社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2004－ＣＳＥＣ－26　(10)
2004／7／20

# PFDHに対するEUF-SACMAの最適安全性証明

サントソバグス†　太田　和夫†　國廣　昇†

† 電気通信大学電気通信学研究科情報通信工学専攻　〒182-8585 東京都調布市調布ケ丘 1-5-1
E-mail: †{bagus,ota,kunihiro}@ice.uec.ac.jp

# Optimal Security Proof for PFDH under Existential Unforgeability against Strong Adaptive Chosen Message Attack

Bagus SANTOSO†, Kazuo OHTA†, and Noboru KUNIHIRO†

† University of Electro-Communications Department of Information Communication Engineering,
Chofugaoka 1-5-1, Chofu-shi, Tokyo, 182-8585 Japan
E-mail: †{bagus,ota,kunihiro}@ice.uec.ac.jp

**Abstract** In EUROCRYPT2002, Coron proposed optimal security proofs for PSS signature scheme and other signature schemes such as PFDH signature scheme. However the proofs only works under the standard security notion of existential unforgeability against adaptive chosen message attack (EUF-ACMA) [5], while in probabilistic signature scheme such as PSS and PFDH, the strongest security notion is the strong existential unforgeability against adaptive chosen message attack (SEUF-ACMA) [1]. In this paper, we introduce a variant of SEUF-ACMA called SEUF-q-ACMA and show a concrete construction of optimal security proof for PFDH signature scheme under the security notion of SEUF-q-ACMA.

**Key words** PSS, FDH, random oracle model, EUF-ACMA, EUF-SACMA. EUF-$q$-SACMA

## 1. Introduction

For digital signature schemes, the standard "strongest" security notion was defined by Goldwasser, Micali and Rivest in [5], as existential unforgeability under an adaptive chosen message attack (EUF-ACMA). This notion captures the property that an attacker can not produce a valid signature, even after obtaining the signature of messages of his choice. More exactly say, the forged message must be different from the messages previously asked by an adversary to a signature oracle.

Recently, a slightly stronger notion of security, called strong existential unforgeability, was considered in [1], where we require that an adversary can not even generate a new signature on a previously signed messages. This concept of signature security is appropriate for the probabilistic signature schemes, where a fixed message might have several different signatures.

Nowadays, the security of provable secure signature schemes are mostly proven using reduction technique which based on the hardness of a hard problem. The general flow of reduction technique on proving the security of signature scheme is as follows. First, assuming that there is a forger which can output forgery with success probability greater than certain value ($\epsilon_F$) within a given time. Next using the forger, we construct an algorithm that breaks a hard problem, such as discrete logarithm problem or RSA-invertion problem with success probability greater than certain value ($\epsilon_R$)within a given time. If the hard problem is very hard, the success probability of the algorithm should be negligible, thus there is no forger with such success probability. A good reduction algorithm is where $\epsilon_F$ and $\epsilon_R$ are very close. The closeness between $\epsilon_F$ and $\epsilon_R$ is called *tightness*. For practical applications of schemes, the tightness of the security reduction guarantees the appropriate selection of security parameter. For a signature scheme, there could be various reduction algorithms. Each has its own construction method. A reduction algorithm is called optimal if there is no other reduction algorithm has tighter reduction. The optimality of reduction algorithm is very important from both the theoretical and practical viewpoints.

The optimality of PSS signature scheme was comprehensively discussed in [4] under the security definition of the standard secure signature security. The paper [4] gave a proof

of an optimality of some specific reduction described explicitly by adopting a similar approach as Boneh and Venkatesan [2]. This result claims that there is no tighter reduction than the specific reduction given in their paper, which is an surprising result in the theory. However, the discussed security in their paper is not the strongest security concept for probabilistic signature scheme introduced in [1], while the PFDH (Probabilistic Full Domain Hash signature Scheme) and PSS (Probabilistic Signature Scheme) are a kind of the probabilistic signature schemes. In this paper, we will discuss the security reduction of PFDH under the security notion which we consider "stronger" in the probabilistic signature scheme than the standard "strongest" security notion.

We will introduce the new concept of an existential unforgeability under a strong adaptive chosen message attack called *EUF-q-SACMA*. In this attack scenario, the forger can dynamically obtain signatures of messages of his choice, and additionally, and always asks at least $q$ signatures of the message which it attempts to output the valid forgery of. Theorem 1 shows the lower bound of success probability of a specific algorithm $\widetilde{R}$ which reduces inverting RSA to breaking PFDH[$k_0$] in EUF-$q$-SACMA. Theorem 2 shows the upper bound of success probability of any reduction algorithm $\mathcal{R}$ which reduces inverting RSA to breaking PFDH[$k_0$] in EUF-$q$-SACMA. As a result, it is proven that the reduction algorithm $\widetilde{R}$ introduced in the proof of Theorem 1 is actually optimal under EUF-$q$-SACMA.

## 2. Definitions

In this section, we briefly describe some definitions used throughout the paper. Most of the definitions here are adopted from [4].

**[Definition 1] (Signature Scheme)** A signature scheme `(Gen,Sign,Verify)` where `Gen`, `Sign`, `Verify` represent key generation algorithm, signing algorithm, and verification algorithm respectively. For detail definition, please refer to [4]. A *valid signature* is a signature which is accepted by verification algorithm.

**[Definition 2] (Unique Signature Scheme)** A unique signature scheme is a signature scheme `(Gen,Sign,Verify)` where given a pair of public and private keys, the signature scheme produces only one unique valid signature for one message.

**[Definition 3] (Probabilistic Signature Scheme)** A probabilistic signature scheme is a signature scheme `(Gen, Sign,Verify)` where given a pair of public and private keys, the signature scheme is able to produces several different valid signatures for one message.

**[Definition 4] (EUF-ACMA)** An existential unforgeability under an adaptive chosen message attack (EUF-ACMA) is a security notion under a scenario of attack towards a signature scheme, where the forger can dynamically obtain signatures of messages of his choice with a condition that it does not make any signature queries of the message it is going to output the valid forgery of. A *valid forgery* is a pair of a message and a valid signature of the message, where the signature was never retrieved by the forger.

