ID-based Deniable Authentication Protocol based on Diffie-Hellman Problem on Elliptic Curve

Jayaprakash Kar
Faculty of Computing and Information Technology, Department of Information Systems
King Abdul Aziz University, KSA
jayaprakashkar@yahoo.com

(Received July 19, 2009; revised and accepted Aug. 25, 2009)

Abstract

Deniable authentication protocol enables a receiver to identify the true source of a given message, but not to prove the identity of the sender to a third party. This property is very useful for providing secure negotiation over the Internet. This paper describes a secure identity based deniable authentication protocol whose security is based on comutational infeasibility of solving Diffie-Hellman Problem on Elliptic Curve (ECDHP).

deniable authentication, ECDLP, ECDHP, HDDHP

1 Introduction

Nowadays, authentication had emerged to be an essential communication process in key establishment. Authentication can be realized by the use of digital signature in which the signature (signers private key) is tied to the signer as well as the message being signed. This digital signature can later be verified easily by using the signer's public key. Hence, the signer will not be able to deny his participation in this communication. Generally, this notion is known as non-repudiation. However, under certain circumstances such as electronic voting system, online shopping and negotiation over the Internet, the non-repudiation property is undesirable [20]. It is important to note that in these applications, the sender's identity should be revealed only to the intended receiver. Therefore, a significant requirement for the protocol is to enable a receiver to identify the source of a given message, and at the same time, unable to convince to a third party on the identity of the sender even if the receiver reveal his own secret key to the third party. This protocol is known as deniable authentication protocol.

In the past several years, numerous deniable authentication protocols have been proposed but many of them have also been proven to be vulnerable to various cryptanalytic attacks [5] [13] [14] . The concept of deniable authentication protocol was initially introduced by Dwork et al. [7], which is based on the concurrent zero knowledge proof. However, this scheme requires a timing constraint. Not only that, the proof of knowledge is also time-consuming [6] [19]. Another notable scheme which was developed by Aumann and Rabin [1] is based on the intractability of the factoring problem, in which a set of public data is needed to authenticate one bit of a given message. Few years later, Deng et al. [6] have proposed two deniable authentication schemes based on Aumaan and Rabins scheme. The proposed schemes are based on the intractability of the factoring problem and the logarithm problem. However, in 2006, Zhu et al. [14] have successfully demonstrated the Man-in-the-Middle attack against Aumann and Rabins scheme and this indirectly results in an insecure implementation of Deng et al.s schemes. In 2003, Boyd and Mao [10]have proposed another two deniable authenticated key establishment for Internet protocols based on elliptic curve cryptography. These schemes are believed to be able to solute the complexity of computation and appear to be more efficient than others but their vulnerability to KCI attack has been exploited by Chou et al. [4] in 2005. Besides that, Fan et al.have proposed a simple deniable authentication protocol based on Diffie-Hellman key distribution protocol in 2002. Unfortunately, in 2005, Yoon et al. [13] have pointed out that their protocol suffers from the intruder masquerading attack and subsequently proposed their enhanced deniable authentication protocol based on Fan et al.'s scheme.

Deniable authentication protocol is a new security authentication mechanism. Compared with traditional authentication protocols, it has the following two features:

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

2) However, the intended receiver can not prove to any third party the identity of the sender.

2 Preliminaries

2.1 Notations

We first introduce common notations used in this paper as follows.

- p is the order of underlying finite field.
- F_p is the underlying finite field of order p.
- E is an an elliptic curve defined on finite field F_p with large order.
- G is the group of elliptic curve points on E.
- P is a point in $E(F_p)$ with order n , where n is a large prime number.
- $\mathcal{H}(\cdot)$ is a secure one-way hash function.
- $\|$ denotes concatenation operation between two bit stings.
- S be the Sender with identity ID_s, $ID_s \in \{0,1\}^*$.
- R be the Receiver with identity ID_r, $ID_r \in \{0,1\}^*$.

2.2 Diffie-Hellman Problem

This section briefs overview of Computational Diffie-Hellman (CDH) problem, Decisional Diffie-Hellman and Hash Diffie-Hellman problem in $\mathbb{G}$.

