Signature Transformation Attacks on Designated Confirmer Signatures

Victor K. Wei

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

October 2, 2006

Abstract. Camenisch and Michels in Eurocrypt 2000 introduced the signature trnaformation 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.

1 The results

Chaum [3] 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 [3, 1].

T. Okamoto [8] gave a formal security model for DCS, and a polynomial equivalence reduction between DCS and public-key encryption. Camenisch and Michels [2] 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. [2] also gave concrete instantiations, using the RSA signature and the Cramer-Shoup encryption. The confirmation and disavowalwere not very efficient as they involved double discrete logarithms or range proofs.

Goldwasser and Waisbard [7] and Gentry, et al. [6] presented DCS without random oracles. [6]'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 [2]'s signature transformation attack on the two DCS's above, before repairing them. The attack on [6] is within their model while the attack on [7] is beyond their model. We also optimize the efficiencies of [6]'s confirmation and disavowal protocols.

In this brief note, we do not include the security model or other definitions of terminologies. Consult the original references for details [2, 7, 6].

Attack and repair on Gentry, et al. [6]. In summary, the main DCS in [6] 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 $\mathsf{sk}_C = (x_1,\ x_2,\ y_1,\ y_2,\ z)$.

The *signature transformation attack*: Generate the transformed signature $\bar{\sigma}' = (\bar{\sigma}^*,\ \bar{\phi},\ \bar{c})$ on message $\bar{m} = m + 1$ by computing $\bar{c} = c$, $\bar{r} = r$, $\bar{\phi} = \phi g$, and computing $\bar{\sigma}^*$ using attacker's knowledge of sk_S which is granted in the security model. 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 signture from an invalid one by interacting with the confirmation oracle, and thus breaking the security of the DCS. Note that replacing Equation (1) by $\phi = g^{H(m)} h^r$ is not a sufficient defense as we can use $\bar{\phi} = \phi g^{H(\bar{m}) - H(m)}$ and achieve the same attack.

Repair: 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 [2].

Using the repair above, we can explicitly upgrade the part of [6]'s security model to the corresponding part in [2] that defends signature transformation attacks. Below, we also optimize [6]'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,

$$\begin{gathered} CZK\{(x_1, x_2, y_1, y_2, s_0, s_1 = s_0 x_1, s_2 = s_0 y_1, s_3 = s_0 x_2, s_4 = s_0 y_2) : \\ d_1 = g_1^{x_1} g_2^{x_2} \ \wedge\ d_2 = g_1^{y_1} g_2^{y_2} \ \wedge\ T = u_4^{-s_0} g_1^{s_1+\alpha s_2} g_2^{s_3+\alpha s_4} \\ \wedge\ 1 = d_1^{s_0} g_1^{-s_1} g_2^{-s_3} \ \wedge\ 1 = d_2^{s_0} g_1^{-s_2} g_2^{-s_4}\} \text{ with } T \neq 1 \\ CZK\{(z, \bar{r}, s_0, s_1 = s_0 \bar{r}) : d_3 = g_1^z \ \wedge\ u_3 = u_1^z g_0^{\bar{r}} \ \wedge\ T = (\phi^{-1} g^m)^{s_0} g^{s_1} \\ \wedge\ T_4 = g_4^{s_0} \ \wedge\ 1 = T_4^{\bar{r}} g_4^{-s_1}\} \text{ with } T \neq 1 \end{gathered}$$

The confirmation costs 4 moves totalling 3 exponentiations. The disavow costs 4 moves totally at most 32 exponentiations. In comparison, [6]'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: $D_3 = g_1^{z_z} d_3^{c'}$, $D_u = u_1^{z_z} g_0^{z_r} u_3^{c'}$, $D_T = (\phi^{-1} g^m)^{z_0} g^{z_1} T^{c'}$, $D_4 = g_4^{z_0} T_4^{c'}$, $D_5 = T_4^{z_r} g_4^{-z_1}$.

Attack generalization and repair: Other DCS schemes that use encryption as a black-box building block, such as those in [7, 6] and elsewhere, may also risk signature transformation attacks. Our results suggest these schemes may have an easy upgrade path by opening the black box slightly and add more parameters to the hash input or to the input of other kinds of *tag* generating mechanisms. We demonstrate a similar signature transformation attack on the DCS in [7] below.

Attack and repair on Goldwasser, et al. [7]. We focus on the first DCS in [7] which is based on the Cramer-Shoup signature [5] and the Cramer-Shoup encryption [4]. The Cramer-Shoup signature on message m is $\sigma' = (e, y', y)$,

$$\begin{aligned} y^e &= x h^{H(x')} \\ x' &= (y')^{e'} h^{-H(m), \mathsf{pk}_S, e, y')} \end{aligned}$$

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 $\sigma = (\sigma_1 = e,\, \sigma_2 = y',\, \sigma_3 = \mathsf{Enc}(\mathsf{pk}_C,\, y)\ )$.

The *signature transformation attack*: Generate the transformed signature $\bar{\sigma} = (\bar{e},\, \bar{y}',\, \sigma_3)$ on a new 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. Note in verifying the signature, it is a common practice to not check e (resp. $\bar{e}$) is a prime, only to check that it is within a certain range [5, 7]. We use this practice in our attack hyphtheses.

Therefore, an adversary $\mathcal{A}$ can distinguish a valid signture from an invalid one by interacting with the confirmation oracle. However, [7] does not claim the indistinguishability between valid and invalid signatures. Our attack is beyond their security model. Their DCS remains secure in their own model. Nevertheless, we suggest to include more parameters in the has inputs wherever possible to defend against signature transformation and potentially other attacks. For example, $x' = (y')^{e'} h^{-H(m),\mathsf{pk}_S,e,y')}$ or to have even more parameters included in the hash inputs.

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

References

1. N. Asokan, Victor Shoup, and Michael Waidner. Optimistic fair exchange of digital signatures. In *EUROCRYPT 1998*, pages 591–606, 1998.
2. J. Camenisch and M. Michels. Confirmer signature schemes secure against adaptive adversaries. In *Eurocrypt 2000*, pages 243–258. Springer-Verlag, 2000. LNCS No. 2729.
3. D. Chaum. Designated confirmer signatures. In *Eurocrypt'94*, pages 86–91. Springer-Verlag, 1994. LNCS No. 435.
4. 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.
5. Ronald Cramer and Victor Shoup. Signature schemes based on the strong rsa assumption. *ACM Trans. Inf. Syst. Secur.*, 3(3):161–185, 2000.
6. 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.
7. 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.

8. T. Okamoto. Designated confirmer signatures and public-key encryption are equivalen. In *Proc. CRYPTO '94*, pages 61–74, 1994.