An Attack on Not-interactive Designated Verifier Proofs for Undeniable Signatures

Guilin Wang

Infocomm Security Department (ICSD)
Institute for Infocomm Research (I[2]R)
21 Heng Mui Keng Terrace, Singapore 119613
`http://i2r.a-star.edu.sg/icsd/staff/guilin`
`glwang@i2r.a-star.edu.sg`

Abstract. At Crypto'89, Chaum and van Antwerpen first introduced the concept of undeniable signatures, which has a special property such that a signature cannot be verified without the signer's cooperation. In 1996, Jakobsson, Sako, and Impagliazzo proposed a not-interactive undeniable signature scheme by employing a new primitive called *designated verifier proofs*. However, this paper shows that their scheme is *insecure* by demonstrating a simple attack that allows a dishonest signer to convince a designated verifier receiving invalid signatures. In addition, two intuitive countermeasures are presented.

Keywords: digital signature, undeniable signature, designated verifier proof.

1 Introduction

Undeniable signature, first proposed at Crypto'89 by Chaum and van Antwerpen [1], is a special type of digital signature in the sense that the validity of an alleged signature cannot be verified without the cooperation of the signer. With this property, undeniable signatures are expected to be useful in many applications [2]. However, undeniable signatures are vulnerable to blackmailing [4, 5] and mafia attacks [3].

In 1996, Jakobsson, Sako, and Impagliazzo introduced a new primitive called *designated verifier proofs* [6]. Such proofs enable a prover Alice to convince a designated verifier Bob that a statement is true. But such proofs cannot be transferred to convince a third party Cindy the same statement. The reason is that Bob himself can also generate such proofs. The basic idea is as follows. When Alice wants to convince only the designated verifier Bob a statement Θ, she actually prove the statement "*Θ is true* `or` *I knows Bob's secret key*". Furthermore, they proposed an elegant not-interactive designated verifier proof for Chaum's zero-knowledge undeniable signature scheme [2] to avoid blackmailing and mafia attacks. In other words, they introduced a not-interactive undeniable signature scheme.

In this paper, however, we show that this not-interactive undeniable signature scheme is *insecure*, since we identify a simple attack that enables the signer Alice to convince the designated verifier Bob receiving invalid signatures.

The rest of this paper is organized as follows. Section 2 reviews Jakobsson et al.'s not-interactive undeniable signature scheme. Then, we demonstrate our attack and propose two countermeasures in Section 3.

2 Jakobsson et al.'s Scheme

Let p, q be two large primes such that $q|(p-1)$, G_q a subgroup of $\mathbb{Z}_p^*$ of order q, and g a generator of G_q. In addition, assume that Alice and Bob have the DLP key pairs $(x_A, y_A = g^{x_A} \bmod p)$ and $(x_B, y_B = g^{x_B} \bmod p)$, respecitively.

In Chaum's undeniable signature scheme [2], Alice's signature on (hashed) message m is $s = m^{x_A} \bmod p$. To realize not-interactive confirmation, Jakobsson et al. proposed the following scheme that allows Alice to convince the designated verifier Bob that a pair (m, s) is valid, i.e., $\log_g y_A = \log_m s$.

Construct A Proof: The prove, Alice, selects three random number $w, r, t \in_R \mathbb{Z}_q$, and computes

$$\begin{aligned} c &= g^w y_B^r \bmod p \\ G &= g^t \bmod p \\ M &= m^t \bmod p \\ h &= hash_q(c, G, M) \\ d &= t + x_A(h + w) \bmod q, \end{aligned} \tag{1}$$

where $hash_q(\cdot)$ is a hash function mapping to $\mathbb{Z}_q$. Then, Alice sends the proof (w, r, G, M, d) to the designated verifier, Bob.

Verify A Proof: The designated verifier, Bob, can verify a proof by computing

$$\begin{aligned} c &= g^w y_B^r \bmod p \\ h &= hash_q(c, G, M), \end{aligned} \tag{2}$$

and checking that

$$\begin{aligned} G y_A^{h+w} &= g^d \bmod p \\ M s^{h+w} &= m^d \bmod p. \end{aligned} \tag{3}$$

Simulating Transcripts: By selecting three random numbers $d, \alpha, \beta \in_R \mathbb{Z}_q$, the designated verifier Bob can simulate correct transcripts as follows:

$$\begin{aligned} c &= g^\alpha \bmod p \\ G &= g^d y_A^{-\beta} \bmod p \\ M &= m^d s^{-\beta} \bmod p \\ h &= hash_q(c, G, M) \\ w &= \beta - h \bmod q \\ r &= (\alpha - w) x_B^{-1} \bmod q. \end{aligned} \tag{4}$$

3 Our Attack

To cheat a designated verifier Bob, Alice selects four random numbers $w, r, t, \bar{t} \in_R \mathbb{Z}_q$, and then computes the proof $P = (w, r, G, \bar{M}, d)$ and an invalid signature $\bar{s}$ for a message m *simultaneously* as follows

$$\begin{array}{ll} c & = g^w y_B^r \bmod p \\ G & = g^t \bmod p \\ \bar{M} & = m^{\bar{t}} \bmod p \\ h & = hash_q(c, G, \bar{M}) \\ d & = t + x_A(h+w) \bmod q \\ \bar{s} & = m^{x_A} \cdot m^{(t-\bar{t})(h+w)^{-1}} \bmod p. \end{array} \tag{5}$$

After that, Alice sends the message-signature pair $(m, \bar{s})$ to Bob, and stores the proof $P = (w, r, G, \bar{M}, d)$ securely. When Bob needs to check the validity of $\bar{s}$, Alice sends the proof P to Bob. It is easy to see that Bob will believe that $\bar{s}$ is Alice's valid signature for message m, since the proof P satisfies equations (2) and (3). However, in later disputes, Alice can convince a third party (e.g. a judge) that $\bar{s}$ is indeed invalid by using a denial protocol [2].

The above attack results from the observation that equation (3) guarantees $\log_m s = x_A$ *only if* $\log_m M = \log_g G$. However, the proof (w, r, G, M, d) does not necessarily guarantee $\log_m M = \log_g G$. Therefore, dishonest signer Alice can change M and s at the same time such equation (3) is still satisfied.

From the above observation, there are two intuitive countermeasures to avoid our attack. The first one is that requiring Alice to provide additional proof to show that $\log_m M = \log_g G$. This improvement will increase the length of the proof. The other is to embed the signature s in the input of the hash function, i.e., let $h = hash_q(c, G, M, s)$. Of course, the security of such improvements needs to be analyzed rigorously in future.

References

1. D. Chaum and H. van Antwerpen. Undeniable signatures. In: *CRYPTO'89, LNCS 435*, pp. 212-216. Springer-Verlag, 1989.
2. D. Chaum. Zero-knowledge undeniable signatures. In: *EUROCRYPT'90, LNCS 473*, pp. 458-464. Springer-Verlag, 1991.
3. Y. Desmedt, C. Coutier, and S. Bengio. Special Uses and Abuses of the Fiat-Shamir Passport Protocol. In: *Crypt'87*, pp. 21-39. Springer-Verlag, 1987.
4. Y. Desmedt and M. Yung. Weakness of undeniable signature schemes. In: *EUROCRYPT'91, LNCS 547*, pp: 205-220. Springer-Verlag, 1991.
5. M. Jakobsson. Blackmailing using undeniable signatures. In: *EUROCRYPT'96, LNCS 950*, pp.: 425-427. Springer-Verlag, 1994.
6. M. Jakobsson, K. Sako, and R. Impagliazzo. Designated verifier proofs and their applications. In: *EUROCRYPT'96, LNCS 1070*, pp. 143-154. Springer-Verlag, 1996.