Definition 2.1 Diffie-Hellman Problem*: Let $(q, \mathbb{G}, P)$ be a 3-tuple generated by polynomial time algorithm $\mathcal{G}(k)$,and let $a, b \in \mathbb{Z}_q^*$, the CDH problem in $\mathbb{G}$ is as follows: Given (P, aP, bP), compute abP. The (t, ϵ)-CDH assumption holds in $\mathbb{G}$ if there is no algorithm $\mathcal{A}$ running in time t such that*

$$\mathbf{Adv}_{\mathbb{G}}^{CDH}(\mathcal{A}) = \ Pr[\mathcal{A}(P, aP, bP) = \ abP] \geq \epsilon$$

where the probability is taken over all possible choices of (a, b).

$\underline{\mathbf{Exp}_{\mathcal{G}(k)}^{CDH}}$

1) $(\mathbb{G}, q, P) \leftarrow \mathcal{G}(1^k)$
2) $a, b, c \leftarrow \mathbb{Z}_q^*$
3) $U_1 = \ aP, U_2 = \ bP$
4) if $W = \ abP$ return 1 else return 0

Definition 2.2 Decisional Diffie-Hellman Problem *:Let $(q, \mathbb{G}, P)$ be a 3-tuple generated by polynomial time algorithm $\mathcal{G}(k)$,and let $a, b, c \in \mathbb{Z}_q^*$, the DDH problem in $\mathbb{G}$ is as follows: Given (P, aP, bP, cP), decide whether it is a Diffie-Hellman tuple.*

Definition 2.3 Hash Decisional Diffie-Hellman Problem *:Let $(q, \mathbb{G}, P)$ be a 3-tuple generated by polynomial time algorithm $\mathcal{G}(k)$,$\mathcal{H} : \{0,1\}^* \to \{0,1\}^l$ be a secure cryptographic hash function, whether l is a security parameter, and let $a, b \in \mathbb{Z}_q^*, h \in \{0,1\}^l$, the HDDH problem in $\mathbb{G}$ is as follows: Given (P, aP, bP, h), decide whether it is a hash Diffie-Hellman tuple $((P, aP, bP, \mathcal{H}(abP))$. If it is right, outputs 1; and 0 otherwise. The (t, ϵ)-HDDH assumption holds in $\mathcal{G}$ if there is no algorithm $\mathcal{A}$ running in time at most t such that*

$$\mathbf{Adv}_{\mathbb{G}}^{HDDH}(\mathcal{A}) = \ |Pr[\mathcal{A}(P, aP, bP, \mathcal{H}(abP) = 1] - Pr[\mathcal{A}(P, aP, bP, h) = \ 1]| \geq \epsilon$$

where the probability is taken over all possible choices of (a, b, h).

3 Model of ID-based Deniable Authentication Protocol

An ID-based deniable authentication protocol (IBDAP) consists of the following four algorithms: **Setup**, **Extract**, **Send** and **Receive** [2] [3]. The functions of each are described as follows.

- **Setup**: On input of the security parameter 1^k the PKG (Private Key Generator) uses this algorithm to produce a pair (params, master-key), where params are the global public parameters for the system and master-key is the master secret key kept secretly by PKG. We assume that params are publicly known so that we do not need to explicitly provide them as input to other algorithms.
- **Extract**: On input of an identity i and the master secret key master-key, the PKG uses this algorithm to compute a public-secret key pair (pk_i, sk_i) corresponding to i.
- **Send**: The sender S uses this algorithm with input (m, sk_S, pk_R) to output a deniable authentication message $\tilde{m}$, where pk_R is the public key of the receiver R.
- **Receive**: The receiver R uses this algorithm with input $(\tilde{m}, m, pk_S, pk_R)$ to output 1 if the deniable authentication message $\tilde{m}$ is valid or 0 otherwise. The above algorithms must have the following consistency requirement. If

  $\tilde{m} \leftarrow$ **Send**(m,sk_S, pk_R), then we must have
  $1 \leftarrow$ **Receive**($\tilde{m}$, m, pk_S, pk_R).

3.1 Security model

This subsection describes about security notions of ID-based deniable authentication protocol [2] [3].

Player. Let $P = \ \{\mathcal{P}_0, \mathcal{P}_1, \ldots \mathcal{P}_n\}$ be a set of players who may be included in the system. Each player $\mathcal{P}_i \in P$

get his public-secret key pair (pk_i, sk_i) by providing his identity i to the **Extract** algorithm. A player $\mathcal{P}_i \in P$ is said to be fresh if $\mathcal{P}_i$'s secret key sk_i has not been revealed by an adversary; while if $\mathcal{P}_i$s secret key sk_i has been revealed, $\mathcal{P}_i$ is then said to be corrupted. With regard of the unforgeability against chosen-message attacks, we define the security notion via the following game played by a challenger and an adversary.

