A Note on Signature Transformation Attacks and Confirmer Signatures

Victor K. Wei

Dept. of Information Engineering, The Chinese Univ. of Hong Kong, Hong Kong
kwwei@ie.cuhk.edu.hk

October 4, 2006

Abstract. Camenisch and Michels in Eurocrypt 2000 introduced the signature transformation attack on designated confirmer signatures (DCS). We apply this attack on Gentry, et al. Asiacrypt 2005's DCS, and on Goldwasser, et al. TCC 2004's DCS before repairing them. We also optimize efficiencies of the former DCS' confirmation and disavowal protocols. The undeniable signature of Laguillaumie, et al. in Indocrypt 2005 is upgraded using techniques above.

1 The results

Chaum [7] introduced the DCS (Designated Confirmer Signature). The signature verification requires the interaction with a confirmer who was designated by the signer when the signature was created. The motivation was to split the power to sign and the power to confirm in order to mitigate the overpower of the signer. Several applications benefit from such a power splitting [7, 2].

T. Okamoto [15] gave a formal security model for DCS, and a polynomial equivalence reduction between DCS and public-key encryption. Camenisch and Michels [5] presented an upgraded DCS security model which included the *signature transformation attacker* who can query the confirmation oracle with adaptively designed signer public key which is not obtained by the given key generation protocol. [5] also gave concrete instantiations, using the RSA signature and the Cramer-Shoup encryption. The confirmation and disavowal were not very efficient as they involved double discrete logarithms or range proofs.

Goldwasser and Waisbard [12] and Gentry, et al. [11] presented DCS without random oracles. [11]'s DCS has $O(1)$-size and the state-of-the-art efficiency of costing 10 (resp. 41) exponentiations in confirm (resp. disavow).

The **contributions** of this note: We apply Camenisch and Michels [5]'s signature transformation attack on the two DCS's above, before repairing them. The attack on [11] is within their model while the attack on [12] is beyond their model. We also optimize the efficiencies of [11]'s confirmation and disavowal protocols. The undeniable signature of Laguillaumie, et al. [14] is upgraded using techniques above.

In this brief note, we do not include the security model or other definitions of terminologies. Consult the original references for details [5, 12, 11].

Attack and repair on Gentry, et al. [11].

Review. The main DCS in [11] on message m is $\sigma' = (\sigma^*, \phi, c)$, where

$$\phi = Commit(m, r) = g^m h^r \in QR_{n^2} \tag{1}$$
$$c = \mathsf{Enc}(\mathsf{pk}_C, r) = (u_1, u_1, u_3, u_4) = (g_1^\rho, g_2^\rho, d_3^\rho g_0^r, (d_1 d_2^\alpha)^\rho) \in QR_{n^2}^4 \tag{2}$$
$$\sigma^* = \mathsf{Sign}(\mathsf{sk}_S, (\phi, c, \mathsf{pk}_S)) \tag{3}$$

where $\alpha = Hash(u_1,\ u_2,\ u_3)$. The commitment is Pedersen's commitment. The base $g_0 = n + 1$ allows the confirmer to compute the *partial discrete logarithm* in the Paillier system, and thus decrypt r. Sign is any secure signature without random oracles, with signer private key sk_S. The confirmer public key pk_C consists of $d_1 = g_1^{x_1} g_2^{x_2}$, $d_2 = g_1^{y_1} g_2^{y_2}$, $d_3 = g_1^z$. Its private key is $\mathsf{sk}_C = (x_1,\ x_2,\ y_1,\ y_2,\ z)$. In this, the main instantiation of [11], the confirmer uses the Cramer-Shoup encryption [9] instantiated in the group $\mathbb{Z}_{n^2}$ [6].

Attack Hypotheses. The attacker needs the following hypotheses:

1. Knowing the private key $\mathsf{sk}_{S'}$ of a signer S', including $S' = S$.
2. Query access to a confirmation oracle which, upon common inputs including a message $\bar{m}$, a signer public key pk_S, and a putative DCS $\bar{\sigma}'$, will confirm or disavow the DCS $\bar{\sigma}'$. Except when the queried tuple $(\bar{m},\ \mathsf{pk}_{S'},\ \bar{\sigma}')$, $\bar{\sigma}' = (\bar{\sigma}^*,\ \bar{\phi},\ \bar{c})$, shares the same m, or the same pk_S, or the same σ^*, or the same ϕ as the attacker's target tuple. Note queries with the same $\bar{c} = c$ are allowed.

