

A Robust Hypercube-based Pairwise Key Establishment for Sensor Networks

Yen-Hua Liao¹, Chin-Laung Lei¹, and Ai-Nung Wang²

Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan¹

Department of Mathematics, National Taiwan University, Taipei, Taiwan²
radha@fractal.ee.ntu.edu.tw, lei@cc.ee.ntu.edu.tw, wang@math.ntu.edu.tw

Abstract. The hypercube-based key predistribution scheme in sensor networks has some attractive features. However, it has no capability of providing basic authentication service for sensor networks due to using polynomial-based approach. In this article, we introduce a novel tame-based approach, in which a symmetric and two-one bivariate map is generated for key predistribution by exploiting a tame automorphism in algebra. This tame-based approach can offer authentication service for sensor networks. We then improve the hypercube-based scheme based on this tame-based approach. It turns out that this improved scheme is able to fulfill fundamental authentication requirement in sensor networks, and still has the nice features of the hypercube-based scheme.

Keywords: sensor networks, key management, key predistribution, pairwise key, authentication

1 Introduction

A sensor network is typically composed of a large number of resource-limited sensor nodes. Academia and industry have paid a lot of attention to wireless sensor networks due to their widespread applications, such as healthcare, intelligent home, battlefield surveillance [1] etc. In such environment, it is not attractive to implement asymmetric cryptography on resource-constrained sensor nodes. In contrast, symmetric cryptography is more desirable to secure sensor networks from attack. Therefore, how to establish a shared key between two communicating sensor nodes is an important and essential task for carrying out the security objectives in sensor networks. There have been many the related researches developed [2-8].

Eschenauer and Gligor [2] proposed a basic probabilistic key predistribution method. Typically, the topology of a sensor network cannot be known before deployment, so this scheme predistributes some keys in each sensor node prior to deployment. After being deployed, if two sensor nodes share one common key, then they can have a direct pairwise key. Otherwise, they can try to establish an indirect pairwise key through other intermediate nodes. In the basis of this scheme [2], many works were developed. In particular, one of them was the hypercube-based key predistribution scheme [3]. This scheme has some good features. First, a sensor node can know directly whether it can share a direct pairwise key with another node, and if it can, which polynomial should be used. Second, any two sensor nodes can establish a pair-

wise key if the nodes can communicate with each other and there are no compromised sensor nodes. Third, even though there are some nodes compromised, it is still a high probability to have another pairwise key between two noncompromised nodes.

Nevertheless, this scheme cannot provide authentication service for sensor networks due to employing polynomial-based approach. In this article, we first present a tame-based key predistribution approach, which is able to fulfill basic authentication requirement for sensor networks. We then apply this tame-based approach to the hypercube-based key predistribution scheme [3]. As a result, in addition to still having the nice features of the hypercube-based scheme, the improved scheme can offer fundamental authentication service for sensor networks.

The rest of this article is organized as follows. We review briefly the related work for sensor networks in Section 2. In Section 3, we present a tame-based key predistribution approach. The details applying this tame-based approach to the hypercube-based scheme [3] are given in Section 4. In Section 5, we have a conclusion of this article.

2 Related Work

Eschenauer and Gligor [2] introduced a basic probabilistic key predistribution technique for pairwise key establishment in sensor networks. Chan et al. [4] proposed the q -composite key predistribution scheme based on this basic scheme [2]. Because a small number of compromised nodes will compromise a large portion of direct pairwise keys between non-compromised sensor nodes in these schemes, they are not good enough for large-scale sensor networks. Chan et al. [4] additionally presented a random-pairwise keys scheme. This scheme can have perfect security against node compromises. That is, the compromised sensor nodes will not compromise any direct pairwise key between two non-compromised sensor nodes. Nevertheless, this scheme suffers from being only able to support limited network size. Liu et al. [3] proposed the polynomial pool-based schemes: the random subset assignment scheme and hypercube-based key predistribution scheme, based on Blundo's polynomial-based approach [9] and the basic scheme [2]. The polynomial pool-based schemes can have a threshold property on the resilience against node compromises.

