Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 929542, 11 pages
doi:10.1155/2012/929542

Research Article

Low-Cost Monitoring and Intruders Detection Using Wireless Video Sensor Networks

Jacques M. Bahi,[1] Christophe Guyeux,[1] Abdallah Makhoul,[1] and Congduc Pham[2]

[1] *FEMTO-ST Institute, University of Franche-Comté, Rue Engel-Gros, BP 527, 90016 Belfort Cedex, France*
[2] *LIUPPA Laboratory, University of Pau, Avenue de l'Université, BP 1155, 64013 Pau, France*

Correspondence should be addressed to Abdallah Makhoul, abdallah.makhoul@univ-fcomte.fr

Received 1 December 2011; Revised 14 February 2012; Accepted 28 February 2012

Academic Editor: Wensheng Zhang

There is a growing interest in the use of video sensor networks in surveillance applications in order to detect intruders with low cost. The essential concern of such networks is whether or not a specified target can pass or intrude the monitored region without being detected. This concern forms a serious challenge to wireless video sensor networks of weak computation and battery power. In this paper, our aim is to prolong the whole network lifetime while fulfilling the surveillance application needs. We present a novel scheduling algorithm where only a subset of video nodes contributes significantly to detect intruders and prevent malicious attacker to predict the behavior of the network prior to intrusion. Our approach is chaos based, where every node based on its last detection, a hash value and some pseudorandom numbers easily compute a decision function to go to sleep or active mode. We validate the efficiency of our approach through theoretical analysis and demonstrate the benefits of our scheduling algorithm by simulations. Results show that in addition to being able to increase the whole network lifetime and to present comparable results against random attacks (low stealth time), our scheme is also able to withstand malicious attacks due to its fully unpredictable behavior.

1. Introduction

Instead of using traditional vision systems built essentially from fixed video cameras, it is possible to deploy autonomous and small wireless video sensor nodes (WVSNs) [1] to achieve video surveillance of a given area of interest. Doing so leads to a much higher level of flexibility, therefore extending the range of surveillance applications that could be considered. More interestingly, this scenario can support dynamic deployment scenario even in so-called object and obstacle-rich environments or hard-to-access areas. Such wireless video sensor nodes can in addition be thrown in mass to constitute a large-scale surveillance infrastructure. In these scenarios, hundreds or thousands of video nodes of low capacity (resolution, processing, and storage) of the same or similar type can be deployed in an area of interest.

Surveillance applications have very specific needs due to their inherently critical nature associated to security [2–4]. The basic objective of video surveillance systems is to allow detection and/or identification of intruders. Therefore, in that context, the main goal of a video sensor network is to ensure the coverage of the whole area of interest at any time t. Another issue of prime importance is related to energy considerations since the scarcity of energy does have a direct impact on coverage, as it is not possible to have all the video nodes in activity at the same time. Therefore, a common approach is to define a subset of the deployed nodes to be active while the other nodes can sleep. There are already some techniques that schedule video nodes to work alternatively while maintaining the complete coverage [5–7]. The main idea in these techniques is to turn off a redundant node. Here redundancy means that the covered area by a node is completely covered by its neighbors too. However, these techniques usually depend on location or directional information, which is costly in energy and complexity. Usually it is very difficult to determine the redundant nodes without the location information. Fortunately, not all applications need a complete coverage at anytime, and in most surveillance applications for intrusion detection, most sensor nodes can move to a so-called "idle mode" in the absence of intrusions. When an intruder is detected by a node, all the network will be alerted. In that context, it is critical to provide an effective scheme for turning off video nodes without degrading the surveillance quality.

In this paper, we present a solution to the scheduling problem in mission-critical surveillance applications using video sensor nodes. We provide a chaotic sleeping scheme and conduct a theoretical and simulation analysis of both performances and security. Until now, only random approaches have been extensively studied in the literature to turn off video nodes without degrading the surveillance quality. Even if such methods present good scores in detecting random intrusions while preserving the lifetime of the network, they do not encompass the situation of a malicious attacker. That is to say, the intruder is not supposed to know something about the surveillance scheme, he cannot observe the WVSN for a while, or he is not authorized to deduce anything from his possible knowledge. In this paper, we intend to tackle with situations where the attacker is not supposed passive: he is smart and does not necessarily choose a random way to achieve his intrusion. In addition to preserving the network lifetime and being able to face random attacks, we show that our scheme is also capable of withstanding attacks of a malicious adversary due to its unpredictable behavior.

The rest of the paper is organized as follows. In Section 2, related works related to surveillance applications with WVSNs are presented. Smart threats and malicious attackers are introduced in Section 3. Basic recalls and terminologies on the fields of the mathematical theory of chaos and chaotic iterations are given in Section 4, and the link unifying them is explained too. The surveillance scheme based on the chaos theory is detailed in Section 5. We show in Section 6 that our proposed scheme can be used against malicious attacks. Simulation results in Section 7 compare our scheme to the classical random schedule in terms of intruder's stealth time, network lifetime, and energy repartition. The paper ends by a conclusion section, where our contribution is summed up, and planned future work is detailed.

2. Related Works

In video sensor networks, minimizing energy consumption and prolonging the system lifetime are major design objectives. Due to the significant energy-saving when a node is sleeping, a frequently used mechanism is to schedule the sensor nodes such that redundant nodes go to sleep as often and for as long as possible. By selecting only a subset of nodes to be active and keeping the remaining nodes in a sleep state, the energy consumption of the network is reduced, thereby extending the operational lifetime of the sensor network.

In this context, the coverage problem for wireless video sensor networks can be categorized as

(i) *known targets coverage problem* which seeks to determine a subset of connected video nodes that covers a given set of target locations scattered in a 2D plane,

(ii) *region-coverage problem* which aims to find a subset of connected video nodes that ensures the coverage of the entire region of deployment in a 2D plane.

Most of the previous works have considered the known-targets coverage problem [8–11]. The objective is to ensure at all time the coverage of some targets with known locations that are deployed in a two-dimensional plane. For example, the authors in [11] organize sensor nodes into mutually exclusive subsets that are activated successively, where the size of each subset is restricted and not all of the targets need to be covered by the sensors in one subset. In [9], a directional sensor model is proposed, where a sensor is allowed to work in several directions. The idea behind this is to find a minimal set of directions that can cover the maximum number of targets. It is different from the approach described in [8] that aims to find a group of nondisjoint cover sets, each set covering all the targets to maximize the network lifetime.

Regarding the region-coverage problem in which this study takes place, existing works focus on finding an efficient deployment pattern so that the average overlapping area of each sensor is bounded. The authors in [12] analyze new deployment strategies for satisfying some given coverage probability requirements with directional sensing models. A model of directed communications is introduced to ensure and repair the network connectivity. Based on a rotatable directional sensing model, the authors in [13] present a method to deterministically estimate the amount of directional nodes for a given coverage rate. A sensing connected subgraph accompanied with a convex hull method is introduced to model a directional sensor network into several parts in a distributed manner. With adjustable sensing directions, the coverage algorithm tries to minimize the overlapping sensing area of directional sensors only with local topology information. Lastly, in [6], the authors present a distributed algorithm that ensures both coverage of the deployment area and network connectivity, by providing multiple cover sets to manage field of view redundancies and reduce objects disambiguation.

All the above algorithms depend on the geographical location information (position and direction) of video nodes. These algorithms aim to provide a complete-coverage network so that any point in the target area would be covered by at least one video node. However, this strategy is not as energy efficient as what we expect because of the following two reasons. Firstly, the energy cost and system complexity involved in obtaining geometric information may compromise the effect of those algorithms. Secondly, video nodes located at the edge of the area of interest must be always in an active state as long as the region is required to be completely covered. These video nodes will die after some time, and their coverage area will be left without surveillance. Thus, the network coverage area will shrink gradually from outside to inside. This condition is unacceptable in video surveillance applications and intrusion detection, because the major goal here is to detect intruders as they cross a border or as they penetrate a protected area.

