Two Factor Authenticated Key Exchange Protocol for Wireless Sensor Networks: Formal Model and Secure Construction

Fushan Wei[1(✉)], Ruijie Zhang[1], and Chuangui Ma[2]

[1] State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou, China
{weifs831020,rjz_wonder}@163.com
[2] Department of Basic Courses, Army Aviation Institute, Beijing, China
chuanguima@sina.com

Abstract. Two-factor authenticated key exchange (TFAKE) protocols are critical tools for ensuring identity authentication and secure data transmission in wireless sensor networks (WSNs). Until now, numerous TFAKE protocols based on smart cards and passwords are proposed for WSNs. Unfortunately, most of them are found insecure against various attacks. Researchers focus on cryptanalysis of these protocols and then fixing the loopholes. Little attention has been paid to design rationales and formal security models of these protocols. In this paper, we first put forward a formal security model for TFAKE protocols in WSNs. We then present an efficient TFAKE protocol for WSNs without using expensive asymmetric cryptology mechanisms. Our protocol can be proven secure in the random oracle model and achieves user anonymity. Compared with other TFAKE protocols, our protocol is more efficient and enjoys provable security.

Keywords: Two-factor authenticated key exchange · Password · Smart card · Provable security · Wireless sensor networks

1 Introduction

With the rapid development of the micro electronic mechanism system, wireless communications and low-power technologies in embedded systems, wireless sensor networks (WSNs) are now widely used in many applications, such as military surveillance, environment monitoring, health care monitoring, disaster relief and natural disaster prevention. WSNs are usually composed of thousands even millions of sensor nodes. Due to its ubiquitous nature, sensor nodes are randomly

F.Wei—This work is supported by the National Natural Science Foundation of China (Nos. 61309016, 61379150, 61501515), Postdoctoral Science Foundation of China (Grant No. 2014M562493), Postdoctoral Science Foundation of Shanxi Province, and Key Scientific Technological Project of Henan Province (Grant Nos. 122102210126, 092101210502).

X. Sun et al. (Eds.): ICCCS 2016, Part I, LNCS 10039, pp. 377–388, 2016.
DOI: 10.1007/978-3-319-48671-0_34

deployed in unattended environments and collect valuable data of interest. In order to protect these valuable data from unauthorized users or even malicious adversaries, user authentication and data confidentiality are primary concerns in WSNs before accessing data from sensor nodes [1–4]. User authentication ensures the validity of the user while data confidentiality requires the user and the sensor node to establish a common secret key to encrypt the collected data.

Typically, an authenticated key exchange protocol in WSNs involves a user, a gateway node and a sensor node. When a user wants to access real-time data from a sensor node, he will send a query to the gateway node. The gateway node will verify the validity of the user. If the user is a qualified one, then the gateway will assist the user and the sensor node to establish a common secret key to realize data integrity and confidentiality for the upcoming data transmission. Currently, two-factor authenticated key exchange (TFAKE) protocols based on smart card and password are the most popular authentication mechanism in WSNs. TFAKE protocol is an approach to authenticate someone which requires the presentation of two different kinds of authentication factors (smart card and password in this case). The adversary has to compromise both the smart card and the password to impersonate the user. By combining the advantages of smart cards and passwords, TFAKE protocols achieve high-level security without additional computation cost.

