International Journal of Innovative Computing, Information and Control
Volume **8**, Number **2**, February **2012**

ICIC International ©2012 ISSN 1349-4198
pp. 1001–1023

TAME POOL-BASED PAIRWISE KEY PREDISTRIBUTION FOR LARGE-SCALE SENSOR NETWORKS*

Yen-Hua Liao[1], Chin-Laung Lei[1], Ai-Nung Wang[2] and Wen-Chi Tsai[3]

[1]Department of Electrical Engineering
[2]Department of Mathematics
National Taiwan University
No. 1, Sec. 4, Roosevelt Road, Taipei 10617, Taiwan
radha@fractal.ee.ntu.edu.tw; lei@cc.ee.ntu.edu.tw; wang@math.ntu.edu.tw

[3]Department of Statistics
National Chengchi University
No. 64, Sec. 2, Zhi-nan Rd., Taipei 11605, Taiwan
wctsai@nccu.edu.tw

Received August 2010; revised February 2011

Abstract. *Due to resource constraints on sensor nodes, public key cryptography is not desirable to use in sensor networks. Instead, symmetric cryptosystem using preloaded keys is suitable for this environment. In such a solution, pairwise key establishment is a fundamental task for securing sensor networks. However, the conventional polynomial-based key predistribution approach, on which many existing key management schemes in sensor networks were based, can only offer probabilistic authentication. In this paper, we first develop a novel tame-based key predistribution approach, where we exploit tame automorphisms to get symmetric and two-one bivariate maps for the pairwise key establishment. This tame-based approach can provide deterministic authentication between two parties. We then present a general framework for the key predistribution, on the basis of the tame-based approach. It turns out that this tame map can substitute the conventional polynomial in any existing polynomial-based scheme to offer deterministic authentication service. In particular, we further introduce a new key predistribution scheme, the local symmetric-tame maps predistribution scheme, based on this framework and the deployment information. The analysis demonstrates that, in addition to being able to provide deterministic authentication service, the scheme not only has significantly better performance, but can also achieve greater resilience on security than existing schemes.*
Keywords: Sensor networks, Tame automorphism, Key management, Pairwise key, Network security, Authentication

1. **Introduction.** Sensor networks can be applied widely in many areas such as battlefield surveillance [1], healthcare [2], intelligent home, intelligent vehicle [3], location system [4], commercial inventory management and industrial applications (e.g., machine monitoring) [5]. A sensor network is typically composed of a large number of resource-limited sensor nodes with low-capacity power supply such as batteries, relatively slow processor and limited memory (For example, a typical MICA 2 mote is equipped with 7.8 MHz processor, 4KB RAM and two AA batteries) [6].

Besides functionality, security services such as confidentiality, authentication and integrity are critical for sensor networks in order to facilitate proper operations in sensor networks. Key management, which provides and manages the cryptographic keys for security mechanisms, is the fundamental of security services and is never a trivial task.

*A preliminary version of this paper appeared in *GLOBECOM*, Nov. 2009.

The difficulty to manage keys over sensor networks mainly comes from two properties of sensor networks: the lack of resource for conventional public key cryptography and the vulnerability to physical attacks such as the destruction or the capture of sensor nodes. Due to the importance and difficulty of key management over sensor networks, many key management schemes based on symmetric cryptosystem are proposed recently [7-25].

The basic issue of key management in symmetric cryptosystem is how to establish a symmetric shared key between two communication parties. Generally, in a sensor network, the network topology cannot be known in advance before deployment. Therefore, a practical method for the key distribution in sensor networks is to preload key(s) in each sensor node prior to deployment. There are two extremes. One is to have all the nodes in network shared one common key. The advantages of this solution are no communication or computation needed for pairwise key establishment and low storage overhead. However, it suffers from the situation that the capture of a single node will compromise the whole network. The other extreme is to have each sensor node preloaded $N-1$ pairwise keys, each of which is shared with each of other $N-1$ nodes, where N is the total number of sensor nodes. Although, in this solution, the captured nodes will not compromise any other pairwise keys used between non-compromised nodes, the storage overhead (e.g., $N = 10,000$) makes it infeasible due to the constrained memory on sensor nodes.

Usually, in sensor networks, the most basic and commonest communication pattern for a sensor node is to communicate with its neighbors. Other communication patterns (such as communicating between two nodes which are more than one-hop away) can be realized based on this basic communication pattern [22]. Therefore, how to establish a pairwise key between two neighboring sensor nodes gets mostly concerned. Eschenauer and Gligor [7] proposed a probabilistic key predistribution technique, where each sensor node randomly picks a subset of keys from a large key pool before deployment. Thus, two neighbors can share at least one common key with a certain probability after deployment. Many schemes were developed based on the basic probabilistic key predistribution scheme [7]. However, the problem of these existing probabilistic key predistribution schemes is that they can only provide probabilistic authentication service in sensor networks, since two pairs of sensor nodes may share the same pairwise key in these schemes.

In this paper, we first take advantage of tame automorphism in algebra to develop a novel tame-based key predistribution approach, in which a pairwise key is uniquely shared between two sensor nodes. Therefore, the approach can provide deterministic authentication service for sensor networks through challenge-response. Based on the approach, we further introduce a key predistribution scheme for large-scale sensor networks. The analysis shows that the scheme not only has significantly higher probability to establish a direct pairwise key between two neighbors, but can also achieve greater resilience against node compromises. Furthermore, the existing key predistribution schemes need to do trade-off between the performance of establishing a direct pairwise key between two neighbors and the resilience against node compromises. However, our scheme can achieve arbitrarily high probability to establish a direct pairwise key between two neighbors without decreasing the resilience of security.

The rest of this article is organized as follows. We briefly review the related works for sensor networks in Section 2. In Sections 3 and 4, we develop a tame-based key predistribution approach and a tame pool-based key predistribution framework respectively. Based on this framework, the conventional polynomial in any existing polynomial pool-based scheme can be substituted by the tame map in order to offer deterministic authentication service. Particularly, in Section 5, we further introduce a new instantiation in details, the *local symmetric-tame maps predistribution scheme*, on the basis of the framework and the deployment knowledge. The analyses of the performance, resilience against node compromise, communication, computation and memory overhead for the scheme are given in Section 6. In Section 7, we compare the effect of decreasing memory overhead to various schemes. We conclude this article in Section 8.

2. **Related Work.** There have been a number of key management schemes developed for sensor networks. Eschenauer and Gligor [7] proposed a basic probabilistic key predistribution technique to establish pairwise keys in sensor networks. Chan et al. [8] further extended the basic technique [7] to the q-composite key predistribution scheme, which allows two sensor nodes to setup a pairwise key only when they share at least q common keys. In these schemes, a small number of compromised nodes will compromise a large portion of direct pairwise keys between non-compromised sensor nodes. Chan et al. [8] additionally proposed a random-pairwise keys scheme. It can provide the property of perfect resilience against node compromises, which means that the compromised sensor nodes will not compromise any direct pairwise key between two non-compromised sensor nodes. However, the scheme suffers from the limited network size. Based on Blundo's polynomial-based approach [26] and the basic scheme [7], Liu et al. [12,14] presented the polynomial pool-based schemes: the random subset assignment scheme and grid-based scheme. Du et al. [10] proposed the multiple-space key predistribution scheme based on Blom's work [27] and the basic scheme [7]. Their scheme is equivalent to the random subset assignment scheme [12]. These schemes have a threshold property on the resilience against node compromises, which means that a sensor network remains perfect resilience up to the capture of a certain fraction of sensor nodes.

Taking advantage of the expected locations of sensor nodes, Du et al. [11] presented a scheme to improve the basic scheme [7]. Ito et al. [28] further extended Du et al.'s [11] scheme to a scheme where different keys are logically mapped to two dimensional positions, and the keys distributed to a node are determined by their positions estimated through the probability density function of node deployment. Liu and Ning [25] proposed three schemes using the expected location or deployment location information of sensor nodes to improve the random-pairwise keys scheme [8] and the random subset assignment scheme [12]. In this article, we present a scheme which further combines predeployment knowledge and postdeployment knowledge of sensor nodes for key predistribution to achieve significantly better performance and resilience on security than aforementioned schemes.

Due to the use of conventional polynomials for key predistribution, the polynomial pool-based schemes (The basic scheme [7] is a special case of the polynomial pool-based approach.) have collision problem, which means that two pairs of sensor nodes may share the same pairwise key. Therefore, they can only provide probabilistic authentication service in sensor networks. Perrig et al. [29] presented two security protocols: SNEP and μTESLA. SNEP is a protocol for data confidentiality, two-party data authentication and data freshness. μTESLA was developed based on TESLA [30] for broadcast authentication in sensor networks. Their schemes use a base station for key management after deployment. Traynor et al. [23,24] also proposed a mechanism using key distribution center (KDC) to perform probabilistic authentication between two sensor nodes. These schemes rely on a KDC/base station for key management after deployment. It restricts the scalability of their works. In this article, we introduce a tame-based approach for key predistribution, which can easily offer deterministic authentication service without the participation of a KDC/base station. Actually, our tame map can substitute the polynomial in any existing polynomial pool-based scheme to provide deterministic authentication service between two parties in sensor networks.

Zhu et al. [22] developed LEAP(+) key management protocol for sensor networks. In this scheme, an attacker can easily compromise all the sensor nodes in a sensor network and then take over the network as long as he (or she) compromises a single node after the deployment and before the expiration of the node's time interval $T_{\min}$ (We assume that nodes are protected well before the deployment.). Besides, for a sensor network with high node density (e.g., the number of neighbors is more than 20), the scheme may not be desirable due to the lack of resilience. However, our scheme can tolerate many nodes to be compromised without threatening the direct communications between two non-compromised nodes, and can be applied to any sensor network with a reasonable node density.

Other related researches were studied. Chan and Perrig [18] proposed a protocol using one or more peer sensor nodes as intermediaries to help to establish pairwise keys. Chuang et al. [31] presented a group-based scheme using one-way function to generate group-to-group pairwise keys. Miller and Vaidya [32] presented a protocol utilizing channel diversity to allow neighboring nodes to establish pairwise keys via broadcasting plaintext keys. In Law et al. [33], they constructed an evaluation framework to identify the suitable candidates of block ciphers for sensor networks.

Additionally, there are some other strategies developed to address various attacks in sensor networks, which further improve the security of sensor networks. They include secure routing [34], secure location verification [35], defending against wormhole attacks [36], preventing false data injection attacks [37], defeating Sybil attacks [38] and recognizing node replication attacks [39].

3. Tame-based Key Predistribution.

In this section, we first develop a new technique using tame automorphism for key predistribution, which is the basis of our framework.