However, the polynomial pool-based schemes (The basic scheme [2] is a special case of polynomial pool-based approach.) cannot provide authentication service for sensor networks, which is one of the essential security requirements. In this paper, we introduce a tame-based approach for key predistribution to fulfill the goal of authentication in sensor networks.

3 Tame-based key predistribution

In this section, we present a new technique for pairwise key establishment in sensor networks, utilizing tame automorphism in Algebra.

A tame automorphism $\varphi_i = (\varphi_{i,1}, \dots, \varphi_{i,k})$ is given the following form in Algebra, where the order of variables x_1, \dots, x_k with polynomials $g_{i,j}$ can be permuted:

$$\begin{aligned}
(1): \varphi_{i,1}(x_1, \dots, x_k) &= x_1 + g_{i,1}(x_2, \dots, x_k) = y_1 \\
(2): \varphi_{i,2}(x_1, \dots, x_k) &= x_2 + g_{i,2}(x_3, \dots, x_k) = y_2 \\
&\dots \\
(j): \varphi_{i,j}(x_1, \dots, x_k) &= x_j + g_{i,j}(x_{j+1}, \dots, x_k) = y_j \\
&\dots \\
(n): \varphi_{i,k}(x_1, \dots, x_k) &= x_k = y_k
\end{aligned}$$

This tame automorphism φ_i defined as above is invertible. Its inverse is $\varphi_i^{-1} = (\varphi_{i,1}^{-1}, \dots, \varphi_{i,k}^{-1})$ with $x_k = \varphi_{i,k}^{-1}(y_1, \dots, y_k) = y_k$ and $x_j = \varphi_{i,j}^{-1}(y_1, \dots, y_k) = y_j - g_{i,j}(\varphi_{i,j+1}^{-1}(y_1, \dots, y_k), \dots, \varphi_{i,k}^{-1}(y_1, \dots, y_k))$ for $j = k-1, \dots, 1$. For example, when $k = 4$, $\varphi_i^{-1} = (\varphi_{i,1}^{-1}, \dots, \varphi_{i,4}^{-1})$ is as follows:

$$\begin{aligned}
\varphi_{i,4}^{-1}(y_1, y_2, y_3, y_4) &= x_4 = y_4 \\
\varphi_{i,3}^{-1}(y_1, y_2, y_3, y_4) &= x_3 = y_3 - g_{i,3}(y_4) \\
\varphi_{i,2}^{-1}(y_1, y_2, y_3, y_4) &= x_2 = y_2 - g_{i,2}(y_3 - g_{i,3}(y_4), y_4) \\
\varphi_{i,1}^{-1}(y_1, y_2, y_3, y_4) &= x_1 = y_1 - g_{i,1}(y_2 - g_{i,2}(y_3 - g_{i,3}(y_4), y_4), y_3 - g_{i,3}(y_4), y_4)
\end{aligned}$$

It follows that this tame automorphism φ_i is a one-one map, since it is invertible.

In our method, we let $k = 2$ and $\varphi_i : K^2 \rightarrow K^2$, where K is a $GF(2^l)$; l is half of a cryptographic key length, and let T be composition map $\varphi_r \dots \varphi_2 \varphi_1$ with $r \geq 4$: $T(x_1, x_2) = \varphi_r \dots \varphi_2 \varphi_1(x_1, x_2) = (T_1(x_1, x_2), T_2(x_1, x_2))$. In this composition map T , the order for φ_i 's variables with polynomials $g_{i,j}$ are permuted at least three times.

