Key-Privacy in Digital Signature (Extended Abstract)

Ryotaro Hayashi   Keisuke Tanaka *

Dept. of Mathematical and Computing Science
Tokyo Institute of Technology
2-12-1 Ookayama, Meguro-ku, Tokyo 152-8552, Japan
{hayashi9, keisuke}@is.titech.ac.jp

February 13, 2003

Summary

Bellare , Boldyreva , Desai, and Pointcheval proposed a new security requirement of encryption schemes called "key-privacy". It asks that the encryption provide (in addition to privacy of the data being encrypted) privacy of the key under which the encryption was performed. They proposed a variant of RSA-OAEP that provides key-privacy in the random oracle model assuming RSA is one-way. After giving a more efficient scheme with respect to the number of exponentiations for encryption, we propose a new security requirement of undeniable signature scheme corresponding to key-privacy in encryption scheme. We prove that the discrete-log based scheme proposed by Michels and Stadler provides anonymity and existential unforgeability in the random oracle model assuming the Decision Diffie–Hellman problem is hard. We also propose a variant of an RSA-based scheme proposed by Gennaro, Krawczyk, and Rabin that provides anonymity and existential unforgeability in the random oracle model assuming RSA is one-way and the EDL problem is hard.

Keywords: key-privacy, anonymity, encryption, undeniable signature

1 Introduction

The classical security requirements of an encryption scheme, for example indistinguishability or non-malleability under chosen-ciphertext attack, provide privacy of the encryption data. In [1], Bellare , Boldyreva , Desai, and Pointcheval proposed a new security requirement of encryption schemes called "key-privacy". It asks that the encryption provide (in addition to privacy of the data being encrypted) privacy of the key under which the encryption was performed.

The standard RSA does not provide key-privacy. Because even if two public-key N_0 and N_1 ($N_0 < N_1$) are the same bits, $N_1 - N_0$ may be large. In [1], they propose a "key-privacy" encryption scheme RSA-RAEP which is a variant of RSA-OAEP[7]. Their variant simply repeats the ciphertext computation, by using new coins each time, until the ciphertext y falls in some "safe" range.

*Supported in part by Grant-in-Aid for Scientific Research, Ministry of Education, Culture, Sports, Science and Technology, 14780190.

Before going into the main part of this paper, we propose a more efficient scheme with respect to the number of exponentiations for encryption. Suppose that N is uniformly distributed in $[2^{k-1}, 2^k)$, the expected number of our scheme is the same as RSA-RAEP. But in our scheme, the number of exponentiations for encryption is at most two, while in RSA-RAEP, the upper bound of this number is $k_1(\gg 2)$ (security parameter).

In this paper, we are mainly concerned with undeniable signature. Digital signatures are easily verified by anyone using the corresponding public key. This property can be advantageous for many users, but it is unsuitable for many other users. In [2], Chaum and Antwerpen provided undeniable signature which cannot be verified without the signer's cooperation, and many undeniable signature schemes are provided which satisfy existential unforgeability. But in such schemes, the verifier may decide it is someone's signature or it isn't. In this paper, we propose a new security requirement of undeniable signature schemes corresponding to key-privacy in encryption schemes. It asks that the undeniable signature scheme provide privacy of the key under which the signing was performed.

An undeniable signature has the property that it cannot be verified without the signer's cooperation. Thus, we can consider the situation that there is a long term between a point of time when the signature (and the message) is issued and that when the signature (and the message) is verified. The signer might want to keep her identity private from an eavesdropping adversary until when the signature is verified. When the signature should be verified, the signer will disclose her identity to the verifier. This situation might not imply direct practical applications, however, we believe that our proposed notion of the key-privacy of signature can give a useful measure of security in a different direction from the existential forgery.

We prove that the discrete-log based scheme proposed by Michels and Stadler provides anonymity and existential unforgeability in the random oracle model assuming the Decision Diffie–Hellman problem is hard. We also propose a variant of an RSA-based scheme proposed by Gennaro, Krawczyk, and Rabin that provides anonymity and existential unforgeability in the random oracle model assuming RSA is one-way and the EDL problem is hard.

2 Key-Privacy Encryption Scheme

2.1 Definitions

In a heterogeneous public-key environment, encryption will probably fail to be anonymous for trivial reasons. For example, different users might be using different cryptosystems, or, if the same cryptosystem, have keys of different lengths. In [1], a public-key encryption scheme is defined as follows.

Definition 1 ([1]). *A public-key encryption scheme $\mathcal{PE} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ consists of four algorithms.*

- *The common-key generation algorithm $\mathcal{G}$ takes as input some security parameter k and returns some common key I.*

- *The key generation algorithm $\mathcal{K}$ is a randomized algorithm that takes as input the common key I and returns a pair (pk, sk) of keys, the public key and a matching secret key.*

- *The encryption algorithm $\mathcal{E}$ is a randomized algorithm that takes the public key pk and a plaintext x to return a ciphertext y.*

- *The decryption algorithm $\mathcal{D}$ is a deterministic algorithm that takes the secret key sk and a ciphertext y to return the corresponding plaintext x or a special symbol $\perp$ to indicate that the ciphertext was invalid.*

□

In [1], they formalize the property of "key-privacy". This can be considered under either chosen-plaintext attack or chosen-ciphertext attack, yielding two notions of security, IK-CPA and IK-CCA. (IK means "indistinguishability of keys".)

Definition 2 (IK-CPA, IK-CCA[1]). *Let $\mathcal{PE} = (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme. Let $b \in \{0,1\}$ and $k \in \mathbf{N}$. Let $A_{\rm cpa}$, $A_{\rm cca}$ be adversaries that run in two stages and where $A_{\rm cca}$ has access to the oracles $\mathcal{D}_{sk_0}(\cdot)$ and $\mathcal{D}_{sk_1}(\cdot)$. Note that* si *is the state information. It contains pk_0, pk_1, and so on. Now, we consider the following experiments:*

