Improvement of an ID-Based Deniable Authentication Protocol

Tzu-Chun Lin

Abstract—Deniable authentication protocol is an important notion that allows a receiver to identify the source of a given message, but not to prove the identity of the sender to a third party. Such property is very useful for providing secure negotiation over the Internet. In 2013, J. Kar introduced a new ID-based deniable authentication protocol based on elliptic Diffie-Hellman key agreement protocol([6]). However, E.-J. Yoon ([17]) pointed out that Kar’s protocol cannot defend sender spoofing attack and message modification attack. In this paper, we present an improved protocol based on double elliptic Diffie-Hellman scheme.

Index Terms—Deniable authentication, elliptic curves, Diffie-Hellman, bilinear pairing.

1. Introduction

The concept of deniable authentication protocol was initially introduced by C. Dwork et al.[12]. Traditional authentication protocol assures the identification of communicated parties. One of drawback is the privacy issue. In compared with the traditional authentication, the deniable authentication has two special characteristics:

1) It enables an intended receiver to identify the source of a given message.

2) The intended receiver cannot prove the identity of the sender to any third party, even if he/she fully cooperates with the third party.

Due to the above two characteristics, the deniable authentication protocol can be broadly used for online shopping, electronic voting system, e-learning system and negotiation over Internet, etc. In 1998, C. Dwork et al.[13] introduced the notion of interactive deniable authentication protocol based on concurrent zero-knowledge proof. At the time, Y. Aumann and M. Rabin[11] proposed an interactive deniable authentication protocol based on the factoring problem. However, both protocols have a common weakness, they are vulnerable to the person-in-the-middle attack. This means that a third party can impersonate the intended receiver to identify the source of a given message. In 2004, Z. Shao[16] proposed a first non-interactive deniable authentication scheme based on ElGamal signature scheme. In general, non-interactive protocols are more effective than interactive protocols because of less communication cost. As a result, many non-interactive deniable authentication protocols have been designed. Their security is based on different mathematical hard problems. However, most of these schemes have some drawbacks. For example, in 2005, R. Lu and Z. Cao proposed two non-interactive deniable authentication protocols based on bilinear pairing and integer factorization problem, respectively. These two protocols still fail to achieve the second feature of the deniable authentication scheme[12][15]. In 2006, W.-B. Lee et al.[8] proposed a deniable authentication protocol based on generalized ElGamal scheme. They claimed that its communication process is shorter than Shao’s protocol. But Y. Zhang[18] pointed out that Lee’s protocol requires more transmission bits than Shao’s protocol. In 2011, C.-Y. Liu et al.[11] showed that Shao’s protocol fails to achieve the second requirement of the deniable authentication scheme. In 2013, J. Kar proposed an ID-based deniable authentication protocol based on elliptic Diffie-Hellman key agreement protocol[6]. However, in 2015, E.-J. Yoon[17] pointed out that Kar’s protocol cannot defend so-called sender spoofing attack and message modification attack. In this note, we present an improved protocol which defends such attacks.

The remainder of this paper is organized as follows. In Section 2, we briefly some required mathematical properties. In Section3 we review Kar’s ID-based deniable authentication protocol. Our protocol with the secure analysis and the performance analysis will be presented in Sections 4-6.

2. Preliminaries

In this section, we briefly outline some basic knowledge which is used in the paper.

2.1 Elliptic Curves and Pairing

Definition 1. An elliptic curve over a finite field \( \mathbb{F}_p \) is roughly defined to be an equation of the Weierstraß form
\[ E: \ y^2 = x^3 + ax + b \]

with \( a,b \in \mathbb{F}_p \) satisfying \( 4a^3 + 27b^2 \neq 0 \).

Let \( \infty \) denote the point at infinity and let \( q \) be a power of the prime \( p \). The set \( E(\mathbb{F}_q) \) of rational points on an elliptic curve \( E \) over \( \mathbb{F}_q \)

\[ E(\mathbb{F}_q) = \{(x,y) \in \mathbb{F}_q^2 \mid y^2 = x^3 + ax + b \} \cup \{ \infty \} \]

under the addition is an abelian group.