Example: Let $r = 5$ and $\varphi_i = (\varphi_{i,1}, \varphi_{i,2})$, $1 \leq i \leq r$ be as follows:

$$\begin{aligned}
[1]: \varphi_{1,1}(x_1, x_2) &= x_1 + g_{1,1}(x_2) = y_1 \\
\varphi_{1,2}(x_1, x_2) &= x_2 = y_2 \\
[2]: \varphi_{2,1}(y_1, y_2) &= y_1 = z_1 \\
\varphi_{2,2}(y_1, y_2) &= y_2 + g_{2,2}(y_1) = z_2 \\
[3]: \varphi_{3,1}(z_1, z_2) &= z_1 + g_{3,1}(z_2) = w_1 \\
\varphi_{3,2}(z_1, z_2) &= z_2 = w_2 \\
[4]: \varphi_{4,1}(w_1, w_2) &= w_1 = q_1 \\
\varphi_{4,2}(w_1, w_2) &= w_2 + g_{4,2}(w_1) = q_2 \\
[5]: \varphi_{5,1}(q_1, q_2) &= q_1 = m_1 \\
\varphi_{5,2}(q_1, q_2) &= q_2 + g_{5,2}(q_1) = m_2
\end{aligned}$$

It follows that

$$\begin{aligned}
T(x_1, x_2) &= \varphi_4 \varphi_3 \varphi_2 \varphi_1(x_1, x_2) = (T_1(x_1, x_2), T_2(x_1, x_2)) = \\
& (x_1 + g_{1,1}(x_2) + g_{3,1}(x_2 + g_{2,2}(x_1 + g_{1,1}(x_2))), x_2 + g_{2,2}(x_1 + g_{1,1}(x_2)) + \\
& g_{4,2}(x_1 + g_{1,1}(x_2) + g_{3,1}(x_2 + g_{2,2}(x_1 + g_{1,1}(x_2))))).
\end{aligned}$$

Since φ_i is a one-one map, it follows that composition map T is a one-one map as well.

In addition to T , we let $h(x, y) = (x + y, xy): K^2 \rightarrow K^2$, where K is a $GF(2^l)$. This $h(x, y)$ is symmetric such that $h(x, y) = h(y, x)$ and a two-one map, that is, if $h(\alpha, \beta) = h(\alpha', \beta')$, then $\alpha = \alpha', \beta = \beta'$ or $\alpha = \beta', \beta = \alpha'$. The reasons are:

- (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 we get $\alpha + \beta = u$ and $\alpha\beta = v$. When $x + y = u$ and $xy = v$, we can Replace $y = u - x$ in $xy = v$. Then we get $x^2 - ux + v = 0$. Factor it into $(x - \alpha)(x - \beta) = 0$. It follows that $(x, y) = (\alpha, \beta)$ or (β, α) , just these two solutions. Therefore, $h(x, y)$ is a two-to-one map.

We further let $\psi(x, y) = T \circ h(x, y) = (\psi_1(x, y), \psi_2(x, y)): K^2 \rightarrow K^2$, where K is a $GF(2^l)$. These $\psi_\zeta(x, y)$ for $\zeta = 1, 2$ are bivariate polynomials. It follows that $\psi(x, y)$ is a symmetric and two-one map, since $h(x, y)$ is a symmetric and two-one map, and T is a one-one map. That is, if $\psi(\alpha, \beta) = \psi(\alpha', \beta')$, then $\alpha = \alpha', \beta = \beta'$ or $\alpha = \beta', \beta = \alpha'$.

Through proper configuration, we can get two t -degree bivariate polynomials, $\psi_\zeta(x, y) = \sum_{i,j=0}^t a_{\zeta ij} x^i y^j$ for $\zeta = 1, 2$, from $\psi(x, y)$. For example, in previous example, we get $t = 99$ if $g_{1,1}$ is a 3-degree polynomial, $g_{2,2}$ is a 33-degree polynomial, and $g_{3,1}, g_{4,2}, g_{5,2}$ are 1-degree polynomials. For the sake of presentation, we refer to this $\psi(x, y)$ as a *symmetric-tame map*. In the following, we assume all the symmetric-tame maps are t -degree (means each $\psi_\zeta(x, y)$ for $\zeta = 1, 2$ is t -degree).