Experiment $\mathbf{Exp}^{\rm ik-cpa-\mathit{b}}_{\mathcal{PE},A_{\rm cpa}}(k)$	Experiment $\mathbf{Exp}^{\rm ik-cca-\mathit{b}}_{\mathcal{PE},A_{\rm cca}}(k)$
$I \stackrel{R}{\leftarrow} \mathcal{G}(k)$	$I \stackrel{R}{\leftarrow} \mathcal{G}(k)$
$(pk_0, sk_0) \stackrel{R}{\leftarrow} \mathcal{K}(I)$; $(pk_1, sk_1) \stackrel{R}{\leftarrow} \mathcal{K}(I)$	$(pk_0, sk_0) \stackrel{R}{\leftarrow} \mathcal{K}(I)$; $(pk_1, sk_1) \stackrel{R}{\leftarrow} \mathcal{K}(I)$
$(x, \mathsf{si}) \leftarrow A_{\rm cpa}(\mathsf{find}, pk_0, pk_1)$	$(x, \mathsf{si}) \leftarrow A_{\rm cca}^{\mathcal{D}_{sk_0}(\cdot),\mathcal{D}_{sk_1}(\cdot)}(\mathsf{find}, pk_0, pk_1)$
$y \leftarrow \mathcal{E}_{pk_b}(x)$	$y \leftarrow \mathcal{E}_{pk_b}(x)$
$d \leftarrow A_{\rm cpa}(\mathsf{guess}, y, \mathsf{si})$	$d \leftarrow A_{\rm cca}^{\mathcal{D}_{sk_0}(\cdot),\mathcal{D}_{sk_1}(\cdot)}(\mathsf{guess}, y, \mathsf{si})$
`return` d	`return` d

Above it is mandated that $A_{\rm cca}$ never queries $\mathcal{D}_{sk_0}(\cdot)$ and $\mathcal{D}_{sk_1}(\cdot)$ on the challenge ciphertext y. For atk $\in$ {cpa, cca} *we define the advantages via*

$$\mathbf{Adv}^{\rm ik-atk}_{\mathcal{PE},A_{\rm atk}}(k) = \left| \Pr[\mathbf{Exp}^{\rm ik-atk-1}_{\mathcal{PE},A_{\rm atk}}(k) = 1] - \Pr[\mathbf{Exp}^{\rm ik-atk-0}_{\mathcal{PE},A_{\rm atk}}(k) = 1] \right|.$$

The scheme $\mathcal{PE}$ is said to be IK-CPA secure (respectively IK-CCA secure) if the function $\mathbf{Adv}^{\rm ik-cpa}_{\mathcal{PE},A_{\rm cpa}}(\cdot)$ (resp. $\mathbf{Adv}^{\rm ik-cca}_{\mathcal{PE},A_{\rm cca}}(\cdot)$) is negligible for any adversary A whose time complexity is polynomial in k. □
The "time-complexity" is the worst case execution time of the experiment plus the size of the code of the adversary, in some fixed RAM model of computation.

2.2 RSA-RAEP

A simple observation that seems to be folklore is that standard RSA encryption does not provide key-privacy, even when all moduli in the system have the same length. Suppose an adversary knows that the ciphertext y is created under one of two keys (N_0, e_0) or (N_1, e_1), and suppose $N_0 \le N_1$. If $y \ge N_0$ then the adversary bets it was created under (N_1, e_1), else it bets it was created under (N_0, e_0). It is not hard to see that this attack has non-negligible advantage.

In [1], they propose an RSA-based encryption scheme which is secure in the sense of IK-CCA. It is RSA-RAEP which is a variant of RSA-OAEP. Their variant simply repeats the ciphertext computation, each time using new coins, until the ciphertext y satisfies $y < 2^{k-1}$, where k is the length of N.

Definition 3 (RSA-RAEP [1]). *RSA-RAEP $= (\mathcal{G}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ is as follows. The common key generator algorithm $\mathcal{G}$ takes a security parameter k and returns parameters k, k_0 and k_1 such that $k_0(k) + k_1(k) < k$ for all $k > 1$. This defines an associated plaintext-length function $n(k) = k - k_0(k) - k_1(k)$. The key generation algorithm $\mathcal{K}$ takes k, k_0, k_1, runs the key-generation algorithm of the RSA family, and get (N, e) and (N, d). The public key for the scheme pk is $(N, e), k, k_0, k_1$ and the secret key sk is $(N, d), k, k_0, k_1$ ($2^{k-1} < N < 2^k$). The other algorithms are depicted below. The oracles G and H which $\mathcal{E}_{pk}$ and $\mathcal{D}_{sk}$ reference below map bit strings as follows: $G : \{0,1\}^{k_0} \to \{0,1\}^{n+k_1}$ and $H : \{0,1\}^{n+k_1} \to \{0,1\}^{k_0}$. Note that $[x]^n$ denotes the n most significant bits of x, while $[x]_m$ denotes the*

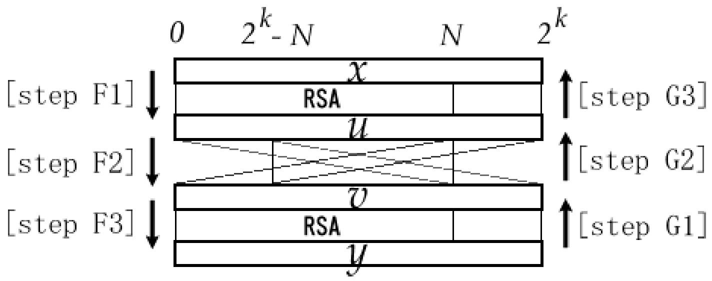

Figure 1: function $f_{N,e}$ and $g_{N,d}$

m least significant bits of x.

Algorithm $\mathcal{E}_{pk}^{G,H}(x)$
  $ctr = -1$
  repeat
    $ctr \leftarrow ctr + 1$
    $r \stackrel{R}{\leftarrow} \{0,1\}^{k_0}$
    $s \leftarrow (x \,||\, 0^{k_1}) \oplus G(r)$
    $t \leftarrow r \oplus H(s)$
    $v \leftarrow (s||t)^e \bmod N$
  until $((v < 2^{k-1}) \vee (ctr = k_1))$
  if $(ctr = k_1)$ $y \leftarrow 1||0^{k_0+k_1}||x$ **else** $y \leftarrow 0||v$
  return y

Algorithm $\mathcal{D}_{sk}^{G,H}(y)$
  $b \leftarrow [y]^1;\ v \leftarrow [y]_{k_0+k_1+n}$
  if $(b = 1)$
    $w \leftarrow [v]^{k_0+k_1};\ x \leftarrow [v]_n$
    if $(w = 0^{k_0+k_1})$ $z \leftarrow x$ **else** $z \leftarrow \perp$
  else
    $s \leftarrow [v^d]^{n+k_1};\ t \leftarrow [v^d]_{k_0}$
    $r \leftarrow t \oplus H(s)$
    $x \leftarrow [s \oplus G(r)]^n;\ p \leftarrow [s \oplus G(r)]_{k_1}$
    if $(p = 0^{k_1})$ $z \leftarrow x$ **else** $z \leftarrow \perp$
  return z

□

They proved RSA-RAEP is secure in the sense of IK-CCA in the random oracle model assuming RSA is one-way. In RSA-RAEP, the expected number of exponentiations for encryption is

$$\sum_{i=1}^{k_1} i\left(1 - \frac{2^{k-1}}{N}\right)^{i-1} \frac{2^{k-1}}{N} = \frac{1-(1-p)^{k_1}}{p} - k_1(1-p)^{k_1}$$

where $p = 2^{k-1}/N$. But in RSA-RAEP, the upper bound of the number of exponentiations for encryption is $k_1 (\gg 2)$ (security parameter). Now we propose a more efficient encryption scheme.

