# Cryptanalysis of Two Dynamic ID-Based Remote User Authentication Schemes for Multi-server Architecture

Ding Wang[1,2,⋆], Chun-guang Ma[1], De-li Gu[1], and Zhen-shan Cui[1]

[1] College of Computer Science and Technology, Harbin Engineering University, Harbin City 150001, China
[2] Automobile Management Institute of PLA, Bengbu City 233011, China
wangdingg@mail.nankai.edu.cn, machunguang@hrbeu.edu.cn

**Abstract.** In NSS'10, Shao and Chin pointed out that Hsiang and Shih's dynamic ID-based remote user authentication scheme for multi-server environment has several security flaws and further proposed an improved version which is claimed to be efficient and secure. In this study, however, we will demonstrate that Shao-Chin's scheme still cannot achieve the claimed security goals, and we report its following flaws: (1) It cannot withstand offline password guessing attack under their non-tamper resistance assumption of the smart card; (2) It fails to provide user anonymity; (3) It is prone to user impersonation attack. More recently, Li et al. found that Sood et al.'s dynamic ID-based authentication protocol for multi-server architecture is still vulnerable to several kinds of attacks and presented a new scheme that attempts to overcome the identified weaknesses. Notwithstanding their ambitions, Li et al.'s scheme is still found vulnerable to various known attacks by researchers. In this study, we perform a further cryptanalysis and uncover its two other vulnerabilities: (1) It cannot achieve user anonymity, which is the essential goal of a dynamic ID-based scheme; (2) It is susceptible to offline password guessing attack. The proposed cryptanalysis discourages any use of the two schemes under investigation in practice and reveals some subtleties and challenges in designing this type of schemes.

**Keywords:** Cryptanalysis, Authentication protocol, Offline password guessing attack, Smart card, Multi-Server architecture.

## 1 Introduction

With the rapid growth of Internet applications, the number of service providing servers proliferates at an ever-increasing rate [1, 2]. The distributed locations of service servers make it convenient and efficient for subscribers to access resources, and it is of great concern to protect the users and systems' security and privacy from malicious adversaries. Accordingly, user authentication is crucial to assure one communicating participant of the legitimacy of the corresponding

⋆ Corresponding author.

L. Xu, E. Bertino, and Y. Mu (Eds.): NSS 2012, LNCS 7645, pp. 462–475, 2012.

party by acquisition of corroborative evidence, preventing unauthorized clients from accessing system services for multi-server environment. Among numerous methods for user authentication, password based authentication using smart cards is the most convenient and effective two-factor authentication mechanism and has been widely adopted in many security-critical applications, such as e-banking, e-commerce and e-health [3].

In 1991, Chang and Wu [4] introduced the first password based remote user authentication schemes using smart cards, since then there have been many of this type of schemes proposed [5–8]. Although the issue of password authentication with smart cards for single-server environment recently has already been well studied in [5–8], it is extremely difficult for a user to remember these numerous different sets of identities and passwords when she employs these single-server architecture schemes to login and access different remote service servers.

To address this issue, a number of smart card based password authentication schemes for multi-server environment has been presented quite recently [9–12]. A sound and practical remote user authentication protocol for multi-server environment should be of high efficiency and can resist various related attacks, as well as the provision of some desirable features, such as mutual authentication, key agreement, local password update, user anonymity and so on. However, all of these schemes for multi-server environment are found impractical or completely insecure shortly after they were first proposed [13–15], which outlines the need for intensive further research and dynamic ID-based schemes that can preserve user anonymity are of particular interest.

In 2010, Shao and Chin [16] proposed an improved dynamic ID-based authentication scheme for multi-server environment to overcome the weakness of Hsiang-Shin's scheme [12]. The authors claimed that their improvement provides mutual authentication and is free from all related cryptographic attacks, such as offline password guessing attack, insider attack and impersonation attack. Although their scheme is efficient and superior to the previous solutions for implementation in resource-constrained applications, we find their scheme cannot achieve the claimed security: their scheme is vulnerable to offline password attack and user impersonation attack, and fails to preserve user anonymity.

More recently, Li et al. [17] pointed out that, besides a design flaw, Sood et al. scheme [15] is susceptible to leak-of-verifier attack and stolen smart card attack, and further proposed an efficient and secure dynamic ID-based authentication scheme using smart cards for multi-server architecture to cope with these identified problems. Unfortunately, just two months after Li et al.'s scheme was first published online, the replay attack, password guessing attack and masquerade attack are identified in their scheme by Han [18]. Later on, Xue et al. [19] also found Li et al.'s scheme cannot withstand the replay attack, denial of service attack, eavesdropping attack, internal attack and impersonation attack. Surprisingly, our further cryptanalysis demonstrates that Li et al.'s scheme still cannot preserve user anonymity, which is the most essential goal of a dynamic ID-based scheme. Besides, we also observed that Li et al.'s scheme is susceptible to another type of offline password guessing attack, which is more effective than and different from Han's. In addition, we point out that Xue et al.'s improvement over Li et al.'s scheme is still vulnerable to a similar password guessing attack.

The remainder of this paper is organized as follows: in Section 2, we review Shao-Chin's scheme. Section 3 describes the weaknesses of Shao-Chin's scheme. Li et al.'s scheme is reviewed in Section 4 and the corresponding cryptanalysis is given is Section 5. Section 6 concludes the paper.

