LETTER

Security Analysis of a Distributed Reprogramming Protocol for Wireless Sensor Networks*

Yong YU[†,††a)], **Jianbing NI**[†,††], *Nonmembers*, *and* **Ying SUN**[††], *Student Member*

SUMMARY Reprogramming for wireless sensor networks is essential to upload new code or to alter the functionality of existing code. To overcome the weakness of the centralized approach of the traditional solutions, He *et al.* proposed the notion of distributed reprogramming where multiple authorized network users are able to reprogram sensor nodes without involving the base station. They also gave a novel distributed reprogramming protocol called SDRP by using identity-based signature, and provided a comprehensive security analysis for their protocol. In this letter, unfortunately, we demonstrate that SDRP is insecure as the protocol fails to satisfy the property of authenticity and integrity of code images, the most important security requirement of a secure reprogramming protocol.

key words: *security analysis, wireless sensor networks, reprogramming, authentication*

1. Introduction

Wireless reprogramming is referred to as the activities that upload a new code or retask the existing code with new parameters [1], [2] in wireless sensor networks. The early reprogramming protocols dealt with spreading new code images but did not take security into account. However, security of a reprogramming protocol is core among all the desirable properties since wireless networks are usually deployed in hostile environments. Observing that all the previous secure reprogramming protocols are centralized in the sense that only the base station has the power to perform the reprogramming, He *et al.* [3] proposed the concept of distributed reprogramming to support multiple authorized users to reprogram sensor nodes. Distributed reprogramming achieves the following merits. Firstly, it overcomes the weakness of single point of failure in the centralized solutions and secondly, different users may have different privilege of reprogramming sensor nodes. Accordingly, distributed reprogramming is more suitable than the traditional approaches for wireless sensor networks. Nine requirements that a secure distributed reprogramming protocol should provide, namely, authenticity and integrity of code images, freshness, node compromise, distributed, supporting different user privileges, partial reprogram capability, user traceability, efficiency and scalability are also specified in [3]. Authenticity and integrity of code images are most highly desirable among all these properties. After a detailed analysis, He *et al.* concluded that PKI based approach, group signature based approach and the solutions of pre-equipping each node with multiple key pairs are not suitable for constructing distributed reprogramming protocols. Then they proposed a novel solution using identity-based cryptography and described a concrete scheme based on their identity-based signature from bilinear maps. They also provided a comprehensive security analysis to show that their distributed reprogramming protocol enjoys many desirable properties: authenticity and integrity of code images, freshness, resistance to node and user compromised attacks, distributed, supporting different user privileges and user traceability.

In this letter, we revisit the distributed reprogramming protocol in [3] and show that there is a security flaw in SDRP: after observing a valid signature on a message m generated by the network user U_j, a passive adversary is able to recover U_j's private key.

2. Review of SDRP

Identity-based cryptography was suggested as a basic tool to construct distributed reprogramming protocols in the following way [3]: the private key generator in identity-based cryptography acts as the network owner in distributed reprogramming, and the users in identity-based cryptography play the role of the network users in distributed reprogramming, and the verifiers in identity-based signature become the sensor nodes. Elliptic curve cryptography was employed in their scheme because of its advantages in shorter signature size, higher computational efficiency and lower communication cost. We briefly review SDRP which involves three phases: system initialization, user preprocessing and sensor node verification.

2.1 System Initialization

The network owner selects groups G and G_T with the same prime order q equipped with a bilinear map $e : G \times G \to G_T$, where G is a cyclic additive group generated by P while G_T

Manuscript received January 7, 2013.

[†]The authors are with School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, 610054, China.

[††]The authors are with School of Computer Science and Software Engineering, University of Wollongong, Wollongong, NSW2522, Australia.

*This work was supported by the National Natural Science Foundation of China under Grants 61003232, 61073176, the National Research Foundation for the Doctoral Program of Higher Education of China under Grant 20100185120012, and the Fundamental Research Funds for the Central Universities under Grant ZYGX2010J066. The first author is supported by University of Wollongong Vice Chancellor Fellowship.

a) E-mail: yyucd2012@gmail.com

DOI: 10.1587/transinf.E96.D.1875

is a cyclic multiplicative group. He also chooses two secure hash functions $H_1 : \{0,1\}^* \rightarrow G$, $H_2 : \{0,1\}^* \rightarrow Z_q^*$, picks a random value $s \in Z_q^*$ as his master key and computes $PK_{owner} = sP$ as his public key. The system public parameters $\{G, G_T, P, q, e, PK_{owner}, H_1, H_2\}$ are loaded in all sensor nodes before deployment.

When a network user U_j with identity $UID_j \in \{0,1\}^*$ registers to the network owner, the network owner first checks the validity of the user. If the registration information is false, he rejects the application. Otherwise, he sets U_j's public key as $PK_j = H_1(UID_j, Pri_j)$ where Pri_j denotes the level of U_j's privilege, generates U_j's private key $SK_j = s \cdot PK_j$ and forwards it to U_j in a secure manner.

2.2 User Preprocessing

We briefly describe how a network user U_j reprograms the sensor nodes with a new program image. Firstly, U_j partitions the program image to fixed-size pages and splits each page into fixed-size packets. Merkle hash tree [4] and hash-chain techniques are employed to facilitate the authentication of the packets. To guarantee the integrity and authenticity of the whole new code image, U_j computes $\sigma_j = H_2(m) \cdot SK_j$ as his signature on message m, which includes the metadata of the code image and the root of the Merkle hash tree. Finally, U_j dispatches $\{UID_j, Pri_j, m, \sigma_j\}$ to the targeted sensor nodes as the notification of the new code image.