2.3 Our Proposed Encryption Scheme

In our scheme, we use the following function instead of the standard RSA function. We set domain and range of this function $\{x | x \in [0, 2^k) \ \wedge \ x \bmod N \in \mathbb{Z}_N^*\}$, and it is clear that $f_{N,e}^{-1} = g_{N,d}$. See Figure 1.

Function $f_{N,e}(x)$
  [step F1] **if** $(x < N)$ $u \leftarrow x^e \bmod N$ **else** $u \leftarrow x$
  [step F2] **if** $(u < 2^k - N)$ $v \leftarrow u + N$ **elseif** $(2^k - N \le u < N)$ $v \leftarrow u$ **else** $v \leftarrow u - N$
  [step F3] **if** $(v < N)$ $y \leftarrow v^e \bmod N$ **else** $y \leftarrow v$
  return y

```
Function g_{N,d}(y)
  [step G1] if (y < N) v ← y^d mod N else v ← y
  [step G2] if (v < 2^k − N) u ← v + N elseif (2^k − N ≤ v < N) u ← v else u ← v − N
  [step G3] if (u < N) x ← u^d mod N else x ← u
  return x
```

Then, we have the following theorem.

Theorem 1. *If there exists a polynomial-time algorithm which inverts $f_{N,e}$ with non-negligible probability, then there exists a polynomial-time algorithm which inverts the standard RSA function $f(X) = X^e \bmod N$ with non-negligible probability.* □

Proof. Let $P = [0, 2^k - N), Q = [2^k - N, N), R = [N, 2^k)$. We specify a inverting algorithm D for the standard RSA function using the inverting algorithm A for $f_{N,e}$ whose advantage is non-negligible.

```
          Algorithm D(N, e, Y)
            x ← A(N, e, Y)
            if (x ∈ R)
[step R]      t ← x − N
            else
              if (Y ∈ P)
[step P]        y ← Y + N; t ← A(N, e, y)
              else
[step Q]        y ← Y^e mod N; t ← A(N, e, y)
            return t
```

Now, we analyze the advantage of D. If A can invert y to get x with non-negligible provability, then

$$\Pr[u \in P]\Pr[A(y) = x|u \in P] + \Pr[u \in Q]\Pr[A(y) = x|u \in Q] + \Pr[u \in R]\Pr[A(y) = x|u \in R]$$

is non-negligible. So, for some $T \in \{P, Q, R\}$, $\Pr[u \in T]\Pr[A(y) = x|u \in T]$ is non-negligible, and for such T, $\Pr[A_T]\Pr[t = Y^d \bmod n|A_T]$ is non-negligible. (A_T denotes "[step T] is executed".) Therefore, the advantage of the above algorithm is non-negligible. □

Now, we describe our proposed scheme.

Definition 4. *$\mathcal{G}, \mathcal{K}$, and the oracles G and H are the same as RSA-RAEP. The other algorithms are described as follows. Note that the valid ciphertext y satisfies $y \in [0, 2^k)$ and $y \bmod N \in \mathbb{Z}_N^*$.*

Algorithm $\mathcal{E}_{pk}^{G,H}(x)$	Algorithm $\mathcal{D}_{sk}^{G,H}(y)$
$r \stackrel{R}{\leftarrow} \{0,1\}^{k_0}$	$s \leftarrow [g_{N,d}(v)]^{n+k_1}$; $t \leftarrow [g_{N,d}(v)]_{k_0}$
$s \leftarrow (x \,\|\, 0^{k_1}) \oplus G(r)$	$r \leftarrow t \oplus H(s)$
$t \leftarrow r \oplus H(s)$	$x \leftarrow [s \oplus G(r)]^n$; $p \leftarrow [s \oplus G(r)]_{k_1}$
$v \leftarrow f_{N,e}(s\|\|t)$	if $(p = 0^{k_1})$ $z \leftarrow x$ else $z \leftarrow \perp$
return y	return z

□

Using Theorem 1, we can prove the following theorem.

Theorem 2. *Our scheme is secure in the sense of IND-CCA2 and IK-CCA under the assumption that RSA is one-way.* □

Proof Idea. If RSA is one-way, our proposed function $f_{N,e}$ is also one-way. Fujisaki, Okamoto, Poincheval, and Stern [7] proved OAEP with one-way function is secure in the sense of IND-CCA2. So, OAEP with $f_{N,e}$ is secure in the sense of IND-CCA2 under the assumption that RSA is one-way. Noticing that the function and the domain of valid signature change, the IK-CCA proof of our scheme is similar to that of RSA-RAEP. □

In this scheme, the expected number of exponentiations in encryption is

$$1 \times \frac{2(2^k - N)}{2^k} + 2 \times \left(1 - \frac{2(2^k - N)}{2^k}\right) = \frac{N}{2^{k-1}}.$$

Suppose that N is uniformly distributed in $[2^{k-1}, 2^k)$, the expected number of our scheme is the same as RSA-RAEP. But in our scheme, the number of exponentiations for encryption is at most two.

3 Key-Privacy in Digital Signature

3.1 Definitions

Digital signatures are easily verified as authentic by anyone using the corresponding public key. This property can be advantageous for many users, but it is unsuitable for many other users. Chaum and Antwerpen provided undeniable signature which cannot be verified without the signer's cooperation [2, 3].