## 2 Review of Shao-Chin's Scheme

In this section, we examine the dynamic ID-based authentication scheme using smart cards proposed by Shao and Chin [16] in NSS 2010. Shao-Chin's scheme consists of five phases: registration phase, login phase, authentication phase, password change phase and track phase. For ease of presentation, we employ some intuitive abbreviations and notations listed in Table 1 and we will follow the notations in Shao-Chin's scheme as closely as possible.

**Table 1.** Notations

| Symbol | Description |
|---|---|
| $U_i$ | $i^{th}$ user |
| $S$ | remote server |
| $ID_i$ | identity of user $U_i$ |
| $CID_i$ | dynamic identity of user $U_i$ |
| $P_i$ | password of user $U_i$ |
| $S_j$ | $j^{th}$ service providing server |
| $SID_j$ | identity of service server $S_j$ |
| $\oplus$ | the bitwise XOR operation |
| $\parallel$ | the string concatenation operation |
| $h(\cdot)$ | collision free one-way hash function |
| $A \rightarrow B : M$ | message $M$ is transferred through a common channel from $A$ to $B$ |
| $A \Rightarrow B : M$ | message $M$ is transferred through a secure channel from $A$ to $B$ |

Besides the users and the service servers, there is another participant, called registration center ($RC$), involved in the system, and $RC$ is trusted by all the users and service servers. Let $x$ and $z$ be two secret keys of $RC$.

### 2.1 Registration Phase

The registration phase is divided into two parts, namely, the server registration and the user registration.

(i) Server registration

1) $S_j$ chooses her identity $SID_j$;
2) $S_j \Rightarrow RC : \{SID_j\}$;
3) $RC$ computes $y_j = h(h(x) \parallel SID_j)$ and $h(z)$;
4) $RC \Rightarrow S_j : \{y_j, h(z)\}$.

(ii) User registration

1) $U_i$ chooses her $ID_i$ and $P_i$;
2) $U_i \Rightarrow RC : \{ID_i, P_i\}$;
3) $RC$ computes $T_i = h(ID_i \parallel x)$, $R_i = h(x) \oplus h(z) \oplus T_i$, $V_i = T_i \oplus h(ID_i \parallel P_i)$ and $H_i = h(T_i)$ and stores $\{R_i, V_i, H_i, h(\cdot)\}$ in the smart card;
4) $RC \Rightarrow U_i$: A smart card containing security parameters $\{R_i, V_i, H_i, h(\cdot)\}$.

### 2.2 Login Phase

When $U_i$ wants to login to $S_j$, the following operations will be performed:

Step L1. $U_i$ inserts her smart card into card reader, and inputs $ID_i$ and $P_i$.
Step L2. Smart card computes $T_i = V_i \oplus h(ID_i \parallel P_i)$, and checks whether $H_i$ equals $h(T_i)$ or not. If they are equal, the user proceeds to the next step. Otherwise, the login request is rejected.
Step L3. Smart card generates a random number $r$ and computes $B_1 = R_i \oplus T_i \oplus h(r \parallel T_i)$
Step L4. $U_i \rightarrow S_j : \{B_1\}$.
Step L5. On receiving $B_1$ from $U_i$, $S_j$ computes $B_2 = B_1 \oplus h(z)$;
Step L6. $S_j \rightarrow U_i : \{B_2\}$.
Step L7. Smart card chooses a random number $N_i$ and computes $y_j = h(B_2 \oplus h(r\|T_i)\|SID_j)$, $CID_i = ID_i \oplus h(B_2 \oplus h(r\|T_i)\|N_i)$, $G_i = CID_i \oplus h(y_j\|N_i)$ and $C = h(CID_i\|G_i\|N_i)$.
Step L8. $U_i \rightarrow S_j : \{C, G_i, N_i\}$.

### 2.3 Authentication Phase

After receiving the login request from $U_i$, $S_j$ performs the following operations:

Step A1. The server $S_j$ Computes $CID_i = G_i \oplus h(y_j\|N_i)$ and, then checks whether the received $C$ is equal to the computed $h(CID_i\|G_i\|N_i)$. If the equality does not hold, the server $S_j$ rejects the login request.
Step A2. $S_j$ generates a random number $N_j$ and computes $M_1 = h(CID \parallel SID_j \parallel N_i)$.
Step A3. $S_j \rightarrow U_i : \{M_1, N_j\}$.
Step A4. Upon receiving the response message from $S_j$, $U_i$ computes $h(CID_i\| SID_j\|N_j)$ and compares it with $M_1$. The equality indicates the legitimacy of $S_j$. Otherwise, the login request is interrupted.
Step A5. $U_i$ computes $M_2 = h(CID_i\|SID_j\|N_j)$.
Step A6. $U_i \rightarrow S_j : \{M_2\}$.
Step A7. On receiving $M_2$, $S_j$ checks whether the received $M_2$ equals the computed $h(CID_i\|SID_j\|N_j)$. The equality indicates the legitimacy of $U_i$. Otherwise, the access request is interrupted.
Step A8. After authenticating each other, $U_i$ and $S_j$ use the same session key $SK = h(CID_i\|SID_j\|N_i\|N_j)$ to secure subsequent communications.

### 2.4 Password Change Phase and Track Phase

Since both the password change phase and track phase have little relevance with our discussions, they are omitted here.

