Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2015, Article ID 159518, 12 pages
http://dx.doi.org/10.1155/2015/159518

Research Article

Collaborative Key Exchange System Based on Chinese Remainder Theorem in Heterogeneous Wireless Sensor Networks

Mohamed Kasraoui, Adnane Cabani, and Houcine Chafouk

Instrumentation, IT and Systems Department, IRSEEM, 76801 Saint-Étienne-du-Rouvray, France

Correspondence should be addressed to Mohamed Kasraoui; mohamed.kasraoui@gmail.com

Received 5 December 2014; Revised 12 March 2015; Accepted 17 March 2015

Academic Editor: Hongxin Hu

IPv6 over Low Power Wireless Personal Area Networks (6LoWPANs), in the next generation of wireless sensor networks, represent an emerging field which can be integrated with Internet technology. Security is one of the most important issues in 6LoWPANs given the vulnerability to security threats from Internet and the inherent constraints such as bandwidth, processing power, memory, and energy. Despite limited resources, data security for nodes adds an additional heavy cost by using various security schemes. Moreover, there is no standard approach to provide the end-to-end security in 6LoWPANs. In this work, we first axed our research to propose a new end-to-end security scheme for IP enabled sensor networks to optimize battery energy consumption and then we adapted the Internet Key Exchange version 2 (IKEv2) to wireless sensor networks while taking into consideration the scarce resources. Hence a novel Cooperative Key Exchange System (CKES) has been proposed in this paper based on Chinese Remainder Theorem (CRT) which has also been implemented in NS2 to analyze energy consumption compared to IKEv2.

1. Introduction

The rapid evolution of wireless communication capabilities and wireless sensors networks made this technology a suitable solution for multiple application scenarios, such as health, military, and logistics. In the context of our project called "Port Transit of Containers (it is project supported by the "Haute-Normandie" Regional Council and the European institutions by the FEDER program)," we aim to develop a system providing tractability based on Heterogeneous Wireless Sensors Networks (HWSNs) technologies. This should trace material type inside containers, mobility of containers, delivering address, and so forth with utmost security.

In order to make the interconnection easier between compatible IPv6 hosts Internet and sensor nodes (SNs), many extensions (of protocols) have been realized such as WirelessHART, ISA 1000.11.a, and ZigBee. This allows a direct communication between sensor nodes (SNs) and compatible IPv6 hosts and makes the interconnection easier with Internet (see Figure 1).

In fact, two different models, as illustrated in Figure 1, are considered to ensure the interconnection between WSNs and Internet. In the first model, all data traffic, exchanged between SNs and Internet host, must pass through a proxy. This proxy is considered as a third member for each communication between peers by presenting an interface between the sender and the receiver. When it receives data, coming from the sender, it decrypts it and then reencrypts it before forwarding secure data to the destination. This model presents many issues such as the visibility of all data passed through the proxy, the decrease of scalability and the complexity of the interoperability between IP-WSN and external IPv6 networks. In the second model, data exchanged between WSNs and Internet host pass through one or more routers (Edge router) and SNs can be connected directly with any IP host. Compared to the previous model, here, an end-to-end security between peers is ensured without the need for proxies which allows more scalability and less complexity.

On the other hand, IETF (Internet Engineering Task Force) group has proposed a new adaptation layer called

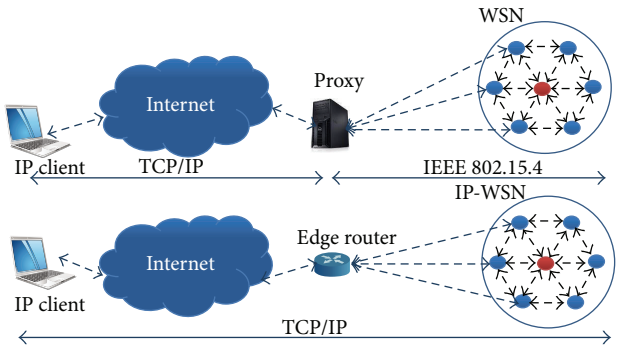

Figure 1: IP-WSN architecture.

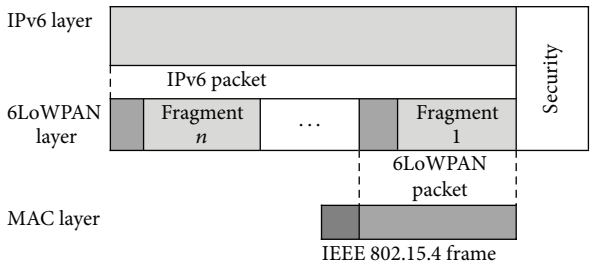

Figure 2: 6LoWPAN adaptation layer.

6LoWPAN (see Figure 2). This layer is located between the link and the network layers to allow the transmission of IPv6 packets over IEEE 802.15.4 network. It also provides header compression and packet fragmentation functionality for IPv6 packets [1]. IETF aims to optimize the resources of wireless sensors networks in terms of power consumption, memory usage, and packet size while providing an interconnection with the Internet.

However, secure and safe protocols should be developed to ensure a suitable interconnection between Internet and IP based sensor networks. Until now, the IPSec protocol has been used to secure the end-to-end communication between IPv6 hosts whereas no standardized solution has been proposed to ensure end-to-end security in 6LoWPANs, except some proposals such as TinySec [2] and MS-SPIN [3]. Hence, it would be interesting to apply the IPSec protocol to 6LoWPANs and adapt it in order to reinforce the security in constrained environments. IPSec could provide end-to-end security by encryption and authentication of all IP traffic. It also provides security to all applications by just turning on the IPsec.

The IPsec standard [4] defines two security services: an Authentication Header (AH), which provides data integrity, and authentication and an Encapsulating Security Payload (ESP), which ESP provides data confidentiality. In IPSec, the security of communication links is defined in terms of security associations (SAs) that should be established between two peers. In fact, a SA allows the establishment of an agreement (between peers) to define all security parameters that should be respected during next exchanges.

In addition, IPSec used IKEv2 protocol [5] in order to ensure a dynamic management of security associations. It is based on two databases: Security Association Database (SAD) and Security Policy Database (SPD), to store all security associations and policies for each device and it also proposes four pairs of messages to negotiate security association between peers.

These requirements make a big challenge to implement the IKEv2 on constrained environments by considering energy and bandwidth limitation. So, there is need to develop a lightweight IKE which can be easily deployed in the target network.

In this paper, we have proposed a new Cooperative Key Exchange System (CKES) based on the concept of Chinese Remainder Theorem (CRT). We present our approach for optimizing the energy consumption, the network lifetime, and the security in IP based Heterogeneous Wireless Sensors Networks (HWSNs).

The rest of the paper is structured as follows: Section 2 looks at the related works of end-to-end security algorithms used in WSNs as well as collaborative security services. Section 3 describes our proposed approach in detail. Section 4 provides the simulation results and discusses it. Finally, we discuss our conclusion and future work in Section 5.