The validity or invalidity of an undeniable signature can be ascertained by conducting a protocol with the signer, assuming the signer participates. If a confirmation protocol Conf is used, the cooperating signer gives exponentially-high certainty to the verifier that the signature does correspond to the message and the signer's public key. If a disavowal protocol Disa is conducted, the signer gives exponentially-high certainty that the signature does not correspond to the message and the signer's public key. In both protocols a cheating signer, even with infinite computing power, has only an exponentially small chance of success and an overwhelming probability of being detected.

There are two basic security requirements on undeniable signature. The first is unforgeability, namely, without access to the private key of the signer no one should be able to produce legitimate signatures by himself. The second is non-transferability of the signature, namely, no attacker should be able to convince any other party, without the cooperation of the legitimate signer, of the validity or invalidity of a given message and signature.

But in such schemes, the verifier may decide it is someone's signature or it isn't. For example, an RSA-based undeniable signature can be distinguished in the same way as in an encryption scheme (See section 2.2).

In this section, we propose a new security requirement of undeniable signature schemes corresponding to key-privacy in encryption schemes. It asks that the undeniable signature scheme provide privacy of the key under which the signing was performed.

Similar to the Definition 1, we define the common-key generation algorithm ComKeyGen and redefine the key-generation algorithm KeyGen. The common-key generation algorithm ComKeyGen takes as input some security parameter k and returns some common key I. The key generation algorithm KeyGen is a randomized algorithm that takes as input the common key I and returns a pair (pk, sk) of keys, the public key and a matching secret key.

We consider four types of attack.

- known-message attack (NMA): An adversary has signatures for a set of messages which are known to the adversary but not chosen by him. He cannot interact in either confirmation or disavowal protocol.

- adaptive chosen-message attack (CMA): An adversary is allowed to use the signer as an oracle. He cannot interact in either confirmation or disavowal protocol.

- known-message and verification attack (NMVA): An adversary has signatures for a set of messages which are known to the adversary but not chosen by him. He can interact in a confirmation and a disavowal protocol except on the challenge.

- adaptive chosen-message and verification attack (CMVA): An adversary is allowed to use the signer as an oracle. He can interact in a confirmation and a disavowal protocol except on the challenge.

Definition 5 (IK-atk secure). *Let* $\mathcal{UDS} = \{\mathsf{ComKeyGen}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Conf}, \mathsf{Disa}\}$ *be a undeniable signature scheme. Let* $b \in \{0,1\}$ *and* $k \in \mathbb{N}$*(security parameter). Let* $A = (A_1, A_2)$ *be adversaries that run in two stages and* A *can use* $\mathcal{O}(\cdot)$*. But when* atk $\in$ {KMVA, CMVA}, A_2 *cannot interact in either a confirmation or a disavowal protocol on the pair* (m, σ)*. Now we consider the following experiments:*

$$
\begin{array}{l}
\texttt{Experiment } \mathbf{Exp}^{\mathrm{IK-atk-}b}_{\mathcal{UDS},A}(k) \\
\quad I \leftarrow \mathsf{ComKeyGen}(1^k) \\
\quad (pk_0, sk_0) \leftarrow \mathsf{KeyGen}(I) \\
\quad (pk_1, sk_1) \leftarrow \mathsf{KeyGen}(I) \\
\quad (m, \mathsf{si}) \leftarrow A_1^{\mathcal{O}(\cdot)}(pk_0, pk_1) \\
\quad \sigma \leftarrow \mathsf{Sign}_{sk_b}(m) \\
\quad d \leftarrow A_2^{\mathcal{O}(\cdot)}(m, \sigma, \mathsf{si}) \\
\quad \texttt{return } d
\end{array}
$$

where

atk	$\mathcal{O}(\cdot)$
KMA	$\mathsf{SignU}_{sk_0}, \mathsf{SignU}_{sk_1}$
CMA	$\mathsf{Sign}_{sk_0}, \mathsf{Sign}_{sk_1}$
KMVA	$\mathsf{SignU}_{sk_0}, \mathsf{SignU}_{sk_1}$, $\mathsf{Conf}_{sk_0}, \mathsf{Conf}_{sk_1}$, $\mathsf{Disa}_{sk_0}, \mathsf{Disa}_{sk_1}$
CMVA	$\mathsf{Sign}_{sk_0}, \mathsf{Sign}_{sk_1}$, $\mathsf{Conf}_{sk_0}, \mathsf{Conf}_{sk_1}$, $\mathsf{Disa}_{sk_0}, \mathsf{Disa}_{sk_1}$

Note that if adversary uses SignU*, message* m *is randomly chosen and the adversary gets* $(m, \mathsf{Sign}(m))$*. We define the advantages of the adversaries via:*

$$\mathbf{Adv}^{\mathrm{IK-atk}}_{\mathcal{UDS},A}(k) = \left| \Pr[\mathbf{Exp}^{\mathrm{IK-atk-1}}_{\mathcal{UDS},A}(k) = 1] - \Pr[\mathbf{Exp}^{\mathrm{IK-atk-0}}_{\mathcal{UDS},A}(k) = 1] \right|.$$

The scheme $\mathcal{UDS}$ *is said to be* IK-atk *secure if the function* $\mathbf{Adv}^{\mathrm{IK-atk}}_{\mathcal{UDS},A}(\cdot)$ *is negligible for any adversary* A *whose time complexity is polynomial in* k*.* $\square$

3.2 Discrete-Log Based Signature Scheme

In [4], Michels and Stadler proposed a discrete-log based (convertible) undeniable signature scheme and gave proofs for the required properties. We prove their scheme with careful key generation is IK-CMVA secure under the assumption that the DDH problem is hard. Before giving the proof, we describe their scheme with the modified key generation.

