Information Fusion 14 (2013) 498–503

Contents lists available at SciVerse ScienceDirect

Information Fusion

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

Cryptanalysis of a remote user authentication scheme for mobile client–server environment based on ECC

Ding Wang [a,b,*], Chun-guang Ma [a]

[a] *College of Computer Science and Technology, Harbin Engineering University, Harbin City 150001, China*
[b] *Training Department, Automobile Management Institute of PLA, Bengbu City 233011, China*

ARTICLE INFO

Article history:
Received 1 August 2012
Received in revised form 12 November 2012
Accepted 31 December 2012
Available online 26 January 2013

Keywords:
Authentication protocol
Cryptanalysis
Mobile device
Elliptic curve cryptography
Random oracle model

ABSTRACT

Understanding security failures of cryptographic protocols is the key to both patching existing protocols and designing future schemes. The design of secure remote user authentication schemes based on elliptic curve crypto-graphy (ECC) for mobile applications is still quite a challenging problem, though many schemes have been published lately. In this paper, we analyze an efficient ID-based scheme for mobile client–server environment without the MapToPoint function introduced by He et al. in 2012. This proposal attempts to overcome many of the well known security and efficiency shortcomings of previous schemes, and it also carries a claimed proof of security in the random oracle model. However, notwithstanding its formal security arguments, we show that He et al.'s protocol even cannot attain the basic goal of mutual authentication by demonstrating its vulnerabilities to reflection attack and parallel session attack. Besides these two security vulnerabilities, their scheme also suffers from some practical pitfalls such as user anonymity violation and clock synchronization problem. In addition, we carry out an investigation into their security proof and propose some changes to the scheme so that it can achieve at least its basic security goal, in the hope that similar mistakes are no longer made in the future.

1. Introduction

With the rapid development of wireless network technologies, such as AMPS, GSM, GPRS, 3G and 4G, over the last couple of decades, more and more electronic transactions are processed on mobile devices (e.g. PDAs, laptops, smart cards and smart phones). Due to the portability of mobile devices, people can accomplish electronic transactions for secret Internet banking, online shopping, online voting, pay-TV, etc. anytime and anywhere, and it is of great concern to protect the systems and the users' privacy and security from malicious adversaries. Accordingly, user authentication becomes an essential security mechanism for remote systems to assure one communicating party of the authenticity of the corresponding party by acquisition of corroborative evidence. Generally, authentication factors are grouped into three categories [1]: (1) what you know (e.g., passwords, PINs); (2) what you have (e.g., tokens, smart card, portable storage devices); and (3) who you are (e.g., fingerprints, irides). Among the numerous methods based on one or more of these three types, the combination of the first two factors is one of the most popular and effective approaches for authentication in security-critical applications [2] such as e-banking, e-commerce, and e-health services.

In 1981, Lamport [3] proposed the first password authentication scheme to authenticate a remote user over an insecure channel. Later on, Chang and Wu [4] introduced the smart cards into remote user authentication schemes. Since then, many password-based two factor authentication schemes [5–11] have been proposed, where a client with a mobile device remembers a password and the corresponding server holds the long-term secret key that is used to verify the client's knowledge of the password. These easy-to-remember passwords, called weak passwords, have low entropy and thus are potentially vulnerable to various sophisticated attacks, especially offline password guessing attack [12], which is the gravest threat a well-designed password authentication scheme must be able to thwart. Halevi and Krawczyk [13] have proved that no password protocol can be free from offline password guessing attack if the public-key techniques are not employed. Since the computation ability and battery capacity of mobile devices are limited, the traditional public-key based remote authentication schemes are not suitable for mobile applications. As a result, a common feature of previous schemes is that computation efficiency and system security cannot be achieved at the same time, e.g., the schemes in [5–9] are found insecure, while the schemes in [10,11,14] need either pairing operation or several exponentiations on the mobile client side.

Fortunately, it seems to see the dawn in recent 2 years, where several schemes based on ECC without pairing on user side have been proposed to reduce computation cost while preserving

[*] Corresponding author at: College of Computer Science and Technology, Harbin Engineering University, Harbin City 150001, China. Tel.: +86 151 0459 6985; fax: +86 0451 8251 9600.
E-mail address: wangdingg@mail.nankai.edu.cn (D. Wang).

