ISSN 1392–124X (print), ISSN 2335–884X (online) INFORMATION TECHNOLOGY AND CONTROL, 2015, T. 44, Nr. 2

An Anonymous and Lightweight Authentication Scheme for Mobile Devices

Kuo-Hui Yeh

The Department of Information Management, National Dong Hwa University,
Hualien 974, Taiwan,
e-mail: khyeh@mail.ndhu.edu.tw

crossref http://dx.doi.org/10.5755/j01.itc.44.2.8335

Abstract. In this paper, we present a lightweight authentication scheme designed to enable mobile devices to achieve robust client-anonymity and computation efficiency. Instead of the heavy encryption and decryption modules of Elliptic Curve Cryptography (ECC), we adopt the key agreement operation of ECC as the core technique in the proposed anonymous authentication scheme. This eliminates significant computation cost and thus does not exceed the inherent resource-limitations on mobile devices. Security analyses are conducted to guarantee the robustness of the proposed authentication scheme. Moreover, when we implement our proposed scheme, the demo-system we have named AuthDroid, into the Android system, the implementation results demonstrate a practical execution time, e.g. 149.7 microseconds, on an Android-based smartphone, i.e. HTC ONE X, to complete the whole authentication procedure of AuthDroid.

Keywords: Android; Anonymity; Authentication; ECC; Mobile devices; Security.

1. Introduction

With the universalness of intelligent mobile devices (e.g. android phones, i-phones and tablets), myriad value-added services have been developed to benefit consumers and businesses. As mobile users use their mobile devices, the value-added applications often require a public wireless connection to provide full functionality. To support the smooth and effective running of applications it is necessary to design suitable operating systems for intelligent mobile devices such as smartphones and tablets. Among current operating system technologies, iOS [10] and Android [1] are two of the most popular system architectures and lead the way in terms of the successful development of numerous value-added applications for intelligent mobile devices. These two system architectures have stimulated the generation of numerous online application markets all over the world. Recently, a critical challenge for mobile devices revolves around how to resolve the tension between the convenience provided by mobile applications and the data transmission threat to mobile clients. Mobile application services, in general, need to possess specific secure transmission designs. Famous mobile software platforms such as WhatsApp [25], Line [12], Skype [17] and Facebook [7], have embedded lightweight authentication schemes to support secure transmission over the Internet. Without appropriate defense mechanisms, the utilization and transmission of user information may be insecure against malicious adversaries on the Internet.

Handheld mobile devices, in general, are embedded with resource-limited hardware components and are restricted by the power saving requirement for device runtime. This limits the ability of a mobile device to run full-fledged security functions, such as real-time antivirus and firewall software connected with the backend application servers. As a result, the properties of computation efficiency (representing the power consumption) need to be carefully investigated when designing new mobile applications or secure communication mechanisms for mobile devices. In addition, mobile applications often interact with sensitive personal data, such as chat records or data retrieved from local sensors such as GPS, cameras, microphones, and accelerometers. Consumers (or business clients) do not always know whether their data is being processed properly or not. Moreover, mobile service applications (or transactions) involving sensitive personal information are becoming more and more common. It is highly risky for consumers if such sensitive data is transmitted within a public network environment without any protection mechanisms in place. From these observations, we believe that a secure communication mechanism with robust data confidentiality and strong privacy protection is a critical requirement for mobile devices.

In this paper, we describe a secure communication protocol for use among mobile devices (or applications) through a trusted third-party. We assume that the data transmission environment is public and insecure, and each legal communication entity intends to negotiate a session key agreement for secure transmissions among numerous mobile devices with robust client privacy and low computation overhead. The primary goal of this scheme is to prevent sensitive personal information from being disclosed during transmission and to facilitate communication between applications by phone users (or external mobile services). Taking into account the need to find a balance between the resource constraint impinging on mobile devices and the desired level of security, we adopt the key agreement property of ECC in the proposed authentication scheme instead of relying on heavy encryption/decryption modules. In addition, we implement a demo-system, called *AuthDroid*, on the Android system to demonstrate the feasibility and practicability of our proposed authentication scheme.

2. Related works

As facilities and computers are linked together, primarily via Internet, resources can be easily shared and exploited. Since the authentication protocol was introduced by Lamport [15], a range of authentication protocols have been developed to ensure legitimate access to resources and secure data exchange. In the following section we discuss research which is the most relevant to our study.

Single-sign on (SSO) is a concept of authentication technology that enables each remote user to access multiple services via a single credential in a distributed computer network. In 2010, Chang and Lee [3] presented a SSO based authentication mechanism for a distributed network environment. Based on their proposed security arguments, the robustness of the mechanism seems to be appropriate, however, two attacks, i.e. a user impersonation attack and a credential recovering attack, can be invoked successfully on Chang and Lee's protocol [23]. Next, Juang et al. [14] proposed a smart card based authenticated key agreement scheme. The authors provided a method to protect user identity during each authentication session. The security of Juang et al.'s mechanism is based on ECC and symmetric cryptosystem. Nevertheless, Sun et al. [18] showed that the security of Juang et al.'s protocol is doubtful and proposed a remedy to eliminate all identified weaknesses. Later, Li et al. [16] demonstrated that Juang et al.'s scheme cannot provide initiator untraceability, and proposed a solution to strengthen the security and efficiency of Juang et al.'s scheme. Unfortunately, Tsai et al. [19] found that Li et al.'s scheme is vulnerable to de-synchronization attack. In addition, the secret update mechanism of Li et al.'s scheme is not well-designed and the scalability of the registration table in thus not efficient. For these reasons, Tsai et al. demonstrated an anonymous authentication scheme. The distinguishing feature of Tsai et al.'s scheme is that the server does not need to maintain a registration table, which makes the scheme suitable for a large scale of service level. Nevertheless, as Tsai et al.'s protocol is a single server based scheme, the scalability may be limited in multi-server environments.