In algebra, a tame automorphism $\varphi_i = (\varphi_{i,1}, \ldots, \varphi_{i,n})$ is given as the following form, where permutations of variables $x_1, \ldots, x_n$ are allowed,

$$\begin{aligned}
(1) &: \varphi_{i,1}(x_1, \ldots, x_n) = x_1 + g_{i,1}(x_2, \ldots, x_n) = y_1 \\
(2) &: \varphi_{i,2}(x_1, \ldots, x_n) = x_2 + g_{i,2}(x_3, \ldots, x_n) = y_2 \\
&\vdots \\
(j) &: \varphi_{i,j}(x_1, \ldots, x_n) = x_j + g_{i,j}(x_{j+1}, \ldots, x_n) = y_j \\
&\vdots \\
(n) &: \varphi_{i,n}(x_1, \ldots, x_n) = x_n = y_n
\end{aligned}$$

Proposition 3.1. *A tame automorphism φ_i is invertible. Its inverse is ${\varphi_i}^{-1} = ({\varphi_{i,1}}^{-1}, \ldots, {\varphi_{i,n}}^{-1})$ with $x_n = {\varphi_{i,n}}^{-1}(y_1, \ldots, y_n) = y_n$ and $x_j = {\varphi_{i,j}}^{-1}(y_1, \ldots, y_n) = y_j - g_{i,j}({\varphi_{i,j+1}}^{-1}(y_1, \ldots, y_n), \ldots, {\varphi_{i,n}}^{-1}(y_1, \ldots, y_n))$ for $j = n-1, \ldots, 1$.*

For example, when $n = 3$, ${\varphi_i}^{-1} = ({\varphi_{i,1}}^{-1}, \ldots, {\varphi_{i,3}}^{-1})$ is as follows:

$$\begin{aligned}
{\varphi_{i,3}}^{-1}(y_1, y_2, y_3) &= x_3 = y_3 \\
{\varphi_{i,2}}^{-1}(y_1, y_2, y_3) &= x_2 = y_2 - g_{i,2}(y_3) \\
{\varphi_{i,1}}^{-1}(y_1, y_2, y_3) &= x_1 = y_1 - g_{i,1}(y_2 - g_{i,2}(y_3), y_3)
\end{aligned}$$

Corollary 3.1. *A tame automorphism is an injective map.*

In our method, we let $n = 2$ and $\varphi_i : K^2 \to K^2$, where K is a $GF(2^l)$; l is a half of a cryptographic key length, and let T be composition map $\varphi_r \ldots \varphi_2\varphi_1$ with $r \geq 4$: $\mathrm{T}(x_1, x_2) = \varphi_r \ldots \varphi_2\varphi_1(x_1, x_2) = (\mathrm{T}_1(x_1, x_2), \mathrm{T}_2(x_1, x_2))$. In composition map T, the order for φ_i's variables with polynomial $g_{i,j}$ is permuted at least three times, for $i = 1, \ldots, r$.

Example 3.1. *Let $r = 4$ and $\varphi_i = (\varphi_{i,1}, \varphi_{i,2})$, $1 \le i \le r$ be as follows:*

$$\begin{array}{ll}
[1] : \varphi_{1,1}(x_1, x_2) = x_1 + g_{1,1}(x_2) = y_1 & \varphi_{1,2}(x_1, x_2) = x_2 = y_2 \\
[2] : \varphi_{2,1}(y_1, y_2) = y_1 = z_1 & \varphi_{2,2}(y_1, y_2) = y_2 + g_{2,2}(y_1) = z_2 \\
[3] : \varphi_{3,1}(z_1, z_2) = z_1 + g_{3,1}(z_2) = w_1 & \varphi_{3,2}(z_1, z_2) = z_2 = w_2 \\
[4] : \varphi_{4,1}(w_1, w_2) = w_1 = q_1 & \varphi_{4,2}(w_1, w_2) = w_2 + g_{4,2}(w_1) = q_2
\end{array}$$

It follows that

$$\begin{aligned}
\mathrm{T}(x_1, x_2) =& \varphi_4\varphi_3\varphi_2\varphi_1(x_1, x_2) = (\mathrm{T}_1(x_1, x_2), \mathrm{T}_2(x_1, x_2)) \\
=& (x_1 + g_{1,1}(x_2) + g_{3,1}(x_2 + g_{2,2}(x_1 + g_{1,1}(x_2))), x_2 + g_{2,2}(x_1 + g_{1,1}(x_2)) \\
& + g_{4,2}(x_1 + g_{1,1}(x_2) + g_{3,1}(x_2 + g_{2,2}(x_1 + g_{1,1}(x_2)))))).
\end{aligned}$$

The order for φ_{i+1}'s variables with polynomial $g_{i+1,j}$ is different from the order for φ_i's variables with polynomial $g_{i,j}$, for $i = 1, 2, 3$. Therefore, the order for φ_i's variables with polynomial $g_{i,j}$ is permuted three times in the example, for $i = 1, \ldots, 4$.

Corollary 3.2. *From Corollary 3.1, we get φ_i is an injective map. It follows that composition map* T *is an injective map as well.*

In addition to T, we let $h(x, y) = (x + y, xy) : K^2 \to K^2$, where K is a $GF(2^l)$.

Proposition 3.2. *$h(x, y)$ is a symmetric and two-one map. That is, $h(\alpha, \beta) = h(\alpha', \beta')$, if and only if $\alpha = \alpha'$, $\beta = \beta'$ or $\alpha = \beta'$, $\beta = \alpha'$.*

Proof:

(a) It is obvious that $h(x, y) = (x + y, xy) = (y + x, yx) = h(y, x)$. Therefore, $h(x, y)$ is symmetric.

(b) Suppose $h(\alpha, \beta) = (\alpha + \beta, \alpha\beta) = (u, v)$. Then, $\alpha + \beta = u$ and $\alpha\beta = v$. When

$$x + y = u \tag{1}$$

$$xy = v \tag{2}$$

Combining Equations (1) and (2), we have

$$x^2 - uv + v = 0 \tag{3}$$

We can factor Equation (3) into $(x - \alpha)(x - \beta) = 0$, and it implies that $(x, y) = (\alpha, \beta)$ or (β, α). Therefore, $h(x, y)$ is a two-to-one map.

From (a) and (b), we can conclude that $h(x, y)$ is a symmetric and two-to-one map.

We further let $f(x, y) = \mathrm{T} \circ h(x, y) = (f_1(x, y), f_2(x, y)) : K^2 \to K^2$, where K is a $GF(2^l)$. These $f_\zeta(x, y)$ for $\zeta = 1, 2$ are bivariate polynomials.

Corollary 3.3. *$f(x, y)$ is a symmetric and two-one map, since $h(x, y)$ is a symmetric and two-one map, and* T *is an injective map. That is, $f(\alpha, \beta) = f(\alpha', \beta')$, if and only if $\alpha = \alpha'$, $\beta = \beta'$ or $\alpha = \beta'$, $\beta = \alpha'$.*

Through proper configuration, we can get two t-degree bivariate polynomials, $f_\zeta(x, y) = \sum_{i,j=0}^{t} a\zeta_{ij} x^i y^j$ for $\zeta = 1, 2$, from $f(x, y)$. For example, in the previous example, we get $t = 99$ if $g_{1,1}$ is a 3-degree polynomial, $g_{2,2}$ is a 33-degree polynomial, $g_{3,1}$ is a 1-degree polynomial, and $g_{4,2}$ is also a 1-degree polynomial. For the sake of presentation, we refer to $f(x, y)$ as a *symmetric-tame map*. In the following, we assume that all the symmetric-tame maps are t-degree (it means that both $f_\zeta(x, y)$ for $\zeta = 1, 2$ are t-degree).

For key predistribution, the setup server randomly generates a symmetric-tame map $f(x, y)$. Here "randomly" means that 1) r is a random number which is not smaller than 4, 2) each polynomial $g_{i,j}$ is generated randomly, 3) the order for φ_i's variables with polynomial $g_{i,j}$ in composition map T, for $i = 1, \ldots, r$, is permuted randomly and at least three times, and 4) t-degree randomly consists of t's factors depending on r and how $\mathrm{T}_\zeta(x, y)$ for $\zeta = 1, 2$ is composed of. For example, in Example 3.1 with $t = 99$ and $r = 4$, the degrees of $g_{1,1}$, $g_{2,2}$, $g_{3,1}$ and $g_{4,2}$, can be respectively 3, 33, 1, 1 or 33, 1, 3, 1 or 9, 11, 1, 1, etc.

In our method, each sensor node has a unique ID. For each sensor node α, the setup server computes a symmetric-tame map share of $f(x, y)$: $f(\alpha, y) = (f_1(\alpha, y), f_2(\alpha, y))$, and store $f(\alpha, y)$ in the memory of node α. Any two sensor nodes α and β can establish a pairwise key through $f(\alpha, \beta) = (f_1(\alpha, \beta), f_2(\alpha, \beta)) = (f_1(\beta, \alpha), f_2(\beta, \alpha)) = f(\beta, \alpha)$. Node α can compute $f(\alpha, \beta) = (f_1(\alpha, \beta), f_2(\alpha, \beta))$ by evaluating $f(\alpha, y)$ at point β, and node β can also compute the common key $f(\alpha, \beta) = f(\beta, \alpha) = (f_1(\beta, \alpha), f_2(\beta, \alpha))$ by evaluating $f(\beta, y)$ at point α. Both of them then use $f_1(\alpha, \beta)||f_2(\alpha, \beta) = f_1(\beta, \alpha)||f_2(\beta, \alpha)$ as their pairwise key. Since $f(x, y)$ is a symmetric and two-one map, the pairwise key is uniquely shared between them. They both can authenticate each other through challenge-response protocol with the pairwise key. Therefore, this tame-based key predistribution approach can provide deterministic authentication service in sensor networks.

Due to the use of two t-degree polynomials $f_\zeta(x, y)$ for $\zeta = 1, 2$, the security analysis for the approach is t-collusion resistant. In other words, an attacker knows nothing about the pairwise key between any two non-compromised nodes if he (or she) compromises no more than t nodes.

In order to establish a pairwise key, a sensor node needs to evaluate the symmetric-tame map share at the ID of the other sensor node. This involves evaluating two t-degree polynomials over $GF(2^l)$. Therefore, the computation complexity for a sensor node is $3t - 1$ multiplications and $2t$ additions over $GF(2^l)$. The addition over $GF(2^l)$ is simply the exclusive-OR (XOR) of two$l$-bit blocks. The multiplication q of two elements $o = (o_{l-1}, \ldots, o_1, o_0)$ and $p = (p_{l-1}, \ldots, p_1, p_0)$ over $GF(2^l)$, which is defined with an irreducible polynomial $g(x) = g_l x^l + \ldots + g_1 x + g_0$ over $GF(2^l)$ $(g = (g_l, \ldots, g_1, g_0))$ with degree l stored in each sensor node, can be computed by shift and XOR operations as follows [40].