2. Related Work

An overview of several works on security issues in IP enabled WSNs is given in this section. The related works presented in this section focus on the end-to-end security to ensure the IP communication between SNs and Internet hosts. Then we are realized that IPSec used in the Internet is more suitable compared to other proposed schemes. This standard has already demonstrated its performances in Internet environment and became mandatory component for IPv6 to ensure the end to end security [6].

2.1. Key Management in WSNs. In WSNs, the public-key cryptography was considered too heavyweight compared to other networks. Thus, symmetric key algorithms were selected as suitable solutions for the establishment of a secret key between two peers [7, 8]. However, these are essentially based on predistribution approach that needs a large memory capacity and is considered as not suitable solution in a large scale Wireless Sensor Network. In order to cope with the memory constraint, Eschenauer and Gligor in [9] proposed the use of the random key predistribution (RKP). In this approach, which is adopted by many authors [10, 11], the number of keys is reduced where each node preloaded with a subset of keys is generated from a global pool of keys. Then, any peer of nodes will share a key with a large probability while it still exists that two nodes cannot establish a key with a small probability.

To bridge this gap, deterministic approaches were developed [12–14]. All this solutions could solve the problem of distributing the secret keys. Among them, we find the mGKE (group based key establishment scheme) [12], used to establish unique pairwise keys in connected networks regardless of sensor density or distribution. Chan and Perrig [15] also proposed the Peer Intermediaries for Key Establishment (PIKE)

Table 1: Proposals for IP communication end-to-end security on WSNs.

	SSNAIL	Sizzle	ContikiSec	SIMWSN	6LoWPAN/IPSec
Authentication	ECDSA	ECDSA	CMAC	ECDSA	AH-HMAC-SHA1-96
Key exchange	ECDH	ECDH	—	ECDH	ISAKMP/ECDH
Confidentiality	RC4	RC4	AES-CBC	AES/CCM or 3DES	AES-CBC, AES_CTR, 3DES
Key size	160	160	128	128 → 256	128 → 196
Hashing	MD5, SHA1	MD5, SHA1	—	SHA1, SHA2	SHA1, tigger (x3SHA1)
Access control	—	Gateway	—	Gateway	—
Layer	Transport	Transport	MAC/network	Network	Network
Gateway	—	Yes	—	Yes	—
End-to-end security	Yes (transport)	Yes (transport)	No	Yes (network)	Yes (network)
Attacks	MIM	MIM	Eavesdropping, replay, and DoS	Replay	MIM, spoofing (UI), DoS, replay, black hole, and MIM
Network layer	—	—	—	IPSec_TM/WSN_SM	IPSec_PAN

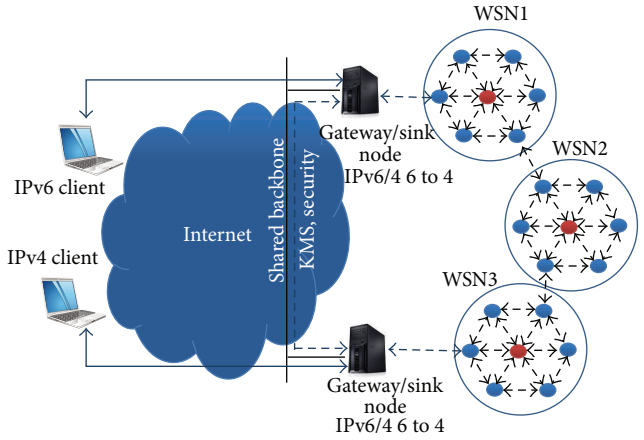

Figure 3: It illustrates the operational scenario of SIMWSN [1].

that uses other nodes in the network as trusted intermediaries to perform key establishment between neighboring nodes.

2.2. End-to-End Security in Wireless Sensor Networks (WSNs). Different techniques for IP-WSNs are proposed to ensure the end-to-end security. Granjal et al. [16] have proposed a new architecture for WSN called SIMWSN (Secure Interconnection Model for WSN). To secure the IP communication link between sensor nodes and Internet hosts, a secure gateway was introduced in this architecture. Figure 3 shows how to employ secure gateways in order to associate all WSNs. For each communication between two peers, an IPSec tunnel has to be established between the Internet host and the gateway while basic IEEE 802.15.4 security mechanisms are used between the gateway and sensor nodes. In order to manage the IPv6 addresses and support both IPv6 and 6to4 tunneling on the Internet interface, SIMWSN applies the rules defined in 6LoWPAN [17].

The concept of SIMWSN has not yet been implemented in the above mentioned work. In [18], authors have followed the idea of using network layer security in IP enabled WSNs. They have implemented a compressed IPsec for 6LoWPAN networks and they have also developed an encoding method for the AH and the ESP extension headers using the LOWPAN Next Header Compression (NHC) format introduced in RFC6282 [19].

In addition, authors have implemented a compressed version of IPSec in the Contiki OS. They have also used the concept of preshared key to establish security associations (SAs). The implementation [18] has been done using the operation mode (HMAC-SHA1-96 for AH and AES-CBC for ESP). Depending on the used protocol, the overall memory footprint of the IPSec varies from 3.9 kB to 9 kB ROM and 0.3 kB to 1.1 kB RAM.

The same concept has been proposed by Gupta et al. in [20]. Their solution, Sizzle, provides a secure communication based on SSL protocol implemented on different gateways. More generic solution, called SSNAIL (Sensor Network for an All-IP World), has been proposed in [21]. It employs the same cryptographic concept but no security gateway has been used. Moreover, Casado and Tsigas have proposed ContkiSec [22] which introduces the concept security profiles in WSNs. To summarize, a comparison study of different discussed solutions is shown in Table 1 in terms of security characteristics.

The proposed idea of Raza et al. [23] brings our attention to adapt IPSec in 6LoWPAN. In fact, some critical points have to be studied in this approach. For example, the using of conventional IKEv2 requires more resources that cannot be available in such energy-constrained wireless networks. So, further measures have to be considered in order to improve the security and make key management more efficient and adaptable to large scale WSNs. In Section 3, a novel Cooperative Key Exchange System (CKES) is presented taking inspiration from the IKEv2.

2.3. IPSec Security for 6LoWPAN. The security defined in IEEE 802.15.4 ensures encryption and authentication schemes (e.g., AES, μTESLA, and SPINS) at the link-layer. Using these algorithms, the security of data in 6LoWPAN is provided only hop-by-hop. While IPsec is mandatory with IPv6, considering the power constraints and limited

Table 2: IKE header format.

IKE_SA initiator's SPI (8 octets)				
IKE_SA responder's SPI (8)				
Nest Payload (1)	Major Version (4)	Minor Version (4)	Exchange Type (1)	Flags (1)
Message ID				
Length				

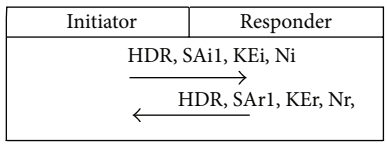

Figure 4: IKEv2 Phase 1 exchange.

processing capabilities of IEEE 802.15.4 capable devices, IPsec is computationally expensive. Meanwhile, most researchers have focused on the use of IPSEC. Varadarajan and Crosby [24] have implemented the use of new compressed IPSec headers together with 15 cryptographic algorithms typically used with IP security architectures. They analyzed the feasibility of applying IPsec and IPv6 in WSNs. The results confirm the possibility of implementing end-to-end security in IPv6 enabled WSNs to create a transition between WSNs and the Internet.