[**Game 1**]

- Initial: The challenger runs Setup to produce a pair $(params, master - key)$, gives the resulting $params$ to the adversary and keeps the master-key secretly.
- Probing: The challenger is probed by the adversary who makes the following queries.
- Extract: The challenger first sets $\mathcal{P}_0, \mathcal{P}_1$ to be fresh players, which means that the adversary is not allowed to make Extract query on $\mathcal{P}_0$ or $\mathcal{P}_1$. Then, when the adversary submits an identity i of player $\mathcal{P}_i, (i = 0, 1)$, to the challenger. The challenger responds with the public-secret key pair (pk_i, sk_i) corresponding to i to the adversary.
- Send: The adversary submits the requests of deniable authentication messages between $\mathcal{P}_0$ and $\mathcal{P}_0$. The challenger responds with deniable authentication messages with respect to $\mathcal{P}_0$ (resp. $\mathcal{P}_1$) to $\mathcal{P}_1$ (resp $\mathcal{P}_0$).
- Forging: Eventually, the adversary outputs a valid forgery $\tilde{m}$ between $\mathcal{P}_0$ and $\mathcal{P}_1$. If the valid forgery $\tilde{m}$ was not the output of a Send query made during the game, we say the adversary wins the game.

Definition 3.1 (Unforgeability). *Let A denote an adversary that plays the game above. If the quantity $Adv^{UF}_{IBDAP}[A] = Pr[Awins]$ is negligible we say that the ID-based deniable authentication protocol in question is existentially unforgeable against adaptive chosen-message attacks.*

To capture the property of deniability of deniable authentication protocol, consider the following game run by a challenger.

[**Game 2**]

- Initial: Let $\mathcal{P}_0$ and $\mathcal{P}_1$ be two honest players that follow the deniable authentication protocol, and let $\mathcal{D}$ be the distinguisher that is involved in the game with $\mathcal{P}_0$ and $\mathcal{P}_0$.
- Challenging: The distinguisher $\mathcal{D}$ submits a message $m \in \{0,1\}^*$ to the challenger. The challenger first randomly chooses a bit $b' \in \{0,1\}^*$, then invokes the player P_b to make a deniable authentication message $\tilde{m}$ on m between $\mathcal{P}_0$ and $\mathcal{P}_1$. In the end, the challenger returns $\tilde{m}$ to the distinguisher $\mathcal{D}$.
- Guessing: The distinguisher $\mathcal{D}$ returns a bit $b \in \{0,1\}^*$. We say that the distinguisher $\mathcal{D}$ wins the game if $b = b'$.

Definition 3.2 (Deniablity). *Let D denote the distinguisher that is involved the game above. If the quantity $Adv^{DN}_{IBDAP}[D] = |Pr[b = b'] - \frac{1}{2}|$ is negligible we say that the ID-based deniable authentication protocol in question is deniable.*

4 Lu et al.'s Simple Deniable Authentication Protocol

Definition 4.1 *A bilinear parameter generator Gen is a probabilistic algorithm that takes a security parameter k as input and out-puts a 5-tuples $(q, \mathbb{G}, \mathbb{G}_T, e, g)$ as the bilinear parameters, including a prime number q with $|q| = k$, two cyclic groups $\mathbb{G}, \mathbb{G}_T$ of the same order q, an admissible bilinear map $e : \mathbb{G}X\mathbb{G} \rightarrow \mathbb{G}_T$ and a generator g of $\mathbb{G}$.*

This protocol also involves two entities: a sender S and an intended receiver R. Let $(q, \mathbb{G}, \mathcal{G}_T, e, g)$ be a 5-tuples generated by the bilinear parameter generator $Gen(k)$, and let $\mathcal{H} : \{0,1\}^* \rightarrow \{0,1\}^l$ be a secure cryptographic hash function, where l is a security parameter. The public key and private key pairs of the sender S and the receiver R are (Y_s, x_s) and (Y_r, x_r) respectively, where $x_s x_r \in \mathbb{Z}_q^*$ and $Y_s = g^{x_s}, Y_r = g^{x_r}$.