## 3 Cryptanalysis of Shao-Chin's Scheme

There are three assumptions explicitly made in Shao-Chin's scheme [16]:

(i) An adversary $\mathcal{A}$ has total control over the communication channel between the user $U$ and the remote server $S$. In other words, the attacker can intercept, block, delete, insert or alter any messages exchanged in the channel.
(ii) The secret parameters stored in the smart card can be revealed once a legitimate user's smart card is somehow obtained (e.g. picked up or stolen) by $\mathcal{A}$.
(iii) The user-memorable passwords are weak, i.e. of low entropy.

Note that the above three assumptions, which are also made in the latest works [5–8, 13–15], are indeed reasonable: (1) Assumption *i* is accordant with the common Dolev-Yao adversary model for distributed communication; (2) Assumption *ii* is practical when taking the state-of-the-art side-channel attack techniques [20–22] into consideration; and (3) Assumption *iii* reveals the reality that users are allowed to choose their own passwords at will during the password change phase and registration phase, usually the users are apt to choose passwords that are related to their personal life [23], such as meaningful dates, phone numbers or license plate numbers, and the human-memorable passwords tends to be "weak passwords" [24].

In the following discussions of the security pitfalls of Shao-Chin's scheme, based on the above three assumptions, we assume that an adversary can extract the secret parameters $\{V_i, R_i, H_i\}$ stored in the legitimate user's smart card, and could also intercept or block the exchanged messages $\{B_1, B_2, C, G_i, N_i, M_i, N_j, M_2\}$ during the login and authentication phase.

### 3.1 No Provision of User Anonymity

A protocol preserving user anonymity prevents an adversary from acquiring sensitive information about an individual's social circle, preferences, lifestyles, shopping patterns, etc. by analyzing the login history, the services requested, or the communications being accessed [25]. In addition, the leakage of user-specific information may cause an unauthorized entity or malicious attacker to track the user's current location and login history [26]. Hence, assuring anonymity not only does protect user privacy but also makes remote user authentication protocols more secure. In Shao-Chin's scheme, the dynamic-ID technique is employed to provide the feature of user anonymity, however, the following attack demonstrates the failure of their attempt.

Let us see how a dishonest service provider $S_k$ colluding with a malicious privileged user $U_m$ successfully breach the anonymity of any legitimate user, say $U_i$. $U_m$ having her own smart card can gather information $R_m, V_m$ from her own smart card, with previously intercepted authentication messages $\{B_1, B_2, N_i, G_i, C\}$ that are exchanged between $U_m$ and any service provider, say $S_j$, $U_m$ and $S_k$ can collude to compute $ID_i$ corresponding to $U_i$ as follows:

**Step 1.** $U_m$ computes $T_m = V_m \oplus h(ID_m \| P_m)$, where $V_m$ is revealed from her own smart card, $ID_m$ and $P_m$ is known to herself;
**Step 2.** $U_m$ computes $h(x) \oplus h(z) = R_m \oplus T_m$, where $R_m$ is revealed;
**Step 3.** $U_m$ and $S_k$ collude to compute $h(x) = (h(x) \oplus h(z)) \oplus h(z) = (R_m \oplus T_m) \oplus h(z)$, where $h(z)$ is known to all service servers, including $S_k$.
**Step 4.** Guesses $U_i$'s identity to $ID_i^*$;
**Step 5.** Computes $CID_i^* = ID_i^* \oplus h(h(x) \| N_i)$, where

$$\begin{aligned} h(x) &= h(x) \oplus (h(r\|T_i) \oplus h(r\|T_i)) \oplus (h(z) \oplus h(z)) \\ &= (h(x) \oplus h(z) \oplus h(r\|T_i)) \oplus (h(z) \oplus h(r\|T_i)) \\ &= B_1 \oplus (h(z) \oplus h(r\|T_i)) = B_2 \oplus h(r\|T_i) \end{aligned}$$

**Step 6.** Computes $C^* = h(CID_i^* \| G_i \| N_i)$, where $G_i$ and $N_i$ is intercepted.
**Step 7.** Verifies the correctness of $ID_i^*$ by checking if the computed $C^*$ is equal to the intercepted $C$;
**Step 8.** Goes back to Step 4 until the correct value of $ID_i$ is found.

In practice, a user's identity is often drawn from a very limited space, say $\mathcal{D}_{id}$, the above procedure can be completed in polynomial time.

It is worth noting that, in the above attack, the malicious user $U_m$ only needs to extract the security parameters stored in her own smart card, she does not need to obtain any information about the victim user $U_i$ except the public authentication messages originating from $U_i$. As a result, the above attack is effective and practical. In conclusion, once an internal user colludes with a dishonest service server, user anonymity will be breached in Shao-Chin's scheme, while user anonymity is the most essential security feature that a dynamic identity-based authentication scheme is designed to provide.

### 3.2 Offline Password Guessing Attack

As stated in Section 3.1, any legitimate user $U_i$'s identity can be breached when an internal malicious user $U_m$ colludes with a service server $S_k$. Once the victim user $U_i$'s identity $ID_i$ is obtained by $U_m$ and $S_k$, $U_i$'s password $P_i$ can also be offline guessed as follows:

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

Let $|\mathcal{D}_{id}|$ and $|\mathcal{D}_{pw}|$ denote the number of identities in identity space $\mathcal{D}_{id}$ and the number of passwords in password space $\mathcal{D}_{pw}$, respectively. The running time of the above attack procedure is $\mathcal{O}(|\mathcal{D}_{id}| * (3T_H + 5T_X) + |\mathcal{D}_{pw}| * (2T_H + T_X))$, where $T_H$ is the running time for Hash operation and $T_X$ is the running time for $XOR$ 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, e.g. $|\mathcal{D}_{id}| \leq |\mathcal{D}_{pw}| = 10^6$ [24]. As $|\mathcal{D}_{id}|$ and $|\mathcal{D}_{pw}|$ are very limited in practice, the above attack can be completed in polynomial time.