In 2012, Wang [24] analyzed the trust between a smart card and card reader. The possibility of user compromise attacks was examined in the situation where an adversary possesses a stolen smart card in conjunction with a compromised user password. The authors then presented important findings under multiple kinds of password based schemes and different attacker types. Namely, the security of both the symmetric key based scheme and the public key HMQV-based scheme is limited, , while the public key ID-based scheme (PSCAb) and the public key based scheme with password validation data at server (PSCAV) are both secure. Chen et al. [6] subsequently proposed a password-based authentication scheme without smart cards; unfortunately, the researches [8] and [13] have proved that Chen et al.'s scheme is not secure. Next, several advancements were made by Tsai et al. in recent years, with two group key agreement protocols [20, 22] being developed for mobile architecture and one password-based authentication scheme [21] being proposed for a multi-server environment. Chang et al. [5] next proposed an authentication scheme to resist against user traceability attack. The authors claimed that their scheme could withstand various attacks such as user impersonation attacks, server counterfeit attacks, replay attacks, and password guessing attacks. However, Chang et al.'s scheme is insecure against server counterfeit attacks, user impersonation attacks, and man-in-the-middle attacks. In addition, their scheme cannot provide user-untraceability. In 2013, Huan et al. [9] identified two specific scenarios for password authentication in distributed systems, i.e. (1) adversaries with pre-computed data stored in a smart card, and (2) adversaries with different data (with respect to different time slots) stored in a smart card. Two attacks were shown to be practicable via implementing attacks on the two authentication schemes, and corresponding countermeasures were proposed.

3. The proposed authentication scheme

In this section, we demonstrate our proposed authentication scheme, in which a trusted registration center, *RC,* is required. The server and *RC* do not require the maintenance of any registration table for the authentication of each communication entity, including the user or the server. In addition, both the user and the server need to store only one set of public parameters, i.e. $\{p, E_p, P, P_{RC}, n, h(.), h_1(.)\}$ and $\{p, E_p, P, P_{RC}, n, h(.), h_2(.)\}$, respectively, published by

RC. Note that RC chooses an elliptic curve E_p over a finite field Z_p with a large prime p, and three one-way hash functions $h(.)$, $h_1(.)$ and $h_2(.)$. Then, RC chooses a generator point P with order n, and computes its private key x_{RC} and its public key $P_{RC}= x_{RC}\times P$. Finally, RC publishes and shares $\{p, E_p, P, P_{RC}, n, h(.), h_1(.)\}$ and $\{p, E_p, P, P_{RC}, n, h(.), h_2(.)\}$ with the user and the server, individually.

Registration Phase of the service provider S_j: In the registration phase, the server S_j will receive the parameters $\{p, E_p, P, P_{RC}, n, h(.), h_2(.)\}$ publicized by RC. In addition, the identity, i.e. SID_j, of S_j is public.

Step1. S_j sends his/her identity SID_j to RC via a secure channel.

Step2. Once obtaining SID_j, RC computes $h(h(SID_j)||y_{RC})$, and sends $h(h(SID_j)||y_{RC})$ to SID_j via a secure channel, where y_{RC} is the secret generated by RC.

Step3. Now S_j possesses $\{p, E_p, P, P_{RC}, n, h(.), h_2(.)\}$ and $h(h(SID_j)||y_{RC})$.

Registration Phase of the user U_i: In the registration phase, the user's mobile device, such as a tablet or a smart phone, has been configured with public parameters $\{p, E_p, P, P_{RC}, n, h(.), h_1(.)\}$. When the user U_i wants to register on RC, the following steps are performed.

Step1. U_i inputs his/her password PW_i to compute $h(PW_i||b)$, where b is a random number generated by the user's mobile device. Next, U_i sends his/her identity ID_i and $h(PW_i||b)$ to RC via a secure channel.

Step2. Upon receiving $\{ID_i, h(PW_i||b)\}$, RC calculates $V=h(h(ID_i)||z_{RC})\oplus h(PW_i||b)$ and sends V to U_i via a secure channel, where z_{RC} is the secret generated by RC.

Step3. When U_i gets V, U_i stores $\{V, b\}$ into the user's mobile device.

Pre-computation Phase: We launch this phase once the session key at the current session is agreed upon successfully. That is, once the session key is established between U_i and S_j, the user's mobile device will generate a new random number N_1 and compute $e_{Ui}=N_1\times P$ and $c_{Ui}=N_1\times P_{RC}$. After that, $\{e_{Ui}, c_{Ui}, N_1\}$ will be stored in the user's mobile device. Furthermore, the server chooses a random number N_3 to compute $e_{Sj}=N_3\times P$ and $c_{Sj}=N_3\times P_{RC}$, and maintains $\{e_{Sj}, c_{Sj}, N_3\}$ for the next authentication.

Login Phase (Fig. 1): When U_i wants to access S_j, U_i searches the public identity SID_j of service provider S_j, and inputs his/her identity ID_i and password PW_i.

Step1. U_i derives $h(h(ID_i)||z_{RC})$ from $V\oplus h(PW_i||b)$, and calculates $C_1=(\ h(ID_i)\ ||\ (h(h(ID_i)||z_{RC})\oplus N_2)\ )\oplus h_1(c_{Ui})$, where N_2 is a random number. Next, Ui sends $\{C_1, e_{Ui}\}$ to S_j according to the public identity SID_j and the corresponding network address.

Step2. After getting $\{C_1, e_{Ui}\}$, S_j computes $C_2=(\ h(SID_j)\ ||\ (h(h(SID_j)||y_{RC})\oplus N_4)\ )\oplus h_2(c_{Sj})$ and sends $\{C_1, e_{Ui}, C_2, e_{Sj}\}$ to RC. Note that N_4 is a random number.

Step3. Once RC receives $\{C_1, e_{Ui}, C_2, e_{Sj}\}$, RC performs the following equations.

