

A Secure Obfuscator for Encrypted Blind Signature Functionality*

Xiao Feng^{1,2} Zheng Yuan^{1,2**}

¹ Beijing Electronic Science & Technology Institute, Beijing 100070, P. R. China
zyuan@tsinghua.edu.cn and yuanzheng@besti.edu.cn

² School of Telecommunications Engineering, Xidian University, Shaanxi 710071, P. R. China

Abstract. This paper introduces a new obfuscation called obfuscation of encrypted blind signature. Informally, encrypted blind signature enable the message is blind to signer, and she couldn't distinguish two encrypted signatures by herself. A obfuscation of encrypted blind signature makes the process of encrypted blind signature unintelligible for any third party, while still keeps the original encrypted blind signature functionality. We use schnorr's blind signature scheme and linear encryption scheme as blocks to construct a new obfuscator. Moreover, we propose two new security definition: blindness w.r.t encrypted blind signature (EBS) obfuscator and one-more unforgeability(OMU) w.r.t EBS obfuscator, and prove them based on Decision Liner Diffie-Hellman(DL) assumption and the hardness of discrete logarithm, respectively. We also demonstrate that our obfuscator satisfies the Average-Case Virtual Black-Box Property(ACVBP) property w.r.t dependent oracle, it is indistinguishable secure. Our paper expand a new direction for the application of obfuscation.

Keywords: Obfuscation, Blind signature, Indistinguishable security.

1 Introduction

Obfuscation in cryptography has been formally proposed by Barak, Goldreich et al.[1] at the first time. Although it has been a theoretically hot area, there isn't much more progress in the past few years. The implementation of obfuscation mainly depends on how to construct a secure obfuscator. Informally, obfuscator is an algorithm program that transforms a program into a new unintelligible program while its functionality remains the same. Barak et al. suggested that an obfuscator ought to satisfy the following three property:

1. **Functionality:** the obfuscated program has the same functionality as the original program.
2. **Polynomial Slowdown:** the description length and running time of the obfuscated program are at most polynomially larger than the original program's.

* This work is supported by the National Natural Science Foundation of China (No.61070250), and Beijing Natural Science Foundation (N0.4132066), and the 12th Five-year Cryptography Development Foundation of China (No.MMJJ201101026), and Scientific Research and Post-graduate Training Cooperation Project-Scientific Research Base-New Theory of Block Cipher and Obfuscation and their Application Research.

** To whom correspondence should be addressed.

3. Virtual Black-Box Property(VBP): Anything that can be efficiently computed from obfuscated program can be efficiently computed given oracle access to the original program.

Obfuscation has profound effects on both theory and application, such as software protection, homomorphic encryption, removing random oracles and transforming private-key encryption into public-key encryption. Despite all that, Barak et al. has prove the impossibility of obfuscation even under a very weak definition. Later, more impossible obfuscation results of natural functionalities were shown in [2][3][4][5]. Even so, cryptographic community has been dedicating to conducting a series of exploration, and they found that there still exist simple classes of functions such as point functions[6][7][8][2][3][9] with the possibility of obfuscation.

Before 2007, several positive results of obfuscation were mainly about simple functions. The first obfuscation of complicated cryptographic functionality was proposed by Hohenberger et al.[10] in TCC'07. They obfuscate re-encryption and proved the security of obfuscator in the standard model. In brief, the re-encryption functionality is the one that takes a ciphertext for a message encrypted under Alice's public key and transforms it into a ciphertext for the same message under Bob's public key. Hohenberger et al. presented an improved security property called ACVBP. Following the security definition of ACVBP[10], Hada [11] showed a secure obfuscation for encrypted signature, which generates a signature on a given message under Alice's secret signing key and then encrypts the signature under Bob's public encryption key. Later, on the basis of Hohenberger's results, Nishanth Chandran et al. [12] refined the delegation of access of re-encryption functionality, demonstrating the security of collusion-resistant obfuscation. These are the only known three obfuscations of complicated cryptographic functionality.

Blind signature has a wide range of applications in e-cash and electronic election. A blind signature is a protocol introduced by Chaum [13] for protecting the anonymity of signer, which was based on the RSA digital signature scheme. Unlike general digital signature scheme, blind signature requires that signer signs the message without knowing the message or the resulting signatures while the user could verify it publicly, it's a interactive protocol between signer and user. A blind signature must satisfies the following property:

1. Unforgeability: Adversary can not produce a legal blind signature on message after interacting with signer.
2. Blindness: The signatures of two given messages are computationally indistinguishable even under a set of known message-signature pairs.

Later, Okamoto[14] put forward to Schorr's blind signature scheme, which is based on discrete logarithm problem. Schnorr[15] then proved its security.