2.3 Sensor Node Verification

Upon receiving a message-signature pair $\{UID_j, Pri_j, m, \sigma_j\}$ from the user U_j, the intended nodes first check the programming privilege of U_j. If it is valid, the nodes verify the signature by checking if the following equation holds:

$$e(P, \sigma_j) = e(H_2(m) \cdot H_1(UID_j, Pri_j), PK_{owner}).$$

3. A Security Flaw in SDRP

SDRP enjoys many advantages as the authors have discussed in [3]. They also explained two main merits of the signature they used. Firstly, the signature overhead is low and secondly, the signing speed is fast. For the security, they argued that $\sigma_j = H_2(m) \cdot SK_j$ is a secure identity-based signature since without the private key SK_j, it is hard to forge a valid signature. Moreover, it is infeasible to derive SK_j merely from $(UID_j, OK_j, P, H_1, H_2, PK_{owner})$. However, below we show that a passive adversary in the network can successfully derive a user's secret key after observing a valid message-signature pair from the user.

In the distributed reprogramming protocol [3], the signature is generated by

$$\sigma_j = H_2(m) \cdot SK_j.$$

If an adversary gets a valid message-signature pair (m, σ_j), he is able to recover the U_j's private key by computing

$$H_2(m)^{-1} \cdot \sigma_j \rightarrow SK_j,$$

where $H_2(m)^{-1} \pmod q$ can be derived by using the Extended Euclidian algorithm [5]. With this private key, an adversary can impersonate the network user U_j for generating signatures on any program image, which leads to the protocol losing authenticity and integrity of code images. However, we stress that the security problem we showed above only lies in the concrete scheme in [3]. The generic solution proposed by He *et al.* remains valid and sound.

Therefore, to fix the problem, we can make use of some ID-based signatures with provable security such as the schemes due to Hess [6], Cha and Cheon [7]. This modification will overcome the security flaws in the concrete protocol [3] without sacrificing any desirable security feature. However, expensive bilinear pairing operations are involved in most of the existing identity-based signature schemes such as [6], [7]. A bilinear pairing operation requires almost 10 times more computations in the underlying finite field than an elliptic curve point scalar multiplication does in the same finite field [8], [9]. For 80-bit security, one pairing computation takes about 1.90 s in an optimized implementation on a standard MICA2 sensor node [10]. The readers can refer to [11] for the latest results of implementation of pairing-based cryptography on a sensor node. Thus, we also suggest applying some more efficient non-pairing-based identity-based signature schemes such as [12] to distributed reprogramming protocols for wireless sensor networks.

4. Conclusion

In this letter, we revisited the distributed reprogramming protocol for wireless sensor networks introduced by He et al. and showed that there is a security flaw in their protocol and suggested a way to resolve the problem without sacrificing any desirable security feature.

References

[1] V.C. Gungor and G.P. Hancke, "Industrial wireless sensor networks: Challenges, design principles, and technical approaches," IEEE Trans. Ind. Electron., vol.56, no.10, pp.4258–4265, 2009.
[2] Y. Li and J. Ma, "Secure message distribution scheme with configurable privacy in heterogeneous wireless sensor networks," IEICE Trans. Inf. & Syst., vol.E93-D, no.3, pp.484–490, March 2010.
[3] D. He, C. Chen, S. Chan, and J. Hu, "SDRP: A secure and distributed reprogramming protocol for wireless sensor networks," IEEE Trans. Ind. Electron., vol.59, no.11, pp.4155–4163, 2012.
[4] R. Merkle, "Protocols for public key cryptosystems," Proc. IEEE, Security and Privacy, pp.122–133, 1980.
[5] V. Shoup, A computational introduction to number theory and algebra, Cambridge University Press, 2008.
[6] F. Hess, "Efficient identity based signature schemes based on pairings," Proc. SAC 2002, LNCS 2595, pp.310–324, 2002.
[7] J.C. Cha and J.H. Cheon, "An identity-based signature from gap Diffie-Hellman groups," Proc. PKC 2003, LNCS 2567, pp.18–30, 2003.
[8] P. Barreto, H. Kim, B. Lynn, and M. Scott, "Efficient algorithms for

pairing-based cryptosystems," Proc. CRYPTO 2002, LNCS 2442, pp.354–368, 2002.
[9] P. Barreto, B. Lynn, and M. Scott, "On the selection of pairing-friendly groups," Proc. SAC 2003, LNCS 3006, pp.17–25, 2004.
[10] L.B. Oliveira, D.F. Aranha, and C.P.L. Gouvea, "TinyPBC: Pairings for authenticated identity-based non-interactive key distribution in sensor networks," Comput. Commun., vol.34, no.3, pp.485–493, 2011.
[11] M. Shirase, Y. Miyazaki, and T. Takagi, "Efficient implementation of pairing-based cryptography on a sensor node," IEICE Trans. Inf. & Syst., vol.E92-D, no.5, pp.909–917, May 2009.
[12] R. Zhu, G. Yang, and D.S. Wong, "An efficient identity-based key exchange protocol with KGS forward secrecy for low-power devices," Theor. Comput. Sci., vol.378, no.2, pp.198–207, 2007.