In 2009, Das [5] presented the first two-factor user authentication scheme using smart card and password. Das claimed his scheme can resist replay attack, stolen-verifier attack, guessing attack, and impersonation attack. However, Das's scheme is found to be insecure against various attacks. Nyang et al. [6] demonstrated that Das's scheme is insecure against off-line dictionary attack, sensor node compromising attack, and does not protect query response messages. They also proposed an improved scheme to overcome the drawbacks of Das's scheme. Chen et al. [7] showed that Das's scheme does not provide mutual authentication and proposed their improvement. He et al. [8] found that Das's scheme is vulnerable to the insider attack and the derived impersonation attack. Khan et al. [9] pointed out that Das's scheme is vulnerable to the gateway node bypassing attack and privileged insider attack, it does not provide methods to change users' passwords, and it does not achieve mutual authentication between the GW-node and the sensor node. Khan et al. also presented an improved scheme to overcome the security weaknesses of Das's scheme. Unfortunately, Sun et al. [10] showed that Khan et al.'s scheme still suffers from the GW-node impersonation attack, the GW-node bypassing attack, and the privileged insider attack. They proposed a new user authentication scheme which is proved to be secure under the security model of Bellare and Rogaway [11]. Recently, Yuan [12] also found that in Khan et al's scheme, there is no provision of non-repudiation, it is susceptible to attack due to a lost smart card, and mutual authentication between the user and the GW-node does not attained. To fix these weaknesses, Yuan proposed an improved scheme using user's biometrics and proved the security of the new scheme by the GNY logic [13]. Nevertheless, Wei et al. [14] pointed out several secure loopholes of Yuan's scheme and also presented their improvement.

Although there are many TFAKE protocols proposed in the literature, most of them only have heuristic security arguments and are found to be insecure. Researchers show great interest to cryptanalyze the existing TFAKE protocols for WSNs and fix the shortcomings. Little attention has been paid to formal security analysis of TFAKE protocols for WSNs. Until now, to the best of our knowledge, there are only two TFAKE protocols for WSNs which have rigorous security proof. The first one is Sun et al.'s protocol [10]. However, their protocol is proven secure in Bellare and Rogaway's security model, which is a security model for key exchange protocols rather than a model for TFAKE protocols. Their protocol employs "challenge-response" technique which makes it inefficient in term of communication. Moreover, their protocol does not provide session key establishment for the user and the sensor node. Another provably secure TFAKE protocol is due to Nam et al. [15]. Their protocol is the first provably secure TFAKE protocol for WSNs with user anonymity. Nevertheless, their protocol uses computation-expensive public key operations, which makes their protocol unsuitable for WSNs because of the resource-constrained nature of sensor nodes. Furthermore, it is easier for TFAKE protocols to achieve two-factor security and user anonymity when public key cryptosystems are employed.

In this paper, we investigate how to design provably secure TFAKE protocols for WSNs without using computation-expensive public key operations. We first present a security model for TFAKE protocols in WSNs based on the security models of [16,17]. We then propose an efficient TFAKE protocol with user anonymity by using symmetric encryptions and hash functions. We also explain the design rationales for a better understanding of our protocol. The novel TFAKE protocol is proven secure in the random oracle model. Based on the security proof and the performance evaluation, we believe that the proposed TFAKE protocol is more secure and efficient than other related protocols.

The remainder of this paper is organized as follows. In Sect. 2, we summarize the attacks and security requirements of TFAKE protocols in WSNs. In Sect. 3, our proposed TFAKE protocol is described. The security proof of our protocol is conducted in Sect. 4. We compare the efficiency and security features of our protocol with related protocols in Sect. 5. In Sect. 6, we conclude the paper with a brief summary.

2 Attacks and Security Requirement

In this section, we first describe the communication model for TFAKE protocols in WSNs. We then summarize the attacks against TFAKE protocols in WSNs and present the security requirements for these protocols.

2.1 Communication Model

A TFAKE protocol in WSNs involves three kinds of participants: users, gateway nodes and sensor nodes. Plenty of resource-constrained sensor nodes are deployed in unattended environments to collect information of interest. These sensor nodes

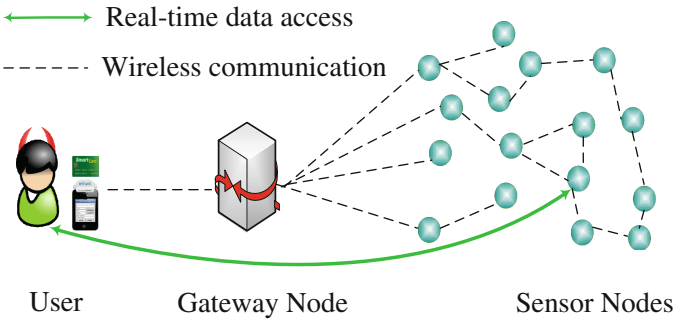

Fig. 1. Communication model

communicate with each other through multi-hop transmissions. In order to get access to the collected data of the sensor nodes, a user should first register himself to the gateway node. Whenever a registered user wants to get real-time data from the sensor nodes, he first sends a query to the gateway node and the gateway node will authenticate the validity of the user. If the user is an enrolled one, the gateway node will further help the user and the target sensor node to establish a common secret session key. The gateway node can be viewed as an interface between the user and the sensor node. A typical communication model for TFAKE protocol in WSNs is shown in Fig. 1.

