Cryptanalysis of a Forward Secure Blind Signature Scheme with Provable Security

Shuhong Wang[1], Feng Bao[2], and Robert H. Deng[1]

[1] School of Information Systems, SMU, Singapore 178902
{shwang, robertdeng}@smu.edu.sg
[2] Institute for Infocomm Research (I2R), Singapore 119613
baofeng@i2r.a-star.edu.sg

Abstract. A forward secure blind signature scheme was proposed by Duc, Cheon and Kim, in ICICS 2003. The security of the scheme was proved to be equivalent to the strong RSA assumption in the random oracle model. In this paper we present an attack to the scheme by forging valid signatures with public keys only. The attack is so efficient that forging a valid signature needs less computation than legally generating a signature, even considering only the user side. Our result implies that the security proof of the scheme must be invalid. Furthermore we point out the fault of the proof and explain why it invalidates the proof.

Keywords: Blind signature, Forward security, Provable security, Strong RSA assumption, Cryptanalysis.

1 Introduction

Due to some unpredictable security faults of underlying system or errors of implementation, key exposure is high likely unavoidable. To mitigate the danger caused by key exposure, the notion of *forward security*[1] was introduced [1] in the context of signature schemes. It is always obtained by employing the so called key evolution strategy, which is economical/practical compared with distribution of keys across multiple systems via secret sharing like threshold methodologies [9,10]. Informally, key evolution means that different secret keys are used in different periods of time, and the key for next time period is updated (through an Update protocol) from the one in previous time period, meanwhile the public key is kept unchanged over its lifetime. There have been many signature schemes with forward security [4,2,13].

However, as claimed in [8], there is no instance of Update supporting unlimited periods[2] key evolution until the proposal of Duc, Cheon and Kim [8] in a

[1] In the context of session key exchange, it was first introduced in [11] known as forward secrecy, meaning that compromise of the current session key should not compromise past established session keys.

[2] In general, the Update protocols only support T times of key evolutions for some predefined integer T.

S. Qing et al. (Eds.): ICICS 2005, LNCS 3783, pp. 53–60, 2005.

blind signature context. The scheme is called forward secure blind signature, and hereinafter, we denote it the DCK scheme for short.

Blind signature, as an extension of digital signature, allows the user to obtain the signer's signature on a message of his choice in a blind way such that the message content is not revealed to the signer. Such a scheme was first proposed by Chaum [6] for the purpose of digital payments where the user is a consumer and the signer is a bank. Along with the rapid development of sensitive e-commerce [16,3,18], blind signature is becoming very concernful. It is unassailable that forward security will provide really useful features for a blind signature scheme.

Desirably, a *forward secure blind signature* [8], especially when used for electronic payment, should at least have following two security features.

- **Blindness.** Besides "obtaining signature without revealing message", the blindness property also implies that the signer cannot statistically distinguish signatures, which is like that the bank cannot trace its client's buying activities. For the formal definition, please refer to [8].
- **Forward security.** In the context of blind signature, forward security implies the basic *unforgeability* as of ordinary signatures. In addition, it implies the unforgeability of signature to be valid in previous time periods even if the current secret key of the signer is compromised.

In this paper, we address the security analysis of the attractive DCK scheme [8] mentioned above. Although the security of the scheme was proved to be equivalent to the strong RSA assumption in the random oracle model [5], we are still able to forge valid signatures at will with public keys only. Note that our attack is so efficient that forging a valid signature needs less computation than legally generating a signature, even when merely considering the user side. Our result implies that the security proof of the scheme must be invalid. Furthermore we point out the fault of the security proof and explain why it invalidates the proof.

The remainder of the paper is arranged as follows. We first briefly review the original DCK scheme in section 2, and then describe our signature forgery attack in section 3. The analysis of their security proof is given in section 4, followed by conclusion in section 5.

2 Description of the DCK Scheme

The DCK scheme is simply described as follows.

System Setup. k is the security parameter. $N = pq$ are product of two random safe primes p and q of $k/2$ bits length. λ is a large prime without nontrivial common divisor with $\varphi(N)$, where φ is the Euler function in number theory. An element $a \in Z_N^*$ is selected to be of order greater than λ. Let $r_0 \in_R Z_\lambda^*$ and $s_0 \in_R Z_N^*$ (with notation $x \in_R X$, we mean x is randomly chosen from X), and compute $V = a^{-r_0} s_0^{-\lambda} \bmod N$.