One direction to solve these problems is to schedule a node to sleep following a probabilistic approach. Each node remains awake with a given probability so that the coverage of the area can be guaranteed. However the probability can be modeled by an observer, who can take benefits from his observations to predict the dynamic of the network. This is obviously a security flaw. These considerations lead us to the introduction of smart threats given in the next section.

3. Smart Threats

3.1. Introduction. Let us suppose that an adversary tries to reach a location X into the area without being detected. We consider that this situation leads to two categories of attacks.

(i) On the one hand, the attacker only knows that the area is under surveillance. He tries to take its chance, for example, by following the shortest way or by trying a random path. In this first category of attack that we call "blind elementary attacks," the intruder does not know how the surveillance is achieved as he does not observe the WVSN.

(ii) On the other hand, in the second category of attacks, called "malicious attacks" in this paper, the intruder is supposed to be intelligent. He can try to take benefits from his observations to understand the behavior of the WVSN. After having recorded the dynamic of the WVSN for a given time, the malicious intruder can try to determine when video nodes are turned on. This prediction can help the intruder to find a way to reach X without being detected.

In our opinion, the most reasonable way to evaluate the consequences of a malicious attack is to suppose that the intruder has access to the surveillance scheme. With this supposition, our security model encompasses the case where an attacker can have a physical access to a given node, thus determining the embedded mechanism used for video surveillance. In this Kerckhoffs-based principle, the attacker knows all but the initial parameters of the nodes. Moreover, he can observe the WVSN for a while. To achieve his intrusion, he can use all of the acquired knowledge—the sole difficulty is his lack of a secret parameter (the secret key) used to initialize the surveillance process.

The context of blind elementary attacks is well-known and understood: it has been studied a lot in the last decade, and various solutions have yet been proposed (Section 2). On the contrary, to the best of our knowledge, the case of an intelligent intruder (smart threat) has not yet really been treated. In this paper, we intend to propose a scheme able to withstand attacks encompassing these malicious intrusions and thus to offer a first solution to the problem raised by the smart threats existence hypothesis.

Technically speaking, the proposed approach offers several benefits. Firstly, the node scheduling algorithm does not need location information. Therefore, the energy consumption is reduced because there is no need to locate the node itself and its neighbors. Secondly, we will show that it performs as well as a random scheduling, in terms of lifetime and intrusion detection against blind elementary attacks (see Section 7). Lastly, due to its chaotic properties, its coverage is unpredictable, and thus a malicious adversary has no solution to attack the network (Section 6).

3.2. Classification of Malicious Attacks. When a malicious adversary attacks a WVSN, he can concentrate his efforts either on the global network or on some specific nodes. Depending on the considered situation, he can perform either an active attack, modifying the network architecture or a node, or a passive attack based only on observations. He can have access to several WVSNs using the same algorithm. Furthermore, he can build its own network to make some experiments. His objective is to find the secret key used in the targeted network: with this knowledge, the attacker will be able to predict the behavior of the video sensor nodes.

Active attacks have been already investigated several times in the literature. These studies encompass the cases where nodes can be added, moved, modified, or removed, where communications between nodes can be observed or modified, and where the global architecture of the network is attacked. However, some WVSN are such that any modification of the network is signaled, leading to the impossibility of such active attacks. On the contrary, passive observations and deductions of a malicious attacker are always possible. To the best of our knowledge, these threats have not yet been investigated.

The passive malicious attacks can be classified as follows.

(i) In the *target only attack (TOA)*, the adversary can only observe targeted networks.

(ii) In the *constant key attack (CKA)*, the adversary has access to several WVSNs using the same secret key. The areas under surveillance and the network architecture change from one WVSN to another, but the attacker knows that all these networks use the same algorithm with the same secret key.

(iii) In the *known original attack (KOA)*, the attacker had previously accessed to the WVSNs and its area. He had the opportunity to test various keys in a previous access. He hopes that this knowledge will help him to determine a way to realize his intrusion when the WVSN is really launched.

(iv) In the *chosen key attack (CKA)*, the adversary has access to an exact copy of the network and area under surveillance than the one he wants to attack. He has realized, for instance, a miniature model or a computer simulator having exactly the same behavior than the targeted network and its area. He can thus try several secret keys, and if he achieves to reproduce exactly the same behavior for the network, then he can reasonably suppose that the true secret key has been discovered.

(v) Finally, in the *estimated original attack (EOA)*, the attacker has only an estimation, an approximation of the network and its area.

In each of these categories, the sole objective of the attacker is to obtain the value of the secret key. With this knowledge, he will able to determine the WVSN behavior, finding by doing so a way to achieve his intrusion.

3.3. Security Levels in CKA. We now take place in the chosen key attack problem. Let k_0 be the secret key used to initiate the video surveillance. Denote by Y_k the probabilistic model

Table 1: Notations and terminologies.

S^n	The nth term of a sequence S
V_i	The ith component of a vector V
$f^k = f \circ \cdots \circ f$	The kth composition of a function f
$[\![1;N]\!]$	The set of integers: $\{1, 2, \ldots, N\}$
$\mathbb{B}$	The set of Booleans $\{0, 1\}$
$\mathbb{N}$	The set of positive integers $\{0, 1, 2, \ldots\}$
$\mathbb{R}$	The set of real numbers
$X \times Y$	The cartesian product between two sets X and Y

that the attacker can build with his observations, and by $\mathbb{K}$ the set of all possible keys.

Definition 1 (Insecurity). The WVSN is insecure against the target only attack if and only if $\exists k_1 \in \mathbb{K}$, $p(Y_{k_1}) = p(Y_{k_0})$ and for all $k_2 \in \mathbb{K}$, $p(Y_{k_2}) \neq p(Y_{k_0})$.

This is on the contrary with the following.

Definition 2 (Security). The WVSN is secure against the Target Only Attack if and only if for all $k \in \mathbb{K}$, $p(Y_k) = p(Y_{k_0})$.

In that situation, it is easy to prove that the mutual information $\mathcal{I}(k_0, Y_{k_0})$ is equal to 0, which is often refered as *perfect secrecy*.

4. Basic Recalls

4.1. Devaney's Chaotic Dynamical Systems. General notations used in this document are given in Table 1.

Consider a topological space $(\mathcal{X}, \tau)$, where $\mathcal{X}$ is a set and τ a topology on $\mathcal{X}$, that is, a collection of subsets of $\mathcal{X}$, satisfying the following well-known axioms.

(1) The empty set and $\mathcal{X}$ are in τ.

(2) τ is closed under arbitrary union.

(3) τ is closed under finite intersection.

Let $f : \mathcal{X} \to \mathcal{X}$ be a function continuous for this topology.

Definition 3. f is said to be *topologically transitive* if, for any pair of open sets $U, V \subset \mathcal{X}$, there exists $k > 0$ such that $f^k(U) \cap V \neq \varnothing$.

Definition 4. An element (a point) x is a *periodic element* (point) for f of period $n \in \mathbb{N}^*$, if $f^n(x) = x$.

Definition 5. f is said to be *regular* on $(\mathcal{X}, \tau)$ if the set of periodic points for f is dense in $\mathcal{X}$: for any point x in $\mathcal{X}$, any neighborhood of x contains at least one periodic point (without necessarily the same period).

Definition 6. f is said to be *chaotic* on $(\mathcal{X}, \tau)$ if f is regular and topologically transitive.