(1) Derive $(\ h(ID_i)\ ||\ (h(h(ID_i)||z_{RC})\oplus N_2)\ )$ from $C_1\oplus h_1(x_{RC}\times e_{Ui})=C_1\oplus h_1(x_{RC}\times N_1\times P)$, where x_{RC} is stored by RC.

(2) Calculate $h(h(ID_i)||z_{RC})$ with $h(ID_i)$ derived in (1) and the secret z_{RC} maintained by RC.

(3) Retrieve N_2 with $h(h(ID_i)||z_{RC})$ calculated in (2) and $(h(h(ID_i)||z_{RC})\oplus N_2)$ derived in (1).

(4) Derive $(\ h(SID_j)\ ||\ (h(h(SID_j)||y_{RC})\oplus N_4)\ )$ from $C_2\oplus h_2(x_{RC}\times e_{Sj})=C_2\oplus h_2(x_{RC}\times N_3\times P)$.

(5) Compute $h(h(SID_j)||y_{RC})$ with $h(SID_j)$ derived in (4) and the secret y_{RC} stored at RC.

(6) Retrieve N_4 with $h(h(SID_j)||y_{RC})$ calculated in (5) and $(h(h(SID_j)||y_{RC})\oplus N_4)$ derived in (4).

(7) Generate a random number N_5.

(8) Calculate $N_5\times e_{Ui}=N_5\times N_1\times P$, $N_5\times e_{Sj}=N_5\times N_3\times P$, $C_3=h_1(h(SID_j), N_5\times e_{Ui}, N_5\times e_{Sj}, N_2)$, and $C_4=h_2(N_5\times e_{Ui}, N_5\times e_{Sj}, N_4)$.

(9) Send $\{N_5\times e_{Ui}, N_5\times e_{Sj}, C_3, C_4\}$ to S_j.

Step4. Upon obtaining $\{N_5\times e_{Ui}, N_5\times e_{Sj}, C_3, C_4\}$, S_j computes $h_2(N_5\times e_{Ui}, N_5\times e_{Sj}, N_4)$ and compares the result with the received value C_4. If it holds, S_j forwards $\{N_5\times e_{Ui}, N_5\times e_{Sj}, C_3, C_5\}$ to U_i, where $SK=N_3\times N_5\times e_{Ui}=N_3\times N_5\times N_1\times P$ and $C_5=h(N_5\times e_{Sj}, N_5\times e_{Ui}, SK)$. After that, U_i calculates $h_1(h(SID_j), N_5\times e_{Ui}, N_5\times e_{Sj}, N_2)$ and compares the result with the received value C_3. If both of these values are equal, U_i computes $SK=N_1\times N_5\times e_{Sj}=N_1\times N_5\times N_3\times P$, and verifies C_5. If this verification is successful, U_i performs $C_6=h(SK, N_5\times e_{Sj}, N_5\times e_{Ui})$ and sends it to S_j. Finally, S_j examines the validity of C_6. If it holds, the session key SK is successfully agreed upon by U_i and S_j.

4. Security analyses

In this section, we introduce the security analyses of our proposed authentication scheme. Before doing so, it is important to define the adversary model. In a public communication environment, there is a probabilistic polynomial-time attacker A who controls the communication links and the schedule of protocol events. A has the following abilities: message modification, transmission injection, and protocol event re-scheduling. Mapping to the real world, A can

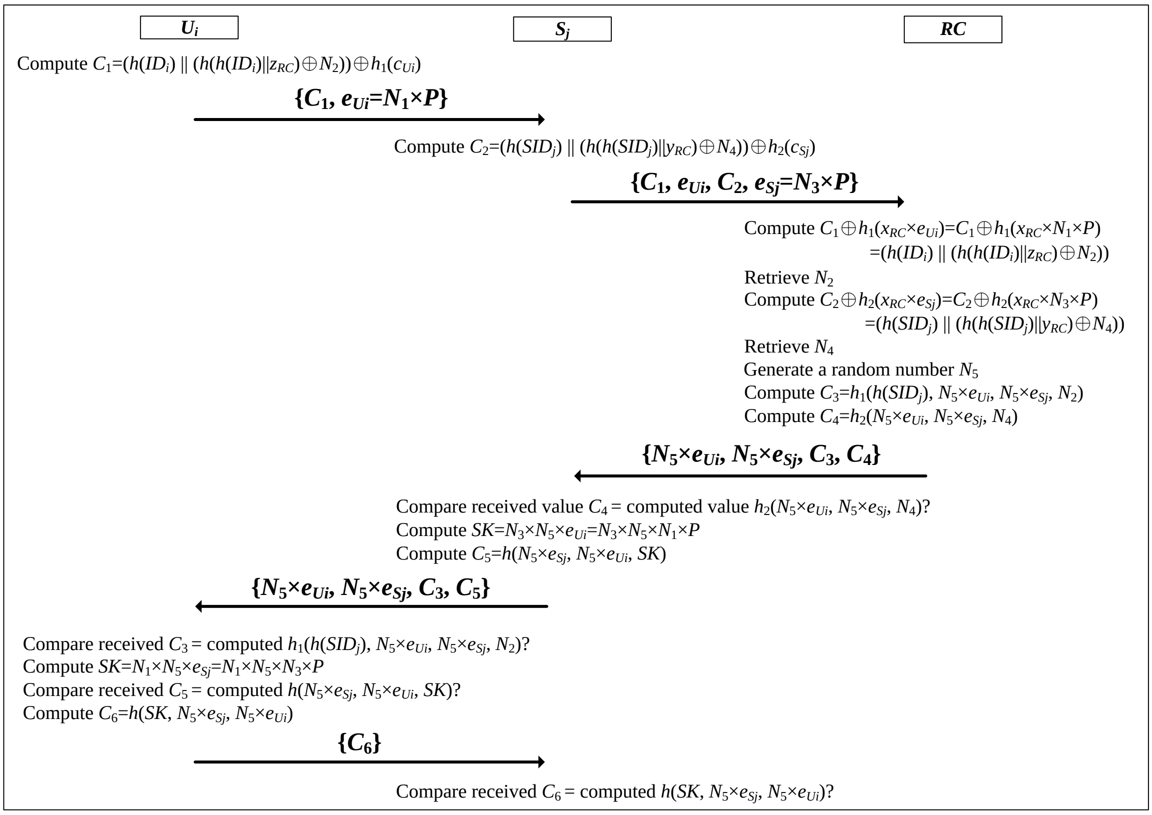

Figure 1. The proposed authentication scheme

be a legitimate user, service provider or system administrator who is legitimate and verified in our system, and possesses the authorization of some system functionalities. On the other hand, there exists another probabilistic polynomial-time attacker A, who is restricted to delivering messages generated from one of the communicating parties to the other one. In the real world, this kind of attacker can be an outsider who does not have the capability to inject or modify the transmitted messages. Note that an insider without entity verification or function authorization can also be an example of this kind of attacker.