In this paper, we first use Schnorr's blind signature scheme and linear encryption scheme[16] as blocks to construct a secure obfuscator for blind signature, which is complete and verifiable. In order to prove the security of the obfuscator, we propose two new security definition, Blindness w.r.t encrypted blind signature(EBS) obfuscator and one-more unforgeability (EBS) obfuscator, to prove our Theorem 5. Our main

method is to construct different adversaries to break the hardness assumption under security definition of ACVBP w.r.t dependent oracle, the scheme is insecure if any of the adversaries is successful. The specific refers to section 5. We also prove that the OMU w.r.t EBS functionality implies OMU w.r.t EBS obfuscator under the assumption that EBS obfuscator satisfies ACVBP w.r.t dependent oracle set. Obviously, we have OMU w.r.t EBS obfuscator. At last, we present the security proof of EBS obfuscator. i.e., the EBS obfuscator satisfies ACVBP w.r.t dependent oracle. Thus, we prove that, under the ACVBP w.r.t dependent oracle, generating a blind signature on a message and then encrypting the signature is functionally equivalent to encrypting the sign key and then generating a blind signature on the message.

The article structure is as follows: Section 2 is preliminaries which contain three parts; We propose new security definitions which is the basis of theorem's proof in section 3; We construct the secure obfuscator for special EBS functionality in section 4 and prove its security in section 5.

2 Preliminaries

In this section, we present the basic definition of security and the hardness assumption that our proofs rely on.

2.1 Bilinear Maps

Set $BMsetup$ be an initialization algorithm: on input security parameter 1^k , outputs the bilinear map parameters as $(q, g, \mathbb{G}, \mathbb{G}_T, \mathbf{e})$, where \mathbb{G}, \mathbb{G}_T are groups of prime order $q \in \Theta(2^k)$, g is a generator of \mathbb{G} and \mathbf{e} is an efficient bilinear mapping from $\mathbb{G} \times \mathbb{G}$ to \mathbb{G}_T . The mapping \mathbf{e} satisfies the following two property:

- Bilinear: For all $g \in \mathbb{G}$ and $a, b \in \mathbb{Z}_q$, $\mathbf{e}(g^a, g^b) = \mathbf{e}(g, g)^{ab}$.
- Non-degenerate: If g generates \mathbb{G} , then $\mathbf{e}(g^a, g^b) \neq 1$.

2.2 Complexity Assumptions

Definition 1. (DL Assumption) For every PPT machine D , every polynomial $p(\cdot)$, all sufficiently large $n \in \mathbb{N}$, and every $z \in \{0, 1\}^{poly(n)}$,

$$\left| \Pr \left[\begin{array}{l} p = (q, \mathbb{G}, \mathbb{G}_T, \mathbf{e}, g) \leftarrow Setup(1^n); \\ a \leftarrow \mathbb{Z}_q; b \leftarrow \mathbb{Z}_q; r \leftarrow \mathbb{Z}_q; s \leftarrow \mathbb{Z}_q; \\ decision \leftarrow D(p, (g^a, g^b), (g^{r+s}, (g^a)^r, (g^b)^s), z). \end{array} \right] - \Pr \left[\begin{array}{l} p = (q, \mathbb{G}, \mathbb{G}_T, \mathbf{e}, g) \leftarrow Setup(1^n); \\ a \leftarrow \mathbb{Z}_q; b \leftarrow \mathbb{Z}_q; r \leftarrow \mathbb{Z}_q; s \leftarrow \mathbb{Z}_q; t \leftarrow \mathbb{Z}_q; \\ decision \leftarrow D(p, (g^a, g^b), (g^t, (g^a)^r, (g^b)^s), z). \end{array} \right] \right| \leq \frac{1}{p(n)}$$

2.3 The Definition of General Security

In this subsection we review the security definition of public-key encryption(PKE) scheme and digital blind signature(DBS) scheme. *Setup* is an algorithm that generates a parameter, on security parameter 1^n , which is used commonly by multiple users in a pair of *PKE* and *DBS* schemes.

A probabilistic public key cryptosystem *PKE* is a probabilistic polynomial time Turing machine *II* that

(1)*EKG*: on inputs p generates a pair of public-secret key (pk, sk) and outputs the description of two algorithms, *E* and *D* such that

(2)*E* is a probabilistic encryption algorithm: for some constants p , public key pk and a plaintext m , returns the ciphertext c , let $MS(p, pk)$ be the message space defined by (p, pk) .

(3)*D* is a deterministic decryption algorithm: for some constants p , secret key sk and ciphertext c , returns the plaintext m .

Definition 2. (*Indistinguishability of Encryptions against CPAs*) A *PKE* scheme (EKG, E, D) satisfies the indistinguishability if the following condition holds: For every PPT machine pair (A_1, A_2) (adversary), every polynomial $p(\cdot)$, all sufficiently large $n \in \mathbb{N}$, and every $z \in \{0, 1\}^{\text{poly}(n)}$,

$$2 \cdot \Pr \left[\begin{array}{l} p \leftarrow \text{Setup}(1^n); (pk, sk) \leftarrow EKG(p); \\ (m_1, m_2, h) \leftarrow A_1(p, pk, z); b \leftarrow \{0, 1\}; c \leftarrow E(p, pk, m_b); \\ d \leftarrow A_2(p, pk, (m_1, m_2, h), c, z); \\ b = d. \end{array} \right] - 1 \leq \frac{1}{p(n)}$$

where we assume that A_1 produces a valid message pair m_1 and $m_2 \in MS(p, pk)$ and a hints h .