- **Step 1**: The sender S chooses a random number $u \in \mathbb{Z}_q^*$ and computes $U = g^u, U' = Y_r^u$ rand then sends his identity ID_s and U to the receiver R.
- **Step 2**: R chooses a random number $v \in \mathbb{Z}_q$ and computes $V = g^v, V' = Y_s^v$. R also uses his private key x_r to compute $U'' = U^{x_r}$, $h_1 = \mathcal{H}(ID_r, U, U'', V)$ and sends (V, h_1) to S.
- **Step 3**: S checks the equality $h_1 = \mathcal{H}(ID_r, U, U'', V)$. If it holds, S is authenticated and V will be accepted, otherwise rejected, since $U' = U'' = g^{ux_r}$.
- **Step 4**: When S wants to send a deniable message $M \in \{0,1\}^l$, he first computes $V'' = V^{x_s}$ and sends both h_2 and h_3 to R, where $h_2 = \mathcal{H}((V'') \oplus M$ and $h_3 = \mathcal{H}(ID_s, V, V'', M)$
- **Step 5**: R recovers M by computing $h_2 \oplus \mathcal{H}(V')$ and verifies its validity by checking equality $h_3 = \mathcal{H}(ID_s, V, V', M)$. If it holds, M will be accepted, otherwise rejected, since $V' = V'' = g^{x_s} v$.

5 Proposed Protocol

Security of the proposed deniable authentication protocol is based on the Computational Diffie-Hellman problem (CDHP), Decisional Diffie-Hellman Problem (CDHP) and Hash Diffie-Hellman problem (HDHP). The protocol involves two entities : a sender S and a intended receiver R. It follows the followings steps.

- **Setup** Let $\mathcal{H} : \{0,1\}^* \rightarrow \{0,1\}^l$ be a secure cryptographic hash function which is of collision. In the proposed protocol the sender has a certificate issued by the certificate authority (CA). The CA contains the public key (π_{pub}) of the Receiver, and the signature of CA for the certificate. The sender can obtain (π_{pub}) and verify the validity of it. The private key (π_{prv}) of receiver is kept secret.

- **Extract** During the extraction phase, the sender S with identity $ID_s \in \{0,1\}^*$ select t_s randomly from $[1, n-1]$ and computes the following

$$a_s = \mathcal{H}(ID_s) \oplus t_s \tag{1}$$

$$Q_s = a_s \cdot P \tag{2}$$

The key pair is (Q_s, a_s). Then concatenate Q_s with the time stamp $T \in \mathbb{Z}_q^*$. Encrypts the concatenated value $(Q_s \| T)$ using receiver's public key π_{pub}.

$$\tilde{Q_s} = E_{\pi_{pub}}(Q_s \| T)$$

Similarly the receiver R with identity $ID_r \in \{0,1\}^*$ selects random number $t_r \in [1, n-1]$. Then computes the following:

$$a_r = \mathcal{H}(ID_r) \oplus t_r \tag{3}$$

$$Q_r = a_r \cdot P \tag{4}$$

So the key pairs of receiver R is (a_r, Q_r).

- **Send** It follows the following steps.

  1) **Step 1**: During this phase the sender S sends the cipher $\tilde{Q_s}$ to the the receiver R. After getting, R will decrypt using his own private key π_{prv} as

  $Q_s = D_{\pi_{prv}}(\tilde{Q_s})$, where D denotes decryption algorithm.

  2) **Step 2** Receiver R use the calculated value a_r from Eq.(3). Computes the session key α_1 and the hashed value as

  $$\alpha_1 = a_r \cdot Q_s \tag{5}$$

  $$\beta = \mathcal{H}(ID_r, Q_r, \alpha_1) \tag{6}$$

  Receiver R sends the computed Q_r and β to S. Similarly after receiveing, Sender also compute the session key as

  $$\alpha_2 = a_s \cdot Q_r \tag{7}$$

  and check the equality $\beta = \mathcal{H}(ID_r, Q_r, \alpha_2)$. If it holds, S is authenticated and Q_r will be accepted, otherwise rejected, since $\alpha = \alpha_1 = a_r \cdot Q_s = a_r a_s \cdot P = a_s \cdot Q_r = \alpha_2$

  3) **Step 3**: When Sender S authenticates the deniable message $M \in \{0,1\}^l$, computes $\gamma_1 = \mathcal{H}(\alpha_2) \oplus (M \| T)$.

  4) **Step 4**: The resulting deniable authenticated message is tuples $\psi = (ID_s, T, \gamma_1)$.

  5) **Step 5**: Finally S sends ψ to the recipient R.

- **Receive**

  1) **Step 1**: After receiving $\psi = (ID_s, T, \gamma_1)$, the receiver R recovers M by computing $\gamma_1 \oplus \mathcal{H}(\alpha_2)$.Then computes $\gamma_2 = \mathcal{H}(\alpha_1) \oplus (M \| T)$.