Setup

$$
\begin{array}{l|l}
\mathsf{ComKeyGen}(1^\ell) & \mathsf{KeyGen}(G, q, \mathcal{H}_\ell, \mathcal{H}_G) \\
\quad q \leftarrow \ell \text{ bits prime} & \quad g \overset{R}{\leftarrow} \{g' | \langle g' \rangle = G\} \\
\quad G \leftarrow \text{multiplicative subgroup of order } q & \quad x_1, x_2 \overset{R}{\leftarrow} \mathbb{Z}_q \\
\quad \mathcal{H}_\ell : \{0,1\}^* \times G \to \{0,1\}^\ell \text{ (Random Oracle)} & \quad y_1 \leftarrow g^{x_1};\ y_2 \leftarrow g^{x_2} \\
\quad \mathcal{H}_G : \{0,1\}^* \to G \text{ (Random Oracle)} & \quad \texttt{return } pk = (g, y_1, y_2) \text{ and} \\
\quad \texttt{return } (G, q, \mathcal{H}_\ell, \mathcal{H}_G) & \qquad\qquad sk = (x_1, x_2)
\end{array}
$$

	common input : $g, y_2, (m, (\hat{\tilde{r}}, \hat{s}))$	
Prover		Verifier
$(sk = x)$		$u, v \xleftarrow{R} \mathbb{Z}_q$
		$a \leftarrow g^u y_2^v$
	$\xleftarrow{\quad a \quad}$	
$\beta \leftarrow \mathcal{H}_G(g^{\hat{s}} y_1^{\mathcal{H}_\ell(m, \hat{\tilde{r}})})$		
$k, \tilde{k}, w \xleftarrow{R} \mathbb{Z}_q$		
$r_g \leftarrow g^k$, $r_\beta \leftarrow \beta^k$		
$\tilde{r}_g \leftarrow g^{\tilde{k}}$, $\tilde{r}_\beta \leftarrow \beta^{\tilde{k}}$		
	$\xrightarrow{r_g, r_\beta, \tilde{r}_g, \tilde{r}_\beta, w}$	
	$\xleftarrow{\quad u, v \quad}$	
`if` $(a \neq g^u y^v)$ `halt`		
`else`		
$s \leftarrow k - (v + w)x \bmod q$		
$\tilde{s} \leftarrow \tilde{k} - (v + w)k \bmod q$		
	$\xrightarrow{\quad s, \tilde{s} \quad}$	
		$\beta \leftarrow \mathcal{H}_G(g^{\hat{s}} y_1^{\mathcal{H}_\ell(m, \hat{\tilde{r}})})$
		`if` $(g^s y^{v+w} \neq r_g \vee g^{\tilde{s}} r_g^{v+w} \neq \tilde{r}_g$
		$\vee \beta^{\tilde{s}} r_\beta^{v+w} \neq \tilde{r}_\beta)$ `halt`
		`else`
		`if` $(\beta^s \hat{\tilde{r}}^{v+w} = r_\beta)$ "valid"
		`else` "invalid"

Figure 2: Confirmation and Disavowal Protocol

Generating a Signature

$\mathsf{Sign}_{sk}(m)$
  $k \xleftarrow{R} \mathbb{Z}_q$
  $r \leftarrow g^k$
  $\tilde{r} \leftarrow \Big(\mathcal{H}_G(r)\Big)^{x_2}$
  $c \leftarrow \mathcal{H}_\ell(m, \tilde{r})$
  $s \leftarrow k - cx_1 \bmod q$
  `return` $(\tilde{r}, s)$

Confirmation and Disavowal Protocol

A signature can be verified or denied by interactively proving the equality or inequality of the discrete logarithms of $\tilde{r}$ and y_2 to the bases $\mathcal{H}_G(g^s y_1^{\mathcal{H}_\ell(m, \tilde{r})})$ and g. See Figure 2.

Theorem 3 ([4]). *This protocol is complete, sound, zero-knowledge under the assumption that there exists no algorithm running in the expected polynomial time which decides with non-negligible probability better than guessing whether two discrete logarithms are equal. And this undeniable signature satisfies existential unforgeability and non-transferability under the assumption that the DDH problem is hard in G.* $\square$

Now, we prove the following theorem.

Theorem 4. *This undeniable signature is IK-CMVA secure assuming that the DDH problem is hard in* G. $\Box$

Proof. Suppose that A is a polynomial-time algorithm such that $\mathbf{Adv}^{\mathrm{IK-CMVA}}_{\mathcal{UDS},A}(k)$ is non-negligible. We first construct a distinguisher D for the DDH problem by using A.

`Algorithm` $D(g, X, Y, T)$
  $t_0, t_1, t_2 \stackrel{R}{\leftarrow} \mathbb{Z}_q$
  $g_0 \leftarrow u \cdot g;\ g_1 \leftarrow v \cdot g^{t_0}$
  $x_{10}, x_{11} \stackrel{R}{\leftarrow} \mathbb{Z}_q$
  $y_{10} \leftarrow g_0^{x_{10}};\ y_{11} \leftarrow g_1^{x_{11}}$
  $y_{20} \leftarrow X;\ y_{21} \leftarrow X \cdot g^{t_1}$

  $\hat{g_0} \leftarrow Y;\ \hat{g_1} \leftarrow Y^{t_0} \cdot g^{t_0 t_2}$
  $\hat{y_{20}} \leftarrow T;\ \hat{y_{21}} \leftarrow T \cdot X^{t_2} \cdot Y^{t_1} \cdot g^{t_1 t_2}$

  $pk_0 = (g_0, y_{10}, y_{20});\ pk_1 = (g_1, y_{11}, y_{21})$
  $sk_0 = (x_{20});\ sk_1 = (x_{21})$

  $(m, \mathsf{si}) \leftarrow A_1^{\mathsf{Sign}_{sk_0}, \mathsf{Sign}_{sk_1}}(pk_0, pk_1)$
  $b \stackrel{R}{\leftarrow} \{0, 1\}$
  $(\tilde{r}, s) \leftarrow \hat{\mathsf{Sign}}_{sk_b}(m)$
  $d \leftarrow A_2^{\mathsf{Sign}_{sk_0}, \mathsf{Sign}_{sk_1}}(m, (\tilde{r}, s), \mathsf{si})$

  `if` $(b = d)$ `return` 1
  `else return` 0

Simulations We also construct the three simulators as follows.

Random Oracle. Distinguisher D uses H_G-List and H_ℓ-List. If the adversary makes a fresh query q to $\mathcal{H}_G$ (resp. $\mathcal{H}_\ell$), D returns the adversary with a random string $h \in G$ (resp. $h \in \{0,1\}^\ell$), and adds (q, h) to the H_G-List (resp. H_ℓ-List).

Signing Oracle. Distinguisher D uses S_0-List, S_1-List, H_G-List, and H_ℓ-List. For $i \in \{0, 1\}$, the simulation of Sign_{sk_i} is as follows.

$\mathsf{Sign}_{sk_i}(m)$
  $k, t \stackrel{R}{\leftarrow} \mathbb{Z}_q$
  $r \leftarrow g_i^k$
  $r' \leftarrow (g_i)^t$
  add (r, r') to H_G-List
  $\tilde{r} \leftarrow (y_{2i})^t$
  $c \stackrel{R}{\leftarrow} \{0, 1\}^\ell$
  add $((m, \tilde{r}), c)$ to H_ℓ-List
  $s \leftarrow k - cx_{1i} \bmod q$
  add $(m, (\tilde{r}, s))$ to S_i-List
  `return` $(\tilde{r}, s)$

For $i \in \{0,1\}$, when D runs $\hat{\mathsf{Sign}}_{sk_i}$, D uses $\hat{g_i}$ and $\hat{y_{2i}}$ instead of g_i and y_{2i}.