Note that, in this attack, the malicious user $U_m$ not only needs to extract the security parameters stored in her own smart card, but also needs to obtain the secret data stored in the smart card of victim user $U_i$. Although this assumption is much constrained, our attack demonstrates the feasibility of offline password guessing attack on Shao-Chin's scheme under their non-tamper resistance assumption of the smart card, thereby contradicting the claim made in [16].

### 3.3 User Impersonation Attack

An internal malicious user $U_m$ and a service server $S_k$ can collude to impersonate any legitimate user (even non-existent user), say $U_{ran}$, to login any service server, say $S_j$, as follows:

**Step 1.** $U_m$ computes $T_m = V_m \oplus h(ID_m \| P_m)$, as $V_m$ is revealed from her own smart card, $ID_m$ and $P_m$ is known to herself;
**Step 2.** $U_m$ computes $h(x) \oplus h(z) = R_m \oplus T_m$, where $R_m$ is revealed;
**Step 3.** $S_k$ and $U_m$ collude to compute $h(x) = (h(x) \oplus h(z)) \oplus h(z) = (R_m \oplus T_m) \oplus h(z)$, where $h(z)$ is known to all service servers, including $S_k$.
**Step 4.** $U_m$ sends a random value $X$ to any service server, say $S_j$;
**Step 5.** $U_m$ ignores the response $\{B_2\}$ sent back by $S_j$ and computes $y_j = h(h(x) \| SID_j)$, where $SID_j$ is $S_j$'s identity.
**Step 6.** $U_m$ computes $CID_{ran} = ID_{ran} \oplus h(h(x) \parallel N_{ran})$, $G_{ran} = CID_{ran} \oplus h(y_j \parallel N_{ran})$ and $C = h(CID_{ran} \parallel G_{ran} \parallel N_{ran})$, where $N_{ran}$ is a random number chosen by $U_m$.
**Step 7.** $U_m$ sends $\{C, G_{ran}, N_{ran}\}$ to $S_j$.
**Step 8.** On receiving the response $\{M_{ran}, N_j\}$ sent back by $S_j$, $U_m$ computes $M_2 = h(CID_{ran} \parallel SID_j \parallel N_j)$ and the session key $SK = h(CID_{ran} \parallel SID_j \parallel N_{ran} \parallel N_j)$.
**Step 9.** $U_m$ sends $\{M_2\}$ to $S_j$.

It is easy to see that: 1) On receiving $X$ sent by $U_m$ in Step 4, $S_j$ will send back $B_2 = X \oplus h(z)$ according to the protocol; 2) On receiving $\{C, G_{ran}, N_{ran}\}$ sent by $U_m$ in Step 7, $S_j$ will find no abnormality when checking the validity of $C$, because $U_m$ indeed has computed the correct $y_j = h(h(x) \| SID_j) = h(B_2 \oplus h(r \| T_i) \| SID_j)$ in Step 5, and the latter expression is justified as follows

$$\begin{aligned} h(x) &= h(x) \oplus (h(r\|T_i) \oplus h(r\|T_i)) \oplus (h(z) \oplus h(z)) \\ &= (h(x) \oplus h(z) \oplus h(r\|T_i)) \oplus (h(z) \oplus h(r\|T_i)) \\ &= B_1 \oplus (h(z) \oplus h(r\|T_i)) = B_2 \oplus h(r\|T_i). \end{aligned}$$

3) On receiving $M_2$ sent by $U_m$ in Step 9, $S_j$ will find no abnormality when checking the validity of $M_2$, because $U_m$ has indeed computed the valid $CID_{ran} = ID_{ran} \oplus h(h(x) \parallel N_{ran})$ in Step 5, where $h(x) = B_2 \oplus h(r \| T_i)$.

It is worth noting that, as with the password guessing attack presented in Section 3.1, in this attack, the malicious user $U_m$ only needs to extract the security parameters stored in her own smart card, she does not need to obtain any information about the victim $U_i$ except the public authentication messages originating from $U_i$. As a result, this impersonation attack is effective and practical.

## 4 Review of Li et al.'s Scheme

In this section, we briefly review the dynamic identity based authentication protocol for multi-server architecture using smart cards proposed by Li et al. in 2012. Li et al.'s protocol also involves three participants, i.e., the user ($U_i$), the service providing server ($S_j$) and the control server ($CS$). It is should be noted that $CS$, a trusted party, is not only responsible for the registration but also involved in the authentication process of $U_i$ and $S_j$. $CS$ is in possession of a master secret key $x$ and a secret number $y$. There are four phases in their protocol: registration, login, authentication and session key agreement, and password change. In the following, we employ the notations listed in Table 1.

### 4.1 Registration Phase

The registration phase can be divided into two parts, namely, the server registration and the user registration.