1) Let $i = 1$. If $p_0 = 0$, then $q = 0$ else $q = o$.
2) Shift o 1 bit to the left.
3) If $o_l = 1$, then $o = o \oplus g$.
4) If $p_i = 1$, then $q = q \oplus o$.
5) Let $i = i + 1$.
6) If $i < l$, go to 2).
7) Output q.

The storage space needed for each sensor node is $2 \times (t + 1) \times l + l$ bits for storing the t-degree symmetric-tame map share and $g(x)$. There is no communication overhead in the approach for pairwise key establishment. Therefore, compared with the conventional polynomial-based approach which can only provide probabilistic authentication, our method can offer sensor networks deterministic authentication service without increasing the computation, memory and communication overhead.

4. Tame Pool-based Key Predistribution.

The tame-based approach presented above is t-collusion resistant. However, the value of t is limited by memory constraint on sensor node (e.g., $t = 98$). For large-scale sensor networks (e.g., the number of sensor nodes is 10,000), this limited collusion resistance property may not be enough. Therefore, in this section, we further develop a general framework for dealing with key predistribution of large-scale sensor networks, in the basis of the tame-based approach. We refer to it as tame pool-based key predistribution because there exists a pool of *symmetric-tame maps* used in the framework. As a result, the polynomial used in any existing polynomial pool-based scheme can be substituted by the symmetric-tame map to offer deterministic authentication service. In particular, we will introduce a new instantiation based on the framework in next section.

The process of the framework consists of three phases: symmetric-tame map predistribution, direct key establishment and indirect key establishment. The setup server distributes symmetric-tame map shares to each sensor node in the symmetric-tame map predistribution phase. After deployment, two sensor nodes will try to establish a direct pairwise key through direct key establishment phase first. If it successes, the process stops; otherwise, the two nodes perform indirect key establishment to establish an indirect pairwise key with assistance of other nodes.

(1) Symmetric-tame map predistribution phase

The setup server randomly generates a set $\mathcal{F}$ of *symmetric-tame maps* and assigns each symmetric-tame map a unique ID to identify the different maps. For each sensor node α, the setup server selects a subset of symmetric-tame maps from $\mathcal{F}$, and then distributes the corresponding shares of these maps to the node.

(2) Direct key establishment phase

In this phase, two sensor nodes first determine whether they share a common symmetric-tame map. To accomplish this, one of them broadcasts all the identifiers of symmetric-tame maps which it has shares of. In order to prevent traffic analysis attack, this node may use encrypted broadcasts [1,3], instead of revealing the identifiers of these maps directly. If the other node also has one of the symmetric-tame map shares associated with the broadcasts, it computes the corresponding pairwise key based on tame-based approach presented in Section 3, and then responds to the source node through challenge-response (It may add the ID of the corresponding map to its response packet.). They can authenticate each other in this phase, since the pairwise key is only shared between them.

(3) Indirect key establishment phase

If two nodes cannot establish a pairwise key through the direct key establishment phase, they need to run the indirect key establishment. They have to discover one or more intermediate nodes to assist them in establishing an indirect pairwise key. A possible way is to have one of them (called source node) to send a request to a number of nodes which have direct pairwise keys with it. If one of these intermediate nodes can also establish direct key with the other node (called destination node), this intermediate node can be used to help establishing an indirect pairwise key. Otherwise, the request would be forwarded continuously by these intermediate nodes. The communication overhead incurred by such procedure may be considerable in this phase.

In order to get high probability to establish a direct pairwise key between two nodes and good resilience against node compromises, the main concern in the framework is how to select the subset of symmetric-tame maps from $\mathcal{F}$ for each node in the predistribution phase. In next section, we propose a scheme, based on the framework, where the selection of the subset of symmetric-tame maps from $\mathcal{F}$ for a node is according to the location of the node.

5. Local Symmetric-tame Maps Predistribution Scheme.

Now, we introduce a key predistribution scheme, the *local symmetric-tame maps predistribution scheme*, based on the framework and location information of sensor nodes. We assume that the setup server can predetermine sensor nodes' locations, and each sensor node can acquire its actual deployment location after being deployed. This assumption is reasonable, since it is possible to predict sensor nodes' locations in many applications in static sensor networks. Moreover, there have been many tasks focusing on the development of location system of sensor nodes [4,41].

The basic idea of this scheme is to combine the expected location information of sensor nodes with the improving random subset assignment scheme with deployment locations (IRSS) [25] for key predistribution. In the following, we first describe the deployment model used in our scheme, and then present the local symmetric-tame maps predistribution scheme.

5.1. **Deployment model.** We assume that the sensor network is deployed in a target field of a two-dimension area. However, this model can be extended to a three-dimension space straightforwardly. In the target field, the location of a sensor node can be represented by a coordinate. The setup server can predetermine each sensor's location called *expected location*. After being deployed, a sensor node is at *deployment location*, which is its actual location. The expected location and the deployment location of a sensor node may be different. The difference is referred to as the *deployment error* for the node. This model is similar to that in IRSS [25].

In the model, the transmission range of a sensor node is denoted as T_r. Two sensor nodes are called *neighbors* if they are within the transmission range of each other.

For simplicity, we analyze the performance of our scheme under the scenario that a sensor node randomly locates anywhere in a circle centered at the node's expected location with a radius e called the *maximum deployment error* (In our model, we assume that the setup server can predetermine the maximum deployment error e.). However, our scheme can be applied to any deployment model.

5.2. **Local symmetric-tame maps predistribution scheme.** In the proposed scheme, the setup server locally chooses symmetric-tame maps in the predistribution phase for each sensor node according to its expected location. Specifically, for each sensor node, the setup server randomly picks a subset of symmetric-tame maps from the circle centered at its expected location with a predetermined radius, and stores the corresponding shares in the node. The analysis indicates that the proposed scheme outperforms previous ones.

The Proposed Scheme. We now describe the local symmetric-tame maps predistribution scheme in details below. We use the notations in Table 1.

Phase 1: Symmetric-tame map predistribution

The setup server first predetermine e, b', b and $|\mathcal{F}|$, where b' is limited by memory constraint and the range of b can be referred to the analysis in Section 6.1. Then the setup server randomly generates a set $\mathcal{F}$ of symmetric-tame maps: $K^2 \to K^2$, where K is a $GF(2^l)$, and each of them is associated with a unique location in the target field. The location's coordinate can also be assigned as the ID of the corresponding symmetric-tame map. These symmetric-tame map locations are evenly distributed over the whole target field. For each sensor node, the setup server randomly chooses a subset of b symmetric-tame maps from the circle centered at the node's expected location with radius $(e + \delta + \omega)$, where we can compute ω according to the analysis in Section 6.1 and $\delta = ((e + \omega) \times \sqrt{b'/b}/(1 - \sqrt{b'/b})$, and then predistributes the corresponding symmetric-tame map shares and their locations to the sensor node. For convenience, we call this circle *local circle*. It is shown in Figure 1(a).

Phase 2: Symmetric-tame map prioritization

Each sensor node first acquires its deployment location after being deployed. Then, according to its deployment location and the related locations of the predistributed symmetric-tame map shares, the node prioritizes these symmetric-tame map shares by

TABLE 1. Notations

Notation	Description
$f(x, y)$	symmetric-tame map
$\|\mathcal{F}\|$	the size of symmetric-tame map pool
b	the number of symmetric-tame maps predistributed in each sensor node before deployment
b'	the number of symmetric-tame maps kept in each sensor node's storage after the prioritization
e	the maximum deployment error
δ	the associated locations of b' symmetric-tame maps are no bigger than δ away from the corresponding node's deployment location
N	the total number of sensor nodes in the network
m	the average number of neighbor sensor nodes
A	the area of the target field
P	the probability to establish a direct pairwise key between two neighbors
P_c	the probability that a direct pairwise key between two non-compromised nodes is compromised
ω	*adaptive parameter*, which is described in details later

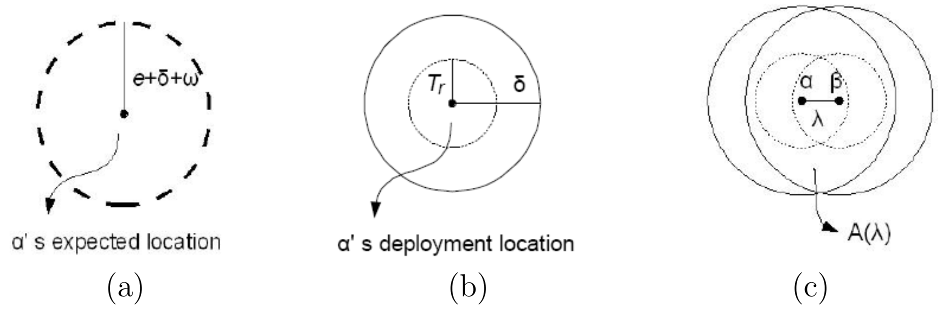


FIGURE 1. (a) Local circle, (b) share circle and (c) shared area of neighbors

their distances to its deployment location. A symmetric-tame map share has higher priority if it is closer to the node. The node then keeps the b' highest-priority symmetric-tame map shares in its memory and removes others.

Phase 3: Pairwise key establishment

This phase can be accomplished by performing the direct key establishment and indirect key establishment phases in the tame pool-based key predistribution framework. Two sensor nodes attempting to establish a pairwise key first determine whether they both have shares of a common symmetric-tame map. To achieve this, one of them broadcasts the corresponding symmetric-tame maps' IDs of b' shares in its memory. If the other node has one of the symmetric-tame map shares associated with the broadcasts, it computes the corresponding pairwise key based on the tame-based approach, and responds to the source node through challenge-response. Since the pairwise key is uniquely shared between them, they both can authenticate each other in the phase. If they fail to establish a direct pairwise key, then they need to start the indirect key establishment phase to discover one or more intermediate nodes which can help them to setup an indirect pairwise key.

Phase 4: Addition and revocation

Due to the inherence that sensor nodes easily get damaged and compromised, it is very likely to add new nodes into the network to replace the damaged or compromised nodes during the lifetime of a sensor network. Whenever a new node is added, the setup server first performs the above symmetric-tame map predistributon phase based on node's expected location. Then, after being deployed, the node simply starts the above symmetric-tame map prioritization phase. There is no communication overhead needed between the setup server and the deployed nodes for adding node. Once a compromised node is detected, it is necessary to revoke that node for reducing the damage of the system. The revocation can be done by the setup server to broadcast a list of revoked nodes' IDs. Each deployed node records the IDs of revoked nodes which share at least a common symmetric-tame map with it. If there is a symmetric-tame map compromised – more than t nodes which share the same symmetric-tame map are compromised, a non-compromised sensor node which has share of this symmetric-tame map discards the corresponding share.