A blind digital signature *DBS* also contains three algorithms:

(1)*SKG*: on inputs p generates a pair of public-secret key (pk, sk) .

(2) (S, U) is a probabilistic interactive signing algorithm: for some constants p , secret key sk and l -bit plaintext $m = m_1 m_2 \cdots m_l \in MS(p, pk)$, the execution of algorithm $S(sk)$ (by signer), and algorithm $U(pk, m)$ (by user) for message m generates the signature σ , where $MS(p, pk)$ is the message space defined by (p, pk) .

(3)*V* is a deterministic verification algorithm: for some constants p , public key pk , message m and signature σ , if σ is the valid signature of m , it accept; Otherwise returns \perp .

Security of a blind signature scheme include one-more unforgeability and blindness.

Definition 3. (*Blindness*) A blind signature scheme $DBS = (SKG, (S, U), V)$ is called blind if for any efficient algorithm A_3 , all sufficiently large $n \in \mathbb{N}$, and every $z \in \{0, 1\}^{\text{poly}(n)}$, there exist

$$2 \cdot \Pr \left[\begin{array}{l} p \leftarrow \text{Setup}(1^n); (pk, sk) \leftarrow SKG(p); \\ b \leftarrow \{0, 1\}; (\sigma_0, \sigma_1) \leftarrow A_3^{\langle \cdot, U(pk, m_b) \rangle^1, \langle \cdot, U(pk, m_{1-b}) \rangle^1}(p, pk, z); \\ b^* \leftarrow A_3(\sigma_0, \sigma_1); \\ b = b^*. \end{array} \right] - 1 \leq \frac{1}{p(n)}$$

where A_3 is the malicious Signer and U is the honest user. If $\sigma_0 = \perp$ or $\sigma_1 = \perp$, then the Signer is not informed about other signature either.

Note that we use $X^{\langle \cdot, Y(y_0) \rangle^1, \langle \cdot, Y(y_1) \rangle^1}$ to define the process that X invoke arbitrarily ordered executions with $Y(y_0)$ and $Y(y_1)$, but interact with each algorithm only once.

Definition 4. (One-more Unforgeability) A DBS scheme $(SKG, (S, U), V)$ is unforgeable if for any efficient algorithm A_4 (the malicious user), every polynomial $p(\cdot)$, all sufficiently large $n \in \mathbb{N}$, and every $z \in \{0, 1\}^{\text{poly}(n)}$, there exist

$$\Pr \left[\begin{array}{l} p \leftarrow \text{Setup}(1^n); (pk, sk) \leftarrow SKG(p); \\ ((m_1^*, \sigma_1^*), \dots, (m_{k+1}^*, \sigma_{k+1}^*)) \leftarrow A_4^{\ll S_{p,sk} \gg^k}(p, pk, z); \\ \text{if } m_i^* \neq m_j^* \text{ for } i \neq j; \\ V(p, pk, m_i^*, \sigma_i^*) = \text{Accept for all } i; \text{ then return } 1. \end{array} \right] \leq \frac{1}{p(n)}$$

where $S_{p,sk}$ is the signing oracle (circuit).

Note that we use $X^{\ll Y \gg^k}$ to define the process that X sample access to Y for at most k times.

3 Construct the Secure Obfuscator for Special EBS Functionality

This section presents a secure obfuscator for the blind signature and prove the security based on the generalized ACVBP definition.

3.1 Schnorr's Blind Signature

We use Schnorr's blind signature scheme[14] as a block to build the EBS functionality. Specific process is as follows:

$SKG(p)$

1. Parse $p = (q, \mathbb{G}, \mathbb{G}_T, \mathbf{e}, g)$.
2. Select $g_1 \in \mathbb{G}$ and $x \in \mathbb{Z}_q$ randomly.
3. Output the secret key $sk = g_1^x$ and public key $pk = (g_1, g^{g_1^x})$, where $y = g^{g_1^x}$.

$Sign(p, sk, m)$

1. Parse $p = (q, \mathbb{G}, \mathbb{G}_T, \mathbf{e}, g)$.
2. Signer select $k \in \mathbb{Z}_q$ randomly and computer $t = g^k \text{ mod } p$, then send t to User.
3. User select $\alpha, \beta \in \mathbb{Z}_q$ randomly and computer $\omega = t g^\alpha y^\beta \text{ mod } p$, then compute $c = H(m || \omega)$ and $c' = c - \beta \text{ mod } q$, send c' to Signer.
4. Signer compute $u = k - c' \cdot sk \text{ mod } q$, and sent u to User.
5. User computer $v = u + \alpha \text{ mod } q$.
6. User output signature $\sigma = (c, v)$.