**[Definition 5] (SEUF-ACMA)** A strong existential unforgeability under an adaptive chosen message attack (SEUF-ACMA) is a security notion under a scenario of attack towards a probabilistic signature scheme, where the forger can dynamically obtain signatures of messages of his choice. The forger is allowed to output the forgery of the message which it does not make any signature queries of, and also the forgery of the message which it has made the signature queries of, as long as the forgery is a valid forgery. A *valid forgery* is a pair of a message and a valid signature of the message, where the exactly same signature was never retrieved by the forger.

**[Definition 6] (EUF-q-SACMA)** An existential unforgeability under a $q$ strong adaptive chosen message attack (EUF-$q$-SACMA) is a security notion under a scenario of attack towards the probabilistic signature scheme, where the forger can dynamically obtain signatures of messages of his choice, and additionally, always asks at least $q$ signatures of the message which it attempts to output the valid forgery of. A *valid forgery* is a pair of a message and a valid signature of the message, where the exactly same signature was never retrieved by the forger.

Note that SEUF-ACMA and EUF-$q$-SACMA are different in the sense of the space of the valid forgery. In EUF-$q$-SACMA, the valid forgery includes the forgery of the message the forger has been made the signature queries of at least $q$ times, but the forgery of the message the forger has not been made the signature queries of will not accepted. In SEUF-ACMA, the valid forgery includes the forgery of all messages, both the ones that has not been made the signature queries of and the ones that has been made the signature queries of.

**[Definition 7]** A forger $\mathcal{F}$ is said to $(t, q_H, q_\Sigma, \epsilon)$-break the probabilistic signature scheme `(Gen,Sign,Verify)` under EUF-ACMA if after at most $q_H(k)$ queries to hash oracle, $q_\Sigma$ signature queries, without making any signature queries of the message to forge, within $t(k)$ processing time, it outputs a valid forgery with probability at least $\epsilon(k)$ for all $k \in \mathbb{N}$.

**[Definition 8]** A forger $\mathcal{F}$ is said to $(t, q_H, q_\Sigma, q, \epsilon)$-break the probabilistic signature scheme `(Gen,Sign,Verify)` under EUF-$q$-SACMA if after at most $q_H(k)$ queries to hash oracle, $q_\Sigma$ signature queries including at least $q$ signature queries of the message to forge, within $t(k)$ processing time.

it outputs a valid forgery with probability at least $\epsilon(k)$ for all $k \in \mathbb{N}$.

**[Definition 9]** A probabilistic signature scheme (Gen, Sign, Verify) is $(t, q_H, q_\Sigma, q, \epsilon)$-secure if there is no forger who $(t, q_H, q_\Sigma, q, \epsilon)$-breaks the scheme.

**[Definition 10] (PFDH Signature Scheme)** PFDH signature scheme is a probabilistic signature scheme based on RSA cryptosystem. For the definition and notation of RSA cryptosystem, please refer [4]. Here, let $\mathcal{RSA} : \{0,1\}^k \rightarrow \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z}$, where for input $1^k$, $\mathcal{RSA}(1^k) = (N, e, d)$ s.t $e \cdot d \equiv 1 \pmod{\Phi(N)}$. The signature scheme PFDH is parameterized by the integer $k_0$. The description of PFDH signature scheme is as follows:

**KeyGenPFDH($k$)**
$(N, e, d) \leftarrow \mathcal{RSA}(1^k)$
$pk \leftarrow (N, e)$
$sk \leftarrow (N, d)$
return $(pk, sk)$.

**SignPFDH$_{sk=(N,d)}(M)$**
$r \xleftarrow{R} \{0,1\}^{k_0}$
$y \leftarrow H(M||r)$
return $(y^d \bmod N, r)$

**VerifyPFDH$_{pk=(N,e)}(M, s, r)$**
$y \leftarrow s^e \bmod N$
if $y = H(M||r)$ return 1
else return 0

†Note: $pk$ is public key, $sk$ is private key.

Figure 1 Probabilistic Full Domain Hash (PFDH) Signature Scheme.

**[Definition 11]** An algorithm $\mathcal{A}$ is said to $(t, \epsilon)$-break RSA if given $(N, e, y^e \bmod N)$, within $t(k)$ processing time, it outputs $y \bmod N$ with probability at least $\epsilon(k)$ for all $k \in \mathbb{N}$.

**[Definition 12]** RSA is $(t, \epsilon)$-secure if there is no algorithm which $(t, \epsilon)$-breaks RSA.

**[Definition 13]** A reduction algorithm $\mathcal{R}$ is said to $(t_R, q_H, q_\Sigma, q, \epsilon_F, \epsilon_R)$-reduce inverting RSA to breaking PFDH[$k_0$] under EUF-$q$-SACMA, if upon input $(N, e, y)$ and after running any forger that $(t_F, q_H, q_\Sigma, q, \epsilon_F)$-breaks PFDH[$k_0$], outputs $y^d \bmod N$ with probability at least $\epsilon_R$, within an additional running time of $t_R$.

**[Definition 14]** A reduction algorithm $\mathcal{R}$ is said to $(t_R, q_H, q_\Sigma, \epsilon_F, \epsilon_R)$-reduce inverting RSA to breaking PFDH[$k_0$] under EUF-ACMA, if upon input $(N, e, y)$ and after running any forger that $(t_F, q_H, q_\Sigma, \epsilon_F)$-breaks PFDH[$k_0$], outputs $y^d \bmod N$ with probability at least $\epsilon_R$, within an additional running time of $t_R$.

## 3. Security Analysis of PFDH in EUF-$q$-SACMA

In this section, we will prove the security of PFDH[$k_0$] in EUF-$q$-SACMA using reduction technique, by constructing a reduction algorithm $\mathcal{R}$ which breaks RSA given a forger of PFDH. Then we also prove that the reduction algorithm $\mathcal{R}$ is optimal using similar technique in [4].