6. Analysis.

6.1. Probability of establishing direct pairwise keys between neighbors.

We assume that there are on average m nodes in each sensor node's transmission range. Thus, the density of the network can be estimated as $(m + 1)/\pi T_r^2$. We use the transmission range T_r as the basic unit for distance metric. That is, $T_r = 1$. Then, on average, there are $(m + 1)$ sensor nodes in area of $\pi T_r^2 = \pi$. If the total number of sensor nodes in the network is N, the area of the target field can be estimated as $A = N\pi/(m + 1)$.

A sensor node α is predistributed b symmetric-tame map shares in its memory during the predistribution phase and leaves b' shares after the prioritization phase. The related locations of these b' shares are not further than δ from the deployment location of node α, where δ is $((e + \omega) \times \sqrt{b'/b}/(1 - \sqrt{b'/b})$. The circle centered at node's deployment location with radius δ is called the *share circle* as shown in Figure 1(b). According to the prioritization process, the b' shares are exactly the predistributed symmetric-tame map shares whose related locations are within the share circle.

Figure 1(c) shows the overlapped area $A(\lambda)$ of the share circles of neighboring node α and node β, the distance between which is λ. Since $\lambda = 1$ and $\delta \le 1/2$, $A(\lambda)$ does not exist. Thus, we only consider the condition that $\delta > 1/2$ in our model. Then, the size of the area can be estimated as

$$A(\lambda) = 2 \times \delta^2 \times \cos^{-1}(\lambda/2\delta) - \lambda \times \sqrt{\delta^2 - (\lambda/2)^2}$$

For node α and node β, the number of predistributed symmetric-tame map shares whose related locations are in the area $A(\lambda)$ can be estimated by $M(\lambda) = \lfloor b \times A(\lambda)/\pi \times (e + \delta + \omega)^2 \rfloor$. That is, in the memory of node α and node β, there are shares of about $M(\lambda)$ symmetric-tame maps which locate in the overlapped area. In addition, the total number of symmetric-tame maps which locate in the area can be estimated by $M_t(\lambda) = \lfloor |\mathcal{F}| \times A(\lambda)/A \rfloor$. Then, the probability that node α and node β have a common symmetric-tame map can be estimated as

$$P(\lambda) = 1 - \frac{[(M_t(\lambda) - M(\lambda))!]^2}{M_t(\lambda)!(M_t(\lambda) - 2M(\lambda))!}$$

Therefore, the probability to share a common symmetric-tame map between node α and any of its neighbors can be estimated as

$$P = \int_0^{T_r} \int_0^{2} \frac{P(\lambda) \times \lambda}{\pi T_r^2} d\lambda d\theta = 2 \int_0^1 P(\lambda)\lambda d\lambda$$

This indicates the probability of establishing a direct pairwise key between node α and any of its neighbors. For simplicity, we estimate the probability to establish a direct pairwise key between any two neighbors by probability P of the node expected to be deployed at the center of the target field having a direct pairwise key with its neighbor in our simulation.

Adaptive parameter ω – We now describe this parameter in details below. There are four necessary conditions in our model:

1. $\delta > 1/2$, since $A(\lambda)$ does not exist when $\lambda = 1$ and $\delta \leq 1/2$, and it is expected that neighbors can have the opportunity to share common symmetric-tame map.
2. $\pi(e+\delta+\omega)^2 \leq A$. The circle from which we pick a set of b symmetric-tame maps for a sensor node should be no bigger than the size of the target field.
3. $M_t(\lambda) \geq M(\lambda)$. The total number of symmetric-tame maps in the overlapped area $A(\lambda)$ should be no smaller than the number of predistributed symmetric-tame maps in the area for both node α and node β.
4. $\omega \geq 0$. This makes the model function well. Considering the situation that the deployment location of a sensor node is e away from its expected location, $\omega < 0$ will fail the model.

From condition 1, we get $\delta > 1/2$; $\delta = ((e+\omega) \times \sqrt{b'/b}/(1-\sqrt{b'/b})$. Then,

$$\begin{aligned}
&\left((e+\omega) \times \sqrt{b'/b}\right) \Big/ \left(1-\sqrt{b'/b}\right) > \frac{1}{2} \\
\Rightarrow &\left((e+\omega) \times \sqrt{b'/b}\right) > \frac{1}{2}\left(1-\sqrt{b'/b}\right) \\
\Rightarrow &(e+\omega) > \frac{1}{2}\left(\sqrt{b'/b}-1\right) \\
\Rightarrow &\omega > \frac{1}{2}\left(\sqrt{b'/b}-1\right) - e
\end{aligned} \tag{4}$$

From condition 2, we have $\pi(e+\delta+\omega) \leq A$. Then,

$$\begin{aligned}
&(e+\delta+\omega)^2 \leq A/\pi \\
\Rightarrow &e+\delta+\omega \leq \sqrt{A/\pi} \\
\Rightarrow &e + \left((e+\omega) \times \sqrt{b'/b}\right) \Big/ \left(1-\sqrt{b'/b}\right) + \omega \leq \sqrt{A/\pi} \\
\Rightarrow &e+\omega \leq \sqrt{A/\pi}\left(1-\sqrt{b'/b}\right) \\
\Rightarrow &\omega \leq \sqrt{A/\pi}\left(1-\sqrt{b'/b}\right) - e
\end{aligned} \tag{5}$$

From condition 3, we have $M_t(\lambda) \geq M(\lambda)$. Additionally, $M_t(\lambda) = \lfloor |\mathcal{F}| \times A(\lambda)/A \rfloor$ and $M(\lambda) = \lfloor b \times A(\lambda)/\pi \times (e+\delta+\omega)^2 \rfloor$. It follows that

$$\begin{aligned}
&|\mathcal{F}| \times A(\lambda)/A \geq b \times A(\lambda)/(\pi \times (e+\delta+\omega)^2) \\
\Rightarrow &|\mathcal{F}| \times (\pi \times (e+\delta+\omega)^2) \geq b \times A \\
\Rightarrow &(e+\delta+\omega)^2 \geq b \times A/(\pi \times |\mathcal{F}|) \\
\Rightarrow &e + \left((e+\omega) \times \sqrt{b'/b}\right) \Big/ \left(1-\sqrt{b'/b}\right) + \omega \geq \sqrt{(b \times A)/(\pi \times |\mathcal{F}|)} \\
\Rightarrow &e+\omega \geq \sqrt{(b \times A)/(\pi \times |\mathcal{F}|)}\left(1-\sqrt{b'/b}\right) \\
\Rightarrow &\omega \geq \sqrt{(b \times A)/(\pi \times |\mathcal{F}|)}\left(1-\sqrt{b'/b}\right) - e
\end{aligned} \tag{6}$$

For the sake of presentation, we call this ω *adaptive parameter*, which satisfies $\omega \geq 0$ and Equations (4)-(6) with fixed b, b', e, A and $|\mathcal{F}|$. We can get the minimum ω from the range of the adaptive parameter. We refer to this minimum ω as $\omega_{\min}$. In the following, when other parameters, b, b', e, A and $|\mathcal{F}|$ are fixed, we will use $\omega = \omega_{\min}$ in our scheme without explicit statement. The reason is that it can get the highest probability of sharing a direct pairwise key between two neighbors according to the result of our analysis: The smaller the local circle is, the better the performance of P is. For instance, given that $b = 11$, $b' = 2$, $e = 5$ and $|\mathcal{F}| = 20$, from Equations (4)-(6), and $\omega \geq 0$, it follows that $2.6402\ldots \leq \omega \leq 5.3021\ldots$, then we can get $\omega = \omega_{\min} = 2.65$, which is used in the symmetric-tame map predistribution phase.

Here we derive the range of b. Since the adaptive parameter ω needs to satisfy the Equations (4)-(6), and $\omega \geq 0$ at the same time, from the intersection of Equations (4) and (5), the intersection of Equations (5) and (6), and the intersection of Equation (5) and $\omega \geq 0$, we have Equations (7)-(9) respectively.

$$\frac{1}{2}\left(\sqrt{b'/b} - 1\right) - e < \sqrt{A/\pi}\left(1 - \sqrt{b'/b}\right) - e \tag{7}$$

$$\sqrt{(b \times A)/\pi \times |\mathcal{F}|}\left(1 - \sqrt{b'/b}\right) - e \leq \sqrt{A/\pi}\left(1 - \sqrt{b'/b}\right) - e \tag{8}$$

$$0 \leq \sqrt{A/\pi}\left(1 - \sqrt{b'/b}\right) - e \tag{9}$$

From Equation (7), it follows that

$$b < \frac{4A \times b'}{\pi} \tag{10}$$

From Equation (8), it follows that

$$b \leq |\mathcal{F}| \tag{11}$$

This is obvious, since we choose b symmetric-tame maps from the symmetric-tame map pool $\mathcal{F}$. Hence, it can be omitted in the following computation.

