PLOS ONE

RESEARCH ARTICLE

Secure Obfuscation for Encrypted Group Signatures

Yang Shi[☯], **Qinpei Zhao**[☯], **Hongfei Fan, Qin Liu***

School of software engineering, Tongji University, Shanghai, China

☯ These authors contributed equally to this work.
* qin.liu@tongji.edu.cn

Citation: Shi Y, Zhao Q, Fan H, Liu Q (2015) Secure Obfuscation for Encrypted Group Signatures. PLoS ONE 10(7): e0131550. doi:10.1371/journal.pone.0131550

Editor: Vince Grolmusz, Mathematical Institute, HUNGARY

Received: November 21, 2014

Accepted: June 3, 2015

Published: July 13, 2015

Data Availability Statement: All relevant data are within the paper.

Funding: This work was supported by the National Natural Science Foundation of China (No. 61202382, www.nsfc.gov.cn) and the Fundamental Research Funds for the Central Universities (No. 2013KJ033, www.moe.gov.cn). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

Abstract

In recent years, group signature techniques are widely used in constructing privacy-preserving security schemes for various information systems. However, conventional techniques keep the schemes secure only in normal black-box attack contexts. In other words, these schemes suppose that (the implementation of) the group signature generation algorithm is running in a platform that is perfectly protected from various intrusions and attacks. As a complementary to existing studies, how to generate group signatures securely in a more austere security context, such as a white-box attack context, is studied in this paper. We use obfuscation as an approach to acquire a higher level of security. Concretely, we introduce a special group signature functionality-an encrypted group signature, and then provide an obfuscator for the proposed functionality. A series of new security notions for both the functionality and its obfuscator has been introduced. The most important one is the average-case secure virtual black-box property w.r.t. dependent oracles and restricted dependent oracles which captures the requirement of protecting the output of the proposed obfuscator against collision attacks from group members. The security notions fit for many other specialized obfuscators, such as obfuscators for identity-based signatures, threshold signatures and key-insulated signatures. Finally, the correctness and security of the proposed obfuscator have been proven. Thereby, the obfuscated encrypted group signature functionality can be applied to variants of privacy-preserving security schemes and enhance the security level of these schemes.

Introduction

Group signature was proposed by Cham and Heyst [1], which is a special type of digital signature for a group of persons. In the group signature setting, there is a group having numerous members and a single manager (the group manager). A single verification key called the group public key is associated with the group. Each group member has its own secret signing key based on which it can produce a signature relative to the group public key. The group manager has a master secret key. Given a signature Σ, based on the master secret key, the group manager can extract the identity of the group member who created Σ. It is called traceability. On the

other hand, those who are not holding the master secret key are unable to extract the identity of the group member who created Σ, which is called anonymity. In fact, a group signature has the following properties[1,2]: (i) only members of the group can sign messages; (ii) the receiver can verify whether it is a valid group signature; (iii) anonymity; (iv) traceability.

Group signature schemes provide functionalities which are applicable in many practical scenarios. In recent years, applications of group signature schemes in many emerging technologies or privacy-sensitive applications are studied, such as in social networks [3,4], medical information systems [5–7], Vehicular Ad hoc Networks (VANets) [8,9], electronic voting [10], Wireless Sensor Networks (WSNs) [11], electronic cash [12,13], and cloud computing [14–18] especially. These studies make great contributions for protecting security of information systems and privacy of users against various attacks. However, all these schemes are developed in the black-box model (or the so-called "black-box attack context"). In a black-box attack context, an adversary can access only the functionality of a cryptosystem. However, there is another security model (the white-box model) in which an adversary has total visibility of the implementation of the cryptosystem and full control over its execution platform. That is, the implementation of the cryptosystem is running in a White-Box Attack Context (WBAC).

In fact, we can find many typical WBACs, such as (1) a server or an endpoint for which a hacker has got the "root" or "admin" privilege; (2) a malicious host where mobile agents are running [19,20]; (3) an outdoor sensor node (of a WSN) captured by an attacker [21,22]; (4) the Digital Right Management (DRM) components in cable TV set-top boxes or IPTV equipments [23,24]; (5) On Board Units (OBUs) and Road Side Units (RSUs) in a VANet (e.g., the device suffers from the so called "Industrial insiders' attack" in [25] or the "On-board tampering" attack in [26], or even the "Malware attack" in [27]); and (6) mobile devices (e.g., smart phones and tablets) captured by an attacker [28].

In WBACs, obfuscating techniques should be used while implementing key-related cryptographic functionalities to protect privacy-preserving schemes or security protocols based on group signature schemes. Hence, we contribute to the security requirements as follows.

1. A special obfuscatable group signature functionality, i.e., the encrypted group signature, is proposed with a concrete scheme, and then a corresponding obfuscator is provided.
2. Security notions of the encrypted group signature functionality and notions of the corresponding obfuscators are proposed. The most important one of the new security notions is average-case secure virtual black-box property (ACVBP) w.r.t. Dependent Oracles and Restricted Dependent Oracles, which describes the security requirement of protecting the output of the proposed obfuscator, i.e., the obfuscated implementation of encrypted group signature functionality against collision attacks from group members. The security notions fit for many other application scenarios.
3. The correctness and security of the proposed obfuscator are proven. The efficiency of the proposed encrypted group signature functionality and its obfuscator is analyzed.

Besides the contributions to cryptography, the result is useful in many applications. For example, the proposed technique can be applied in cloud computing. In this application, an inner user can upload a file anonymously on the private cloud of a company. However, in the case that an investigation is needed, the group manager is capable of opening the identity of the user. For another example, in a privacy-preserving emergency call (PEC) scheme for mobile healthcare social networks, the obfuscatable encrypted group signature scheme and its obfuscator can be used to implement a decentralized emergency response system for a rapid response of emergency care in the network. The application demonstrates that, with the help of encrypted group signature technique, the PEC preserves users' privacy by hiding their

identities, and it avoids unnecessary disclosure of personal health information. The details of these applications are provided in Section 6.1.1 and 6.1.2 to demonstrate the applicability in concrete scenarios.

Furthermore, we found that the proposed solution can be adapted to identity-based cryptography and key-insulated cryptography. Therefore, how to transform the proposed obfuscatable encrypted group signature scheme into an obfuscatable encrypted identity-based signature scheme and an obfuscatable encrypted key-insulated signature scheme are sketched out in Section 6.1.3.

The remainder of this paper is organized as follows. The next section presents background information about obfuscation and obfuscators. The section also provides a brief introduction on the complexity assumptions needed in this paper. Section 3 first proposes an overall scheme that is a combination of a group signature scheme and an asymmetric encryption scheme, then an obfuscatable Encrypted Group Signature (EGS) functionality based on the overall scheme is provided. Section 4 presents an obfuscator for the proposed EGS functionality and proves the correctness and security of the obfuscator. Moreover, a series of security notions of the functionality and the obfuscators is also introduced in Section 4. In Section 5, we compare the results of the proposed scheme with the ones in other studies on obfuscation for cryptographic purpose. A discussion on possible applications and extensions of the proposed scheme and the obfuscator is provided in Section 6. The rationale behind the obfuscatable sign-then-encrypt functionalities and our main contributions is also discussed in this section. Finally, the article concludes in Section 7 with future work.

Preliminaries and Background

2.1 Obfuscation and Its Recent Advances

Informally, the goal of obfuscation is to make a computer program "unintelligible" while preserving its functionality, and an obfuscator is a “compiler” that performs such transformations [29,30]. As the results in [29–33] are mainly about the difficulties or even impossibilities of obfuscation, it is a hard work to find a secure obfuscator, even though for a special functionality. Some positive results were reported besides these negative results for general-purpose obfuscation, such as in [29–33]. However, these positive results mainly serve as theoretical illustrations and many of these positive results are not suitable for practical applications.

Despite these positive results for theoretical study, obfuscators for cryptographic-related functionalities with acceptable runtime cost remained elusive until the obfuscatable encrypted signature [34] and the obfuscatable re-encryption [35] were proposed. Fortunately, after these two articles were published, several obfuscatable cryptographic-related functionalities were identified and the corresponding secure obfuscators were proposed in recent years [36–41].

We note that these positive results do not violate the general impossibility results. One of the reasons is that they are not general-purpose obfuscation. The other one is that they have used distinct weak security criteria such as the ACVBP proposed by Hohenbergeret al. [35] or the simulation-based average-case virtual black-box property proposed by Hofheinz et al. [42].

Two general-purpose obfuscators [43,44] were proposed in recent two years, however, they either use a weak security notion or use the homomorphic encryption techniques with high costs of space and time.

Both general-purpose obfuscators and specialized obfuscators for cryptographic functionalities usually set the input as a (probabilistic) circuit in theoretical analysis. A brief review on probabilistic circuits and circuit obfuscators is provided in the following subsection.

2.2. Probabilistic Circuits and Circuit Obfuscators

As prerequisites of security analysis, we use the conventional definitions and notations of probabilistic circuits and circuit obfuscators following those in [29,30,34,35].

Let Poly(λ) denote the set of all polynomials of λ. Let $\mathbb{C}_\lambda$ be a set of polynomial-size circuits with input length $l_{Input}(\lambda) \in \text{Poly}(\lambda)$ and output length $l_{Output}(\lambda) \in \text{Poly}(\lambda)$. Usually, we use $\mathbb{C} = \{\mathbb{C}_\lambda\}_{\lambda \in \mathbb{N}}$ to denote a class of such circuits, where there exists an associated Probabilistic Polynomial Time (PPT) generation algorithm which takes as input 1^λ and generates a random circuit $\mathbb{C} \in_{\$} \mathbb{C}_\lambda$. In studies on obfuscation for cryptographic purpose, it usually corresponds to the random generation of system parameters or cryptographic keys on the security parameter 1^λ. The generation process is denoted by $C \leftarrow \mathbb{C}_\lambda$. When a circuit C is used as an input argument or an output result of an algorithm, it is assumed that a specification of the circuit is used implicitly.

Let *para* be the set of regular input parameters and *rand* be the random input variable. Suppose that *C(para*, *rand)* is a probabilistic circuit. Given an arbitrary regular input *para*, we can consider $C(para,\cdot)$ (or $C_{para}(\cdot)$) as a sampling algorithm for the distribution obtained by evaluating the output of *C(para*, *rand)* on random variable *rand*. Given a couple of probabilistic circuits (C_0, C_1) whose regular inputs are of the same length, the two distributions produced by $C_0(para,\cdot)$ and $C_1(para,\cdot)$ are denoted by the pair $(C_0(para), C_1(para))$. The statistical difference between them is defined in (1) [34].

$$\begin{aligned}&\text{StaDiff}(C_0(para),\ C_1(para))\\&= \frac{1}{2}\sum_{y\in\{0,1\}^{l_{Output}(\lambda)}}\left|\Pr[out \leftarrow C_0(para)\ : out = y] - \Pr[out \leftarrow C_1(para)\ : out = y]\right|\end{aligned} \tag{1}$$

Moreover, for a Turing machine M, its black-box access to a probabilistic circuit C can be divided into two categories, i.e., "oracle access" and "sampling access". Oracle access means that the Turing machine M is allowed to set both regular and random inputs. It is denoted by M^C as the traditional way. Sampling access means that the Turing machine M is only allowed to set the regular input. It is denoted by $M^{\ll C \gg}$.

An obfuscator for a class of circuits $\mathbb{C} = \{\mathbb{C}_\lambda\}_{\lambda \in \mathbb{N}}$ is a PPT machine which takes a circuit $C \in \mathbb{C}_\lambda$ as input and outputs an unintelligible circuit C' which preserves the functionality. The formal description of "preserving functionality" is given by Definition 1 in section 4.2.

2.3. Complexity Assumptions

The overall scheme containing the obfuscatable EGS functionality in this paper is built based on a group signature scheme [45] and an asymmetric linear encryption scheme [46], so, we make use of the following four complexity assumptions as the theoretical basis of our work.

Remark 1. Although most cryptosystems based on pairings assume for simplicity that bilinear groups have prime order. In our case, it is important that the pairing is defined over a group G containing $|G| = n$ elements, where $n = pq$ has a (ostensibly hidden) factorization in two large primes, $p \neq q$. Moreover, G_p and G_q denote the cyclic subgroups of G with respective order p and q.

i. The Computational Diffie-Hellman (CDH) assumption in bilinear groups. This assumption states that, given a triple $(g,\ g^a,\ g^b) \in G_p^3$ for random exponents $a, b \in Z_p$, there is no PPT algorithm that can compute $g^{ab} \in G_p$ with non-negligible probability. Because of the bilinear pairing, CDH in G_p implies a "Gap Diffie Hellman" assumption.

ii. The subgroup decision assumption. Consider an "instance generator" algorithm $\mathcal{GG}$ that, on input a security parameter 1^λ, outputs a tuple (p, q, G, G_T, e), in which p and q are

independent uniform random λ-bit primes, G and G_T are cyclic groups of order $n = pq$ with efficiently computable group operations (over their respective elements, which must have a polynomial size representation in λ), and e: $G \times G \rightarrow G_T$ is a bilinear map. The subgroup decision assumption is that on input a tuple $(n = pq, G, G_T, e)$ derived from a random execution of $\mathcal{GG}(1^\lambda)$, and an element w selected at random either from G or from G_q, there is no (PPT) algorithm can decide whether $w \in G_q$ with non-negligible advantage.

iii. The Hidden Strong Diffie-Hellman assumption. We first define the l-HSDH problem as follows: On input three generators g, h and g^ω of G_p, and l-1 distinct triples $(g^{1/(\omega+c_i)}, g^{c_i}, h^{c_i}) \in G_p^3$ where $c_i \in Z_p$, output another such triple $(g^{1/(\omega+c)}, g^c, h^c) \in G_p^3$ distinct of all the others. The Hidden Strong Diffie-Hellman assumption states that, in a family of prime order bilinear groups generated by the instance generator $\mathcal{GG}$, there is no PPT algorithm can solve the HSDH problem in the bilinear group $(p, G_p, \hat{e}) \leftarrow \mathcal{GG}(1^\lambda)$ with non-negligible probability for sufficiently large $\lambda \in \mathbb{N}$.

iv. The Decisional Linear (DLIN) assumption. We first define the Decision Linear Problem in G as follows: Given u, v, h, u^a, v^b, $h^c \in G$ as input, output yes if $a+b = c$ and no otherwise. The DLIN assumption states that, there is no (PPT) algorithm can solve the Decision Linear Problem with non-negligible advantage.