### 3.1 A Security Proof of PFDH in EUF-$q$-SACMA

**[Theorem 1]** If RSA is $(t_R, \epsilon_R)$-secure, then PFDH[$k_0$] is $(t_F, q_H, q_\Sigma, q, \epsilon_F)$-secure where $\forall k \in \mathbb{N}$ and $q < 2^{k_0 - 1}$,

$$t_R = t_F + (q_H + q_\Sigma)\mathcal{O}(k^3) \quad (1)$$

$$\epsilon_R = \epsilon_F \frac{1}{1 + 14(q_\Sigma - q)2^{-k_0}} \quad (2)$$

**Proof:** The proof is given in appendix A.

### 3.2 Optimal Security Proof for PFDH in EUF-$q$-SACMA

**[Theorem 2]** Let $\mathcal{R}$ a reduction algorithm which $(t_R, q_H, q_\Sigma, q, \epsilon_F, \epsilon_R)$-reduces inverting RSA to breaking PFDH[$k_0$], with $q_H \geqq q_\Sigma$ and $q < 2^{k_0 - 1}$. $\mathcal{R}$ can run or rewind the forger at most $r$ times. Then, from $\mathcal{R}$ we can construct an algorithm $\mathcal{I}$ which $(t_I, \epsilon_I)$-breaks RSA with:

$$t_I = (r + 1) \cdot t_R \quad (3)$$

$$\epsilon_I = \epsilon_R - r \cdot \epsilon_F \cdot \frac{2^{k_0 + 2}}{q_\Sigma - q} \quad (4)$$

**Proof:** The proof is given in appendix C. □

## 4. Discussion

The flow of the discussion is following the discussion section in [4].

### 4.1 Discussion on Theorem 1

From (2), we can see that when $k_0 = 0$, we lost about $\log_2 q_\Sigma$ of security compared to RSA(*1). In this case, since PFDH becomes a unique signature scheme, EUF-$q$-SACMA and EUF-ACMA scenario are completely coincided. When $0 < k_0 < \log_2(q_\Sigma - q)$, as long as $q < 2^{k_0 - 1}$, every bit of $k_0$ adds one bit security to the scheme. Finally, when $k_0 \geqq \log_2(q_\Sigma - q)$, we can obtain a security level that is almost the same as inverting RSA. Note that the larger $k_0$ is, the closer $\epsilon_F$ to $\epsilon_R$ is. For example, given time $t_F$ taking $k_0 = \log_2(q_\Sigma - q)$ then the probability of breaking PFDH[$k_0$] where $q < 2^{k_0 - 1}$ is at least $\epsilon_F = \epsilon_R \cdot 15$ where $\epsilon_R$ is the least probability to break RSA in at most time close to $t_F$. Suppose we want to design PFDH[$k_0$] with $q_\Sigma$ being the number of signatures to publish within time $t_F$, against any forger that can output valid forgery from at least $q$ signatures of target message to forge. So, in this case, in order to obtain the security level of PFDH[$k_0$] being almost as secure as inverting RSA, we can set $k_0 \geqq \max\{\log_2 q, \log_2(q_\Sigma - q)\}$.

(*1) : $q = 0$ because when $k = 0$ PFDH becomes FDH which is a unique signature scheme.

### 4.2 Discussion on Theorem 2

#### 4.2.1 General Discussion

Let consider a forger $\mathcal{F}$ which breaks PFDH[$k_0$] after $q_\Sigma$ signature queries (including $q$ signature queries of message to forge) and outputs forgery with $\epsilon_F = 1/2$. Then, according to theorem 2, from any reduction $\mathcal{R}$ that succeeds with probability at least $\epsilon_R$ when running the $\mathcal{F}$ in EUF-$q$-SACMA scenario where $q < 2^{k_0-1}$ once, we can obtain a polynomial time algorithm $\mathcal{I}$ which succeeds in breaking RSA with probability at least $\epsilon_R - 2^{k_0+1}/(q_\Sigma - q)$, without using $\mathcal{F}$. If inverting RSA is hard, the probability of breaking RSA should be negligible. Thus, the success probability of any such reduction algorithm should be at most $2^{k_0+1}/(q_\Sigma - q) + \mathrm{negl}$. Hence, in order to get tight reduction ($\epsilon_R \backsimeq \epsilon_F = 1/2$), we need to set $k_0 \backsimeq \log_2(q_\Sigma - q)$. This result is almost the same as one shown in theorem 1. Thus, this shows that our reduction algorithm used in theorem 1 is almost optimal.

#### 4.2.2 Concrete Example

For more precise illustration, let us observe above discussion with a concrete example. Here we assume that we work on 1024-bit modulus RSA and $q < 2^{k_0-1}$.

First, recall that our reduction algorithm $\widetilde{\mathcal{R}}$ in theorem 1 succeeds in inverting RSA using PFDH[$k_0'$] with probability at least:

$$\epsilon_{\widetilde{R}} = \frac{\epsilon_F}{1 + 14 \cdot (q_\Sigma - q)2^{-k_0'}}$$

Setting $k_0' = \log_2(q_\Sigma - q)$, for any forger $\mathcal{F}$ with success probability at least $\epsilon_F = 1/2$ ,we obtain that $\widetilde{\mathcal{R}}$ will invert RSA with probability at least 1/30. Here, assume that the running time of $\widetilde{\mathcal{R}}$ is less than $2^{50}$.

Now, let us consider another reduction $\mathcal{R}$ on PFDH[$k_0$] in the same EUF-$q$-SACMA with the same running time $2^{50}$ which succeeds with probability at least $\epsilon_R \geqq \epsilon_{\widetilde{R}}$ using the same forger $\mathcal{F}$. Thus, from theorem 2, we can construct an algorithm $\mathcal{I}$ which inverts RSA without $\mathcal{F}$ with success probability at least $\epsilon_I = \epsilon_R - 2^{k_0+1}/(q_\Sigma - q)$ within running time $t_I = (1+1)2^{50} = 2^{51}$.