From Equation (9), it follows that

$$b \geq \frac{b'}{\left(1 - e \times \sqrt{\pi/A}\right)^2} \tag{12}$$

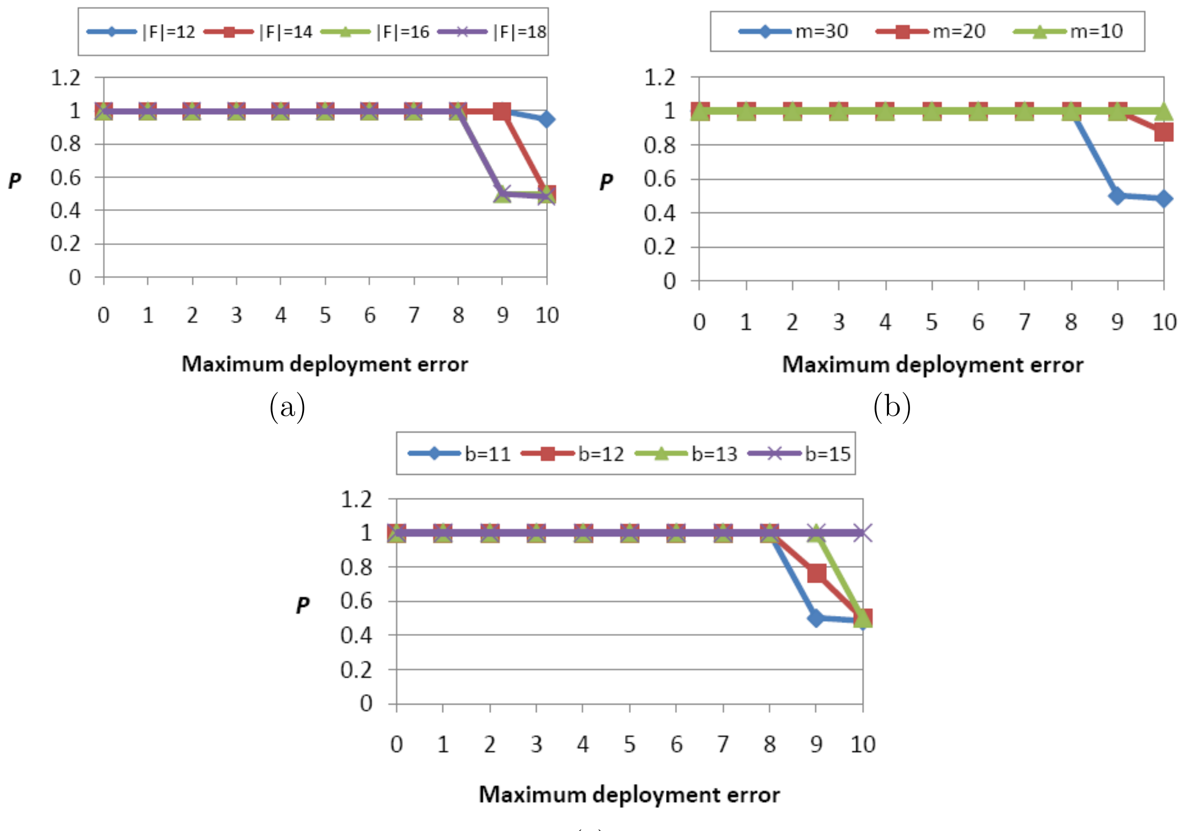
Take the intersection of Equations (10) and (12). Then, we can get the range of b:

$$\frac{b'}{\left(1 - e \times \sqrt{\pi/A}\right)^2} \leq b < \frac{4A \times b'}{\pi} \tag{13}$$

In the following analysis, we assume that the total number of sensor nodes in the network is $N = 10,000$, on average there are $m = 30$ neighbors in each node's transmission range, and the memory overhead for each node after the prioritization phase is equivalent to 200 cryptographic keys unless otherwise specified. For convenience, we configure b to fit the range of b as $e = 10$ given other parameters fixed. Figure 2(a) shows the probability of establishing a direct pairwise key between any two neighbors for different e and $|\mathcal{F}|$ given that $b = 11$ and $b' = 2$. Obviously, the probability is extremely high especially when the maximum deployment error e is less than nine times of the transmission range. We can also see the probability becomes higher when the size of the symmetric-tame map pool $\mathcal{F}$ decreases given certain b, b' and e.

Figure 2(b) presents the probability of establishing a direct pairwise key between any two neighbors for different e and m given that $b = 11$, $b' = 2$ and $|\mathcal{F}| = 18$. Given certain b, b', $|\mathcal{F}|$ and e, we can see that the probability grows higher when the average number of neighbors m decreases. The area A of target field increases as m decreases, with fixed total number of sensor nodes in the network. Thus, Figure 2(b) implies that the larger the ratio of the area of target field to the size of the local circle is, the more the performance of probability P can be improved, given certain b, b', $|\mathcal{F}|$ and e.

Figure 2(c) shows the probability of establishing a direct pairwise key between any two neighbors for different e and b given that $b' = 2$ and $|\mathcal{F}| = 18$. We can see the increase of the number of predistributed symmetric-tame maps in the predistribution phase will increase probability P with fixed b', m, $|\mathcal{F}|$ and e.



(a) (b) (c)

FIGURE 2. Probability to establish a direct pairwise key between two neighbors for different e and (a) different $|\mathcal{F}|$ given that $b = 11$ and $m = 30$; (b) different m given that $b = 11$ and $|\mathcal{F}| = 18$; (c) different b given that $|\mathcal{F}| = 18$ and $m = 30$. $b' = 2$.

6.2. **Resilience against node compromises.** We assume that an attacker can read the secrets stored in node's memory once he (or she) compromises the node. Additionally, in the security analysis, an attacker knows nothing about the direct pairwise key between two non-compromised nodes if he (or she) compromises no more than t nodes which share the same symmetric-tame map generating that pairwise key.

In this section, we consider the security against node compromises in two ways. One is that an attacker locally compromises sensor nodes in a certain area. This kind of attack is called *local attack*. The other is *random attack*, where an attacker randomly compromises sensor nodes in the entire network.

In local attack, in order to compromise the direct pairwise key between two non-compromised nodes in a particular area, an attacker needs to compromise more than t nodes sharing the same symmetric-tame map which generates the pairwise key. Nevertheless, since symmetric-tame maps are selected locally for each sensor node, the attacker cannot compromise the security of any other area even though he (or she) compromises the security of a particular area.

Consider that an attacker launches random attack. Obviously, there are more symmetric-tame map shares stored in node before the prioritization phase than after it. In other words, an attacker can learn more secrets from a node if he (or she) compromises the node before its prioritization phase. Thus, we consider the resilience against node compromises for random attack in two cases.

Case 1: In this case, we assume there is no node compromised between the predistribution phase and the prioritization phase. After the prioritization, the nodes in the network may be compromised by an attacker. Assume that the attacker randomly compromises N_c nodes in the network after the prioritization phase. For any symmetric-tame map f in $\mathcal{F}$, the probability that a node has share of f is $b'/|\mathcal{F}|$. Thus, the probability that exactly i out of N_c compromised nodes own shares of this map can be estimated as

$$P_c(i) = \frac{N_c!}{(N_c - i)!i!}\left(\frac{b'}{|\mathcal{F}|}\right)^i \left(1 - \frac{b'}{|\mathcal{F}|}\right)^{N_c - i}$$

If an attacker knows more than t different shares of a symmetric-tame map, he (or she) can evaluate any pairwise key generated through the map. Therefore, the probability that f is compromised can be estimated as $P_c = 1 - \sum_{i=0}^{t} P_c(i)$. As a result, the probability that a direct pairwise key between two non-compromised nodes is compromised can be estimated by P_c, since f is any symmetric-tame map in $\mathcal{F}$.

We use the probability that a direct pairwise key between two non-compromised nodes is compromised to evaluate the resilience against node compromises. Figure 3 shows the relationship between the probability that a direct pairwise key between two non-compromised nodes is compromised, and the number of compromised nodes under different configurations given that $e = 5$. We assume that the probability to establish a direct pairwise key between two neighbors is 0.33 and there are $b' = 2$ shares stored in each node after the prioritization. For different values of b, the size of the symmetric-tame map pool $\mathcal{F}$ is configured to maintain probability P being 0.33. It shows that we can achieve arbitrarily great security resilience by increasing the number of predistributed symmetric-tame maps in the predistribution phase.

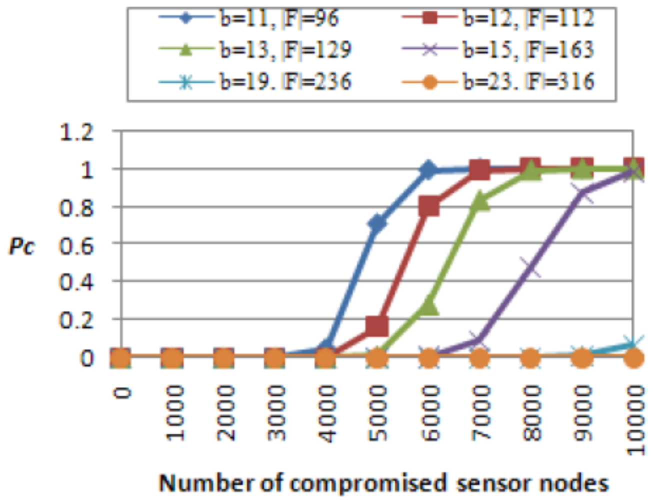


FIGURE 3. Probability that a direct pairwise key between two non-compromised nodes is compromised vs. the number of compromised sensor nodes under the different configurations in Case 1 given that $e = 5$ and $b' = 2$. Assume $P = 0.33$.

Case 2: In this case, we assume an attacker can randomly compromise N_b nodes between the predistribution phase and the prioritization phase, and randomly compromise N_c nodes after the prioritization. For any symmetric-tame map f in $\mathcal{F}$, the probability that a node compromised before the prioritization has share of f is $b/|\mathcal{F}|$ and the probability that a node compromised after the prioritization has share of f is $b'/|\mathcal{F}|$. Then, we can estimate the probability that exactly k out of $N_b + N_c$ compromised nodes have shares of the map by

$$P_c(k) = \sum_{i+j=k} \frac{N_b!}{(N_b-i)!i!}\left(\frac{b}{|\mathcal{F}|}\right)^i\left(1-\frac{b}{|\mathcal{F}|}\right)^{N_b-i}\frac{N_c!}{(N_c-j)!j!}\left(\frac{b'}{|\mathcal{F}|}\right)^j\left(1-\frac{b'}{|\mathcal{F}|}\right)^{N_c-j}$$

Similarly, the probability that the map f is compromised can be calculated by $P_c = 1 - \sum_{k=0}^{t} P_c(k)$. Thus, the probability that a direct pairwise key between two non-compromised nodes is compromised can also be estimated by P_c, since f is any symmetric-tame map in $\mathcal{F}$.

Figure 4 illustrates the relationship between the probability that a direct pairwise key between two non-compromised nodes is compromised and the number of compromised nodes under the different configurations in the second case given that $e = 5$. We also assume that the probability to establish a direct pairwise key between two neighbors is 0.33 and $b' = 2$. We consider two situations: one is $N_b = 100$ in Figure 4(a) and the other is $N_b = 500$ in Figure 4(b). It turns out that even under this case, although the resilience in situation 2 is not as great as that in situation 1 under the same configuration, we can still get arbitrarily great security resilience by increasing b.

In general, a sensor node is more likely to be compromised after deployment than before it. Moreover, after being deployed, a node starts immediately the prioritization phase once it acquires its deployment location. Therefore, it is difficult for an attacker to compromise a large number of sensor nodes during the short period of time between node deployment and prioritization.