1566-2535/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.inffus.2012.12.002

security strength [15–18]. However, The reality of the situation is that this dilemma is only partially addressed and most of the ECC-based schemes were found severely flawed shortly after they were first put forward [19–21] (for a typical example, see [15]). In 2012, He et al. [22] pointed out that previous schemes are problematic both in security and performance, and thus proposed an efficient ID-based scheme for mobile client–server environment on ECC without the MapToPoint function. This proposal attempts to cope with many of the well known security and efficiency problems of previous schemes, and even includes a formal security proof in the random oracle model. Despite of its claim of provable security, He et al.'s scheme is in fact insecure in the face of an active adversary. In this study, we demonstrate this by presenting a reflection attack and a parallel session attack that breach the essential goal of mutual authentication. In addition, their scheme suffers from user anonymity violation and clock synchronization problem. Besides reporting these defects, we also work out what has gone wrong with the protocol and how to fix it.

The remainder of this paper is organized as follows: in Section 2, we review He et al.'s scheme. Section 3 describes the weaknesses of He et al.'s scheme. The proof analysis and security enhancement are given in Section 4 and Section 5 concludes the paper.

2. Review of He et al.'s scheme

2.1. Preliminaries

Throughout the paper, we will follow the original notations in He et al.'s scheme [22] as closely as possible.

- p, n: two large prime numbers;
- F_p: a finite field;
- E: an elliptic curve defined on finite field F_p with prime order n;
- G: the group of elliptic curve points on E;
- P: a base point on elliptic curve E with order n;
- $H_1(\cdot)$: a secure one-way hash function, where $H_1 : \{0,1\}^* \to Z_n^*$;
- $H_2(\cdot)$: a secure one-way hash function, where $H_2 : \{0,1\}^* \to Z_p^*$;
- $H_3(\cdot)$: a secure one-way hash function, where $H_3 : \{0,1\}^* \to Z_p^*$;
- $MAC_k(m)$: the secure message authentication code of m under the key k;
- (S, ID_S): the server and its corresponding identity string;
- (C_i, ID_{C_i}): a client and its corresponding identity string;
- (x, P_S): the server S's private/public key pair, where $P_S = xP$;
- $A \Rightarrow B{:}M$: message M is transferred through a secure channel from A to B;
- $A \to B{:}M$: message M is transferred through a common channel from A to B.

The security of He et al.'s scheme is based on the intractability of the following two mathematical problems on elliptic curves:

(i) *Computational Diffie–Hellman Assumption (CDHA)*: Given $P, xP, yP \in G$, it is hard to compute $xyP \in G$.
(ii) *Collision Attack Assumption 1 (k-CAA1)*: For an integer k, and $x \in Z_n^*$, $P \in G$, given $\left(P, xP, h_0, \left(h_1, \frac{1}{h_1+x}P\right), \ldots, \left(h_k, \frac{1}{h_k+x}P\right)\right)$, where $h_i \in Z_n^*$ and distinct for $0 \leqslant i \leqslant k$, it is hard to compute $\frac{1}{h_0+x}P$.

2.2. He et al.'s protocol

He et al.'s scheme consists of three phases, namely, system initialization phase, client registration phase and mutual authentication with key agreement phase.

System initializing phase. In this phase, S generates parameters of the system.

(1) S chooses an elliptic curve equation E;
(2) S selects a base point P with the order n over E;
(3) S selects its master key x and computes public key $P_S = xP$;
(4) The server chooses three secure one-way hash functions $H_1(\cdot)$, $H_2(\cdot)$, $H_3(\cdot)$ and a message authentication code $MAC_k(\cdot)$. The server keeps x in private and publishes $(F_p, E, n, P, P_S, H_1, H_2, H_3, MAC_k(\cdot))$.

Client registration phase. When a low-power computing client C_i with the identity ID_{C_i} wants to register to the server S, the following operation will be performed:

(1) $C_i \Rightarrow S : \{ID_{C_i}\}$.
(2) On receiving the registration message from C_i, the server S computes $h_{C_i} = H_1(ID_{C_i})$ and then uses the system private key x to compute the client's private key $D_{C_i} = \frac{1}{x+h_{C_i}}P \in G$.
(3) $S \Rightarrow C_i$: D_{C_i}.

Two approaches may be used to deliver the private key D_{C_i} to the client C_i. One is off-line approach that the server S stores the identity ID_{C_i} and the private key D_{C_i} into a smart card and sends it to the client C_i. The other is on-line approach, the server S may use the Secure Socket Layer (SSL) channel in the https mode to deliver the private key D_{C_i} to the client C_i.