2.2 Summary of Attacks

In this subsection, we summarize all the attacks against TFAKE protocols in WSNs. The attacks against TFAKE protocols in WSNs are listed and explained in the following:

1. **Privileged-Insider Attack.** It is a common practice that a user only remembers several passwords and uses the same password in different application scenarios for convenience. In a TFAKE protocol in WSNs, if the gateway node knows the password of the user, he can use the password to impersonate the user to get access to other servers. So it is desirable the gateway node does not know the password of the user to thwart the privileged-insider attack.
2. **Impersonation Attack.** The adversary impersonates a participant to deceive other protocol participants. Many attacks can be classified as the impersonation attack.
3. **Stolen-Verifier Attack.** Typically, the gateway node maintains a password verifier table of all the registered users. If the adversary compromises the gateway node and steals the password verifier table, he can extract the password from the verifier table and impersonate all the users at will. Consequently, it is desirable the gateway node does not maintain such a password-related verifier table.

4. **Replay Attack.** The adversary records a valid protocol messages and later replays the intercepted message to impersonate the legal user to the gateway node. The replay attack can be viewed as a special kind of impersonation attack.
5. **Password Guessing Attack/Dictionary Attack.** The password is human-memorable and low-entropy. If an adversary can get a verification equation of the password, he can enumerate all the passwords from the dictionary and verify its guess until the corrected password is found. Password guessing attack can be classified into on-line attack and off-line attack. It is desirable that the off-line password guessing attack should be impossible and the on-line password guessing attack can be detected by the victim participant.
6. **Node Capture Attack.** Since sensor nodes are usually deployed in unattended or hostile environments, the adversary can easily compromise a sensor node and extract the secret information stored in it. However, the adversary should not impersonate other protocol participants (such as un-compromised sensor nodes or a user) by the captured sensor node.
7. **GW-Node Bypassing Attack.** The gateway node shares different secrets with the users and the sensor nodes, respectively. GW-node bypassing attack details that the adversary can compromise a user's secret and authenticates himself directly to the sensor node. In other words, the gateway node is bypassed during the authentication. This attack is a special kind of impersonation attack.
8. **Password Guessing Attack by Insiders.** This attack basically belongs to the password guessing attack. If an TFAKE protocols in WSNs is not well-designed, the insiders (e.g. other users, the gateway node) have some advantage in guessing the victim user's password. It is desirable that the insiders should not get the password information.
9. **Parallel Session Attack.** The adversary executes two protocol sessions in parallel. The adversary tries to use the transcripts of one session to impersonate a valid participant in another session. This attack is basically an impersonation attack.
10. **Stolen Smart Card Attack/The Smart Card Breach Attack.** The adversary steals a user's smart card and then extracts all the information stored in the smart card via side channel attacks. It is desirable that the adversary cannot impersonate the victim user with the breached smart card. In other words, the adversary should not compromise the password by the stolen smart card.
11. **Many Logged-in Users with the Same Login-ID Attack.** When the users register to the gateway node, the gateway node may store the same secret in different smart cards which are belong to different users. As a result, a malicious user can use the common secret to log in the name of other users.
12. **Reflection Attack.** In this attack, The adversary simply sends back (reflect) the message generated by the target participant to himself. The essential idea of the attack is to trick the target into providing the response to its own challenge.

