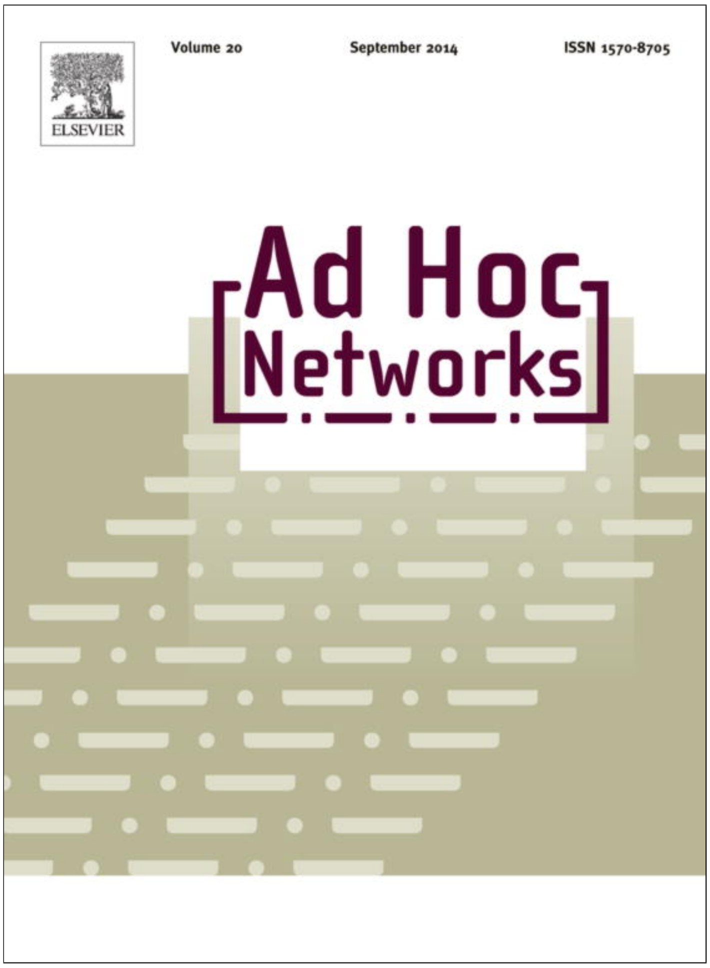
ELSEVIER
Volume 20
September 2014
ISSN 1570-8705
Ad Hoc Networks

Ad Hoc Networks 20 (2014) 1–15

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

Understanding security failures of two-factor authentication schemes for real-time applications in hierarchical wireless sensor networks

Ding Wang [*], Ping Wang

College of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China
National Engineering Research Center for Software Engineering, Beijing 100871, China

ARTICLE INFO

Article history:
Received 19 June 2013
Received in revised form 10 January 2014
Accepted 11 March 2014
Available online 21 March 2014

Keywords:
Password authentication
Hierarchical wireless sensor networks
User anonymity
Smart card
Non-tamper resistant

ABSTRACT

Understanding security failures of cryptographic protocols is the key to both patching existing protocols and designing future schemes. In this work, we investigate two recent proposals in the area of smart-card-based password authentication for security-critical real-time data access applications in hierarchical wireless sensor networks (HWSN). Firstly, we analyze an efficient and DoS-resistant user authentication scheme introduced by Fan et al. in 2011. This protocol is the first attempt to address the problems of user authentication in HWSN and only involves lightweight cryptographic primitives, such as one-way hash function and XOR operations, and thus it is claimed to be suitable for the resource-constrained HWSN environments. However, it actually has several security loopholes being overlooked, and we show it is vulnerable to user anonymity violation attack, smart card security breach attack, sensor node capture attack and privileged insider attack, as well as its other practical pitfalls. Then, A.K. Das et al.'s protocol is scrutinized, and we point out that it cannot achieve the claimed security goals: (1) It is prone to smart card security breach attack; (2) it fails to withstand privileged insider attack; and (3) it suffers from the defect of server master key disclosure. Our cryptanalysis results discourage any practical use of these two schemes and reveal some subtleties and challenges in designing this type of schemes. Furthermore, using the above two foremost schemes as case studies, we take a first step towards investigating the underlying rationale of the identified security failures, putting forward three basic principles which we believe will be valuable to protocol designers for advancing more robust two-factor authentication schemes for HWSN in the future.

1. Introduction

With the rapid development of micro-electromechanical systems and wireless network technologies, wireless sensor networks (WSNs) have drawn increasing interest from both academic and industrial areas due to its easy deployment, ubiquitous nature and wide range of applications. Generally speaking, there are two architectures available for WSNs: the distributed flat architecture and the hierarchical architecture. Most large-scale WSNs prefer to follow the latter, for it is more energy-efficient and has more operational advantages than its flat counterpart [1]. In hierarchical wireless sensor networks (HWSN), there is a hierarchy among the nodes based on their capabilities: base station, cluster heads and sensor nodes. The HWSN is divided into a number of clusters to enhance its flexibility and to save energy consumption. Each cluster

[*] Corresponding author at: College of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China. Tel.: +86 185 1134 5776; fax: +86 010 6275 4993.

E-mail address: wangdingg@mail.nankai.edu.cn (D. Wang).

http://dx.doi.org/10.1016/j.adhoc.2014.03.003
1570-8705/© 2014 Elsevier B.V. All rights reserved.

is administered by a cluster head. Sensor nodes communicate with each other in the same cluster and finally communicate with the cluster head via one-hop or multi-hop transmission(s). The base station is typically a gateway to another network, a powerful data processing and storage center, or also an access point for human interface. In this study, we focus mainly on HWSN, and more details about HWSN can be found in [2,3].

In many security-critical applications, such as real-time traffic control, industrial process control, healthcare monitoring and military surveillance, external users are generally interested in accessing real-time information from sensor nodes. To facilitate the external users to access the real-time data directly from the desired sensor nodes inside HWSN without involving the base station or gateway node as and when demanded (as illustrated in Fig.1), it is of utter importance to protect the users and systems' security and privacy from malicious adversaries because of the broadcast nature of wireless communications. Accordingly, user authentication becomes an essential security mechanism for the external user to be first authorized to the base station (or the gateway node) as well as the sensor nodes before granting his/her access to the real-time data. However, given the stringent constraints on memory capacity, computation capability, bandwidth and energy consumption of sensor nodes, it still remains quite a challenging problem to design an efficient and secure remote user authentication scheme for real-time data access directly from the desired sensor nodes inside HWSN.

To address the above issues, in 2011, Fan et al. [4] proposed the first smart-card-based password authentication scheme for HWSN. This proposal involves only lightweight operations, such as one-way hash function and exclusive-OR operations, which is well-suited to the large-scale resource-limited sensor networks. The authors claimed that their scheme is free from various related cryptographic attacks, such as smart card security breach attack, offline password guessing attack, replay attack and impersonation attack. Although their scheme is efficient and has been equipped with a formal security proof, we find it actually cannot achieve the claimed security goals: (1) it cannot preserve user anonymity and (2) it is vulnerable to smart card security breach attack. The authors also overlook dimensions on which their scheme fares poorly, such as the feature of local password change, resistance to node capture attack and insider-attack.

More recently, A.K. Das et al. [5] proposed a novel user authentication scheme based on traditional password and smart card to provide user access to real-time data by authorizing her directly at node level for HWSN. As with Fan et al.'s scheme [4], this protocol also only involves hash and XOR operations, with no additional symmetric encryption or asymmetric computations, and thus it is very efficient. This scheme also possesses many attractive features, such as dynamic node addition, local password change and session key establishment. Therefore, it exhibits great potential for practical applications. The authors claimed that their scheme provides better security as compared with the other related works, such as mutual authentication between the user and the cluster heads, resistance to privileged-insider attack, denial-of-service attack, node capture attack and smart card security breach attack. However, in this study, we demonstrate that it still cannot achieve the claimed security goals: (1) It is vulnerable to smart card security breach attack; (2) it fails to withstand privileged-insider attack; and (3) it suffers from the problem of server master key disclosure.

There have been a number of papers [6–13] dealing with security vulnerabilities in smart-card-based password authentication (i.e., two-factor authentication [14]) schemes for WSNs. In these studies, the authors only focus on presenting attacks on target protocols and proposing 'enhanced schemes', nevertheless, little (to the best of our knowledge, actually no) attention is paid to the underlying rationales of the identified security failures, and to the design principles of a sound proposal. Unsurprisingly, the same mistakes are repeated over and over again. For example, many schemes using non-tamper resistant smart

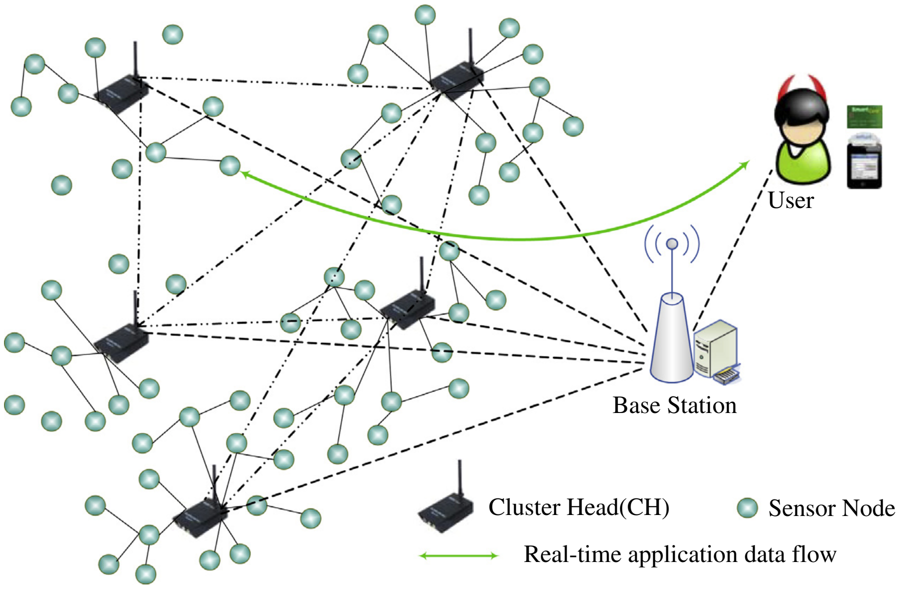

Fig. 1. Direct data access in hierarchical wireless sensor networks (HWSN).

cards attempt to only employ symmetric cryptographic operations, such as hash functions, symmetric encryption and XOR operations, to achieve some advanced security goals like user anonymity. And the scheme proposed by He et al. [8] in 2010 falls into this category. In 2011, Kumar and Lee found this scheme actually cannot preserve user anonymity. Being fully aware of this vulnerability in He et al.'s scheme, the authors of [5,15–17] still endeavor to construct schemes that can preserve user anonymity while not employing public-key techniques. Unfortunately, as we will point out later, such attempts are to beat the air. To improve this situation, in this work, through the security analysis of the above two schemes and based on our past cryptanalysis experience, we put forward three suggestions that are crucial for designing more robust two-factor authentication schemes for WSNs, in the hope that no similar structural mistakes are made in the future.