$Verify(p, pk, m, \sigma)$

1. Parse $p = (q, \mathbb{G}, \mathbb{G}_T, \mathbf{e}, g)$, $pk = (g_1, g^{g_1^x})$, $m = m_1, m_2, \dots, m_n$, and $\sigma = (c, v)$.
2. Compute $g^v y^c = \omega$.
3. Accept if $H(m || \omega) = c$; otherwise output \perp .

3.2 Linear Encryption Scheme

The Boneh's linear encryption scheme[16] is another block to build the EBS functionality, which is as follows:

$EKG(p)$:

1. Parse $p = (q, \mathbb{G}, \mathbb{G}_T, \mathbf{e}, g)$.
2. Select $a \in \mathbb{Z}_q$ and $b \in \mathbb{Z}_q$ randomly.
3. Output the secret key $sk_e = (a, b)$ and public key $pk_e = (g^a, g^b)$.

$Enc(p, pk_e, m)$

1. Parse $p = (q, \mathbb{G}, \mathbb{G}_T, \mathbf{e}, g)$.
2. Select $r \in \mathbb{Z}_q, s \in \mathbb{Z}_q$ randomly.
3. Compute $(c_1, c_2, c_3) = ((g^a)^r, (g^b)^s, g^{r+s}m)$.
4. Output $c = (c_1, c_2, c_3)$.

$Verify(p, sk_e, c)$

1. Parse $p = (q, \mathbb{G}, \mathbb{G}_T, \mathbf{e}, g)$, $sk_e = (a, b)$, and $c = (c_1, c_2, c_3)$.
2. Output $m = c_3 / (c_1^{1/a} / c_2^{1/b})$.

Theorem 1. [16] *Under DL assumption, the linear encryption scheme satisfies the indistinguishability.*

3.3 The Obfuscator for the EBS Functionality

Our EBS functionality consists of the blind signature scheme and encryption scheme above. We can construct a circuit C_{p,sk,pk_e} which contains a common parameter p , the signing secret key sk and the public encryption key pk_e . Note that the important point of obfuscation is that how to rerandomize the Enc to make the two results are scalar homomorphic. Here, we use the $ReRand$ algorithm, given a ciphertext (c_1, c_2, c_3) and public key $pk_e = (g^a, g^b)$, to rerandomize the ciphertext (c_1, c_2, c_3) as following: $(c_1(g^a)^{r'}, c_2(g^b)^{s'}, c_3g^{r'+s'}) \leftarrow ReRand(p, pk_e, (c_1, c_2, c_3))$, where $r', s' \in \mathbb{Z}_q$ are random parameter.

Given a circuit C_{p,sk,pk_e} , the detail of our obfuscator for the EBS Functionality Obf_{EBS} is below:

1. Extracts (p, sk, pk, pk_e) , where $sk = g_1^x, pk = g_1^{s_1}$ and $pk_e = (g^a, g^b)$.
2. Parse $p = (q, \mathbb{G}, \mathbb{G}_T, \mathbf{e}, g)$.
3. *Signer* runs $Enc(p, pk_e, sk) \rightarrow (c_1, c_2, sk') = ((g^a)^r, (g^b)^s, g^{r+s}g_1^x)$ to obtain a new signing secret key $sk' = g^{r+s}g_1^x$, computes the corresponding public signing key $pk' = (g_1, g^{g^{r+s}g_1^x})$, where $y' = g^{g^{r+s}g_1^x}$, and sends (c_1, c_2) to *User*.
4. *Signer* selects a random parameter $k \in \mathbb{Z}_q$, then sends $t = g^k$ to *User*.
5. Randomly chooses $\alpha, \beta \in \mathbb{Z}_q$, *User* counts $\omega' = tg^\alpha(y')^\beta$, $c' = H(m||\omega')$, and $c'' = c' - \beta$, then transmits c'' to *Signer*.

6. *Signer* gives *User* u' , where $u' = k - c'' \cdot sk'$.
7. *User* gets $(c', v') = (H(m || \omega'), u' + \alpha)$.
8. *User* computes $c_3 = c_1^{1/a} c_2^{1/b} c'$, rerandomize the ciphertext (c_1, c_2, c_3) as $C_1 = (c'_1, c'_2, c'_3) \leftarrow \text{ReRand}(p, pk_e, (c_1, c_2, c_3))$
(Note: $(c'_1, c'_2, c'_3) = ((g^a)^{r+r'}, (g^b)^{s+s'}, c' g^{r+r'+s+s'})$).
9. *User* compute $C_2 \leftarrow \text{Enc}(p, pk, v')$. (We define $C_2 = (c''_1, c''_2, c''_3)$).
10. *User* output the encrypted blind signature $\sigma = (C_1, C_2)$.