Finally, the signer's initial secret key SK_0 is $(0, r_0, s_0, v_0 = V, f_0 = 1)$, the public key $PK = (N, \lambda, a, V)$. All other parameters are erased. Also, a collision-free hash function $H : \{0,1\}^* \rightarrow Z_\lambda^*$ is assumed and made public.

Secret key Update. Hereafter, we definite $a \div b$ the integer quotient of $\frac{a-(a \bmod b)}{b}$. After each execution of the Update protocol, all parameters except the new secret key are erased from the memory. See Fig. 1.

Input SK_i	$e \in_R Z_N^*$	Output SK_{i+1}
	$f_{i+1} = f_i^2 a^e \bmod N$	
$(i,\ r_i,\ s_i,\ f_i) \Rightarrow$	$l = (2r_i - e) \div \lambda$	$\Rightarrow (i+1, r_{i+1}, s_{i+1}, f_{i+1})$
	$r_{i+1} = (2r_i - e) \bmod \lambda$	
	$s_{i+1} = a^l s_i^2 \bmod N$	

Fig. 1. The key Update protocol

Signature Issuing. This procedure involves two entities: signer and user. The protocol is illustrated in Fig. 2, where $||$ denotes the string concatenation. Note that (i, f_i) are available to the user when contacting with the signer, but they are unavailable to verifier. Otherwise, it contradicts the original intention for offline electronic system, where payments are made available without online communication with the signer (bank) [8].

Signature Verification. Given the signature $(m, i, \sigma(m)) = (m, i, f, c', y', z')$ and $PK = (N, \lambda, a, V)$, the verifier first computes $v_i = V^{2^i} f \bmod N$ and $x'' = a^{y'} z'^{\lambda} v_i^{c'}$, then checks whether or not $c' \stackrel{?}{=} H(i||f||m||x'')$. If it is right, then accept; Otherwise reject.

3 The Signature Forgery Attack

In this section we describe our forgery attack on the DCK scheme. The attack is so strong that anyone can forge signatures on any message valid in any time period. The only needed information is the public key $PK = (N, \lambda, a, V)$. The attack is also very efficient even compared with the computation on the user side only. The attack behaves as the following.

1. Obtain the public key $PK = (N, \lambda, a, V)$ and a valid time period i.
2. Choose $\alpha, \gamma \in_R Z_\lambda^*$ and $\beta, v \in_R Z_N^*$.
3. Compute $f = V^{-2^i} v^\lambda \bmod N$.
4. Compute $x = a^\alpha \beta^\lambda v^{\lambda\gamma} \bmod N$.
5. Compute $c = H(i||f||m||x)$, where m is any message of the attacker's choice.
6. Compute $z = v^{\gamma - c} \beta \bmod N$ and set $y = \alpha$.
7. Output the 6-tuple (m, i, f, c, y, z) as the forged signature on m, intending for the i-th time period.

$$\begin{array}{rcl}
\textit{Signer } (SK_i) & & \textit{User } (PK, m) \\
 & & \text{Get } (i, f_i) \text{ from signer} \\
t \in_R Z_\lambda^* \text{ and } u \in_R Z_N^* & & \\
x = a^t u^\lambda \bmod N & & \\
 & \xrightarrow{\qquad x \qquad} & \\
 & & \alpha, \gamma \in_R Z_\lambda^*,\ \ \beta \in_R Z_N^* \\
 & & v_i = V^{2^i} f_i \bmod N \\
 & & x' = x a^\alpha \beta^\lambda v_i^\gamma \bmod N \\
 & & c' = H(i||f_i||m||x') \\
 & & c = (c' - \gamma) \bmod \lambda \\
 & \xleftarrow{\qquad c \qquad} & \\
y = (t + cr_i) \bmod \lambda & & \\
w = (t + cr_i) \div \lambda & & \\
z = a^\omega u s_i^c \bmod N & & \\
 & \xrightarrow{\qquad y,\ z \qquad} & \\
 & & y' = (y + \alpha) \bmod \lambda \\
 & & \omega' = (y + \alpha) \div \lambda \\
 & & \omega'' = (c - c') \div \lambda \\
 & & z' = a^{\omega'} v_i^{-\omega''} z\beta \bmod N \\
 & & \sigma(m) = (f_i, c', y', z')
\end{array}$$

Fig. 2. The signature Issuing protocol. As noted in [8], the index i of f_i is omitted in $\sigma(m)$ on consideration that attackers do not have to use the correct f for a period. Thus the signature is denoted as $(m, i, \sigma(m)) = (m, i, f, c', y', z')$.