For key predistribution, the setup server randomly generates a symmetric-tame map $\psi(x, y)$. Here “randomly” means r is a random number not smaller than 4, each polynomial $g_{i,j}$ is generated randomly, the order for φ_i 's variables with polynomials $g_{i,j}$ in composition map T , for $i = 1 \dots r$, are permuted randomly and at least three times, and t -degree randomly consists of t 's factors depending on r value and how $T_\zeta(x, y)$ for $\zeta = 1, 2$ is composed of. For example, in aforementioned example with $t = 99$ and $r = 5$, the degrees of $g_{1,1}, g_{2,2}, g_{3,1}, g_{4,2},$ and $g_{5,2}$ can be, respectively, 3, 33, 1, 1, 1, or 33, 1, 3, 1, 1, or 1, 11, 1, 1, 9 etc.

In our method, each node has unique ID. For each node α , the setup server computes a symmetric-tame map share of $\psi(x, y): \psi(\alpha, y) = (\psi_1(\alpha, y), \psi_2(\alpha, y))$, and store $\psi(\alpha, y)$ in node α 's storage. Then node α can establish a pairwise key with node β through $\psi(\alpha, \beta) = (\psi_1(\alpha, \beta), \psi_2(\alpha, \beta)) = (\psi_1(\beta, \alpha), \psi_2(\beta, \alpha)) = \psi(\beta, \alpha)$. Node α can compute $\psi(\alpha, \beta) = (\psi_1(\alpha, \beta), \psi_2(\alpha, \beta))$ by evaluating $\psi(\alpha, y)$ at point β , and Node β can also compute the common key $\psi(\alpha, \beta) = \psi(\beta, \alpha) = (\psi_1(\beta, \alpha), \psi_2(\beta, \alpha))$ by evaluating $\psi(\beta, y)$ at point α . Both of them then use $\psi_1(\alpha, \beta) \parallel \psi_2(\alpha, \beta) = \psi_1(\beta, \alpha) \parallel \psi_2(\beta, \alpha)$ as their pairwise key. Since $\psi(x, y)$ is a symmetric and two-one map, this pairwise key is uniquely shared between them. They both can authenticate each other through challenge-response protocol with this key. Therefore, this tame-based approach can provide the authentication service.

Due to two t -degree bivariate polynomials $\psi_\zeta(x, y)$ for $\zeta = 1, 2$ involved, the security analysis for this approach is t -collusion resistant. In other words, an attacker

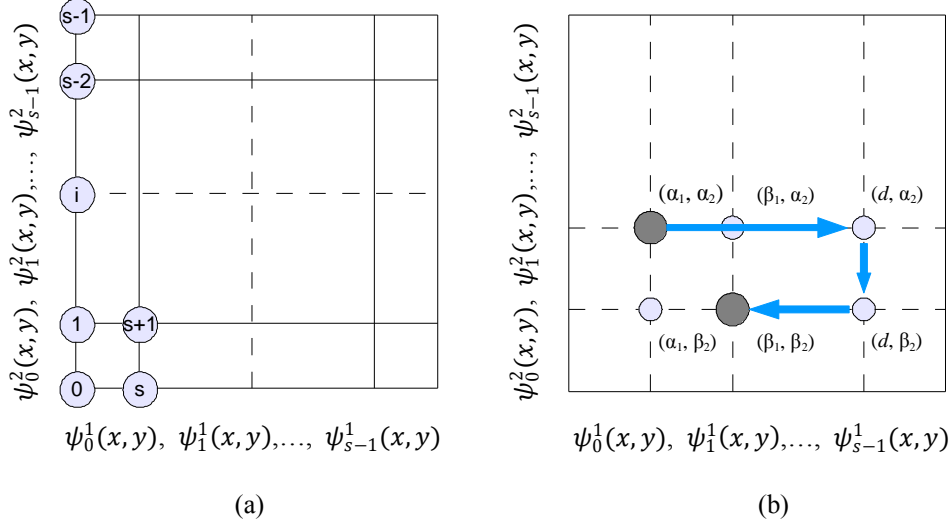


Fig. 1. The improved hypercube-based key predistribution for $n = 2$. (a) An example of order for node's assignment. (b) Hypercube for $n = 2$.

knows nothing about the pairwise key between any two non-compromised nodes if he (or she) compromises no more than t nodes.