Attack consequence and procedure: Given a (message, signer private key, putative DCS) tuple, denoted $(m,\ \mathsf{pk}_S,\ \sigma')$, our attacker computes the validity of the putative DCS (i.e. distinguishes a valid DCS from a non-valid simulation DCS) and consequently cracks the security of the DCS scheme by cracking its transcript simulatability [11]. It does so by interacting once with the confirmation oracle with the following transformed tuple: $(\bar{m},\ \mathsf{pk}_{S'},\ \bar{\sigma}')$, $\bar{\sigma}' = (\bar{\sigma}^*,\ \bar{\phi},\ \bar{c})$, where $\bar{m} = m + 1$, $\mathsf{pk}_{S'}$

is from the attack hypotheses, and $\bar{c} = c$, $\bar{\phi} = \phi g$ (which corresponds to $\bar{r} = r$), and $\bar{\sigma}^* = \mathsf{Sign}(\mathsf{sk}_{S'},\ (\bar{\phi},\ \bar{c},\ \mathsf{pk}_{S'}))$. The transformed DCS has the same validity/invalidity as the pre-transformation DCS. Interacting with the confirmation oracle yields the validity/invalidity of the transformed DCS, and consequently the validity/invalidity of the original pre-transformation DCS.

Attack generalization. Replacing Equation (1) by $\phi = g^{H(m,\mathsf{pk}_S,\mathsf{pk}_C,c)}h^r$ is not a sufficient defense as we can achieve the same attack using

$$\bar{\phi} = \phi g^{H(\bar{m},\mathsf{pk}_S,\mathsf{pk}_C,c)-H(m,\mathsf{pk}_S,\mathsf{pk}_C,c)}$$

Other DCS schemes that use public-key encryption as a black-box building block, such as those in [12, 11] and elsewhere, may also risk signature transformation attacks. In fact, we demonstrate a signature transformation attack on [12] subsequently.

Attack mitigation: Change α above to

$$\alpha = Hash(u_1, u_2, u_3, \phi, \mathsf{pk}_S, \mathsf{pk}_C, m)$$

When queried with anything other than the (DCS, pk_S, m), the confirmation oracle will not yield any non-negligible advantage on the invisibility of the validity the DCS [5]. Using the repair above, we can explicitly upgrade the part of [11]'s security model to the corresponding part in [5] that defends signature transformation attacks.

Our results suggest that other schemes that use public-key encryption as a black-box building block, such as those in [12, 11] and elsewhere, should also use our easy mitigation technique: Open the black box slightly and add more parameters to the hash input or to the input of other kinds of *tag* generating mechanisms [1].

Discussions. Below, we also optimize [11]'s four-move concurrent zero-knowledge confirmation/disavowal protocols.

We omit the straightforward confirmation protocol $CZK\{r : \phi g^{-m} = h^r\}$. To disavow, prove either of the following:

$$\begin{gathered} CZK\{(x_1, x_2, y_1, y_2) : d_1 = g_1^{x_1} \ \wedge \ d_2 = g_1^{y_1} g_2^{y_2} \\ \wedge \ u_4 \neq g_1^{x_1+\alpha y_1} g_2^{x_1+\alpha y_2}\} \\ CZK\{(z, \bar{r}) : d_3 = g_1^{z} \ \wedge \ u_3 = u_1^{z} g_0^{\bar{r}} \ \wedge \ \phi g^{-m} \neq h^{\bar{r}}\} \end{gathered}$$