Mutual authentication with key agreement phase. Whenever the client C_i wants to access the services on the server S, this phase is executed. The detail of the phase is illustrated as follows:

(1) The client C_i chooses a random number $r_{C_i} \in Z_n^*$, and computes $M = r_{C_i} \cdot P, M' = r_{C_i} \cdot D_{C_i}$. Then C_i computes $k = H_2(ID_{C_i}, T_{C_i}, M, M')$, where T_{C_i} is the current timestamp on user side.
(2) $C_i \to S : M_1 = \{ID_{C_i}, T_{C_i}, M, MAC_k(ID_{C_i}, T_{C_i}, M)\}$.
(3) After receiving M_1, S checks the validity of ID_{C_i} and the freshness of T_{C_i}. The freshness of T_{C_i} is checked by performing $T' - T_{C_i} \leqslant \Delta T$, where T' is the time when S receives the above message and ΔT is a valid time interval. If ID_{C_i} is not valid or T_{C_i} is not fresh, S aborts the current session. S computes $h_{C_i} = H_1(ID_{C_i})$, $M' = \frac{1}{x+h_{C_i}}M$ and $k = H_2(ID_{C_i}, T_{C_i}, M, M')$. Then, S checks the integrity of $MAC_k(ID_{C_i}, T_{C_i}, M)$ with the key k. S will quit the current session if the check produces a negative result. Otherwise, S chooses a random number $r_S \in Z_n^*$, and computes $W = r_S \cdot P$, $K_S = r_S \cdot M$ and the session key $sk = H_3(ID_{C_i}, T_{C_i}, T_S, M, W, K_S)$, where T_S is the current timestamp on server side.
(4) $S \to C_i : M_2 = \{ID_{C_i}, T_S, W, MAC_k(ID_{C_i}, T_S, W)\}$.
(5) Upon receiving M_2, C_i checks the integrity of $MAC_k(ID_{C_i}, T_S, W)$ with the key k. C_i will quit the current session if the integrity check fails. Otherwise, C_i computes $K_{C_i} = r_{C_i} \cdot W$ and the session key $sk = H_3(ID_{C_i}, T_{C_i}, T_S, M, W, K_{C_i})$.

3. Cryptanalysis of He et al.'s scheme

With many attractive properties, such as provision of forward secrecy, resistance to known key attack and high efficiency, that their scheme possesses being presented, He et al.'s scheme seems to be quite decent and promising. Furthermore, their protocol is claimed to be provably secure against active adversaries under the assumptions of k-CAA1 and MAC in the random oracle model. In this section, however, we will demonstrate that it fails to achieve all the claimed security goals and is still far from a secure protocol to be applicable for practical use before the identified flaws are appropriately fixed.

3.1. Reflection attack

Note that, Client C_i's login request M_1 is symmetric to server S's response M_2 and the only differences between them are the timestamps and DH values. This symmetry leads to the following reflection attack:

Step 1. An adversary $\mathcal{A}$ intercepts and blocks a valid login request message $M_1 = \{ID_{C_i}, T_{C_i}, M, MAC_k(ID_{C_i}, T_{C_i}, M)\}$ of client C_i from the public communication channel.
Step 2. $\mathcal{A}$ sends $M_2 = \{ID_{C_i}, T_{C_i}, M, MAC_k(ID_{C_i}, T_{C_i}, M)\}$ to C_i timely.

It is easy to see that, upon receiving the server S's (actually $\mathcal{A}$'s) response, client C_i will accept the server's (actually $\mathcal{A}$'s) response, because the response message is computed with the correct key k and the integrity check will succeed. As a result, $\mathcal{A}$ successfully masquerades the server S. Since C_i cannot actually authenticate S, He et al.'s scheme fails to provide mutual authentication, contradicting the claimed security goal.

It should be noted that, although $\mathcal{A}$ can fool C_i into believing himself is indeed communicating with the server, $\mathcal{A}$ cannot further compute the corresponding session key sk. That is, this attack breaches the basic goal of mutual authentication with limited damage.

3.2. Parallel session attack

In the authentication phase, consider the scenario of the parallel session attack where an adversary $\mathcal{A}$ without knowing client C_i's password wants to masquerade as a legal user by creating a valid login message from the eavesdropped communication between S and C_i. When C_i wants to login the remote server S, C_i sends the login message $M_1 = \{ID_{C_i}, T_{C_i}, M, MAC_k(ID_{C_i}, T_{C_i}, M)\}$ to the server S, where T_{C_i} is the current timestamp. If M_1 is valid, the identification of C_i is successful and S will respond $M_2 = \{ID_{C_i}, T_S, W, MAC_K(ID_{C_i}, T_S, W)\}$ to C_i, where T_S is the current timestamp on server side.