The remainder of this paper is organized as follows: in Section 2, we review Fan et al.'s scheme. Section 3 describes the weaknesses of Fan et al.'s scheme. A.K. Das et al.'s scheme is reviewed in Section 4 and the corresponding cryptanalysis is given in Section 5. Section 6 discusses the lessons learned from the cryptanalysis and the conclusion is drawn in Section 7.

2. Review of Fan et al.'s scheme

For a self-contained discussion, we briefly review the first efficient and denial-of-service (DoS)-resistant user authentication scheme for HWSN proposed by Fan et al. [4] in 2011. Fan et al.'s protocol involves four participants, namely, the user (U_i), the base station (BS), the master node (MN_j) (i.e., the cluster head) and the sensor node (SN). Note that, the base station BS is only responsible for the registration and is not involved in the authentication process of U_i, MN_j and SN. There are three phases in Fan et al.'s protocol: registration, login and authentication. For ease of presentation, we employ some intuitive notations listed in Table 1 and will follow the notations in Fan et al.'s scheme as closely as possible.

2.1. Registration phase

Before the running of this phase, it is supposed that each master node, say MN_j, has already been deployed in the designated area and shares the secret number Y_j with the base station, say BS; meanwhile, each sensor node in a cell shares the secret parameter S_k with the master node. Suppose a new user U_i wants to register with the BS for accessing services, the following operations will be performed:

(1) The user U_i chooses the identity ID_i and the password PW_i.
(2) $U_i \Rightarrow BS : \{ID_i, PW_i\}$.
(3) After receiving the registration request, the base station BS selects a random large number R_i and computes $RID_i = h(R_i\|ID_{BS}) \oplus ID_i \oplus ID_{BS}$, where ID_{BS} represents the identity of BS. Then BS calculates $A_i = h(X) \oplus h^2(ID_i\|PW_i)$ and $V_i = h^3(ID_i\|PW_i)$.
(4) BS generates a data table including user U_i's warrant N_{ij} (e.g., the network user's permissions, the period of validity, and the master node's identity) for the relative master nodes and the value of $B_{ij} = h(N_{ij}\|Y_j) \oplus h(ID_i\|PW_i)$. Note that U_i may not need to access all the cells in HWSN, and the warrant table aims to restrict users as to which cell they can login.
(5) BS stores $\{RID_i, R_i\}$ corresponding to ID_i in its database.
(6) $GWN \Rightarrow U_i$: A smart card containing security parameters $h(\cdot), RID_i, V_i, A_i$, and the data table of warrants $\{N_{ij}, B_{ij}\}$.

Table 1
Notations and abbreviations.

Symbol	Description
U_i	ith User
BS	Base station
MN_j	jth Master node
SN	Sensor node
$\mathcal{A}$	The adversary
ID_i	Identity of user U_i
PW_i	Password of user U_i
X	Secret key maintained by BS
Y_j	Secret parameter shared between MN_j and BS
S_k	Secret parameter shared between MN_j and its sensor nodes
N_{ij}	ith User's warrant for jth master node
$\oplus$	The bitwise XOR operation
$\|$	The string concatenation operation
$h(\cdot)$	Collision free one-way hash function
$\rightarrow$	A common communication channel
$\Rightarrow$	A secure communication channel

2.2. Login phase

When user U_i wants to acquire real-time data directly from the sensor nodes, the following operations will be performed:

(1) U_i inserts her smart card into a card reader and inputs her identity ID_i^* and password PW_i^*.
(2) The smart card computes $V_i^* = h^3(ID_i^*\|PW_i^*)$ and verifies whether the computed V_i^* equals the stored V_i. If the verification fails, it indicates that U_i is not a legal card holder and the smart card rejects.
(3) U_i chooses which master node to login, and the smart card reads corresponding values of the selected master node: N_{ij} and B_{ij}. Then, it computes $TK_i = h((B_{ij} \oplus h(ID_i\|PW_i))\|T) = h(h(N_{ij}\|Y_j)\|T)$ and $SID_i = RID_i \oplus h((A_i \oplus h^2(ID_i\|PW_i))\|T) = RID_i \oplus h(h(X)\|T)$, where T is the current timestamp on user side.
(4) The smart card computes $C_1 = SID_i \oplus TK_i$ and $C_2 = h(TK_i\|SID_i\|N_{ij}\|T)$.
(5) $U_i \rightarrow MN_j : \{N_{ij}, C_1, C_2, T\}$.
(6) The smart card erases ID_i, PW_i, SID_i, $h(N_{ij}\|Y_j)$, $h(ID_i\|PW_i)$, and $h^2(ID_i\|PW_i)$ from its memory, but records its last login timestamp T.

2.3. Authentication phase

This phase aims to achieve the goal of mutual authentication among U_i, MN_j and SN. Meanwhile, a session key is established between U_i and SN.

(1) Upon receiving the login request $\{N_{ij}, C_1, C_2, T\}$, MN_j first checks the validity of T, and then computes $TK_i^* = h\left(h\left(N_{ij}^*\|Y_j\right)\|T\right)$, $SID_i^* = C_1 \oplus TK_i^*$ and $C_2^* = h(TK_i^*\|SID_i^*\|N_{ij}\|T)$, where Y_j is the pre-shared key between MN_j and BS. Furthermore, MN_j checks $C_2^* \overset{?}{=} C_2$. If the equality holds, MN_j accepts the login request. Otherwise, the master node MN_j simply rejects.

(2) MN_j stores $\{SID_i, T\}$ for the subsequent operations. Note that MN_j can report U_i's abnormal behaviors to BS; meanwhile, BS can collect usage information about U_i in each MN_j for pricing. If BS receives a message $\{SID_i, T\}$ sent by MN_j, it can calculate $RID_i^* = SID_i \oplus h(h(X)\|T)$ and reveal the real identity of U_i by computing $ID_i = RID \oplus ID_{BS} \oplus h(R_i\|ID_{BS})$.

(3) MN_j computes the temporary authentication key $TKM_j = h(h(N_{ij}\|Y_j)\|SID_i\|T_{MN})$ for U_i, where T_{MN} is the current timestamp, and selects a random number K to generate the session key $Key = h(S_k\|K)$. Afterwards, MN_j computes $D_1 = Key \oplus TKM_j$, $D_2 = h(Key\|TKM_j\|T_{MN})$ and $D_3 = h(K\|T_{MN}\|S_k)$.

(4) $MN_j \rightarrow U_i : \{D_1, D_2, T_{MN}\}$.

(5) $MN_j \rightarrow SN : \{K, T_{MN}, D_3\}$.

(6) Upon receiving the message $\{K, T_{MN}, D_3\}$ from its master node, SN computes $D_3^* = h(K\|T_{MN}\|S_k)$ and checks $D_3^* \overset{?}{=} D_3$, where S_k is shared between MN and its sensor nodes (including the sensor node SN). If the check holds, SN computes the session key $Key = h(S_k\|K)$.

(7) Upon receiving the response from MN_j, the smart card checks the validity of T_{MN}. Then, the smart card requires U_i to re-enter her identity and password for verification. If it is correct, the smart card will recalculate $h(N_{ij}\|Y_j)$ and SID_i as in Step 3 of the login phase. As the last login timestamp T has been cached, $h(N_{ij}\|Y_j)$ and SID_i can be easily computed. Finally, the smart card calculates $TKM_j^* = h(h(N_{ij}\|Y_j)\|SID_i\|T_{MN})$ and $Key^* = D_1 \oplus TKM_j^*$. If $D_2^*(= h(Key^*\|TKM_j^*\|T_{MN}))$ is equal to the received D_2, U_i accepts the session key Key to secure its ensuing data communications.

Remark 1. Note that, Fan et al.'s scheme can only deal with remote user authentication of the application-layer, other security provisions in the same or below layers, such as resistance to physical-layer jamming, sybil attack and bogus-message injection, are out of reach, and thus additional security mechanisms (e.g., [18]) should be adopted for the secure operation of the whole system. Besides, appropriate MAC, clustering and routing schemes (e.g., [19–21]) should also be in place. Another subtlety that needs special attention is that the nodes (i.e., cluster heads and sensor nodes) shall be relatively stationary after deployment, otherwise the roaming authentication (like [22]) will be involved once a sensor has moved from its one cluster to another cluster, which is too computationally and communicationally expensive for sensor nodes.

The above first two implications should also be observed in any of this kind of schemes [5,8,10,12,13], while the last implication does not exist in schemes (e.g., [5]) that can accommodate dynamism of the nodes.

Remark 2. From Fan et al.'s scheme as well as the scheme examined in Section 4 (i.e., A.K. Das et al.'s scheme [5]), we can see that: (1) this kind of schemes are three-party cryptographic protocols and (2) the cluster heads perform much more cryptographic operations (e.g, hash and encryption/decryption) than the normal sensor nodes in the login and authentication process. As these cryptographic operations are computationally intensive, they are undesirable (or even unaffordable) for common sensor nodes which are typically low-powered, computation-constrained. Consequently, this kind of user authentication schemes generally need to exploit the heterogeneities of HWSN, i.e. computational heterogeneity and energy heterogeneity, may also link heterogeneity. This in turn well explains why most works in this area (including ours) mainly focus on HWSN but not general WSNs.

3. Cryptanalysis of Fan et al.'s scheme

There are four assumptions about the adversary's capabilities explicitly made in Fan et al.'s scheme [4]:

(i) The sensitive data stored in the sensor nodes as well as cluster heads can be revealed once they are captured by an adversary $\mathcal{A}$.

(ii) The secret data stored in the smart card can be extracted once a legitimate user's smart card is somehow obtained (e.g. picked up or stolen) by $\mathcal{A}$.

(iii) $\mathcal{A}$ has total control over the communication channel among the user U_i, the base station BS, the cluster head CH_j and the sensor nodes. In other words, the attacker can intercept, block, delete, insert or alter any messages exchanged in the channel.

(iv) The user-memorable identities and passwords are weak, i.e., of low entropy.