Some ordinary attacks, such as the man-in-the-middle attack and the denial of service attack, are omitted. The reason is that these attacks are either included in another attack (i.e. the man-in-the-middle attack belongs to impersonation attack) or cannot be solved by cryptology methodology (i.e. denial of service attack is always possible no matter how the protocol is designed). It also should be noted that our summary of attacks is based on earlier work of [18,19]. However, our summary is more comprehensive and tailers to TFAKE protocol in WSNs. Based on the summary of the attacks, we present a formal security model for TFAKE protocols in WSNs. The security model will be presented in the full version due to lack of space.

3 Our Proposed Protocol

In this section, we propose an efficient TFAKE protocol for WSNs based on robust authenticated encryption (RAE) schemes [20]. Unlike traditional symmetric encryption schemes which only ensure data confidentiality, an RAE scheme can achieve both data confidentiality and authenticity. The authenticity of the ciphertext enables us to prove the security of our protocol rigorously. For a better understanding of the paper, we briefly introduce the definition of RAE schemes. Fix an alphabet Σ. Typically Σ is $\{0,1\}$ or $\{0,1\}^8$. An RAE scheme is defined as a triple $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$. The key space $\mathcal{K}$ is a set of strings with an associated distribution. The encryption algorithm $\mathcal{E}$ is deterministic and maps a five-tuple $(K, N, A, M, \lambda) \in (\Sigma^*)^3 \times N \times \Sigma^*$ to a string $C = \mathcal{E}_K^{N,A,\lambda}(M)$ of length $|M| + \lambda$, where K is the encryption key, N is a nonce, A is the associated data, M is the message and λ is the ciphertext expansion. λ can be 0 and thus can be omitted sometimes. The decryption algorithm $\mathcal{D}$ is deterministic and takes a five-tuple (K, N, A, M, λ, C) to a value $\mathcal{D}_K^{N,A,\lambda}(C) \in \Sigma^* \cup \{\perp\}$. It is required that $\mathcal{D}_K^{N,A,\lambda}(\mathcal{E}_K^{N,A,\lambda}(M)) = M$ for all K, N, A, M, λ. If there is no M such that $C = \mathcal{E}_K^{N,A,\lambda}(M)$, then $\mathcal{D}_K^{N,A,\lambda}(C) = \perp$. For more details, refer to [20].

There are three phases in our protocol: the registration phase, the authentication and key exchange phase and the password updating phase.

3.1 Registration Phase

When registering with the gateway node GW, the user U_i chooses his own low-entropy password PW_i and a high-entropy random number b, then U_i computes $PW_i^* = h(PW_i \| b)$ and sends the message (ID_i, PW_i^*) to the gateway node through a secure channel, where ID_i is U_i's real identity. Upon receiving the message, the gateway node GW computes $V_i = h(ID_i \| K)$ and $N_i = V_i \oplus h(ID_i \| PW_i^*)$. The GW-node GW also chooses U_i's pseudo identity DID_i for user anonymity and records the list (DID_i, ID_i) in its data base. At last, GW generates a smart card with parameters $(DID_i, N_i, h(\cdot))$, and sends the user's smart card to U_i through a secure channel. Upon receiving the smart card, U_i updates the parameters by adding the random number b to the smart card.

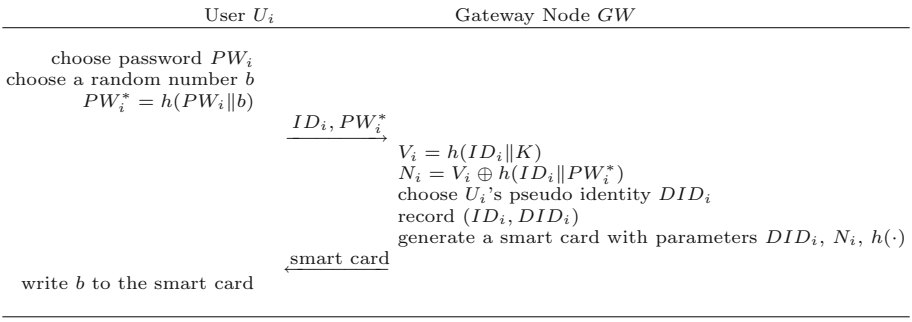

Fig. 2. Registration phase of the proposed scheme

3.2 Authentication and Key Exchange Phase

When a user U_i wants to access the real-time data from a sensor node SN_j, U_i invokes the authentication and key exchange phase with the gateway node GW. If U_i is a valid user, he will share a common secret key with the sensor node SN_j at the end of this phase. For a pictorial description, refer to Fig. 3.