As standard of measurement, let us use the best factoring algorithm known (number field sieve (NFS)).The running time of NFS for factoring a modulus $N$ is about

$$T_{NFS}(k) = exp(C \cdot (\log N)^{1/3} \cdot (\log\log N)^{2/3})$$

where $C \backsimeq 1.923$ [4]. Therefore we can assume RSA is $(t, \epsilon)$-secure for any $(t, \epsilon)$ satisfying $t(k)/\epsilon(k) < T_{NFS}(k)$. For 1024-bit modulus, we have $T_{NFS}(k) \backsimeq 2^{86}$. Therefore, we can assume that for a 1024-bit modulus, we can obtain that RSA is $(t, t \cdot 2^{-86})$-secure for all $t < 2^{86}$. Putting $t = 2^{51}$, we can obtain that RSA is $(2^{51}, 2^{-35})$ -secure which means that given $t_I = 2^{51}$, $\epsilon_I < 2^{-35}$. Thus, the success probability of reduction algorithm $\mathcal{R}$ can not be greater than $2^{k_0+1}/(q_\Sigma - q) + 2^{-35}$. Since $\epsilon_R \geqq \epsilon_{\widetilde{R}}$, we must have $2^{k_0+1}/(q_\Sigma - q) + 2^{-35} \geqq 1/30$, which gives $k_0 \geqq \log_2(q_\Sigma - q) - 6$. This shows that the difference between $k_0$ from any reduction algorithm $\mathcal{R}$ and the one from $\widetilde{\mathcal{R}}$, where $\epsilon_R \geqq \epsilon_{\widetilde{R}}$, can be bounded up to a constant factor. Consequently, the reduction algorithm $\widetilde{\mathcal{R}}$ in theorem 1 is optimal.

### 4.3 Comparison of PFDH[$k_0$] in EUF-ACMA and EUF-$q$-SACMA

Assume that RSA is $(t, \epsilon)$-secure. Then from [4], we can obtain that PFDH[$k_0$] is $(t_F, q_\Sigma, q_H, \epsilon_F^0)$-secure under EUF-ACMA where $\epsilon_F^0 = \epsilon_R(1 + 6 \cdot \log_2 q_\Sigma \cdot 2^{-k_0})$. Whereas, from theorem 1, with the same condition, we can obtain that PFDH[$k_0$] is $(t_F, q_\Sigma, q_H, q, \epsilon_F^q)$-secure under EUF-$q$-SACMA where $\epsilon_F^q = \epsilon_R(1 + 14 \cdot \log_2(q_\Sigma - q) \cdot 2^{-k_0})$. It is easy to see that in order to gain the same security level as inverting RSA, the minimal requirement of $k_0$ differs between those scenarios. Under EUF-ACMA, we need to set $k_0 \backsimeq \log_2 q_\Sigma$, while under EUF-$q$-SACMA, we only need to set $k_0 \backsimeq \log_2(q_\Sigma - q)$ which is smaller(*2) than $\log_2 q_\Sigma$.Hence, the scenario attack in EUF-$q$-SACMA is *weaker* ( easier to handle ) than the scenario attack in EUF-ACMA.

At first glance, this seems to contradict our intuition, since the scenario attack in EUF-$q$-SACMA seems stronger than the one in EUF-ACMA, due to the fact that the scenario attack in EUF-$q$-SACMA is allowing to gain some information about the signature of the message the forger is going to forge, whereas the forger in EUF-ACMA is not.

However, *is it true* that the scenario attack in EUF-$q$-SACMA is stronger than the scenario attack in EUF-ACMA ? Again, it is indeed true that the forger in EUF-$q$-SACMA can gain some information about the signature of the message it is going to forge, whereas the forger in EUF-ACMA can not. However, looking more detail into inside EUF-$q$-SACMA, we get some restrictions on EUF-$q$-SACMA as follows:

- The forger in EUF-$q$-SACMA is able to only forge the message which it has received the signature of, for at least $q$ times. This restriction makes the forger *is not allowed* to output valid forgery of the message it has never asked the signature of.
- Let $M^*$ denote the candidate of the message the forger

(*2) : In fact, we should examine this argument more carefully by using non-simplified version of the result of EUF-$q$-SACMA. Because the simplified version of the result of EUF-$q$-SACMA here has bigger constant coefficient than the one in EUF-ACMA, which can cause misleading arguments, such as $\epsilon_F^q > \epsilon_F^0$ when $q$ is small, which is actually not true at all.

is going to output. Suppose that the forger has received at least $q$ signatures of $M^*$. Thus, along with the signatures of $M^*$, it also has received a set of random salts $\bar{r} = \{r_i^*\}_{i=1}^{j}$ where $j \geqq q$. Remind that a valid forgery is a forgery of $M^*$ with random salt $r^*$ which is not contained in $\bar{r}$. Thus the more forger receives signatures of $M^*$, the less the number of candidate of $r^*$ it is allowed to output in valid forgery.

These restrictions decreases the degree of freedom of forgery in EUF-$q$-SACMA and thus decreases the strength of the attack scenario. As an comparison let us analyze the forger of EUF-ACMA.

- The forger of EUF-ACMA is allowed to output any valid forgeries of any messages it has never asked the signature of.
- Since the message that the forger of EUF-ACMA is going to forge, is always the one it has never asked to the signature oracle, there is no restriction on the candidate of the random salt it can output in forgery.

From above arguments, it is obvious that actually the forgery in EUF-$q$-SACMA is more restricted than the forgery in EUF-ACMA.

## 5. Conclusion and Further Research

The strongest security notion for probabilistic signature scheme is the strong existential unforgeability under an adaptive chosen message attack (SEUF-ACMA). However the optimal security proofs for PSS and PFDH signature schemes in [4] only works under the standard security notion of existential unforgeability against adaptive chosen message attack (EUF-ACMA) [5]. In this paper, we have show a concrete reconstruction of an optimal security proof for PFDH[$k_0$] signature scheme under a variant of SEUF-ACMA called EUF-$q$-SACMA where $q < 2^{k_0-1}$.