It is worth noting that the above four assumptions, which are also made in the latest works for HWSN [6–9,11,15,17,23–25], are indeed reasonable: (1) Assumption i is realistic for wireless sensor networks are often deployed in hostile environments, and both the cluster heads and common sensor nodes, which are usually not equipped with tamper-resistant hardware due to cost constraints, can be physically captured by the adversary; (2) Assumption ii is practical when taking the state-of-the-art side-channel attacks [26–29], software attacks (launched on software-supported card, e.g., Java Card) [30] and reverse engineering techniques [31] into consideration, and also it has been widely accepted as a more prudent and desirable manner in designing two-factor authentication schemes since 2004 (see the dividing line in Fig.1 of [32]): put aside any special security features that could be provided by a smart-card, and simply assume that once the smart-card is in the possession of an adversary, all the sensitive parameters stored in it are known to the adversary; (3) Assumption *iii* is consistent with the common Dolev-Yao adversary model [33] for distributed computing; and (4) Assumption *iv* reveals the reality that users are allowed to choose their passwords at will during the registration phase (and the password change phase), usually the users tend to choose passwords that are related to their personal life for convenience [34,35], such as phone numbers, meaningful dates or license plate numbers, and thus these human-memorable passwords are likely to be "weak passwords" [36]. In addition, even if password-composition policies are in place to increase password strength, their effectiveness is greatly hampered by circumvention strategies employed by the users [37,38], or by inconsistent and even contradictory requirements across systems and web sites [39,40]. User's identity, chosen in the way with the password, is often confined to a predefined format and kept static in its entire life-cycle, and thus it is as weak as (probably weaker [41] than) user's password.

In the following discussions of the security pitfalls of Fan et al.'s scheme, based on the above four assumptions, we assume that an adversary $\mathcal{A}$ can extract the secret parameters $\{h(\cdot), RID_i, V_i, A_i\}$ stored in the legitimate user's smart card, and could also intercept or block the exchanged messages $\{N_{ij}, C_1, C_2, T, D_1, D_2, T_{MN}, K, D_3\}$ during the login and authentication processes. Although Fan et al.'s scheme has many attractive properties, such as provision of mutual authentication between the external user, master node (i.e., gateway node) and sensor node, high efficiency and key agreement, it fails to achieve the claimed security goals of resistance to user anonymity violation attack and smart card security breach attack, and it is susceptible to node capture attack and privileged-insider attack. What's more, it has other practical pitfalls.

3.1. No provision of user anonymity

A protocol with user anonymity protects an individual's sensitive personal information, such as preferences, lifestyles, shopping patterns, and social circle, from being acquired by an adversary through analyzing the login information, the services or the resources being accessed [42,43]. Moreover, in wireless environments, the leakage of user-specific information may facilitate an unauthorized entity to track the user's current location and login history [44]. Hence, user anonymity is a highly desirable feature of user authentication schemes for HWSN.

To preserve user anonymity, a feasible approach is to employ the "dynamic ID technique" [45]: a user's real identity is concealed in the session-variant pseudonym identities. The authentication schemes adopting this technique are the so-called "dynamic-ID" schemes [46,47], and Fan et al.'s scheme falls into this category. Let us see how a dishonest master node MN_m colluding with a malicious privileged user U_m manage to breach the anonymity of any legitimate user, say U_i. Note that the collusion of insiders is a rather practical threat in WSNs, especially when WSNs are deployed in hostile environments [48–50]. U_m having her own smart card can gather information A_m from her own smart card, then she can compute $h(X) = A_m \oplus h^2(ID_m \| PW_m)$ because the malicious user U_m knows her own identity ID_m and password PW_m corresponding to her smart card. With the correct value of $h(X), U_m$ and MN_m can collude to successfully learn some sensitive user-specific information about the legitimate user U_i through the following steps:

Step 1. Upon receiving U_i's login message $\{N_{im}, C_1, C_2, T\}$, MN_m computes $TK_i = h(h(N_{im} \| Y_m) \| T)$, where Y_m is shared between MN_m and BS.

Step 2. MN_m Computes $SID_i = C_1 \oplus TK_i$.

Step 3. MN_m Computes $RID_i = SID_i \oplus h(h(X) \| T)$, where $h(X)$ is obtained through colluding with the malicious privileged user U_m.

It should be noted that, RID_i is kept the same for all the login requests of user U_i and is specific to user U_i. Accordingly, this value RID_i can be seen as user U_i's identification, and the attacker MN_m can, therefore, use this information to identify and track the user U_i's login requests. Our attack well serves to demonstrate the feasibility of user anonymity (un-traceability) violation attack on Fan et al.'s scheme, thereby contradicting the claim "nobody except *BS* can trace the user U_i" that is made in [4].

3.2. Smart card security breach attack

Let us consider the following scenarios. In case a legitimate user U_i's smart card is stolen by the adversary $\mathcal{A}$, and the stored secret parameters $h(\cdot)$ and V_i can be extracted. Note that this assumption is reasonable as described in Assumption *ii* and it is also explicitly made in Fan et al.'s scheme. As a concrete example, in 2008 Nohl and Evans [31] succeeded in conducting a reverse engineering attack on a MIFARE Classic chip (MIFARE) by revealing its proprietary cryptographic algorithms and subsequently breaking the entire system. With the extracted $V_i, \mathcal{A}$ can successfully guess the password of U_i as follows:

Step 1. Randomly choose a pair (ID_i^*, PW_i^*) from $\mathcal{D}_{id} \times \mathcal{D}_{pw}$, where $\mathcal{D}_{id}$ denotes the identity space and $\mathcal{D}_{pw}$ denotes the password space.

Step 2. Computes $V_i^* = h^3(ID_i^* \| PW_i^*)$.

Step 3. Verifies the correctness of PW_i^* by checking if the computed V_i^* equals the extracted V_i.

Step 4. Repeats the above steps until the correct value of $(ID_i^*,\ PW_i^*)$ pair is found.

Note that, in most password-based authentication schemes, for the sake of user-friendliness, a user is often allowed to select her own identity *ID* at will (often confined to a predefined format) during the registration phase; the user usually tends to choose an identity which is easily remembered for her convenience. Consequently, these easy-to-remember identities are of low entropy and thus can also be offline enumerated by an adversary $\mathcal{A}$ within polynomial time in the same way with the passwords. Hence, in practice, it is reasonable and realistic to assume that $\mathcal{A}$ can offline enumerated all the (ID, PW) pairs in the Cartesian product $\mathcal{D}_{id} * \mathcal{D}_{pw}$ within polynomial time, where $|\mathcal{D}_{id}|$ and $|\mathcal{D}_{pw}|$ denote the number of identities in $\mathcal{D}_{id}$ and the number of passwords in $\mathcal{D}_{pw}$, respectively.

The running time of the above attack procedure is $\mathcal{O}(|\mathcal{D}_{id}| * |\mathcal{D}_{pw}| * 3T_H)$, where T_H is the running time for Hash operation. Since both password and identity are human-memorable short strings but not high-entropy keys, in other words, they are often chosen from two corresponding dictionaries of small size. Recently empirical study of large revealed passwords sets also confirmed that, $|\mathcal{D}_{id}|$ and $|\mathcal{D}_{pw}|$ are very limited in practice, e.g. $|\mathcal{D}_{id}| \leqslant |\mathcal{D}_{pw}| \leqslant 10^6$ [35,36]. In conclusion, the above attack can be completed in polynomial time, demonstrating the invalidity of the claim "Through clever design, our proposed scheme can prevent smart card breaches". For a better grasp of the effectiveness of our attack, we further acquire the experimental timings (see Table 2) of the related cryptographic operations by using the publicly-available rational arithmetic C/C++, cryptographic library MIRACL [51]. Each kind of cryptographic operation has been performed one thousand times and the corresponding value is taken an average over them.

One may argue that, if the scheme falls due to the bad behaviors of the user (e.g., choosing a "weak password"), it is the user's fault and not the proposed scheme. However, it's a general principle that an admired and practical cryptographic protocol should follow the users' habits and rely its security on as less strong assumptions as possible, but not simply assume that the protocol will function well only if the user never behaves badly. And particularly, in the password-based protocol setting, what one must ensure is that the protocol can prevent an adversary from attempting to determine the correct password even if it is drawn from a relatively small dictionary (a known space of possibilities) [52]. In this light, our smart card security breach attack is meaningful and constitutes a real challenge to two-factor authentication schemes for HWSN.

3.3. Privileged-insider attack

In WWW 2007, researchers [34] from Microsoft Research conducted a large scale study on password reuse habits. Their statistical data was obtained from more than half a million users over a period of three months, and it is reported that the average user maintains about 6.5 passwords and 25 accounts that require passwords, which means each password is shared across 3.9 different sites. This work and some subsequent ones [53,54] well cohere with the common practice that, the users tend to use the same passwords (or slight variations) to access several servers for their convenience. In such situations, if a privileged-insider of *BS*, e.g., the administrator, has learned the user's password, she may manage to impersonate the victim user to access other application systems. Therefore, the resistance to privileged-insider attack is an admired security feature for authentication schemes in WSNs [5,9,17].

In Fan et al.'s scheme, U_i simply submits the plain-text password to *BS* in the registration phase. Now, if U_i uses this PW_i to access other systems for her convenience, the malicious insider can impersonate U_i to login by abusing the legitimate user's password and thus gets access to other systems. Therefore, Fan et al.'s scheme is susceptible to privileged-insider attack. Though it may be possible that all the privileged insiders of *BS* are trusted parities and U_i never uses the same password to access other servers, the implementers and users of the scheme shall be well aware of such a potential risk.

3.4. Node capture attack

The resistance against node capture attack is a basic security requirement for authentication schemes in WSNs [5,17]. In [4], Fan et al. explicitly stated that, "the sensor nodes can be easily captured by the intruder, since WSNs are deployed in a hostile environment." In the light of this statement, however, we find Fan et al.'s scheme fails to achieve this security requirement. Once a sensor node has been captured by $\mathcal{A}$, the node capture attack can be launched as follows:

Step 1. Extracts the security parameter S_k from the captured node's memory, which is practical as stated in Assumption *i*.

Step 2. Intercepts the message $\{K^*, T^*_{MN}, D^*_3\}$ that MN_j sends to any other non-captured sensor nodes in the same cell.

Step 3. Computes the session key $Key = h(S_k \| K^*)$, where S_k is shared among all the sensor nodes and its master node.

The above attack demonstrates that, once a sensor node inside a cell is captured, the communications that are conducted by other non-captured sensor nodes in the same cell are endangered, which is obviously undesirable.

3.5. Other practical pitfalls

In Fan et al.'s scheme, there is no password change procedure provided. That is to say, the users cannot change their passwords regularly, which is not a recommended practice, for the fixed password is definitely more vulnerable than a periodically changed one.

Table 2
Experimental timings of related operations on common Laptop PCs.

Experimental platform (common PCs) (GHz)	Asymmetric operation (RSA dec. with $\|n\| = 1024$) (ms)	Hash operation (SHA-1) (μs)	Other lightweight operations (e.g., XOR and Concatenation) (μs)
Intel E5500 2.80	7.216	0.753	0.010
Intel i3-530 2.93	6.359	0.693	0.009
Intel i5-3210M 2.50	4.390	1.132	0.008

What's more, the scheme is not scalable, as there is no method provided for adding new nodes in the existing network. If some sensor nodes or master nodes (i.e., cluster heads) are captured by an attacker, or some nodes expire as a result of the dissipation of energy, it is often required to add some new nodes in order to replace those ineffective nodes.