1. U_i inserts his smart card into the card reader, inputs his identity ID_i and password PW_i. The smart card computes $PW_i^* = h(PW_i \| b)$ and recovers $V_i^* = N_i \oplus h(ID_i \| PW_i^*)$. After this, the smart card computes an encryption key $k_1 = h(V_i^* \| T_1)$, where T_1 is the current timestamp in U_i's system. The smart card also chooses a random number R_1 from the space $\{0,1\}^l$ and encrypts the random number (R_1) using a robust authenticated encryption scheme to get the ciphertext $C_1 = E_{k_1}^{T_1, DID_i}(R_1)$, where the timestamp T_1 is used as the nonce and the pseudo identity DID_i of U_i is the associated data. Finally, the smart card sends the message (DID_i, SN_j, C_1, T_1) to the gateway node GW.
2. Upon receiving the message (DID_i, C_1, T_1) at time T_1^*, the gateway node GW checks whether $T_1^* - T_1 \leq \triangle T$ or not, where $\triangle T$ denotes the upper bound of time interval for the transmission delay. If it is true, GW finds U_i's real identity ID_i in the data base using the pseudo identity DID_i and computes the decryption key $k_1 = h(h(ID_i \| K) \| T_1)$. GW then decrypts the ciphertext C_1 and computes $R_1 = D_{k_1}^{T_1, DID_i}(C_1)$. If the decryption operation succeeds, U_i is authenticated by the gateway node GW. GW computes an encryption key $k_2 = h(ID \| h(K \| SN_j) \| T_2)$, where T_2 is the current timestamp of the GW-node's system. Finally, GW encrypts the random number R_1 using the robust authenticated encryption scheme to get the ciphertext $C_2 = E_{k_2}^{T_2, ID}\{R_1\}$, where the timestamp T_2 is used as the nonce and $ID = (DID_i \| GW \| SN_j)$ is the associated data. Finally, GW sends the message (ID, C_2, T_2) to the sensor node SN_j.
3. Upon receiving the message (ID, C_2, T_2) at time T_2^*, the sensor node SN_j checks if $T_2^* - T_2 \leq \triangle T$. If it is true, S_n computes $k_2 = h(ID \| x_j \| T_2)$ and decrypts C_2 and computes $R_1 = D_{k_2}^{T_2, ID}\{C_2\}$. If the decryption operation

succeeds, SN_j computes the encryption key $k_3 = h(ID\|R_1\|T_3)$, where T_3 is the current timestamp of SN_j's system. After that, S_n chooses a random number R_2 from the space $\{0,1\}^l$ and encrypts R_2 using k_3 to get the ciphertext $C_3 = E_{k_3}^{T_3,ID}\{R_2\}$, where the timestamp T_3 is used as the nonce and $ID = (DID_i\|GW\|SN_j)$ is the associated data. SN_j sends the message (C_3, T_3) to the user U_i. Finally, SN_j computes the session key $sk = h(ID\|T_3\|R_1\|R_2)$ for future communications with the user U_i and accepts the session.

4. Upon receiving the message (C_3, T_3) at time T_3^*, U_i checks if $T_3^* - T_3 \leq \triangle T$. If it is true, GW computes $k_3 = h(ID\|R_1\|T_3)$ and decrypts $R_2 = D_{k_3}^{T_3,ID}\{C_4\}$. If the decryption operation succeeds, U_i accepts the session and computes the session key $sk = h(ID\|T_3\|R_1\|R_2)$ for future communications with the sensor node SN_j.

Finally, U_i and SN_j could use the common session key sk in upcoming private communications.