Finally, the optimal security proof for general $q$ under EUF-$q$-SACMA is still needed to be investigated. The optimal proof of probabilistic signature scheme under the strongest notion SEUF-ACMA still also remains as an open problem.

### References


[1] Jee Hea An, Yevgeniy Dodis, and Tal Rabin. On the security of joint signature and encryption. In *EUROCRYPT 2002*, volume 2332 of *LNCS*. Springer-Verlag, 2002.

[2] D. Boneh and R. Venkatesan. Breaking rsa may not be equivalent to factoring. In *EUROCRYPT 1998*, volume 1233 of *Lecture Notes in Computer Science*, pages 59–71. Springer-Verlag, 1998.

[3] Dan Boneh and Xavier Boyen. Short signatures without random oracles. In *Advances in Cryptology—EUROCRYPT 2004*. volume 3027 of *Lecture Notes in Computer Science*, pages 56–73. Berlin: Springer-Verlag, 2004.

[4] Jean-Sebastien Coron. Optimal security proofs for pss and other signature schemes. In L.R. Knudsen, editor, *EUROCRYPT 2002*. volume 2332 of *Lecture Notes in Computer Science*, pages 272–287. Springer-Verlag, 2002.

[5] S. Goldwasser, S. Micali, and R. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. In *SIAM Journal of computing*, volume 17(2), pages 281–308, April 1988.


## Appendix

### A Proof of Theorem 1

Our proof is following the proof in [4]. We will show that for any forger that $(t_F, q_H, q_\Sigma, q, \epsilon_F)$-breaks PFDH[$k_0$], we can construct reduction algorithm $\mathcal{R}$ which $(t_R, \epsilon_R)$-breaks RSA where the variables satisfy (1) and (2).

The reduction algorithm $\mathcal{R}$ here is also adopted from [4]. It uses a forger $\mathcal{F}$ to invert RSA, and it has ability of answering hash and signature queries of forger $\mathcal{F}$.

**Algorithm for $\mathcal{R}$:**

Input: $(N, e, y)$, and $(q_H, q_\Sigma, q)$, where
$(N, e) \leftarrow RSA(1^k)$ and $y \stackrel{R}{\leftarrow} \mathbb{Z}^*{}_N$

Output: $y^d \bmod N$

1. Set $i \leftarrow 0$
2. Send $(N, e)$ to $\mathcal{F}$.
3. If $\mathcal{F}$ makes a hash query for $M||r$ then
   Find $j \in \mathbb{N}$ s.t $M_j = M$
   If $\nexists j \in \mathbb{N}$ s.t $M_j = M$,
   $i \leftarrow i + 1$; $M_i \leftarrow M$; $j \leftarrow i$;
   $L_j \leftarrow \texttt{InitList}(q_\Sigma, \beta)$
   If $r \in L_j$ then
   $x \stackrel{R}{\leftarrow} \mathbb{Z}_N^*$, Return $H(M||r) = x^e \bmod N$.
   Else if $r \notin L_j$ then
   Return $H(M||r) = y \cdot^e \bmod N$.
4. If $\mathcal{F}$ makes a signature query for $M$ then
   Find $j \in \mathbb{N}$ s.t $M_j = M$
   If $\nexists j \in \mathbb{N}$ s.t $M_j = M$,
   $i \leftarrow i + 1$; $M_i \leftarrow M$; $j \leftarrow i$;
   $L_j \leftarrow \texttt{InitList}(q_\Sigma, \beta)$
   If $L_j \neq \emptyset$ then pick an element $r'$ of $L_j$ and discard it from $L_j$. Otherwise stop.
   If $M||r'$ has been asked before, it is easy to see that $H(M||r') = x^e \bmod N$.
   If $M||r'$ has never been asked yet, $x \stackrel{R}{\leftarrow} \mathbb{Z}_N^*$;
   $H(M||r') \leftarrow x^e \bmod N$.
   Return $\sigma(M) = (x, r')$.
5. If $\mathcal{F}$ outputs a forgery $(M^*, \sigma^*, r^*)$,
   If $H(M^*||r^*) = y \cdot x^e \bmod N$ then
   output $y^d = \sigma^*/x \bmod N$.
   Otherwise stop.

**Algorithm** `InitList`

Input: $q_\Sigma$ and $\beta \in [0, 1)$

Output: $L \in \mathcal{P}(\mathbb{Z}_{2^{k_0}}^{q_\Sigma})$

1. $L \leftarrow \emptyset$, $j \leftarrow 0$.
2. If $j < q_\Sigma$ then
   Flip a coin $c \in \{0,1\}$ which is biased by $\beta$, where $c = 0$ with probability $\beta$ and $c = 1$ with $1 - \beta$.
   Else go to step 4.
3. If $c = 0$ then
   $j \leftarrow j + 1$;
   $r'_j \stackrel{R}{\leftarrow} \{0,1\}^{k_0}$;
   Add $r'_j$ into $L$ and go to step 2.
   Else if $c = 1$ then goto step 4.
4. Return $L$.

First, in order to analyze the reduction algorithm $\mathcal{R}$, we will try to find the success probability of the signature oracle simulation by $\mathcal{R}$, e.g., even after at most $q_\Sigma$, $\mathcal{F}$ does not know that the signature oracle is in fact a simulated one. Next, from step 4 of reduction algorithm $\mathcal{R}$, we know that every time $\mathcal{F}$ makes a signature query $M_i$, the number element of its corresponding list $(L_i)$ is decreased by one. Whereas, from the construction of list $L_i$ in algorithm `InitList`, we have :

$$\Pr[|L_i| = j] = \begin{cases} \beta^j(1-\beta) & if\ j < q_\Sigma \\ \beta^{q_\Sigma} & if\ j = q_\Sigma, \end{cases} \quad (A\cdot1)$$

where $|L_i|$ denote s the number of elements of $L_i$.