Let us recall that a *metric space* is an ordered pair $(\mathcal{X}, d)$, where $\mathcal{X}$ is a set and d is a metric on $\mathcal{X}$, that is, a function $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that for any $x, y, z \in \mathcal{X}$, the following holds:

(i) $d(x, y) \geq 0$ (non-negative),

(ii) $d(x, y) = 0$ if and only if $x = y$ (identity of indiscernibles),

(iii) $d(x, y) = d(y, x)$ (symmetry),

(iv) $d(x, z) \leq d(x, y) + d(y, z)$ (triangle inequality).

The chaos property is strongly linked to the notion of "sensitivity," defined on a metric space $(\mathcal{X}, d)$ by the following.

Definition 7. f has sensitive dependence on initial conditions if there exists $\delta > 0$ such that, for any $x \in \mathcal{X}$ and any neighborhood V of x, there exists $y \in V$ and $n > 0$ such that $d(f^n(x), f^n(y)) > \delta$. δ is called the constant of sensitivity of f.

Indeed, Banks et al. have proven in [14] that when f is chaotic and $(\mathcal{X}, d)$ is a metric space, then f has the property of sensitive dependence on initial conditions (this property was formerly an element of the definition of chaos). To sum up, quoting Devaney in [15], a chaotic dynamical system "is unpredictable because of the sensitive dependence on initial conditions. It cannot be broken down or simplified into two subsystems which do not interact because of topological transitivity. And in the midst of this random behavior, we nevertheless have an element of regularity." Fundamentally different behaviors are consequently possible and occur in an unpredictable way.

4.2. Chaotic Iterations. Let us consider a *system* of a finite number $\mathsf{N} \in \mathbb{N}^*$ of elements (or *cells*), so that each cell has a Boolean *state*. A sequence of length N of Boolean states of the cells corresponds to a particular *state of the system*. A sequence whose elements are subsets of $[\![1;\ \mathsf{N}]\!]$ is called a *strategy*. The set of all strategies is denoted by $\mathbb{S}$.

Definition 8. The set $\mathbb{B}$ denoting $\{0, 1\}$, let $f : \mathbb{B}^{\mathsf{N}} \to \mathbb{B}^{\mathsf{N}}$ be a function and $S \in \mathbb{S}$ be a strategy. The so-called *chaotic iterations* (CIs) are defined by [16] $x^0 \in \mathbb{B}^{\mathsf{N}}$ (x^0, the initial condition, is a chosen Boolean vector of size N) and

$$\forall n \in \mathbb{N}^*,\ \forall i \in [\![1;\ \mathsf{N}]\!], \quad x_i^n = \begin{cases} x_i^{n-1} & \text{if } i \notin S^n, \\ \left(f\left(x^{n-1}\right)\right)_{S^n} & \text{if } i \in S^n. \end{cases} \tag{1}$$

In other words, at the nth iteration, only the S^nth cell is "iterated."

Note that in a more general formulation, S^n can be a subset of components and $f(x^{n-1})_{S^n}$ can be replaced by $f(x^k)_{S^n}$, where $k < n$, describing for example, delays transmission. For the general definition of such chaotic iterations, see, for example, [16].

The term "chaotic", in the name of these iterations, has *a priori* no link with the mathematical theory of chaos recalled previously. However, we have proven in [17] that in a relevant metric space $(\mathcal{X}, d)$, the vectorial negation $f_0(x_1, \ldots, x_{\mathsf{N}}) = (\overline{x_1}, \ldots, \overline{x_{\mathsf{N}}})$ satisfies the three conditions for Devaney's chaos. This result is recalled in the next section.

4.3. Chaotic Iterations and Devaney's Chaos. Denote by Δ the *discrete Boolean metric*: for all $x, y \in \mathbb{B}$, $\Delta(x, y) = 0 \Leftrightarrow x = y$. Given a function $f : \mathbb{B}^{\mathsf{N}} \to \mathbb{B}^{\mathsf{N}}$, define the function $F_f : [\![1;\ \mathsf{N}]\!] \times \mathbb{B}^{\mathsf{N}} \to \mathbb{B}^{\mathsf{N}}$ such that

$$F_f(k, E) = \left(E_j \cdot \Delta(k, j) + f(E)_k \cdot \overline{\Delta(k, j)}\right)_{j \in [\![1;\ \mathsf{N}]\!]}, \tag{2}$$

where + and $\cdot$ are the Boolean addition and product operations and E_j is the jth coordinate of the Boolean vector E. The *shift* function is defined by $\sigma : (S^n)_{n\in\mathbb{N}} \in \mathbb{S} \to (S^{n+1})_{n\in\mathbb{N}} \in \mathbb{S}$, and the *initial function* i is the map which associates to a sequence, its first term: $i : (S^n)_{n\in\mathbb{N}} \in \mathbb{S} \to S^0 \in [\![1;\ \mathsf{N}]\!]$.

Consider the phase space: $\mathcal{X} = [\![1; \mathsf{N}]\!]^{\mathbb{N}} \times \mathbb{B}^{\mathsf{N}}$ and the map

$$G_f(S, E) = \left(\sigma(S), F_f(i(S), E)\right). \tag{3}$$

The chaotic iterations can be described by the following iterations

$$\begin{aligned} X^0 &\in \mathcal{X}, \\ X^{k+1} &= G_f\left(X^k\right). \end{aligned} \tag{4}$$

Let us define a new distance between two points (S, E), $(\check{S}, \check{E}) \in \mathcal{X}$ by

$$d\left((S, E); \left(\check{S}, \check{E}\right)\right) = d_e\left(E, \check{E}\right) + d_s\left(S, \check{S}\right), \tag{5}$$

where

(i) $d_e(E, \check{E}) = \sum_{k=1}^{\mathsf{N}} \Delta(E_k, \check{E}_k) \in [\![0;\ \mathsf{N}]\!]$,

(ii) $d_s(S, \check{S}) = (9/\mathsf{N}) \sum_{k=1}^{\infty} (|S^k - \check{S}^k|/10^k) \in [0;\ 1]$.

This new distance has been introduced in [17, 18] to satisfy the following requirements. When the number of different cells between two systems is increasing, then their distance should increase too. In addition, if two systems present the same cells and their respective strategies start with the same terms, then the distance between these two points must be small because the evolution of the two systems will be the same for a while. The distance presented above follows these recommendations. Indeed, if the floor value $\lfloor d(X, Y) \rfloor$ is equal to n, then the systems $E, \check{E}$ differ in n cells. In addition, $d(X, Y) - \lfloor d(X, Y) \rfloor$ is a measure of the differences between strategies S and $\check{S}$. More precisely, this floating part is less than 10^{-k} if and only if the first k terms of the two strategies are equal. Moreover, if the kth digit is nonzero, then the kth terms of the two strategies are different.

The following propositions are proven in [17, 18] by using the sequential continuity.

Proposition 9. *For all* $\mathsf{N} \in \mathbb{N}^*$, *for all* $f : \mathbb{B}^{\mathsf{N}} \to \mathbb{B}^{\mathsf{N}}, G_f$ *is a continuous function on* $(\mathcal{X}, d)$.

It is then checked in [17, 18] that in the metric space $(\mathcal{X}, d)$, the vectorial negation $f_0(x_1, \ldots, x_{\mathsf{N}}) = (\overline{x_1}, \ldots, \overline{x_{\mathsf{N}}})$ satisfies the three conditions for Devaney's chaos: regularity, transitivity, and sensitivity. This has led to the following result.

Proposition 10. *CIs are chaotic on* $(\mathcal{X}, d)$ *as it is defined by Devaney.*

These chaotic iterations have been used to define in [17] a hash function and a pseudorandom number generator (PRNG) able to pass the stringent TestU01 battery of tests in [19].

5. Chaos-Based Scheduling

5.1. The General Algorithm

5.1.1. Network Capabilities. The WVSN is supposed to be constituted by 2^{N} nodes V_i, $i \in [\![0; 2^{\mathsf{N}} - 1]\!]$. If the number of nodes is not a power of two, then N will be the smallest integer such that 2^{N} is greater than the number of nodes.