(i) Server registration
1) $S_j$ chooses her identity $SID_j$;
2) $S_j \Rightarrow CS : \{SID_j\}$;
3) $CS$ computes $h(SID_j \parallel y)$ and $h(x \parallel y)$;
4) $CS \Rightarrow S_j : \{h(x \parallel y), h(SID_j \parallel y)\}$.

(ii) User registration
1) $U_i$ freely chooses her $ID_i$ and $P_i$, and chooses a random number $b$. Then, $U_i$ computes $A_i = h(b \parallel P_i)$;
2) $U_i \Rightarrow CS : \{ID_i, A_i\}$;
3) $CS$ computes $B_i = h(ID_i \parallel x)$, $C_i = h(ID_i \parallel h(y) \parallel A_i)$, $D_i = B_i \oplus h(ID_i \parallel A_i)$, $E_i = B_i \oplus h(y \parallel x)$, and stores $\{C_i, D_i, E_i, h(\cdot), h(y)\}$ in the smart card;
4) $CS \Rightarrow U_i$: A smart card containing parameters $\{C_i, D_i, E_i, h(\cdot), h(y)\}$.
5) Upon receiving the smart card, $U_i$ enters $b$ into it.

### 4.2 Login Phase

When $U_i$ wants to login to $S_j$, the following operations will be performed:

Step L1. User $U_i$ inserts her smart card into a card reader and inputs her identity $ID_i$, password $P_i$ and the service server's identity $SID_j$.

Step L2. The smart card computes $A_i = h(b \parallel P_i)$ and $C_i' = h(ID_i \parallel h(y) \parallel A_i)$, and checks whether $C_i' = C_i$. If they are equal, it indicates that $U_i$ is a legal card holder.

Step L3. The smart card generates a random number $N_{i1}$, and computes $B_i = D_i \oplus h(ID_i \parallel A_i)$, $F_i = h(y) \oplus N_{i1}$, $P_{ij} = E_i \oplus h(h(y) \parallel N_{i1} \parallel SID_j)$, $CID_i = A_i \oplus h(B_i \parallel F_i \parallel N_{i1})$, $G_i = h(B_i \parallel A_i \parallel N_{i1})$.

Step L4. $U_i \rightarrow S_j : \{F_i, G_i, P_{ij}, CID_i\}$.

### 4.3 Authentication and Session Key Agreement Phase

Step A1. On receiving the login request, $S_j$ chooses a random number $N_{i2}$, and computes $K_i = h(SID_j \parallel y) \oplus N_{i1}$ and $M_i = h(h(x \parallel y) \parallel N_{i2})$.

Step A2. $S_j \rightarrow CS : \{F_i, G_i, P_{ij}, CID_i, SID_j, K_i, M_i\}$.

Step A3. Upon receiving the login request $\{F_i, G_i, P_{ij}, CID_i, SID_j, K_i, M_i\}$, $CS$ computes $N_{i2} = K_i \oplus h(SID_j \parallel y)$, $M_i' = h(h(x \| y) \| N_{i2})$, and checks whether $M_i'$ equals the received $M_i$. If they are equal, the validity of the server $S_j$ is verified by the control server $CS$. Otherwise, the $CS$ terminates the session.

Step A4. $CS$ computes $N_{i1} = F_i \oplus h(y)$,$B_i = P_{ij} \oplus h(h(y) \| N_{i1} \| SID_j) \oplus h(y \| x)$ $(= E_i \oplus h(y \| x))$, $A_i = CID_i \oplus h(B_i \parallel F_i \parallel N_{i1})$, $G_i' = h(B_i \parallel A_i \parallel N_{i1})$ and checks $G_i' \stackrel{?}{=} G_i$. If the verification holds, the legitimacy of user $U_i$ is authenticated by $CS$. Otherwise $CS$ terminates the session.

Step A5. $CS$ generates a random number $N_{i3}$, and computes $Q_i = N_{i1} \oplus N_{i3} \oplus h(SID_j \parallel N_{i2})$,$R_i = h(A_i \parallel B_i) \oplus h(N_{i1} \oplus N_{i2} \oplus N_{i3})$, $V_i = h(h(A_i \parallel B_i) \parallel h(N_{i1} \oplus N_{i2} \oplus N_{i3}))$, $T_i = N_{i2} \oplus N_{i3} \oplus h(A_i \parallel B_i \parallel N_{i1})$.

Step A6. $CS \rightarrow S_j : \{Q_i, R_i, V_i, T_i\}$.

Step A7. On receiving the authentication message $\{Q_i, R_i, V_i, T_i\}$ from $CS$, server $S_j$ computes $N_{i1} \oplus N_{i3} = Q_i \oplus h(SID_j \parallel N_{i2})$, $h(A_i \parallel B_i) = R_i \oplus h(N_{i1} \oplus N_{i3} \oplus N_{i2})$, $V_i' = h(h(A_i \parallel B_i) \parallel h(N_{i1} \oplus N_{i3} \oplus N_{i2}))$, and checks $V_i' \stackrel{?}{=} V_i$. If they are not equal, $S_j$ terminates the session. Otherwise, the legitimacy of $CS$ is authenticated by the server $S_j$.

Step A8. $S_j \rightarrow U_i : \{V_i, T_i\}$.