Confirmation and Disavowal Protocol. We describe the simulator of the confirmation and disavowal protocol with pk_i.

1. The adversary sends a.
2. The simulator chooses $s, \tilde{s}, w$, and $c \overset{R}{\leftarrow} \mathbb{Z}_q$, and computes $\beta \leftarrow \mathcal{H}_G(g^s y_1^{\mathcal{H}_\ell(m,\tilde{r})})$, $r_g \leftarrow g^s y^c$, $\tilde{r}_g \leftarrow g^{\tilde{s}} r_g^c$, $\tilde{r}_\beta \leftarrow \beta^{\tilde{s}} r_\beta^c$. If $(m,(\tilde{r},s)) \in S_i$-List, then $r_\beta \leftarrow \beta^s z^c$, otherwise $r_\beta \overset{R}{\leftarrow} G\backslash\{\beta^s z^c\}$. The simulator sends r_g, r_β, $\tilde{r}_g$, $\tilde{r}_\beta$, and w.
3. The adversary sends u and v.
4. If $a \neq g^u y^v$, the simulator halts. Otherwise, the simulator rewinds to step 2 and sets to $c = v + w$ and recomputes $s, \tilde{s}, w, r_g, \tilde{r}_g, \tilde{r}_\beta$, and r_β. The simulator sends r_g, r_β, $\tilde{r}_g$, $\tilde{r}_\beta$, and w.
5. The adversary sends u and v.
6. If $a \neq g^u y^v$, the simulator halts. Otherwise, the simulator sends s and $\tilde{s}$.

Analysis Now, we analyse the above construction. If (g, X, Y, T) is a Diffie-Hellman 4-tuples, the input X, Y, T satisfy $X = g^u, Y = g^v, T = g^{uv}$ for some $u, v \in \mathbb{Z}_q$. $(g_0, y_{10}, y_{20}, g_1, y_{11}, y_{21})$ has the proper distribution of public keys for the original system, and the simulator's output of the signing oracle and the confirmation and disavowal protocol also has the proper distribution. Since $\log_{g_0} y_{20} = \log_{\hat{g_0}} \hat{y_{20}} = u$ and $\log_{g_1} y_{21} = \log_{\hat{g_1}} \hat{y_{21}} = (u + t_1)t_0^{-1}$, the challenge signature is distributed exactly like the original generation of signature of m under secret key sk_b. So, D outputs 1 with 1/2 plus non-negligible probability.

If (g, X, Y, T) is a non-Diffie-Hellman 4-tuples, $(g_0, y_{10}, y_{20}, g_1, y_{11}, y_{21})$ has the proper distribution of public keys for the original system, and the simulator's output of the signing oracle and the confirmation and disavowal protocol also has the proper distribution. But the challenge signature $(\tilde{r}, s)$, c, and r' are random elements and are independent of anything else. This implies that the challenge signature gives A no information about b in an information-theoretic sense. So, D outputs 1 with 1/2 plus negligible probability. □

3.3 RSA-Based Signature Scheme

In [5], Gennaro, Krawczyk, and Rabin proposed an RSA-based (convertible) undeniable signature scheme and gave proofs for the required properties. Their scheme provides existential unforgeability by using the PSS scheme[6]. But their scheme does not provide anonymity with the same reason as RSA-OAEP does not provide anonymity. We describe their scheme with the modification, and prove the modified scheme is secure in the sense of IK-CMVA under the EDL assumption.

Common input : $(n, w, S_w), (m, (\hat{S}_m, \hat{r}))$

Prover		Verifier
$(sk = (e, d))$		$i, j \xleftarrow{R} \mathbb{Z}_n;\ \ Q \leftarrow (\hat{S}_m)^{2i}(S_w)^j \bmod n$
	$\xleftarrow{\ Q\ }$	
$A \leftarrow Q^e \bmod n$		
	$\xrightarrow{BC(A)}$	
	$\xleftarrow{\ i, j\ }$	
`if` $(Q \neq (\hat{S}_m)^{2i}(S_w)^j \bmod n)$ `halt`		
	$\xrightarrow{\ A\ }$	
		$\bar{m} \leftarrow \mathsf{Sign2}(m, \hat{r})$
		`if` $(A = \bar{m}^{2i} w^j \bmod n)$ "valid"
		`else` "undetermined"

Figure 3: Confirmation Protocol

Setup

$\mathsf{ComKeyGen}(1^k)$
- $k = k_0 + k_1 + 1$
- $h : \{0,1\}^* \to \{0,1\}^{k_1}$ (Random Oracle)
- $g : \{0,1\}^{k_1} \to \{0,1\}^{k-k_1-1}$ (Random Oracle)
- $(g_1(w) = [g(w)]^{k_0}$ and $g_2(w) = [g(w)]_{k-k_0-k_1-1})$
- $\mathcal{N}_k = \{n | n = pq,\ 2^{k-1} \leq n < 2^k,\ p = 2p' + 1,$ $q = 2q' + 1,$ and $p, q, p', q'$are prime$\}$
- `return` $(\mathcal{N}_k, k, k_0, k_1)$

$\mathsf{KeyGen}(\mathcal{N}_k, k, k_0, k_1)$
- $n \leftarrow \mathcal{N}_k$
- $e \xleftarrow{R} \mathbb{Z}_{\phi(n)}$
- $d \leftarrow e^{-1}$
- $w \xleftarrow{R} \{a | a \in \mathbb{Z}_n^* \wedge ord(a) \geq p'q' \wedge a^d < 2^{k-1}\}$
- $S_w \leftarrow w^d \bmod n$
- `return` $pk = (n, w, S_w)$ and $sk = (e, d)$

Generating a Signature

```
Sign(m)
  ctr ← −1
  repeat
    ctr ← ctr + 1
    r ←R {0,1}^k0
    m̄ ← Sign2(m, r)
    S_m ← m̄^d mod n
  until (S_m < 2^(k−1) ∨ ctr = k1)
  return (S_m, r)
```

```
Sign2(m, r)
  w ← h(m||r)
  r* ← g1(w) ⊕ r
  m̄ ← 0||w||r*||g2(w)
  return m̄
```

Confirmation and Disavowal Protocols

We show the confirmation protocol in Figure 3 and the disavowal protocol in Figure 4.