Each V_i is able to wake-up on a specific signal, to survey a given area (and to detect intrusions), to send a wake up signal to another node V_j, and to go to sleep when it is required. Furthermore, it is supposed that V_i embeds the following.

(i) The mechanisms required by the intrusion detection: a sensing function $c_i(t)$, such as a camera, which returns some digital data at each listening time, and a decision function $d_i(c)$ which returns if an intrusion is detected in these sensing values $(c_i(t))$ or not.

(ii) An internal clock having the time $T_i = r_i T_0$ as a reference. In other words, there exists a global minimal time interval T_0 such that the internal clock of frequency f_i of each node i is such that $T_i = 1/f_i$ is a multiple of T_0: $T_i = r_i T_0$, with $r_i \in \mathbb{N}$.

(iii) A vector of N binary digits, called *the state of the system* V_i, and the capability to swap each bit of this vector ($0 \leftrightarrow 1$).

(iv) An integer e_i, called *listening time*, initialized to 0.

In other words, each node V_i can achieve CIs. Thus, each node can compute, easily and by using a few resources, a hash value and some pseudorandom numbers as it is recalled in Section 4.2. We will denote by g_i the seed of the PRNG used in node V_i, which is equal to a secret parameter p_i at time $t = 0$. This secret parameter with N bits has been generated by a cryptographically secure PRNG, and thus it is uniformly distributed into $[\![0; 2^{\mathsf{N}} - 1]\!]$. The state V_i is initialized to the binary decomposition of g_i.

5.1.2. Deploying the Network. The deployment of video sensor nodes in the physical environment is the first operation (step) in the network lifecycle. It may take several forms. Sensor nodes may be randomly deployed dropping them from a plane and placed one by one by a human or a robot. Deployment may be a one-time activity or a continuous process. These methods have been extensively studied in the literature. In our method, the sole requirement to satisfy is to guarantee the uniform repartition into the region of interest.

5.1.3. Initialization of the WVSN. At time $t = 0$, a subset $\mathcal{I} \subset [\![0; 2^{\mathsf{N}} - 1]\!]$ (or $\mathcal{I} \subset [\![0; m]\!]$ if the network has m nodes, and if m is not a power of 2) of nodes are woken up and for all $i \in \mathcal{I}, e_i^{t_0} = T_i$.

5.1.4. Surveillance. The principle of surveillance applications is defined as follows. At each time $t_j = j \times T_0$, $j = 1, 2, \ldots$.

(1) If a sleeping node V_i has received $n_i^{t_j-1} \geq 1$ wake-up orders during the time interval $[t_j - 1, t_j]$, then it goes into active mode and sets its listening time $e_i^{t_j}$ to $n_i^{t_j-1} T_i$.

(2) If an active node V_i has received $n_i^{t_j-1} \geq 1$ orders to wake up during the time interval $[t_{j-1}, t_j]$, then it increments its listening time: $e_i^{t_j} = e_i^{t_j-1} + n_i^{t_j-1} T_i$.

(3) For each node V_i having a listening time $e_i^{t_j} \neq 0$.

  (i) V_i ensures the surveillance of its area during T_0.

  (ii) If, during this time interval, an intrusion is detected, then the WVSN is under alert.

  (iii) If t_j is the first listening time of V_i after having activated, then:

    (a) the hash value $h_i^{t_j}$ of the sensed value $c_i(t_j)$ is computed (c.f. Section 4.2),

    (b) the seed g_i of the PRNG of V_i is set to $h_i^{t_j} + t_j$, where + is the concatenation of the digits of $h_i^{t_j}$ and t_j (thus even if $h_i^{t_j} = h_i^{t_k}, k < j$, we have $g_i^{t_j} \neq g_i^{t_k}$),

    (c) the N bits of the state of the system V_i are set to $E_i^{t_j}$, where $E_i^{t_j}$ is the binary decomposition of i shown as a binary vector of length N.

(4) N bits are computed with the PRNG of V_i. These bits define an integer $S_i^{t_j} \in [\![0; 2^{\mathsf{N}} - 1]\!]$. Then the bit of $E_i^{t_j}$ in position $S_i^{t_j}$ is switched, which leads to a new state $E_i^{t_{j+1}}$. By doing so, CIs are realized.

(5) Each active node V_i decreases its listening time: $e_i^{t_j} = e_i^{t_j} - 1$.

(6) For each active node having its listening time $e_i^{t_j} = 0$:

  (i) V_i sends the wake-up order to node V_k, where $k \in [\![0; 2^{\mathsf{N}} - 1]\!]$ is the integer whose binary decomposition is the last state of the system $V_i(E_i^{t_{j+1}})$. If the number of nodes m is not a power of 2, then V_i tests first if $k < m$ before sending its signal.

  (ii) V_i goes to sleep.

5.2. Reducing Communication Overheads

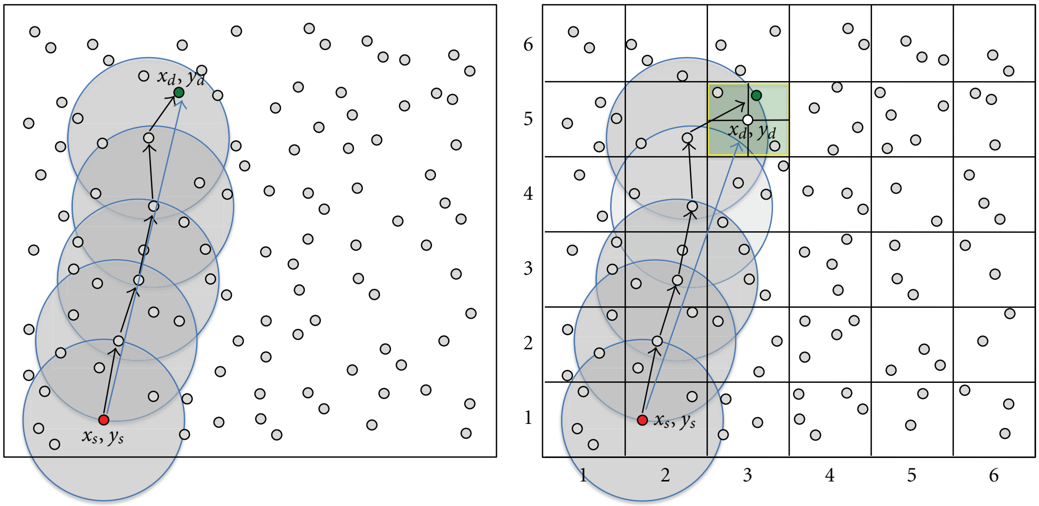
5.2.1. Avoiding Broadcast by Geographic Routing Waking up. In the previous section, the general algorithm requires in step 6 that for each active node V_i having its listening time $e_i^{t_j} = 0$, V_i sends a wake-up order to node V_k, k being determined by the chaotic iterations. Practically, if nodes' position is random, this wake-up message needs to be broadcasted to the entire network with a high communication overhead. It is possible to significantly reduce this overhead by using a geographic routing approach and having each node knowing the geographic position of the other nodes. A preliminary phase of neighbor advertising/discovering is then needed at network initialization whose cost could be regarded as low when the network lifetime becomes longer. Geographic position could be obtained by node's GPS capabilities or by a combination of GPS and other localization techniques. Figure 1(a) illustrates this method: node V_i at coordinate (x_s, y_s) needs to wake up node V_k at coordinate (x_d, y_d). Using geographic routing, the best neighbor is selected hop by hop to reach V_k.

