The Development of Deniable Authentication Protocol Based on The Bivariate Function Hard Problem

Normahirah Nek Abd Rahman[1] and Muhammad Rezal Kamel Ariffin[1,2]
(Corresponding author: Normahirah Nek Abd Rahman)
Al-Kindi Cryptography Research Laboratory, Institute for Mathematical Research, Universiti Putra Malaysia[1]
Department of Mathematics, Faculty of Science, Universiti Putra Malaysia[2]
43400 UPM Serdang, Selangor, Malaysia.
(Email: mahirah_mayrah@yahoo.com, rezal@upm.edu.my)
(Received Jan. 28, 2015; revised and accepted Apr. 25 & June 30, 2015)

Abstract

A 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 the third party. Non-interactive protocol is more efficient than interactive protocol in terms of communication overhead, and thus several non-interactive deniable authentication protocols have been proposed. So, it is very necessary to design a deniable authentication protocol which is non-interactive, secure and efficient. This paper proposes a deniable authentication protocol based on the bivariate function hard problem (BFHP) cryptographic primitive. An improvement based on the BFHP is suggested since the problem of the BFHP provides the needed security elements plus its fast execution time. At the same time, the proposed protocol has properties of completeness, deniability, security of forgery attack, security of impersonation attack and security man-in-the-middle attack also has been proved.

Keywords: Bivariate function hard problem, deniable authentication protocol, non-interactive protocol

1 Introduction

Deniability is a privacy property that ensures protocol participants can later deny taking part in a particular protocol run while authentication is used to ensure that users are who they say they are. So, a deniable authentication protocol is a protocol that 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. There are many interactive and non-interactive deniable authentication protocols have been proposed. However, the interactive manner makes deniable protocols inefficient.

Deniable authentication has two characteristics that differ from traditional authentication. The first one is only the intended receiver can identify the true source of a given message (i.e. able to identify the signature of the sender) and the second one is the receiver cannot prove the source of the message to a third party (i.e. unable to prove the signature of the sender to a third party that the signature belongs to the sender). In other words, once the receiver has obtained and authenticated the message from the sender, the receiver cannot impersonate as the sender to a third party. Because of these two characteristics, the deniable authentication protocol is very useful for providing secure negotiation over internet.

For example, suppose that a customer wants to order an item from a merchant, so the customer should make an offer to the merchant and create an authenticator for the offer because the merchant must be sure that this offer really comes from the customer. However, the merchant wants to be able to prevent the customer from showing this offer to another party in order to elicit a better deal. Therefore, we need a protocol that enables a receiver to identify the source of a given message, but prevents a third party from learning the sender's identity.

In 1998, Dwork et al. [4] proposed an interactive deniable authentication protocol based on concurrent zero knowledge proof while Aumann and Rabin [2] proposed an interactive deniable authentication protocol based on the integer factorization problem (IFP). Later, Deng et al. (2001) [3] introduced two interactive deniable authentication protocols based on the discrete logarithm problem (DLP) and IFP respectively. In 2002, Fan et al. [5] introduced another simple interactive deniable authentication protocol based on Diffie-Hellman Key Distribution Protocol. However, there is a common weakness in the four previous protocols which the sender does not know to whom he proves the source of a given message. That is, a third party can impersonate the intended receiver to identify the source of a given message. Meanwhile, these four protocols are interactive and less efficient.

This scenario has led many cryptographers to come up with non-interactive deniable authentication protocol in order to enhance the efficiency. Shao (2004) [12] proposed a non-interactive deniable authentication protocol based on generalized ElGamal signature scheme. Lu and Cao (2005) [10, 11] proposed two deniable authentication protocols based on bilinear pairing and IFP respectively but their protocol is still unable to achieve the second characteristic of being a deniable authentication protocol.

Later, in 2008, Hwang and Ma [8] proposed deniable authentication protocol with anonymous sender protection. The sender's anonymity is also used to protect the sender's privacy. Though the sent message is forgeable by the receiver, but the sender can provide evidence to prove the message was really sent by him. Hence, to reduce the computational cost of proposed protocols with anonymous sender protection, Hwang and Chao (2010) [7] proposed a new deniable authentication protocol with anonymous sender protection in an efficient way based on Schnorr signature scheme.