Remark 2. The DLIN problem is originally defined in a prime-order group in [46]. In our case, we use the DLIN assumption over composite-order group, similar assumptions could be found in literatures such as [47] and [48].

An Obfuscatable Encrypted Group Signature Functionality

We propose an overall scheme that is a combination of a group signature scheme and an asymmetric encryption scheme. The scheme consists of seven algorithms: Setup, Enroll, Sign, Verify, EKGen, Enc and Dec. Based on the scheme, an obfuscatable EGS algorithm implements the EGS functionality is then provided.

3.1. The Overall Scheme

There are seven types of roles (see Fig 1) in the scheme: The first type is the group master. It is a trusted authority (TA) which is in charge of initializing system parameters, generating public parameters, setting up the group and issuing secret signing keys to the group members. Furthermore, the TA also certificates public encryption keys for all users (member or nonmember of the group). Sometimes, the master key MK is destroyed once the group is set up. The second type is the group manager, which is given the ability to identify signers using the tracing key TK, but not to enroll new users or create new signing keys. The third type is regular member users (group members, or signers) that each one is given a distinct secret signing key K_{ID}. The fourth type is verifiers, who can verify signatures using the public parameters. The fifth type is encryptors and the sixth type is decryptors which are not shown in Fig 1 as they are too simple. These three types (i.e., the fourth type, the fifth type and the sixth type) of users could be either a group member of nonmember. The seventh type is obfuscators.

Algorithms proposed in [45] (a group signature scheme) and [46] (an asymmetric linear encryption scheme) are used to construct the overall scheme. Hence, the algorithms of the overall scheme are similar to the corresponding algorithms in [45] and [46]; in fact, they are slightly modified or only different in description. Note that it is not a trivial work to the overall scheme because the signing algorithm Sign and the encryption algorithm Enc should be "compatible" to construct the obfuscatable EGS functionality. These algorithms in the scheme are

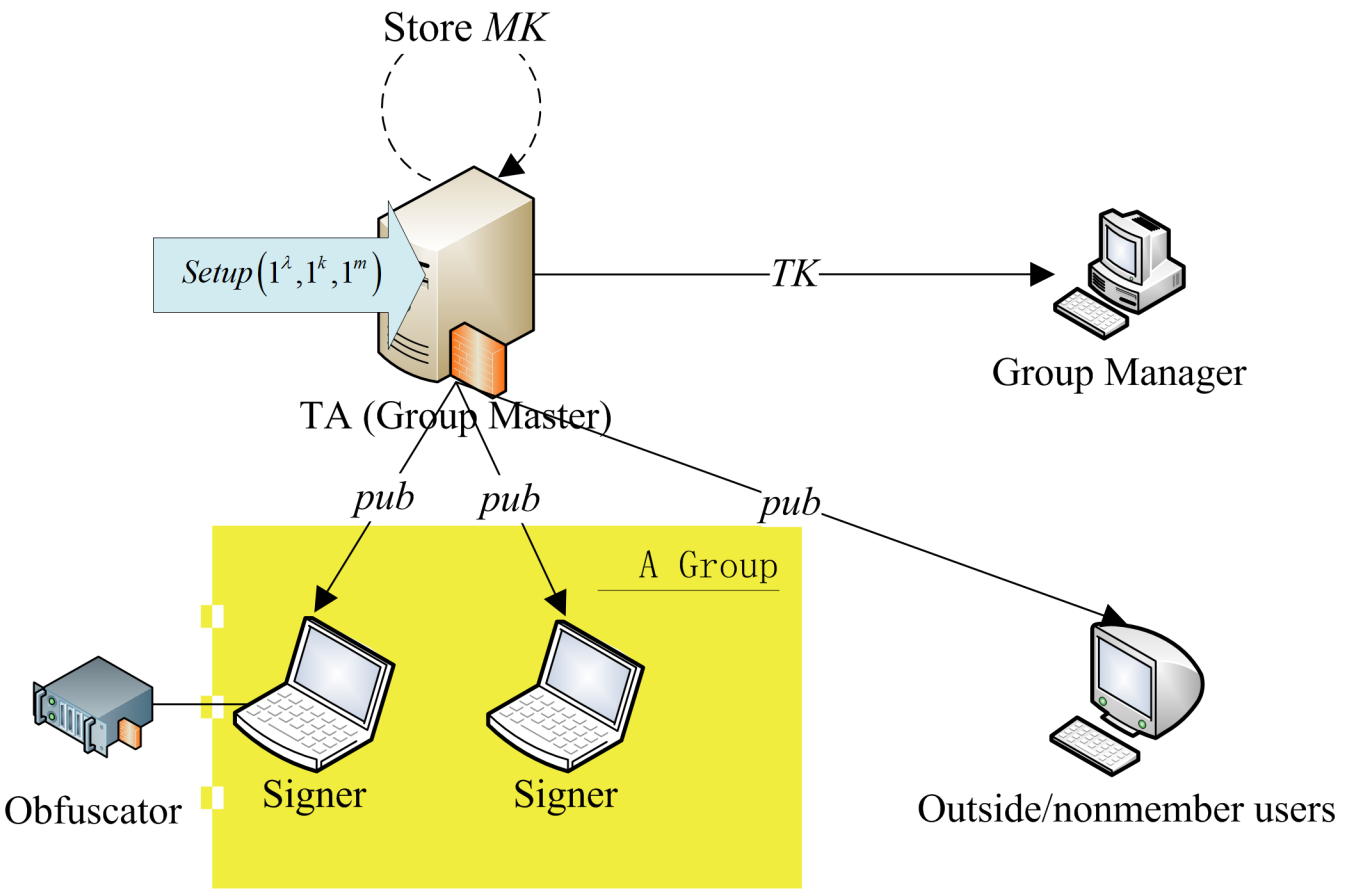


Fig 1. The usage of the Setup algorithm.

doi:10.1371/journal.pone.0131550.g001

introduced in the following subsections. To make the algorithms easier to understand, we list frequently used symbols in Table 1.

3.1.1. Setup. The Setup algorithm takes a security parameter in unary as input. The algorithm outputs *pub* (a tuple consists of system parameters and public values), the master enrollment key *MK*, and the group manager's tracing key *TK*. Suppose that up to 2^k signers are supported in the group, and the message space is $\{0,1\}^m$, where $k = O(\lambda)$ and $m = O(\lambda)$. Let *Gen* [*group*] denote the set of generators of the "*group*". The usage of the algorithm is illustrated in Fig 1.

The algorithm proceeds as follows.

Algorithm $Setup(1^{\lambda},1^{k},1^{m})$

Begin
$p, q \xleftarrow{\$} \mathbb{N}$, p and q are prime numbers s.t. $\log_2 p = \Theta(\lambda) > k \wedge \log_2 q = \Theta(\lambda) > k$
$n \leftarrow p \cdot q$
builss a cyclic bilinear group G of order n
Let G_p be the cyclic subgroups of G of order p
Let G_q be the cyclic subgroups of G of order q
$h \xleftarrow{\$} Gen[G_q]$
$g \xleftarrow{\$} Gen[G]$

Table 1. Symbols.

Symbol	Description
pub	A tuple consists of system parameters and public values
MK	The master enrollment key
TK	The group manager's tracing key
2^k	The maximum number of signers
$\{0,1\}^m$	The message space
G, G_T	Cyclic groups
G_p, G_q	Subgroups of G
ID	The user's identity
s_{ID}	A secret unique value corresponding to ID
K_{ID}	The private signing key
M	A message
σ	A signature
PK_e	The encryption key
SK_e	The decryption key
C	Ciphertext

doi:10.1371/journal.pone.0131550.t001

$\alpha, \omega \xleftarrow{\$} Z_n$
$A \leftarrow \hat{e}(g,\ g) \in G_T$
$\Omega \leftarrow g^{\omega} \in G$
$u, v', v_1, \ldots, v_m \in Gen[G]$
$PP \leftarrow (g, h, u, v', v_1, \ldots, v_m, \Omega, A) \in G \times G_q \times G^{m+3} \times G_T$
$pub \leftarrow (n, \hat{e}, G, G_T, PP)$
$MK \leftarrow (g^{\alpha}, \omega) \in G \times Z_n$
$TK \rightarrow q \in \mathbb{N}$
output (pub, MK, TK)
End

3.1.2. Enroll. The algorithm Enroll serves for creating a signing key for a user whose identity is ID, where $0 \leq \mathrm{ID} < 2^k < p$. The enroll algorithm takes *pub* (a tuple consists of system parameters and public values), the master key *MK*, and the user's identity ID as input. It outputs a unique identifier $s_{\mathrm{ID}} \in Z_n$ and a corresponding private signing key K_{ID}. The secret unique value s_{ID} can be later used for tracing purposes. This value must be chosen so that $\omega + s_{\mathrm{ID}}$ lies in Z_n^*. The usage of the algorithm is illustrated in Fig 2.

The algorithm proceeds as follows.

Algorithm *Enroll* (pub, MK, ID)

Begin
$(n, \hat{e}, G, G_T, PP) \leftarrow pub$
$(g, h, u, v', v_1, \ldots, v_m, \Omega, A) \leftarrow PP$
$(g^{\alpha}, \omega) \leftarrow MK$
$s_{\mathrm{ID}} \in_{\$} Z_n, s.t. \omega + s_{\mathrm{ID}} \in Z_n^*$
$K_{\mathrm{ID}} = (K_1, K_2, K_3) \leftarrow \left((g^{\alpha})^{\frac{1}{\omega + s_{\mathrm{ID}}}},\ g^{s_{\mathrm{ID}}},\ u^{s_{\mathrm{ID}}} \right) \in G^3$
End

3.1.3. Sign. The signing algorithm takes *pub*(a tuple consists of system parameters and public values), a group member's private signing key K_{ID}, and the message $M = (\mu_1, \ldots, \mu_m) \in$

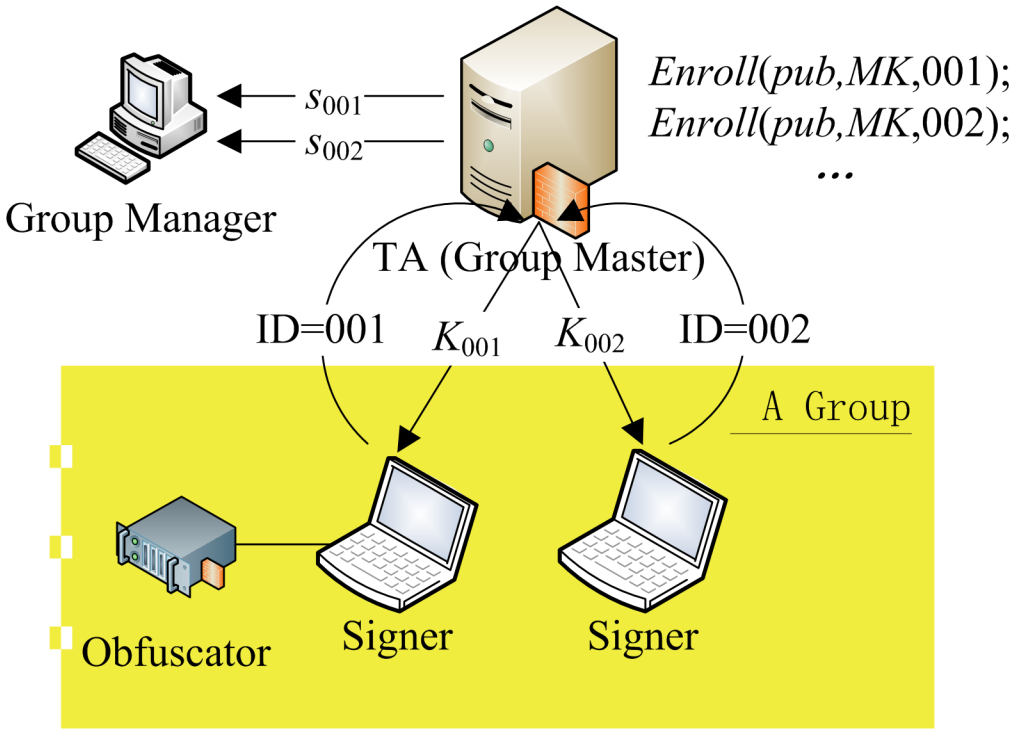


Fig 2. The usage of the Enroll algorithm.

doi:10.1371/journal.pone.0131550.g002

$\{0,1\}^m$ as input. The algorithm outputs a signature $(\sigma_1, \sigma_2, \sigma_3, \sigma_4, \pi_1, \pi_2)$ using the following procedure.

Algorithm $Sign\,(pub, K_{ID}, M)$

Begin

$(n, \hat{e}, G, G_T, PP) \leftarrow pub$

$(g, h, u, v', v_1, \ldots, v_m, \Omega, A) \leftarrow PP$

$s \xleftarrow{\$} Z_n$

$(\mu_1, \ldots, \mu_m) \leftarrow M$

$\theta = (\theta_1, \theta_2, \theta_3, \theta_4) = \left(K_1,\ K_2,\ K_3 \cdot \left(v' \prod_{i=1}^{m} v_i^{\mu_i}\right)^s, g^{-s}\right)$

$t_1, t_2, t_3, t_4 \xleftarrow{\$} Z_n$

$\sigma_1 \leftarrow \theta_1 \cdot h^{t_1}$

$\sigma_2 \leftarrow \theta_2 \cdot h^{t_2}$

$\sigma_3 \leftarrow \theta_3 \cdot h^{t_3}$

$\sigma_4 \leftarrow \theta_4 \cdot h^{t_4}$

$\pi_1 \leftarrow h^{t_1 t_2} \cdot (\theta_1)^{t_2} \cdot (\theta_2 \Omega)^{t_1}$

$\pi_2 \leftarrow u^{t_2} \cdot g^{-t_3} \cdot \left(v' \prod_{i=1}^{m} v_i^{\mu_i}\right)^{t_4}$

$output\ (\sigma_1, \sigma_2, \sigma_3, \sigma_4, \pi_1, \pi_2)$

End

3.1.4. Verify. The group signature verification algorithm Verify takes *pub*(a tuple consists of system parameters and public values), a signature σ from an unknown group member, and a

message M as input. If the signature σ is valid, it outputs 1, otherwise, outputs 0. Note that the verification algorithm outputs 1 only implies that the signature is generated by a member of the given group, but does not reveal the identity of the original signer.