It is also possible to reduce the constraints on the geographical coordinates of individual nodes by using geographic routing with area coordinates instead. The principle is to divide the area of interest in several smaller rectangular areas and to use the area's index instead of a node's index for the chaotic iterations. Once the area to wake-up is determined, we can use the geographic coordinates of the center of the area to propagate the wake up message without knowing the coordinate of individual nodes. Figure 1(b) illustrates this behavior where node V_i at coordinate (x_s, y_s) in area (2,1) has to wake up area (3,5). The coordinate of the center of area (3,5) could be computed/estimated by node V_i which will then use geographic routing to reach this area represented by a virtual node of coordinates (x_d, y_d). The last step of the geographic routing will wake up a node V_k (green dot) in the target area since this node is the closest to coordinates (x_d, y_d). Compared to the previous case, we can see that the wake-up node can be different. Note that it is not required that the estimation of an area's center be accurate as local nodes are selected by local broadcast at the radio level.

5.2.2. Neighborhood-Scoped Waking up. Another solution is to use neighbor-scoped waking up instead of considering the entire network. Since the network starts with a given number of active nodes, if each active node selects the next node to be woken up in their respective neighborhood, the chaotic iteration properties are still valid to make predictions of the future impossible for a smart intruder. Figure 2 illustrates this solution and shows 3 active nodes at step i (red dots) selecting one of their neighbors (green dots) with chaotic iterations on the neighbor index which in turn will select at step $i + 1$ another node (yellow dots) among their own neighbors. This solution has the advantage to avoid multi-hop transmissions and GPS coordinates.

5.3. Relaxing the Global Synchronization Assumption. The chaos-based scheduling algorithm, as presented previously, assumes that the sensor network is globally synchronized (each node maintains an internal clock $T_i = r_i \times T_0$ as a reference). However, this time synchronization can be costly or hard to maintain in various sensor networks. If so, this strong hypothesis can be relaxed as follows. T_0 can be replaced by T_0^i, where T_0^i is the smallest time interval between two operations in node V_i. That is, $T_0^i = 1/f_0^i$, where f_0^i is the CPU frequency of node i.

The sentence at the beginning of Section 5.1.4 "At each time $t_j = j \times T_0$, $j = 1, 2, \ldots$" can be changed in "For all node i, for all time $t_j = j \times T_0^i$, $j = 1, 2, \ldots$," and "V_i ensures



(a)

(b)

Figure 1: (a) avoiding broadcast with direct geographic routing. (b) area geographic routing.

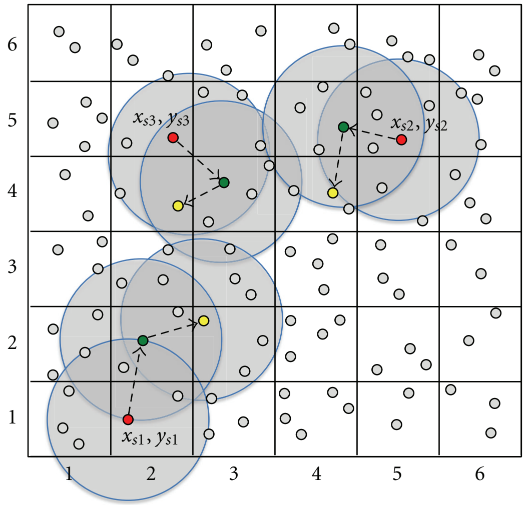


Figure 2: Neighbor-scoped activation.

the surveillance of its area during T_0" can be replaced by "V_i ensures the surveillance of its area during T_0^i" in step 3 of the surveillance algorithm.

The global synchronization assumption has been formulated initially to simplify the scheme description and because we need a common discrete time for all the network when establishing the proofs that makes the algorithm secure. In case where such a hypothesis cannot be supposed without loss of performances, we can obtain such a common discrete time by ordering and reindexing the set:

$$\left\{T_{i,j} = j \times T_0^i, i \in [\![0; 2^{\mathsf{N}} - 1]\!], j \in \mathbb{N}\right\} \text{ in } \{t_k, k \in \mathbb{N}\}, \tag{6}$$

and by considering the following order:

$$T_{i,j} \prec T_{i',j'} \iff \left(T_{i,j} < T_{i',j'}, \text{or } T_{i,j} = T_{i',j'} \text{ and } j \leqslant j'\right). \tag{7}$$

6. Theoretical Study

6.1. Scheduling as Chaotic Iterations. The scheduling scheme presented above can be described as CIs. The global state E^t of the whole system is constituted by the reunion of each internal state E_i^t of each node i. This is an element of $\mathbb{B}^{\mathsf{N}\times 2^{\mathsf{N}}}$. The strategy at time t is the subset of $[\![0; \mathsf{N} \times 2^{\mathsf{N}}]\!]$ constituted by all of the strategies that are computed into the awakened nodes at time t. More precisely, if the node V_k has computed the strategy S_k^t at time t, then the global strategy S^t will contain the value $S_k^t + k \times \mathsf{N}$. Lastly, the iteration function is the vectorial negation defined: $\mathbb{B}^{\mathsf{N}\times 2^{\mathsf{N}}} \rightarrow \mathbb{B}^{\mathsf{N}\times 2^{\mathsf{N}}}$. A subsequence E^{m^t} is extracted from E^t, which determines the changes that occur in the network: nodes whose binary id is into E^{m^t} are nodes that achieve the surveillance at the considered time. Let us remark that S^k and m^k depend both on the outside world, due to the fact that S_i^t are regularly seeded with the digest of some sensed values.

6.2. Complexity. Even if the hash function and the PRNG taken from [17, 19], respectively, can be replaced by any cryptographically secure hash function and PRNG, we do not recommend their substitution. Indeed, all of the operations used by our scheme can be achieved by CIs. Each iteration of CIs is only constituted by the negation of a few binary digits. Obviously, such an operation is fast and does not consume a lot of energy. By doing so, we thus obtain an efficient video surveillance scheduling scheme compliant with WVSN requirements. Section 7 will detail more quantitatively this fact.

6.3. Coverage. The coverage of the whole area is guaranteed due to the following reasons.

Firstly, the scheduling process corresponds to CIs. These iterations are chaotic according to Devaney; thus they are transitive. This transitivity property is the formulation of an uniform distribution in terms of topology. It claims that the system will never stop to visit any subregion of the whole area, regardless of how tiny the region is.

Secondly, as the choice of the nodes to wake up at each time is done by using CIs, this selection corresponds to the returned value of our PRNG proposed in [19]. This "CI(X, Y)-generator" takes two PRNGs X, Y as input sequences, realizes CIs with X as strategy, the vectorial negation as update function, and selects the states to publish as outputs by using the second PRNG Y. By such a combination, we improve the statistical properties of the input PRNG used as the strategy, and we add chaotic properties. The scheduling process corresponds to the CI(X, Y)-generator, with $X = m$ and $Y = S$. As Y is statistically perfect (Y is CI(ISAAC, ISAAC), which can pass the whole NIST, DieHARD, and TestU01 batteries of tests), the uniform repartition of the states is then guaranteed.

Lastly, experiments in Section 7 will show that this intended uniform coverage is well obtained in practice.

6.4. Security Study

6.4.1. Qualitative Approach. Let us suppose that Oscar, an intruder, knows that the scheduling process is based on CIs, that is, he knows the whole algorithm, except the seeds that have been used to initiate the PRNGs of each node. By doing so, we respect the Kerckhoffs' principle: the adversary has all except the secret key. Oscar's desire is to reach a particular location X of the area without being detected. To achieve his goal, he can choose two strategies. On the one hand, he can try a blind elementary attack, either by following a random way from its position to X, or by choosing the shortest path. The next subsection and the experiments will show that such an attack cannot work. On the other hand, Oscar can try to take benefits both from his knowledge and his observations. However, if he can determine the nodes that are awakened at time t, he cannot predict the awaken nodes at time $t+1$, $t+2, \ldots$. To do so, he should be able to obtain $S^{t+1}, S^{t+2}, \ldots$, which are computed from the digests of some values that will be sensed in the future. As our hash function satisfies the avalanche effect, due to its chaotic properties, any error on the sensed value leads to a completely different digest.