Each sensor node needs to use storage space of $2 \times (t+1) \times 1$ bits for storing the t -degree symmetric-tame map share, which is equal to $(t+1)$ cryptographic keys. Two sensor nodes need to evaluate the symmetric-tame map share at the ID of the other sensor node for establishing 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 [3]. There is no communication overhead in this approach for pairwise key establishment.

4 Improving hypercube-based key predistribution

Now, we improve the hypercube-based key predistribution scheme [3], based on the tame-based approach presented in the previous section. The details of the improved scheme are as follows:

(1) Setup Phase:

Suppose the total number of sensor nodes in the network is N . The setup server first constructs an n -dimensional hypercube. For each sensor node, the setup server assigns the smallest unoccupied coordinate (i_1, \dots, i_n) to this node. This is in order to make any two nodes have at least one key path when there is no node compromised and any two nodes can communicate with each other. This coordinate can be used as node's ID. Figure 1(a) presents an example of order for node assignment when $n = 2$. We can see that if the nodes (a_1, a_2) and (b_1, b_2) exist, it must either node (a_1, b_2) or node (b_1, a_2) exist, or both. For dimension i of the hypercube, the setup server randomly generates s^{n-1} symmetric-tame maps

$\psi_{(\tau_1, \dots, \tau_{n-1})}^i(x, y)$, where $s = \lceil \sqrt[n]{N} \rceil$, $1 \leq i \leq n$, $0 \leq \tau_1, \dots, \tau_{n-1} < s$. For each node (i_1, \dots, i_n) , the setup server distributes $\{\text{ID}, \psi_{(i_2, \dots, i_n)}^1(i_1, y), \dots, \psi_{(i_1, \dots, i_{n-1})}^n(i_n, y)\}$ to this node. Figure 1(b) shows the hypercube for $n = 2$.

(2) Direct Key Establishment Phase:

When node α wants to establish a pairwise key with node β , it first determines whether the Hamming distance of their IDs is equal to one (In other words, their coordinates have same values in $n-1$ dimensions). If it is, then they share a common symmetric-tame map according to our symmetric-tame map assignment method. Therefore, they can establish a direct pairwise key based on the tame-based key predistribution approach. For instance, node α and node β have the same values in the first $n-1$ dimensions, $\alpha_j = \beta_j$, for $1 \leq j \leq n-1$, then they share the symmetric-tame map $\psi_{(\alpha_1, \dots, \alpha_{n-1})}^n(x, y)$; hence can establish a direct pairwise key through this map. If they cannot establish direct key, they need to run the indirect key establishment.

(3) Indirect Key Establishment Phase:

If the Hamming distance of nodes α and β is bigger than one, they need to find a key path for establishing an indirect pairwise key. According to our node assignment scheme, they can find at least one key path between them to establish an indirect key if there is no node compromised and any two nodes can communicate with each other. They can predetermine such a key path by the following key path discovery algorithm without communicating with each other. For instance, in Figure 1(b), node (α_1, β_2) or node (β_1, α_2) can help node (α_1, α_2) and node (β_1, β_2) to establish an indirect pairwise key. One possible way is to let node (α_1, α_2) generate the pairwise key $K_{\alpha\beta}$ and send it to node (β_1, β_2) through the intermediate node (say, node (α_1, β_2)). Every message transmitted between two adjacent nodes in the path is encrypted and authenticated with the direct pairwise key shared between them. Assume $\alpha > \beta$. To find such a key path, node α or node β performs the following algorithm.