Theorem 1. (Correctness of the Attack) Suppose an attacker follows above seven steps and obtains the 6-tuple (m, i, f, c, y, z). Then, the verifier always accepts (m, i, f, c, y, z) as a valid signature in period i.

Proof. To prove the theorem, we simply simulate what the verifier does with the signature (m, i, f, c, y, z). He/She first retrieves the public key $PK = (N, \lambda, a, V)$ and computes $v_i = V^{2^i} f = V^{2^i}(V^{-2^i} v^\lambda) = v^\lambda \bmod N$. Then he/she computes $x'' = a^y z^\lambda v_i^c \bmod N$. Because we have

$$\begin{aligned}
x'' &= a^y z^\lambda v_i^c \bmod N \\
&= a^y (v^{\gamma - c}\beta)^\lambda v_i^c \bmod N \\
&= a^\alpha v^{(\gamma - c)\lambda} \beta^\lambda (v^\lambda)^c \bmod N \\
&= a^\alpha \beta^\lambda v^{(\gamma - c)\lambda + c\lambda} \bmod N \\
&= a^\alpha \beta^\lambda v^{\gamma\lambda} \bmod N = x,
\end{aligned}$$

it is always the case that $c = H(i||f||m||x) = H(i||f||m||x'')$. As a result, the verifier accepts (m, i, f, c, y, z) as a valid signature in period i. ■

Remarks on Efficiency. Roughly speaking, our forgery attack only expends 6 T_E (modulo exponentials) and 4 T_M (modulo multiplications), while there are 5

T_E and 6 T_M on only the user side for legally obtaining a signature. Both have a modulo reciprocal and therefore are eliminated. Note that we do not count in the computation of $v_i = V^{s^i} f_i \bmod N$ which is assumed to be available for legal users from the signer. In fact, the attack can also contact with the signer and get the V^{-2^i} by computing $f_i v_i^{-1} \bmod N$, thus 1 T_M and 1 reciprocal replace 1 T_E and 1 reciprocal. Accordingly, the forgery gains 1 T_M efficiency compared to honestly obtaining a signature by the user (5 T_E and 5 T_M for forgery to 5 T_E and 6 T_M for generation).

4 The Failure of Security Proof

The DCK scheme [8] is constructed from the provably secure Okamoto-Guilou-Quisquater (OGQ for short) blind signature scheme [15,12]. Using the same methodology (oracle replay) for proving OGQ scheme due to Pointcheval and Stern [16], authors of [8] proved the security of DCK scheme under the strong RSA assumption.

Strong RSA Assumption. The strong RSA assumption is described as follows: Given a RSA modulus N (which is a product of two large primes) and a random element $c \in Z_N^*$, it is intractable to find two elements $m, r \in Z_N^*$ such that $m^r = c \bmod N$. It is a well-known assumption in cryptography and has been extensively used for security proofs.

4.1 Sketch of the Security Proof

There are two theorems regarding the security of the DCK scheme, one for the blindness property (Theorem 2 of [8]) and one for the forward security (Theorem 3 of [8]). The later is outlined as follows.

Theorem 3 of [8]. *If there exists a forger who can break forward security of our scheme. Then, with non-negligible probability, we can violate the strong RSA assumption.*