  2) **Step 2**: Verifies the validity of the equality $\gamma_1 = \gamma_2$ and the time stamp T. If holds then accepts M otherwise reject.

The protocol is illustrated in table 1.

6 Correctness

Theorem 1 *If $\psi = (ID_s, T, \gamma_1)$ is a authentication message produced by the Sender S honestly, then the recipient R will always accept it.*

Proof. The proposed protocol satisfies the property of correctness. In effect, if the deniable authentication message ψ is correctly generated, then

$$\gamma_1 = \mathcal{H}(\alpha_2) \oplus (M \| T) = \mathcal{H}(\alpha_1) \oplus (M \| T) = \gamma_2$$
Since $\alpha_1 = a_r \cdot Q_s = a_r a_s \cdot P = a_s \cdot Q_r = \alpha_2$

□

7 Security and Performance Analysis

The security of the protocol is based on Computational Diffie-Hellman (CDH), Decisional Diffie-Hellman (DDH) and the Hashed Diffie-Hellman (HDDH) Problems. The proposed protocol will be achieving the following properties.

- **Deniable authentication**: The intended receiver can identify the source of a given message, but cannot prove the source to any third party.

- **Authentication**: During the protocol execution, the sender and the intended receiver can authenticate each other.

- **Confidentiality**: Any outside adversary has no ability to gain the deniable authentication message from the transmitted transcripts.

Theorem 2 *Assume that the collision-free hash function $\mathcal{H}$ behaves as a random oracle. Then the proposed authentication scheme is secure provided that the Diffie-Hellman algorithm assumption holds in $\mathbb{G}$.*

Table 1: Proposed Protocol

Sender S		Receiver R
Select random number $t_s \in [1, n-1]$ Computes $a_s = \mathcal{H}(ID_s) \oplus t_s$ where $ID_s \in \{0,1\}^*$ Computes $Q_s = a_s \cdot P$ Encrypt Q_s as $\tilde{Q}_s = E_{\pi_{pub}}(Q_s \| T)$ where $T \in \mathbb{Z}^*$ is the time stamp		
	$\xrightarrow{\tilde{Q}_s}$	
		Decrypt as $D_{\pi_{prv}}(\tilde{Q}_s \| T) = Q_s$ Select random number $t_r \in [1, n-1]$ $a_r = \mathcal{H}(ID_r) \oplus t_r$ where $ID_r \in \{0,1\}^*$ Computes $\alpha_1 = a_r \cdot Q_s$ Computes $Q_r = a_r \cdot P$ $\beta = \mathcal{H}(ID_r, Q_r, \alpha_1)$
	$\xleftarrow{Q_r, \beta}$	
Compute $\alpha_2 = a_s \cdot Q_r$ Verify $\beta \overset{?}{=} \mathcal{H}(ID_r, Q_r, \alpha_2)$ and $\gamma_1 = \mathcal{H}(\alpha_2) \oplus (M \| T)$ $\psi = (ID_s, T, \gamma_1)$		
	$\xrightarrow{\psi}$	
		Computes $\gamma_2 = \mathcal{H}(\alpha_1) \oplus (M \| T)$ if time stamp T is valid and $\gamma_1 = \gamma_2$ accept M otherwise reject

Proof. Assume that A is an adversary, who can with non-negligible probability, break the proposed authentication scheme. Then, we can use A to construct another algorithm $\tilde{A}$, which is having parameters $(q, \mathbb{G}, P)$ and $\mathcal{H}$, where $\mathcal{H} : \{0,1\}^* \rightarrow \{0,1\}^l$ be a secure cryptographic hash function, behaves a random oracle [24], and a DH instance (P, aP, bP), where $a, b \in \mathbb{Z}_q^*$ as her challenge, and her task here is to compute $(ab) \cdot P$. Let $U = U_1, U_2 \ldots U_n$ be a set of n users who may participate in the system. $\tilde{A}$ first picks a random number j from $\{1, 2 \ldots n\}$, and sets the user U_j's public key $Q_j = t_j \cdot P$. Then, $\tilde{A}$ chooses another $n-1$ random numbers $t_i \in \mathbb{Z}_q^*$ as user U_i's secret key, where $1 \leq i \leq n$ and $i \neq j$, and computes the corresponding public key $Q_i = t_i \cdot P$. Finally, $\tilde{A}$ sends all public key $Q_1, Q_2 \ldots Q_n$ to the adversary A. □

Theorem 3 *The proposed Protocol achieves the mutual authentication between the sender and the intended receiver.*

Proof. The sender S authenticates the receiver R in Step-2 of **Send** phase by verifying the equality $\beta = \mathcal{H}(ID_r, Q_r, \alpha_2)$ and similary receiver R also authenticates the sender S in Step-2 of **Receive** phase by checking the validity of time stamp T and the equality $\gamma_1 = \gamma_2$.