Then, Zhang et al. (2011) [13] proposed a new non-interactive deniable authentication protocol based on generalized ElGamal signature scheme, which is more efficient than the previous two protocols (Shao 2004, Lee et al. 2007) [9, 12] both in computation and communication. To authenticate the source of a message, although the proposed protocol needs one more modular exponentiation than Shao's protocol, but as to the length of the communicated messages, just 2|h| are required to be transmitted compared to 3|h| in Shao's protocol. Lee et al.'s protocol needs five exponentiation computations altogether compared to proposed protocol which needs only four. The transmitted bits of the proposed protocol are reduced to 320 bits compared to Lee et al.'s protocol which is $1184 \sim 2208$ bits.

In this paper, we propose a new non-interactive deniable authentication protocol based on the Bivariate Function Hard Problem (BFHP) (Ariffin et al. 2013) [1]. We prove our protocol is secure against forgery attack, impersonation attack and man-in-the-middle attack and prove the properties of completeness and deniability of this protocol. With its guaranteed security, we also show that the performance of the protocol requires reasonable numbers of operation in both sign and verify phases.

The layout of the paper is as follows. In Section 2, we will first review the definition of the BFHP. Proof will be given on the uniqueness and intractability of the BFHP. We will also review in this section, deniable authentication protocol in the standard model. In Section 3, we propose the standard model of the deniable authentication protocol followed by the security analysis in which proof is given. In Section 4, we provide efficiency analysis and comparison of the protocol. In Section 5, the conclusion about our deniable authentication protocol is made.

2 Preliminaries

2.1 Linear Diophantine Equations with Infinitely Many Solutions

Definition 1. *The successful process of prf-solving a Diophantine equation which has infinitely many solutions is the process of determining a preferred solution from a set of infinitely many solutions for the Diophantine equation.*

To further understand and obtain the intuition of Definition 1, we will now observe a remark by Herrmann and May (2008) [6]. It discusses the ability to retrieve variables from a given linear Diophantine equation. But before that we will put forward a famous theorem of Minkowski that relates the length of the shortest vector in a lattice to the determinant.

Theorem 1. *In an ω-dimensional lattice, there is exists a non-zero vector with*

$$\|\upsilon\| \leq \sqrt{\omega}\det(L)^{\frac{1}{\omega}}$$

We now put forward the remark.

Remark 1. There is a method for finding small roots of linear modular equations $a_1x_1+a_2x_2+...+a_nx_n \equiv 0 \pmod N$ with known modulus N. It is further assumed that $\gcd(a_i, N) = 1$. Let X_i be upper bound on $|x_i|$. The approach to solve linear modular equation requires to solve the shortest vector in a certain lattice. We assume that there is only one linear independent vector that fulfills Minkowski bound (Theorem 1) for the shortest vector. Herrmann and May (2008) [6] showed that under heuristic assumption that the shortest vector yields the unique vector $(y_1, ..., y_n)$ whenever

$$\prod_{i=1}^{n} X_i \leq N.$$

If in turn we have

$$\prod_{i=1}^{n} X_i \geq N^{1+\epsilon}.$$

Then the linear equation usually has N^{ϵ} many solutions, which is exponential in the bit-size of N. So, there is no hope to find efficient algorithms that in general improve on this bound, since one cannot even output all roots in polynomial time. We now put forward a corollary.

Corollary 1. *A linear Diophantine equation*

$$\begin{aligned} f(x_1, x_2, ..., x_n) &= a_1x_1 + a_2x_2 + ... + a_nx_n \\ &= N \end{aligned}$$

with

$$\prod_{i=1}^{n} x_i \geq N^{1+\epsilon}$$

is able to ensure secrecy of the preferred sequence $x = \{x_i\}$.

Remark 2. In fact if one were to try to solve the linear Diophantine equation $N = a_1x_1+a_2x_2+...+a_nx_n$, where

$$\prod_{i=1}^{n} x_i \geq N^{1+\epsilon}$$

any method will first output a short vector x= $\{x_i\}$ as the initial solution. Then there will be infinitely many values from this initial condition that is able to reconstruct N.

2.2 Bivariate Function Hard Problem

The following proposition gives a proper analytical description of the Bivariate Function Hard Problem (BFHP).