Step A9. Upon receiving $\{V_i, T_i\}$ from $S_j$, the smart card computes $N_{i2} \oplus N_{i3} = T_i \oplus h(A_i \parallel B_i \parallel N_{i1})$, $V_i' = h(h(A_i \parallel B_i) \parallel h(N_{i2} \oplus N_{i3} \oplus N_{i1}))$, and checks $V_i' \stackrel{?}{=} V_i$. If the verification fails, the user $U_i$ terminates the session. Otherwise, the legitimacy of the control server $CS$ and the server $S_j$ is authenticated by user $U_i$.

Finally, the user $U_i$, the server $S_j$ and the control server $CS$ agree on a common session key $SK = h(h(A_i \parallel B_i) \parallel (N_{i1} \oplus N_{i2} \oplus N_{i3}))$.

### 4.4 Password Change Phase

This phase is performed locally. When the user wants to update her password, this phase is invoked. Since this phase has little relevance with our discussions, it is omitted here.

## 5 Cryptanalysis of Li et al.'s Scheme

The three assumptions presented in Section 3 is also explicitly made in Li et al.'s paper when they analyze the security of Sood et al.'s scheme, and thus our following cryptanalysis is also based on these three assumptions.

Although Li et al.'s scheme has many attractive properties, such as provision of local password change, high efficiency and no time-synchronization problem, it fails to achieved many of the claimed security goals and has been found vulnerable to replay attack, password guessing attack and user impersonation attack by Han [18]. Besides these security pitfalls, later on Xue et al. further found it prone to leak-of-verifier attack, server spoofing attack and denial of service attack,[1] and they also presented an improvement.

Surprisingly, our further cryptanalysis demonstrates that Li et al.'s scheme still cannot preserve user anonymity, which is the most crucial goal of a dynamic ID-based scheme. Besides, we also observe that Li et al.'s scheme is susceptible to another type of offline password guessing attack, which is more effective than and different from Han's. Furthermore, we point out that Xue et al.'s improvement over Li et al.'s scheme is still vulnerable to a similar password guessing attack.

### 5.1 No Provision of User Anonymity

Let us see how a dishonest service provider $S_k$ colluding with a malicious internal user $U_m$ successfully breach the anonymity of any legitimate user, say $U_i$. $U_m$ having her own smart card can gather information $h(y)$ from her own smart card, with previously intercepted authentication messages $\{P_{ij}, SID_j\}$ that are exchanged between $U_m$, $CS$ and any service provider, say $S_j$, $U_m$ and $S_k$ can collude to compute $E_i$ corresponding to any user $U_i$ as follows:

**Step 1.** $U_m$ extracts $h(y)$ from her own smart card;
**Step 2.** $U_m$ and $S_k$ collude to compute $N_{i1} = F_i \oplus h(y)$, where $F_i$ is intercepted from the public channel;
**Step 3.** $U_m$ and $S_k$ collude to compute $E_i = P_{ij} \oplus h(h(y) \parallel N_{i1} \parallel SID_j)$, where $P_{ij}$ and $SID_j$ are intercepted from the public channel.

As $E_i$ is kept the same for all the login requests of user $U_i$ and is specific to $U_i$, this $E_i$ can be seen as user $U_i$'s identification. And an adversary can, therefore, use this information to identify and trace $U_i$'s login requests and activities. By

[1] We think Xue et al.'s internal attack and eavesdropping attack only constitute parts of replay attack, server spoofing attack, etc, and they may not be considered as independent kinds of attacks, and thus they are not listed here.

generalizing the above attack, any legal user who logins to service servers would be exposed to $U_m$ and $S_k$, and thus user anonymity is not preserved.

It should be noted that, in the above attack, the malicious user $U_m$ only needs to extract the security parameters stored in her own smart card, she does not need to obtain any information about the victim user $U_i$ except the public authentication messages originating from $U_i$. As a result, the above attack is effective and practical. In conclusion, once an internal user colludes with a dishonest service server, user anonymity will be breached in Li et al.'s scheme, while user anonymity is the most crucial security feature that a dynamic identity-based authentication scheme is designed to provide.

### 5.2 Offline Password Guessing Attack

Let us consider the following scenarios. In case a legitimate user $U_i$'s smart card is stolen by a malicious internal user $U_m$, and the stored secret values $h(y)$, $D_i$, $E_i$ and $b$ can be extracted. Note that this assumption is reasonable as described in Assumption *iii* and it is also explicitly made in Li et al.'s scheme. With the previously eavesdropped message $\{F_i, CID_i, G_i\}$, this malicious internal user $U_m$ can successfully guess the password of $U_i$ as follows:

**Step 1.** Extracts $h(y)$ from her own smart card;
**Step 2.** Computes $N_{i1} = F_i \oplus h(y)$, where $F_i$ is intercepted from the public channel;
**Step 3.** Computes $E_i = P_{ij} \oplus h(h(y) \parallel N_{i1} \parallel SID_j)$, where $P_{ij}$ and $SID_j$ are intercepted from the public channel.
**Step 4.** Computes $h(y\|x) = E_m \oplus B_m = E_m \oplus D_m \oplus h(ID_m\|A_m) = E_m \oplus D_m \oplus h(ID_m\|h(b\|P_m))$, where $E_m$, $D_m$ and $b$ are revealed from $U_m$'s own smart card;
**Step 5.** Computes $B_i = E_i \oplus h(y\|x)$, where $E_i$ is revealed from $U_i$'s smart card;
**Step 6.** Computes $A_i = CID_i \oplus h(B_i\|F_i\|N_{i1})$;
**Step 7.** Guesses the value of $P_i$ to be $P_i^*$ from the password space $\mathcal{D}$.
**Step 8.** Computes $A_i^* = h(b\|P_i^*)$, where $b$ is revealed from $U_i$'s smart card.
**Step 9.** Verifies the correctness of $P_i^*$ by checking if $A_i^*$ equals to $A_i$.
**Step 10.** Repeats Steps 7, 8 and 9 until the correct value of $P_i$ is found.