In the proposed protocol, if the receiver accepts the authentication message ψ, receiver R can always identify the source of the message. If an adversary wants impersonate the sender S, he can obtain a time stamp $T \in \mathbb{Z}_q^*$, a message M. But, he could not construct the α_2. If the adversary tries to compute α_2 he has to know the sender's private key a_s for that it needs to solve ECDLP. □

Definition 7.1 *Informally, a deniable authentication protocol is said to achieve the property of confidentiality, if there is no polynomial time algorithm that can distinguish the transcripts of two distinct messages.*

Theorem 4 *The proposed protocol achieves the property of confidentiality provided that the HDDH problem is hard in $\mathbb{G}$.*

Proof. $\gamma_1 = \mathcal{H}(\alpha_2) \oplus (M \| T)$ is actually a hashed ElGamal cipher text [25]. Hashed ElGamal encryption is semantically secure in the random oracle model under the Computational Diffie-Hellman (CDH) assumption. This is the assumption that given P, aP, bP, it is hard to compute $ab \cdot P$ in $\mathbb{G}$, where a, b are random elements of $\mathbb{Z}_q^*$. The CDH assumption is more precisely formulated as follows.
Let $\mathbb{A}$ be an algorithm that takes as input a pair of group elements, and outputs a group element. CDH-advantage of $\mathcal{A}$ to be

$$[a, b \leftarrow \mathbb{Z}_q^* : \mathcal{A}(aP, bP) = ab \cdot P]$$

The CDH assumption on $(\mathbb{G})$ is that any efficient algorithms CDH advantage is negligible. As a result, the proposed protocol can achieves the confidentiality. □

Theorem 5 *The proposed protocol also achieves the property of deniability.*

Proof. To prove that the proposed protocol has deniable property, first we should prove that it enables an intended receiver R to identify the source of the given message M. Since the authenticated message $\psi = (ID_s, T, \gamma_1)$ contains the sender identity ID_s, R can easily identify the source of the message. After verifying $\gamma_1 = \gamma_2$, R can be assured that the message is originated from S. If R intends to expose the message's identity to third party, S would be repudiate as he would argue that S could also generate ψ, since R can compute γ_2 and $\gamma_1 = \gamma_2$, *i.e* transcripts transmitted between the sender S and the receiver R could be simulated by the receiver R himself in polynomial time algorithm. Hence the deniable property is satisfied.

Also we can prove considering the security model describe in section-5. Let us consider a distinguisher $\mathcal{D}$ and two honest players $\mathcal{P}_0$ and $\mathcal{P}_1$ involved in **Game** 2. The distinguisher $\mathcal{D}$ first submits a message $m \in \{0,1\}^*$ to the challenger. Then, the challenger chooses a bit $b \in \{0,1\}$ uniformly at random, and invokes the player $\mathcal{P}_b$ to make a deniable authentication message $\psi = (ID_b, T_b, MAC_b, C)$ on m between $\mathcal{P}_0$ and $\mathcal{P}_1$. In the end, the challenger returns $\psi = (ID_b, T_b, MAC_b, C)$ to the distinguisher $\mathcal{D}$. Since both $\mathcal{P}_0$ and $\mathcal{P}_1$ can generate a valid deniable authentication message $\psi = (ID_b, T_b, MAC_b, C)$, which can pass the verification equation, in an indistinguishable way, when $\mathcal{D}$ returns the guessed value b, we can sure that the probability $\Pr[b = b']$ is $\frac{1}{2}$, and the quantity $Adv_{IBDAP}^{DN}[D] = |Pr[b = b'] - \frac{1}{2}| = |\frac{1}{2} - \frac{1}{2}| = 0$. Based upon the analysis above, we can conclude that the proposed protocol can achieve the deniable authentication. □

Theorem 6 *The Protocol authenticates the source of the message.*

Proof. If someone proves $\mathcal{H}(\alpha_2) \oplus (M \| T)$ to R, where $\alpha_2 = a_s \cdot Q_r$, he must be S. If an adversary gets all the information Q_s in **Extract** phase, he can not compute the session key α_1. For that he has to solve Elliptic Curve Discrete Logarithm Problem. □

