Multi-Layer Encryption for Multi-Level Access Control in Wireless Sensor Networks

Po-Yuan Teng, Shih-I Huang, and Adrian Perrig

Abstract The purpose of Multi-Layer Encryption (MLE) is to have only one cipher text, but users with different keys (e.g., in different groups) will obtain different levels of data after they decrypt with their own key. This property is especially useful in surveillance applications, which requires an efficient mechanism for multi-level data access. In this paper, we first address specific requirements for Wireless Sensor Networks (WSNs), and then propose a MLE scheme which has good properties of forward/backward secrecy, without the necessity of time synchronization. In this scheme, users only need to store a constant number of keys regardless of defined secret layers, and higher-level users are able to decrypt more data than lower-level users.

Key words: Security, Multi-Layer Encryption, Forward Secrecy, Wireless Sensor Networks, Multi-Level Access Control

1 Introduction

Some applications with multiple priority groups need different layers of sensed data (e.g., in a metropolitan surveillance application, the police can see all data, but citizen can only see a subset of the data), this requirement is the main reason to develop MLE. In our architecture, we expect a data server to store encrypted data from sensor nodes, and this data server will authenticate users whenever they request reading specific data.

Po-Yuan Teng
Industrial Technology Research Institute (ITRI), Taiwan. e-mail: pppk@itri.org.tw

Shih-I Huang
Industrial Technology Research Institute (ITRI), Taiwan. e-mail: si.huang@gmail.com

Adrian Perrig
Carnegie Mellon University (CMU). e-mail: adrian@ece.cmu.edu

In the following section, we shortly describe our target environment, and supply a summary of notation used throughout this paper. We will precisely describe our multi-layer encryption schemes in Section 3 and discuss some possible attacks in Section 4, then make a conclusion in Section 5.

2 Background

In this section, we briefly describe the fundamental network architecture applied in our scheme, and the notations we used throughout this paper.

- Sensor Network Environments

In general, sensor nodes face some limitations, including constrained computational power, limited battery life, limited storage space, and deployments in networks [5].

Because of these sensor node limitations, one popular solution is to have a data server as a supplementary controller (also known as base station [4]). In this architecture, the data server stores sensed data from sensor nodes, broadcasts beacon signals periodically to maintain the routing topology, and schedules the duty cycle for each node. This periodically broadcasted beacon signals are utilized to develop our MLE scheme.

- Notation

For clarity, we list the symbols and notations used throughout this paper below:

Table 1 Notation

MK	Master Key	$H^n(M)$	Hash n times of message M
M_i	Plaintext of layered message i	$H(M_1, M_2)$	Hash of M_1 concatenates M_2
C_i	Cipher text of corresponding M_i	KB_{ID_i,L_n}	Base key for layer n of node ID_i
ID_i	The identity of sensor node i	KE_{ID_i,L_n,T_j}	Encryption key for layer n of node ID_i, during time period T_j
UID_i	The identity of user i	T_i	The ith Time period
UK_i	User key for user i	TMK_{ID_i}	Time Master Key for node ID_i
K_{G_i}	Group Key (ex. K_{G_1} for Group G_1)	TK_{ID_i,T_j}	Time Key for node ID_i, during time period T_j
$Enc(K,M)$	Encrypt message M using key K		
$\{M\}_K$	Encrypted message M by K		

3 Multi-Layer Encryption Scheme

Informally, forward secrecy ensures that the past messages are protected even if the current secret key is exposed [3], and backward secrecy means that the exposed secret key is no longer useful in the future.

As shown in Fig. 1, this scheme requires the data server to hold one master key MK. The data server randomly generates K_{G_1} and computes K_{G_2}, K_{G_3} by performing one-way hash function, and then gives these keys to G_1, G_2, and G_3 users respectively.

The data server periodically broadcasts $seedT_i$ to sensor nodes, the value of $seedT_i$ could be a computational result of time period T_i. For example, as Fig. 1 shows, the date "2007-10-15" could be the 18th time period of our system, so the value of $seedT_i$ that the server broadcasts in this time period is $seedT_{18}$, where $seedT_{18} = AES(MK, \text{"2007-10-15"})$.