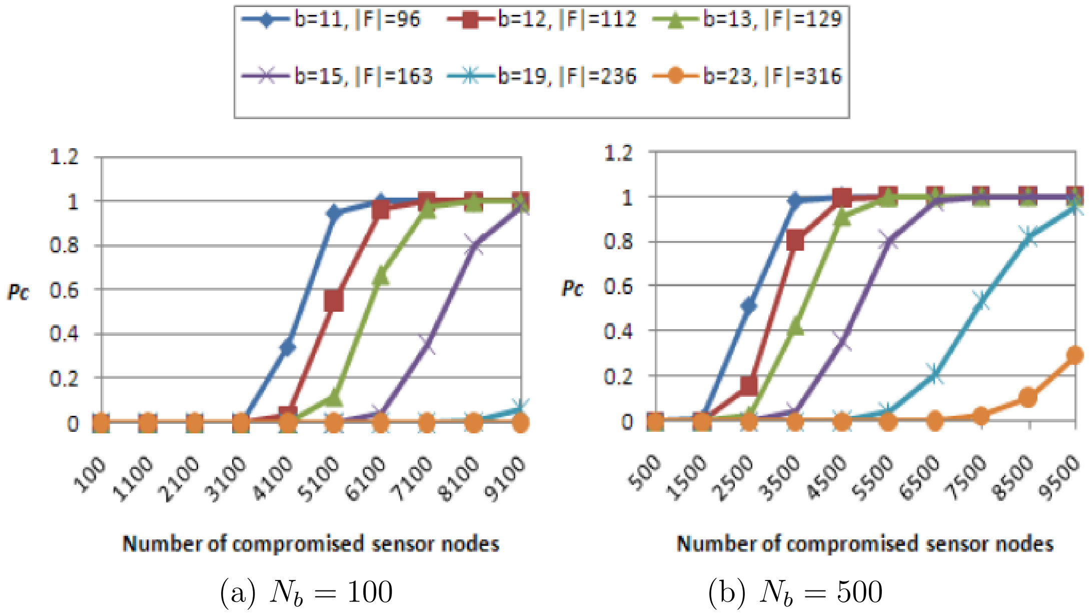


FIGURE 4. Probability that a direct pairwise key between two non-compromised nodes is compromised vs. the number of compromised sensor nodes under different configurations in Case 2 given that $e = 5$ and $b' = 2$. Assume $P = 0.33$.

6.3. **Comparison.** In this paper, we mainly compare the performance of our scheme with the well-known schemes in Liu and Ning [25]: the closest pairwise keys scheme (CPKS), the closest polynomials predistribution scheme (CPPS) and IRSS. These schemes use either predeployment or postdeployment knowledge to improve the pairwise key predistribution. However, in our scheme, we further combine the expected location and deployed location information to do the key predistribution. As a result, both the probability to establish a direct pairwise key between two neighbors and the resilience against node compromises in our scheme outperform those schemes. Moreover, due to the use of the tame-based approach, our scheme can provide deterministic authentication service. However, CPPS and IRSS can only offer probabilistic authentication, since they are based on the polynomial-based approach.

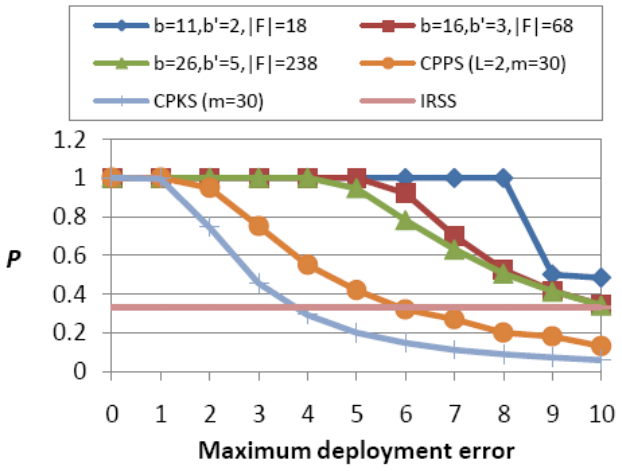


FIGURE 5. Compare the probability to establish a direct pairwise key between two neighbors between our scheme and the schemes in [25], under different configurations. The parameters in IRSS are the same with those in our scheme given $c = b$ and $c' = b'$.

Figure 5 compares the probability to establish a direct pairwise key between two neighbors between our scheme and the schemes in [25] under different configurations. Assume each node has memory overhead equivalent to 200 cryptographic keys in each scheme. For each configuration, the size of the polynomial pool in IRSS is configured to make probability P be 0.33, and the size of the symmetric-tame map pool $\mathcal{F}$ in our scheme is configured to be the same as the polynomial pool size in IRSS with $b = c$ and $b' = c'$. Obviously, probability P in our scheme is significantly higher than that in other schemes [25].

Moreover, given the limits on the maximum deployment error e, network density, and the memory overhead for the key predistribution, probability P in CPKS is fixed. The previous two parameters depend on specific application and the last parameter is constrained by the memory size of sensor node. Thus, it is hard to adapt these parameters to achieve higher probability P. However, in each configuration of our scheme, given the constraints above, we can always get an arbitrarily high probability P by increasing the number of predistributed symmetric-tame maps in the predistribution phase, as shown in Figure 2(c). Therefore, a designer can choose a proper configuration depending on the need of the system. Although in CPPS, given the above constraints, probability P can also be improved by increasing the cell side length L, it will reduce the resilience against node compromises. On the contrary, in our scheme, the increase of b will not affect the resilience of security when there is no node compromised before the prioritization. We can see this from the analysis in Section 6.2, since the resilience only depends on t and the ratio $b'/|\mathcal{F}|$. In addition, although in IRSS, increasing c can also improve probability P, it is obvious that the performance of our scheme is greatly better than IRSS under the same configuration, as shown in Figure 5.

Precisely combining the expected location and deployment location information for key predistribution, not only significantly improve the performance of probability P, but can also achieve greater resilience against node compromises. Figure 6 compares the probability that a direct pairwise key between two non-compromised nodes is compromised between our scheme, the schemes in Liu and Ning [25], the basic probabilistic scheme [7], the q-composite scheme [8] and the random subset assignment scheme [12]. Considering that there is no node compromised before the prioritization phase. We assume that probability P is 0.33 in each scheme. We can see that probability P_c in our scheme can achieve significantly greater security guarantee than other schemes. The smaller the maximum deployment error e is, the greater the improvement of the resilience in our scheme can be.

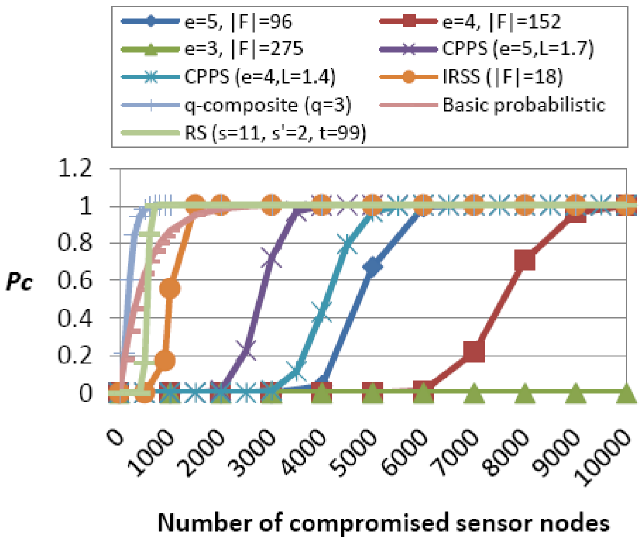


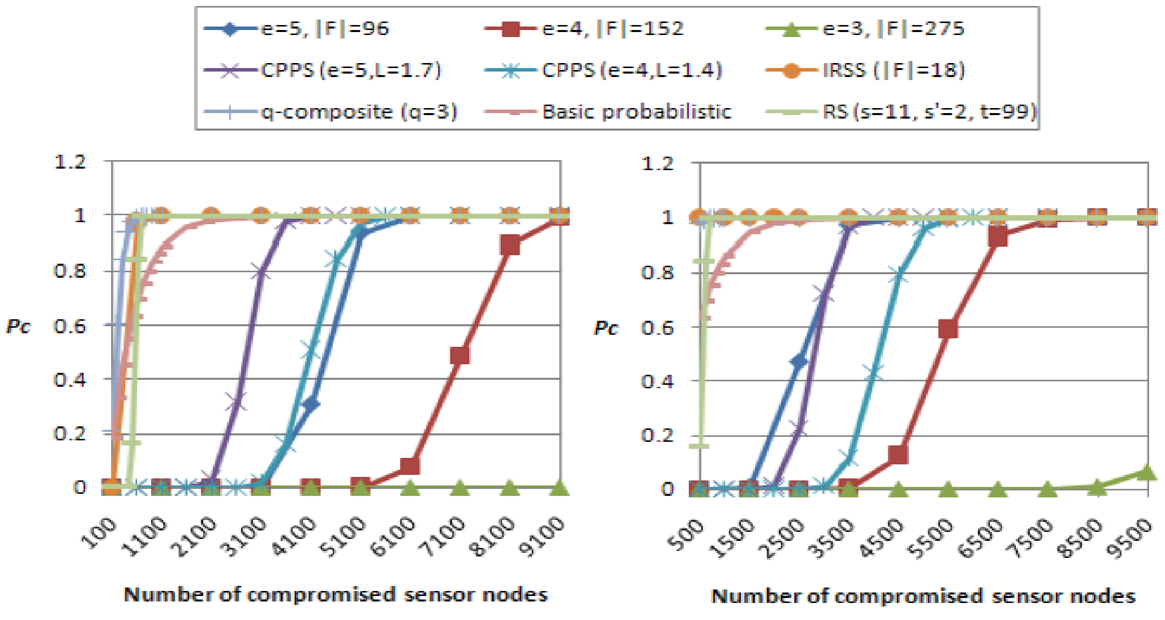
FIGURE 6. Compare the probability that a direct pairwise key between two non-compromised nodes is compromised under the different numbers of compromised sensor nodes between our scheme and other schemes, considering there is no node compromised before the prioritization. Assume $b = c = 11$ and $b' = c' = 2$ in our scheme and IRSS. $P = 0.33$.

Furthermore, given a certain key pool size, polynomial pool size, cell side length L, maximum deployment error, and memory constraint on sensor node, probability P is fixed in the basic probabilistic scheme, the q-composite scheme, the random subset assignment scheme and CPPS. However, our scheme can still achieve arbitrarily high probability P by increasing b, given a fixed symmetric-tame map pool size, e, and memory constraint on sensor node, as shown in Figure 2(c). In other words, the basic probabilistic scheme, the q-composite scheme, the random subset assignment scheme, and CPPS need to do trade-off between the performance of the probability to establish a direct pairwise key between two neighbors and the resilience against node compromises. However, our scheme can achieve arbitrarily high probability to establish a direct pairwise key between two neighbors without decreasing the resilience of security.

In addition, in the basic probabilistic scheme, the q-composite scheme, the random subset assignment scheme and CPPS, probability P_c is fixed, given the required probability

P, memory constraint, network density and e. However, in our scheme, we can get arbitrarily great resilience of security by increasing b given the constraints above, as shown in Figure 3. Despite by increasing c, IRSS can also get better resilience, the resilience in our scheme is significantly greater than that in IRSS, as shown in Figure 6.