**Definition 2.** Let \( G \) be an additively-written group and let \( G_T \) be a multiplicatively-written group of the same order. A mapping \( \hat{e} : G \times G \rightarrow G_T \) is called a bilinear pairing, if it satisfies the following conditions:

1) Bilinearity: For all \( P, Q, R \in G \), we have

\[ \hat{e}(P, R + Q) = \hat{e}(P, R) \hat{e}(P, Q), \]

\[ \hat{e}(P + R, Q) = \hat{e}(P, Q) \hat{e}(R, Q). \]

2) Non-degeneracy: If \( \hat{e}(P, R) = 1_{G_T} \), for all \( R \in G \), then \( P = 1_G \).

3) Computability: \( \hat{e} \) is efficiently computable.

Note that if a bilinear pairing is used on elliptic curves, then usually the Weil pairing or Tate pairing is to recommend. As a convenient reference for elliptic curves and bilinear pairing see [3],[13],and [14].

**2.2 Security Problems**

The security of the protocols described in the paper is dependent on the intractability of the following problems. Let \( G \) be an elliptic curve over a finite field of the order \( n \).

**Definition 3.** (Elliptic curve discrete logarithm problem (ECDLP)) Given \( Q, R \in G \), find an integer \( 0 < r < n \) such that \( R = rQ \).

**Definition 4.** (Decisional Diffie-Hellman Problem (DDHP)) Given \( P, aP, bP, cP \in G \) and. Determine whether \( c \equiv ab \mod n \).

**Definition 5.** (Hash Decisional Diffie-Hellman Problem (HDDHP)) Let \( H \) be a secure cryptographic hash function. Given \( P, aP, bP \in G \) and \( k \). Determine whether \( k = H(abP) \).

**Definition 6.** (Computational Diffie-Hellman Problem (CDHP)) Given \( P, aP, bP, cP \in G \). Compute \( abP \).

**3. Related Works**

We review the Kar’s ID-based deniable authentication protocol. Suppose that Alice (as sender) wants to send a message \( m \) to Bob (as receiver). At the beginning, some parameters are selected: A large finite field \( \mathbb{F}_p \), an elliptic curve \( E \) over \( \mathbb{F}_p \) in order to construct an additive abelian subgroup \( G_1 \) of \( E(\mathbb{F}_p) \) and a base point \( P \in G_1 \) of large order. Let \( h_1 : \{0,1\}^l \rightarrow \{0,1\}^l \) and \( h_2 : G_1 \times G_1 \rightarrow \{0,1\}^l \) be two cryptographic hash functions. Moreover, \( ID_s \) and \( ID_r \) denoted as Alice’s and Bob’s identification, respectively. The Kar’s ID-based deniable authentication protocol is described as follows.

**3.1 Kar’s Protocol**

**Key Generation:**

Alice chooses an integer \( t_s \) and sets the value \( a_s \) and the point \( Q_s \)

\[ a_s := h_1(ID_s) \oplus t_s, \quad Q_s := a_sP. \]

Alice’s private key is \( \{t_s, a_s, Q_s\} \).

Similarly, Bob chooses an integer \( t_r \) and sets the value \( a_r \) and the point \( Q_r \). In order to encrypt the information, Bob needs to add an encryption key denoted as \( \pi_e \) and the corresponding decryption key \( \pi_d \). Bob’s private key is \( \{t_r, a_r, \pi_d\} \) and public key is \( \{Q_r, \pi_e\} \).

**Send Phase:**

1) Alice concatenates \( Q_s \) with the time stamp \( T \) and encrypts it by using Bob’s public key \( \pi_e \)

\[ \bar{Q}_s := E_{\pi_e}(Q_s || T) \]

where \( E \) is denoted as the encryption function.

Alice sends the cipher \( \bar{Q}_s \) to Bob.

2) Bob decrypts it by using his private decryption key \( \pi_d \)

\[ Q_s := D_{\pi_d}(\bar{Q}_s || T) \]

where \( D \) is denoted as the decryption function.

