情報保護學會論文誌
第20卷 第5號, 2010. 10

결함 있는 안전성 증명을 갖는 수신자 지정 서명기법들에 대한 정확한 안전성분석

김 기 태,[1†] 양 대 헌[1], 이 경 희[2‡]
[1]인하대학교, [2]수원대학교

Exact Security Analysis of Some Designated Verifier Signature Schemes With Defective Security Proof

KiTae Kim,[1†], DaeHun Nyang,[1] KyungHee Lee[2‡]
[1]INHA University, [2]The University of Suwon

요 약

수신자 지정 서명은 서명자가 지정된 검증자에게 서명의 유효성을 증명할 수 있도록 하는 서명기법이다. 한편 그 지정된 검증자는 제삼자에게 서명문서의 소스 즉 두 가능한 서명자들 중에서 누구에 의한 서명인지를 확인시킬 수 없다. 일반적인 전자서명과 달리, 수신자 지정 서명은 서명자가 지정된 수신자를 제외한 누구에게든 자신의 서명을 부인할 수도 있다. 그동안 제안된 몇몇 기법들 중에서, 최근 Zhang등의 기법과 Kang등의 기법이 다양한 공격에 취약하다는 사실이 밝혀졌다. 본 논문에서, 위의 기법들이 저자들에 의해서 안전성 증명이 제시되었음에도 불구하고 공격을 허용하게 되는 근본적인 이유를 밝히고, 더불어 Huang-Chou 기법과 Du-Wen 기법이 같은 문제를 갖는다는 사실을 보인다. 나아가 Huang-Chou의 기법에 대하여 실질적인 공격들을 제안한다. 마지막으로, Du-Wen 기법은 안전성 증명과정에서 저자들이 위 기법들의 저자들과 동일한 오류를 범하였으나 그 오류를 수정하여 실제 증명가능한 안전성을 갖는 기법임을 보인다.

ABSTRACT

Designated verifier signatures allow a signer to prove the validity of a signature to a specifically designated verifier. The designated verifier can be convinced but unable to prove the source of the message to a third party. Unlike conventional digital signatures, designated verifier signatures make it possible for a signer to repudiate his/her signature against anyone except the designated verifier. Recently, two designated verifier signature schemes, Zhang et al.'s scheme and Kang et al.'s scheme, have been shown to be insecure by concrete attacks. In this paper, we find the essential reason that the schemes open attacks while those were given with its security proofs, and show that Huang-Chou scheme and Du-Wen scheme have the same problem. Indeed, the security proofs of all the schemes reflect no message attackers only. Next, we show that Huang-Chou scheme is insecure by presenting universal forgery attack. Finally, we show that Du-Wen scheme is, indeed, secure by completing its defective security proof.

Keywords: Digital signature, designated verifier signature, cryptanalysis

접수일: 2010년 5월 14일, 수정일: 2010년 8월 12일,
게재확정일: 2010년 8월 31일

† 주저자, ktkim@inha.ac.kr
‡ 교신저자, khlee@suwon..ac.kr

I. Introduction

The primitive, designated verifier signatures, was introduced at Eurocrypt'96 by Jakobsson, Sako, and Impagliazzo [10], inspired on undeniable signatures. In the same year, Chaum independently introduced a similar concept in [3] under the name of private signatures. In undeniable signatures, the concept suggested by Chaum and Antwerpen [4], signers should participate during verification process to avoid undesirable verifier getting convinced of the validity of signatures. The signer can reject invalid signatures but cannot deny valid signatures. One feature of undeniable signature schemes is that signers can decide when their signatures are verified but do not know to whom they are proving the validity of signatures.

To solve some issues of this feature, Jakobsson et al. suggested designated verifier signatures and strong designated verifier signatures in which each signer is allowed to specify the verifier [10]. The notions were formalized and further investigated by Saeednia, Kremer and Markowitch [15]. In designated verifier signature (DVS, for short), verifiers can simulate signatures that are indistinguishable from signatures created by signers. Due to this property, signatures cannot be transferred to a third party even if the verifiers' private keys are revealed. Normal DVS schemes are designed to be publicly verifiable and so everyone has access to verification algorithm. But, anyone, except for the verifier, should not be convinced whether signatures are valid ones from the signer or simulated ones from the designated verifier. On the other hand, if the designated verifier is assumed to be honest then such schemes may not achieve the goal of the designated verifier signatures. For use even in such a scenario, strong designated verifier signatures require an additional property that everyone can simulate signatures from which no one, except for the verifier, can distinguish the real signatures. To achieve this requirement, strong designated verifier signature schemes are constructed with private verifiability: the secret key of the designated verifier is necessary to perform the verification algorithm. In other words, only the designated verifier can verify the validity of signatures and even the signer cannot verify the signatures if he does not keep track of the signatures.

Recently, many researchers have attempted to construct (strong) ID-based designated verifier signatures. Susilo, Zhang and Mu proposed an ID-based strong DVS schemes based on bilinear Diffie-Hellman assumption [16]. Huang, Susilo, Mu, and Zhang also proposed a strong DVS scheme and a short ID-based strong DVS scheme [8,9]. Kumar, Shailaja and Saxena proposed a novel ID-based strong DVS scheme [13]. Zhang and Mao proposed an ID-based strong DVS scheme which enjoys non-delegatability [17]. Kang, Boyd and Dawson proposed an ID-based DVS scheme [11] and an ID-based strong DVS scheme [12]. More recently, Du and Wen pointed out Kang et al.'s scheme in [11] is vulnerable to universal forgery attack and suggested another identity based DVS to enhance the security [5]. Huang and Chou pointed out Kang et al.'s another scheme in [12] is also insecure and proposed an improvement [7].

As provable security is desirable in cryptographic community, cryptographic scheme are usually given with its security proofs under suitable hardness problems. Following this trend, the authors of the above mentioned schemes except for [12] give proofs to claim that their proposed schemes are secure. However, unfortunately, most of

the scheme turn out to be insecure by several types of attacks [5,7,12]. Nevertheless, no research work has investigated which part of the proofs is misleading while it is important to explore the possibility that the scheme can be improved.

In this paper, we first point out the common mistake the authors took in their security proofs in the papers [5,7,11,17]. Next, we show that Huang et al.'s scheme suffers from the universal forgery attack while the authors claimed their scheme is the first really secure strong designated verifier signature scheme having the source hiding security. Our attacks on the scheme give another example of the evidence that new construction without a careful proof of security is likely to contain serious flaws. For completeness, we finally give the correct security proof on Du et al.'s scheme.

II. Preliminaries

In this section, we review the definition of bilinear pairing and a related hardness assumption. Throughout this paper, we denote G_1 and G_2 by cyclic groups of the same prime order q.

Definition 1 (Bilinear Pairing).

An admissible bilinear pairing is a map $e: G_1 \times G_1 \rightarrow G_2$ that has the following properties:

(1) Computable: There is a polynomially bounded algorithm to compute $e(P,Q)$ for any $P,Q \in G_1$.

(2) Bilinear: $e(aP,bQ) = e(P,Q)^{ab}$ for all $P,Q \in G_1$ and $a,b \in Z_q^*$.

(3) Non-degenerate: There is a $P \in G_1$ such that $e(P,P) \neq 1_{G_2}$. That is, for non-identity elements $P,Q \in G_1$, we have $e(P,Q) \neq 1$.

In the above case, we say that G_1 is a bilinear group and (G_1,G_2) is a bilinear group pair. Note that the original Weil pairing for an elliptic curve does not satisfy non-degeneracy, but a modified Weil pairing over super-singular curve and Tate pairing have the above properties.

Definition 2
(Bilinear Diffie-Hellman Assumption : BDH).

The bilinear Diffie-Hellman problem in G_1 is as follows: Given (P,aP,bP,cP) for randomly chosen $a,b,c \in Z_q^*$, it is infeasible to compute $e(P,P)^{abc}$.

An algorithm A has advantage ϵ in solving BDH in G_1 if

$$\Pr[A(P,aP,bP,cP) = e(P,P)^{abc}] \geq \epsilon$$

where the probability is over the random choice of a,b,c, the choice of P, and the random coins of A. We say that the (ϵ,t)-BDH assumption holds in G_1 if no t-time algorithm has advantage at least ϵ in solving BDH problem in G_1.

III. ID-based Strong Designated Verifier Signature Schemes and its Security Models

Definition 3
(ID-based Strong Designated Verifier Signatures).

A strong designated verifier signature scheme (SDVS) consists of a tuple of (possibly randomized) algorithms (Setup, Extract, Sign, Vrfy, TrSim) where

Setup: The setup algorithm, on input security parameter κ, outputs the public parameters par and the master secret key msk.

Extract: The key extraction algorithm takes the public parameters par

and an identity ID, and outputs the private key sk_{ID} for the identity.

Sign: The signature generation algorithm, takes as input the public parameters par, private (signing) key sk_{ID},, the designated signer's identity V (and hence pk_V) and message m in the message space, outputs the signature σ on the message.

Verify: The signature verification algorithm is a deterministic algorithm that takes as input the public parameters par, the designated verifier's private key sk_V, a message m, and a signature σ, outputs "accept" or "rejec".

TrSim: The transcript simulation algorithm, takes as input a signer's public key, designated verifier's private key, and a message, outputs a signature on the message.

We say that a signature σ on m is valid with respect to (pk_S, pk_V) if $Verify_{sk_V}(m, \sigma, pk_S, pk_V)$ outputs "accept". As usually, we require that a designated verifier signature scheme is correct, that is, for all (pk_S, sk_S) and (pk_V, sk_V) generated by Extract, and for message m in the message space, we should have

$$Verify_{sk_V}(Sign_{sk_S}(m)) = accept$$

For the sake of simplicity, we sometimes omit to explicitly include public parameter par that is a part of the input of all but one algorithm.

Strong DVS schemes are required to satisfy several properties, namely unforgeability, non-transferability and strongness(or privacy of signer's identity). We give formal definition of unforgeability for our purpose, and list other security requirements only for the sake of completeness in informal argument since we will not need them in this paper. The notions are following the papers [9,10,15,16,17].

1. Correctness: If the signer properly generates a signature by running the signing algorithm, the signature must be accepted by the verification algorithm.

2. Unforgeability: Informally, without the knowledge of the private key of either the signer or the designated verifier, it is infeasible to create a valid signature with respect to the signer and the verifier. Formally, this security can be defined in the following experiment argument.

Experiment $\mathrm{Exp}^{uf-cma}_{ID-SDVS,A}(\kappa)$
$(par, msk) \leftarrow Setup(1^{\kappa})$
$(ID_S, ID_V, m, \sigma) \leftarrow A^{OExtract, OSign, OTrSim}(par)$
If $Verify_{sk_S}(ID_V, m, \sigma) = 0$ then return 0
If the followings are satisfied then return 1
(i) ID_S and ID_V have never been queried to the $OExtract$ oracle.
(ii) (ID_S, ID_V, m) has never been queried to the $OSign$ oracle.
Return 0

In the above experiment, $OExtract$ is the key extraction oracle that takes ID as input and returns the corresponding private key sk_{ID}, $OSign$ is the signing oracle that, on inputs (ID_S, ID_V, m), outputs σ as a response by running the Sign algorithm, and OTrSim is the transcripts simulation oracle, on inputs (ID_S, ID_V, m), outputs the result of $TrSim_{sk_V}(ID_S, m)$.

We assume that the signature output by the adversary is in the signature space, without loss of generality.

Definition 4 (Unforgeability). An ID-based strong designated verifier signature scheme $ID-SDVS$ is unforgeable under chosen message if for any polynomial-time adversary A, the advantage $Adv^{uf-cma}_{ID-SDVS,A}(\kappa)$ defined by

$\Pr[\mathrm{Exp}^{uf-cma}_{ID-SDVS,A}(\kappa) = 1]$ is negligible in κ.

In words, the adversary is explicitly given public parameters as input and has oracle access to $OExtract$, $OSign$, and $OTrSim$. The adversary wins if he creates a valid designated verifier signature (ID_S, ID_V, σ, m), under the restriction that he never been queried (ID_S, ID_V) and $(ID_S, ID_V, m)$to the key extraction oracle and the signing oracle, respectively. In a secure designated verifier signature scheme, we require that the adversary A not be able to create such a signature.

3. Non-transferability: The designated verifier, even after being convinced of a signature on some message, is not able to convince any other user of this fact. Informally, this property is defined as follows: given a signature on a message, it is infeasible to determine who, from the original signer or the designated verifier, created the signature, even if one knows all secret keys. Non-transferability is usually ensured by allowing the designated verifier to simulate signatures that are intended for him. This notion is often called *Source Hiding* or perfectly non-transferability if the signatures and the transcripts are perfectly indistinguishable.

4. Strongness (Privacy of signer's identity): Anyone except the designated verifier can not derive useful knowledge from a signature, even when the designated verifier is believed to be honest and signer's secret key is revealed. Note that the signer should not able to distinguish the signatures generated by himself from the transcripts simulated by the designated verifier.

5. Non-delegatability: It is hard for the signer to delegate his signing capability to any third party, without disclosing his secret key. A weaker notion, called the *Verifier-only delegatability*, means that only the designated verifier is able to delegate its signing capability without transferring its secret key.

IV. Some Strong DVS Schemes, Revisited

Though Zhang et al.'s scheme [17] and Kang et al.'s scheme [11] have their security proofs, the schemes are shown to be insecure by Kang et al. [12] and Du et al. [5], respectively. In the paper [12], Kang et al. suggested two different designated verifier signature schemes without security proof. Later, one of the schemes is shown to be lack of source hiding. To resist their attacks, Huang et al. [7] suggested an improvement of Kang et al.'s scheme [12], and Du et al. [5] also proposed an improvement of Kang et al.'s scheme [11] by using Cha-Cheon signature scheme [2]. However, we find that the improvements have the same problems with the schemes they attacked in design or security proofs. It seems that the authors of [5,7,12] ignore the reason why their attack is possible. As a result, they made the same mistake in their improvements. In the following, we will pin down the problem that is inherent to the above constructions.

The security proofs of unforgeability of the above schemes were given with reduction technique: if there is a forger against the proposed DVS scheme then one can construct a solver to underlying hardness problem. We remark that the forger in their security definition is modeled as an active adversary who can see all communicated messages and has access to signing oracles. That is, a designated verifier signature scheme must possess unforgeability property under chosen message attack. However, the forger in their proofs is much more powerful than that of security model. To be more precise, we outline their proof procedure: the simulator initially guesses the target identities which the forger will at-

tempt a forgery against. Let ID_S and ID_V denote the target identities of the signer and the designated verifier, respectively. During signing queries of the adversary, the simulator cannot answer any signing queries with respect to S and V. At final stage, the attacker outputs a forgery with which the simulator expects to solve the problem instance. To summarize, the simulator in this game expects the forger to succeed in outputting a forgery with respect to the target signer and verifier. Then, the proofs end up with the claim that this adversary can be used to contradict a given hardness assumption.

But, we stress that the adversary is not allowed to see any signature of the target identities. This means that the adversary having a valid signature could attack their scheme. Most of attacks on designated verifier signature schemes start with a valid signature. As a result, their security proofs do not give the evidence that the proposed constructions are provably secure. This is the reason that the attacks, as well as our attack we will describe in the next section, are possible.

In this section, we have shown that the proof of security of the above mentioned schemes are given against no message attackers while it is no guarantee of any real security of designated verifier signatures. Due to this, the schemes except Du-Wen scheme open attacks even though the authors claimed the schemes are provably secure.

V. Huang-Chou Scheme and Attacks

As we already explained, even though Huang et al.'s scheme was claimed to be secure strong DVS scheme, their proof was somewhat misleading. In this section, we will show that the problem appeared in their proof leads to some attacks, by describing concrete attack. As far as we know there is no known attack against this scheme.

5.1. Review of Huang-Chou Scheme

The Huang et al.'s strong DVS scheme can be described as follows:

Setup.

Let G_1 be an additive cyclic group generated by P and G_2 be a multiplicative cyclic group. The groups are of the same prime order q. Let $e: G_1 \times G_1 \rightarrow G_2$ be a bilinear map and $H_1 : \{0,1\}^* \rightarrow G_1$ and $H_2 : \{0,1\}^* \times G_2 \rightarrow Z_q^*$ be cryptographic hash functions. Then, the key generation center(KGC) picks a random value $s \in Z_q^*$ as the system master secret key and computes the corresponding public key as $P_{pub} = sP$. The system parameter set is $\{G_1, G_2, P, P_{pub}, H_1, H_2, e, q\}$.

Extract.

Given a user's identity ID, KGC computes $Q_{ID} = H_1(ID), S_{ID} = sQ_{ID}$ and returns (S_{ID}, Q_{ID}) to the user ID as his private key and public key.

Sign.

To create signature on m, the signer with an identity A does the following: Select a random value $\alpha \in Z_q^*$, and then compute (δ, ϵ, ξ) as $\delta = \alpha Q_A$, $\epsilon = e(P_{pub}, Q_B)$, $\xi = H_2(m, \epsilon) S_A$; Compute $\sigma = e(\xi + S_A, Q_B)^{\alpha}$. Then (δ, σ) is the signature on m intended for the verifier B.

Verify.

After receiving (δ, σ), the verifier B checks the validity of the signature by testing

whether or not $\sigma = e(\delta, S_B)^{H_2(m,\epsilon)+1}$. The verifier accepts the signature if and only if the equation holds.

TrSim(Transcript Simulation).

The designated verifier B can simulate correct signature transcript for message m to be verified successfully as follows:

(1) Pick a random value $\beta \in Z_q^*$

(2) Compute $\tilde{\delta}$ and $\tilde{\sigma}$ as follows:

$$\tilde{\delta} = \beta\delta \; ; \; \tilde{\sigma} = e(\tilde{\delta}, S_B)^{H_2(m,\epsilon)+1}$$

The simulated signature is of m is $(\tilde{\delta}, \tilde{\sigma})$.

5.2. Analysis of the scheme

Huang et al.'s scheme is vulnerable to universal forgery attack since an attacker who knows a valid signature can freely generate signatures by himself. To see this, the adversary does the following:

(1) Query a signature on m^* with respect to signer A and verifier B to get a signature (δ^*, σ^*)

(2) Compute $e(\delta^*, S_B)$ by

$$(\sigma^*)^{\frac{1}{H_2(m^*, e(P_{pub}, Q_B))+1}}.$$

Now the adversary can freely sign any message on behalf of A to convince B on the message:

(1) Choose a message m that he wants to sign.

(2) Compute $\sigma = e(\delta^*, S_B)^{H_2(m, e(P_{pub}, Q_B))+1}$.

Notice that the resulting values (δ^*, σ) pass the verification test with respect to the signer A and the verifier B, and the adversary can perform above procedure for any message. Therefore, the adversary is able to impersonate the signer A to convince the designated verifier B on messages of his choice.

Remark. If the scheme allows the verification algorithm to check the first component δ of signatures is ever used, then the above attack is avoidable. But, to do so, the verifier should keep track of every signatures intended for him and this, we believe, makes the scheme to be impractical. Moreover, even if we assume that this is not a problem, the scheme still has security problems. First, the scheme does not possess delegatability: the signer A can delegate his signing capability to any third party by sending $e(S_A, Q_B)$. That is, the signer can delegate his signing capability to any third party without disclosing his secret key. Second, the scheme does not satisfy the strongness property: In strong DVS, anyone should not be able to verify the validity of signatures without verifier's secret key even if the verifier is assumed to be honest and not to forge (or simulate) signatures. However, an attacker who knows one valid signature (δ, σ) can compute $e(Q_A, S_B)$ via the above attack. With this value, the attacker can easily verify the validity of the subsequent signatures without the secret key of the designated verifier.

VI. Du-Wen scheme

Du et al. showed that Kang et al.'s scheme is insecure and suggested an improvement based on the Cha-Cheon signature scheme to resist their attacks. Though the authors claimed that their improved scheme is secure and gave its security proofs, the proof also has the same problem as we explained before. In this section, we review the Du et al.'s improvement and then give its correct security proof of unforgeability.

6.1. The Scheme

Setup.

Let G_1 be an additive cyclic group gen-

erated by P and G_2 be a multiplicative cyclic group. The groups are of the same prime order q. Let $e: G_1 \times G_1 \rightarrow G_2$ be a bilinear map and $H_1 : \{0,1\}^* \rightarrow G_1$ and $H_2 : \{0,1\}^* \times G_2 \rightarrow Z_q^*$ be cryptographic hash functions. Then, the key generation center(KGC) picks a random value $s \in Z_q^*$ as the system master secret key and computes the corresponding public key as $P_{pub} = sP$. The system parameter set is $\{G_1, G_2, P, P_{pub}, H_1, H_2, e, q\}$.

Extract.

Given a user's identity ID, KGC computes $Q_{ID} = H_1(ID), d_{ID} = sQ_{ID}$ and returns (d_{ID}, Q_{ID}) to the user ID as his private key and public key.

Sign.

To create signature on m, the signer with an identity A does the following:

(1) Choose a random value $r \in Z_q^*$ and compute $t = rQ_A$.

(2) Set $h = H_2(m,t)$.

(3) Compute $T = (r+h)d_A$ and $\sigma = e(T, Q_B)$.

The signature on the message m is (t,σ)

Verify.

After receiving (t,σ), the designated verifier B checks the validity of the signature by testing whether or not $\sigma = e(t + hQ_A, d_B)$. If it does not hold, he rejects.

TrSim.

At this stage, the designated verifier B can simulate correct signature transcript for message m to be verified successfully as follows:

(1) Choose a random value $k' \in Z_q^*$ and compute $t' = k' Q_A$.

(2) Set $h' = H_2(m,t')$

(3) Compute $\sigma' = e(t' + h' Q_A, d_B)$.

The simulated signature is (t',σ') on m is a valid one in the sense that it passes the verification algorithm.

In the paper [5], the adversary in the proof only reflects no-message attack in the sense that the forger should create a valid signature with respect to the signer A and the designated verifier B without seeing any signature corresponding to A and B. On the other hand, their construction seems to be secure since they used well-studied signature scheme as building block. Indeed, we can correct their proof in the security model given in the section 3, so as to show the improvement is actually secure strong designated verifier signature scheme.

Theorem. The Du-Wen designated verifier signature scheme is unforgeable under adaptively chosen message. That is, if there is a forger F to the Du-Wen designated verifier signature scheme which has running time τ and advantage ϵ with $\epsilon \geq 10(q_S+1)(q_S+q_{H_2})/q$ then one can build an attacker (simulator) to solve the BCDH problem within the expected running time $\tau' \leq \frac{120686\, q_{H_1}^2\, q_{H_2}\, \tau}{\epsilon(1-1/q)^2}$ where q_{H_1}, q_{H_2} and q_S denote the maximal number of queries of H_1, H_2 and the signing queries, respectively.

Proof. For security, we show that any adversary F that can break the security of the scheme with non-negligible probability ϵ after making at most q_{H_1}, q_{H_2} hash queries corresponding to H_1, H_2, respectively, and requesting q_K public keys can be used to build an adversary S that solves the BCDH problem in G_1. On input a bilinear Diffie-Hellman instance P, aP, bP, cP, the BDH adversary S simulates the unforgeability security game for F as follows: To begin the simulation, S guesses which one F will attempt a forgery

against. We denote the target identities ID_A and ID_B, where ID_A is the signer and ID_B is the designated verifier's identity. The simulator prepares two hash tables H_1-list and H_2 -List for the corresponding hash functions, and the master public key P_{pub} as $P_{pub}=cP$ and implicitly set the virtual master secret key s as c. Of course, the simulator does not know the secret key.

H_1 Oracle Query: For each query to H_1 on input ID_i, the simulator checks if there is an entry in the table H_1-List. (The table is initially set to empty) If so, it outputs the corresponding value. Otherwise, it outputs aP if $ID_i=ID_A$, bP if $ID_i=ID_B$, or k_iP by choosing a random k_i.

Extract Query: For each key extraction query on input ID_i, the simulator checks if ID_i is ID_A or ID_B. If so then it aborts. Otherwise, it does the following: (i) Lookup H_1-List to check $(ID_i, H_1(ID_i)=k_iP)$ is already in the list. (ii) If so, returns k_iP_{pub} as the private key d_{ID_i} (iii) Otherwise, it chooses a random value $k_i \in Z_q^*$, records (ID_i, k_iP) in the table, and returns k_iP_{pub} as the private key d_{ID_i}

H_2 Query: For each query to H_2 on input (m,t) where m is a message and a group element $t \in G_1$, the simulator checks if there is an entry in H_2-List. If so, it returns the corresponding value h_i. Otherwise, it chooses h_i at random and returns the value as the response, and then add (m_i, t, h_i) to the H_2-List.

Sign Query: For each signing query on input (ID_i, ID_j, m_i) where ID_i is the signer and ID_j is the designated verifier, and m_i is a message to be signed, the simulator creates the signature using control over the output of H_1 and H_2 as follows:

(1) If $ID_i=ID_A$ then it chooses random values r_i and h_i and computes $t_i=r_iP-h_iaP$. If (m_i,t_i) is already in the H_2-List then it recomputes t_i with different h_i, computes $\sigma_i=e(r_iP_{pub}, Q_j)$, add (m_i,t_i,h_i) to H_2-List and returns (t_i,σ_i) as the signature on m_i

(2) If $ID_i=ID_B$ then the simulator can respond in a similar way as the above case.

(3) If $ID_i \notin \{ID_A, ID_B\}$ then it selects r_i at random, and computes $t_i=r_iQ_i$. If (m,t_i) is in the H_2-List, it takes the corresponding value, otherwise, it selects a random value h_i. Next it computes $\sigma_i=e\left((r_i+h_i)d_{ID}, Q_{ID_j}\right)$ and returns (t_i,σ_i) as the signature on m_i. It adds (m_i,t_i,h_i) to H_2-List.

After a number of queries, F will output a purported forgery (ID_i, ID_j, m, σ). If $(i,j)\neq(A,B)$ then the attacker S guessed the wrong target signer/designated verifier and must abort. If $Verify_{sk_B}(m,\sigma,ID_A)\neq 1$ or (m,σ) is the result of any signing oracle query, the adversary F has failed, so S also aborts.

Now we analyze the probability that the simulator S completes the simulation without aborting so as to get a successful forgery (ID_A, ID_B, m, σ). We remark that

$$\Pr\left[(ID_i, ID_j, t, \sigma) \text{ is valid}\right] \geq \epsilon;$$

$$\Pr\left[i,j \in [q_{H_1}] \mid (ID_i, ID_j, t, \sigma) \text{is valid}\right] \geq 1-1/q^2 \ ;$$

$$\Pr\left[\begin{matrix}(i,j)=(A,B) \mid \\ i,j \in [q_{H_1}] \wedge (ID_i, ID_j, t, \sigma) \text{is valid}\end{matrix}\right] \geq \frac{1}{q_{H_1}(q_{H_1}-1)}$$

From the above equations, we have the bound

$$\Pr\left[\begin{matrix}(ID_i, ID_j)=(ID_A, ID_B) \\ \wedge (ID_i, ID_j, t, \sigma) \text{is valid}\end{matrix}\right] \geq \epsilon\left(1-\frac{1}{q^2}\right)\frac{1}{q_{H_1}^2}$$

Applying the Forking Lemma [14] (or the Reset Lemma [1]) with the same random tape but different choices of H_2, S finally gets two forgery (m,t,h,σ) and (m,t,h',σ') where $h\neq h'$, $\sigma=e(t+hQ_A, d_B)$ and $\sigma'=e(t+h'Q_A, d_B)$.

Then the simulator computes the BDH solution $e(P,P)^{abc}$ as follows:

$$\frac{\sigma}{\sigma'} = e((h-h')Q_A, d_B)\ ;$$

$$\left(\frac{\sigma}{\sigma'}\right)^{\frac{1}{h-h'}} = e(Q_A, d_B) = e(aP, cbP) = e(P,P)^{abc}$$

If the forger F against the signature scheme has advantage $\epsilon \geq 10(q_S+1)(q_S+q_{H_2})/q$ then, from the Forking Lemma, the expected time for the simulator S to solve the BCDH problem is bounded above by $120686 q_{H_1}^2 q_{H_2} \tau \cdot 1/\epsilon \cdot 1/(1-1/q)^2$ as required.

The other security requirements, such as non-transferability and strongness, are analyzed in the same way that Du et al. did. We remark that Huang et al. argued that Du et al.'s scheme does not provide the source hiding because the verification equation is $\sigma = e(t+hQ_A, d_B)$ and the verifier uses the signer's public key Q_A for doing the verification. However, on the contrary to their demonstration, this fact does not mean the lack of source hiding. Indeed, since the verifier can simulate the transcripts the adversary cannot tell the source even if the private keys are revealed. Moreover, even if the verifier is honest, the attacker cannot know the source without the knowledge of the verifier's private key.

VII. Conclusion

In this paper, we have pointed out that the security proofs of some designated verifier signature schemes do not capture real adversary but only reflect no message attackers. We also have presented concrete attacks on the Huang-Chou DVS scheme. Du-Wen DVS (improved) scheme has the same problem especially in the security proof of unforgeability though it is secure. To show Du-Wen scheme is a secure one, we have given correct security proof of unforgeability against adaptively chosen message. Our work alerts to the possibility of danger appeared in DVS schemes, as well as other cryptographic schemes, without rigorous analysis.

참 고 문 헌

[1] M. Bellare and A. Palacio, "GQ and Schnorr Identification Schemes: Proofs and Security against Impersonation under Active and Concurrent Attacks," Crypto'02, LNCS 2442, Springer-Verlag, pp. 162-177, 2002.

[2] J.C. Cha and J.H. Cheon, "An identity-based signature from gap Diffie-Hellman groups," PKC'03, LNCS 2567, Springer-Verlag, pp. 18-30, 2003.

[3] D. Chaum, "Private signature and proof systems?, US Patent, No. 5493614, 1996.

[4] D. Chaum and H. van Antwerpen, "Undeniable signature," Crypto'89, LNCS 485, Springer-Verlag, pp. 212-216, 1990.

[5] H. Du and Q. Wen, "Attack on Kang et al.'s Identity-based strong designated verifier signature scheme," IACR ePrint 2008-297, 2008.

[6] F. Hess, "Efficient identity based signature schemes based on pairing," SAC 2002, LNCS 2595, Springer-Verlag, pp. 310-324, 2002.

[7] H. Huang and J. Chou, "A provably secure really source hiding designated verifier signature scheme based on random oracle model," IACR ePrint 2009-348, 2009.

[8] X. Huang, W. Susilo, Y. Mu, and F. Zhang, "Short designated verifier signature scheme and its identity-based variants," International Journal of Network Security, Vol. 6, No. 1, pp. 82-93, Jan. 2008.

[9] X. Huang, W. Susilo, Y. Mu and F. Zhang, "Short (identity-based) designated verifier signature schemes," ISPEC 2006,

LNCS 3903, Springer-Verlag, pp. 214-225, 2006.

[10] M. Jakobsson, K. Sako and R. Impagliazzo, "Designated Verifier Proofs and Their Applications," Eurocrypt'96, LNCS 1070, Springer-Verlag, pp. 142-154, 1996.

[11] B. Kang, C. Boyd and Ed Dawson, "Identity-based strong designated verifier signature schemes: Attacks and new construction," Computers & Electrical Engineering, Volume 35, Issue 1, Elsevier, pp. 49-53, 2009.

[12] B. Kang, C. Boyd and Ed Dawwon, "A novel identity-based strong designated verifier signature scheme," Journal of Systems and Software, Volume 82, Issue 2, Elsevier, pp. 270-273, 2009.

[13] K. Kumar, G. Shailaja and A. Saxena, "Identity based strong designated verifier signature scheme", IACR ePrint 2006-134, 2006.

[14] D. Pointcheval, J. Stern, "Security arguments for digital signatures and blind signatures," Journal of Cryptology, Vol. 13, No. 3, Springer-Verlag, pp. 361-396, 2000.

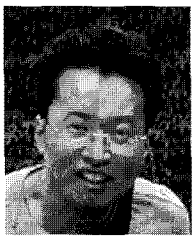
[15] S. Saeednia, S. Kremer, and O. Markovitch, "An efficient strong designated verifier signature scheme," In ICISC 2003, LNCS 2869, Springer-Verlag, pp. 40-54, 2003.

[16] W. Susilo, F. Zhang, and Y. Mu, "Identity-based strong designated verifier signature scheme," In ACISP 2004, LNCS 3108, Springer-Verlag, pp. 313-324, 2004.

[17] J. Zhang and J. Mao, "A novel ID-based designated verifier signature scheme," Information Sciences, Volume 178, Issue 3, pp. Elsevier, 766-773, 2008.

〈著者紹介〉

김 기 태 (Kitae Kim) 정회원
1997년 2월: 건양대학교 수학과 졸업
2000년 2월: 인하대학교 수학과 석사
2009년 8월: 인하대학교 수학과 박사
2009년 9월~현재: 인하대학교 정보통신대학원 박사 후 연구원
〈관심분야〉 전자서명, 암호분석, 대수적 정수론



양 대 헌 (DaeHun Nyang) 정회원
1994년 2월: 한국과학기술원 과학기술대학 전기·전자공학과 졸업
1996년 2월: 연세대학교 컴퓨터과학과 석사
2000년 8월: 연세대학교 컴퓨터과학과 박사
2000년 9월~2003년 2월: 한국전자통신연구원 정보보호연구본부 선임연구원
2003년 2월~현재: 인하대학교 컴퓨터정보공학부 부교수
〈관심분야〉 암호 이론, 암호 프로토콜, 인증 프로토콜, 무선 인터넷 보안

이 경 희 (KyungHee Lee) 정회원
1993년 2월: 연세대학교 컴퓨터과학과 학사
1998년 8월: 연세대학교 컴퓨터과학과 석사
2004년 2월: 연세대학교 컴퓨터과학과 박사
1993년 1월~1996년 5월: LG소프트(주) 연구원
2000년 12월~2005년 2월: 한국전자통신연구원 선임연구원
2005년 3월~현재: 수원대학교 전기공학과 조교수
〈관심분야〉 바이오인식, 정보보호, 컴퓨터비전, 인공지능, 패턴인식