Definition 2. *We define $\mathbb{Z}^+_{(2^{m-1},2^m-1)}$ as a set of positive integers in the interval as a set of positive integers in the interval $(2^{m-1}, 2^m - 1)$. In other words, if $x \in \mathbb{Z}^+_{(2^{m-1},2^m-1)}$, then x is a m-bit positive integer.*

Proposition 1. (Ariffin et.al (2013)) Let $F(x_1, x_2, ..., x_n)$ be a multiplicative one-way function that maps $F : \mathbb{Z}^n \rightarrow \mathbb{Z}^+_{(2^{m-1},2^m-1)}$. Let F_1 and F_2 be such function (either identical or non-identical) such that $A_1 = F(x_1, x_2, ..., x_n), A_2 = F(y_1, y_2, ..., y_n)$ and $\gcd(A_1, A_2) = 1$. Let $u, v \in \mathbb{Z}^+_{(2^{n-1},2^n-1)}$. Let (A_1, A_2) be public parameters and (u, v) be private parameters. Let

$$G(u, v) = A_1u + A_2v \tag{1}$$

with the domain of the function G is $\mathbb{Z}^2_{(2^{n-1},2^n-1)}$ since the pair of positive integers $(u, v) \in \mathbb{Z}^2_{(2^{n-1},2^n-1)}$ and $\mathbb{Z}^+_{(2^{m+n-1},2^{m+n}-1)}$ is the codomain of G since $A_1u+A_2v \in \mathbb{Z}^+_{(2^{m+n-1},2^{m+n}-1)}$.

If at minimum $n - m - 1$, where (n, m) is chosen such that the value k results in 2^k to be accepted as exponentially large for any probabilistic polynomial time (PPT) adversary to sieve through all possible answers, it is infeasible to determine (u, v) over $\mathbb{Z}$ from $G(u, v)$. Furthermore (u, v) is unique for $G(u, v)$ with high probability.

Remark 3. We remark that the preferred pair (u, v) in $\mathbb{Z}$, is *prf*-solution for Equation (1). The preferred pair (u, v) is one of the possible solutions for Equation (1) given by

$$u = u_0 + A_2t \tag{2}$$

and

$$v = v_0 - A_1t \tag{3}$$

for any $t \in \mathbb{Z}$.

Remark 4. Before we proceed with the proof, we remark here that the Diophantine equation given by $G(u, v)$ is solved when the preferred parameters (u, v) over $\mathbb{Z}$ are found. That is the BFHP is *prf*-solved when the preferred parameters (u, v) over $\mathbb{Z}$ are found.

Proof. We begin by proving that (u, v) is unique for each $G(u, v)$ with high probability. Let $u_1 \neq u_2$ and $v_1 \neq v_2$ such that

$$A_1u_1 + A_2v_1 = A_1u_2 + A_2v_2 \tag{4}$$

We will then have

$$Y = v_2 - v_1 = \frac{A_1(u_1 - u_2)}{A_2}$$

Since $\gcd(A_1, A_2) = 1$ and $A_2 \approx 2^m$, the probability that $Y = v_1 - v_2$ is an integer solution not equal to zero is 2^{-m}. Thus, we have $v_1 = v_2$ with probability $1 - \frac{1}{2^m}$. (i.e. $1 - \frac{1}{2^m}$ is the probability that A_2 divides $u_1 - u_2$).

Next we proceed to prove that to *prf*-solve the Diophantine equation given by Equation (1) is infeasible to be *prf*-solved. From the general solution for $G(u, v)$ is given by Equation (2) and Equation (3) for some integer t to find u within the stipulated interval $u \in (2^{n-1}, 2^n-1)$ we have to find the integer t such that the inequality $2^{n-1} < u < 2^n - 1$ holds. This gives

$$\frac{2^{n-1} - u_0}{A_2} < t < \frac{2^n - 1 - u_0}{A_2}.$$

Then, the difference between the upper and the lower bound is

$$\frac{2^n - 1 - 2^{n-1}}{A_2} = \frac{2^{n-1} - 1}{A_2} \approx \frac{2^{n-2}}{2^m} = 2^{n-m-2}.$$

Since $n-m-1 = k$ where 2^k is exponentially large for any probabilistic polynomial time (PPT) adversary to sieve through all possible answers, we conclude that the difference is very large and finding the correct t is infeasible. This is also the same scenario for v. □