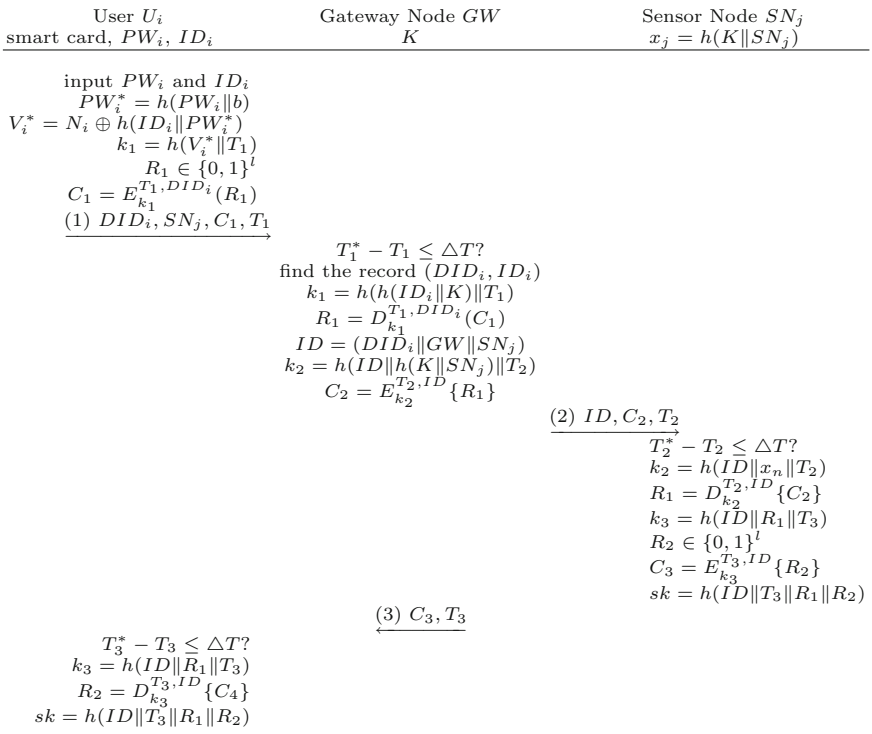

Fig. 3. Authentication and key exchange phase

3.3 Password Updating Phase

This phase is invoked whenever U_i wants to change his password PW_i with a new one, say $PW_i^{'}$. U_i inserts his smart card into the terminal and inputs his identity ID_i, the old password PW_i and the new password $PW_i^{'}$. The smart card computes $N_i^{'} = N_i \oplus h(ID_i \| h(PW_i \| b)) \oplus h(ID_i \| h(PW_i^{'} \| b))$ and replaces N_i with $N_i^{'}$.

4 Security Proof

In this section, we present the security proof of our protocol within the security model given in Sect. 3. Due to lack of space, the security proof of Theorem 1 will be presented in the full version.

Theorem 1. *Let $\mathcal{P}$ be our TFAKE protocol. If the encryption scheme used in our protocol achieves RAE security, and the hash function used in our protocol is a random oracle. Let $\mathcal{A}$ be an PPT adversary, then the adversary's advantage in attacking the session key security and authentication security of the proposed protocol is negligible.*

5 Performance Analysis

In this section, we compare the performance of our protocol with other related protocols [5,6,8–10,15]. The comparison of computation and communication costs are demonstrated in Table 1. In terms of computation, let "H" denote the

Table 1. Comparisons of efficiency

	Our protocol	Das's protocol [5]	N-L protocol [6]	H-G-C protocol [8]	K-A protocol [9]	S-L-F protocol [10]	N-K-P protocol [15]
E1	H	0	0	H	H	0	H
E2	2H	3H	3H	5H	2H	2H	T_{sym}
E3	5H+2T_{sym}	4H	7H+T_{sym}	5H	4H	2H	5H+3T_{pub}
E4	4H+2T_{sym}	4H	8H+T_{sym}	5H	5H	5H	6H+T_{pub}
E5	3H+2T_{sym}	H	4H+2T_{sym}	H	2H	2H	3H+2T_{pub}
E6	4H	N/A	N/A	6H	4H	2H	2H
E7	832 bits	832 bits	1344 bits	928 bits	992 bits	1056 bits	2144 bits
E8	3	3	3	3	3	8	4