2.3.1. Basic Internet Key Exchange IKEv2. All communication channels in IPSec are secured in terms of security associations (SAs) [25]. Each one can be established between two or more entities describing all security parameters such as cryptography algorithms, key lifetimes, and key sizes. In order to provide a dynamic management of SAs, an Internet Key Exchanged version 2 (IKEv2) [5] has been introduced in IPSec. For each SA, there are four pairs of messages exchanged between peers to negotiate all the security parameters. All message headers have to be according to the IKE header format shown in Table 2. To store all security associations and policies, two databases Security Association Database (SAD) and Security Policy Database (SPD) have to be added to each device (see Figure 6).

In IKEv2, all communications consist of pairs of messages which are called "request/response pairs." To maintain a security association, two phases are required by IKEv2. Phase 1 in Figure 4 performs mutual authentication between two parts and establishes an IKE_SA.

At this stage, a secret is shared between peers, which is used for further IKEv2 exchanges to perform encryption and integrity protection.

They will agree with each other on the following parameters of their IKE_SA:

(i) Cryptographic algorithms: they are algorithms to protect IKE exchanges, Diffie-Hellman Groups (Group 1: 768-bit MODP and Group 2: 1024-bit MODP), and a pseudorandom function;

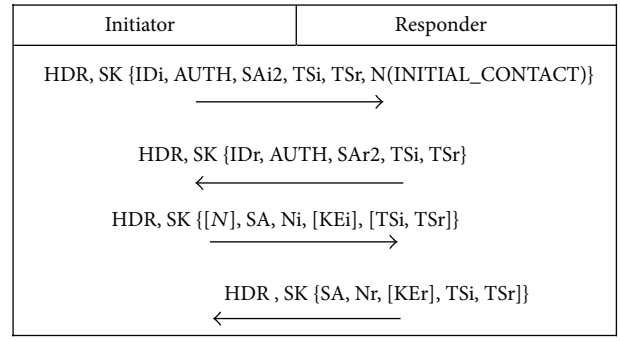

Figure 5: IKv2 Phase 2 exchange.

(ii) SKEYSEED: it is the secret keys from which all keys are derived for IKE SA (SKe: encryption key to ensure confidentiality, SKa: authentication key to ensure integrity, and SKd: derivation key master secret to compute further CHILD SAs keys);

(iii) IKE_SPI standing for IKE Security Parameter Index and uniquely identifying an IKE_SA;

(iv) lifetime: duration of IKE SAs;

(v) nonce: they are INITIATOR nonce (Ni) and RESPONDER nonce (Nr). These are randomly generated values to reinforce the security;

(vi) message ID counters: the ID counters provide antireplay for IKEv2 exchanges by increasing the ID counter by one for every emitted IKEv2 message;

(vii) IKEv2 window size: if the window size has a value of "N," it implies that there can be N unacknowledged IKEv2 requests at any given time during communication.

During Phase 2 of IKEv2 in Figure 5, the INITIATOR sends an IKE_AUTH request and the RESPONDER replies with an IKE_AUTH response.

When Phase 2 is finished, both nodes agree on the following parameters of their CHILD SAs:

(i) CHILD SA SPI: it is a 32 bits unique identifier of the CHILD SA;

(ii) IP addresses: they are source/destination IP address of the IKEv2 compliant nodes;

(iii) IPsec protocol: it is AH (Authentication Header) or ESP (Encapsulating Security Payload);

(iv) sequence number counter: it is value to control every incoming/outgoing IP packet protected with IPsec, preventing replay or unauthorized reinjection of already processed IPsec traffic;

(v) antireplay window size N: any packet with the sequence number $X + N$ is discarded, where X is the awaited sequence number;

(vi) ESP/AH information: it is encryption and/or authentication algorithms, keys, initialization values, and key lifetimes;

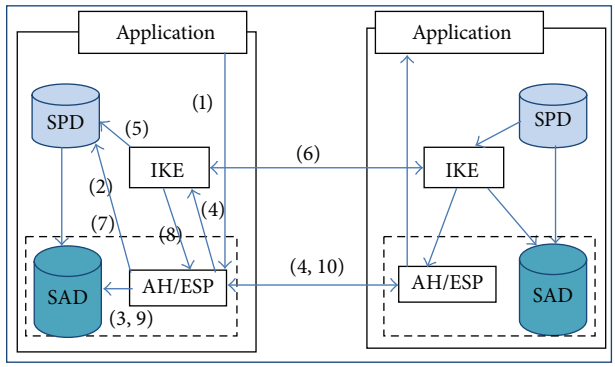

Figure 6: IKE architecture.

(vii) lifetime: it is time interval or byte count after which a SA must be replaced with a new SA (and new SPI).

Figure 6 shows the steps to maintain the SAs between two end points. Each SA is identified by using Security Parameter Index (SPI), destination address, and AH or ESP. The SPI identifies the SA in the IPSec header. During the packet transmission or reception each sensor node holds a SAD as in Step (4, 10). SAD will be used to get the information about SPI, keys, algorithms, and so forth as shown in Step (3, 9). The node looks up the corresponding security association and fetches the necessary keys to apply security to the IP packet. For the initial negotiation between peers IKE uses the SPD to define how the data should be protected shown in Step (5). Then, IKE can process the negotiation in Step (7) to request for new associations.

2.3.2. Diffie-Hellman (DH) Key Agreement Protocol. The establishment of a shared secret in IKEv2 is based on the Diffie-Hellman (DH) protocol [26]. This allows two peers A and B to securely exchange cryptographic keys. Once "p" and "g," representing, respectively, the prime number and the generator, are chosen, A and B will generate their private keys "a" and "b." Finally, public values are computed and exchanged according to functions depicted in Figure 7.