Example 1. *Let $A_1 = 191$ and $A_2 = 229$. Let $u = 41234$ and $v = 52167$. Then $G = 19821937$. Here we take $m = 8$ and $n = 16$. We now construct the parametric solution for this BFHP. The initial points are $u_0 = 118931622$ and $v_0 = -99109685$. The parametric general solution are $u = u_0 + A_2t$ and $v = v_0 - A_1t$. There are approximately $286 \approx 2^9$ (i.e. $\frac{2^{16}}{229}$) values of t to try (i.e. difference between upper and lower bound), while at minimum the value is $t \approx 2^{16}$. In fact, the correct value is $t = 519172 \approx 2^{19}$.*

Case 1. For $(t', v' \notin \mathbb{Z})$, we can find the value of t' which $u' = u_0 + A_2t'$ such that $u \approx 2^n$. Let $u' = 43571 \approx 2^8$. Then $t' = 519161.7948$ and the value of $v' \notin \mathbb{Z}$ since $v' = 50217.79913$ which clearly results v will not be integer if u is not the *prf*-solution.

Case 2. For $(t', v' \in \mathbb{Z})$, we will obtain $v' \in \mathbb{Z}$ with probability $\frac{1}{2^m}$ which $u' = u_0 + A_2t'$ such that $t' = t_0$ and $u \approx 2^n$. Let $u' = 44211 \approx 2^8$. Then the value of $v' \in \mathbb{Z}$ since $v' = 49684$ for $t' = 519159$. Even we get $(t', u', v' \in \mathbb{Z})$ but $u' \neq u$ and $v' \neq v$. In fact there are 2^{19} choices in the example.

2.3 Deniable Authentication Protocol in Standard Model

A deniable authentication protocol in standard model consists of four phases (Setup, Key Generation, Signing, Verifying) which are defined as follows:

1) **Setup:** The authority determines the parameters that can be used by sender and the receiver to generate their private and public key.
2) **Key Generation:** An algorithm that generates private and public key. The private key which is randomly chosen and remain secret, to be used to generate the public key that will be published in public.
3) **Signing:** An algorithm that generates message authentication code (MAC) from the original message which involves hash function.
4) **Verifying:** An algorithm that involves verification of the new MAC generated with the MAC that has been sent by the sender. If both hold, the original message is authentic and has not been altered.

3 The Standard Model of Deniable Authentication Protocol Based on the BFHP

3.1 Proposed Deniable Authentication Protocol

Setup. The authority randomly chooses the following public parameters:

1) p is a large prime number of n-bit size.
2) g is a primitive root in $\mathbb{Z}_p$.
3) $H(\cdot)$ is a collision free hash function with an output is n bits.

Key Generation. When a user wishes to join the system, he chooses a random number $t \in \mathbb{Z}_p$ as his private key and compute $v = g^t (\text{mod } p)$ as his public key. The public key of each user is certificated by certification authority. The sender, S chooses his secret key $t_s \in \mathbb{Z}^+_{(2^{2n-1}, 2^{2n}-1)}$ and computes $v_s = g^{t_s} (\text{mod } p)$ as his public key. The reason why t_s is chosen out of $\mathbb{Z}_p$ can be observe in step 2(i) of signing phase in order for BFHP to hold.

The receiver, R chooses his secret key $t_R \in \mathbb{Z}_p$ and computes $v_R = g^{t_R} (\text{mod } p)$ as his public key.

Signing. When S wants to deniably authenticate a message M to the intended receiver R, he computes the following protocol:

1) Chooses randomly value $\alpha \in \mathbb{Z}^+_{(2^{2n-1}, 2^{2n}-1)}$.
2) Computes
   a. $\sigma = H_1(M)t_s + H_2(M)\alpha$;
   b. $k_1 = (v_R)^{-H_1(M){t_s}^2} (\text{mod } p)$;
   c. $k_0 = (v_R)^{\alpha H_2(M)t_s} (\text{mod } p)$;
   d. $MAC = H(k_0 \| M)$.

   Then, S sends (k_1, σ, MAC) together with message M to R.

Verifying. After receiving (k_1, σ, MAC) together with message M from S, receiver, R computes

1) ${k_1}^* = (v_s)^{\sigma t_R}$;
2) ${k_0}' = k_1 \cdot {k_1}^*$;
3) $MAC = H({k_0}' \| M)$.