E1: Computation cost of the registration phase for a user
E2: Computation cost of the registration phase for a GW-node
E3: Computation cost of the authentication phase for a user
E4: Computation cost of the authentication phase for a GW-node
E5: Computation cost of the authentication phase for a sensor node
E6: Computation cost of the password updating phase for a user
E7: Bandwidth of the authentication phase
E8: Message flows of the authentication phase
N/A:Not Available

computation cost of one hash operation, "T_{pub}" denote the computation cost of one public key operation, "T_{sym}" denote the computation cost of one symmetric key encryption/decryption. Note that the encryption/decryption cost of an RAE scheme is the same as that of symmetric key encryption/decryption. In terms of communication, we consider bandwidth and round complexity. We assume the identifications can be represented with 32 bits, the output size of secure hash functions/Nonces is 160 bits, the timestamp can be represented with 64 bits. The ciphertext is the same size with the plaintext in symmetric encryptions, and the size of the ciphtext is usually doubled in public key encryptions.

We can see from Table 1 that the computation costs of the registration phase and the password updating phase are more or less the same. Consequently, we focus on the computation cost of the authentication phase. Our protocol needs 12 hash operations and 6 symmetric encryption/decryption operations in the authentication phase. The symmetric encryption/decryption operations arise from the distribution of the session key. Without the symmetric encryption/decryption cost, our protocol is as efficient as other protocols. Nam et al.'s protocol [15] uses public key operation, so it is very inefficient compared with other protocols. In terms of communication, our protocol is the most efficient with respect to bandwidth and achieves optional round complexity. In wireless sensor networks, transmitting radio signals on resource-constrained wireless devices usually consumes much more power than computation does, so it is more important to reduce the communication cost than the computation cost. As a result, our protocol is very attractive in terms of efficiency.

Table 2. Comparisons of security features

	Our protocol	Das's protocol [5]	N-L-protocol [6]	H-G-C-protocol [8]	K-A-protocol [9]	S-L-F-protocol [10]	N-K-P protocol [15]
C1	Yes	Yes	Yes	Yes	Yes	Yes	Yes
C2	Yes	No	No	Yes	No	Yes	Yes
C3	Yes	No	No	Yes	No	Yes	Yes
C4	Yes	Yes	Yes	Yes	Yes	Yes	Yes
C5	No	No	No	No	No	No	Yes
C6	Yes	No	Yes	Yes	Yes	Yes	Yes
C7	Yes	No	No	No	No	Yes	Yes
C8	Yes	No	No	No	No	Yes	Yes
C9	Yes	No	Yes	No	No	No	Yes
C10	Yes	No	No	No	No	No	Yes
C11	Yes	No	Yes	No	No	No	Yes

C1: Resist the replay attack
C2: Resist the privileged insider attack
C3: Resist the impersonation attack
C4: Resist the stolen verifier attack
C5: Resist the stolen smart card attack
C6: Resist the off-line dictionary attack
C7: Resist the node capture attack
C8: Mutual authentication
C9: Session key distribution
C10: User anonymity
C11: Provable security

Table 2 summarizes security features of our protocol with related protocols [5,6,8–10,15]. We can see from Table 2 that our protocol provides more security features than other related protocols. The only disadvantage of our protocol is its vulnerability against the stolen smart attack. However, it is noted in [19] that it is impossible to resist the stolen smart card attack merely using symmetric cryptology mechanism.

Considering the computation cost, communication cost and security features as a whole, our protocol achieves provable security and outperforms other related protocols. Therefore, our protocol is more secure than related scheme while preserving high efficiency. As a result, it is more suitable for real-life applications in WSNs.

6 Conclusions

In this paper, we summarize the security requirements of TFAKE protocols in WSNs and present a formal security model to evaluate their security. We also put forward an efficient TFAKE protocol based on robust authenticated encryption schemes and prove the security of our protocol in the random oracle model. Comparison shows that our protocol not only enjoys provable security but also has high efficiency in terms of communication and computation. To the best of our knowledge, our protocol is the first TFAKE protocol which introduces robust authenticated encryption schemes to achieve provable security.