Algorithm *Verify*(pub, σ, M)

Begin
$(\sigma_1, \sigma_2, \sigma_3, \sigma_4, \pi_1, \pi_2) \leftarrow \sigma$
$(n, \hat{e}, G, G_T, PP) \leftarrow pub$
$(g, h, u, v', v_1, \ldots, v_m, \Omega, A) \leftarrow PP$
$(\mu_1, \ldots, \mu_m) \leftarrow M$
$T_1 \leftarrow A^{-1} \cdot \hat{e}(\sigma_1, \sigma_2\Omega)$
$T_2 \leftarrow \hat{e}(\sigma_2, u) \cdot \hat{e}(\sigma_3, g)^{-1} \cdot \hat{e}\left(\sigma_4, v' \prod_{i=1}^{m} v_i^{\mu_i}\right)^{-1}$
IF$(T_1 = \hat{e}(h, \pi_1) \wedge T_2 = \hat{e}(h, \pi_2))$
output 1
Else
output 0
End

3.1.5. Open. As it is illustrated in Fig 3, to recover the identity of the signer, the group manager using the algorithm Open to test whether the signature is generated by a specific

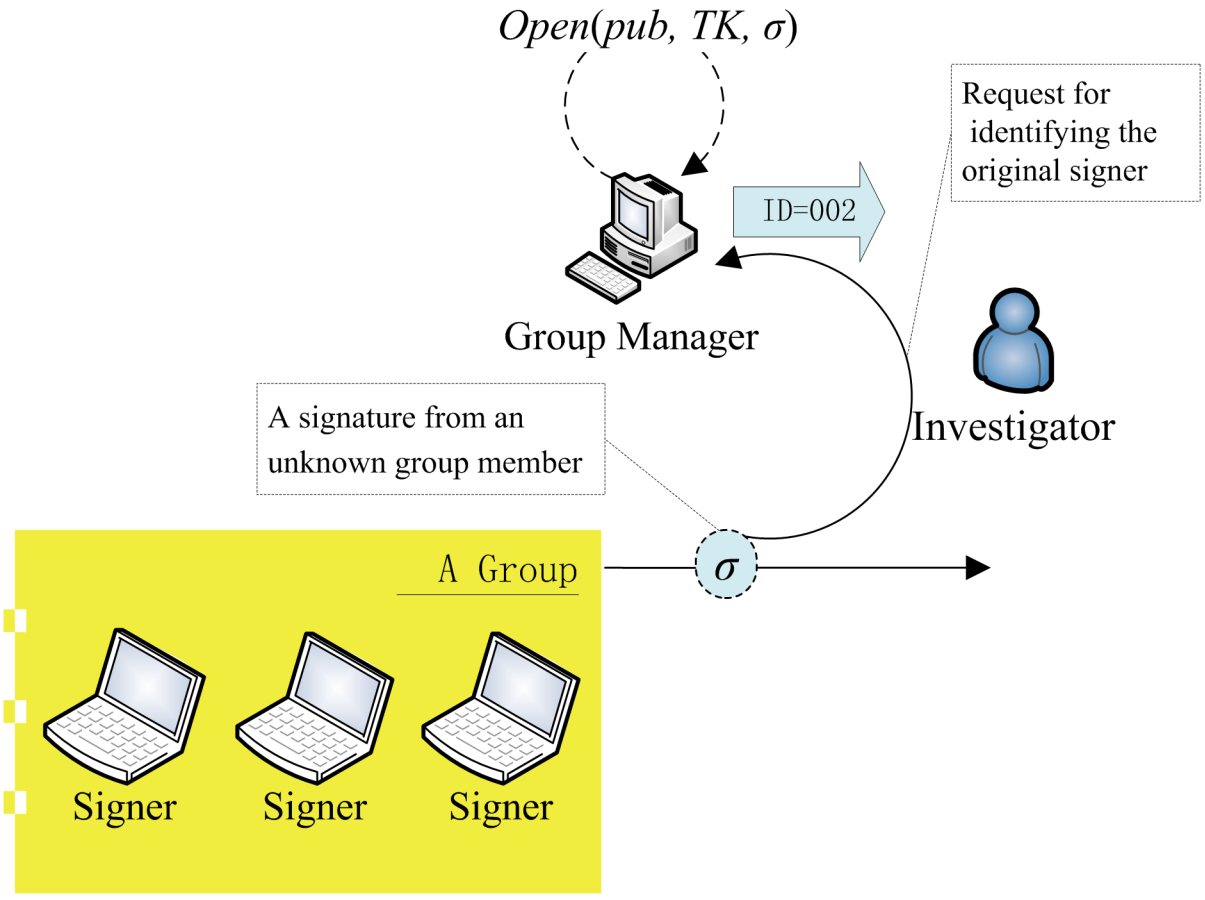


Fig 3. The usage of the Open algorithm.

doi:10.1371/journal.pone.0131550.g003

member of the group. If it is true, the algorithm outputs the identity of the member, otherwise, the output is $\perp$.

The algorithm proceeds as follows.

Algorithm $Open(pub, \sigma, TK)$

Begin
  $(\sigma_1, \sigma_2, \sigma_3, \sigma_4, \pi_1, \pi_2) \leftarrow \sigma$
  $(n, \hat{e}, G, G_T, PP) \leftarrow pub$
  $(g, h, u, v', v_1, \ldots, v_m, \Omega, A) \leftarrow PP$
  For Each ID
  IF$\left((\sigma_2)^{TK} = (g^{s_{ID}})^{TK}\right)$
    $output$ 1; $exit$;
  $output$ $\perp$
End

3.1.6. EKGen. The encryption key generation algorithm EKGen takes the public parameters *pub* as input, and generates a key pair (PK_e, SK_e) as follows. Note that for each user, the TA should provide a certification for the public encryption key PK_e.

Algorithm $EKGen(pub)$

Begin
  $(n, \hat{e}, G, G_T, PP) \leftarrow pub$
  $(g, h, u, v', v_1, \ldots, v_m, \Omega, A) \leftarrow PP$
  $a, b \xleftarrow{\$} Z_n$
  $PK_e \leftarrow (g^a, g^b)$, $SK_e \leftarrow (a, b)$
  $output(PK_e, SK_e)$
End

3.1.7. Enc and Dec. The encryption algorithm Enc takes the public parameters *pub*, the encryption key PK_e, and a (encoded) plaintext $M \in G$ as input. The algorithm encrypts the plaintext and then outputs the ciphertext as follows.

Remark 3. For simplicity, we use $Enc_{PK_e}(\cdot)$ to denote the encryption algorithm while the encryption key is PK_e. Similarly, we use $Dec_{SK_e}(\cdot)$ to denote the decryption algorithm while the decryption key is SK_e.

Algorithm $Enc_{PK_e}(M)$

Begin
  $(n, \hat{e}, G, G_T, PP) \leftarrow pub$
  $(g, h, u, v', v_1, \ldots, v_m, \Omega, A) \leftarrow PP$
  $x, y \xleftarrow{\$} Z_n$
  $(PK_{e,0}, PK_{e,1}) \leftarrow PK_e$
  $C \leftarrow (PK_{e,0}^{x}, PK_{e,1}^{y}, g^{x+y} \cdot M)$
  $Output$ C
End

The decryption algorithm Dec works as follows.

Algorithm $Dec_{SK_e}(C)$

Begin
  $(n, \hat{e}, G, G_T, PP) \leftarrow pub$
  $(C_1, C_2, C_3) \leftarrow C$
  $(a, b) \leftarrow SK_e$

$M \leftarrow C_3/\left(C_1^{\frac{1}{a}} \cdot C_2^{\frac{1}{b}}\right)$
output M
End

3.2. The Encrypted Group Signature Functionality

The encrypted group signature (EGS) functionality EGS_{pub,sk,PK_e}, with respect to a tuple *pub* that consists of system parameters and public values, the signing key *sk*, and the encryption key PK_e, works as follows.

Algorithm $EGS_{pub,sk,PK_e}(M)$

Begin
IF $(M = NULL)$
output (pub, PK_e)
Else
$(\sigma_1, \sigma_2, \sigma_3, \sigma_4, \pi_1, \pi_2) \leftarrow Sign(pub, sk, M)$

$$output: \begin{pmatrix} Enc_{PK_e}(\sigma_1), Enc_{PK_e}(\sigma_2), \\ Enc_{PK_e}(\sigma_3), Enc_{PK_e}(\sigma_4), \\ Enc_{PK_e}(\pi_1), Enc_{PK_e}(\pi_2) \end{pmatrix}$$

End

The activities of the overall scheme and the EGS functionality, include setting up the system, activating a group manager, enrolling a group member, generating and verification of a encrypted group signature, and opening the signature, are illustrated in Fig 4.

3.3. Efficiency Analysis

In Table 2, we list the numbers of various operations that are needed to perform each algorithm in section 3.1 and 3.2 column by column. It is shown from the table that the scheme has high efficiency in general, as the time-consuming pairing operation is only used in the Verify algorithm.

Remark 4. The algorithm Setup also needs some operations that have not been listed in Table 2, such as generation of groups and large prime numbers, and random selection of group generators.

Remark 5. The algorithm Open can be done in constant time. Because the value $(g^{s_{ID}})^q$ can be calculated once and for all for each user, we can construct a lookup table of $(g^{s_{ID}})^q$ for all users in the group. With the help of the lookup table, the algorithm "Open" only needs to compute $(\sigma_2)^q$.

In the next section, we introduce an obfuscator for the proposed EGS functionality, and prove the correctness and security of the obfuscator.

A Secure Obfuscator for the Special EGS Functionality

In this section, we introduce a secure obfuscator for the special EGS functionality. The proposed obfuscator can either be deployed in a physically secure device or in the signer's host.

Fig 5 illustrates the workflow of obfuscation.

4.1. Design of the Obfuscator

In this subsection, we propose an obfuscator Obf_{EGS} for the C_{pub,sk,PK_e} that implements the encrypted group signature (EGS) functionality as follows.

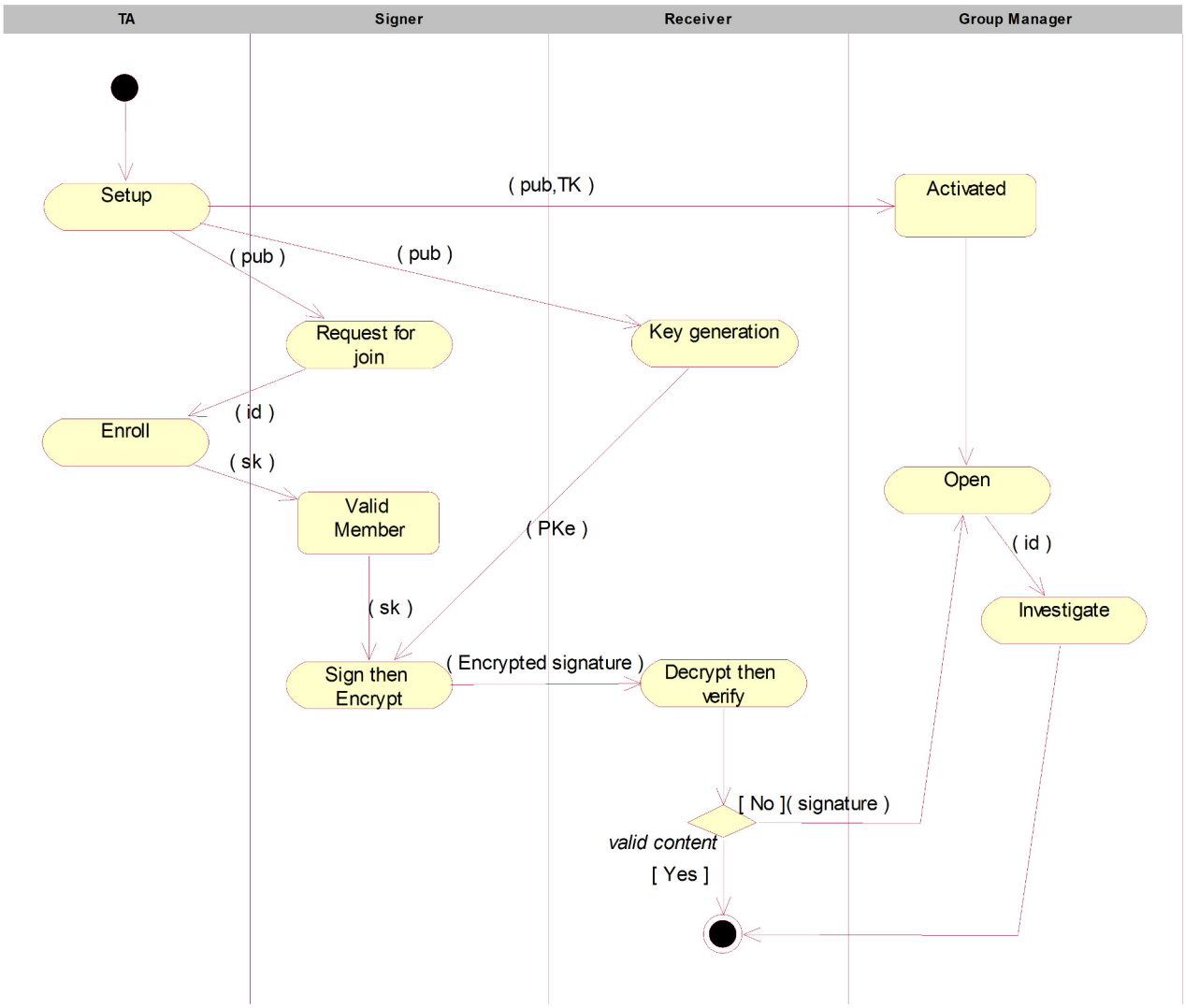


Fig 4. The activity diagram.

doi:10.1371/journal.pone.0131550.g004

$\mathtt{Obf}_{EGS}(C_{pub,sk,PK_e})$
$(PK_{e,0}, PK_{e,1}) \leftarrow PK_e$
$(K_1, K_2, K_3) \leftarrow sk$
$(n, \hat{e}, G, G_T, PP) \leftarrow pub$
$x_1, y_1 \xleftarrow{\$} Z_n, x_2, y_2 \xleftarrow{\$} Z_n, x_3, y_3 \xleftarrow{\$} Z_n$
$\tilde{K}_1 \leftarrow g^{x_1+y_1} \cdot K_1, \tilde{K}_2 \leftarrow g^{x_2+y_2} \cdot K_2, \tilde{K}_3 \leftarrow g^{x_3+y_3} \cdot K_3$
$C_{x_1} \leftarrow PK_{e,0}{}^{x_1}, C_{y_1} \leftarrow PK_{e,1}{}^{y_1}$
$C_{x_2} \leftarrow PK_{e,0}{}^{x_2}, C_{y_2} \leftarrow PK_{e,1}{}^{y_2}$
$C_{x_3} \leftarrow PK_{e,0}{}^{x_3}, C_{y_3} \leftarrow PK_{e,1}{}^{y_3}$
$z \leftarrow (C_{x_1}, C_{y_1}, C_{x_2}, C_{y_2}, C_{x_3}, C_{y_3}, \tilde{K}_1, \tilde{K}_2, \tilde{K}_3)$
Constructs and outputs an obfuscated circuit R_{pub,z,PK_e} that does the following on the input message M:
Begin
IF (M=$NULL$)

Table 2. Efficiency of the algorithms (listed in number of operations).