R verifies whether $H(k_0 \| M) = H({k_0}' \| M)$. If two equations hold, R accepts the received information. Otherwise, R rejects it. Note that $\|$ is the concatenate operator of strings.

Proposition 2. (Completeness) *If the sender and the receiver follow the protocol, the receiver is able to calculate ${k_0}'$ and then identify the source of the message.*

Proof. From the proposed protocol, we have

$$\begin{aligned} {k_0}' &= k_1 \cdot k_1^* \\ &= (v_R)^{-H_1(M){t_s}^2} \cdot (v_s)^{\sigma t_R} (\text{mod } p) \\ &= g^{-t_R H_1(M){t_s}^2} \cdot g^{{t_s}^2 H_1(M) t_R} \cdot g^{H_2(M)t_s \alpha t_R} (\text{mod } p) \\ &= g^{H_2(M)t_s \alpha t_R} (\text{mod } p) \\ &= (v_R)^{H_2(M)t_s \alpha} (\text{mod } p) \\ &= k_0 \end{aligned}$$

So, $H({k_0}' \| M) = H(k_0 \| M)$. □

Example 2. *The authority randomly chooses $p = 137$ and $g = 101$ as a primitive root in $\mathbb{Z}_p$. The sender, S chooses his secret key $t_s = 781$ and computes $v_s = 118$ as his public key. The receiver, R chooses his secret key $t_R = 157$ and computes $v_R = 11$.*

When S wants to deniably authenticate a message $M = 888$ to the intended receiver R, he chooses randomly value of $\alpha = 813$. He computes $\sigma = 35228$ since $H_1(M) = 17$ and $H_2(M) = 27$. Then, he computes $k_1 = 37$ and $k_0 = 36$. Next, he generates MAC by applying the hash function to the concatenation between M and k_0. He gets $MAC = 1dfc4f553a94cfbf96633b16b2b6e1b5$. Then, S sends (k_1, σ, MAC) together with message M to R.

After receiving (k_1, σ, MAC) together with message M from S, receiver, R computes ${k_1}^* = 38$, ${k_0}' = 36$ and $MAC = 1dfc4f553a94cfbf96633b16b2b6e1b5$ R verifies whether $H(k_0 \| M) = H({k_0}' \| M')$. If two equations hold, R accepts the received information. Otherwise, R rejects it.

3.2 Security Analysis of Deniable Authentication Protocol

Proposition 3. *The proposed protocol is deniable.*

Proof. If the receiver can simulate all the transmitted information between him and the sender, then he cannot prove to any third party where the message is from because the third party cannot identify whether the message is from the sender or is forged by receiver himself.

So, if the receiver tells a third party that the data is from the sender, then the sender can deny it and claims that the receiver himself forge the data. Hence the third party cannot identify who tells the truth.

After receiving (k_1, σ, MAC), the receiver can identify the source of the (k_1, σ, MAC) with his own private key, t_R. However, he cannot prove the source of the message to any party because the receiver can calculate k_0, so he can select any other message M' and construct $MAC' = H(k_0{}'\|M')$ and tells the third party (k_1, σ, MAC') is the information he gets from S.

Without the randomly selected $\alpha \in \mathbb{Z}^+_{(2^{2n-1},2^{2n}-1)}$, the secret key t_s of S and secret key t_R of R, the third party cannot derive k_0 and k_0'. So he cannot prove whether the receiver is telling the truth. □

Proposition 4. *If the attacker cannot personate as the sender by using another pair of $(\alpha', t_s{}')$ in order to communicate with the intended receiver, then the proposed protocol can withstand forgery attack.*

Proof. The attacker chooses his secret key $t_s{}' \in \mathbb{Z}^+_{(2^{2n-1},2^{2n}-1)}$. When attacker wants to deniably authenticate a message M' to the intended receiver R, he computes as follows:

1) Chooses randomly value $\alpha' \in \mathbb{Z}^+_{(2^{2n-1},2^{2n}-1)}$.

2) Computes

   a. $\sigma = H_1(M)t_s{}' + H_2(M)\alpha'$;

   b. $\overline{k_1} = (v_R)^{-H_1(M)(t_s{}')^2} (\mathrm{mod}\, p)$;

   c. $\overline{k_0} = (v_R)^{\alpha' H_2(M)(t_s{}')} (\mathrm{mod}\, p)$;

   d. $MAC = H(\overline{k_0}\|M')$.

