Fully Deniable Mutual Authentication Protocol Based on RSA Signature

Xi-Jun Lin [*]and Lin Sun [†]

November 14, 2013

Abstract: Deniable authentication protocols allow a sender to authenticate a receiver, in a way that the receiver cannot convince a third party that such authentication (or any authentication) ever took place. In this study, we construct a fully deniable mutual authentication protocol based on RSA signature, and then a deniable authenticated key exchange protocol is constructed from the proposed protocol.

Key words: Authentication; Deniability; RSA; Signature

1 Introduction

Authentication has received a lot of attention in the cryptographic literature. In 1998, Dwork et al. [6] first introduced the concept of deniable authentication protocols. Subsequently, we have seen many studies of such protocols in the crypto/security community. Many methods, such as Encryption [4, 6, 10], zero knowledge proof [1, 5, 14], symmetric key [3, 7], commitment scheme [11–13] and projective hash function [8, 13], are used.

In this study, we focus on weak signature-based construction. We propose a novel fully deniable mutual authentication protocol by using RSA signature, since RSA is de facto Internet standard. Then, a deniable authenticated key exchange protocol is constructed from the proposed protocol.

After recalling the relevant technical definitions in the next section, a fully deniable authentication protocol based on RSA signature is proposed in Section 3. We also construct a deniable authenticated key exchange protocol from the proposed protocol in Section 4. In the last section, we conclude our work.

2 Preliminaries

2.1 Communication Model and Security Definitions

The model and the definitions specified below are based on the description and discussions given in [1, 2, 5].

[*] X.J.Lin is with the Department of Computer Sciences and Technology, Ocean University of China. Qingdao 266100, P.R.China. email: linxj77@163.com

[†] L.Sun is with the College of Liberal Arts, Qingdao University. Qingdao 266071, P.R.China. email: sunlin9@126.com

The system for a deniable authentication protocol Π consists of two parties: a sender and a receiver. Each of sender and receiver is an interactive algorithm. The sender interacts with the receiver on input some system parameters and the public information of the receiver. The receiver on input some private information, corresponding public information, and some system parameters, interacts with the sender, and outputs either *accept* or *reject.* The behavior of the sender and the receiver will always follow the protocol specification.

For defining deniability, two games are considered. In one game, a normal run of the deniable authentication protocol between the sender and the receiver is carried out. The game output is the transcript of the protocol. In another game, a simulator which has all the information known by the receiver in the first game is executed. The game output is the output of the simulator. By deniability, we should show that there exists a computable simulator whose output is computationally indistinguishable from the output generated in the first game.

Definition 1 (*Deniability)* *A authentication protocol* Π *is deniable if the outputs of these two games are computationally indistinguishable [10].*

Definition 2 (*authentication)* *A deniable authentication protocol* Π *captures the authentication property if it is negligible for the receiver to ouput* accept *but the sender has never interacted with him.*

2.2 RSA Signature Scheme

we recall the RSA signature scheme which involves three algorithms: *Setup, Sign* and *Verify.*

- *Setup:* Take security parameter l as input, it performs as follows.
  1. Pick two distinct safe primes p and q of length l randomly.
  2. Compute $n = pq$.
  3. Choose an integer e such that $1 < e < \phi(n)$ and $gcd(e, \phi(n)) = 1$.
  4. Determine d such that $ed \equiv 1 \pmod{\phi(n)}$, where $\phi(n) = (p-1)(q-1)$.
  5. Output the public key (n, e), and the private key d.
- *Sign:* Take the message $m \in Z_n^*$ and the private key d as input, output $s = m^d \pmod{n}$ as the signature.
- *Verify:* Take the pair (m, s) and the public key (n, e) as input, check whether $s^e = m \pmod{n}$. If it holds, output 'True'; otherwise 'False'.

3 Our Construction

3.1 The Idea Behind Our Construction

Needless to say, the RSA signature scheme has following two properties.