Proof outline. Assume a forger $\mathcal{F}$ who obtains PK and SK_i of time period i can output a signature $\sigma(m)$ valid at some time period j for $j < i$. Also assume $\mathcal{F}$ should query the hashing oracle on $(j||f||m||x')$ before its output. Upon the answer of the oracle say H_1, $\mathcal{F}$ successfully forge a signature $(j, \sigma_1(m)) = (j, f, x'_1, y'_1, z'_1)$. Then, by replaying another oracle H_2 which has the same answer to oracle H_1 until the query of $(j||f||m||x')$. With non-negligible probability, $\mathcal{F}$ will again output a forged signature $(j, \sigma_2(m)) = (j, f, x'_2, y'_2, z'_2)$ based on oracle H_2, this is assured by the well-known forking lemma [16]. Since the two forged signatures have the same verifying equation, it must be the case that $a^{y'_1} z_1'^{\lambda} (V^{2^j} f)^{c'_1} = a^{y'_2} z_2'^{\lambda} (V^{2^j} f)^{c'_2}$. By assuming $V^{2^j} f$ equals to v_j and therefore has the form of $a^{-r_j} s_j^{-\lambda} \bmod N$, the authors of [8] claimed being able to come up with an equation of the form $a^{\rho} = b^{\lambda} \bmod N$, and thus the strong RSA problem is solvable with a high probability, if only $gcd(\rho, \lambda) = 1$ (see Lemma 1 of [8]). Note that it is high likely $gcd(\rho, \lambda) = 1$ with λ being prime.

4.2 Fault of the Proof

As above mentioned, our result implies that the security proof of the scheme must be invalid. Although there exists negative examples [7] such that schemes provably secure in random oracle model [5] may result in insecure ones when the oracle is implemented by cryptographic hash functions, our attack has nothing to do with the hash function. In fact, it is on the basis of the problem in the scheme construction itself.

Keeping our attack in mind to check through the security proof, it is not hard to find out its fault. For a forged signature, the expectation [8] that $V^{2^j} f \bmod N$ would equal to the correct $v_j = a^{-r_j} s_j^{-\lambda} \bmod N$ as in the Update protocol is unreliable. In the proposed attack, we have $V^{2^j} f = v^{\gamma} \bmod N$ with $\gamma \in_R Z_{\lambda}^*$ and $v \in_R Z_N^*$, clearly it is not in the form of $a^r s^{\lambda} \bmod N$ for some $r \in Z_{\lambda}^*$ and $s \in Z_N^*$.

In the following, we show how critical the fault is. To equalize the security of DCK scheme and the strong RSA assumption, it is sufficient to get an equation like $a^{\rho} = b^{\lambda} \bmod N$. Let us take an observation on the equation obtained by oracle replay: $a^{y_1'} z_1'^{\lambda} (V^{2^j} f)^{c_1'} = a^{y_2'} z_2'^{\lambda} (V^{2^j} f)^{c_2'}$, which can be transferred to $a^{y_1'-y_2'} = (z_2'/z_1')^{\lambda} \cdot (V^{2^j} f)^{c_2'-c_1'} \bmod N$. Obviously, one can get equation $a^{\rho} = b^{\lambda} \bmod N$ with some $\rho \in Z$ and $b \in Z_N^*$ if and only if $(V^{2^j} f)^{c_2'-c_1'} \bmod N$ can be expressed as $a^r s^{\lambda} \bmod N$. However, unless with negligible probability $c_2' = c_1' \bmod \lambda$ (then, $(V^{2^j} f)^{c_2'-c_1'} = [(V^{2^j} f)^{(c_2'-c_1') \div \lambda}]^{\lambda} \bmod N$), being able to express an random elements in Z_N^* as the form of $a^r s^{\lambda} \bmod N$ means that one can easily break the OGQ scheme [15] by just using $(\lambda - r, as)$ as an OGQ signing key pair ($a^r s^{\lambda} = a^{-(\lambda-r)}(as)^{\lambda} \bmod N$). This result contradicts the proof of Pointcheval and Stern [16]. And if the proof in [16] is correct, expressing an random element in Z_N^* as $a^r s^{\lambda}$, itself is at least as hard as the strong RSA problem. In other words, the authors in [8] implicitly assumed the solvability of the strong RSA problem. Assume a problem has already been solved and then turn back to solve that problem, which is logically incorrect.

Remarks. However, the proof of [8] still implies the security of the Update protocol, i.e., it is impossible for an attacker to forge a signature by forging the secret key in advance. Since in this case, the equation $V^{2^j} f = v_j = a^{-r_j} s_j^{-\lambda} \bmod N$ holds.

5 Conclusion

In this paper, we successfully illustrated the insecurity of a forward-secure blind signature scheme which is proved to be equivalent to the strong RSA assumption. The attack is strong and very efficient. Anyone can forge signatures on any message valid in any time period using the unchanged public keys only. Furthermore, we also pointed out the fault of the security proof and explained why it invalidates the proof. Our work implies that regardless of the failure caused by oracle implementations, the security proof itself still needs time to validate its correctness.