Then, attacker sends $(\overline{k_1}, \sigma, MAC)$ together with message M' to R. After receiving $(\overline{k_1}, \sigma, MAC)$ together with message M' from attacker, receiver, R computes

1) ${k_1}^* = (v_s)^{\sigma t_R}$;

2) $k_0{}' = \overline{k_1} \cdot {k_1}^*$;

3) $MAC = H(k_0{}'\|M')$.

Hence, $H(\overline{k_0}\|M') \neq H(k_0{}'\|M')$. The message authentication code, $H(\overline{k_0}\|M') \neq H(k_0{}'\|M')$ since $\overline{k_0} \neq k_0{}'$ and the receiver always uses the sender?s public key v_s to calculate ${k_1}^*$ and identify the source of the message as follows:

$$\begin{aligned} k_0{}' &= k_1 \cdot k_1^* \\ &= (v_R)^{-H_1(M)(t_s{}')^2} \cdot (v_s)^{\sigma t_R} (\mathrm{mod}\, p) \\ &= g^{-t_R H_1(M)(t_s{}')^2} \cdot g^{t_s(H_1(M)t_s{}' + H_2(M)\alpha')t_R} (\mathrm{mod}\, p) \\ &= g^{-t_R H_1(M)(t_s{}')^2} \cdot g^{(t_s{}')t_s H_1(M)t_R} \cdot g^{H_2(M)t_s\alpha' t_R} (\mathrm{mod}\, p) \\ &= g^{-t_R H_1(M)(t_s{}')^2} \cdot g^{(t_s{}')t_s H_1(M)t_R} \cdot (v_R)^{H_2(M)t_s\alpha'} (\mathrm{mod}\, p) \\ &\neq \overline{k_0} \end{aligned}$$

The session secret key $k_0 = (v_R)^{H_2(M)t_s\alpha} (\mathrm{mod}\, p)$ is protected by BFHP. That is, the pair (α, t_s) is protected by BFHP on σ. If the BFHP surrounding σ is *prf*-solved, then both (α, t_s) are found. Hence, no third party can forge a valid k_0 to cheat the receiver although he uses another pair of (α, t_s). □

Remark 5. On the other hand, if the DLP is solved, $t_s \in \mathbb{Z}_p$ would be found. However, the corresponding preferred α would not be obtained. In fact, both the preferred integers (α, t_s) is still not obtained.

Observed from $v_s = g^{t_s} (\mathrm{mod}\ p)$. Solving the DLP, we will get $t_{s0} \in \mathbb{Z}_p$. If $t_s \equiv t_{s0} (\mathrm{mod}\ p)$, then the attacker may initiate search for t_s since $t_s = t_{s0} + pj$ for some $j \in \mathbb{Z}$. Observe that since $t_{s0}, p \sim 2^n$ and $t_s \sim 2^{2n}$, we have $j \sim 2^n$ Hence the probability to obtain the correct j is $\frac{1}{2^n}$.

If $t_s \not\equiv t_{s0} (\mathrm{mod}\ p)$, the attacker may not initiate search for t_s since he cannot find $j \in \mathbb{Z}$ as j is the number of time t_{s0} is reduced by p until t_s is obtained.

The following is an example continued from Example 2 in which we illustrated an attacker utilities attacked parameters (t_s', α') as depicted in Proposition 4.

Example 3. *The authority randomly chooses $p = 137$ and $g = 101$ as a primitive root in $\mathbb{Z}_p$. The attacker, A chooses his secret key $t_s{}' = 727$. The receiver, R chooses his secret key $t_R = 157$ and computes $v_R = 11$.*

When A wants to deniably authenticate a message $M' = 555$ to the intended receiver R, he chooses randomly value of $\alpha = 847$. He computes $\sigma = 35228$ since $H_1(M) = 17$ and $H_2(M) = 27$. Then, he computes $\overline{k_1} = 37$ and $\overline{k_0} = 126$. Next, he generates MAC by applying the hash function to the concatenation between M' and k_0. He gets $MAC = 7306b18193e101e4e2b9a5bff79241e1$. Then, A sends $(\overline{k_1}, \sigma, MAC)$ together with message M' to R.