Thus, the probability that $\mathcal{R}$ can answer one signature query of $M_i$ is $\Pr[|L_i| \neq 0] = \beta$. Let $Y_i$ denote the number of signature queries of $M_i$, and especially let $Y^*$ denote the number of signature queries of $M^*$, the candidate message to forge, with $L^*$ as its corresponding list. It is easy to see that probability $\mathcal{R}$ can answer $Y_i$ signature queries of $M_i$ is $\Pr[|L_i| \geqq Y_i] = \beta^{Y_i}$. Let vector $Y_1, Y_2, \ldots, Y^*, \ldots, Y_{q_H}$ denote the list of the number of signature queries of $M_1, M_2, \ldots, M^*, \ldots, M_{q_H}$. Note that although in EUF-$q$-SACMA model $Y^* \geqq q$, this will not affect the total $\sum_{i=1}^{q_H} Y_i \leqq q_\Sigma$. Hence, from the independency of each $Y_i$, we can conclude that the probability that $\mathcal{R}$ can answer all signature queries from $\mathcal{F}$ is at least $\beta^{q_\Sigma}$. Consequently, the total probability that $\mathcal{R}$ gets forgery from $\mathcal{F}$ is at least $\beta^{q_\Sigma} \cdot \epsilon_F$.

Let $\overline{L_i}$ denote the part of $L_i$ which is still left after $\mathcal{F}$ has finished all its signature queries. From step 6, we know that the inversion of RSA will success only when forgery $(M^*, \sigma^*, r^*)$ satisfies $r^* \notin \overline{L^*}$. Since $\mathcal{F}$ knows nothing about $\overline{L^*}$, $r^*$ from forgery and the contents of $\overline{L^*}$ are independent each other, the probability of $r^* \notin \overline{L^*}$ when $|L^*| = \ell$ is $\left(1 - 2^{-k_0}\right)^\ell$. Thus, the total probability that $r^* \notin \overline{L^*}$ is:

$$f(\beta, Y^*) = \sum_{\ell=0}^{q_\Sigma - Y^*} (1 - 2^{-k_0})^\ell \Pr[\overline{L^*} = \ell], \quad (A\cdot2)$$

where

$$\Pr[|\overline{L^*}| = \ell] = \begin{cases} \beta^\ell(1-\beta) & if\ \ell < q_\Sigma - Y^* \\ \beta^{q_\Sigma - Y^*} & if\ \ell = q_\Sigma - Y^*. \end{cases} \quad (A\cdot3)$$

Since $Y^* \geqq q$, it is easy to see that $f(\beta, Y^*) \geqq f(\beta, q)$ for a fixed $\beta$. Thus the total probability that $\mathcal{R}$ success in inverting RSA is at least $\beta^{q_\Sigma} \cdot \epsilon_F \cdot f(\beta, Y^*)$. And for any $(q_\Sigma, k_0, q)$ when $q < 2^{k_0 - 1}$, there exists $\beta_0$ such that:

$$\beta_0^{q_\Sigma} \cdot f(\beta_0, q) \geqq \frac{1}{1 + 14 \cdot (q_\Sigma - q) 2^{-k_0}}, \quad (A\cdot4)$$

which gives (2). Detail proof is given in appendix B.
Similarly, the running time of $\mathcal{R}$ is the running time of $\mathcal{F}$ plus the time to compute answer for hash queries and signature queries, approximately within $\mathcal{O}(k^3)$ for each query. This leads to (1). □

## B Proof of inequality A·4

The proof here is mostly adopted from appendix B of [4]. Let

$$g(\beta) = \beta^{q_\Sigma} \cdot \sum_{\ell=0}^{q_\Sigma - q} (1 - 2^{-k_0})^\ell \cdot \Pr[|L^*| = \ell] \quad (A\cdot5)$$

where

$$\Pr[|\overline{L^*}| = \ell] = \begin{cases} \beta^\ell(1-\beta) & if\ \ell < q_\Sigma - Y^* \\ \beta^{q_\Sigma - Y^*} & if\ \ell = q_\Sigma - Y^*. \end{cases} \quad (A\cdot6)$$

Let $g_0 = \max\{g(\beta); \beta \in [0,1]\}$, we will show that

$$g_0 \geqq \left(1 - \frac{1}{2(q_\Sigma - q)}\right)^q \frac{1}{1 + 6(q_\Sigma - q)2^{-k_0}}$$

Put $\gamma = 2^{-k_0}$, from (A·5) and (A·6), we get:

$$g(\beta) = \frac{\beta^{q_\Sigma}}{1 - (1-\gamma)\beta} \cdot \left(1 - \beta + \gamma\beta\left(\left((1-\gamma)\beta\right)^{q_\Sigma - q}\right)\right) \quad (A\cdot7)$$

When $\gamma \cdot (q_\Sigma - q) \geqq 1/2$, first we derive:

$$g(\beta) \geqq \beta^{q_\Sigma} \cdot \frac{1}{1 - \beta + \gamma},$$

then we take $\beta = 1 - 1/(2(q_\Sigma - q))$ and obtain:

$$g_0 \geqq \left(1 - \frac{1}{2(q_\Sigma - q)}\right)^{q_\Sigma} \cdot \frac{1}{1 + 2 \cdot \gamma(q_\Sigma - q)}.$$

For $(q_\Sigma - q) \geqq 1$ we have

$$\left(1 - \frac{1}{2(q_\Sigma - q)}\right)^{q_\Sigma - q} \geqq \frac{1}{2}.$$

Also, from $\gamma \cdot (q_\Sigma - q) \geqq 1/2$ and $q < 2^{k_0 - 1}$ we have

$$\left(1 - \frac{1}{2(q_\Sigma - q)}\right)^q \geqq \left(1 - \frac{1}{2^{k_0}}\right)^q \geqq \left(1 - \frac{1}{2 \cdot q}\right)^q \geqq \frac{1}{2}.$$

Hence, we obtain

$$g_0 \geqq \frac{1}{4 \cdot (1 + 2 \cdot \gamma \cdot (q_\Sigma - q))} = \frac{1}{4 + 8 \cdot \gamma \cdot (q_\Sigma - q)}.$$