Bob computes the session key

\[ a_1 := a_sQ_s \]

and the hashed value

\[ \beta := h_2(ID_r, Q_r, a_1). \]

Bob sends \( \{Q_r, \beta\} \) to Alice.

3) Alice computes the session key

\[ a_2 := a_rQ_r \]

and the hashed value

\[ \beta' := h_2(ID_r, Q_r, a_2). \]

Alice checks whether \( \beta' = \beta \). If it holds, Alice accepts \( Q_r \); otherwise rejects. Thus, if \( \beta' = \beta \), then Alice performs the following step.

4) Alice computes

\[ y_1 := h_1(a_2) \oplus (m || T) \]

where \( m \) is the deniable message.

Alice sends the deniable authentication information \( \psi = \{ID_s, T, y_1\} \) to Bob.

**Receive Phase:**

1) Bob recovers the message \( m \) by computing

\[ y_1 \oplus h_1(a_1) \]
and then calculates
\[ \gamma_2 := h_1(\alpha_1) \oplus (m || T). \]

2) Bob checks whether \( \gamma_1 = \gamma_2 \) and the time stamp \( T \).
If it holds then Bob accepts \( m \); otherwise rejects.

The security of Kar’s protocol is based on DDHP and HDDHP. It is an non-interactive protocol and can be easy implemented in mobile devices such as PDA, smart card, etc. The major computational cost for the performance of the protocol amount to one encryption (decryption), one point multiplication and three hash evaluations by both sender and receiver. However, E.-J. Yoon\textsuperscript{[17]} has discovered two weaknesses of Kar’s protocol. Following, we explain why Kar’s protocol cannot resist these two attacks.

3.2 Sender Spoofing Attack

Suppose that an attacker, named Eve, can intercept the pair \((\vec{Q}_e, \Psi)\). Then Eve performs the following steps:
1) Eve selects a random integer \( a_e \) and computes
\[ Q_e = a_e P \]
\[ \vec{Q}_e = E_{\pi_e}(Q_e || T_e) \]
where \((Q_e, a_e)\) is the key pair of Eve who impersonated the sender Alice and \( T_e \) is the corresponding time stamp.

2) Bob decrypts \( Q_e = D_{\pi_e}(\vec{Q}_e) \) and computes the session key
\[ \alpha_1 = a_e Q_e \]
and the hashed value
\[ \beta = h_2(ID_r, Q_r, \alpha_1). \]
Then, Bob sends \((Q_e, \beta)\) to Eve.

3) Let \( m_e \) be a fake message of Eve. After receiving \((Q_r, \beta)\), Eve computes the session key
\[ \alpha_2 = a_e Q_r \]
and
\[ \gamma_1^* = h_1(\alpha_2) \oplus (m_e || T_e). \]
Then, Eve sends the information \( \Psi = \{ID_s, T_e, \gamma_1^*\} \) to Bob.

4) Bob computes
\[ m_e = \gamma_1^* \oplus h(\alpha_1) \]
and
\[ \gamma_2 = h_1(\alpha_1) \oplus (m_e || T_e). \]
Bob verifies the validity of the equality \( \gamma_1^* = \gamma_2 \) and the time stamp \( T_e \). Because \( \gamma_1^* = \gamma_2 \), Bob will believe the trustworthiness of the attacker Eve.

3.3 Message Modification Attack

Suppose that Eve can intercept \( \Psi = \{ID_s, T, \gamma_1\} \) and the deniable message \( m \). Then
1) Eve extracts the hash value \( h_1(\alpha_2) \) by computing
\[ \gamma_1 \oplus (m || T) = h_1(\alpha_2). \]
2) Eve computes a modified
\[ \gamma_1^* := h_1(\alpha_2) \oplus (m_e || T_e), \]
where \( T_e \) is the current time stamp of Eve.
3) Eve sends the fake authenticated information \( \Psi^* = \{ID_s, T_e, \gamma_1^*\} \) to Bob.
4) After receiving \( \Psi^* \), Bob recovers the fake message
\[ m_e = \gamma_1^* \oplus h_1(\alpha_1). \]
5) Bob computes \( \gamma_2 \) by using his session key \( \alpha_1 \)
\[ \gamma_2 = h_1(\alpha_1) \oplus (m_e || T_e). \]
6) Bob verifies the validity of the equality \( \gamma_1^* = \gamma_2 \) and the time stamp \( T_e \). Because \( \gamma_1^* = \gamma_2 \), Bob will believe the trustworthiness of the attacker Eve.