- (i) The source node (say, node α) keeps a set $\mathcal{E} = \{e_1, \dots, e_c\}$, a most recently discovered intermediate node γ , and a list \mathcal{L} , where the set \mathcal{E} records the dimensions that ID α and ID β have different values and the list \mathcal{L} records the key path discovered by this algorithm. In the beginning, $\gamma = \alpha$ and $\mathcal{L} = \{\alpha\}$.
- (ii) The next intermediate node w is decided by randomly choosing an e from \mathcal{E} so that $w = (\gamma_1, \dots, \gamma_{e-1}, \beta_e, \gamma_{e+1}, \dots, \gamma_n)$ exists. Then remove e from \mathcal{E} , let $\gamma = w$, and append w to \mathcal{L} . This step is repeated until $|\mathcal{E}| = 1$. When $|\mathcal{E}|$ is one, append β to \mathcal{L} ; then return \mathcal{L} as the key path discovered.

For instance, if $\alpha = (1, 2, 4)$ and $\beta = (0, 3, 5)$, then $\{(1, 2, 4), (0, 2, 4), (0, 2, 5), (0, 3, 5)\}$ is a possible key path between node α and node β . If the intermediate nodes discovered by this algorithm are compromised or out of communication range for some reasons, there is still alternative algorithm, called dynamic key path discovery, to find key paths. For instance, in Figure 1(b), if node (d, α_2) can have a predetermined key path with node (β_1, β_2) through node (d, β_2) , then node (α_1, α_2) can use

node (d, α_2) to establish an pairwise key with node (β_1, β_2) . We refer interested readers to [3] for details of this dynamic key path discovery algorithm.

Since the procedure of the improved scheme is similar to the hypercube-based scheme [3], the analysis of performance, security and overhead of this scheme is the same with the hypercube-based scheme. However, due to using tame-based approach, this scheme is able to provide authentication service for sensor networks, in addition to preserving the nice features of the hypercube-based scheme.

5 Conclusion

In this paper, we introduce a new tame-based pairwise key establishment to fulfill authentication function in sensor networks. We then improve the hypercube-based key predistribution scheme [3] based on this tame-based approach. As a result, this improved scheme can satisfy the basic authentication requirement in sensor networks, besides still having the nice features of the hypercube-based key predistribution scheme.

Acknowledgements. This work is supported in part by the National Science Council under Grants NSC 96-2628-E-002-023-MY3, NSC 98-2219-E-011-001 and by the iCAST project under the Grants No. NSC97-2745-P-001-0012.

References

- [1] Y. Zhou, Y. Fang, and Y. Zhang, "Securing Wireless Sensor Networks: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 10, no. 3, pp. 6-28.
- [2] L. Eschenauer and V. Gligor, "A Key Management Scheme for Distributed Sensor Networks," *Proc. Ninth ACM Conf. Computer and Comm. Security (CCS '02)*, Nov. 2002.
- [3] D. Liu, P. Ning and R. Li, "Establishing Pairwise Keys in Distributed Sensor Networks," *ACM Trans. Inf. Sys. Sec.*, vol. 8, no. 1, February 2005, pp. 41-77.
- [4] H. Chan, A. Perrig, and D. Song, "Random Key Predistribution Schemes for Sensor Networks," *Proc. IEEE Symp. Security and Privacy*, May 2003, pp. 197-213.
- [5] W. Du, J. Deng, Y.S. Han, and P.K. Varshney, "A Pairwise Key Predistribution Scheme for Wireless Sensor Networks," *Proc. 10th ACM Conf. Computer and Comm. Security (CCS '03)*, Oct. 2003.
- [6] D. Liu, and P. Ning, "Improving key pre-distribution with deployment knowledge in static sensor networks," *ACM Trans. Sensor Networks*, vol. 1, no. 2, Nov. 2005, pp. 204-239.
- [7] P. Traynor, R. Kumar, H. Choi, G. Cao, S. Zhu, and T. L. Porta, "Efficient Hybrid Security Mechanisms for Heterogeneous Sensor Networks," *IEEE Trans. Mobile Computing*, vol. 6, no. 6, Jun. 2007, pp. 663-677
- [8] 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, Nov. 2006, pp. 500-528.
- [9] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, "Perfectly-Secure Key Distribution for Dynamic Conferences," *Proc. Conf. Advances in Cryptology (CRYPTO '92)*, vol. 740, 1992, pp. 471-486.