Using $\gamma \cdot (q_\Sigma - q) \geqq 1/2$, we obtain

$$(2(4-1)+8-8) \cdot \gamma(q_\Sigma - q) \geqq (4-1)$$
$$1 + 14 \cdot \gamma(q_\Sigma - q) \geqq 4 + 8 \cdot \gamma(q_\Sigma - q).$$

Thus,

$$g_0 \geqq \frac{1}{1 + 14 \cdot \gamma \cdot (q_\Sigma - q)}.$$

If $\gamma \cdot (q_\Sigma - q) \leqq 1/2$, we take $\beta = 1$ and obtain using (A·7):

$$g_0 \geqq (1-\gamma)^{q_\Sigma - q} \geqq 1 - \gamma \cdot (q_\Sigma - q) \geqq \frac{1}{1 + 2 \cdot \gamma(q_\Sigma - q)}$$
$$\geqq \frac{1}{1 + 14 \cdot \gamma(q_\Sigma - q)}$$

□

## C Proof of Theorem 2

The general flow of the construction of algorithm $\mathcal{I}$ is following [4].

### C 1 Preliminaries

#### C 1.1 Construction of $\mathcal{I}$ under unique signature scheme

First, suppose we have a reduction algorithm $\mathcal{R}$ which satisfies the requirements in theorem 2. Then we simulate the forger oracle $\mathcal{F}$ inside $\mathcal{R}$ by using rewinding technique as follows: in the first run, in some arbitrary point we make a signature query of $M^*$ and stop. Next, we rewind $\mathcal{R}$ to the point before we make any signature queries and make signature queries exactly as in the first run with exception that we do not make a signature query of $M^*$, and to continue making at most $q_\Sigma$ signature queries to $\mathcal{R}$. After that, we make $\mathcal{F}$ output a forgery of $M^*$ which is retrieved in the first run. Unless $\mathcal{R}$ does not give valid answer of the signature query of $M^*$ in the first run, $\mathcal{R}$ will not be able to distinguish between the simulated forger and a real forger which runs exactly the same signature queries of the second run. For detail discussion about the simulation, please refer to [4].

#### C 1.2 Construction of $\mathcal{I}$ under non-unique signature scheme

Unfortunately, the simulation of forger oracle showed above only works only if the underlying signature scheme is a unique signature scheme. In the case of probabilistic signature scheme as PFDH[$k_0$], the forgery result of simulated forger will always be producible by $\mathcal{R}$, whereas the forgery of a real forger will not always be producible by $\mathcal{R}$. Thus, $\mathcal{R}$ can construct an algorithm to distinguish between a simulated forger and a real forger. For example, after forger oracle outputs forgery of $M^*$, $\mathcal{R}$ can rewind itself and try to send a signature query of $M^*$ in any point. Since the number of signature queries is finite, if $M^*$ is producible by $\mathcal{R}$, it will find exactly the same forgery of $M^*$ in a finite time, thus the forger is a simulated forger. Otherwise, it will not find exactly same forgery of $M^*$ in a given time, thus the forger is a real forger.

### C 2 Sketch of Proof

Let consider a variant of PFDH[$k_0$], called PFDH0[$k_0$] which the random salt is fixed to $0^{k_0}$. Note that PFDH0[$k_0$] is a unique signature scheme. First we convert a forger $\mathcal{F}_0$ for PFDH0[$k_0$] into a forger $\mathcal{F}$ for PFDH[$k_0$]. $\mathcal{F}$ should work under the scenario of EUF-$q$-SACMA inside a reduction algorithm $\mathcal{R}$ which reduces inverting RSA into breaking PFDH[$k_0$] and satisfies the requirements in theorem 2. Since $\mathcal{R}$ is indeed a reduction algorithm that inverts RSA, using $\mathcal{F}$ which is constructed from $\mathcal{F}_0$, it is easy to see that we can construct a reduction $\mathcal{R}_0$ which reduces inverting RSA into breaking PFDH0[$k_0$], by just forwarding the queries from $\mathcal{F}_0$ to $\mathcal{R}_0$. For detail discussion, please refer to appendix J of [4].

**[Lemma 3]** Let $\mathcal{F}_0$ be a forger which $(t_F^0, q_H^0, q_\Sigma^0, \epsilon_F^0)$-breaks PFDH0[$k_0$]. From $\mathcal{F}_0$ we can construct a forger $\mathcal{F}$ which $(t_F, q_H, q_\Sigma, q, \epsilon_F)$-breaks PFDH[$k_0$], with:(*3)

$$q_H = q_H^0, \quad q_\Sigma = 2^{k_0+1} \cdot q_\Sigma^0 + q, \quad \textit{and} \quad \epsilon_F = \epsilon_F^0/2 \cdot (1-2^{-k_0})^q$$

**Proof:** We construct $\mathcal{F}$ from $\mathcal{F}_0$ by forwarding the queries from $\mathcal{F}_0$ to the hash or signature oracle of PFDH[$k_0$], and forwarding the answers from the hash or signature oracle to $\mathcal{F}_0$. Eventually the forger $\mathcal{F}_0$ will output forgery of PFDH0[$k_0$] which also a forgery of PFDH[$k_0$].

The most important thing to note is that $\mathcal{F}_0$ will always expect the random salt part of the signature of $M$ be $0^{k_0}$ as in PFDH0[$k_0$], whereas the signature of $M$ retrieved from the signature oracle of PFDH[$k_0$] is equal to $0^{k_0}$ only with probability $2^{-k_0}$. Also remind that $\mathcal{F}$ needs to send signature queries of the message it is going to output as forgery, here $M^*$, at least $q$ times. Here, we fix the number of signature queries of the message to forge to $q$ times(*4)

So the total number of queries sent to signature oracle in order to serve $\mathcal{F}_0$, need to be reduced from $q_\Sigma$ to $q_\Sigma - q$. And the $q$ signature queries of $M^*$ need not to contain the one with $0^{k_0}$ random salt, otherwise the forgery from $\mathcal{F}_0$ is useless.