References

1. Guo, P., Wang, J., Li, B., Lee, S.: A variable threshold-value authentication architecture for wireless mesh networks. J. Internet Technol. **15**(6), 929–936 (2014)
2. Shen, J., Tan, H., Wang, J., Wang, J., Lee, S.: A novel routing protocol providing good transmission reliability in underwater sensor networks. J. Internet Technol. **16**(1), 171–178 (2015)
3. Xie, S., Wang, Y.: Construction of tree network with limited delivery latency in homogeneous wireless sensor networks. wirel. Pers. Commun. **78**(1), 231–246 (2014)
4. He, D.B., Kumar, N., Chen, J.H., et al.: Robust anonymous authentication protocol for health-care applications using wireless medical sensor networks. Multimedia Syst. **21**(1), 49–60 (2015)
5. Das, M.L.: Two-factor user authentication in wireless sensor networks. IEEE Trans. Wirel. Commun. **8**(3), 1086–1090 (2009)
6. Nyang, D.H., Lee, M.K.: Improvement of Das's two-factor authentication protocol in wireless sensor networks. Cryptology, ePrint archive. http://eprint.iacr.org/2009/631.pdf
7. Chen, T.H., Shih, K.K.: A robust mutual authentication protocol for wireless sensor networks. ETRI J. **32**(5), 704–712 (2010)
8. He, D.J., Gao, Y., Chan, S.: An enhanced two-factor user authentication scheme in wireless sensor networks. Ad Hoc Sens. Wirel. Netw. **10**(4), 1–11 (2010)

9. Khan, M.K., Alghathbar, K.: Cryptanalysis and security improvements of two-factor user authentication in wireless sensor networks. Sensors **10**(3), 2450–2459 (2010)
10. Sun, D.Z., Li, J.X., Feng, Z.Y.: On the security and improvement of a two-factor user authentication scheme in wireless sensor networks. Pers. Ubiquit. Comput. **17**(5), 895–905 (2013)
11. Bellare, M., Rogaway, P.: Entity authentication and key distribution. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 232–249. Springer, Heidelberg (1994)
12. Yuan, J.J.: An enhanced two-factor user authentication in wireless sensor networks. Telecommun. Syst. **55**(1), 105–113 (2014)
13. Gong, L., Needham, R., Yahalom, R.: Reasoning about belief in cryptographic protocols. In: Proceedings of 1990 IEEE Computer Society Symposium Research in Security and Privacy, pp. 234–246 (2009)
14. Wei, F.S., Ma, J.F., Jiang, Q., et al.: Cryptanalysis and improvement of an enhanced two-factor user authentication scheme in wireless sensor networks. Inf. Technol. Control **45**(1), 62–70 (2016)
15. Nam, J., Kim, M., Paik, J., et al.: A provably-secure ECC-based authentication scheme for wireless sensor networks. Sensors **14**(11), 21023–21044 (2014)
16. Pointcheval, D., Zimmer, S.: Multi-factor authenticated key exchange. In: Bellovin, S.M., Gennaro, R., Keromytis, A., Yung, M. (eds.) ACNS 2008. LNCS, vol. 5037, pp. 277–295. Springer, Heidelberg (2008)
17. Bellare, M., Pointcheval, D., Rogaway, P.: Authenticated key exchange secure against dictionary attacks. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 139–155. Springer, Heidelberg (2000). doi:10.1007/3-540-45539-6_11
18. Alsaleh, M., Mannan, M., Van Oorschot, P.C.: Revisiting defenses against large-scale online password guessing attacks. IEEE Trans. Dependable Secure Comput. **9**(1), 128–141 (2012)
19. Wang, D., He, D., Wang, P., et al.: Anonymous two-factor authentication in distributed systems: certain goals are beyond attainment. IEEE Trans. Dependable Secure Comput. **12**(4), 428–442 (2015)
20. Hoang, V.T., Krovetz, T., Rogaway, P.: Robust authenticated-encryption AEZ and the problem that it solves. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9056, pp. 15–44. Springer, Heidelberg (2015). doi:10.1007/978-3-662-46800-5_2