In traditional security verification of authentication, the random oracle model is widely used to guarantee protocol robustness by showing that an attacker would require impossible behavior from the oracle or would have to solve some mathematical problem believed to be hard. On the other hand, the famous BAN logic technique is always adopted to ensure the mutual authentication property. In this section, we will first show that our proposed scheme is insecure against malicious attackers under the hardness of elliptic curve discrete logarithm. Then, we present the mutual authentication of our proposed scheme via the BAN logic technique.

Definition. Let E be an elliptic curve over a finite field F_p with a prime order q. Suppose that G is a base point over $E(F_p)$, and a (t, ε)-ECDL attacker in $E(F_p)$ is a probabilistic Turing machine Δ running in a time period t such that $\mathrm{Succ}_G^{ECDLP}(\Delta) = \Pr[\Delta(aG, bG) = abG] \geq \varepsilon$, where the probability is taken over the random values a and b. The Elliptic Curve Discrete Logarithm Problem (ECDLP) is (t, ε)-intractable if there exists no (t, ε)-attacker in $E(F_p)$. The Elliptic Curve Discrete Logarithm Assumption is the case for all polynomial t and any non-negligible ε.

Theorem 1. *Let A be an adversary against the Authenticated Key Agreement (AKA) security of our proposed authentication scheme within a time bound t, with less than q_s interactions with the communication entities, and asking q_h times $h(.)$, q_{h_1} times $h_1(.)$ and q_{h_2} times $h_2(.)$ hash-queries. Then,*

$$\mathrm{Adv}_P^{AKA}(A) = \left(\frac{q_h^2 \times q_{h_1}^2 \times q_{h_2}^2}{(2^{l+1})(2^{l_1+1})(2^{l_2+1})}\right) + \left(\frac{q_s^2}{2^{k+1}}\right) + \max\left[\left(\frac{q_{h_1}^2}{2^{l_1+1}}\right), \left(\frac{q_{h_2}^2}{2^{l_2+1}}\right)\right] + \left(\frac{3 \times q_h^2}{2^{l+1}}\right) + q_s \times \mathrm{Succ}_G^{ECDLP}(t'),$$

where $t' \leq t + q_s \times \tau_G$, and τ_G denotes the computational time for a multiplication in G with order q.

Proof. We define a sequence of games starting at the real game G_0. In each game, the adversary possesses different advantages for winning the game. Once all the games are analyzed, we then derive the possibility (or probability) of compromising our authentication scheme.

Game G_0. This is the real attack game in the random oracle models. For any game G_n, we define some events as follows. First, event E_n occurs if $b=b'$, where b is the binary bit involved in the Test-query, and b' is the output of the adversary. By this definition, we have $\mathrm{Adv}_P^{AKA}(A) = 2\Pr[E_0] - 1$. If the adversary has not stopped playing the game after q_s Send-queries lasting for more than time t, the game is terminated and a random bit b' will be chosen as the output, where q_s and t are predefined upper bounds.

Game G_1. In this game, we first simulate three hash oracles:

$h(.): \{0,1\}^* \rightarrow \{0,1\}^l$, with a hash list Λ_h.

$h_1(.): \{0,1\}^* \rightarrow \{0,1\}^{l_1}$, with a hash list Λ_{h_1}.

$h_2(.): \{0,1\}^* \rightarrow \{0,1\}^{l_2}$, with a hash list Λ_{h_2}.

All instances such as U_i and S_j can be simulated to conform to real player behavior, for Send, Execute, Reveal, Corrupt and Test-queries [4, 19]. From this simulation, we can easily see that this game is indistinguishable from a real attack unless the permutation properties of $h(.)$, $h_1(.)$ and $h_2(.)$ do not hold. According to the birthday paradox, for example, the probability of collisions happening under $h(.)$ is at most $q_h^2/2^{l+1}$. For the same reason, we have

$$|\Pr[E_1] - \Pr[E_0]| \leq \left(\frac{q_h^2}{2^{l+1}}\right) \times \left(\frac{q_{h_1}^2}{2^{l_1+1}}\right) \times \left(\frac{q_{h_2}^2}{2^{l_2+1}}\right)$$

Game G_2: In this game, we modify the game so that the adversary may guess the correct authentic values $\{C_1, C_2\}$, $\{C_3, C_4\}$, $\{C_5\}$ or $\{C_6\}$ without hash queries. Thus, games G_1 and G_2 are indistinguishable under the following probability, where the maximum bit-length among $\{C_1, C_2\}$, $\{C_3, C_4\}$, $\{C_5\}$ and $\{C_6\}$ is k.