Let $Y_i$ denote the number of signature queries of $M_i$ need to be sent to signature oracle by in order to retrieve a signature of $M_i$ with $0^{k_0}$ as its random salt. Let $Y = \sum_{i=1}^{q_\Sigma^0} Y_i$, thus $\mathcal{F}$ will output forgery of $(M^*, \sigma^*, 0_0^k)$ with probability at least $\epsilon_F^0 \cdot \Pr[Y \leqq q_\Sigma - q] \cdot (1-2^{-k_0})^q$.

Reusing the result shown in [4], we get $\Pr[Y \leqq (q_\Sigma - q)] \geqq 1/2$. Thus, the forger $\mathcal{F}$ outputs a forgery for PFDH[$k_0$] with

(*3) : Although the attack framework of $\mathcal{F}_0$ and $\mathcal{F}$ are different, it does not affect the generality of our proof, since a forger of EUF-$q$-SACMA will behave exactly with a forger of EUF-ACMA in a unique signature scheme.

(*4) : Even though we fix the number of signature queries of the message to forge into $q$ times, it does not affect the generality of our proof as long as we can construct $\mathcal{F}$

probability at least $\epsilon_F = \epsilon_F^0/2 \cdot (1 - 2^{-k_0})^q$ after at most $q_\Sigma$ signature queries, including $q$ signature queries of message to forge. □

**[Lemma 4]** Let $\mathcal{R}$ be a reduction which $(t_R, q_H, q_\Sigma, q, \epsilon_F, \epsilon_R)$-reduces inverting RSA to breaking PFDH[$k_0$]. From $\mathcal{R}$ we can construct a reduction $\mathcal{R}_I$ which $(t_R^0, q_H^0, q_\Sigma^0, \epsilon_F^0, \epsilon_R)$-reduces inverting RSA to breaking PFDH0[$k_0$], with:

$$q_\Sigma = q_\Sigma^0 \cdot 2^{k_0+1} + q \qquad q_H = q_H^0 \qquad \text{(A·8)}$$

$$\epsilon_F = \epsilon_F^0/2 \cdot (1 - 2^{-k_0})^q \qquad \epsilon_R = \epsilon_R^0 \qquad \text{(A·9)}$$

$$t_R^0 = t_R \qquad \text{(A·10)}$$

**Proof:** From a forger $\mathcal{F}_0$ which $(t_F^0, q_H^0, q_\Sigma^0, \epsilon_F^0)$-breaks PFDH0[$k_0$], it is obvious from lemma 3 that we can construct a forger $\mathcal{F}$ which $(t_F, q_H, q_\Sigma, q, \epsilon_F)$-breaks PFDH[$k_0$], with parameter settings as (A·8), (A·9) and (A·10). Accordingly, given $\mathcal{R}$, using $\mathcal{F}$ we can invert RSA. Notice that this procedure shows a algorithm to invert RSA given a forger $\mathcal{F}_0$ using $\mathcal{R}$. Thus, it is easy to see that in fact we have succeeded in constructing an $\mathcal{R}_I$ where $\epsilon_R^0 = \epsilon_R$ and $t_R^0 = t_R$.

□

Now, let $\mathcal{R}$ be a reduction algorithm which $(t_R, q_H, q_\Sigma, q, \epsilon_F, \epsilon_R)$-reduces inverting RSA to breaking PFDH[$k_0$]. According to lemma 4, we can construct a reduction algorithm $\mathcal{R}_0$ which $(t_R^0, q_H^0, q_\Sigma^0, \epsilon_F^0, \epsilon_R^0)$-reduces inverting RSA to breaking PFDH0[$k_0$] with parameter settings as shown in (A·8), (A·9) and (A·10). If $\mathcal{R}$ can run or rewind the forger at most $r$ times, so can $\mathcal{R}_I$. First recall the following theorem from [4].

**[Theorem 5]** Let $\mathcal{R}_I$ be a reduction which $(t_R^0, q_H^0, q_\Sigma^0, \epsilon_F^0, \epsilon_R^0)$-reduces inverting RSA to breaking a unique signature scheme. If $\mathcal{R}_0$ is allowed to run or rewind a forger at most $r$ times, then from $\mathcal{R}_0$ we can construct an inverter algorithm $\mathcal{I}$ which $(t_I, \epsilon_I)$-breaks RSA, with:

$$t_I = (r+1) \cdot t_R^0 \qquad \text{(A·11)}$$

$$\epsilon_I = \epsilon_R^0 - r \cdot \epsilon_F^0 \cdot \frac{exp(-1)}{q_\Sigma^0} \cdot \left(1 - \frac{q_\Sigma^0}{q_H^0}\right)^{-1} \qquad \text{(A·12)}$$

Using equations (A·8), (A·9) with $q_H \geqq q_\Sigma$, $q < 2^{k_0-1}$ and $exp(-1) \leqq 1/2$, we obtain:

$$\begin{aligned} & r \cdot \epsilon_F^0 \cdot \frac{exp(-1)}{q_\Sigma^0} \cdot \left(1 - \frac{q_\Sigma^0}{q_H^0}\right)^{-1} \\ & \leqq r \cdot \frac{2\epsilon_F}{(1-2^{-k_0})^q} \frac{1}{2(q_\Sigma - q)2^{-k_0-1}} \frac{1}{(1-(q_\Sigma - q)2^{-k_0-1}q_H^{-1})} \\ & \leqq r \cdot \epsilon_F \frac{2^{k_0+1}}{(1-2^{k_0})^q (q_\Sigma - q)} \leqq r \cdot \epsilon_F \frac{2^{k_0+2}}{q_\Sigma - q} \end{aligned}$$

Hence, the inverter algorithm $\mathcal{I}$ will succeed with probability at least:

$$\epsilon_R - r \cdot \epsilon_F \cdot \frac{2^{k_0+2}}{q_\Sigma - q},$$

and thus theorem 2 is proven. □