Suppose that $\mathcal{A}$ has intercepted and blocked this response message, i.e., $M_2 = \{ID_{C_i}, T_S, W, MAC_k(ID_{C_i}, T_S, W)\}$. Then $\mathcal{A}$ impersonates as C_i to immediately start a new session with S by sending fabricated login request $M_1^* = \left\{ID_{C_i}, T_{C_i}^* = T_S, M^* = W, MAC_k^*(ID_{C_i}, T_{C_i}, M) = MAC_k(ID_{C_i}, T_S, W)\}\right\}$ to server S. Suppose the login message M_1^* arrives at S at time T', it will pass the verification of S for the following two reasons:

(1) The relationship $T' - T_{C_i}^* \leqslant \Delta T$ will hold with a high probability. On the one hand, the new session is timely started by $\mathcal{A}$. On the other hand, the time-delay in wide-area networks (especially mobile networks) is unpredictable and fluctuates largely, ΔT often is larger than the timespan of a full round trip [23–25].
(2) The integrity check will succeed because $MAC_k^*(ID_{C_i}, T_{C_i}, M)$ is actually computed by server S with the correct k.

This attack is summarized in Fig. 1. It also should be noted that, although the session key sk computed by S is unknown to the adversary $\mathcal{A}$, it reveals a potential weakness that may be employed to carry out other subtle (or unintended) damages to its application systems. For example, suppose that the services provided by S have to be charged, and S will begin to charge C_i once C_i has successfully logged into S. Since $\mathcal{A}$ can masquerade C_i to login S, he will fool S into wrongly charging C_i. What's more, such an impersonation attack may distort the system logs. In practice, false system logs may have great negative effects upon the application systems. Hence, the above parallel session attack is practical and indeed rather damaging.

3.3. Failure of protecting the user's anonymity

In many cases (e.g., electronic voting, electronic auditing, secret online-order placement, etc.), it is of utmost importance to preserve user privacy so that the adversary cannot trace user activity. For example, in e-commerce applications, once the identity of a user is leaked, the sensitive personal information about this user, such as preferences, social circle and shopping patterns, can be learned and abused for marketing purposes [26]. A protocol with user anonymity prevents an adversary from acquiring an individual's sensitive personal information, such as lifestyles and current location, by analyzing the login information. Therefore, user anonymity is a glamorous and important feature that a practical authentication scheme should achieve [17,20,27].

Instead of a single "anonymity" property, there are a number of different aspects of anonymity, such as sender identity protection, sender un-traceability, blender anonymity, and sender k-anonymity [28]. Different application systems may implement quite different (and sometimes even contradictory) notions of anonymity. As for remote authentication schemes, since it is necessary to know who is providing what services, the identity of the server shall be disclosed to the public. Accordingly, user anonymity is defined

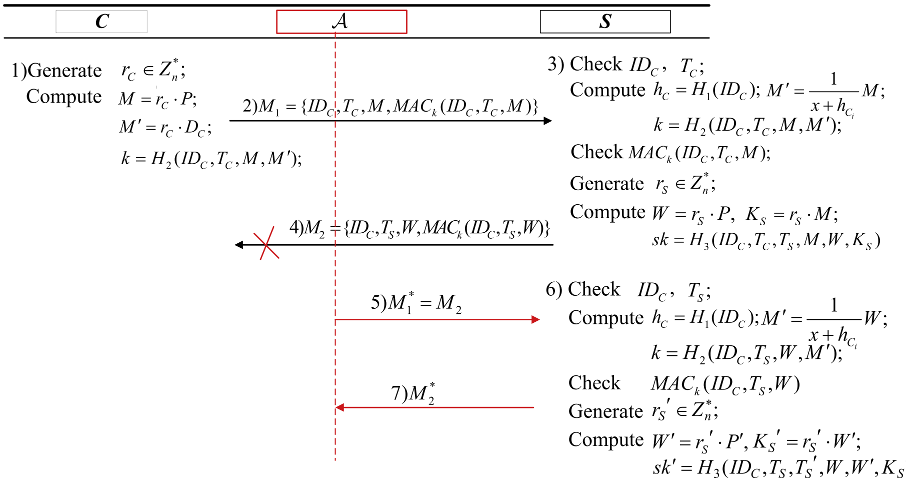
C
$\mathcal{A}$
S
1)Generate $r_C \in Z_n^*$;
Compute $M = r_C \cdot P$;
$M' = r_C \cdot D_C$;
$k = H_2(ID_C, T_C, M, M')$;
2)$M_1 = \{ID_C, T_C, M, MAC_k(ID_C, T_C, M)\}$
3) Check ID_C, T_C;
Compute $h_C = H_1(ID_C)$; $M' = \frac{1}{x+h_{C_i}} M$;
$k = H_2(ID_C, T_C, M, M')$;
Check $MAC_k(ID_C, T_C, M)$;
Generate $r_S \in Z_n^*$;
Compute $W = r_S \cdot P$, $K_S = r_S \cdot M$;
$sk = H_3(ID_C, T_C, T_S, M, W, K_S)$
4)$M_2 = \{ID_C, T_S, W, MAC_k(ID_C, T_S, W)\}$
5)$M_1^* = M_2$
6) Check ID_C, T_S;
Compute $h_C = H_1(ID_C)$; $M' = \frac{1}{x+h_{C_i}} W$;
$k = H_2(ID_C, T_S, W, M')$;
7)M_2^*
Check $MAC_k(ID_C, T_S, W)$
Generate $r_S' \in Z_n^*$;
Compute $W' = r_S' \cdot P'$, $K_S' = r_S' \cdot W'$;
$sk' = H_3(ID_C, T_S, T_S', W, W', K_S')$