As Oscar cannot determine the sensed values of each node, at each time and with an infinite precision, he does not have the knowledge of the current state of the global system. He has only access to an approximation of this state. As the global scheduling process is chaotic, this error on the initial condition is magnified at each iteration, leading to the impossibility for Oscar to predict the scheduling process. This qualitative approach for security will be formalized in the second section below.

6.4.2. Chaotic Properties. We now investigate the topological properties presented by the proposed video-surveillance scheme. First of all, let us recall two fundamental definitions from the mathematical theory of chaos.

Definition 11. A function f is said to be expansive if $\exists \varepsilon > 0$, for all $x \neq y$, $\exists n \in \mathbb{N}$ such that $d(f^n(x), f^n(y)) \geq \varepsilon$.

Definition 12. A discrete dynamical system is said to be *topologically mixing* if and only if, for any pair of disjoint open sets $U, V \neq \varnothing$, $n_0 \in \mathbb{N}$ can be found so that for all $n \geq n_0$, $f^n(U) \cap V \neq \varnothing$.

As proven in [20], chaotic iterations are expansive and topologically mixing when f is the vectorial negation f_0. Consequently, these properties are inherited by the WVSN presented previously, which induce a greater unpredictability. Any difference on the initial parameter of the WVSN is, in particular, magnified up to be equal to the expansivity constant.

Now, what are the consequences for a wireless sensor network to be chaotic according to Devaney's definition? Firstly, the topological transitivity property implies indecomposability.

Definition 13. A dynamical system $(\mathcal{X}, f)$ is *indecomposable* if it is not the union of two closed sets $A, B \subset \mathcal{X}$ such that $f(A) \subset A$, $f(B) \subset B$.

Hence, reducing the observed area in order to simplify its complexity is impossible if $\Gamma(f)$ is strongly connected. Moreover, under this hypothesis the surveillance scheme is strongly transitive.

Definition 14. A dynamical system $(\mathcal{X}, f)$ is *strongly transitive* if for all $x, y \in \mathcal{X}$, for all $r > 0$, $\exists z \in \mathcal{X}$, $d(z, x) \leq r \Rightarrow \exists n \in \mathbb{N}^*$, $f^n(z) = y$.

According to this definition, for all pair of points x, y in the phase space, a point z can be found in the neighborhood of x such that one of its iterates $f^n(z)$ is y. Indeed, this result has been stated during the proof of the transitivity presented in [17]. Among other things, the strong transitivity leads to the fact that without the knowledge of the initial awakened nodes, all scheduling is possible. Additionally, no nodes of the output space can be discarded when studying the video-surveillance scheme: this space is intrinsically complicated and it cannot be decomposed or simplified.

Finally, these WVSNs possess the instability property.

Definition 15. A dynamical system $(\mathcal{X}, f)$ is unstable if for all $x \in \mathcal{X}$, the orbit $\gamma_x : n \in \mathbb{N} \mapsto f^n(x)$ is unstable; that means $\exists \varepsilon > 0$, for all $\delta > 0$, $\exists y \in \mathcal{X}$, $\exists n \in \mathbb{N}$, such that $d(x, y) < \delta$ and $d(\gamma_x(n), \gamma_y(n)) \geq \varepsilon$.

This property, which is implied by sensitive point dependence on initial conditions, leads to the fact that in all neighborhoods of any point x there are points that can be apart by ε in the future through iterations of the WVSN. Thus, we can claim that the behavior of these networks is unstable when $\Gamma(f)$ is strongly connected.

6.4.3. Cryptanalysis in CKA Framework. As stated in Section 6.1, the proposed video-surveillance scheme can be rewritten as

$$\begin{aligned} X^0 &\in \mathcal{X}, \\ X^{k+1} &= G_{f_0}\left(X^k\right), \end{aligned} \tag{8}$$

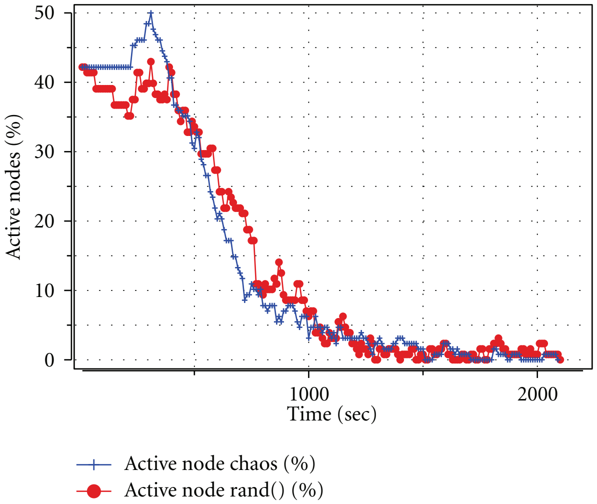


Figure 3: Percentage of active nodes.

where the phase space is $\mathcal{X} = [\![1;\ \mathsf{N}\times 2^{\mathsf{N}}]\!]^{\mathbb{N}} \times \mathbb{B}^{\mathsf{N}\times 2^{\mathsf{N}}}$, X^0 depends on a secret parameter $p = (p_1,\dots,p_{\mathsf{N}}) \in (\mathbb{B}^{\mathsf{N}})^{\mathbb{N}}$ whose binary digits are uniformy distributed, and f_0 stands for the vectorial negation on $\mathbb{B}^{\mathsf{N}\times 2^{\mathsf{N}}}$.

We will now show the following.

Proposition 16. *The videosurveillance scheme proposed in this document is secure when facing a chosen key attack.*

Proof. Let $N = \mathsf{N}\times 2^{\mathsf{N}}$. We will prove by a mathematical induction that for all $n \in \mathbb{N}, X^n \sim \mathbf{U}(\mathbb{B}^N)$. The base case is immediate, as the initial state of the WVSN is initialized by $(g_1,\dots,g_{\mathsf{N}}$, which are produced by a cryptographically secure PRNG, so $X^0 \sim \mathbf{U}(\mathbb{B}^N)$. Let us now suppose that the statement $X^n \sim \mathbf{U}(\mathbb{B}^N)$ holds for some n. Let $e \in \mathbb{B}^N$ and $\mathbf{B}_k = (0,\dots,0,1,0,\dots,0) \in \mathbb{B}^N$ (the digit 1 is in position k). So $P(X^{n+1} = e) = \sum_{k=1}^{N} P(X^n = e + \mathbf{B}_k, S^n = k)$. These two events are independent, thus $P(X^{n+1} = e) = \sum_{k=1}^{N} P(X^n = e + \mathbf{B}_k) \times P(S^n = k)$. According to the inductive hypothesis, $P(X^{n+1} = e) = (1/2^N)\sum_{k=1}^{N} P(S^n = k)$. The set of events $\{S^n = k\}$ for $k \in [\![1;N]\!]$ is a partition of the universe of possible, so $\sum_{k=1}^{N} P(S^n = k) = 1$.