1. *Existential forgery*: An adversary can firstly picks $s \in_R Z_n^*$ and then computes $m = s^e \pmod{n}$. Obviously, the pair (m, s) satisfies the verification equation $s^e = m \pmod{n}$. Hence, the adversary generates a valid forgery pair (m, s) even without the knowledge of the private key d.

2. If the adversary has to output a message m before knowing the public key e, he cannot generate a valid forgery signature s, even given the public key e after m is outputted (he also does not know the private key d). That is, the adversary cannot forge a valid signature s if the message m is fixed before he knows the public key e.

The idea behind our construction is to use the above two properties. The deniability is captured since the RSA signature is existentially forgeable. Furthermore, the public key e, which is generated after the message is outputted, is computed from two random numbers picked by the sender and the receiver respectively. That is, the message is fixed before the generation of e in the viewpoint of the receiver. Then, if a valid signature s is sent by the sender, the receiver can be assured that the signature is produced by a private key owner. Hence, the authentication property is captured.

3.2 The Proposed Protocol

To setup the protocol, each user picks two distinct safe primes p and q of length l randomly, and computes $n = pq$, then n is public key and the pair (p, q) is private key (which is different from the RSA signature scheme). Moreover, $H_1 : \{0,1\}^* \to \{0,1\}^t, H_2 : \{0,1\}^* \to \{0,1\}^t$ are secure hash functions, where $t < l$, and $MAC(\cdot)$ is secure message authentication code algorithm.

Suppose user P_1's private key is (p_1, q_1) and public key is n_1, and user P_2's private key is (p_2, q_2) and public key is n_2. ID_1 and ID_2 are P_1's identifier and P_2's identifier, respectively. Then, P_1 and P_2 perform the following protocol.

1. $P_1 \to P_2$: P_1 picks $r_1 \in_R \{0,1\}^*$ and k randomly such that $k \in \mathbb{Z}_{n_1}^*$ and $k \in \mathbb{Z}_{n_2}^*$, and then sends (ID_1, k, r_1) to P_2.

2. $P_2 \to P_1$: Upon receipt of (ID_1, k, r_1), P_2 performs as follows.

   - Pick $r_2 \in_R \{0,1\}^*$ and compute $e' = H_1(ID_2, ID_1, r_2, r_1)$, and d' such that $e'd' \equiv 1 \pmod{\phi(n_2)}$.
   - Compute $s' = (k \cdot H_2(ID_2, ID_1, r_2, r_1))^{d'} \pmod{n_2}$.
   - Compute $\delta' = MAC_k(r_2, r_1, s')$.

   Then, P_2 sends $(ID_2, r_2, (s', \delta'))$ to P_1.

3. $P_1 \to P_2$: Upon receipt of $(ID_2, r_2, (s', \delta'))$, P_1 checks whether $s'^{H_1(ID_2, ID_1, r_2, r_1)} = k \cdot H_2(ID_2, ID_1, r_2, r_1) \pmod{n_2}$ and $\delta' = MAC_k(r_2, r_1, s')$. If the equations do not hold, P_1 rejects; otherwise, P_1 accepts and performs as follows.

   - Compute $e = H_1(ID_1, ID_2, r_1, r_2)$ and d such that $ed \equiv 1 \pmod{\phi(n_1)}$.
   - Compute $s = (k \cdot H_2(ID_1, ID_2, r_1, r_2))^d \pmod{n_1}$.
   - Compute $\delta = MAC_k(r_1, r_2, s)$.

Then, P_1 sends (s, δ) to P_2.

4. Upon receipt of (s, δ), P_2 checks whether $s^{H_1(ID_1, ID_2, r_1, r_2)} = k \cdot H_2(ID_1, ID_2, r_1, r_2) \pmod{n_1}$ and $\delta = MAC_k(r_1, r_2, s)$. If the equations hold, P_2 accepts; otherwise, P_2 rejects.

Theorem 1 *The proposed protocol is deniable in the context of Definition 1.*

Proof: To prove the deniability property, we should prove that the transcript between P_1 and P_2 could be simulated by a probabilistic polynomial time simulator only with P_1's private key or P_2's private key. That is, P_1 or P_2 can construct the transcript by himself. Thus, the deniability property can be proved via the simulation process of the simulator.

For security analysis, we consider H_1 and H_2 as random oracles. Given ID_1, ID_2, P_1's public key n_1 and P_2's private key (p_2, q_2), we show a construction of the simulator as follows:

1. Pick $r_1, r_2 \in_R \{0,1\}^*$ and $s \in_R \mathbb{Z}_{n_1}^*$, and then compute $e = H_1(ID_1, ID_2, r_1, r_2)$.
2. Compute $k = s^e / H_2(ID_1, ID_2, r_1, r_2) \pmod{n_1}$ and $\delta = MAC_k(r_1, r_2, s)$.
3. Compute $e' = H_1(ID_2, ID_1, r_2, r_1)$ and d' such that $e'd' \equiv 1 \pmod{\phi(n_2)}$.
4. Compute $s' = (k \cdot H_2(ID_2, ID_1, r_2, r_1))^{d'} \pmod{n_2}$ and $\delta' = MAC_k(r_2, r_1, s')$.
5. Output $(ID_1, k, r_1; ID_2, r_2, (s', \delta'); (s, \delta))$.

Based on the construction of the simulator, r_1, r_2 are uniformly distributed over $\{0,1\}^*$, H_1, H_2 are random oracles and s is uniformly distributed over $\mathbb{Z}_{n_1}^*$, so we have that k is also uniformly distributed. Moreover, δ is uniformly distributed since $MAC(\cdot)$ is secure message authentication code algorithm. Similarly, s' and δ' are uniformly distributed. That is, P_2 can construct the transcript by himself. We can prove in the same way that P_1 can construct the transcript by himself. Thus, the transcript of the simulator is indistinguishable from a real transcript to a third party, then the deniability property is captured.

Theorem 2 *The authentication property of the proposed protocol is captured in the context of Definition 2.*

Proof: Since k is fixed before e and e' are computed, with the property of RSA signature scheme, we know that the adversary cannot forge valid signatures s and s' on k. Hence, he cannot construct valid message authentication codes δ and δ' which are generated with the key k. Thus, the authentication property is captured. □

4 Deniable Authenticated Key Exchange Protocol

In this section, we construct a deniable authenticated key exchange protocol from the proposed protocol.

To setup the protocol, a multiplicative group of prime order is generated and g is its generator, the other system parameters are same as the proposed protocol. To share a session key, P_1 and P_2 perform the following protocol.

1. $P_1 \to P_2$: P_1 picks k randomly such that $k \in \mathbb{Z}_{n_1}^*$ and $k \in \mathbb{Z}_{n_2}^*$, computes $r_1 = g^x$ where x is picked randomly, and then sends (ID_1, k, r_1) to P_2.

2. $P_2 \to P_1$: Upon receipt of (ID_1, k, r_1), P_2 performs as follows.

   - Pick y randomly, then compute $r_2 = g^y$, $e' = H_1(ID_2, ID_1, r_2, r_1)$ and d' such that $e'd' \equiv 1 \pmod{\phi(n_2)}$.
   - Compute $s' = (k \cdot H_2(ID_2, ID_1, r_2, r_1))^{d'} \pmod{n_2}$.
   - Compute $\delta' = MAC_k(r_2, r_1, s')$.

   Then, P_2 sends $(ID_2, r_2, (s', \delta'))$ to P_1.

3. $P_1 \to P_2$: Upon receipt of $(ID_2, r_2, (s', \delta'))$, P_1 checks whether ${s'}^{H_1(ID_2, ID_1, r_2, r_1)} = k \cdot H_2(ID_2, ID_1, r_2, r_1) \pmod{n_2}$ and $\delta' = MAC_k(r_2, r_1, s')$. If the equations do not hold, P_1 rejects; otherwise, P_1 accepts and performs as follows.

   - Compute $e = H_1(ID_1, ID_2, r_1, r_2)$ and d such that $ed \equiv 1 \pmod{\phi(n_1)}$.
   - Compute $s = (k \cdot H_2(ID_1, ID_2, r_1, r_2))^d \pmod{n_1}$.
   - Compute $\delta = MAC_k(r_1, r_2, s)$.

   Then, P_1 sends (s, δ) to P_2, and computes the shared session key $SK = r_2^x$.

4. Upon receipt of (s, δ), P_2 checks whether $s^{H_1(ID_1, ID_2, r_1, r_2)} = k \cdot H_2(ID_1, ID_2, r_1, r_2) \pmod{n_1}$ and $\delta = MAC_k(r_1, r_2, s)$. If the equations hold, P_2 accepts, and computes the shared session key $SK' = r_1^y$; otherwise, P_2 rejects.

It is clear that $SK' = r_1^y = g^{xy} = r_2^x = SK$. That is, if P_1 and P_2 both output *accept*, they can share a same session key.

Obviously, the deniability property and the authentication property are captured since the key exchange protocol is same as the proposed protocol except that r_1 is replaced with g^x and r_2 is replaced with g^y. Moreover, since Computational Diffie-Hellman problem is hard, we conclude that no probabilistic polynomial time adversary can break this protocol with non-negligible probability.

5 Conclusion

In this study, we propose a fully deniable mutual authentication protocol based on RSA signature, and then construct a deniable authenticated key exchange protocol.

References

[1] Aumann, Y., Rabin, M.O.: ‘Authentication, enhanced security and error correcting codes’. CRYPTO’98, Santa Barbara, California, August 1998, pp. 299-303.

[2] Aumann, Y., Rabin, M.O.: ‘Efficient deniable authentication of long messages‘. Int. Conf. on Theoretical Computer Science, 1998.

[3] Boyd, C., Mao, W., Paterson, K.G.: 'Deniable authenticated key establishment for Internet protocols'. Security Protocols, 11th Int. Workshop, Cambridge, UK, April 2003, pp. 255-271.

[4] Brown D. R. L., 'Deniable authentication with RSA and multicasting', Cryptology ePrint Archive, http://eprint.iacr.org/2005/056.pdf, Feb 2005.

[5] Deng, X., Lee, C.H., Zhu, H.: 'Deniable authentication protocols', IEE Proc. Comput. Digit. Tech., 2001, 148, (2), pp. 101-104.

[6] Dwork, C., Naor, M., Sahai, A.: 'Concurrent zero knowledge'. Proc. 30th Annual ACM Symp. on the Theory of Computing, Dallas, Texas, May 1998, pp. 409-418.

[7] Fan, L., Xu, C.X., Li, J.H.: 'Deniable authentication protocol based on DiffieCHellman algorithm', Electron. Lett., 2002, 38, (4), pp. 705-706.

[8] Feng, T., Hua, L.F., Feng, M.J., Moon, S.: 'A new approach for UC security concurrent deniable authentication', Sci. China F: Inf. Sci., 2008, 51, (4), pp. 352-367.

[9] Goldreich, O.: 'Foundations of Cryptography: Basic Tools'. Cambridge University Press, 2001.

[10] Harn, L., Ren, J.: 'Design of fully deniable authentication service for E-mail applications'. IEEE Commun. Lett., 2008, 12(3). pp.210-221.

[11] Jiang, S.Q., Naini, R.S.: 'An efficient deniable key exchange protocol (extended abstract)'. FC 2008, Cozumel, Mexico, January 2008, pp. 47-52.

[12] Pass, R.: 'On deniability in the common reference string and random oracle model'. CRYPTO 2003, Santa Barbara, California, August 2003, pp. 316-337.

[13] Raimondo, M.D., Gennaro, R.: 'New approaches for deniable authentication'. CCS 2005, Alexandria, VA, USA, November 2005, pp. 112-121.

[14] Zhu, R.W., Wong, D.S., Lee, C.H.: 'Cryptanalysis of a suite of deniable authentication protocols', Commun. Lett., 2006, 10, (6), pp. 504-506.