		Setup	Enroll	Sign	Verify	Open	EKGen	Enc	Dec	EGS
In Z_n	Rand	2	1	5	0	0	2	2	0	17
	Add	0	1	0	0	0	0	1	0	6
	Mult	0	1	1	0	0	0	0	0	1
	Inv	0	1	0	0	0	0	0	2	0
	Neg	0	0	2	0	0	0	0	0	0
In G	Rand	0	0	0	0	0	0	0	0	0
	Mult	0	0	2m+9	m+1	0	0	1	2	2m+15
	Exp	2	3	2m+12	m	1	2	3	2	2m+30
	Inv	0	0	0	0	0	0	0	2	2
In G_T	Mult	0	0	0	3	0	0	0	0	0
	Inv	0	0	0	3	0	0	0	0	0
$G^2 \rightarrow G_T$	Pair	1	0	0	6	0	0	0	0	0

"Rand" denotes the operation that generates a random element of the group or ring. "Add" and "Mult" denote the addition and multiplication, respectively. "Neg" denotes the operation that generates an addictive inverse and "Inv" denotes the operation that generates a multiplicative inverse. "Pair" denotes the pairing operation.

doi:10.1371/journal.pone.0131550.t002

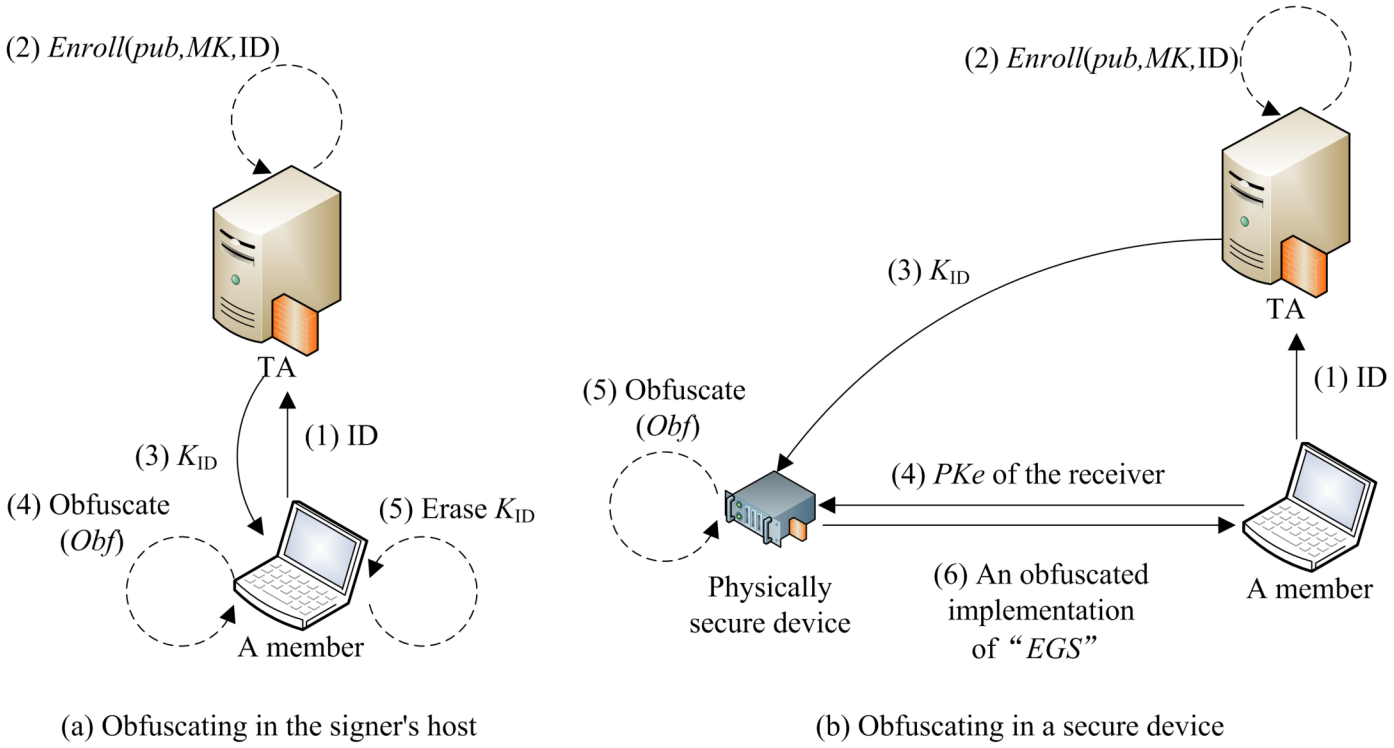


Fig 5. The workflow of obfuscation.

doi:10.1371/journal.pone.0131550.g005

output (pub, PK_e)
Else
$(\mu_1 \ldots \mu_m) \leftarrow M$
$(PK_{e,0}, PK_{e,1}) \leftarrow PK_e$
$(C_{x_1}, C_{y_1}, C_{x_2}, C_{y_2}, C_{x_3}, C_{y_3}, \tilde{K}_1, \tilde{K}_2, \tilde{K}_3) \leftarrow z$
$(n, \hat{e}, G, G_T, PP) \leftarrow pub$
$s \xleftarrow{\$} Z_n$
For(i=1; $i \leq 4$; i++)
$t_i \xleftarrow{\$} Z_n$
EndFor
For(i=1; $i \leq 6$; i++)
$x_i^*, y_i^* \xleftarrow{\$} Z_n$
IF($i \leq 3$)
$C_{x_i}^* \leftarrow (PK_{e,0})^{x_i^*} \cdot C_{x_i}$
$C_{y_i}^* \leftarrow (PK_{e,1})^{y_i^*} \cdot C_{y_i}$
Else IF(i=5)
$C_{x_i}^* \leftarrow (PK_{e,1})^{x_i^*} \cdot C_{x_1}^{t_2} \cdot C_{x_2}^{t_1}$
$C_{y_i}^* \leftarrow (PK_{e,1})^{y_i^*} \cdot C_{y_1}^{t_2} \cdot C_{y_2}^{t_1}$
Else
$C_{x_i}^* \leftarrow (PK_{e,0})^{x_i^*}$
$C_{y_i}^* \leftarrow (PK_{e,1})^{y_i^*}$
EndFor
$\tilde{\theta}_1 \leftarrow \tilde{K}_1, \tilde{\theta}_2 \leftarrow \tilde{K}_2$
$s \xleftarrow{\$} Z_n$
$\tilde{\theta}_3 \leftarrow \tilde{K}_3 \cdot \left(v' \prod_{i=1}^{m} v_i^{\mu_i} \right)^s$
$\tilde{\theta}_4 \leftarrow g^{-s}$
For(i=1; $i \leq 4$; i++)
$\tilde{\sigma}_i \leftarrow g^{x_i^* + y_i^*} \cdot \tilde{\theta}_i \cdot h^{t_i}$
EndFor
$\tilde{\pi}_1 \leftarrow g^{x_5^* + y_5^*} \cdot h^{t_1 t_2} \cdot (\tilde{\theta}_1)^{t_2} \cdot (\tilde{\theta}_2 \Omega)^{t_1}$
$\tilde{\pi}_2 \leftarrow g^{x_6^* + y_6^*} \cdot u^{t_2} \cdot g^{-t_3} \cdot \left(v' \prod_{i=1}^{m} v_i^{\mu_i} \right)^{t_4}$
output $\langle C_{x_i}^*, C_{y_i}^*, \tilde{\sigma}_i \rangle_{(i=1,2,3,4)}, \langle C_{x_5}^*, C_{y_5}^*, \tilde{\pi}_1 \rangle, \langle C_{x_6}^*, C_{y_6}^*, \tilde{\pi}_2 \rangle$
End

In Table 3, we list the numbers of various operations that are needed in an obfuscating transformation. Although the obfuscator looks complicated, it remains in high efficiency because the time-consuming pairing operation is never used and the complexity has nothing to do with the number of signers.

In the next two subsections, we study the "correctness" and "security" of the obfuscator respectively.

Table 3. Efficiency of the obfuscator.

Operation	In Z_n					In G			
	Rand	Add	Mult	Inv	Neg	Rand	Mult	Exp	Inv
Number	24	9	7	0	2	0	2m+29	2m+43	0

"Rand" denotes the operation that generates a random element of the group or ring. "Add" and "Mult" denote the addition and multiplication, respectively. "Neg" denotes the operation that generates an addictive inverse and "Inv" denotes the operation that generates a multiplicative inverse.

doi:10.1371/journal.pone.0131550.t003

4.2. Preserving Functionality

The correctness of an obfuscator Obf requires that, on a circuit C, Obf(C) behaves identically to C on all inputs with probability 1. Formally, this property is called "Preserving Functionality" as described in Definition 1 [34,35].

Definition 1. Preserving Functionality. A PPT machine Obf is a circuit obfuscator for a class of probabilistic circuits $\mathbb{C} = \{\mathbb{C}_\lambda\}_{\lambda \in \mathbb{N}}$ if, for every probabilistic circuit $C \in \mathbb{C}_\lambda$, (2) holds:

$$\Pr[C' \leftarrow \mathrm{Obf}(C) \ : \forall x, \mathrm{StaDiff}(C(x),\ C'(x)) = 0] \ = \ 1 \tag{2}$$

where the statistical difference StaDiff between $C(x)$ and $C'(x)$ is given by (1).

Theorem 1. (Preserving Functionality). Consider any circuit $C_{pub,sk_{\mathrm{ID}},PK_e} \in C_\lambda$ and let circuit $R_{pub,z,PK_e} = \mathrm{Obf}_{EGS}(C_{pub,sk,PK_e})$. On every possible input, the output distributions of C_{pub,sk,PK_e} and R_{pub,z,PK_e} are identical.

Proof. Suppose that $PK_e = (g^a, g^b)$. On an arbitrary message $M \neq NULL$, the output of C_{pub,sk,PK_e} is $(c_i = Enc(\sigma_i, PK_e)_{i\,=\,1,2,3,4}, c_{i+4} = Enc(\pi_i, PK_e)_{i\,=\,1,2})$ where $(\sigma_1, \sigma_2, \sigma_3, \sigma_4, \pi_1, \pi_2) = Sign(M, sk_{\mathrm{ID}})$. Let $sk_{\mathrm{ID}} = (K_1, K_2, K_3)$ and $M = (\mu_1 \ldots \mu_m) \in \{0,1\}^m$. We have

$$c_1 = g^{r_1 a}, g^{s_1 b}, g^{r_1+s_1} \cdot K_1 \cdot h^{t_1} \tag{3}$$

$$c_2 = g^{r_2 a}, g^{s_2 b}, g^{r_2+s_2} \cdot K_2 \cdot h^{t_2} \tag{4}$$

$$c_3 = g^{r_3 a}, g^{s_3 b}, g^{r_3+s_3} \cdot K_3 \cdot \left(v' \prod_{i=1}^{m} v_i^{\mu_i}\right)^s \cdot h^{t_3} \tag{5}$$

$$c_4 = g^{r_4 a}, g^{s_4 b}, g^{r_4+s_4} \cdot g^{-s} \cdot h^{t_4} \tag{6}$$

$$c_5 = g^{r_5 a}, g^{s_5 b}, g^{r_5+s_5} \cdot h^{t_1 t_2} \cdot (K_1)^{t_2} \cdot (K_2 \Omega)^{t_1} \tag{7}$$

$$c_6 = g^{r_6 a}, g^{s_6 b}, g^{r_6+s_6} \cdot u^{t_2} \cdot g^{-t_3} \cdot \left(v' \prod_{i=1}^{m} v_i^{\mu_i}\right)^{t_4} \tag{8}$$

where $s, t_1, t_2, t_3, t_4, r_1, \ldots, r_6, s_1, \ldots, s_6$ are uniformly random elements of Z_n.

Because $R_{pub,z,PK_e} = \mathrm{Obf}_{EGS}(C_{pub,sk,PK_e})$, we have:

$$z = (C_{x_1}, C_{y_1}, C_{x_2}, C_{y_2}, C_{x_3}, C_{y_3}, \tilde{K}_1, \tilde{K}_2, \tilde{K}_3) \tag{9}$$

where

$$\begin{aligned} &\tilde{K}_1 = g^{x_1+y_1} \cdot K_1, \tilde{K}_2 = g^{x_2+y_2} \cdot K_2, \tilde{K}_3 = g^{x_3+y_3} \cdot K_3 \\ &C_{x_1} = g^{ax_1}, C_{y_1} = g^{by_1} \\ &C_{x_2} = g^{ax_2}, C_{y_2} = g^{by_2} \\ &C_{x_3} = g^{ax_3}, C_{y_3} = g^{by_3} \end{aligned} \tag{10}$$

In (10), $x_1, y_1, x_2, y_2, x_3, y_3$ are uniformly random elements of Z_n.