The output signature $\sigma = (C_1, C_2)$ is blind to the *Signer*, as *Signer* couldn't recognize either (c', v') or (α, β) . But *Verifier* can verify the signature σ by following verification algorithm $V(p, pk, m, \sigma)$:

1. Compute $c' = c'_3 / ((c'_1)^{1/a} (c'_2)^{1/b})$, $v' = c''_3 / ((c''_1)^{1/a} (c''_2)^{1/b})$, $g^{v'} y^{c'} = \hat{\omega}$, and $H(m || \hat{\omega}) = \hat{c}$.
2. If $\hat{c} = c'$, accept $\sigma = (C_1, C_2)$; otherwise output \perp .

Obviously, the obfuscation can be executed in polynomial time and have the same functionality compare with the original blind signature. So we omit the two proofs about functionality and polynomial slowdown.

4 The New Security Definition of the Blind Signature in the Context of EBS

We modify the above definitions to adapt to our proposals in the context of EBS. Because we need to prove the security of blind signature in the presence of the obfuscator we proposed. In this section, we allow the *Signer* to access the obfuscation circuit as follows:

Definition 5. (*Blindness w.r.t. EBS Obfuscator*) An encrypted signature scheme $EBS = (SKG, EKG, (S, U), V)$ w.r.t obfuscator is called blind if for any efficient algorithm A_3 , all sufficiently large $n \in \mathbb{N}$, and every $z \in \{0, 1\}^{\text{poly}(n)}$, there exist

$$2 \cdot \Pr \left[\begin{array}{l} p \leftarrow \text{Setup}(1^n); (pk, sk) \leftarrow SKG(p); (pk_e, sk_e) \leftarrow EKG(p); \\ C' \leftarrow \text{Obf}(C_{p,sk,pk_e}); \\ b \leftarrow \{0, 1\}; (\sigma_0, \sigma_1) \leftarrow A_3^{< \cdot, U(pk, m_b) >^1, < \cdot, U(pk, m_{1-b}) >^1}(p, pk, pk_e, C', z); \\ b^* \leftarrow A_3(\sigma_0, \sigma_1); \\ b = b^*. \end{array} \right] - 1 \leq \frac{1}{p(n)}$$

where A_3 is the malicious *Signer* and U is the honest user. If $\sigma_0 = \perp$ or $\sigma_1 = \perp$, then the *Signer* is not informed about other signature either.

Definition 6. (*One-more Unforgeability w.r.t. EBS Obfuscator*) An EBS scheme $(SKG, EKG, (S, U), V)$ is unforgeable if for any efficient algorithm A_4 (the malicious user), every polynomial $p(\cdot)$, all sufficiently large $n \in \mathbb{N}$, and every $z \in \{0, 1\}^{\text{poly}(n)}$, there exists

$$\Pr \left[\begin{array}{l} p \leftarrow \text{Setup}(1^n); (pk, sk) \leftarrow SKG(p); (pk_e, sk_e) \leftarrow EKG(p); \\ C' \leftarrow \text{Obf}(C_{p,sk,pk_e}); \\ ((m_1^*, \sigma_1^*), \dots, (m_{k+1}^*, \sigma_{k+1}^*)) \leftarrow A_4^{\ll S_{p,sk} \gg^k}(p, pk, pk_e, C', z); \\ \text{if } m_i^* \neq m_j^* \text{ for } i \neq j; \\ V(p, pk, m_i^*, \sigma_i^*) = \text{Accept for all } i; \text{ then return } 1. \end{array} \right] \leq \frac{1}{p(n)}$$

where $S_{p,sk}$ is the signing oracle (circuit).

Definition 7. (ACVBP w.r.t Dependent Oracles) Let $T(C)$ be a set of oracles dependent on the circuit C . A circuit obfuscator Obf for C satisfies the ACVBP w.r.t dependent oracle set T if the following condition holds: There exists a PPT oracle machine S (simulator) such that, for every PPT oracle machine D (distinguisher), every polynomial $p(\cdot)$, all sufficiently large $n \in \mathbb{N}$, and every $z \in \{0, 1\}^{\text{poly}(n)}$,

$$\left| \Pr \left[\begin{array}{l} C \leftarrow C_n; \\ C' \leftarrow Obf(C); \\ b \leftarrow D^{\llbracket C, T(C) \rrbracket}(C', z). \end{array} : b = 1 \right] - \Pr \left[\begin{array}{l} C \leftarrow C_n; \\ C'' \leftarrow S^{\llbracket C \rrbracket}(1^n, z); \\ b \leftarrow D^{\llbracket C, T(C) \rrbracket}(C'', z). \end{array} : b = 1 \right] \right| \leq \frac{1}{p(n)}$$

where $D^{\llbracket C, T(C) \rrbracket}$ means that D has sampling access to all oracles contained in $T(C)$ in addition to C .

5 The Security of Special EBS Obfuscator

In this section, we attribute the the security of special EBS obfuscator to DL assumption and the random oracle model. Although our obfuscation can remove the random oracle in theory, there still have no effective methods to do so. The reason why we prove it in random oracle model is that the signature scheme we choose is secure in random model, which is inherent property of the original signature scheme.