Let $|\mathcal{D}|$ denote the number of passwords in the password space $\mathcal{D}$. Then the running time of the attacker $U_m$ is $\mathcal{O}(|\mathcal{D}| * (5T_H + 6T_X))$, where $T_H$ is the running time for Hash operation and $T_X$ is the running time for $XOR$ operation. So, the time for $U_m$ to recover the password is a linear function of the number of passwords in the password space. When the password space is small, e.g., $|\mathcal{D}| = 10^6$ [24], $U_m$ may recover the password in seconds on a PC.

It should be noted that, in this attack, the malicious user $U_m$ only needs to guess $U_i$'s password, while in the offline password guessing attack proposed by Han [18], the attacker needs to guess both $U_i$'s password and identity correctly at the same time. From this point of view, our attack is more effective. But our disadvantage is that, the adversary in our attack should be an internal user, while the adversary in Han's attack is not subject to this restriction.

### 5.3 Offline Password Guessing Attack on Xue et al.'s Improvement

In [19], Xue et al. pointed out that Li et al.'s scheme vulnerable to several attacks and further proposed an improvement that is claimed to be secure.[2] However, we find Xue et al.'s improvement is still vulnerable to an offline password guessing attack as described in the following.

Let us consider the following scenarios. In case a legitimate user $U_i$'s smart card is stolen by an adversary $\mathcal{A}$, and the stored secret values such as $C_i$, $D_i$ and $b$ can be extracted. Note that this assumption is explicitly made in Xue et al.'s improvement. With a previously eavesdropped message $\{F_i, PID_i, TS_i\}$, $\mathcal{A}$ can acquire $U_i$'s password $PW_i$ by performing the following attack procedure:

**Step 1.** Guesses the value of $P_i$ to be $P_i^*$ from the password space $\mathcal{D}$;
**Step 2.** Computes $A_i^* = h(b \| P_i^*)$, where $b$ is revealed from $U_i$'s smart card;
**Step 3.** Computes $B_i^* = D_i \oplus h(PID_i \oplus A_i^*)$, where $PID_i$ is intercepted from the public channel;
**Step 4.** Computes $N_{i1}^* = F_i \oplus B_i^*$;
**Step 5.** Computes $G_i^* = b \oplus h(B_i^* \| N_{i1}^* \| TS_i \| \text{“11”})$;
**Step 6.** Verifies the correctness of $P_i^*$ by checking if $G_i^*$ equals to the intercepted $G_i$;
**Step 7.** Repeats the above steps until the correct value of $P_i$ is found.

Since the size of password dictionary, i.e.$|\mathcal{D}|$, often is very limited in practice, the above attack procedure can be completed in polynomial time.

***Notes and Countermeasure.*** We have analyzed more than sixty recently proposed smart card based password authentication schemes for single-server environment and twelve schemes for multi-server architecture, and find these schemes (no matter for single-server environment or multi-server architecture) that do not employ public-key techniques definitely vulnerable to the offline password guessing attack under the three assumptions (most essentially, the non-tamper resistance assumption of the smart card) introduced in Section 3. In other words, all these schemes that do not employ public-key techniques but claim to be secure under these three assumptions are found problematic. A related work done by Halevi and Krawczyk [27] provides very strong evidence (with the probability of $\boldsymbol{P \neq NP}$) that, under the common Dolev-Yao adversary model, no password protocol (the traditional one-factor password authentication) can be free from offline password guessing attack if the public-key techniques are not employed. Here, we conjecture that under the three assumptions introduced in Section 3, no smart card based password protocol (two-factor authentication) can be free from offline password guessing attack if the public-key techniques are not employed. And now the countermeasure is obvious: resorting to public-key techniques like [5–8].

[2] According to Xue et al.'s statement, this improvement has been submitted to Journal of Network and Computer Applications.

## 6 Conclusion

Understanding security failures of cryptographic protocols is the key to both revising existing protocols and proposing future schemes. In this paper, we have shown that two dynamic ID-based remote user authentication schemes for multi-server environment are completely broken and only radical revisions of the protocols can possibly eliminate the identified defects and thus the two schemes under investigation are not recommended for practical application. Remarkably, our cryptanalysis highlights the difficulties and challenges in designing secure and efficient dynamic ID-based remote user authentication schemes for multi-server architecture, in the hope that no similar mistakes are made in the future.

**Acknowledgment.** This research was partially supported by the National Natural Science Foundation of China (NSFC) under No. 61170241 and the International Exchange Program of Harbin Engineering University for Innovation-oriented Talents Cultivation.