Basically, $seedT_i$ are used to update encryption keys, since these $seedT_i$ values are broadcasted in plaintext, an attacker could record all the values and endanger the system. To avoid this problem, the server gives each sensor node an unique time master key (TMK_{ID_i}) through function $TMK_{ID_i} = Enc(MK, ID_i)$. TMK_{ID_i} are used to generate time keys (TK_{ID_i,T_j}), and time keys are aimed to update encryption keys for each secret layer through hash function $KE_{ID_i,L_n,T_j} = H(KB_{ID_i,L_n}, H^{L_{n-1}}(TK_{ID_i,T_j}))$. The reason to perform one-way hash on time keys in each secret layer here is to prevent colluding attack.

The server gives each sensor node a different key set, here we denote it as base key (KB_{ID_i,L_n}), where the subscript ID_i denotes that this key belongs to node ID_i and L_n denotes the secret level of the key. Combining base keys and TK_{ID_i,T_j} values can generate encryption keys (KE_{ID_i,L_n,T_j}). This is known as key-insulated methods which are mainly used to provide forward/backward secrecy.

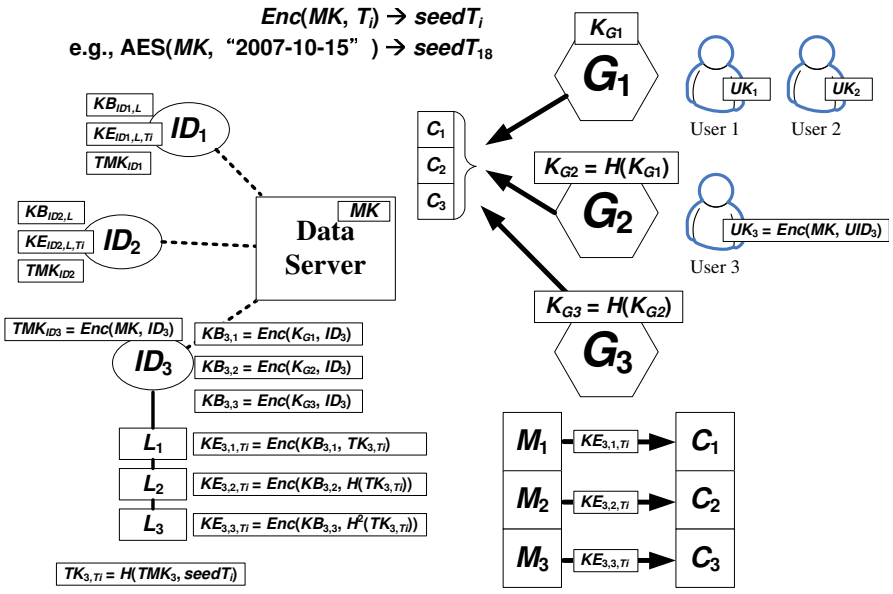

Fig. 1 Our MLE scheme

In this scheme we also give each user an user key (UK_i) by the generating function $UK_i = Enc(MK, UID_i)$. These user keys can be used to perform user authentication and encrypt time key (TK_{ID_i,T_i}) before sending to users.

We use the example that user 1 requests data "2007-10-15" from node ID_3, the flow is as follows:

1. User 1 requests data from ID_3 in time period "2007-10-15"
2. Server authenticates user 1 by UK_1
3. Server compute $Enc(MK,$ "2007-10-15") = $seedT_{18}$
4. Server computes $Enc(MK, ID_3) = TMK_3$
5. Server computes $H(TMK_3, seedT_{18}) = TK_{3,18}$
6. Server sends $Enc(UK_1, TK_{3,18}) = \{TK_{3,18}\}_{UK_1}$ for user 1
7. User 1 decrypts $\{TK_{3,18}\}_{UK_1}$ and obtains $TK_{3,18}$
8. User 1 owns K_{G_1} and now she has $TK_{3,18}$
   a. User 1 has K_{G_1} and knows ID_3 (public information), so she can derive node's base key $KB_{3,1}$ by computing $Enc(K_{G_1}, ID_3) = KB_{3,1}$
   b. User 1 knows $TK_{3,18}$, so she can derive node's encryption key $KE_{3,1,18}$ by computing $Enc(KB_{3,1}, TK_{3,18}) = KE_{3,1,18}$
   c. Then user 1 has encryption key $KE_{3,1,18}$ and can decrypt C_1 to obtain M_1
   d. Because user 1 has K_{G_1}, she can deduce K_{G_2} and K_{G_3} by performing one-way hash function, then she can derive base key $KB_{3,2}$ and $KB_{3,3}$ by computing $Enc(K_{G_2}, ID_3) = KB_{3,2}$ and $Enc(K_{G_3}, ID_3) = KB_{3,3}$
   e. User 1 can derive encryption key $KE_{3,2,18}$ and $KE_{3,3,18}$ to decrypt C_2 and C_3, where $KE_{3,2,18} = Enc(KB_{3,2}, H(TK_{3,18}))$ and $KE_{3,3,18} = Enc(KB_{3,3}, H^2(TK_{3,18}))$
9. User 3 in group G_2 only has K_{G_2} and can deduce K_{G_3}. If user 3 requests time key from the server, after authenticated herself using UK_3, the server will know she is in group G_2, and gives her the value of $H(TK_{3,18})$ instead of $TK_{3,18}$. This can prevent the colluding attack because even though user 3 can get K_{G_1} from a left G_1 user, she still cannot obtain $TK_{3,18}$ value, so user 3 can at most obtain M_2 and M_3

4 Discussion

There are many known attacks in sensor networks, including Denial-of-Service, blackhole, wormhole, Sybil, traffic analysis, node replication, and so on [1, 2]. As a complementary solution, we concentrate on the security of our MLE scheme. There are some possible attacks to our proposed scheme and we evaluate the security here.

- **Eavesdropping** In our proposed scheme, the only plaintext data adversaries can get is $seedT_i$ value, but without the time master key TMK_{ID_i}, the $seedT_i$ value is useless because it is only a seed value for generating the time key TK_{ID_i,T_j}.
- **Colluding Attack** If a user in lower privileged level collude with a left user in higher privileged level (e.g., user 3 colludes with user 1), although she can get the

group key of higher level, after authenticated, the data server will only give the time key of corresponding privileged level (i.e., $H^{L_{n-1}}(TK_{ID_i,T_j})$) to her. Without time keys of higher level, the user cannot derive specific encryption keys to obtain the data.

- **Compromised sensor nodes** In our scheme, each sensor node is given a set of distinct keys, and these keys are only the computational results. Even if specific sensor node is compromised, the adversary will only know these computational results and cannot take any advantage to compromise the other nodes, so the damage will be limited in the range of compromised nodes.

5 Conclusion

With the proliferation of sensor networks, the amount of privacy-sensitive data that is collected increases continuously. Based on the inherent properties of numerous applications, we observe a tension and tradeoff between privacy and the usefulness of information: very fine-grained data often contains privacy-sensitive information but is the most useful, whereas coarse-grained data protects privacy but is often less useful.

In this paper, we observe that we can break this tradeoff by simultaneously providing access to varying granularities of information, based on the access right of the data consumer. In fact, our efficient cryptographic construction enables sensor nodes to encrypt different granularities of data under different cryptographic keys. We find that our approach is viable even on highly resource-constrained sensor nodes, enabling us to simultaneously achieve several points in the tradeoff space between fine granularity/resolution of data and privacy.

Acknowledgements Special thanks to Lee-Chun Ko for inspecting the article and providing precious comments.

References

1. Akyildiz LF, Su W, Sankarasubramaniam Y and Cayirci E (2002) A survey on sensor networks. In IEEE Communications Magazine, 40(8):102–114
2. Callaway EH (2004) Wireless Sensor Networks: Architectures and Protocols. CRC Press.
3. Itkis G (2006) Forward Security: Adaptive Cryptography - Time Evolution. Invited chapter for the Handbook of Information Security, John Wiley and Sons, Inc.
4. Perrig A, Szewczyk R, Wen V et al (2002) SPINS: Security Protocols for Sensor Networks. In Wireless Networks Journal (WINE), September 2002.
5. Walters JP, Liang Z, Shi W, and Chaudhary V (2006) Wireless sensor network security: A survey. In Security in Distributed, Grid, and Pervasive Computing, Chapter 17, Auerbach Publications, CRC Press.