4. Review of A.K. Das et al.'s scheme

In this section, we briefly review the dynamic password-based user authentication scheme for HWSN proposed by A.K. Das et al. [5] in 2012. This protocol only involves three participants, i.e., the user (U_i), the base station (BS) and the cluster head (CH_j). It should be noted that the base station BS is not only responsible for the registration but also involved in the authentication process of U_i and CH_j. There are seven phases in this scheme: pre-deployment, post-deployment, registration, login, authentication, password change and dynamic node addition. In the following, we employ the notations listed in Table 1 and some additional notations illustrated in Table 3, and will follow the original notations in [5] as closely as possible. As we shall see, the descriptions of the scheme are rather tedious, but we manage to go through the jungle of the protocol specifications and to identify two serious security flaws.

4.1. Pre-deployment phase

The setup server (i.e., the base station) performs offline the following steps before the deployment of the sensor nodes and cluster heads in a target field:

(1) The setup server assigns a unique identifier, say ID_{CH_j}, to each cluster head CH_j. For each deployed regular sensor node S_i, the setup server also assigns a unique identifier, say ID_{S_i}.

(2) The setup server then selects randomly a unique master key, say MK_{CH_j}, for each cluster head CH_j. Note that the master key is shared between the cluster head CH_j and the base station only. Similarly, the setup server also assigns a unique randomly generated master key, say MK_{S_i} for each deployed regular sensor node S_i, which will be shared with the base station only.

(3) Finally, the setup server loads the following information into the memory of each cluster head CH_j its own identifier ID_{CH_j} and its own master key MK_{CH_j}. Each deployed regular sensor node S_i in the cluster C_j is loaded with its own identifier ID_{S_i} and its own master key MK_{S_i}.

Table 3
Additional notations in A.K. Das et al.'s scheme.

Symbol	Description
CH_j	Cluster head in the jth cluster
ID_{CH_j}	Identity of cluster head CH_j
$E(\cdot)/D(\cdot)$	Symmetric key encryption/decryption algorithm
X_s	Master secret key maintained by BS
X_A	Secret key shared between the user and BS
y	A random number chosen by the user

4.2. Post-deployment phase

As the concern is user authentication, it is simply assumed that the deployed nodes in a cluster can establish secret keys using some existing efficient and secure key establishment techniques. After key establishment, sensor nodes can securely communicate with other neighboring sensor nodes and their cluster head in the cluster, and cluster heads can also securely communicate with other neighboring cluster heads and finally to the base station.

4.3. Registration phase

This phase involves two participants: the user U_i and the base station BS. When U_i wants to resister with BS, the following operations will be performed:

(1) The user U_i selects a random number y, the identifier ID_i and the password PW_i. U_i then computes $RPW_i = h(y \| PW_i)$.

(2) $U_i \Rightarrow BS : \{ID_i, RPW_i\}$.[1]

(3) BS computes $f_i = h(ID_i \| X_s)$, $x = h(RPW_i \| X_A)$, $r_i = h(y \| x)$, and $e_i = f_i \oplus x = h(ID_i \| X_s) \oplus h(RPW_i \| X_A)$.

(4) BS then selects all m deployed cluster heads in the network, denoted by $CH_1, CH_2, \ldots, CH_m$, which have been deployed during the initial deployment phase, and computes the m key-plus-id combinations $\{(K_j, ID_{CH_j}) | 1 \leqslant j \leqslant m\}$, where $K_j = E_{MK_{CH_j}}(ID_i \| ID_{CH_j} \| X_s)$.

(5) For dynamic cluster head addition phase, assume that another m' cluster heads, denoted by CH_{m+1}, $CH_{m+2}, \ldots, CH_{m+m'}$, will be deployed later after the initial deployment in the network to replace some compromised cluster heads. For this purpose, the BS computes another m' key-plus-id combinations $\{(K_{m+j},\ ID_{CH_{m+j}}) | 1 \leqslant j \leqslant m'\}$, where $K_{m+j} = E_{MK_{CH_{m+j}}}(ID_i \| ID_{CH_{m+j}} \| X_s)$. Note that $ID_{CH_{m+j}}$ is the unique identifier generated by the BS for the cluster head CH_{m+j} to be deployed during the dynamic node addition phase, and $MK_{CH_{m+j}}$ is CH_{m+j}'s unique master key and is only shared with the BS.

(6) Finally, BS generates a non-tamper-proof smart card with the following parameters $\{ID_i, y, X_A, r_i, e_i, h(\cdot), [(K_j, ID_{CH_j}) | 1 \leqslant j \leqslant m + m']\}$.
The value of $m + m'$ is chosen according to the memory capacity of the smart card and more discussions can be found in the original paper [5].

Remark 3. The original specification of the above Step 6 in [5] states that "Finally, the BS generates a tamper-proof smart card with the following parameters ...". However, when evaluating the security of their protocol, A.K. Das et al. explicitly assumed that the smart card can be breached and secret data stored in its memory can be revealed. On the other hand, it is more desirable to base a scheme on a non-tamper-proof smart card in practice, for fully tamper-proof smart cards are costly and even may be unattainable. Hence, we deduce that a typo has occurred here.

[1] As we will show later in Section 5.1, there is a design flaw here, actually, the random number y should be sent along with ID_i and RPW_i.

4.4. Login phase

If the user U_i wants to access real-time data from the HWSN, the user U_i needs to perform the following steps:

(1) U_i inserts her smart card into the card reader, and keys her password PW_i.
(2) The smart card then computes $RPW'_i = h(y\|PW_i)$, $x' = h(RPW'_i\|X_A)$ and $r'_i = h(y\|x')$, and then verifies whether r'_i equals the stored r_i. If this equality holds, it implies that U_i has entered her password correctly; otherwise, the smart card rejects. Then, the smart card computes $N_i = h(x'\|T_1)$, where T_1 is the timestamp on the user side
(3) The user U_i selects a cluster head, say CH_j, from which the real-time data can be accessed inside *WSN*. The smart card selects the encrypted master key corresponding to CH_j, i.e. K_j, from its memory and computes $E_{ij} = E_{k_j}(ID_i\|ID_{CH_j}\|N_i\|e_i\|T_1)$.
(4) $U_i \rightarrow BS : \{ID_i, ID_{CH_j}, E_{ij}\}$.

4.5. Authentication phase

After receiving the login request message from U_i, BS performs the following operations:

(1) The base station BS computes $K'_j = E_{MK_{CH_j}}(ID_i\|ID_{CH_j}\|X_s)$, and obtains $(ID_i\|ID_{CH_j}\|N_i\|e_i\|T_1)$ by decrypting E_{ij} using MK_{CH_j}, where MK_{CH_j} is shared between BS and CH_j.
(2) BS checks if the retrieved ID_i is equal to the received ID_i and also if retrieved ID_{CH_j} is equal to received ID_{CH_j}. If the verification holds, BS further checks if $|T_1 - T_1^*| < \Delta T_1$, where T_1^* is the current timestamp of BS and ΔT_1 is the expected time interval for the transmission delay. If T_1 is valid, BS further computes $X = h(ID_i\|X_s)$, $Y = e_i \oplus X$, and $Z = h(Y\|T_1)$. If $Z = N_i$, then BS considers U_i as a valid user and accepts U_i's login request. Otherwise, the session is terminated.
(3) BS computes $u = h(Y\|T_2)$ and $E_{sj} = E_{MK_{CH_j}}(ID_i\|ID_{CH_j}\|u\|T_1\|T_2\|X\|e_i)$, where T_2 is the current timestamp on BS side.
(4) $BS \rightarrow CH_j : \{ID_i, ID_{CH_j}, E_{sj}\}$.
(5) Upon receiving the request from BS, the cluster head CH_j obtains $(ID_i\|ID_{CH_j}\|u\|T_1\|T_2\|X\|e_i)$ by decrypting E_{sj} using its own master key MK_{CH_j}. CH_j then checks the validity of ID_i, ID_{CH_j} and T_2. If any check fails, the session is terminated.
(6) CH_j computes $v = e_i \oplus X = h(RPW_i\|X_A)$ and $w = h(v\|T_2) = h(h(RPW_i\|X_A\|T_2))$, and checks whether w equals the retrieved u. If the equality holds, U_i is authenticated by CH_j. Otherwise, the session is terminated. CH_j computes the session key, which is shared with U_i, as $SK = h(ID_i\|ID_{CH_j}\|e_i\|T_1)$.
(7) $CH_j \rightarrow BS : \{ack\}$.
(8) $BS \rightarrow U_i : \{ack\}$.
(9) On receiving the acknowledgment from CH_j, U_i computes $SK = h(ID_i\|ID_{CH_j}\|e_i\|T_1)$.
From now on, U_i and CH_j can use the established session key SK for securing future data communications.

4.6. Password change phase

In this phase, the user U_i can change her password locally without the interaction with the base station BS.

(1) U_i inputs her smart card into the card reader, and provides her identifier ID_i, old password PW_i^{old} as well as the new password PW_i^{new}.
(2) The smart card first computes $RPW_i^* = h\left(y\|PW_i^{old}\right)$, $M_1 = h\left(RPW_i^*\|X_A\right)$ and $M_2 = h(y\|M_1)$, and then compares the computed M_2 with the stored r_i. If they do not equal, this means that the user U_i has entered her old password PW_i^{old} incorrectly and hence, the password change phase terminates immediately.
(3) The smart card computes $M_3 = e_i \oplus M_1 = h(ID_i\|X_s)$, $M_4 = h(y\|PW_i^{new})$, $r'_i = h(y\|M_4)$, $M_5 = h(M_4\|X_A)$ and $e'_i = M_3 \oplus M_5 = h(ID_i\|X_s) \oplus h(h(y\|PW_i^{new})\|X_A)$, and replaces r_i and e_i with r'_i and e'_i, respectively.

4.7. Dynamic node addition phase

Since the dynamic node addition phase has little relevance with our discussions, it is omitted here.

5. Cryptanalysis of A.K. Das et al.'s scheme

Due to its efficiency and provision of various admired features such as dynamic node addition and local password change, A.K. Das et al.'s scheme exhibits great application prospects, and yet it has some security flaws being overlooked as with Fan et al.'s scheme. The four assumptions listed in Section 3 are also explicitly made in A.K. Das et al.'s paper when they analyze the security of their scheme, and thus our following cryptanalysis is also based on these four assumptions. Accordingly, we assume that an adversary can extract the secret parameters $\{ID_i, y, X_A, r_i, e_i, h(\cdot), [(K_j, ID_{CH_j})|1 \leqslant j \leqslant m + m']\}$ stored in the legitimate user's smart card, and could also intercept or block the exchanged messages $\{ID_i, ID_{CH_j}, E_{ij}, E_{sj}\}$ during the login and authentication processes. A.K. Das et al. argued that their scheme can be free from privileged-insider attack and smart card security breach attack, however, the following attacks evidently invalidate their claims.