$$|\Pr[E_2] - \Pr[E_1]| \leq \left(\frac{q_s^2}{2^{k+1}}\right)$$

Game G_3: In this game, we avoid collisions amongst the hash queries asked by the adversary to RC's ephemeral secrets, i.e. x_{RC}, y_{RC} and z_{RC}, maintained by RC. Assume that no collision has been found by the adversary for RC's ephemeral secrets. Choose two random elements $\mathrm{r} \in \{0,1\}^l$ and $\mathrm{r}_1 \in \{0,1\}^{l_1}$. If this query is directly asked by the adversary and $\{(*,\mathrm{r}),(*,\mathrm{r}_1)\} \in \Lambda_A$, where Λ_A denotes the queried list of the adversary, then we abort the game. Note that x_{RC} is involved with $h_1(.)$ and $h_2(.)$, and y_{RC} and z_{RC} are involved with only $h(.)$. The two games G_3 and G_2 are indistinguishable once the adversary causes the game to abort. Hence, we obtain

$$|\Pr[E_3] - \Pr[E_2]| \leq \max\left[\left(\frac{q_{h_1}^2}{2^{l_1+1}}\right), \left(\frac{q_{h_2}^2}{2^{l_2+1}}\right)\right] + \left(\frac{2 \times q_h^2}{2^{l+1}}\right)$$

Game G_4: This game considers the collisions amongst the hash queries asked by the adversary to the current session key SK. Choose a random set of elements $\mathrm{r}_{sk} \in \{0,1\}^l$. If $(*, \mathrm{r}_{sk})\} \in \Lambda_A$, the game is terminated. Note that SK is involved with $\mathrm{h}(.)$. In that case, games G_4 and G_3 are indistinguishable unless the adversary terminates the game. Therefore, we can derive

$$|\Pr[E_4] - \Pr[E_3]| \leq \left(\frac{q_h^2}{2^{l+1}}\right)$$

Game G_5: In this game, we simulate the executions under the random self-reducibility of *ECDLP*. Given a pair *ECDLP* instance (X, Y), where $X=\alpha A$ and $Y=\beta B$, we wish to derive Z=*ECDLP*(X, Y). With the list Λ_A, we can obtain the elliptic curve discrete logarithm secret values with the probability $1/q_s$. We thus can find the values α and β such that *ECDLP*(X, Y) = *ECDLP*(αA, βB) = *ECDLP*$(A, B)^{\alpha\beta}$. Finally, we have $|\Pr[E_5]\text{-}\Pr[E_4]| \leq q_s \times \mathrm{Succ}_G^{\mathrm{ECDLP}}(t')$ where $t' \leq t + q_s \times \tau_G$.

And this completes the proof.

Theorem 2. *The proposed authentication scheme guarantees mutual authentication.*

Proof. The mutual authentication of the proposed authentication scheme is proved via BAN logic [2]. Basic constructs and logic postulates are defined as follows. Note that in this section the symbols P and Q range over principals, X and Y range over statements, and K ranges over encryption keys (or long-term secrets).

Constructs:

- P believes X: The principal P believes that X is true.
- P sees X: Someone has sent a message containing X to P, who can read and repeat X (possibly after doing some decryption).
- P said X: P has actually sent a message including statement X at the current session of the protocol or before.
- P controls X: P has jurisdiction over X, i.e. the principal P is an authority on X and this matter should be trusted.
- fresh(X): X has not been sent in a message before the current session of the protocol.
- $P \overset{K}{\longleftrightarrow} Q$: The key K is shared between the principals P and Q.

- $P \overset{X}{\longleftrightarrow} Q$: The formula X is a secret known only to P and Q. Only P and Q may use X to prove their identities to each other.
- $\{X\}_K$: This symbol represents the formula X encrypted or protected under the key K.

Logical postulates:

- Rule 1 (the message-meaning rules): If P believes $P \overset{K}{\longleftrightarrow} Q$ and P sees $\{X\}_K$, then we postulate P believes Q said X.
- Rule 2 (the nonce-verification rule): If P believes fresh(X) and P believes Q said X, then we postulate P believes Q believes X.
- Rule 3 (the jurisdiction rule): If P believes Q controls X and P believes Q believes X, then we postulate P believes X.
- Rule 4:
  a. If P sees (X, Y) then P sees X.
  b. If P believes $P \overset{X}{\longleftrightarrow} Q$ and P sees $\{X\}_K$, then P sees X.
- Rule 5: If one part of a formula is fresh, then the entire formula must also be fresh. If P believes fresh(X), then P believes fresh (X, Y).

Assumption:

Before analyzing the authentication scheme, the assumptions are given as follows. Note that all symbols are the same as those in the proposed authentication scheme presented in Section III.

Assumption 1: U_i, RC believe $U_i \overset{z_{RC}, x_{RC}, SID_j}{\longleftrightarrow} RC$

Assumption 2: S_j, RC believe $S_j \overset{y_{RC}, x_{RC}, SID_j}{\longleftrightarrow} RC$

Assumption 3: *U_i, S_j, RC believe* fresh(N_1), fresh(N_2), fresh(N_3), fresh(N_4), fresh(N_5)

Assumption 4: U_i, S_j *believe RC* controls N_5

The concrete realization of the proposed authentication scheme:

Step 1: $U_i \rightarrow S_j \rightarrow RC$: $\{C_1, e_{Ui}, C_2, e_{Sj}\}$

Step 2: $RC \rightarrow S_j$: $\{N_5 \times e_{Ui}, N_5 \times e_{Sj}, C_3, C_4\}$

Step 3: $S_j \rightarrow U_i$: $\{N_5 \times e_{Ui}, N_5 \times e_{Sj}, C_3, C_5\}$

Step 4: $U_i \rightarrow S_j$: $\{C_6\}$

The formal analysis of mutual authentication:

1. S_j sees $\{N_5 \times e_{Ui}, N_5 \times e_{Sj}, C_3, C_4\}$.
2. S_j believes $S_j \overset{y_{RC}, x_{RC}, SID_j}{\longleftrightarrow} RC$ (From assumption 2).
3. S_j believes RC said $\{N_5 \times e_{Ui}, N_5 \times e_{Sj}, C_3, C_4\}$ ((1) & (2), Inferred by Rule 1).
4. S_j believes fresh(N_1), fresh(N_3), fresh(N_5) (From assumption 3).
5. S_j believes RC believes $\{N_5 \times e_{Ui}, N_5 \times e_{Sj}, C_3, C_4\}$ ((3) & (4), Inferred by Rule 2).
6. S_j believes RC controls $\{N_5\}$ (From assumption 4).
7. S_j believes $\{N_5 \times e_{Ui}, N_5 \times e_{Sj}, C_3, C_4\}$ ((5) & (6), Inferred by Rule 3).
8. U_i sees $\{N_5 \times e_{Ui}, N_5 \times e_{Sj}, C_3\}$.
9. U_i believes $U_i \overset{z_{RC}, x_{RC}, SID_j}{\longleftrightarrow} RC$ (From assumption 1).
10. U_i believes RC said $\{N_5 \times e_{Ui}, N_5 \times e_{Sj}, C_3\}$ ((8) & (9), Inferred by Rule 1).
11. U_i believes fresh(N_1), fresh(N_3), fresh(N_5) (From assumption 3).
12. U_i believes RC believes $\{N_5 \times e_{Ui}, N_5 \times e_{Sj}, C_3\}$ ((10) & (11), Inferred by Rule 2).
13. U_i believes RC controls $\{N_5\}$ (From Assumption 4).
14. U_i believes $\{N_5 \times e_{Ui}, N_5 \times e_{Sj}, C_3\}$ ((12) & (13), Inferred by Rule 3).

The final results are as follows.

S_j believes RC believes $\{N_5 \times e_{Ui}, N_5 \times e_{Sj}, C_3, C_4\}$ (From (5))

S_j believes $\{N_5 \times e_{Ui}, N_5 \times e_{Sj}, C_3, C_4\}$ (From (7))

U_i believes RC believes $\{N_5 \times e_{Ui}, N_5 \times e_{Sj}, C_3\}$ (From (12))

U_i believes $\{N_5 \times e_{Ui}, N_5 \times e_{Sj}, C_3\}$ (From (14))

With the four results (5), (7), (12) and (14), and the assumption of the trustworthiness of RC, both the remote user U_i and the service provider S_j can be authenticated by each other via RC. In addition, the session key SK can be perfectly constructed by U_i and S_j as only they can verify C_3, C_4, C_5, and C_6 successfully.

□

Claim 1: The proposed authentication scheme guarantees data security and session key security.

In the proposed authentication scheme, all transmitted messages $\{C_1, e_{Ui}\}$, $\{C_1, e_{Ui}, C_2, e_{Sj}\}$, $\{N_5 \times e_{Ui}, N_5 \times e_{Sj}, C_3, C_4\}$, $\{N_5 \times e_{Ui}, N_5 \times e_{Sj}, C_3, C_5\}$ and $\{C_6\}$ are well-protected via high-entropy secrets x_{RC}, y_{RC} and z_{RC} chosen by RC. Without knowing these three secrets, attackers cannot obtain any useful information from transmitted ciphertexts. In addition, as some transmitted ciphertexts such as C_3, C_4, C_5 and C_6 are involved with the hash function, it is difficult for attackers to derive any secrets such as random numbers and session key values. This is because of the irreversibility of the one way hash function. Moreover, an attacker may eavesdrop e_{Ui}, e_{Sj}, $N_5 \times e_{Ui}$ and $N_5 \times e_{Sj}$, and intend to derive the session key value. However, due to the difficulty of solving ECDLP, the protection of the session key is guaranteed. Therefore, the data confidentiality and session key security can be ensured in the proposed authentication scheme.

Claim 2: The proposed authentication scheme guarantees user anonymity.

In each session of the proposed authentication scheme, five random numbers N_1, N_2, N_3, N_4 and N_5 are generated and utilized to randomize the messages transmitted among the user, the service provider and the registration center. Without revealing the real identity in public, all the communication entities only need to know whether the involved partners are legitimate or not. In a more detailed way, in the proposed authentication scheme all the identities are transmitted in cipher format instead of plaintext and these identities will be randomized at each new session. As a result, the proposed authentication scheme can guarantee the property of user anonymity.

Claim 3: The proposed authentication scheme guarantees known-key security and forward security.

In the proposed authentication scheme, the session key $SK{=}N_3{\times}N_5{\times}N_1{\times}P$ is involved with three one-time valid random numbers, i.e. N_1, N_3 and N_5, at each session. Even if an attacker can acquire one or more previous session keys, the attacker cannot derive any useful information regarding the currently involved session key from previous session keys. That is, since the current session key is constructed with N_1, N_3 and N_5, it is hard to derive the current session key without knowing these one-time valid numbers N_1, N_3 and N_5. Hence, the proposed authentication scheme can provide known-key security. In addition, once the attacker obtains the long-term secrets x_{RC}, y_{RC} and z_{RC}, the attacker may derive the numbers N_2 and N_4. Nevertheless, the one-time valid numbers N_1, N_3 and N_5 still cannot be retrieved as they are well-protected in the values e_{Ui} and e_{Sj}. In other words, under the difficulty of solving the ECDLP *problem*, we know that these three random numbers N_1, N_3 and N_5 cannot be derived. Therefore, the proposed authentication scheme can ensure forward security.

Claim 4: The proposed authentication scheme guarantees the non-repudiation property and the resistance to man-in-the-middle based attacks such as server counterfeit attack, user impersonation attack and man-in-the-middle attack.