Theorem 5 ([5]). *These confirmation and disavowal protocols are complete, sound, zero-knowledge under the assumption that RSA is one-way. And this undeniable signature satisfies existential unforgeability and non-transferability under the assumption that RSA is one-way and the EDL assumption.* □

Common input : $(n, w, S_w), (m, (\hat{S}_m, \hat{r}))$

$\boxed{\text{Prover}}$		$\boxed{\text{Verifier}}$
$(sk = (e, d))$		$b \stackrel{R}{\leftarrow} \mathbb{Z}_\ell,\ i = 4b,\ j \stackrel{R}{\leftarrow} \mathbb{Z}_n$
		$\bar{m} \leftarrow \mathsf{Sign2}(m, \hat{r})$
		$Q_1 \leftarrow (\bar{m})^i w^j \bmod n$
		$Q_2 \leftarrow (\hat{S}_m)^i (S_w)^j \bmod n$
	$\xleftarrow{(Q_1, Q_2)}$	
$\bar{m} \leftarrow \mathsf{Sign2}(m, \hat{r})$		
`find` $k = 4b$ $(b \in \mathbb{Z}_\ell)$ `s.t.` $Q_1/Q_2^e = \left(\frac{\bar{m}}{(\hat{S}_m)^e}\right)^k$		
`if` (k is found uniquely) $A \leftarrow k$ `else` $A \leftarrow 0$		
	$\xrightarrow{BC(A)}$	
	$\xleftarrow{i, j}$	
`if` $(Q_1 \neq (\bar{m})^i w^j \bmod n \vee$ $Q_2 \neq (\hat{S}_m)^i (S_w)^j \bmod n)$ `halt`		
	$\xrightarrow{A}$	
		`if` $(A = i)$ "invalid"
		`else` "undetermined"

Figure 4: Disavowal Protocol

Remark [EDL assumption [5]] Let $T \stackrel{R}{\leftarrow} \{z' | z' \in \mathbb{Z}_n^* \wedge z' < 2^{k-1}\}$. For values n, w, S_w and $\bar{m}$ as defined above it is infeasible to decide the validity of $\log_{\bar{m}} T = \log_w S_w$ over $\mathbb{Z}_n$. We define the EDL problem as follows. This EDL problem is considered to be hard, and we might be able to assume this.

Definition 6 (EDL problem). *Let* $(\mathcal{N}_k, k, k_0, k_1) \leftarrow \mathsf{ComKeyGen}(1^k)$ *and* $((n, w, S_w), (e, d)) \leftarrow \mathsf{KeyGen}(\mathcal{N}_k, k, k_0, k_1)$*. We define two distributions as follows.*

$$\begin{aligned}
\mathbf{X} : & \ m \stackrel{R}{\leftarrow} \{0,1\}^*;\ (S_m, r) \leftarrow \mathsf{Sign}_{sk}(m);\ \bar{m} \leftarrow \mathsf{Sign2}(m, r); \\
& \ \mathtt{return}((\mathcal{N}_k, k, k_0, k_1), (n, w, S_w), \bar{m}, S_m) \\
\mathbf{Y} : & \ m \stackrel{R}{\leftarrow} \{0,1\}^*;\ r \stackrel{R}{\leftarrow} \{0,1\}^{k_0};\ \bar{m} \leftarrow \mathsf{Sign2}(m, r);\ z \stackrel{R}{\leftarrow} \{z' | z' \in \mathbb{Z}_n^* \wedge z' < 2^{k-1}\}; \\
& \ \mathtt{return}((\mathcal{N}_k, k, k_0, k_1), (n, w, S_w), \bar{m}, z)
\end{aligned}$$

We say that the EDL problem is hard if

$$|\Pr[D(\mathbf{X}) = 1] - \Pr[D(\mathbf{Y}) = 1]|$$

is negligible for any algorithm D whose time complexity is polynomial in k. □

We can prove the following theorem.

Theorem 6. *This undeniable signature is IK-CMVA secure assuming RSA is one-way and the EDL problem is hard.* □

Proof. The proof is similar to that of the discrete-log based scheme in section 3.2. We can construct a distinguisher for breaking the EDL assumption. See Appendix A. □

4 Conclusion

In this paper, we propose a new security requirement of undeniable signature schemes corresponding to key-privacy in encryption schemes. It asks that the undeniable signature scheme provide privacy of the key under which the sign was performed. We prove that the discrete-log based scheme proposed by Michels and Stadler provides anonymity and existential unforgeability in the random oracle model assuming the Decision Diffie–Hellman problem is hard. We also propose a variant of an RSA-based scheme proposed by Gennaro, Krawczyk, and Rabin that provides anonymity and existential unforgeability in the random oracle model assuming RSA is one-way and the EDL problem is hard. It should be noticed that it does not seem to be easy to make a more efficient RSA-based signature scheme like as the encryption scheme in section 2.3.

Acknowledgements

We thank Tatsuaki Okamoto for helpful discussion and variable comments.

References

[1] M. Bellare, A. Boldyreva, A. Desai and D. Pointcheval. Key-Privacy in Public-Key Encryption. Proc. Asiacrypt 2001, pp. 566-582.

[2] D. Chaum and H. V. Antwerpen. Undeniable Signatures. Advance is Cryptology - Crypto '89, LNCS 435, Springer-Verlag, 1989, pp. 212-217.

[3] D. Chaum. Zelo-knowledge Undeniable Signatures. Proc. Eurocrypt '90, LNCS 473, Springer-Verlag, 1991, pp. 458-464.

[4] M. Michels and M. Stadler. Efficient Convertible Undeniable Signature Schemes. In Proc. 4th Workshop on Selected Areas in Cryptography (SAC'97), Ottawa, Canada, 1997, pp. 231-244.

[5] R. Gennaro, H. Krawczyk, and T. Rabin. RSA-based undeniable signatures. CRYPTO '97, LNCS 1294, Springer-Verlag, 1997, pp.132-149.

[6] M. Bellare and P. Rogaway. The exact security of digital signatures: How to sign with RSA and Rabin. Advances in Cryptology - Eurocrypt 96, LNCS 1070, Springer-Verlag, 1996, pp. 339-416.