2.4. Collaborative Security Services in WSNs. Collaboration techniques are considered as efficient solutions to provide security services. At first, it has been suggested by cryptographers to deal with secret sharing. Many different schemes of secret sharing have been proposed in the literature. They can be used for any situation in which the access to an important resource has to be restricted. The main idea of these schemes is to split a secret into multiple shares and distribute the result among a set of participants. Shamir [27] and Blakley [28] are historically the first who introduced this concept based, respectively, on polynomial interpolation and linear projective geometry. Usually, a trusted party, chosen as administrator, coordinates the secret sharing scheme, but there are some other schemes conceived without using dealer to configure. All or a part of the participants can reconstruct the secret after collecting all or part of their shares called combiner. A global threshold can be used to determine

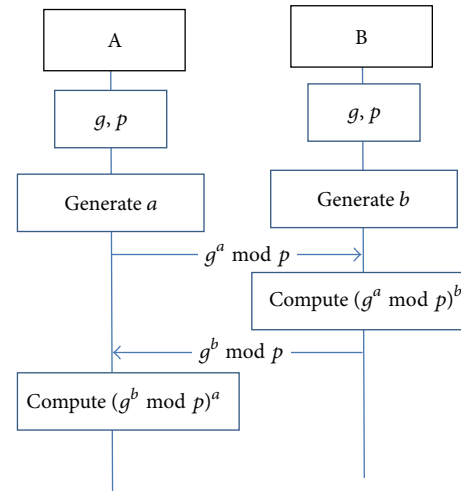

Figure 7: Diffie-Hellman key agreement.

the number of participants from which the secret can be recovered. Recently, the study of the secret sharing schemes based on the CRT (Chinese Remainder Theorem) [29, 30] has been reactivated, given to the application of these schemes in threshold cryptography [31], proxy signature [32], or cloud computing security [33].

The concept of CRT can be presented as follows: let $m_1, m_2, m_3, \ldots, m_n$ be a set of prime numbers and it should be coprime to each other. Let $a_1, a_2, a_3, \ldots, a_n$ be a set of positive integers such that $a_i < m_i \ \forall i \in [1, \ldots, N]$. Let S be a congruence system presented as in

$$S = \begin{cases} x \equiv a_1 \pmod{m_1} \\ x \equiv a_2 \pmod{m_2} \\ \vdots \\ x \equiv a_k \pmod{m_k}. \end{cases} \tag{1}$$

CRT states that a unique solution for S exists and lies between $[1, M-1]$. This unique solution is given by

$$X = \sum_{i=1}^{k} (a_i * M_i * y_i) \,(\mathrm{mod} M), \tag{2}$$

where

$$M = m_1 * m_2 * \cdots * m_k \prod_{i=1}^{k} m_i,$$
$$M_i = \frac{M}{m_i}, \tag{3}$$
$$y_i = M_i^{-1} \,(\mathrm{mod}\, m_i),$$

where y_i is determined by using extended Euclid's theorem.

As mentioned in [34], the heterogeneous WSN consists of sensor nodes with different abilities, such as various sensor types and communication/sensing range. The heterogeneity

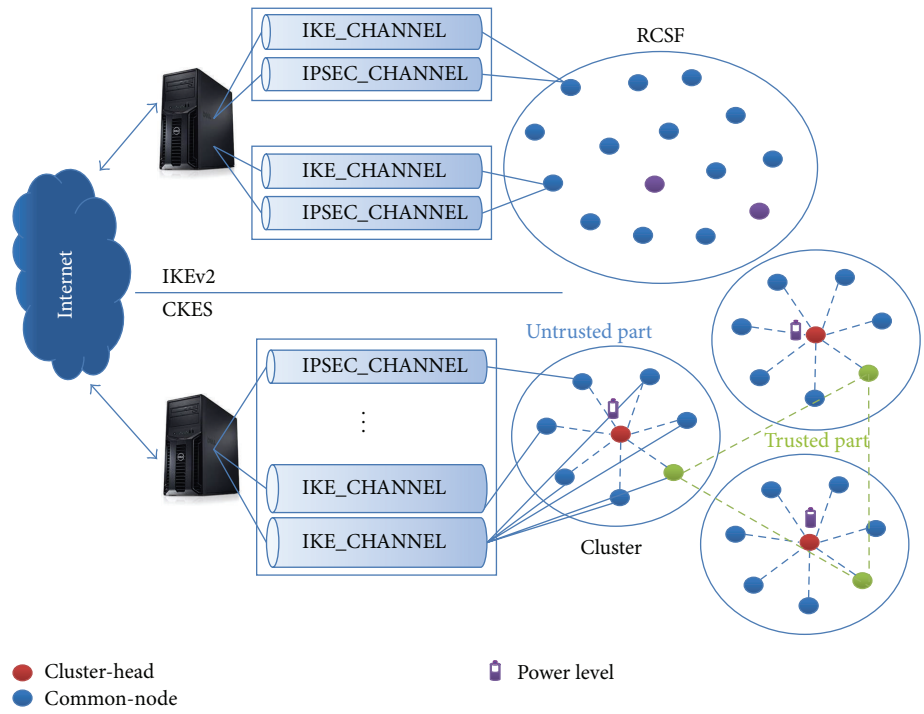

Figure 8: Cooperative Key Exchange System (CKES) architecture.

can be defined in terms of energy levels, and here we can designate two categories of nodes: "constrained node" means nodes with less energy constraint compared to normal nodes (energy-constrained nodes) and "unconstrained node" means a node that has a permanent energy source (node is able to renew their energy from solar sources, temperature differences, or vibration, etc.). So, it will be very interesting to exploit HWSN characteristics to apply some collaboration techniques.

In this section, we reviewed existing end-to-end security algorithms as well as some key management schemes used in WSNs. In addition, we addressed the applying of IPSec on WSNs and specially its key exchange protocol IKEv2, which has been detailed in Section 2.3. Finally, we provided a quick survey of collaborative security services in WSNs.

3. Proposed Architecture (New Collaborative Key Exchange System (CKES))

In this section, we present a new adaptation of IKEv2 for HWSN titled "the Collaborative Key Exchange System (CKES)." Compared to the basic IKEv2, this collaborative approach prolongs the network lifetime and improves the energy efficiency. In IKEv2, a sensor node has to process all cryptographic operations detailed in Section 2.1 that are considered as heavy computational loads in WSN. To deal with this issue, we considered that each node could request its neighbors to process all heavy cryptographic operations. We considered also that same security policies should be maintained to ensure the integrity and confidentiality during data exchange. As explained in Section 2, there are two main channels (IPSEC_CHANNEL, IKE_CHANNEL) used to establish a secure communication between peers. As shown in Figure 8, we proposed to divide the IKE_CHANNEL into subchannels between neighbor sensor nodes and one Internet host. This makes a collaborative process allowing the decrease of the computational load of energy-constrained nodes.

In next subsections, more details will be presented about our approach. We will present all assumptions and requirement that have to be involved. We will also present the method used to choose collaborative nodes as well as the different cryptography operations.

3.1. Assumptions. In CKES approach, the following assumptions have been considered.

(i) Prime numbers $(m_1, \ldots, m_n)$ are generated by a base station (BS); they have also to be coprimes.

(ii) LEACH clustering algorithm [35] is used organize sensor nodes into clusters.

(iii) Nodes are assigned with prime numbers that is generated by the Base Station (BS).

(iv) To manage the trust values of sensors network, Group-Based Trust Management Scheme (GTMS) is be used.

(v) Highly trusted nodes (HTNs) have more resources in terms of battery power and memory capacity. It supports also IKEv2 negotiations and coordinates between all collaborative nodes.

(vi) The HTN can request CH to get the power level information of all nodes.

Table 3: Cryptography algorithms costs.