An attacker may issue counterfeit messages to deceive the legal communication users or the service providers. However, without the knowledge of three high-entropy secrets x_{RC}, y_{RC} and z_{RC}, and five one-time valid numbers N_1, N_2, N_3, N_4 and N_5, it is difficult for the attacker to compute legitimate request or response messages such as $\{C_1, e_{Ui}\}$, $\{C_1, e_{Ui}, C_2, e_{Sj}\}$, $\{N_5{\times}e_{Ui}, N_5{\times}e_{Sj}, C_3, C_4\}$, $\{N_5{\times}e_{Ui}, N_5{\times}e_{Sj}, C_3, C_5\}$ and $\{C_6\}$. Even if the attacker sends a previously eavesdropped message to a victim party, the verification of these old messages will fail. This is because all of these random numbers N_1, N_2, N_3, N_4 and N_5 have been used at a previous session. In addition, the verification procedures at the registration center side will help the communicating parties to prevent against man-in-the-middle based attacks. In a more detailed way, N_2 and N_4 can temporarily be represented as the legitimate pseudonyms of the user and the service provider, respectively, instead of revealing the real identities in public. Moreover, the values $h(ID_i)$ and $h(SID_j)$ retrieved by the registration center can serve as evidence for each service request. This design will result in man-in-the-middle based attacks always failing at the registration center side. Furthermore, in the case that some service conflicts happen, the maintained evidence will play a useful role in dealing with these troubles. Obviously, the proposed authentication scheme delivers the property of non-repudiation.

Based the above analyses, we present a comparison (i.e. Table 1) of our proposed protocol and other relevant schemes. In the next section, we will introduce the implementation on current mobile device to demonstrate the feasibility and practicability of our proposed scheme.

Table 1. Comparison of our proposed protocol and other schemes

	The Proposed Scheme	**Tsai et al. [19]**	**Chang et al. [5]**
Suaitable to multi-server architecture (Scabability)	Yes	No	Yes
User anonymity	Yes	Yes	No
Data Confidentiality	Yes	Yes	Yes
Mutual authentication	Yes	Yes	Yes
Resistance to user impersonation attack	Yes	Yes	No
Resistance to server counterfeit attack	Yes	Yes	No
Resistance to man-in-the middle attack	Yes	Yes	No

5. Implementation

In this section, we introduce the environment setup followed by the implementation results of the proposed authentication scheme. The overview of the implementation environment is shown in Table 2. In order to evaluate the performance of the proposed authentication scheme, we implemented a demo system, called *AuthDroid*, which is realized with JAVA and Java Elliptic Curve Cryptography project (JECC)

[11]. The client program runs on a HTC ONE X with Android version 4.1.1, and the server program runs on a cloud-based machine, called MyCloud Pro, AMD 7450 Dual-Core 2.4 G, DDR2 1.5 G, Fedora Linux 12. We next report the implementation results.

Table 2. Environment Description

User's Smartphone	HTC ONE X: 1.5 GHz, quad-core, RAM 1 GB, Android 4.1.1
Server	MyCloud Pro: AMD 7450 Dual-Core 2.4G, DDR2 1.5G, Fedora Linux 12
Development Environment	Eclipse Java EE IDE

Below are the instantiations of the cryptographic primitives involved in the implementation of *AuthDroid*:

- Hash Functions: SHA-2 (256 bits, 384 bits, 512 bits)
- ECC: Java Elliptic Curve Cryptography project (JECC)

Fig. 2 shows the client program. To initiate a login, the user needs to input his identity, password and fingerprint. Then, the user clicks the "Submit" button, and the client program starts *AuthDroid* with the server program, as shown in Fig. 3. Our implementation results show that *AuthDroid* takes about 149.7 microseconds for the client program to complete the whole authentication procedures of *AuthDroid* with the server program. We obtained this average time from 200 runs of *AuthDroid*. As the HTC ONE X is a common smartphone, our implementation reflects the practicability and feasibility of the proposed authentication scheme.

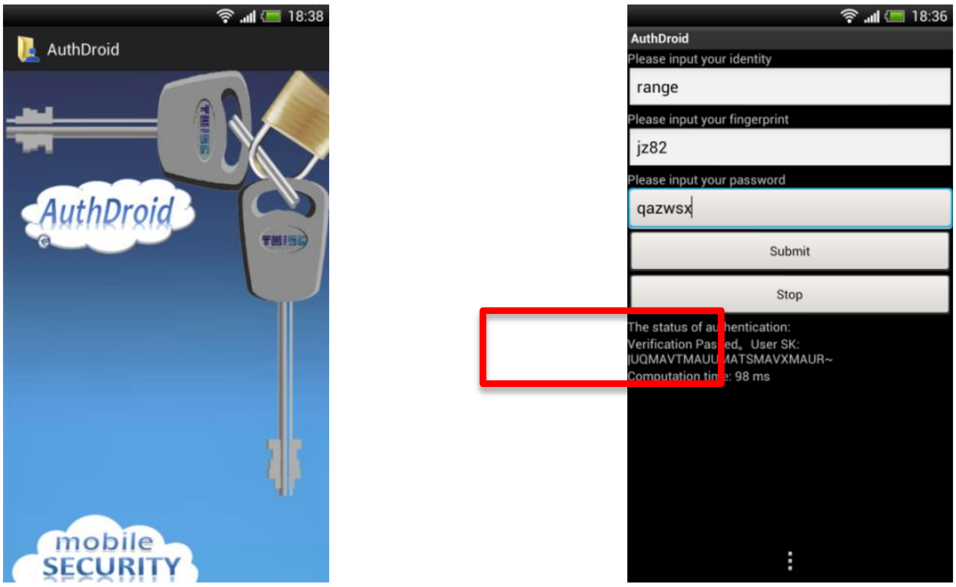

Figure. 2. Client Program on Android (HTC ONE X: 1.5 GHz, quad-core, RAM 1 GB, Android 4.1.1)

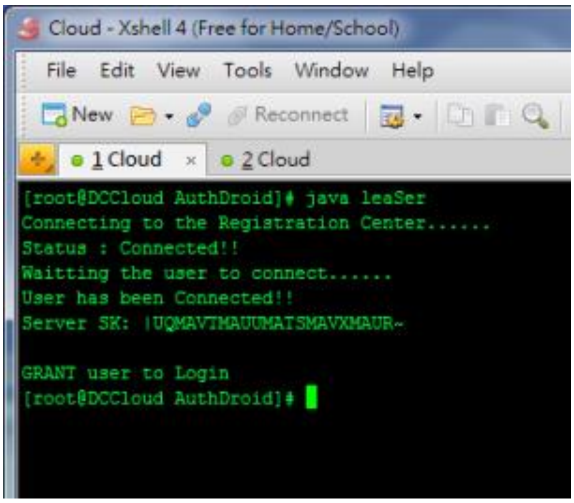

Figure 3. Server Program on the Server (MyCloud Pro with Fedora 12)

6. Conclusion

This paper presents a lightweight authentication scheme for mobile devices. The proposed authentication scheme enjoys the advantages of the convenience of password based authentication and preserves client-privacy protection as well. Formal analyses are demonstrated to promise the security robustness. We further implemented a prototype *AuthDroid* on a common Android-based smartphone, i.e. HTC ONE X, to show the practicability and feasibility of the proposed authentication scheme. The implementation results present that *AuthDroid* delivers a good performance on Android 4.1.1, where a short execution time period of 149.7 microseconds is required to mutually agree on a robust session key.