4. Proposed Protocol

In this section, we present a mutual deniable authentication scheme based on double Diffie-Hellman key agreement and pairing.

Choose a cyclic subgroup \( G_1 \) of \( E(F_p) \) and two generators \( P \) and \( R \) of the order \( n \) and a bilinear pairing \( \hat{e} : G_1 \times G_1 \to G_T \), where \( G_T \) is a multiplicative group of the order \( n \). Let \( h_1 : G_1 \to \{0,1\}^l \) and \( h_2 : G_1 \times G_1 \to \{0,1\}^l \) be two cryptographic hash functions. Moreover, \( ID_s \) and \( ID_r \) denote as Alice’s and Bob’s identification, respectively.

**Key Generation:**

Alice chooses an integer \( t_s < n \) and sets the value \( a_s = h_1(ID_s) \oplus t_s < n \) and the points \( Q_s = a_s P \) and \( U_s = t_s R \). Then Alice’s public key is \( U_s \) and private key is \( \{t_s, a_s, Q_s\} \). Similarly, Bob chooses an integer \( t_r \) and sets the value \( a_e \) and the points \( Q_e \) and \( U_e \). In order to encrypt the information, Bob needs to add an encryption key denoted as \( \pi_e \) and the corresponding decryption key \( \pi_d \). Bob’s public key is \( \{Q_e, U_e, \pi_e\} \) and private key is \( \{t_e, a_e, \pi_d\} \).

**Send Phase:**

1) Alice encrypts her private key \( Q_s \) by using Bob’s public key \( \pi_e \)
\[ \vec{Q}_s := E_{\pi_e}(Q_s). \]
where \( E \) is denoted as the encryption function. Then, Alice sends the cipher \( \vec{Q}_s \) to Bob.

2) Bob decrypts it by using his private decryption key \( \pi_d \)
\[ Q_s = D_{\pi_d}(\vec{Q}_s). \]
where \( D \) is denoted as the decryption function.

Bob computes the session key \( \alpha_1 \) and the hashed value \( \beta \)
\[ \alpha_1 := a_e Q_s, \quad \beta = h_2(ID_r, Q_r, \alpha_1). \]
Then, Bob sends \(\{Q_r, \beta\}\) to Alice.

3) Alice computes a session key
\[
a_2 = a_1 Q_r
\]
and the hashed value
\[
h_2(ID_r, Q_r, a_2).
\]
Alice checks whether
\[
h_2(ID_r, Q_r, a_2) = \beta.
\]
If it holds, Alice accepts \(Q_r\); otherwise rejects. If
\[
h_2(ID_r, Q_r, a_2) = \beta,\text{ then Alice performs the following step.}
\]
4) Alice computes
\[
\beta_2 = a_2 + t_s U_r, \quad \gamma_1 = m + t_s U_r.
\]
and a bilinear pairing
\[
k = e(m, t_s U_r),
\]
where \(m \in G_s\) is the deniable message.

Then, Alice sends the deniable authenticated information \(\Psi = \{ID_r, \gamma_1, \beta_2, k\}\) to Bob.

**Receive Phase:**

After receiving \(\Psi\), Bob performs the following steps.

1) Bob computes
\[
\beta_1 := a_1 + t_s U_s
\]
and checks whether \(\beta_1 = \beta_2\).

If it holds then Bob computes
\[
\gamma_1 - t_s U_s = m, \quad e(m, t_s U_r).
\]
2) Bob checks whether \(e(m, t_s U_r) = k\). If it holds, then Bob accepts the message \(m\); otherwise rejects.

The major computational cost for the performance of the protocol amount to one encryption (decryption), two point multiplications, one hash evaluation and one pairing evaluation by both sender and receiver.