Definition 7.2 Secure against Man-in-the-middle *An authentication protocol is secure against an Man-in-the-middle, if Man-in-the-middle can not establish any session key with either the sender or the receiver.*

Theorem 7 *The proposed protocol is secure with respect to the man-in-the-middle (MIA) attack.*

Proof. In the extraction phase, the message is encrypted with the public key π_{pub}. It is difficult for the adversary to get the key π_{prv}. An intruder can intercept the message from S and act as R to negotiate the session key α with S. If he wants execute MIA attack, he must act as the sender S to cheat R. To construct the cipher $\tilde{Q_s}$, first he has to

Table 2: Security properties comparisions

Security	Protocol [9]	Protocol [8]	Proposed Protocol
Denaiable Authentication	✓	✓	✓
Mutual Authentication	✗	✓	✓
Man-in-middle Attack	✗	✗	✓

find out π_{pub} and a_s. For that he has to solve ECDLP, which is computationally infeasible takes fully exponential time. If he fakes an $\tilde{Q}_s$, R can not get correct Q_s. so it resist MIA attack. Similary from the other direction, an intruder can intercept the message from receiver R and and act as sender S to negotiate the session key α with R. To execute MIA attack, he will act as R to cheat S as man in the middle. To compute the session key, he has to find a_r by solving ECDLP which is infeasible. □

7.1 Performance Analysis

Based on computational costs and the communication overloads, we can evaluate the performance of a protocol. The protocol proposed in [8] is based on simple Diffie-Hellman Problem. The computational cost depend upon the exponent operation in group $\mathbb{G}$ where as in the proposed protocol it depend upon scalar multiplication. Since it is based on Elliptic curve Diffie-Hellman Problem. Exponentiation is costly operation than scalar multiplication. For communication overheads, let us assume that the length of ID_S and the digest $\mathcal{H}(\cdot)$ of the input message are of length are 160 bits. The following notations areused for analyzing the computational costs:

- T_M : denotes the computational time of scalar multiplication.
- T_H : Computational time for execution of Hash function.
- T_{Enc}: Time for public key encryption operation.
- T_{Dec} : Time for decryption operation.

8 Conclusion

The security of the proposed protocol is based on difficulty of breaking the Elliptic Curve Diffie-Hellman problem and one way hash function. It archives deniable authentication, mutual authentication as well as confidentiality. Also it is resistant against Man-in-Middle attack. It is an non-interactive protocol and can be easy implemented in mobile devices such as PDA, smart card etc. Since the protocol is based on the elliptic curve cryptography (ECC), it has high security complexity with short key size.

References

[1] Yonatan Aumann and Michael O. Rabin *Efficient Deniable Authentication of Long Messages* Int. Conf. on Theoretical Computer Science in honour of Professor Manuel Blums 60th birthday, 1998. (http://www.cs.cityu.edu.hk/dept/video.html

[2] J.P Kar *ID-based Deniable Authentication Protocol Suitable for Mobile devices*, 3rd International ICST Conference on Security and Privacy in Mobile Information and Communication Systems (MobiSec 2011), Aalborg, Denmark.

[3] J.P.Kar and B.Majhi *A Novel Deniable Authentication Protocol based on Diffie-Hellman Algorithm using Pairing techniques*, ACM International Conference on Communication, Computing and Security (ICCCS 2011), NIT, Rourkela, India. Pp-493-498.

[4] J. S. Chou, Y. L. Chen, J. C. Huangk *A ID-Based Deniable Authentication Protocol on pairings*, Cryptology ePrint Archive: Report, (335)(2006).

[5] J. S. Chou, Y. L. Chen, M. D. Yang *Weaknesses of the Boyd-Mao Deniable Authenticated key Establishment for Internet Protocols*, Cryptology ePrint Archive: Report,(451)(2005).

[6] X. Deng, Lee, C. H. Lee, and H. Zhu *Deniable authentication protocols*, IEEE Proc. Comput. Digit. Tech., Vol. 148 (2), March 2001, pp. 101104.

[7] C. Dwork, M. Naor, A. Sahai *Concurrent zero-knowledge*, Proc. 30th ACM STOC 98,Dallas TX, USA, 1998, pp. 409418.

