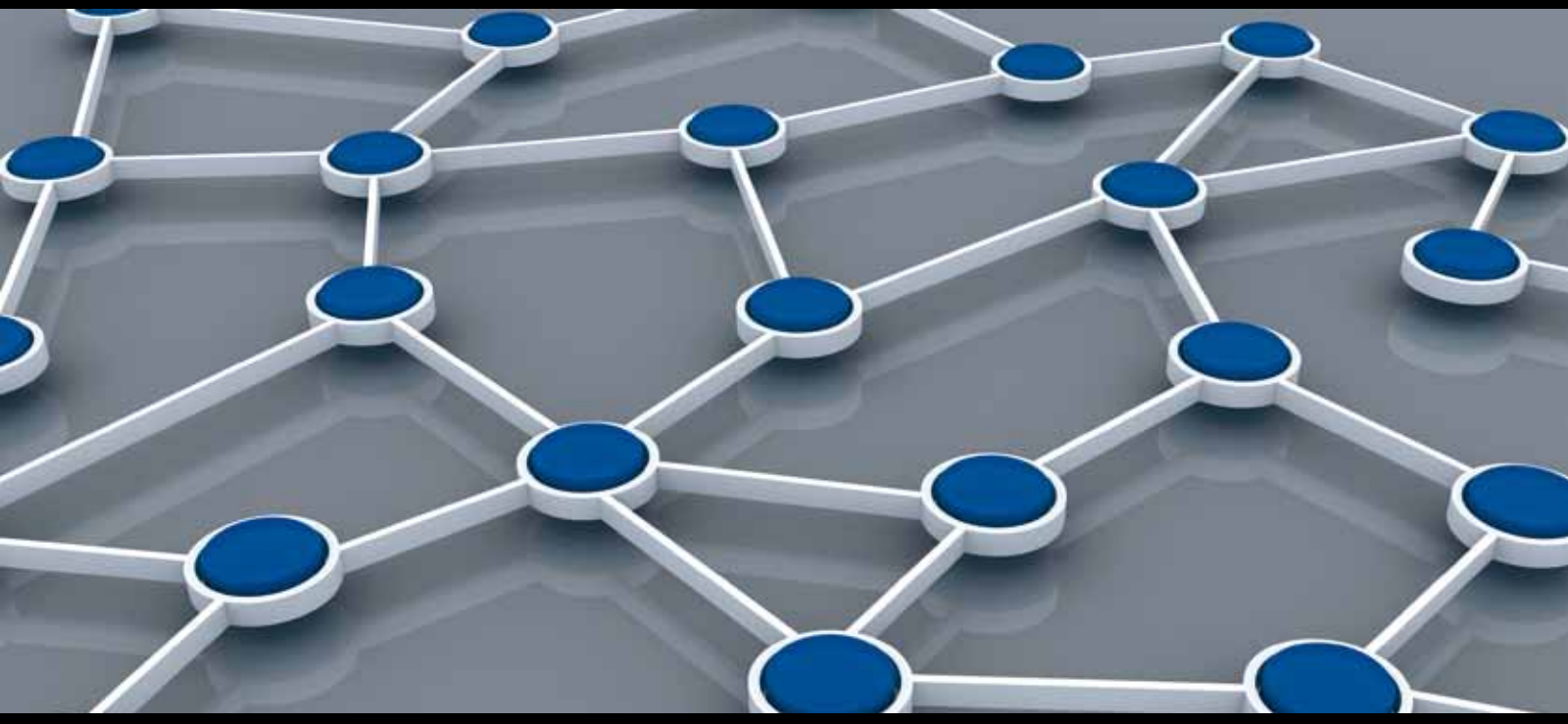
International Journal of Distributed Sensor Networks

Wireless Sensor Network Security

Guest Editors: An Liu, Mihui Kim, Leonardo B. Oliveira, and Hailun Tan



Wireless Sensor Network Security

International Journal of Distributed Sensor Networks

Wireless Sensor Network Security

Guest Editors: An Liu, Mihui Kim, Leonardo B. Oliveira, and Hailun Tan

Editorial Board

Contents

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 362385, 1 page
http://dx.doi.org/10.1155/2013/362385

Editorial
Wireless Sensor Network Security

An Liu,[1] Mihui Kim,[2] Leonardo B. Oliveira,[3] and Hailun Tan[4]

[1] *Intelligent Automation, Inc. Rockville, MP 20855, USA*
[2] *Hankyong National University, Anseong, Republic of Korea*
[3] *UFMG, Pampulha, MG, Brazil*
[4] *University of New South Wales, Kensington, NSW, Australia*

Correspondence should be addressed to An Liu; aliu@i-a-i.com

Received 16 December 2012; Accepted 16 December 2012

Wireless sensor networks consist of a large number of low-cost, low-power, and multifunctional sensor nodes that communicate over short distances through wireless links. Such sensor networks are ideal candidates for a wide range of applications such as monitoring of critical infrastructures, data acquisition in hazardous environments, industry control systems, vehicular networks, and military operations. Wireless sensor networks introduce new security challenges due to their dynamic topology, severe resource constraints, and absence of a trusted infrastructure.

The purpose of this special issue is to publish high-quality research papers as well as review articles addressing recent advances of wireless sensor network security.

In this special issue, we will explore the topic of security challenges and solutions for the wide application of wireless sensor networks in different areas, such as RFID and smart grid. RFID has been widely used in logistics and internet of things. Smart grid leverages sensor networks to better balance the load of the power grid. Both of them have great impact on the daily life and are weak points of the whole economy, which attract the attention of terrorists. The feasible attacks and defense solutions are still not clear. The papers in this volume also cover several important foundational security services for wireless sensor networks, such as key management, key distribution, secure cluster formation, secure architecture for multihop communication, intrusion prediction, and secure localization. All these foundational security services are critical bases for application of wireless sensor networks.

We hope that the papers in this volume engender further thinking about new security challenges for wireless sensor networks and even further "out-of-the-box" ideas about how to solve new security problems for wireless sensor networks.

An Liu
Mihui Kim
Leonardo B. Oliveira
Hailun Tan

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 768489, 17 pages
http://dx.doi.org/10.1155/2013/768489

Research Article

Logic-Based Security Architecture for Systems Providing Multihop Communication

Iman Almomani,[1] Eman Al-Banna,[1] and Mousa AL-Akhras[2]

[1] *Computer Science Department, King Abdullah II School for Information Technology, The University of Jordan, Amman 11942, Jordan*
[2] *Computer Information Systems Department, King Abdullah II School for Information Technology, The University of Jordan, Amman 11942, Jordan*

Correspondence should be addressed to Iman Almomani; i.momani@ju.edu.jo

Received 16 June 2012; Revised 1 September 2012; Accepted 2 September 2012

Academic Editor: Leonardo B. Oliveira

Security is a basic element of distributed systems such as ad hoc and sensor communication networks. Several standards define security requirements and enforcers, such as ITU-T Recommendations X.800 and X.805. It is essential to specify and analyze protocols to know which security requirements they achieve. This paper presents a logic-based security architecture (LBSA). LBSA is a systematic way to test if a protocol is secure by checking what security requirements are achieved. Different rules, actions, and sets which fit into the proposed LBSA are included, new ones are also added to complete the architecture. The key advantage of LBSA is that it enables a security protocol to prove its correctness mathematically. Mathematical proofs provided by LBSA cover more cases that usually cannot be covered exhaustively by simulation tools. This paper also specifies and analyzes several security enforcers and protocols and mathematically proves which security requirements they achieve. Mapping between security requirements and inference rules/actions is also provided to facilitate the use of LBSA. Some enforcers are analyzed using LBSA to demonstrate how they achieve security requirements. Finally, we take Ariadne protocol as a case study and show how to use the proposed LBSA architecture to prove that this protocol is secure.

1. Introduction

When two entities communicate to obtain a certain service(s), they must ensure secure end-to-end communication. Systems do not provide efficient services without applying proper security mechanism due to the existence of different types of attackers. Security can be defined through a set of requirements that must be achieved by the communicating parties to communicate securely and protect services from attackers.

Because of the importance of security for end-to-end communication, many secure protocols have been proposed as will be discussed later in the paper. Some of these protocols had taken prevention measures to stop attackers while others had taken the detection approaches. A way to analyze these protocols is required to check if these protocols are secure as their designers claim and to know which security requirements these protocols achieve. In this paper, we propose a logic-based security architecture (LBSA) which is an easy, fast, and reliable way to specify and analyze secure protocols.

Some security requirements, as defined by several standards such as ITU-T Recommendations X.800 and X.805 [1, 2], must be achieved to declare that a protocol is secure. Using LBSA, a protocol can be tested to check which security requirements it achieves. Several efforts have been done to utilize logic in such test [3–8], but logic was used to specify and analyze specific protocols by defining global and local sets, actions, and inference rules. In this paper security architecture is taken as a whole, different actions, sets, and rules which fit into the architecture have been included, that means specifying and analyzing different enforcers and determining the requirements they satisfy. Also, we had labeled the actions, sets and rules for easy access and we added our own to complete the architecture. Additionally, we mapped different security requirements into inference rules/actions. This mapping shows the inference rules/actions

which are used during the protocol analysis to achieve security requirement(s).

Utilizing LBSA, protocols can be tested to check if they achieve the security requirements specified by their designers. This checking can be performed by analyzing the protocol and applying appropriate actions and rules. If the rules are applied successfully we conclude that their claim is true, otherwise it is false.

The rest of this paper is organized as follows: Section 2 illustrates the security architecture. Section 3 presents related work. Section 4 presents the proposed Logic-Based Security Architecture (LBSA) for systems providing multihop communication. Section 5 provides possible mapping between security requirements and inference rules/actions. In Section 6, the specification and analysis of message authentication code (MAC) and digital certificate based on LBSA are discussed. Section 7 illustrates how to specify and analyze protocols using LBSA by taking Ariadne routing protocol as a case study. Finally, Section 8 concludes the paper and presents avenues for future work.

2. Security Architecture

Security architecture as defined by ITU-T Recommendations X.800 [1] and X.805 [2] defines a set of security requirements that must be achieved. Figure 1 illustrates this architecture.

Security architecture as shown in Figure 1 includes a set of security requirements that are used to protect systems from different types of attackers. To achieve the objectives of these requirements, a set of security enforcers must be applied.

The requirements and their definitions are illustrated as the following.

(i) *Authentication*: Prove the identity of the communicating parties.

(ii) *Authorization*: control the access to the system's resources. Give different entities different privileges according to their roles.

(iii) *Confidentiality*: ensure the secrecy of data. Secret data must be read only by intended recipients.

(iv) *Integrity*: protect the received data from any kind of modifications during transmission from the sender to the receiver.

(v) *Nonrepudiation*: prevent the users from denying the sending of messages or initiating events that they had performed.

(vi) *Privacy*: protect the identity and/or the location of the node and sometimes the routing protocol being used.

(vii) *Availability*: ensure that no one prevents authorized users from getting access to the system available services.

These requirements must be achieved to guarantee secure communication and to prevent and detect different attackers which can be classified into the following.

(i) *Internal or external attackers*: internal attackers are compromised users who are authorized to enter the system but they are misbehaving. This type of attack needs a detection mechanism to be discovered as they have the authority to access the system's services. External attackers, on the other hand, are unauthorized entities attempting to access the system's services; these attackers must be prevented from accessing the system's resources.

(ii) *Passive or active attackers*: passive attackers monitor the system without taking any action and usually it is a phase that precedes an active attacking. Active attacker takes actions and does modifications.

(iii) *Intentional or accidental*: intentional attack is a planned attack while accidental attacking results from system malfunctions such as software bugs.

Table 1: Examples of security requirements enforcers.

Security requirement	Security enforcer
Authentication	(i) Digital certificate
	(ii) Digital signature
	(iii) Message authentication code (MAC)
Authorization	Firewall
Confidentiality	Encryption
Integrity	(i) Hash function
	(ii) MAC
	(iii) Digital signature
Nonrepudiation	Digital signature
Privacy	(i) Achieved partially by encryption
	(ii) Hide node location
	(iii) Hide node identity
	(iv) Hide routing protocol
Availability	(i) Intrusion detection scheme (IDS)
	(ii) Intrusion prevention scheme (IPS)

Several mechanisms and techniques have been defined to achieve different security requirements. Table 1 gives examples of different security enforces that are used to implement the objectives of different security requirements.

Secure protocols usually use the above enforces, among others, to achieve the security requirements. There are different ways to analyze security protocols in terms of achieved security requirements such as simulation and mathematical models; this paper uses logic to do such analysis.

3. Related Work

The importance of security in providing successful services in any distributed system raises the necessity of having formal way to analyze security protocols. Previous effort in using logic for analyzing security is Rubin logic [3, 4].

Rubin logic is an approach that specifies and analyzes nonmonotonic cryptographic protocols. It is one of the first approaches to allow reasoning about nonmonotonic protocols [3]. In nonmonotonic protocols, beliefs are changed

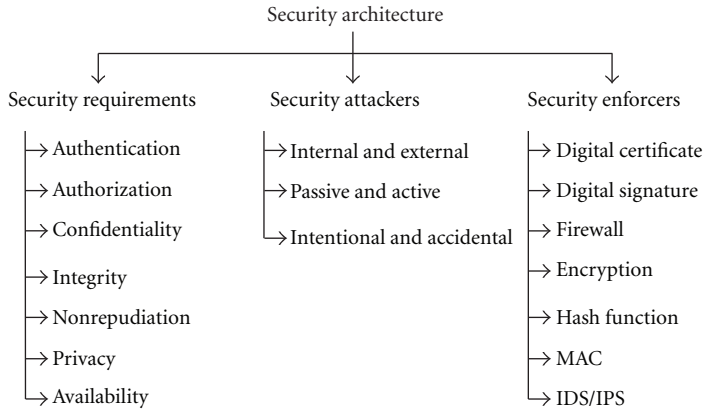


Figure 1: Security Architecture extracted from ITU-T Recommendations X.800 and X.805.

during protocol execution time. An example of nonmonotonicity is the belief that a key must be changed when a node becomes compromised. To achieve the protocol specification and analysis, Rubin logic defines global and local sets, actions and inference rules.

Rubin and Honeyman [3] focused on authentication protocols. They took KHAT protocol [9] as an example to discover its flaws. KHAT protocol was built to solve the problem of long running jobs in an authenticated environment where a trusted server issues tickets with limited lifetimes for services. The authors gave special attention to ensure the freshness of data using fresh nonce. The main problem they attempted to solve is that principal B cannot achieve the belief that the session key with principal A is fresh. Finally, the authors defined most of global and local sets that are used later in the literature.

Rubin [4] extended the work presented in [3] by adding one set. Rubin [4] aimed to make sure that keys are observed by their intended parties and data items are fresh, especially the public keys. Rubin [4] made link between certificates and requests which reveals weakness in Needham and Schroeder public key protocol [10].

Xu and Xie presented in a series of papers [5–8] the utilization of Rubin logic in analyzing the security for specific protocols.

In [5], Xu and Xie extended the work presented in [4] to analyze nonrepudiation in routing protocols proposed for wireless mobile ad hoc networks (MANET). This work took ARAN [11] routing protocol to test nonrepudiation.

Xu and Xie [6] use the work presented in [5] to analyze electronic commerce protocols and in [7] they have chosen Zhou-Gollmann [12] protocol which is a simple and effective nonrepudiation protocol to illustrate how an electronic commerce protocol is analyzed using the extended Rubin logic.

Two examples of Rubin logic's applications are given by Xu and Xie in [8]. First example is the Andrew secure RPC [13] protocol using symmetric keys. The second one is X.509 [14] authentication protocol using asymmetric keys.

As can be illustrated from the above-related work, all attempts to utilize Rubin logic have either focused on a specific requirement or a specific protocol. This paper proposes a logic-based security architecture (LBSA) that presents a formal way to analyze any security requirement in any system providing multihop communication. All sets, actions, and rules presented in previous efforts are considered and generalized; new ones are added to complete the architecture. After that, we illustrate how LBSA will be used to test security requirements and issues in different security enforcers and protocols.

4. Logic-Based Security Architecture (LBSA) for Systems Providing Multihop Communication

In this section we illustrate the proposed architecture which defines a logical way for specifying and analyzing different enforcers and different protocols that achieve any security requirement as defined in [2]. Some researchers used logic as mentioned earlier in Section 3. All sets, actions, and rules defined in [3–8] which fit in our architecture have been included and labeled by the proposed LBSA.

Global sets define the specification of the protocol as a whole. Local sets are private to each principal in the specification. Actions are specified as part of the protocol (i.e., how the protocol works) while inference rules are used to reason about the beliefs during the protocol executing. Consequently, the relation between sets, actions, and rules and the result of the action may update the sets. This achieves some rules and conditions which are followed by applying inference rules which in turn update another set(s). Figure 2 shows this relation.

Using different actions, the protocol can be specified exactly as it works. While executing these actions, local and global sets will have new values which lead to applying some rules. This is the process of analyzing protocols.

The purpose of LBSA is to generalize rules and actions and not to specify them. Accordingly, these rules and actions can be customized according to the context. Sections 6 and 7 illustrate how these sets, rules, and actions can be customized according to the context. Additionally, each set, rule, and action are described in the following tables using the description column.

Table 2 defines different global and local sets. Global sets are labeled as GS_i and local sets are labeled as LS_i, where i is

Table 2: Global and local sets of LBSA.

Set label	Set name	Description
GS_1	Principal set	All principals (i.e., nodes) participate in the protocol are in this set. $P = \{P_1, P_2, \dots, P_n\}$. There is one initiator of the protocol and it could be any P_i.
GS_2	Rule set	Inference rules needed to derive new statements from existed assertions that are in this set. $R = \{R_1, R_2, \dots, R_n\}$, where R_i is of the form $(C_1, C_2, \dots, C_n)/D$ C_i is a condition and D is a statement.
GS_3	Secret set	Each secret in the system exist in this set. During the analysis, the content of this set will be changed for the emergence of new secrets such as session keys. $S = \{S_1, S_2, \dots, S_n\}$.
GS_4	Observers set	For each S_i, observers (S_i) contain all the principals who could possibly know the secret S_i by listening to network traffic or generating it themselves. The members of the Observers set can be stated explicitly or maintained as formulas representing their membership.
GS_5	FireWall set(P_i)	This set presents the firewall list that contains the access control rules that are needed to be applied to the incoming packets of P_i in order to allow or prevent them.
LS_1	Possession set(P_i)	This set contains all the data relevant to security that this principal knows or possesses. This includes secret encryption keys, public keys, data that must remain secret, and any other information that is not publicly available. $\text{POSS}(P_i) = \{\text{poss}_1, \text{poss}_2, \dots, \text{poss}_n\}$. Note: poss_i contains two fields—the actual data and the origin of this data.
LS_2	Belief set(P_i)	This set contains all the beliefs held by a principal. This includes the belief that the keys it holds between itself and other principals are good, beliefs about jurisdiction, beliefs about freshness, and beliefs about the possessions of other principals. $\text{BEL}(P_i) = \{\text{bel}_1, \text{bel}_2, \dots, \text{bel}_n\}$.
LS_3	Opaque(P_i)	This set contains candidates to be added to the Seen set. It is used by the Update function (Table 3(b)). The set contains text parts (data) of the message and a list of the associated keys needed to read them.
LS_4	Seen(P_i)	This set contains text parts (data) that P_i sees from messages sent across the network. The Seen sets collectively contain the same information as the observers set.
LS_5	Haskeys(P_i)	This set contains keys that P_i sees either because they are in the initial possession set, or because they appear in a message sent across the network and are added to P_i's seen set.
LS_6	Behavior list(P_i)	This item is a list rather than a set because the elements are ordered. $\text{BL} = \{\text{AL}, \text{bvr}_1, \text{bvr}_2, \dots, \text{bvr}_n\}$. AL is an action list as will be defined below.
LS_7	Bindings set(P_i)	This set contains the legal bindings of keys held by a principal. These are bindings that are created by P_i, and bindings that are received in certificates from trusted servers. Bindings $(P_i) = \{k_1 \bowtie P_1, k_2 \bowtie P_2, \dots, k_n \bowtie P_n\}$.
LS_8	Proofs set(P_i)	This set contains all the other principals' nonrepudiation that P_i can prove.
LS_9	PL(P_i)	Principals list contains all the principals that must receive Messages (Msgs) from principals P_i.
LS_{10}	AbnBeh(P_i)	Abnormal behavior set contains all the principals that P_i detect their abnormal behaviors.

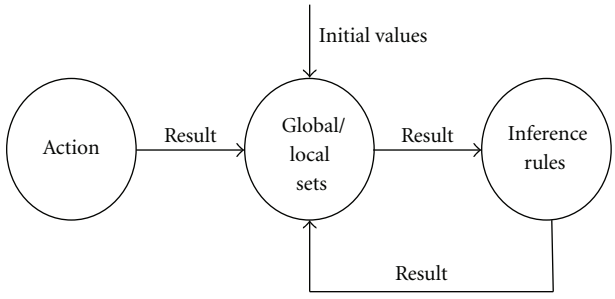


Figure 2: Relation between sets, actions and inference rules.

the set number. These sets contain the protocol information. There are five global sets GS_1–GS_5 and ten local sets LS_1–LS_{10}. These sets have initial values and these values are changed whenever actions and rules are executed. Global sets (GS_1–GS_4) were initilly defined in [3] for a specific protocol called KHAT [9]. These global sets have been generalized by LBSA and new global sets are also added. Local sets (LS_1–LS_8) were initially defined in [3–5] for specific protocols, but they are generalized and labeled by LBSA which also defines new local sets needed to complete the architecture.

Tables 3(a)–3(c) define the LBSA actions with their descriptions, conditions, and results. Actions are labeled as ACT_i, where i is the action number and these tables contain 30 actions. The actions describe the node operations such as sending messages, receiving messages, generating key pairs among other operations. Actions ($AC\,T_1$–$AC\,T_{21}$) that are considered in [3–6] had been generalized and fitted in the proposed LBSA. Moreover, new actions are defined and added by LBSA.

Table 3: (a) Actions (ACT_1–ACT_{10}) of LBSA, (b) actions (ACT_{11}–ACT_{20}) of LBSA, and (c) actions (ACT_{21}–ACT_{30}) of LBSA.

(a)

Action label	Action name	Description	Condition	Result
ACT_1	Send(P_j, X)	P_i sends msg X to P_j.	—	Principal P_i sends msg X to P_j.
ACT_2	Receive(P_j, X)	P_i receives msg X from P_j.	—	Principal P_i receives msg X from P_j.
ACT_3	Encrypt(X, k)	If P_i possesses msg X and knows the key k then it can encrypt X using k and possess $\{X\}_k$.	$X, k \in \text{POSS}(P_i)$, $P_i \in$ Observers (k).	$\text{POSS}(P_i) = \text{POSS}(P_i) \cup \{\{X\}_k\}$.
ACT_4	Decrypt($\{X\}_k$, k)	If P_i possesses msg X which encrypted under k and knows the key k (symmetric encryption) or its inverse (Asymmetric encryption), then P_i can decrypt X and possess it.	$P_i \in$ Observers(k), $\{X\}_k$, $k \in \text{POSS}(P_i)$.	$\text{POSS}(P_i) = \text{POSS}(P_i) \cup \{X\}$.
ACT_5	Generate-nonce(N)	Is used to check out the freshness. A principal generates a nonce to link a challenge and a response. When the response is received, LINK(N) is removed from BEL(P_i).	—	$\text{POSS}(P_i) = \text{POSS}(P_i) \cup \{N\}$, $\text{BEL}(P_i) = \text{BEL}(P_i) \cup \text{LINK}(N)$.
ACT_6	Generate-secret(s)	Is used to generate a new secret s. Note that the Observers set will be updated.	—	$S := S \cup \{s\}$, Observers(s) = $\{P_i\}$, $\text{POSS}(P_i) = \text{POSS}(P_i) \bowtie \{s \bowtie P_i\}$, $\text{BEL}(P_i) = \text{BEL}(P_i) \cup \#(s)$, *where # indicates that the value is fresh.*
ACT_7	Concat($x_1, x_2, \ldots, x_n$)	Is used to construct a message X out of submessages $x_1, x_2, \ldots, x_n$.	$x_1, x_2, \ldots, x_n \in \text{POSS}(P_i)$.	$\text{POSS}(P_i) = \text{POSS}(P_i) \cup \{x_1, x_2, \ldots, x_n\}$.
ACT_8	Split(X)	When a message contains a set of components, this action is used to break it into its components.	X contains $x_1, x_2, \ldots, x_n$, $X \in \text{POSS}(P_i)$.	$\text{POSS}(P_i) = \text{POSS}(P_i) \cup \{x_1, x_2, \ldots, x_n\}$.
ACT_9	Forget(X)	When P_i no longer in possession of X, this action will be used. If other principles trust P_i then they can believe that P_i is no longer possesses X.	$X \in \text{POSS}(P_i)$.	$\text{POSS}(P_i) = \text{POSS}(P_i) - \{X\}$, $\forall P_j \in P$ if $\text{TRUST}[j, i] = 1$. then $\text{BEL}(P_j) = \text{BEL}(P_j) - X \in \text{POSS}(P_i)\}$.
ACT_{10}	Check-freshness(X)	Is used to verify the freshness of timestamp X.	$X \in \text{POSS}(P_i)$, X is not expired.	$\text{BEL}(P_i) = \text{BEL}(P_i) \cup \{\#(X)\}$.

(b)

Action label	Action name	Description	Condition	Result
ACT_{11}	Forget-secret(s)	When P_i no longer knows the secret s. If other principals trust P_i then they can believe that P_i no longer possesses s.	$P_i \in$ Observers (s), $s \in \text{POSS}(P_i)$.	Observers(s) = Observers(s) $- \{P_i\}$, $\text{POSS}(P_i) = \text{POSS}(P_i) - \{s\}$, $\forall P_j \in P$ if $\text{TRUST}[j, i] = 1$ then $\text{BEL}(P_j) = \text{BEL}(P_j) - \{s \in \text{POSS}(P_i)\}$
ACT_{12}	Apply(f, X)	Is used to apply function f to X.	$f, X \in \text{POSS}(P_i)$.	$\text{POSS}(P_i) := \text{POSS}(P_i) \cup \{f(X)\}$.
ACT_{13}	Abort	The protocol aborts when different cracks and events that affect the protocol specification happen.	Protocol run is illegal.	Analysis reports failure.
ACT_{14}	Generate-key-pair(k^+, k^-)	Is used to generate two keys k^+ (public) and its inverse k^- (private).	—	$\text{POSS}(P_i) = \text{POSS}(P_i) \cup \{k^+ \bowtie P_i, k^- \bowtie P_i\}$.

(b) Continued.

Action label	Action name	Description	Condition	Result
ACT_{15}	Apply-Asymkey(X, k)	There are two cases for applying asymmetric key operation on X. The first one when X is in the form of $\{Y\}_{k'}$ and k and $k^{'}$ are inverses to each other. The second case when X is encrypted using k.	$X, k \in \text{POSS}(P_i)$, $P_i \in \text{Observers}(k)$, k is an asymmetric key.	if $X = \{Y\}_{k'}$ and k is the inverse of $k^{'}$, then $\text{POSS}(P_i) = \text{POSS}(P_i) \cup Y$ else $\text{POSS}(P_i) = \text{POSS}(P_i) \cup \{\{X\}_k\}$.
ACT_{16}	Bind(k, P_j)	Before sending a key, principal P_i can bind this key to other principle P_j using this action. After this binding, both P_i's possession and Binding sets will contain $k \bowtie P_j$.	$k \in \text{POSS}(P_i)$, $P_i \in \text{Observers}(k)$, k is a key intended for P_j.	$\text{POSS}(P_i) = \text{POSS}(P_i) \cup \{(k \bowtie P_j)\}$, $\text{Bindings}(P_i) = \text{Bindings}(P_i) \cup \{(k \bowtie P_j)\}$.
ACT_{17}	Update(X)	After sending a message this action is used to maintain the observers set.	—	Maintains the observers of X.
ACT_{18}	Broadcast(X)	Message X is broadcasted by P_i to each of its neighbors.	—	P_i broadcasts X to each of its neighbors.
ACT_{19}	Check-certificate(cert_R)	Is used to verify the freshness of the certificate cert_R.	$\text{cert}_R \in \text{POSS}(P_i)$, cert_R is not expired	$\text{BEL}(P_i) = \text{BEL}(P_i) \cup \{\#(K_R)\}$, *where* K_R *is the public key specified in* cert_R.
ACT_{20}	Check-request($\{\text{IP}_R, N_R\}$)	Is used to record new routing record. Each node receives route discovery packet it records the predecessor of that packet and the successor toward the destination.	$\text{IP}_R \in \text{POSS}(P_i)$, $N_R \in \text{POSS}(P_i)$, and $\{\text{IP}_R, N_R\}$ is not already seen.	{new routing record} $\in \text{POSS}(P_i)$ *Where*: IP_R *is* IP *address of principle* R and N_R *is nonce of principle* R.

(c)

Action label	Action name	Description	Condition	Result
ACT_{21}	Comp(A, B)	This action is used to compare two values if they are equal or not.	A, B $\in \text{POSS}(P_i)$.	If A and B are equal: $\{X\}^* \in \text{Proof}(P_i)$, *means message X is checked and not modified* If they are not equal: $\text{POSS}(P_i) = \text{POSS}(P_i) - \{X\}$ *means message X is checked and it is modified*
ACT_{22}	Renew-key-pair(k^+, k^-)	This action is used to renew the public and private keys.	—	POSS = $(\text{POSS}(P_i) - \{k^+ \bowtie P_i, k^- \bowtie P_i\}) \cup \{\#k^+ \bowtie P_i, \#k^- \bowtie P_i\}$.
ACT_{23}	Multicast($\forall P_i \in \text{PL}, X$)	It means that Principle P_i sends msg X to all principles in the PL.	—	Principle P_i sends message X to all principles in the PL.
ACT_{24}	Modify(Cert_R)	This action modifies certificate content.	—	Modify the certificate content.
ACT_{25}	Monitor (P_i, abnormal behavior)	It means that principle P_i monitors the abnormal behavior of other principals.	—	Principal P_i detects abnormal behaviors of other principal $P_i, P_j \in \text{AbnBeh}(P_i)$.
ACT_{26}	Generate-Alert(P_i, P_j)	P_i generates an alert at time t because it detects abnormal behavior of P_j.	$P_j \in$ A bnBeh (P_i).	$\text{Alert}(t, P_i, P_j)$
ACT_{27}	Alert-Filtering ($\text{Alert}(t, P_i, P_j)$)	This action is used to do the process of categorizing attack alerts produced from principals in order to distinguish false positive from actual attack.	$\text{Alert}(t, P_i, P_j) \in \text{POSS}(P_n)$, *where* P_n *is any Principle* $\in P$, *but not* P_i*or* P_j.	If true positive P_n takes an action.
ACT_{28}	Access (Resource)	This action is used to allow or deny the access of resources.	$\text{AccessInfo}(P_i) \in \text{FireWall}$, Condition = Allow	Allow resource access.
ACT_{29}	$X \leftarrow Y$	This action is used to rename value Y to X.	$Y \in \text{POSS}(P_i)$	$X \in \text{POSS}(P_i)$, Use X instead of Y.
ACT_{30}	Process (Pra, Fun)	This action is used in key agreement process. Process parameters received from P_j.	Fun $\in \text{POSS}(P_i)$, parameters received from P_j.	Key $\in \text{POSS}(P_i)$

Different inference rules are defined in Tables 4(a)–4(d). RL_i is the rule label where i is the rule number. As the rule conditions are executed correctly, the sets' values will be changed or the protocol will be aborted. Rules defined for specific protocols in [3–6] had been extracted, generalized, and mapped to LBSA. New rules were also generated by LBSA to complete the security architecture.

Section 5 provides mapping between security requirements and inference rules/actions. It also provides mapping between some protocol services and inference rules/actions.

5. Possible Mapping between Security Requirements and Inference Rules/Actions

To ensure that the required security requirement is achieved by a security protocol, some inference rules must be applied and some actions must be performed. In this section we provide possible mappings between each security requirement and the inference rule that must be applied or the action that must be executed to achieve it. Table 5 summarizes these mappings.

Actions and rules are needed to complete the specification and the analysis processes. In other words, actions and rules add new values to the global and local sets which may result in achieving the mapped rules conditions.

Other actions and rules are needed to perform certain protocol services, including key management which deals with key and certificate generations, verification, and renewing. Certain services are related to nodes' interaction and protocol functions. Data item refreshments such as messages, keys, certificates, and nonces are among the data that need to be kept fresh.

Table 6 summarizes the mappings between protocol services and the inference rule that must be applied or the action that must be executed to achieve it.

The following section presents how the above-presented architecture will be used to formally analyze security enforcers and security protocols.

6. Security Enforcers Specification and Analysis

Different security enforcers are used as part of different protocols to achieve security such as in [15–18]. In this section we have chosen well-known enforcers that can be provided at different network layers to illustrate how LBSA could be used to specify and analyze the security enforcers that could be used by many different security protocols. These enforcers are message authentication code (MAC) and digital certificate which also includes digital signature. The reason for using these enforcers is that they are the most used enforcers, in many protocols including routing protocols, to achieve authentication and integrity (MAC) and authentication, integrity and nonrepudiation (digital certificate) [19–25]. The following sections show how these enforcers could be verified using LBSA.

6.1. Message Authentication Code (MAC). MAC [26] achieves two security requirements, namely, authentication and integrity. In MAC the sender and the receiver of the message must agree on a shared key before initiating the communication. The sender of the message uses the shared key and the message as an input to a hash function to generate the message digest. Then, the sender sends the message and the digest to the receiver. The receiver checks the originality of the message and its correctness by applying the hash function on the received message and the shared key. After that, the receiver compares the resulted digest with the received message digest. If the two digests are equal, this will prove the authenticity of the message as the shared key is only known by sender and receiver. Moreover, it ensures that the message is not modified during the transmission since any simple modification will produce a different digest. To mathematically prove that MAC achieves both authentication and integrity using LBSA the following steps must be performed.

(i) Define the Values of the Global Sets. The first step in the protocol specification is to define the values of the global sets:

(GS1) P = {Alice, Bob} // *Alice is assumed to be the initiator of the protocol.*

(GS2) R = {contains the rules defined in the Tables 4(a)–4(d)}

(GS3) $S = \{K_{AB}\}$ // K_{AB} *is the shared key between Alice and Bob*

(GS4) Observers (K_{AB}) = {Alice, Bob}.

(ii) Define the Initial Values of the Local Sets. The second step is to define the initial values of the local sets to each principal. Note that some of the global and the local sets values will be changed during the protocol analysis.

Principal A (Alice):

(LS1) POSS(A) = {Msg, K_{AB}, (B $\bowtie$ K_{AB}) from B}

(LS2) BEL(A) = {#(K_{AB}), (B $\bowtie$ K_{AB})}

(LS6) BL =

(ACT_7) Concat(Msg, K_{AB})

(ACT_{12}) Apply(H, {Msg.K_{AB}}) // *Apply hash function to the concatenation*

(ACT_7) Concat(Msg, H(Msg.K_{AB})) // *prepare the message and digest*

(ACT_1) Send(B, {Msg.H(Msg.K_{AB})}) // *send message and digest to* B.

Principal B (Bob):

(LS1) POSS(B) = {K_{AB}, (A $\bowtie$ K_{AB}) from A}

(LS2) BEL(B) = {#(K_{AB}), (A $\bowtie$ K_{AB})}

(LS6) BL =

(ACT_2) Receive(A, {Msg.H(Msg.K_{AB})})

(ACT_8) Split(Msg.H(Msg.K_{AB}))

Table 4: (a) Rules (RL_1–RL_6) of LBSA, (b) rules (RL_7–RL_{11}) of LBSA, (c) rules (RL_{12}–RL_{16}) of LBSA, and (d) rules (RL_{17}–RL_{19}) of LBSA.

(a)

Rule label	Rule	Description
RL_1	Nonce verification rule: $\frac{\#(X) \in \text{BEL}(P), X \text{ from } Q \in \text{POSS}(P)}{\text{BEL}(P) = \text{BEL}(P) \cup \{Q \text{ believes } \#(X)\}}$	Principle P can decide that principle Q believes that X is fresh if P's Belief set contains $\#(X)$ and X from Q is in the P's possession set.
RL_2	Message meaning rule: $\frac{\{X\}_k \text{ from } Q \in \text{POSS}(P), \{P, Q\} \subseteq \text{Observers}(k)}{\text{BEL}(P) = \text{BEL}(P) \cup \{X \in \text{POSS}(Q)\}}$	P can believe that Q possesses X if it receives $\{X\}_k$ from Q and both P and Q know the key k.
RL_3	Sub-message freshness rule: $\frac{\#(x_1) \in \text{BEL}(P), \{X \text{ contains } x_1, X \text{ contains } x_2\} \subseteq \text{POSS}(P)}{\text{BEL}(P) = \text{BEL}(P) \cup \#(x_2)}$	P can believe that x_2 is fresh if it believes that x_1 is fresh and have x_1 and x_2 in the same formula and this formula is in P's Possession set. That is, the part of the message that contains something fresh is fresh.
RL_4	Linkage rule (symmetric keys): $\frac{\#(k) \in \text{BEL}(P), P \in \text{Observers}(k), \text{LINK}(Na) \in \text{BEL}(P), X \text{ contains } f(Na), X \text{ contains } x_1, \{X\}_k \text{ from } Q \in \text{POSS}(P)}{\text{BEL}(P) = (\text{BEL}(P) - \text{LINK}(Na)) \cup \{\#(x_1)\}}$	This rule is applied only once for the same nonce since the LINK item will be removed after applying it successfully. P_i can believe that a submessage x_1 of a message X is fresh when several conditions are applied. The first one is when P's Belief set contains LINK (Na) which means the nonce Na has not been used before. Also, the message X that contains the nonce must be encrypted using the key k and k must be fresh and available to P; this is what other conditions state.
RL_5	Linkage rules (asymmetric keys): (a) $\frac{\#(k^-) \in \text{BEL}(P), P \in \text{Observers}(k^-), \text{LINK}(Na) \in \text{BEL}(P), X \text{ contains } f(Na), X \text{ contains } x_1, \{X\}_k^+ \text{ from } Q \in \text{POSS}(P)}{\text{BEL}(P) = (\text{BEL}(P) - \text{LINK}(Na)) \cup \{\#(x_1)\}}$ (b) $\frac{\#(k^+) \in \text{BEL}(P), P \in \text{Observers}(k^+), \text{LINK}(Na) \in \text{BEL}(P), X \text{ contains } f(Na), X \text{ contains } x_1, \{X\}_k^- \text{ from } Q \in \text{POSS}(P)}{\text{BEL}(P) = (\text{BEL}(P) - \text{LINK}(Na)) \cup \{\#(x_1)\}}$	This rule is applied only once for the same nonce since the LINK item will be removed after applying it successfully. P_i can believe that a submessage x_1 of a message X is fresh when several conditions are applied. The first one is when P's Belief set contains LINK (Na) which means the nonce Na has not been used before. Also, the message X that contains the nonce must be encrypted using the public key k^+ and the corresponding inverse (private) key k^- must be fresh and available to P; this is what other conditions state.
RL_6	Possible origins rule: $\frac{X \in \text{POSS}(P), X \text{ contains } x_1, R \in \text{Observers}(x_1), R \neq P}{x_1 \text{ from } R \in \text{POSS}(P)}$	When P's possession set contains message X which contains submessage x_1 and this submessage observed by more than one principal, other than P, it will be marked as coming from all principals by adding new item for each of the P's Possession set.

(b)

Rule label	Rule	Description
RL_7	Submessage origin rule: $\frac{\{X\}_k^+ \in \text{POSS}(P), X \text{contains } x_1 \text{ from } Q, R \in \text{Observers}(k^-), X \text{ contains } x_2, R \neq P}{x_2 \text{ from } Q \in \text{POSS}(P), x_2 \text{ from } R \in \text{POSS}(P)}$	A submessage x_2 can be marked as coming from Q if P possesses a message X which contains x_1 and x_2 encrypted using an asymmetric key, and submessage x_1 is marked as coming from Q. And any other principle who observes the inverse key, the submessage is marked as coming from it.
RL_8	Submessage origin rule for asymmetric keys: (a) $\frac{\{X\}_{kp}^+ \in \text{POSS}(P), X \text{ contains } x_1 \text{ from } Q, X \text{ contains } x_2}{x_2 \text{ from } Q \in \text{POSS}(P)}$ (b) Submessage origin rule for private keys: $\frac{\{X\}_{kQ}^- \in \text{POSS}(P), X \text{ contains } x_2}{x_2 \text{ from } Q \in \text{POSS}(P)}$	This rule is simplified version of the *submessage origin rule*. In (a) If P's Possession set contains message X which is encrypted using P's public key. And X contains a submessage x_1 which is marked as coming from Q. Then all other submessages in X are marked as coming from Q. This is true because P is the only principle that has the corresponding private key so no one can change the content of X. In (b) since Q is the constructor of the message X, any submessage of X can be marked as coming from Q. This is true because Q is the only principle that has Q's private key.

(b) Continued.

Rule label	Rule	Description
RL_9	Unbound key rule: $\dfrac{k \in \text{POSS}(P), \nexists Q : (k \bowtie Q) \in \text{POSS}(P)}{\text{Observers}(k) = W}$	When key k is in P's Possession set and this key is not bounded to any principal then all principals (W) observe it. That is, the key is not a secret.
RL_{10}	Bound key origin rule for symmetric keys: $\dfrac{(k \bowtie Q) \in \text{POSS}(P), Q \neq P, (k \bowtie Q) \notin \text{Bindings}(P)}{\text{Abort}}$	The protocol abort when P's Possession set contains the key k which is bound to principle Q, and $Q \neq P$ but this binding is not in P's bindings set which contains legal bindings for P.
RL_{11}	Bound key origin rules for asymmetric keys: (a) $\dfrac{(k^+ \bowtie Q) \in \text{POSS}(P), (k^+ \bowtie Q) \notin \text{Bindings}(P) \{X\}_k^- \text{ from } R \in \text{POSS}(P), R \neq Q}{\text{Abort}}$ (b) $\dfrac{(k^- \bowtie Q) \in \text{POSS}(P), (k^- \bowtie Q) \notin \text{Bindings}(P), \{X\}_k^+ \text{ from } R \in \text{POSS}(P_i), R \neq Q}{\text{Abort}}$	If principle P has the binding of an asymmetric key to principle Q in its possession and binding sets and P's possession set contains a message that is encrypted under the inverse key this means that the message comes from Q because it is the only principle that have the inverse key. Otherwise, the protocol aborts.

(c)

Rule label	Rule	Description
RL_{12}	Conjunction rule: (a) $\dfrac{(Q \rightarrow (X,Y)) \in \text{Proofs}(P)}{(Q \rightarrow X) \in \text{Proofs}(P_i); (Q \rightarrow Y) \in \text{Proofs}(P)}$ (b) $\dfrac{(Q \rightarrow X) \in \text{Proofs}(P); (Q \rightarrow Y) \in \text{Proofs}(P)}{(Q \rightarrow (X,Y)) \in \text{Proofs}(P)}$	When P's Proof set contains the proof that Q is accountable for the conjunction X and Y, then it will contain the proof that Q is accountable for X and Y, respectively. Contrariwise, the second one is used.
RL_{13}	Ciphertext understanding rule: $\dfrac{(Q \rightarrow \{X\}_k) \in \text{Proofs}(P), (k \in Q) \in \text{Proofs}(P)}{(Q \rightarrow X) \in \text{Proofs}(P)}$	When P's Proof set contains the prove that Q is accountable for $\{X\}_k$ and Q holds k then it will contain the prove that Q is accountable for message X.
RL_{14}	Transfer rule: $\dfrac{(\text{TTP} \rightarrow X) \in \text{Proofs}(P)}{(X \in Q) \in \text{Proofs}(P)}$	When P's Proof set contains the proof that Trusted Third Party (TTP) is accountable for message X, then it will contain the proof that Q holds message X.
RL_{15}	Timestamp rule (symmetric keys): $\dfrac{\#(K) \in \text{BEL}(P), K \in \text{POSS}(P), \#(T) \in \text{BEL}(P), X \text{ contains } x_1, \{X\}_K \text{ from } Q \in \text{POSS}(P)}{\text{BEL}(P) = \text{BEL}(P) \cup \{\#(x_1)\}}$	Timestamp rule defined to reason about the message.
RL_{16}	Timestamp rule (asymmetric keys): $\dfrac{\#(K_Q^+) \in \text{BEL}(P), K_Q^+ \in \text{POSS}(P), \#(T) \in \text{BEL}(P), X \text{ contains } x_1, \{X\}_{kQ}^- \text{from } Q \in \text{POSS}(P)}{\text{BEL}(P) = \text{BEL}(P) \cup \{\#(x_1)\}}$	Timestamp rule defined to reason about the message.

(d)

Rule Label	Rule	Description
RL_{17}	Authentication Rule: $\dfrac{(K_Q^+ \bowtie Q) \text{ from } Z \in \text{POSS}(P), (K_Q^+ \bowtie Q) \in \text{BEL}(P), \{X\}^* \in \text{Proofs}(P)}{\text{Bindings}(P) = \text{Bindings}(P) \cup \{(K_Q^+ \bowtie Q)\}}$	This rule adds a new binding to the binding set. Note that its condition will not complete until the comparison action (ACT_{21}) is executed successfully. Z could be any principle except P or TTP.

(d) Continued.

Rule Label	Rule	Description
RL_{18}	Integrity Rule: $\dfrac{\{X\}^* \in \text{Proof}(P)}{\text{BEL}(P) = \text{BEL}(P) \cup \{X \in \text{Poss}(Q)\}}$	This rule states that if $\{X\}^* \in \text{Proof}(P_i)$ which means that X received without any modification (i.e., as it sent from Q) then we add $X \in \text{Poss}(Q)$ to the believe set.
RL_{19}	Non-Repudiation Rule: $\dfrac{\{X\}^-_{kQ} \in \text{Poss}(P),\ \left(K^+_Q \bowtie Q\right) \in \text{Binding}(P),}{(Q \rightarrow X) \in \text{Proof}(P)}$	This rule states that if we have msg encrypted by K^-_Q and the corresponding public key K^+_Q in the Binding (P), we can prove that Q is responsible for sending message X.

Table 5: Mappings between security requirements and Inference rules/actions.

Security Requirement	Inference Rules/Actions
Authentication	RL_2, RL_6, RL_7, RL_8, RL_{13}, RL_{17}
Authorization	ACT_{28}
Confidentiality	ACT_3, ACT_4, ACT_{15}
Integrity	RL_{18}
Non-Repudiation	RL_2, RL_6, RL_7, RL_8, RL_{13}, RL_{19}
Privacy	ACT_3, ACT_{15}
Availability	ACT_{13}, ACT_{25}, ACT_{26}, ACT_{27}, ACT_{28}

(ACT_7) Concat(Msg, K_{AB}) // *concatenate the received message with the key*

(ACT_{12}) Apply(H, {Msg.K_{AB}})

(ACT_{21}) Comp(H(Msg.K_{AB}), H(Msg.K_{AB})) // *compare the calculated digest with the received one*.

(iii) MAC Analysis. The last step is to analyze the security enforcer which is MAC in this case to check security requirements it achieves.

Alice is assumed to be the initiator of the protocol. Therefore, four actions in the BL(A) are executed which add new values to the POSS(A) set:

$$\text{POSS}(\text{A}) = \{\text{Msg}, K_{AB}, (\text{B} \bowtie K_{AB}) \text{ from B}, \{\text{Msg}.K_{AB}\}, H(\text{Msg}.K_{AB})\}. \quad (1)$$

So far no inference rules can be applied. Now, actions in BL(B) are executed and this will produce new values that are added to POSS(B) and the Proof(B) sets. Consequently, both integrity and authentication inference rules can be applied:

$$\text{POSS}(\text{B}) = \{K_{AB}, (\text{A} \bowtie K_{AB}) \text{ from A}, \text{Msg}, H(\text{Msg}.K_{AB}), \{\text{Msg}.K_{AB}\}, H(\text{Msg}.K_{AB})\}$$
$$\text{Proof}(\text{B}) = \{\{\text{Msg}\}^*\}. \quad (2)$$

Since {Msg}* ∈ Proof(B), we can apply the authentication rule (*RL*17) and the integrity rule (*RL*18) as follows.

Authentication Rule (RL17):

$$\frac{(\text{A} \bowtie K_{AB}) \text{ from A} \in \text{POSS}(\text{B})}{\text{Binding}(\text{B}) = \text{Binding}(\text{B}) \cup \{(\text{A} \bowtie K_{AB})\}}$$
$$\frac{(\text{A} \bowtie K_{AB}) \in \text{BEL}(\text{B}), \{\text{Msg}\}^* \in \text{Proof}(\text{B})}{\text{Binding}(\text{B}) = \text{Binding}(B) \cup \{(\text{A} \bowtie K_{AB})\}}. \quad (3)$$

By applying the authentication rule, we proved that MAC successfully authenticated the sender. Since the result of comparison ends successfully, the binding of the shared key and the identity of sender (Alice in this scenario) will be added to the binding set of Bob which ensures the authenticity of the received message.

Integrity Rule (RL18):

$$\frac{\{\text{Msg}\}^* \in \text{Proof}(\text{B})}{\text{BEL}(\text{B}) = \text{BEL}(\text{B}) \cup \{\text{Msg} \in \text{POSS}(\text{A})\}}. \quad (4)$$

The equality of the two digests also proves the correctness of the received message which leads Bob to adding a new value to Bob's Belief set indicating that the received message is the same as the message sent by Alice.

As can be concluded, LBSA has been used to mathematically prove the achievement of authentication and integrity by MAC. The above process is also applicable to any other security enforcer.

6.2. Digital Certificate. In hierarchical trust model, digital certificate [14] is usually issued by a certification authority (CA). It contains the user identity, public key (K^+), CA's digital signature, and other information. Using the CA's digital signature, the user can proof the authenticity of the issuer and ensure correct binding between the identity and the public key in the certificate.

We assume that the CA is the initiator of the protocol which will make one certificate for Alice and another for Bob. It sends each certificate after attaching its signature. Bob needs Alice's public key to send messages to her securely, therefore Alice sends her certificate to Bob then Bob starts the signature verification to ensure correct binding between Alice's public key and her identity.

Table 6: Mappings between protocol services and inference rules/actions.

Protocol Services	Inference rules/actions
Key management	ACT_6, ACT_{11}, ACT_{14}, ACT_{16}, ACT_{19}, ACT_{22}, ACT_{24}, ACT_{30}, RL_4, RL_5, RL_9, RL_{10}, RL_{11}
Nodes interaction and protocol functions	ACT_1, ACT_2, ACT_7, ACT_8, ACT_9, ACT_{12}, ACT_{17}, ACT_{18}, ACT_{20}, ACT_{21}, ACT_{23}, ACT_{29}, RL_{12}, RL_{14}
Data refreshment	ACT_5, ACT_{10}, RL_1, RL_3, RL_4, RL_5, RL_{15}, RL_{16}

Using LBSA to prove the achieved security requirements using the digital certificate, we start by presenting the enforcer specification in terms of global and local sets and then we performed the analysis part.

(i) Define the Values of the Global Sets. Consider

(GS1) P = {CA, Alice, Bob}

(GS2) R = {contains the rules defined in Tables 4(a)–4(d)}

(GS3) $S = \{K_{CA}^-, K_A^-, K_B^-\}$

(GS4) Observers $(K_{CA}^-) = \{CA\}$

(GS4) Observers $(K_A^-) = \{A\}$

(GS4) Observers $(K_B^-) = \{B\}$.

(ii) Define the Initial Values of the Local Sets to each Principal

Principal (CA):

(LS1) POSS(CA) = $\{K_{CA}^-, K_{CA}^+\}$

(LS2) BEL(CA) = $\{\#(K_{CA}^-), \#(K_{CA}^+)\}$

(LS7) Binding(CA) = $\{CA \bowtie K_{CA}^+\}$

(LS6) BL =

// Generation of Alice's certificate

(ACT_{12}) Apply(H, $cert_A$) // *digest generation*

(ACT_{15}) Apply-asymkey($H(cert_A)$, K_{CA}^-) // *generation of CA's digital signature*

(ACT_7) Concat($cert_A$, $\{H(cert_A)\}_{K_{CA}^-}$)

(ACT_1) Send(Alice, $cert_A.\{H(cert_A)\}_{K_{CA}^-}$)

// Generation of Bob's certificate

(ACT_{12}) Apply(H, $cert_B$)

(ACT_{15}) Apply-asymkey($H(cert_B)$, K_{CA}^-)

(ACT_7) Concat($cert_B$, $\{H(cert_B)\}_{K_{CA}^-}$)

(ACT_1) Send(Bob, $cert_B.\{H(cert_B)\}_{K_{CA}^-}$).

Principal (A) (Alice):

(LS1) POSS(A) = $\{K_A^-, K_A^+, K_{CA}^+\}$

(LS2) BEL(A) = $\{\#(K_A^-), \#(K_A^+), \#(K_{CA}^+)\}$

(LS7) Binding(CA) = $\{A \bowtie K_A^+, CA \bowtie K_{CA}^+\}$

(LS6) BL =

(ACT_2) Receive (CA, $cert_A.\{H(cert_A)\}_{K_{CA}^-}$)

(ACT_1) Send (Bob, $cert_A.\{H(cert_A)\}_{K_{CA}^-}$)—A(1).

Principal (B) (Bob):

(LS1) POSS(B) = $\{K_B^-, K_B^+, K_{CA}^+\}$

(LS2) BEL(B) = $\{\#(K_B^-), \#(K_B^+), \#(K_{CA}^+)\}$

(LS7) Binding(B) = $\{B \bowtie K_B^+, CA \bowtie K_{CA}^+\}$

(LS6) BL =

(ACT_2) Receive(CA, $cert_B.\{H(cert_B)\}_{K_{CA}^-}$) // *Bob receives its certificate from the CA*

(ACT_2) Receive(Alice, $cert_A.\{H(cert_A)\}_{K_{CA}^-}$) // *Bob receives Alice's certificate from the Alice*—B(1)

(ACT_8) Split($cert_A.\{H(cert_A)\}_{K_{CA}^-}$)—B(2)

(ACT_{21}) Comp(Apply(H, $cert_A$), Apply-asymkey($\{H(cert_A)\}_{K_{CA}^-}$, K_{CA}^+))

// Bob compares the calculated digest and the digest resulted from the decryption of the digital signature—B(3).

(iii) Digital Certificate Analysis. CA is assumed to be the initiator of the protocol and the first four actions in the BL(CA) are executed which will add new values to the POSS(CA) as follows:

POSS(CA) = $\{K_{CA}^-, K_{CA}^+, H(cert_A), \{H(cert_A)\}_{K_{CA}^-}, cert_A.\{H(cert_A)\}_{K_{CA}^-}\}$.

The next action to be executed is the first action in BL(A) which will receive the msg and update the POSS(A) to include the msg content:

POSS(A) = $\{K_A^-, K_A^+, K_{CA}^+, cert_A.\{H(cert_A)\}_{K_{CA}^-}\}$. Then CA does the rest of the actions which followed by executing the first action in BL(B) and these actions update the POSS(CA) again and the POSS(B).

POSS(CA) = $\{K_{CA}^-, K_{CA}^+, H(cert_A), \{H(cert_A)\}_{K_{CA}^-}, cert_A.\{H(cert_A)\}_{K_{CA}^-}, H(cert_B), \{H(cert_B)\}_{K_{CA}^-}, cert_B.\{H(cert_B)\}_{K_{CA}^-}\}$.

POSS(B) = $\{K_B^-, K_B^+, K_{CA}^+, cert_B.\{H(cert_B)\}_{K_{CA}^-}\}$.

Now Alice sends her certificate to Bob which is the action A(1) in the BL(A) and the next actions to be executed are B(1)-B(2) which update the POSS(B):

$$\text{POSS(B)} = \{K_{\text{B}}^{-}, K_{\text{B}}^{+}, K_{\text{CA}}^{+}, \text{cert}_{\text{B}}.\{H(\text{cert}_{\text{B}})\}_{K_{\text{CA}}^{-}}, \text{cert}_{\text{A}}.\{H(\text{cert}_{\text{A}})\}_{K_{\text{CA}}^{-}}, \text{cert}_{A}, \{H(\text{cert}_{\text{A}})\}_{K_{\text{CA}}^{-}}\}.$$

Since $\text{cert}_{\text{A}}.\{H(\text{cert}_{\text{A}})\}_{K_{\text{CA}}^{-}} \in \text{POSS(B)}$ we can apply Submessage origin rule (RL7):

$$\frac{\text{cert}_{\text{A}}.\{H(\text{cert}_{\text{A}})\}_{K_{\text{CA}}^{-}} \in \text{POSS(B)}}{(K_{\text{A}}^{+} \bowtie \text{A}) \text{ from CA} \in \text{POSS(B)}}, \quad \frac{\text{cert}_{\text{A}}.\{H(\text{cert}_{\text{A}})\}_{K_{\text{CA}}^{-}} \text{contains } K_{\text{A}}^{+}}{(K_{\text{A}}^{+} \bowtie \text{A}) \text{ from CA} \in \text{POSS(B)}}. \tag{5}$$

Which adding a new value to the POSS(B):

$$\text{POSS(B)} = \{K_{\text{B}}^{-}, K_{\text{B}}^{+}, K_{\text{CA}}^{+}, \text{cert}_{\text{B}}.\{H(\text{cert}_{\text{B}})\}_{K_{\text{CA}}^{-}}, \text{cert}_{\text{A}}.\{H(\text{cert}_{\text{A}})\}_{K_{\text{CA}}^{-}}, \text{cert}_{A}, \{H(\text{cert}_{\text{A}})\}_{K_{\text{CA}}^{-}}, (K_{\text{A}}^{+} \bowtie \text{A}) \text{ from CA}\}.$$

Finally, executing the action B(3) will add new values to POSS(B) (assuming the comparison action ensures equality) which achieves some rules conditions and these rules can be applied:

$$\text{POSS(B)} = \{K_{\text{B}}^{-}, K_{\text{B}}^{+}, K_{\text{CA}}^{+}, \text{cert}_{\text{B}}.\{H(\text{cert}_{\text{B}})\}_{K_{\text{CA}}^{-}}, \text{cert}_{\text{A}}.\{H(\text{cert}_{\text{A}})\}_{K_{\text{CA}}^{-}}, \text{cert}_{\text{A}}.\{H(\text{cert}_{\text{A}})\}_{K_{\text{CA}}^{-}}, (K_{\text{A}}^{+} \bowtie \text{A}) \text{ from CA}, H(\text{cert}_{\text{A}}), H(\text{cert}_{\text{A}}), \{\text{cert}_{\text{A}}\}^{*}\}.$$

Since $(K_{\text{A}}^{+} \bowtie \text{A}) \text{ from CA} \in \text{POSS(B)}, \{\text{cert}_{\text{A}}\}^{*} \in \text{POSS(B)}$ we can apply authentication and integrity rules as follows.

Authentication Rule (RL17):

$$\frac{(K_{\text{A}}^{+} \bowtie \text{A}) \text{ from CA} \in \text{POSS(B)}}{\text{Binding(B)} = \text{Binding(B)} \cup \{(K_{\text{A}}^{+} \bowtie \text{A})\}}, \quad \frac{(K_{\text{CA}}^{+} \bowtie \text{A}) \in \text{BEL(B)}, \{\text{cert}_{\text{A}}\}^{*} \in \text{Proof(B)}}{\text{Binding(B)} = \text{Binding(B)} \cup \{(K_{\text{A}}^{+} \bowtie \text{A})\}}. \tag{6}$$

Integrity Rule (RL18):

$$\frac{\{\text{cert}_{\text{A}}\}^{*} \in \text{Proof(B)}}{\text{BEL(B)} = \text{BEL(B)} \cup \{\text{cert}_{\text{A}} \in \text{Poss(A)}\}}. \tag{7}$$

Nonrepudiation Rule (RL19):

$$\frac{\{H(\text{cert}_{\text{A}})\}_{K_{\text{CA}}^{-}} \in \text{Poss(B)}, (K_{\text{CA}}^{+} \bowtie \text{CA}) \in \text{Binding(B)}}{(\text{CA} \longrightarrow \text{cert}_{\text{A}}) \in \text{Proof(B)}}. \tag{8}$$

Consequently, applying Authentication Rule (*RL17*) ensures the authenticity of Alice's public key. Integrity Rule (*RL18*) confirms the correctness of Alice's certificate. Finally, Nonrepudiation Rule (*RL19*) proves the originality of Alice's certificate, in other words it proves that Alice's certificate was issued and signed by the CA.

The purpose of LBSA is to generalize the rules, but the way they are applied depends on the enforcer itself. As shown this section, Rules 17 and 18 are used by both MAC and digital certificates but in different meanings. The authentication rule in MAC ensures the authenticity of the received message, whereas in digital certificate it ensures the authenticity of the public key. The integrity rule is used in MAC to ensure the correctness of the received message whereas in digital certificate it used to ensure the correctness of the certificate's contents.

7. Case Study: Ariadne Protocol

Many secure routing protocols have been proposed in the literature [27–32]. In this section, Ariadne protocol [27], which is a secure, on-demand routing protocol for multihop wireless networks was chosen to illustrate how to use LBSA to specify and analyze secure protocols. Ariadne is based on Timed Efficient Stream Loss-tolerant Authentication (TESLA) [33] which is an efficient broadcast authentication scheme that requires time synchronization. TESLA is commonly used in wireless sensor networks due to its low communication and computation overhead.

Before we start the security verification of Ariadne, we will present an overview of Ariadne route discovery using TESLA. For simplicity, we will use four nodes but what is applied in case of four nodes can be applied to more nodes. In all cases we will have a source, a destination, and a set of intermediate nodes which their number can vary depending on the chosen route. In the case of using more nodes, the change will be in the length of the chain of nodes. The way the authentication and integrity are achieved is defined by the Ariadne protocol itself. In this case study, we mapped Ariadne to LBSA to check Ariadne security correctness.

The source node S begins route instantiation to destination D by broadcasting a route request packet. A and B are intermediate nodes, that is, $S \rightarrow A \rightarrow B \rightarrow D$. A route reply packet is unicasted by the destination D as a reply to the request packet along the reverse path to the source.

In the route discovery process there are two types of messages REQUEST and REPLY. REQUEST and REPLY messages include the following fields.

Route REQUEST packet in Ariadne contains eight fields: REQUEST (define the type of the message), initiator address, target address, id (uniquely identifies the request), time interval, hash chain, node list, and MAC list. Note that the last three fields are updated by each intermediate node receiving that request.

Table 7 shows the protocol parameters.

The following steps illustrate how Ariadne route discovery using TESLA works:

S: $h_0 = \text{MAC}_{KSD}$ (REQUEST, S, D, id, ti)

$S \rightarrow$ broadcast: (REQUEST, S, D, id, ti, h_0, ( ), ( ))

A: $h1 = H(A, h_0)$

$M_A = \text{MAC}_{KA\text{ti}}$ (REQUEST, S, D, id, ti, $h1$, (A), ( ))

Table 7: Ariadne protocol parameters.

Parameter name	Meaning
S	Initiator (source) node.
D	Target (destination) node.
K_{SD}, K_{DS}	shared secret keys between S and D for message authentication in each direction.
t_i	Time interval.
Id	Message identifier.
$K_{A\text{ti}}$	TESTLA key (K) used by principal (A), i is the index for the time interval specified in the request.
$K_{B\text{ti}}$	TESTLA key (K) used by principal (B), i is the index for the time interval specified in the request.

$A \rightarrow$ broadcast: (REQUEST, S, D, id, ti, $h1$, (A), (M_A))

B: $h2 = H(B, h1)$

$M_B = \text{MAC}_{KB\text{ti}}$ (REQUEST, S, D, id, ti, $h1$, (A, B), (M_A))

$B \rightarrow$ broadcast: (REQUEST, S, D, id, ti, $h2$, (A, B), (M_A, M_B))

D: $M_D = \text{MAC}_{KSD}$ (REPLY, D, S, ti, (A, B), (M_A, M_B))

$D \rightarrow B$: (REPLY, D, S, ti, (A, B), (M_A, M_B), M_D, ( ))

$B \rightarrow A$: (REPLY, D, S, ti, (A, B), (M_A, M_B), M_D, ($K_{B\text{ti}}$))

$A \rightarrow S$: (REPLY, D, S, ti, (A, B), (M_A, M_B), M_D, ($K_{B\text{ti}}$, $K_{A\text{ti}}$)).

In the Ariadne protocol, the initiator of the REQUEST initializes the hash chain (h_0) to MAC_{KSD} (initiator, target, id, time interval) and the node list and MAC list to empty lists.

When any node A receives a route REQUEST for which it is not the target, the node checks its local table of (initiator, id) values from recent REQUESTs it has received to determine if it has already seen a REQUEST from this same Route Discovery. If it has, the node discards the packet. The node also checks whether the time interval in the REQUEST is valid. Then, the node modifies the REQUEST by appending its own address, A, to the node list in the REQUEST, replacing the hash chain field with H (A, hash chain), and appending a MAC of the entire REQUEST to the MAC list. The node uses the TESLA key $K_{A\text{ti}}$ to compute the MAC, where i is the index for the time interval specified in the REQUEST. Then, the node rebroadcast the modified REQUEST.

If the target node (destination) determines that the REQUEST is valid, it returns a route REPLY to the initiator, containing eight fields: REPLY (define the type of the message), target address (which is the initiator address in the REQUEST message), initiator address (which is the target address in the REQUEST message), time interval (the same as in the REQUEST message), node list, MAC list, target MAC, and key list. The target MAC is set to a MAC computed on the preceding fields in the REPLY with the key K_{DS}, and the key list is initialized to the empty list. The route REPLY is then returned to the initiator of the REQUEST along the source route obtained by reversing the sequence of hops in the node list of the REQUEST.

7.1. Ariadne Specification. First we will present the assumptions of Ariadne protocol which are as follows.

(i) The source S and destination D share the MAC keys K_{SD} and K_{DS}.

(ii) Every node has a TESLA one-way key chain.

(iii) All nodes know the authentication key of TESLA one-way key chain of each other node.

(i) Define the Values of the Global Sets. Consider

(GS1) $P = \{S, A, B, D\}$

(GS2) R = {contains the rules defined in Tables 4(a)–4(d)}

(GS3) $S = \{K_{SD}, K_{DS}, K_{B\text{ti}}, K_{A\text{ti}}\}$

(GS4) Observers (K_{SD}) = $\{S, D\}$

(GS4) Observers ($K_{A\text{ti}}$) = $\{A, S\}$

(GS4) Observers ($K_{B\text{ti}}$) = $\{B, S\}$

(GS4) Observers (K_{DS}) = $\{D, S\}$.