**5. Security Analysis**

In the following, we demonstrate that our protocol not only guarantees the deniability, but also resists the sender spoofing and the message modification attacks.

**Proposition 1.** The sender Alice can identify the receiver Bob.

**Proof.** Suppose that Eve impersonated Bob to communicate with Alice. Eve has to send the hashed value \(\beta\) (see the step (2) of the send-phase) to Alice. Also, Eve has to create the session key \(a_1 = a_2 Q_x\). To this end, Eve has to decrypt the cipher \(Q_x\) and solve ECDLP: \(Q_x = a_x P\). Clearly, it is difficult for Eve to derive the decryption key \(p_x\). In addition, it is well known that solving ECDLP is computationally infeasible. So no one can forge \(a_1\) without knowing the Bob’s private key \(\{\pi, a_r\}\).

**Proposition 2.** After receiving the information \(\Psi\), Bob authenticates the identity of the sender Alice by checking \(\beta_1 = \beta_2\).

**Proof.** The value \(\beta_1 = \beta_2\) is the common session key of Alice and Bob. If Eve wants to impersonate Alice, Eve needs to derive Alice’s private key \(\{a_2, t_s\}\) or Bob’s private key \(\{a_1, t_r\}\). Eve faces the ECDLP again.

**Proposition 3.** The proposed authentication protocol is deniable.

**Proof.** Bob has the same ability as Alice to generate the authenticating information \(\Psi = \{ID_r, \gamma_1, \beta_2, k\}\), since \(\gamma_2 = \gamma_1\) and \(\beta_1 = \beta_2\). Thus, it is indistinguishable the authenticated information \(\Psi\) whether was sent by Alice or forged by Bob. This means that Bob can identify the source of the information but cannot prove to any third party that the information comes from Alice.

**Proposition 4.** Let \(G\) be a cyclic subgroup of \(E(\mathbb{F}_p)\) of large order. Suppose that \(e\) is a bilinear pairing on \((G, G_\ell)\).

If \(s \in G_\ell\) and \(A \in G\) such that \(s = e(C, A)\) and if \(s\) and \(A\) are known, then the probability for finding \(C\) in \(G\) is negligible.

**Proof.** Since \(e(aP, bR) = e(P, R)^{ab}\), for nonzero integers \(a\) and \(b\), the problem is also the discrete logarithmic problem.

**Proposition 5.** The protocol can withstand a sender spoofing attack.

**Proof.** Suppose that Eve intercepts the cipher \(Q_x\) and the authenticated information \(\Psi = \{ID_r, \gamma_1, \beta_2, k\}\). Eve selects a random integer \(t_e\) to compute
\[
a_e = h_1(ID_s) \oplus t_e, \quad Q_e = a_e P, \quad Q_e = E_{a_e}(Q_e).
\]
Thus, Bob computes the points \(Q_x\) and \(a_1 = a_2 Q_x\). Let \(m_e\) be a fake message from Eve. Eve creates
\[
a_e = a_e Q_r, \quad \beta_e = a_e + X, \quad \gamma_e = m_e + X, \quad k^* = e(m_e, X),
\]
where \(X\) may be the point \(t_s U_r\) or \(t_e U_s\). Eve sends \(\Psi^* = \{ID_s, \gamma_e, \beta_e, k^*\}\) to Bob. After receiving \(\Psi^*\), Bob computes \(\beta_1\) and \(e(\gamma_e - t_s U_r, t_s U_s)\), and checks whether \(\beta_1 = \beta_e\) and \(e(\gamma_e - t_s U_r, t_s U_s) = k^*\). Only when both the equations hold, Bob will accept the message \(m_e\).