Fig. 1. Parallel session attack on He et al.'s scheme.

against the public rather than the server because the server has to identify and check the validity of the user for accounting and/or billing purposes [29]. Hence, in such situations, user anonymity basically means user identity protection (i.e., initiator/sender anonymity), more precisely, it means the adversary could not have any knowledge of the real identity of the initiator but may know whether two independent conversations originate from the same (unknown) entity. Comparatively, a more desirable anonymity property is initiator un-traceability (i.e., sender un-traceability) [30], which means that the adversary can figure out neither who the initiator is nor whether two conversations originate from the same (unknown) initiator.

In He et al.'s scheme, the user's identity *ID* is transmitted in plain-text, which may leak the identity of the logging user once the login messages were eavesdropped. In other words, without employing any effort an adversary can distinguish and recognize the particular transactions performed by the specific user C_i. Moreover, the user's identity *ID* is static in all the login phases, which may facilitate the attacker to trace out the different login request messages belonging to the same user and to derive some information related to the user C_i. In a word, neither initiator anonymity nor initiator un-traceability can be preserved in their scheme.

3.4. The clock synchronization problem

It is well known that, remote user authentication schemes employing timestamps to provide message freshness may still suffer from replay attacks as the transmission delay is unpredictable in existing networks [31]. In addition, clock synchronization is difficult and expensive in existing network environments, especially in wireless and mobile networks [24] and distributed networks [23,25,32]. Hence, these schemes employing the timestamp mechanism to resist replay attacks are not suitable for mobile applications [17,33]. In He et al.'s scheme, this principle is violated.

4. Proof analysis and security enhancement

Now a paradox arises: How can a protocol that was provably secure later be found insecure? To answer this question, we investigate into the formal security proof of He et al.'s protocol. After figuring out the flaw in the reasoning of the proof, we propose a simple but effective fix to the protocol.

4.1. Flaw in He et al.'s security proof

Once again, our attacks demonstrate that having a formal security model and a "provably secure" protocol in that model is no panacea and the protocol may perform poorly in reality since the security proof only works well within the model of the security. The failure of a formal proof is unavoidable if an insufficient security model, which do not entirely capture all the attacks that might be considered realistic, is adopted. Recent examples include [8,9,18,34]. Formal security proofs based on the intractability of the computational hard problems (e.g., computational Diffie–Hellman problem and elliptic curve discrete logarithm problem) seem to be routine and similar. However, experience over the past two decades has told us that, formal methods are often misused and security proofs are very intricate and prone to errors. Particularly, security proofs often fail to cover various kinds of attacks, such as impersonation attack [9], man-in-the-middle attack [35], reflection attack [36] and parallel session attack [37], due to the symmetry (or algebraic relationships) between the protocol transcripts.

As stated in Section 1, He et al.'s scheme is equipped with a claimed proof of its security in a formal model of communication and adversarial capabilities. The model that they adopted is a variant of the Bellare–Rogaway 1993 model (BR93 model) [38]. Here we refer the reader to [38] for details. After defining the security model, He et al.'s proof proceeds with a standard reduction argument. Since both our reflection attack and parallel session attack break the goal of mutual authentication (called "*MA-security*" in [22]) but not the session key security, we only focus on the proof of "*MA-security*" for the protocol.