For the same input M, suppose that the output of R_{pub,z,PK_e} is

$$(c'_{i(i=1,2,3,4,5,6)}) = \left(\langle C_{x_i}{}^*, C_{y_i}{}^*, \tilde{\sigma}_i\rangle_{(i=1,2,3,4)}, \langle C_{x_5}{}^*, C_{y_5}{}^*, \tilde{\pi}_1\rangle, \langle C_{x_6}{}^*, C_{y_6}{}^*, \tilde{\pi}_2\rangle\right) \tag{11}$$

Consequently, we have:

$$c_1' = \langle C_{x_1}{}^*, C_{y_1}{}^*, \tilde{\sigma}_1 \rangle = \left\langle (PK_{e,0})^{x_1{}^*} \cdot C_{x_1}, (PK_{e,1})^{y_1{}^*} \cdot C_{y_1}, g^{x_1{}^*+y_1{}^*} \cdot \tilde{K}_1 \cdot h^{t_1{}^*} \right\rangle$$
$$= \langle g^{ax_1{}^*+ax_1}, g^{by_1{}^*+by_1}, g^{x_1{}^*+y_1{}^*+x_1+y_1} \cdot K_1 \cdot h^{t_1{}^*} \rangle = \left\langle g^{a(x_1{}^*+x_1)}, g^{b(y_1{}^*+y_1)}, g^{(x_1{}^*+x_1)+(y_1{}^*+y_1)} \cdot K_1 \cdot h^{t_1{}^*} \right\rangle \quad (12)$$

$$c_2' = \langle C_{x_2}{}^*, C_{y_2}{}^*, \tilde{\sigma}_2 \rangle = \langle g^{a(x_2{}^*+x_2)}, g^{b(y_2{}^*+y_2)}, g^{(x_1{}^*+x_1)+(y_1{}^*+y_1)} \cdot K_1 \cdot h^{t_2{}^*} \rangle \quad (13)$$

$$c_3' = \langle C_{x_3}{}^*, C_{y_3}{}^*, \tilde{\sigma}_3 \rangle = \left\langle g^{ax_1{}^*+ax_1}, g^{by_1{}^*+by_1}, g^{x_1{}^*+y_1{}^*+x_1+y_1} \cdot K_1 \cdot \left(v' \prod_{i=1}^{m} v_i^{\mu_i} \right)^{s^*} \cdot h^{t_3{}^*} \right\rangle$$
$$= \left\langle g^{a(x_3{}^*+x_3)}, g^{b(y_3{}^*+y_3)}, g^{(x_3{}^*+x_3)+(y_3{}^*+y_3)} \cdot K_1 \cdot \left(v' \prod_{i=1}^{m} v_i^{\mu_i} \right)^{s^*} \cdot h^{t_3{}^*} \right\rangle \quad (14)$$

$$c_4' = \langle C_{x_4}{}^*, C_{y_4}{}^*, \tilde{\sigma}_4 \rangle = \langle g^{ax_4{}^*}, g^{by_4{}^*}, g^{x_4{}^*+y_4{}^*} \cdot g^{-s^*} \cdot h^{t_4{}^*} \rangle \quad (15)$$

$$c_5' = \langle C_{x_5}{}^*, C_{y_5}{}^*, \tilde{\pi}_1 \rangle$$
$$= \langle (PK_{e,1})^{x_i{}^*} \cdot C_{x_1} \cdot C_{x_2}, (PK_{e,1})^{y_i{}^*} \cdot C_{y_1} \cdot C_{y_2}, g^{x_5{}^*+y_5{}^*} \cdot h^{t_1{}^*t_2{}^*} \cdot (\tilde{\theta}_1)^{t_2{}^*} \cdot (\tilde{\theta}_2\Omega)^{t_1{}^*} \rangle$$
$$= \langle g^{ax_5{}^*} \cdot g^{ax_1t_2{}^*} \cdot g^{ax_2t_1{}^*}, g^{ay_5{}^*} \cdot g^{ay_1t_2{}^*} \cdot g^{ay_2t_1{}^*}, g^{x_5{}^*+y_5{}^*} \cdot h^{t_1{}^*t_2{}^*} \cdot (\tilde{K}_1)^{t_2{}^*} \cdot (\tilde{K}_2\Omega)^{t_1{}^*} \rangle$$
$$= \langle g^{a(x_5{}^*+x_1t_2{}^*+x_2t_1{}^*)}, g^{b(y_5{}^*+y_1t_2{}^*+y_2t_1{}^*)}, g^{x_5{}^*+y_5{}^*} \cdot h^{t_1{}^*t_2{}^*} \cdot (g^{x_1+y_1} \cdot K_1)^{t_2{}^*} \cdot (g^{x_2+y_2} \cdot K_2\Omega)^{t_1{}^*} \rangle \quad (16)$$
$$= \langle g^{a(x_5{}^*+x_1t_2{}^*+x_2t_1{}^*)}, g^{b(y_5{}^*+y_1t_2{}^*+y_2t_1{}^*)}, g^{x_5{}^*+y_5{}^*} \cdot h^{t_1{}^*t_2{}^*} \cdot g^{x_1t_2{}^*+y_1t_2{}^*} \cdot g^{x_2t_1{}^*+y_2t_1{}^*} \cdot h^{t_1{}^*t_2{}^*} \cdot (K_1)^{t_2{}^*} \cdot (K_2\Omega)^{t_1{}^*} \rangle$$
$$= \langle g^{a(x_5{}^*+x_1t_2{}^*+x_2t_1{}^*)}, g^{b(y_5{}^*+y_1t_2{}^*+y_2t_1{}^*)}, g^{(x_5{}^*+x_1t_2{}^*+x_2t_1{}^*)+(y_5{}^*+y_1t_2{}^*+y_2t_1{}^*)} \cdot h^{t_1{}^*t_2{}^*} \cdot (K_1)^{t_2{}^*} \cdot (K_2\Omega)^{t_1{}^*} \rangle$$

$$c_6' = \left\langle C_{x_6}{}^*, C_{y_6}{}^*, \tilde{\pi}_2 \right\rangle = \langle g^{ax_6{}^*}, g^{by_6{}^*}, g^{x_6{}^*+y_6{}^*} \cdot u^{t_2{}^*} \cdot g^{-t_3{}^*} \cdot \left(v' \prod_{i=1}^{m} v_i^{\mu_i} \right)^{t_4{}^*} \rangle \quad (17)$$

In the above six equations, $s^*, t_1{}^*, t_2{}^*, t_3{}^*, t_4{}^*, x_1{}^*, \cdots, x_6{}^*, y_1{}^*, \cdots, y_6{}^*$ are uniformly random elements of Z_n.

Let $V = v' \prod_{i=1}^{m} v_i^{\mu_i}$, the two tuples of ciphertexts are listed in Table 4.

Clearly, the two tuples of ciphertexts are identically distributed.

If the input is a null message $M = NULL$, both the output of C_{pub,sk,PK_e} and the output of R_{pub,z,PK_e} are (pub, PK_e).

This ends the proof.

Table 4. Comparison of ciphertexts.

i	c_i	c_i'
1	$g^{r_1a}, g^{s_1b}, g^{r_1+s_1} \cdot K_1 \cdot h^{t_1}$	$g^{a(x_1{}^*+x_1)}, g^{b(y_1{}^*+y_1)}, g^{(x_1{}^*+x_1)+(y_1{}^*+y_1)} \cdot K_1 \cdot h^{t_1{}^*}$
2	$g^{r_2a}, g^{s_2b}, g^{r_2+s_2} \cdot K_2 \cdot h^{t_2}$	$g^{a(x_2{}^*+x_2)}, g^{b(y_2{}^*+y_2)}, g^{(x_1{}^*+x_1)+(y_1{}^*+y_1)} \cdot K_1 \cdot h^{t_2{}^*}$
3	$g^{r_3a}, g^{s_3b}, g^{r_3+s_3} \cdot K_3 \cdot V^s \cdot h^{t_3}$	$g^{a(x_3{}^*+x_3)}, g^{b(y_3{}^*+y_3)}, g^{(x_3{}^*+x_3)+(y_3{}^*+y_3)} \cdot K_1 \cdot V^{s^*} \cdot h^{t_3{}^*}$
4	$g^{r_4a}, g^{s_4b}, g^{r_4+s_4} \cdot g^{-s} \cdot h^{t_4}$	$g^{ax_4{}^*}, g^{by_4{}^*}, g^{x_4{}^*+y_4{}^*} \cdot g^{-s^*} \cdot h^{t_4{}^*}$
5	$g^{r_5a}, g^{s_5b}, g^{r_5+s_5} \cdot h^{t_1t_2} \cdot (K_1)^{t_2} \cdot (K_2\Omega)^{t_1}$	$g^{a(x_5{}^*+x_1t_2{}^*+x_2t_1{}^*)}, g^{b(y_5{}^*+y_1t_2{}^*+y_2t_1{}^*)}, g^{(x_5{}^*+x_1t_2{}^*+x_2t_1{}^*)+(y_5{}^*+y_1t_2{}^*+y_2t_1{}^*)} \cdot h^{t_1{}^*t_2{}^*} \cdot (K_1)^{t_2{}^*} \cdot (K_2\Omega)^{t_1{}^*}$
6	$g^{r_6a}, g^{s_6b}, g^{r_6+s_6} \cdot u^{t_2} \cdot g^{-t_3} \cdot V^{t_4}$	$g^{ax_6{}^*}, g^{by_6{}^*}, g^{x_6{}^*+y_6{}^*} \cdot u^{t_2{}^*} \cdot g^{-t_3{}^*} \cdot V^{t_4{}^*}$

doi:10.1371/journal.pone.0131550.t004

4.3. Security Properties

4.3.1. Security notions for the EGS functionality and the obfuscator. Average-Case Secure Virtual Black-box Property (ACVBP) was proposed in [35], and it was extended to ACVBP w.r.t. Dependent Oracles in [34]. The generalization allows distinguishers to have sampling access not only to $<<C>>$ but also to a set of oracles dependent on C.

Definition 2. A circuit obfuscator Obf for $\mathbb{C}$ 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 $\mathcal{D}$ (distinguisher), every polynomial f, all sufficiently large $\lambda \in \mathbb{N}$, and every $z \in \{0,1\}^{\mathrm{poly}(\lambda)}$,

$$\left| \Pr\left[\begin{array}{ll} C \leftarrow \mathbb{C}_\lambda; & \\ C' \leftarrow \mathrm{Obf}(C); & : b = 1 \\ b \leftarrow \mathcal{D}^{<<C,T(C)>>}(C', z) & \end{array} \right] - \Pr\left[\begin{array}{ll} C \leftarrow \mathbb{C}_\lambda; & \\ C'' \leftarrow S^{<<C>>}(1^\lambda, z); & : b = 1 \\ b \leftarrow \mathcal{D}^{<<C,T(C)>>}(C'', z) & \end{array} \right] \right| < \frac{1}{f(\lambda)} \quad (18)$$

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

To the best of our knowledge, in obfuscating sig-then-encrypt functionalities, $T(C)$ is always assigned to the signature function, such as in [34,36,38,41,49,50]. However, we have to investigate the effects of collision attacks from some members of the same group against the proposed obfuscator. In this scenario, the adversary against the obfuscator can get the signing key of a corrupted group member, that is, the adversary can query the *Enroll* oracle on the identity of a corrupted member. Because there are some restrictions on these kinds of queries, we define a set of restricted oracles dependent on C as $\mathcal{R}(C)$. Each element of $\mathcal{R}(C)$ is an oracle-restrictions pair. For example, in this paper $\mathcal{R}(C) = \{(Enroll, id \neq \mathrm{ID})\}$. Conventionally, when $\mathcal{R}(C)$ consists of only one element, we omit the braces. Moreover, we suggest that the restrictions could be written as superscripts of the oracle, e.g., $Enroll^{[id \neq \mathrm{ID}]}$.

Based on the above intuition, we extended ACVBP w.r.t. Dependent Oracles (**Definition 2**) to ACVBP w.r.t. Dependent Oracles and Restricted Dependent Oracles as follows.

Definition 3. A circuit obfuscator Obf for $\mathbb{C}$ satisfies the ACVBP w.r.t. dependent oracle set T and restricted dependent oracle set $\mathcal{R}$ if the following condition holds: There exists a PPT oracle machine S (simulator) such that, for every PPT oracle machine $\mathcal{D}$ (distinguisher), every polynomial f, all sufficiently large $\lambda \in \mathbb{N}$, and every $z \in \{0,1\}^{\mathrm{poly}(\lambda)}$,

$$\left| \begin{array}{l} \Pr\left[\begin{array}{ll} C \leftarrow \mathbb{C}_\lambda; & \\ C' \leftarrow \mathrm{Obf}(C); & : b = 1 \\ b \leftarrow \mathcal{D}^{<<C,T(C),\mathcal{R}(C)>>}(C', z) & \end{array} \right] \\ -\Pr\left[\begin{array}{ll} C \leftarrow \mathbb{C}_\lambda; & \\ C'' \leftarrow S^{<<C>>}(1^\lambda, z); & : b = 1 \\ b \leftarrow \mathcal{D}^{<<C,T(C),\mathcal{R}(C)>>}(C'', z) & \end{array} \right] \end{array} \right| < \frac{1}{f(\lambda)} \quad (19)$$

where $\mathcal{D}^{<<C,\ T(C),\ \mathcal{R}(C)>>}$ means that $\mathcal{D}$ has sampling access to all oracles contained in $T(C)$ and $\mathcal{R}(C)$ in addition to C.