**Proposition 6.** The protocol can withstand a message modification attack.

**Proof.** Suppose that Eve intercepts the authenticated information \(\Psi = \{ID_r, \gamma_1, \beta_2, k\}\).

Case 1. If Eve substitutes \(\{m_e, \gamma_1', k^*\}\) for \(\{m_1, \gamma_1, k\}\), where \(k^* = e(m_e, t_s U_r)\) and \(\gamma_1' = m_1 + t_s U_r\). After receiving the information \(\Psi^* = \{ID_r, \gamma_1', \beta_2, k^*\}\) from Eve, Bob obtains a queer message \(m^* = \gamma_1' - t_s U_r \neq m_e\), since \(t_s U_r\) (or, \(t_e U_s\)) \(\neq t_s U_s\). Afterwards, Bob computes the
bilinear pairing $\hat{\ell}(m', t, U_s)$. Thus, Bob rejects the message $m_e$, since the value of the bilinear pairing $\hat{\ell}(m', t, U_s) \neq k^*$. 

Case 2. If Eve substitutes $\{m, y'_1, k\}$ for $\{m, y_1, k\}$. After receiving the information $w^* = \{I_{y'}, y'_1, y_2, k\}$ from Eve, Bob computes the fake message $m_e = y'_1 - t_s U_s$ by using his private key $t_r$ and Alice’s public key $U_s$. Bob will discover that $\hat{\ell}(m_e, t, U_s) \neq k$. Also, Bob rejects the message.

Proposition 7. (Completeness) If a sender and a receiver follow the proposed protocol to negotiate with each other, the receiver can identify the source of a message.

Proof. It can be seen that the sender and receiver share the common session secret $U := t_s t_r$. Therefore, the receiver can identify the source of the message $m$ according to $y_1 = y_2$.

6. Comparison

In this section, we give a performance comparison of our protocol and other three deniable protocols in [6], [10], and [12]. The Table 1 shows the computational cost of four deniable protocols. For convenience, the following notation is used: $T_h$ is the time for executing a hash function; $T_i$ is the time for executing a modular inverse operation; $T_{ip}$ is the time for executing a bilinear pairing operation; $T_{pm}$ is the time for executing a point multiplication operation; $T_{en}$ is the time for executing an encryption; $T_{de}$ is the time for executing a decryption.

It is well known that pairing operations are more expensive than point multiplications. For example, if the order $|G_1|$ is a 160-bit prime and $q$ is a 512-bit prime, then the performance simulation result in [2] indicates that one $T_{bp}$ is about 7 times that of one $T_{pm}$ and 15 times that of one $T_h$, respectively. The other operations applied in the protocols of Table 1 are negligible compared to them. Thus, the total computational cost (no $T_{en}$ and $T_{bp}$) of the protocols [6], [10], [12] and our’s are approximately $5.14 T_h$, $20.3 T_h$, $19.33 T_h$ and $21.3 T_h$, respectively.

In Table 2, we give a comparison of security. Kar’s protocol [6] is insecure. Lu et al.’s protocol [15] fails to achieve the second requirement of the deniable authentication scheme [15].

7. Conclusions

In this paper, we presented a novel non-interactive deniable authentication protocol based on elliptic Diffie-Hellman and bilinear pairing. It guarantees mutual and deniable authentication as well as confidentiality. To the best of our knowledge, only a few of the deniable protocols have a mechanism for mutual authentication. In 2013, J. Kar [6] proposed an efficient mutual deniable authentication protocol. However, this protocol still has some security vulnerabilities. It cannot resist sender spoofing attack and message modification attack. As can be seen from Table 2, the proposed protocol is more secure than other protocols.

<table>
<thead>
<tr>
<th>Table 1: Computational cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schemes</td>
</tr>
<tr>
<td>Sender</td>
</tr>
<tr>
<td>[6]</td>
</tr>
<tr>
<td>[10]</td>
</tr>
<tr>
<td>[12]</td>
</tr>
<tr>
<td>Our’s</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2: Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schemes</td>
</tr>
<tr>
<td>Sender</td>
</tr>
<tr>
<td>[6]</td>
</tr>
<tr>
<td>[10]</td>
</tr>
<tr>
<td>[12]</td>
</tr>
<tr>
<td>Our’s</td>
</tr>
</tbody>
</table>

References


**Tru-Chun Lin** received the Ph.D. in Mathematics from Goettingen University, Germany. She is an associate professor with the Department of Applied Mathematics of Feng Chia University, Taichung. Her current research interests include commutative algebras, invariant theory of finite groups, and public key cryptography.