Finally, $P(X^{n+1} = e) = 1/2^N$, which leads to $X^{n+1} \sim \mathbf{U}(\mathbb{B}^N)$. This result is true for all $n \in N$; we thus have proven that;

$$\forall p, Y_p = X^{N_0} \sim \mathbf{U}\left(\mathbb{B}^N\right). \tag{9}$$

So the video surveillance defined in this paper is secure in CKA. □

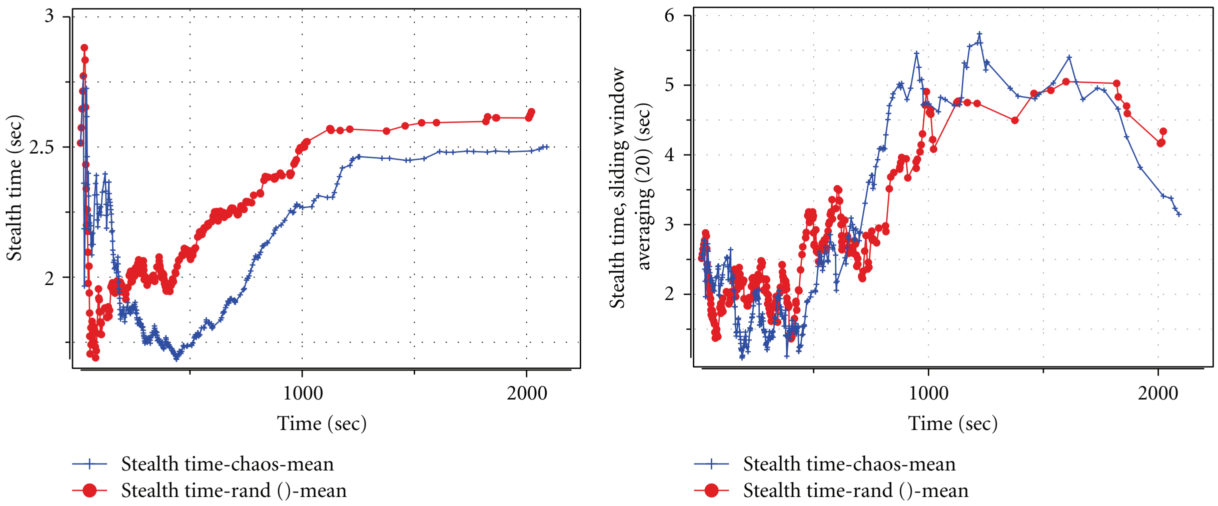
7. Simulation Results

This section presents simulation results on comparing our chaotic approach to the standard C++ rand()-based approach with random intrusions. We use the OMNET++ simulation environment, and the next node selection will either use chaotic iterations or the C++ rand() function (rand() %2^n) to produce a random number between 0 and 2^n. For this set of simulations, 128 sensor nodes (therefore $n = 7$) are randomly deployed in a 75 m $*$ 75 m area. Unless specified, sensors have an 36° AoV, and sensor node captures at the rate of 0.2 fps. Each node starts with a battery level of 100 units, and taking 1 picture consumes 1 unit of battery. When a node V_i is selected to wake up, it will be awake for T_i seconds. We set all $T_i = T = 20$ s. According to the behavior defined in Section 5, before going to sleep after an activity period of $e_i T$, V_i will determine the next node to be woken up. It can potentially elect itself in which case V_i stays active for at least another T period. The elected node can be already active, in which case it simply increases its e_i counter. We set about 50% of the sensor nodes to be active initially (each sensor draws a random value between 0 and 1 and if the value is greater than 0.5, it will be active). This initial threshold is tunable but we did not try to vary this parameter in this paper. The results presented here have been averaged over 10 simulation runs with different initial seeds. Figure 3 shows the percentage of active nodes. Both the chaotic and the standard rand() function have similar behavior: the percentage of active nodes progressively decreases due to battery shortage.

To compare both approaches in terms of surveillance quality, we record stealth time when intrusions are introduced in the area of interest. The stealth time is the time during which an intruder can travel in the field without being seen. The first intrusion starts at time 10 s at a random position in the field. The scan line mobility model is then used with a constant velocity of 5 m/s to make the intruder move to the right part of the field. When the intruder is seen for the first time by a sensor, the stealth time is recorded and the mean stealth time computed. Then a new intrusion appears at another random position. This process is repeated until the simulation ends (i.e., no more sensor nodes with energy). Figure 4(a) shows the mean stealth time over the whole simulation duration. Figure 4(b) shows the same data but with a sliding window averaging filter of 20 values. As the nodes are uniformly distributed in the area of interest, there is a strong correlation between the percentage of active nodes and the stealth time as it can be expected. The result we want to highlight here is that our chaotic node selection approach has a slightly better level of performance in presence of random intrusions than standard rand() function in addition to providing a formal proof of non-prediction by malicious intruders.

The last result we want to show is the energy consumption distribution. We recorded every 10 s the energy level of each sensor node in the field and computed the mean and the standard deviation. Figure 5 shows the evolution of the standard deviation during the network lifetime. We can see that the chaotic node selection provides a slightly better distribution of activity than the standard rand() function.

This better distribution of the node's activity has the beneficial effect to increase the detection quality: as nodes are used more equally, there are less "holes" in the surveillance network due to some nodes having battery shortage earlier.



(a) Stealth time

(b) Stealth time with a sliding window

Figure 4

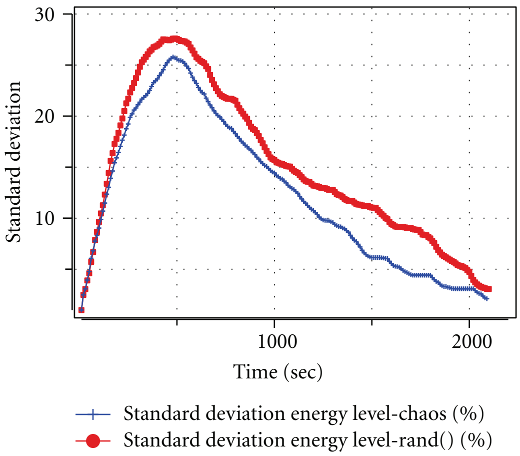


Figure 5: Evolution of the energy consumption's standard deviation.

8. Comparison with the Random Approach

Experiments show that the proposed algorithm is as good as random scheduling in case of random attacks. Lifetime of the network ant stealth time is quite similar for the two approaches. Under this point of view, when facing a random attack, there is no improvement compared to existing methods. However, in the case of a malicious attack, using the proposed chaos-based approach instead of a random one is of importance. Indeed, the random approach is either insecure or not realizable in that context. It is due to the constraints inherent to the wireless video sensor networks. The problem can be summarized as follows: either the generators embedded into each node in a random scheduling are cryptographically secure (as ISAAC or the Blum-Blum-Shub—BBS—generator e.g.), and thus they are very slow and need a lot of computational resources, which are two issues incompatible with the objectives and limitations of such networks or they are fast but insecure and thus not adapted when a high level of security is required.

Security of a pseudorandom generator is a characteristic that shows how hard it is to tell the difference between the pseudorandom sequences and truly random sequences. A pseudorandom generator is said to be provably secure if distinguishing these two classes of sequences is as difficult as solving a well-known and supposedly hard (typically number-theoretic) problem. For instance, the BBS pseudorandom generator is secure under the assumption that factoring large Blum integers are a difficult problem. Such a security level cannot be attained by fast and light PRNGs that are in common use in WVSN. These PRNGs are fast but insecure: either bias appears in their iterations, leading to the possibility to take benefits of this bias to predict the future evolution of the network, or they have been cryptanalyzed by more specific and refined attacks.

The approach we propose does not rely its security on the quality of the randomness but on topological properties of chaos, making the attacks described above inefficient to break the scheduling program. In that situation, chaotic properties reinforce the security of the scheme by making it impossible to forecast the future evolution of the network by using the knowledge acquired during previous observations. Furthermore, we have proven in Section 6.4.3 that the proposed scheme is secure when considering attacks in the CKA framework. To the best of our knowledge, it is not the case for the random approach. Therefore our approach has a solid and formally proven security foundation while the random approach has no such proofs.