They are equivalent to, respectively,

$$CZK\{(x_1, x_2, y_1, y_2, s_0, s_1 = s_0x_1, s_2 = s_0y_1, s_3 = s_0x_2, s_4 = s_0y_2) :$$
$$d_1 = g_1^{x_1}g_2^{x_2} \wedge d_2 = g_1^{y_1}g_2^{y_2} \wedge T = u_4^{-s_0}g_1^{s_1+\alpha s_2}g_2^{s_3+\alpha s_4}$$
$$\wedge\ 1 = d_1^{s_0}g_1^{-s_1}g_2^{-s_3} \wedge 1 = d_2^{s_0}g_1^{-s_2}g_2^{-s_4}\} \text{ with } T \neq 1$$

$$CZK\{(z, \bar{r}, s_0, s_1 = s_0\bar{r}) : d_3 = g_1^z \wedge u_3 = u_1^z g_0^{\bar{r}} \wedge T = (\phi^{-1}g^m)^{s_0}g^{s_1}$$
$$\wedge\ T_4 = g_4^{s_0} \wedge 1 = T_4^{\bar{r}}g_4^{-s_1}\} \text{ with } T \neq 1$$

The confirmation costs 4 moves totalling 3 exponentiations. The disavow costs 4 moves totally at most 32 exponentiations. In comparison, [11]'s confirmation (resp. disavowal) costs 4 moves and 10 exponentiations (resp. 16 moves and 41 exponentiations). We demonstrate the second CZK:

1. Verfier select random c', sends $c'' = Hash(c')$.
2. Prover sends T, T_4, and $D_3 = g_1^{r_z}$, $D_u = u_1^{r_z}g_0^{r_r}$, $D_T = (\phi^{-1}g^m)^{r_0}g^{r_1}$, $D_4 = g_4^{r_0}$, $D_5 = T_4^{r_r}g_4^{-r_1}$.
3. Verifier checks $T \neq 1$ and sends c'.
4. Prover checks $c'' = Hash(c')$, sends $z_z = r_z - c'z$, $z_r = r_r - c'\bar{r}$, $z_0 = r_0 - c's_0$, $z_1 = r_1 - c's_1$.

Finally, Verifier checks the following before outputting 1: $D_3 = g_1^{z_z}d_3^{c'}$, $D_u = u_1^{z_z}g_0^{z_r}u_3^{c'}$, $D_T = (\phi^{-1}g^m)^{z_0}g^{z_1}T^{c'}$, $D_4 = g_4^{z_0}T_4^{c'}$, $D_5 = T_4^{z_r}g_4^{-z_1}$.

Signature transformation attack and Goldwasser, et al. [12]'s DCS. *Review.* We focus on the first concrete DCS in [12] which is based on the Cramer-Shoup signature [10] and the Cramer-Shoup encryption [9]. The Cramer-Shoup signature on message m is $\sigma' = (e, y', y)$,

$$y^e = xh^{H(x')}$$
$$x' = (y')^{e'}h^{-H(m)}$$

where the signer's public key is $\mathsf{pk}_S = (n,\ h,\ x,\ e')$, n is a product of two primes, e' and e' are distinct primes, h and x are random. The Goldwasser, et al.'s DCS is $\sigma = (\sigma_1 = e,\ \sigma_2 = y',\ \sigma_3 = \mathsf{Enc}(\mathsf{pk}_C,\ y)\ )$.

Attack Hypotheses. The attacker needs the following hypotheses:

1. Knowing the private key sk_S of the signer.
2. Query access to a confirmation oracle which, upon common inputs including a message $\bar{m}$, a signer public key $\mathsf{pk}_{S'}$, and a putative DCS $\bar{\sigma}$, will confirm or disavow the DCS $\bar{\sigma}$. Except when the queried tuple $(\bar{m},\ \mathsf{pk}_{S'},\ \bar{\sigma})$, $\bar{\sigma} = (\bar{e},\ \bar{y}',\ \bar{\sigma}_3)$, shares the same m, or the same pk_S, or the same σ_1, or the same σ_2 as the attacker's target tuple. Queries with the same σ_3 are allowed.

3. The signature verification protocol does not check e is a prime. It only checks that it is within a certain range. This is a common practice in using the Cramer-Shoup encryption, e.g. [10, 12], to keep computational complexities low.

Attack consequence and procedure. The attacker can compute the validity/invalidity of a given putative DCS by interacting once with the confirmation oracle with the following transformed putative DCS: $\bar{\sigma} = (\bar{e}, \bar{y}', \sigma_3)$ on a new arbitrary message $\bar{m}$ for a new signer public key $\bar{\mathsf{pk}}_S = (n, \bar{h}, \bar{x}, e')$ where $\bar{x} = y^e$, $\bar{h} = y$, $\bar{x}' = (y')^{e'}\bar{h}^{-H(\bar{m})}$, $\bar{e} = e + H(\bar{x}')$. It is mechanical to verify that the transformed DCS has the same validity/invalidity as the pre-transformation DCS. Interacting with the confirmation oracle yields the validity/invalidity of the transformed DCS, and consequently the validity/invalidity of the original pre-transformation DCS.