References

1. Ross Anderson, Two Remarks on Public Key Cryptography, Invited Lecture, in *Fourth Annual Conference on Computer and Communications Security*, ACM, 1997.
2. Michel Abdalla and Leonid Reyzin. A new forward-secure digital signature scheme. In *Advances in Cryptology - ASIACRYPT 2000*, Springer-Verlag, 2000.
3. Feng Bao, Robert H. Deng, and Wenbo Mao, Efficient and practical fair exchange protocols with off-line TTP, in *IEEE Symposium on Security and Privacy*, IEEE Computer Society Press, pp. 77 - 85, 1998.
4. Mihir Bellare and Sara K. Miner, A Forward-Secure Digital Signature Scheme, in *Advances in Cryptology - CRYPTO '99*, LNCS 1666, Springer-Verlag, pp. 431 - 448, 1999.
5. Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM CCS '93*, pages 62 - 73, November 1993.
6. David Chaum, Blind Signatures For Untraceable Payments, in *Advances in Cryptology - CRYPTO '82*, Plenum Publishing, pp. 199 - 204, 1982.
7. Ran Canetti, Oded Goldreich, and Shai Halevi, The Random Oracle Methodology, Revisited (Extend abstract), in *Proc. of the 30th ACM Symp. on Theory of Computing - STOC'98*, pages 209-218, 1998.
8. Dang N. Duc, Jung H. Cheon, and Kwangjo Kim, A Forward-Secure Blind Signature Scheme Based on the Strong RSA Assumption, in *Proceedings of the 5-th International Conference on Information and Communications Security - ICICS '03*, LNCS 2836, Springer-Verlag, pp. 11 - 21, 2003.
9. Yvo G. Desmedt and Yair Frankel, Threshold cryptosystems, in *Advances in Cryptology - Crypto '89*, LNCS 435, Springer-Verlag, pp. 307 - 315, 1989.
10. Y. Desmedt, Y. Frankel and M. Yung, Multi-receiver/Multi-sender network security: efficient authenticated multicast/feedback, *Proceedings of IEEE Infocom '92*, pp. 2045 - 2054, 1992.
11. C. Günther, An Identity-based Key-exchange Protocol, in *Proceedings of Eurocrypt '89*, LNCS 434, Springer-Verlag, 1989.
12. Louis S. Guillou and Jean J. Quisquater, A Practical Zero-Knowledge Protocol Fitted to Security Microprocessors Minimizing both Transmission and Memory, in *Advances in Cryptology - EUROCRYPT '88*, LNCS 330, Springer-Verlag, pp. 123 - 128, 1988.
13. Gene Itkis and Leonid Reyzin, Forward-Secure Signatures with Optimal Signing and Verifying, in *Advances in Cryptology - CRYPTO '01*, LNCS 2139, Springer-Verlag, pp. 332 - 354, 2001.
14. Wenbo Mao and Colin Boyd, Towards Formal Analysis of Security Protocols, In *Proceedings of the 4-th Computer Security Foundations Workshop*, Franconia, New-Hampshire, June 1993.
15. Tatsuki Okamoto, Provably Secure and Practical Identification Schemes and Corresponding Signature Schemes, in *Advances in Cryptology - CRYPTO '92*, LNCS 740, Springer-Verlag, pp. 31 - 53, 1992.
16. David Pointcheval and Jacques Stern, Security Arguments for Digital Signatures and Blind Signatures, *Journal of Cryptology*, Vol. 13(3), pp. 361 - 396, Springer-Verlag, 2000.

    The full version of the authors' "Security proofs for Signature Schemes" in Eurocrypt '96 and "Provably Secure Blind Signature Schemes" in Asiacrypt '96.

17. A. De Santis, Y. Desmedt, Y. Frankel, and M. Yung, How to share a function securely, in *Proceedings of 26th STOC*, pp. 522 - 533, 1994.
18. S. Wong and Victor K. Wei, A method for imposing spending limit on electronic coins, in *Proceedings of Int'l Symp. on Information Theory*, 1998.
19. Fangguo Zhang and Kwangjo Kim, ID-Based Blind Signature and Ring Signature from Pairings, in *Advances in Cryptology - ASIACRYPT '02*, LNCS 2501, Springer-Verlag, pp. 533 - 547, 2002.