5.1. Privileged-insider attack

In many scenarios, the user tends to use a common password to access several systems for his convenience [34,54]. If the user registers to the BS with password in plain-text or in some other forms that could be easily derived by an insider of BS, the malicious insider can impersonate the legitimate user to login by abusing the legitimate user's password and thus get access to other systems [55,56]. Let's see how A.K. Das et al.'s scheme is susceptible to this threat.

First of all, we have to point out that there is a design flaw in the registration phase. In Step 2 of the registration phase, U_i only sends $\{ID_i, RPW_i\}$ to BS, whereas BS needs to compute $r_i = h(y\|x)$ in Step 3 of the registration phase. As y is randomly chosen by U_i, how can BS obtain the value of y

to compute r_i? Clearly, the parameter y should be sent along with $\{ID_i, RPW_i\}$ to BS in Step 2 of the registration phase.

Unfortunately, if y is sent along with $\{ID_i, RPW_i\}$ in Step 2 of the registration phase, the privileged-insider attack can be performed by $\mathcal{A}$ as follows:

Step 1. Guesses the value of PW_i to be PW_i^* from a dictionary space $\mathcal{D}_{pw}$.
Step 2. Computes $RPW_i^* = h(y\|PW_i^*)$.
Step 3. Verifies the correctness of PW_i^* by checking if the computed RPW_i^* is equal to the received RPW_i.
Step 4. Repeats the above steps until the correct value of PW_i is found.

Let $|\mathcal{D}_{pw}|$ denote the number of passwords in the password space $\mathcal{D}_{pw}$. The running time of the above attack procedure is $\mathcal{O}(|\mathcal{D}_{pw}| * T_H)$, where T_H is the running time for Hash operation. Since passwords are human-memorable short strings but not high-entropy keys, in other words, they are often drawn from a dictionary with small size, e.g. $|\mathcal{D}_{pw}| = 10^6$ [35,36]. As $|\mathcal{D}_{pw}|$ is very limited in practice, the above attack can be completed in polynomial time. As a result, the scheme is vulnerable to privileged-insider attack.

5.2. Smart card security breach attack

Suppose a legitimate user U_i's smart card is somehow obtained by $\mathcal{A}$, and all the secret parameters $\{ID_i, y, X_A, r_i, e_i, h(\cdot), [(K_j, ID_{CH_j})|1 \leqslant j \leqslant m + m']\}$ that are stored in the card can be extracted. In this case, A.K. Das et al.'s scheme is completely insecure.

Firstly, let us show that U_i's password PW_i can be offline guessed as follows:

Step 1. Guesses the value of PW_i to be PW_i^* from a dictionary space $\mathcal{D}_{pw}$.
Step 2. Computes $RPW_i^* = h(y\|PW_i^*)$ and $r_i^* = h(y\|h(RPW_i^*\|X_A))$, where y and x_A is extracted from U_i's smart card.
Step 3. Verifies the correctness of PW_i^* by checking if the computed r_i^* is equal to the revealed r_i.
Step 4. Repeats the above steps until the correct value of PW_i is found.

Let $|\mathcal{D}_{pw}|$ denote the number of passwords in the password space $\mathcal{D}_{pw}$. The running time of the above attack procedure is $\mathcal{O}(|\mathcal{D}_{pw}| * 3T_H)$, where T_H is the running time for Hash operation. According to Assumption iv, the user-chosen passwords tend to be weak passwords, in other words, they are often chosen from a dictionary of small size, e.g. $|\mathcal{D}_{pw}| = 10^6$ [35,36]. As $|\mathcal{D}_{pw}|$ is very limited in practice, the above attack can be completed in polynomial time. As a result, the scheme is vulnerable to offline password guessing attack.

Secondly, let us show that any of U_i's previous session keys, say SK_m, which is established during U_i's mth authentication process, can be computed by $\mathcal{A}$ as follows:

Step 1. Decrypts E_{ij}^m using K_j to obtain e_i and T_1^m, where E_{ij}^m is intercepted from U_i's mth authentication process and K_j is revealed from U_i's smart card.
Step 2. Computes $SK_m = h(ID_i\|ID_{CH_j}\|e_i\|T_1^m)$, where ID_i and ID_{CH_j} is intercepted from U_i's mth authentication process.

Once SK_m is obtained, the whole mth session will be exposed. In general, it is evident that all the previous sessions participated by U_i can be compromised.

The above analysis reveals that, once the secret data in the card is extracted, U_i's password and all the previous session keys established by U_i can be obtained by $\mathcal{A}$. In other words, our analysis demonstrates the feasibility of smart card security breach attack on A.K. Das et al.'s scheme under their non-tamper resistance assumption of the smart card, thereby contradicting the claim made in [5].

5.3. Server master key disclosure

As X_s is the master secret key of the base station BS, what will happen if X_s is revealed to other parties than BS? Obviously, the security of the entire system will collapse. In the following, we show that a malicious (but legitimate) user can successfully obtain X_s by just compromise only one cluster head. Assume that a malicious user U_m has compromised a cluster head CH_j. U_m can recover X_s as follows:

Step 1. Extracts the master key MK_{CH_j} stored on CH_j.
Step 2. Extracts $K_j = E_{MK_{CH_j}}(ID_i\|ID_{CH_j}\|X_s)$ from U_m's own smart card.
Step 3. Recovers $ID_i\|ID_{CH_j}\|X_s$ by decrypting K_j using MK_{CH_j}.

The above attack illustrates that, under the Assumptions i and ii (see Section 3), BS's master secret key X_s can be easily disclosed, then there is no security left at all. This is really a serious threat, considering the practicality of the Assumptions i and ii. Fortunately, we find this issue can be well addressed by modifying how K_j is computed. One can see that, instead of setting $K_j = E_{MK_{CH_j}}(ID_i\|ID_{CH_j}\|X_s)$, if we let $K_j = E_{MK_{CH_j}}(\boldsymbol{h}(ID_i\|ID_{CH_j}\|X_s))$ and other parts of the protocol remain unchanged, the presented attack can be thwarted, while all the other security provisions of the original protocol will be unaffected.

6. Some suggestions for designing more practical schemes

Since smart-card-based password authentication can provide strong two-factor authentication [14], it is well-suited to security-critical applications in WSNs. A number of this type of schemes have been proposed, and some quite recent ones include [4,6,7,15–17,23,24,57,58]. However, according to [6,8,9,11] and our cryptanalysis results [25], most of these proposals have been found prone to various practical attacks. Previous research has shown that, it is really not an easy task to design a practical authentication scheme for WSNs, for the protocol designers are faced with the difficult task of reconciling efficiency, security and functionality requirements [4,59], and often must make design decisions that are seemingly well motivated but may have unintended consequences.

Generally, there are three major issues challenging the design. Firstly, efficiency needs to be taken into account. Sensor nodes and user-held mobile devices (e.g., Laptops, mobile phones, PDAs and smart cards) are typically resource-constrained in terms of storage capacity and processing power. As a result, an authentication scheme should be computationally efficient. Secondly, security and privacy are serious concerns for the authentication service. Owing to the hostile environments where WSNs are usually deployed, nodes can be physically captured by an attacker. What's worse, the participants in the scheme communicate with each other using wireless technologies, such as 802.15.4 and 802.11g, while wireless networking's broadcast nature makes the transmitted messages readily available to anyone. In particular to HWSN, collusion attacks pose a serious threat to security, because cluster heads (or master nodes) and external users are not trusted parties and they could collude to commit malicious actions. Thirdly, a practical scheme shall support various basic functionalities, such as mutual authentication, key agreement and local user password change. Besides these basic properties, an admired scheme may be expected to provide more advanced features like dynamic node addition, insider resistance and user anonymity.

Understanding security failures of cryptographic protocols is vital for designing more robust schemes, for lessons learned from the past often pave the way for the future. There have been a number of papers [6–9,11,12,60,61] dealing with security flaws in two-factor authentication schemes for WSNs, but until now, to the best of our knowledge, little attention has been paid to the underlying rationale of the identified security failures. Consequently, quite similar (sometimes even the same) mistakes are repeated time and time again, resulting in the unpleasant but unending "break-fix-break-fix" cycle (see Fig.2). For example, the scheme proposed by M.L. Das [58] in 2009 was found vulnerable to privileged insider attack by Khan and Alghathbar [7] in 2010. Unfortunately, precisely the same defect still exists in its several improved versions [7,17,24,57], while the authors of [7,17,24,57] are clearly aware of this vulnerability in M.L. Das's scheme. For another example, Kumar et al. [61] pointed out that Yoo et al.'s scheme [23] cannot preserve user anonymity and developed an enhanced version. However, Kumar et al.'s scheme uses the same cryptographic primitives (i.e., symmetric encryption and Hash) with Yoo et al.'s scheme. As we will show later, both these two schemes are intrinsically unable to preserver user anonymity due to the same reason.

As far as we know, the work in [25] is the only one that presents a principle for designing more robust two-factor authentication schemes for WSNs: the public-key techniques are indispensable to resist smart card security breach attack (offline password guessing attack) if the smart cards are assumed to be non-tamper resistant. To further ameliorate the above situation, through the security analysis of Fan et al.'s and A.K. Das's schemes and based upon our past cryptanalysis experience (some of our recent results include [25,32,47,62,63]), we put forward three suggestions that are helpful to construct more practical two-factor schemes for WSNs in the following.

6.1. Employ public-key techniques to preserve user anonymity

First of all, there are some comments needed to be made about the notion of user anonymity in remote user authentication schemes. Briefly speaking, user anonymity includes a basic property called "initiator anonymity" and an advanced property called "sender untraceability" [62]. The former means that the adversary cannot determine the real identity of the initiator (i.e., the user), while the latter says that the adversary should be not only infeasible to acquire any knowledge of the real identity of the

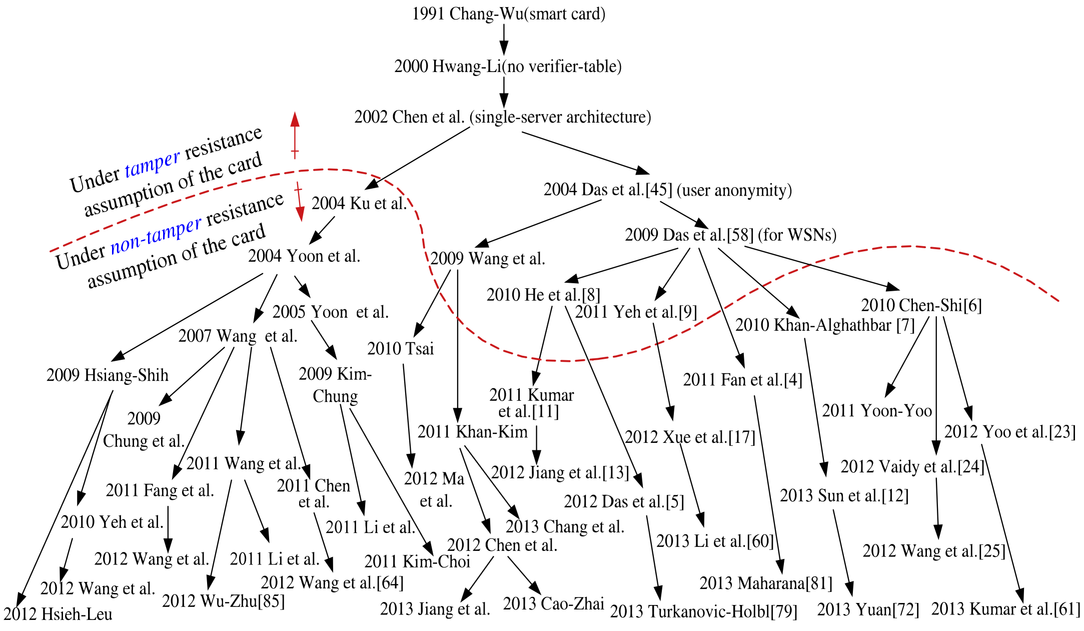

Fig. 2. A brief history of two-factor authentication for WSNs.

user but also cannot determine whether two conversations originate from the same (unknown) user.