Operations	Energy consumption (mJ)	Operation time (s)
RSA_Sign	359.87	12.04
Sign_Verif	14.05	0.47
DH exchange	1185	54.11

3.2. Heavyweight Cryptographic Operations. Most of the previous works on security of WSN have used Diffie-Hellman (DH) key agreement protocol and signature scheme which have been considered as heavyweight encryption algorithms. So the use of these algorithms could increase the communication delay, the power consumption, and resource utilization of constrained nodes. As the experimental example given in [36], the energy consumption on the MiCA2 platform is about 359.87 mJ using RSA-1024 for signature and 12.04 mJ using RSA-1024 for verification.

As the memory footprint, there are 4.4 kB used of ROM (Read Only Memory) space and about 1 kB used of RAM space. An example of DH exchange is given in [37]. It shows the consumption of 1185 mJ to share the DH values and compute the master key.

In CKES, the heavyweight cryptographic algorithms mentioned in Table 3 will be moved from the constraint nodes to the less constraint or unconstraint neighbor nodes.

3.3. Proposed CKES Procedure. The procedure consists of fourteen steps as described below.

Step 1. A constraint node A (initiator) requests HTN to start a key exchange session with a node B (responder). In this request, A has to mention the B's ID and the maximum number of collaborative nodes.

Step 2. The HTN multicasts the request to N less constraint nodes which are available to support heavyweight cryptographic algorithms.

Step 3. Each requested cluster member (CM) starts to update its power level, computation power, availability, and network threshold (Ct). Based on these values it accepts the CH request.

Step 4. The HTN sends "k" collaborative CMs IDs to A and as well as their CRT coefficients $(y_1 * M_1, y_2 * M_2, \ldots, y_k M_k)$ given by the CH.

Step 5. A starts to generate secret value "a" that will be used for the DH exchange and the master key computation. This value should be the sum of "k" elements $a_1, a_2, \ldots, a_k$ such that $a_i < \min(m_i)\ \forall i \in [1, \ldots, k]$ and $a = \sum_{i=1}^{k} a_i$.

Step 6. A computes X using (2), but without applying modulo M. This X value generates a solution for CRT as in (1) and it satisfies a set of congruence $X = a_i \pmod{m_i}$, where $i = 1, 2, \ldots, k$.

Step 7. A sends the solution X, the security association SAi1, and the Message Authentication Code MAC to the HTN.

Step 8. HTN multicasts X to all collaborative nodes and sends the "*IKE packet*" which consists of (HDR, SAi1, Ni, CERT_HTN) to the responder B.

Step 9. After receiving X, each collaborative node computes its own, where $a_i = X \bmod (m_i)$. Then, it calculates the DH parties as $g^{a_i} \bmod p$ and sends it to the responder B.

Step 10. To compute the A's DH public key, B makes the product of the values received from the CMs as follows:

$$\prod_{i=1}^{k} g^{a_i} \bmod p = g^{\sum_{i=1}^{k} a_i} \bmod p = g^{a} \bmod p. \quad (4)$$

Step 11. After checking the certification and computing the master key $g^{a*b} \bmod p$, node B sends $g^{b} \bmod p$ to the HTN as well as the "IKE packet" (HDR, SAr1, etc.).

Step 12. Each collaborative node computes its own master key par $g^{b*a_i} \bmod p$ and sends it to the initiator A including the HTN part.

Step 13. After receiving the master key portions, A makes the product of received value in order to compute the master key

$$\prod_{i=1}^{k} g^{b*a_i} \bmod p = g^{b\sum_{i=1}^{k} a_i} \bmod p = g^{b*a} \bmod p. \quad (5)$$

Step 14. Once the master key is calculated, peers A and B can start the second IKE exchange IKE_AUTH based on the negotiated SA and the master key.

All the details of CEKS protocol are illustrated in Figure 9. As explained above, once clusters are formed and CHs are selected using LEACH, highly trusted nodes (HTNs) will be chosen based on GTMS algorithm [38]. In fact, GTMS calculates the trust value based on direct or indirect observations. Direct observations represent the number of successful and unsuccessful interactions, and the indirect observations represent the recommendations of trusted peers about a specific node. Trust values can be changed over time as well as the residual energy level of each node. The CKES process starts by sending a "CKES message request" to the HTN. Then, collaborative nodes are chosen based on residual energy levels and trust values. The minimum required number to launch the CKES process is fixed at 3 in order to ensure a minimum level of security using the CRT.

4. Simulation

In this section, we present simulation results to evaluate the efficiency of the CKES compared with the basic IKEv2 implemented in [39]. Some improvements have been done since the last version of CKES presented in [40]. We carried out the simulations using NS2 simulator in which we have modified the energy model class to estimate the energy