After receiving $(\overline{k_1}, \sigma, MAC)$ together with message M' from A, receiver, R computes $k_1^* = (v_S)^{\sigma t_R}$ using sender's public key, $v_s = 118$. He gets $k_1^* = 38$ and $k_0{}' = 36$. Then he computes $MAC = 4eca496522032ec8a7132e441c6725d1$. R verifies that $H(\overline{k_0}\|M') \neq H(k_0{}'\|M')$. Then, R does not accept the information he gets from attacker.

Proposition 5. *If an attacker wants to impersonate as the intended receiver in order to identify the source of a given message, then the proposed protocol can withstand such an impersonation attack.*

Table 1: The comparison among deniable authentication protocols

	Fan et al. protocol		Zhang et al. protocol		The proposed protocol	
	S	R	S	R	S	R
Exponentiation	2+1	2+2	2	3	2	1
Hashing Computation	1+1	1+1	2	2	3	1
Data Transmission Overhead	$2\|n\|+2\|h\|$		$2\|h\|$		$3\|n\|+\|r\|$	
Interactive	Yes		No		No	

Proof. In our protocol, any third party want to impersonate as the intended receiver cannot identify the source of the message even if he obtains (k_1, σ, MAC). If he can verify the message authenticator, he must find k_0 and $k_0{}'$. As we prove above, he cannot forge k_0 and $k_0{}'$ as

$$\begin{aligned} k_0{}' &= k_1 \cdot k_1^* \\ &= (v_R)^{-H_1(M){t_s}^2} \cdot (v_s)^{\sigma t_R} (\mathrm{mod}\, p) \\ &= g^{-t_R H_1(M){t_s}^2} \cdot g^{{t_s}^2 H_1(M) t_R} \cdot g^{H_2(M) t_s \alpha t_R} (\mathrm{mod}\, p) \\ &= g^{H_2(M) t_s \alpha t_R} (\mathrm{mod}\, p) \\ &= (v_R)^{H_2(M) t_s \alpha} (\mathrm{mod}\, p). \end{aligned}$$

It is shown that t_R is required in each step to calculate $k_0{}'$. Without the receiver's private key, t_R, it is impossible for the attacker to forge $k_0{}'$. □

Proposition 6. *The proposed protocol is secure against man-in-the-middle attack if man-in-the-middle cannot establish any session key with either the sender or the receiver.*

Proof. Objective of the man-in-the-middle attack is to pretend to be the sender and cheat the receiver. In order to pretend as a sender, he needs to compute σ' for the corresponding M'. But this is infeasible because the pair (α, t_s) is protected by BFHP within the initial σ. On the other hand, the man-in-the-middle cannot pretend to be the receiver to cheat the sender because he needs to obtain the receiver's private key, t_R to compute ${k_1}^* = (v_S)^{\sigma t_R}$. This is also infeasible because t_R is protected by the DLP within v_R. Therefore, the attacker is unable to pretend to be the sender or the receiver. □

4 Comparison

To study the performance of the proposed protocol, we compare it with some previous proposed deniable authentication protocols. We make comparison against the most known efficient interactive protocol (Fan et al. 2002) and non-interactive protocol (Y. Zhang et al. 2011). The comparison is summarized as in Table 1.

To authenticate the source of a message in Fan et al.'s interactive protocol, two modular exponentiation computation and one hashing computation are required by both sender and receiver. In addition, the sender needs to compute a signature with a message recovery which requires one modular and one hash function computation. The receiver needs to verify the signature which requires two modular exponentiation computation and one hash function computation. The data transmission overhead for Fan et al.'s protocol is $2|n| + 2|h|$ bits which $2|n|$ is the modular size and $2|h|$ is output size of hash function.

Our proposed protocol is non-interactive so that the communication process is shorter than in any interactive protocol. In signing phase, the sender needs two modular exponentiation computation and three hash function computation. The receiver needs one modular exponentiation computation and one hash function computation in verifying phase. Data transmission overhead for our proposed protocol is $3|n| + |r|$ bits, $|r|$ denotes the size of α and t_s while Y. Zhang et al.'s protocol is $2|h|$ bits.

5 Conclusion

A new deniable authentication protocol based on the bivariate function hard problem has been developed. One can observe from the Table 1 that the number of exponentiation computation needed is less that known efficient deniable authentication schemes. This suggested that the proposed method has better computational complexity on both the sender and the receiver's end.