We have analyzed more than one hundred recently proposed smart-card-based password authentication schemes: more than ninety schemes for general applications (including both the single-server architecture like [32,47,62,64] and the multi-server architecture like [65]), more than thirty schemes for WSNs (e.g., [25]) and more than twenty schemes for mobile wireless networks (e.g., [22]). And we find these schemes (no matter for general purpose like [66–68] or for WSNs like [5,8,13,15–17,24,60,61,69]), which do not employ public-key techniques but claim to preserve user anonymity, definitely vulnerable to user anonymity violation attack under the non-tamper resistance assumption of the smart cards. Accordingly, we come to the conjecture that, schemes based only on symmetric cryptographic primitives (e.g., hash functions, XOR operations, symmetric encryption) are intrinsically unable to preserve user anonymity.

As we have observed, there are two factors that may contribute to this failure. Firstly, for the sake of user-friendliness, a user is usually allowed to select her own identity ID at will (sometimes confined to a predefined format) during the registration phase; the user tends to choose an identity which is easily remembered for her convenience. In other words, user identities are human-memorable and as weak as passwords. Now, it is not difficult to see: there is a possibility that user's identity can also be offline guessed by $\mathcal{A}$ using the same method as guessing user's password. And this potential has been confirmed by our user anonymity violation attack [25] on Xue et al.'s dynamic ID-based scheme. Secondly, the threats from malicious collusion often cause the scheme to fail to achieve the property of "sender untraceability" [65,70]. A malicious external user having her own smart card can gather and compute some critical parameters, e.g. $H(x)$ in Section 3.1. Then, if the dishonest authentication server colludes with this malicious user (e.g., MN_m colludes with U_m in Section 3.1 and S_k colludes with U_m in Section 5.2 of [65]), some information specific to the victim user that logins to the dishonest server may be computed. In this way, "sender untraceability" is breached.

In [25], the authors managed to prove that, under the non-tamper resistance assumption of the smart cards, all the smart-card-based password authentication schemes for WSNs that only employ symmetric cryptographic primitives are vulnerable to offline password guessing attack (smart card security breach attack) with a probability of $\boldsymbol{P} \neq \boldsymbol{NP}$. And now the countermeasure is obvious: resorting to public-key techniques like [43,64,71,72], in which ECC or RSA public key technique is employed, while the studies in [73–76] have well established the feasibility and acceptability of performing a few public-key operations (e.g., scalar multiplication, Tate pairing and RSA encryption) on resource-constrained sensor nodes and smart cards. If the heterogeneities (e.g., computational heterogeneity and energy heterogeneity) of HWSN and the computational imbalance of public-key operations (e.g, RSA encryption is typically more lightweight than RSA encryption [77]) are further exploited, it is promising to see that secure, efficient and privacy-preserving two-factor schemes can be built by use of the existing public-key techniques. In addition, there might be some relationships between this proved assertion and the above conjecture, however, we promote the proof of our conjecture as one open problem.

6.2. Adopt a fuzzy-verifier to balance usability and security

In practice, it is a widely recommended security policy for security-critical applications that a user should update her password regularly. To provide better user-friendliness, many schemes support the feature of user local password update [78], which means the users can update their passwords locally without the hassle of interacting with the authentication server (i.e., the base station in [5] or the master node in [4]). To provide secure password update, a verification of the old password should be performed before the new password update occurs. Accordingly, to support securely local password update, there should be a password-related verifier stored in the smart card. This just well explains why the parameter $r_i = h(y\|h(h(y\|PW_i)\|X_A))$ is stored in the smart card in A.K. Das et al.'s scheme [5].

It is important to notice that, A.K. Das et al.'s scheme stores password-related verifiers $\{r_i, y, X_A\}$ in a form that is plaintext-equivalent to the password, and an adversary $\mathcal{A}$ can just exploit r_i to confirm whether her guessed password is correct or not. This well explicates the reason for the success of the smart card security breach attack introduced in Section 5.2. Now, it is trivial to see that, all these newly proposed schemes [5,24,60,61,67,68,79–81] that store a password-related verifier similar to r_i in the smart card to serve as the comparing target when updating the old password are definitely vulnerable to offline password guessing attack, once the smart card security is breached.

As far as we know, under the non-tamper resistance assumption of the smart cards, until now there has been no two-factor authentication scheme for WSNs that can support securely local password change while be free from offline password guessing attack. In [63,82], the authors observed that there is an avoidable trade-off between the security requirement of resistance to smart security breach attack and the usability goal of local password change. Fortunately, they further introduced a notion of "fuzzy verifier", which can well balance the aforementioned security requirement and usability goal.

To gain more insights into this issue, we take Fan et al.'s scheme [4] as another example. Suppose a password verifier is also stored in U_i's smart card (actually, there already exists such a verifier in Fan et al.'s original scheme, i.e., $V_i = h^3(ID_i\|PW_i)$). Now, whenever U_i wants to update her password, she first keys ID_i and the original password PW_i, the smart card then validates the correctness of PW_i by checking $V_i \overset{?}{=} h^3(ID_i\|PW_i)$. If the check fails, the password change request is denied. Unfortunately, now it is not difficult to see that, once V_i is revealed by $\mathcal{A}$, an smart card security breach attack (as demonstrated in Section 3.2) will be successfully launched by exhaustively checking whether $V_i \overset{?}{=} h^3(ID_i^*\|PW_i^*)$, where ID_i^* and PW_i^* are the guessed identity and password. Now if we adopt the "fuzzy verifier" technique and compute the parameter $V_i = h(h(h(ID_i)\|h(PW_i)) \bmod n)$, where n is a medium number (e.g., $n = 2^{10}$), one can be assured that there exists $\frac{|\mathcal{D}_{id}|*|\mathcal{D}_{pw}|}{n} \approx 2^{30}$ candidates of (ID, PW) pair to frustrate $\mathcal{A}$ when $|\mathcal{D}_{id}| = |\mathcal{D}_{pw}| = 10^6$ [36,35] and $n = 2^{10}$, where $|\mathcal{D}_{id}|$ and $|\mathcal{D}_{pw}|$ denote the size of the identity space and password space, respectively. Even with the victim user's identity ID_i learnt, $\mathcal{A}$ will still be prevented from determining the exactly correct password PW_i, because there exists $\frac{|\mathcal{D}_{pw}|}{n} \approx 2^{10}$ password candidates. To further eliminate the specious passwords from the remaining password candidates, there is no other way than launching an online password guessing attack by interacting with the authentication server, which can be effectively thwarted by non-cryptographic techniques [83], e.g., by blocking the victim's account for a period of time after a few consecutive failed attempts [84,85]. In this manner, we frustrate $\mathcal{A}$'s attempt to obtain the correct password. We refer the readers to [82] for more details.

At the mean time, a new question arises: what will happen if a user happens to submit a wrong $(\widetilde{ID_i}, \widetilde{PW_i})$ pair in the password change process such that $h(h(h(\widetilde{ID_i})\|h(\widetilde{PW_i}))$ mod $n) = V_i$, while $\widetilde{PW_i} \neq PW_i$? Actually, this will rarely occur in reality for two reasons. For simplicity, we assume $n = 2^{10}$. First, it is not difficult to determine that the probability of this kind of misfortune is less than $\frac{1}{1000}$ whenever a user updates her password, under the assumption that the outputs of hash function $h(\cdot)$ can be treated as random integers with fixed length. Here the integer n serves as the usability-security tradeoff factor, and the treatment of hash functions as random functions has been a widely accepted method in modern cryptography research, called the random oracle model, first introduced by Bellare and Rogaway in [86]. Second, in practice, both for usability and security, a user's password should only be changed regularly but not frequently, e.g. every ninety days [87], which means such a misfortune will only happen every 256 $\left(= 2^{10} \ * \frac{1}{4}\right)$ years. It should be noted that, even if such a misfortune has occurred, the user can restore the function of the card by re-registering with BS.

6.3. Add "salts" to resist privileged insider attack

In the registration phase of M.L. Das's scheme [58], the user just submits her plain-text password to the registration server (i.e., the gateway node), and therefore this scheme is obviously vulnerable to privileged insider attack. Be aware of this defect, Khan and Alghathbar [7] adopt another strategy: the user does not present her plain-text password PW_i but its hashed value $h(PW_i)$ to the registration server (i.e., the gateway node). And some subsequent schemes, e.g. [17,24,72], also adopt similar approaches. One can easily see that, on receiving $h(PW_i)$, a malicious insider of the registration server can launch a similar attack to the one introduced in Section 5.1 to offline guess the correct password PW_i without much difficulty.

However, if some "salts" are added to the submitted password-verifier in the registration phase, this threat can be thwarted thoroughly. The following is a recommended practice: in the registration phase, besides the identity ID_i and the password PW_i, U_i chooses a large random number r (serving as the "salts") and computes $RPW = h(PW_i \oplus r)$, and then sends RPW instead of PW_i or $h(PW_i)$ to the registration server. Upon receiving the smart card, U_i keys r into the smart card, and she does not need to memorize r. This approach has been employed in several two-factor authentication schemes [43,61,82].

7. Conclusion

Owing to the broadcast nature of wireless communications, extremely large network size, severe resource constraints and the lack of the infrastructure support, how to design secure and efficient remote user authentication schemes for real-time data access directly from sensor nodes in HWSN still remains a quite challenging problem, though several schemes have been proposed lately. In this paper, we have analyzed two efficient two-factor authentication schemes for HWSN without employing public-key techniques. The two schemes are equipped with a claimed proof of security, however, we pointed out that both protocols have various security flaws being overlooked. There have been several works on the security analysis of two-factor authentication schemes for HWSN, yet little (or even no) rationale is revealed and not surprisingly, similar mistakes are repeated over and over again. To mitigate this situation, through the cryptanalysis of two quite recent schemes for HWSN, i.e. Fan et al.'s and A.K. Das et al.' schemes, we put forward three suggestions that are helpful to explicate many of the security failures repeated in the past and vital for designing more robust schemes for HWSN in the future.