At first, we will prove the completeness property of our special EBS obfuscator. Informally, the signature is complete if for any message m , verification algorithm $V(p, pk, m, \sigma)$ always set up, i.e., the probability: $\Pr_{V(p, pk, m, \sigma)} = 1$.

Lemma 1. *The EBS obfuscation is complete.*

Proof. Once the verifier receives the signature $\sigma = (C_1, C_2)$, he finish the following proceeds in a polynomial reduction:

1. Compute $c' = c'_3 / ((c'_1)^{1/a} (c'_2)^{1/b})$.
2. Compute $v' = c''_3 / ((c''_1)^{1/a} (c''_2)^{1/b})$.

According to the verification algorithm, he has $g^{v'}(y')^{c'} = g^{u'+\alpha} g^{g_1^{x'} c'} = g^{u'+\alpha+g_1^{x'} c'}$. As $u' = k - c'' \cdot sk'$ and $c'' = c' - \beta$, obtain the equation $u' + \alpha + g_1^{x'} c' = k + \alpha + \beta sk'$. Thus, $g^{v'}(y')^{c'} = g^k g^\alpha g^{\beta sk'}$. Because $t = g^k$ and $y' = g^{sk'}$, he have $g^{v'}(y')^{c'} = t g^\alpha g^\beta = \omega'$. Then, the equation $H(m || \omega') = c'$ must be established. We outcome the completeness of EBS obfuscation.

Theorem 2. *Under DL assumption, for the EBS obfuscator and two messages m_0, m_1 selected by the malicious Signer A_3 , the distributions of σ_0 and σ_1 are computationally indistinguishable.*

Proof. The blindness of EBS obfuscator follows directly from the hardness of DL assumption in the group \mathbb{G} . More formally, we show that if an adversary A_3 can distinguish the signatures (σ_0, σ_1) of two message m_0 and m_1 under sk with non-negligible

probability, then we construct an adversary A' that will break the DL assumption with advantage ϵ as well.

At first, we analyze the result of EBS obfuscator: we get $\sigma = (C_1, C_2) = ((g^a)^{r+r'}, (g^b)^{s+s'}, c' g^{r+r'+s+s'}, (g^a)^{r''}, (g^b)^{s''}, v' g^{r''+s''})$, where r, s, r', s', r'', s'' are all random parameters. Through the process of obfuscation above, we have $c' = H(m||\omega')$, $v' = k - c' \cdot sk' + \beta \cdot sk' + \alpha$, where k, α, β are random and $\omega' = g^k g^\alpha (y')^\beta$. So when fix the secret key sk' , and v' depends the value of c' (i.e, v' and c' are linearly dependent). Thus the value of C_2 relay on c' . Cause C_1 and C_2 have the same form, so we can only consider C_1 in the following work(C_2 also has the same result, we omit it here). Let $\hat{s} = s + s', \hat{r} = r + r'$, so we have $C_1 = (g^{\hat{r}}, g^{\hat{s}}, g^{\hat{r}+\hat{s}})$.

A' works as follows:

- A' receives as input a tuple $(g, (a, b), B = g^{\hat{r}}, K = g^{\hat{s}}, W)$ where g is a random generator of the group \mathbb{G} and r, s are random exponents. The goal of A' is to determine whether $W = g^{\hat{r}+\hat{s}}$.
- A' picks a random generator g of group \mathbb{G} .
- On receiving two message m_0 and m_1 from A_3 , A' flips a bit b at random and sends the signature $\sigma_b := ((g^a)^{\hat{r}}, (g^b)^{\hat{s}}, c_b W)$ as the signature of m_b to A_3 .
- A_3 replies with a bit b^* . A' simply output 1 if $b = b^*$ (i.e., guessing that $W = g^{\hat{r}+\hat{s}}$); otherwise output a random bit(i.e., W is a random parameter).

It is easy to see that when W is random, the signature σ_b is independent of b and hence the success probability of A_3 in this case is exactly $\frac{1}{2}$. When $W = g^{\hat{r}+\hat{s}}$, the signature σ_b has the same distribution as the result of EBS obfuscator. According to the assumption, the adversary A_3 has advantage at least ϵ . That is, A' succeeds in determining whether $W = g^{\hat{r}+\hat{s}}$ with non-negligible advantage, A' break the DL assumption.

Theorem 3. [15] *The blind signature is one-more unforgeable if discrete logarithm is hard.*

Theorem 4. *Let $T(C_{p,sk,pk_e})$ be $S_{p,sk}$. If the EBS obfuscator satisfies ACVBP w.r.t dependent oracle set T , then the one-more unforgeability(OMU) w.r.t the EBS functionality implies the one-more unforgeability w.r.t EBS obfuscator.*

Proof. We show that, if there exists an adversary A_4 to break the OMU w.r.t Obf when the OMU w.r.t EBS is satisfied, then it will contradict the ACVBP w.r.t dependent oracle set T of EBS obfuscator. Let the distinguisher D have sample access to $T(C_{p,sk,pk_e})$ to check whether A_4 succeeds in breaking OMU w.r.t Obf.

1. Inputs a circuit C (either an obfuscated circuit or a simulated circuit) and an auxiliary-input z .
2. Extracts (p, pk, pk_e) through sampling access to C_{p,sk,pk_e} .
3. Samples access to $S_{p,sk}$ at most k times $((m_1^*, \sigma_1^*), \dots, (m_k^*, \sigma_k^*)) \leftarrow A_4^{\ll S_{p,sk} \gg^k}(p, pk, pk_e, C, z)$ to simulate $(m_{k+1}^*, \sigma_{k+1}^*)$.
4. $V(p, pk, m_{k+1}^*, \sigma_{k+1}^*) = \text{Accept}$ for $m_{k+1} \neq m_i$ where $i \in \{1, k\}$.

If C is an obfuscated circuit, then the probability D output 1 equal to the probability that A_4 breaks OMU w.r.t Obf, which is non-negligible by the assumption. And if C

is a simulated circuit, then the probability D output 1 is negligible, otherwise, A_4 can break the OMU w.r.t EBS functionality. So the probability which ACVBP established is non-negligible. Hence it will contradict the ACVBP w.r.t dependent oracle set T of EBS obfuscator. Theorem is established.

Theorem 5. *Let $T(C_{p,sk,pk_e})$ be $S_{p,sk}$. The EBS obfuscator satisfies ACVBP w.r.t dependent oracle set T under DL assumption.*

Proof. According to the EBS obfuscator we proposed, the security proof of obfuscator that contains an interactive process between Signer and User is a little different from the previous work. We use the variant of Hada's proof method. At first, we construct an simulator S to simulate the behaves of the obfuscated circuit; the two execution process is as follows (Note that the value (p, pk, pk_e) is easy to get through sampling access to C_{p,pk,pk_e} . So we mainly focus on $(sk', (c_1, c_2))$):

1. Inputs the security parameter 1^n and an auxiliary-input z .
2. Extracts (p, pk, pk_e) through sampling access to C_{p,pk,pk_e} .
3. Parses $p = (q, \mathbb{G}, \mathbb{G}_T, \mathbf{e}, g)$ and $pk = (g_1, g_1^{g_1^x})$.
4. Randomly select $Junk \leftarrow \mathbb{G}$.
5. Computes $c_1, c_2, c_3 \leftarrow Enc(p, pk_e, Junk)$ and set $sk' = c_3$.
6. Outputs $(sk', (c_1, c_2))$.

And then we consider the worst case that the interactive values are captured by adversary already, i.e., the value of $k, t, c'', u', v', \omega'$ are known (ω' can get by computing $g^{y'} y'^{c'}$), we proved the output distribution of S is indistinguishable from the real distribution (C_1, C_2) for any PPT distinguisher. In particular, when the distinguisher is permitted to sampling access to $CS = \{C_{p,sk,pk_e}, S_{p,sk}\}$, assume the probability that a distinguisher $D^{\ll C, S \gg}$ distinguish the two output distribution above is non-negligible. That is the probability of the following formula is non-negligible. And let $z = (k, t, c'', u', v', \omega')$ to be the auxiliary-input, we have:

Real one:

$$Pr \left[\begin{array}{l} p \leftarrow Setup(1^n); (pk_e, sk_e) \leftarrow EKG(p); \\ (pk, sk) = (g_1^{g_1^x}, g_1^x) \leftarrow SKG(p); \\ (c_1, c_2, sk') \leftarrow Enc(p, pk_e, sk); \\ pk' = (g_1, g_1^{g_1^{r+s} g_1^x}); \\ b \leftarrow D^{\ll C, S \gg}((p, pk_e, pk', sk', (c_1, c_2)), (k, t, c'', u', v', \omega')); \\ b = 1. \end{array} \right]$$

Junk one:

$$Pr \left[\begin{array}{l} p \leftarrow Setup(1^n); (pk_e, sk_e) \leftarrow EKG(p); \\ (pk, sk) = (g_1^{g_1^x}, g_1^x) \leftarrow SKG(p); \\ Junk \leftarrow \mathbb{G}; \\ (c_1, c_2, sk') \leftarrow Enc(p, pk_e, Junk); \\ pk' = (g_1, g_1^{g_1^{r+s} g_1^x}); \\ b \leftarrow D^{\ll C, S \gg}((p, pk_e, pk', sk', (c_1, c_2)), (k, t, c'', u', v', \omega')); \\ b = 1. \end{array} \right]$$

Third we construct an adversary (A_1, A_2) to break the indistinguishability of the linear encryption scheme. A_1 produces a message pair $(m_1, m_2) = (sk, Junk)$ and an associated hint $h = pk$. Given a ciphertext c (of either m_1 or m_2), A_2 distinguish the results of m_1 and m_2 by distinguisher D as follows:

1. Parse $p = (q, \mathbb{G}, \mathbb{G}_T, \mathbf{e}, g)$ and pk_e , ciphertext c and auxiliary $z = (k, t, c'', u', v', \omega')$.
2. Get the output m_1, m_2 of A_1 , $h = pk = g^{s^x}$ and $c = (c_1, c_2, sk')$, let $pk' = g^{c^3}$.
3. Simulate $D^{\ll C, S \gg}((p, pk_e, pk', sk', (c_1, c_2)), (k, t, c'', u', v', \omega'))$.
4. Output the result of D .

If c is a ciphertext of m_1 , then the probability A_2 output 1 equal to first probability, otherwise, it is equal to the later probability. According to the Theorem 1, the difference of the two probability above is negligible which contradicts to the assumption. Theorem is established.

6 Conclusion

A new functionality for obfuscation has been proposed in this paper under DL assumption and the hardness of discrete logarithm. Following Hohenberger and Hada's steps, we present two new security definition and our scheme is a further application which not only protect the Signer's secret key from revealing, but also keep the signature blinding from the Signer. This functionality is very useful in E-Cash and E-Vote. At the same time, our scheme resists different PPT adversaries and satisfies ACVBP w.r.t dependent oracle property. Furthermore, we will continue to focus on the research and application of obfuscation.

References

1. Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil Vadhan and Yang K. On the (im)possibility of obfuscating program. In CRYPTO, Advances in Cryptology-CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001. Proceedings, pages 1-18, 2001.
2. Shafi Goldwasser and Yael Tauman Kalai. On the impossibility of obfuscation with auxiliary input. In FOCS, 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS) 2005, 23-25 October 2005, Pittsburgh, PA, USA. Proceedings, pages 553-562, 2005.
3. Hoeteck Wee. On obfuscating point functions. In STOC, Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005, pages 523-532, 2005.
4. Dennis Hofheinz, John Malone-Lee, and Martijn Stam. Obfuscation for cryptographic purposes. In TCC 2007, Theory of Cryptography, 4th Theory of Cryptography Conference, TCC 2007, Amsterdam, The Netherlands, February 21-24, 2007. Proceedings, pages 214-232, 2007.
5. Nir Bitansky and Ran Canetti. On strong simulation and composable point obfuscation. In CRYPTO, Advances in Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001. Proceedings, pages 1-18, 2001.

6. Ran Canetti. Towards realizing random oracles: Hashing functions that hide all partial information. In CRYPTO, Advances in Cryptology-CRYPTO'97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 1997. Proceedings, pages 455-469, 1997.
7. Ran Canetti, Daniele Miccianico, and Omer Reingold. Perfectly one-way probabilistic hash functions. In STOC, pages 72-89, 2010.
8. Ben Lynn, Manoj Prabhakaran, and Amit Sahai. Positive results and techniques for obfuscation. In EUROCRYPT, Advances in Cryptology-EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004. Proceedings, pages 20-39, 2004.
9. Ran Canetti and Ronny Ramzi Dakdouk. Obfuscating point functions with multibit output. In EUROCRYPT, Advances in Cryptology-EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings, pages 92-112, 2010.
10. Susan Hohenberger, Guy N.Rothblum, Abhi Shelat, and Vinod Vaikuntanathan. Securely obfuscating re-encryption. In TCC, Theory of Cryptography, 4th Theory of Cryptography Conference, TCC 2007, Amsterdam, The Netherlands, February 21-24, 2007. Proceedings, pages 233-252, 2007.
11. Satoshi Hada. Secure obfuscation for encrypted signatures. In EUROCRYPT, Advances in Cryptology-EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30-June 3, 2010. Proceedings, pages 92-112, 2010.
12. Nishanth Chandran, Melissa Chase and Vinod Vaikuntanathan. Functional Re-encryption and Collusion-Resistant Obfuscation. In TCC, 9th Theory of Cryptography Conference, TCC 2012, Taormina, Sicily, Italy, March 19-21, 2012. Proceedings, pages 404-421, 2012.
13. David Chaum. Blind Signatures for Untraceable Payments. In R.L.Rivest, A.Sherman, and D.Chaum, editors. In CRYPTO 82, New York. Proceedings pages 199-203, 2010.
14. Tatsuaki Okamoto. Provably Secure and Practical Identification Schemes and Corresponding Signature Schemes. In CRYPTO, Advances in Cryptology \dagger CRYPTO92, 12th Annual International Cryptology Conference Santa Barbara, California, USA August 16-20. Proceedings pages 31-53, 1992.
15. Claus Peter Schnorr. Enhancing the Security of Perfect Blind DL-Signatures. In Information Sciences, Volume 176, Issue 10, 22 May 2006. Proceedings Pages 1305C1320, 2006.
16. Dan Boneh, Xavier Boyen, and Hovav Shacham. Short Group Signatures. In CRYPTO, Advances in Cryptology C CRYPTO 2004, 24th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 2004. Proceedings pages 41-55.