In its reductionist approach, the proof of server-to-client authentication (*S2C* in short) assumes an active adversary $\mathcal{A}$ who breaks the *S2C-security*. A forger F is then constructed from $\mathcal{A}$ to break the assumption of MAC function. In He et al.'s arguments, the probability that F has called its MACing oracle to produce the flow $\{ID_{C_i}, T_S, W, MAC_k(ID_{C_i}, T_S, W)\}$ is negligible, because F could only have called the MACing oracle on this flow in two cases: (1) "on behalf of a responder Π_S^t which received $MAC_k(ID_{C_i}, T_{C_i}, M)$ as its own first flow"; (2) "on behalf of an initiator $\Pi_{C_i}^u$ with $u \neq s$ which also chooses $MAC_k(ID_{C_i}, T_{C_i}, M)$ and needs to decide whether or not it should accept". Indeed, the probabilities of these two cases are negligible. However, there is a significant gap in this reasoning as another (the third) case has been overlooked. $\mathcal{A}$ in our reflection attack can take $\Pi_{C_i}^s$ as the MACing oracle by just sending back $M_2 = \{ID_{C_i}, T_S = T_{C_i}, W = M, MAC_k(ID_{C_i}, T_{C_i}, M)\}$ to $\Pi_{C_i}^s$ timely. In other words, F have called MACing oracle on $\{ID_{C_i}, T_S, W\}$ on behalf of $\Pi_{C_i}^s$, and this probability is not negligible. This means that even equipped with $\mathcal{A}$ who breaks the *S2C-security* with a non-negligible advantage, F is unable to obtain a non-negligible advantage in invalidating the assumption of MAC function. Consequently, the proof fails. And a similar mistake is made in the security proof of *C2S-security*, which makes their security claim of resistance to parallel session attack invalid.

4.2. Security enhancement

Both our reflection attack and parallel session attack exploit the symmetric structure of the transcripts exchanged between communicating parties, and are based on sending the authentication message directly back to the sender. One may argue that, to defeat the reflection attack (resp. the parallel session attack), it is enough for the user (resp. the server) to check if the received message is the same as the one that has been sent in the valid time interval. We call this approach as "the session-specific information checking strategy". However, this strategy will not work. Firstly, in reality, it is common practice to keep session-specific parameters independent for each session, so it is difficult or even impossible for one session to check on information of other sessions [39]. Secondly, when implementing security protocols, it is recommended to erase secret temporary values from memory as soon as they are no longer required, in case the system is compromised by a Trojan horse or other forms of malicious code [36,40]. Thirdly, to check whether the received message has already been sent in the valid time interval, one has to record all the messages sent in this time interval, which is quite expensive and undesirable for resource-constrained client devices and heavily-loaded servers.

Here is another point that ought to be noted. One may think that the reflection attack and the parallel session attack could be easily identified and thus thwarted by checking whether the (received) timestamp has been sent before. It is not difficult to see that, this "timestamp-checking" strategy is just a special case of the above "session-specific information checking strategy". The point here is, this strategy is effective to withstand these two attacks but inefficient to be adopted in resource-constrained mobile environments. Let us demonstrate its impracticability. On the one hand, to check whether the (received) timestamp has been sent before, the server has to record all the timestamps that are involved during the valid time interval for each active user. That's to say, the server has to maintain (search, read and write) a large timestamp table for verification, which will put a tremendous burden on the server and greatly impair the server's capability of processing concurrent access requests. On the other hand, to check whether the (received) timestamp has been sent before, the mobile user has to record all the timestamps that are involved in the valid time interval, which is undesirable because the mobile devices are often in shortage of storage capacity.

These two attacks support the observation that two-flow authenticated key establishment protocols that do not contain asymmetry in the computations of the transcripts will not meet the basic security requirement for mutual authentication in a distributed computing environment [41]. A straight-forward way to counter these two attacks is to break the symmetry between $MAC_k(ID_{C_i}, T_{C_i}, M)$ and $MAC_k(ID_{C_i}, T_S, W)$. Here we adopt the approach suggested by Wang et al. [37], i.e. including both the sender and responder's identity and their roles into the calculation of MACs:

$$MAC = \begin{cases} MAC_k(ID_{C_i}, ID_S, T_{C_i}, M) & \text{for } C_i, \\ MAC_k(ID_S, ID_{C_i}, T_S, W) & \text{for } S. \end{cases}$$

Now consider the attack scenario illustrated in Fig. 1. If the adversary sends $MAC_k(ID_S, ID_{C_i}, T_S, W)$ back to S, the attack will be detected as soon as the server checks the equality $MAC_k(ID_S, ID_{C_i}, T_S, W) \overset{?}{=} MAC_k(ID_{C_i}, ID_S, T_S, W)$. As a result, the proposed attacks will no longer be valid against the patched protocol.

With our patch implemented, He et al.'s protocol now can attain its main goal of mutual authentication without additional computation host, communication and storage overhead. However, as for the other identified defects, there seems no simple countermeasure but radical revisions: (1) To achieve user anonymity, pseudonym-identity (or dynamic identity) technique shall be employed [10,19]; (2) to cope with the clock synchronization problem, timestamps shall be replaced with nonces to provide message freshness, which will result in one more message flow.