Besides the security notion for the obfuscator in WBAC, we should also provide security notions for the EGS functionality. There is a number of existing security notions of group

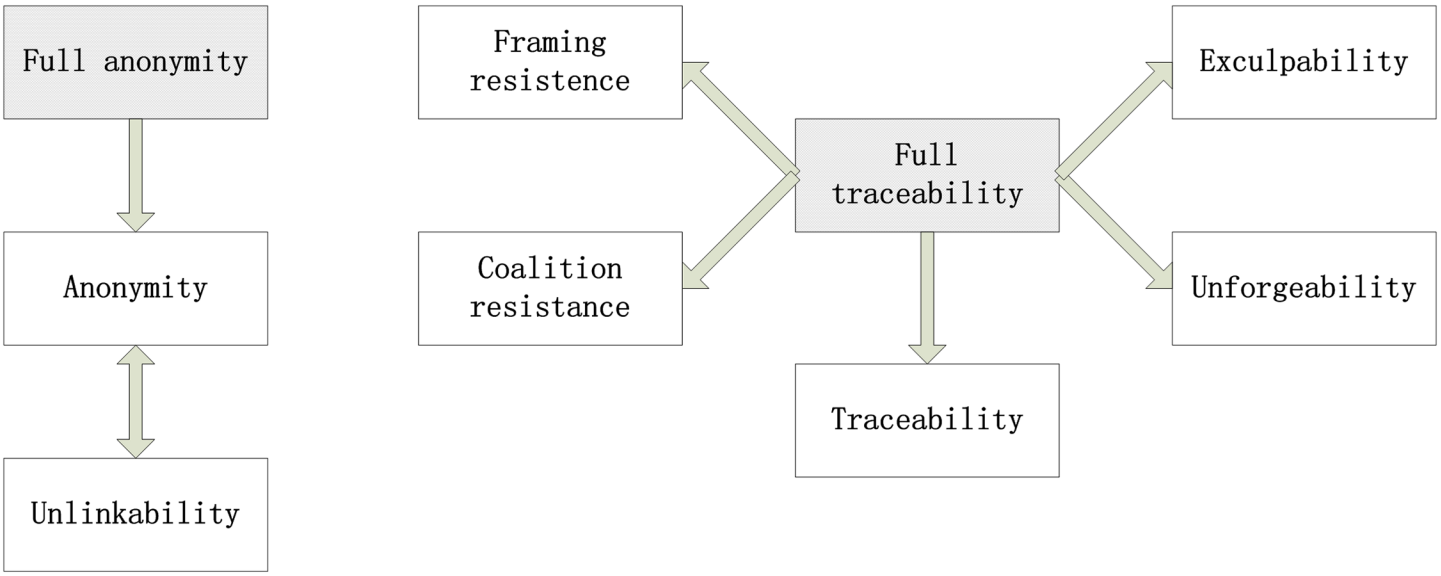


Fig 6. Security notions of group signature schemes.

doi:10.1371/journal.pone.0131550.g006

signature schemes. Fortunately, as shown in Fig 6, there are two cores that imply other security notions as it was discussed in [2]. Hence, we focus on the full traceability (FT) and full anonymity (FA). Note that we use the CPA-full-anonymity[46] instead of the CCA2-full-anonymity[2].

We formally define full-traceability (FT w.r.t. EGS Functionality) using the following experiment.

$Exp^{Trace}_{EGS,\mathcal{F}}(\lambda, k, m)$
Begin
  $(pub, priv) = ((n, \hat{e}, G, G_T, PP), (MK, TK)) \leftarrow Setup(1^{\lambda}, 1^{k}, 1^{m})$
  $(PK_e, SK_e) \leftarrow EKGen(pub)$
  $st \leftarrow (pub, TK, PK_e); \Theta \leftarrow \emptyset; L \leftarrow \emptyset; \Gamma \leftarrow \varepsilon; count \leftarrow true$
While $(cont = true)$ do
    $(cont, st, id) \xleftarrow{\$} \mathcal{F}^{\langle\langle Sign_{sk_{ID}}(\cdot)\rangle\rangle}(choose, st, \Gamma)$
    $L \leftarrow L \cup \{(M, \mathrm{ID}) | Sign_{sk_{ID}}(M)$ has been queried by $\mathcal{F}\}$
    IF $(cont = true)$
      $\Theta \leftarrow \Theta \cup \{id\}$; $\Gamma \leftarrow Enroll(pub, MK, id)$
  EndWhile
  $(M, \sigma) \xleftarrow{\$} \mathcal{F}^{\langle\langle Sign_{sk_{ID}}(\cdot)\rangle\rangle}(guess, st)$
  IF $(0 = Verify(pub, \sigma, M))$ return 0;
  IF $(\perp = Open(pub, TK, \sigma, M))$ return 0;
  IF $(\exists id \in [0, \ldots, 2^k - 1]$, s.t. $id = Open(TK, \sigma, M) \wedge id \notin \Theta \wedge (M, \mathrm{ID}) \notin L)$
    return 0
  Else
    return 1
End

Definition 4. (FT w.r.t. EGS Functionality). Let (*Setup*, *Enroll*, *Sign*, *Verify*) and (*EKGen*, *Enc*, *Dec*) be a pair of group signature and public key encryption schemes. The group signature scheme is FT w.r.t. the EGS functionality if the following condition holds: For every PPT oracle machine $\mathcal{F}$ (adversary), every polynomial f, all sufficiently large $\lambda \in \mathbb{N}$, and every $st \in \{0,1\}^{\mathrm{poly}(\lambda)}$,

$$\Pr\left[Exp^{Trace}_{EGS,\mathcal{F}}(\lambda, k, m)\right] < \frac{1}{f(\lambda)} \tag{20}$$

We formally define full-anonymity (FA w.r.t. EGS Functionality) using the following experiment.

$Exp^{CPA-Anon-b}_{EGS,\mathcal{F}}(\lambda, k, m)$
Begin
  $(pub, priv) = ((n, \hat{e}, G, G_T, PP), (MK, TK)) \leftarrow Setup(1^{\lambda}, 1^{k}, 1^{m})$
  $(PK_e, SK_e) \leftarrow EKGen(pub)$
  $SK \leftarrow \emptyset$; $id \leftarrow 0$
  While $(id < 2^{k})$ do
    $SK \leftarrow SK \cup Enroll(pub, MK, id)$; $id \leftarrow id + 1$
  EndWhile
  $(st, id_0, id_1, M) \xleftarrow{\$} \mathcal{F}(choose, pub, SK, PK_e)$
  $\sigma \leftarrow Sign_{sk_{id_b}}(M)$
  $d \xleftarrow{\$} \mathcal{F}^{\langle\langle Sign_{sk}(\cdot)\rangle\rangle}(guess, st, \sigma)$
  return d
End

Definition 5. (FA w.r.t. EGS Functionality). Let (*Setup*, *Enroll*, *Sign*, *Verify*) and (*EKGen*, *Enc*, *Dec*) be a pair of group signature and public key encryption schemes. The group signature scheme is full anonymous w.r.t. the EGS functionality if the following condition holds: For every PPT oracle machine $\mathcal{F}$ (adversary), every polynomial f, all sufficiently large $\lambda \in \mathbb{N}$, and every $st \in \{0,1\}^{\mathrm{poly}(\lambda)}$,

$$\left|\Pr\left[d \leftarrow Exp^{CPA-Anon-b}_{EGS,\mathcal{F}} : b = d\right] - \frac{1}{2}\right| < \frac{1}{f(\lambda)} \tag{21}$$

Next, we consider a pair of stronger security notions, which require that the group signature scheme is still secure even when the adversary is given an obfuscated circuit. The following experiments are used to describe the strengthened definitions of full-traceability and full-anonymity respectively.

$Exp^{Trace}_{EGS,\mathrm{Obf}_{EGS},\mathcal{F}}(\lambda, k, m)$
Begin
  $(pub, priv) = ((n, \hat{e}, G, G_T, PP), (MK, TK)) \leftarrow Setup(1^{\lambda}, 1^{k}, 1^{m})$
  $(PK_e, SK_e) \leftarrow EKGen(pub)$
  $st \leftarrow (pub, TK, PK_e)$; $SK \leftarrow \emptyset$; $\Theta \leftarrow \emptyset$; $L \leftarrow \emptyset$; $\Gamma \leftarrow \varepsilon$; $cont \leftarrow true$
  While $(id < 2^{k})$ do
    $SK \leftarrow SK \cup Enroll(pub, MK, id)$; $id \leftarrow id + 1$
  EndWhile
  While $(cont = true)$ do
    $(cont, st, id) \xleftarrow{\$} \mathcal{F}^{\langle\langle Sign_{sk_{\mathrm{ID}}}(\cdot), O_{EGS}(\cdot)\rangle\rangle}(choose, st, \Gamma)$
    $L \leftarrow L \cup \{(M, \mathrm{ID}) | Sign_{sk_{\mathrm{ID}}}(M)$ has been queried by $\mathcal{F}\}$
    IF $(cont = true)$
      $\Theta \leftarrow \Theta \cup \{id\}$; $\Gamma \leftarrow Enroll(pub, MK, id)$
  EndWhile
  $(M, \sigma) \xleftarrow{\$} \mathcal{F}^{\langle\langle Sign_{sk_{\mathrm{ID}}}(\cdot), O_{EGS}(\cdot)\rangle\rangle}(guess, st)$

IF(0 = $Verify(pub, \sigma, M)$) return 0;
IF($\perp = Open(pub, TK, \sigma, M)$) return 0;
IF($\exists id \in [0, \ldots, 2^k - 1]$, s.t. $id = Open(TK, \sigma, M) \wedge id \notin \Theta \wedge (M, \mathrm{ID}) \notin L$)
return 0
Else
return 1
End
Algorithm $O_{EGS}(id)$
Begin
Extract sk_{id} from SK
return $\mathrm{Obf}_{EGS}(C_{pub,sk_{id},PK_e})$
End
$Exp^{CPA-Anon-b}_{EGS,\mathrm{Obf}_{EGS},\mathcal{F}}(\lambda, k, m)$
Begin
$(pub, priv) = ((n, \hat{e}, G, G_T, PP), (MK, TK)) \leftarrow Setup(1^\lambda, 1^k, 1^m)$
$(PK_e, SK_e) \leftarrow EKGen(pub)$
$SK \leftarrow \emptyset$; $id \leftarrow 0$
While($id < 2^k$) do
$SK \leftarrow SK \cup Enroll(pub, MK, id)$; $id \leftarrow id + 1$
EndWhile
$(st, id_0, id_1, M) \xleftarrow{\$} \mathcal{F}(choose, pub, SK, PK_e)$
$\sigma \leftarrow Sign_{sk_{id_b}}(M)$
$d \xleftarrow{\$} \mathcal{F}^{\langle\langle Sign_{sk}(), O_{EGS}()\rangle\rangle}(guess, st, \sigma)$
return d
End
Algorithm $O_{EGS}(id)$
Begin
Extract sk_{id} from SK
return $\mathrm{Obf}_{EGS}(C_{pub,sk_{id},PK_e})$
End

Now we give definitions of full-traceability (FT) and full-anonymity (FA) w.r.t. EGS Obfuscator as follows.

Definition 6. (FT w.r.t. EGS Obfuscator). Let (*Setup*, *Enroll*, *Sign*, *Verify*) and (*EKGen*, *Enc*, *Dec*) be a pair of group signature and public key encryption schemes. The group signature scheme is FT w.r.t. Obf_{EGS} if the following condition holds: For every PPT oracle machine $\mathcal{F}$ (adversary), every polynomial f, all sufficiently large $\lambda \in \mathbb{N}$, and every $st \in \{0,1\}^{\mathrm{poly}(\lambda)}$,

$$\Pr\left[Exp^{Trace}_{EGS,\mathrm{Obf}_{EGS},\mathcal{F}}(\lambda, k, m)\right] < \frac{1}{f(\lambda)} \tag{22}$$

Definition 7. (FA w.r.t. EGS Obfuscator). Let (*Setup*, *Enroll*, *Sign*, *Verify*) and (*EKGen*, *Enc*, *Dec*) be a pair of group signature and public key encryption schemes. The group signature scheme is FA w.r.t. Obf_{EGS} if the following condition holds: For every PPT oracle machine $\mathcal{F}$ (adversary), every polynomial f, all sufficiently large $\lambda \in \mathbb{N}$, and every $st \in \{0,1\}^{\mathrm{poly}(\lambda)}$,

$$\left|\Pr\left[d \leftarrow Exp^{CPA-Anon-b}_{EGS,\mathrm{Obf}_{EGS},\mathcal{F}} : b = d\right] - \frac{1}{2}\right| < \frac{1}{f(\lambda)} \tag{23}$$

Definition 8. A circuit obfuscator Obf for $\mathbb{C}$ is rerandomizable (RR) w.r.t. dependent oracle set T and restricted dependent oracle set $\mathcal{R}$ if the following condition holds: There exists a PPT oracle machine *RR* such that, for every PPT oracle machine $\mathcal{D}$ (distinguisher), every

polynomial f, all sufficiently large $\lambda \in \mathbb{N}$, and every $z \in \{0,1\}^{\mathrm{poly}(\lambda)}$, when $C \in \mathbb{C}_\lambda$ and $C' \leftarrow \mathrm{Obf}(C)$, (24) holds.

$$\left| \begin{array}{l} \Pr\left[b \leftarrow \mathcal{D}^{<<C,T(C),\mathcal{R}(C)>>}(C', z) \quad : b = 1 \right] \\ -\Pr\left[\begin{array}{l} C'' \leftarrow RR(C'); \\ \\ b \leftarrow \mathcal{D}^{<<C,T(C),\mathcal{R}(C)>>}(C'', z) \end{array} \quad : b = 1 \right] \end{array} \right| < \frac{1}{f(\lambda)} \tag{24}$$

In (24), $\mathcal{D}^{<<C,\ T(C),\ \mathcal{R}(C)>>}$ means that $\mathcal{D}$ has sampling access to all oracles contained in $T(C)$ and $\mathcal{R}(C)$ in addition to C.

Note that if a circuit obfuscator Obf for $\mathbb{C}$ is rerandomizable w.r.t. dependent oracle set T and restricted dependent oracle set $\mathcal{R}$ and Obf also satisfies the ACVBP w.r.t. dependent oracle set T and restricted dependent oracle set $\mathcal{R}$, we say that Obf is RR&ACVBP w.r.t. dependent oracle set T and restricted dependent oracle set $\mathcal{R}$.

There are six new security notions that are proposed in this section. Relationships among the proposed security notions and known security notions are shown in Fig 7. In Fig 7, the arrows denote the "imply" relationships. Most of the "imply" relationships are easy to verify so we omit the detail analysis, except the complex one that is investigated in Theorem 3.

As illustrated in Fig 7, some new security notions are equal to old ones. Especially, Definition 5 (FA w.r.t. EGS Obfuscator), Definition 7 (FA w.r.t. EGS functionality) and the CAP-full-anonymity (in [46]) are equivalent. Hence, from the practice point of view, it seems that Definition 5 and Definition 7 are somewhat useless because in fact depict the same security of CAP-full-anonymity. However, to provide these security notions and investigate them are necessary, both to acquire the result and to fulfill the completeness of theory.

4.3.2. The main security theorem.

Theorem 2. The proposed obfuscator Obf_{EGS} for the proposed EGS functionality satisfies ACVBP w.r.t. dependent oracle $T(C) = Sign_{sk_{\mathrm{ID}}}$ and restricted dependent oracle $\mathcal{R}(C) = Enroll_{MK}^{[id \neq \mathrm{ID}]}$.