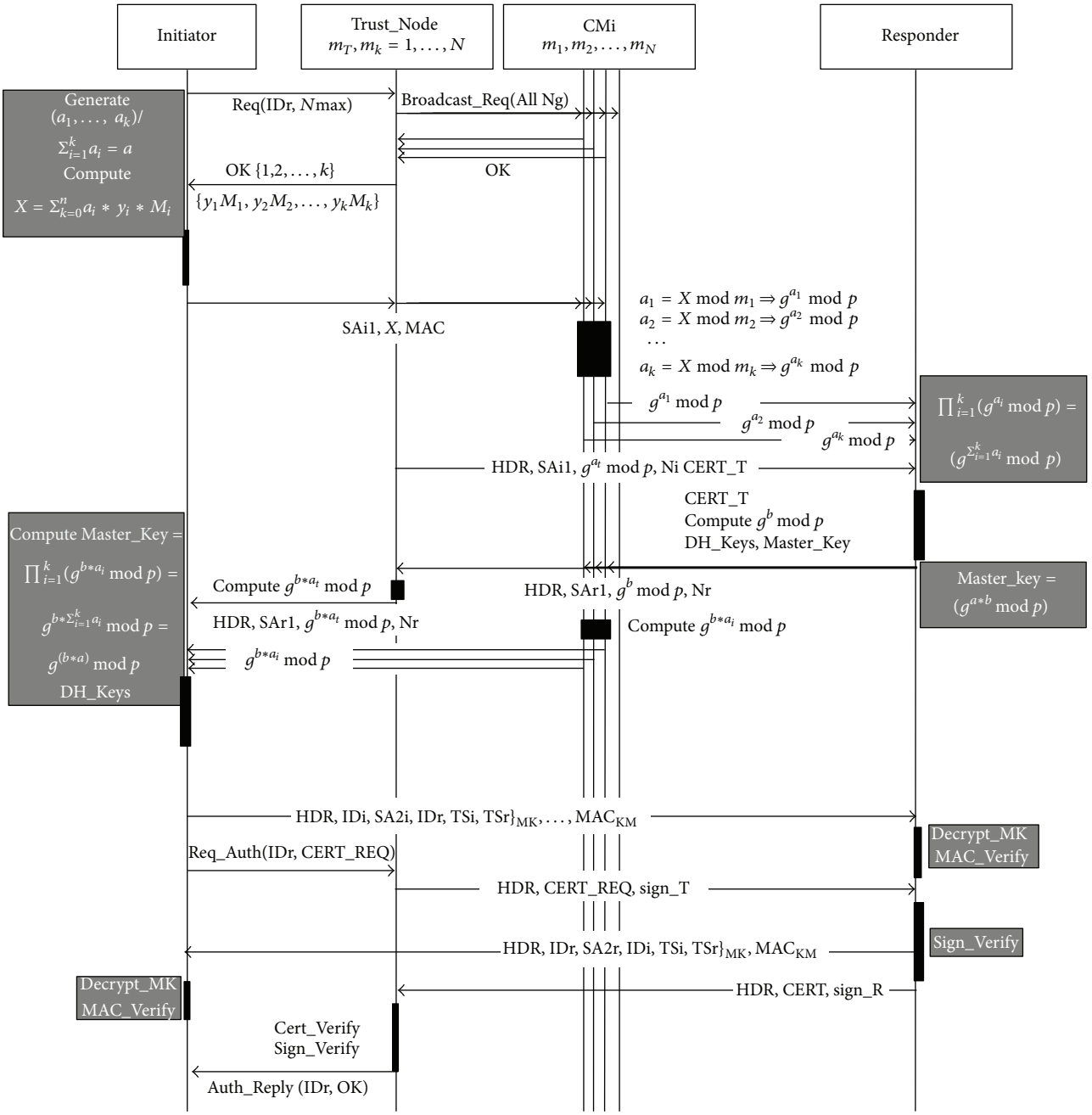

Figure 9: Cooperative Key Exchange System in HWSN.

consumption of cryptography operations of each SN as well as the communication energy costs. This model presents a linear decrement in the computation of the residual energy.

4.1. Simulation Parameters. We have considered WSN in which a security gateway connects sensors to every other Internet host using IPsec protocol. We have used a network configuration of 80 wireless sensors (60% normal nodes and 40% unconstrained nodes) and a single security gateway (SG). In fact, we defined two categories of nodes, unconstrained nodes, and normal nodes with initial energy 100 J. The simulation parameters are outlined in Table 4.

NS2 simulator is capable of producing performance measures at various protocol levels and observation points in WSNs. In our work, we have implemented the minimum IKEv2 features as described in RFC 5996 by using OpenSSL library for the cryptography operations:

(i) IKE_INIT_SA and IKE_AUTH phases for the initiator and responder nodes;

(ii) DH protocol (group 1);

(iii) RSA signature;

(iv) a simple traffic selector negotiation;

Table 4: Simulation parameter.

Parameters	Value
Simulator	NS-2 with Mannasim Patch
Traffic type	UDP
Bandwidth	250 kbps
Scenario size	400 m × 400 m
Transmission range of nodes	70 m
MAC protocol	IEEE 802.11
Routing protocol	AODV
Propagation model	Two-ray ground
Simulation time	100 s
Initial energy	100 joules

Table 5: Energy communication costs of the IKEv2 and CKES protocol.

IKEv2	Sent (bytes)	Recv (bytes)
IKE_INIT	124	124
IKE_AUTH	497	497
IKEv2 communication costs (mJ)	1.29	1.43
CEKS communication costs (mJ)	0.89	1.24

(v) one-child SAs per IKE SA;

(vi) DES3 encryption algorithm and the SHA-1 hash function.

4.2. Simulation Results

4.2.1. Communication Costs. First of all, we determined the communication energy cost at one constrained node (initiator) in order to bring out the difference between IKEv2 and CKES protocols on one node type which presents 40% of the network. The energy model is based on the reception and the transmission costs to compute the communication energy during the IKE_INIT and IKE_AUTH steps. Table 5 shows the communication energy costs of IKEv2, the proposed CKES, and the sent and received bytes. The energy communication cost is reduced by 22 percent with the use of CKES. These results show the efficiency of our CKES system especially at the constrained node level.

Secondly, we compared the energy consumed for the data communication at network level. Thus, it is important to take into consideration the communication cost added to route packets from the initiator to the responder.

Table 6 compares the network energy communication cost between the two protocols using the same routing and clustering algorithms. Results show that the communication cost using IKEv2 is lower than CKES. Due to the important number of messages exchanged between constrained and collaborative nodes, CKES caused an additional energy cost compared to the IKEv2.

4.2.2. Energy Consumption of Cryptographic Operations. In order to compare the energy consumption at the constrained nodes, we implemented the minimum of cryptographic operations needed in IKEv2 using OpenSSL library.

Table 6: Network energy communication costs of the IKEv2 and CKES protocol.

Network communication costs using IKEv2 (mJ)	Network communication costs using CKES (mJ)
42.81	96.5

Table 7: Energy computation costs of the IKEv2 and CKES protocol.

	IKEv2	CKES
Total computation costs (mJ)	252.87	4.8

Table 8: Total energy costs of the IKEv2 and CKES protocol.

	IKEv2	CKES
Total communication costs (mJ)	45.53	98.63
Total computation costs (mJ)	252.87	4.8
Total costs (mJ)	298.4	103.43

Table 7 summarizes the whole energy costs of cryptographic operations for both protocols. It shows that the computation cost using CKES is lower than using IKEv2.

As shown in the table, using CEKS, we can save around 98% of energy for the constrained nodes and we can also reduce the network energy consumption and maximize the network lifetime, whereas using IKv2, the heavyweight cryptographic algorithms were done by the initiator and consequently it consumes a lot of energy.

4.2.3. Total Energy Cost. To summarize, Table 8 provides the total energy costs of the two protocols. It shows that CKES consumes lower energy and we can reduce by 30% the total energy cost in the constrained nodes.

Unlike the IKEv2, the computation of cryptographic operations, in CKES, is happening in the collaborative nodes. This reduces the consumption of energy and the utilization of memory and this also improves the network lifetime by maintaining the same security level.

According to these results, our proposed cooperative approach is considered as a suitable key exchange system in HWSNs. In addition, we are studying the efficiency of our protocol at the network level and we aim to develop a framework based on CEKS, trust, and resource manager distributed approach.

4.2.4. No Constraints Nodes Distributions Impact. In order to make a comparison between IKEv2 and CKES, we present, in Figures 10 and 11, the overall energy costs of both systems. Figure 10 presents energy cost for IKEv2 with three colors representing the initiator computation cost, the initiator communication cost, and the network communication cost. Figure 11 presents our proposal CKES with the same colors and shown metrics (costs).

As shown in the Figure 11, we can identify one threshold at 5% of no constrained nodes using the CKES system. This is

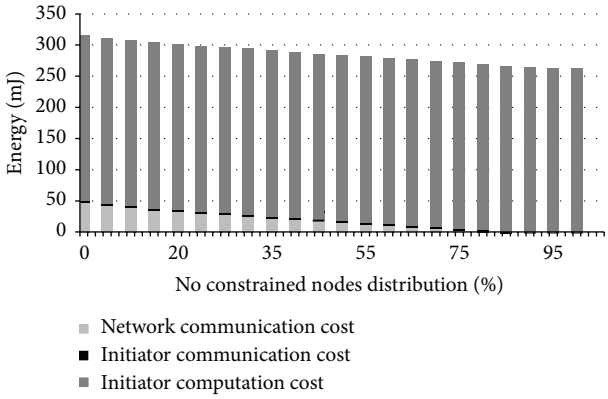

Figure 10: Overall energy consumption for IKEv2.