The work we report here is merely a first step towards investigating the underlying rationales of the security failures that have continue to occur in a great proportion of recent two-factor schemes for HWSN. Even for the list of suggestions, we have just considered the most urgent ones, leaving many security requirements (e.g., resistance to denial of service attack and *GWN* by-passing attack) unstudied. From a microscopic point of view, for example, we have conjectured that public-key techniques are indispensable to resist against user anonymity violation attack under the non-tamper resistance assumption of the smart cards, and an interesting and meaningful question might be whether one can find/construct a counter-example to this conjecture. Or if not, providing a proof of this conjecture suggests an important direction for future study. We believe our work provides a better understanding of the security challenges in designing two-factor user authentication schemes for HWSN, and may constitute a useful guidance to promote further development of more practical schemes.

Acknowledgments

The authors would like to thank the anonymous reviewers for their constructive suggestions and comments that highly improve the readability and completeness of the paper. This research was partially supported by the National Basic Research Program of China (973) under Grant No. 2009CB320703, and the National Natural Science Foundation of China (NSFC) under Grant No. 61170263.

References

[1] Y. Cheng, D. Agrawal, An improved key distribution mechanism for large-scale hierarchical wireless sensor networks, Ad Hoc Networks 5 (1) (2007) 35–48.

[2] Y. Zhao, Y. Zhang, Z. Qin, T. Znati, A co-commitment based secure data collection scheme for tiered wireless sensor networks, J. Syst. Archit. 57 (6) (2011) 655–662.

[3] G. Maia, D. Guidoni, A. Viana, A. Aquino, R. Mini, A. Loureiro, A distributed data storage protocol for heterogeneous wireless sensor networks with mobile sinks, Ad Hoc Networks 11 (5) (2013) 1588–1602.

[4] R. Fan, D. He, X. Pan, L. Ping, An efficient and dos-resistant user authentication scheme for two-tiered wireless sensor networks, J. Zhejiang Univ. – Sci. C 12 (7) (2011) 550–560.

[5] A.K. Das, P. Sharma, S. Chatterjee, J.K. Sing, A dynamic password-based user authentication scheme for hierarchical wireless sensor networks, J. Network Comput. Appl. 35 (52) (2012) 1646–1656.

[6] T.H. Chen, W.K. Shih, A robust mutual authentication protocol for wireless sensor networks, ETRI J. 32 (5) (2010) 704–712.

[7] M.K. Khan, K. Alghathbar, Cryptanalysis and security improvements of two-factor user authentication in wireless sensor networks, Sensors 10 (3) (2010) 2450–2459.

[8] D. He, Y. Gao, S. Chan, C. Chen, J. Bu, An enhanced two-factor user authentication scheme in wireless sensor networks, Ad Hoc Sensor Wireless Networks 10 (4) (2010) 361–371.

[9] H. Yeh, T. Chen, P. Liu, T. Kim, H. Wei, A secured authentication protocol for wireless sensor networks using elliptic curves cryptography, Sensors 11 (5) (2011) 4767–4779.

[10] C.-C. Lee, C.-T. Li, S. der Chen, Two attacks on a two-factor user authentication in wireless sensor networks, Parallel Process. Lett. 21 (1) (2011) 21–26.

[11] P. Kumar, H. Lee, Cryptanalysis on two user authentication protocols using smart card for wireless sensor networks, in: Wireless Advanced, IEEE, 2011, pp. 241–245.

[12] D. Sun, J. Li, Z. Feng, Z. Cao, G. Xu, On the security and improvement of a two-factor user authentication scheme in wireless sensor networks, Pers. Ubiquitous Comput. 17 (5) (2013) 895–905.

[13] Q. Jiang, Z. Ma, J.F. Ma, G. Li, Security enhancement of robust user authentication framework for wireless sensor networks, China Commun. 9 (10) (2012) 103–111.

[14] G.M. Yang, D.S. Wong, H.X. Wang, X.T. Deng, Two-factor mutual authentication based on smart cards and passwords, J. Comput. Syst. Sci. 74 (7) (2008) 1160–1172.

[15] P. Kumar, A.J. Choudhury, M. Sain, S.M. Lee, H.J. Lee, Ruasn: A robust user authentication framework for wireless sensor networks, Sensors 11 (5) (2011) 5020–5046.

[16] P. Kumar, S.-G. Lee, H.-J. Lee, E-sap: Efficient-strong authentication protocol for healthcare applications using wireless medical sensor networks, Sensors 12 (2) (2012) 1625–1647.

[17] K. Xue, C. Ma, P. Hong, R. Ding, A temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks, J. Network Comput. Appl. 36 (1) (2013) 316–323.

[18] J. Shi, R. Zhang, Y. Zhang, Secure range queries in tiered sensor networks, in: Proceedings of 28th IEEE Conference on Computer Communications (INFOCOM 2009), IEEE, 2009, pp. 945–953.

[19] M. Chatterjee, S.K. Das, D. Turgut, WCA: A weighted clustering algorithm for mobile ad hoc networks, Cluster Comput. 5 (2) (2002) 193–204.

[20] J. Ai, D. Turgut, L. Boloni, A cluster-based energy balancing scheme in heterogeneous wireless sensor networks, in: P. Lorenz, P. Dini (Eds.), International Conference on Networking (ICN 2005), LNCS, vol. 3420, Springer, Berlin Heidelberg, 2005, pp. 467–474.

[21] A. Bari, S. Wazed, A. Jaekel, S. Bandyopadhyay, A genetic algorithm based approach for energy efficient routing in two-tiered sensor networks, Ad Hoc Networks 7 (4) (2009) 665–676.

[22] D. Wang, P. Wang, J. Liu, Improved privacy-preserving authentication scheme for roaming service in mobile networks, in: Proceedings of 15th IEEE Wireless Communications and Networking Conference (WCNC 2014), 2014, pp. 1–6. <http://wangdingg.weebly.com/uploads/2/0/3/6/20366987/wcnc2014.pdf>.

[23] S.G. Yoo, K.Y. Park, J. Kim, A security-performance-balanced user authentication scheme for wireless sensor networks, Int. J. Distrib. Sensor Networks (2012), http://dx.doi.org/10.1155/2012/38281.

[24] B. Vaidya, D. Makrakis, H. Mouftah, Two-factor mutual authentication with key agreement in wireless sensor networks, Secur. Commun. Networks (2012), http://dx.doi.org/10.1002/sec.51.

[25] D. Wang, C.-G. Ma, On the (in)security of some Smart-card-based Password Authentication Schemes for WSN, Cryptology ePrint Archive, Report 2012/581, 2012. <http://eprint.iacr.org/2012/581.pdf>.

[26] T.S. Messerges, E.A. Dabbish, R.H. Sloan, Examining smart-card security under the threat of power analysis attacks, IEEE Trans. Comput. 51 (5) (2002) 541–552.

[27] S. Mangard, E. Oswald, T. Popp, Power Analysis Attacks: Revealing the Secrets of Smart Cards, Springer-Verlag, 2007.

[28] N. Veyrat-Charvillon, F.-X. Standaert, Generic side-channel distinguishers: improvements and limitations, in: P. Rogaway (Ed.), CRYPTO 2011, LNCS, vol. 6841, Springer, Berlin/Heidelberg, 2011, pp. 354–372.

[29] T.H. Kim, C. Kim, I. Park, Side channel analysis attacks using AM demodulation on commercial smart cards with SEED, J. Syst. Softw. 85 (12) (2012) 2899–2908.

[30] X. Leroy, Smart Card Security from a Programming Language and Static Analysis Perspective, INRIA Rocquencourt & trusted logic, Tecnical Report, 2013. <http://pauillac.inria.fr/xleroy/talks/language-security-etaps03.pdf>.

[31] K. Nohl, D. Evans, S. Starbug, H. Plötz, Reverse-engineering a cryptographic RFID tag, in: Proceedings of the 17th USENIX Security symposium (USENIX Security 2008), USENIX Association, 2008, pp. 185–193.

[32] D. Wang, P. Wang, Offline dictionary attack on password authentication schemes using smart cards, in: Y. Desmedt, B. Thuraisingham, K. Hamlen (Eds.), Proceedings of 16th Information Security Conference (ISC 2013), Lecture Notes in Computer Science, Springer-Verlag, 2013, pp. 1–16. <http://wangdingg.weebly.com/uploads/2/0/3/6/20366987/isc2013.pdf>.

[33] D. Dolev, A. Yao, On the security of public key protocols, IEEE Trans. Inform. Theory 29 (2) (1983) 198–208.

[34] D. Florencio, C. Herley, A large-scale study of web password habits, in: Proceedings of the 16th international conference on World Wide Web (WWW 2007), ACM, 2007, pp. 657–666.

[35] J. Bonneau, The science of guessing: analyzing an anonymized corpus of 70 million passwords, in: 33th IEEE Symposium on Security and Privacy (S&P 2012), IEEE Computer Society, 2012, pp. 538–552.

[36] M. Dell'Amico, P. Michiardi, Y. Roudier, Password strength: an empirical analysis, in: Proceedings of 29th IEEE Conference on Computer Communications (INFOCOM 2010). IEEE Communications Society, 2010, pp. 1–9.

[37] S. Komanduri, R. Shay, P.G. Kelley, M.L. Mazurek, Of passwords and people: measuring the effect of password-composition policies, in: Proceedings of the 22th ACM SIGCHI Conference on Human Factors in Computing Systems (CHI 2011), ACM, 2011, pp. 2595–2604.

[38] J. Campbell, W. Ma, D. Kleeman, Impact of restrictive composition policy on user password choices, Behav. Inform. Technol. 30 (3) (2011) 379–388.

[39] M. Weir, S. Aggarwal, M. Collins, H. Stern, Testing metrics for password creation policies by attacking large sets of revealed passwords, in: Proceedings of the 17th ACM Conference on Computer and Communications Security (CCS 2010), ACM, 2010, pp. 162–175.

[40] P.G. Kelley, S. Komanduri, M.L. Mazurek, Guess again (and again and again): measuring password strength by simulating password-cracking algorithms, in: 33th IEEE Symposium on Security and Privacy (S&P 2012), IEEE Computer Society, 2012, pp. 523–537.

[41] J. Bonneau, M. Just, G. Matthews, Whats in a name?, in: R. Sion (Ed.), Proceedings of the 14th International Conference on Financial Cryptography and Data Security (FC 2010), LNCS, vol. 6052, Springer, Berlin/Heidelberg, 2010, pp. 98–113.

[42] F. Bao, R. Deng, Privacy protection for transactions of digital goods, in: S. Qing, T. Okamoto, J. Zhou (Eds.), ICICS 2001, LNCS, vol. 2229, Springer, Berlin/Heidelberg, 2001, pp. 202–213.