5. Conclusion

In this paper, we have analyzed an efficient and provably secure ID-based scheme for mobile client–server environment on ECC without the MapToPoint function introduced by He et al. in 2012. Although their scheme is equipped with a claimed proof of provable security, we have pointed out that, besides suffering from the problems of clock synchronization and user privacy violation, He et al.'s protocol actually cannot achieve the basic goal of mutual authentication by demonstration its vulnerabilities to reflection attack and parallel session attack. To explicate this seemingly paradoxical situation, we have investigated into He et al.'s proof of security and uncovered the flaw in the reasoning of the proof. We further presented a simple but effective fix to resist against the reflection attack and parallel session attack, while the other identified defects can only be eliminated by radical revisions. As for limitations, our improvement still cannot preserve user privacy and thus a natural direction for further research is to design a secure and efficient remote user authentication scheme for mobile devices with user anonymity.

Acknowledgments

We are grateful to the four anonymous reviewers for their valuable suggestions and constructive comments that highly improve the readability and completeness of the paper. This research was partially supported by the National Natural Science Foundation of China (NSFC) under Grant No. 61170241, and the Specialized Foundation Project for Science and Technology Innovation Talent of Harbin City under No. 2012RFXXG086.

References

[1] L. O'Gorman, Comparing passwords, tokens, and biometrics for user authentication, Proceedings of the IEEE 91 (12) (2003) 2021–2040.

[2] J. Bonneau, C. Herley, P. van Oorschot, F. Stajano, The quest to replace passwords: a framework for comparative evaluation of web authentication schemes, in: Proceedings of the IEEE Symposium on Security and Privacy (S&P 2012), IEEE Computer Society, San Francisco, CA, 2012, pp. 553–567, http://dx.doi.org/10.1109/SP.2012.44.

[3] L. Lamport, Password authentication with insecure communication, Communications of the ACM 24 (11) (1981) 770–772.

[4] C. Chang, T. Wu, Remote password authentication with smart cards, IEE Proceedings-Computers and Digital Techniques 138 (3) (1991) 165–168.

[5] J. Shen, C. Lin, M. Hwang, A modified remote user authentication scheme using smart cards, IEEE Transactions on Consumer Electronics 49 (2) (2003) 414–416.

[6] I. Liao, C. Lee, M. Hwang, A password authentication scheme over insecure networks, Journal of Computer and System Sciences 72 (4) (2006) 727–740.

[7] C. Lee, M. Hwang, I. Liao, Security enhancement on a new authentication scheme with anonymity for wireless environments, IEEE Transactions on Industrial Electronics 53 (5) (2006) 1683–1687.

[8] J. Xu, W. Zhu, D. Feng, An improved smart card based password authentication scheme with provable security, Computer Standards & Interfaces 31 (4) (2009) 723–728.

[9] K. Yeh, C. Su, N. Lo, Y. Li, Y. Hung, Two robust remote user authentication protocols using smart cards, Journal of Systems and Software 83 (12) (2010) 2556–2565.

[10] 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.

[11] Y. Wang, Password protected smart card and memory stick authentication against off-line dictionary attacks, in: D. Gritzalis, S. Furnell, T.M. (Eds.), Proceedings of the 27th IFIP International Information Security and Privacy Conference (SEC 2012), IFIP AICT, vol. 37, Springer Boston, 2012, pp. 489–500.

[12] D. Klein, Foiling the cracker: a survey of, and improvements to, password security, in: Proceedings of the 2nd USENIX Security Workshop, 1990, pp. 5–14.

[13] S. Halevi, H. Krawczyk, Public-key cryptography and password protocols, ACM Transactions on Information and System Security 2 (3) (1999) 230–268.

[14] D. 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.

[15] J. Yang, C. Chang, An id-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem, Computers & Security 28 (3) (2009) 138–143.

[16] S. Islam, G. Biswas, Design of improved password authentication and update scheme based on elliptic curve cryptography, Mathematical and Computer Modelling (2012), http://dx.doi.org/10.1016/j.mcm.2011.07.001.

[17] S. Islam, G. Biswas, A more efficient and secure id-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem, Journal of Systems and Software 84 (11) (2011) 1892–1898.

[18] R. Wang, W. Juang, C. Lei, Robust authentication and key agreement scheme preserving the privacy of secret key, Computer Communications 34 (3) (2011) 274–280.