Figure 7 compares the probability that a direct pairwise key between two non-compromised nodes is compromised between our scheme and other schemes, considering that there are some nodes compromised before the prioritization. Similarly, we assume that probability P is 0.33 in each scheme. We can see that the result of the comparison in this case is similar with that in the previous case that there is no node compromised before the prioritization. From Figure 7(b) and Figure 4(b), it is obvious that the security performance of our scheme can be increased arbitrarily by increasing b given other fixed parameters; therefore, even though the number of compromised nodes is large before the prioritization, our scheme can always provide significantly better performance of security than other schemes.



(a) $N_b = 100$ (b) $N_b = 500$

FIGURE 7. Compare the probability that a direct pairwise key between two non-compromised nodes is compromised under the different numbers of compromised sensor nodes between our scheme and other schemes, considering that there are some nodes compromised before the prioritization. Assume $b = c = 11$ and $b' = c' = 2$ in our scheme and IRSS. $P = 0.33$.

A feature of CPKS is that an attacker cannot compromise a direct pairwise key between two non-compromised nodes if he (or she) compromises some nodes in the network. By configuring that the number of sensor nodes sharing the same symmetric-tame map or sharing the same polynomial is no more than t, CPPS, IRSS, and our work can achieve this security property. That is, when $N \times b'/|\mathcal{F}| \leqq t$ in our scheme, considering that there is no node compromised before the prioritization, $N_s \leqq t$ in CPPS and $N \times c'/|\mathcal{F}| \leqq t$ in IRSS, perfect resilience against node compromises can also be achieved.

Figure 8 compares the probability to establish a direct pairwise key between two neighbors under this situation and different e between other schemes and ours, considering that there is no node compromised before the prioritization phase. The parameter $|\mathcal{F}|$ in IRSS and our scheme, and the parameter L in CPPS are chosen to make them perfectly

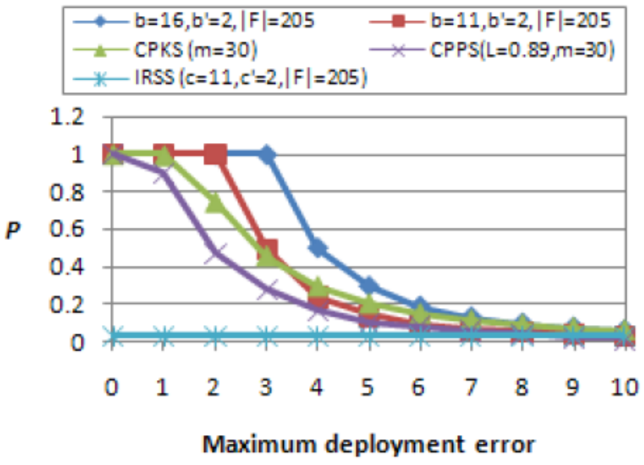


FIGURE 8. Compare the probability to establish a direct pairwise key between two neighbors between our scheme and other schemes. Consider that there is no node compromised before the prioritization in our scheme. The parameter $|\mathcal{F}|$ in our scheme and IRSS, and the parameter L in CPPS are chosen to make them perfectly resistant against node compromises.

resistant against node compromises. We can see that the performance in our scheme is much better than other schemes from the aspect of probability P, especially when e is not big.

Moreover, given the constraints above and fixed e, probability P in CPKS and CPPS is fixed, while probability P in our scheme can still be much higher by increasing b. Although this probability in IRSS can also be improved by increasing c, the performance of our scheme surpasses its performance under the same parameters, as shown in Figure 8.

Now, we consider the other case that there are some nodes compromised before the prioritization. To achieve perfect resilience against node compromises, we configure $N_b \times b/|\mathcal{F}|+(N-N_b)\times b'/|\mathcal{F}| \leqq t$ in our scheme and $N_b\times c/|\mathcal{F}|+(N-N_b)\times c'/|\mathcal{F}| \leqq t$ in IRSS. Figures 9(a) and 9(b) compare probability P with different e between other schemes and ours, considering that there are $N_b = 100$ and $N_b = 500$ compromised nodes before the prioritization, respectively. Similarly, the parameter $|\mathcal{F}|$ in our scheme and IRSS, and the parameter L in CPPS are chosen to make them perfectly resistant. We can see that even though there are some nodes compromised before the prioritization, the results of the comparisons are similar to that in the previous case that no node is compromised before the prioritization.

From the analysis above, we can conclude that our scheme can provide much higher probability to establish a direct pairwise key between two neighbors than other schemes when perfect resilience against node compromises is requested.

6.4. **Overhead.** The computation overhead is mainly caused by the evaluation of a symmetric-tame map share when a sensor node wants to compute a pairwise key. This involves evaluating two t-degree polynomials over $GF(2^l)$, which can be done in an efficient way as discussed in Section 3.

In addition, there are b symmetric-tame map shares and these shares' related locations stored in the memory of each sensor node during the predistribution phase, and b' out of them left after the prioritization phase. If we encode each location coordinate with 8 bytes, 4 bytes for x coordinate and 4 bytes for y coordinate, the memory overhead for a location coordinate will be equivalent to one key space. Hence, for each sensor node,

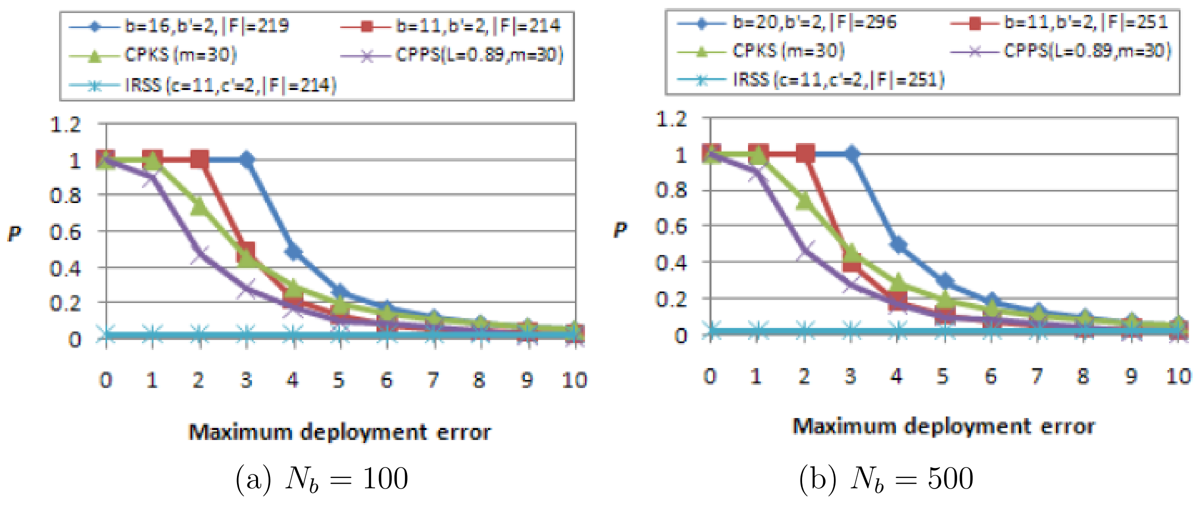


FIGURE 9. Compare the probability to establish a direct pairwise key between two neighbors. Consider that there are some nodes compromised before the prioritization in our scheme. The parameter $|\mathcal{F}|$ in our scheme and IRSS, and the parameter L in CPPS are chosen to make them perfectly resistant against node compromises.

the memory overhead is about $b \times 2l \times (t + 2) + l$ bits in the predistribution phase and $b' \times 2l \times (t+2)+l$ bits after the prioritization phase. Considering node compromises, each non-compromised sensor node records only the IDs of compromised sensor nodes sharing at least one common symmetric-tame map with it. For each of the b' symmetric-tame map shares, if the number of compromised sensor nodes sharing the same symmetric-tame map is more than t, the sensor node will delete the corresponding share and all IDs of the related compromised nodes. Thus, the memory overhead for this is no more than $b' \times t \times \log N$ bits after the prioritization phase.

To establish a common pairwise key between two neighbors, one of them (called source node) first broadcasts its b' IDs of symmetric-tame maps. After receiving this message, the other node determines whether it has share of at least one common symmetric-tame map with the source node and replies a short message to the source node.

7. Discussion.

Figure 10 compares the probability to establish a direct pairwise key between two neighbors under different e between our scheme and other schemes, with memory overhead equivalent to 100 and 200 cryptographic keys, respectively, for each sensor node. We can see that when the memory overhead is reduced ($t = 48$), with other parameters fixed, the performance of P will not be affected in our scheme, CPPS and IRSS, while the performance of P in CPKS decreases. Moreover, as analyzed in Section 6.3, given the limits of e, network density, and the memory overhead for key predistribution, probability P in CPKS is fixed. However, in our scheme, given the constraints above, we can always get arbitrarily high probability P by increasing b.

From the analysis in Section 6.2, the less t will incur the decrease of security guarantee in our scheme, CPPS and IRSS, with other parameters fixed. Nevertheless, by increasing the number of symmetric-tame maps predistributed in the predistribution phase with fixed e, b', t, m and P, we can still achieve arbitrarily good security guarantee in our scheme as shown in Figure 11. Figure 11 presents the probability that a direct pairwise key between two non-compromised nodes is compromised with 50% memory overhead, when there is no node compromised before the prioritization. However, in CPPS, the security guarantee is fixed as thee, m, t and P are fixed. Although IRSS can also get

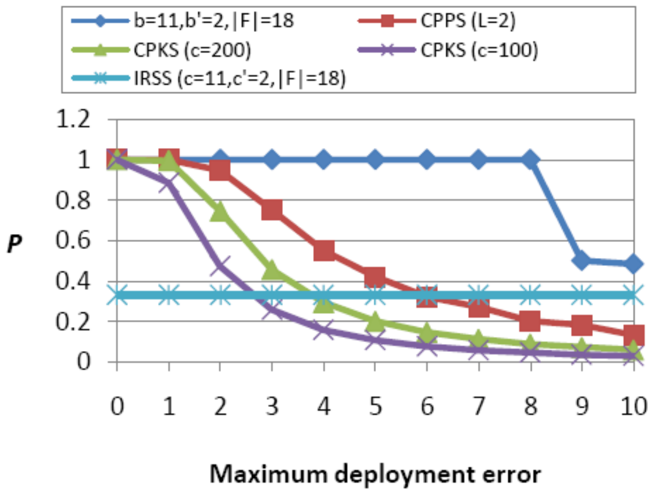


FIGURE 10. Compare the probability to establish a direct pairwise key between two neighbors under different e between our scheme and other schemes, with memory overhead equivalent to 100 and 200 cryptographic keys for each node, respectively. $b' = c' = 2$.

better security guarantee by increasing c with fixed c', t and P, the security guarantee of our scheme is significantly better than that of it as shown in Figure 6. While CPKS can provide perfect security against node compromises, by configuring $N \times b'/|\mathcal{F}| \leqq t$, our scheme can also have this advantage, and still has higher probability P than it as shown in Figure 8.