[7] E. Fujisaki, T. Okamoto, D. Pointcheval, and J. Stern, RSA-OAEP is Secure under the RSA Assumption, Advances in Cryptology - Proceedings of CRYPTO '01, LNCS 2139, Springer-Verlag, 2001, pp. 260-274.

A Proof of Theorem 6

Suppose that A is a polynomial-time algorithm such that $\mathbf{Adv}_{\mathcal{UDS},A}^{\mathrm{IK-CMVA}}(k)$ is non-negligible. We construct a distinguisher D for breaking the EDL assumption by using A.

```
Algorithm D((N_k, k, k_0, k_1), (n, w, S_w), t, T)
  b <-R- {0,1}
  pk_b = (n_b, w_b, S_wb) <- (n, w, S_w)
  (pk_{1-b}, sk_{1-b}) <- KeyGen(N_k, k, k_0, k_1)

  ŵ_b <- t;  Ŝ_wb <- T

  (m, si) <- A_1^{Sign_{sk_0}, Sign_{sk_1}}(pk_0, pk_1)
  (r̃, s) <- Ŝign_{sk_b}(m)
  d <- A_2^{Sign_{sk_0}, Sign_{sk_1}}(m, (r̃, s), si)

  if (b = d) return 1
  else return 0
```

Simulations We also construct the four simulators as follows.

Random Oracle. Distinguisher D uses g-List and h-List. If the adversary makes a fresh query q to g (resp. h), D returns the adversary with a random string $a \in \{0,1\}^{k-k_1-1}$ (resp. $a \in \{0,1\}^{k_1}$), and adds (q, a) to the g-List (resp. h-List).

Signing Oracle. Because distinguisher D has sk_{1-b}, D can simulate $\mathsf{Sign}_{sk_{1-b}}$ perfectly. So, we only describe the simulator of Sign_{sk_b}. D uses S_b-List, g-List, and h-List.

```
Sign_{sk_b}(m)
  r <-R- {0,1}^{k_0}
  ctr <- -1
  repeat
    ctr <- ctr + 1
    t <- Z_{n_b}
    S_m <- (S_wb)^t mod n_b
    m̄ <- (w_b)^t mod n_b
  until { (S_m ∈ Z*_{n_b} ∧ S_m < 2^{k-1} ∧ [m̄]^1 = 0) ∨ ctr = k_1 }
  α <- [m̄]_{k-1};
  w <- [α]^{k_1};  β <- [α]_{k-k_1-1}
  γ* <- [β]^{k_0};  γ <- [β]_{k-k_0-k_1-1}
  add (w, (r ⊕ γ*)||γ) to g-List
  add (m||r, w) to h-List
  add (m, (S_m, r)) to h-List
  return (S_m, r)
```

When D runs $\hat{\mathsf{Sign}}_{sk_b}$, D uses $\hat{w}_b$ and $\hat{S_{wb}}$ instead of w_b and S_{wb}.

Confirmation Protocol. Because D has sk_{1-b}, D can simulate $\mathsf{Conf}_{sk_{1-b}}$ perfectly. So, we only describe the simulator of the confirmation protocol with pk_b.

1. The adversary sends Q.
2. The simulator chooses $A \xleftarrow{R} \mathbb{Z}_{n_b}$, and sends $BC(A)$.
3. The adversary sends i and j.

4. If $Q \neq (\hat{S_m})^{2i}(S_{wb})^j \bmod n_b$, the simulator halts. Otherwise, the simulator rewinds to step 2, and if $(m, (\hat{S_m}, \hat{r})) \in S_b$-List, then $A \leftarrow \bar{m}^{2i}(w_b)^j \bmod n_b$, otherwise $A \xleftarrow{R} \mathbb{Z}_{n_b} \backslash \{\bar{m}^{2i}(w_b)^j \bmod n_b\}$. The simulator sends $BC(A)$.

5. The adversary sends i and j.

6. If $Q \neq (\hat{S_m})^{2i}(S_{wb})^j \bmod n_b$, the simulator halts. Otherwise, the simulator sends A.

Disavowal Protocol. Because D has sk_{1-b}, D can simulate $\mathsf{Disa}_{sk_{1-b}}$ perfectly. So, we only describe the simulator of the disavowal protocol with pk_b.

1. The adversary sends (Q_1, Q_2).

2. The simulator chooses $A \xleftarrow{R} \mathbb{Z}_{n_b}$, and sends $BC(A)$.

3. The adversary sends i and j.

4. If $Q_1 \neq \bar{m}^i(w_b)^j \bmod n_b$ or $Q_2 \neq (\hat{S_m})^i(S_{wb})^j \bmod n_b$, the simulator halts. Otherwise, the simulator rewinds to step 2, and if $(m, (\hat{S_m}, \hat{r})) \notin S_b$-List, then ,$A \leftarrow i$, otherwise , $A \xleftarrow{R} \{k | k = 4b,\ b \in \mathbb{Z}_\ell\} \backslash \{i\}$. The simulator sends $BC(A)$.

5. The adversary sends i and j.

6. If $Q_1 \neq \bar{m}^i(w_b)^j \bmod n_b$ or $Q_2 \neq (\hat{S_m})^i(S_{wb})^j \bmod n_b$, the simulator halts. Otherwise, the simulator sends A.

Analysis Now, we analyse the above construction. If $((\mathcal{N}_k, k, k_0, k_1), (n, w, S_w), t, T) \in \mathbf{X}$, the input w, S_w, t, T satisfy $\log_w S_w = \log_t T = d$ over $\mathbb{Z}_n$. $pk_b = (n, w, S_w)$ has the proper distribution of public keys for the original system, and the simulator's output of the signing oracle and the confirmation and disavowal protocol also has the proper distribution. Since $\log_w S_w = \log_t T = \log_{\hat{w}_b} \hat{S_{wb}} = d$, the challenge signature is distributed exactly like the original generation of signature of m under secret key sk_b. So, D outputs 1 with 1/2 plus non-negligible probability.

If $((\mathcal{N}_k, k, k_0, k_1), (n, w, S_w), t, T) \in \mathbf{Y}$, $pk_b = (n, w, S_w)$ has the proper distribution of public keys for the original system, and the simulator's output of the signing oracle and the confirmation and disavowal protocol also has the proper distribution. But the challenge signature S_m and r are random elements and are independent of anything else. This implies that the challenge signature gives A no information about b in an information-theoretic sense. So, D outputs 1 with 1/2 plus negligible probability.