Proof. We have $C = C_{pub,sk_{\mathrm{ID}},PK_e}$, $T(C) = Sign_{sk_{\mathrm{ID}}}$ and $\mathcal{R}(C) = Enroll_{MK}^{[id \neq \mathrm{ID}]}$. Then we define a pair of probabilities in (25) and (26). They are the probabilities that $\mathcal{D}^{<<C,\ T(C),\ \mathcal{R}(C)>>}$ outputs 1, given the real and simulated distributions, respectively. $sk_{\mathrm{ID}} = (K_1, K_2, K_3)$ is encrypted in the real distribution while $(K_1', K_2', K_3') \in_{\$} G^3$ is encrypted in the simulated distribution. It is the only difference.

Let

$$\begin{aligned} &\Pr_{Nick} \\ &= \Pr\left[\begin{array}{l} (pub, priv) = ((n, \hat{e}, G, G_T, PP), (MK, TK)) \leftarrow Setup(1^\lambda, 1^k, 1^m) \\ (PK_e, SK_e) \xleftarrow{\$} EKGen(pub) \\ sk_{\mathrm{ID}} = (K_1, K_2, K_3) \xleftarrow{\$} Enroll(pub, MK, \mathrm{ID}) \\ C' \leftarrow \mathrm{Obf}_{EGS}(C_{pub,sk_{\mathrm{ID}},PK_e}) \\ b \leftarrow \mathcal{D}^{<<C_{pub,sk_{\mathrm{ID}},PK_e}(),Sign_{sk_{\mathrm{ID}}}(),Enroll_{MK}^{[id \neq \mathrm{ID}]}()>>}(C') \end{array} \quad : b = 1 \right] \end{aligned} \tag{25}$$

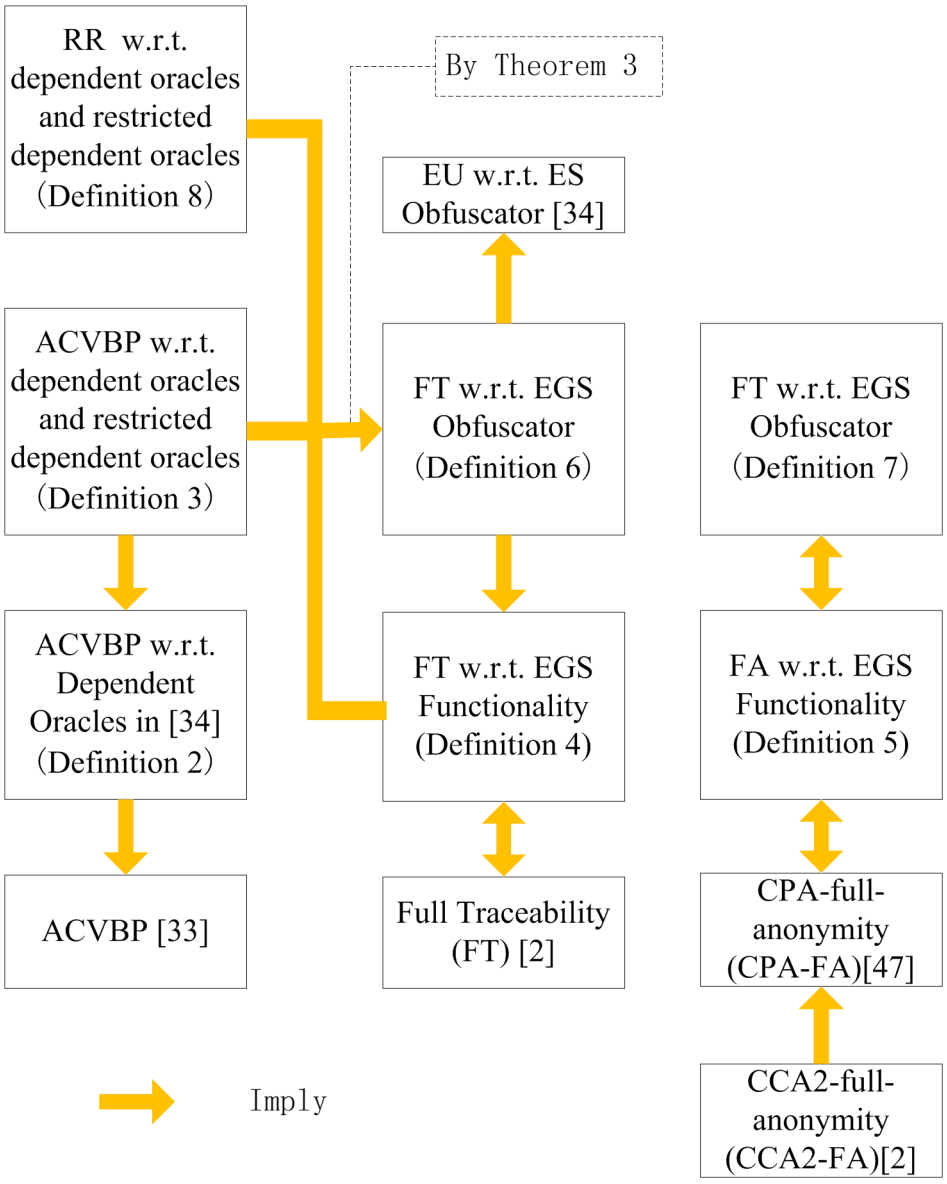


Fig 7. Relationships among the proposed security notions and known security notions.

doi:10.1371/journal.pone.0131550.g007

and

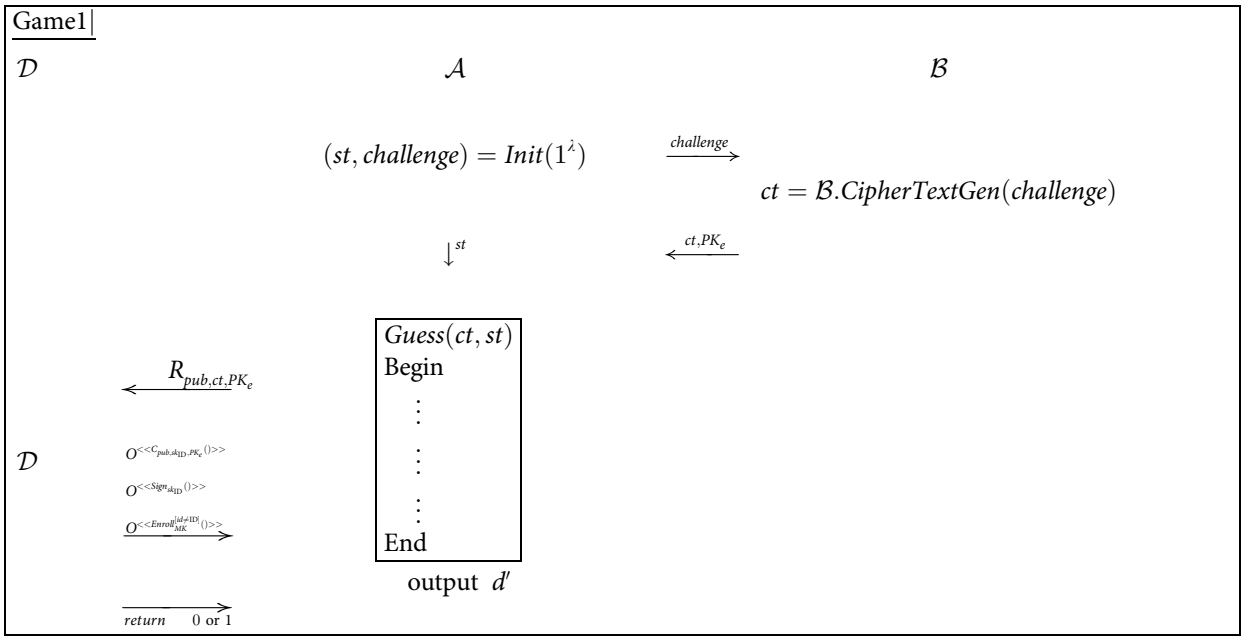
$$\mathrm{Pr}_{Junk} = \Pr\left[\begin{array}{ll} (pub, priv) = ((n, \hat{e}, G, G_T, PP), (MK, TK)) \leftarrow Setup(1^{\lambda}, 1^{k}, 1^{m}) & \\ (PK_e, SK_e) \xleftarrow{\$} EKGen(pub) & \\ sk_{\mathrm{ID}} = (K_1, K_2, K_3) \xleftarrow{\$} Enroll(pub, MK, \mathrm{ID}) & : b = 1 \\ C'' \leftarrow Sim^{<<C_{pub,sk_{\mathrm{ID}},PK_e}()>>}() & \\ b \leftarrow \mathcal{D}^{<<C_{pub,sk_{\mathrm{ID}},PK_e}(),Sign_{sk_{\mathrm{ID}}}(),Enroll_{MK}^{[id\neq \mathrm{ID}]}()>>}(C'') & \end{array}\right] \quad (26)$$

We construct a simulator *Sim* that works as follows

$Sim^{\langle\langle C_{pub,sk_{\mathrm{ID}},PK_e}()\rangle\rangle}()$
$(pub, PK_e) \leftarrow C_{pub,sk_{\mathrm{ID}},PK_e}()$
$(n, \hat{e}, G, G_T, PP) \leftarrow pub$
$(g^a, g^b) \leftarrow PK_e$
$x_1, y_1 \xleftarrow{\$} Z_n, x_2, y_2 \xleftarrow{\$} Z_n, x_3, y_3 \xleftarrow{\$} Z_n$
$K_1', K_2', K_3' \xleftarrow{\$} G$
$\tilde{K}_1' \leftarrow g^{x_1+y_1} \cdot K_1', \tilde{K}_2' \leftarrow g^{x_2+y_2} \cdot K_2', \tilde{K}_3' \leftarrow g^{x_3+y_3} \cdot K_3'$
$C_{x_1}' \leftarrow g^{ax_1}, C_{y_1}' \leftarrow g^{by_1}$
$C_{x_2}' \leftarrow g^{ax_2}, C_{y_2}' \leftarrow g^{by_2}$
$C_{x_3}' \leftarrow g^{ax_3}, C_{y_3}' \leftarrow g^{by_3}$
$Junk \leftarrow (C_{x_1}', C_{y_1}', C_{x_2}', C_{y_2}', C_{x_3}', C_{y_3}', \tilde{K}_1', \tilde{K}_2', \tilde{K}_3')$
output $R_{pub,Junk,PK_e}()$ that works the same as $R_{pub,z,PK_e}()$

For contradiction, assume that the probability that a distinguisher $\mathcal{D}^{<<C_{pub,sk_{\mathrm{ID}},PK_e}(),Sign_{sk_{\mathrm{ID}}}(),Enroll_{MK}^{[id\neq \mathrm{ID}]}()>>}$ can distinguish between C' and C'' is not negligible. That is, the difference between the following two probabilities is not negligible. Without loss of generosity, we suppose that $\Pr_{Nick}$-$\Pr_{Junk} = \delta > 0$.

Then an adversary pair $(\mathcal{A}, \mathcal{B})$ which breaks the indistinguishability of the linear encryption scheme is constructed as follows. $\mathcal{A}$ produces a plaintext pair $\langle sk_{\mathrm{ID}}, sk' \rangle$, and the associated public setting and some global parameters of the asymmetric encryption scheme using $\mathcal{A}.Init$ and plays the following security game with $\mathcal{D}$ and $\mathcal{B}$.



We list the usage of the algorithms that are used in Game1, i.e., $\mathcal{A}.Init$, $\mathcal{A}.OC$, $\mathcal{A}.OS$, $\mathcal{A}.OE$, $\mathcal{A}.Guess$, and $\mathcal{B}.CipherTextGen$, in Table 5.

The descriptions of the algorithms are as follows.

Table 5. The algorithms in Game1.

Algorithm	Usage
$\mathcal{A}.Init$	Initiate the parameter.
$\mathcal{A}.OC$	Reply the $O^{<<C_{pub,sk_{ID},PK_e}()>>}$ queries from $\mathcal{D}$.
$\mathcal{A}.OS$	Reply the $O^{<<Sign_{sk_{ID}}()>>}$ queries from $\mathcal{D}$.
$\mathcal{A}.OE$	Reply the $O^{<<Enroll|_{MK}^{|id\neq ID|}()>>}$ queries from $\mathcal{D}$.
$\mathcal{A}.Guess$	Guess the value of d in $\mathcal{B}.CipherTextGen$
$\mathcal{B}.CipherTextGen$	Generate the challenge ciphertext.

doi:10.1371/journal.pone.0131550.t005

Algorithm $\mathcal{A}.Init(1^{\lambda})$

Begin
$(pub, priv) = ((n, \hat{e}, G, G_T, PP), (MK, TK)) \leftarrow Setup(1^{\lambda}, 1^{k}, 1^{m})$
$sk_{\mathrm{ID}} = (K_1, K_2, K_3) \xleftarrow{\$} Enroll(pub, MK, \mathrm{ID})$
$sk' = (K_1', K_2', K_3') \xleftarrow{\$} G$
$challenge \leftarrow (sk_{\mathrm{ID}}, sk', n, \hat{e}, G, G_T, g)$
$st \leftarrow (pub, priv)$
output $(st, challenge)$
End

Remark 6. We suppose that $\mathcal{A}$ generates the system parameters honestly, that is, $\mathcal{A}$ does not set any "backdoor" in the parameters. Otherwise, the system parameters could be generated by a trusted third party.

Remark 7. After $\mathcal{A}$ is initialized, *pub*,ID and sk_{ID} are private "member variables" of $\mathcal{A}$.