Therefore, an adversary $\mathcal{A}$ can distinguish a valid signature from an invalid one by interacting with the confirmation oracle. However, [12] does not claim the indistinguishability between valid and invalid signatures, called the *invisibility of the signature* in [15, 5]. Our attack is beyond their security model. Their DCS remains secure in their own model.

Mitigation. Nevertheless, we suggest to include more parameters in the has inputs wherever possible to defend against signature transformation and potentially other attacks. For example, letting $x' = (y')^{e'} h^{-H(m, \mathsf{pk}_S, e, y')}$ or having even more parameters included in the hash inputs can contribute to enhanced security.

Upgrading [14]'s undeniable signature to achieve signature invisibility. Undeniable signatures [8] are DCS's where signer and confirmer are the same entity. Using techniques developed above, we can modify Laguillaumie, et al. [14]'s undeniable signature without random oracles to upgraded security model with signature invisibility and defense against signature transformation attackers. Consult original references for details of the security model.

1. *Setup.* The signer public key $\mathsf{pk} = (n, y_1, y_2, {}_1, d_2, d_3)$, $\mathsf{sk} = (x_1, x_2, \bar{x}_1, \bar{x}_2, \bar{y}_1, \bar{y}_2, z)$, where $y_1 = g^{x_1}$, $y_2 = g^{x_2}$, $d_1 = g_1^{\bar{x}_1} g_2^{\bar{x}_2}$, $d_2 = g_1^{\bar{y}_1} g_2^{\bar{y}_2}$, $d_3 = g_1^z$, n is a product of two safe primes p and q, pairings $\hat{\mathbf{e}} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$, order$(\mathbb{G}_1) = n$, $g \in \mathbb{G}_1$, $g_1, g_2 \in \mathbb{Z}_{n^2}$, $g_0 = n + 1$.
2. *Sign.* Select random $R \in \mathbb{Z}_n$, compute $\sigma = (\sigma_1, \sigma_2)$, where $\sigma_1 = g^{1/(x_1 + R + mx_2)}$, $\sigma_2 = \mathsf{Enc}(R) = (u_1, \cdots, u_4)$ with $u_1 = g_1^r$, $u_2 = g_2^r$, $u_3 = d_3^r g_0^R$, $u_4 = (d_1 d_2^\alpha)^r$, where $\alpha = Hash(u_1, u_2, u_3, m, \mathsf{pk}, \sigma_1)$.

3. *Confirm/disavow.* To confirm, prove the following concurrent zero-knwoledge protocols:

$$CZK\{R : \hat{\mathbf{e}}(\sigma_1, y_1 y_2^m)\hat{\mathbf{e}}(\sigma_1, g)^R = \hat{\mathbf{e}}(g, g) \ \wedge \ u_3 = u_1^z g_0^R \ \wedge \ d_3 = g_1^z\}$$

To disavow, prove the following concurrent zero-knwoledge protocol

$$\begin{aligned} &CZK\{(\bar{x}_1, \bar{x}_2, \bar{y}_1, \bar{y}_2, z, R') : d_1 = g_1^{\bar{x}_1} g_2^{\bar{x}_2} \\ &\quad \wedge \ d_2 = g_1^{\bar{y}_1} g_2^{\bar{y}_2} \ \wedge \ d_3 = g_1^z \ \wedge \ u_3 = u_1^z g_0^{R'} \\ &\quad \wedge \ [u_4 \neq u_1^{\bar{x}_1 + \alpha\bar{y}_1} u_2^{\bar{x}_2 + \alpha\bar{y}_2} \ \vee \ \hat{\mathbf{e}}(\sigma_1, y_1 y_2^m)\hat{\mathbf{e}}(\sigma_1, g)^{R'} \neq \hat{\mathbf{e}}(g, g)\} \end{aligned}$$

Note order$(g_0) = n$ in $\mathbb{Z}_{n^2}$. There is no need to prove for the *proof of range* that R (and R') lie in the interval $[0, n)$. Th invisibility of signature mainly follows the use of concurrent zero-knowledge protocols. The unforgeability of the undeniable signature can be proved similarly to [14]. Methods to instantiate a pairings group (or *gap Diffie-Hellman group*) $\mathbb{G}_1$ with a composite order n were described in Boneh, et al. [4] and Groth, et al. [13].