[43] X. Li, W. Qiu, D. Zheng, K. Chen, J. Li, Anonymity enhancement on robust and efficient password-authenticated key agreement using smart cards, IEEE Trans. Indust. Electron. 57 (2) (2010) 793–800.

[44] C. Tang, D. Wu, Mobile privacy in wireless networks-revisited, IEEE Trans. Wireless Commun. 7 (3) (2008) 1035–1042.

[45] M.L. Das, A. Saxena, V.P. Gulati, A dynamic id-based remote user authentication scheme, IEEE Trans. Consumer Electron. 50 (2) (2004) 629–631.

[46] R. Madhusudhan, R.C. Mittal, Dynamic id-based remote user password authentication schemes using smart cards: a review, J. Network Comput. Appl. 35 (4) (2012) 1235–1248.

[47] D. Wang, C.G. Ma, P. Wu, Secure password-based remote user authentication scheme with non-tamper resistant smart cards, in: N. Cuppens-Boulahia, F. Cuppens, J. Garcia-Alfaro (Eds.), Proceedings of the 26th Annual IFIP Conference on Data and

Applications Security and Privacy (DBSec 2012), LNCS, vol. 7371, Springer, Berlin/Heidelberg, 2012, pp. 114–121.

[48] D. He, J. Bu, S. Zhu, S. Chan, C. Chen, Distributed access control with privacy support in wireless sensor networks, IEEE Trans. Wireless Commun. 10 (10) (2011) 3472–3481.

[49] Y. Yu, K. Li, W. Zhou, P. Li, Trust mechanisms in wireless sensor networks: attack analysis and countermeasures, J. Network Comput. Appl. 35 (3) (2012) 867–880.

[50] J. Duan, D. Gao, C. Foh, H. Zhang, Tc-bac: A trust and centrality degree based access control model in wireless sensor networks, Ad Hoc Networks 11 (8) (2013) 2675–2692.

[51] Miracl library, Shamus Software Ltd. <http://www.shamus.ie/index.php?page=home>.

[52] J. Katz, R. Ostrovsky, M. Yung, Efficient and secure authenticated key exchange using weak passwords, J. ACM 57 (1) (2009) 1–39.

[53] R. Shay, S. Komanduri, P.G. Kelley, Encountering stronger password requirements: user attitudes and behaviors, in: Proceedings of the Sixth Symposium on Usable Privacy and Security (SOUPS 2010), ACM, New York, NY, USA, 2010, pp. 1–10.

[54] S. Haque, M. Wright, A study of user password strategy for multiple accounts, in: Proceedings of the Third ACM Conference on Data and Application Security and Privacy (CODASPY 2013), ACM, 2013, pp. 173–176.

[55] T.H. Chen, Y.C. Chen, W.K. Shih, H.W. Wei, An efficient anonymous authentication protocol for mobile pay-tv, J. Network Comput. Appl. 34 (4) (2011) 1131–1137.

[56] D. He, S. Wu, J. Chen, Note on design of improved password authentication and update scheme based on elliptic curve cryptography, Math. Comput. Model. 55 (3) (2012) 1661–1664.

[57] B. Vaidya, D. Makrakis, H. Mouftah, Improved two-factor user authentication in wireless sensor networks, in: Proeedings of IEEE 6th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob 2010), IEEE, 2010, pp. 600–606.

[58] M.L. Das, Two-factor user authentication in wireless sensor networks, IEEE Trans. Wireless Commun. 8 (3) (2009) 1086–1090.

[59] S. Ravi, A. Raghunathan, P. Kocher, S. Hattangady, Security in embedded systems: design challenges, ACM Trans. Embed. Comput. Syst. 3 (3) (2004) 461–491.

[60] C.-T. Li, C.-Y. Weng, C.-C. Lee, An advanced temporal credential-based security scheme with mutual authentication and key agreement for WSNs, Sensors 13 (8) (2013) 9589–9603.

[61] P. Kumar, A. Gurtov, M. Ylianttila, S.-G. Lee, H. Lee, A strong authentication scheme with user privacy for wireless sensor networks, ETRI J. 35 (5) (2013) 889–899.

[62] D. Wang, C.G. Ma, Cryptanalysis of a remote user authentication scheme with provable security for mobile client-server environment based on ECC, Inform. Fusion 14 (4) (2013) 498–503.

[63] C.G. Ma, D. Wang, S.D. Zhao, Security flaws in two improved remote user authentication schemes using smart cards, Int. J. Commun. Syst. (2012), http://dx.doi.org/10.1002/dac.2468.

[64] D. Wang, C. Ma, Cryptanalysis and security enhancement of a remote user authentication scheme using smart cards, Elsevier J. China Univ. Posts Telecommun. 19 (5) (2012) 104–114.

[65] D. Wang, C.G. Ma, S. Zhao, C. Zhou, Cryptanalysis of two dynamic id-based remote user authentication schemes for multi-server architecture, in: L. Xu, E. Bertino, Y. Mu (Eds.), Proceedings of the 6th International Conference on Network and System Security (NSS 2012), LNCS, vol. 7645, Springer, Berlin/Heidelberg, 2012, pp. 462–475.

[66] M.K. Khan, S. Kim, K. Alghathbar, Cryptanalysis and security enhancement of a 'more efficient and secure dynamic id-based remote user authentication scheme', Comput. Commun. 34 (3) (2011) 305–309.

[67] S. Sood, A. Sarje, K. Singh, A secure dynamic identity based authentication protocol for multi-server architecture, J. Network Comput. Appl. 34 (2) (2011) 609–618.

[68] X. Li, Y. Xiong, J. Ma, W. Wang, An enhanced and security dynamic identity based authentication protocol for multi-server architecture using smart cards, J. Network Comput. Appl. 35 (2) (2012) 763–769.

[69] T.-C. Hsiao, Y.-T. Liao, J.-Y. Huang, T.-S. Chen, G.-B. Horng, An authentication scheme to healthcare security under wireless sensor networks, J. Med. Syst. 36 (6) (2012) 3649–3664.

[70] D.B. He, H. Hu, Cryptanalysis of a dynamic id-based remote user authentication scheme with access control for multi-server environments, IEICE Trans. Inform. Syst. 96 (1) (2013) 138–140.

[71] M.K. Khan, D.B. He, A new dynamic identity-based authentication protocol for multi-server environment using elliptic curve cryptography, Secur. Commun. Networks 5 (11) (2012) 1260–1266.

[72] J.-J. Yuan, An enhanced two-factor user authentication in wireless sensor networks, Telecommun. Syst. (2013), http://dx.doi.org/10.1007/s11235-013-9755-5.

[73] A.S. Wander, N. Gura, H. Eberle, V. Gupta, S.C. Shantz, Energy analysis of public-key cryptography for wireless sensor networks, in: Third IEEE International Conference on Pervasive Computing and Communications (PerCom 2005), IEEE, 2005, pp. 324–328.

[74] A. Liu, P. Ning, Tinyecc: A configurable library for elliptic curve cryptography in wireless sensor networks, in: Proceedings of the 7th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN 2008), IEEE, 2008, pp. 245–256.

[75] M. Hutter, E. Wenger, Fast multi-precision multiplication for public-key cryptography on embedded microprocessors, in: B. Preneel, T. Takagi (Eds.), Proceedings of the 13th International Workshop on Cryptographic Hardware and Embedded Systems (CHES 2011), LNCS, vol. 6917, Springer, Berlin/Heidelberg, 2011, pp. 459–474.

[76] D.B. He, An efficient remote user authentication and key agreement protocol for mobile client–server environment from pairings, Ad Hoc Networks 10 (6) (2012) 1009–1016.

[77] F. Zhu, D. Wong, A. Chan, R. Ye, Password authenticated key exchange based on RSA for imbalanced wireless networks, in: Proceedings of 5th Information Security Conference (ISC 2002), LNCS, vol. 2433, Springer-Verlag, 2002, pp. 150–161.

[78] I.-E. Liao, C.-C. Lee, M.-S. Hwang, A password authentication scheme over insecure networks, J. Comput. Syst. Sci. 72 (4) (2006) 727–740.

[79] M. Turkanovic, M. Holbl, An improved dynamic password-based user authentication scheme for hierarchical wireless sensor networks, Electron. Electr. Eng. 19 (6) (2013) 109–116.

[80] X. Li, J. Niu, M.K. Khan, J. Liao, An enhanced smart card based remote user password authentication scheme, J. Network Comput. Appl. (2013), http://dx.doi.org/10.1016/j.jnca.2013.02.034.

[81] R. Maharana, An Improved User Authentication Protocol for Hierarchical Wireless Sensor Networks using Elliptic Curve Cryptography, Master's Thesis, National Institute of Technology Rourkela, India, 2013.

[82] D. Wang, C.G. Ma, P. Wang and Z. Chen, Robust Smart Card based Password Authentication Scheme Against Smart Card Security Breach, Cryptology ePrint Archive, Report 2012/439, 2012. <http://eprint.iacr.org/2012/439.pdf>.

[83] B. Pinkas, T. Sander, Securing passwords against dictionary attacks, in: Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS 2002), ACM, 2002, pp. 161–170.

[84] W. Burr, D. Dodson, R. Perlner, W. Polk, S. Gupta, E. Nabbus, NIST 800-63-1: NIST Special Publication 800-63-1 Electronic Authentication Guideline, Tech. Rep., National Institute of Standards and Technology, Gaithersburg, MD, December 2011.

[85] S.H. Wu, Y.F. Zhu, Q. Pu, Robust smart-cards-based user authentication scheme with user anonymity, Secur. Commun. Networks 5 (2) (2012) 236–248.

[86] M. Bellare, P. Rogaway, Entity authentication and key distribution, in: D. Stinson (Ed.), Advances in Cryptology – CRYPTO'93, LNCS, vol. 773, Springer, Berlin/Heidelberg, 1994, pp. 232–249.

[87] C. Herley, P. Van Oorschot, A research agenda acknowledging the persistence of passwords, IEEE Secur. Privacy 10 (1) (2012) 28–36.

Ding Wang received his B.S. Degree in Information Security from Nankai University, Tianjin, China, in 2008. Then he went to Information Engineering University of PLA to work toward Information Security Engineering. Now he is pursuing his Ph.D. degree at Peking University, Beijing, China. He has published more than 20 referred research papers at Elsevier, IEEE and Wiley journals, and conferences such as DBSec 2012, ICICS 2012, ISC 2013 and WCNC 2014. He was awarded the Top-Ten Distinguished Graduate Academic Star of the University in 2012. His research interests include cryptography and wireless network security.

Ping Wang received his B.S. degree from University of Electronic Science and Technology of China, China, in 1983, and Ph.D. degree in Computer Science from University of Massachusetts in 1996. He has served as chief engineer in Open Software Foundation and Lucent Technologies Ltd. Currently, he is a full-time Professor and director of the PKU-PSU Joint Laboratory of Intelligent Computing and Smart Sensing. He served as technique committee co-chairs of RFIDSec'11 Asia. He has wide interests in distributed computing, system software and system security.