The proposed protocol is proved to have the following characteristics which only intended receiver can be authenticated and it is deniable. Some possible attacks have also been considered and we showed that our proposed protocol is secure against forgery attack, impersonation attack and man-in-the-middle attack. Hence, our proposed deniable authentication protocol is more desirable than existing schemes. In the future studies, we will focus to improve the efficiency while still maintain the security of the protocol.

Acknowledgment

The author gratefully acknowledged the financial support of My PhD from the Ministry of Education Malaysia.

References

[1] M. R. K. Ariffin, M. A. Asbullah, N. A. Abu, and Z. Mahad, "A new efficient asymmetric cryptosystem based on the integer factorization problem of n =

p^2q," *Malaysian Journal of Mathematical Sciences*, vol. 7, pp. 19–37, 2013.

[2] Y. Aumann and M. O. Rabin, "Authentication, enhanced security and error correcting codes," in *Advances in Cryptology (CRYPTO'98)*, LNCS 1462, pp. 299–303, Springer-Verlag, 1998.

[3] X. Deng, C. H. Lee, and H. Zhu, "Deniable authentication protocol," *IEE Proceedings on Computer and Digital Techniques*, vol. 148, no. 2, pp. 101–108, 2001.

[4] C. Dwork, M. Noar, and A. Sahai, "Concurrent zero-knowledge," in *Proceedings of 30th Annual ACM Sympossium of Theory of Computing*, pp. 409–418, 1998.

[5] L. Fan, C. X. Xu, and J. H. Li, "Deniable authentication protocol based on diffie hellman algorithm," *Electronic Letters*, vol. 38, no. 4, pp. 705–706, 2002.

[6] M. Herrmann and A. May, "Solving linear equation modulo divisors: on factoring given any bits," *LAdvances in Cryptology (ASIACRYPT'08)*, LNCS 5350, pp. 406–424, Springer, 2008.

[7] S. J. Hwang and C. H. Chao, "An efficient non-interactive deniable authentication protocol with anonymous sender protection," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 13, no. 3, pp. 219–231, 2010.

[8] S. J. Hwang and J. C. Ma, "Deniable authentication protocol with anonymous sender protection," in *International Computer Symposium*, pp. 412–419, Tamsui, Taiwan, 2008.

[9] W. B. Lee, C. C. Wu, and W. J. Tsaur, "A novel deniable authentication protocol using generalized ElGamal signature scheme," *Information Sciences*, vol. 177, pp. 1376–1381, 2007.

[10] R. Lu and Z. Cao, "Non-interactive deniable authentication protocol based form bilinear pairings," *Applied Mathematics and Computation*, vol. 168, pp. 954–961, 2005.

[11] R. Lu and Z. Cao, "Non-interactive deniable authentication protocol based on factoring," *Computer Standard & Interfaces*, vol. 27, no. 4, pp. 401–405, 2005.

[12] Z. Shao, "Efficient deniable authentication protocol based on generalized elgamal signature scheme," *Computer Standard & Interface*, vol. 26, no. 5, pp. 449–454, 2002.

[13] Y. Zhang, Q. Xu, and Z. Liu, "A new non-interactive deniable authentication protocol based on generalized elgamal signature scheme," in *2011 6th IEE Joint International Conference on Information Technology and Artificial Intelligence Conference (ITAIC'11)*, pp. 193–197, Aug. 2011.

Normahirah Nek Abd Rahman received a Bachelor's degree and a Master's degree in Mathematics from Universiti Kebangsaan Malaysia in 2011 and 2012 respectively, and currently doing a PhD in Mathematical Cryptography in Universiti Putra Malaysia since 2013. Her current research topics include Diophantine equation, new attack on RSA cryptosystem using continued fraction and Coppersmith's method.

Muhammad Rezal Kamel Ariffin received his PhD in Mathematics from Universiti Kebangsaan Malaysia (National University of Malaysia) in 2009. He is a lecturer in the Department of Mathematics, Faculty of Science, Universiti Putra Malaysia (UPM). He is also an associate researcher at the Institute for Mathematical Research, UPM conducting research mainly in the mathematical aspects of cryptography. His current research interest is designing and analyzing number theoretic based cryptosystems.