(ii) Define the Initial Values of the Local Sets for each Principal

Principal (S):

(LS1) POSS(S) = $\{K_{DS}, K_{SD}, K_{A\text{ti}}, K_{B\text{ti}}, (D \bowtie K_{DS})$ from D, $(D \bowtie K_{SD})$ from D, $(A \bowtie K_{A\text{ti}})$ from A, $(B \bowtie K_{B\text{ti}})$ from $B\}$

(LS2) BEL(S) = $\{\#(K_{DS}), \#(K_{SD}), \#(K_{A\text{ti}}), \#(K_{B\text{ti}}), (D \bowtie K_{DS}), (D \bowtie K_{SD}), (A \bowtie K_{A\text{ti}}), (B \bowtie K_{B\text{ti}})\}$

(LS6) BL =

(ACT_7) Concat ({REQUEST, S, D, id, ti}, K_{SD})—S(1)

(ACT_{12}, ACT_{29}) $h_0 \leftarrow$ Apply(H, {REQUEST, S, D, id, ti}. K_{SD})—S(2)

(ACT_{18}) Broadcast ({REQUEST, S, D, id, ti, h_0, (), () })—S(3)

(ACT_2) Receive (A, {REPLY, D, S, ti, (A, B), (M_A, M_B), M_D, ($K_{B\text{ti}}$, $K_{A\text{ti}}$)})—S(4)

(ACT_7) Concat({REPLY, D, S, id, ti, (A, B), (M_A, M_B)}, K_{DS})—S(5)

(ACT_{12}, ACT_{29}) $C1 \leftarrow$ Apply(H, {REPLY, D, S, id, ti, (A, B), (M_A, M_B)}.K_{DS})—S(6)

(ACT_{21}) Comp($M_D, C1$)—S(7)

(ACT_7) Concat ({REQUEST, S, D, id, ti, $h1$, (A), ()}, A_{ti})—S(8)

(ACT_{12}, ACT_{29}) $C2 \leftarrow$ Apply(H,{REQUEST, S, D, id, ti, $h1$, (A), ()}.A_{ti})—S(9)

(ACT_{21}) Comp($M_A, C2$)—S(10)

(ACT_7) Concat({REQUEST, S, D, id, ti, $h2$, (A, B), ()}, B_{ti})—S(11)

(ACT_{12}, ACT_{29}) $C3 \leftarrow$ Apply(H, {REQUEST, S, D, id, ti, $h2$, (A, B), ()}.B_{ti})—S(12)

(ACT_{21}) Comp($M_B, C3$)—S(13).

Principal (A):

(LS1) POSS(A) = {A_{ti}}

(LS2) BEL(A) = {#(A_{ti})}

(LS7) Binding(A) = {$A_{ti} \bowtie$ ti}

(LS6) BL =

(ACT_2) Receive({REQUEST, S, D, id, ti, h_0, (), ()})—A(1)

(ACT_{20}) Check-request(S, id)—A(2)

(ACT_{10}) Check-freshness(ti)—A(3)

(ACT_7) Concat(A, h_0)—A(4)

(ACT_{12}, ACT_{29}) $h1 \leftarrow$ Apply(H, {$A.h_0$})—A(5)

(ACT_7) Concat({REQUEST, S, D, id, ti, $h1$, (A), ()}, A_{ti})—A(6)

(ACT_{12}, ACT_{29}) $M_A \leftarrow$ Apply(H, {REQUEST, S, D, id, ti, $h1$, (A), ()}.A_{ti})—A(7)

(ACT_{18}) Broadcast({REQUEST, S, D, id, ti, $h1$, (A), (M_A)}—A(8)

(ACT_2) Receive(B,{REPLY, D, S, ti, (A, B), (M_A, M_B), M_D, (K_{Bti})})—A(9)

(ACT_1) Send(S, {REPLY, D, S, ti, (A, B), (M_A, M_B), M_D, (K_{Bti}, K_{Ati})})—A(10).

Principal (B):

(LS1) POSS(B) = {B_{ti}}

(LS2) BEL(B) = {#(B_{ti})}

(LS7) Binding(B) = {$B_{ti} \bowtie$ ti}

(LS6) BL =

(ACT_2) Receive({REQUEST, S, D, id, ti, $h1$, (A), (M_A)})— B(1)

(ACT_{20}) Check-request(S, id)— B(2)

(ACT_{10}) Check-freshness(ti)— B(3)

(ACT_7) Concat($B, h1$)— B(4)

(ACT_{12}, ACT_{29}) $h2 \leftarrow$ Apply(H, {$B.h1$})— B(5)

(ACT_7) Concat({REQUEST, S, D, id, ti, $h2$, (A, B), ()}, B_{ti})—B(6)

(ACT_{12}, ACT_{29}) $M_B \leftarrow$ Apply (H, {REQUEST, S, D, id, ti, $h2$, (A, B), ()}.B_{ti})—B(7)

(ACT_{18}) Broadcast({REQUEST, S, D, id, ti, $h2$, (A, B), (M_A, M_B)}—B(8)

(ACT_2) Receive(D, {REPLY, D, S, ti, (A, B), (M_A, M_B), M_D, ()})—B(9)

(ACT_1) Send(A, {REPLY, D, S, ti, (A, B), (M_A, M_B), M_D, (K_{Bti})}— B(10).

Principal (D):

(LS1) POSS(D) = {K_{DS}, K_{SD}, ($S \bowtie K_{SD}$) from S, ($S \bowtie K_{DS}$) from S}

(LS2) BEL(D) = {#(K_{SD}), #(K_{SD}), $K_{SD} \bowtie S$, $K_{DS} \bowtie S$}

(LS6) BL =

(ACT_2) Receive({REQUEST, S, D, id, ti, $h2$, (A, B), (M_A, M_B)}—D(1)

(ACT_{20}) Check-request(S, id)—D(2)

(ACT_{10}) Check-freshness(ti)—D(3)

(ACT_{12}, ACT_{29}) C1 $\leftarrow$ Apply(H, Concat(A, Apply(H, Concat (B, Apply(H, (ACT_7) Concat({REQUEST, S, D, id, ti}, K_{SD}))))))—D(4)

(ACT_{21}) Comp ($h2, C1$)—D(5)

(ACT_7) Concat({REPLY, D, S, id, ti, (A, B), (M_A, M_B)}, K_{DS}) —D(6)

(ACT_{12}, ACT_{29}) $M_D \leftarrow$ Apply(H, {REPLY, D, S, id, ti, (A, B), (M_A, M_B)}. K_{DS})—D(7)

(ACT_1) Send (B, {REPLY, D, S, ti, (A, B), (M_A, M_B), M_D, ()})—D(8).

7.2. Ariadne Analysis. Node S is the initiator of the protocol and its first three actions will be executed which result in new values to be added to the POSS(S) set:

POSS(S):= POSS(S) ∪({{REQUEST, S, D, id, ti}.K_{SD}}, {h_0 = H({REQUEST, S, D, id, ti}.K_{SD})}).

Now we assume that node A receives the request so first eight actions in BL(A) will be executed and the result is

POSS(A) = POSS(A) ∪{{REQUEST, S, D, id, ti, h_0, (), ()}, $A.h_0, h1 = H(A.h_0)$,

$M_A = H$({REQUEST, S, D, id, ti, $h1$, (A), ()}.A_{ti})}

BEL(A) =BEL(A) ∪ {#(ti)}.

Now B(1–8) will be executed and the result is

POSS(B) = POSS(B) ∪{{REQUEST, S, D, id, ti, $h1$, (A), (M_A)}, $B.h1, h2 = H(B.h1)$, {REQUEST, S, D, id, ti, $h2$, (A, B), ()}.B_{ti}, $M_B = H$({REQUEST, S, D, id, ti, $h2$, (A, B), ()}.B_{ti})}

BEL(B) = BEL(B) ∪ {#(ti)}.

Now the request reaches its target which is node D and its BL(1–5) actions will be executed and the result will be as follows:

$\text{POSS}(D) = \text{POSS}(D) \cup \{\{\text{REQUEST}, S, D, \text{id}, \text{ti}, h2, (A, B), (M_A, M_B)\}, C1 = H(A.B.\ \{\text{REQUEST}, S, D, \text{id}, \text{ti}\}.K_{SD},\ \{\text{REPLAY}, D, S, \text{id}, \text{ti}, (A, B), (M_A, M_B)\})$

$\text{BEL}(D) = \text{BEL}(D) \cup \{\#(\text{ti})\}$

$\text{Proof}(D) = \text{Proof}(D) \cup \{h2\}^*$

Since $\{h2\}^* \in \text{Proof}(D)$ we can apply integrity rule RL(18): $\{h2\}^* \in \text{Proof}(D)/\ \text{BEL}(D) = \text{BEL}(D) \cup \{\{\text{REQUEST}, S, D, \text{id}, \text{ti}, h_0, (), ()\} \in \text{Poss}(S)\}$.

After the execution of D(7) and D(8), the POSS set will be updated as follows

$\text{POSS}(D) = \text{POSS}(D) \cup \{\{\text{REQUEST}, S,\ D, \text{id}, \text{ti}, h2, (A, B), (M_A, M_B)\}, \text{C1} = H(A.\ B.\ \{\text{REQUEST}, S,\ D,\ \text{id},\ \text{ti}\}.K_{SD},\ \{\text{REPLAY},\ D, S,\ \text{id},\ \text{ti}, (A, B), (M_A, M_B)\}.K_{DS},\ M_D = H(\{\text{REPLAY}, D, S, \text{id}, \text{ti}, (A, B), (M_A, M_B)\}.K_{DS})$.

Therefore, the target (destination) node makes sure that the request is from node S and it will send a reply message using the reverse path since the comparison action confirms equality. Each intermediate node (A and B in our example) will receive the reply, adds its key to the key list, and forwards it to the next node until it reaches the source. The destination will authenticate each intermediate node using the same process as the target node did.

When the target node S receives the REPLY the rest of its actions will be executed to ensure the validity of the target's MAC, and each MAC in the MAC list. Node S local sets will be updated as follows:

$\text{POSS}(S) = \{K_{DS}, K_{SD}, K_{A\text{ti}}, K_{B\text{ti}}, (A \bowtie K_{A\text{ti}})\ \text{from}\ A, (B \bowtie K_{B\text{ti}})\ \text{from}\ B, \{\text{REPLY}, D, S,\ \text{ti}, (A, B), (M_A, M_B),\ M_D,\ (K_{B\text{ti}}, K_{A\text{ti}})\},\ \{\text{REPLY}, D, S,\ \text{id},\ \text{ti}, (A, B), (M_A, M_B)\}.K_{DS}\}, \{\text{REQUEST}, S, D, \text{id}, \text{ti}, h1, (A), ()\}.A_{\text{ti}}, \{\text{REQUEST}, S, D, \text{id}, \text{ti}, h2, (A, B), ()\}.B_{\text{ti}}\}$

$\text{Proof}(S) = \text{Proof}(S) \cup \{\{M_D\}^*, \{M_B\}^*, \{M_A\}^*\}$.

Now we can apply the authentication rule (RL17) for each entry in the proof set.

Authentication Rule (RL17):

$$\frac{(D \bowtie K_{DS})\ \text{from}\ D \in \text{POSS}(S)}{\text{Binding}(S) = \text{Binding}(S) \cup \{(D \bowtie K_{DS})\}},$$

$$\frac{(D \bowtie K_{DS}) \in \text{BEL}(S), \{M_D\}^* \in \text{Proof}(S)}{\text{Binding}(S) = \text{Binding}(S) \cup \{(D \bowtie K_{DS})\}},$$

$$\frac{(B \bowtie K_{B\text{ti}})\ \text{from}\ B \in \text{POSS}(S)}{\text{Binding}(S) = \text{Binding}(S) \cup \{(B \bowtie K_{B\text{ti}})\}},$$

$$\frac{(B \bowtie K_{B\text{ti}}) \in \text{BEL}(S), \{M_B\}^* \in \text{Proof}(S)}{\text{Binding}(S) = \text{Binding}(S) \cup \{(B \bowtie K_{B\text{ti}})\}},$$

$$\frac{(A \bowtie K_{A\text{ti}})\ \text{from}\ A \in \text{POSS}(S)}{\text{Binding}(S) = \text{Binding}(S) \cup \{(A \bowtie K_{A\text{ti}})\}},$$

$$\frac{(A \bowtie K_{A\text{ti}}) \in \text{BEL}(S), \{M_A\}^* \in \text{Proof}(S)}{\text{Binding}(S) = \text{Binding}(S) \cup \{(A \bowtie K_{A\text{ti}})\}}. \tag{9}$$

Also we can apply the integrity rule (RL18) for each entry in the proof set.

Integrity Rule (RL18):

$$\{M_D\}^* \in \text{Proof}(S) \times (\text{BEL}(S) := \text{BEL}(S) \cup \{\{\text{REPLY}, D, S, \text{ti}, (A, B), (M_A, M_B), M_D, ()\} \in \text{Poss}(D)\})^{-1} \tag{10}$$

$$\{M_B\}^* \in \text{Proof}(S) \times (\text{BEL}(S) := \text{BEL}(S) \cup \{\{\text{REPLY}, D, S, \text{ti}, (A, B), (M_A, M_B), M_D, ()\} \in \text{Poss}(B)\})^{-1} \tag{11}$$

$$\{M_A\}^* \in \text{Proof}(S) \times (\text{BEL}(S) := \text{BEL}(S) \cup \{\{\text{REPLY}, D, S, \text{ti}, (A,B), (M_A,M_B), \text{M}_A, (K_{B\text{ti}}, K_{A\text{ti}})\} \in \text{Poss}(A)\})^{-1}. \tag{12}$$

Since we applied the authentication and the integrity rules successfully on the target's MAC and on each MAC in the MAC list, it can be concluded that the reply is valid and authenticated; consequently node S can accept it. Therefore, Ariadne achieves both the authentication and the integrity requirements which detect attackers when they attempt to impersonate the message's sender or modify its content.

Flaws in this protocol lie in the assumptions of the authors that may affect the usability of the protocol. Ariadne assumed the presence of a working public key infrastructure (PKI) or certificate authority (CA). Consequently, key management services are assumed to be already existed but are not provided by Ariadne. Also, Ariadne did not take the possibility of having internal attackers into consideration and therefore, no IDS are applied to detect them.

Based on the above assumptions, actions such as ACT_6, ACT_{11}, ACT_{13}, ACT_{14}, ACT_{16}, ACT_{19}, ACT_{22}, ACT_{24}, ACT_{25}, ACT_{26}, ACT_{27}, and ACT_{28} cannot be activated and rules like R4, R5, R9, R10, and R11 cannot be applied. Based on this observation, LBSA can detect the absence of key management system and IDS in Ariadne. As a result, key management services will fail and intruders cannot be detected.

8. Conclusions and Future Work

The success of applications over multihop communication networks principally depends on the achievement of security requirements. Many secure protocols are proposed, consequently, there should be a reliable way to test if these protocols satisfy the security requirements as their designers claim.

Previous attempts to utilize logic in analyzing security have either focused on a specific requirement or a specific protocol. In this paper, we proposed a logic-based security architecture (LBSA) to specify and analyze any security requirement in any system providing multihop communication using logic. Any system or protocol claiming security can be mapped into our architecture in other words it must be specified using various global/local sets and actions, then it can be analyzed by applying different rules. LBSA provides a formal way to analyze security enforcers and security protocols instead of depending only on the simulation tools which are arguable by many researchers in terms of implementation accuracy.

Additionally, using simulation tools usually covers a small set of scenarios whereas using LBSA more cases can be covered that usually cannot be covered exhaustively by simulation tools.

This paper also illustrated how LBSA can be used to verify security enforcers by applying it to two well-known enforcers which are MAC and digital certificate. In addition to that, a case study for a secure routing protocol used in ad hoc and sensor networks named Ariadne was also evaluated in terms of security using LBSA.

To the best of our knowledge, the proposed security architecture covers the most important enforcers. Further development of new enforcers in the future may simply need other sets, actions, and rules to be added. Also, LBSA could be extended by considering new rules and actions for the Interaction between Intrusion Detection System (IDS) and Intrusion Prevention System (IPS). Moreover, LBSA can utilize the computational Intelligence tools in analyzing the security.

References

[1] ITU-T Recommendation X. 800, "Security architecture for Open Systems Interconnection for CCITT applications," 1991.

[2] ITU-T Recommendation X. 805, "Security architecture for systems providing end-to-end communication," 2003.

[3] A. D. Rubin and P. Honeyman, "Nonmonotonic cryptographic protocols," in *Proceedings Computer Security Foundations Workshop*, pp. 100–116, June 1994.

[4] A. D. Rubin, "Extending NCP for protocols using public keys," *Mobile Networks and Applications*, vol. 2, no. 3, pp. 227–241, 1997.

[5] Y. Xu and X. Xie, "Security analysis of routing protocol for MANET based on extended Rubin logic," in *Proceedings of the IEEE International Conference on Networking, Sensing and Control (ICNSC '08)*, pp. 1326–1331, Sanya, China, April 2008.

[6] Y. Xu and X. Xie, "Extending rubin logic for electronic commerce protocols," in *Proceedings of the 2nd International Conference on Anti-counterfeiting, Security and Identification (ASID '08)*, pp. 448–451, Guiyang, China, August 2008.

[7] Y. Xu and X. Xie, "Analysis of electronic commerce protocols based on extended rubin logic," in *Proceedings of the 9th International Conference for Young Computer Scientists (ICYCS '08)*, pp. 2079–2084, Hunan, China, November 2008.

[8] Y. Xu and X. Xie, "Analysis of authentication protocols based on rubin logic," in *Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '08)*, Dalian, China, October 2008.

[9] A. D. Rubin and P. Honeyman, "Long running jobs in an authenticated environment," in *Proceedings of the USENIX Security Conference IV*, pp. 19–28, October 1993.

[10] R. M. Needham and M. D. Schroeder, "Using encryption for authentication in large networks of computers," *Communications of the ACM*, vol. 21, no. 12, pp. 993–999, 1978.

[11] K. Sanzgiri, D. LaFlamme, B. Dahill, B. N. Levine, C. Shields, and E. M. Belding-Royer, "Authenticated routing for ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 3, pp. 598–610, 2005.

[12] J. Zhou and D. Gollmann, "Fair non-repudiation protocol," in *Proceedings of the 17th IEEE Symposium on Security and Privacy*, pp. 55–61, Oakland, Calif, USA, May 1996.

[13] M. Satyanarayanan, "Integrating security in a large distributed system," *ACM Transactions on Computer Systems*, vol. 7, no. 3, pp. 247–280, 1989.

[14] ITU-T Recommendation X. 509, "Information technology—Open System Interconnection—The Directory: public-key and attribute certificate framework," 2008.

[15] J. Weiss, "Message digest, message authentication codes and digital signature," in *Jave Cryptography Extensions*, pp. 101–118, 2004.

[16] J. Zhou and K. Y. Lam, "Securing digital signatures for non-repudiation," *Computer Communications*, vol. 22, no. 8, pp. 710–716, 1999.

[17] Z. Shao, "Certificate-based fair exchange protocol of signatures from pairings," *Computer Networks*, vol. 52, no. 16, pp. 3075–3084, 2008.

[18] C. H. Lima and P. J. Lee, "Korean certificate-based digital signature algorithm," *Computers and Electrical Engineering*, vol. 25, no. 4, pp. 249–265, 1999.

[19] S. Madria and J. Yin, "SeRWA: a secure routing protocol against wormhole attacks in sensor networks," *Ad Hoc Networks*, vol. 7, no. 6, pp. 1051–1063, 2009.

[20] S. Ozdemir and H. Çam, "Integration of false data detection with data aggregation and confidential transmission in wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 18, no. 3, pp. 736–749, 2010.

[21] E. Stavrou and A. Pitsillides, "A survey on secure multipath routing protocols in WSNs," *Computer Networks*, vol. 54, no. 13, pp. 2215–2238, 2010.

[22] O. Delgado-Mohatar, A. Fúster-Sabater, and J. M. Sierra, "A light-weight authentication scheme for wireless sensor networks," *Ad Hoc Networks*, vol. 9, no. 5, pp. 727–735, 2011.

[23] I. Almomani and M. Saadeh, "Security model for tree-based routing in wireless sensor networks: structure and evaluation," *KSII Transactions on Internet and Information Systems*, vol. 6, no. 4, pp. 1223–1247, 2012.

[24] K. Bicakci, I. E. Bagci, and B. Tavli, "Communication/computation tradeoffs for prolonging network lifetime in wireless sensor networks: the case of digital signatures," *Information Sciences*, vol. 188, pp. 44–63, 2012.

[25] X. Fan and G. Gong, "Accelerating signature-based broadcast authentication for wireless sensor networks," *Ad Hoc Networks*, vol. 10, no. 4, pp. 723–736, 2012.

[26] ISO/IEC 9797-2, "Information technology-security techniques-Message Authentication Code (MACs)—Part 2: mechanisms using a dedicated hash function".

[27] Y. C. Hu, A. Perrig, and D. B. Johnson, "Ariadne: a secure on-demand routing protocol for ad hoc networks," *Wireless Networks*, vol. 11, no. 1-2, pp. 21–38, 2005.

[28] R. Mavropodi, P. Kotzanikolaou, and C. Douligeris, "SecMR—a secure multipath routing protocol for ad hoc networks," *Ad Hoc Networks*, vol. 5, no. 1, pp. 87–99, 2007.

[29] J. Kim and G. Tsudik, "SRDP: secure route discovery for dynamic source routing in MANETs," *Ad Hoc Networks*, vol. 7, no. 6, pp. 1097–1109, 2009.

[30] S. Gupte and M. Singhal, "Secure routing in mobile wireless ad hoc networks," *Ad Hoc Networks*, vol. 1, no. 1, pp. 151–174, 2003.

[31] K. K. Chauhan, A. K. S. Sanger, and V. S. Kushwah, "Securing on-demand source routing in MANETs," in *Proceedings of the 2nd International Conference on Computer and Network Technology (ICCNT '10)*, pp. 294–297, Bangkok, Thailand, April 2010.

[32] S. G. Yoo, K. Y. Park, and J. Kim, "A security-performance-balanced user authentication scheme for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 382810, 11 pages, 2012.

[33] A. Perrig, R. Canetti, J. D. Tygar, and D. Song, "The TESLA broadcast authentication protocol," *CryptoBytes*, RSA Laboratories, vol. 5, no.2, 2002.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 763187, 12 pages
doi:10.1155/2012/763187

Research Article

Secure Neighbor Discovery in Wireless Sensor Networks Using Range-Free Localization Techniques

Mariano García-Otero and Adrián Población-Hernández

Departmento de Señales, Sistemas y Radiocomunicaciones, ETSI de Telecomunicación, Universidad Politécnica de Madrid, Avenida Complutense 30, 28040 Madrid, Spain

Correspondence should be addressed to Mariano García-Otero, mariano@gaps.ssr.upm.es

Received 14 July 2012; Revised 27 September 2012; Accepted 27 September 2012

Academic Editor: An Liu

If a wireless sensor network (WSN) is deployed in a hostile environment, the intrinsic limitations of the nodes lead to many security issues. In this paper, we address a particular attack to the location and neighbor discovery protocols, carried out by two colluding nodes that set a wormhole to try to deceive an isolated remote WSN node into believing that it is a neighbor of a set of local nodes. To counteract such threat, we present a framework generically called detection of wormhole attacks using range-free methods (DWARF) under which we derive two specific wormhole detection schemes: the first approach, DWARFLoc, performs jointly the detection and localization procedures employing range-free techniques, while the other, DWARFTest, uses a range-free method to check the validity of the estimated position of a node once the location discovery protocol is finished. Simulations show that both strategies are effective in detecting wormhole attacks, and their performances are compared with that of a conventional likelihood ratio test (LRT).

1. Introduction

Wireless sensor networks (WSNs) are composed of a potentially large number of low-cost and resource-constrained devices which are often distributed over a wide area. Thus, if a WSN is deployed in an unfriendly environment, providing security to the involved network protocols is a challenging task that usually requires the use of different combined strategies [1].

A protocol that deserves special attention from a security point of view is neighbor discovery (ND). This is because one of the most basic requirements in a WSN is the ability of every node to reliably determine which of the other nodes are within its radio range so that it can establish single-hop links with them. Trustworthy ND is a cornerstone for securing higher-level network protocols and system functionalities, such as physical and network access control, data routing, and node localization [2].

In a hostile environment, a WSN can be compromised by different threats, but the so-called wormhole or relay attack lies among the most devastating [3]. A wormhole is a high-speed direct communication link between two malicious nodes that act in collusion by capturing network packets on one end, sending them through the wormhole and replaying them at the other end. Thus, to launch a wormhole attack, an adversary does not need to infect any network node or break any cryptographic system, making it a quite severe threat to WSNs.

Wormholes completely distort the network topology, making distant nodes to appear as local for a given node looking for its neighbors. As a side effect of a failed ND due to a wormhole, most location discovery (LD) protocols will also be compromised; this is because the wormhole severely distorts all the measurements related to the relative positions of the nodes. However, in some cases, the high sensitivity of

LD protocols to wormholes can be turned into an advantage, because the localization process can be suitably modified to detect the presence of an attack.

In this paper we address this approach for the detection of wormholes. Specifically, we propose a general framework called detection of wormhole attacks using range-free methods (DWARF) that has two modes of operation: the first one (DWARFLoc) performs the detection of a wormhole simultaneously with the localization procedure, while the second one (DWARFTest) is a postlocalization detector that tries to validate the node position after this latter is obtained. The principles of DWARF are rooted in the exploitation of the ideas underlying the operation of a range-free localization method, namely, the so-called "sensor localization with Ring Overlapping based on Comparison of Received Signal Strength Indicator" (ROCRSSI) algorithm [4].

The main contributions of this paper are as follows.

(i) The formulation of a simplified attack model for which the detection of a wormhole can be rigorously formulated as a binary hypothesis testing problem.

(ii) The derivation of the likelihood ratio test (LRT) as the asymptotically optimal solution for the wormhole detection problem. However, the LRT requires a precise statistical model for the observations.

(iii) The derivation of DWARFLoc and DWARFTest as robust alternatives to the LRT, because they are not tied to any particular channel model.

(iv) The evaluation of the relative performances of both categories of tests (LRT and DWARF) through simulations.

The rest of the paper is organized as follows. Section 2 reviews related work concerning wormhole detection. Section 3 presents basic ideas about range-free localization and briefly describes the ROCRSSI algorithm. Section 4 defines the particular attack to be counteracted. Section 5 formulates the wormhole detection problem under the framework of statistical hypothesis testing and derives the LRT. Section 6 presents the two wormhole detection strategies DWARFLoc and DWARFTest. Section 7 evaluates the performance of the different wormhole detection strategies through simulations. Finally, section 8 draws some conclusions.

2. Related Work

In recent years, the topic of secure ND has been extensively studied and a lot of different defensive measures against wormhole attacks are described in the related literature.

For instance, it is proposed in [3] the use of location and time stamps, that is, geographical and temporal "leashes", attached to network packets to detect wormhole attacks; therefore, this strategy assumes that all the nodes know their exact positions and are synchronized in time, which are probably unrealistic hypotheses if the network is under attack.

In [5], a wormhole detection algorithm for a multihop wireless network is presented, based on a search of forbidden substructures in the connectivity graph.

The authors of [6] present different preventive mechanisms against wormholes and propose an intruder detection system, LIDeA, in which every node analyzes their neighbors and collaborates to detect suspicious nodes using a voting strategy.

In [7], the authors introduce a graph-based and beaconless solution that detects wormholes visually by reconstructing the network topology using only inaccurate distances between the nodes; however, an irregular-shaped network or multiple wormholes may lead to an incorrect detection.

The cryptographic concept of "pairing" is introduced in [8]. The article describes a node-to-node neighborhood authentication protocol based on location-based keys (private keys of individual nodes that are bound to their identities and positions), to avoid malicious nodes to join the network.

Wu et al. [9] propose a localization scheme based on hop counts (DV-Hop) by labeling the neighboring nodes of beacon nodes according to different algorithms to detect wormhole attacks; nevertheless, the proposed scheme does not work well if the network has packet losses or the transmission ranges of all nodes are not identical.

Robust localization techniques were described in [10, 11], using the concept of "verifiable multilateration." Both are range-based approaches: while ROPE [10] provides secure localization and location verification using directional antennas and distance bounding, SPINE [11] estimates the distances between the nodes by measuring the time of flight of the radio signal. These solutions require either perfectly known directional antennas or specific transceivers capable of measuring the time of flight.

A secure range-free localization method called SeRLoc was proposed in [12], where the nodes are supposed to be equipped with static directional antennas with a fixed communication range, the nodes are localized by overlapping regions within communication range, and the wormholes are detected by checking the properties of message uniqueness and communication range violation. HiRLoc [13] is the evolution of SeRLoc and provides a high-resolution localization by adding two variables to the localization algorithm, the angle of rotation of the antennas, and the transmission power, increasing the complexity of the nodes.

Recently, ConSetLoc [14] proposes a robust range-free localization scheme based on evaluating the relationship between hops and distances and then applying convex constraints in geometry to reduce localization errors induced by wormholes.

For moving nodes, a secure ND protocol called MSDN [15] has been proposed, applying the notion of graph rigidity to aid moving network nodes in the verification of neighbors.

All the procedures for secure ND described above assume that the two colluding nodes forming a wormhole are located within the network deployment area. However, as we will see in Section 4, the particular threat we will address in this paper assumes that one of the wormhole nodes is situated out of the range of the WSN nodes but in the vicinity of an isolated node which is the target of the attack. So, this particular

wormhole attack to the LD and ND protocols cannot be detected by conventional techniques.

3. Range-Free Localization

Traditional localization techniques rely on providing network nodes with auxiliary devices capable of self-acquiring their coordinates in a geographical reference system, such as global positioning system (GPS) receivers. Such solutions, however, have severe drawbacks in terms of their cost and energy consumption and are unable to operate indoors. A much more flexible approach to LD is obtained if we assume that only a small number of network nodes are assumed to know their own locations (through GPS receivers or system configuration), while the other nodes are only able to measure their relative distances to other neighbor nodes and use these data to position themselves. Focusing on the physical layer (PHY) level, received signal strength (RSS) is a parameter readily available in most commercial sensor nodes, usually in a coarsely quantized form called RSS indicator (RSSI). RSS measurements can be used for localization, because they are related to the distances between nodes [16, 17]; however, as they strongly depend on the particular hardware used and also on often unpredictable environment conditions, in many cases they cannot be used to directly estimate distances. Therefore, in recent times several "range-free" alternatives to localization have been proposed; these methods use an indirect approach and provide localization without the need of accurate distance estimations.

We point here that there is some controversy regarding the expression "range-free" when applied to localization because, for some authors, this term only refers to techniques based on connectivity information, which can be interpreted as a binary quantization of RSS. We will, however, adopt a broader interpretation of "range-free" schemes as those that use RSS values but do not rely on the existence of any precise relationship between RSSs and distances, only assuming there is a loose link between these parameters [18]. We will also call these methods "nonparametric," as opposed to "parametric" or "range-based" approaches, which require a precise model relating RSS values to distances.

For instance, if we denote the Euclidean distance between two arbitrary network nodes at positions $\mathbf{x}$ and $\mathbf{y}$ as $d(\mathbf{x},\mathbf{y}) \equiv ||\mathbf{x}-\mathbf{y}||$ and the RSS (in dBm) measured at the receiver of node $\mathbf{y}$ for a signal transmitted by node $\mathbf{x}$ as $r(\mathbf{x},\mathbf{y})$, a common basic assumption in many range-free methods is the validity of a simple monotonicity constraint:

$$r(\mathbf{x},\mathbf{y}) > r(\mathbf{x},\mathbf{y}) \Longleftrightarrow d(\mathbf{x},\mathbf{y}) < d(\mathbf{x},\mathbf{y}),\ \forall \mathbf{x},\mathbf{y},\mathbf{z} \in \mathbb{R}^2. \tag{1}$$

Notice that because the transmitted power is assumed to be unknown, RSS measurements are not expected to be symmetric that is, $r(\mathbf{x},\mathbf{y}) \neq r(\mathbf{y},\mathbf{x})$. One of the most straightforward approaches to the solution of the problem of localizing a node based on the restriction (1) is given by the so-called ROCRSSI algorithm [4].

This range-free localization method assumes that there is a node trying to estimate its own unknown position $\mathbf{p}$, surrounded by N "anchor nodes" located at known positions $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_N$. Every anchor node is continuously broadcasting beacon packets that include, along with its own location, the RSS values corresponding to beacon signals received from all the other anchor nodes in its vicinity. Therefore, for every anchor $\mathbf{a}_i$ $(i = 1, 2, \ldots, N)$ in the neighborhood of $\mathbf{p}$, we will assume that the following RSS values are available:

One anchor-to-node RSS: $r(\mathbf{a}_i, \mathbf{p})$,

$N-1$ anchor-to-anchor RSSs: $r(\mathbf{a}_i, \mathbf{a}_j)$, for all $i \neq j$.

Now, by applying the monotonicity constraint (1) to this set of RSS measurements, the localization algorithm obtains the tightest possible lower and upper bounds, $\rho_1^{(i)}$ and $\rho_2^{(i)}$, respectively, for the possible values of the distance between the ith anchor and the node to be located; this, in turn, translates to a restriction in the position of the node as a ring $R(\mathbf{a}_i, \rho_1^{(i)}, \rho_2^{(i)})$, centered around $\mathbf{a}_i$ and with inner and outer radii $\rho_1^{(i)}$ and $\rho_2^{(i)}$; respectively,

$$R\left(\mathbf{a}_i, \rho_1^{(i)}, \rho_2^{(i)}\right) = \left\{\mathbf{p} \in \mathbb{R}^2 : \rho_1^{(i)} < d(\mathbf{a}_i, \mathbf{p}) < \rho_2^{(i)}\right\}, \quad i = 1, 2, \ldots, N, \tag{2}$$

with $\rho_1^{(i)}$ and $\rho_2^{(i)}$ obtained as

$$\rho_1^{(i)} = \begin{cases} d(\mathbf{a}_i, \mathbf{a}_m), & \text{if } \exists r(\mathbf{a}_i, \mathbf{a}_m) \\ & = \inf\left\{r\left(\mathbf{a}_i, \mathbf{a}_j\right), j \neq i : r\left(\mathbf{a}_i, \mathbf{a}_j\right) > r\left(\mathbf{a}_i, \mathbf{p}\right)\right\}, \\ 0, & \text{otherwise,} \end{cases}$$

$$\rho_2^{(i)} = \begin{cases} d(\mathbf{a}_i, \mathbf{a}_n), & \text{if } \exists r(\mathbf{a}_i, \mathbf{a}_n) \\ & = \sup\left\{r\left(\mathbf{a}_i, \mathbf{a}_j\right), j \neq i : r\left(\mathbf{a}_i, \mathbf{a}_j\right) < r\left(\mathbf{a}_i, \mathbf{p}\right)\right\}, \\ \infty, & \text{otherwise,} \end{cases} \tag{3}$$

where $\inf(S)$ and $\sup(S)$ denote the infimum and supremum of the set S, respectively.

After repeating this procedure for all the anchors, the node is found to be located on the intersection of a set of rings $R(\mathbf{a}_i, \rho_1^{(i)}, \rho_2^{(i)})$, $i = 1, 2, \ldots, N$, of the form (2); finally, the node position is estimated as the centroid of the intersection region. Notice that unlike range-based methods, range-free localization techniques cannot obtain the exact node position $\mathbf{p}$, even in the absence of measurement errors, because they only provide bounds for the location; however, these bounds tend to be tighter as the number of anchors N increases.

With actual measurements, the condition (1) does not hold for every pair of nodes because the radio channel is usually anisotropic, so that not all the rings (2) have a common intersection. The compromise solution in such cases is to assume the UN to be in the region of the plane where most of the rings intersect. This is equivalent to assume that every anchor "votes" for a given ring as a candidate to hold the UN, and the region of the plane that gets the higher number of votes is finally elected. Such voting strategy has the added benefit of providing a good degree of robustness to attacks to the localization process triggered by malicious anchors [19, 20].

On the other hand, the achieved number of votes (i.e., intersecting rings) for the region of the plane finally elected is also an indicator of the "degree of success" of the localization process: a high value for this number (relative to its absolute maximum, i.e., the number of anchors N) implies that RSS measurements are highly correlated to actual distances between nodes, so that the monotonicity constraint (1) is fulfilled in most situations. This fact is illustrated in Figure 1, where we can see examples of two extreme cases: Figure 1(a) represents the distribution of the number of votes when the node to be located receives RSS measurements that are independent of distances, while Figure 1(b) illustrates a situation where RSSs are deterministically related to distances; notice the presence of a sharp peak in this latter case, highlighting the area where the node is located.

The ROCRSSI algorithm, unlike other range-free approaches, does not require any special hardware at the nodes (like directive antennas) and its implementation does not depend on parameters that are somewhat imprecisely defined such as the "communication range," commonly employed by range-free methods based on connectivity.

4. Attack Model

We will assume the existence of an adversary who tries to deceive both the location and neighborhood discovery protocols by forcing a remote compromised node to appear as a neighbor of the local network nodes. To accomplish this, the attacker uses a wormhole link with two endpoints: one in the vicinity of the anchor nodes, and the other within the radio range of the compromised node (see Figure 2); the wormhole local node captures beaconing packets sent from the anchor nodes and tunnels them to the wormhole remote node through a dedicated high-speed link, so that they arrive unmodified at the compromised node. This latter node, then, applies the localization procedure using these packets as if they came directly from the anchors, therefore resulting in a fake position within the deployment area of the local network. As the wormhole nodes act as simple relays and do not manipulate the information contained in the packets, wormhole attacks resist defensive measures solely based on cryptographic protocols.

Once the compromised node is falsely positioned, the network can become vulnerable to different exploits. For instance, the compromised node could inadvertently inject misleading information into the local network or obtain sensitive data from other nodes and flow them through the wormhole link. Another possibility for an adversary comes from the fact that the wormhole local node can be easily masqueraded as an authenticated local node by impersonating the compromised node; in this way, anyone who physically bears the wormhole local node could gain access to restricted areas or secret information [2].

The model of Figure 2, in spite of its simplicity, captures the essential mechanism of a wormhole attack to LD and ND protocols. Ironically, however, most existing wormhole detection schemes cannot cope with this simple attack for several reasons as follows.

(i) The simple scenario of Figure 2 assumes that in a normal situation (no attack), all the active nodes are neighbors; this precludes the use of secure LD or ND techniques solely based on connectivity information or hop counts. Obviously, methods for wormhole detection based on the analysis of "network layer" parameters (routes, traffic, etc.) are also inapplicable.

(ii) The compromised node only communicates with the remote wormhole node, so it cannot get cooperation from "real" neighbors in the localization process or the detection of the attack.

(iii) A wormhole attack is undetectable by "network-based" localization techniques [21]: if the position of the node is obtained from signals received by the anchors, the compromised node will be always located at the position of the wormhole local node. Therefore, the LD procedure should be performed at the unlocalized node, using data it received from the anchors, because the unlocalized node is the only one that can detect inconsistencies caused by a wormhole attack.

On the other hand, the model of Figure 2 is simple enough to allow the application of standard tools of statistical decision theory to the problems of node localization and wormhole detection.

As a wormhole attack challenges higher-level protocols, most effective procedures to detect such attacks are based on looking for inconsistencies in measurements performed at the physical layer level. In the next sections, we develop different detection strategies that analyze the RSS values measured by the nodes interacting in the localization procedure.

5. Wormhole Detection Using RSS: Parametric Approach

Any wormhole detection procedure can be stated as a binary hypothesis testing problem: given a vector of N RSS observations $\mathbf{r} = [r_1, r_2, \ldots, r_N]^T$, we must decide between hypothesis H_0 (no wormhole is present) and H_1 (a wormhole attack is active). However, to formalize the test we need a suitable statistical description of the observations. In the sequel, we will use the standard log-distance path-loss model [22] that links RSS values (in logarithmic scale) to distances as

$$r(\mathbf{x}, \mathbf{y}) = K - 10\alpha \log_{10} d(\mathbf{x}, \mathbf{y}) + e, \tag{4}$$

where $\mathbf{x}$ and $\mathbf{y}$ are the positions of the transmitter and receiver, respectively, K is the mean received power (in dBm) at unit distance, α is the path-loss exponent (which depends on the environment), and e is a zero-mean Gaussian random variable with standard deviation σ (in dB) that takes into account shadowing effects. Therefore, $r(\mathbf{x}, \mathbf{y})$ is also a Gaussian random variable with standard deviation σ and mean $\mu(\mathbf{x}, \mathbf{y})$, with

$$\mu(\mathbf{x}, \mathbf{y}) = K - 10\alpha \log_{10} d(\mathbf{x}, \mathbf{y}). \tag{5}$$

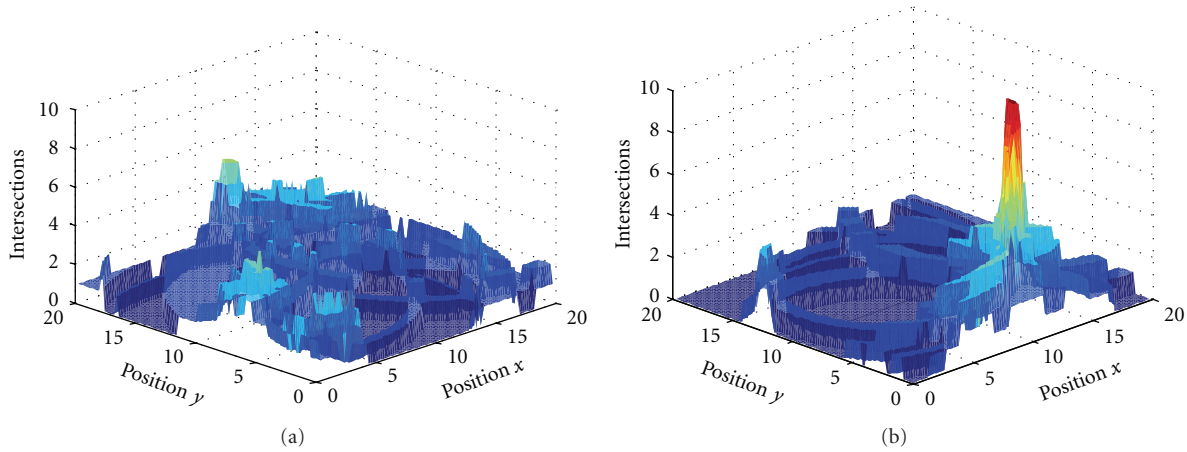


FIGURE 1: Spatial distributions of the number of intersecting rings with $N = 10$ anchors. (a) RSS measurements independent of distances. (b) RSS measurements inversely related to distances.

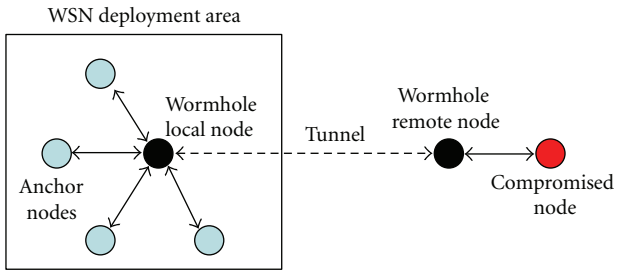


FIGURE 2: Wormhole attack to location and neighbor discovery.

Now, assuming that the observations are independent and identically distributed (IID), the distribution of $\mathbf{r}$ is multivariate normal with mean vector $\boldsymbol{\mu} = [\mu_1, \mu_2, \ldots, \mu_N]^T$ and covariance matrix $\sigma^2\mathbf{I}$, where $\mathbf{I}$ is the identity matrix, so that the joint probability density function (PDF) of the RSS measurements is

$$f(\mathbf{r};\boldsymbol{\mu}) = (2\pi\sigma^2)^{-N/2} \exp\left[-\frac{1}{2\sigma^2}(\mathbf{r}-\boldsymbol{\mu})^T(\mathbf{r}-\boldsymbol{\mu})\right], \quad (6)$$

where the superscript T denotes "transpose." The assumption of IID observations is valid whenever they are associated to transmitters and/or receivers located at different positions and shadow fading is spatially uncorrelated.

According to (5), the only parameter of (6) that depends on the position is $\boldsymbol{\mu}$, so we will formulate the wormhole detection problem as a test of the mean vector of $\mathbf{r}$:

$$\begin{aligned} H_0 &: \boldsymbol{\mu} = \boldsymbol{\mu}_0, \\ H_1 &: \boldsymbol{\mu} \neq \boldsymbol{\mu}_0, \end{aligned} \quad (7)$$

where $\boldsymbol{\mu}_0$ is determined assuming there is no wormhole present.

Now, depending on the origin of the measurements, we can define two different tests. The first one is carried out by the unlocalized node, which performs the localization and wormhole detection processes simultaneously, using RSS values obtained from packets supposedly transmitted by the anchors. The second strategy can be applied after the node is localized and is performed by the anchors, which analyze the RSS measurements obtained from packets supposedly transmitted by the localized node. Both schemes are presented in Sections 5.1 and 5.2, respectively.

5.1. Simultaneous Localization and Wormhole Detection: Likelihood Ratio Test. In this scheme, the anchors broadcast beaconing packets containing their positions, conveniently enciphered and authenticated to prevent other kinds of attacks. These packets are intended to be received by the unlocalized node, which measures their RSS values and then decrypts them to obtain the positions of the anchors. As stated previously, assuming that the statistical model for the RSS observations (6) is valid, then the wormhole detection procedure can be formulated as a hypothesis testing problem of the form (7), where the measurements $\mathbf{r} = [r_1, r_2, \ldots, r_N]^T$ are, in our case, collected by the unlocalized node.

Therefore, if the hypothesis H_0 (no wormhole) is true, then the observations are RSS values of packets transmitted by the anchors at known positions $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_N\}$ (see Figure 3(a)), so that we have the null hypothesis

$$H_0 : r_i = r(\mathbf{a}_i, \mathbf{p}), \quad i = 1, 2, \ldots, N \quad (8)$$

and according to (5), the elements of vector $\boldsymbol{\mu}_0$ in (7) are

$$\mu_{0,i} = K - 10\alpha \log_{10} d(\mathbf{a}_i, \mathbf{p}), \quad i = 1, 2, \ldots, N. \quad (9)$$

However, under H_1 (wormhole attack), the packets obtained by the unlocalized node come from the remote wormhole node, as shown in Figure 2; therefore, the RSS values for these packets will be totally unrelated to the anchors positions We will further assume that the remote wormhole node "randomizes" the observations (e.g., by changing its transmitted power) to avoid that they all take the same value and so circumvent a trivial detection; thus, the assumption of IID observations also holds true under H_1.

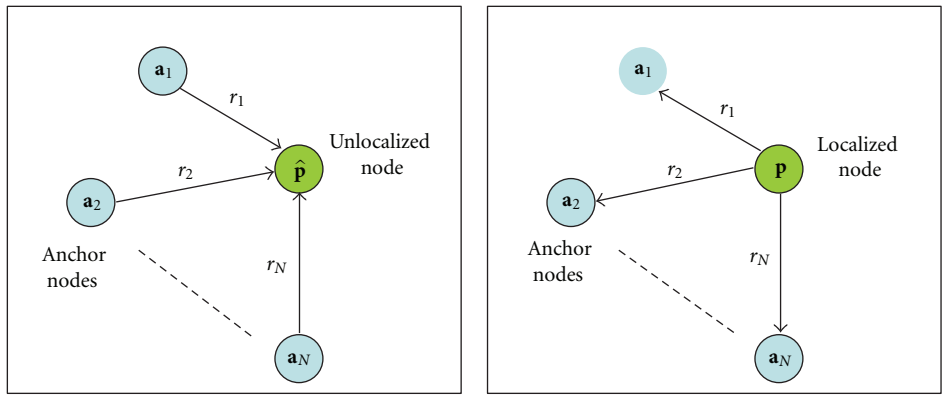


Figure 3: Scenarios for secure neighbor discovery assuming no wormhole is present. (a) Simultaneous localization and wormhole detection. (b) Wormhole detection after localization.

Notice that, as the position of the node $\mathbf{p}$ is unknown, both H_0 and H_1 of (7) are composite hypotheses. Therefore, we can obtain the likelihood ratio test (LRT) [23] as

$$\text{Decide } H_1 \,(\text{wormhole present}) \text{ iff } \Lambda(\mathbf{r}) > \eta, \tag{10}$$

where $\Lambda(\mathbf{r})$ is the likelihood ratio

$$\Lambda(\mathbf{r}) = \frac{\max_{\boldsymbol{\mu}} f(\mathbf{r};\boldsymbol{\mu})}{\max_{\boldsymbol{\mu}_0} f\left(\mathbf{r};\boldsymbol{\mu}_0\right)} \tag{11}$$

and η is a threshold selected so that we have a given probability of false alarm (PFA). Taking into account (6) and (9), we have

$$f\left(\mathbf{r};\boldsymbol{\mu}_0\right) = \left(2\pi\sigma^2\right)^{-N/2} \exp\left[-\frac{1}{2\sigma^2} V(\mathbf{p})\right] \tag{12}$$

with

$$V(\mathbf{p}) = \sum_{i=1}^{N} \left[r_i - K + 10\alpha \log_{10} d(\mathbf{a}_i, \mathbf{p})\right]^2. \tag{13}$$

The numerator of (11) is easily obtained, according to (6), as

$$\max_{\boldsymbol{\mu}} f(\mathbf{r};\boldsymbol{\mu}) = \left(2\pi\sigma^2\right)^{-N/2} \tag{14}$$

while the denominator of (11) is, according to (12),

$$\max_{\boldsymbol{\mu}_0} f\left(\mathbf{r};\boldsymbol{\mu}_0\right) = \left(2\pi\sigma^2\right)^{-N/2} \exp\left[-\frac{1}{2\sigma^2} V(\hat{\mathbf{p}})\right], \tag{15}$$

where $\hat{\mathbf{p}}$ is the maximum likelihood estimate (MLE) of $\mathbf{p}$ under H_0, defined as

$$\hat{\mathbf{p}} = \arg\max_{\mathbf{p}} f\left(\mathbf{r};\boldsymbol{\mu}_0\right). \tag{16}$$

Taking into account the inverse relationship between $f(\mathbf{r};\boldsymbol{\mu}_0)$ and $V(\mathbf{p})$, (16) can be also expressed as

$$\hat{\mathbf{p}} = \arg\min_{\mathbf{p}} V(\mathbf{p}) \tag{17}$$

so that $\hat{\mathbf{p}}$ is obtained as the solution of a nonlinear least square (NLS) problem. Finding the global solution of (17) is, in general, a difficult optimization problem because of the existence of multiple local minima in the objective function. Therefore, it is customary to resort to simpler suboptimal alternatives to the exact MLE that guarantee a single local minimum [24, 25].

Now, taking into account (11), (14), and (15), we can compute the logarithm of the likelihood ratio as

$$\ln \Lambda(\mathbf{r}) = \frac{V(\hat{\mathbf{p}})}{2\sigma^2} \tag{18}$$

so that a test equivalent to (10) is

$$\text{Decide } H_1 \text{ iff } V(\hat{\mathbf{p}}) > \eta', \tag{19}$$

where η' is another suitable threshold, selected so that

$$P\left[V(\hat{\mathbf{p}}) > \eta' \mid H_0\right] = P_{\text{FA}} \tag{20}$$

with P_{FA} the probability of false alarm. The LRT is summarized in Algorithm 1.

We can see from (13) that $V(\hat{\mathbf{p}})$ is the sum of the squared residuals, so it represents a measure of the "quality" of the MLE $\hat{\mathbf{p}}$.

5.2. Wormhole Detection after Localization: Likelihood Ratio Test. Another wormhole detection strategy could be implemented after a given node has completed the localization procedure, and as a result of this, it has obtained a position within the local network deployment area. The idea now is to use the anchor nodes to check the validity of the node location.

To accomplish this, the localized node broadcasts cryptographically secured packets containing its position $\mathbf{p}$ to be verified. These packets are received by the anchors, which use them to obtain RSS measurements and the declared node position. So, in this case, the observations $\mathbf{r} = [r_1, r_2, \ldots, r_N]^T$ are collected by the anchors and under H_0 (no wormhole), correspond to the RSS values of packets

Inputs:
Set of trustworthy anchor positions: $\{\mathbf{a}_i, i = 1, 2, \ldots, N\}$
Set of untrustworthy anchor to node RSSs: $\{r_i = r(\mathbf{a}_i, \mathbf{p}), i = 1, 2, \ldots, N\}$
Parameters of the path-loss model: K and α.
Detection threshold: η'
Steps:
(1) Obtain $\hat{\mathbf{p}}$ as the maximum likelihood estimate (MLE) of the position of the node using (17) and (13).
(2) Compute the test statistic $V(\hat{\mathbf{p}})$ using (13).
(3) **if** $V(\hat{\mathbf{p}}) > \eta'$ then
(4) **set** *wormhole_flag* ←*true*
(5) **else**
(6) **set** *wormhole_flag* ← *false*
(7) **end if**
(8) **return** *wormhole_flag* and estimated position $\hat{\mathbf{p}}$

Algorithm 1: Simultaneous localization and wormhole detection. Parametric approach: likelihood ratio test.

transmitted by the node at position $\mathbf{p}$ and received by the anchors at positions $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_N\}$ (see Figure 3(b)). Therefore, we have the null hypothesis

$$H_0 : r_i = r(\mathbf{p}, \mathbf{a}_i), \quad i = 1, 2, \ldots, N, \tag{21}$$

and according to (5) and taking into account that $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$, for all $\mathbf{x}, \mathbf{y}$, the elements of vector $\boldsymbol{\mu}_0$ are also given by (9).

On the other hand, under H_1 (wormhole attack), the packets received by the anchors are transmitted by the local wormhole node, as shown in Figure 2; therefore, the RSS values for these packets will be unrelated to the declared position of the compromised node $\mathbf{p}$.

Therefore, the only difference with the previous case is that now the position of the node $\mathbf{p}$ is known, so H_0 is a simple hypothesis and the likelihood ratio is

$$\Lambda(\mathbf{r}) = \frac{\max_{\boldsymbol{\mu}} f(\mathbf{r}; \boldsymbol{\mu})}{f\left(\mathbf{r}; \boldsymbol{\mu}_0\right)}, \tag{22}$$

where $f(\mathbf{r}; \boldsymbol{\mu})$ and $f(\mathbf{r}; \boldsymbol{\mu}_0)$ are given by (6) and (12), respectively.

Following analogous steps to the previous section, we arrive at a test similar to (19) but using the reported position instead of the MLE (see Algorithm 2):

$$\text{Decide } H_1 \text{ iff } V(\mathbf{p}) > \eta'', \tag{23}$$

where $V(\mathbf{p})$ was defined in (13) and η'' is chosen so that

$$P[V(\mathbf{p}) > \eta'' \mid H_0] = P_{\text{FA}}. \tag{24}$$

Again, the test statistic $V(\mathbf{p})$ is a measure of "goodness of fit" of the declared position to the observations.

6. Wormhole Detection Using RSS: Nonparametric Approach

The detection strategies of Section 5 assume the existence of a well-defined measurement model that describes the statistical relationship between observed RSS values and distances. However, in most instances, such model can only be stated under idealized conditions or is tied to a specific scenario; in this latter case, estimating its parameters often requires a costly calibration phase which must be repeated every time the environmental conditions change.

Therefore, it would be desirable to devise wormhole detection procedures that are "nonparametric" in the sense that unlike the test (7), these strategies do not impose a particular distribution for the observations; thus, such tests will be robust against departures from any predefined model. In particular, we will base our derivations of nonparametric detection schemes on the underlying ideas of the range-free positioning techniques described in Section 3.

As above, depending on the source of the measurements, we will derive a procedure for simultaneous localization and wormhole detection performed by the unlocalized node, using RSS values obtained from packets transmitted by the anchors, and a postlocalization wormhole detection scheme performed by the anchors, employing RSS measurements obtained from packets transmitted by the localized node. Both schemes are presented in Sections 6.1 and 6.2, respectively.

6.1. Simultaneous Localization and Wormhole Detection: DWARFLoc. We can check the presence of a wormhole without assuming any specific model for the observations by exploiting the fact that under no attack, the RSS values collected by the unlocalized node will be related to the distances from the node to the anchors, no matter which is the exact form of this relationship; on the other hand, if a wormhole is present, the RSS values measured by the compromised node are totally unrelated to its actual position.

Thus, under a wormhole attack and assuming that the compromised node uses the ROCRSSI scheme described in Section 3 to localize itself, it is very unlikely for the rings provided by the anchor nodes to share a common intersection, even in the absence of measurement errors; so, if a voting strategy is adopted to estimate the unknown node position, the number of votes received by any region in the plane will be well below the maximum attainable score (see

Inputs:
Set of trustworthy anchor positions: $\{\mathbf{a}_i, i = 1, 2, \ldots, N\}$
Untrustworthy node position: $\mathbf{p}$
Set of untrustworthy node to anchor RSSs: $\{r_i = r(\mathbf{p}, \mathbf{a}_i), i = 1, 2, \ldots, N\}$
Parameters of the path-loss model: K and α
Detection threshold: η''
Steps:
(1) Obtain the anchor to node distances $\{d(\mathbf{a}_i, \mathbf{p}), i = 1, 2, \ldots, N\}$.
(2) Compute the test statistic $V(\mathbf{p})$ using (13).
(3) **if** $V(\mathbf{p}) > \eta''$ then
(4) **set** *wormhole_flag* ← *true*
(5) **else**
(6) **set** *wormhole_flag* ← *false*
(7) **end if**
(8) **return** *wormhole_flag*

Algorithm 2: Detection after localization. Parametric approach: likelihood ratio test.

Figure 1(a)). On the other hand, if no wormhole is present, we should expect that most anchors agree on the existence of a region of the plane that satisfies the set of constraints (2); this region, therefore, will receive a high number of votes (relative to the number of anchors), as Figure 1(b) illustrates. For these reasons, the test statistic proposed for this nonparametric detection strategy is the deviation of the maximum number of votes attained by any region of the plane from the average number of votes.

Therefore, in this scheme the anchor nodes broadcast beaconing packets that contain their positions and the RSSs they measure for packets transmitted by other anchor nodes; such packets should be conveniently enciphered and authenticated. Then, the unlocalized node collects and decrypts the beaconing packets and computes RSS values for them (see Figure 3(a)); these measurements, along with the positions of the anchors and the anchor-to-anchor RSSs, are used to estimate the position of the node, via the ROCRSSI method. The quality of the estimated position is determined by the number of votes it received, and if this number (after mean centering) is above a predefined threshold, the localization process is considered valid; otherwise, an attack is presumed and the unlocated node refrains from joining the network. As usual, the detection threshold is selected to obtain a given PFA. The whole DWARFLoc procedure is described in Algorithm 3.

6.2. Wormhole Detection after Localization: DWARFTest. Once the node is successfully located, we can proceed to verify the validity of the node position $\mathbf{p}$ by reversing the previous roles of the tested node and the anchor nodes (see Figure 3(b)): now the former broadcasts packets containing its estimated location, while the latter receive these transmissions, compute RSS values, and use them to look for possible violations of the monotonicity constraint (1). If the tested node has been compromised by a wormhole attack like that of Figure 2, the source of those packets will be the wormhole local node, whose position is, with a high probability, different from that reported by the compromised node, so that many of the anchor nodes will find that the measured RSSs do not agree with the expected ones. Obviously, beside the anchor nodes, any other node whose position has been previously validated can also participate in this wormhole detection procedure. Notice also that the RSS values collected by the anchors should be transmitted to a central node in order to process them.

As a measure of dissimilarity between distances and RSS measurements, we have used a slight modification of the classical Kendall tau distance [26], which is a metric that counts the number of pairwise disagreements between two lists. In our case, the test statistic counts the number of violations of the monotonicity constraint (1) for every possible pair of node-to-anchor distances and their corresponding measured RSS values as

$$\tau(\mathbf{p}) = \Big|\Big\{(i,j), i<j : \Big(d(\mathbf{p},\mathbf{a}_i) < d\Big(\mathbf{p},\mathbf{a}_j\Big) \wedge r(\mathbf{p},\mathbf{a}_i) < r\Big(\mathbf{p},\mathbf{a}_j\Big)\Big) \vee \Big(d(\mathbf{p},\mathbf{a}_i) > d\Big(\mathbf{p},\mathbf{a}_j\Big) \wedge r(\mathbf{p},\mathbf{a}_i) > r\Big(\mathbf{p},\mathbf{a}_j\Big)\Big)\Big\}\Big|, \tag{25}$$

where $|S|$ denotes the cardinal number of a set S.

As the test statistic $\tau(\mathbf{p})$ is a discrete random variable (it only takes integer values), the decision procedure should include two parameters to exactly obtain a predefined PFA: an integer detection threshold η and a real number γ ($0 \leq \gamma \leq 1$), such that

$$P[\tau(\mathbf{p}) > \eta \mid H_0] + \gamma P[\tau(\mathbf{p}) = \eta \mid H_0] = P_{\mathrm{FA}}, \tag{26}$$

where P_{FA} is the desired probability of false alarm. The steps to implement the DWARFTest procedure are illustrated in Algorithm 4.

7. Simulation Results

We have conducted some simulations to evaluate and compare the performance of the wormhole detection strategies described in Sections 5 and 6. The simulated WSN is composed of a set of anchor nodes whose positions are uniformly distributed in a square room of 20 m × 20 m.

Inputs:
Set of anchor positions: $\{\mathbf{a}_i, i = 1, 2, \ldots, N\}$
Set of untrustworthy anchor to node RSSs: $\{r(\mathbf{a}_i, \mathbf{p}), i = 1, 2, \ldots, N\}$
Set of trustworthy anchor to anchor RSSs: $\{r(\mathbf{a}_i, \mathbf{a}_j), i = 1, 2, \ldots, N; j = 1, 2, \ldots, N; i \neq j\}$
Detection threshold: η
Steps:
(1) Define a grid $\mathbf{G}$ of L points in the plane, covering the WSN deployment region and an array $\mathbf{V}$ of L counters.
(2) **set** $\mathbf{V} \leftarrow \mathbf{0}$
(3) **for** every anchor $\mathbf{a}_i, i = 1, 2, \ldots, N$ **do**
(4) Obtain a ring $R(\mathbf{a}_i, \rho_1^{(i)}, \rho_2^{(i)})$ of the form (2) that should ideally contain the node position, using (3)
(5) **for** every point of the grid $\mathbf{g} \in \mathbf{G}$ **do**
(6) **if** $\mathbf{g} \in R(\mathbf{a}_i, r_1^{(i)}, r_2^{(i)})$ **then**
(7) Increment counter of votes for point $\mathbf{g}$: $\mathbf{V}(\mathbf{g}) \leftarrow \mathbf{V}(\mathbf{g}) + 1$
(8) **end if**
(9) **end for**
(10) **end for**
(11) Obtain the intersection region as the set of grid points with maximum number of "votes":

$$v_M = \max_{\mathbf{g} \in \mathbf{G}} \mathbf{V}(\mathbf{g})$$
$$\mathbf{M} = \{\mathbf{g} \in \mathbf{G} : \mathbf{V}(\mathbf{g}) == v_M\}$$

(12) Estimate the position of the node as the centroid of the intersection area:

$$\hat{\mathbf{p}} = \frac{\mathbf{1}}{|\mathbf{M}|} \sum_{\mathbf{g} \in \mathbf{M}} \mathbf{g}$$

(13) Compute the sample mean of the number of votes:

$$\overline{v} = \frac{1}{L} \sum_{\mathbf{g} \in \mathbf{G}} \mathbf{V}(\mathbf{g})$$

(14) **if** $v_M - \overline{v} \leq \eta$ then
(15) **set** *wormhole_flag* $\leftarrow$ *true*
(16) **else**
(17) **set** *wormhole_flag* $\leftarrow$ *false*
(18) **end if**
(19) **return** *wormhole_flag* and estimated position $\hat{\mathbf{p}}$

Algorithm 3: Simultaneous localization and wormhole detection. Nonparametric approach: DWARFLoc.

Inputs:
Set of trustworthy anchor positions: $\{\mathbf{a}_i, i = 1, 2, \ldots, N\}$
Untrustworthy node position: $\mathbf{p}$
Set of untrustworthy node to anchor RSSs: $\{r(\mathbf{p}, \mathbf{a}_i), i = 1, 2, \ldots, N\}$
Detection threshold and "PFA adjustment" parameter: η, γ
Steps:
(1) Obtain the node to anchor distances $\{d(\mathbf{p}, \mathbf{a}_i), i = 1, 2, \ldots, N\}$.
(2) Compute the test statistic $\tau(\mathbf{p})$, using (25)
(3) **if** $\tau(\mathbf{p}) > \eta$ then
(4) **set** *wormhole_flag* $\leftarrow$ *true*
(5) **else if** $\tau(\mathbf{p}) = \eta$
(6) **set** *wormhole_flag* $\leftarrow$ *true* with probability γ
(7) **else**
(8) **set** *wormhole_flag* $\leftarrow$ *true*
(9) **end if**
(10) **return** *wormhole_flag*

Algorithm 4: Wormhole detection after localization. Nonparametric approach: DWARFTest.

For RSS values, we have assumed the log-distance path-loss model (4) for which we set $\alpha = 3$ as a typical value for indoor environments.

The range-free localization scheme uses a square grid of 20×20 elements, which implies a spatial resolution of 1 m in the proposed scenario. The range-based (parametric) approach uses as an approximation for the MLE the best linear unbiased estimator (BLUE) of the node position, because it is much simpler to implement than the exact MLE and its variance is close to the Cramér-Rao lower bound [25].

A wormhole attack is simulated according to the model of Figure 2. The distance between the wormhole remote node and the compromised node, $d(\mathbf{w}, \mathbf{c})$, is randomly chosen, and both nodes are assumed to be located beyond the radio range of any other WSN node. To avoid a trivial detection, the remote wormhole node performs random changes in its transmitted power, so that the RSS values measured by the compromised node are obtained as

$$r_i^c = K - 10\alpha \log_{10} d(\mathbf{w}, \mathbf{c}) + e + u_i, \quad i = 1, 2, \ldots, N, \tag{27}$$

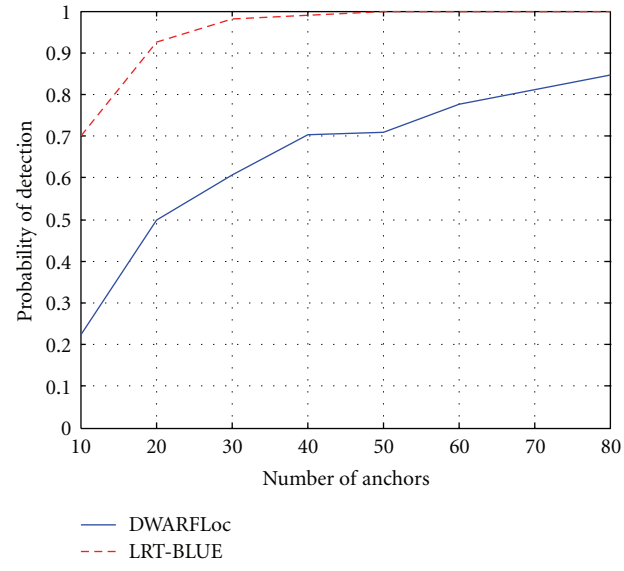
where $d(\mathbf{w}, \mathbf{c})$ is uniformly distributed between 0 and 20 m, e is a zero-mean Gaussian random variable with standard deviation σ, and $\{u_i, i = 1, 2, \ldots, N\}$ are IID random variables with uniform distribution in the interval $(-6, 6)$. These RSSs are first processed by the simultaneous detection and localization schemes of Sections 5.1 and 6.1.

Once the node has been located, the detection procedures of Sections 5.2 and 6.2 are started and the tested node begins to broadcast its estimated position. However, according to Figure 2, if this node has been compromised by a wormhole attack, the RSS values measured by the anchors are related to their distances to the wormhole local node, because this node is acting as a repeater.

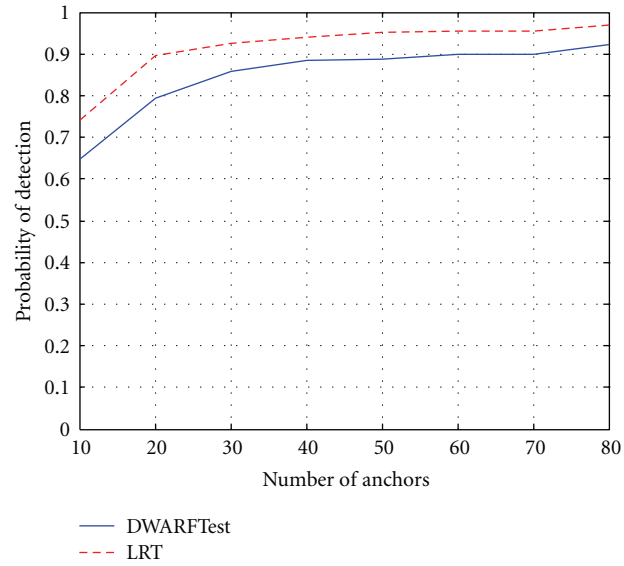
To determine the detection thresholds for the tests, we have also simulated the scenarios of Figure 3, using a reference node whose position is uniformly distributed in the WSN deployment area. Then, for each of the four tests, the empirical cumulative distribution function (CDF) of the test statistic is used to obtain the critical value that ensures a given PFA.

Some results are represented in Figures 4 and 5, where we have plotted the attained probability of detection for the wormhole detection schemes of Sections 5 and 6 under different situations. The PFA is fixed at 0.05 and we conducted 1000 simulation runs in all cases.

By examining Figures 4(a) and 5(a), we can observe that the parametric approach for simultaneous wormhole detection and localization (LRT-BLUE) performs clearly better than the range-free procedure (DWARFLoc); this was expected, because range-free localization methods do not use *a priori* information about any model for the RSS observed values. However, we can see form Figures 4(b) and 5(b) that the range-free version of the scheme for detection after localization (DWARFTest) competes in performance with its parametric counterpart (LRT) and even surpasses it for high values of the path-loss standard deviation; this is attributable to the rapid degradation of the BLUE estimator when the RSS measurements are subject to significant errors.



(a)



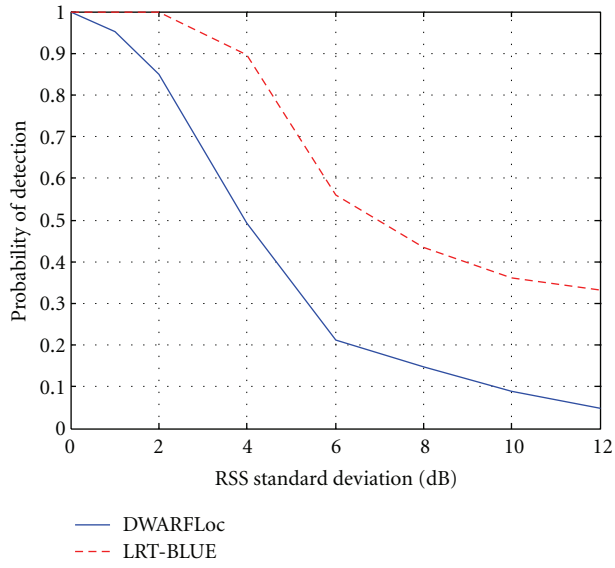
(b)

FIGURE 4: Probability of wormhole detection for the proposed strategies with varying number of anchor nodes ($P_{FA} = 0.05$ and $\sigma = 3$ dB). (a) Simultaneous localization and detection. (b) Detection after localization.

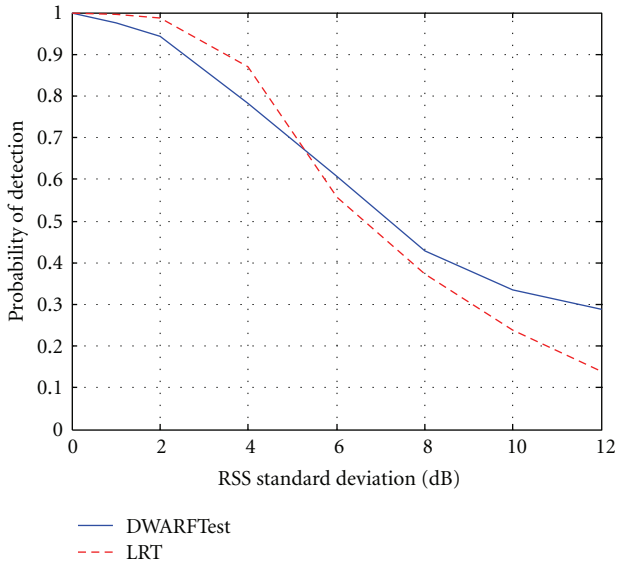
8. Conclusions

In this paper we presented a minimalist model for a wormhole attack to a WSN that can be effectively counteracted by two different detection procedures, based on the underlying ideas of RSS-based range-free localization methods. The first one (DWARFLoc) operates simultaneously with the localization procedure, and the second one (DWARFTest) is a postlocalization detector that tries to validate *a posteriori* the estimated node position. Simulations suggest that DWARFTest has much better detection performance than DWARFLoc but requires more transmissions to be carried out.

Furthermore, assuming that the RSS values follow the standard log-normal path-loss model, we have also derived exact likelihood ratio tests for the detection of a wormhole, which can be used as benchmarks for any other detection scheme.



(a)



(b)

FIGURE 5: Probability of wormhole detection for the proposed strategies with varying path-loss standard deviation ($P_{FA} = 0.05$ and $N = 40$). (a) Simultaneous localization and detection. (b) Detection after localization.

Acknowledgments

This research was partially supported by the Spanish Ministry of Science and Innovation under Grant TEC2009-14219-C03 (AMURA) and the European Commission under Grant FP7-ICT-2009-4-248894 (WHERE2).

References

[1] X. Du and H. H. Chen, "Security in wireless sensor networks," *IEEE Wireless Communications*, vol. 15, no. 4, pp. 60–66, 2008.

[2] P. Papadimitratos, M. Poturalski, P. Schaller et al., "Secure neighborhood discovery: a fundamental element for mobile ad hoc networking," *IEEE Communications Magazine*, vol. 46, no. 2, pp. 132–139, 2008.

[3] Y. C. Hu and A. Perrig, "Wormhole attacks in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 370–379, 2006.

[4] C. Liu, K. Wu, and T. He, "Sensor localization with ring overlapping based on comparison of received signal strength indicator," in *Proceedings of IEEE International Conference on Mobile Ad-Hoc and Sensor Systems*, pp. 516–518, October 2004.

[5] R. Maheshwari, J. Gao, and S. R. Das, "Detecting wormhole attacks in wireless networks using connectivity information," in *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM '07)*, pp. 107–115, May 2007.

[6] T. Giannetsos, T. Dimitriou, and N. R. Prasad, "State of the art on defenses against wormhole attacks in wireless sensor networks," in *Proceedings of the 1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace and Electronic Systems Technology (Wireless VITAE '09)*, pp. 313–318, May 2009.

[7] W. Wang and B. Bhargava, "Visualization of wormholes in sensor networks," in *Proceedings of the 3rd ACM Workshop on Wireless Security (WiSe '04)*, pp. 51–60, October 2004.

[8] Y. Zhang, W. Liu, W. Lou, and Y. Fang, "Location-based compromise-tolerant security mechanisms for wireless sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 247–260, 2006.

[9] J. Wu, H. Chen, W. Lou, Z. Wang, and Z. Waang, "Label-based DV-Hop localization againstwormhole attacks in wireless sensor networks," in *Proceedings of the 5th IEEE International Conference on Networking, Architecture and Storage (NAS '10)*, pp. 79–88, July 2010.

[10] L. Lazos, R. Poovendran, and S. Čapkun, "ROPE: Robust position estimation in wireless sensor networks," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, pp. 324–331, April 2005.

[11] S. Čapkun, "Secure positioning in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 221–232, 2006.

[12] L. Lazos and R. Poovendran, "SeRLoc: secure range-independent localization for wireless sensor networks," in *Proceedings of the 2004 ACM Workshop on Wireless Security (WiSe '04)*, pp. 21–30, October 2004.

[13] L. Lazos and R. Poovendran, "HiRLoc: high-resolution robust localization for wireless sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 233–246, 2006.

[14] Y. Niu, D. Gao, S. Gao, and P. Chen, "A robust localization in wireless sensor networks against wormhole attack," *Journal of Networks*, vol. 7, no. 1, pp. 187–194, 2012.

[15] R. Stoleru, H. Wu, and H. Chenji, "Secure neighbor discovery and wormhole localization in mobile ad hoc networks," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1179–1190, 2012.

[16] N. Patwari, A. O. Hero, M. Perkins, N. S. Correal, and R. J. O'Dea, "Relative location estimation in wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 51, no. 8, pp. 2137–2148, 2003.

[17] F. Gustafsson and F. Gunnarsson, "Mobile positioning using wireless networks: possibilities and fundamental limitations based on available wireless network measurements," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 41–53, 2005.

[18] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-free localization and its impact on large scale

sensor networks," *ACM Transactions on Embedded Computing Systems*, vol. 4, no. 4, pp. 877–906, 2005.

[19] K. Wu, C. Liu, J. Pan, and D. Huang, "Robust range-free localization in wireless sensor networks," *Mobile Networks and Applications*, vol. 12, no. 5-6, pp. 392–405, 2007.

[20] M. García-Otero, T. Zahariadis, F. Álvarez et al., "Secure geographic routing in ad hoc and wireless sensor networks," *Eurasip Journal on Wireless Communications and Networking*, vol. 2010, Article ID 975607, pp. 1–12, 2010.

[21] A. H. Sayed, A. Tarighat, and N. Khajehnouri, "Network-based wireless location: Challenges faced in developing techniques for accurate wireless location information," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 24–40, 2005.

[22] T. S. Rappaport, *Wireless Communications, Principles and Practice*, Prentice Hall, 2nd edition, 2002.

[23] S. M. Kay, *Fundamentals of Statistical Signal Processing, Volume II: Detection Theory*, Prentice Hall, 1998.

[24] P. Tarrío, A. M. Bernardos, J. A. Besada, and J. R. Casar, "A new positioning technique for RSS-based localization based on a weighted least squares estimator," in *Proceedings of IEEE International Symposium on Wireless Communication Systems (ISWCS '08)*, pp. 633–637, October 2008.

[25] L. Lin and H. C. So, "Best linear unbiased estimator algorithm for received signal strength based localization," in *Proceedings of the 19th European Signal Processing Conference (EUSIPCO '11)*, pp. 1989–1993, August 2011.

[26] M. Kendall and A. Stuart, *The Advanced Theory of Statistics*, vol. 2, Charles Griffin, 1979.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 732381, 11 pages
doi:10.1155/2012/732381

Research Article

Secure Localization in Wireless Sensor Networks with Mobile Beacons

Ting Zhang,[1] Jingsha He,[2] and Hong Yu[1]

[1] *College of Computer Science, Beijing University of Technology, Beijing 100124, China*
[2] *School of Software Engineering, Beijing University of Technology, Beijing 100124, China*

Correspondence should be addressed to Ting Zhang, zhangting06@emails.bjut.edu.cn

Received 6 July 2012; Revised 4 September 2012; Accepted 4 September 2012

Academic Editor: An Liu

We present a scheme, called SLMB, for secure sensor localization in WSNs in which we propose to use a mobile beacon node with the goal of reducing the overall energy consumption in sensor nodes during sensor localization. In the SLMB scheme, a mobile beacon node traverses through the network, collects information from unknown sensor nodes, figures out position relationship with these nodes, and sends the information to the base station where analysis and location calculation is carried out to relieve unknown sensor nodes from energy-consuming computation. The proposed SLMB scheme is also designed to resist wormhole attacks, and localization is developed based on a mathematical model to design a path for the mobile beacon node to traverse in order to cover the entire sensor network. To evaluate our scheme, we have performed simulations to demonstrate that the SLMB scheme can improve the success rate and the accuracy of sensor localization compared to other sensor localization schemes in hostile environments. Our simulation results also show that the SLMB scheme consumes much less energy than traditional distributed sensor localization schemes, which is an important metric in measuring the effectiveness and usefulness of any schemes targeted for applications in WSNs.

1. Introduction

Sensor localization in wireless sensor networks (WSNs) is a fundamental technical issue, for it is critical for monitoring applications and for most location-based routing protocols and services. Therefore, in recent years, sensor localization has generated a great deal of interest in which researchers have considered various technical issues, such as efficiency [1], accuracy [2], and security [3] during sensor localization. Methods for the localization of wireless sensor nodes are generally classified into two types: range-based localization and range-free localization. The first type includes schemes in which positions of the unknown sensor nodes are calculated using measurement means to derive relevant information about distances and angles between sensor nodes [4]. The second type includes schemes in which positions of the unknown sensor nodes are estimated using connectivity information as well as multihop routing information to derive relevant information between sensor nodes [5].

In real applications, however, there may be other types of localization methods owing to different application scenarios. Therefore, specific localization methods in real applications need to be continuously developed and improved based on orientation methods in order to adapt basic localization schemes to different network scenarios. Consequently, in order to develop effective sensor localization methods, we should analyze and understand the main characteristics of specific networks and develop proper performance metrics that can be used to measure the performance of localization schemes. Meanwhile, we should also consider the main constraints of wireless sensor networks such as constrained energy supply of the sensor nodes as well as the complexity of network environments in the development of effective localization methods.

Most existing localization algorithms, whether range-free or range-based, are distributed in nature in which unknown sensor nodes need to get position information about nearby beacon nodes so that they can calculate their own positions. The calculated position results are then sent to the base station or a central server to be used in real applications. One major drawback of such distributed localization algorithms is that it makes energy-constrained unknown sensor nodes

bear all the responsibility of communication and computation, resulting in high energy consumption in the sensor nodes. Another problem with such distributed algorithms is the increased security risks due to frequent communication between the sensor nodes.

In this paper, we propose a secure centralized sensor localization scheme by using a mobile beacon node (SLMB) to address the above-mentioned critical issues for WSNs and by developing some secure mechanisms to resist wormhole attacks in sensor localization. The proposed SLMB scheme has the following general features.

(1) It uses a mobile beacon node to travel along a calculated path in the network to collect information about the position relationship with nearby unknown sensor nodes. The collected information is then sent to the base station where the positions of the unknown sensor nodes are calculated, which can greatly lower the communication cost for the unknown sensor nodes.

(2) It takes a centralized approach so as to reduce the amount of calculation in the unknown sensor nodes by transferring the calculation work to the base station, which is a node in the network that is considered to be free from resource constraint.

(3) It calculates a reasonable mobile path for the beacon node to traverse so as to cover the entire network to ensure that every unknown sensor node can get connected to the mobile beacon node at some point of time so that necessary information can be collected for position calculation. The development of the mobile path follows the design principle of the cellular network and includes a quantitative method for the determination of efficient and necessary points for the mobile beacon node to visit for information collection.

(4) It includes some secure mechanisms to fight against wormhole attacks, thus improving the security of the centralized sensor localization algorithm in general.

The rest of this paper is organized as follows. In Section 2, we review some related work on sensor localization in WSNs. In Section 3, we present our centralized sensor localization scheme and describe some implementation and application issues. In Section 4, we describe the experiment we have performed to evaluate the proposed SLMB scheme and show some favorable simulation results in comparison to other localization methods. Finally, in Section 5, we conclude this paper in which we also discuss some future work.

2. Related Work

Existing sensor localization schemes can be generally classified into two types, that is, distributed and centralized schemes based on where calculation of sensor positions is performed in the localization process. In distributed localization, the unknown sensor nodes collect position information about nearby beacon nodes and calculate their own coordinates by themselves [6, 7]. That is, unknown sensor nodes are responsible for position calculation. In contrast, in centralized localization, beacon nodes collect the position information about unknown sensor nodes and send the information to the base station for data integration and position calculation [8, 9]. That is, the base station is responsible for position calculation.

Although distributed localization schemes have been widely popular, since in most WSNs, the number of beacon nodes is usually too limited, and the status of such nodes is too static to meet the needs of large WSNs. For these reasons, if an unknown sensor node wants to use beacon information more effectively, it may need to get the beacon information through multihop data transmission. In [10], the authors proposed a self-positioning algorithm that can run efficiently and independently at individual sensor nodes based on locally collected information. However, the requirement on the distance measurement error is quite strict. In [11], the authors proposed an algorithm and showed that even when only the connectivity information was given, the Euclidean distance between the estimated and the correct position of every unknown sensor node can be bounded and would decay at a rate that is inversely proportional to the radio range. However, this scheme incurs a larger amount of calculation in the unknown sensor nodes. In [12], the authors proposed a classic distributed localization scheme called DV-Hop based on distance vector routing. In DV-Hop, each unknown node needs to get the hop-count to the beacon nodes which estimate the average size for one hop between nodes in the network. Then the unknown nodes calculate their positions using the obtained information about the distances between the beacon nodes and themselves. DV-Hop can provide approximate positions for the nodes in a network where only a small fraction of nodes have self-positioning capability, but it requires more message exchanges between nodes in the network.

Due to energy constraint and thus limited life of sensor nodes, many researchers have proposed some centralized localization methods to reduce energy consumption through lowering computation and communication cost for the sensor nodes. Such localization approaches can bring significant benefits to applications, for it can extend the life of sensor nodes since most computations will now be completed at the central server or base station. In [13], the authors presented a multihop localization technique for WSNs by exploiting the strength indications of received signals. The proposed scheme aims at providing a solution for the localization of sensor nodes in static WSNs. In [14], the authors made some major modifications to improve the performance of the simulated annealing-based localization algorithm to increase localization accuracy. However, this type of localization schemes requires a large number of beacon nodes and involves complicated localization algorithms in order to complete the localization of all unknown sensor nodes in the network.

In order to overcome the shortcomings of requiring a large number of beacon nodes, in [15], some schemes based on mobile beacon nodes were proposed to transfer beacon information to help unknown sensor nodes in

performing self-localization. The problem is that some of the methods cannot be easily integrated into the centralized framework and some others lack methods for concise calculation of effective mobile beacon path. In [16], the authors demonstrated a range-free localization mechanism based on the location information from mobile beacons and on the principles of elementary geometry. But all the position calculation is still completed in the unknown sensor nodes, which makes it more like a distributed localization scheme. In [17], the authors proposed a novel mobile beacon-assisted localization algorithm based on network density clustering for WSNs by combining node clustering, incremental localization, and mobile beacon assisting together. Although this scheme is suitable for clustering large networks, it may not be suitable for networks that require faster convergence.

Some more research work has also be carried out to address security issues in sensor localization for WSNs. In [18], the authors improved the security and accuracy of sensor localization using location-based key distribution. In [19], the authors presented a novel defense mechanism against attacks in the DV-Hop localization algorithm. However, the security mechanisms proposed in these algorithms are not applicable to the mobile beacon scenario in WSNs.

In this paper, we introduce a mobile beacon node into centralized localization while improving the security of the scheme. In order to keep the computation cost and therefore the energy cost low for the sensor nodes, we propose a specific centralized localization scheme in which the mobile beacon node traverses the entire network following a well-designed path during which it stops at every collection point to collect position information from nearby unknown sensor nodes before moving to the next collection point. The mobile beacon node sends a position request at each collection point to nearby unknown sensor nodes, estimates the position relationship to these unknown sensor nodes based on received information and sends the information along with its own current position to the base station. It is the base station that will eventually complete all the position calculation.

The proposed SLMB scheme has the following obvious advantages.

(1) It can balance energy consumption of sensor nodes in the network, for it would prevent the sensor nodes that are closer to the base station from consuming excessive energy to deliver position information to the base station from far away sensor nodes in a multihop manner.

(2) It can improve localization accuracy as well as success rate compared to other similar schemes.

(3) It can improve the security of localization since securing only the beacon node should be much easier than securing a large number of unknown sensor nodes in the network.

(4) It can effectively reduce the communication overhead for the sensor nodes and overall transmission delay.

3. The Proposed Scheme

3.1. The Network Model. There are three types of nodes in the network model for our SLMB scheme. The first type includes the base station, which is capable of managing and integrating data for the entire network including the calculation of positions of unknown sensor nodes and the application of the results in real applications. The second type includes the mobile beacon nodes, which is capable of positioning themselves, traversing the network to collect information from unknown sensor nodes, and transmitting the collected information to the base station for position calculation. In addition, beacon nodes are mobile nodes that are assumed to have unlimited energy supply. The third type includes the unknown sensor nodes whose positions or locations in the network need to be determined through calculation based on collected information.

3.2. The Localization Model. The scheme that we propose is appropriate for applications and networks in which there is not enough stationary beacon nodes as position references for the unknown sensor nodes but localization still needs to be finished in time. In the proposed SLMB scheme, the information about the distribution of the unknown sensor nodes in the network can be obtained by using a mobile beacon node, and the positions of the unknown sensor nodes can be calculated quickly by the base station. In addition, in the SLMB scheme, we use a mathematical model to make the mobile beacon node follow a designated path to cover the entire network so as to improve the effectiveness and efficiency of sensor localization.

Following are the main steps of our centralized sensor localization algorithm, that is, the SLMB scheme.

(1) The mobile beacon node moves along a calculated path, sending position requests at every collection point to nearby unknown sensor nodes, collecting responses from unknown sensor nodes, and sending the collected information along with its current position to the base station.

(2) The mobile beacon node moves to the next collection point after completing the work at a previous collection point until it completes the traversal of the whole path to cover the entire network. The mobile beacon node can decide to aggregate information collected at more than one collection point before sending the collected information to the base station to further improve the performance of communication although energy consumption is not an issue under consideration.

(3) The base station integrates all the information received from the mobile beacon node and calculates the positions of all the unknown sensor nodes.

3.3. The Mobile Path Model. Mobility of the beacon node is required in our SLMB scheme. Consequently, the path that the mobile beacon node travels is very important for the performance of the scheme.

The purpose of using a mobile beacon node is to collect position information from unknown sensor nodes. Therefore, the path for the mobile beacon node to travel needs to meet the following two requirements.

(1) It must cover the entire network. Since sensor nodes in the network may be deployed randomly, the beacon node needs to connect to as many unknown sensor nodes as possible in order to improve the efficiency of localization.

(2) It must complete localization quickly. The path for the mobile beacon node to travel along should support efficient localization and make the number of collection points as minimal as possible.

The area that the mobile beacon node can effectively cover at anytime is modeled by a round area or circle with its present position as the center point and the signal transmission range as the radius. We can thus build a mathematical model to optimize the path that the mobile beacon node should follow as it traverses the entire network, which can be viewed and solved as the area coverage problem.

We assume that all sensor nodes in the network are deployed within a rectangular area, and the size of the area as well as the communication radius of the sensor nodes are known in advance. Our objective is to have the circles of the beacon node cover the entire rectangular area as it traverses through the network while keeping the overlapping regions of the circles as minimal as possible. This requires that with a collection of points $\{(x_{c_1}, y_{c_1}), (x_{c_2}, y_{c_2}), \ldots, (x_{c_n}, y_{c_n})\}$ that the mobile beacon node stops during its journey, for any arbitrary point (x_o, y_o) that represents the position of an unknown sensor node, the following condition must be met: if (x_o, y_o) is located in the rectangular network area, it must be covered by at least one circle of the mobile beacon node with a collection point of the mobile beacon node as the center and the signal transmission range as the radius.

From the above analysis, we can see that the circular areas that the mobile beacon node generates as it moves along a path could have some parts overlapping with each other in order to cover the entire rectangular area. Therefore, we have to make sure that the circles would cover each and every unknown sensor node deployed in the network while making the overlapping parts as small as possible, which is the basic principle in the design of the mobile path for the mobile beacon node to traverse and cover the entire network. When the overlapping areas of different circles are the same, the polygon that is constructed with the chords of each circle becomes straight polygons. We can thus transform the original area coverage problem of using circles to cover a rectangular area to the problem of using the polygons to cover a rectangular area.

Lemma 1. *The number of edges of a straight polygon for dividing a rectangle can only be 3, 4, or 6.*

Proof. We assume that the rectangle is covered by one or more straight polygons each of which has p edges ($p \geq 3$). If α is the interior angle of the polygon, then $\alpha = (180^\circ(p - 2))/p$.

Let q be the total number of polygons to which a vertex belong. Then, $q = 360^\circ/\alpha$. Since q must be a natural number and q can be calculated using

$$q = 360^\circ \cdot \frac{p}{180^\circ(p-2)} = \frac{2p}{p-2}. \tag{1}$$

Thus, p can be calculated using

$$p = \frac{2q}{q-2} = \frac{2(q-2)+4}{q-2} = 2 + \frac{4}{q-2}. \tag{2}$$

From formula (2), we can get the following results:

$$\begin{cases} q = 3 \\ p = 6, \end{cases} \quad \begin{cases} q = 4 \\ p = 4, \end{cases} \quad \begin{cases} q = 6 \\ p = 3. \end{cases} \tag{3}$$

Therefore, the number of edges p can only be 3, 4, or 6. The three specific coverage situations are demonstrated in Figure 1. In the figure, S_c shows the overlapping part between two adjacent circles. Let S_s and S_t denote the area of the sector and that of the triangle, respectively. Thus, $S_c = 2(S_s - S_t)$. We can then use formula (4) to get S_s, S_t, and S_c, respectively, in which χ is the degree of the central angle of the sector, and x is the percentage of S_c over the circle.

$$\begin{aligned} S_s &= \left(\frac{\chi}{360^\circ}\right) \cdot \pi R^2, \\ S_t &= \frac{1}{2} R^2 \sin\chi, \\ S_c &= x\pi R^2. \end{aligned} \tag{4}$$

We can then derive

$$x\pi R^2 = 2\left(\left(\frac{\chi}{360^\circ}\right) \cdot \pi R^2 - \frac{1}{2} R^2 \sin\chi\right). \tag{5}$$

Thus, we get

$$x = 2\left(\frac{\chi}{360^\circ}\right) - \frac{\sin\chi}{\pi}, \tag{6}$$

when $p = 3, 4, 6$, $\chi = 120^\circ, 90^\circ, 60^\circ$, and $x = 0.39, 0.18, 0.06$, respectively.

It is thus clear that using straight hexagons to cover a rectangular area can achieve the highest efficiency, which coincides with the core idea of the honeycomb network principle.

We design the path for the mobile beacon node to travel as follows.

First, we need to determine the minimum number of circles to cover the rectangle with hexagons. Let's deduce the formulas for solving this problem.

Suppose the size of a rectangular area is $M * N$ and the communication radius of the wireless nodes is R. Let m be the number of circles in one odd-numbered horizontal line, n be the number of circles in one vertical line. Let l and d be the distances shown in Figure 1. We can thus calculate l and d by using

$$\begin{aligned} l &= 2 \cdot R \cdot \cos\frac{\pi}{6} = \sqrt{3}R, \\ d &= R \cdot \sin\frac{\pi}{6}. \end{aligned} \tag{7}$$

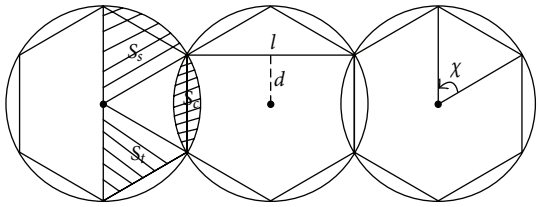


Figure 1: Relationship between the coverage areas with the mobile beacon node at different collection points.

Then, n can be calculated using formulas (8) or (9) when it is an odd or an even number, respectively.

$$\left(\underbrace{\left(\frac{1}{2}+1\right)+1+2+1+2+\cdots+1+2+1}_{n-1}\right)R<N$$
$$\leq\left(\underbrace{\left(\frac{1}{2}+1\right)+1+2+1+2+\cdots+1+2+1+\left(1+\frac{1}{2}\right)}_{n}\right)R. \tag{8}$$

$$\left(\underbrace{\left(\frac{1}{2}+1\right)+1+2+1+2+\cdots+1+\left(1+\frac{1}{2}\right)}_{n-1}\right)R<N$$
$$\leq\left(\underbrace{\left(\frac{1}{2}+1\right)+1+2+1+2+\cdots+1+2+1}_{n}\right)R. \tag{9}$$

And m can be calculated using

$$\sqrt{3}r\cdot(m-1)<M\leq\sqrt{3}r\cdot m. \tag{10}$$

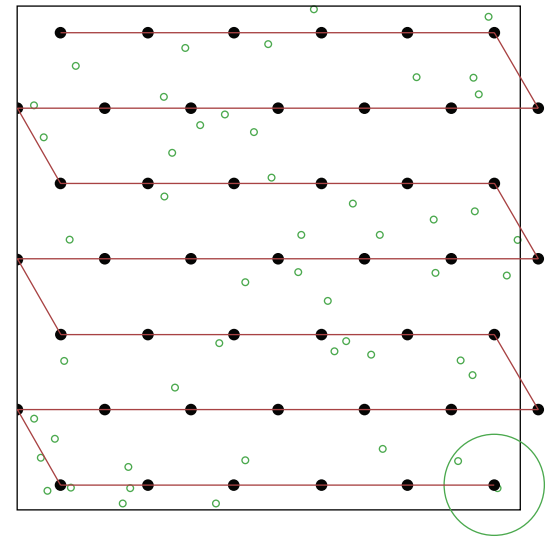
We are now ready to compute the minimum number of circles that can cover the entire rectangular network area through using formula (11) in which P is the total number of collection points.

$$P=\begin{cases}n\cdot m+\dfrac{n-1}{2}, & n \text{ is an odd number.}\\ n\cdot m+\dfrac{n}{2}, & n \text{ is an even number.}\end{cases} \tag{11}$$

The path thus derived for the mobile beacon node to traverse and cover the entire network is shown in Figure 2. □

3.4. Position Calculation. The calculation of the position of each and every unknown sensor node is performed by the base station in our SLMB scheme, which is different from the traditional range-based localization methods in order to reduce the convergence time of localization as well as the cost of information collection by the mobile beacon node. Most existing range-based localization methods need multiple measurement points to measure the distances to unknown sensor nodes, whether they are based on the means of arrival time, signal strength, or angle. In the SLMB scheme, we combine the measurements of angle and arrival time to determine the distances so as to reduce the requirement on the number for collection points. As shown in Figure 3, the mobile beacon node can sense the directional angle θ of received messages from an unknown sensor node using an antenna array and, at the same time, measure the distance to the same node using time information in the messages. Then, the position of the unknown sensor node can be calculated using both pieces of information.

Since there are only a limited number of collection points, the measurement in the proposed SLMB scheme may incur errors. As shown in Figure 3 in which the angle error is $\pm\Delta\theta$ and the range error is $\pm\Delta d$, we take the centroid of the area with angle interval $\{(\theta-\Delta\theta),(\theta+\Delta\theta)\}$ and length interval $\{(d-\Delta d),(d+\Delta d)\}$ as the position of the unknown sensor node.



○ Unknown nodes
● Collection points for the beacon nodes

Figure 2: Mobile path for the mobile beacon node.

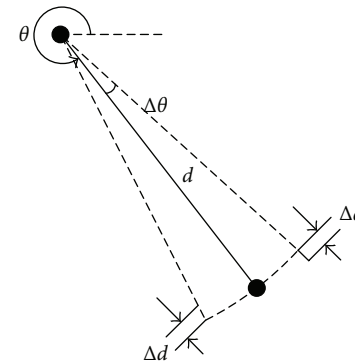


Figure 3: Position relationship between the mobile beacon node and a to-be-located unknown sensor node.

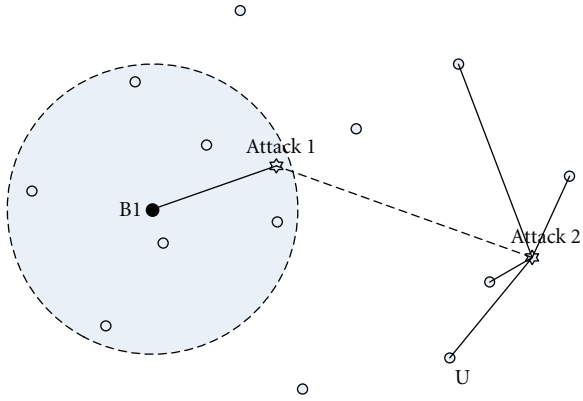


Figure 4: The wormhole attack model.

3.5. The Security Mechanism. Wormhole attacks are the primary type of attacks that can be launched without compromising any cryptographic keys. It can cause serious consequences to localization, especially when the beacon node wakes up the neighboring unknown sensor nodes through a localization request and when an unknown sensor node responds to the request. A communication channel between two attackers is shown in Figure 4 from which we can see that attack1 can transmit a request from B1 to the unknown sensor nodes that are outside of the coverage area of B1. The communication channel can also be used to replay the response of U to B1.

In order to detect information that is replayed from outside of the normal communication range, when the beacon node receives some information from the same unknown sensor node at different collection points, the beacon node should check whether one position information has been received repeatedly from the same exit of a wormhole, then compare the distances of the repeated positions d with the threshold T. If $d \leq T$, it means that this is a normal error caused by the overlapping area of the two collection points. If $d > T$, it could mean an attack. If the wormhole attack is launched against just one node, the beacon node is not able to determine the location of the attacker. However, if the wormhole attack is launched towards multiple nodes, the attacker could be detected according to the wormhole attack filtering principle that is based on the same exit.

In addition, the beacon node may also receive messages with the same ID of an unknown sensor node since the replayed information is within the same communication radius. According to the signal transmission characteristics, we will only accept the first received information, discard the latter ones, and include it in the blacklist since the replayed information couldn't arrive at the object earlier than the original signal with the same transmission power.

3.6. Application Issues. The SLMB scheme has been designed to make sure that the mobile beacon node fully covers the entire deployment area, thus making it suitable for static WSNs. In dynamic WSNs in which the location of a sensor node may change from time to time due to mobility or the network environment, the SLMB scheme can be enhanced so that the mobile beacon node will periodically traverse the network to calculate and update the information on sensor locations. The interval between repeated SLMB applications can follow a strategy that can be determined based on application requirements as well as network environments. In addition, we can also adapt the basic SLMB scheme for huge WSNs by dividing the sensor deployment area into multiple regions and then deploying multiple mobile beacon nodes in the area with each for a different region to meet the real-time requirement of sensor localization. The model allows us to derive satisfactory localization results by making each beacon node cover minimum number of collection points with particular time constraints to achieve desired performance for sensor localization.

As it has been widely known, the application of WSNs has now spread to a lot of different areas including those in harsh environments such as battlefields and wildlife monitoring as well as many emerging applications in our daily life. Both distributed and centralized sensor localization schemes have their distinctive strengths and weaknesses to deal with different application scenarios. In a harsh environment where it is almost impossible for human beings to get near the sensors, a remotely controlled wireless mobile device can be used to traverse the deployment area acting as the beacon node to accomplish the functionality of sensor localization. If there are mountains and hills in the deployment area, we can manage to map the three-dimensional area into our two-dimensional model and thus still use a wireless flying device to collect location information from the sensor nodes. If sensor nodes are deployed in a well-developed area, a vehicle can be operated to move along a designated route to cover the entire deployment area to collect location information from the sensor nodes. In extreme situations where it is not feasible to use a mobile device as the beacon, distributed sensor localization algorithms should be considered as a complementary scheme. As WSNs find more and more diverse applications ranging from traditional applications to the Internet of Things scenario, there are certainly many applications in which our SLMB scheme can be used to perform sensor localization to achieve a wide variety of performance objectives.

We also would like to note that the proposed SLMB scheme is appropriate for WSNs that do not have a too high requirement on the accuracy of localization. To improve the accuracy of localization though, we can increase the number of collection points for the mobile beacon node to collect more position information about unknown sensor nodes and calculate the positions of the unknown sensor nodes through maximum likelihood estimation, which is a future work in our research in which we will demonstrate how accuracy improves along with the increase in the number of collection points. This is a tradeoff between accuracy and required completion time in addition to some other considerations such as the cost of communication and computation.

4. Simulation and Analysis

4.1. Performance on Sensor Localization. We have performed several simulations to evaluate the performance of

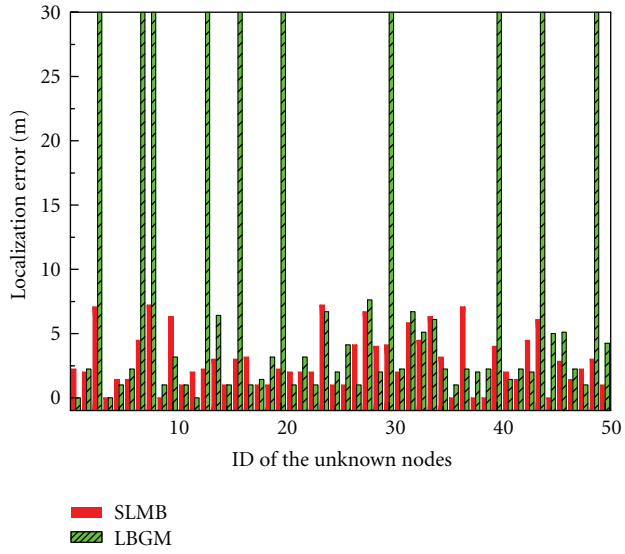


Figure 5: Comparison of localization errors.

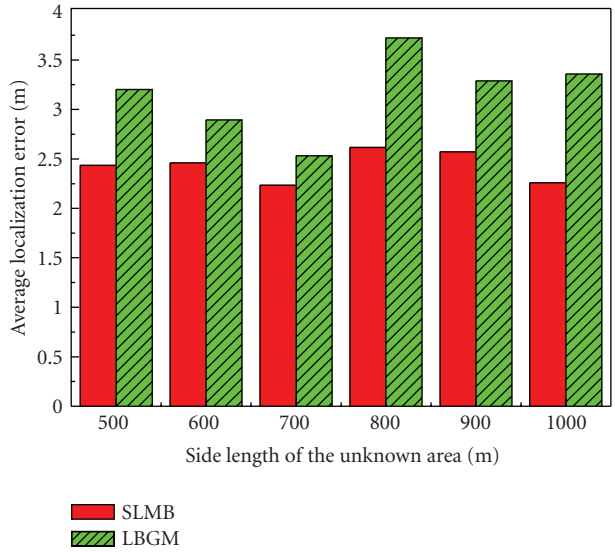


Figure 6: Average localization error of unknown sensor nodes for various network sizes.

the proposed SLMB scheme on sensor localization. The network configuration for our first simulation is set up as follows: there are 50 unknown sensor nodes and a mobile beacon node deployed randomly in an area of $800 \times 800\,\text{m}^2$. The transmission range R of the wireless nodes is set up to be 100 m. The distance error and angle error between the mobile beacon node and any unknown sensor node are set up in the range of 0–0.05 and 0–$0.05 * \pi$, respectively.

We compare localization error between our proposed SLMB scheme and a localization scheme based on the general mobile path (LBGM). Localization error is an important metric to measure the performance of sensor localization in WSNs, which is the distance between localization coordinates and the actual coordinates calculated using (12) in which (x_U, y_U) and (x'_U, y'_U) denote the measured and the actual coordinates of unknown sensor node U, respectively.

$$e_U = \sqrt{(x_U - x'_U)^2 + (y_U - y'_U)^2}. \tag{12}$$

The simulation results on localization error for 50 unknown sensor nodes are shown in Figure 5 from which we can see that there are several unknown sensor nodes that have an localization error of infinite value, which means that these nodes cannot be located using the LBGM scheme. Our proposed SLMB scheme is shown to be more effective, for it can improve the success rate of localization of unknown sensor nodes for about 20% while reducing localization errors in general.

Since there are a variety of applications that need the location information about deployed sensor nodes but sensors may be different, it is worthwhile to investigate the performance of the proposed SLMB scheme for different network sizes in terms of coverage area and for different transmission ranges of the sensor nodes. We hereby use the notion of average localization error in evaluating our SLMB scheme using (13) in which N denotes the number of unknown sensor nodes in a network as follows:

$$\bar{e} = \frac{\sum_{i=1}^{N} e_i}{N}. \tag{13}$$

We first investigate the effect of network size on sensor localization. In the evaluation, 100 unknown sensor nodes and a mobile beacon node are deployed in the network, the network is set up to cover an area of $500 * 500\,\text{m}^2$, $600 * 600\,\text{m}^2$, $700 * 700\,\text{m}^2$, $800 * 800\,\text{m}^2$, $900 * 900\,\text{m}^2$, and $1000 * 1000\,\text{m}^2$, respectively, and R is set up to be 100 m. The distance error and angle error between the mobile beacon node and any unknown sensor node are also set up in the range of 0–0.05 and 0–$0.05 * \pi$, respectively.

The average localization error of the unknown sensor nodes using the proposed SLMB scheme and that using the LBGM scheme are shown in Figure 6 and the success rates of localization of these two schemes are shown in Figure 7. From these two figures, we can see that our SLMB scheme is more effective in covering the entire network area and in improving the accuracy of localization of unknown sensor nodes.

We then investigate the effect of transmission range of the nodes on localization. In the evaluation, 100 unknown sensor nodes and a mobile beacon node are deployed in a network area of $600 * 600\,\text{m}^2$, and the transmission range of the wireless nodes is set up to be 50 m, 60 m, 70 m, 80 m, 90 m, and 100 m, respectively. The distance error and angle error between the mobile beacon node and unknown sensor nodes are also set up in the range of 0–0.05 and 0–$0.05 * \pi$, respectively. The average localization errors for the 100 unknown sensor nodes using the proposed SLMB and the LBGM schemes as well as the localization success rates are shown in Figures 8 and 9, respectively. From these two figures, we can see that the SLMB scheme can achieve better performance both on localization accuracy and on

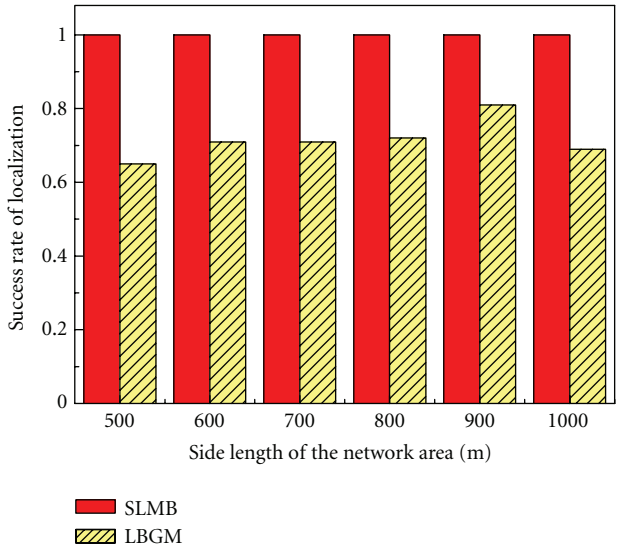


Figure 7: Success rate of sensor localization for various network sizes.

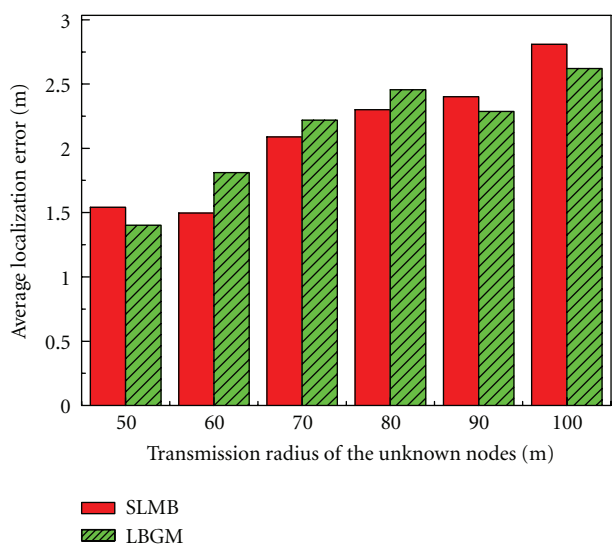


Figure 8: Average localization error of unknown sensor nodes for various transmission radiuses.

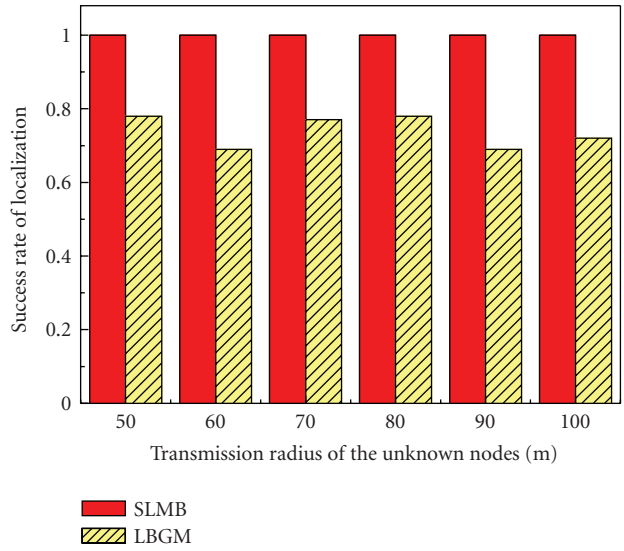


Figure 9: Success rate of localization for various transmission radiuses.

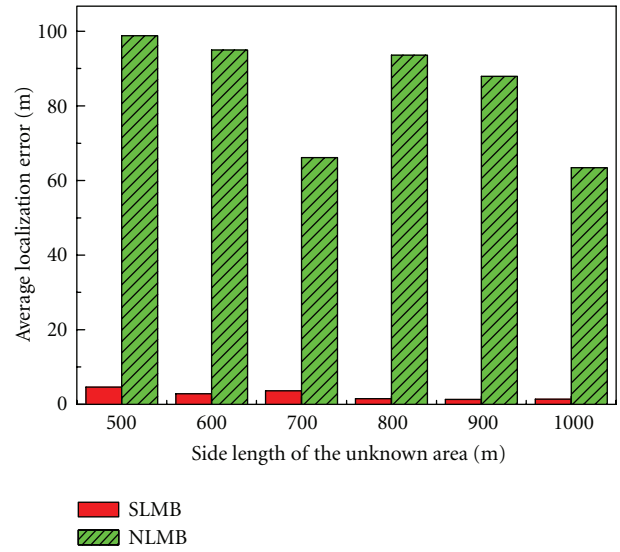


Figure 10: Average localization error for various network sizes in a hostile environment.

localization success rate with very stable results. The reason for the small difference shown in Figure 8 in localization accuracy between SLMB and LBGM is that it only includes the simulation results of those nodes that can be successfully located.

Finally, we investigate the performance of the SLMB scheme in terms of its ability of resisting against wormhole attacks. We randomly distribute two pairs of wormhole attackers in the experiment environments that we set up above for various network sizes and different transmission radiuses. The average localization errors of the unknown sensor nodes under these two environments are shown in Figures 10 and 11, respectively, from which, we can see that the SLMB scheme is able to fight against wormhole attacks, thus improving the localization accuracy for WSNs compared to normal localization by using a mobile beacon (NLMB).

4.2. Performance on Energy Consumption. Batteries are usually used to supply power in the sensor nodes in WSNs, and a sensor node is considered to be no longer functional when the battery in the node is exhausted. Therefore, the efficiency of energy usage must be considered in any protocol design for WSNs. The energy consumption of a sensor node mainly consists of energy consumption for data transmission and that for data processing. We now analyze the performance

of SLMB with respect to energy consumption and compare it to DV-Hop [12], a classic distributed sensor localization method.

First, let us develop an energy consumption model for the proposed SLMB scheme. In our model, the operations in each sensor node that consumes energy include data transmission, data reception, and position calculation, and the energy consumed for each of these operations is denoted as E_s, E_r, and E_c in which it is widely recognized that E_s and E_r are normally much higher than E_c. The total amount of energy consumed by each sensor node can then be calculated using Formula (14) in which E_s and E_r can be calculated using formulas (15) and (16), respectively. In all the formulas, k_1 and k_2 denote the number of bits that have been sent and received, respectively, during sensor localization and E_0 denotes energy consumption for sending or receiving a single bit of data. Energy consumption for sending a message includes two parts; one is calculated based on the amount of data sent and the other is on the distance between the sender and the receiver that we denote as E_{s1} and E_{s2}, respectively, and d is the distance and x a constant multiplier.

$$E = E_s + E_r + E_c, \tag{14}$$

$$E_s = E_{s1} + E_{s2} = E_0 * k_1 + x * k_1 * d^2, \tag{15}$$

$$E_r = E_0 * k_2. \tag{16}$$

In our SLMB scheme, we assume that the amount of data is fixed and is the same for every message sent and received and that any data that is sent by a node can be received by all the neighboring nodes within the radius of the communication of the sending node. We now compare the performance on energy consumption of SLMB to that of DV-Hop. In DV-Hop, an unknown sensor node needs to transmit localization information through multiple hops and calculates its position coordinates by itself.

The network configuration for our simulation on energy consumption is set up as follows: 500 unknown sensor nodes are deployed randomly in an area of $500 \times 500\,\text{m}^2$; the transmission range R of the wireless nodesis assumed to be 50 m. We can then get $E_0 = 50\,\text{nj/bit}$ and $x = (0.1\,\text{nj/bit})/m^2$. The localization results may also need to be updated in some applications. Thus, we evaluate the performance on energy consumption for multiple applications of sensor localization, and the results for the accumulative energy consumption are shown in Figure 12. We now investigate the energy consumption for varying numbers of unknown sensor nodes in the network, and the results are shown in Figure 13.

We can see from the above evaluation that energy consumption in SLMB is much smaller than that in DV-Hop. SLMB can keep energy consumption at a very low level with various numbers of unknown sensor nodes, especially for some networks in which multiple applications of sensor localization are needed. Under both circumstances, our proposed SLMB scheme achieves a much better performance on energy consumption. Another obvious advantage of the SLMB scheme is that it not only can lower energy consumption in each sensor node, but it can also keep energy consumption evenly across all the unknown sensor nodes in the network, thus preventing some unknown sensor node from exhausting energy prematurely and becoming unusable before some others and, as a result, prolonging the life of the network. The main factors that lead to the improved performance are that the SLMB scheme has been designed to achieve the goals of reducing the number of data transmissions, making the unknown sensor nodes in the network transmit messages with same amount of data and same signal strength, all contributing to significant reduction in the total amount of energy consumed in unknown sensor nodes for the functionality of sensor localization.

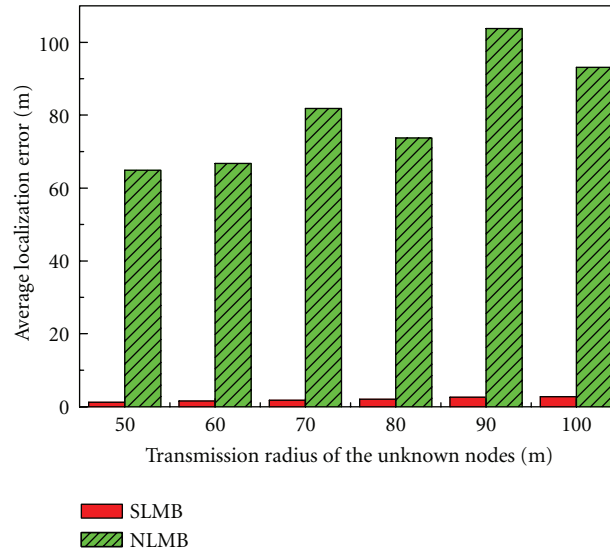


Figure 11: Average localization error for various transmission radiuses in a hostile environment.

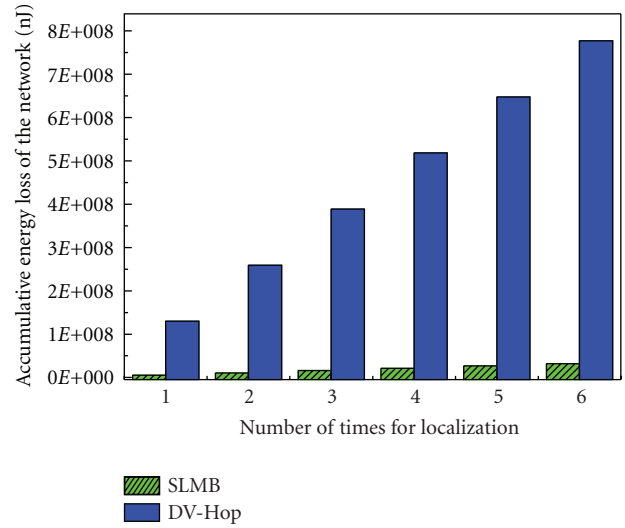


Figure 12: Accumulative network energy consumption for multiple applications of sensor localization.

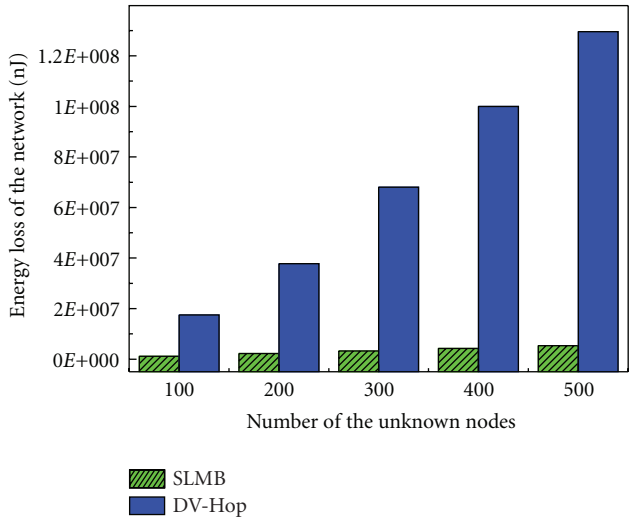


Figure 13: Network energy consumption for varying numbers of unknown sensor nodes in the network.

5. Conclusions

In this paper, we presented a secure centralized localization scheme by using a mobile beacon node. In the scheme, the mobile beacon node is responsible for collecting information about position relationship with unknown sensor nodes and for sending the information to the base station where the positions of the unknown sensor nodes are calculated. The scheme can greatly reduce the computation cost compared to distributed localization algorithms and lower the communication overhead for sending position information to the base station compared to some other centralized localization algorithms for the unknown sensor nodes. In the scheme, most work on collecting and sending information is done by the mobile beacon node, thereby also reducing the security risks in sensor localization. Specifically, the proposed scheme is designed to resist wormhole attacks in localization to improve the security. The scheme also includes a mathematical computation model to determine the collection points for the mobile beacon node to completely and efficiently cover the entire sensor network. The proposed scheme only requires that the beacon node to have an antenna array.

In the future, we will extend our secure localization scheme to improve the security of localization in the presence of other kinds of malicious attacks without incurring too much computational overhead and communication cost. We will also investigate the performance of sensor localization schemes that use different mobile beacon paths, different types of deployment, and different transmission radius for the sensor nodes.

Acknowledgment

This work is supported by National Natural Science Foundation of China (61272500).

References

[1] S. K. Yang and K. F. Ssu, "An energy efficient protocol for target localization in wireless sensor networks," *Proceedings of World Academy of Science, Engineering and Technology*, vol. 56, pp. 398–407, 2009.

[2] M. Boushaba, A. Hafid, and A. Benslimane, "High accuracy localization method using AoA in sensor networks," *Computer Networks*, vol. 53, no. 18, pp. 3076–3088, 2009.

[3] R. Sugihara and R. K. Gupta, "Sensor localization with deterministic accuracy guarantee," in *Proceedings of IEEE Conference on Computer Communications (INFOCOM '11)*, pp. 1772–1780, April 2011.

[4] J. Park, Y. Lim, K. Lee, and Y. H. Choi, "A Polygonal Method for Ranging-Based Localization in an Indoor Wireless Sensor Network," *Wireless Personal Communications*, vol. 60, no. 3, pp. 521–532, 2011.

[5] Y. W. E. Chan and B. H. Soong, "A new lower bound on range-free localization algorithms in wireless sensor networks," *IEEE Communications Letters*, vol. 15, no. 1, pp. 16–18, 2011.

[6] K. Langendoen and N. Reijers, "Distributed localization in wireless sensor networks: a quantitative comparison," *Computer Networks*, vol. 43, no. 4, pp. 499–518, 2003.

[7] M. Karakaya and H. Qi, "Distributed target localization using a progressive certainty map in visual sensor networks," *Ad Hoc Networks*, vol. 9, no. 4, pp. 576–590, 2011.

[8] A. Karbasi, "From centralized to distributed sensor localization," in *Proceedings of Annual International Conference on Mobile Computing and Networking (MOBICOM '10)*, pp. 5–7, September 2010.

[9] M. Simek, D. Komosny, R. Burget, P. Moravek, and R. Silva, "Centralized boundary discovery algorithms for anchor-free localization in wireless sensor networks," in *Proceedings of the International Conference on Ultra Modern Telecommunications and Workshops (ICUMT '09)*, pp. 1–7, October 2009.

[10] S. Zhu and Z. Ding, "Distributed cooperative localization of wireless sensor networks with convex hull constraint," *IEEE Transactions on Wireless Communications*, vol. 10, no. 7, pp. 2150–2161, 2011.

[11] A. Karbasi and S. Oh, "Distributed sensor network localization from local connectivity: performance analysis for the HOP-TERRAIN algorithm," in *Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS'10)*, pp. 61–70, June 2010.

[12] D. Niculescu and B. Nath, "DV based positioning in ad hoc networks," *Telecommunication Systems*, vol. 22, no. 1–4, pp. 267–280, 2003.

[13] C. Alippi and G. Vanini, "A RSSI-based and calibrated centralized localization technique for wireless sensor networks," in *Proceedings of the 4th Annual IEEE International Conference on Pervasive Computing and Communications Workshops*, pp. 301–305, March 2006.

[14] M. Shahrokhzadeh, A. T. Haghighat, and B. Shahrokhzadeh, "An efficient centralized localization method in wireless sensor networks," in *Proceedings of the 17th International Workshop on Energy-Aware Communications (EUNICE '11)*, vol. 6955 of *Lecture Notes in Computer Science*, pp. 217–220, September 2011.

[15] E. C. Kim and K. Kim, "Distance estimation with weighted least squares for mobile beacon-based localization in wireless sensor networks," *IEEE Signal Processing Letters*, vol. 17, no. 6, pp. 559–562, 2010.

[16] K. F. Ssu, C. H. Ou, and H. C. Jiau, "Localization with mobile anchor points in wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 54, no. 3, pp. 1187–1197, 2005.

[17] F. Zhao, H. Y. Luo, and Q. Lin, "An mobile beacon-assisted localization algorithm based on network-density clustering for wireless sensor networks," in *Proceedings of the 5th International Conference on Mobile Ad-hoc and Sensor Networks (MSN '09)*, pp. 304–310, December 2009.

[18] Q. Mi, J. A. Stankovic, and R. Stoleru, "Practical and secure localization and key distribution for wireless sensor networks," *Ad Hoc Networks*, vol. 10, no. 6, pp. 946–961, 2012.

[19] N. Labraoui, M. Gueroui, and M. Aliouat, "Secure DV-Hop localization scheme against wormhole attacks in wireless sensor networks," *European Transactions on Telecommunications*, vol. 23, no. 2012, pp. 303–316, 2012.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 301750, 14 pages
doi:10.1155/2012/301750

Research Article

A Secure Cluster Formation Scheme in Wireless Sensor Networks

Gicheol Wang,[1] Dongkyun Kim,[1] and Gihwan Cho[2]

[1] *Department of Advanced KREONET Service, Korea Institute of Science and Technology Information, Daejeon 305-806, Republic of Korea*
[2] *Division of Computer Engineering, Jeonbuk National University, Jeonju 561-756, Republic of Korea*

Correspondence should be addressed to Gihwan Cho, ghcho@chonbuk.ac.kr

Received 15 June 2012; Accepted 11 September 2012

Academic Editor: Mihui Kim

In wireless sensor networks, clustering expedites many desirable functions such as load balancing, energy savings, and distributed key management. For secure clustering, it is very important to find compromised nodes and remove them during the initial cluster formation process. If some nodes are compromised and survive the censorship process, they can make some nodes have a different cluster view and can split a cluster into multiple clusters to deteriorate cluster quality as a whole. To resolve these problems, we propose a robust scheme against such attacks in this paper. First, our scheme generates large-sized clusters where any two nodes are at most two hops away from each other to raise the quality of clusters. Second, our scheme employs the verification of two-hop distant nodes to preserve the quality of the large-sized clusters and refrains from splitting the clusters. Last, our scheme prefers broadcast transmissions to save the energy of nodes. Security analysis proves that our scheme can identify compromised nodes and preserves the cluster membership agreement against the compromised nodes. In addition, simulation results prove that our scheme generates fewer clusters and is more secure and energy efficient than the scheme producing only small-sized clusters.

1. Introduction

The recent advancement of wireless technology, sensor technology, and low-power embedded systems have enabled the development of low-power wireless sensing devices and they can collectively constitute a wireless sensor network for various applications. Wireless sensor networks have been widely deployed for military surveillance, pollution and structure monitoring, industrial equipment monitoring, mountain fire monitoring, and so on. Clustering in such networks provides many advantages such as the reduction of energy consumption [1, 2], load balancing [3], and distributed key management [4, 5].

The most prominent benefit of clustering is that it can greatly reduce the energy consumption of nodes and lengthen the network lifetime. Clustering is grouping physical network nodes into a small number of logical assemblies and maintaining them during the network operation. The logical assemblies are called clusters. For the initial formation of clusters, each node performs a cluster formation protocol. If each cluster requires a leader, nodes in each cluster should perform a leader election protocol. Hereafter, we call the leader as cluster head. Since a cluster head plays a crucial role such as collecting sensed data from other nodes and transferring the collected data to the sink, compromised nodes try to become cluster heads. In order to keep compromised nodes from being a cluster head, we can use two main strategies. First, we can identify compromised nodes and remove them during the initial cluster formation. If a compromised node survives the censorship process, it can easily obtain candidacy for being a cluster head. Therefore, removing the compromised nodes during the cluster formation is the first defense line for secure clustering. Second, we can keep the compromised nodes from predicting and manipulating results in cluster head elections and expediting their wins in the elections. This strategy is the second defense line for secure clustering. In this paper, we only focus on the first defense line as the second defense line was covered in [6–8].

Essentially, protecting the first defense line is very important for secure clustering. If the first defense breaks down, the second defense line is also put in danger and suffers from compromised nodes. To keep the first defense line from external attackers, thus far, several schemes using symmetric cryptography [9–11] were proposed. However, they cannot

keep compromised nodes from obstructing the operation of the protocols while they also prevent the participation of external attackers. Sun et al. proposed a scheme using the protocol conformity check and asymmetric cryptography [12]. It is effective at keeping the two types of compromised nodes from obstructing the operation of the protocol. However, this scheme operates with only small-sized clusters (i.e., cliques) and splits the cliques whenever a suspicious node is found in the cliques. This generates many clusters and the average size of clusters also decreases. Moreover, this scheme causes a lot of communication overhead to verify the protocol conformity of nodes.

To resolve the above problems, we propose a novel cluster formation scheme in this paper. First, our scheme generates large-sized clusters where the hop distance between any two members is at most two. Second, our scheme minimizes the separation of clusters by exchanging information between two-hop distant nodes. Last, instead of unicast transmissions, our scheme mainly employs broadcast transmissions to reduce energy consumption. Generating and maintaining large-sized clusters can provide many useful applications. First, a large-sized cluster has a long TDMA transmission schedule in which each node transmits its reading less frequently than the case of a short TDMA schedule. Therefore, a long TDMA schedule saves the amount of energy consumption at nodes, and consequently lengthens the network longevity. Second, generating large-sized clusters for a network shortens the hop distance between any two clusters and provides routing efficiency when nodes deliver their reading to the sink through multiple clusters. Last, a large-sized cluster enhances security if its CH-role node changes time after time. That is, if a CH-role node changes, the keys used for communication of the CH and its members also change accordingly. Since a large-sized cluster has more keys used for communication of the CH and its members than a small sized cluster, it guarantees higher confidentiality and integrity for their readings against malicious nodes.

The rest of this paper is organized as follows. Related work concerning secure cluster formation is briefly described in Section 2. Section 3 provides the network and threat model. The details of the proposed scheme are described in Section 4, and Section 5 provides the security analysis and simulation results. Finally, Section 6 concludes this paper.

2. Related Work

Heinzelman et al. proposed LEACH (Low-Energy Adaptive Clustering Hierarchy) in which sensors declare themselves as a cluster head according to a probability of being a cluster head [1]. Because cluster heads consume much more energy than normal nodes, this scheme attempts to extend the network lifetime by assigning cluster head roles to all nodes alternately. In this scheme, some nodes with a higher probability than threshold declare themselves as cluster heads and other nodes join in one of them. However, this scheme has no measure to protect the cluster formation.

F-LEACH protects the cluster head election in LEACH [9]. In this scheme, a node declares itself as a cluster head using common keys shared with the sink, and the sink authenticates the declaration message using the same keys. Then, the sink securely broadcasts the authenticated cluster heads using μTESLA [13]. Normal nodes only join one of the authenticated cluster heads. However, this scheme has no mechanism that authenticates the joining of normal nodes. Oliveira et al. proposed SecLEACH [10] in which the sink authenticates the cluster head nodes and the cluster heads authenticate the joining nodes. In F-LEACH and SecLEACH, some keys for the authentication are distributed to sensors prior to deployment. However, both F-LEACH and SecLEACH can only protect the cluster formation from external attackers. In other words, they cannot prevent compromised nodes from declaring themselves as cluster heads and from joining in any cluster head.

Liu proposed a cluster formation scheme where only predetermined nodes declare themselves as cluster heads and other nodes join in any cluster directly or via a relay node [11]. Since the declaration of any cluster head or the join of any normal node is authenticated by preassigned polynomial shares, an external attacker cannot participate in the cluster formation. This scheme also has a wormhole prevention mechanism in which a node with many neighbors shuts itself down or the sink enforcedly shuts down a node with many neighbors by reporting the node as a wormhole attacker. However, if any relay node is compromised by attackers, the relay node can invoke a DoS (Denial of Service) attack by cutting down the connection between the cluster head and its serving nodes. In addition, a compromised node can disturb a relay node determination and break all connections using the relay node. Moreover, attackers can target the predetermined cluster heads for compromise because their roles are fixed.

Sun et al. proposed a secure cluster formation scheme which verifies the protocol conformity of nodes to identify malicious nodes [12]. In this scheme, all nodes are grouped into cliques, in which all nodes are directly connected with each other. After the clique formation, each node checks whether all members in the clique agree on the clique membership or not. If a normal node finds a disagreement, it performs the protocol conformity verification for other nodes in the clique in order to recognize and remove compromised nodes. This scheme well identifies and removes compromised nodes through the protocol conformity check. However, the scheme increases the number of clusters in the network because it only produces small-sized clusters (i.e., cliques) and splits a cluster whenever a suspicious node is identified in the cluster. Moreover, it burdens nodes with a lot of communication overhead because it requires an abundance of unicast communication during the protocol conformity check. To elaborate on why the unicast transmission causes more overhead than broadcast communication, we assume that a node transmits a message to its all neighbors serially. If we use unicast communication, the node should transmit the message as frequently as the number of the neighbors. Contrarily, the broadcast communication brings the same effect through only one-time transmission.

Rifà-Pous and Herrera-Joancomartí divided the cluster formation process into three phases; cluster discovery phase,

cluster head designation phase, and cluster maintenance phase [14]. In the cluster discovery phase, members in a cluster build a consensus in the cluster membership. In the cluster head designation phase, members in a cluster elect a cluster head based on the number of neighbors and the frequency of cluster-head-role performance. In the cluster maintenance phase, the elected cluster heads play the role of a local CA (Certification Authority), and issue an authorization certificate to each member. However, this scheme assumes that all nodes conform to the cluster discovery protocol. For instance, if a compromised node transmits a message to some nodes and avoids the transmission to other nodes in the cluster discovery phase, the nodes in the same cluster have a different membership. This separates a cluster into multiple ones, and the separated clusters elect their cluster head, respectively, in the cluster head designation phase. So, this scheme is vulnerable to such an attack and can generate many clusters in the network.

3. Network and Threat Model

3.1. Network Model. We assume that sensor nodes are randomly scattered in the work field by an aircraft. Following the deployment, they never change their locations and are grouped into clusters to perform an energy-efficient operation like TDMA communication in clusters. The sink acts like a data collection center and a gateway to the wired networks. We assume the following.

First, any wormhole attack is invalidated by a wormhole prevention scheme such as the scheme in [15] so that each node can identify its neighbors correctly. Second, there is no message loss during the cluster formation process except for the intentional transmission avoidance of compromised nodes. Even though it seems to be quite an immoderate assumption, we need this assumption to discriminate a compromised node from normal nodes. Without this assumption, normal nodes cannot discriminate the misbehavior of compromised nodes from message loss. Removal of this assumption is an interesting future research item. Third, we assume that each node can adjust its transmission power when they send a message so that at most two-hop distant nodes can receive the message. Last, each node can support lightweight public key operations such as ECC (Elliptic Curve Cryptography) operations. It has been proven in [16] that energy-constrained sensors can well support lightweight public key operations such as ECDSA (Elliptic Curve Digital Signature Algorithm) signature generation and ECDSA verification.

3.2. Threat Model. Numerous attacks are available in wireless sensor networks. One of the most serious attacks is the DoS attack. Even though this sort of attacks is very stealthy and difficult to defeat, some promising countermeasures were introduced in [17]. Especially in a cluster structure, a greedy attacker might affiliate many normal nodes into its cluster by transmitting a bogus and long-distant message. Thanks to a wormhole prevention scheme, our scheme can easily defeat this type of attack. Sybil attack [18] also has a bad impact on the network because nodes cannot identify their legitimate neighbors. We assume that such an attack can be paralyzed by the schemes in [19].

To focus on the secure cluster formation problem, we assume that compromised nodes launch an attack by not conforming to a cluster formation protocol. Especially, we concentrate on two kinds of attacks on a cluster formation protocol. First, a compromised node can transmit a message to a section of nodes in a cluster while avoiding transmission to the rest of the nodes. Hereafter, this type of attack will be termed "selective transmission attack." In addition, a compromised node can utterly avoid the transmission of a message. We term this type of attack as "silence attack" hereafter. These attacks force nodes to have a different view on the cluster membership. This splits a cluster into multiple ones and consequently decreases the average size of clusters (i.e., the average number of members). In a cluster, the number of members affects the probability that a compromised node is elected as a CH on the basis of random selection. To elaborate on this problem, we assume that there is one cluster with a few members and another cluster with more members and they have one compromised node, respectively. We also assume that a cluster head is elected randomly like in [6] and compromised nodes cooperate with the election protocol. In this situation, the cluster with a few members is likely to elect the compromised node as a cluster head.

4. Secure Cluster Formation Using Two-Hop Conformity Verification

Before the detailed explanation of our scheme, we first establish some definitions, which are used throughout the paper.

(i) *Dominants*: At the beginning of the cluster formation, each node exchanges its certificate with neighbors. Due to the exchange, each node comes to know the neighbors whose IDs are lower than its own ID. Those nodes are referred as *dominants* hereinafter and their IDs are maintained in each node's storage.

(ii) *CS (Cluster Separator) node*: A node whose ID is the lowest among the neighbors becomes a *CS node*. In the same manner, a node expects that the lowest ID node among the dominants is going to declare itself as a *CS node*.

After the deployment of sensor nodes, each node exchanges its certificate with neighbors. This exchange enables each node to identify its neighbors and dominants. The lowest ID node in the neighborhood broadcasts a CS (Cluster Separator) message to determine a cluster border. Note that a CS node is not a cluster head but a protocol initiator. If the receivers of the CS message do not belong to any clusters, they join the cluster and respond with a broadcast message to inform the CS node of their joining. Hereafter, we call this message as a CR (Cluster Response) message. If a sender of a CR message is a dominant, the receivers remove the sender from their dominant list and check if there is no dominant. A node having no dominants becomes a CS node. If there

is no attack on the protocol, this process creates clusters in the network incrementally. After the border determination of clusters, CS nodes broadcast a FCS (Final Cluster Separator) message to request members to accept their joining the cluster. This message contains a list of CR messages, which were sent from its cluster and it enables the comparison of the same membership between nodes. A receiver accepts the request to join only if the received membership is exactly same with its own membership. If a CS node attempts to exclude a specific node, it might prevent the propagation of the victim's CR message and omit the victim's CR message in its FCS message. When the victim notifies this misbehavior, the victim searches a node which holds the victim's CR message to be a witness by sending a Solicitation message. Such a node signs the victim's CR message with its private key and sends it with a pre-received FCS message to the victim. This message is called the Solicitation Response message hereafter. Through this message reception, the victim acquires a witness to prove its legitimacy. Because the CS node's misbehavior has been confirmed, the victim now accuses the CS node with an Attacker Report message for other members to exclude it. Table 1 shows the time when the messages are sent and what functions they do.

The cluster formation is divided into two steps. Step 1 determines the border of clusters and verifies it. Step 2 merges a CS node into its cluster and verifies the merge.

4.1. Cluster Border Determination and Verification

4.1.1. Cluster Border Determination. A CS node broadcasts a CS (Cluster Separator) message to initiate the cluster formation. The message consists of the message type and the sender ID, which is signed by the sender's private key to prevent a spoofing attack. If a node receives the message from multiple CS nodes, the receiver joins the first comer and ignores other messages.

If a node receives a CS message, it verifies the received signature. If it is correctly verified, it joins the cluster and broadcasts the CR (Cluster Response) message. The CS message and the CR message determine the border of a cluster. The CR message consists of the message type, the CS node's ID, and the received signature from the CS node. The node also signs the message with its private key before broadcasting to protect the integrity and the authenticity of the CR message. Last, the sender's certificate is attached to the tail of the signed message because it might be propagated further than one hop. Note that each node holds only neighbors' certificates. A node in the same cluster verifies the CR message and stores the message if it is correctly verified. Note that a CS node in the same cluster uses the stored CR messages as a proof when it requires the members to accept its merge into the cluster in step 2. The receiver of a CR message first checks whether the message comes from a member of the same cluster or not. If the message originator belongs to the same cluster, and it is the first message from the originator, the receiver rebroadcasts the message to prevent silence attacks. However, if the CR message comes from another cluster, the receiver does nothing except check whether they are dominants or not. In case of dominants, the receiver removes the sender from the dominants and it becomes a CS node if the dominant list is empty. Figure 1 illustrates the flowchart of cluster border determination in step 1.

4.1.2. Verification of Cluster Border Determination. After the cluster border determination, each node checks if there are any deviations from the protocol. If some deviations are found, each node employs the following actions.

Attack Type 1. If a CS node avoids the broadcast of a CS message, members in the cluster cannot receive any message from its cluster. In this case, the members remove the CS node from dominants and the neighbor list and check if there is no dominant. If so, they become a CS node and broadcast a CS message. Figure 2 shows this countermeasure following the "yes" control flow from the comparative symbol "No message from same cluster."

Attack Type 2. when a CS node is the sole link point among nodes in a cluster, it may avoid rebroadcasting a CR message from a node. In that case, the nodes which are connected via only the CS node cannot receive the CR message and those nodes have a different view on cluster membership. To agree on the cluster membership between those nodes, we employ the following countermeasures. Figure 2 shows these countermeasures following the control flow from connector 3.

(i) If a node receives no CR messages from any nonneighbors, the node broadcasts its CR message with two-hop transmission power. Then, it waits for unknown nodes to respond with their CR message. When the node receives a CR message from an unknown node (i.e., a nonneighbor node) in the same cluster, it registers the sender into the member list.

(ii) If a node has any CR message from a nonneighbor node, it waits for other CR messages from other nonneighbor nodes. When it receives a CR message from another unknown node, it registers the sender into member list and broadcasts its CR message with two-hop transmission power to notify its existence.

Attack Type 3. If a CS node broadcasts its CS message to only a part of its neighbors, some neighbors cannot receive the CS message but can receive a CR message. In that case, the nonreceivers of the CS message can recognize that the CS node selectively transmits the CS message. If a node receives only CR messages not a CS message, the node has two choices such as the following.

(i) First, the node can request other nodes in the same cluster to accept its affiliation. Because the requested nodes cannot assure if the requestor is lying or the CS node is a malicious node, the situation is very ambiguous. So, they can remove the requestor and the CS node from the cluster to resolve the ambiguity.

Table 1: Function of messages and the time when they are sent.

Message type	Time when message is sent *Function*
CS (Cluster Separator) message	At the beginning of step 1 *Determines a cluster border*
CR (Cluster Response) message	Upon receiving a CS message *Request to join a cluster*
FCS (Final Cluster Separator) message	At the beginning of the step 2 *Merges a CS node into a cluster*
Solicitation message	When a victim does not receive a FCS message from its CS node *Acquire witnesses to prove its legitimacy*
Solicitation response message	When a node receives a Solicitation message and holds any evidence *Proves a victim's legitimacy*
Attacker report message	When a victim confirms misbehavior of a CS node *Excludes a CS node from members*

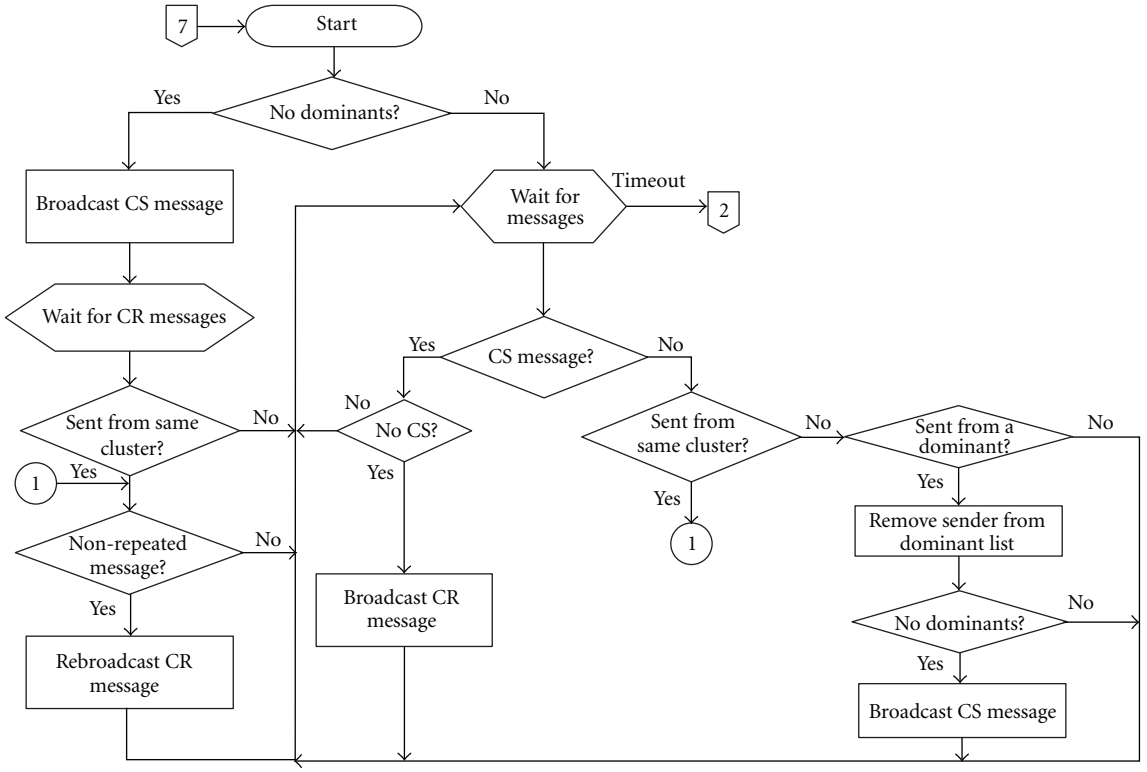


Figure 1: Flowchart of cluster border determination in step 1.

Since they are now separated from the original cluster, they need to restart the cluster formation. If the CS node is the real compromised node, it is likely to broadcast the CS message at this time. Otherwise, it is going to be separated alone.

(ii) Second, the node can remove the CS node from the member and neighbor list and restart the cluster formation to define a new cluster. In this case, because the non-receiver of the CS message does not need to send an affiliation request message, it can save its energy. If the CS node is the real compromised node, the receivers of the CS message have a malicious CS node in their cluster.

Regardless of taking any choice, the cluster is separated into two and one of them has a malicious CS node. Therefore, we select the second choice in order to save energy consumption. Following the second choice, some members might lose their CS node. If a node loses the connection with

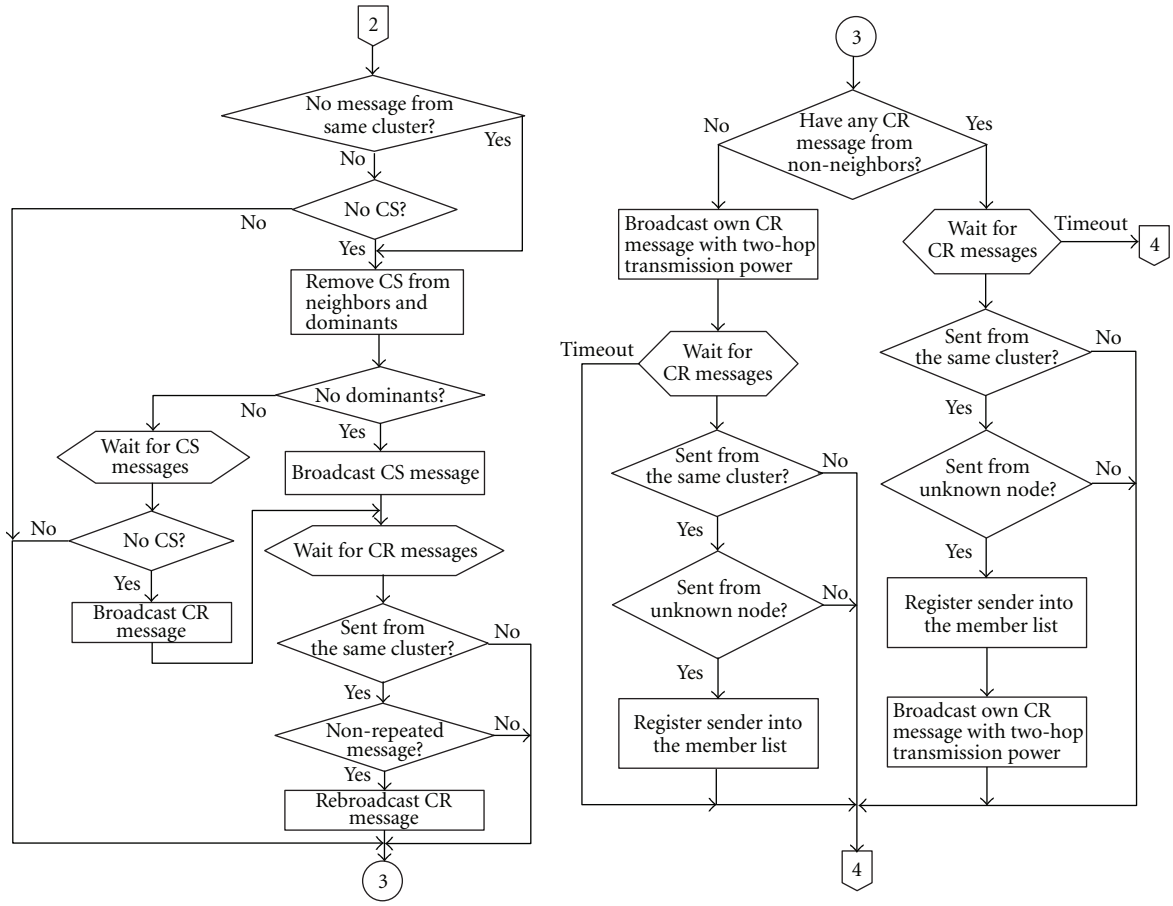


Figure 2: Flowchart of cluster border verification in step 1.

its CS node, it checks if there is no dominant in the neighborhood. If so, the node becomes a CS node and broadcasts a CS message. Figure 2 shows this countermeasure following the "no" control flow from the comparative symbol "No message from same cluster."

4.2. Final Cluster Formation

4.2.1. Merger of CS Nodes. A CS node broadcasts an FCS message using the received CR messages. The message consists of the message type and the list of the received CR messages. The CS node signs the message with its private key and transmits the signed message. Upon receiving an FCS message, the receiver verifies the signature and compares the list of CR messages with its own list. If they are exactly the same, the receiver registers the sender into the member list. Otherwise, the receiver discards the message.

4.2.2. Verification of Merged CS Nodes. After the merger of a CS node, each node checks if the CS node deviates from the protocol. If it identifies a deviation, it takes the following actions.

Attack Type 4. A CS node may remove some CR messages from its FCS (Final Cluster Separator) message and avoid the transmission to the removed nodes to exclude them from the cluster. In this case, the nonreceivers of the FCS message can employ the following countermeasures. Figure 3 shows the flowchart of the merger and verification of the CS node.

(i) If a node does not receive an FCS message from its CS node, it broadcasts a Solicitation message with two-hop transmission power.

(ii) A node which receives a Solicitation message checks if it has the sender's CR message. A compromised node may hide any misbehavior of the CS node by not responding to the Solicitation message even if it has the solicitor's CR message (Attack type 5). Otherwise, it makes a Solicitation Response message and responds to the solicitor with the message. First, it signs the solicitor's CR message with its private key and transmits the signed message in a unicast manner along with the FCS message and its certificate.

(iii) If a solicitor receives a Solicitation Response message, it first checks the FCS message, which is included in the message. The solicitor checks whether its CR message exists in the list of CR messages or not. Recall that a FCS message consists of CR messages received from members. So, if a FCS message misses the solicitor's CR message, this is unavoidable proof

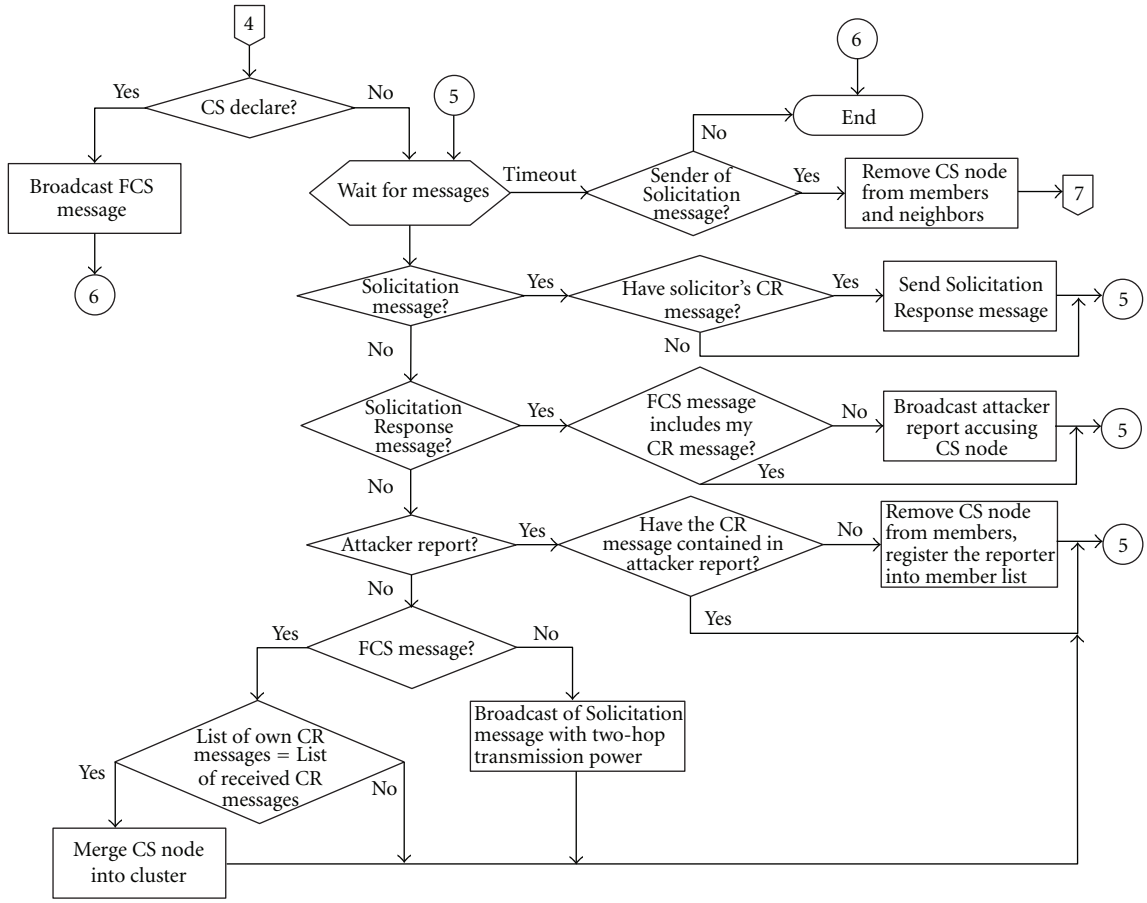


Figure 3: Flowchart of merger and verification of CS node.

that the CS node is a compromised node. In that case, the solicitor advertises the CS node as an attacker using a two-hop broadcast message, which is called an Attacker Report. The message consists of the signed CR message and the signer's certificate. If a solicitor is compromised, it may omit the transmission of the Attacker Report to avoid the exclusion of the CS node (Attack type 6).

(iv) If a node receives an Attacker Report, it verifies the message and checks if the witness really has received the CR message of which it proclaimed the reception. If so, it removes the CS node and registers the reporter into the member list. As a matter of fact, the node which receives the Attacker Report cannot assure that the accused CS node is really responsible for the nonreception of a CR message at any node. However, in any case, because the CS node is connected to all nodes in the cluster, it is most responsible for the nonreception.

(v) If a non-receiver of a FCS message receives no message even after it broadcasts a Solicitation message, it removes its CS node from the member and the neighbor list and restarts the protocol from the beginning.

4.3. Cluster Formation Example. We introduce an illustrative example to help the quick comprehension for our scheme. In step 1, nodes determine their cluster border. Figure 4(a) shows the initial process for the cluster border determination in step 1. Nodes 1, 4, and 5 become a CS node because they have no dominants in their neighborhood. So, they broadcast a CS message. However, a CS node (i.e., 5) deviates from the protocol by selectively transmitting its CS message. Node 5 does not send its CS message to nodes 9 and 27 to exclude them from the cluster. Nodes 9 and 27 cannot assure whether CS node 5 has deviated from the protocol or not until they receive a CR message heading for node 5 from another node in the cluster. Such a CR message can be a proof that CS node 5 broadcasted its CS message.

Figure 4(b) shows that each node receiving a CS message broadcasts a CR message in step 1. If a node receives a CR message and it is not a duplicate message, it rebroadcasts the message. However, a malicious node may avoid rebroadcasting the message. Node 4 carries out such an attack to hide nodes 30 and 40 from 29 and vice versa. To defeat this kind of attack, node 29 transmits its CR message with two-hop transmission power to find a hidden member because it does not receive a CR message from any two-hop member. Nodes 30 and 40 register the node 29 into their member list and broadcast their own CR message with two-hop transmission

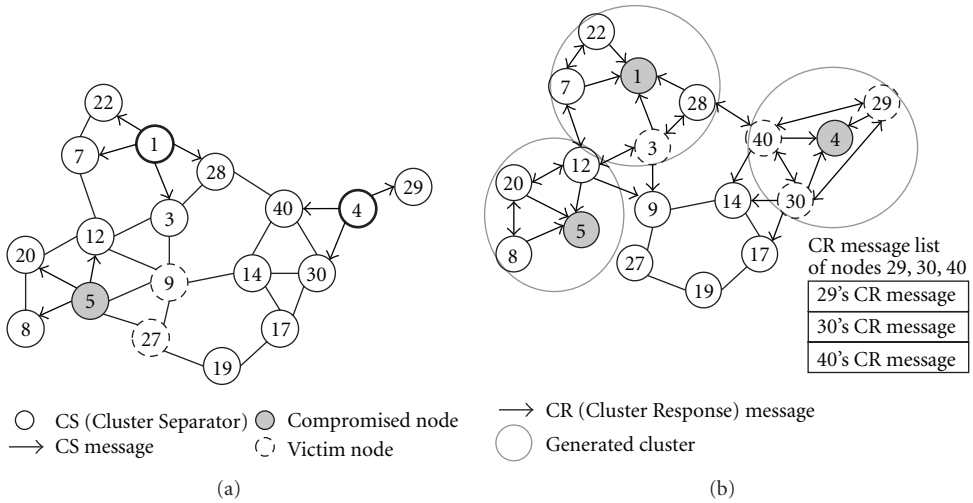


Figure 4: Cluster border determination in step 1 (a) Broadcast of CS message (b) Broadcast of CR message.

power. Node 29 also registers the nodes 30 and 40 into its member list. As a result, they share the same cluster mem-bership among them as shown in the right bottom of Figure 4(b).

Node 1 rejected the relay of the CR message of node 3 in order to exclude the node 3 from its cluster in Figure 4(b). However, the neighbor node 28 can get the CR message because it is directly connected with the node 3. As a result, the nodes 7 and 22 have a different view on cluster membership from nodes 3 and 28 as shown in Figure 5(a). Even though nodes 9 and 27 do not receive any CS message, they can receive a CR message from node 12 as shown in Figure 4(b). So, they can assure that CS node 5 avoided the transmission of its CS message to them. Therefore, they remove CS node 5 from their member list and neighbor list. Now, they restart the cluster formation process to define a new cluster border. Since node 9 no longer has any dominants, it declares itself as a CS node as shown in Figure 5(a). Upon receiving the CS message, nodes 14 and 27 broadcast their CR message to join the cluster as shown in Figure 5(b). Node 17 performs the same process as node 9 and the result is shown as Figure 6(a).

In step 2, nodes merge the CS node into their cluster if it conforms to the protocol. CS nodes trigger step 2 as in step 1. Nodes 1, 4, 5, 9, and 17 initiate step 2 by broadcasting an FCS message as shown in Figure 6(a). Since compromised nodes 4 and 5 behave as if they are normal nodes in Figure 6(a), we concentrate on the absorption of CS node 1. Recall that CS node 1 avoided rebroadcasting node 3's CR message in step 1 in order to separate it from the cluster. At the beginning of step 2, CS node 1 broadcasts its FCS message including the received CR messages. Of course, the CS node omits node 3's CR message to cheat other nodes and avoids the transmission toward node 3 as shown in Figure 6(a). Receivers 7 and 22 compare the list of CR messages with their own list. Because nodes 7 and 22 discover that they are exactly the same, they register node 1 into their member list. However, node 28 does nothing because it discovers that node 1's CR message list disagrees with its own list. Now, node 3 broadcasts a Solicitation message with two hop transmission power to obtain proof that it broadcasted its CR message as shown in Figure 6(b). Since receiver 28 has node 3's CR message, it first signs node 3's CR message by its private key and transmits the signed message along with 1's FCS message. The signed CR message of node 3 and the FCS message of node 1 constitute a Solicitation Response message as shown in Figure 6(b).

Upon receiving the Solicitation Response message, node 3 checks if there are unknown nodes in the FCS message. If so, it registers those nodes into its member list. Then, node 3 checks whether node 1's FCS message includes its CR message or not. Since node 1's FCS message does not include its CR message, it is definite proof of node 1's deviation from the protocol. So, node 3 reports node 1 as an attacker using a two-hop broadcast message as shown in Figure 7(a). The Attacker Report includes node 3's CR message which is signed by node 28's private key and node 28's certificate. Receivers 7, 22, and 28 verify the signature. If the verification succeeds, they remove node 1 from the member list and the neighbor list. This is because node 1 is a CS node, and it is connected to all nodes. That is, it is most responsible for the nonreception of 3's CR message at nodes 7 and 22. Nodes 7 and 22 register node 3 into their member list because they find a new normal node whose normalcy is guaranteed by node 28.

Now, all processes for cluster formation are completed. We have a clustered network such as Figure 7(b) after the completion of the protocol.

5. Evaluation

5.1. Security Analysis. Our scheme prevents external attackers from joining the cluster formation process by using message authentication. For such a reason, we focus on the insider attacks, which are launched by compromised nodes. If compromised nodes transmit the same false message or

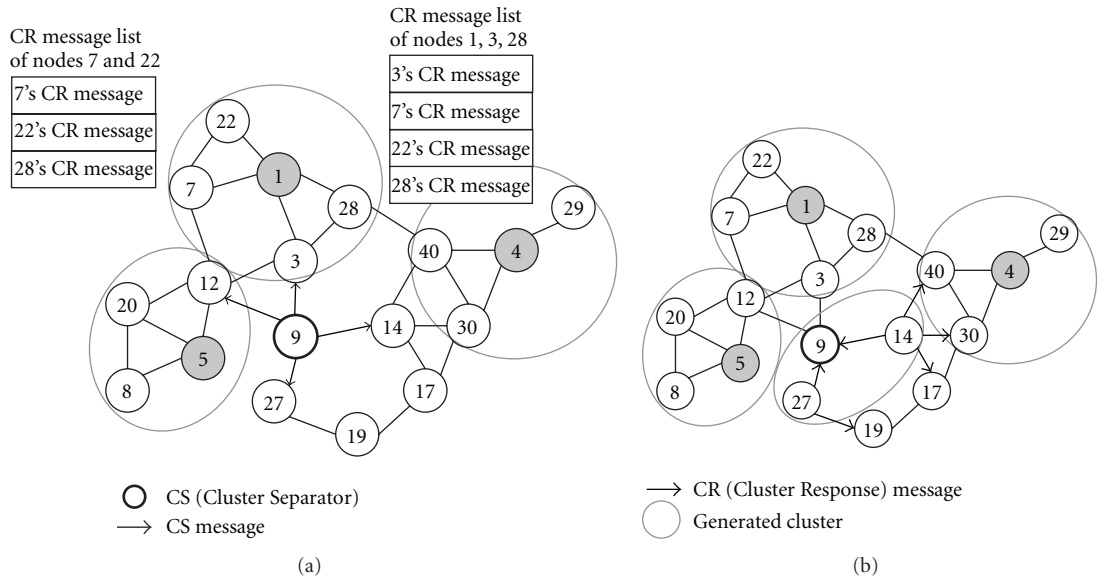


Figure 5: New cluster border determination in step 1 (a) Disconnection from existing CS node and new cluster border determination (b) Broadcast of CR message for new CS node.

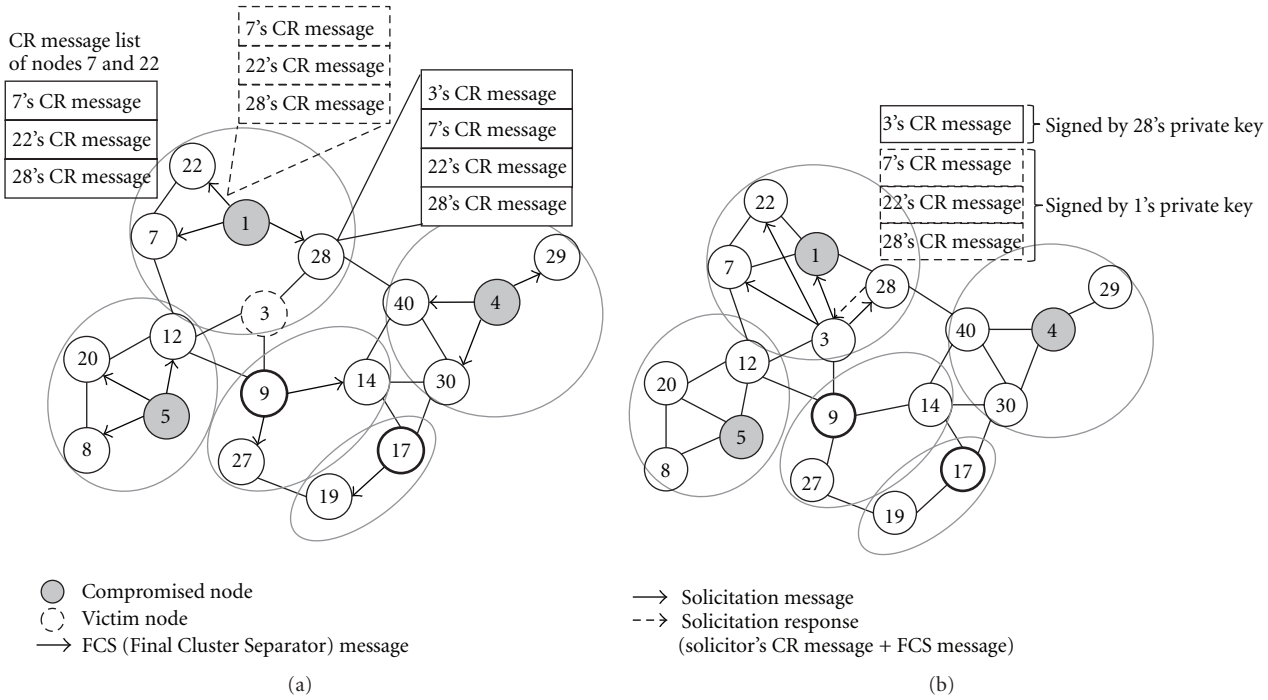


Figure 6: Merger of CS node and unknown normal nodes in step 2 (a) Broadcast of FCS message (b) Solicitation request and response.

avoid transmission to all members, they cannot create any inconsistency in the cluster membership. Therefore, any inconsistency in cluster membership can only result from transmitting different messages to different members or from avoiding the transmission by compromised nodes. In Section 5.1.1, we prove that compromised nodes are identified if they transmit inconsistent messages. In Section 5.1.2, we prove that our scheme preserves the agreement in cluster membership against the misbehavior of compromised nodes.

5.1.1. Identification of Compromised Nodes. First, we explain notations employed in the following. Suppose that nodes i, j, and k are normal members in a cluster and node l is the CS node in the cluster. We denote node i's view on cluster membership as Ci in the following. We introduce Lemma 1 and employ it to prove Theorem 2.

Lemma 1. *For two normal members i and j, if $k \in Ci$, then we should have $k \in Cj$.*

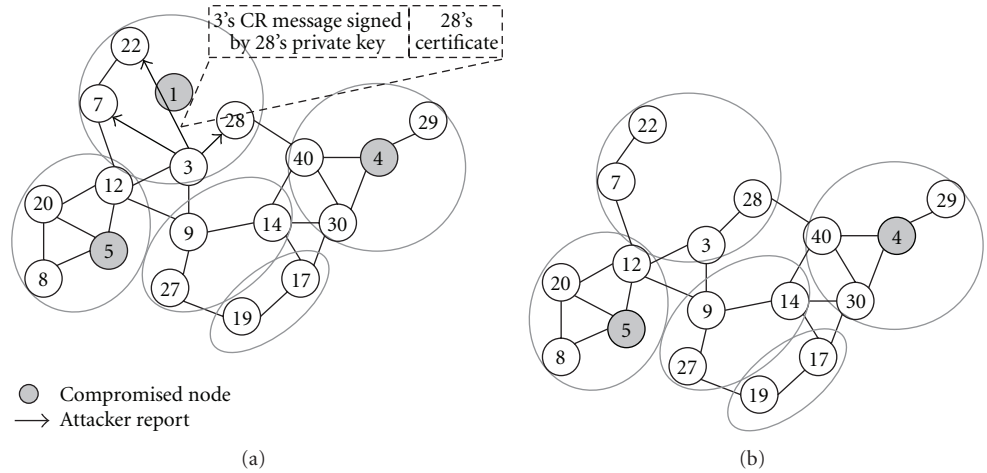


Figure 7: Attacker report and completion of cluster formation (a) Attacker report (b) Completion of cluster formation.

Proof. We prove it by contradiction. First, we suppose $k \notin Cj$. If the supposition is true, we should have $k \notin Cj$ in the first step of our scheme. If $k \notin Cj$ in the first step of our scheme, we also have $k \notin Cj$. Therefore, it contradicts the condition $k \in Ci$. □

Theorem 2. *If any inconsistency in cluster membership is caused by compromised nodes which transmit inconsistent messages to members, our scheme can identify the compromised nodes.*

Proof. As shown in Lemma 1, if $k \in Ci$, then we have $k \in Ci$. Therefore, if we identify any inconsistency in cluster membership, it has been definitely caused by the CS node l which is connected to all members i, j, and k. When the node k identifies inconsistency in cluster membership, the node k transmits a Solicitation message and node i responds to the Solicitation message. Using the Solicitation Response message, we come to know that the CS node l transmits inconsistent messages to members. □

5.1.2. Preservation of Cluster Membership Agreement. In our scheme, the silence attacks can only cause any inconsistency in membership in the second step. First, if a CS node causes any inconsistency in membership during the first step, the victims are separated from the original cluster. Since the victims form a cluster for themselves, the separation and the formation cannot affect the final cluster membership. Second, if some compromised nodes do not relay CR messages in the first step, members themselves agree on the cluster membership with each other through a two-hop broadcast of the CR message. In the following Theorem 3, we prove that our scheme preserves the agreement of cluster membership even if compromised nodes launch a silence attack or a selective transmission attack to certain members.

Theorem 3. *For a CS node l and two normal members i and j, if $l \in Cj$, then we must have $Ci = Cj$.*

Proof. We prove it by contradiction. First, we suppose that $Ci \neq Cj$. Without loss of generality, we assume that $k \in Ci$ but $k \notin Cj$. In this case, node j identifies the CS node l's misbehavior through the node k's Attacker Report. Then node j removes the CS node l from its member list and registers the node k into its member list. Therefore, we have $l \notin Cj$. It is contrary to the condition $l \in Cj$. □

5.2. Simulation Results. In this subsection, we evaluate the security and energy-efficiency of our scheme using the ns-2 network simulator. In the simulation environment, 100 nodes were randomly deployed in a 100 m × 100 m square field. Each node consumed their energy using the energy model of [1] during the cluster formation. Table 2 shows the simulation parameters and their values. We compare our scheme with Sun's scheme [12] because its goal and method are most similar with our scheme. Even though some schemes [9–11] present their secure cluster formation techniques, they are excluded from comparison because their methodology and attackers' aim are significantly different from our scheme. Rifà-Pous' scheme is similar to our scheme in terms of cluster formation methodology but it has no countermeasures against compromised nodes which deviate from the protocol. So, comparison with Rifà-Pous' scheme is unfair.

In Figure 8(a), we demonstrate how many clusters the two schemes generate when compromised nodes launch silence attacks and selective transmission attacks on the network. The result shows their resiliency when they are under attack. As shown in Figure 8(a), both schemes increase the number of clusters as the number of compromised nodes increases. Our scheme greatly reduces the generated clusters as compared with Sun's scheme, especially under a small number of compromised nodes. This result is caused by two facts. First, our scheme initially reduces the number of generated clusters by generating large-sized clusters. In addition, our scheme minimizes the separation of the clusters even if the clusters are under an attack.

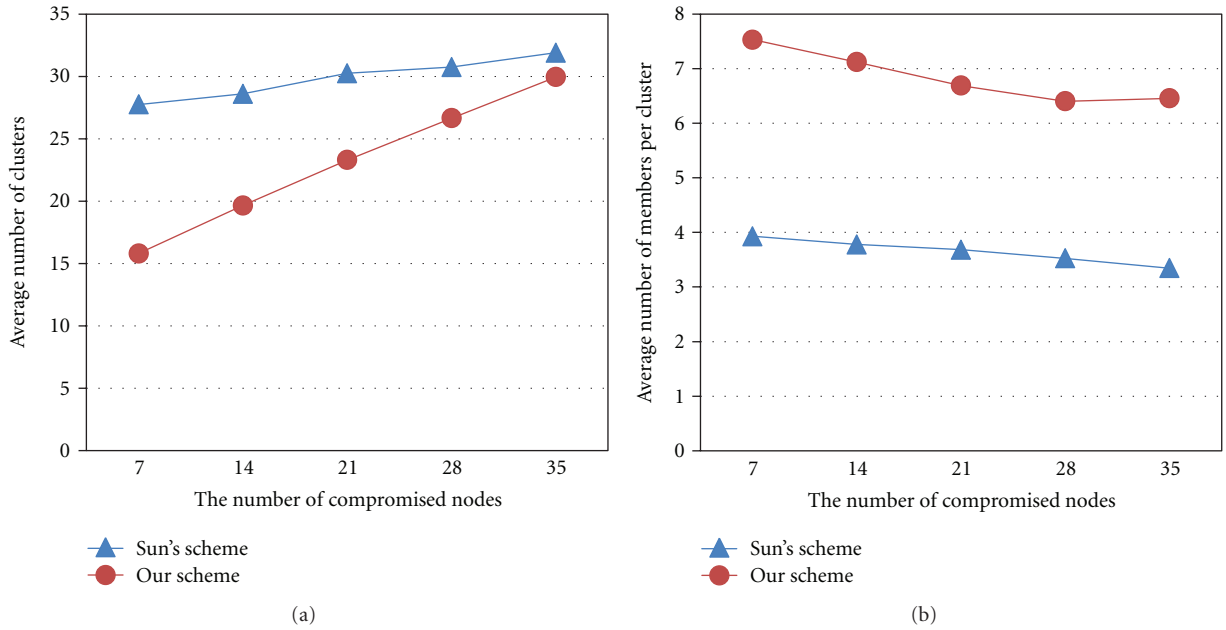


Figure 8: Variation in the number of clusters and the number of members per cluster (a) Avg. number of clusters versus compromised nodes (b) Avg. number of members per cluster versus compromised nodes.

Table 2: Simulation parameters.

Parameter	Value
Simulation area	100 m. × 100 m.
Number of nodes	100
Number of compromised nodes	7~35
Initial energy	20 Joules
Energy consumption model	Energy model of [1]
Bandwidth	1 Mbps
Packet header size	25 bytes
Transmission range	25 meters
Signature algorithm	ECDSA (elliptic curve digital signature algorithm)-160
Data encryption and decryption algorithm	AES (advanced encryption standard)-128
Hash Algorithm	SHA-1

Figure 8(b) shows how two schemes affect the average size of clusters. The result shows the quality of clusters when the clusters are under attacks. As shown in Figure 8, both schemes deteriorate the quality of clusters as the number of compromised nodes increases. Sun's scheme excludes a node whenever the node is identified as a compromised node or a suspicious node. So, it gradually deteriorates the quality of clusters as the number of compromised nodes. Our scheme keeps the quality of clusters higher than Sun's scheme because it separates much fewer nodes from clusters compared to Sun's scheme.

When a normal node identifies that a node has deviated from the protocol, it is likely to exclude the suspicious node from the cluster. As a result, many single clusters and double clusters are generated during the cluster formation. A single cluster consists of only one node, and a double cluster consists of only two nodes. Note that the clusters in Figure 8(a) do not include single clusters. Single and double clusters are almost meaningless in the viewpoint of clustering, and we label them as bad clusters. Figures 9(a) and 9(b) show how many bad clusters two schemes generate. As shown in Figures 9(a) and 9(b), Sun's scheme generates more bad clusters than our scheme. Especially, because Sun's scheme generates more single clusters than our scheme, we can say that it generates more useless clusters in comparison to our scheme.

Figure 10(a) shows how well two schemes remove compromised nodes from clusters during the protocol. The survived attackers in Figure 10(a) signify the nodes which have been compromised during the cluster formation and have survived until the end of the protocol. The result represents the healthiness of generated clusters in two schemes. Sun's scheme outperforms our scheme under a small number of compromised nodes (i.e., 7). Sun's scheme expels compromised nodes well through its one-hop conformity check when their population is small. However, because compromised nodes are not going to respond to normal nodes' protocol conformity check, the increase of compromised nodes significantly deteriorates performance. Whenever a node finds any inconsistency in the cluster membership, it attempts to obtain proofs from neighboring witnesses. However, if all witnesses are compromised, the node can never obtain any proof. So, the compromised node which causes the inconsistency can survive the protocol conformity check.

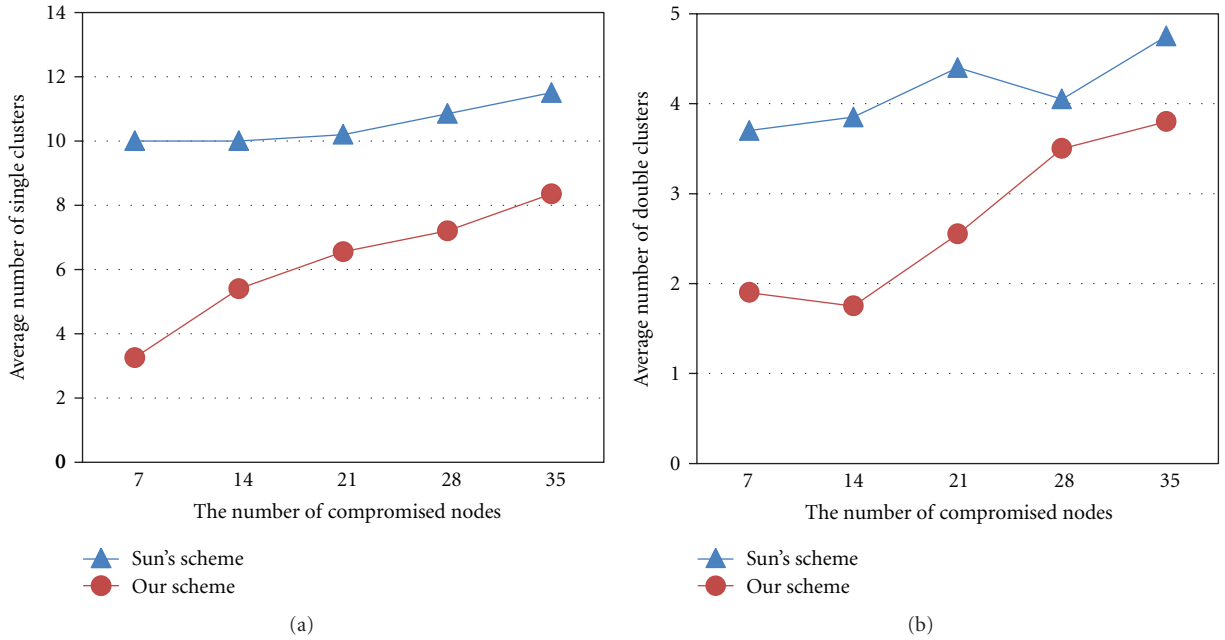


FIGURE 9: Variation in the number of bad clusters (a) Avg. number of single clusters versus compromised nodes (b) Avg. number of double clusters versus compromised nodes.

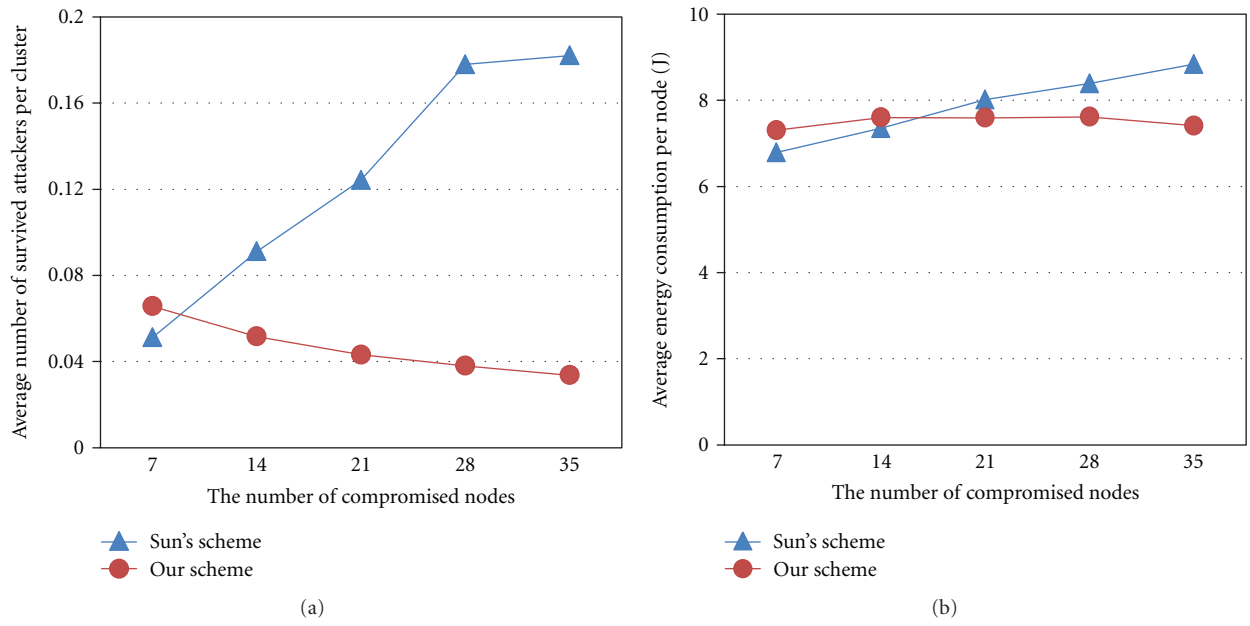


FIGURE 10: Variation in the number of survived attackers and energy consumption per node (a) Avg. number of survived attackers per cluster versus compromised nodes (b) Avg. energy consumption per node versus compromised nodes.

That is, the survival rate of compromised nodes increases when the population of compromised nodes grows up. However, our scheme employs two-hop transmissions to check the protocol conformity of a node. So, our scheme can obtain more proofs than Sun's scheme, and it can expel the compromised nodes regardless of the unresponsive nodes. Especially, as the number of compromised nodes increases, our scheme removes more compromised nodes from the network using the two-hop transmissions as shown in Figure 10(a).

Figure 10(b) shows how the increase of compromised nodes affects the energy consumption at each node. In Sun's scheme, if a compromised node causes any inconsistency of cluster membership between any two nodes, a normal node checks the protocol conformity of the node with the inconsistent membership. During the protocol conformity check

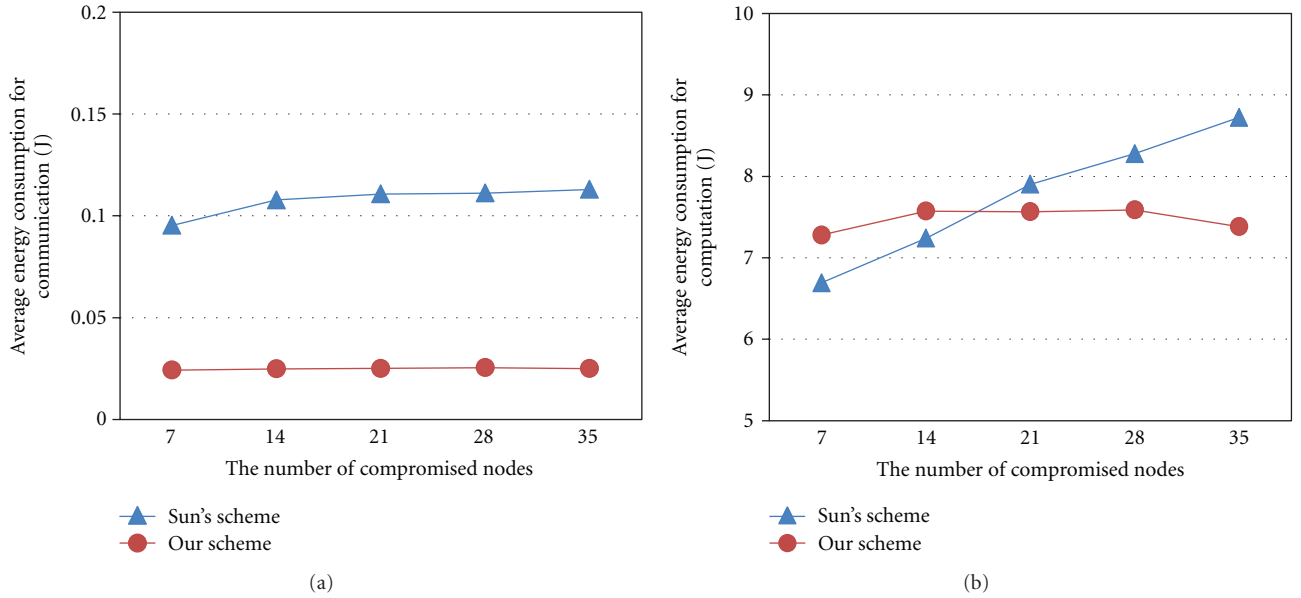


Figure 11: Variation in energy consumption for communication and computation (a) Avg. energy consumption for communication versus compromised nodes (b) Avg. energy consumption for computation versus compromised nodes.

process, the normal node requests the neighboring members to provide the messages previously received with a signature in a unicast manner. So, if compromised nodes increase, more normal nodes find the inconsistency, and they all begin the protocol conformity check of neighboring members. Consequently, the neighboring members consume much more energy than less compromised nodes. Even though our scheme increases the energy consumption of nodes under a small number of compromised nodes, it is a transient increase. As the number of compromised nodes increases, our scheme greatly reduces energy consumption of nodes as shown in Figure 10(b). This is because our scheme suppresses the employment of unicast transmissions and mainly employs broadcast transmissions.

Figure 11(a) shows the amount of energy each node consumed for communication during the protocol. As shown in Figure 11(a), Sun's scheme slightly increases energy consumption as the number of compromised nodes increases. Our scheme greatly decreases energy consumption as well as preserves the amount of energy consumption constantly regardless of the population of compromised nodes Figure 11(b) shows the amount of energy each node consumed for computation during the protocol. Both schemes consumed much more energy for computation than that for communication. It shows that the total energy consumption of both schemes highly depends on the energy consumption for the computation of both schemes. As shown in Figure 11(b), Sun's scheme incrementally increases the amount of energy consumption as the number of compromised nodes increases. This is mainly caused by the increase of one-hop conformity checks. Our scheme preserves the amount of energy consumption almost constantly regardless of the increase of compromised nodes. As a result, it greatly reduces the energy consumption of nodes especially under many compromised nodes.

Now, we compare the storage overhead of two schemes. Let Ni be the number of neighbors of node i. In Sun's scheme, because each node i should store the messages of all neighbors in each step, and the messages are received from the neighbors during the four steps, its storage overhead becomes $4Ni$ messages. If node i detects that a neighbor j has inconsistent cluster membership, it should receive an extra message from j and store it to complete the verification. In this case, node i's storage overhead is $4Ni + 1$ messages. Let M be the number of members in a cluster. So, we have an inequality of $4Ni > M(\approx 2Ni) > Ni$ according to the result of Figure 8(b). In our scheme, each node first stores Ni messages which are sent from its neighbors due to the exchange of certificates. Besides, each node i should store all CR messages sent from its members in order to build a consensus on the cluster membership. So, each node's storage overhead is $M + Ni$ messages. When a normal node does not receive any FCS message, it requests other members to send a proof of its normalcy. If the node succeeds in obtaining the proof, it should also store the proof message to persuade other members of the CS node's misbehavior. In this case, the node's storage overhead is $M + Ni + 1$ messages. According to the above analysis, our scheme's storage overhead is lower than Sun's scheme.

6. Conclusions

In this paper, we proposed a cluster formation scheme, which generates large-sized clusters in a secure manner. Our

scheme initially produces large sized clusters and preserves them using the two-hop conformity verification. Then, our scheme merges the cluster formation initiators into clusters only when they conform to the cluster formation protocol. Security analysis proves that our scheme can easily identify compromised nodes which cause any inconsistency in cluster membership and preserves the agreement in cluster membership against the compromised nodes. Simulation results show that our scheme reduces the number of generated clusters, and it suppresses the generation of useless clusters. Other simulation results prove that our scheme enhances the quality of clusters and expels more compromised nodes than the compared scheme. The simulation for energy efficiency evaluation proves that our scheme greatly reduces the amount of energy consumption, especially under many compromised nodes.

Acknowledgment

This work was supported by research funds of Chonbuk National University in 2011.

References

[1] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.

[2] O. Younis and S. Fahmy, "HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, 2004.

[3] G. Gupta and M. Younis, "Performance evaluation of load-balanced clustering of wireless sensor networks," in *Proceedings of the 10th International Conference on Telecommunications (ICT '03)*, vol. 2, pp. 1577–1583, March 2003.

[4] G. Wang, K. S. Song, and G. Cho, "DIRECT: dynamic key renewal using secure cluster head election in wireless sensor networks," *IEICE Transactions on Information and Systems*, vol. E93-D, no. 6, pp. 1560–1571, 2010.

[5] G. Wang and G. Cho, "Clustering-based key renewals for wireless sensor networks," *IEICE Transactions on Communications*, vol. E92-B, no. 2, pp. 612–615, 2009.

[6] M. Sirivianos, D. Westhoff, F. Armknecht, and J. Girao, "Non-manipulable aggregator node election protocols for wireless sensor networks," in *Proceedings of the 5th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt '07)*, pp. 1–10, April 2007.

[7] Q. Dong and D. Liu, "Resilient cluster leader election for wireless sensor networks," in *Proceedings of the 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON '09)*, pp. 108–116, June 2009.

[8] G. Wang and G. Cho, "Secure cluster head election using mark based exclusion inwireless sensor networks," *IEICE Transactions on Communications*, vol. E93-B, no. 11, pp. 2925–2935, 2010.

[9] A. C. Ferreira, M. A. Vilaca, L. B. Oliveira, E. Habib, H. C. Wong, and A. A. Loureiro, "On the security of cluster-based communication protocols for wireless sensor networks," in *Proceedings of the 4th IEEE International Conference on on Networking*, pp. 449–458, 2005.

[10] L. B. Oliveira, H. C. Wong, M. Bern, R. Dahab, and A. A. F. Loureiro, "SecLEACH-A random key distribution solution for securing clustered sensor networks," in *Proceedings of the 5th IEEE International Symposium on Network Computing and Applications (NCA '06)*, pp. 145–152, July 2006.

[11] D. Liu, "Resilient cluster formation for sensor networks," in *Proceedings of the 27th International Conference on on Distributed Computing Systems (ICDCS '07)*, pp. 40–48, 2007.

[12] K. Sun, P. Peng, P. Ning, and C. Wang, "Secure distributed cluster formation in wireless sensor networks," in *Proceedings of the 22nd Annual Computer Security Applications Conference (ACSAC '06)*, pp. 131–140, December 2006.

[13] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "SPINS: security protocols for sensor networks," *Wireless Networks*, vol. 8, no. 5, pp. 521–534, 2002.

[14] H. Rifà-Pous and J. Herrera-Joancomartí, "A fair and secure cluster formation process for Ad hoc networks," *Wireless Personal Communications*, vol. 56, no. 3, pp. 625–636, 2011.

[15] Y. C. Hu and A. Perrig, "Wormhole attacks in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 370–379, 2006.

[16] A. S. Wandert, N. Gura, H. Eberle, V. Gupta, and S. C. Shantz, "Energy analysis of public-key cryptography for wireless sensor networks," in *Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications (PerCom '05)*, pp. 324–328, March 2005.

[17] D. R. Raymond and S. F. Midkiff, "Denial-of-service in wireless sensor networks: attacks and defenses," *IEEE Pervasive Computing*, vol. 7, no. 1, pp. 74–81, 2008.

[18] J. R. Douceur, "The sybil attack," in *Proceedings of the 1st International Workshop on Peer-to-peer Systems*, March 2002.

[19] B. Parno, A. Perrig, and V. Gligor, "Distributed detection of node replication attacks in sensor networks," in *Proceedings of the 2005 IEEE Symposium on Security and Privacy (S&P '05)*, pp. 49–63, May 2005.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 243841, 14 pages
doi:10.1155/2012/243841

Research Article

Toward Intelligent Intrusion Prediction for Wireless Sensor Networks Using Three-Layer Brain-Like Learning

Jun Wu,[1] Song Liu,[1] Zhenyu Zhou,[1] and Ming Zhan[2]

[1] *Global Information and Telecommunication Institute, Waseda University, Tokyo 169-0051, Japan*
[2] *National Key Laboratory of Science and Technology on Communications, University of Electronic Science and Technology of China, Chengdu 611731, China*

Correspondence should be addressed to Jun Wu, junwu@akane.waseda.jp

Received 16 June 2012; Accepted 23 August 2012

Academic Editor: Mihui Kim

The intrusion prediction for wireless sensor networks (WSNs) is an unresolved problem. Hence, the current intrusion detection schemes cannot provide enough security for WSNs, which poses a number of security challenges in WSNs. In many mission-critical applications, such as battle field, even though the intrusion detection systems (IDSs) without prediction capability could detect the malicious activities afterwards, the damages to the WSNs have been generated and could hardly be restored. In addition, sensor nodes usually are resource constrained, which limits the direct adoption of expensive intrusion prediction algorithm. To address the above challenges, we propose an intelligent intrusion prediction scheme that is able to enforce accurate intrusion prediction. The proposed scheme exploits a novel three-layer brain-like hierarchical learning framework, tailors, and adapts it for WSNs with both performance and security requirements. The implementation system of the proposed scheme is designed based on agent technology. Moreover, an attack experiment is done for getting training and test data set. Experiment results show that the proposed scheme has several advantages in terms of efficiency of implementation and high prediction rate. To our best knowledge, this paper is the first to realize intrusion prediction for WSNs.

1. Introduction

Wireless sensor networks (WSNs) have become a technology for the new millennium with endless applications ranging from civilian to military [1–3]. A wireless sensor network is consisted of a large number of wireless-capable sensor devices working collaboratively to achieve a common objective. As a matter of fact, WSNs are often deployed in potentially adverse or even hostile environments where adversaries can launch various kinds of attacks [3–5]. The nodes of WSNs are vulnerable to these attackers, because unmanned sensors are often deployed through open medium and dynamic network topology. Intrusion detection is a security technology that attempts to identify those who are trying to break into and misuse a system without authorization and those who have legitimate access to the system but are abusing their privileges. Recently, the problem of intrusion detection in WSNs has received considerable attention [4–15].

In the existing intrusion detection schemes of WSNs [5–15], two approaches have been used: signature-based detection and anomaly detection. Signature-based detection [7–12] lies in the monitoring of system activity and the identification of behaviors which are similar to pattern signatures of known attacks or intrusions stored in a signature database. This category of intrusion detection systems (IDSs) detects accurately known attacks, and the signatures are often generalized in order to detect the many variations of a given known attack. But this generalization leads to the increase of false positives (i.e., false alarms). The main limitation of such IDSs concerns their incapability to detect unknown intrusions that are not already present in the signature database.

On the other hand, anomaly detection systems [6, 13–15] detect attacks by observing deviations from a preestablished normal system or user behavior. This approach makes detecting new or unknown attacks, if these attacks imply an abnormal use of the system. The main difficulty in implementing

reliable anomaly detection systems is the creation of the normal behavior model. Since it is difficult to define correctly these models and only incomplete or incorrect models can be obtained, which leads to false negatives or false positives.

However, there is an important limitation of the existing intrusion detection schemes in WSNs, which is shown as follows. The existing intrusion detection schemes of WSNs have no concept of intrusion prediction. In many mission-critical applications, such as battle field, some attack processes are executed in a very short time [5] when the threat environment for WSNs includes a well-resourced adversary. Even though IDSs can detect these malicious activities afterwards, damages could have been done to the compromised WSNs which could hardly be restored in some cases. Therefore, it is very important to develop algorithms and tools to track and predict attacks in advance to remove potential threats. The intrusion prediction mechanisms for existing applications, such as computer networks, grid computing systems, and automated substation, are developed to predict various attacks, but cannot be applied directly to WSNs [16–21].

Recently, in the intrusion detection community, interest has been growing applying machine learning techniques to get high performances in execution time and overall classification accuracy [22–24]. Machine learning is a technology which is concerned with the design of algorithms that allow systems to evolve behaviors based on empirical data. A learner can take advantage of examples (data) to capture characteristics of interest of their unknown underlying probability distribution. Data can be seen as examples that illustrate relations between observed variables. A major focus of machine learning research is to automatically learn how to recognize complex patterns and to make intelligent decisions based on data. Hence, the learner must generalize from given examples, so as to be able to produce a useful output in new cases. Machine learning based intrusion detection for WSNs [25] has gained limited attention so far, not to mention intrusion prediction or implementation on the current generation of sensor nodes.

From the above discussion, it is clear that achieving prediction with high accuracy using machine learning is still an open challenge in WSNs. Towards addressing this challenge, we proposed in this paper a machine learning based intelligent intrusion prediction scheme. By exploring a three-layer brain-like hierarchical learning model, we proposed a novel intelligent intrusion prediction scheme, namely, BLID, which is specially tailored for WSNs. We based our design on the observation of the inherent nature of WSNs that different nodes own different recourses. Hence, we design this intelligent intrusion prediction as a hierarchical model. In the proposed scheme, supervised learning with relatively low complexity is performed in the resource-restrained sensors. Inversely, unsupervised learning and reinforcement learning are implemented in the sinks and base stations which have powerful resources. The learning modules in different layers can interact with each other. Our solutions have several advantages. First, BLID is efficient in terms of storage, computation, and communication overhead on the sensor side. Most important, BLID can perform intrusion prediction. To the best of our knowledge, the design of intrusion prediction in WSNs has not been addressed in previous work.

In summary, our paper makes the following contributions. (1) It introduces the intrusion prediction problem for the first time in WSNs. (2) The proposed scheme applies and tailors brain-like hierarchical learning to WSNs for achieving intrusion prediction with high accuracy. (3) The implementation of BLID is simulated on the current generation of sensor nodes.

The rest of this paper is organized as follows. Section 2 describes the system model and assumptions as well as some technical preliminaries on which our scheme is based. Section 3 presents the proposed scheme in detail. Section 4 describes the wireless attack experiment through which we get the training and testing data set for evaluating our scheme. In Section 5, we evaluate our scheme in terms of efficiency and accuracy. Finally, we conclude this paper in Section 6.

2. Models and Assumptions

2.1. Network Model. In this work, we consider a WSN with three-layer structure which includes base station layer, sink layer, and sensor layer. This structure is a popular way for deploying WSNs [5, 26–28]. Usually, the sensor nodes are scattered in a sensor field, and each sensor can collect data and deliver the data to the sink or base station (BS). Sensor nodes are static most of the time, whereas mobile nodes can be deployed according to application requirements. One or several base stations (BSs) can be deployed together with the network. BSs and sinks can be either static or mobile. BS acts as an interface between the WSN and the external world.

Data storage and access in WSNs mainly follows two approaches which are centralized and distributed approaches [29]. In centralized scheme, sensed data are collected from individual sensors and transmitted back to the sink, for storage and access. In the distributed case, the sensors store the data locally or at some designated nodes within the network. The stored data can be further accessed in distributed manner by the users of the WSN. BS, sink, and sensor are the access points (AP) when users access the data in the WSN. An access scenario is illustrated in Figure 1. Local users can access WSNs through wireless links directly. However, remote users need to access the WSN through satellite, Internet, or mobile network.

2.2. Intrusion Model. This paper considers that adversaries could be either external intruders or unauthorized network users. Due to lack of physical protection, sensor nodes are usually vulnerable to strong attacks. In particular, we consider the adversary with both passive and active capabilities, which can (1) eavesdrop all the communication traffics in the WSN, and (2) compromise and control a small number of sensor nodes. In addition, (3) unauthorized users may collude to compromise the encrypted data.

2.2.1. Wire Intrusions. The base station can act as an interface between the WSNs and other communication network,

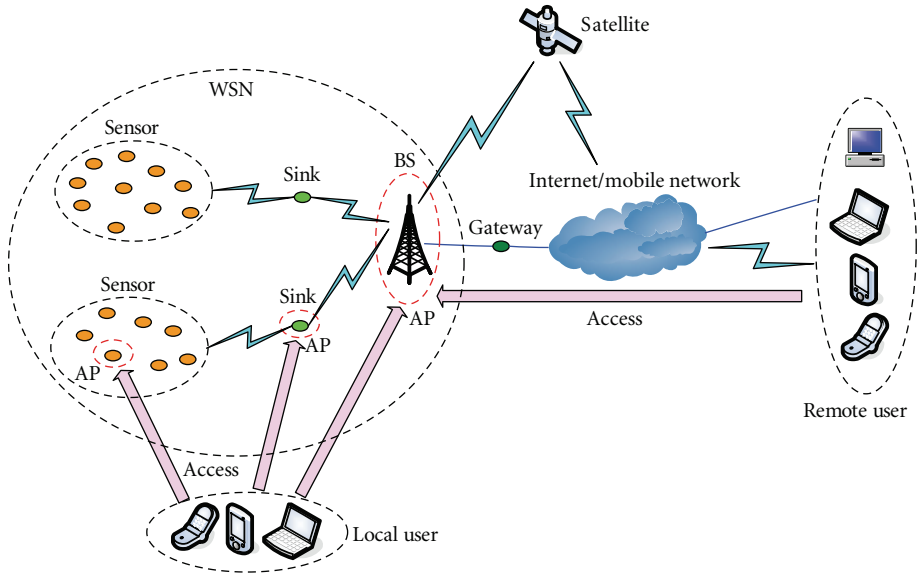


Figure 1: An access scenario of a WSN.

which is most likely to be Internet. In other words, remote users usually access the data in the base station through Internet. Hence, the intrusion attacks of Internet can present to the base station. There have been many studies focused on intrusion detection and prediction for Internet [16, 18, 20–22, 24, 25]. The Internet intrusions present to base station can be resolved on these schemes.

2.2.2. Wireless Intrusions. Comparing with wire networks, wireless networks face more intrusions because the wireless communication medium is open physically to adversaries. Various adversaries can attack wireless networks through wireless links [30]. More seriously, a lot of free tools are available on the Internet that allows novice hackers to exploit wireless protocol weaknesses to deny access to a network. All these facts raise the challenge of intelligent intrusion prediction for WSNs.

The wireless communication infrastructure of WSNs is the choice of application. Because many existing WSNs are deployed by IEEE 802.11 and mote device technologies [31], we consider in this paper IEEE 802.11 as the wireless communication infrastructure of WSNs. In WSNs, all the sensors, the sink, and the base station can act as the wireless access points; hence, all the three kinds of nodes could be intruded by the wireless attacks. There is a free collection of tools to attack 802.11-based networks available for download on Internet [32]. These tools operate on WEP and WPA-protected networks.

In this paper, we take four kinds of attacks, for example, doing experiments, which are ARP replay attack, forgery attack, ongoing dictionary attack, and chopchop attack. These attacks are the common attacks in 802.11 networks [33, 34].

2.3. Preliminaries. This section briefly describes the technique preliminaries on which our scheme is designed.

2.3.1. Brain-Like Hierarchical Learning. Recently, brain-like learning and computation have attracted a lot of attentions in the area of machine learning. Our brain is a highly complicated structure and there have been many studies focused on brain-like learning [35–38]. In this paper, we consider the brain-like learning model in [37], which is developed into a system structure in [38]. This brain-like model is based on the fact that the cerebellum is a specialized organism for supervised learning (SL), the basal ganglia are for reinforcement learning (RL), and the cerebral cortex is for unsupervised learning (UL). In the framework, a particular function, such as the control of arm movement, can be realized by a global network combing different learning modules in the cerebellum, the basal ganglia, and the cerebral cortex. The three learning modules of brain-like learning are described as follows.

Supervised Learning (SL) in the Cerebellum. This learning module is which constructs an input-output mapping. It is characterized by the parameter update based on the correlation between the output error and the presynaptic input.

Unsupervised Learning (UL) in the Cerebral Cortex. This learning module is characterized by the relaxation dynamics for determining the output as well as the Hebbian synaptic rule under a certain regularization.

Reinforcement Learning (RL) in the Basal Ganglia. This learning module is concerned with how an agent ought to take actions in an environment so as to maximize some notion of cumulative reward.

2.3.2. Agent Technology. The agent technology is an important technique in recent researches of the artificial intelligence [39]. In the area of WSNs, a lot of new works

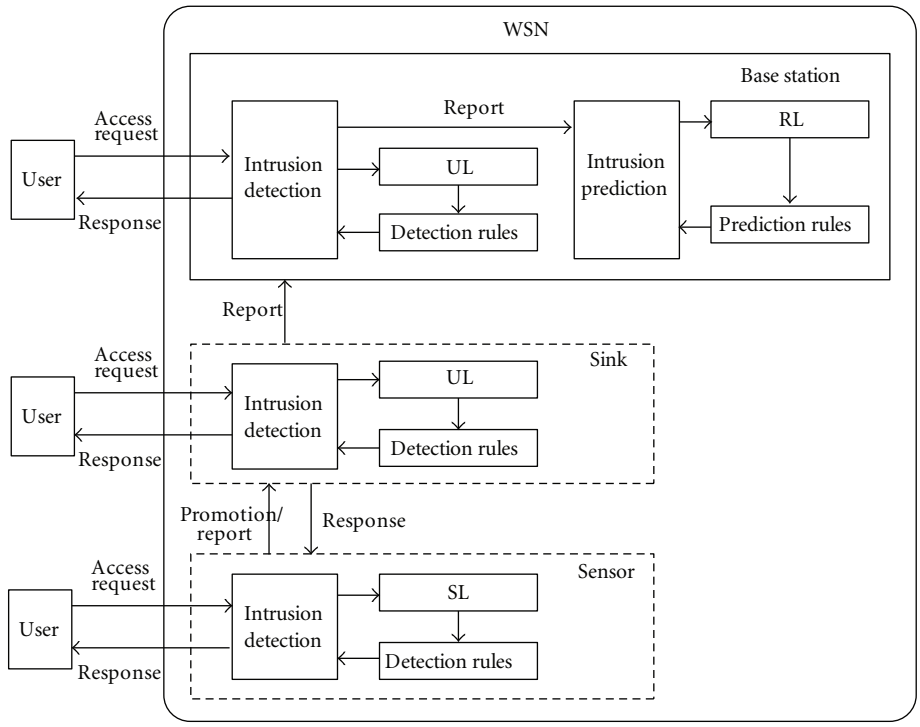


Figure 2: Systemic design of BLID.

introduce agent technology into WSNs [40–42]. Using agent technology in WSNs leads to a number of advantages, such as scalability, event-driven actions, task-orientation, and adaptivity.

3. Brain-Like Hierarchical Learning Intelligent Intrusion Prediction Scheme

This section presents the proposed intelligent intrusion prediction scheme for WSNs. We first introduce the systemic design. Then, we present the detailed description of our basic scheme, which is followed by an advanced design.

3.1. Systemic Design. In this section, we set up the system model of BLID, as shown in Figure 2. The basic idea of the intelligent intrusion prediction scheme is *distributed detection and centralized prediction*. Every node of the WSN can perform intrusion detection. However, only base station can perform intrusion prediction for the whole WSN.

Considering the limited resources of sensors and the powerful resources of sink as well as base station, we define two levels of intrusion detection: (1) supervised learning based detection and (2) unsupervised learning based detection. The supervised learning based detection is a low level detection which is performed in sensors. This part is corresponding with the cerebellum of the brain. On the other hand, the unsupervised learning based detection is a high level detection which is performed both in sinks and base station. This part is corresponding with the cerebral of the brain. If some unknown attacks occur to a sensor, the sensors will send the unknown features to the sink. This operation is marked as "*promotion.*" Then the sink will determine whether the access is an attack or not by its high level rules. If the sink cannot identify based on current rules, it will adaptively update the rules based on unsupervised learning. Then the sink sends the response to the sensor. In short, that sink and the base station perform intrusion detection by theirselves. The sensor performs low level detection by itself, but it needs the help of sink for performing high level detection.

In the WSNs, only the base station can perform intrusion prediction which is based on reinforcement learning. This part is corresponding with the basal ganglia of the brain. In case of an intrusion, the sensors and sinks send the related features of the attack to the base station through a "*report*" operation. Similarly, the detection modular in the base station reports every local intrusion to prediction module. Note that sensor reports every intrusions to base station via sink because sensors cannot communicate directly with the base station.

Through the basic idea above, a particular detection or prediction function can be realized by a global network combing different learning modules in sensor, sink, and base station.

3.2. Supervised Learning Based Intrusion Detection in Sensor. Decision tree is a kind of classifier for supervised learning. In order to perform supervised learning with low complexity, we use decision tree (DT) as a classifier for data analyzing. Usually, there are three criteria for constructing a decision

tree: the information gain, the gain ratio, and the Gini index [43].

There are three steps to design decision tree based intrusion detection. The first step is defining and initializing variables that will be used in the ensuing process. The second step is defining a set of primary detection rules. A detection rule contains a set of keywords that must be checked to trigger an alarm. Finally, the third step is defining a set of primary action rules that describe the behaviour after analyzing the attribute data. The core part is how to construct a decision tree.

The decision tree construction scheme for sensor must have low complexity because the resources of sensors are limited.

The decision tree in our scheme contains three types of nodes: ordinary, leaf, and promotion nodes. Each node is represented by $N\ (A, D, M)$ where A is an attribute set, D is a set of detection rules, and C is a set of countermeasure. The attribute set A denotes the set of attributes already used to decompose the tree and D is the set of detection rules that are matched at that node. The initial root node contains the whole set of detection rules, an empty set of attributes, and an empty set of matched rules. Then, we iteratively decompose each node according to the set of possible attributes using the appropriate inference rules. Leaves are nodes that cannot be transformed anymore. They can be used to report attacks thanks to the detection rules contained in their last field. A promotion node is a node at which can be further processed by the sink as a root node of subtree.

Before we present our construction scheme, we define some notations and auxiliary functions employed in the decision tree construction scheme.

Definition 1. Let $T = \{t_1, t_2, \ldots, t_k\}$ be a set of criterion variable and d be a rule which is $\{(v_1 = t_1) \wedge (v_2 = t_2) \wedge \cdots \wedge (v_k = t_k)\}$. k is the dimension of T. We define the function $\text{Drawn}(d) = \{v_1, v_2, \ldots, v_k\}$. The function can be extended to a set of rules L by

$$\text{Drawn}(D) = \bigcup_{d \in D} \text{Drawn}(d). \tag{1}$$

Definition 2. We define the function $\text{Obtain}(N(F, D, M)) = \{\text{Subtree} \mid N_1(F, D_1, M_1) \cup N_2(F, D_2, M_2) \cdots \cup N_m(F, D_m, M_m)\}$. $N_1, N_2, \ldots, N_m$ are the member nodes of the subtree. This function sends $N(F, D, M)$ to a sink. Then $N(F, D, M)$ can be further processed by the sink and a subtree will be returned to the sensor. The root node of the subtree is $N(F, D, M)$, so that the subtree can be integrated with the current tree.

We use function Drawn to extract the parameters of the local rules, which are low level rules. Also, we use the function Obtain to get a subtree from the sink. In other words, if the sensor cannot deal with some situations, the sink can help to decompose the current node N into a subtree base on high level rules. We assume that the root node of the tree has been selected. For each nonempty branch of the current node, we use the following scheme to construct a decision tree.

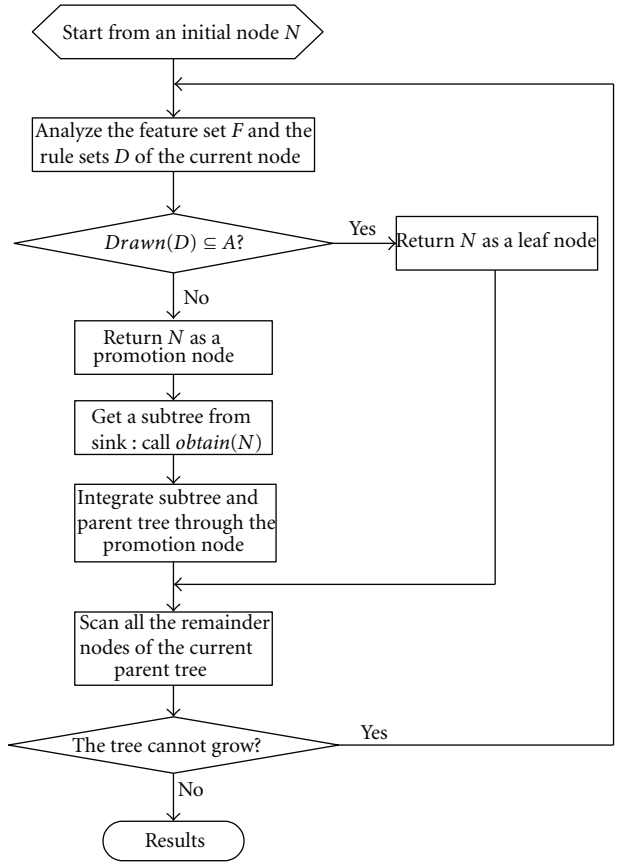


Figure 3: Decision tree learning in sensor.

The scheme of tree construction is shown in Figure 3. The process begins from an initial node N. The current node will become a leaf node if all the attributes have been considered. Otherwise, function Obtain will be used. When $\text{Obtain}(N)$ function is performed, the connection point of the subtree and the parent tree is the current node N. Note that the parent tree is the decision tree in the sensor, and the subtree is generated in the sink. The rule set in the sensor is a subset of the rule set in the sink. All leaf nodes cannot be processed further. The construction process is stopped when all reduced nodes are leaf nodes.

3.3. Unsupervised Learning Based Intrusion Detection in Sink and Base Station. Traditionally, a decision tree is viewed as a supervised learning method, because the splitting is guided by an impurity measure, which depends on the class labels of the data. On the other hand, clustering is an important exploratory data analysis task. It aims to organize objects (data records) into similarity groups or clusters. Clustering is often called unsupervised learning as no classes denoting an a priori partition of the objects are known. This is in contrast with supervised learning (e.g., classification), for which the data records are already labeled with known classes.

As mentioned before, we have designed the supervised learning in sensor based on decision tree. In order to correspond with the learning scheme in sensors, we base our

unsupervised learning on decision tree. As a matter of fact, there have been several studies focused on decision tree based clustering. The scheme in [44] needs to introduce additional data points into the existing points. The operation is complex for WSNs. In [45], an unsupervised decision tree is proposed for information retrieval, which cannot be applied in WSNs directly. In this section, we present the decision tree based clustering for WSNs.

Clustering is required to divide initial sets of objects on many groups (clusters) so that objects inside each group would be the much alike in some sense, while the objects of different groups will be as more as possible "different." It is required to find out such clusters of objects in space of characteristics, which will in the best way satisfy to a criterion of a grouping quality. It is supposed that the characteristics, describing objects, may be both quantitative and qualitative. Various methods of the cluster analysis differ in the ways of understanding of similarity, criterion of quality, and ways of finding groups.

At first, we define a criterion of quality of the grouping. Let characteristic of a request from a user or an attacker be a data sample. All the samples consist of the sample space. The decision tree with L leaves splits space of characteristics into L nonoverlapping subareas $S^1, S^2, \ldots, S^L$. This splitting space corresponds to the splitting of the set of observation Samples into L subsets $\text{Sample}^1, \text{Sample}^2, \ldots, \text{Sample}^L$. Thus, the number of leaves in a tree coincides with the number of groups of objects. We will consider a group of objects Sample^i.

The description of this subset will be the following conjunction of statements:

$$U\left(\text{Sample}^i, V^i\right) = \left(X_1 \in V_1^i\right) \wedge \left(X_2 \in V_2^i\right) \wedge \cdots \wedge \left(X_n \in V_n^i\right), \tag{2}$$

where V_j^i is interval which is calculated as follows:

$$V_j^i = \left[\min_{\text{Sample}^i}\left\{x_j\right\}, \max_{\text{Sample}^i}\left\{x_j\right\}\right], \quad \text{or } V_j^i = \left\{x_j \mid x_j \in \text{Sample}^i\right\}, \tag{3}$$

where the previous equation is for quantitative characteristic, and the second one is for qualitative characteristic.

A characteristic subspace R^i, corresponding to the group's description, we call a taxon (plural taxa). It is important to note, although in a decision tree the part of characteristics can be absent, in the description of each group all available characteristics must participate.

Relative capacity (volume) of taxon can be calculated by

$$\lambda^i = \prod_{j=1}^{n} \frac{\left|V_j^i\right|}{\left|D_j\right|}, \tag{4}$$

where $|V_j^i|$ designates the length of an interval (in case of the quantitative characteristic) or capacity (number of values) of appropriate subset V_j^i (in case of the qualitative characteristic); $|D_j|$ is the length of an interval between the minimal and maximal values of characteristic X_j for all objects from initial sample (for the quantitative characteristic) or the general number of values of this characteristic (for the qualitative characteristic).

When the number of clusters is known, the criterion of quality of a grouping is the amount of the relative volume of taxa:

$$g = \sum_{i=1}^{L} \lambda^i. \tag{5}$$

The grouping with minimal value of the criterion is called optimum grouping.

If the number of clusters is not given beforehand, it is possible to understand the next value as the criterion of quality

$$P = g + aL, \tag{6}$$

where $a > 0$ is a given parameter.

When minimizing this criterion, we receive on the one hand taxa of the minimal size and on the other hand aspire to reduce the number of taxa. Notice that in a case when all characteristics are quantitative, minimization of criterion means minimization of the total volume of multivariate parallelepipeds, which contain the groups.

For the construction of a decision tree, the method of consecutive branching described in Section 3.2 can be used. On each step of this method, a group of the objects corresponding to the leaf of the tree is divided into two new subgroups.

Division occurs with a glance on criterion of quality of a grouping, that is, the total volume of received taxa should be minimal. The node will be divided if the volume of the appropriate taxon is more than a given value. The division proceeds until there is at least one node for splitting or the current number of groups is less than the given number.

Note that learning mechanism in sink not only constructs decision tree for itself but also decomposes the promotion node from sensor to construct a subtree for sensor.

3.4. Reinforcement Learning Based Intrusion Prediction in Base Station. The process of monitoring user behavior and making predictions on network is a nonlinear problem [46]. Especially, comparing with traditional communication networks, WSNs have more dynamic and nonlinear facts. Hence, the linear method cannot work well for intrusion prediction in WSNs. Nonlinear prediction by reinforcement learning [47] and related algorithm can be used to solve intrusion prediction.

Different from supervised learning and unsupervised learning, reinforcement learning is a kind of goal-directed learning which is of great use for a learner (agent) adapting unknown environments [48]. When the environment belongs to Markov decision process (MDP), or partially observable Markov decision process (POMDP), a learner acts some trial-and-error searches according to certain policies and receives reward or punishment. The scheme in [47] uses Stochastic Gradient Ascent (SGA) algorithm [49] as

the reinforcement learning algorithm. However, the main shortcoming of the scheme is that off-policy sampling, as well as nonlinear function approximation, can cause the algorithms to become unstable (i.e., the parameters of the approximator may diverge). Moreover, the unstability will decrease the accuracy of prediction.

By using the convergent temporal-difference learning [50], we develop the nonlinear prediction into a stable scheme. Furthermore, we use the modified scheme as the intrusion prediction scheme base station.

As mentioned before, if an intrusion presents to the sensor, sink, or base station, the attribute parameter of the attack must be reported to the predictor in the base station. Here, the attribute parameter of the attack is denoted as a vector Attack $= [a_1, a_2, \ldots, a_n]$, which includes the concerned node ID, node address, attack type, attack time, and so forth. The selection of the attributes depends on the application scenarios. The architecture of the intrusion prediction system is illustrated in Figure 4.

The neural network in the prediction system is composed by 4 layers: input layer, hidden layer, stochastic layer, and output layer.

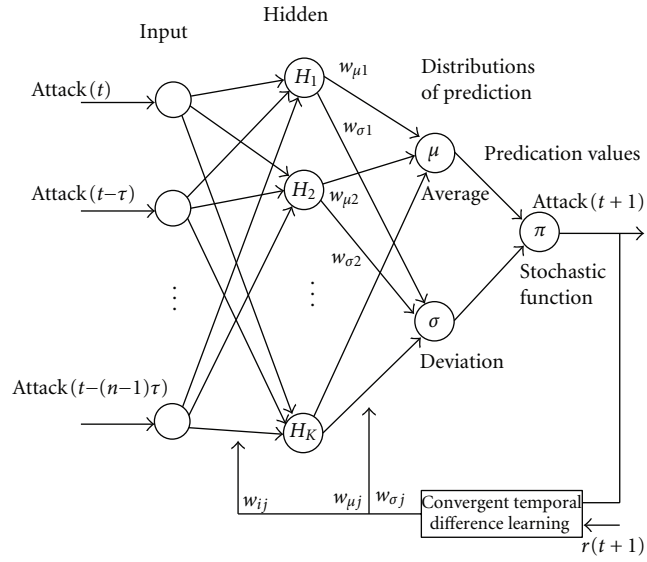


Figure 4: Architecture of intrusion prediction.

Input Layer. The inputs of prediction system on time t can be constructed as an n dimensions vector space $X(t)$, which includes n observed points with same intervals on time series Attack(t).

$$\begin{aligned} X(t) &= (x_1(t), x_2(t), \ldots, x_n(t)) \\ &= (\text{Attack}(t), \text{Attack}(t-\tau), \ldots, \text{Attack}(t-(n-1)\tau)), \end{aligned} \tag{7}$$

where τ is time delay (interval of sampling) and n is the embedding dimension.

Hidden Layer. Multiple nodes accept input with weights w_{ij}, and their output is given by

$$H_j(t) = \frac{1}{1 + e^{-\beta_H \sum a_i(t) w_{ij}}}, \tag{8}$$

where β_H is a constant.

Stochastic Layer. To each hidden node $H_j(t)$ in hidden layer, parameters of distribution function are connected in weight $w_{\mu j}$ and weight $w_{\sigma j}$ when we consider the output is according to Gaussian distribution. Nodes in stochastic layer give their output μ, σ as

$$\begin{aligned} \mu\left(H_j(t), w_{j\mu}\right) &= \frac{1}{1 + e^{-\beta_\mu \sum H_j(t) w_{j\mu}}} \\ \sigma\left(H_j(t), w_{j\sigma}\right) &= \frac{1}{1 + e^{-\beta_\sigma \sum H_j(t) w_{j\sigma}}}, \end{aligned} \tag{9}$$

where β_μ, β_σ is constant, respectively.

Output Layer. The node in output layer means a stochastic policy in reinforcement learning. Here we use a 1-dimension Gaussian function

$$\pi(\text{Attack}(t+1), W, X(t)) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(\text{Attack}(t+1)-\mu)^2/2\sigma^2}, \tag{10}$$

where Attack$(t+1)$ is the value of one-step ahead prediction, produced by regular random numbers. W means weights w_{ij}, $w_{\mu j}$, and $w_{\sigma j}$. This function causes learner's action, so it is called stochastic policy in reinforcement learning.

In order to update weights, we consider a prototypical case of temporal-difference learning, that of learning a linear approximation to the state-value function for a given policy and Markov decision process (MDP) from sample transitions. We take both the MDP and the policy to be stationary, so their combination determines the stochastic dynamics of a Markov chain. The state of the chain at each time t is a random variable, denoted as $s_t = \{1, 2, \ldots, N\}$. On each transition from s_t to s_{t+1}, there is also a reward r_{t+1}, whose distribution depends on both states. We seek to learn the parameter $\theta \in \mathbf{R}^n$ of an approximate value function $V_\theta : S \to \mathbf{R}^n$ such that

$$V_\theta(s) = \theta^T \phi_s \approx V(s) = E\left\{ \sum_{t=1}^{\infty} \gamma^t r_{t+1} \mid s_0 = s \right\}, \tag{11}$$

where $\theta_s \in \mathbf{R}^n$ is feature vector characterizing state s, and $\gamma \in [0, 1)$ is a constant called the discount rate.

Temporal difference error is defined as follows:

$$\delta_k = r_k + \gamma \phi_k^T \phi' - \theta_k^T \phi_k. \tag{12}$$

Following the method in [50], the weight W can calculated by

$$W = E\left[\phi\phi^T\right]^{-1} E[\delta\phi]. \tag{13}$$

Note that that δ depends on θ, hence w depends on θ.

Therefore, w can be updated as follows:

$$w_{k+1} = w_k + \psi_k\left(\delta_k - \phi_k^T w_k\right)\phi_k, \tag{14}$$

where ψ_k factors are step-size parameters, possibly decreasing over time.

3.5. Implementation System Based on Agent Technology

3.5.1. Design of Agent System. We use multiagent to realize the function of intrusion detection and prediction in WSNs. There are four kinds of agents designed in WSNs, which are detection agent (DA), communication agent (CA), database agent (BA), and prediction agent (PA). All the four agents are designed in base station. However, only DA, CA, and BA are designed in sensor and sink. Figure 5 shows the structure of the agent system of a node in wireless sensor network. In Figure 5, prediction agent (PA) is coloured grey, which means that PA does not exist in every node of the WSN but only base station node.

Detection Agent (DA)

(1) The *Detection Learning Module (DLM)* performs the learning algorithm described in Section 3. The module acts as a classifier to perform intrusion detection. It implements the proposed supervised decision tree learning algorithm for sensor. For sink and base station, this module runs the proposed decision tree based cluster algorithm. The rules for making decision are called from detection rule module. The results of learning can be sent to detection rule module for updating rules.

(2) The *Detection Rule Module (DRM)* contains the rule sets for intrusion detection. The rules are the choice of application design. The rules can be updated by the learning algorithm in the DLM.

Communication Agent (CA). This agent provides an interface for the node communicating with other nodes. Also, it preprocesses the raw data into the format required by the data classification techniques. On one hand, this module acts as an interface for the node interoperating with other nodes in WSNs. For sensor, this module sends the packet of *promotion* and reports as well as receives the response from sink. In sink, this module reports every attack, which occurs to the sensor and sink, to the base station. When this module is performed in a base station, it receives the report packets from the sinks. On the other hand, communication agent performs an interface to receive request and send responses for the user who accesses the node. It transfers the parameters of the request to DA and PA for further processing.

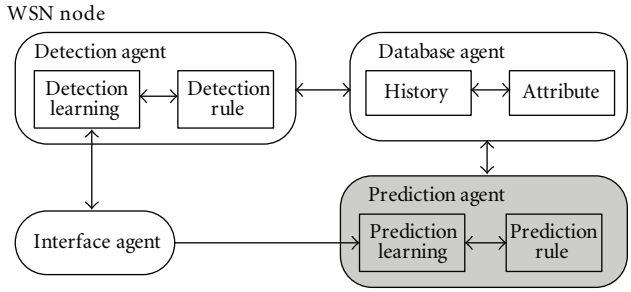


Figure 5: Node model of agent system.

Database Agent (BA)

(1) The *History Module (HM)* provides two distinct functionalities: a convenient mechanism to log events and actions that have occurred and an efficient mechanism to query these logged events. This module provides history data for detection learning and prediction learning.

(2) The *Attribute Module (AM)* provides an interface for the detection agent (DA) and prediction agent (PA) to query and update attributes of the data and users.

Prediction Agent (DA)

(1) *The Prediction Learning Module (PLM)* is designed only in base station. It performs the reinforcement learning algorithm described in Section 3. The module acts as a classifier to perform intrusion prediction. The rules for making decision are called from prediction rule module. The results of learning can be sent to prediction rule module for updating rules.

(2) *The Prediction Rule Module (PRM)* contains the rule sets for intrusion prediction. The rules can be updated adaptively by the learning algorithm in the PLM.

3.5.2. Sequence Model of Interaction. In BLID, there are several cases of interoperation among the learning modules in different nodes in WSNs, which is similar with the cerebellum, the basal ganglia, and the cerebral cortex. On one hand, when unknown request presents to the sensor, the sensor performs a *promotion* action for further detection by helping of sink. After the sink finished the further detection by using the clustering algorithm, it sends the responses to the sensor. The response includes the subtree (see Section 3.2) for updating the detection rules in the sensor. On the other hand, for both sink and sensor, they must report the features of the request to the base station if they find the request comes from an attacker. Note that sensors report the attack to base station via the sink, because sensor cannot communicate with the base station directly. Because report is an operation which is easy to understand, we just illustrate the sequence of *promotion* operation in Figure 6.

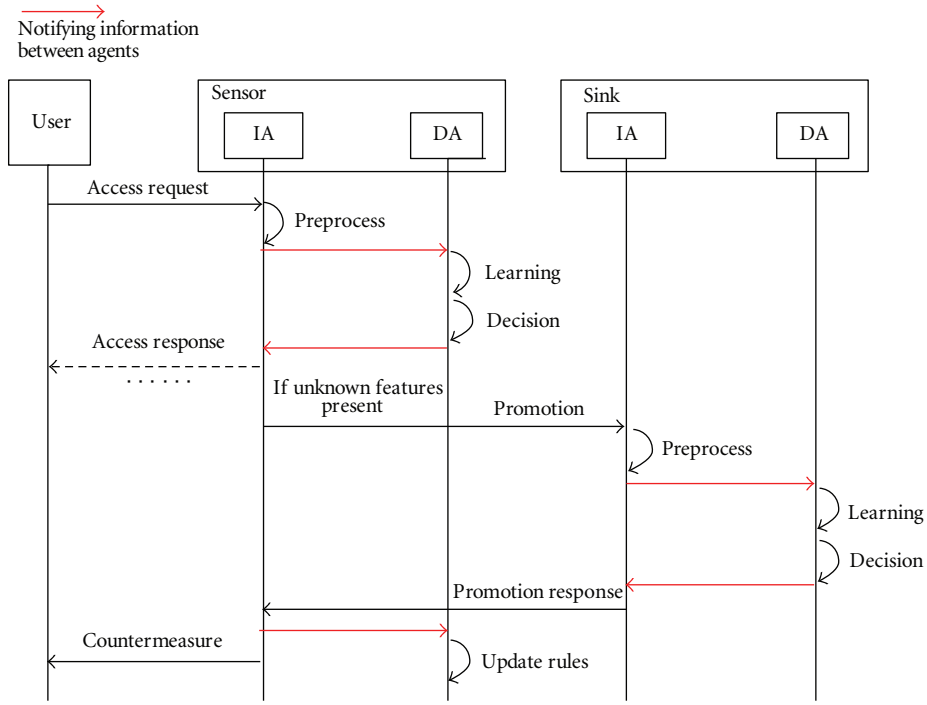


Figure 6: Sequence model of *promotion* operation.

4. Getting Data Set via Wireless Attack Experiment

In this section, we report the attack experimentation, through which we can get the data set for training and test. Because many existing WSNs are deployed by IEEE 802.11 and mote device technologies [31], we use IEEE 802.11 based wireless link for our experiment. Moreover, for access control, a role-based access control (RBAC) policy is used.

Feature selection is an important issue for intrusion prediction. In order to enhance the prediction accuracy for the attack from different layers, we consider both the application layer features and MAC layer features to construct the data set. We combine the features of access control and 802.11 wireless traffics [51] to construct the feature data set. On one hand, we select the important features of access control based on the feature selection method for access control in [52]. On the other hand, according to IEEE 802.11 standard [51], the fields of the MAC header can be extracted. We used the information gain ratio (IGR) as a measure to determine the relevance of each feature [53]. We can order the features according to the score assigned by the IGR measure. The IGR measure is based on the data set of frames collected from our testing network. The features of access control and 802.11 traffics which we used for experiment are shown in Tables 1 and 2, respectively. The number of the selection features depends on the requirements of security and the recourses of the system. As a case study for resource-constrained WSNs, we select 5 access control features and 7 traffic features of 802.11 for test.

We did the attack experiment in an 802.11 network. We take ARP replay attack, forgery attack, ongoing dictionary attack [32], and chopchop attack [33], which are the common attacks in 802.11 networks, as the examples for evaluation. The tool we use to generate attacks is Backtrack, which is available from the website [34].

In our experiment, the network was composed of three wireless stations. We use one machine as a server node (access point). Then, we use another machine to generate normal traffic firstly and later attacks. The last machine was used

Table 1: Features of access control.

Order	Features	Description
1	LoginResult	Access decision results before access
2	NumbWr	Number of write operation on access control files
3	NumbCrea	Number of create operation on rule file
4	NumbAccess	Number of access
5	NumbDe	Number of delete operation on access control files

Table 2: Features of traffic.

Order	Features	Description
1	WepResult	The result of WEP ICV check
2	Duration	The time the medium is expected to be busy
3	More_Frag	Whether a frame is nonfinal fragment or not
4	Desti_Addr	The MAC address of the receiving node
5	Fram_Type	The type of the frame
6	IfRetransmit	If the frame is a retransmitted frame
7	Sour_Addr	The MAC address of sending node

Table 3: Data set.

Traffic type	Training set	Test set
ARP replay attack	200	200
Forgery attack	200	200
Ongoing dictionary attack	200	200
Chopchop attack	0	200
Normal	1200	1200

to collect and record both normal and intrusion traffic. The number of related records in the data set is shown in Table 3. There is no training set for chopchop attack, because we use this attack as unknown attack for test. The other three kinds of attacks can be regarded as usual attacks.

5. Evaluation and Comparisons

This section evaluates BLID in terms of overhead and accuracy.

5.1. Overhead and Complexity Evaluation

5.1.1. Time Overhead and Memory Consumption in Sensor. Usually the resources of sensor are limited, but the resources of sink and base station are powerful. Hence, the evaluation of sensor is crucial and typical. We focus on the time overhead and memory consumption caused by our scheme on sensor. We have implemented BLID for TinyOS and tested it using TOSSIM [54]. The mote that TOSSIM simulates is MicaZ.

There are two phases of the learning, training phase and test set. Before the sensors being deployed, the training process can be performed on some other well-resourced devices, such as laptop, because the recourses of sensors are limited. Hence, the initial detection rules can be constructed on well-resourced devices and then loaded into sensors. In this paper, the initial detection rules training is based on the training data set in Section 4. Based on the above reasons, we just focus on the test phase. The overhead caused by BLID and related schemes during detection is reported in Figure 7, which is the time needed by a sensor from receiving a request to making a local detection decision.

In Figure 7, the vertical coordinates denote the overhead caused by intrusion detection system. Four groups of columns denote four cases which are corresponding with four kinds of attacks described in Section 4. As shown in Figure 7, the time overhead caused by BLID is lower than that of the schemes in [12, 25]. The results show that detecting unknown attacks usually needs more time than detecting known attacks.

Loading the rules intrusion detection requires memory. The memory consumption of our scheme is an important measure of its feasibility and usefulness on memory-constrained sensor nodes. The memory consumption is shown in Table 4. Because MicaZ has 128 KB of instruction memory and 512 KB of flash memory, the experiment results mean that BLID leaves enough space in the mote's memory for user applications. In addition, we use PC act as the sink and base station nodes. The memory consumption of high level detection and prediction is shown in Table 5.

Table 4: Memory consumption in sensor.

Agent	Size (bytes)
Detection agent	10274
Database agent	21857
Communication agent	3216
Total	35347

Table 5: Memory consumption in sink and base station.

Agent	Size (bytes)
Prediction agent in BS	535341
Database agent in BS	732219
Detection agent in sink	152796
Database agent in sink	225678
Communication agent	3216

5.1.2. Energy Consumption of Sensor. Energy cost is one of the most critical problems in resource-constrained sensors. In this subsection, we estimate the energy consumption of sensor using PowerTOSSIM [55], which is an energy modeling extension of TOSSIM. PowerTOSSIM is often used to evaluate the energy consumption of WSNs [56–58]. The energy consumption is measured for five components: CPU, RADIO, LED, SENSOR, and EEPROM. According to the attack experiment in Section 4, ARP reply attack, forgery attack, and ongoing dictionary attack act as the known attacks, while chopchop attack acts as unknown attack. In addition, based on the time overhead in Figure 7, the time overhead of ongoing attack has higher time complexity than ARP replay attack and forgery attack. Therefore, here we take ongoing attack as the example of known attack for energy consumption evaluation. Chopchop attack is still used as unknown attack for the energy consumption evaluation. Then, we fix the time of execution equal to 1200 simulated seconds, which is because the motes in PowerTOSSIM take boot time of 10 seconds. Here we consider three cases, which are continuous known attack, continuous unknown attack, and continuous normal access. In the simulation, the radio is set as sleep mode if there is no *promotion* or *report* message being transferred. The energy costs of different cases are shown in Figure 8.

In our proposed system, storing feature and rule data performed by EEPROM component and classification analysis performed by CPU component course the corresponding additive energy consumption in EEPROM and CPU, while radio transmission is not always necessary which depends on the *promotion* and *report* operations. As shown in Figure 8, the CPU energy costs for unknown and known attacks are higher than that of normal access and known attack. This is because that the attacks course more computations of the features analysis and must perform report operation to the base station. RADIO energy consumption of unknown

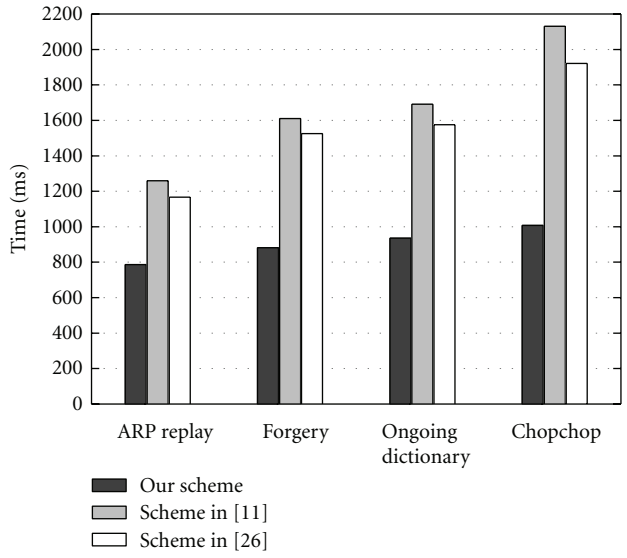


Figure 7: Time overhead in sensor.

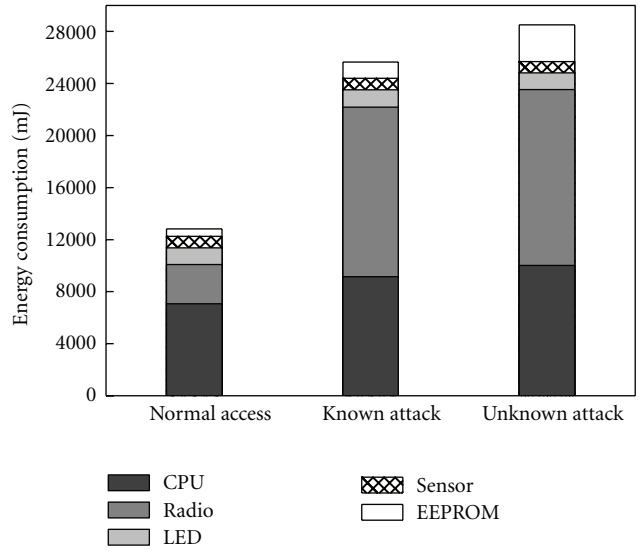


Figure 8: Energy consumption of sensor.

attack detection is higher than that of known attack, because of the additive *promotion* messages. Energy costs of the all the three cases are lower than that of related application schemes of WSNs, such as some ECG monitoring schemes in [57] and some data-stream protocol in [58]. Therefore, the energy consumption of our scheme is acceptable for resource-constrained WSNs.

5.1.3. Communication Overhead. BLID can cause communication overhead into WSNs. In a WSN, the number of sensor is usually much more than that of sink and base station, and some sensors usually are deployed far from base station and sink. In other words, the communication overhead is mainly caused by sensors. Hence, we evaluate the case that the attacks occur to sensors. Figure 9 depicts the communication cost of BLID measured in overhead packets in WSNs.

As shown in Figure 9, the communication overhead in case of unknown attack is higher than that in case of known attack, because the sink needs to return a subtree to the sensor in case of unknown attacks. The communication overhead also depends on the number of hop from the intruded sensor to the sink. For small scale WSNs, such as the number of hop is 3, the communication cost is only 4 for known attacks and 7 for known attacks, respectively. Moreover, for larger scale WSNs, such as the case of 7 hop, the overhead still remains low (8 packets for known attacks and 15 packets for unknown attacks). The communication cost of BLID is lower than that of cooperative intrusion detection scheme in [59]. For the scheme in [59], the communication overhead is 12 for 4 sensors cooperate for detection, and the overhead increase to 19 when 8 sensors cooperate.

5.2. Prediction Rate. The evaluation of the accuracy of prediction was obtained using Matlab and NeuroSolutions [60]. The detection accuracy of BLID depends on the learning algorithm in sink and base station, because "*promotion*"

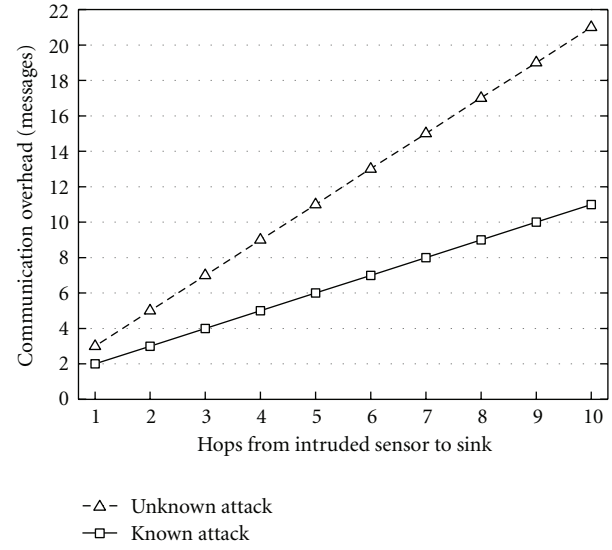


Figure 9: Communication overhead.

operation exists in the low level detection in sensor. The prediction accuracy depends on the reinforcement learning scheme performed in base station.

We use two metrics to evaluate the intrusion prediction performance, namely, prediction rate q and false alarm rate η. The prediction rate is formally defined by

$$q = \frac{d}{n}, \tag{15}$$

where d is the number of prediction attacks, and n is the total number of actual attacks.

We evaluate the prediction rate in this section. Because the real sample cannot be gotten in WSNs for intrusion prediction, DARPA Intrusion Detection Evaluation Data [61] is used as the training and test data set to verify the prediction rate of related schemes. The training data consist of five

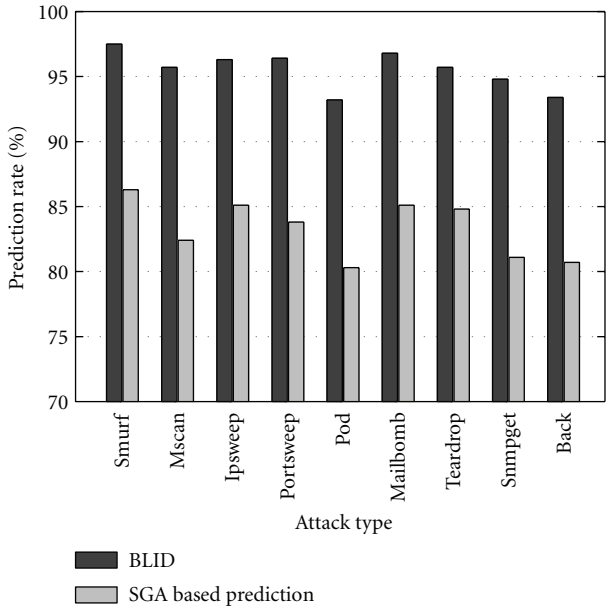


Figure 10: Prediction rate.

weeks of network-based attacks in the midst of normal background data. Attacks are labeled in training data. The test data consist of two weeks network-based attacks and normal background data. We also use the features defined in DARPA 1998 as the feature parameters of our prediction scheme. The prediction rate is shown in Figure 10. In Figure 10, the vertical coordinates denote the prediction rate. Nine groups of columns denote nine cases which are corresponding with nine kinds of attacks in DARPA Intrusion Detection Evaluation Data. As shown in Figure 10, the prediction rate of BLID is on average 12 percentages higher than that of SGA based scheme in [48].

6. Conclusion

In this paper, we analyzed the important issues of accurate intrusion detection and prediction in WSNs. To address the problems, we proposed a brain-like hierarchical learning based intelligent intrusion prediction scheme, called BLID, in which the sensor, sink, and base station perform different kinds of learning algorithms and interoperate optimally with each other. Referring to brain-like hierarchical learning model, we designed a relatively simple decision tree learning algorithm in the sensor for low level intrusion detection, which is corresponding with the supervised learning of cerebellum. Then, we proposed a decision tree based clustering mechanism in sink and base station for intrusion detection, which has a correspondence with unsupervised learning of cerebral cortex. Furthermore, we developed a stable reinforcement learning model in base station for high level intrusion prediction, which is referenced to reinforcement learning of the basal ganglia. Through combing and connecting different learning modules in the sensor, the sink, and the base station as a global network, the function of distributed detection and centralized prediction can be realized. The implementation system of BLID is designed based on agent technology. Our experiment shows that the proposed scheme has several advantages in terms of efficiency of implementation and high prediction rate. Although we assume in this paper that WSNs is deployed through the three-layer architecture, BLID can also be applicable for the WSNs deployed in two-layer architecture, which only includes base station and sensor. This is because both unsupervised learning and reinforcement learning modules are designed in base station, then sensor can interoperate directly with base station for *promotion* operation. An interesting future work of BLID may be on the efficiently distributed intrusion prediction of WSNs.

Acknowledgments

The authors thank Dr. Rana Ashour for the invitation. This work was supported by Japan Society for the Promotion of Science (JSPS) under Grant-in-Aid for Scientific Research (C) (no. 20560373) and China Scholarship Council (CSC) under Grant no. 2008638003.

References

[1] E. Fontana, J. F. Martins-Filho, S. C. Oliveira et al., "Sensor network for monitoring the state of pollution of high-voltage insulators via satellite," *IEEE Transactions on Power Delivery*, vol. 27, no. 2, pp. 953–962, 2012.

[2] J. Valverde, V. Rosello, G. Mujica, J. Portilla, A. Uriarte, and T. Riesgo, "Wireless sensor network for environmental monitoring: application in a coffee factory," *International Journal of Distributed Sensor Networks*, vol. 2012, pp. 1–18, 2012.

[3] J. Wu and S. Shimamoto, "Usage control based security access scheme for wireless sensor networks," in *Proceedings of the IEEE International Conference on Communications (ICC '10)*, Cape Town, South Africa, May 2010.

[4] B. Sun, L. Osborne, Y. Xiao, and S. Guizani, "Intrusion detection techniques in mobile ad hoc and wireless sensor networks," *IEEE Wireless Communications*, vol. 14, no. 5, pp. 56–63, 2007.

[5] J. Wu and S. Shimamoto, "Integrated UCON-based access control and adaptive intrusion detection for wireless sensor networks," in *Proceedings of the 53rd IEEE Global Communications Conference (GLOBECOM '10)*, Miami, Fla, USA, December 2010.

[6] S. Budhaditya, D. S. Pham, M. Lazarescu, and S. Venkatesh, "Effective anomaly detection in sensor networks data streams," in *Proceedings of the 9th IEEE International Conference on Data Mining (ICDM '09)*, Miami, Fla, USA, December 2009.

[7] F. Bao, I. R. Chen, M. J. Chang, and J. H. Cho, "Hierarchical trust man-agement for wireless sensor networks and its applications to trust-based routing and intrusion detection," *IEEE Transactions on Network and Service Management*, vol. 9, no. 2, pp. 169–182, 2012.

[8] L. M. Wang, T. Jiang, and X. Y. Zhu, "Updatable key management scheme with intrusion tolerance for unattended wireless sensor network," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM '11)*, December 2011.

[9] R. Roman, J. Zhou, and J. Lopez, "Applying intrusion detection systems to wireless sensor networks," in *Proceedings of*

the 3rd IEEE Consumer Communications and Networking Conference (CCNC '06), Las Vegas, Nev, USA, January 2006.

[10] E. C. H. Ngai, J. Liu, and M. R. Lyu, "On the intruder detection for sinkhole attack in wireless sensor networks," in *Proceedings of the IEEE International Conference on Communications (ICC '06)*, Istanbul, Turkey, July 2006.

[11] C. E. Loo, M. Y. Ng, C. Leckie, and M. Palaniswami, "Intrusion detection for routing attacks in sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2, no. 4, pp. 313–332, 2006.

[12] K. Q. . Yan, S. C. Wang, and C. W. Liu, "A hybrid intrusion detection system of cluster-based wireless sensor networks," in *Proceedings of the International MultiConference of Engineers and Computer Scientists (IMECS '09)*, Hong Kong, China, March 2009.

[13] I. C. Paschalidis and Y. Chen, "Anomaly detection in sensor networks based on large deviations of markov chain models," in *Proceedings of the 47th IEEE Conference on Decision and Control (CDC '08)*, Cancun, Mexico, December 2008.

[14] S. Rajasegarar, C. Leckie, M. Palaniswami, and J. C. Bezdek, "Quarter sphere based distributed anomaly detection in wireless sensor networks," in *Proceedings of the IEEE International Conference on Communications (ICC '07)*, Scotland, UK, June 2007.

[15] H. Alipour, M. Gholami, and A. Vahdani, "A base-station oriented anomaly detection for wireless sensor networks," in *Proceedings of the 4th IEEE/IFIP International Conference in Central Asia on Internet (ICI '08)*, pp. 1–5, Tashkent, Uzbekistan, September 2008.

[16] F. Jemili, M. Zaghdoud, and M. B. Ahmed, "Hybrid intrusion detection and prediction multiAgent system, HIDPAS," *International Journal of Computer Science and Information Security*, vol. 5, no. 1, pp. 62–71, 2009.

[17] Z. Zhang, Z. Peng, and Z. Zhou, "The study of intrusion prediction based on HsMM," in *Proceedings of the IEEE Asia-Pacific Services Computing Conference (IEEE APSCC '08)*, Yilan, Taiwan, December 2008.

[18] F. Ozgul, Z. Erdem, and C. Bowerman, "Prediction of past unsolved terrorist attacks," in *Proceedings of the IEEE International Conference on Intelligence and Security Informatics (ISI '09)*, Dallas, Tex, USA, June 2009.

[19] M. R. Ahmadi, "An intrusion prediction technique based on co-evolutionary immune system for network security (CoCo-IDP)," *International Journal of Network Security*, vol. 9, no. 3, pp. 290–300, 2009.

[20] N. Ye, Q. Chen, and C. M. Borror, "EWMA forecast of normal system activity for computer intrusion detection," *IEEE Transactions on Reliability*, vol. 53, no. 4, pp. 557–566, 2004.

[21] K. Haslum, A. Abraham, and S. Knapskog, "DIPS: a framework for distributed intrusion prediction and prevention using hidden Markov models and online fuzzy risk assessment," in *Proceedings of the 3rd Internationl Symposium on Information Assurance and Security (IAS '07)*, Manchester, UK, August 2007.

[22] S. Li and Y. Luo, "Discernibility analysis and accuracy improvement of machine learning algorithms for network intrusion detection," in *Proceedings of the IEEE International Conference on Communications (ICC '09)*, Dresden, Germany, May 2009.

[23] C. F. Tsai, Y. F. Hsu, C. Y. Lin, and W. Y. Lin, "Intrusion detection by machine learning: a review," *Expert Systems with Applications*, vol. 36, no. 10, pp. 11994–12000, 2009.

[24] T. Pietraszek and A. Tanner, "Data mining and machine learning—towards reducing false positives in intrusion detection," *Information Security Technical Report*, vol. 10, no. 3, pp. 169–183, 2005.

[25] Z. Yu and J. J. P. Tsai, "A framework of machine learning based intrusion detection for wireless sensor networks," in *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC '08)*, Taichung, Taiwan, June 2008.

[26] C. Chen, J. Ma, and K. Yu, "Designing energy-efficient wireless sensor networks with mobile sinks," in *Proceedings of the ACM Sensys'06 Workshop WSW*, 2006.

[27] J. Zhang and V. Varadharajan, "A new security scheme for wireless sensor networks," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '08)*, New Orleans, La, USA, December 2008.

[28] I. Bisio and M. Marchese, "Efficient satellite-based sensor networks for information retrieval," *IEEE Systems Journal*, vol. 2, no. 4, pp. 464–475, 2008.

[29] B. Thuraisingham, "Secure sensor information management and mining," *IEEE Signal Processing Magazine*, vol. 21, no. 3, pp. 14–19, 2004.

[30] P. Goyal, S. Batra, and A. Singh, "A literature review of security attack in mobile Ad-hoc networks," *International Journal of Computer Applications*, vol. 9, no. 12, pp. 11–15, 2010.

[31] G. Anastasi, E. Borgia, M. Conti, E. Gregori, and A. Passarella, "Understanding the real behavior of Mote and 802.11 ad hoc networks: an experimental approach," *Pervasive and Mobile Computing*, vol. 1, no. 2, pp. 237–256, 2005.

[32] http://www.backtrack-linux.org/.

[33] J. Lei, X. Fu, D. Hogrefe, and J. Tan, "Comparative studies on authentication and key exchange methods for 802.11 wireless LAN," *Computers and Security*, vol. 26, no. 5, pp. 401–409, 2007.

[34] A. Bittau, M. Handley, and J. Lackey, "The final nail in WEP's coffin," in *Proceedings of the IEEE Symposium on Security and Privacy*, May 2006.

[35] A. Roy, "On connectionism, rule extraction, and brain-like learning," *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 2, pp. 222–227, 2000.

[36] M. A. Sharbafi, C. Lucas, and R. Daneshvar, "Motion control of omni-directional three-wheel robots by brain-emotional-learning-based intelligent controller," *IEEE Transactions on Systems, Man and Cybernetics Part C*, vol. 40, no. 6, pp. 630–638, 2010.

[37] K. Doya, "What are the computations of the cerebellum, the basal ganglia and the cerebral cortex?" *Neural Networks*, vol. 12, no. 7-8, pp. 961–974, 1999.

[38] J. Hu, T. Sasakawa, K. Hirasawa, and H. Zheng, "A hierarchical learning system incorporating with supervised, unsupervised and reinforcement learning," in *Proceedings of the 4th International Symposium on Neural Networks (ISNN '07)*, Nanjing, China, June 2007.

[39] J. M. Bradshaw, "Introduction to software agents," in *Soft Agents*, J. M. Bradshaw, Ed., AAAI Press/MIT Press, Cambridge, Mass, USA, 1997.

[40] F. Bai, K. S. Munasinghe, and A. Jamalipour, "An ecologically inspired intelligent agent assisted wireless sensor network for data reconstruction," in *Proceedings of the IEEE International Conference on Communications (ICC '10)*, May 2010.

[41] A. Rogers, N. R. Jennings, and D. D. Corkill, "Agent technologies for sensor networks," *IEEE Intelligent Systems*, vol. 24, no. 2, pp. 13–17, 2009.

[42] Z. Deng and W. Zhang, "Localization and dynamic tracking using wireless-networked sensors and multi-agent technology: first steps," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 85, no. 11, pp. 2386–2395, 2002.

[43] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[44] B. Liu, Y. Xia, and P. S. Yu, "Clustering through decision tree construction," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Dallas, Texas, USA, May 2000.

[45] P. Bellot and M. El-Bèze, "Clustering by means of unsupervised decision trees or hierarchical and K-means-like algorithm," in *Proceedings of the RIAO Conference, Collège de France (RIAO '00)*, Paris, France, April 2000.

[46] V. Kodogiannis and A. Lolis, "Forecasting financial time series using neural network and fuzzy system-based techniques," *Neural Computing and Applications*, vol. 11, no. 2, pp. 90–102, 2002.

[47] T. Kuremoto, M. Obayashi, and K. Kobayashi, "Nonlinear prediction by reinforcement learning," in *Proceedings of the International Conference on Intelligent Computing (ICIC '05)*, Hefei, China, August 2005.

[48] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, The MIT Press, 1998.

[49] H. Kimura and S. Kobayashi, "Reinforcement learning for continuous action using stochastic gradient ascent," in *Proceedings of the 5th International Conference on Intelligent Autonomous Systems (IAS '98)*, 1998.

[50] H. R. Maei, C. Szepesvari, S. Bhatnagar, D. Precup, D. Silver, and R. S. Sutton, "Convergent temporal-difference learning with arbitrary smooth function approximation," in *Proceedings of the 23rd Annual Conference on Neural Information Processing Systems (NIPS '09)*, Vancouver, Canada, December 2009.

[51] IEEE 802.11-1999, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Higher Speed Physical Layer Extension in the 2.4 GHz Band.

[52] W. Lee and S. J. Stolfo, "A framework for constructing features and models for intrusion detection systems," *ACM Transactions on Information and System Security*, vol. 3, no. 4, pp. 227–261, 2000.

[53] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[54] P. Levis, N. Lee, M. Welsh, and D. Culler, "TOSSIM: accurate and scalable simulation of entire TinyOS applications," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (ACM SenSys '03)*, pp. 126–137, November 2003.

[55] V. Shnayder, M. Hempstead, B. R. Chen, G. W. Allen, and M. Welsh, "Simulating the power consumption of large-scale sensor network applications," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, November 2004.

[56] E. T. H. Chu, H. J. Lee, T. Y. Huang, and C. T. King, "Sample assignment for ensuring sensing quality and balancing energy in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 9, pp. 1578–1584, 2011.

[57] M. Zeng, I. Y. Chung, J. A. Lee, and J. G. Lee, "An on-node intelligent based energy efficient ECG monitoring system," in *Proceedings of the International Conference on ICT Convergence (ICTC '11)*, September 2011.

[58] N. Erratt and Y. Liang, "Compressed data-stream protocol: an energy-efficient compressed data-stream protocol for wireless sensor networks," *IET Communications*, vol. 5, no. 18, pp. 2673–2683, 2011.

[59] I. Krontiris, Z. Benenson, T. Giannetsos, F. C. Freiling, and T. Dimitriou, "Cooperative intrusion detection in wireless sensor networks," in *Proceedings of the 6th European Conference on Wireless Sensor Networks (EWSN '09)*, Cork, Ireland, February 2009.

[60] NeuroSolutions, Inc., 2010, http://www.neurosolutions.com/.

[61] DARPA, DARPA, Intrusion Detection Evaluation, 1998, http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/data/1998data.html.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 821486, 7 pages
doi:10.1155/2012/821486

Research Article

Self-Healing Key-Distribution Scheme with Collusion Attack Resistance Based on One-Way Key Chains and Secret Sharing in Wireless Sensor Networks

Dong Jiao,[1] Mingchu Li,[1] Yan Yu,[2] and Jinping Ou[3]

[1] *School of Software Technology, Dalian University of Technology, Dalian 116621, China*
[2] *School of Electronic Science and Technology, Dalian University of Technology, Dalian 116024, China*
[3] *School of Civil Engineering, Dalian University of Technology, Dalian 116024, China*

Correspondence should be addressed to Yan Yu, yuyan@dlut.edu.cn

Received 14 June 2012; Accepted 21 August 2012

Academic Editor: Leonardo B. Oliveira

In wireless sensor networks, self-healing key-distribution schemes are used to ensure that, even if the message packets that are broadcast in some sessions get lost, the group nodes can still recover the lost session keys simply by using their personal secret keys and broadcast messages that have been received without requesting additional transmissions from the group manager. These schemes reduce network traffic, decrease the group manager's workload, and lower the risk of node exposure through traffic analysis. However, most existing schemes have many deficiencies, such as high overhead for storage and communication and collusion attacks. In this paper, we have proposed a modified, self-healing, key-distribution scheme based on one-way key chains and secret sharing. Our scheme has the properties of constant storage, lower communication overhead, long lifespan, forward secrecy, backward secrecy, and resistance to collusion attacks.

1. Introduction

Wireless sensor networks (WSNs) are composed of a large number of sensor nodes with limited power, storage, computation, and communication capabilities. WSNs have wide applications in military operations and scientific exploration [1, 2] in which there may be inadequate support by the infrastructure of the network, allowing adversaries to potentially intercept, modify, or partially interrupt communication. In such applications, security is a critical concern. In addition, in some deployment scenarios, sensor nodes must operate under adversarial conditions. Therefore, determining how to distribute group session keys for secure communication to a large dynamic group over an unreliable network is a serious issue. In WSNs, packet loss occurs frequently. Messages that are broadcast by the group manager (base station) might never reach some authorized nodes (sensor nodes). So, it is important to guarantee the reliable transmission of information for updating the group's session keys to the authorized nodes. An easy solution is requesting retransmission, but requesting retransmission increases the overhead associated with communication incurs a high risk of revealing the nodes' physical locations, which is not acceptable in some high-security environments.

A self-healing, key-distribution scheme is proposed to solve the problem described above. The main concept of self-healing, key-distribution schemes is that, even if the message packets that are broadcast in some sessions get lost, the group nodes can still recover the lost session keys simply by using their personal secret keys and broadcast messages that have been received without requesting additional transmissions from the group manager. These schemes reduce network traffic, decrease the group manager's workload, and lower the risk of node exposure through traffic analysis. Figure 1 shows network topology in a key distribution scheme under adversarial conditions.

In 2002, Staddon et al. [3] proposed the first self-healing, key-distribution scheme with revocation using secret sharing [4]. However, Staddon et al.'s schemes incur high overhead for storage and communication. Later, several other schemes

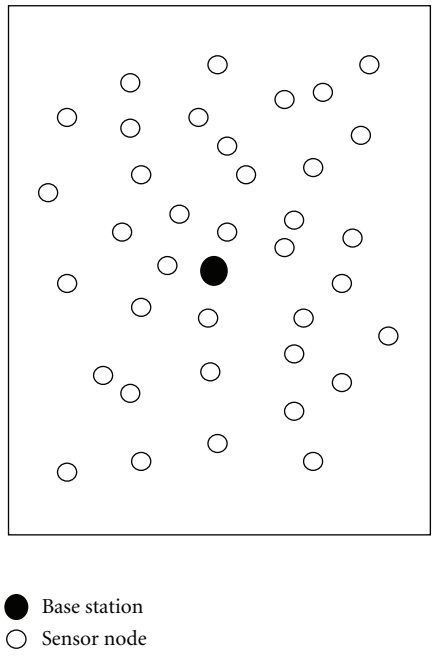


Figure 1: Network topology in a key-distribution scheme.

were proposed [5–9] based on Staddon et al.'s schemes. Liu et al. (2003) generalized the definitions and security notions and proposed a new scheme that significantly decreased the overhead for communication by introducing a novel, personal key-distribution [5]. Blundo et al. [10] showed that the first scheme in [3] is insecure. An adversary could recover the group's session key with just broadcast messages. In [11], Dutta et al. proposed two self-healing, key-distribution schemes with revocation that were secure, but they did not consider collusion attacks. In [12], Dutta et al. proposed a new self-healing key-distribution scheme with a constant storage overhead by using only one secret polynomial. But Xu and He's scheme [13] and Du and He's scheme [14] showed that the scheme in [12] was insecure. Any user can recover the manager's secret polynomial, which should not been known by any user. Xu and He (2009) proposed two schemes in [13], one of which improved the scheme in [12] by using an access polynomial instead of the revocation polynomial with the other, which was based on the scheme in [11], still using an access polynomial. But neither of the two schemes proposed in [13] considered collusion attacks between the revoked user and the newly-joined user. In [14], Du and He proposed a new self-healing, key-distribution scheme with revocation and resistance to collusion attacks. However, Bao and Zhang (2011) showed that the scheme in [14] was vulnerable to collusion attacks [15]. A revoked user and a newly-joined user easily could recover the session keys that they should not know. However, Bao and Zhang (2011) used m secret polynomials for m sessions and an access polynomial in the broadcast phase, which resulted in an excessive communication overhead.

In this paper, we propose a self-healing key-distribution scheme for WSNs based on one-way key chains and secret sharing. In our scheme, only one secret polynomial is used in all sessions, and modified access polynomials are used, which produces a lower communication overhead. Also, our scheme can resist collusion attacks between a newly-joined user and a revoked user.

The rest of the paper is organized as follows. In Section 2, the security model is presented and Bao and Zhang's scheme [15] is reviewed briefly. In Section 3, our modified, self-healing, key-distribution scheme is proposed. Then, we discuss the security and performance of our scheme in Section 4. Our conclusions are presented in Section 5.

2. Preliminaries

In this section, we briefly introduce Bao and Zhang's scheme [15] and the security definitions. The following notations will be used in the rest of the paper.

U is the set of all users (sensor nodes) in wireless sensor networks.

U_i is the user i in U.

u_i is the identity of U_i.

GM is the group manager (base station).

n is the total number of users in U.

m is the total number of sessions.

t is the maximum number of compromised users in all sessions.

p is a large prime modulus, where $2^{799} < p < 2^{800}$.

q is a large prime divisor of $p-1$, where $2^{159} < q < 2^{160}$ and $q^2 \mid (p-1)$.

$\{g_i\}_{i=1}^{m}$ are m generators with order q in GF(p).

$f(x) \in F_q[x]$ is the secret polynomial of degree t generated by GM.

S_i is the personal secret of user U_i.

B_j is the broadcast message generated by GM for session j.

β_j is the self-healing key generated by GM for session j.

K_j is the session key in session j generated by GM.

K_0 is the initial key seed generated by GM.

R_j is the set of all revoked users in and before session j.

G_j is the set of nonrevoked users in session j.

H_1, H_2 are two cryptographically secure, one-way functions, and $H_1 : \{0,1\}^* \to F_p, H_2 : \{0,1\}^* \to F_p$.

$D_k(\cdot)$ is a symmetric decryption function.

$E_k(\cdot)$ is a symmetric encryption function.

2.1. Security Model

Definition 1 (self-healing key-distribution with t-revocation capability [11]). A key-distribution scheme is a self-healing, key-distribution scheme with t-revocation capability if the following conditions are true.

(a) For any nonrevoked user U_i in session j, the group session key K_j is efficiently determined by the broadcast message B_j and the personal secret S_i.

(b) The group session key K_j cannot be determined by what the non-revoked users learn from B_j or their own personal secret alone.

(c) t-revocation capability: for each session j, let R_j denote a set of revoked users in and before session j, where $|R_j| \leq t$, the group manager can generate a broadcast message B_j such that all the revoked users in R_j cannot recover the group session key K_j.

(d) Self-healing property: any U_i who joins in or before session j_1 and is not revoked before session j_2 ($1 \leq j_1 < j_2$) can recover all the keys $K_j (j_1 \leq j \leq j_2)$ by the broadcast messages B_{j_1}, B_{j_2}, and the personal secret S_i.

Definition 2 (t-wise forward secrecy [11]). Let $R_j \subseteq U$ denote a set of all revoked users in and before session j, where $|R_j| \leq t$. A key-distribution scheme guarantees forward secrecy if the members in R_j together cannot get any information about K_j, even with the knowledge of group session keys before session j.

Definition 3 (t-wise backward secrecy [11]). Let $J_j \subseteq U$ denote a set of users who join the group after session j, where $|J_j| \leq t$. A key-distribution scheme guarantees backward secrecy if the members in J_j together cannot get any information about K_j, even with the knowledge of group session keys after session j.

Definition 4 (resistance to the collusion attack [16]). Let $R \subseteq U$ denote a set of all revoked users in and before session j_1 and let $J \subseteq U$ denote a set of users who join the group after session j_2, where $1 \leq j_1 < j_2$ and $|R \cup J| \leq t$. A key-distribution scheme with resistance to collusion attacks means that, even if all users in R and J cooperate, they cannot get any information about keys K_j, for all $j_1 < j < j_2$.

2.2. Review of Bao and Zhang's Scheme. In [15], Bao and Zhang proposed an improved key-distribution scheme for [14] that included resistance to collusion attacks. The scheme is divided into the four phases described below.

Phase 1: Setup. First, the GM randomly chooses m polynomials $f_1(x), \ldots, f_m(x) \in F_p[x]$, each of degree t.

Second, the GM randomly chooses numbers $\alpha_1, \ldots, \alpha_m \in F_p$ for each session.

Third, the GM chooses a random secret value $t_i \in F_p$ for user U_i and the t_i values are different from each other. Then, the GM sends the personal secret $S_i = \{t_i, \alpha_{j'}, f_{j'}(t_i)\}$ to user U_i in a secure manner. (The term j' denotes the session number when the user joins the group and $\alpha_{j'} \in \{\alpha_1, \ldots, \alpha_m\}$.)

Then, the GM randomly chooses a prime, initial key seed $K_0 \in F_p$, which is kept secret and m numbers $\{\beta_j\}_{j=1}^{m} \in F_p$ as the self-healing keys.

The GM computes a key seed and corresponding key chain for each session using two one-way hash functions H_1, H_2 and m numbers $\{\beta_j\}_{j=1}^{m}$. For $1 \leq j \leq m$, the key seed of session j is computed as shown:

$$K_j^0 = H_1\left(K_{j-1}, \beta_j\right). \tag{1}$$

And the key chain of session j of length j is computed as shown:

$$K_j^{j-1} = H_2\left(K_j^{j-2}\right) = H_2^{j-1}\left(K_j^0\right), \tag{2}$$

where $H_2^i()$ means applying the hash operation i times. Then, $\{K_j^0, K_j^1, \ldots, K_j^{j-1}\}$ is the key chain of session j, and the group session key in session j is $K_j = K_j^{j-1}$.

Phase 2: Broadcast. Let $\mathcal{U}_{\text{act}_j} = \{U_{\text{act}_1}, \ldots, U_{\text{act}_{a_j}}\}$ be the set of all active users for session j, where a_j is the number of active users in session j. Let $\mathcal{T}_{\text{act}_j} = \{t_{\text{act}_1}, \ldots, t_{\text{act}_{a_j}}\}$ be the set of all active users' secret values in session j. Then, the GM generates $\{G_j^1, G_j^2, \ldots, G_j^j\}$ of size j as a masking key sequence for session j by applying XOR on both $\alpha_{j'}$, and every key forms the key chain of session j, where

$$G_j^{j'} = K_j^{j'-1} \oplus \alpha_{j'}. \tag{3}$$

In session j, the GM broadcasts the following message:

$$\begin{aligned} B_j = &\left\{Z_j^{j'}(x) = A_j^{j'}(x) G_j^{j'} + f_{j'}(x)\right\}_{j'=1}^{j} \\ &\cup \left\{E_{K_j^0}(\beta_1), E_{K_j^1}(\beta_2), \ldots, E_{K_j^{j-1}}\left(\beta_j\right)\right\}, \end{aligned} \tag{4}$$

where $A_j^{j'}(x) = (S_j^{j'} \cdot x - T_j) \prod_{i=1}^{a_{j'}} (x - t_{\text{act}_i}) + 1$ is an access polynomial. When an active user U_{act_i} receives the broadcast message B_j of session j, U_{act_i} can evaluate $A_j^{j'}(t_{\text{act}_i}) = 1$ by using its secret value t_{act_i}, where j' denotes that U_{act_i} has joined the group in session j'. However, a revoked user can only evaluate a random value.

Phase 3: Group Session Key and Self-Healing Key Recovery. When a nonrevoked user U_i in session j, who joins in the group in session j', receives the broadcast message B_j of session j, U_i can recover the group session key K_j as follows.

First, U_i computes $G_j^{j'} = z_j^{j'}(t_i) - f_{j'}(t_i)$ from (4), where $f_{j'}(t_i) \in s_i$ and $A_j^{j'}(t_i) = 1$. Then, U_i evaluates $K_j^{j'-1} = G_j^{j'} \oplus \alpha_{j'}$ from (3), where $\alpha_{j'}$ is secret value of U_i.

Then U_i can compute all the future keys $\{K_j^{j'}, K_j^{j'+1}, \ldots, K_j^{j-1}\}$ in the key chain of session j by using the one-way hash function $H_2()$. The group session key of session j is $K_j = K_j^{j-1} = H_2^{j-j'}(K_j^{j'-1})$.

Then, U_i can decrypt $\{E_{K_j^{j'-1}}(\beta_{j'}), E_{K_j^{j'}}(\beta_{j'+1}),\ldots, E_{K_j^{j-1}}(\beta_j)\}$ by using the corresponding keys $\{K_j^{j'-1}, K_j^{j'},\ldots, K_j^{j-1}\}$ to get the corresponding self-healing keys $\{\beta_{j'}, \beta_{j'+1},\ldots,\beta_j\}$.

However, a revoked user can recover neither the group session key nor the self-healing keys of session j, since $A_j^{j'}(t_i)$ is a random number for any user $U_i \in R_j$.

Phase 4: Add Group Members. If a new user wants to join the group in session j, the GM chooses a never-used identity $v \in \{1,2,\ldots,n\}$ for U_v. Then, the GM selects a random secret value $t_v \in F_p$ and sends the personal secret key $S_v = \{t_v, \alpha_j, f_j(t_v)\}$ to U_v using RSA algorithm.

3. The Proposed Scheme

In this section, we propose an improved version of Bao and Zhang's scheme [15] using secret sharing. In our scheme, we use only one secret polynomial and modified access polynomials, which lower the communication overhead. Our scheme is divided into four phases, as follows.

Phase 1: Initiation. First, the GM creates a polynomial $f(x) \in F_q[x]$ of degree t as the secret polynomial. Then, the GM chooses $\{g_i\}_{i=1}^m$ as m generators with order q in GF(p) and $\{\alpha_i\}_{i=1}^m$ for each session.

Second, the GM selects a unique identity $u_i \in F_q$ for user U_i and sends $S_i = \{u_i, \alpha_{j'}, f(u_i) \bmod q\}$ to user U_i for $i = 1,\ldots,n$ as personal secret keys via a secure communication channel, where j' denotes the session number when the user joined the group. For example, user U_r, who joins the group in session 1, will receive $S_r = \{u_r, \alpha_1, f(u_r) \bmod q\}$.

Then, the GM randomly chooses a prime initial key seed $K_0 \in F_p$, which is kept secret, and m numbers $\{\beta_j\}_{j=1}^m \in F_p$ as the self-healing keys.

In our scheme, as in Du-He's scheme [14], we still use key chains. The GM computes a key seed and corresponding key chain for each session using two one-way hash functions H_1, H_2 and m numbers $\{\beta_j\}_{j=1}^m$. For $1 \le j \le m$, the key seed of session j is computed by (1): $K_j^0 = H_1(K_{j-1}, \beta_j)$.

And the key chain of session j of length j is computed by (2): $K_j^{j-1} = H_2(K_j^{j-2}) = H_2^{j-1}(K_j^0)$, where $H_2^i()$ means applying the hash operation i times. Then, $\{K_j^0, K_j^1,\ldots, K_j^{j-1}\}$ is the key chain of session j and the group session key in session j is $K_j = K_j^{j-1}$.

Phase 2: Broadcast. Assume that $R_j \subseteq U$ and $|R_j| \le t$ are the sets of all revoked users in and before session j, respectively. Let G_j be the set of all nonrevoked users in session j. In session j, the GM chooses a set of nonzero indices $X_j = \{x_{j,1}, x_{j,2},\ldots, x_{j,t}\}$ such that $I_{R_j} \subseteq X_j$, but $W_j \cap I_{G_j} = \emptyset$, where I_{R_j} denotes the set of indices of the users in R_j, and I_{G_j} represents the indices of users in G_j. Let $\mathcal{U}_j^{j'} = \{u_{j',1}, u_{j',2},\ldots, u_{j',n_{j'}}\}$ be the set of indices of the users who join the group in session j' and are still active in session j, where $n_{j'}$ is the number of users of the set and $1 \le j' \le j$ and $G_j = \bigcup_{j'=1}^{j} \mathcal{U}_j^{j'}$. Then, the GM computes a sequence $\{Z_j^{j'}\}_{j'=1}^{j}$ using the key chain of session j as shown:

$$Z_j^{j'}(x) = \left(K_j^{j'-1} \oplus \alpha_{j'}\right) + g_j{}^{f(0)} A_j^{j'}(x) \bmod p, \tag{5}$$

where $A_j^{j'}(x) = (Y_j^{j'} x - T_j)\prod_{i=1}^{n_{j'}}(x - u_{j',i}) + 1$ is a modified-access polynomial. $Y_j^{j'}$ and T_j are randomly selected by the GM in F_p, such that $Y_j^{j'}/T_j$ is different from all users' indices. When an active user U_i receives the broadcast message B_j of session j, U_i can evaluate $A_j^{j'}(u_i) = 1$ by using its secret identity value u_i, where j' denotes that U_i joined the group in session j'. However, a revoked user or an active user who does not join in the group in session j' only can evaluate a random value.

Then, the GM broadcasts the following message B_j:

$$\begin{aligned} B_j = g_j \cup \left\{x_{j,i}, g_j^{f(x_{j,i})}\right\}_{i=1}^{t} \cup \left\{Z_j^{j'}(x)\right\}_{j'=1}^{j} \\ \cup \left\{E_{K_j^0}(\beta_1), E_{K_j^1}(\beta_2),\ldots, E_{K_j^{j-1}}\left(\beta_j\right)\right\}. \end{aligned} \tag{6}$$

Phase 3: Group Session Key Recovery and Self-Healing Key Recovery. When a non-revoked user U_i, who joins the group in session j', receives the broadcast message B_j of session j, he or she can recover $g_j^{f(0)}$ by Lagrange's interpolation using B_j and her or his personal secret keys as following:

$$g_j^{f(0)} = \prod_{l=0}^{t}\left(g_j^{f(x_{j,l})}\right)^{w_l} \bmod p, \tag{7}$$

where

$$w_l = \prod_{\substack{k=0 \\ k \ne l}}^{t} \frac{-x_{j,k}}{x_{j,l} - x_{j,k}}. \tag{8}$$

With $x_{j,0} = u_i$, user U_i can recover $g_j^{f(0)}$, then he or she can recover $K_j^{j'-1}$ by (5) with $A_j^{j'}(x) = 1$, as follows:

$$K_j^{j'-1} = \left(Z_j^{j'}(u_i) - g_j^{f(0)} \bmod p\right) \oplus \alpha_{j'}, \tag{9}$$

where j' denotes the session number when U_i joined the group, and $\alpha_{j'}$ is the secret of user U_i distributed by the GM when he or she joins the group in session j'.

Then, U_i computes the group session key of session j as $K_j = K_j^{j-1} = H_2^{j-j'}(K_j^{j'-1})$.

User U_i also can recover the self-healing key $\{\beta_{j'}, \beta_{j'+1},\ldots,\beta_j\}$ using $K_j^{j'-1}$ and B_j. First, U_i computes all the keys $\{K_j^{j'}, K_j^{j'+1},\ldots, K_j^{j-1}\}$ in the key chain of session j by using the one-way hash function $H_2()$. Then, U_i can decrypt $\{E_{K_j^{j'-1}}(\beta_{j'}), E_{K_j^{j'}}(\beta_{j'+1}),\ldots, E_{K_j^{j-1}}(\beta_j)\}$ by using the keys $\{K_j^{j'-1}, K_j^{j'},\ldots, K_j^{j-1}\}$ to get $\{\beta_{j'}, \beta_{j'+1},\ldots,\beta_j\}$. Then,

the user with session key $K_{j'}$ can recover all session keys between session j' to j based on (1) and (2).

A user who was revoked in session j cannot recover the current group session key or the self-healing key even with the B_j, since he or she cannot recover $g_j^{f(0)}$ based on Lagrange's interpolation.

Phase 4: New User Added. If a user U_x wishes to be added to the group in session j, GM chooses a unique and never-used identity u_x for U_x and sends the secret $S_x = \{u_x, \alpha_j, f(u_x) \bmod q\}$ to U_x using the RSA algorithm.

4. Security and Performance Analyses

In this section, we show that our proposed scheme has self-healing property, forward security, backward security, and resistance to collusion attacks. Compared with Bao and Zhang's scheme [15], our scheme has lower communication overhead.

4.1. Self-Healing Property. Assume that U_i, who join the group in session j', are active in session j_1 and session j_2, where $1 \le j' \le j_1 \le j_2$. And U_i receive session-key broadcast messages B_{j_1} and B_{j_2} but lose the session key broadcast message B_j, where $j_1 < j < j_2$. Users U_i can still recover all the lost session keys K_j for $j_1 < j < j_2$ as follows.

(1) When the broadcast message B_{j_1} is received, U_i can recover $g_{j_1}^{f(0)}$ using their personal secrets by (7) and (8). Then, because U_i are active users in session j_1, U_i can recover $K_{j_1}^{j'-1}$ by (5) and (9), where $A_{j_1}^{j'}(u_i) = 1$. Then, U_i compute the group session key of session j as $K_{j_1} = K_{j_1}^{j_1-1} = H_2^{j_1-j'}(K_{j_1}^{j'-1})$.

(2) When the broadcast message B_{j_2} is received, U_i can recover $g_{j_2}^{f(0)}$ using their personal secrets by (7) and (8). Then, because U_i are active users in session j_2, U_i can recover $K_{j_2}^{j'-1}$ by (5) and (9), where $A_{j_2}^{j'}(u_i) = 1$. U_i compute all the keys $\{K_{j_2}^{j'}, K_{j_2}^{j'+1}, \ldots, K_{j_2}^{j_2-1}\}$ in the key chain of session j_2 by using the one-way hash function $H_2()$. Then, U_i can recover $\{\beta_{j'}, \beta_{j'+1}, \ldots, \beta_{j_1}, \ldots, \beta_{j_2}\}$ using the keys $\{K_{j_2}^{j'-1}, K_{j_2}^{j'}, \ldots, K_{j_2}^{j_2-1}\}$ by decryption $\{E_{K_{j_2}^{j'-1}}(\beta_{j'}), E_{K_{j_2}^{j'}}(\beta_{j'+1}), \ldots, E_{K_{j_2}^{j_2-1}}(\beta_{j_2})\}$.

(3) With K_{j_1} and $\{\beta_{j_1}, \ldots, \beta_{j_2}\}$, U_i can recover all session keys K_j for $j_1 < j < j_2$ by (1) and (2).

Therefore, our scheme achieves the self-healing property.

4.2. Forward Secrecy. Let $R_j \subseteq U$ and $|R_j| \le t$ be the set of all revoked users in and before session j, respectively. Then, we show that the coalition R_j cannot get any information about the current session key K_j, even with the previous group session keys before session j. To recover the session key $K_j = K_j^{j-1} = H_2^{j-j'}(K_j^{j'-1})$, user $U_i \in R_j$ must recover $K_j^{j'-1} = (Z_j^{j'}(u_i) - g_j^{f(0)} A_j^{j'}(u_i)) \oplus \alpha_{j'}$ by (5), where j' denotes the session number when U_i joined the group. But for revoked users U_i, $A_j^{j'}(u_i)$ is a random value that is not known by U_i. Moreover, U_i cannot recover $g_j^{f(0)}$ even with all of the information of all revoked users, because, according to Lagrange's interpolation, to recover $g_j^{f(0)}$, U_i must know at least $(t+1)$ number pairs, such as $(x_i, g_j^{f(x_i)})$, where $(x_i, f(x_i))$ is a point on $f(x)$. Since the size of the coalition R_j is, at most, t, the coalition R_j cannot recover $g_j^{f(0)}$. In [17], Harn showed that U_i may be able to recover $g_j^{f(0)}$ with the previous $g_{j_1}^{f(0)}$ and $g_{j_2}^{f(0)}$ when $g_j = g_{j_1} g_{j_2}$. But in our scheme, the probability is 2^{-160}, which is extremely low and can be almost neglected. After all, the coalition R_j cannot get any information about the current session key K_j.

The above analysis shows that our scheme is forward secure.

4.3. Backward Secrecy. Let $J_j \subseteq U$, where $|J_j| \le t$, be the set of all users who join the group after session j. We will show that the coalition J_j cannot get any information about any previous session key K_{j_1} for $j_1 \le j$, even with the knowledge of group keys after session j.

Users in J_j can get only the session keys $\{K_{j+1}, K_{j+2}, \ldots\}$ and self-healing keys $\{\beta_{j+1}, \beta_{j+2}, \ldots\}$. Without loss of generality, one can get $K_{j+1} = H_2^j(K_{j+1}^0)$ and $K_{j+1}^0 = H_1(K_j, \beta_{j+1})$ by (1) and (2), where $H_1(\cdot)$ and $H_2(\cdot)$ are two one-way hash functions. It is computationally infeasible for any user in J_j to compute any previous session key K_{j_1} with keys $\{K_{j+1}, K_{j+2}, \ldots\}$ and self-healing keys $\{\beta_{j+1}, \beta_{j+2}, \ldots\}$ for $j_1 \le j$.

However, users in J_j could attempt to recover the previous session keys by their personal secret keys and the previous broadcast messages. However, by (5) and (6), it is evident that the previous broadcast messages do not have the equations for users in J_j. So users in J_j cannot recover the previous session keys.

The above analysis shows that our scheme is backward secure.

4.4. Resistance to Collusion Attack. Let $R_{j_1} \subseteq U$ be the set of all revoked users in and before session $j_1 + 1$ and let $J_{j_2} \subseteq U$ be the set of all users who join the group from session j_2. We will show that collusion of R_{j_1} and J_{j_2} cannot recover any session key K_j $(j_1 < j < j_2)$ with their personal secret keys and the broadcast message B_{j_1} and B_{j_2}.

To recover session key K_j $(j_1 < j < j_2)$, $R_{j_1} \cup J_{j_2}$ must recover the self-healing keys $\beta_{j_1+1}, \beta_{j_1+2}, \ldots, \beta_{j_2-1}$. Without loss of generality, assume that U_a joins the group in session j_1 and that U_b joins the group in session j_2. For the equation, $Z_{j_2}^{j_1}(x) = (K_{j_2}^{j_1-1} \oplus \alpha_{j_1}) + g_{j_2}^{f(0)} A_{j_2}^{j_1}(x)$, since user U_a, who joined the group in session j_1, is not active in session j_2, $A_{j_2}^{j_1}(u_a)$ is a random number. Then, U_a cannot recover $K_{j_2}^{j_1-1}$ even with α_{j_1} and $g_{j_2}^{f(0)}$ provided by user U_b. Therefore, users in $R_{j_1} \cup J_{j_2}$

Table 1: Comparison among different schemes.

Schemes	Storage overhead	Communication overhead	Long lifespan	Forward secrecy	Backward secrecy	Collusion attack resistance
Staddon et al.'s scheme 3 [3]	$(m-j-1)^2 \log p$	$(mt^2+2mt+m+t)\log p$	No	Yes	Yes	Yes
Liu et al.'s scheme 3 [5]	$(m-j+1)\log p$	$(2tj+j)\log p$	No	Yes	Yes	Yes
Dutta et al.'s scheme [11]	$3\log p$	$(t+1+j)\log p$	Yes	No	No	No
Xu and He's scheme 1 [13]	$4\log p$	$\left(\max\left\{t+j+1, a_j+j+2\right\}\right)\log p$	Yes	Yes	Yes	No
Du and He's scheme [14]	$(m-j+2)\log p$	$[(t+1)j+j]\log p$	No	Yes	Yes	Yes
Bao and Zhang's scheme [15]	$3\log p$	$\left(\max\left\{a_j+2, t+1\right\}\cdot j+j\right)\log p$	Yes	Yes	Yes	Yes
Our scheme	$3\log p$	$\left(a_j+3j+2t+1\right)\log p$	Yes	Yes	Yes	Yes

cannot recover the self-healing keys $\beta_{j_1+1}, \beta_{j_1+2}, \ldots, \beta_{j_2-1}$ and session key K_j $(j_1 < j < j_2)$.

The above analysis shows that our scheme can resist collusion attacks.

4.5. Constant Storage Overhead and Lower Communication Overhead. Our scheme has a constant storage overhead, which comes only from the user's personal secret keys $\{u_i, \alpha_{j'}, f(u_i) \bmod q\}$. So, the storage overhead is $(3\log p)$ bits.

In our scheme, we use only one secret polynomial and modified access polynomials, which lower the communication overhead. The communication overhead is $(a_j + 3j + 2t + 1)\log p$, where t is the maximum number of revoked users, and a_j is the number of active users in session j. Table 1 shows the comparison among the different schemes.

5. Conclusions

In this paper, we proposed a modified and an improved version of Bao and Zhang's scheme. Our scheme uses only one secret polynomial and modified access polynomials, which achieve a lower communication overhead. In addition, our scheme has the properties of constant storage, long lifespan, forward secrecy, backward secrecy, and resistance to collusion attacks. And, compared with the previous schemes, our proposed scheme is an efficient and secure, self-healing, key-distribution scheme for WSNs.

Acknowledgments

Financial supports for this study provided by grant from National Natural Science Foundation of China (Project nos. 51108060, 50921001, 90815022), National Key Technology Research and Development Program during the Twelfth Five-Year Plan Period (Project no. 2011BAK02B01), and the Fundamental Research Funds for the Central Universities (Project no. DUT12JR13) are gratefully acknowledged.

References

[1] Y. Yu, J. Ou, J. Zhang, C. Zhang, and L. Li, "Development of wireless MEMS inclination sensor system for swing monitoring of large-scale hook structures," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 4, pp. 1072–1078, 2009.

[2] Y. Yu and J. Ou, "Wireless collection and data fusion method of strain signal in civil engineering structures," *Sensor Review*, vol. 29, no. 1, pp. 63–69, 2009.

[3] J. Staddon, S. Miner, M. Franklin, D. Balfanz, M. Malkin, and D. Dean, "Self-healing key distribution with revocation," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 241–257, May 2002.

[4] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[5] D. Liu, P. Ning, and K. Sun, "Efficient self-healing group key distribution with revocation capability," in *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS'03)*, pp. 231–240, New York, NY, USA, October 2003.

[6] S. M. More, M. Malkin, J. Staddon, and D. Balfanz, "Sliding-window self-healing key distribution," in *Proceedings of the ACM Workshop on Survivable and Self-Regenerative Systems (In Association with 10th ACM Conference on Computer Communications Security)*, pp. 82–90, October 2003.

[7] C. Blundo, P. D'Arco, A. Santis, and M. Listo, "Definitions and bounds for self-healing key distribution," in *31st International Colloquium on Automata, Languages, and Programming (ICALP 2004)*, J. Díaz, J. Karhumaki, A. Lepistö, and D. Sannella, Eds., vol. 3142 of *Lecture Notes in Computer Science*, pp. 234–246, Springer, NewYork, NY, USA, 2004.

[8] D. Hong and J. S. Kang, "An efficient key distribution scheme with self-healing property," *IEEE Communications Letters*, vol. 9, no. 8, pp. 759–761, 2005.

[9] T. Biming and H. Mingxing, "A Self-healing key distribution scheme with novel properties," *International Journal of Network Security*, vol. 7, no. 1, pp. 115–120, 2008.

[10] C. Blundo, P. D'Arco, and M. Listo, "A flaw in a self-healing key distribution scheme," in *Proceedings of the Information Theory Workshop*, pp. 163–166, Paris, France, 2003.

[11] R. Dutta, E. Chang, and S. Mukhopadhyay, "Efficient self-healing key distribution with revocation for wireless sensor networks using one way hash chains," in *Proceedings of the 5th*

International Conference on Applied Cryptography and Network Security (ACNS'07), J. Katz and M. Yung, Eds., vol. 4521 of *Lecture Notes in Computer Science*, pp. 385–400, Springer, Heidelberg, Germany, 2007.

[12] R. Dutta, Y. D. Wu, and S. Mukhopadhyay, "Constant storage self-healing key distribution with revocation in wireless sensor network," in *Proceedings of the IEEE International Conference on Communications (ICC'07)*, pp. 1323–1328, Glasgow, UK, June 2007.

[13] Q. Y. Xu and M. X. He, "Improved constant storage self-healing key distribution with revocation in wireless sensor network," in *Information Security Applications (WISA 2008)*, vol. 5379 of *Lecture Notes in Computer Science*, pp. 41–55, Springer, Heidelberg, Germany, 2009.

[14] W. Du and M. X. He, "Self-healing key distribution with revocation and resistance to the collusion attack in wireless sensor networks," in *Provable Security (ProvSec 2008)*, vol. 5324 of *Lecture Notes in Computer Science*, pp. 345–359, Springer, Heidelberg, Germany, 2008.

[15] K. H. Bao and Z. F. Zhang, "Collusion attack on a self-healing key distribution with revocation in wireless sensor networks," in *Information Security Applications (WISA 2010)*, vol. 6513 of *Lecture Notes in Computer Science*, pp. 221–233, Springer, Heidelberg, Germany, 2011.

[16] C. Blundo, P. D'Arco, A. de Santis, and M. Listo, "Design of self-healing key distribution schemes," *Designs, Codes, and Cryptography*, vol. 32, no. 1–3, pp. 15–44, 2004.

[17] L. Harn, "Efficient sharing (broadcasting) of multiple secrets," *IEE Proceedings: Computers and Digital Techniques*, vol. 142, no. 3, pp. 237–240, 1995.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 547471, 8 pages
doi:10.1155/2012/547471

Research Article

A Secure Hierarchical Key Management Scheme in Wireless Sensor Network

Yiying Zhang,[1,2] Xiangzhen Li,[3] Jianming Liu,[2] Jucheng Yang,[4] and Baojiang Cui[1]

[1] *School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2] *Department of Production & Technology, State Grid Information & Telecommunication Company Ltd., Beijing 100761, China*
[3] *IoT Center, State Grid Electric Power Research Institute, Nanjing 210003, China*
[4] *College of Computer Science & Information Engineering, Tianjin University of Science & Technology, Tianjin 300222, China*

Correspondence should be addressed to Yiying Zhang, zhangyiying1973@hotmail.com

Received 12 June 2012; Revised 28 July 2012; Accepted 12 August 2012

Academic Editor: Hailun Tan

Due to the vulnerable environment, limited recourse and open communication channel, wireless sensor networks (WSNs) are necessary to be protected from various attacks. The key management is an important way to protect the communication in WSNs. In this paper, we present a hierarchical key management scheme (HKMS) which can efficiently enhance the security and survivability for the clustered WSNs. Different from previous works, the HKMS distributes keys based on hop counts and one-way function by the clustered architecture, which not only localizes the key things but also has no overhead. The HKMS provides the session keys among sensors and the cluster key between the cluster head and member nodes. The HKMS dynamically generates different keys based on different hops in different periods which can protect the network from the compromised nodes and reduce the high probability of the common keys without any special sensors (such as the anchor nodes). The security analysis and simulation show the HKMS can prevent several attacks effectively and reduce the energy consumption.

1. Introduction

Wireless sensor network (WSN) is usually considered as a large-scale network with thousands of tiny sensors and deployed in smart grid, smart city, smart home, and so forth to sense the information. As the most important part in the perception layer of internet of things (IoTs), WSNs are deployed for sensing, monitoring, or controlling various objects [1, 2]. However, there are still some limitations, such as the capability of computation, low energy, small storage and open communication channel. Therefore, WSNs are vulnerable to various attacks, and the security in WSNs is required [3–8].

Some literatures focus on localizing the key things. In [2, 3], the authors presented RPKH and location-dependent key management (LDK) schemes to provide the local key management. The RKPH and LDK utilize different nodes including the normal nodes and anchor nodes to generate keys by different transmission ranges.

In [3], LDK has been presented and it employs the heterogeneous sensors to build a clustered sensor network. In LDK, there are higher ability nodes, the anchor nodes, as the management nodes. The anchor nodes use the different location information to generate sets of keys. Neighboring nodes can establish secure communication link by determining the common keys via exchanging their key materials. LDK takes advantage of relative location of nodes after deployment by utilizing anchor nodes at different power levels. Based on different locations, nodes can receive different sets of keys from anchor nodes. Neighboring nodes can establish secure communication link through the common keys. LDK can increase the direct connectivity ratio among nodes. However, in LDK, nodes need to transmit a message that consists of all the key materials when determining common keys to establish secure links. Therefore, it consumes lots of communicating energy and is not efficient for WSNs, and the adversary also can eavesdrop on the key materials during nodes exchanging packets. Moreover, the special node (anchor node) makes it difficult to deploy.

In [5], the ARPKH is designed based on random key distribution in the heterogeneous sensor networks, which

uses separate keys in different clusters and take into consideration distance of sensors from theirs cluster head. Compared with the RPKH, the ARPKH considers a multiple shared keys between pairwise nodes. When a key that used for establishing the secure link between two nodes is revealed, the link has been expired and then the connectivity is broken. Moreover, ARPKH will change the alternative shared keys to replace the revealed key and establish a new secure link between two nodes again. However, the ARPKH needs alternative shared key replacement, which makes sensors predistribute more keys and occupy larger storage. Moreover, ARPKH also needs the anchor nodes as the cluster head, which makes it impracticable.

In this paper, we present a hierarchical key management scheme (HKMS) in the clustered wireless sensor network [9]. Different from the previous works, our network needs no any special nodes (e.g., high-energy or high-capability nodes), which makes it more practicable to deploy. Meanwhile, HKMS also distributes keys through the key seed (nonce) according to TTL (time to live), which has higher level security than previous works transferring key things directly.

The HKMS builds the key system with the clustered architecture formation. The cluster head gets the hop counts from cluster head to the member nodes with ACK packets and then uses the hop count to determine TTL as well as a certain numbers of nonces for building the key system. During the key distribution, nodes in different hop ranges will be obtained different keys. With the cluster head reselection, the key system will be rebuilt, and then the key should be reassigned.

Considering about the security and the life time of WSNs, we will rekey to refresh the cluster and the keys. During the rekey phase, the cluster will elect new cluster head which calculates the new distance from CH to member nodes and then generate the new key system based on the old one.

Our solution has the following scientific research contributions: (1) HKMS utilizes the hierarchical architecture to localize the key things, which prevents the compromised nodes threat the entire network. (2) Without any overhead, HKMS counts the hop count in the cluster formation, which can effectively reduce energy consumption. (3) HKMS employs the normal wireless sensor network but not special nodes, which makes it more practicable.

The rest of this paper is organized as follows Section 2 presents the system model. Section 3 describes the key management in detail. Section 4 evaluates HKMS using security analysis, meanwhile, we simulate the solutions to evaluate the performances of HKMS. Finally, we end the paper with a conclusion as well as the further work in Section 5.

2. System Model

2.1. Network Model. Given G is a WSN which consists of m clusters, that is, $G = C_1 \cup C_2 \cdots \cup C_m$ and $C_i \cap C_j = \phi$, $i \neq j$, where C_i is a cluster with the cluster head (CH or CH_i) and member nodes [10, 11]. In a cluster, the CH collects and aggregates packets from its member nodes and then forwards them to the base station (BS). Normally, a member sensor can transfer packets to CH through several hops. Assume a normal node $v_j^i \in C_i$, it communicates with CH through multihop, as shown in Figure 1.

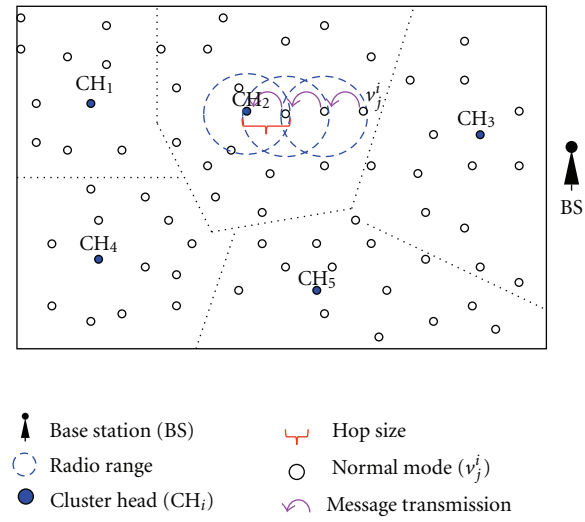


Figure 1: The considered wireless sensor network.

Table 1: Notations.

Notation	Explanation
K_I	The initial key shared by all nodes
ID_{CH}	ID of the cluster head
n_i	The ith nonce in the set of nonces N
ID_{v_j}	ID of member node v_j
$f()$	The *one-way* function
TTL	Time to live
C_i	The ith cluster in the WSN
k_j^i	The ith key for the member node v_j
v_j^i	The member node in C_i

2.2. Assumptions. In our network, all sensor nodes are deployed in the network uniformly and randomly and are static. Each sensor has a unique ID. If a node is compromised, all of the information in this node will be revealed including the key materials [12]. The sensors in network should be in at least one cluster.

2.3. Notations. In Table 1, we list some notations used in this paper.

3. The Hierarchical Key Management Scheme

In this section, we introduce the hierarchical key management scheme (HKMS) in detail. Before the deployment of the sensor network, each sensor is predistributed an initial key K_I for the security in deployment phase, the initial key provides the communication during the formation phase and will be erased after key deployment [12].

3.1. The Cluster Head Election. As mentioned above, considering the energy efficiency and management facilitation of

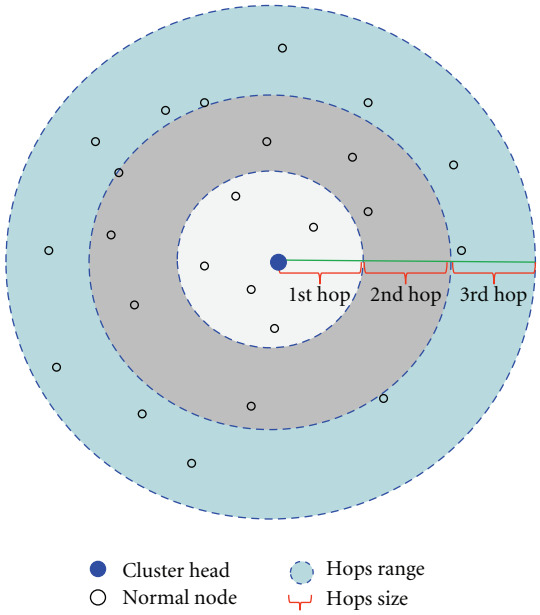


Figure 2: The nodes in different hop range to CH (assume these nodes join in the same cluster, TTL = 3).

WSN, we adopt the hierarchical architecture for our network [10, 11, 13]. Firstly, a node itself decides whether it becomes a candidate CH or not according to the cluster head election algorithm [10, 14]. The node will announce the candidate information to other nodes. And other nodes which may accept several election campaign messages, and they will choose one to join it as follows.

3.2. The Cluster Formation. Once a node becomes a cluster head, it will send a *beacon* message to other sensors to form a cluster. Each sensor may receive several different *beacon* messages from different candidate cluster heads, but it only can join one cluster.

When the CH broadcasts a *beacon* message encrypted by K_I contains ID of CH, the transmission range is limited by TTL, that is, calculated by hop count. Usually, TTL is the hop count plus one. And then, TTL and a set of *nonces* named N, the random numbers. And the *nonces* will be different with different TTL, that is, if the TTL = 3, then the sensor will get four (TTL + 1) nonces, such as $N = \{n_1, n_2, n_3, n_4\}$. We generate more nonces (e.g., TTL + 1) for the connectivity, especially the common keys. Where TTL is to limit the cluster size, for example, TTL = 3, and it will be decreased for each forwarding until it becomes 0 and the beacon message will be dropped.

Therefore, depending on the cluster size (TTL), other nodes can receive different sets of *beacon* messages from different CHs as (1) in different distance (hop ranges) as shown in Figure 2:

$$N = \{n_i \mid n_1, \ldots, n_{\mathrm{TTL}+1}\}$$
$$\mathrm{CH} \longrightarrow * : \{\mathrm{ID}_{\mathrm{CH}}, N, \mathrm{TTL}\}_{K_I}. \quad (1)$$

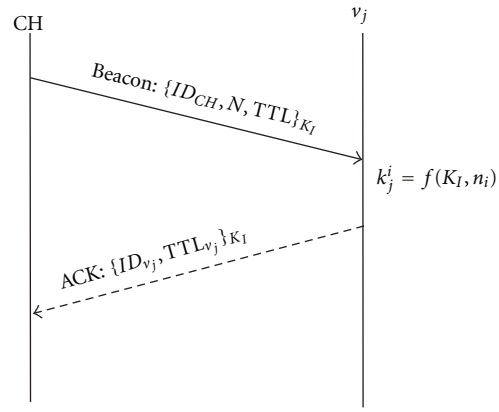


Figure 3: The initial key generation process.

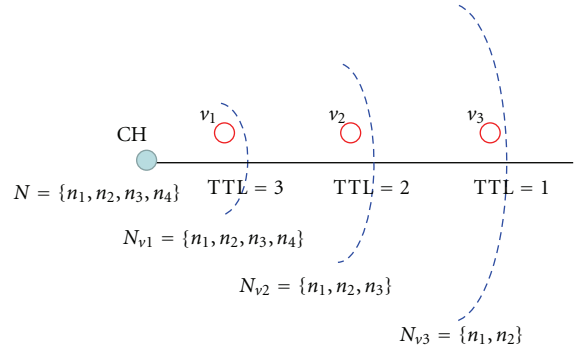


Figure 4: The deployment of nonces from the cluster head (TTL = 3).

3.3. The Initial Key Generation. Assume $v_j \in C_i$, $v_j \neq \mathrm{CH}$, we call v_j as member node. When member node v_j receives a beacon message and wants to join the cluster C_i, it counts the TTL and sends the ACK including the ID_{v_j} and the TTL_{v_j} back to its interest cluster, and then the cluster head knows the hops from ID_{v_j} to CH.

The *beacon* messages are orderly transmitted at different distance levels. And then, the member node v_j decrypts the *beacon* messages and obtains $\mathrm{ID}_{\mathrm{CH}}$ and then set of *nonces* N_i.

And then, v_j calculates the candidate keys k_j^i and gets the key set K^i based on the received *nonces* as follows:

$$K^i = \left\{k_j^i \mid k_1^i, \ldots, k_{\mathrm{TTL}_{v_j}}^i\right\}, \quad k_j^i = f(k_I, n_i), \quad (2)$$

where $f()$ denotes a one-way hash function. The specific algorithm of key distribution is designed as in Algorithm 1.

And the initial key generation process is as shown in Figures 3 and 4.

After the calculations, nodes erase K_I. Consequently, v_j stores the key things as follows in Table 2.

According to Table 2, we can find that the nodes can communicate with its neighbour nodes for the common keys. The specific algorithm of hop count and key information acquirement is as in Algorithm 2.

(1) CH broadcasts *beacon* messages with different *nonces*:
CH $\rightarrow * : \{\mathrm{ID}_{\mathrm{CH}}, N, \mathrm{TTL}\}_{K_I}$
(2) v_j decrypts$\{\mathrm{ID}_{\mathrm{CH}}, N, \mathrm{TTL}\}_{K_I}$
(3) $K^i = \phi$
(4) $k_{\mathrm{length}} = \mathrm{TTL}_{v_j^i}$
(5) *For* $j = 1$ *to* k_{length} {
$k_j^i = f(k_I, n_i)$ // generate the key pool
$K^i = K^i \cup \{k_j^i\}$
}
(6) *Erase* K_I.
(7) *end*.

Algorithm 1: Key distribution algorithm (TTL = 3).

(1) CH broadcasts *beacon* messages with different *nonces*:
CH $\rightarrow * : \{\mathrm{ID}_{\mathrm{CH}}, n_1 \cdots n_{\mathrm{TTL}+1}, \mathrm{TTL}\}_{K_I}$
(2) $v_j \rightarrow \mathrm{CH} : \{\mathrm{ID}_{v_j}, \mathrm{TTL}_{v_j}\}_{K_I}$, *Key pool generation for* v_j by $k_j^i = f(k_I, N_i)$
(3) *Erase* K_I.
(4) CH obtains the hop count: $N_{\mathrm{hops}} = \mathrm{TTL}_{\mathrm{CH}} - \mathrm{TTL}_{v_j} + 1$
(5) According to the N_{hops}, nonces N, and oneway function $f()$, the cluster head can get the ID_{v_j}'s key information.
(6) **end**

Algorithm 2: Hop count and key information acquirement algorithm (TTL = 3).

Table 2: Keys table of member nodes in different hop ranges (TTL = 3).

Keys for 1st hop	Keys for 2nd hop	Keys for 3rd hop
$k_j^1, k_j^2, k_j^3, k_j^4$	k_j^2, k_j^3, k_j^4	k_j^3, k_j^4
$k_j^1, k_j^2, k_j^3, k_j^4$	k_j^2, k_j^3, k_j^4	k_j^3, k_j^4
...	...	...
$k_j^1, k_j^2, k_j^3, k_j^4$	k_j^2, k_j^3, k_j^4	k_j^3, k_j^4

3.4. The Common Key Discovery. For communication with ites neighbouring nodes, member node should establish secure link between them which needs the common keys to encrypt/decrypt messages. According to those candidate keys, member nodes in the same distance receive the same *beacon* messages and they can also generate the same keys. Moreover, the nodes in the adjacent areas also have some duplicate candidate keys.

If a node can receive $\{\mathrm{ID}_{\mathrm{CH}}, N_i, \mathrm{TTL}_i\}_{K_i}$, it also can receive $\{\mathrm{ID}_{\mathrm{CH}}, N_j, \mathrm{TTL}_j\}_{K_i}$, where $\mathrm{TTL}_i > \mathrm{TTL}_j$ $(i < j)$. Since the distance range of hops j covers the distance range of hops i, the node near cluster head has more keys than the one far away from cluster head.

Therefore, each member node v_j generates a list L_j which just stores the keys as follows:

$$L_j = \left\{ k_j^i \mid k_j^{\mathrm{Hops}}, \ldots, k_j^{\mathrm{TTL}_{v_j}+1} \right\}. \tag{3}$$

According to the principle of key generation, given two nodes v_i and v_j $(i < j)$, the set of common keys is S, then we have: $S = L_j \cap L_i = L_j$.

Moreover, since the packets from members will be collected by the cluster head, the cluster head should have the ability to decrypt these messages. During this process, the member nodes should report its keys, which will increase the transmission. Because the *nonces* are sent by the cluster head, it also knows the function, and then it can calculate the keys of members as mentioned in Algorithm 2.

Due to the keys generated by hop count, nodes in the same cluster can be connected. And the path key between v_i and v_j is calculated as follows:

$$k_{ij} = f^{\mathrm{abs}(i-j)}(k_I, N_i). \tag{4}$$

Equation (4) makes it possible for any two nodes in the cluster to communicate with each other. Actually, there is another way to make every two nodes communicate, that is, the last nonce is the same in a cluster, which makes the same key for the cluster. (the last nonce is used to the cluster key).

3.5. The Cluster Key. The cluster key is the key which is used for communication between the CH and its members also for generating new key in next round. Since there are TTL + 1 *nonces*, according to Algorithms 1 and 2, the last nonce in the set N will be transferred to every sensor.

3.6. The Rekey Process. For prolonging the lifetime of the whole network, it is necessary to change the cluster head. On the other hand, the key should be rekeyed for the security [15, 16], otherwise, when CH receives a certain amount of encrypted messages (more than $2^{2k/3}$, k is the length of key) [15], the keys will be no longer safe. According to the

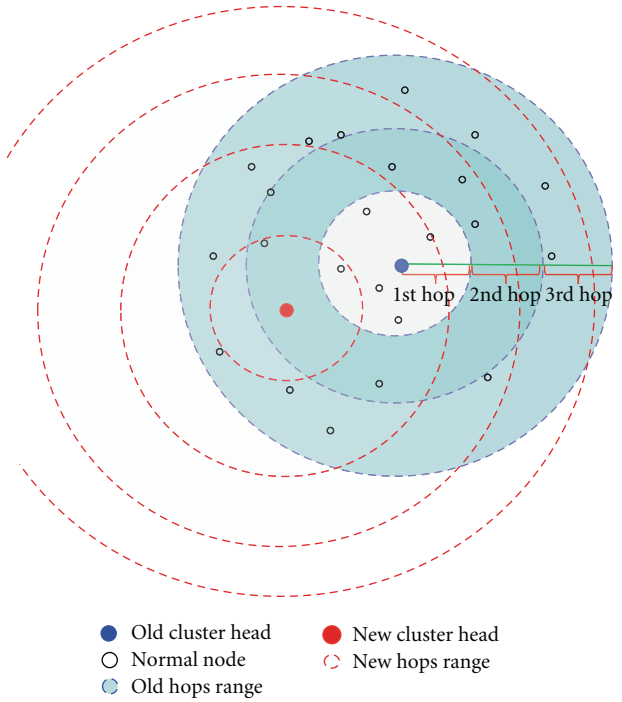


Figure 5: The reselection of cluster head.

requirements above, we should recluster after a certain phase. Assume that the process of reclustering happens inside the cluster, which can reduce the energy consumption.

During the reselection of CH, we can rekey as the initial phase. When the new cluster head has been selected according to the algorithm [10], it will announce itself as the cluster head and recalculate the distance from its members.

As shown in Figure 5, after reclustering, the new cluster head changes not only the relative position but also hop counts from CH to members, which make the *nonces* as well as the key things different.

4. Security Analyses

4.1. The Security Analyses. Compared to previous works, the salient advantage of our solution is that we addressed challenging runtime security issues using localizing key things and design a dynamic key management.

During the cluster formation phase, the cluster head can calculate the hop counts from the cluster head to member nodes, and then the member nodes can generate keys by the *nonces* and hops from the *beacon* message. According to the different hop counts, the cluster is divided into several security belts as shown in Figure 2, the nodes in different belts have different keys. Because the keys are generated by the set of *nonces*, the adjacent nodes have some common keys, which makes it possible to communicate with each other.

Moreover, nodes near the cluster head have more keys than the nodes far away from CH, which means that the far nodes just can submit message to the CH. And then the messages just can be decrypted by near nodes, which

Table 3: Analysis in local key management.

Attack types	RPKH	LDK	HKMS
Selective forwarding	×	×	√
Sink-Hole attack	×	√	√
Sybil attack	√	√	√
Worm-Hole	√	√	√
HELLO flood	√	√	√
DoS	×	×	√

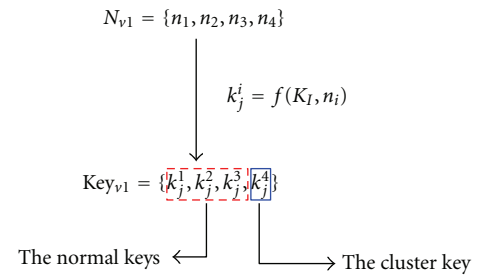


Figure 6: The cluster key in the HKMS.

makes the HKMS protocol more suitable to the collection type wireless sensor networks. The one-way security model prevents the eavesdrop attack, selective-forwarding attack DoS attack (denial-of-service) and hello flood attack, and so forth as shown in Table 3.

To communicate with members, the cluster head utilizes the last nonce as the seed of cluster key which is shared with all the sensors (including the CH) as shown in Figure 6. The cluster also can be used to rekey during the next round cluster, since the rekey process is with the redistribution of nonces. Comparing with RPKH and LDK [2, 3], the HKMS has no special requirements about the nodes, which makes it more feasibly. Also the HKMS utilizes the process of cluster formation to generate the key system without overhead, which reduces more energy consumption than previous works.

Furthermore, the key system forms during the cluster formation, which almost does not consume any energy overhead.

Furthermore, as described above, if a node can receive *beacon* message in the jth hop range, meanwhile, it also can receive *beacon* message transmitted at kth hop range, where $j < k \leq \text{TTL}$. And then, the probability can be given based on the binomial distribution as follows:

$$p_j^{\text{TTL}} = \binom{\text{TTL}}{j} \left(\frac{\text{TTL} - j + 1}{\text{TTL}}\right)^{\text{TTL}-j+1} \times \left(1 - \frac{\text{TTL} - j + 1}{\text{TTL}}\right)^{j-1}. \quad (5)$$

According to (5), with the increase of TTL, the probability also increases, that is, the common keys between two nodes are increased, which enhances the connectivity. The increase of TTL also can shrink the size of each subregion, which decreases the number of nodes who use the

Table 4: Simulation parameters.

Parameter	Value
Area size	$100 * 100$
Quantity of sensor	100
BS position	(50, 250)
Initial energy	2 J
Cluster radius	40 m
Packet size	500 Bytes
E_{elec}	50 nJ/bit
ε_{fs}	10 pJ/bit/m^2
ε_{amp}	0.0013 pJ/bit/m^4
E_{DA}	5 nJ/bit/signal
d_0	86.2 m

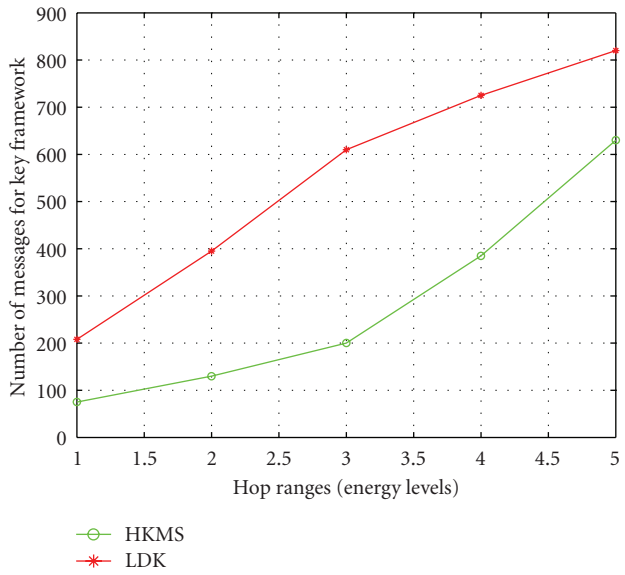


Figure 7: The comparison in energy consumption for key framework with different hop ranges (energy levels).

same communicating key and then localizes the impact of attacks. Moreover, the pair of nodes who do not have enough common keys can communicate with each other via a key path in HKMS. It also can improve the indirect network connectivity and then improve the whole network connectivity.

4.2. Simulation. In this section, we evaluate the performance of HKMS implemented in Visual C++ and MATLAB. The network scenario that we consider in simulation contains 100 nodes. According to the requirement of the HKMS, we designed a wireless sensor network simulation incorporating ECDG, essentially a multihop hierarchical sensor network [17]. The parameters for the simulations are listed in Table 4.

In Table 4, the E_{elec} is for running the transmitter or receiver circuitry; the ε_{amp} is for the transmit amplifier.

Firstly, we compare the performance of HKMS with that of LDK in energy consumption. Figure 7 shows the

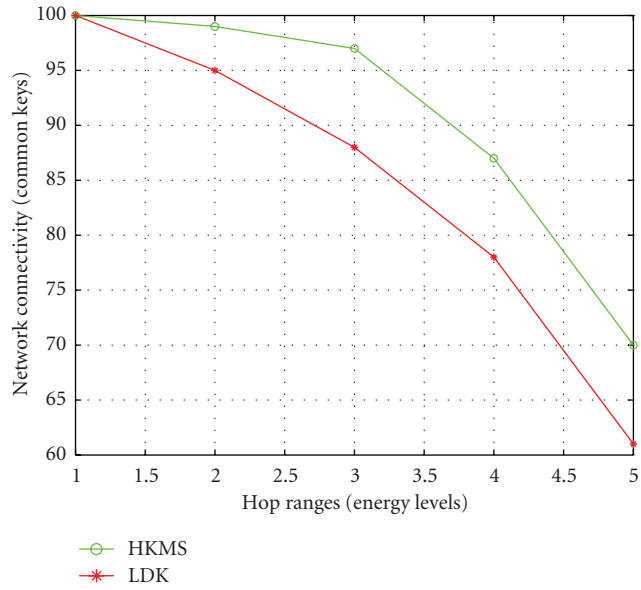


Figure 8: Comparing the performance of HKMS with that of LDK on energy consumption versus network connectivity.

comparison LDK versus HKMS in energy consumption for key framework. From Figure 7, we can see that the number of messages for key formation increased with the increase of number of hops (energy levels in LDK). For LDK, the curve looks smooth for the anchor node which has more ability to enhance the energy level to form clusters and keys. However, LDK needs transfering more messages to generate the keys. Meanwhile, with the increase of hops of HKMS, it needs more messages to be forwarded from far nodes to generate keys. Since the key things are included in the packets of ACK, the HKMS needs less message transmission. As shown in Figure 7, the HKMS uses 50% energy of LDK to form the key framework. However, due to the noise and attenuation, more hops will increase energy consumption when the hops are more than 5.

Under the same simulation environment, Figure 8 demonstrates the comparison of the connectivity of HKMS and LDK. And we can observe that more hops (energy levels) will reduce connectivity. Since the HKMS happens in one cluster, which makes it possible to communicate with each other. However, with the hop count increase, the coverage becomes bigger, which makes it difficult to forward packets. When the hops are more than 3, the QoS almost is less than 80%. For LDK, it also faces the same problem. The sensors in the radio of anchor may be not the number of the cluster. When the energy level increases, the uncertainty also increases, which reduces the connectivity.

Figure 9 shows the excepted number of keys for each member node with different TTL, which indicates that we can adjust the value of TTL to adapt to the network with the different density. Compared with LDK, HKMS has more ability to be employed for different WSNs, even in a network, there can be different size clusters because of the different TTL. In our solution, we can adjust the TTL value to average

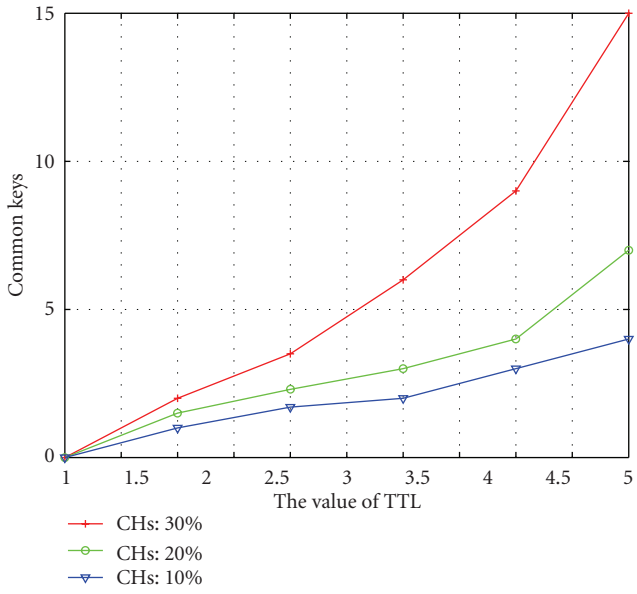


Figure 9: Excepted number of keys for each RN.

the cluster size. Figure 9 shows that the TTL value will change the density of cluster head number in the network. Here, given the TTLs are 3, 4, 5 respectively, the cluster head will be about 10%, 20%, and 30% of all nodes, respectively. From Figure 9, we can see that with the increase of TTL, the number of common keys also increases. As shown in Figure 9, when TTL is 5, the number of CHs is about 20%.

5. Conclusion and Future Work

In this paper, we propose a hierarchical key management scheme (HKMS) to enhance network security and survivability. Unlike previous works, we employ the hierarchical architecture but not fixed-node network. In contrast to other clustered architectural security solutions, the salient advantage of this work is that we addressed challenging security issues by localizing key things. We generate new keys in different hop ranges in a cluster. Also we present a rekey mechanism in the cluster head selection with low energy consumption. Meanwhile, HKMS can adjust the TTL to control the cluster size and the connectivity of nodes in the common keys. The simulations and security analysis show that our solution cannot only reduce the energy consumption effectively but also enhance the security level. In the future, we will focus on how to enhance security in mobile and scalable WSNs.

Acknowledgments

This work was supported by China Postdoctoral Science Foundation Funded Project (2012M510367) and this work was also supported by the following foundation: Important National Science & Technology Specific Projects of China: Next-generation broadband wireless mobile communication networks (2011ZX03005-002).

References

[1] Y. Zhou, Y. Fang, and Y. Zhang, "Securing wireless sensor networks: a survey," *IEEE Communications Surveys and Tutorials*, vol. 10, no. 3, pp. 6–28, 2008.

[2] S. Banihashemian and A. G. Bafghi, "A new key management scheme in heterogeneous wireless sensor networks," in *Proceedings of the 12th International Conference on Advanced Communication Technology (ICACT '10)*, pp. 141–146, February 2010.

[3] F. Anjum, "Location dependent key management in sensor networks without using deployment knowledge," *Wireless Networks*, vol. 16, no. 6, pp. 1587–1600, 2010.

[4] X. Du, Y. Xiao, M. Guizani, and H. H. Chen, "An effective key management scheme for heterogeneous sensor networks," *Ad Hoc Networks*, vol. 5, no. 1, pp. 24–34, 2007.

[5] S. Banihashemian and A. G. Bafghi, "Alternative shared key replacement in heterogeneous wireless sensor networks," in *Proceedings of the 8th Annual Conference on Communication Networks and Services Research (CNSR '10)*, pp. 174–178, May 2010.

[6] M. Luk, G. Mezzour, A. Perrig, and V. Gligor, "MiniSec: a secure sensor network communication architecture," in *Proceedings of the 6th International Symposium on Information Processing in Sensor Networks (IPSN '07)*, pp. 479–488, April 2007.

[7] W. Du, J. Deng, Y. S. Han, S. Chen, and P. K. Varshney, "A key management scheme for wireless sensor networks using deployment knowledge," in *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, pp. 586–597, March 2004.

[8] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *Proceedings of the IEEE Symposium on Security And Privacy*, pp. 197–213, May 2003.

[9] K. Vivek, C. Narottam, and S. Surender, "A survey on clustering algorithms for heterogeneous wireless sensor networks," *International Journal of Advanced Networking and Applications*, vol. 2, no. 4, pp. 745–754, 2011.

[10] K. Ramesh and K. Somasundaram, "A comparative study of clusterhead selection algorithms in wireless sensor networks," *International Journal of Computer Science & Engineering Survey*, vol. 2, no. 4, pp. 153–164, 2011.

[11] A. Ray and D. De, "Energy efficient cluster head selection in wireless sensor network," *Recent Advances in Information Technology*, pp. 306–311, 2012.

[12] S. Zhu, S. Setia, and S. Jajodia, "LEAP+: efficient security mechanisms for large-scale distributed sensor networks," *ACM Transactions on Sensor Networks*, vol. 2, no. 4, pp. 500–528, 2006.

[13] A. Manjeshwar and D. P. Agrawal, "TEEN: a protocol for enhanced efficiency in wireless sensor networks," in *Proceedings of the 15th international workshop on parallel and distributed computing issues in wireless networks and mobile computing*, pp. 2009–2015, San Francisco, Calif, USA, 2001.

[14] J. S. Chen, Z. W. Hong, N. C. Wang, and S. H. Jhuang, "Efficient cluster head selection methods for wireless sensor networks," *Journal of Networks*, vol. 5, no. 8, pp. 964–970, 2010.

[15] H. N. Seyed, H. J. Amir, and D. Vanesa, "A distributed group rekeying scheme for wireless sensor networks," in *Proceedings of The 6th International Conference on Systems and Networks Communications (ICSNC '11)*, pp. 127–135, 2011.

[16] F. R. Kong and C. W. Li, "Dynamic key management scheme for wireless sensor network," *Journal of Software*, vol. 21, no. 7, pp. 1679–1691, 2010 (Chinese).

[17] S. Bandyopadhyay and E. J. Coyle, "An energy efficient hierarchical clustering algorithm for wireless sensor networks," in *Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies (INFOCOM '03)*, pp. 1713–1723, April 2003.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 492819, 8 pages
doi:10.1155/2012/492819

Review Article

Wireless Sensor Network Applications in Smart Grid: Recent Trends and Challenges

Yide Liu

Faculty of Management and Administration, Macau University of Science and Technology, Taipa, Macau

Correspondence should be addressed to Yide Liu, ydliu@must.edu.mo

Received 28 April 2012; Revised 11 July 2012; Accepted 16 July 2012

Academic Editor: An Liu

Smart grid revolutionizes the current electric power infrastructure by integrating with communication and information technologies. With wireless sensor network, smart grid enables both utilities and customers to transfer, monitor, predict, and manage energy usage effectively and costly. However, the increased application of wireless sensor network also introduce new security challenges, especially related to privacy, connectivity, and security management, causing unpredicted expenditure and disaster to both utilities and consumers. In order to build a reliable wireless sensor network for smart grid, an application review and taxonomy of relevant cyber security and privacy issues is presented in this paper. A unified framework for identification of applications and challenge issues of wireless sensor network in smart grid is developed. Future research directions are discussed at the end of this paper.

1. Introduction

Smart grid can provide efficient, reliable, and safe energy automation service with two-way communication and electricity flows. Through wireless sensor network, it can capture and analyze data related to power usage, delivery, and generation efficiently. According to the analysis results, smart grid can provide predictive power information (e.g., meter reading data, monthly charge, and power usage recommendation) to both utilities and consumers. It can also diagnose power disturbances and outages to avoid the effect of equipments failure and natural accidents. Wireless senor network is adopted by utility companies and suppliers for substation automation management, and it is also widely applied in wireless automatic meter reading (WAMR) system. Based on wireless sensor network, energy usage and management information, including the energy usage frequency, phase angle and the values of voltage, can be read real time from remote devices. Therefore, utility companies can manage electricity demand efficiently. They can reduce operational costs by eliminating the need for human readers and provide an automatic pricing system for customers. Customers can enjoy highly reliable, flexible, readily accessible and cost-effective energy services.

However, wireless sensor network also brings cyber security and privacy challenges to smart grid—many security, privacy and reliability issues appear during electric power delivery. For example, cascading-failure-induced disasters might appear if attackers disrupt the grid at a later date from a remote location; smart grid customers' privacy information might be accessed illegally through wireless sensing network; the adversary might also compromise selected nodes in a tactical delay-tolerant network and thus fail the critical mission of the supervisory control and data acquisition (SCADA) systems [1, 2]. Any of these forms of attack can be highly dangerous to the grid—millions of homes might be left without electric power and businesses could be closed. Besides, power grids are a major resource to the national defense. Therefore, a secure wireless ad hoc and sensor network communication with high capacity must be addressed to ensure a reliable and efficient smart grid.

However, some of current guidelines for electric power system were designed for connectivity, without consideration of wireless risks [3], and some of electric power system security standards do not cover threats through wireless sensor network communication. It may lead to an unsatisfied result to simply transplant wireless sensor network security techniques into the smart grid. An understanding of system

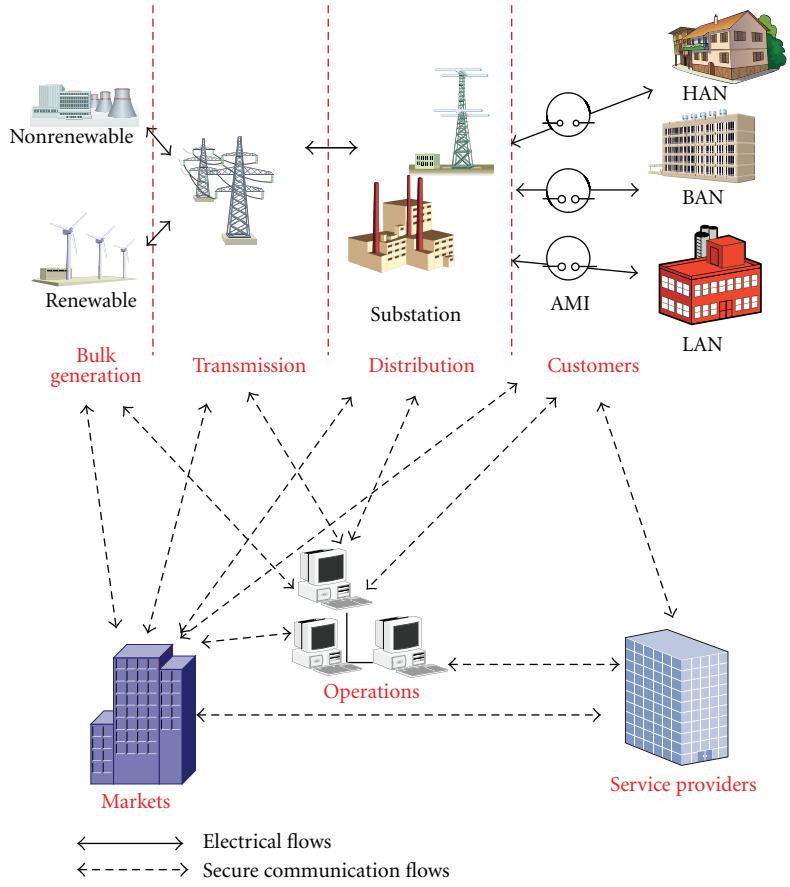


Figure 1: NIST reference model for the smart grid [5].

components with wireless sensor network and associated cyber vulnerabilities is therefore necessary for the smart grid deployments and is the motivation of this paper.

The remainder of this paper is organized as follows. Section two reviews the application of wireless sensor network in smart grid, including WirelessHART, International Society of Automation (ISA) 100.11a, and ZigBee. In Sections three and four, related cyber security and privacy issues in the smart grid are discussed and classified. Section five provides several potential research fields.

2. Wireless Sensor Network Applications in Smart Grid

For distributing energy power from power plants to end customers, smart grid contains three major processes: power generation, power delivery, and power utilization, wherein seven specific domains are going on: power plant domain, substation domain, distribution domain, market domain, operation domain, service provider domain, and customer domain (as show in Figure 1). Recently, WSN has been widely recognized as a vital component of the electric power system, different from wireless ad hoc networks, wireless sensor network contains a large number of low cost, low power, and multifunctional sensor nodes which can be of benefit to electric system automation applications, especially in urban areas [4]. These sensor nodes take advantage of demographic, action, communication, situation, or other data (physical environment, location data, distance, temperature, sound, air pressure, time, lighting levels, people nearby, customer preferences and even customer emotional state, etc.). They can also map the physical characteristics of the environment to quantitative measurements [4].

The collaborative and context-awareness nature of WSN brings several advantages over traditional sensing including greater fault tolerance, improved accuracy, larger coverage area, and extraction of localized features. Sensor nodes can monitor the overall network and to communicate with the control center in the power utility (e.g., a substation), in order to help operators decide the appropriate actions. The sensor node can communicate with the task manager via Internet or satellite. As shown in Figure 2, for developing a wireless sensor network for smart grid, there are three alternatives based on the IEEE 802.15.4 protocol: ZigBee, WirelessHART, and ISA100.11a. For example, ZigBee is a choice for smart grid system networking within home. WirelessHART or ISA100.11a can be used in substation or a generation plant. In this section, the wireless sensor network

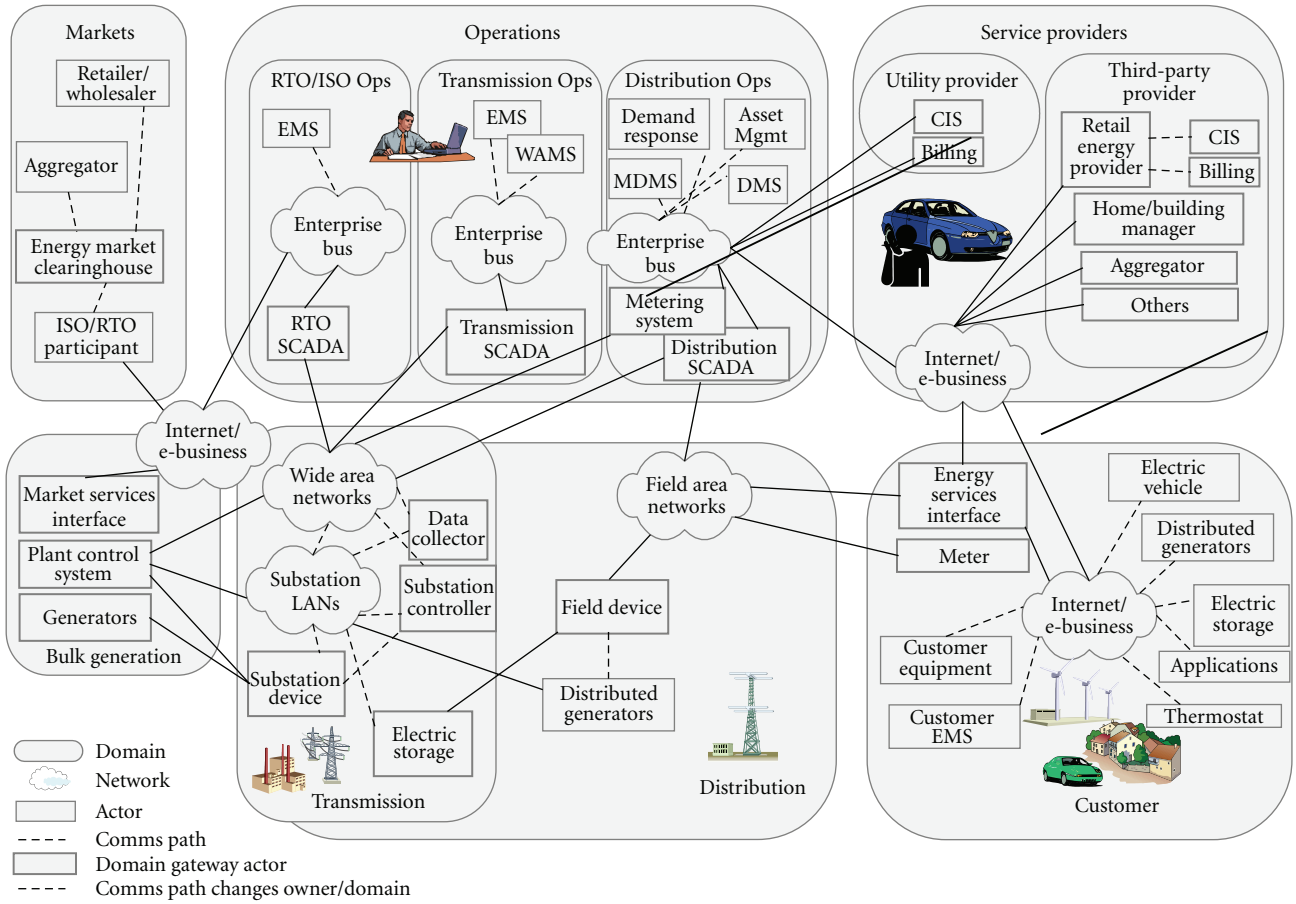


Figure 2: NIST Smart Grid Framework 1.0, September 2009 [6].

application for smart grid will be discussed separately in the context of power generation, power delivery, and power utilization.

2.1. Power Utilization. Wireless sensor network can be used in home area networks (HANs). As mentioned, ZigBee is a suitable choice for HANs. It provides the reliable wide-area coverage and predictable latencies that are expected for smart grid. A typical application of WSN for smart grid is wireless automatic meter reading (WAMR) systems, which can determine real-time energy consumption of the customers as customers can download their archives and take it to meter reading through a mobile device. WAMR can also improve business performance and technical reliability for power utility operations, as utility companies can identify more valuable customers by comparing the data between the distributed generation sources and overall power consumption [4]. WAMR system can remotely control light, heat, air conditioning, and other appliances of different customers.

Smart grid system needs to provide benefits to both the customer and the utility, and the smart meter within HANs perform as an interface that translates, summarizes, and aggregates data of power usage and presents it to the power utility [9]. Inside home, a wireless sensor network can link the various equipments and a central power router as shown in Figure 3. This network could connect to the utility network via a smart meter. The smart meter serves as an interface for a variety of operational signals so that both metering and operational data are carried on the wireless sensor network. For example, a utility can implement the "demand response" function within a home through a price-based incentive signal, and the smart meter infrastructure carry the signal [9]. Specifically, in areas of high population

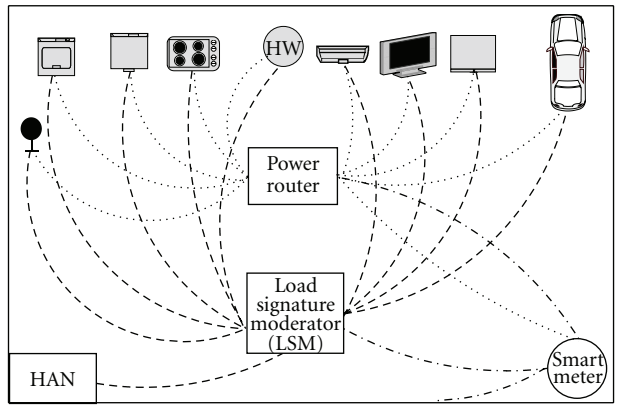


Figure 3: Home area networks (HANs) example [7].

density, the smart meter must be able to different smart grid nodes (SGNs) that belong to each customer by collecting usage information through SGNs [9]. The smart meter may also assert an incentive signal to cause the SGNs to switch to a power-saving profile when the amount of information exchanged is not large [9].

2.2. Power Delivery. Wireless sensor networks can also be used in electric power system operations and substation automation. For example, sensors could be installed to monitor the delivery systems and power use in the system, and sensors can be further classified according to their location. Substations could also be monitored as circuit currents, power usage and station apparatus are checked here [9]. WSNs can also provide a feasible and cost-effective sensing and communication solution for remote system monitoring systems. The conditions of different smart grid operation process, (e.g., generation units, transformers, transmission lines, and motors), can be monitored by the large-scale deployment of smart sensor nodes in a remote, and these nodes can be installed on the critical equipment of smart grid. Therefore, a single system contingency in the power grid can be detected and isolated before it causes cascading effects [10]. Besides, measuring voltages and currents associated with transformers, circuit breakers, and switches in a substation or a distribution station, power quality sensors, transformer temperature sensors, and breaker position indicators may also be monitored [9].

2.3. Power Generation. A bulk generation plant may contain several generation units, and several hundred actuators may control fuel, air, and water flows to optimize heat rate (efficiency of the generator) control emissions, and adjust generator output within each unit [9]. Wireless sensors could be installed to monitor the generation systems in power plants, and WirelessHART or ISA100.11a could be used to deploy sensors here.

Sensors that use IEEE 802.15.4-based radio transceivers can function for several years in harsh environments without requiring any external power (e.g., WirelessHART can route around not only single but also multiple node failures) [9]. Besides, sensors can be easily relocated and supplementary sensors can be deployed within a few hours. Therefore, each generation unit may measure parameters such as steam temperature and air, water, or fuel flow rates based on sensors. This information is fed into the data acquisition system in the power plant [9].

3. Challenges of Wireless Sensor Network in Smart Grid

Although the wireless sensor networks have been facilitating different smart grid operation processes, the characteristics of different WSNs applications are vastly different in features, data rate, and related standards. Therefore, different challenges might appear in different application contexts, which increase the risk of smart grid operation and maintenance.

Common challenges associated with wireless sensor networks are probabilistic channel behavior, accidental and directed interference or jamming, and eavesdropping or unauthorized modification of the communications if not protected by authentication and encryption [9]. Customers' metering information must also be secure. In this section, we detail challenges found in the research literature and map them onto the CERT taxonomy [8].

CERT taxonomy provides a useful framework and uniform terminology to security researchers (see Figure 4).

3.1. Security Requirements. Secrecy, integrity, and availability are three fundamental security requirements, and previous research has provided several basic goals for establishing secure smart grid over the wireless sensor network [1–3, 11–14].

3.1.1. Secrecy. The target of secrecy is to prevent passive attacks and unauthorized access to sensitive data, that is, power usage and billing information. In a wireless sensor network, the issue of confidentiality should address the following requirements [15–17]: (i) a sensor node should not allow its neighbors to read its readings unless they are authorized, (ii) key distribution mechanism should be robust, and (iii) public information (e.g., sensor identities and public keys of the nodes) should be encrypted to protect against traffic analysis attacks. Early detection method could be used for preventing unwarranted communication delays, any manipulation of information must be detected as early as possible. Early detection can also eliminate or reduce false alarms. Besides, privacy is also a critical issue and can be attacked easily, especially in context such as submitting service request for emergency and checking energy usage from smart meters. However, it is not easy to describe the scope of privacy issues for smart grid, as privacy problems can exit not only in personal communications, but also in business transaction among power plant, substations and customers. Unfortunately, there has not been a well-established standard for smart grid privacy issues. Standard-based privacy protection schemes could be a solution. For example, EG2 made a suggestion to separate the smart metering data into low-frequency attributable data (e.g., data used for billing) and high-frequency anonymous technical data (e.g., data used for demand side management) aiming to protect privacy [18].

3.1.2. Integrity. The target of integrity is to ensure that the transmitted data is not illegally modified (e.g., changing, deleting, creating, delaying, or replaying data) from the sender to the recipient, and the identity and content of the received data must be verified to be the same as the original source. An authentication method could be developed for ensuring that the origin and destination of information is correctly identified, the injection of corrupted data by unauthorized entities must be prevented.

3.1.3. Availability. The target of availability is to ensure the wireless sensor network services to be available to authorized users on time, even in presence of an internal or external attack (e.g., denial of service attack). To reach this target, both additional communication among nodes and a central

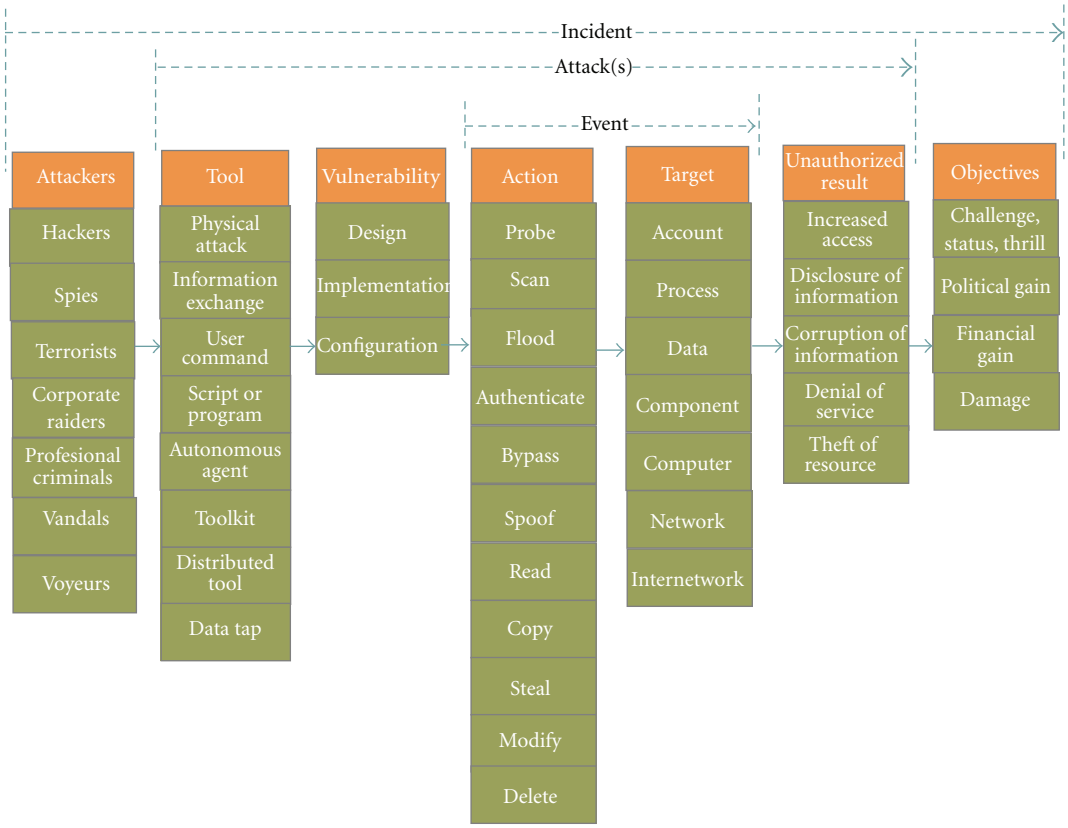


Figure 4: Attack taxonomy by CERT Coordination Center [8].

access control system may be adopted for successful delivery of every message to its recipient [15, 19]. A solution is to make sure all actions performed on any information must be logged for a time period.

3.2. Attacks Taxonomy. CERT taxonomy focuses on incidents, and an incident within CERT taxonomy means that an attacker executes one or more attacks to achieve specific objectives. Additionally, based on the target each incident, different tools are used to exploit vulnerabilities to produce an unauthorized result. Table 1 listed the main attackers and objectives.

Table 1: Attackers and objectives of wireless sensor network in smart grid.

Attacker	Objective
Professional criminals	Damage or steal smart grid devices like smart meters and home appliances
Terrorists	Cause harm
Vandals	Crack
Hackers	Crack
Voyeurs	Gain access to related devices and related data

3.2.1. Device Issues. Devices related with wireless sensor network include smart meter and AMI devices. These devices bring significant advantages for users and create challenge issues at the same time because date and signals transmitted by these devices contain the information about presence of people at their residence and what appliances are in use. Depuru et al. listed certain sections of people who might be interested in collecting and analyzing the data transmitted through wireless network, including revengeful exspouses, civil litigant, illegal consumers of energy, extortionists, terrorists, political leaders with vested interests, thieves, and so forth [4]. For example, professional criminals may damage smart grid devices and steal costly device components for financial gain. Therefore, the location of smart meters should not be easy to touch. Hackers may gain access to related devices and related data (e.g., metering database, meters battery change, removal, and modification information) for challenging themselves [20, 21]. Voyeurs may remote connect/disconnect meters and outage reporting [20, 22]. Therefore, it needs high security to protect customer information and devices. Possible solutions include ensuring the integrity of meter data, detecting unauthorized changes on meter, and authorizing all accesses to/from AMI networks [23]. In fact, challenges are not only from deliberate attacks, but also include other possible human errors and system vulnerabilities, such as weak smart grid user authentication

Table 2: Attacks of wireless sensor network in smart grid.

Target	Vulnerability	Actions	Unauthorized result	Tool
Device	Design/implementation	Steal, destroy, or replace devices	Meter storage tampering	Thieves tools
Sensor data	Inadequate physical tamper protections	Jamming attacks	Service or data lost	Jamming
Communication ability	Network failures/crypto or protocol issues	Jamming attacks/spoof/scan	Communication interception	Script or program
Account/data	No firmware integrity protections	Spoof/scan/read	Password extraction	Malicious software

control, weak communication protocol, and improper communication management.

3.2.2. Networking Issues. Routing information in wireless sensor networks can be changed, and this challenge can result in unauthorized control of the communication network. For example, an intruder can take over vulnerable equipments and mislead the data presented to smart grid operators.

Jamming attacks could be seen as the most well-known attacks that compromise availability of wireless sensor networks. The possibility of jamming may appear with any radio-based medium, and the sensor nodes may be deployed in hostile or insecure environments where an attacker has the physical access. Jamming is a type of attack which interferes with the radio frequencies that the sensor nodes use for communication [15, 19, 24]. A jamming source may be powerful enough to disrupt the entire network. Even an intermittent jamming may cause negative effect as the message communication in a WSN may be extremely time-sensitive [15, 25]. Besides, the integration of other communication systems might result in arduous challenges of protecting smart grid, especially when integrating smart gird with existing public network [3]. AES (advanced encryption standard) encryption [26, 27] could be a possible solution for protecting sensor network.

WSNs' vulnerabilities include design and implementation of wireless sensor networks for smart grid. The design and implementation of WSNs are constrained by three types of resources: (i) energy, (ii) memory, and (iii) processing [23]. During different communication processes, the lack of sensor battery may lead to the failure of smart grid. Sensor nodes have limited battery energy supply [28], but in smart grid, the batteries of the sensors can be charged by the energy supplies [23]. The collaborative effort of sensor nodes can handle the problems of limited memory and processing capabilities of the sensor nodes [23]. Table 2 described the wireless sensor networks attacks.

3.2.3. Other Technical Challenges. Other technical challenges for wireless sensor network in smart grid include harsh environmental conditions, reliability and latency requirements, and packet errors and variable link capacity [10]. In smart grid environment, sensors may also be subject to RF interference, highly caustic or corrosive environments, high humidity levels, vibrations, dirt and dust, or other conditions; furthermore, the topology and wireless connectivity of the network may vary [10]. The harsh environmental conditions may disturb a portion of sensor nodes in information delivery process.

When wireless sensor communicating across power utilities and customers, the power plants are in charge of exchanging data (e.g., peer transmission and distribution system operation) or regional transmission organization (e.g., substations, end users, or other power plants), and substations are in charge of exchanging important information (e.g., protection data among substations) and alarms. In short, power plants provide operation services such as switching operation, changing setups, recommendation of optimized operations, starting emergency procedure and performing system restorations [3], and substations always take the responsibility of power system protection, load shedding, recovery from load shedding, shunt control and compensation control [3]. Therefore, the wide variety of applications of WSNs in smart grid will have different requirements on quality-of-service (QoS), reliability, latency, network throughput, and so forth [10]. In addition, sensor data are typically time sensitive [10].

In WSNs, the bandwidth of each wireless link depends on the interference level of the receiver, and high bit error rates (BER $= 10^{-2} - 10^{-6}$) are required in communication [10]. Deliberate attacks which can overwhelm the forwarding capability of nodes, and they can also consume sparsely available bandwidth. These challenges can result in a denial of service to advanced metering infrastructure (AMI) applications based on WSNs. In addition, wireless links perform varying characteristics over time and space due to obstructions and harsh environment in smart grid. Therefore, it may be difficult for wireless links to meet QoS requirements due to the bandwidth and communication latency at each wireless link are location-dependent and can vary continuously [10]. Figure 5 is a modified version of CERT taxonomy based on what we discussed, and it can be seen as a unified framework for identification challenge issues of wireless sensor network in smart grid.

4. Conclusion

The number of applications of smart grid over wireless sensor networks has been steadily increasing, such as wireless

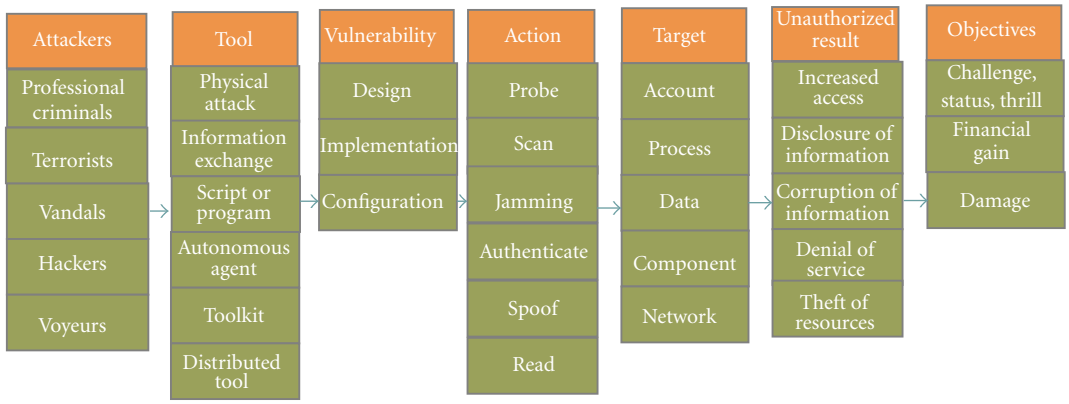


Figure 5: Modified attack taxonomy.

automatic meter reading (WAMR) and remote monitoring systems. However, since radio waves in wireless communication spread in the air, one common risk is that wireless channels are more insecure and susceptible to numerous attacks than wired networks [1]. Much existing work has attempted to incorporate security into smart grid.

To better understand securing service for smart grid over wireless networks, we have presented known attacks that can disrupt wireless sensor network in smart grid communication based on CERT taxonomy. We modified the taxonomy in Figure 5 based on the security analysis in Section 3. We have discussed the recent trends of wireless sensor networks and illustrated basic security requirements to safeguard smart grid against these attacks. We have also reported several existing solutions to wireless sensor network security in smart grid.

It is important to note that there is no single implementation that will define the communications architecture of smart grid. Although we realized security issues, the solutions may also require management effort with policy. For example, a power plant could dine security policies and procedures for maintaining and controlling collaboration with both substations and market, and the next generation of smart metering technology might depend on the policies of utility companies and respective governments [18]. It is misleading to suggest that IT people should take the full responsibility for wireless smart grid network security. However today, there are little common rules or standards for the data exchange or resources usage in the wireless smart grid communication. We are studying this challenge in a case study in related companies.

Acknowledgment

This work was supported by Faculty Research Grant of Macau University of Science and Technology (Project 0236 name: Research on Key Technologies of Context-Aware Computing Based on Mobile Social Network and System Design).

References

[1] A. R. Metke and R. L. Ekl, "Security technology for smart grid networks," *IEEE Transactions on Smart Grid*, vol. 1, no. 1, pp. 99–107, 2010.

[2] P. Sakarindr and N. Ansari, "Security services in group communications over wireless infrastructure mobile ad hoc, and wireless sensor networks," *IEEE Wireless Communications*, vol. 14, no. 5, pp. 8–20, 2007.

[3] D. Wei, Y. Lu, M. Jafari, P.M. Skare, and K. Rohde, "Protecting smart grid automation systems against cyberattacks," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 782–795, 2011.

[4] S. S. S. R. Depuru, L. Wang, and V. Devabhaktuni, "Smart meters for power grid: challenges, issues, advantages and status," *Renewable and Sustainable Energy Reviews*, vol. 15, no. 6, pp. 2736–2742, 2011.

[5] U.S. NIST, "NIST framework and roadmap for smart grid interoperability standards, release 1. 0," NIST Special Publication 1108, 2010, http://www.nist.gov/public_affairs/releases/upload/smartgrid_interoperability_final.pdf.

[6] A draft version of this publication by NIST, http://www.nist.gov/public_affairs/releases/upload/smartgrid_092409_fr.pdf.

[7] G. Kalogridis, C. Efthymiou, S.Z. Denic, T. A. Lewis, and R. Cepeda, "Privacy for smart meters: towards undetectable appliance load signatures," in *Proceedings of the 1st IEEE International Conference on Smart Grid Communications*, pp. 232–237, Gaithersburg, Md, USA, October 2010.

[8] J. D. Howard and T. A. Longstaff, "A common language for computer security incidents," Sandia National Laboratories, Sandia Representative, SAND98-8867, 1998.

[9] B. A. Akyol, H. Kirkham, S. L. Clements, and M. D. Hadley, "A survey of wireless communications for the electric power system," U.S. Department of Energy, 2010.

[10] V. C. Gungor, B. Lu, and G. P. Hancke, "Opportunities and challenges of wireless sensor networks in smart grid," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 10, pp. 3557–3564, 2010.

[11] V. C. Gungor and F. C. Lambert, "A survey on communication networks for electric system automation," *Computer Networks*, vol. 50, no. 7, pp. 877–897, 2006.

[12] G. N. Ericsson, "Cyber security and power system communication—essential parts of a smart grid infrastructure," *IEEE*

Transactions on Power Delivery, vol. 25, no. 3, pp. 1501–1507, 2010.

[13] W. Lou and K. Ren, "Security, privacy, and accountability in wireless access networks," *IEEE Wireless Communications*, vol. 16, no. 4, pp. 80–87, 2009.

[14] H. S. Yang, H. S. Jang, Y. W. Kim et al., "Communication networks for interoperability and reliable service in substation automation system," in *Proceedings of the 5th ACIS International Conference on Software Engineering Research, Management, and Applications (SERA'07)*, pp. 160–165, Busan, Korea, August 2007.

[15] J. Sen, "A survey on wireless sensor network security," *International Journal of Communication Networks and Information Security*, vol. 1, no. 2, pp. 59–82, 2009.

[16] D. W. Carman, P. S. Krus, and B. J. Matt, "Constraints and approaches for distributed sensor network security," Tech. Rep. 00-010, NAI Labs, Network Associates Inc., Glenwood, Md, USA, 2000.

[17] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "SPINS: security protocols for sensor networks," *Wireless Networks*, vol. 8, no. 5, pp. 521–534, 2002.

[18] F. Zhong, S. Gormus, C. Efthymiou et al., "Smart grid communications: overview of research challenges, solutions, and standardization activities," *IEEE Communications Surveys and Tutorials*, vol. PP, no. 99, pp. 1–18, 2012.

[19] Z. Lu, W. Wang, and C. Wang, "From jammer to gambler: modeling and detection of jamming attacks against time-critical traffic," in *Proceedings of the IEEE INFOCOM 2011*, pp. 1871–1879, Shanghai, China, April 2011.

[20] U.S. NIST, "Guidelines for smart grid cyber security (vol. 1 to 3)," NIST IR-7628, 2010, http://csrc.nist.gov/.

[21] D. Wei, Y. Lu, M. Jafari, P. Skare, and K. Rohde, "An integrated security system of protecting smart grid against cyber attacks," in *Proceedings of the Innovative Smart Grid Technologies Conference (ISGT'10)*, pp. 1–7, Gaithersburg, Md, USA, January 2010.

[22] R. Anderson and S. Fuloria, "Who controls the off switch?" in *Proceedings of the 1st IEEE International Conference on Smart Grid Communications (SmartGridComm'10)*, pp. 96–101, Gaithersburg, Md, USA, 2010.

[23] Y. Xiao, Y. Xiao, S. Li, W. Liang, and C. Chen, "Cyber security and privacy issues in smart grids," *IEEE Communications Surveys and Tutorials*, vol. PP, no. 99, pp. 1–17, 2012.

[24] Q. Zeng, H. Li, and P. Dai, "Frequency hopping based wireless metering in smart gird: code design and performance analysis," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM'11)*, pp. 1–5, Houston, Tex, USA, December 2011.

[25] A. D. Wood and J. A. Stankovic, "Denial of service in sensor networks," *Computer*, vol. 35, no. 10, pp. 54–62, 2002.

[26] P. Zhang, O. Elkeelany, and L. McDaniel, "An implementation of secured Smart Grid Ethernet communications using AES," in *Proceedings of the IEEE SoutheastCon 2010 Conference: Energizing Our Future*, pp. 394–397, Concord, NC, USA, March 2010.

[27] A. Bartoli, J. Hernández Serrano, M. Soriano et al., "Secure lossless aggregation for smart grid M2M networks," in *Proceedings of the 1st IEEE International Conference on Smart Grid Communications (SmartGridComm'10)*, pp. 333–338, Gaithersburg, MD, USA, October 2010.

[28] Idaho National Laboratory, "Common cyber security vulnerablilities observed in control system assessments by the INL NSTB program," Idaho National Laboratory Technical Report INL/EXT-08-13979, Idaho National Laboratory, 2008, http://www.inl.gov/scada/publications.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 891584, 8 pages
doi:10.1155/2012/891584

Research Article

Prevention and Detection Methods for Enhancing Security in an RFID System

Jing Huey Khor, Widad Ismail, and Mohammad Ghulam Rahman

Auto-ID Laboratory, School of Electrical and Electronic Engineering, Universiti Sains Malaysia (USM), 14300 Nibong Tebal, Penang, Malaysia

Correspondence should be addressed to Widad Ismail, eewidad@eng.usm.my

Received 27 April 2012; Accepted 29 June 2012

Academic Editor: An Liu

Low-cost radio frequency identification (RFID) tag is exposed to various security and privacy threats due to computational constraint. This paper proposes the use of both prevention and detection techniques to solve the security and privacy issues. A mutual authentication protocol with integration of tag's unique electronic fingerprint is proposed to enhance the security level in RFID communication. A lightweight cryptographic algorithm that conforms to the EPCglobal Class-1 Generation-2 standard is proposed to prevent replay attack, denial of service, and data leakage issues. The security of the protocol is validated by using formal analysis tool, AVISPA. The received power of tag is used as a unique electronic fingerprint to detect cloning tags. t-test algorithm is used to analyze received power of tag at single-frequency band to distinguish between legitimate and counterfeit tag. False acceptance rate (FAR), false rejection rate (FRR), receiver operating characteristic (ROC) curve, and equal error rate (EER) were implemented to justify the robustness of t-test in detecting counterfeit tags. Received power of tag at single frequency band that was analyzed by using t-test was proved to be able to detect counterfeit tag efficiently as the area under the ROC curve obtained is high (0.922).

1. Introduction

Radio frequency identification (RFID) tags that conform to EPC Class-1 Generation-2 (Gen 2) standards are broadly used in supply chain management, logistic, person identification, and access control. Global RFID market is expected to grow at a compound annual growth rate (CAGR) of roughly 17% to a value of approximately $9.7 billion in the period 2011–2013. However, the privacy and security of the usage of RFID technology are not guaranteed. The issues that raise security concerns are possibility of tag cloning issue, denial of service (DoS) attack, replay attack, and data leakage.

Gen 2 tags are susceptible to cloning attack due to lack of explicit authentication and security functionalities. Complex cryptographic algorithms, including hash function, and symmetric and asymmetric algorithms, are not supported by Gen 2 tags [1–3]. This is because Gen 2 tags have low-computation capabilities that are only able to support simple mathematical functions. Hence, strong adversaries are capable of skimming on transmission channels to obtain tag information [4]. This information may be used to create counterfeit tags that bear the same information as that of a legitimate tag. Counterfeit tags can be attached to bogus products and disguise these as authentic products in the market. The counterfeit tag issue is very serious because it is capable of causing a menace ranging from public privacy and safety issues to loss of industry revenues.

Lightweight cryptographic algorithm (i.e., CRC, PRNG, and XOR functions) can be used to prevent data leakage problem in Gen 2 tag. In addition, received power of tag can be used as tag's unique electronic fingerprint to detect counterfeit tags. Detection techniques are deployed to minimize the negative effects of tag cloning threats [5]. Counterfeit tags can be detected by employing the electronic fingerprinting system in an RFID system since each RFID tag is unique, based on their radio frequencies and manufacturing differences. Received power of tag at

single frequency band is analysed by using t-test to distinguish between legitimate and counterfeit tag. Hence, the combination of prevention and detection methods could be the countermeasure to the privacy and security issues being faced by Gen 2 tags.

The remaining of this paper is structured as follows: Section 2 describes the related works and Section 3 illustrates the overview of proposed lightweight cryptographic mutual authentication protocol. Section 4 outlines the experiment setup and data collection for fingerprint-matching method. Section 5 explains the t-test algorithm in details and Section 6 analyzes the accuracy and performance of fingerprint-matching method. Section 7 shows the overall security analysis and Section 8 concludes the paper.

Table 1: Notations used in the protocol.

Notation	Interpretation
E_T	Tag's electronic product code
Rn	Random number
CRC	Cyclic redundancy code
PRNG	Pseudorandom number generator
K_i	Current session key
K_{i+1}	New session key
K_t	Tag's temporary key
K_s	Server's temporary key
$\oplus$	XOR function
$\|\|$	Concatenation

2. Related Works

In Chien and Chen [2], PRNG, CRC, and XOR are used as the fundamentals in the protocol. Two sets of authentication and access keys are designed to defend DoS attack. However, the scheme is vulnerable to replay attack and information leakage. Chien and Huang [6] presented a lightweight mutual authentication protocol to solve replay attack and secret disclosure problem of Li et al. [7] scheme. But cloning attack problem is not resolved in this scheme. Song and Mitchell [8] proposed an authentication protocol that uses challenge-response approach and simple functions such as right and left shifts and bitwise XOR operation in the scheme. However, the scheme is vulnerable to tag impersonation attack and server impersonation attack. Song [9] presented an authentication protocol for tag ownership transfer that meets new owner privacy, old owner privacy, and authorization recovery requirements. However, the ownership transfer protocol is vulnerable to a desynchronization attack that prevents a legitimate reader from authenticating a legitimate tag, and vice versa. Burmester and Munilla [10] proposed a lightweight mutual authentication protocol that supports session unlinkability, forward and backward secrecy. The protocol is optimistic with constant key lookup, and can easily be implemented on a Gen 2 platform. However, the scheme is susceptible to replay and cloning attacks. Chen and Deng [11] proposed mutual authentication protocol that is able to reduce database loading and ensure user privacy. But the authentication protocol did not take into consideration cloning attack issues.

In [12], minimum power responses measured at multiple frequencies are used as unique electronic fingerprint. The power is measured at the range from 860 MHz to 960 MHz in increments of 1 MHz. Two-way analysis of variance (two-way ANOVA) is used to test the equality of means of two groups in terms of minimum power response and different physical characteristic of tags. 10-fold cross-validation on the classifier is used to validate the result obtained, and the AUC is 0.999. The average true positive rate and false positive rate are 0.905 and 0.001, respectively. The research focused on using minimum power responses at multiple frequencies as a unique electronic fingerprint for RFID tags. Hence, this paper extends the idea to show that received power of tag at single frequency band can be used to fingerprint RFID tags. Physical-layer identification of passive UHF RFID tags from three different manufacturers is analyzed in [13]. RFID reader that is capable to simulate an inventory protocol is built to activate tags. RF signal features are extracted from the preambles of tags' replies. Time domain and spectral features of the collected signals are analyzed. The tags can be classified with an accuracy of 71.4% from different locations and distances to the reader based on the time domain features. In addition, UHF RFID tag that is proved can be uniquely identified in controlled environment based on the signal spectral features with 0% of EER. The physical-layer identification method is complex, and the reader used in conducting the experiment is purposely built. In contrast, the proposed method in this paper is simple and applicable to any Gen 2 reader.

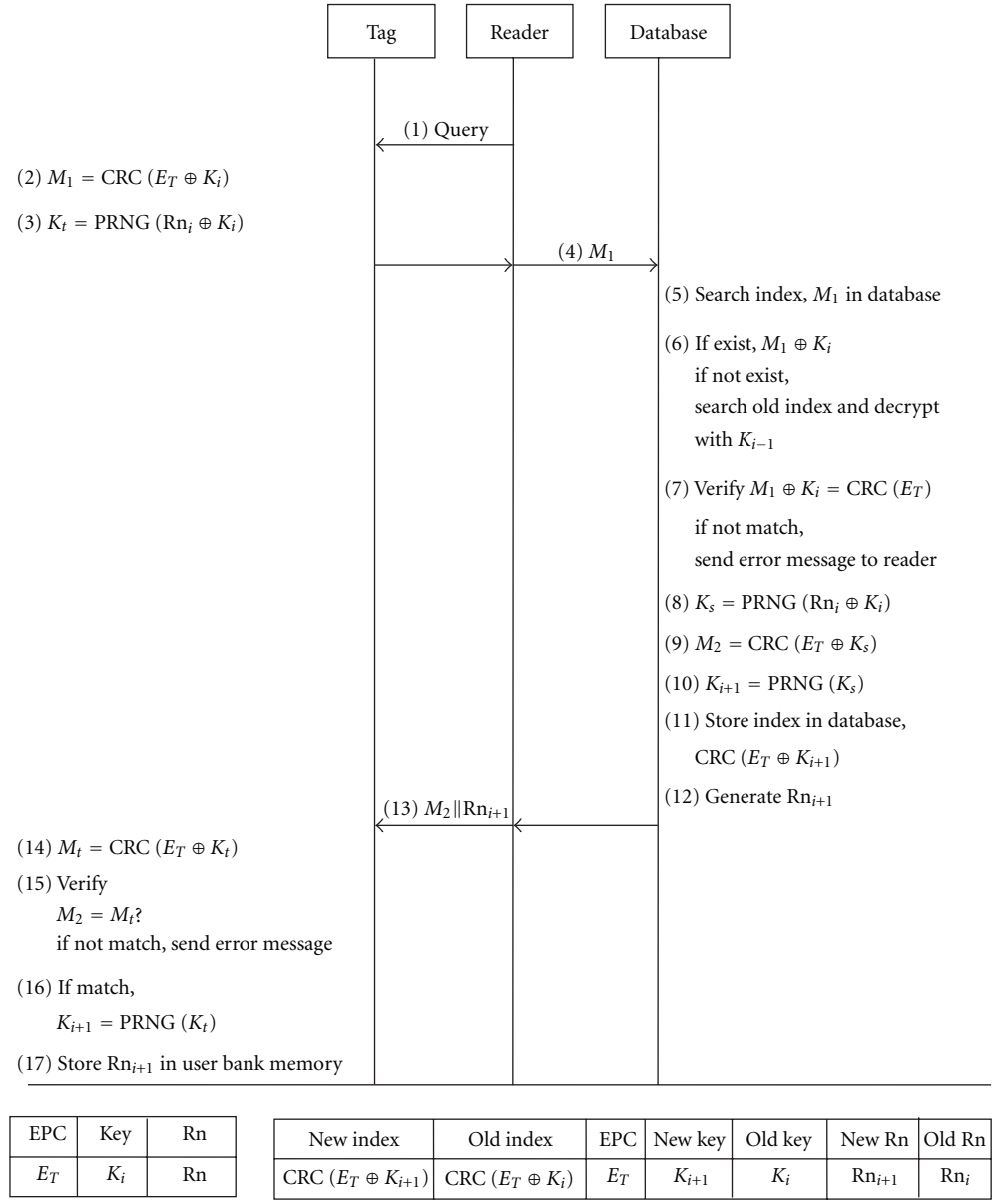
3. Lightweight Cryptographic Mutual Authentication Protocol

A lightweight cryptographic mutual authentication protocol that conforms to Gen 2 standards is proposed. The proposed protocol consists of initialization phase and authentication phase. The channel between a back-end server and a reader is assumed secure. On the other hand, the channel between a reader and a tag is assumed insecure.

The notations used in the description of proposed protocol are shown in Table 1.

In the initialization phase, a back-end server and tag store information are required to perform authentication. The back-end server initially stores seven values of each tag in its database. These are new index denotes as CRC ($E_T \oplus K_{i+1}$), old index denotes as CRC ($E_T \oplus K_i$), tag's electronic product code denotes as E_T, new session key denotes as K_{i+1}, old session key denotes as K_i, new random number denotes as Rn_{i+1}, and old random number denotes as Rn_i. On the other hand, three values that are stored in the tag are E_T, K_i, and Rn_i. Session key of current session is denoted as K_i, and the session key after a successful session is denoted as K_{i+1}. The tag's temporary key is denoted as K_t, and server's temporary key is denoted as K_s. The overall protocol scheme is shown in Figure 1.

In authentication phase, the reader will send query command to the tag. The tag computes $M_1 = \text{CRC}(E_T \oplus K_i)$.



EPC	Key	Rn
E_T	K_i	Rn

New index	Old index	EPC	New key	Old key	New Rn	Old Rn
CRC ($E_T \oplus K_{i+1}$)	CRC ($E_T \oplus K_i$)	E_T	K_{i+1}	K_i	Rn_{i+1}	Rn_i

Figure 1: Lightweight cryptographic mutual authentication protocol.

At the same time, PRNG generates tag's temporary key, K_t, based on the seed number, $\mathrm{Rn}_i \oplus K_i$. The encrypted message, M_1, is sent via the reader to the back-end server. The back-end server searches for an index, CRC ($E_T \oplus K_i$), in its database that is matching with the encrypted message. If matching index is found, the encrypted message is decrypted using the session key, K_i, that is in the same row as indicated by index. Otherwise, the server searches the matching of M_1 with the old index, CRC ($E_T \oplus K_{i-1}$). If the matching of old index is found, old session key, K_{i-1}, is used to decrypt the message. The authentication of the message is then verified. If the decrypted message does not match the message recorded in the database for both new and old indexes, an error message will be sent to the reader. On the other hand, if the server successfully authenticates the tag, a server's temporary key, K_s, is generated. If the M_1 is decrypted with old index, then K_s is generated by XOR K_{i-1} and Rn_{i-1} as a seed. Then, the back-end server computes M_2 = CRC ($E_T \oplus K_s$). A new session key, K_{i+1}, is generated, and CRC ($E_T \oplus K_{i+1}$) is computed and updated as a new index in the database. In addition, a new random number, Rn_{i+1}, is generated and concatenates with M_2. The new session key and random number are stored in the row that indicated by the new index. Afterwards, the back-end server forwards $M_2 \| \mathrm{Rn}_{i+1}$ to the tag through the reader. The tag computes M_t = CRC ($E_T \oplus K_t$), and the authentication of the reader is verified by the tag where a comparison of M_2 and M_t is made. If both messages are matched, then the tag will update a new session key, K_{i+1}, where K_{i+1} = PRNG (K_t). Otherwise, the key will be maintained as current session key, K_i. The tag stores the

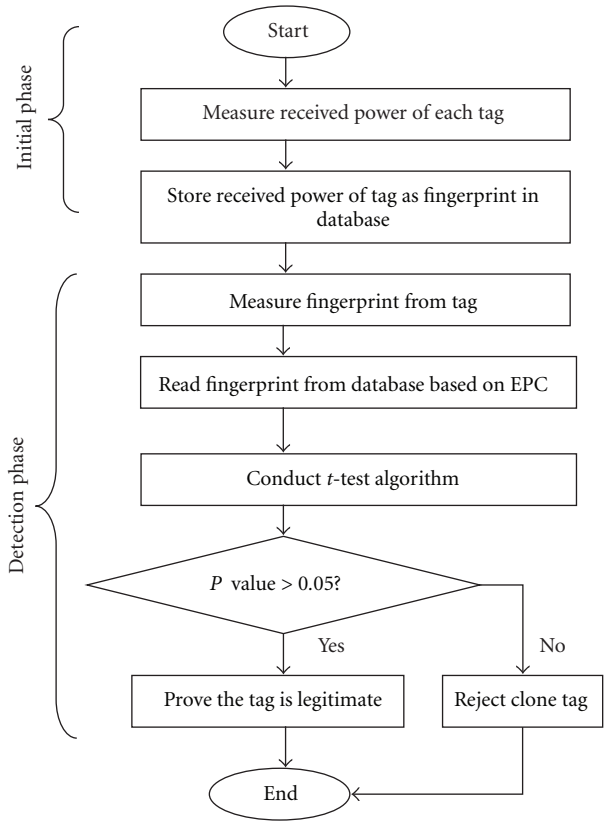


Figure 2: Overall process of fingerprint-matching method.

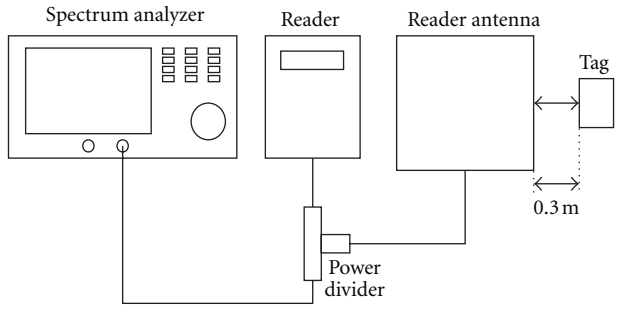


Figure 3: Measurement of received power of tag platform.

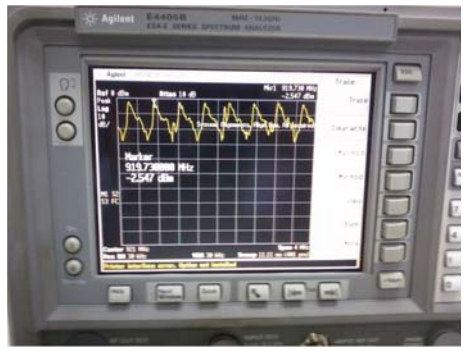


Figure 4: Reader transmitted power measured with spectrum analyzer.

received Rn_{i+1} in the user memory bank for the usage in the next session.

4. Experimental Setup and Fingerprint Data Collection

The proposed RFID tag fingerprint-matching method illustrated in Figure 2 consists of initial phase and detection phase. In the initial phase, received power of each EPC tag is calculated using Friis transmission equation. Reader transmitted power used in the equation is measured using a spectrum analyzer. The received power is measured once the power is held constant. Each tag received power is stored in database. In the detection phase, stored fingerprint and measured fingerprint are compared using t-test algorithm. The tag being measured is proved to be a legitimate tag if P value of t-test algorithm is greater than 0.05. Otherwise, the tag is proved to be a counterfeit tag.

The received power of tag is calculated based on the reader's transmitted power, which is measured at 919–923 MHz. The frequency band is used based on the Malaysian UHF RFID standard governed by Malaysian Communications and Multimedia Commission (MCMC) [14]. However, the measurement is still applicable to other countries, RFID frequency band. The transmitted power of tag is measured for 100 passive RFID tags at fixed temperature and controlled environment. The legitimate tag fingerprint template is determined by obtaining the average received power of 50 readings per tag. The received power that acts as a unique fingerprint for each tag is measured in dBm. The received power is stored in the database only in order to protect the secrecy of fingerprint value from being obtained by adversaries. The unique fingerprint value that stored in the database can be searched based on the EPC. Hence, the stored fingerprint value in database and measured fingerprint value that obtained from experimental measurement can be compared to verify the genuineness of the tag.

Figure 3 shows the measurement of reader transmitted power platform. The setup consists of a passive RFID reader and antenna, passive EPC tag, and spectrum analyzer. The reader operates at UHF 919–923 MHz and supports Gen 2 protocol. The antenna and tag are placed at fixed position to obtain an accurate and reliable result. To determine precise reader transmitted power, cable loss and power loss within the power splitter must be considered. Hence, power value obtained from the spectrum analyzer is added to the total power loss measured to obtain an accurate reader transmitted power. Figure 4 shows a measurement of reader transmitted power using spectrum analyzer.

The tag received power is calculated using Friis transmission equation, as demonstrated in

$$P_r = P_t G_t G_r \left(\frac{\lambda}{4\pi r}\right)^2, \quad (1)$$

where P_r is the power received by the tag antenna and P_t is the power input to the reader antenna. In addition, G_t is the

Table 2: Notations used in the Protocol.

Parameters	Value
Gain of reader antenna	6 dBi
Gain of tag antenna	2.15 dBi
Gain of reader antenna in power ratio, G_t	3.981
Gain of tag antenna in power ratio, G_r	1.641
Frequency, f	919.73 MHz
Wavelength, λ	0.33 m
Distance between reader and tag antennas, R	0.3 m

Table 3: t-test for Tag A and suspicious tag.

	Tag A	Suspicious tag
Mean	0.167092	0.316192
Variance	0.0492	0.07446
Observations	50	5
Pooled Variance	0.05117	
Hypothesized Mean Difference	0	
df	53	
t Stat	−1.40613	
$P(T \le t)$ one tail	0.082761	
t Critical one-tail	1.674116	
$P(T \le t)$ two-tail	**0.165522**	
t Critical two-tail	2.005746	

antenna gain of the reader antenna, G_r is the antenna gain of the tag antenna, λ is the wavelength, and R is the distance between reader and tag antennas. Friis transmission equation is only applicable in Fraunhofer region. Hence, a minimum Fraunhofer region is determined by using

$$r_{ff} = \frac{2D^2}{\lambda}, \tag{2}$$

where, r_{ff} is the minimum far field distance, D is the diameter of transmitting antenna, and λ is the wavelength. The diameter of transmitting antenna is 0.185 m, and the wavelength is 0.33 m because the frequency chosen is 919.73 MHz. Hence, the minimum far field distance is 0.21 m. The tag should be placed at a distance greater than 0.21 m such that it is in the Fraunhofer region. In this setup, the distance between the tag and reader antenna is 0.3 m in order to satisfy Fraunhofer region condition. Parameters used in the measurement are shown in Table 2.

5. t-test Algorithm

Cloning tags may be detected by comparing extracted received power and stored fingerprint using t-test algorithm, as illustrated in

$$t = \frac{\overline{X_1} - \overline{X_2}}{\sqrt{{S_p}^2((1/N_1) + (1/N_2))}},$$
$$S_p{}^2 = \frac{(N_1 - 1){S_1}^2 + (N_2 - 1){S_2}^2}{N_1 + N_2 - 2}, \tag{3}$$

Table 4: t-test for Tag B and suspicious tag.

	Tag B	Suspicious tag
Mean	−0.30055	0.316192
Variance	0.051325	0.07446
Observations	50	5
Pooled Variance	0.053072	
Hypothesized Mean Difference	0	
df	53	
t Stat	−5.70766	
$P(T \le t)$ one tail	2.6E−07	
t Critical one-tail	1.674116	
$P(T \le t)$ two-tail	**5.26E−07**	
t Critical two-tail	2.005746	

where $\overline{X_1}$ and $\overline{X_2}$ are the means of legitimate and suspicious tag groups, N_1 and N_2 are the number of samples for legitimate and suspicious groups, respectively, and $S_p{}^2$ is the pooled variance. t-test algorithm is a statistical test used to identify differences in the means and variances of two populations, namely, legitimate tag and suspicious tag populations. Significant level equals to 0.05 is chosen in conducting the t-test in order to verify the probability of a false rejection. The tag used can be considered as counterfeit if P value obtained from t-test is less than significant level, 0.05. The tag is proved as counterfeit tag because the matching probability between stored fingerprint and measured fingerprint is less.

When a tag is suspected to be counterfeit, comparison of stored and measured tag's fingerprint experiment needs to be conducted. In Case 1, a suspicious tag claims to belong to Tag A based on the stored fingerprint. As demonstrated in Table 3, P value obtained from the t-test within Tag A and the suspicious tag is higher than 0.05. This proves that no significant difference exists between the suspicious tag and Tag A. Hence, the suspicious tag is a legitimate tag. The higher the P value is, the more likely that the two groups will match. Otherwise, the tag is proved to be a counterfeit one. In Case 2, a suspicious tag claims to belong to Tag 4. A t-test is conducted between the suspicious tag and Tag B. The P value obtained from Table 4 is less than 0.05. Hence, the suspicious tag from Case 2 is proved to be a counterfeit.

6. Fingerprint-Matching Performance Analysis

The accuracy of proposed fingerprint-matching method in distinguishing between legitimate and counterfeit tags as shown in Case 2 is analyzed by using FAR, FRR, ROC, and EER. A 2 × 2 contingency table is used to verify four outcomes from the data obtained from Case 2. The outcome is a true acceptance (TA) when measured fingerprint is verified as a genuine value and the tag identity is found in the database. When the measured fingerprint has genuine value but the tag identity is not found in the database, the outcome is false acceptance (FA). Conversely, true reject (TR) is obtained when measured fingerprint has bogus value and the tag identity is not found in the database. False reject

Table 5: Four outcomes from fingerprint matching method.

		Existence of measured fingerprint in database	
		Yes	No
Genuineness of measured fingerprint	Yes	True acceptance (TA) 50	False acceptance (FA) 2
	No	False reject (FR) 0	True reject (TR) 48

Table 6: FAR and FRR for Case 2.

Measurement	Percentage (%)
FAR	4
FRR	0

Table 7: Accuracy of test categorization.

AUC range	Categories
0.50–0.60	Failure
0.60–0.70	Poor
0.70–0.80	Fair
0.80–0.90	Good
0.90–1.00	Excellent

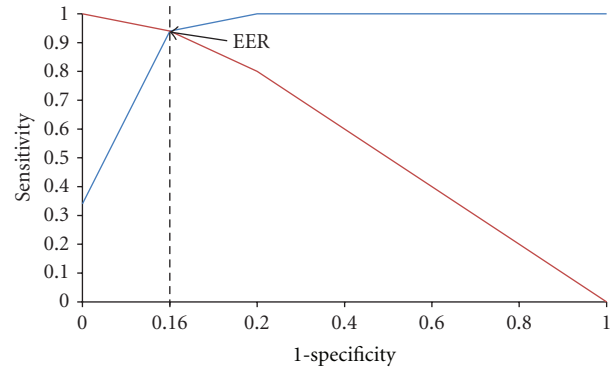


Figure 5: Receiver operating curve with equal error rate.

(FR) is obtained when measured fingerprint is verified as a bogus value but the tag identity is found in the database. Table 5 illustrates four outcomes obtained from fingerprint-matching method for Case 2.

False acceptance rate (FAR) is the measurement of probability in which the fingerprint-matching method falsely verifies different tags as identical. False rejection rate (FRR) is the measurement of probability in which the fingerprint-matching method falsely verifies identical tags as different. FAR and FRR are calculated using (4) and (5), respectively [15],

$$\text{FAR} = \frac{\text{FA}}{\text{FA} + \text{TR}}, \tag{4}$$

$$\text{FRR} = \frac{\text{FR}}{\text{FR} + \text{TA}}. \tag{5}$$

FAR and FRR for Case 2 are shown in Table 6.

ROC curve and EER are used to evaluate the performance of t-test algorithm in verifying measured fingerprint with stored fingerprint. ROC curve illustrated in Figure 5 plots the true acceptance rate (TAR) versus its false acceptance rate (FAR). EER is the rate at which both FAR and FRR are equal. Based on the ROC curve, EER for Case 2 is 0.16, which is considered as a low value. The lower the EER is, the more accurate will be the fingerprint-matching method.

The area under curve (AUC) of the ROC curve is a measurement of the performance of t-test algorithm in distinguishing between two fingerprint data sets. The accuracy of the t-test algorithm is verified using a rough guide for classifying the accuracy of a test as shown in Table 7 [16, 17].

AUC for Case 2 that obtained from SPSS statistical analysis result is 0.922 as shown in Table 8, which is considered an excellent performance according to the accuracy guide. This proves that the t-test algorithm offers high accuracies in distinguishing fingerprints between data sets of two tags.

7. Security Analysis

The security of proposed protocol that is written in HLPSL is validated using AVISPA tool. The intruder under the Dolev-Yao model has capability to full control over the network [18]. The intruder may intercept and analyze transmitted message as well as impersonate one of the agents (tag, reader, and server) to send modified message to others. Data secrecy and mutual authentication are the security goals that needed to achieve in AVISPA tool. The E_T as well as session keys K_i and K_s are kept secret in the transmission channel. An attack is considered happened if intruder is able to obtain any secret values. In addition, tag and back-end server are only allowed to reveal their identity information to the authorized parties. Back-end server needs to ensure that the current session's message, M_1, is the message that computed by legitimate tag. This is to prevent replay attack where intruder sends previous session's message to the legitimate reader. Same case is applied to M_2. The authentication of M_2 must be verified by the tag as a legitimate message that sent by the legitimate reader. Figure 6 shows that OFMC and CL-AtSe back-ends found no man-in-the-middle attacks and the stated security goals were satisfied for a bounded number of sessions as specified in environment role. The strong authentication between the tag and back-end system is established and the secrecy of the EPC and session keys are protected from eavesdropping. The analysis using SATMC and TA4SP on the proposed protocol is inconclusive because the back-ends only support protocols that are free of algebraic equation.

Replay attack can be prevented in this proposed protocol because the value transmitted for each session is different. The proposed protocol is a challenge-response

Table 8: Accuracy of test categorization.

Area	Std. error[a]	Asymptotic sig.[b]	Asymptotic 95% confidence interval	
			Lower bound	Upper bound
0.922	0.027	0.000	0.869	0.975

[a]Under the nonparametric assumption.
[b]Null hypothesis: true area = 0.5.

Table 9: Comparison between schemes.

Scheme	Replay attack	DoS attack	Cloning attack	Forward security	EPC Class-1 Gen-2
Chien and Chen [2]	X	O	X	X	O
Chien and Huang [6]	O	O	X	O	O
Song and Mitchell [8]	X	O	X	X	X
Song [9]	O	X	X	O	X
Burmester and Munilla [10]	X	O	X	O	O
Chen and Deng [11]	O	O	X	O	O
Proposed Scheme	O	O	O	O	O

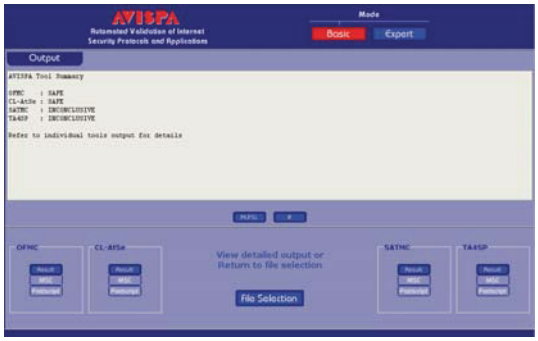


Figure 6: AVISPA validation result.

mutual authentication protocol that is based on one-time pad encryption. Hence, different value of session key is utilized in individual session and PRNG plays a vital role in providing different value of session key to encrypt with E_T. A random number, n, is XOR with K_i to use as a seed of the PRNG. The seed is regenerated for each session to reduce the possibility of successfully cracked by adversaries. For each session, M_1 and M_2 are enciphered by using corresponding session keys, K_i and K_s by the tag and server respectively. These session keys are synchronously updated during mutual authentication by both tag and server. Hence attacker is unable to use the session keys, K_i and K_s, of a particular session to decipher encrypted message for any of the following sessions.

DoS attack can be defended by using updated session key. The legitimate tag can be identified by verifying the encrypted message with message recorded in the database. On the other hand, the authentication of the reader is verified by the tag by comparing the decrypted message with message recorded in the tag. Both new and old indexes, session keys, and random numbers that are stored in the back-end server are used to prevent desynchronized issue. Desynchronization problem occurred when variables stored in the tag are different with the one stored in the database. Hence, the server can use old variables to resynchronize with the tag.

The secrecy of the tag's information is safe from eavesdropping attack. The E_T is enciphered with session key where the session key will be updated after each complete session. In addition, tag is hard to compromise because M_1 and M_2 are enciphered by using different key. If M_1 and M_2 are eavesdropped between legitimate tag and reader, the attacker is unable to obtain any secret information. For example, $M_1 \oplus M_2 = [\text{CRC}\ (E_T \oplus K_i)] \oplus [\text{CRC}\ (E_T \oplus K_s)] = [\text{CRC}\ (E_T \oplus E_T \oplus K_i \oplus K_s)] = [\text{CRC}\ (0 \oplus K_i \oplus K_s) = \text{CRC}\ [(0 \oplus K_i \oplus K_s)]$. Hence, attacker is only able to get enciphered key and is impossible to guess its original key value.

The proposed protocol can prevent the issue of cloning tags by using fingerprint information stored in the database to detect counterfeit tags. Each tag has it own unique received power of tag value. Even though adversaries are able to copy all the data from a tag, they are unable to create a counterfeit tag that has the exact same physical feature as original tag. Thus, any counterfeit tag can be found when the fingerprint of tag detected is not matched with the fingerprint information stored in the tag. The proposed method is analyzed by using one factor only, which is received power of tag at single frequency, whereas two factors, namely, minimum power responses at multiple frequencies and physical characteristic of tags, are tested by using ANOVA in [12]. The accuracy of the proposed method and method of [12] is excellent in both, with the values of 0.922 and 0.999, respectively. The proposed method is simpler but capable to produce comparable accuracy of method [12] which analyses two factors to detect cloning tags.

Table 9 indicates a comparison of results between proposed scheme and related security schemes in terms of replay attack, DoS attack, cloning attack, forward secrecy, and Gen 2 standards compliance. The proposed lightweight

cryptographic mutual authentication protocol is proved to possess more security protection compared to existing security schemes.

8. Conclusions

This paper proposed the use of both prevention and detection methods to enhance the security level in an RFID system. The lightweight cryptographic mutual authentication protocol that consists of lightweight cryptographic algorithm, including XOR, CRC, and PRNG functions, is used as prevention method. The security of proposed protocol is validated using AVISPA tool and is proved safe from replay attack, denial of service threats, and data leakage problem.

In addition, tag's fingerprint extraction and matching method is presented as a detection method in detecting counterfeit tags. Each tag received power is measured, calculated, and stored in the database for further reference. Tag received power can be used as unique fingerprint as these are significantly different in the frequency range of 919–923 MHz. t-test algorithm is used to determine the identity of measured tag. Measured tag is proved as counterfeit if the P-value of the t-test conducted is less than 0.05. Accuracy of the fingerprint-matching method is tested, and 4% of FAR and 0% of FRR is achieved. In addition, fingerprint-matching is proved to be an excellent method, as the area under the ROC curve is 0.922 and ERR is 0.16. Hence, t-test algorithm was proved to be able to protect RFID communication system from tags cloning attack by efficiently distinguishing between legitimate and counterfeit tags.

Acknowledgments

The authors would like to thank the School of Electrical and Electronic Engineering, USM and the USM RU (Research University) grant secretariat, for sponsoring this work.

References

[1] D. Bailey and A. Juels, "Shoehorning security into the EPC tag standard," in *Proceeding of Security and Cryptography for Networks*, pp. 303–320, Berlin, Germany, 2006.

[2] H. Y. Chien and C. H. Chen, "Mutual authentication protocol for RFID conforming to EPC Class 1 Generation 2 standards," *Computer Standards and Interfaces*, vol. 29, no. 2, pp. 254–259, 2007.

[3] M. Bouet and G. Pujolle, "RFID in eHealth systems: applications, challenges, and perspectives," *Annals of Telecommunications*, vol. 65, no. 9-10, pp. 497–503, 2010.

[4] A. Razaq, W. T. Luk, K. M. Shum, L. M. Cheng, and K. N. Yung, "Second-generation RFID," *IEEE Security and Privacy*, vol. 6, no. 4, pp. 21–27, 2008.

[5] B. Danev, T. S. Heydt-Benjamin, and S. Capkun, "Physical-layer identification of RFID devices," in *Proceedings of the USENIX Security Symposium*, pp. 199–214, 2009.

[6] H. Y. Chien and C. W. Huang, "A lightweight authentication protocol for low-cost RFID," *Journal of Signal Processing Systems*, vol. 59, no. 1, pp. 95–102, 2010.

[7] Y. Z. Li, Y. B. Cho, N. K. Um, and S. H. Lee, "Security and privacy on authentication protocol for low-cost RFID," in *Proceedings of the International Conference on Computational Intelligence and Security (ICCIAS '06)*, pp. 1101–1104, October 2006.

[8] B. Song and C. J. Mitchell, "RFID authentication protocol for low-cost tags," in *Proceedings of the 1st ACM Conference on Wireless Network Security (WiSec '08)*, pp. 140–147, April 2008.

[9] B. Song, "RFID tag ownership transfer," in *Proceedings of the 4th Workshop on RFID Security*, Budaperst, Hungary, 2008.

[10] M. Burmester and J. Munilla, "A flyweight RFID authentication protocol," in *Proceedings of the 4th Workshop on RFID Security*, Budaperst, Hungry, 2008.

[11] C. L. Chen and Y. Y. Deng, "Conformation of EPC Class 1 Generation 2 standards RFID system with mutual authentication and privacy protection," *Engineering Applications of Artificial Intelligence*, vol. 22, no. 8, pp. 1284–1291, 2009.

[12] S. C. G. Periaswamy, D. R. Thompson, and D. Jia, "Fingerprinting RFID tags," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 6, pp. 938–943, 2011.

[13] D. Zanetti, B. Danev, and S. Căpkun, "Physical-layer identification of UHF RFID tags," in *Proceedings of the 16th Annual Conference on Mobile Computing and Networking (MobiCom '10)*, pp. 353–364, September 2010.

[14] MCMC, "Strategies for a National RFID Roadmap for The Next Five Years," 2010.

[15] J. Zhou, H. Shirai, I. Takahashi, J. Kuroiwa, T. Odaka, and H. Ogura, "A Hybrid Command Sequence Model for Anomaly Detection," in *Proceeding of 11th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Nanjing, China, 2007.

[16] Y. Kutlu and D. Kuntalp, "A multi-stage automatic arrhythmia recognition and classification system," *Computers in Biology and Medicine*, vol. 41, no. 1, pp. 37–45, 2011.

[17] C. J. Chevillotte, M. H. Ali, R. T. Trousdale, D. R. Larson, R. E. Gullerud, and D. J. Berry, "Inflammatory laboratory markers in periprosthetic hip fractures," *Journal of Arthroplasty*, vol. 24, no. 5, pp. 722–727, 2009.

[18] Avispa, "HLPSL Tutorial—A Beginner's Guide to Modeling and Analyzing Internet Security Protocols," 2006.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 847805, 12 pages
doi:10.1155/2012/847805

Research Article

Secrecy Transfer

Zhihong Liu,[1] Jianfeng Ma,[1] Yong Zeng,[1] Li Yang,[1] and YoungHo Park[2]

[1] *School of Computer Science and Technology, Xidian University, Xi'an 710071, China*
[2] *Department of Electrical Engineering, Kyungpook National University, Daegu 702-701, Republic of Korea*

Correspondence should be addressed to Zhihong Liu, liuzhihong@mail.xidian.edu.cn

Received 7 March 2012; Revised 8 June 2012; Accepted 10 June 2012

Academic Editor: Mihui Kim

Suppose that n nodes with n_0 acquaintances per node are randomly deployed in a two-dimensional Euclidean space with the geographic restriction that each pair of nodes can exchange information between them directly only if the distance between them is at most r, the acquaintanceship between nodes forms a random graph, while the physical communication links constitute a random geometric graph. To get a fully connected and secure network, we introduce secrecy transfer which combines random graph and random geometric graph via the propagation of acquaintanceship to produce an acquaintanceship graph G_{n,n_0}, a kind of random geometric graph with each edge representing an acquaintanceship between two nodes. We find that components of graph G_{n,n_0} that undergoes a phase transition from small components to a giant component when n_0 is larger than a threshold, the threshold for G_{n,n_0} to be a connected graph is derived. In addition, we present its implementation method and applications in wireless sensor networks.

1. Introduction

Suppose at a classroom with n students, each of whom initially has n_0 acquaintances who are randomly chosen among them. Students can only communicate with its direct neighbors. At first, students are isolated. If two adjacent students are acquainted with each other, a link forms between them. As time goes on, some small acquaintance groups emerge. Two stranger students, say Alice and Bob, belonging to different groups may be adjacent, but if there are students in the two groups, respectively, familiar with each other, Alice and Bob may use them as introducers to establish a link between them. By repeating this process, students will be increasingly interwoven by such links, creating a web of acquaintances. We denote this construction as *secrecy transfer* and the resulting network as the *acquaintanceship graph* G_{n,n_0}. We are here interested in the question: for which critical threshold of n_0 is there likely to be a connected acquaintanceship graph?

At first glance, the acquaintanceship graph is a kind of social networks, such as the patterns of friendships between individuals. Technically, the acquaintanceship graph is a combination of random graph [1] and random geometric graph [2]. A random geometric graph $G_{n,r}$ is a graph resulting from placing n nodes randomly in a plane and connecting each pair of nodes if their distance is at most the radius r, while a random graph $G_{n,p}$ is a graph with n nodes in which each edge (out of the $\binom{n}{2}$ possible edges) is chosen independently at random with an edge probability p. The acquaintanceship graph G_{n,n_0} has both properties of random graph and random geometric graph. Initially, n nodes are placed randomly on a plane, every node has n_0 acquaintances. In the view of acquaintanceship, it can be considered logically as a random graph $G_{n,p}$ without geographical position restriction. If graph G_{n,n_0} is connected, everyone can make the acquaintance of arbitrary nodes in the graph. Intuitively, we think that there is a threshold value. If n_0 is larger than that value, the graph G_{n,n_0} may be connected. If the communication between any pair of acquaintances is considered secure and trusted, through the introduction of acquaintances, any one can extend his circle of acquaintance and eventually get secure communication with arbitrary nodes in the graph G_{n,n_0}.

Random graphs and random geometric graphs have been studied extensively, but in a separate way. Random graph and its variations have been used as models of social structure in, for example, epidemiology [3], while random geometric graph is always viewed as a wireless communication network

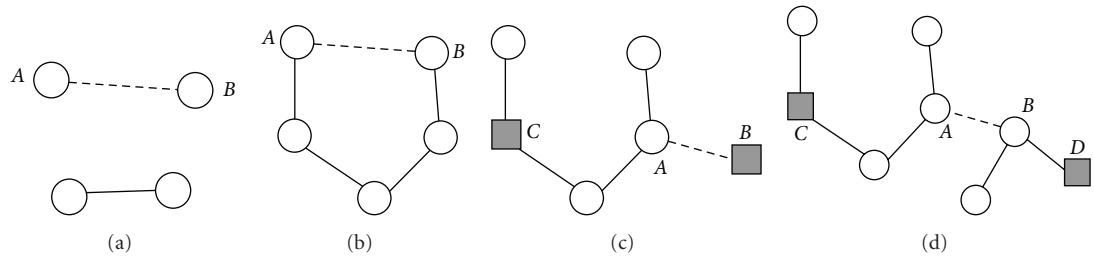


Figure 1: Secrecy transfer.

[4, 5], such as Ad hoc, Mesh, or sensor networks. In fact, random graphs and random geometric graphs have different structural properties. Any two nodes in a random graph can be connected by a link with certain probability regardless of their geographical position. Random key graphs have been recently been used by Di Pietro et al. [6] to model the random key predistribution scheme of Eschenauer and Gligor [7]. The random key graph is a random graph obtained as follows: n nodes, each assigned a subset of keys, are distributed uniformly at random on a given field. An edge is added if two nodes are within a radius r and share at least one common key. Formally, the resulting graph, matching a random graph with identical link probability to a random geometric graph, can be considered as the initial graph of the acquaintanceship graph G_{n,n_0}, if two nodes, sharing one common key, are referred to as acquaintances. Note that, unlike random key graphs, secrecy transfer is a *growth* model and can be considered as a stochastic process. We are interested in the crucial property, connectivity, of the resulting acquaintanceship graph. In [8], we use secrecy transfer to enhance the security performance of key infection [9]. In fact, secrecy transfer in [8] only focuses on key establishment between adjacent nodes who are in a connected component; otherwise, key infection is applied to establish a secret link key. Obviously, it is a tradeoff between key infection and secrecy transfer discussed in this paper. In this paper, some results are given and complemented by simulations, especially the connectivity threshold.

Organization. First, secrecy transfer is presented in Section 2. We derive the connectivity threshold of value n_0 in Section 3. Next, in Section 4, we present the analysis of secrecy transfer in heterogeneous networks. Section 5 gives an implementation method of secrecy transfer. In Section 6, some applications are given. Finally, we conclude the paper in Section 7.

2. Secrecy Transfer

Let n nodes be distributed uniformly and independently at random in a field $[0,1]^2$, each of them has n_0 acquaintances. A pair of nodes are adjacent only if the distance between them is at most the radius r. Suppose nodes A and B are adjacent, that is, the distance between them $|A - B| < r$. Initially, A and B are connected if they are acquainted with each other (*initialization phase*, see Figure 1(a)). If nodes A and B are connected by a path, an edge $A - B$ is added (see Figure 1(b)). As time goes on, the graph G_{n,n_0} evolves continuously, and gradually consists of some components. If node A belongs to a component C_A, and B has acquaint with at least one of nodes in the component C_A, say node C in C_A, we connect A and B by a new edge (see Figure 1(c)). For the case where A and B belong to different component C_A and C_B, if there exist two acquaintance nodes C and D in C_A and C_B, respectively, we introduce an edge between A and B; Otherwise, A and B are disconnected at present stage. If there is no new edge is added for any pair of adjacent nodes, secrecy transfer reaches the *stable state* and the algorithm terminates. Continuing this process, we can get a acquaintanceship graph G_{n,n_0}.

As depicted in Figure 2, 100 nodes are randomly distributed over a $100 \times 100\,\mathrm{m}^2$ field, $n_0 = 4$, and the radius $r = 20\,\mathrm{m}$. At first, two adjacent nodes connect with the probability $p = n_0/n = 0.04$, and we get the initial acquaintanceship graph G_{n,n_0}, as illustrated in Figure 2(a). After repeating secrecy transfer process on the graph G_{n,n_0}, it gradually evolves into the graph shown in Figure 2(g), which approximates to the underly random geometric graph $G_{n,r}$.

One of our goals is to design a *security mechanism* to enable any two adjacent nodes to establish a pairwise key after they are deployed in a field. More specifically, suppose that every node in the network has been preloaded before its deployment with n_0 secret keys, each of which is shared with one of its acquaintances. Let nodes A and B be two adjacent nodes, $|A - B| < r$. If nodes A and B happen to be acquaintances, they share a key K_{AB} which can be used to protect their communication link. If A and B are not acquaintances, but are connected by a path (Figure 1(b)), A can generate a new key K_{AB} and send it to B along the path. As more secure edges are added to the graph G_{n,n_0}, larger components emerge. Suppose node A belongs to a component C_A, (as plotted in Figure 1(c)) if node B acquaints with a node $C \in C_A$, which means that nodes B and C have a shared key K_{BC}. In this case, node A randomly generates a key K_{AB} and sends it along a path in the component C_A to node C. Node C encrypts K_{AB} with the key K_{BC}, that is, $\{K_{AB}\}_{K_{BC}}$ and sends the result back to A. Node A, then, sends $\{K_{AB}\}_{K_{BC}}$ to B via the unsecure channel. Finally, node B can get key K_{AB}, for it has the key K_{BC}. In another case, where nodes A and B belong to different components C_A and C_B, but node $C \in C_A$ acquaints with node $D \in C_B$, as shown in Figure 1(d). Node A first sends a key K_{AB} to node C. Node C encrypts K_{AB} with key K_{CD} which is shared with

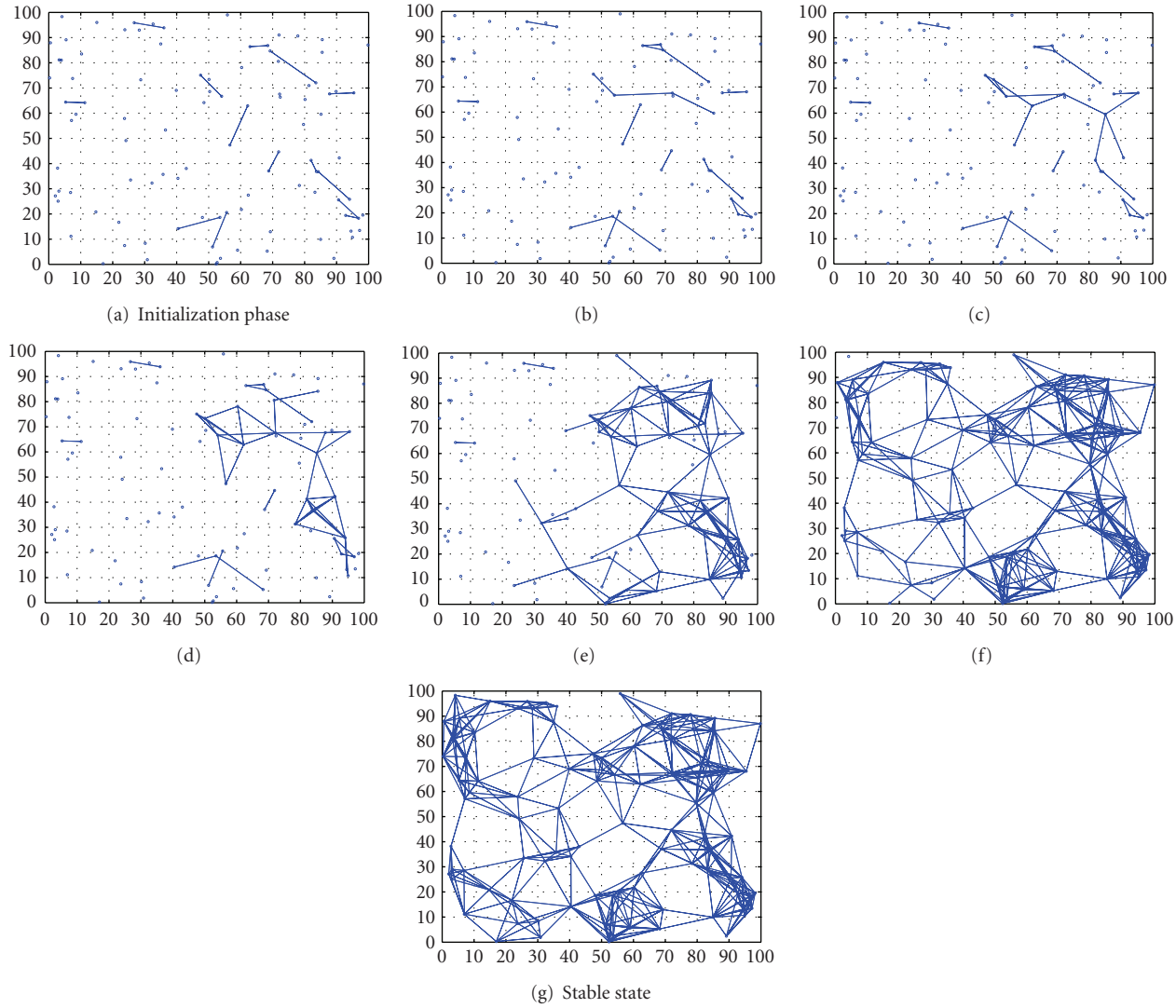


FIGURE 2: An example of secrecy transfer process, with $n = 100$ nodes randomly distributed over a $100 \times 100\,\text{m}^2$ field, $n_0 = 4$, and $r = 20\,\text{m}$.

node D and sends $\{K_{AB}\}_{K_{CD}}$ to node D via nodes A and B. For node D has K_{CD}, it can decrypt the message $\{K_{AB}\}_{K_{CD}}$ to obtain K_{AB}.

Given a randomly deployed network with n nodes, we can view it as a random geometric graph $G_{n,r}$ with each edge representing a possible communication link. Without the protection of a secret key, an adversary can eavesdrop conversations between nodes. If each node has several trusted nodes initially, the trust relationship can be considered as a random graph $G_{n,p}$ with each edge connecting a pair of nodes which have shared a secret key. However, random graph does not consider the transmission radius of nodes, but simply assumes any two nodes have the same probability p to establish a connection. When the distance between two nodes is larger than the transmission radius r, they cannot communicate directly. Roughly speaking, $G_{n,p}$ reflects the *logical* trust relationship between nodes, while $G_{n,r}$ depicts the *physical* communication structure of nodes in the network. Secrecy transfer constructs a graph G_{n,n_0} from $G_{n,r}$ and $G_{n,p}$ (where $p = n_0/n$) and turns it to a *secure* random geometric graph by adding secure edges to it.

The construction of secrecy transfer above reveals that, the graph G_{n,n_0} is robust against eavesdrop attack, for each edge is added via the existing trustiness between nodes. If cryptographic attacks are considered impractical, the adversary cannot break $\{K_{AB}\}_{K_{CD}}$ to get the key K_{AB}, for the key K_{CD} shared between C and D is loaded initially.

3. Connectivity Threshold

The component structure of the graph G_{n,n_0} changes gradually as secrecy transfer is applied. As illustrated in Figure 2(a), after the initialization phase, the greatest component of G_{n,n_0} is tree of small order. Gradually, a giant component emerges, swallowing the whole network, provided the underly random geometric graph $G_{n,r}$ is connected and n_0 is large enough.

Suppose two adjacent components, C_A and C_B, have, respectively, m_1 and m_2 vertices, nodes A in C_A and B in C_B

are adjacent. We first estimate the probability P_{m_1,m_2} that two adjacent components C_A and C_B may get connected to form a larger component.

Let random variable $\mathbb{X}$ be the total number of nodes with whom nodes in component C_A are familiar, $\mathbf{X}_i$ be a bernoulli random variable, where $\mathbf{X}_i = 1$ when the circle of acquaintances of node i includes at least one node in the component C_A, $\mathbf{X}_i = 0$ otherwise. Therefore,

$$\mathbb{X} = \mathbf{X}_1 + \mathbf{X}_2 + \cdots + \mathbf{X}_n. \tag{1}$$

If component C_A consists of m_1 nodes, we have the probability of $\mathbf{X}_i = 1$,

$$\mathbb{P}(\mathbf{X}_i = 1) = 1 - (1 - P)^{m_1}, \tag{2}$$

where $P = n_0/n$.

Thus, the expectation of random variable $\mathbf{X}_i$ is $\mathbb{E}(\mathbf{X}_i) = 1 - (1 - P)^{m_1}$.

For $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n$ are mutually independent, the expectation of $\mathbb{X}$ is

$$\mathbb{E}(\mathbb{X}) = \sum_{i=1}^{n} \mathbb{E}(\mathbf{X}_i) = n\left[1 - (1 - P)^{m_1}\right], \tag{3}$$

which means that, for a component of order m_1, the circle of acquaintances of this component may consist of $n[1 - (1 - P)^{m_1}]$ nodes on average. Let $a = n[1 - (1 - P)^{m_1}]$, the probability P_{m_1,m_2} that there is at least one common acquaintance between components C_A and C_B is

$$P_{m_1,m_2} = 1 - \left(1 - \frac{a - m_1}{n - m_1}\right)^{m_2}. \tag{4}$$

For example, one may see that, for a network $n = 10,000$, $m_1 = 200$, and $m_2 = 1$, the probability P_{m_1,m_2} tends to 1 when $P > 0.02$. This provides intuition that, a component of order 200 is *attractive* and will *swallow* nodes nearby to form a larger component, a kind of "*rich get richer*" phenomenon. For two components of order $m_1 = m_2 = 50$, the probability P_{m_1,m_2} approximates 1 if $P > 0.002$. In general, the larger the components are, the more likely they are to be mixed together.

In a random graph $G_{n,N}$ with n vertices and N edges, if $N \sim cn$ with $c \geq 1/2$, the greatest component has (with probability tending to 1 for $n \to +\infty$) approximately $n^{2/3}$ vertices [10]. As a special case, when $n = 10,000$, $n^{2/3} \approx 464$, such large component in graph G_{n,n_0} will swallow the whole network with high probability.

Next, we investigate the relationship between the connectivity property of graph G_{n,n_0} and value n_0. To determine the value n_0 which will guarantee the connectivity of graph G_{n,n_0}, we employ a well-known algorithm to generate random graphs $G_{n,p}$ with a given degree distribution [11]. Each graph generated has n nodes and $n_0 = np$ links per node. The algorithm may lead to a multigraph, either by connecting a node to itself or by connecting two nodes together more than once. However, as n increases, these events become rare and their number becomes statistically insignificant.

We first generate a random graph with n nodes and n_0 links per node, then deploy the nodes into a square region to obtain a random topology. For $n = 500$, $r = 35$ m, and n_0 varying, we repeat our simulations 50 times to yield an acceptable confidence of results. For each simulation, we measure empirical values for the maximum component and the second component for each trial, averaged over 50 random topologies. In Figure 3, an interesting phenomenon observed is a "*phase transition*" as n_0 increases. There is a critical value of n_0, above which the graph will almost surely be connected. The maximum component grows rapidly from a component of small size to a giant component when $n_0 > 10$. On the contrary, the size of the second component decreases as $n_0 > 10$.

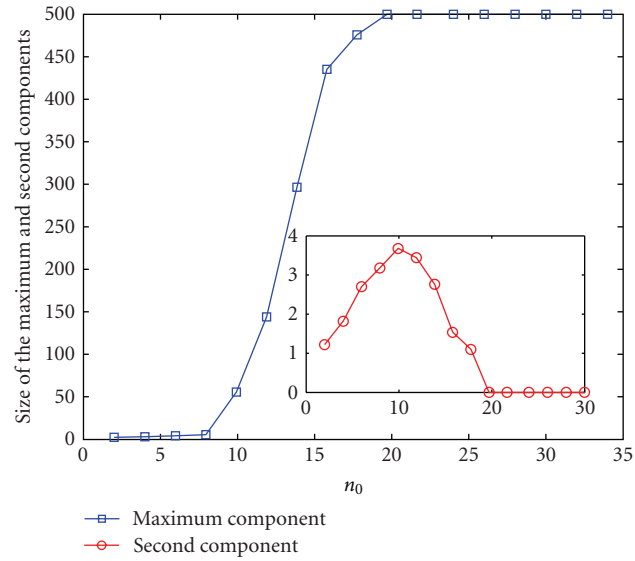


Figure 3: Size of the maximum and second components in graph G_{n,n_0} for $n = 500$, $r = 35$ m.

Within this context, we want to know, under what conditions is the graph G_{n,n_0} be connected? How can we choose n_0 such that the graph G_{n,n_0} constructed by secrecy transfer will be connected with high probability? The answer to this question is crucial in determining the number of acquaintances that an arbitrary node should have initially.

3.1. Lower Bound of Connectivity Threshold. To get a fully connected graph G_{n,n_0}, two conditions must be satisfied. First, the graph $G_{n,r}$ must be connected, which means that, given the value n and a deployment region, the value r should be large enough to guarantee a connected random geometric graph $G_{n,r}$. Assume n nodes are uniformly deployed in a unit square $[0,1]^2$, the well-known connectivity threshold $r_c = \sqrt{(\log n \pm O(1))/\pi n}$ [5]. Second, the value n_0 must be large enough to get the random graph $G_{n,p}$ fully connected. Consider an arbitrary pair of adjacent nodes A and B in graph G_{n,n_0} which have not established secret key between them. For $G_{n,p}$ is connected, there is at least one path in graph $G_{n,p}$, say $P_{AB} = Ax_0x_1 \cdots x_kB$ between nodes A and B. Given any adjacent nodes in path P_{AB}, say x_i and x_{i+1}, there must exist a path $P' = x_iy_1y_2 \cdots y_tx_{i+1}$ from x_i to x_{i+1} in graph $G_{n,r}$, for graph $G_{n,r}$ is connected.

For a random graph $G_{n,p}$, when p is zero, the graph does not have any edge, whereas when p is one, the graph is fully connected. Bollobás showed that, for monotone properties, there exists a value of p such that the property moves from "nonexistent" to "certainly true" in a very large random graph [1]. The function defining p is called the *threshold function* of a property. Given a desired probability P_c for graph connectivity, the threshold function p of $G_{n,p}$ is defined by

$$P_c = \lim_{n\to\infty} P_r\Big[G_{n,p} \text{ is connected}\Big] = e^{e^{-c}}, \tag{5}$$

where $p = \ln(n)/n + c/n$ and c is any real constant.

Therefore, given n we can find p for which the resulting graph $G_{n,p}$ is connected with desired probability P_c. Thus, the lower bound of connectivity threshold of n_0 is

$$n_0 = p \times (n-1) = \frac{n-1}{n}[\ln(n) - \ln(-\ln(P_c))]. \tag{6}$$

3.2. Analysis Results of Connectivity Threshold. Notice that when n_0 is below the lower bound of connectivity threshold mentioned above, the graph $G_{n,p}$ is not connected with high probability, and hence the graph G_{n,n_0} also is not connected with high probability. However, a greater n_0 above the lower bound cannot guarantee a connected graph G_{n,n_0} when n' is small, where n' is the average number of neighbors of a node. For a tighter bound of n_0, correlated with n, n', is not known yet, we only present some analysis results of n_0 below.

After the initialization phase of secrecy transfer, we get a random graph. Erdös and Rényi showed that for random graphs, a giant component exists if the average degree of node $\langle k\rangle > 1$[10] . If $\langle k\rangle < 1$ only small components exist, and the size of the largest component is proportional to $\ln n$ (n is the number of nodes in the graph). Exactly at the threshold, $\langle k\rangle = 1$, a component whose size is proportional to $n^{2/3}$ emerges. In the sequel, when $n = 10,000$, $n' = 10$, and $n_0 = 200$, after the initialization phase of secrecy transfer, the average degree of nodes in the graph G_{n,n_0}, $\langle k\rangle = n'(n_0/n) = 10 \times (200/10000) = 0.2$. In this occasion, the initial graph G_{n,n_0} only consists of small components, such as trees of small size. As the simulation results of initialization phase shown in Figure 2(a), the graph G_{n,n_0} only contains small isolated trees. Our goal is to determine how many such components exist, and the probability that they will be connected to form a giant component.

Let the graph G_{n,n_0} in the initialization phase contains components $C_1, C_2, \ldots, C_i$, such that the size of component C_i is $|C_i| = \lambda_i$, and $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_i \geq 2$. For two components C_i and C_j of order λ_i and λ_j, if they are adjacent, the probability that they will be connected through secrecy transfer is

$$P_{\lambda_1,\lambda_2} = 1 - \left(1 - \frac{a-\lambda_1}{n-\lambda_1}\right)^{\lambda_2}, \tag{7}$$

where $a = n[1-(1-P)^{\lambda_1}]$, $P = n_0/n$.

When secrecy transfer is applied, the graph G_{n,n_0} will evolve continuously. A larger component is more attractive and will swallow nodes nearby to form a larger component with high probability. *Popularity is attractive*. If a component can absorb nodes nearby by secrecy transfer, it is termed as *expandable*. Next we estimate the asymptotic probability $P_{\text{expandable}}$ that at least one component in $C_1, C_2, \ldots, C_i$ is expandable. At first we estimate the number of neighbors n'_λ of a component of size λ.

Suppose n nodes distributed uniformly and independently at random in a unit area S, that is, $S = 1$. Let a component C of size $|C| = \lambda$ lie inside a circle of radius R, n'_λ neighbors of the component C lie outside the circle and be at distance at most r from nodes in the component. Let S' be a subarea in the deployment area S, $S' \ll S$. The probability that a node is placed within area S' is $P = S'/S = S'$. Then, the probability $P(x)$ that of t nodes are placed in the area S' is

$$P(x=t) = \binom{n}{t} P^t (1-P)^{n-t}. \tag{8}$$

When $n \gg 1$ and $S' \ll S$, we can approximate it with a Poisson distribution,

$$P(x=t) \approx \frac{e^{-nP}\cdot(np)^t}{t!} = \frac{e^{-nS'}(nS')^t}{t!}, \tag{9}$$

and the average number of nodes within area S' is

$$t = nS'. \tag{10}$$

In the initial graph G_{n,n_0}, any component of size λ is small ($\lambda \ll n$), and the area S' it occupied is also small, that is, $S' \ll S = 1$. Therefore, we have approximately

$$\lambda = n\pi R^2, \qquad n' = n\pi r^2. \tag{11}$$

Thus,

$$n'_\lambda = n\Big[(R+r)^2\pi - r^2\pi\Big] = n' + 2\sqrt{\lambda n'}. \tag{12}$$

Therefore, the probability P_λ that a component of size λ is expandable is

$$P_\lambda = 1 - (1 - P_{\lambda,1})^{n'_\lambda}, \tag{13}$$

where $P_{\lambda,1} = (a-\lambda)/(n-\lambda)$, $a = n[1-(1-P)^\lambda]$, $P = n_0/n$.

For a set of components $C_1, C_2, \ldots, C_i$, $|C_i| = \lambda_i$, the probability $P_{\text{expandable}}$ can be calculated as

$$\begin{aligned} P_{\text{expandable}} &= 1 - (1-P_{\lambda_1})(1-P_{\lambda_2})\cdots(1-P_{\lambda_i}) \\ &= 1 - (1-P_{\lambda_1,1})^{n'_{\lambda_1}}(1-P_{\lambda_2,1})^{n'_{\lambda_2}}\cdots(1-P_{\lambda_i,1})^{n'_{\lambda_i}} \\ &\approx 1 - \exp\Big\{-\Big(n'_{\lambda_1}P_{\lambda_1,1} + n'_{\lambda_2}P_{\lambda_2,1} + \ldots + n'_{\lambda_i}P_{\lambda_i,1}\Big)\Big\}. \end{aligned} \tag{14}$$

Note that for two sets of components $\mathbb{C} = \{C_1, C_2, \ldots, C_i\}$ ($|C_i| = \lambda_i$) and $\mathbb{C}' = \{C'_1, C'_2, \ldots, C'_i\}$ ($|C'_i| = \lambda'_i$), if $\lambda_1 \geq \lambda'_1, \lambda_2 \geq \lambda'_2, \ldots \lambda_i \geq \lambda'_i$, then

$$\begin{aligned} n'_{\lambda_1} \geq n'_{\lambda'_1}, \ldots, n'_{\lambda_i} \geq n'_{\lambda'_i}, \\ P_{\lambda_1,1} \geq P_{\lambda'_1,1}, \ldots, P_{\lambda_i,1} \geq P_{\lambda'_i,1}. \end{aligned} \tag{15}$$

Therefore, the probability $P_{\text{expandable}}$ that at least one component in set $\mathbb{C}$ is expandable is greater than that in set $\mathbb{C}'$.

Given parameters n, n', and n_0, after the initialization phase of secrecy transfer, the graph G_{n,n_0} may contain some components $C_1, C_2, \ldots, C_i$. The expandable probability of this component set $P_{\text{expandable}} \to 1$, implies that at least one component in $C_1, C_2, \ldots, C_i$ is expandable and will grow larger with high probability. Let a component C_1 be expandable and become a larger component C_1' by swallowing nodes nearby. For $|C_1'| > |C_1|$, the expandable probability $P'_{\text{expandable}}$ of components $C_1', C_2, \ldots, C_i$ is greater than the expandable probability $P_{\text{expandable}}$ of components $C_1, C_2, \ldots, C_i$, that is,

$$P'_{\text{expandable}} > P_{\text{expandable}} \longrightarrow 1. \tag{16}$$

Thus, if the expandable probability $P_{\text{expandable}}$ of the initial graph approximates 1, it will become even greater almost surely, and the graph G_{n,n_0} will eventually evolve into a connected graph with high probability if both $G_{n,p}$ and $G_{n,r}$ are connected graphs. However, small expandable probability of the initial graph G_{n,n_0} cannot guarantee a connected graph with high probability and secrecy transfer will terminate with isolated components.

However, exact results of the critical threshold n_0 are not known yet, we only present some analysis results below.

In an Erdös-Rényi random graph $G_{n,N}$ with n nodes and N links [10], if $N \sim l \cdot n^{(k-2)/(k-1)}$ where $l > 0$, then the number of trees of order k contained in $G_{n,N}$ has in the limit for $n \to +\infty$ a Poisson distribution with mean value

$$\bar{\lambda} = \frac{(2l)^{k-1}k^{k-2}}{k!}. \tag{17}$$

Among these trees, the probability $P_{\text{expandable}}$ that at least one tree of order k is expandable is

$$P_{\text{expandable}} = 1 - (1 - P_k)^{\bar{\lambda}}, \tag{18}$$

where P_k is the probability that a tree of order k is expandable, $P_k = 1 - (1 - P_{k,1})^{n'_k}$, $n'_k = n' + 2\sqrt{kn'}$, $P_{k,1} = (a-k)/(n-k)$, $a = n[1-(1-n_0/n)^k]$, and $\bar{\lambda} = (2l)^{k-1}k^{k-2}/k!$.

For the graph G_{n,n_0} after the initialization phase of secrecy transfer, the number of links in G_{n,n_0} is

$$N = \frac{1}{2}n \cdot n'P = \frac{1}{2}n \cdot n'\frac{n_0}{n} = \frac{1}{2}n'n_0 \sim l \cdot n^{(k-2)/(k-1)}, \tag{19}$$

which yields the result

$$k \sim 1 + \frac{\ln n}{\ln(2ln/n'n_0)}. \tag{20}$$

Using the above considerations, we can estimate the expandable probability $P_{\text{expandable}}$ of the components (trees) of order k.

We see that, for $n = 10,000$, $n' = 10$, and $n_0 = 100$, the probability $P_{\text{expandable}} \approx 0.6991$; for $n_0 = 200$, the probability $P_{\text{expandable}} \approx 1$. This indicates that, in this occasion, for $n_0 \geq 200$, the graph G_{n,n_0} will eventually evolve into a connected graph by secrecy transfer with high probability. However, for $n_0 < 100$, the graph G_{n,n_0} may be fragmented and contains no giant component of order n. Furthermore, the critical threshold of n_0 is rather sensitive to n'. For $n = 100,000$, $n' = 30$, and $n_0 = 200$, the probability $P_{\text{expandable}} \approx 0.9337$. If we reduce n' to 5, then $P_{\text{expandable}} \approx 0.5391$. This is because the decline in n' will result in the decline in the number of neighbors of a component, so does the expandable probability.

On the other hand, the probability P_0 that secrecy transfer cannot take place after the initialization phase is

$$P_0 = \left[\left(1 - \frac{n_0}{n}\right)^{n'}\right]^n = \left(1 - \frac{n_0}{n}\right)^{n \cdot n'} \approx e^{-n_0 \cdot n'}, \tag{21}$$

which is only dependent on n_0 and n'. Less n_0 or n', higher the probability P_0.

4. Heterogeneous Network

Consider graph $G_{n,r}$ of n nodes with n_0 acquaintances per node randomly selected among the nodes in the graph, we are also interested in the number of rounds needed for secrecy transfer to reach a *stable state*. It is shown in Section 3 that a necessary condition for graph G_{n,n_0} to be connected is that graphs $G_{n,p}$ and $G_{n,r}$ must be fully connected. However, the speed of the convergence of secrecy transfer depends on the values of n_0, r for given n. To gain insight, we first consider the value r and perform a simulation-based study of it. Employing a uniform random generator, we position $n = 500$ nodes in a square planar region of $500 \times 500\,\text{m}^2$, following our deployment from Section 3. For each random topology, we estimate the speed of the convergence of secrecy transfer as the number of rounds that it needs to perform to reach its stable state. At each round, each pair of adjacent nodes in the graph G_{n,n_0} employ secrecy transfer to try to get connected. If there is no new edge is added in this round, secrecy transfer reaches its stable state and terminates. We observe from Figure 4 that, as the value r increases, the stable state is reached with a faster speed, and for value n_0, the number of rounds reaches its peak when n_0 approximates to its connectivity threshold.

Conventionally, a wireless network consists of some nodes as supernodes with greater communication radius than normal nodes. The use of these supernodes lead to important characteristics of complex networks [12]: a small average shortest path length between all nodes, and a high-cluster coefficient, which help us saving network resources, avoiding excessive communication, and reducing the time to data delivery. Figure 5 depicts plots of secrecy transfer with $n = 500$ nodes deployed over a $500 \times 500\,\text{m}^2$ field, $n_0 = 20$, and $r = 35\,\text{m}$, among them there are 25 supernodes (squares in the figures) with a larger communication radius $R = 150\,\text{m}$ and only bidirectional links are considered.

From the simulation results illustrated in Figure 6, we conclude that, compared to the homogeneous network case, for a heterogeneous network with supernodes, as the radius R of supernodes grows, the value of n_0 required to maintain

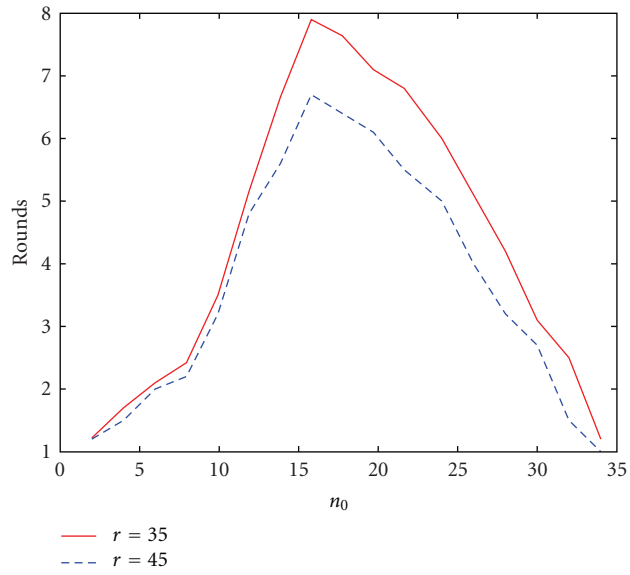


FIGURE 4: Value n_0 versus. number of rounds of secrecy transfer for various values of the radius r.

connectivity of graph G_{n,n_0} decreases, the speed of the convergence of secrecy transfer accelerates. Hierarchically, supernodes can form a higher layer, while normal nodes constitute a lower layer of the network. An implication of a heterogeneous network is that it has better performance with regard to improving energy, power and topology control, scalability, and fault-tolerance and routing efficiency.

One of the most important properties of a network is the degree distribution, or the fraction $P(k)$ of nodes having k connections (degree k). Although the degree distribution alone is not enough to characterize the network, it has great influence on the network's structure and behavior. A well-known result for Erdös-Rényi random graph is that the degree distribution is Poissonian, $P(k) = e^{-\lambda}\lambda^k/k!$, where $\lambda = \langle k \rangle$ is the average degree. For many real networks, such as the Internet, WWW, citations of scientific articles, airline networks, and many more, they often exhibit a scale-free degree distribution, $P(k) = Ck^{-\gamma}$, $k = m, \ldots, K$, where $C \approx (\gamma - 1)m^{\gamma-1}$ is a normalization faction, and m and K are the lower and upper cutoffs for the degree of a node, respectively. A scale-free network with $2 < \gamma < 3$ and N nodes have diameter $d \sim \ln\ln N$ and can be considered as "ultra small-world" network. In fact, the diameter of network is relevant in many fields regarding communication and computer networks, such as routing, searching, and transport of information. All these processes become more efficient when the diameter is smaller.

Intuitively, compared to homogeneous networks, a heterogeneous network with supernodes has a degree distribution different from Poisson distribution. Next, we discuss the degree distribution of heterogeneous networks. Let n nodes and s supernodes be distributed uniformly and independently at random in a square of area 1, $[0,1]^2$, the communication radii of nodes and supernodes are r and R ($r < R$), respectively, and only bidirectional links are taken into considerations.

For a heterogeneous network, there are two degree distributions, one for each type of nodes. For normal nodes with radii r, the degree distribution $P_n(k)$ is Poissonian,

$$P_n(k) = \frac{e^{-\lambda_n}{\lambda_n}^k}{k!}, \quad \text{where } \lambda_n = \langle k \rangle = \pi(n+s)r^2, \tag{22}$$

whereas the degree distribution of supernodes is

$$P_s(k) = \frac{e^{-\lambda_s}{\lambda_s}^k}{k!}, \quad \text{where } \lambda_s = \langle k \rangle = \pi(nr^2 + sR^2). \tag{23}$$

Therefore, the degree distribution $P_{2-h}(k)$ of the heterogeneous network is

$$P_{2-h}(k) = \frac{n}{n+s}P_n(k) + \frac{s}{n+s}P_s(k). \tag{24}$$

From the degree distribution $P_{2-h}(k)$ derived above, we depict and compare it with two different networks. In Figure 7(a) we show the degree distribution of a heterogeneous network with $n = 9,000$, $s = 1,000$, $r = 0.01$, and $R = 0.1$. In Figure 7(b), we compare $P_{2-h}(k)$ with power law $P(k) = k^{-2}$ and Poisson distribution $P(k) = \lambda^k e^{-\lambda}/k!$, where $\lambda = (n+s)\pi r^2$. As the figures shown, the degree distribution of a heterogeneous network with two peaks is different from a Poisson distribution and right-skewed to a power law. The results imply that the behavior of a heterogeneous network has some characteristics of scale-free network, such as small diameter.

Now consider the placement of nodes with more types. Suppose that the network contains t types of nodes, denoted as $T_1, T_2, \ldots, T_t$. For nodes of type T_i, the number of nodes $|T_i|$ in it and node's communication radius r_i satisfy the conditions,

$$\begin{aligned} &r_1 < r_2 < \cdots < r_t, \\ &|T_1| > |T_2| > \cdots > |T_t|. \end{aligned} \tag{25}$$

For simplicity, let n denote the total number of nodes in the network, and

$$|T_1| = \frac{n}{2}, \qquad |T_2| = \frac{n}{2^2}, \ldots, \qquad |T_t| = \frac{n}{2^t}. \tag{26}$$

It is clear that when $n \to +\infty$ and $t \to +\infty$,

$$|T_1| + |T_2| + \cdots + |T_t| = n \cdot \sum_{i=1}^{t} \frac{1}{2^i} \longrightarrow n. \tag{27}$$

Recall that the degree of each type of nodes has a Poisson distribution with different mean value. To derive the degree distribution $P_{t-h}(k)$ of the network, we first determine the degree distribution $P_i(k)$ of nodes of type T_i,

$$P_i(k) = \frac{e^{-\lambda_i}\lambda_i^k}{k!}, \quad (i = 1, \ldots, t), \tag{28}$$

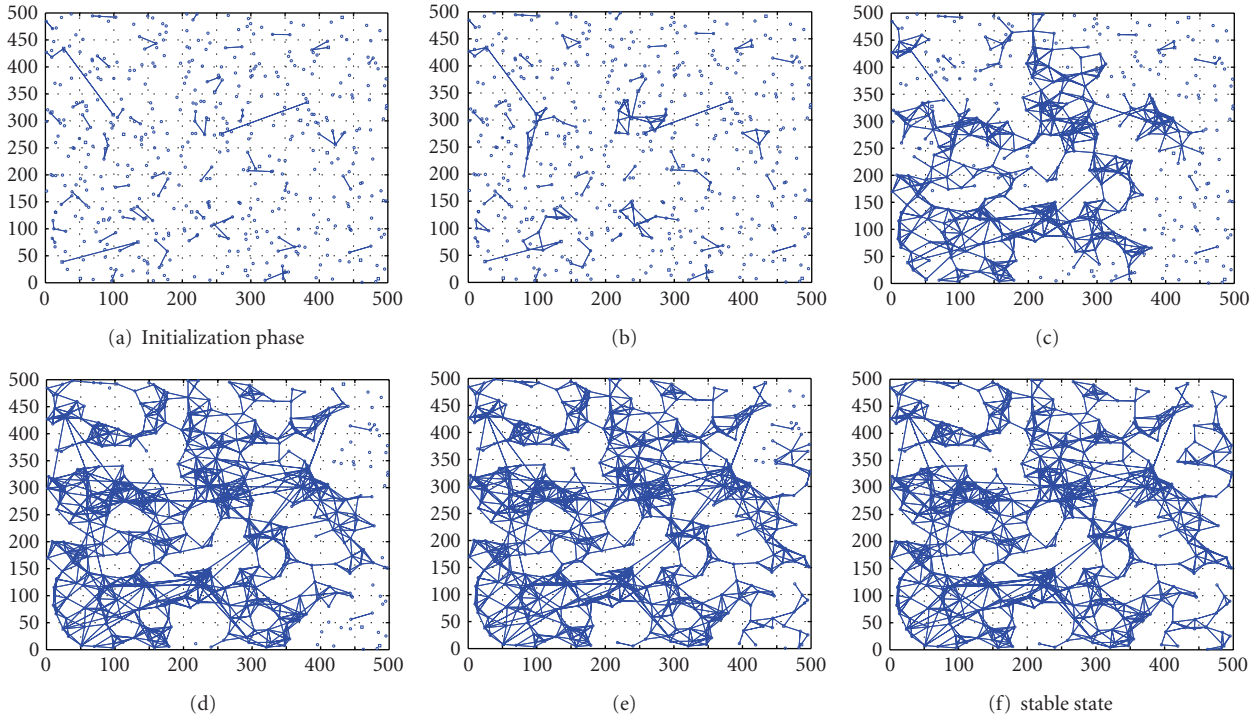


Figure 5: Secrecy transfer process in a heterogeneous network, $n = 500$, $s = 25$, $r = 35$ m, and $R = 150$ m.

where,

$$\begin{aligned}
\lambda_1 &= \pi r_1^2 n,\\
\lambda_2 &= \pi \frac{r_1^2 + r_2^2}{2} n,\\
\lambda_3 &= \pi \frac{2r_1^2 + r_2^2 + r_3^2}{4} n,\\
\lambda_4 &= \pi \frac{4r_1^2 + 2r_2^2 + r_3^2 + r_4^2}{8} n,\\
\lambda_5 &= \pi \frac{8r_1^2 + 4r_2^2 + 2r_3^2 + r_4^2 + r_5^2}{16} n,\\
&\cdots\cdots.
\end{aligned} \tag{29}$$

Therefore, the degree distribution of the heterogeneous network with t types of nodes is

$$\begin{aligned}
P_{t-h}(k) &= \frac{1}{2}P_1(k) + \frac{1}{4}P_2(k) + \cdots + \frac{1}{2^t}P_t(k)\\
&= \frac{1}{k!}\left(\frac{1}{2}e^{-\lambda_1}\lambda_1^k + \frac{1}{4}e^{-\lambda_2}\lambda_2^k + \cdots + \frac{1}{2^t}e^{-\lambda_t}\lambda_t^k\right).
\end{aligned} \tag{30}$$

For $P_{t-h}(k)$ is complicated, we present some numerical results on it. Figure 8 shows the degree distribution $P_{t-h}(k)$ for $r_1 = 0.01$, $r_2 = 0.03$, $r_3 = 0.05$, $r_4 = 0.07$, $r_5 = 0.09$, $r_6 = 0.11$, $r_7 = 0.13$, and $r_8 = 0.2$. As expected, the degree distribution of the heterogeneous network approaches power law $P(k) \propto k^{-2}$. This implies that heterogeneous network maintains some statistical properties of a scale-free network. Therefore, it is plausible that the convergence speed of secrecy transfer in a heterogeneous network is faster than that in a homogeneous network.

5. Implementation of Secrecy Transfer

In this section, we elaborate the implementation method of secrecy transfer. The method contains three phases: the initialization phase, the secrecy transfer phase, and the update phase. To implement secrecy transfer efficiently, we use Bloom Filter [13] for membership queries.

Bloom Filter. A Bloom Filter is a popular data structure used for membership queries. It represents a set $S = [s_1, \ldots, s_n]$ using k independent hash functions $h_1, \ldots, h_k$ and a string of m bits, each of which is initially set to 0. For each $s \in S$, we hash it with all the k hash functions and obtain their values $h_i(s)$ $(1 \le i \le k)$. The bits corresponding to these values are then set to 1 in the string. To determine whether an item s' is in S, bits $h_i(s')$ are checked. If all these bits are 1s, s' is considered to be in S.

Since multiple hash values may map to the same bit, Bloom Filter may yield false positives. That is, an element is not in S but its bits $h_i(s)$ are collectively marked by elements in S. If the hash is uniformly random over m values, the probability that a bit is 0 after all the n elements are hashed and their bits marked is $(1 - 1/m)^{kn} \approx e^{-kn/m}$. Therefore, the probability for a false positive is $(1 - (1 - 1/m)^{kn})^k \approx (1 - e^{-kn/m})^k$. The right hand side is minimized when $k = (m/n) \ln 2$ in which case it becomes $(1/2)^k = (0.6185)^{m/n}$.

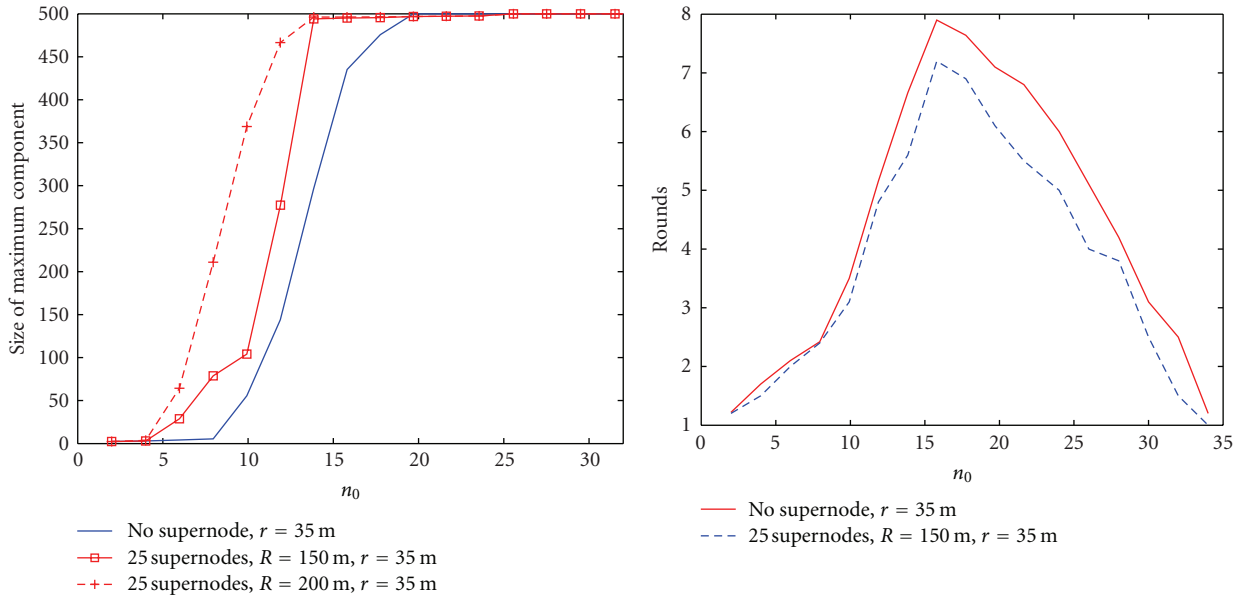


Figure 6: The maximum component and rounds of secrecy transfer in heterogeneous networks.

5.1. Initialization Phase. We first generate a random graph with n nodes and n_0 links per node. For each link a secret key is assigned to it. Each node stores the ID of its neighbors and the corresponding secret key between them. For instance, if node i has n_0 neighbors $i_1, \ldots, i_{n_0}$, it constructs an *acquaintanceship set*

$$A_i = \left\{ (i_1, K_{i,i_1}), \ldots, \left(i_{n_0}, K_{i,i_{n_0}}\right) \right\}, \tag{31}$$

where K_{i,i_1} is the assigned secret key between node i and its neighbor i_1.

After that nodes are deployed randomly over a field.

5.2. Secrecy Transfer Phase. Suppose two adjacent components, C_A and C_B, have, respectively, m_1 and m_2 nodes, nodes $A \in C_A$ and $B \in C_B$ are adjacent. For component C_A, a *component head* (at first after initialization phase, each node is a component head of its own since all nodes are isolated. After several rounds of secrecy transfer process, some large components emerge. To reduce the communication cost, a node is selected to be a component head according to its centrality in the component. To simply the procedure, the node with the highest degree is chosen to be the component head) is selected. He stores all the ID of nodes belonging to the component C_A in a *component member set*

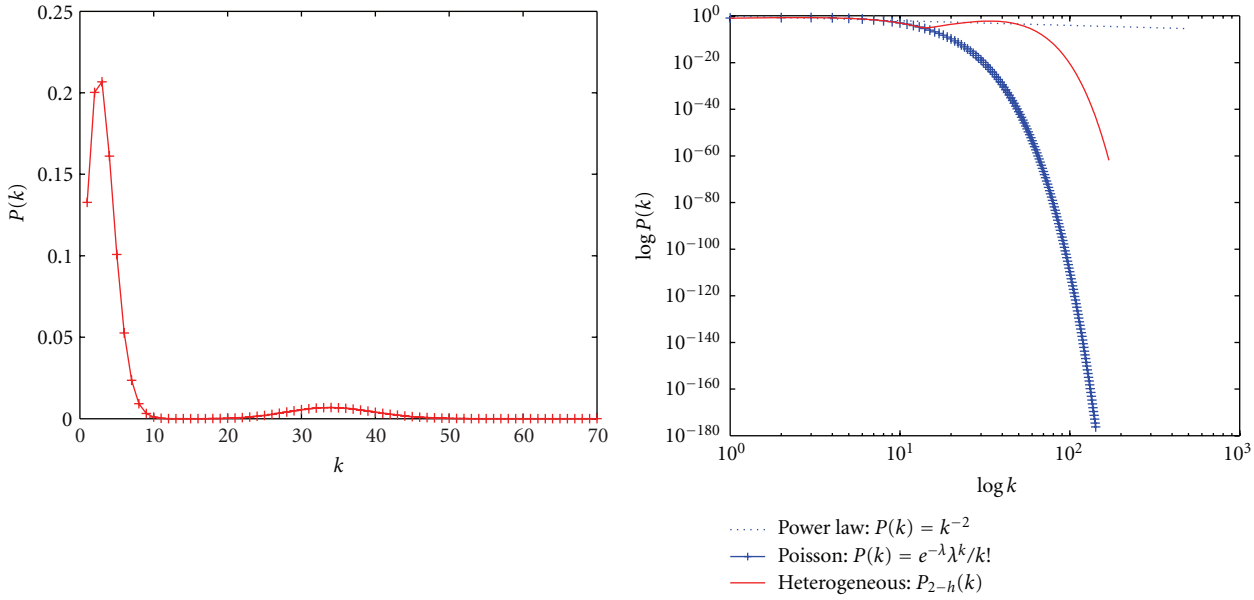
$$\mathrm{CM}_{C_A} = \{a_1, \ldots, a_{m_1}\}, \tag{32}$$

where $a_i \in C_A$.

Each node stores a Bloom Filter BF_{C_A} which contains all the nodes in the acquaintance circle of C_A. That is, the nodes in C_A and the acquaintances of node i for all $i \in C_A$. If an adjacent node k is added to C_A, the Bloom Filter BF_k of node k is inserted into BF_{C_A}, that is, a new Bloom Filter $\mathrm{BF}_{C'_A}$ for the new component $C'_A = C_A + k$ is created, that is, $\mathrm{BF}_{C'_A} = \mathrm{BF}_{C_A} + \mathrm{BF}_k$.

If two components C_A and C_B get connected and melt into a larger component C_{AB}, a new Bloom Filter of component C_{AB}, $\mathrm{BF}_{C_{AB}} = \mathrm{BF}_{C_A} + \mathrm{BF}_{C_B}$, is created and stored in nodes of C_{AB}. To further improve the performance, not all nodes in C_A or C_B need to update its BF_{C_A} or BF_{C_B} to $\mathrm{BF}_{C_{AB}}$, only nodes whose neighbors are not all connected to them need to store the updated Bloom Filter $\mathrm{BF}_{C_{AB}}$ of the new component C_{AB}. As depicted in Figure 9, C_A and C_B melt into a larger component C_{AB}, an isolated node E is adjacent to nodes A, B, F, and G. After C_A and C_B get connected, only nodes A, B, F, and G in C_{AB} have unconnected neighbor. Therefore, they need to store the new $\mathrm{BF}_{C_{AB}}$ and will broadcast it later.

Next, we give an overview of the operations of secrecy transfer. In general, the operation of secrecy transfer is initiated by a new created component. Let C_A be a new component that has "swallowed" node H, nodes A and F have already updated their BF_{C_A} (to insert the ID of node H into it), and let C_B be an adjacent component of C_A. After that, nodes A and F broadcast BF_{C_A} to their adjacent nodes B and E. On receiving the BF_{C_A} from component C_A, node B sends a query message containing BF_{C_A} to the component head of C_B, say node I, where the component member set of C_B, $\mathrm{CM}_{C_B} = \{b_1, \ldots, b_{m_2}\}$, is stored. The component head I then determines whether the nodes in set CM_{C_B} are in the Bloom Filter BF_{C_A}. If a node, say $D \in \mathrm{CM}_{C_B}$, is found in the Bloom Filter BF_{C_A}, node I answers node B by sending D to it. Node B then tells A that there is a node $D \in C_B$ belonging to the acquaintance circle of component C_A. After that, nodes A broadcasts a query message with the ID of node D in component C_A. Each node in C_A verifies whether node D belongs to its acquaintanceship set. As illustrated in Figure 9,



(a) (b)

FIGURE 7: Degree distributions for different networks. The horizontal axis is node degree k (or $\log k$), and the vertical axis is the probability distribution $P(k)$ (or $\log P(k)$) of degrees, that is, the fraction of nodes that have degree greater than or equal to k. The network shown in (a) is a heterogeneous network with two types of nodes, three networks (of power law, heterogeneous, and Poisson distribution) are depicted in (b) with a logarithmic degree scale.

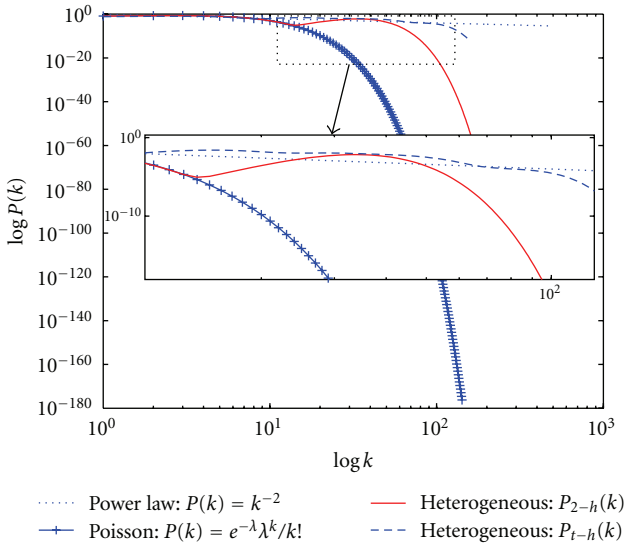


FIGURE 8: Degree distributions.

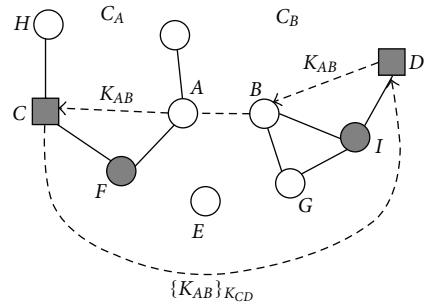


FIGURE 9: Secrecy transfer phase.

if the acquaintanceship set of node $C \in C_A$ contains node D, that is, $A_C = \{\ldots, (D, K_{CD}), \ldots\}$, the node C transmits a response message (C, D) to node A. After obtaining the acquaintance node pair (C, D) from C, node A knows that nodes $C \in C_A$ and $D \in C_B$ are acquaint with each other (they have a shared key K_{CD}). Now nodes A and B can establish a secret key K_{AB} as mentioned in Section 2.

5.3. Updated Phase. After the secret key K_{AB} between nodes A and B is established, two components become a larger component C_{AB}, we then should update the acquaintance circle of C_{AB} for nodes who have unconnected neighbors. A new component head is also need to be selected according to the degree distribution of nodes in C_{AB}. As to the network in Figure 9, if node I is the new component head of C_{AB}, the component member set is updated to be

$$\mathrm{CM}_{A_B} = \mathrm{CM}_{C_A} + \mathrm{CM}_{C_B} = \{a_1, \ldots, a_{m_1}, b_1, \ldots, b_{m_2}\}. \quad (33)$$

Finally, if nodes have updated their Bloom Filter $\mathrm{BF}_{C_{AB}}$, they broadcast the new $\mathrm{BF}_{C_{AB}}$ to their neighbors to find chances for new links. Recursively, this procedure is applied until there is no node has updated its Bloom Filter.

5.4. Security Analysis. As discussed in Section 2, secrecy transfer is robust against eavesdrop attack, for each edge is added via the existing trustiness between nodes. In this subsection, we study the resilience of secrecy transfer against the node compromise attack. Let n_c denote the number of nodes that have been captured. Suppose the compromised nodes are independently and random distributed among the entire deployment region.

Theoretically, as depicted in Figure 9, if any node in the paths $A - F - C$ and $D - I - B$ is compromised, the key K_{AB} between nodes A and B is not secure. Suppose that the length of two paths are l_1 and l_2, respectively. It is easy to estimate the probability that a new established key K_{AB} is compromised as the following:

$$P\{K_{AB} \text{ is compromised}\} = 1 - \frac{\binom{l_1+l_2}{n-n_c}}{\binom{l_1+l_2}{n}}, \tag{34}$$

where n is the number of node in the network.

Unfortunately, even if all nodes in two paths are not compromised, the key K_{AB} may be unsecure. For instance, let a path from A to C be $A - H_1 - H_2 - H_3 - H_4 - C$, and all nodes in the path have not been compromised. Node A sends K_{AB} to H_1 by sending $\{K_{AB}\}_{K_{AH_1}}$, H_1 then transmits $\{K_{AB}\}_{K_{H_1H_2}}$ to node H_2 until K_{AB} reaches the last node C. If $K_{AH_1}, K_{H_1H_2}, \ldots, K_{H_4C}$ are not compromised, K_{AB} is still secure after it is transmitted across the path. However, if a key, such as $K_{H_1H_2}$, is compromised, an adversary may eavesdrop on the communication flows between nodes H_1 and H_2 to obtain $\{K_{AB}\}_{K_{H_1H_2}}$, thus K_{AB} is leaked.

In general, if there are compromised nodes in the network, any key established by secrecy transfer between two neighbors H_1 and H_2 may be unsecure unless nodes H_1 and H_2 are acquaint with each other initially. For any pair of acquaintance nodes, the secret key between them is preloaded before the network is deployed and is considered unbreakable (unless the node is compromised). As to any key established by secrecy transfer, compromised nodes may degrade its security since lots of nodes are involved in the process of the negotiation of a new link key.

In order to set up a more secure channel between nodes A and C, it is reasonable to use the acquaintanceship set of nodes. Suppose in a path $A - H_1 - H_2 - H_3 - H_4 - C$, ($A$, H_3), (H_1, H_3), and (H_1, C) are three pair of acquaintances. To send a secret key K_{AB} to C, node A can send $\{K_{AB}\}_{K_{AH_3}}$ to H_3, H_3 then sends $\{K_{AB}\}_{K_{H_1H_3}}$ to H_1. At last, node C can get $\{K_{AB}\}_{K_{H_1C}}$ from H_1. The advantage of this method is that all communications are encrypted with predistributed keys. If nodes A, C, H_1, and H_3 are not compromised, the key K_{AB} is secure after the transmission. However, such a secure logical path in a set of nodes may not exist. For a path of l nodes, their initial acquaintanceship can be viewed as a random graph $\widehat{G}_{n,p}$, where $n = l$ and $p = n_0/n$. If $\widehat{G}_{n,p}$ is connected, a logical path exists.

If an adversary is not present at the network before secrecy transfer has completed, or it takes more time than a secure interval to compromise nodes, the communication links established by secrecy transfer are secure; otherwise, undetected malicious nodes may degrade the security of secrecy transfer and jeopardize the network. In [14], authors investigated the potentially disastrous threat of node compromise spreading (via communication and preestablished mutual trust) in wireless sensor networks and proposed an epidemiological model to investigate the probability of a breakout. This model can be adapted to analyze the spread of malicious behavior of compromised nodes in the process of secrecy transfer. But how to design efficient countermeasures is still unknown.

5.5. Storage Overhead. A node, say i, needs to store

(1) an *acquaintanceship set*

$$A_i = \left\{(i_1, K_{i,i_1}), (i_2, K_{i,i_2}), \ldots, \left(i_{n_0}, K_{i,i_{n_0}}\right)\right\}, \tag{35}$$

where $i_1, i_2, \ldots, i_{n_0}$ are the acquaintances of node i, $K_{i,i_1}, K_{i,i_2}, \ldots, K_{i,i_{n_0}}$ are the corresponding secret keys with each acquaintance, respectively.

(2) n' secret keys established with its neighbors,

(3) a Bloom Filter BF_{C_A} ($i \in C_A$).

For a component head j ($j \in C_A$), in addition to the secret values a normal node stores, it also stores a *component member set*

$$\mathrm{CM}_{C_A} = \{a_1, \ldots, a_m\}, \tag{36}$$

where $a_1, \ldots, a_m$ are the members of component C_A, m is the cardinality of the component.

6. Applications of Secrecy Transfer

The need for untethered distributed communications and computing continues to drive advances in mobile communications and wireless networking. To serve this purpose, wireless sensor networks have been envisioned to consist of groups of lightweight sensor nodes that may be randomly and densely deployed to observe data within a physical region of interest [15].

In many applications, such as target tracking, battlefield surveillance, and intruder detection, sensor networks are often deployed in hostile environments. To protect the sensitive data, secret keys should be established to achieve data confidentially, integrity, and authentication between communicating parties [16–19]. The first practical key predistribution scheme for sensor network is random key predistribution scheme introduced by Eschenauer and Gligor [7]. Its operation can be briefly described as follows. A random pool S of keys is selected from the key space. Each sensor node receives a random subset of m keys (key ring) from the key pool before deployment. Any two nodes able to find one common key within their respective subsets can use that key as their shared secret to initiate communication. Moreover, the network can be viewed as a random graph $G_{n,p}$, each edge added if two adjacent nodes can find one common key within their key rings. A major advantage of this scheme is the exclusion of base stations in key management, but a fixed number of compromised

sensors causes a fraction of the remaining network to become insecure. Successive sensor captures enable the adversary to reveal network key pool and use them to attack other sensors. In addition, the storage overhead is still high for lightweight sensor nodes. As mentioned previously, secrecy transfer can turn a random graph to a secure random geometric graph. If secrecy transfer is applied with the random key predistribution scheme, the storage overhead of nodes is lower and it can achieve better resilience against node capture attack.

In [20], an asymmetric key predistribution scheme AKPS for sensor networks was proposed. Each nodes only store two secret values initially, a large amount of storage is shifted to keying material servers (KMS). Since AKPS needs to provide public keying material for any pair of nodes, a KMS should store $\binom{n}{2}$ public keying material for a network of n nodes. Roughly speaking, AKPS is not viable for arbitrary large network. We find that, if secrecy transfer is used, a KMS does not need to be preloaded with $\binom{n}{2}$ public keying material. Specially, suppose n^* out of $\binom{n}{2}$ public keying material are randomly picked, the initial probability that two arbitrary sensors can establish a secret key is $p = n^*/\binom{n}{2} = 2n^*/n(n-1)$, which means that, any nodes has $n_0 = n \times p = 2n^*/(n-1)$ "acquaintances" on average. As before, if n_0 is larger than the connectivity threshold, we can repeat the construction process of secrecy transfer to get a connected graph G_{n,n_0} which will guarantee that any pair of adjacent nodes can establish secret keys.

7. Conclusion

This work presented a secrecy transfer algorithm which is directly based on the idea that networks form primarily by people introducing pairs of their acquaintances to one another. The resulting network, showing both properties of random graph and random geometric graph, cannot only model the introduction process in social networks, but also be used to protect the network.

Acknowledgments

This work is supported by Program for Changjiang Scholars and Innovative Research Team in University (IRT1078), the Key Program of NSFC-Guangdong Union Foundation (U1135002), Major national S&T program (2011ZX03005-002), National Natural Science Foundation of China (61173135, 61100230, 61100233), and the Natural Science Basic Research Plan in Shaanxi Province of China (2011JM8004). It is also supported partly by Mobile Network Security Technology Research Center of Kyungpook National University, Republic of Korea.

References

[1] B. Bollobás, *Random Graphs*, Cambridge University Press, 2nd edition, 2001.

[2] M. Penrose, *Random Geometric Graphs*, Oxford Studies in Probability, Oxford University Press, Oxford, UK, 2003.

[3] M. Altmann, "Susceptible-infected-removed epidemic models with dynamic partnerships.," *Journal of Mathematical Biology*, vol. 33, no. 6, pp. 661–675, 1995.

[4] C. Bettstetter, "On the minimum node degree and connectivity of a wireless multihop network," in *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC '02)*, pp. 80–91, ACM, June 2002.

[5] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388–404, 2000.

[6] R. Di Pietro, L. V. Mancini, A. Mei, A. Panconesi, and J. Radhakrishnan, "Redoubtable sensor networks," *ACM Transactions on Information and System Security*, vol. 11, no. 3, article 13, 2008.

[7] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pp. 41–47, November 2002.

[8] Z. Liu, J. Ma, Q. Pei, L. Pang, and Y. Park, "Key infection, secrecy transfer, and key evolution for sensor networks," *IEEE Transactions on Wireless Communications*, vol. 9, no. 8, pp. 2643–2653, 2010.

[9] R. Anderson, H. Chan, and A. Perrig, "Key infection: smart trust for smart dust," in *Proceedings of the 12th IEEE International Conference on Network Protocols (ICNP '04)*, pp. 206–215, Berlin, Germany, October 2004.

[10] P. Erdös and A. Rényi, "On the evolution of random graphs," *Publication of the Mathematical Institute of the Hungarian Academy of Sciences*, vol. 5, pp. 17–61, 1960.

[11] B. Bollobás, "A probability proof of an asymptotic formula for the number of labelled regular graphs," *European Journal of Combinatorics*, vol. 1, pp. 311–316, 1980.

[12] M. E. J. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, no. 2, pp. 167–256, 2003.

[13] B. H, Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM*, vol. 13, no. 7, pp. 422–426, 1970.

[14] P. De, Y. Liu, and S. K. Das, "Deployment-aware modeling of node compromise spread in wireless sensor networks using epidemic theory," *ACM Transactions on Sensor Networks*, vol. 5, no. 3, pp. 1–33, 2009.

[15] N. M. Freris, H. Kowshik, and P. R. Kumar, "Fundamentals of large sensor networks: connectivity, capacity, clocks, and computation," *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1828–1846, 2010.

[16] V. C. Giruka, M. Singhal, J. Royalty, and S. Varanasi, "Security in wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 8, no. 1, pp. 1–24, 2008.

[17] J. Jeong and Z. J. Haas, "Predeployed secure key distribution mechanisms in sensor networks: current state-of-the-art and a new approach using time information," *IEEE Wireless Communications*, vol. 15, no. 4, pp. 42–51, 2008.

[18] L. Ma, X. Cheng, F. Liu, F. An, and J. Rivera, "iPAK: an in-situ pairwise key bootstrapping scheme for wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 8, pp. 1174–1184, 2007.

[19] J. C. Lee, V. C. M. Leung, K. H. Wong, J. Cao, and H. C. B. Chan, "Key management issues in wireless sensor networks: current proposals and future developments," *IEEE Wireless Communications*, vol. 14, no. 5, pp. 76–84, 2007.

[20] Z. Liu, J. Ma, Q. Huang, and S. Moon, "Asymmetric Key Predistribution Scheme for sensor networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 3, pp. 1366–1372, 2009.