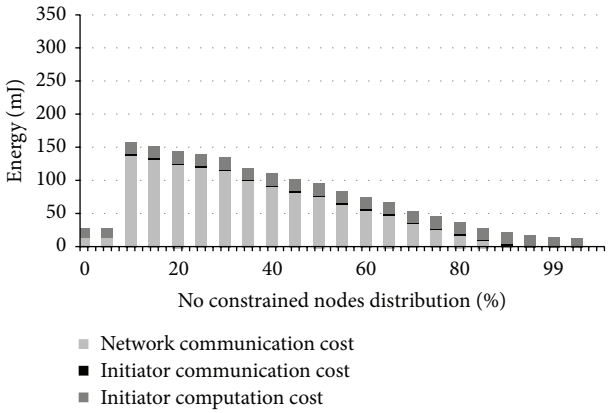

Figure 11: Overall energy consumption for CKES.

due to the minimum number of collaborative nodes required to launch the CKES process. In fact, we fixed the number of collaborative nodes at 3 ($c = 3$) in all simulation scenarios in order to avoid a minimum security level. We can see that the energy consumption cost increases significantly from 20 mJ to 150 mJ at 5% (threshold). After that, the energy cost decreases when the no constrained nodes number increases. Regarding the initiator consumption energy, the cost is still higher using IKEv2 than CKES in all distributions. Indeed, the initiator makes all of its own computational loads using the IKEv2 whereas it delegates a lot of load to constrained nodes when it uses the CKES processes.

As the network communication, its cost decreases proportionally with the constrained node number in both security systems (IKEv2 and CKES). According to results, the communication cost in CKES is more than the double of IKEv2 communication cost in all distributions. This is caused by the increase of the exchanged messages number between the initiator and collaborative nodes. In fact, increasing the number of no constraint nodes makes more chance to find a no constraint neighbor node of the initiator. This can significantly reduce the hops number between initiator and the collaborative nodes which allows making more power savings in the entire network.

Table 9: Network lifetime of the IKEv2 and CKES protocol.

	Network lifetime (rounds)
IKEv2	391
CKES	14367

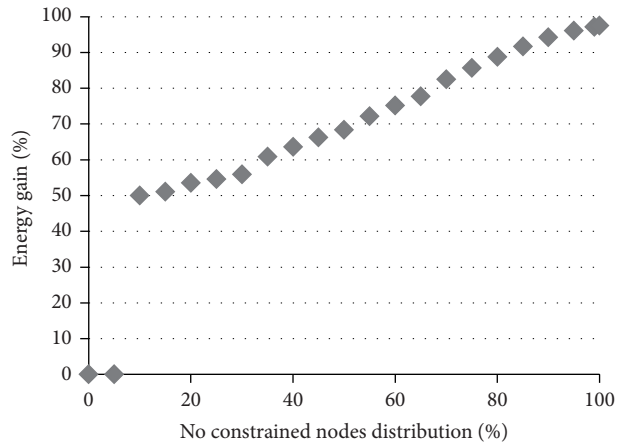

Figure 12: Energy gain of CKES.

In addition, we determined the gain (in %) on residual energy of CKES compared with IKEv2. As shown in Figure 12, the constrained node saves around 50% of its energy with 5% of no constrained nodes and also 94% of energy with 95% of no constrained nodes as compared with what is spent during the IKEv2.

On the other hand, we evaluated the network lifetime metric (parameter) in both approaches (Table 9). For this purpose we start the simulation with 80 nodes in which there are 40% of constrained nodes. Each one has 100 joules as initial energy (Ei). The establishment of one security association SA is made between one randomly chosen node (initiator) and the base station during one round in the simulation. The initiator makes periodically the Key Exchange process during the simulation (each round) and it continues the process at the time till the death of the first node in the network which is considered as the network lifetime. The simulation is done over 20000 rounds for each of the above two protocols.

Simulation results were expected since delegating the heavy computation leads to more energy savings at constrained nodes than offloading signature and encryption operations in exchange system. As shown in Figure 10, the comparison between IKEv2 and CKES confirms the efficiency of the Cooperative key Exchange System in terms of energy costs. It also proves the viability of the proposed collaborative approach in the studied context of IP-based WSN, which involves highly resource-constrained nodes. In addition, providing almost equivalent security level compared to the basic IKEv2, the CKES introduces an additional delay to establish security associations SAs between pairs but it saves more energy and it increases the network lifetime.

5. Conclusion and Future Work

This paper has presented a Cooperative Key Exchange System (CKES) based on the concept of CRT. The proposed approach is an adaptation of the IKEv2 in IP based WSN. We have modified it in order to provide more balanced energy consumption and longer lifetime comparing to a basic IKEv2 implementation.

We have presented the details of the design and implementation of CKES in NS2. We have compared this with the IKE and implemented the main functionalities that can be used in WSNs. The improvement of key exchange system in WSNs which we have proposed with the help of this new module can offer a better lifetime of network.

Our future work would explore the possibility to add a trust management system which could play an important role to make decisions in the collaborative system. We also aim to develop a resources management system in order to improve the balance of energy consumption between SNs.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work has been supported by the "Haute-Normandie" Regional Council and the European institutions by the FEDER program.

References

[1] H. Yu, J. He, T. Zhang, P. Xiao, and Y. Zhang, "Enabling end-to-end secure communication between wireless sensor networks and the Internet," *World Wide Web*, vol. 16, no. 4, pp. 515–540, 2013.

[2] C. Karlof, N. Sastry, and D. Wagner, "TinySec: a link layer security architecture for wireless sensor networks," in *Proceedings of the Second International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 162–175, November 2004.

[3] S. Dhurandher, M. S. Obaidat, G. Jain, I. Mani Ganesh, and V. Shashidhar, "An efficient and secure routing protocol for wireless sensor networks using multicasting," in *Proceedings of the IEEE/ACM International Conference on Green Computing and Communications (GreenCom '10), & International Conference on Cyber, Physical and Social Computing (CPSCom '10)*, pp. 374–379, Hangzhou, China, December 2010.

[4] V. Manral, "Cryptographic algorithm implementation requirements for encapsulating security payload (ESP) and authentication header (AH)," RFC 4835, IP Infusion, 2007.

[5] https://tools.ietf.org/html/rfc5996#page-132.

[6] J. Granjal, R. Silva, E. Monteiro, J. S. Silva, and F. Boavida, "Why is ipsec a viable option for wireless sensor networks," in *Proceedings of the 5th IEEE International Conference on Mobile Ad-Hoc and Sensor Systems (MASS '08)*, pp. 802–807, October 2008.