Conflict of interests

The author declares that there is no conflict of interest regarding the publication of this article.

Acknowledgments

This work was partly supported by the Taiwan Information Security Center (TWISC) and the Ministry of Science and Technology, Taiwan, under the Grants Numbered MOST 103-2221-E-259-016-MY2 and MOST 103-2221-E-011-090-MY2.

References

[1] Android, https://www.google.com/mobile/android/.

[2] **M. Burrows, M. Abadi, R. Needham**. A logic of authentication. *ACM Transactions on Computer Systems*, 1990, Vol. 8, No. 1, 18-36.

[3] **C. C. Chang, C. Y. Lee.** A Secure Single Sign-On Mechanism for Distributed Computer Networks. *IEEE Trans. on Industrial Electronics*, 2012, Vol. 59, No. 1, 629-637.

[4] **C. C. Chang, H. C. Tsai.** An Anonymous and Self-Verified Mobile Authentication with Authenticated Key Agreement for Large-Scale Wireless Networks. *IEEE Trans. on Wireless Communications*, 2010, Vol. 9, No.11, 3346-3353.

[5] **Y.-F. Chang, W.-L. Tai, H.-C. Chang.** Untraceable dynamic-identity-based remote user authentication scheme with verifiable password update. *International Journal of Communication Systems*, 2014, Vol. 27, No. 11, 3430-3440.

[6] **B.-L. Chen, W.-C. Kuo, L.-C. Wuu**. Secure Password-based Remote User Authentication Scheme without Smart Cards. *Information Technology and Control*, 2012, Vol. 41, No.1, 53-59.

[7] FACEBOOK, https://www.facebook.com/.

[8] **D. He, D. Wang, S. Wu**. Cryptanalysis and Improvement of a Password-based Remote User Authentication Scheme without Smart Cards. *Information Technology and Control*, 2013, Vol. 42, No. 2, 170-177.

[9] **X. Huan, X. Chen, J. Li, Y. Xiang, L. Xu**. Further Observations on Smart-Card-Based Password-Authenticated Key Agreement in Distributed Systems. *IEEE Transactions on Parallel and Distributed Systems*, 2014, Vol. 25, No. 7, 1767-1775.

[10] iOS, https://www.apple.com/tw/.

[11] Java Elliptic Curve Cryptography project, http://jecc.sourceforge.net/.

[12] LINE, http://line.me/zh-hant/.

[13] **Q. Jiang, J. F. Ma, G. Li, Z. Ma.** An Improved Password-based Remote User Authentication Scheme without Smart Cards. *Information Technology and Control*, 2013, Vol. 42, No.2, 150-158.

[14] **W. S. Juang, S. T. Chen, H. T. Liaw**. Robust and Efficient Password-Authenticated Key Agreement Using Smart Cards. *IEEE Trans. Industrial Electronics*, 2008, Vol. 55, No. 6, 2551-2556.

[15] **L. Lamport**. Password Authentication with Insecure Communication. *ACM Communications*, 1981, Vol. 24, No. 11, 770-772.

[16] **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. on Industrial Electronics*, 2010, Vol. 57, No. 2, 793-800.

[17] Skype, http://www.skype.com/zh_TW/.

[18] **D. Z. Sun, J. P. Huai, J. Z. Sun, J. W. Zhang, Z. Y. Feng.** Improvements of Juang et al.'s Password-Authenticated Key Agreement Scheme Using Smart Cards. *IEEE Trans. on Industrial Electronics*, 2009, Vol. 56, No. 6, 2284-2291.

[19] **J.-L. Tsai, N.-W. Lo, T.-C. Wu**. Novel Anonymous Authentication Scheme Using Smart Cards. *IEEE Trans. on Industrial Informatics*, 2013, Vol. 9, No. 4, 2004-2013.

[20] **J.-L. Tsai, N.-W. Lo, T.-C. Wu**. ID-Based Authenticated Group Key Agreement Protocol from Bilinear Pairings for Wireless Mobile Devices. *Ad Hoc & Sensor Wireless Networks*, 2013, Vol. 17, Issue 3-4, 221–231.

[21] **J.-L. Tsai, N.-W. Lo, T.-C. Wu.** A New Password-Based Multi-server Authentication Scheme Robust to Password Guessing Attacks. *Wireless Personal Communications*, 2013, Vol. 71, No. 3, 1977–1988.

[22] **J.-L. Tsai.** A novel authenticated group key agreement protocol for mobile environment. *Annales des Télécommunications*, 2011, Vol. 66, Issue 11-12, 663–669.

[23] **G. Wang, J. Yu, Q. Xie**. Security Analysis of a Single Sign-on Mechanism for Distributed Computer Networks. *IEEE Trans. on Industrial Informatics*, 2013, Vol. 9, No. 1, 294-302.

[24] **Y. Wang**. Password Protected Smart Card and Memory Stick Authentication against Off-Line Dictionary Attacks. In: *Proceedings of SEC 2012, IFIP AICT 376*, 2012, pp. 489–500.

[25] WhatsApp, http://www.whatsapp.com/?l=zh_tw.

Received January 2014.