## References

1. Bouyoucef, K., Khorasani, K.: A robust distributed congestion-control strategy for differentiated-services network. IEEE Transactions on Industrial Electronics 56(3), 608–617 (2009)
2. Barolli, L., Xhafa, F.: JXTA-OVERLAY: A P2P platform for distributed, collaborative and ubiquitous computing. IEEE Transactions on Industrial Electronics 58(6), 2163–2172 (2010)
3. Lin, S., Hung, M., Tsai, C., Chou, L.: Development of an ease-of-use remote healthcare system architecture using rfid and networking technologies. Journal of Medical Systems, 1–15 (2012), doi:10.1007/s10916-012-9836-0
4. Chang, C.C., Wu, T.C.: Remote password authentication with smart cards. IEE Proceedings-Computers and Digital Techniques 138(3), 165–168 (1991)
5. Ma, C.-G., Wang, D., Zhang, Q.-M.: Cryptanalysis and Improvement of Sood et al.'s Dynamic ID-Based Authentication Scheme. In: Ramanujam, R., Ramaswamy, S. (eds.) ICDCIT 2012. LNCS, vol. 7154, pp. 141–152. Springer, Heidelberg (2012)
6. Wu, S.H., Zhu, Y.F., Pu, Q.: Robust smart-cards-based user authentication scheme with user anonymity. Security and Communication Networks 5(2), 236–248 (2012)
7. Wang, D., Ma, C.G., Wu, P.: Secure Password-Based Remote User Authentication Scheme with Non-tamper Resistant Smart Cards. In: Cuppens-Boulahia, N., Cuppens, F., Garcia-Alfaro, J. (eds.) DBSec 2012. LNCS, vol. 7371, pp. 114–121. Springer, Heidelberg (2012)
8. Wang, Y.: Password Protected Smart Card and Memory Stick Authentication against Off-Line Dictionary Attacks. In: Gritzalis, D., Furnell, S., Theoharidou, M. (eds.) SEC 2012. IFIP AICT, vol. 376, pp. 489–500. Springer, Heidelberg (2012)
9. Lin, I., Hwang, M., Li, L.: A new remote user authentication scheme for multi-server architecture. Future Generation Computer Systems 19(1), 13–22 (2003)
10. Tsaur, W., Wu, C., Lee, W.: A smart card-based remote scheme for password authentication in multi-server internet services. Computer Standards & Interfaces 27(1), 39–51 (2004)
11. Liao, Y., Wang, S.: A secure dynamic id based remote user authentication scheme for multi-server environment. Computer Standards & Interfaces 31(1), 24–29 (2009)

12. Hsiang, H., Shih, W.: Improvement of the secure dynamic id based remote user authentication scheme for multi-server environment. Computer Standards & Interfaces 31(6), 1118–1123 (2009)
13. Tan, Z.: Cryptanalysis of two id based password authentication schemes for multi-server environments. International Journal of Digital Content Technology and its Applications 5(1), 87–94 (2011)
14. Yeh, K., Lo, N., Li, Y.: Cryptanalysis of hsiang-shihs authentication scheme for multi-server architecture. International Journal of Communication Systems 24(7), 829–836 (2011)
15. Sood, S., Sarje, A., Singh, K.: A secure dynamic identity based authentication protocol for multi-server architecture. Journal of Network and Computer Applications 34(2), 609–618 (2011)
16. Shao, M., Chin, Y.: A novel approach to dynamic id-based remote user authentication scheme for multi-server environment. In: 2010 4th International Conference on Network and System Security (NSS 2010), pp. 548–553. IEEE Press, New York (2010)
17. Li, X., Xiong, Y., Ma, J., Wang, W.: An enhanced and security dynamic identity based authentication protocol for multi-server architecture using smart cards. Journal of Network and Computer Applications 35(2), 763–769 (2012)
18. Han, W.: Weaknesses of a dynamic identity based authentication protocol for multi-server architecture. Arxiv preprint arXiv:1201.0883 (2012), `http://arxiv.org/abs/1201.0883`
19. Xue, K., Hong, P., Ma, C.: A lightweight dynamic pseudonym identity based authentication and key agreement protocol without verification tables for multi-server architecture. Arxiv preprint arXiv:1204.3831 (2012), `http://arxiv.org/abs/1204.3831`
20. Kocher, P., Jaffe, J., Jun, B.: Differential Power Analysis. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 388–789. Springer, Heidelberg (1999)
21. Messerges, T.S., Dabbish, E.A., Sloan, R.H.: Examining smart-card security under the threat of power analysis attacks. IEEE Transactions on Computers 51(5), 541–552 (2002)
22. Kasper, T., Oswald, D., Paar, C.: Side-Channel Analysis of Cryptographic RFIDs with Analog Demodulation. In: Juels, A., Paar, C. (eds.) RFIDSec 2011. LNCS, vol. 7055, pp. 61–77. Springer, Heidelberg (2012)
23. Florencio, D., Herley, C.: A large-scale study of web password habits. In: Proceedings of WWW 2007, pp. 657–666. ACM Press, New York (2007)
24. Klein, D.V.: Foiling the cracker: A survey of, and improvements to, password security. In: Proceedings of the 2nd USENIX Security Workshop, pp. 5–14 (1990)
25. Bao, F., Deng, R.: Privacy Protection for Transactions of Digital Goods. In: Qing, S., Okamoto, T., Zhou, J. (eds.) ICICS 2001. LNCS, vol. 2229, pp. 202–213. Springer, Heidelberg (2001)
26. Tang, C., Wu, D.: Mobile privacy in wireless networks-revisited. IEEE Transactions on Wireless Communications 7(3), 1035–1042 (2008)
27. Halevi, S., Krawczyk, H.: Public-key cryptography and password protocols. ACM Transactions on Information and System Security 2(3), 230–268 (1999)