[19] S. Wu, Y. Zhu, Q. Pu, Robust smart-cards-based user authentication scheme with user anonymity, Security and Communication Networks 5 (2) (2012) 236–248.

[20] D. Wang, C. Ma, L. Shi, Y.-H. Wang, On the security of an improved password authentication scheme based on ECC, in: B. Liu, M. Ma, J. Chang (Eds.), Proceedings of International Conference on Information Computing and Applications, LNCS, vol. 7473, Springer, Berlin/Heidelberg, 2012, pp. 181–188.

[21] T. Truong, M. Tran, A. Duong, Improvement of the more efficient and secure id-based remote mutual authentication with key agreement scheme for mobile devices on ECC, in: Proceedings of the 26th International Conference on Advanced Information Networking and Applications Workshops, IEEE, 2012, pp. 698–703.

[22] D. He, J. Chen, J. Hu, An id-based client authentication with key agreement protocol for mobile client–server environment on ECC with provable security, Information Fusion 13 (3) (2012) 223–230.

[23] D. Mills, Internet time synchronization: the network time protocol, IEEE Transactions on Communications 39 (10) (1991) 1393–1482.

[24] A. Giridhar, P. Kumar, Distributed clock synchronization over wireless networks: algorithms and analysis, in: Proceedings of the 45th IEEE Conference on Decision and Control, IEEE, 2006, pp. 4915–4920.

[25] J. Han, D. Jeong, A practical implementation of ieee 1588–2008 transparent clock for distributed measurement and control systems, IEEE Transactions on Instrumentation and Measurement 59 (2) (2010) 433–439.

[26] F. Bao, R. Deng, Privacy protection for transactions of digital goods, in: S. Qing, T. Okamoto, J. Zhou (Eds.), Proceedings of International Conference on Information and Communications Security (ICICS 2001), LNCS, vol. 2229, Springer, Berlin/Heidelberg, 2001, pp. 202–213.

[27] C.G. Ma, D. Wang, S.D. Zhao, Security flaws in two improved remote user authentication schemes using smart cards, International Journal of Communication Systems (2012), http://dx.doi.org/10.1002/dac.2468.

[28] D. Hughes, V. Shmatikov, Information hiding, anonymity and privacy: a modular approach, Journal of Computer Security 12 (1) (2004) 3–36.

[29] K. Mangipudi, R. Katti, A secure identification and key agreement protocol with user anonymity (SIKA), Computers & Security 25 (6) (2006) 420–425.

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

[31] L. Gong, A security risk of depending on synchronized clocks, ACM SIGOPS Operating Systems Review 26 (1) (1992) 49–53.

[32] R. Baldoni, A. Corsaro, L. Querzoni, S. Scipioni, S. Piergiovanni, Coupling-based internal clock synchronization for large-scale dynamic distributed systems, IEEE Transactions on Parallel and Distributed Systems 21 (5) (2010) 607–619.

[33] C. Chang, C. Lee, A secure single sign-on mechanism for distributed computer networks, IEEE Transactions on Industrial Electronics 59 (1) (2012) 629–637.

[34] J. Tsai, T. Wu, K. Tsai, New dynamic id authentication scheme using smart cards, International Journal of Communication Systems 23 (12) (2010) 1449–1462.

[35] K. Shim, Cryptanalysis of two identity-based authenticated key agreement protocols, IEEE Communications Letters 16 (4) (2012) 554–556.

[36] J. Nam, S. Kim, D. Won, Security analysis of a nonce-based user authentication scheme using smart cards, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences 90 (1) (2007) 299–302.

[37] S. Wang, Z. Cao, K. Choo, L. Wang, An improved identity-based key agreement protocol and its security proof, Information Sciences 179 (3) (2009) 307–318.

[38] 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.

[39] R. Bird, I. Gopal, A. Herzberg, P. Janson, S. Kutten, R. Molva, M. Yung, Systematic design of a family of attack-resistant authentication protocols, IEEE Journal on Selected Areas in Communications 11 (5) (1993) 679–693.

[40] R. Canetti, H. Krawczyk, Analysis of key-exchange protocols and their use for building secure channels, in: B. Pfitzmann (Ed.), Advances in Cryptology – EUROCRYPT 2001, Lecture Notes in Computer Science, vol. 2045, Springer, Berlin/Heidelberg, 2001, pp. 453–474.

[41] S. Blake-Wilson, D. Johnson, A. Menezes, Key agreement protocols and their security analysis, in: M. Darnell (Ed.), Proceedings of the Sixth IMA International Conference on Cryptography and Coding, LNCS, vol. 1355, Springer, Berlin/Heidelberg, 1997, pp. 30–45.