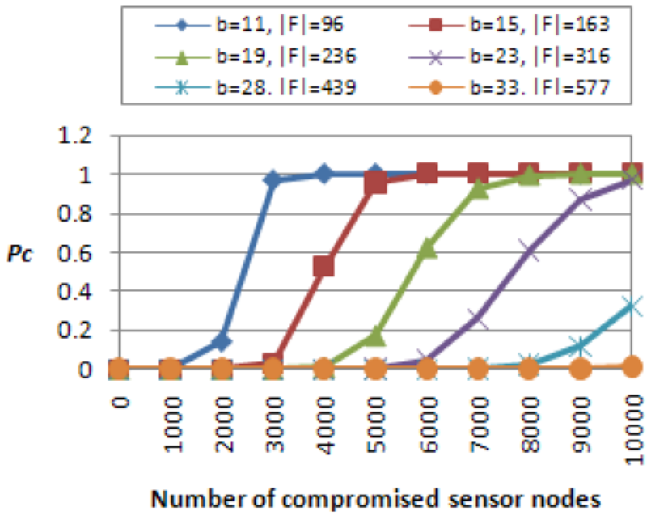


FIGURE 11. Probability that a direct pairwise key between two non-compromised nodes is compromised vs. the number of compromised sensor nodes under different b given that $e = 5$, $b' = 2$ and $P = 0.33$ when there is no node compromised before the prioritization. Assume each node has memory overhead equivalent to 100 cryptographic keys.

For simplicity, we compare the performance between our scheme and other schemes when the memory overhead is reduced, only in the case that there is no node compromised before the prioritization. The result of the comparison in the other case that there are some nodes compromised before the prioritization will be similar to that in the first case. This would be straightforward.

8. **Conclusions.** In this paper, we first take advantage of tame automorphism in algebra to develop a tame-based approach for key predistribution in sensor networks. Based on this approach, we further develop a general framework for pairwise key establishment in sensor networks, a tame pool-based key predistribution. As a result, the tame map can substitute the conventional polynomial in any existing polynomial pool-based scheme in order to provide deterministic authentication service. Particularly, we present a new key predistribution scheme: the local symmetric-tame maps predistribution scheme, based on the framework and the deployment information. The analysis indicates that in our method, exactly combing the expected location and the deployment location for the key predistribution can significantly improve the performance of pairwise key establishment in sensor networks, including the probability to establish a direct pairwise key between two neighbors and the resilience against node compromises.

Acknowledgment. This work is supported in part by the National Science Council under Grants NSC 99-2218-E-002-024 and NSC 99-2218-E-002-026. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

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

[2] L. Ho, M. Moh, Z. Walker, T. Hamada and C. F. Su, A prototype on RFID and sensor networks for elder healthcare: Progress report, *SIGCOMM'05 Workshops*, 2005.

[3] T. H. Kim, S. I. Lee, Y. D. Lee and W. K. Hong, Design and evaluation of in-vehicle sensor network for web based control, *Proc. of the 13th IEEE Int. Symp. and Workshop on Engineering of Computer Based Systems*, 2006.

[4] *The Cricket Indoor Location System an NMS Project @ MIT CSAIL.htm.*

[5] L. Krishnamurthy, R. Adler, P. Buonadonna, J. Chhabra, M. Flanigan, N. Kushalnagar, L. Nachman and M. Yarvis, Design and deployment of industrial sensor networks: Experiences from a semiconductor plant and the north sea, *The 3rd ACM Conf. Embedded Network Sensor Systems*, 2005.

[6] *Crossbow Technology*, http://www.xbow.com/, 2012.

[7] L. Eschenauer and V. Gligor, A key management scheme for distributed sensor networks, *Proc. of the 9th ACM Conf. Computer and Comm. Security*, 2002.

[8] H. Chan, A. Perrig and D. Song, Random key predistribution schemes for sensor networks, *Proc. of IEEE Symp. Security and Privacy*, pp.197-213, 2003.

[9] C.-T. Li, M.-S. Hwang and Y.-P. Chu, An efficient sensor-to-sensor authenticated path-key establishment scheme for secure communications in wireless sensor networks, *International Journal Innovative Computing, Information and Control*, vol.5, no.8, pp.2107-2124, 2009.

[10] W. Du, J. Deng, Y. S. Han and P. K. Varshney, A pairwise key predistribution scheme for wireless sensor networks, *Proc. of the 10th ACM Conf. Computer and Comm. Security*, 2003.

[11] W. Du, J. Deng, Y. S. Han, S. Chen and P. K. Varshney, A key management scheme for wireless sensor networks using deployment knowledge, *Proc. of INFOCOM*, 2004.

[12] D. Liu and P. Ning, Establishing pairwise keys in distributied sensor networks, *Proc. of the 10th ACM Conf. Computer and Comm. Security*, 2003.

[13] I.-C. Lin, P.-Y. Chang and C.-C. Chang, A key management scheme for sensor networks using bilinear pairings and gap Diffie-Hellman group, *International Journal of Innovative Computing, Information and Control*, vol.6, no.2, pp.809-816, 2010.

[14] D. Liu, P. Ning and R. Li, Establishing pairwise keys in distributed sensor networks, *ACM Trans. Information and System Security*, vol.8, no.1, pp.41-77, 2005.

[15] G. Horng, C.-L. Wang and T.-H. Chen, An efficient concealed data aggregation scheme for sensor networks based on secret sharing, *International Journal of Innovative Computing, Information and Control*, vol.5, no.10(A), pp.3085-3098, 2009.

[16] R. Wei and J. Wu, Product construction of key distribution schemes for sensor networks, *Proc. of the 11th Int. Workshop Selected Areas in Cryptography*, 2004.

[17] D. Huang, M. Mehta, D. Medhi and L. Harn, Location-aware key management scheme for wireless sensor networks, *Proc. of the 2nd ACM Workshop Security of Ad Hoc and Sensor Networks*, 2004.

[18] H. Chan and A. Perrig, Pike: Peer intermediaries for key establishment in sensor networks, *Proc. of INFOCOM*, 2005.

[19] H. Nakano, A. Utani, A. Miyauchi and H. Yamamoto, An efficient data gathering scheme using a chaotic pcnn in wireless sensor networks with multiple sink nodes, *ICIC Express Letters*, vol.3, no.3 (B), pp.805-812, 2009.

[20] Y. Zhou, Y. Zhang and Y. Fang, Key establishment in sensor networks based on triangle grid deployment model, *Proc. of IEEE Military Comm. Conf. (MILCOM'05)*, 2005.

[21] P. Traynor, H. Choi, G. Cao, S. Zhu and T. La Porta, Establishing pair-wise keys in heterogeneous sensor networks, *Proc. of INFOCOM*, 2006.

[22] S. Zhu, S. Setia and S. Jajodia, LEAP+: Efficient security mechanisms for large-scale distributed sensor networks, *ACM Trans. Sensor Networks*, vol.2, no.4, pp.500-528, 2006.

[23] P. Traynor, R. Kumar, H. B. Saad, G. Cao and T. La Porta, LIGER: A hybrid key management scheme for heterogeneous sensor networks, *Proc. of ACM/USENIX the 4th Int. Conf. Mobile Systems Applications and Services (MobiSys'06)*, 2006.

[24] P. Traynor, R. Kumar, H. Choi, G. Cao, S. Zhu and T. La Porta, Efficient hybrid security mechanisms for heterogeneous sensor networks, *IEEE Trans. Mobile Computing*, vol.6, no.6, pp.663-677, 2007.

[25] D. Liu and P. Ning, Improving key pre-distribution with deployment knowledge in static sensor networks, *ACM Trans. Sensor Networks*, vol.1, no.2, pp.204-239, 2005.

[26] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro and M. Yung, Perfectly-secure key distribution for dynamic conferences, *Proc. of Conf. Advances in Cryptology (CRYPTO'92)*, vol.740, pp.471-486, 1992.

[27] R. Blom, An optimal class of symmetric key generation systems, *Proc. of EUROCRYPT'84*, 1985.

[28] T. Ito, H. Ohta, N. Matsuda and T. Yoneda, A key pre-distribution scheme for secure sensor networks using probability density function of node deployment, *SASN'05*, 2005.

[29] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen and D. E. Culler, SPINS: Security protocols for sensor networks, *Wireless Networks*, vol.8, pp.521-534, 2002.

[30] A. Perrig, R. Canetti, D. Tygar and D. Song, The Tesla broadcast authentication protocol, *RSA CryptoBytes*, vol.5, no.2, pp.2-13, 2002.

[31] P. J. Chuang, T. H. Chao and B. Y. Li, A scalable grouping random key predistribution scheme for large scale distributed sensor networks, *Proc. of the 3rd Int. Conf. Information Technology and Applications*, 2005.

[32] M. J. Miller and N. H. Vaidya, Leveraging channel diversity for key establishment in wireless sensor networks, *Proc. of INFOCOM*, 2006.

[33] Y. W. Law, J. Doumen and P. Hartel, Survey and benchmark of block ciphers for wireless sensor networks, *ACM Trans. Sensor Networks*, vol.2, no.1, pp.65-93, 2006.

[34] C. Karlof and D. Wagner, Secure routing in wireless sensor networks: Attacks and countermeasures, *Proc. of the 1st IEEE Int. Workshop Sensor Network Protocols and Applications*, 2003.

[35] N. Sastry, U. Shankar and D. Andwagner, Secure verification of location claims, *Proc. of the 2nd ACM Workshop on Wireless Security*, 2003.

[36] L. Hu and D. Evans, Using directional antennas to prevent wormhole attacks, *Proc. of the 11th Network and Distributed System Security Symp.*, 2004.

[37] F. Ye, H. Luo, S. Lu and L. Zhang, Statistical en-route detection and filtering of injected false data in sensor networks, *Proc. of INFOCOM*, 2004.

[38] J. Newsome, R. Shi, D. Song and A. Perrig, The sybil attack in sensor networks: Analysis and defenses, *Proc. of IEEE Int. Conf. Information Processing in Sensor Networks*, 2004.

[39] B. Parno, A. Perrig and V. Gligor, Distributed detection of node replication attacks in sensor networks, *Proc. of IEEE Symp. Security and Privacy*, 2005.

[40] R. Lidl and H. Niederreiter, *Introduction to Finite Fields and Their Applications*, Cambridge University Press, 1986.

[41] W. Du, L. Fang and P. Ning, Lad: Localization anomaly detection for wireless sensor networks, *Proc. of the 19th IEEE Int. Parallel and Distributed Processing Symp.*, 2005.