[7] X. Chen, K. Makki, K. Yen, and N. Pissinou, "Sensor network security: a survey," *IEEE Communications Surveys and Tutorials*, vol. 11, no. 2, pp. 52–73, 2009.

[8] S. A. Camtepe and B. Yener, "Key distribution mechanisms for wireless sensor networks: a survey," Tech. Rep., Rensselaer Polytechnic Institute, 2005.

[9] 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, ACM Press, November 2002.

[10] D. H. Yum and P. J. Lee, "Exact formulae for resilience in random key predistribution schemes," *IEEE Transactions on Wireless Communications*, vol. 11, no. 5, pp. 1638–1642, 2012.

[11] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," *Proceedings of the IEEE Symposium on Security and Privacy (SP '03)*, pp. 197–213, May 2003.

[12] L. Zhou, J. Ni, and C. V. Ravishankar, "Supporting secure communication and data collection in mobile sensor networks," in *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM '06)*, pp. 1–12, IEEE, April 2006.

[13] J. Lee and D. R. Stinson, "Deterministic key predistribution schemes for distributed sensor networks," in *Selected Areas in Cryptography*, vol. 3357 of *Lecture Notes in Computer Science*, pp. 294–307, Springer, Berlin, Germany, 2005.

[14] F. Delgosha and F. Fekri, "Key pre-distribution in wireless sensor networks using multivariate polynomials," in *Proceedings of the 2nd Annual IEEE Communications Society Conference on Sensor and AdHoc Communications and Networks (SECON '05)*, pp. 118–129, September 2005.

[15] H. Chan and A. Perrig, "PIKE: peer intermediaries for key establishment in sensor networks," in *Proceedings of the IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies (NFOCOM '05)*, pp. 524–535, March 2005.

[16] J. Granjal, E. Monteiro, and J. S. Silva, "A secure interconnection model for IPv6 enabled wireless sensor networks," in *Proceedings of the IFIP Wireless Days (WD '10)*, pp. 1–6, Venice, Italy, October 2010.

[17] G. Montenegro, N. N. Kushalnagar, J. Hui, and D. Culler, "RFC 4944—Transmission of IPv6 Packets over IEEE 802.15.4 Networks," September 2007, http://tools.ietf.org/html/rfc4944.

[18] S. Raza, S. Duquennoy, A. Chung, D. Yazar, T. Voigt, and U. Roedig, "Securing communication in 6LoWPAN with compressed IPsec," in *Proceedings of the 7th International Conference on Distributed Computing in Sensor Systems (DCOSS '11)*, pp. 1–8, Barcelona, Spain, June 2011.

[19] J. Hui and P. Thubert, *RFC 6282: Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks*, Arch Rock Corporation, Cisco, San Francisco, Calif, USA, 2011.

[20] V. Gupta, M. Millard, S. Fung et al., "Sizzle: a standards-based end-to-end security architecture for the embedded internet," in *Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications (PerCom '05)*, pp. 247–256, March 2005.

[21] W. Jung, S. Hong, M. Ha, Y.-J. Kim, and D. Kim, "SSL-based lightweight security of ip-based wireless sensor networks," in *Proceedings of the International Conference on Advanced Information Networking and Applications Workshops (WAINA '09)*, pp. 1112–1117, May 2009.

[22] L. Casado and P. Tsigas, "ContikiSec: a secure network layer for wireless sensor networks under the Contiki operating system," in *Identity and Privacy in the Internet Age*, vol. 5838 of *Lecture Notes in Computer Science*, pp. 133–147, Springer, Berlin, Germany, 2009.

[23] S. Raza, S. Duquennoy, J. Höglund, U. Roedig, and T. Voigt, "Secure communication for the internet of things—a comparison of link-layer security and IPsec for 6LoWPAN," *Security and Communication Networks*, vol. 7, no. 12, pp. 2654–2668, 2014.

[24] P. Varadarajan and G. Crosby, "Implementing IPsec in wireless sensor networks," in *Proceedings of the 6th International Conference on New Technologies, Mobility and Security (NTMS '14)*, pp. 1–5, IEEE, April 2014.

[25] S. Kent and K. Seo, "Security architecture for the internet protocol," RFC 4301, IETF, 2005.

[26] http://tools.ietf.org/html/rfc2631.

[27] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[28] G. R. Blakley, "Safeguarding cryptographic keys," in *Proceedings of the National Computer Conference, AFIPS Conference Proceedings*, vol. 48, pp. 313–317, New York, NY, USA, June 1979.

[29] M. Mignotte, "How to share a secret," in *Cryptography: Proceedings of the Workshop on Cryptography Burg Feuerstein, Germany, March 29-April 2, 1982*, T. Beth, Ed., vol. 149 of *Lecture Notes in Computer Science*, pp. 371–375, Springer, Berlin, Germany, 1983.

[30] C. A. Asmuth and J. Bloom, "A modular approach to key safeguarding," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 208–210, 1983.

[31] Y. Desmedt, "Some recent research aspects of threshold cryptography," in *Information Security—Proceedings of the First International Workshop on Information Security (ISW '97)*, E. Okamoto, G. I. Davida, and M. Mambo, Eds., vol. 1396 of *Lecture Notes in Computer Science*, pp. 158–173, Springer, Berlin, Germany, 1998.

[32] M. Mambo, K. Usuda, and E. Okamoto, "Proxy signatures: delegation of the power to sign messages," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E79-A, no. 9, pp. 1338–1353, 1996.

[33] M. K. Alam and K. S. Banu, "An approach secret sharing algorithm in cloud computing security over single to multi clouds," *International Journal of Scientific and Research Publications*, vol. 3, no. 4, 2013.

[34] http://www.cs.nthu.edu.tw/~ychung/conference/GPC2007_Irregular-sensor.pdf.

[35] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, IEEE Computer Society, Washington, DC, USA, January 2000.

[36] G. de Meulenaer, F. Gosset, F.-X. Standaert, and O. Pereira, "On the energy cost of communication and cryptography in wireless sensor networks," in *Proceedings of the 4th IEEE International Conference on Wireless and Mobile Computing, Networking and Communication (WiMob '08)*, pp. 580–585, October 2008.

[37] https://etd.ohiolink.edu/rws_etd/document/get/ysu1253597142/inline.

[38] R. A. Shaikh, H. Jameel, B. J. d'Auriol, H. Lee, S. Lee, and Y.-J. Song, "Group-based trust management scheme for clustered wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 11, pp. 1698–1712, 2009.

[39] M. Kasraoui, A. Cabani, and H. Chafouk, "IKEv2 authentication exchange model in NS-2," in *Proceedings of the International Symposium on Computer, Consumer and Control (IS3C '14)*, pp. 1074–1077, Taichung, Taiwan, June 2014.

[40] M. Kasraoui, A. Cabani, and H. Chafouk, "Secure collaborative system in heterogenous wireless sensor networks," *Journal of Applied Research and Technology*. In press.