[8] R. Lu, X.Lin, Z.Cao, L.Qin, X.Liang *A simple Deniable authentication protocol based on Diffie-Hellman Algorithm*, International Journal of Computer Mathematics, Vol.00, No.00, 20007

[9] L. Fan, C. X. Xu, J. H. Li *Deniable authentication protocol based on Diffie-Hellman algorithm*, Electronics Letters 38. (4) (2002) 705-706.

[10] C. Boyd, W. Mao, K. G. Paterson *Deniable authenticated key establishment for Internet protocols* 11th International Workshop on Security Protocols, Cambridge (UK), April 2003.

[11] S. Q. Jiang *Deniable Authentication on the Internet*, Cryptology ePrint Archive: Report,(082)(2007).

[12] S. B. Wilson, and A. Menezes *Authenticated Diffie-Hellman key agreement protocols*, Proceedings of the 5th Annual Workshop on Selected Areas in Cryptography (SAC 98)LNCS, (1999) (339-361)

[13] E. J. Yoon, E. K. Ryu, K. Y. Yoo *Improvement of Fan et al.'s Deniable Authentication Protocol based*

Table 3: Comparision of two deniable authentication Protocols

Performance	**Protocol [9]**	**Protocol [8]**	**Proposed Protocol**
Computational Costs	$4T_{Exp} + 2T_H + T_{Enc} + T_{Dec}$	$6T_{Exp} + 6T_H$	$6T_H + 4T_M + T_{Enc} + T_{Dec}$
Communication overheads	2368 bits	989 bits	480 bits

on Diffie-Hellman Algorithm, Applied Mathematics and Computation, Vol. 167 (1), August 2005, pp. 274-280

[14] Robert W. Zhu, Duncan S. Wong, and Chan H. Lee *Cryptanalysis of a Suite of Deniable Authentication Protocols*, IEEE Communication Letters, Vol. 10, No. 6, June 2006, pp. 504-506.

[15] N. Koblitz. *A course in Number Theory and Cryptography* 2nd edition Springer-Verlag-1994.

[16] K. H Rosen *Elementary Number Theory in Science and Communication*, 2nd ed., Springer-Verlag, Berlin, 1986.

[17] A. Menezes, P. C Van Oorschot and S. A Vanstone *Handbook of applied cryptography* CRC Press, 1997.

[18] D. Hankerson, A .Menezes and S.Vanstone. *Guide to Elliptic Curve Cryptography*, Springer Verlag, 2004.

[19] Maged Hamada Ibrahim *Receiver-deniable public-key encryption* International Journal of Network Security, vol. 8, no. 2, pp. 159165, 2009.

[20] Chi-Yu Liu, Cheng-Chi Lee and Tzu-Chun Lin *Cryptanalysis of an Efficient Deniable Authentication Protocol Based on Generalized ElGamal Signature Scheme* International Journal of Network Security, vol.12, no. 1, pp. 5860, 2011.

[21] Dwork C., Naor M and Sahai A *Concurrent zero-knowledge* Proceedings of 30th ACM STOC'98, 409-418,1998

[22] Abdalla M., Bellare M. and Rogaway P *The oracle Diffie-Hellman assumptions and an analysis of DHIES*, in Topics in Cryptology - CT-RSA 2001, LNCS, 2020, 143-158,2001

[23] Aumann, Y. and Rabin, M. *Authentication, enhanced security and error correcting codes* in Advances in Cryptology - Crypto'98, LNCS, 1462, 299-303.

[24] Bellar M and Rogaway P *Random oracles are practical: a paradigm for designing efficient protocols* in Proceedings of the 1st CSS, 62-73, 1993

[25] Shoup V *Sequences of games: a tool for taming complexity in security proofs*, in Cryptology ePrint Archive: Report 2004/332, available at: http://eprint.iacr.org/2004/332

Jayaprakash Kar Has received his M.Sc and M.Phil in Mathematics from Sambalpur University, India, M.Tech and Ph.D in Computer Science from Utkal University, India. He has worked on key agreement, password-based protocols, word-based public key cryptography, Side Channel Attack on ECC and RFID authentication Protocol. His current research interests are on Elliptic Curve Cryptosystem and Cryptographic Protocols include key management problem of broadcast encryption, Deniable authentication protocols, Proxy Blind Signature scheme and Wireless Sensor Network. He has 15 Journal papers and Conference articles to his credit. Dr. Kar is editorial board member of many peer reviewed Journals and life member of Cryptology Research Society of India , International Association of Computer Science & Information Technology, Singapore and International Association of Engineers, USA.