9. Conclusions and Perspectives

In this paper, a sleeping scheme for nodes has been proposed as an effective and secure solution to the scheduling problem in mission-critical surveillance applications using WVSNs. It has been evaluated through theoretical and practical aspects of performance and security. As opposed to existing works, this scheduling scheme is not based only on randomness but on the mathematical theory of chaos also. By doing so, we reinforce coverage and lifetime of the network, while obtaining a more secure scheme. We have considered in this paper the case where the intruder is smart and active. Furthermore, we have supposed that he can know the scheme and observe the behavior of the network. We have shown that, in addition to being able to preserve WVSN lifetime and to present comparable results against random attacks, our scheme is also able to withstand such malicious attacks due to its unpredictable behavior.

In future work, we intend to enlarge the security field in WVSN-based video surveillance, by making a classification of attacks that Oscar can achieve depending on the data he has access to. Our desire is to distinguish between several levels of security into each category of malicious attacks, from the weakest one to the strongest one. Additionally, we will study more precisely the topological properties of the scheduling scheme presented in this paper.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[2] T. He, S. Krishnamurthy, J. A. Stankovic et al., "Energy-efficient surveillance system using wireless sensor networks," in *Proceedings of the 2nd International Conference on Mobile Systems, Applications and Services (MobiSys '03)*, pp. 270–283, Boston, Mass, USA, 2004.

[3] S. Oh, P. Chen, M. Manzo, and S. Sastry, "Instrumenting wireless sensor networks for real-time surveillance," in *Proceedings of the International Conference on Robotics and Automation (ICRA '06)*, pp. 3128–3133, Orlando, Fla,USA, May 2006.

[4] Y. Zhu and L. M. Ni, "Probabilistic approach to provisioning guaranteed QoS for distributed event detection," in *Proceedings of the 27th IEEE Communications Society Conference on Computer Communications (INFOCOM '08)*, pp. 592–600, Phoenix, Ariz, USA, April 2008.

[5] J. Wang, C. Niu, and R. Shen, "Randomized approach for target coverage scheduling in directional sensor network," in *Proceedings of the 3rd International Conference on Embedded Software and Systems (ICESS '07)*, vol. 4523 of *Lecture Notes in Computer Science*, pp. 379–390, 2007.

[6] C. Pham, A. Makhoul, and R. Saadi, "Risk-based adaptive scheduling in randomly deployed video sensor networks for critical surveillance applications," *Journal of Network and Computer Applications*, vol. 34, no. 2, pp. 783–795, 2011.

[7] C. Pham and A. Makhoul, "Performance study of multiple cover-set strategies for mission-critical video surveillance with wireless video sensors," in *Proceedings of the 6th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob '10)*, pp. 208–216, Niagara Falls, Canada, October 2010.

[8] Y. Cai, W. Lou, M. Li, and X.-Y. Li, "Target-oriented scheduling in directional sensor networks," in *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM '07)*, pp. 1550–1558, 2007.

[9] J. Ai and A. A. Abouzeid, "Coverage by directional sensors in randomly deployed wireless sensor networks," *Journal of Combinatorial Optimization*, vol. 11, no. 1, pp. 21–41, 2006.

[10] H. Liu, P. Wan, and X. Jia, "Maximal lifetime scheduling for sensor surveillance systems with K sensors to one target," *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 12, pp. 1526–1536, 2006.

[11] M. X. Cheng, L. Ruan, and W. Wu, "Achieving minimum coverage breach under bandwidth constraints in wireless sensor networks," in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '05)*, pp. 2638–2645, March 2005.

[12] H. Ma and Y. Liu, "Some problems of directional sensor networks," *International Journal of Sensor Networks*, vol. 2, no. 1-2, pp. 44–52, 2007.

[13] D. Tao, H. Ma, and L. Liu, "Coverage-enhancing algorithm for directional sensor networks," in *Proceedings of the 2nd International Conference, Mobile Ad-Hoc and Sensor Network*, Lecture Notes in Computer Science, pp. 256–267, Springer, Hong Kong, 2006.

[14] J. Banks, J. Brooks, G. Cairns, and P. Stacey, "On devaney's definition of chaos," *American Mathematical Monthly*, vol. 99, pp. 332–334, 1992.

[15] L. R. Devaney, *An Introduction to Chaotic Dynamical Systems*, Westview, 2nd edition, 2003.

[16] F. Robert, *Discrete Iterations: A Metric Study*, vol. 6 of *Computational Mathematics*, Springer, 1986.

[17] J. M. Bahi and C. Guyeux, "Hash functions using chaotic iterations," *Journal of Algorithms & Computational Technology*, vol. 4, no. 2, pp. 167–181, 2010.

[18] J. M. Bahi and C. Guyeux, "Topological chaos and chaotic iterations, application to hash functions," in *Proceedings of the IEEE World Congress on Computational Intelligence (WCCI '10)*, pp. 1–7, Barcelona, Spain, July 2010.

[19] J. M. Bahi, C. Guyeux, and Q. Wang, "A pseudo random numbers generator based on chaotic iterations. application to watermarking," in *Proceedings of the International Conference on Web Information Systems and Mining*, pp. 202–211, 2010.

[20] C. Guyeux, N. Friot, and J. M. Bahi, "Chaotic iterations versus spread-spectrum: chaos and stego security," in *Proceedings of the 6th International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP '10)*, pp. 208–211, Darmstadt, Germany, October 2010.

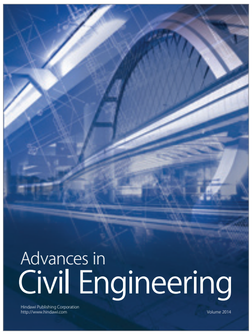
Journal of
Engineering

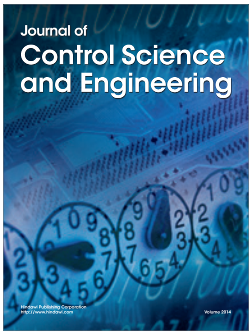
The Scientific World Journal

International Journal of
Rotating Machinery

Journal of
Sensors

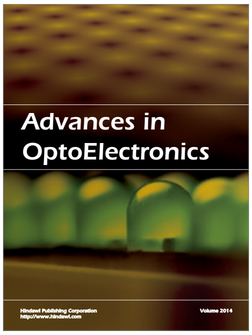
International Journal of
Distributed Sensor Networks


Advances in
Civil Engineering


Journal of
Control Science and Engineering

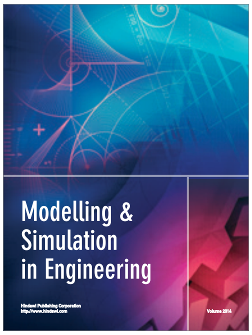
Journal of
Robotics

Journal of
Electrical and Computer Engineering

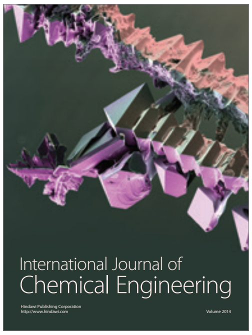

Advances in
OptoElectronics

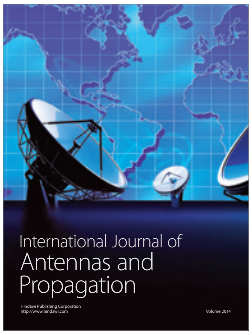
VLSI Design

International Journal of
Navigation and Observation

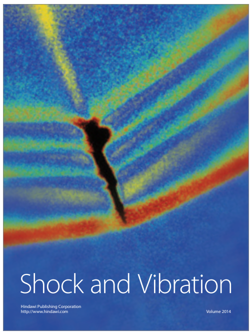

Modelling &
Simulation
in Engineering

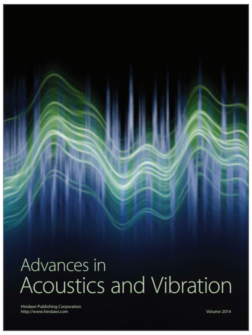
International Journal of
Aerospace Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive
Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration