Security Remarks on a Convertible Nominative Signature Scheme

Guilin Wang and Feng Bao

Institute for Infocomm Research (I^2R)
21 Heng Mui Keng Terrace, Singapore 119613
{glwang,baofeng}@i2r.a-star.edu.sg

Abstract. A nominative signature scheme allows a nominator (i.e. the signer) and a nominee (i.e. a designated verifier) to jointly generate and publish a signature so that *only* the nominee can check the validity of a nominative signature and further convince a third party to accept this fact. Recently, Huang and Wang proposed such a new scheme at ACISP 2004, and claimed that their scheme is secure under some standard computational assumptions. In this paper, we remark that their scheme is *not* a nominative signature in fact, since it fails to meet the crucial security requirement: *verification untransferability*. Specifically, we identify an adaptively chosen-message attack against their scheme such that the nominator can determine the validity of a new message-signature pair with some indirect help from the nominee. Moreover, we point out that using our attack the nominator is further able to demonstrate the validity of nominative signatures to a third party. Therefore, the Huang-Wang scheme does not meet *confirmation/disavowal untransferability* either.

Keywords: Nominative signature, digital signature, attacks, information security.

1 Introduction

As an important primitive in modern cryptography, digital signatures are widely used to provide integrity, authenticity, and non-repudiation for authenticating electronic messages. Standard digital signatures are *universally or publicly verifiable*. That is, according to a publicly known signature verification algorithm anybody can check whether an alleged signature-message pair is valid or not with respect to a given public key. However, this may not be a desired property in some situations, where messages to be authenticated are personally private or commercially sensitive. To restrict the universal verifiability, therefore, some variants of digital signatures have been proposed, such as undeniable signatures (US), designated confirmer signatures (DCS), and nominative signatures (NS).

In undeniable signature schemes [4, 15, 9], the validity of a signature can *only* be verified under the cooperation of the signer. Undeniable signatures find various applications, such as licensing softwares, e-cash, e-voting, e-auctions,

etc. In [6, 1, 11], designated confirmer signatures (DCS) are further proposed to guarantee that the validity of signatures can also be verified under the help of a semi-trusted party, called designated confirmer. This is a useful enhancement for verifiers, since they may need to check a signature's validity even in the case the signer is unavailable, due to some subjective or objective reasons.

Compared with these schemes, nominative signatures [14, 13] hand over the power of signature verification to the verifier. That is, without the help of a designated verifier (called *nominee*), anybody including the signer (called *nominator*) cannot determine whether an alleged signature-message pair is valid or not. Actually, nominative signatures can be considered as the dual concept of undeniable signatures. In addition, as suggested in [14, 13], nominative signatures have potential applications in the scenarios where a signed message is personally private or commercially sensitive, such as a tax bill, a medical examination report, an ID certificate, etc.

In 1996, Kim et al. [14] first introduced the concept of *nominative signatures*. Intuitively, a nominative signature scheme allows a nominator (or signer) and a nominee (or verifier) to jointly generate and publish a signature for an arbitrary message. Different from standard signatures, however, *only* the nominee (holding his/her own private key) can verify the validity of a published nominative signature. Furthermore, if necessary, the nominee can also convince a third party to accept this fact by proving (in zero-knowledge) that such a signature is indeed issued by a specific nominator.

At ACISP 2004, Huang and Wang [13] mounted an attack showing that Kim et al.'s scheme is *not* nominative, since the nominator can also verify and prove the validity of a signature to a third party. To avoid this weakness, they further proposed a new nominative signature scheme. Actually, their scheme are also *convertible*. That is, the nominee can also convert a nominative signature into a publicly verifiable one, if necessary.

Soon, Susilo and Mu [18] claimed that the Huang-Wang scheme is not nominative either. Specifically, they described three deterministic algorithms that allow the nominator alone (i.e., without any help from the nominee) to (a) verify the validity of a nominative signature, (b) convince a third party that the signature is valid, and (c) convert the signature into a publicly verifiable one. However, Guo et al. [12] pointed out that all these attacks are actually *invalid*. In particular, they showed that there exists *no deterministic algorithm* allowing the nominator to check the validity of a published nominative signature if the decisional Diffie-Hellman (DDH) problem is hard. So, it is still not clear whether the Huang-Wang scheme is a truly secure nominative signature against adaptively chosen-message attacks.

In this paper, we present a detailed security analysis of the Huang-Wang scheme [13] and show that it is indeed *not* a secure nominative signature scheme, since it fails to meet the crucial security requirement: *verification untransferability*. Specifically, we identify an adaptively chosen-message attack against their scheme such that the nominator can determine the validity of a new message-signature pair with some indirect help from the nominee. Moreover,

we point out that using this attack the nominator is further able to prove the validity of nominative signatures to a third party. Therefore, the Huang-Wang scheme also *does not* satisfy the *confirmation/disavowal untransferability*, which requires that except the nominee anybody should be unable to convince a third party accepting the validity of an alleged signature. In addition, we analyze the reasons why their scheme is insecure.

This paper is organized as follows. We first review Huang-Wang scheme in Section 2, and then present our attacks in Section 3. Finally, a brief conclusion is given in Section 4.

2 Review of the Huang-Wang Scheme

2.1 Security Model of Convertible Nominative Signature

Definition 1 (Syntax of Convertible Nominative Signatures). *According to [13], a convertible nominative signature scheme consists of a six-tuple (KG,Sig,Ver,Conf,Conv,UVer) of algorithms and protocols:*

1. *KG is a probabilistic algorithm to produce public/secret key pairs (pk_s, sk_s) and (pk_v, sk_v) for a nominator S and a nominee V respectively, when a security parameter 1^n is given.*
2. *Sig is an interactive protocol between the nominator S and nominee V to generate a nominative signature σ for a given message m.*
3. *Ver is a deterministic algorithm so that the nominee V can verify the validity of a presumed signature-message pair (σ, m) using V's private key x_v.*
4. *Conf is an interactive confirmation/disavowal protocol between the nominee V and a third party T so that V can convince T to accept the validity or invalidity of a presumed nominative signature. Conf should satisfy the regular requirements of both completeness and soundness.*
5. *Conv is a polynomial-time algorithm that allows the nominee V to convert a nominative signature into a universally verifiable signature.*
6. *Uver is a deterministic algorithm that allows anybody to verify the validity of a converted nominative signature.*

In [13], a convertible nominative signature is called *secure* if it satisfies the following three security requirements: *unforgeability*, *verification untransferability*, and *confirmation/disavowal untransferability*. Unforgeability requires that except the nominator, anybody including the nominee cannot forge a valid nominative signature on behalf of the nominator with non-negligible probability. The essential meaning of verification untransferability is that anybody including the nominator cannot determine the validity of a presumed nominative signature with non-negligible advantage, even if he/she already checked the validity of many other signatures with the nominee. The last requirement means that anybody including the nominator cannot convince a third party to accept the validity or invalidity of a given nominative signature, even if he/she

already interacted with the nominee for many times. In the following, we recall the formal definitions of those three security requirement, which are specified in [13].

Definition 2 (Unforgeability). *Let F be a probabilistic polynomial-time (PPT) algorithm, called forger, who takes input the security parameter 1^n, the nominator S's public key pk_s, the nominee V's public and private key pair (pk_v, sk_v). F can interact with the signer S by running* Sig *to request valid signatures on polynomial-many adaptively chosen messages m_i, can request the execution of* Conf *and* Conv *for polynomial-many adaptively chosen strings, and finally outputs a forged message-signature pair (m, σ) with $m \notin \{m_i\}$. We say a convertible nominative signature is* `unforgeable`*, if for all such F, for any constant $c > 0$, and for sufficiently large n, the probability that F outputs (m, σ) for which at least one of* Ver *or* Conf *outputs 1 is less than n^{-c}. That is,*

$$\Pr\left[\begin{array}{l}(m,\sigma) \leftarrow F^{\mathsf{Sig},\ \mathsf{Conf},\ \mathsf{Conv}}(1^n, pk_s, pk_v, sk_v) : m \notin \{m_i\} \wedge \\ (\mathsf{Ver}(1^n, m, \sigma, pk_s, pk_v, sk_v) = 1 \vee \mathsf{Conf}(1^n, m, \sigma, pk_s, pk_v, sk_v) = 1)\end{array}\right] < n^{-c}.$$

Here, the probability is taken over the coin tosses of F, S, V, m and σ.

Definition 3 (Verification Untransferability). *Let A be a PPT attacking algorithm, which takes input security parameter 1^n, the nominator S's public/private key pair (pk_s, sk_s), the nominee V's public key pk_v, and a presumed signature-message pair (m, σ) which is valid with exact probability 1/2. A can request the execution of* Conf *and* Conv *for polynomial-many adaptively chosen strings except (m, σ), and finally outputs either 0 or 1. We say a convertible nominative signature is* `verification untransferable`*, if for any such PPT algorithm A, for any positive constant $c > 0$, and for any sufficiently large n, the following inequality holds:*

$$\left|\Pr\left[\begin{array}{l}A^{\mathsf{Conf},\ \mathsf{Conv}}(1^n, m, \sigma, pk_s, sk_s, pk_v) \\ = \mathsf{Ver}(1^n, m, \sigma, pk_s, pk_v, sk_v)\end{array}\right] - \frac{1}{2}\right| < n^{-c}.$$

Here, the probability is taken over the coin tosses of A, S, V, m and σ.

Definition 4 (Confirmation/Disavowal Untransferability). *Let A be a PPT attacking algorithm, which takes input the security parameter 1^n, the nominator S's public and private key pair (pk_s, sk_s), the nominee V's public key pk_v, and a target message-signature pair (m, σ) which is valid with exact probability 1/2. A can request the execution of* Conf *and* Conv *for polynomial-many adaptively chosen strings, where A can request execution of* Conf *with the nominee V on the target pair (m, σ) but cannot request execution of* Conv *on (m, σ). Then, at some point, the attacker A as the role of prover and a honest third party T as the role of verifier engage in a confirmation/disavowal protocol* Conf'*, which could be different from* Conf*, to confirm or disavow the given pair (m, σ). When they stop, the third party T outputs either 0 or 1. We say a convertible nominative signature is* `confirmation/disavowal untransferable`*, if for all such*

A and Conf'*, for any constant* $c > 0$*, and for sufficiently large* n*, the following inequality holds:*

$$\left| \Pr \left[\begin{matrix} \mathsf{Conf}'^{\mathsf{Conf},\ \mathsf{Conv}}_{(A,T)}(1^n, m, \sigma, pk_s, sk_s, pk_v) \\ = \mathsf{Ver}(1^n, m, \sigma, pk_s, pk_v, sk_v) \end{matrix} \right] - \frac{1}{2} \right| < n^{-c}.$$

Here, the probability is taken over the coin tosses of A*,* T*,* S*,* V*,* m *and* σ*.*

2.2 The Huang-Wang Scheme

We now review the concrete Huang-Wang nominative signature scheme [13], which works in the discrete logarithm setting. In the following description, notation $x \in_R X$ means that an element x is uniformly chosen from set X at random, while $\|$ denote the concatenation of strings.

1. Key Generation (KG): Let p, q be two large primes such that $q|(p-1)$, and g an element in $\mathbb{Z}_p^*$ of order q. Assume that the discrete logarithm problem (DLP) in the group $\langle g \rangle$ is hard. The nominator S and the nominee V set their public/private key pairs as (y_s, x_s) and (y_v, x_v) respectively, where $x_s, x_v \in_R \mathbb{Z}_q$, $y_s = g^{x_s} \bmod p$, and $y_v = g^{x_v} \bmod p$. In addition, a one-way hash function $H : \{0,1\}^* \to \mathbb{Z}_q$ is publicly available.

2. Signature Generation (Sig): To generate a nominative signature $\sigma = (b, c, s)$ for a message m, the nominator S and nominee V jointly perform as follows.

1) The nominee V first randomly picks $R_1, R_2 \in_R \mathbb{Z}_q^*$, then sends (a, c) to the nominator S by computing

$$a = g^{R_1} \bmod p \quad \text{and} \quad c = y_v^{R_2} \bmod p.$$

2) Upon receiving (a, c), the nominator S chooses $r \in_R \mathbb{Z}_q$, and sends (b, e, s') to V by computing

$$\begin{aligned} b &= ag^{-r} \bmod p, \\ e &= H(y_v \| b \| c \| m), \\ s' &= r - x_s \cdot e \bmod q. \end{aligned} \tag{1}$$

3) Then, nominee V checks whether both of the following equations hold:

$$e \equiv H(y_v \| b \| c \| m) \quad \text{and} \quad a \equiv g^{s'} y_s^e b \bmod p. \tag{2}$$

If not, output "False". Otherwise, nominee V outputs $\sigma = (b, c, s)$ as the nominative signature for message m by setting

$$s = s' + R_2 - R_1 \bmod q. \tag{3}$$

3. Verification (Ver): Given a signature $\sigma = (b, c, s)$ and a message m, the nominee V accepts σ as valid if and only if

$$(g^s y_s^e b)^{x_v} \equiv c \bmod p, \text{ where } e = H(y_v \| b \| c \| m). \quad (4)$$

4. Confirmation and Disavowal (Conf): For an alleged nominative signature $\sigma = (b, c, s)$ for message m, let $e = H(y_v||b||c||m)$ and $d = g^s y_s^e b \bmod p$. The nominee V uses Michels-Stadler's protocol [15] to confirm or disavow the validity of σ via proving $\log_d c = \log_g y_v$ or $\log_d c \neq \log_g y_v$. Refer to Section 2.2 in [13] for the detail how this proof is conducted interactively.

5. Signature Conversion (Conv): To convert a nominative signature $\sigma = (b, c, s)$ into a universally verifiable one, V just needs to release σ together with a non-interactive proof π showing that $\log_d c = \log_g y_v$, where $d = g^s y_s^e b \bmod p$ and $e = H(y_v||b||c||m)$. Please check Section 2.3 in [13] to know how to generate and verify the proof π.

6. Universal Verification (UVer): Anybody can verify the validity of (σ, π) by checking that π is a correct non-interactive proof for $\log_d c = \log_g y_v$, where $d = g^s y_s^e b \bmod p$ and $e = H(y_v||b||c||m)$.

3 Security Remarks on the Huang-Wang Scheme

In [13], the authors provided some security arguments to show that their nominative signature scheme meets the three desirable security requirements: Unforgeability, verification untransferability, and confirmation/disavowal untransferability. Note that those security arguments are informal explanations instead of formal proofs, so the security of Huang-Wang scheme is *not* guaranteed in fact: It may be secure or insecure.

As pointed out in [13], it seems that the unforgeability of Huang-Wang scheme is related to that of Schnorr signature [17]. This can be informally explained as follows. Let a PPT algorithm F be a forger, who is given the security parameter 1^n, the nominator S's public key pk_s, and the nominee V's public and private key pair (pk_v, sk_v). According to Definition 2, F's goal is to forge a signature $\sigma = (b, c, s)$ for a new message m. For this purpose, F can adaptively choose messages m_i and then run the interactive protocol Sig with the signer, under the limitation that m is not equal to any m_i. If F's output $(m, \sigma = (b, c, s))$ is a valid message-signature pair, we have $(g^s y_s^e b)^{x_v} \equiv c \bmod p$, where $e = H(y_v \| b \| c \| m)$. By letting $s' = -s \bmod q$, this implies that the forger F can get a triple (b, c, s') for message m such that $g^{s'} \equiv y_s^e \cdot (bc^{-x_v^{-1}}) \bmod p$, where $e = H(y_v \| b \| c \| m)$. This triple (b, c, s') is very *similar* to a Schnorr signature (r, t) for message m so that $g^t = y_s^e \cdot r \bmod p$, where $e = H(r \| m)$. However, Schnorr signature is proved to be unforgeable if the discrete logarithm is hard [16]. So it is likely that Huang-Wang scheme is also

unforgeable, though elaborated work is needed to formally prove this result by using the forking lemma proposed in [16].

For other two security requirements, however, it is another story. In the following sections, we present some direct attacks to show that the Huang-Wang scheme satisfies *neither* verification untransferability *nor* confirmation/disavowal untransferability.

3.1 Verification Untransferability

In [13], the authors argued that the Huang-Wang scheme satisfies the verification untransferability, because both of the following two statements hold:

(a) A presumed signature $\sigma = (b, c, s)$ is valid for a message $m \Longleftrightarrow \log_g y_v = \log_d c$, where $d = g^s y_s^e b \bmod p$ for $e = H(y_v||b||c||m)$. However, without the knowledge of both x_v (nominee V's private key) and R_2 (a random number selected by V) an attacker A cannot determine the validity of σ unless it resorts to nominee V's direct help on pair (m, σ) or it can solve the decisional Deffie-Hellman (DDH) problem w.r.t the tuple (g, y_v, d, c).

(b) Michels-Stadler's interactive protocol [15] is an untransferable zero knowledge proof for proving whether two discrete logarithm is equal or not.

However, we notice that neither of the above two statements are correct (and shall be explained below). Moreover, based on this observation we can mount a concrete attack on the Huang-Wang scheme so that the verification untransferability is violated.

Statement (a) is invalid, because the attacker A can "solve" the DDH problem in this scenario. This is due to the fact that according to Definition 3, an attacker A for verification untransferability is allowed to access DDH oracle by running Conf with nominee V. Hence, for given (d, c) attacker A can know whether (g, y_v, d, c) is a DH tuple by asking V whether $(g, y_v, \bar{d}, \bar{c})$ is a DH tuple, where $\bar{d} = dg^{\bar{R}_2}$ and $\bar{c} = cy_v^{\bar{R}_2} \bmod p$ are set by the attacker for some randomly chosen number $\bar{R}_2$.

Actually, using the above fact we identify an concrete attack to show that the Huang-Wang scheme is not verification untransferable. In this attack, we assume the attacker A has the knowledge of nominator S's private key sk_s. Note that this is consistent with Definition 3, which formally specified the verification untransferability. In other words, the nominator S is also allowed to be an attacker for this security requirement. Now, we describe the attack in detail.

Attack 1. To check whether an alleged nominative signature $\sigma = (b, c, s)$ is valid for a message m, the attacker A (or the nominator S) can perform as follows.

1. A first selects another message $\bar{m}$, and two random numbers $\bar{R}_1, \bar{R}_2 \in_R \mathbb{Z}_q^*$.
2. Then, using x_s the attacker A can compute a triple $\bar{\sigma} = (\bar{b}, \bar{c}, \bar{s})$ by

$$\begin{aligned} \bar{b} &= bg^{\bar{R}_1} \bmod p, \\ \bar{c} &= cy_v^{\bar{R}_2} \bmod p, \\ \bar{s} &= s + x_s(e - \bar{e}) + (\bar{R}_2 - \bar{R}_1) \bmod q, \end{aligned} \tag{5}$$

where $e = H(y_v||b||c||m)$ and $\bar{e} = H(y_v||\bar{b}||\bar{c}||\bar{m})$.

3. Finally, A interacts with the nominee V to check the validity of message-signature pair $(\bar{m}, \bar{\sigma})$. The attacker A concludes that the presumed message-signature pair (m, σ) is valid, if $(\bar{m}, \bar{\sigma})$ is valid. Otherwise, A concludes $(\bar{m}, \bar{\sigma})$ is invalid.

The correctness of Attack 1 can be justified as follows. According to Eq.(5), we have

$$\begin{aligned} g^{\bar{s}} y_s^{\bar{e}} \bar{b} &= g^s y_s^{e-\bar{e}} g^{\bar{R}_2 - \bar{R}_1} y_s^{\bar{e}} b g^{\bar{R}_1} \bmod p \\ &= (g^s y_s^e b) g^{\bar{R}_2} \bmod p. \end{aligned}$$

Therefore, (m, σ) is valid $\Leftrightarrow$ $[\exists R_2 \ s.t. \ g^s y_s^e b = g^{R_2} \bmod p \wedge c = y_v^{R_2} \bmod p] \Leftrightarrow$ $[g^{\bar{s}} y_s^{\bar{e}} \bar{b} = g^{\bar{R}_2 + R_2} \bmod p \wedge \bar{c} = c y_v^{\bar{R}_2} = y_v^{\bar{R}_2 + R_2} \bmod p] \Leftrightarrow (\bar{m}, \bar{\sigma})$ is valid.

Now, we return to Statement (b): It is also false, since Michels-Stadler's protocol [15] is *not* zero-knowledge, as first pointed out by Camenisch and Shoup (See Section 5 of [3]). Namely, (in our setting) the value of d^{x_v} is additionally revealed in the case of $\log_g y_v \neq \log_d c$ and the verifier dishonestly selects d such that he knows $\log_g d$. This weakness shall naturally affect the formal security of the Huang-Wang scheme, though we do not any find direct attack by using it [1].

3.2 Confirmation/Disavowal Untransferability

According to Definition 4, the confirmation/disavowal untransferability requires that given a presumed message-signature pair (m, σ), any PPT attacker A (including the nominator S but not the nominee V) cannot convince a third party T to accept the validity or invalidity of this pair (m, σ), where A is allowed to run Conf with V on any string and Conv with V on any string other than the target (m, σ).

However, we find out that using Attack 1 against verification untransferability as a subroutine, we can break the confirmation/disavowal untransferability as well. We now briefly present this attack.

Attack 2. To convince a third party T accepting the validity or invalidity of a target message-signature pair (m, σ), an attacker (who knows the private key x_s of the nominator S) can perform as follows.

1. From the given pair (m, σ), the attacker A first creates a new message-signature pair $(\bar{m}, \bar{\sigma})$ as in Attack 1 (see Eq. (5)).

[1] In the submission version of this paper, we did showed "an attack" by employing this flaw in the ZK protocol. However, when preparing this final version we noticed that "this attack" is actually invalid since it requires the attacker know the value of $\log_g d$, i.e., the random number R_2 selected by the nominee.

2. After that, A asks the nominee V to convert $(\bar{m}, \bar{\sigma})$. Therefore, A can get a non-interactive proof $\bar{\pi}$ that shows whether $\bar{\sigma}$ is a valid signature for message $\bar{m}$.
3. Then, the attacker A forwards $(\bar{m}, \bar{\sigma}, \bar{\pi}, \pi')$ to the third party T, where π' is a non-interactive zero-knowledge proof showing that $\bar{\sigma} = (\bar{b}, \bar{c}, \bar{s})$ is properly generated according to Eq. (5), i.e., A knows that there are two random numbers $(\bar{R}_1, \bar{R}_2)$ so that the following conditions hold simultaneously:

$$\bar{b}/b = g^{\bar{R}_1} \bmod p \wedge \bar{c}/c = y_v^{\bar{R}_2} \bmod p \wedge g^{\bar{s}} y_s^{\bar{e}} \bar{b}/(g^s y_s^e b) = g^{\bar{R}_2} \bmod p. \quad (6)$$

4. Finally, the third party T validates whether π' is a correct proof for Eq. (6). If no, halt. Otherwise, T further checks whether $(\bar{\sigma}, \bar{\pi})$ is valid for message $\bar{m}$. If yes, T concludes (m, σ) is valid. Otherwise, (m, σ) is invalid.

Note that using the well-known technique called *signature proof of knowledge* [2], it is very easy for A to issue proof π' for Eq. (6). Alternatively, A can run an interactive protocol with T to prove that the conditions in Eq.(6) hold. Moreover, during this procedure it is infeasible for T to derive x_s since this proof is zero-knowledge.

The correctness of Attack 2 is almost obvious, since $(\bar{m}, \bar{\sigma})$ and (m, σ) have the same validity or invalidity and the attacker does not ask the nominee V to convert σ at all. Therefore, Attack 2 breaks the confirmation/disavowal untransferability of Huang-Wang scheme. That is, at least for the nominator the Huang-Wang is not confirmation/disavowal untransferable.

3.3 Countermeasures

As we mentioned above, the Huang-Wang scheme employs a flawed building block: Michels-Stadler's protocol, which is designed to prove whether or not two discrete logarithms are equal. However, this protocol is *not* zero-knowledge, contrary to the claims made in [15]. To avoid this weakness, we can choose two truly zero-knowledge protocols from [5] and [3] to prove the equality or inequality of two discrete logarithms, respectively.

Moreover, according to the specification of confirmation/disavowal untransferability (Definition 4), in the setting of nominative signatures one should use *concurrent zero-knowledge* (CZK) protocols rather than *special honest-verifier zero-knowledge* (SHVZK) protocols [3]. The reason is that an attacker here may act as an arbitrary cheating verifier during the execution of Conf protocol to confirm or disavow an alleged nominative signature. Fortunately, by using the techniques suggested in any of [7, 8, 10], we can easily transform SHVZK protocols from [5] and [3] to CZK protocols. Using such CZK protocols to confirm or disavow nominative signatures, the confirmation/disavowal untransferability can be guaranteed. Moreover, the formal proofs can be adapted from that given in [1, 11], where the transcripts of verifying designated confirmer signatures are also required to be untransferable.

However, we do not find any effective countermeasure to prevent Attack 1 at this moment. In fact, we believe this is the essential security flaw of Huang-Wang nominative signature scheme.

4 Conclusions

In this paper, we presented a security analysis of the Huang-Wang convertible nominative scheme [13]. According to our results, the Huang-Wang scheme is not secure, since it fails to meet two desirable security requirements: *verification untransferability* and *confirmation/disavowal untransferability*. Specifically, we identified two attacks to show that their scheme violates those two security requirements. In fact, those two attacks are due to an essential design flaw in the scheme. In addition, we also remarked that the Huang-Wang scheme employs a flawed zero-knowledge protocol. Moreover, we pointed out the reasons why their scheme is insecure. As the future work, it is interesting to consider how to prevent our Attack 1 against verification untransferability and how to design newly secure nominative signatures.

References

1. J. Camenisch, and M. Michels. Confirmer Signature Schemes Secure Against Adaptive Adversaries. In: *Proc. of Advances in Cryptology - EUROCRYPT '00*, LNCS 1870, pp. 243-258. Springer-Verlag, 2000.
2. J. Camenisch and M. Stadler. Efficient Group Signature Schemes for Large Groups. In: *Proc. of Advances in Cryptology - CRYPTO '97*, LNCS 1294, pp. 410-424. Springer-Verlag, 1997.
3. J. Camenisch and V. Shoup. Practical Verifiable Encryption and Decryption of Discrete Logarithms. In: *Proc. of Advances in Cryptology - CRYPTO '03*, LNCS 2729, pp. 126-144. Springer-Verlag, 2003.
4. D. Chaum and H. Antwerpen. Undeniable Signatures. In: *Proc. of Advances in Cryptology - CRYPTO '89*, LNCS 435, pp. 212-216. Springer-Verlag, 1989.
5. D. Chaum and T. P. Pedersen. Wallet Database with Observers. In: *Proc. of Advances in Cryptology - CRYPTO '92*, LNCS 740, pp. 89-105. Springer-Verlag, 1993.
6. D. Chaum. Designated Confirmer Signatures. In: *Proc. of Advances in Cryptology - EUROCRYPT '94*, LNCS 950, pp. 86-91. Springer-Verlag, 1994.
7. R. Cramer, I. Damgård, and P. MacKenzie. Efficient Zero-Knowledge Proofs of Knowledge Without Intractability Assumptions. In: *Proc. of PKC '00*, LNCS 1751, pp. 354-373. Springer-Verlag, 2000.
8. I. Damgård. Efficient Concurrent Zero-Knowledge in the Auxiliary String Model. In: *Proc. of Advances in Cryptology - EUROCRYPT '00*, LNCS 1807, pp. 418-430, Springer-Verlag, 2000.
9. S. D. Galbraith and W. Mao. Invisibility and Anonymity of Undeniable and Confirmer Signatures. In: *Proc. of CT-RSA '03*, LNCS 2612, pp. 80-97. Springer-Verlag, 2003.

10. R. Gennaro. Multi-trapdoor Commitments and Their Applications to Proofs of Knowledge Secure Under Concurrent Man-in-the-Middle Attacks. In: *Advances in Cryptology - CRYPTO '04*, LNCS 3152, pp. 220-236. Springer-Verlag, 2004.
11. C. Gentry, D. Molnar, and Z. Ramzan. Efficient Designated Confirmer Signatures without Random Oracles or General Zero-knowledge Proofs. In: *Advances in Cryptology - ASIACRYPT 2005*, LNCS 3788, pp. 662-681. Springer-Verlag, 2005.
12. L. Guo, G. Wang, and D. Wong. Further Discussions on the Security of a Nominative Signature Scheme. *IACR ePrint archive*, http://eprint.iacr.org/2006/007.
13. Z. Huang and Y. Wang. Convertible Nominative Signatures. In: *Proc. of Information Security and Privacy (ACISP '04)*, LNCS 3108, pp. 348-357. Springer-Verlag, 2004.
14. S.J. Kim, S.J. Park, and D.H. Won. Zero-Knowledge Nominative Signatures. In: *Proc. of PragoCrypt' 96, International Conference on the Theory and Applications of Cryptology*, pp. 380-392, 1996.
15. M. Michels and M. Stadler. Efficient Convertible Undeniable Signature Schemes. In: *Proc. of 4th Annual Workshop on Selected Areas in Cryptography (SAC'97)*, pp. 231-244, 1997.
16. D. Pointcheval and J. Stern. Security Arguments for Digital Signatures and Blind Signatures. *Journal of Cryptology*, 13(3): 361-396, 2000.
17. C.P. Schnorr. Efficient Signature Generation by Smart Cards. *Journal of Cryptology*, 4(3): 161-174, 1991.
18. W. Susilo and Y. Mu. On the Security of Nominative Signatures. In: *Proc. of Information Security and Privacy (ACISP '05)*, LNCS 3547, pp. 329-335. Springer-Verlag, 2005.