Generalization. The undeniable signature above combines Boneh, et al. [3]'s signature without random oracles and the famous Cramer-Shoup encryption [9] without random oracles. It can be modified into a DCS by separating the signing key (given to the signer) and the encryption key (given to the confirmer). But then the confirmer key, $\mathsf{pk}_C = (d_1,\, d_2,\, d_3)$, is dependent of the signer public key n, as the three entries lie in $\mathbb{Z}_{n^2}$. Although security is not compromised because the security of the Cramer-Shoup encryption reduces to the decisional Diffie-Hellman assumption in $\mathbb{Z}_{n^2}$ which continues to hold, this dependence is not desirable. If entries of pk_C are in $\mathbb{Z}_{\bar{n}^2}$ with $\bar{n} \neq n$, then inefficient range proofs may have to be used in the confirmation/disavowal protocol.

Acknowledgements to Hong Kong Earmarked Grants 4232-03E and 4328-02E for sponsorship.

References

1. Masayuki Abe, Rosario Gennaro, Kaoru Kurosawa, and Victor Shoup. Tag-KEM/DEM: A new framework for hybrid encryption and a new analysis of Kurosawa-Desmedt KEM. In *EUROCRYPT 2005*, pages 128–146, 2005.
2. N. Asokan, Victor Shoup, and Michael Waidner. Optimistic fair exchange of digital signatures. In *EUROCRYPT 1998*, pages 591–606, 1998.
3. D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In *Proc. CRYPTO 2004*, pages 41–55. Springer-Verlag, 2004. Lecture Notes in Computer Science No. 3152.

4. Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. Evaluating 2-DNF formulas on ciphertexts. In *TCC 2005*, pages 325–341, 2005.
5. J. Camenisch and M. Michels. Confirmer signature schemes secure against adaptive adversaries. In *Eurocrypt 2000*, pages 243–258. Springer-Verlag, 2000. LNCS No. 2729.
6. J. Camenisch and V. Shoup. Practical verifiable encryption and decryption of discrete logarithms. In *CRYPTO'03*, volume 2729 of *LNCS*, pages 126–144. Springer-Verlag, 2003.
7. D. Chaum. Designated confirmer signatures. In *Eurocrypt'94*, pages 86–91. Springer-Verlag, 1994. LNCS No. 435.
8. D. Chaum and H. van Antwerpen. Undeniable signatures. In *Crypto'89*, pages 286–299, 1989.
9. R. Cramer and V. Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 45–64. Springer-Verlag, 2002.
10. Ronald Cramer and Victor Shoup. Signature schemes based on the strong rsa assumption. *ACM Trans. Inf. Syst. Secur.*, 3(3):161–185, 2000.
11. Craig Gentry, David Molnar, and Zulfikar Ramzan. Efficient designated confirmer signatures without random oracles or general zero-knowledge proofs. In *ASIACRYPT 2005*, volume 3788 of *LNCS*, pages 662–681. Springer-Verlag, 2005.
12. Shafi Goldwasser and Erez Waisbard. Transformation of digital signature schemes into designated confirmer signature schemes. In *TCC 2004*, volume 2951 of *LNCS*, pages 77–100. Springer-Verlag, 2004.
13. Jens Groth, Rafail Ostrovsky, and Amit Sahai. Perfect non-interactive zero knowledge for NP. Cryptology ePrint Archive, Report 2005/290.
14. Fabien Laguillaumie and Damien Vergnaud. Short undeniable signatures without random oracles: The missing link. In *INDOCRYPT 2005*, volume 3797 of *LNCS*, pages 283–296. Springer-Verlag, 2005.
15. T. Okamoto. Designated confirmer signatures and public-key encryption are equivalen. In *Proc. CRYPTO '94*, pages 61–74, 1994.