Algorithm $\mathcal{A}.OS(M)$

Begin
output $Sign_{sk_{\mathrm{ID}}}(M)$
End

Algorithm $\mathcal{A}.OS(id)$

Begin
IF(ID = id) output $\perp$
Else output $Enroll(pub, M, id)$
End

Algorithm $\mathcal{B}$.CipherTextGen(challenge)

Begin
$(sk, sk', n, \hat{e}, G, G_T, g) \leftarrow challenge$
$(PK_e, SK_e) \leftarrow AsymEnc.KGen(n, \hat{e}, G, G_T, g)$
$d \xleftarrow{\$} \{0, 1\}$
$x_1, y_1 \xleftarrow{\$} Z_n, x_2, y_2 \xleftarrow{\$} Z_n, x_3, y_3 \xleftarrow{\$} Z_n$
$C_{x_1} \leftarrow g^{ax_1}, C_{y_1} \leftarrow g^{by_1}$
$C_{x_2} \leftarrow g^{ax_2}, C_{y_2} \leftarrow g^{by_2}$
$C_{x_3} \leftarrow g^{ax_3}, C_{y_3} \leftarrow g^{by_3}$
IF(d = 0)
$(K_1, K_2, K_3) \leftarrow sk$
Else
$(K_1, K_2, K_3) \leftarrow sk'$

$\tilde{K}_1 \leftarrow g^{x_1+y_1} \cdot K_1, \tilde{K}_2 \leftarrow g^{x_2+y_2} \cdot K_2, \tilde{K}_3 \leftarrow g^{x_3+y_3} \cdot K_3$
$ct \leftarrow (C_{x_1}, C_{y_1}, C_{x_2}, C_{y_2}, C_{x_3}, C_{y_3}, \tilde{K}_1, \tilde{K}_2, \tilde{K}_3)$
output ct, PK_e
End

Algorithm $\mathcal{A}.Guess(ct, st)$

Begin
$((n, \hat{e}, G, G_T, PP), (MK, TK)) = (pub, priv) \leftarrow st$
Generate C_{pub,sk_{ID},PK_e} according to the EGS functionality;
Generate R_{pub,ct,PK_e} that works the same as R_{pub,z,PK_e} that is generated by Obf_{EGS};
IF$(1 = \mathcal{D}^{\langle\langle C_{pub,sk_{ID},PK_e}(), Sign_{sk_{ID}}()\rangle\rangle}(R_{pub,ct,PK_e}))$
$d' \leftarrow 0$
Else
$d' \xleftarrow{\$} \{0,1\}$
output d'
End

Note that if $d = 1$, R_{pub,ct,PK_e} is the same as $R_{pub,Junk,PK_e}()$ which is generated by *Sim*, otherwise, R_{pub,ct,PK_e} is really a valid output of Obf_{EGS}.

Algorithm $\mathcal{A}.OC(M)$

Begin
//Compute $EGS_{pub,sk_{ID},PK_e}(M)$
output $C_{pub,sk_{ID},PK_e}(M)$
End

We compute $\Pr[d' = 0 \wedge d = 0]$ and $\Pr[d' = 1 \wedge d = 1]$ in the security game as follows.

$$\begin{aligned}
&\Pr[d' = 0 \wedge d = 0] \\
&= \Pr[d = 0] \cdot \Pr\left[1 \leftarrow \mathcal{D}^{<<C_{pub,sk_{ID},PK_e}(),Sign_{sk_{ID}}(),Enroll_{MK}^{[id\neq ID]}()>>}(R_{pub,ct,PK_e}) | d = 0\right] \\
&\quad + \Pr[d = 0] \cdot \left(\frac{1}{2} \cdot \Pr\left[1 \nleftarrow \mathcal{D}^{<<C_{pub,sk_{ID},PK_e}(),Sign_{sk_{ID}}(),Enroll_{MK}^{[id\neq ID]}()>>}(R_{pub,ct,PK_e}) | d = 0\right]\right) \\
&= \frac{1}{2} \cdot \Pr_{Nice} + \frac{1}{2} \cdot \left(\frac{1}{2} \cdot (1 - \Pr_{Nice})\right) \\
&= \frac{\Pr_{Nice} + 1}{4}
\end{aligned} \tag{27}$$

$$\begin{aligned}
&\Pr[d' = 1 \wedge d = 1] \\
&= \Pr[d = 1] \cdot \left(\frac{1}{2} \cdot \Pr[1 \nleftarrow \mathcal{D}^{<<C_{pub,sk_{ID},PK_e}(),Sign_{sk_{ID}}(),Enroll_{MK}^{[id\neq ID]}()>>}(R_{pub,ct,PK_e}) | d = 1]\right) \\
&= \frac{1}{2} \cdot \frac{1}{2} \cdot (1 - \Pr_{Junk}) \\
&= \frac{1 - \Pr_{Junk}}{4}
\end{aligned} \tag{28}$$

Table 6. Some of the scenarios that Definition 3 should be used.

Cryptosystem	Restricted oracle
Identity-based cryptosystem	Extract the private key of an identity $id \neq ID$
Forward-secure cryptosystem	Get the private key of a time period $t' < t$
t, N-key-insulated cryptosystem	Get the user keys of at most t time periods
t, N-threshold cryptosystem	Get at most t-1 pieces of the shared secret

doi:10.1371/journal.pone.0131550.t006

Finally, the advantage of $\mathcal{A}$ (as an adversary in a chosen-plaintext attack against the linear encryption scheme *LE*) can be calculated as follows.

$$\begin{aligned} & Adv_{\mathcal{A}}^{IND-CPA,LE} \\ & = 2 \cdot \Pr[d' = d] - 1 = 2 \cdot (\Pr[d' = 0 \wedge d = 0] + \Pr[d' = 1 \wedge d = 1]) - 1 \\ & = 2 \cdot \left(\frac{\Pr_{Nice} + 1}{4} + \frac{1 - \Pr_{Junk}}{4} \right) - 1 = 2 \cdot \left(\frac{1}{2} + \frac{\Pr_{Nice} - \Pr_{Junk}}{4} \right) - 1 = \frac{\Pr_{Nice} - \Pr_{Junk}}{4} \\ & = \frac{\delta}{2} \end{aligned} \tag{29}$$

Recall that $\delta = \Pr_{Nick}\text{-}\Pr_{Junk}$, if δ is non-negligible, so is $Adv_{\mathcal{A}}^{IND-CPA,LE}$. This contradicts the security property (i.e., semantically secure against chosen-plaintext attacks) of the linear encryption scheme based on the decisional linear assumption. This ends the proof.

Remark 8. By a natural extension on the proof of the security of the ElGamal encryption scheme, the linear encryption scheme is semantically secure against chosen-plaintext attacks, assuming the decisional linear assumption holds [46].

Remark 9. Definition 3 (ACVBP w.r.t. Dependent Oracles and Restricted Dependent Oracles) fits for many application scenarios. Examples are shown in Table 6. Moreover, the proof of Theorem 3 does not work under the ACVBP or ACVBP w.r.t. Dependent Oracles.

4.3.3. FT and FA w.r.t. the proposed EGS Obfuscator. We prove the relationship between Definition 3, 4, 6 and 8 which is already shown in Fig 7 as follows.

Theorem 3. If an obfuscator Obf_{EGS} for a EGS functionality satisfies RR&ACVBP w.r.t. dependent oracle $T(C) = Sign_{sk_{\mathrm{ID}}}$ and restricted dependent oracle $\mathcal{R}(C) = Enroll_{MK}^{[id \neq \mathrm{ID}]}$, then the FT w.r.t. EGS Functionality implies the FT w.r.t. EGS Obfuscator.

Proof. Suppose that a group signature scheme is FT w.r.t. the EGS functionality but not FT w.r.t. Obf_{EGS}. There exists a PPT oracle machine $\mathcal{F}$ (forgery) and $Q_{\mathrm{Obf}} \in \mathbb{N}$ such that $Adv_{\mathcal{F}}^{(Q_{\mathrm{Obf}})} = \Pr[Exp_{EGS,\mathrm{Obf}_{EGS},\mathcal{F}}^{Trace}(\lambda, k, m)]$ is not a negligible value where the superscript Q_{Obf} implies that $\mathcal{F}$ queries the oracle $\mathrm{O}_{EGS}(\cdot)$ at most Q_{Obf} times.

We denote the maximum advantage of the forgery $\mathcal{F}$ which can query the oracle $\mathrm{O}_{EGS}(\cdot)$ at most $q_{\mathrm{Obf}}(1 \leq q_{\mathrm{Obf}} \leq Q_{\mathrm{Obf}})$ times in the experiment $Exp_{EGS,\mathrm{Obf}_{EGS},\mathcal{F}}^{Trace}(\lambda, k, m)$ as $Adv_{\mathcal{F}}^{(q_{\mathrm{Obf}})}$. Clearly, we have

$$Adv_{\mathcal{F}}^{(Q_{\mathrm{Obf}})} \geq Adv_{\mathcal{F}}^{(Q_{\mathrm{Obf}}-1)} \geq \cdots \geq Adv_{\mathcal{F}}^{1} \geq 0 \tag{30}$$

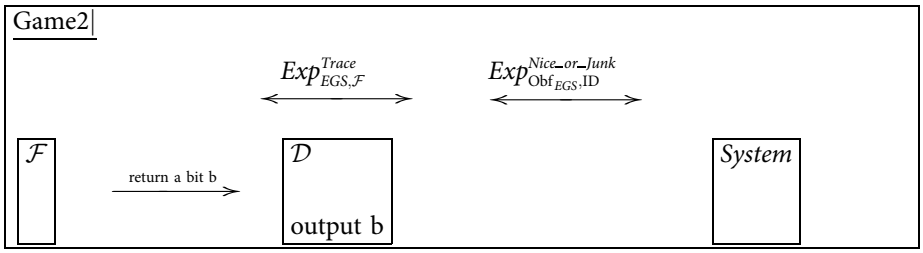
Therefore, $\exists q'_{\mathrm{Obf}}(1 \leq q'_{\mathrm{Obf}} \leq Q_{\mathrm{Obf}})$, s.t.

$$Adv_{\mathcal{F}}^{(q'_{\mathrm{Obf}})} - Adv_{\mathcal{F}}^{(q'_{\mathrm{Obf}}-1)} \geq \frac{Adv_{\mathcal{F}}^{(Q_{\mathrm{Obf}})}}{Q_{\mathrm{Obf}}} \tag{31}$$

We design the experiment $Exp^{Nice\_or\_Junk}_{\mathrm{Obf}_{EGS},\mathrm{ID}}$ as follows. In the experiment, the simulator *Sim* works the same as in the proof of Theorem 2.

$Exp^{Nice\_or\_Junk}_{\mathrm{Obf}_{EGS},\mathrm{ID}}$
Begin
$(pub, priv) = ((n, \hat{e}, G, G_T, PP), (MK, TK)) \leftarrow Setup(1^{\lambda}, 1^{k}, 1^{m})$
$(PK_e, SK_e) \xleftarrow{\$} EKGen(pub)$
$sk_{\mathrm{ID}} = (K_1, K_2, K_3) \xleftarrow{\$} Enroll(pub, MK, \mathrm{ID})$
$C^{(0)} \leftarrow Sim^{\langle\langle C_{pub,sk_{\mathrm{ID}},PK_e}()\rangle\rangle}(); C^{(1)} \leftarrow \mathrm{Obf}_{EGS}(C_{pub,sk_{\mathrm{ID}},PK_e})$
$coin \xleftarrow{\$} \{0,1\}$
$C^* = C^{(coin)}$
$b \leftarrow \mathcal{D}^{\langle\langle C_{pub,sk_{\mathrm{ID}},PK_e}(),Sign_{sk_{\mathrm{ID}}}(),Enroll^{[id\neq \mathrm{ID}]}_{MK}()\rangle\rangle}(C^*)$
End

Furthermore, we design a security game as follows.



We list the usage of the algorithms of $\mathcal{D}$ that are used in Game2, i.e., $\mathcal{D}.Init()$, $\mathcal{D}.Sig(id, M)$, $\mathcal{D}.Answer\_O_{EGS}(id)$ and $\mathcal{D}.RR$, in Table 7.

The existence of $\mathcal{D}.RR$ is guaranteed by the hypothesis. The other three algorithms work as follows.

Algorithm $\mathcal{D}.Init()$

Begin
  While($id<2^k$) do
    IF ($id \neq \mathrm{ID}$)
      $SK[id] \leftarrow Enroll^{[id\neq \mathrm{ID}]}_{MK}(id)$
    $id \leftarrow id + 1$
  EndWhile
End

Algorithm $\mathcal{D}.Sig(id, M)$

Begin
  IF ($id = ID$)
    output $Sign_{sk_{\mathrm{ID}}}(M)$ //Get the output via oracle query
  Else

Table 7. The algorithms of $\mathcal{D}$ in the Game2.

Algorithm	Usage
$\mathcal{D}.Init()$	Initiate the values of private keys except for $id = ID$.
$\mathcal{D}.Answer\_O_{EGS}(id)$	Reply the $O^{<<O_{EGS}(id)>>}$ queries from the forgery $\mathcal{F}$.
$\mathcal{D}.Sig(id, M)$	Reply the $O^{<<Sign_{sk_{ID}}()>>}$ queries from the forgery $\mathcal{F}$.
$\mathcal{D}.RR(C)$	Re-randomize the input obfuscated circuit C.

doi:10.1371/journal.pone.0131550.t007

```
    output Sign_{sk[id]} (M) //Compute the output directly
End
```

Algorithm $\mathcal{D}$.Answer_O_{EGS}(id)

```
Begin
  IF (id = ID)
    D.RR(C*)
  Else
    output Obf_EGS(C_{pub,SK[id],PK_e})
End
```

Let $Adv_{\mathcal{F}}^{(q'_{\mathrm{Obf}})-Junk}$ be the probability of $\mathcal{F}$ (and $\mathcal{D}$) outputs 1 when $coin = 0$ and $\mathcal{F}$ queries $O^{<<O_{EGS}(id)>>}$ at most q'_{Obf} times. Clearly, we have

$$Adv_{\mathcal{F}}^{(q'_{\mathrm{Obf}})-Junk} \leq Adv_{\mathcal{F}}^{(q'_{\mathrm{Obf}}-1)} \tag{32}$$

Let $Adv_{\mathcal{F}}^{(q'_{\mathrm{Obf}})-Nick}$ be the probability of $\mathcal{F}$ (and $\mathcal{D}$) outputs 1 when $coin = 1$ and $\mathcal{F}$ queries $O^{<<O_{EGS}(id)>>}$ at most q'_{Obf} times. $Adv_{\mathcal{F}}^{(q'_{\mathrm{Obf}})-Nick}$ equals to $Adv_{\mathcal{F}}^{(q'_{\mathrm{Obf}})}$.

Hence, $\mathcal{D}$ is a distinguisher of a simulator and a real obfuscator because:

$$Adv_{\mathcal{D}}^{(q'_{\mathrm{Obf}})} \triangleq \left| Adv_{\mathcal{F}}^{(q'_{\mathrm{Obf}})-Nice} - Adv_{\mathcal{F}}^{(q'_{\mathrm{Obf}})-Junk} \right| \geq Adv_{\mathcal{F}}^{(q'_{\mathrm{Obf}})} - Adv_{\mathcal{F}}^{(q'_{\mathrm{Obf}}-1)} \geq \frac{Adv_{\mathcal{F}}^{(Q_{\mathrm{Obf}})}}{Q_{\mathrm{Obf}}} \tag{33}$$

In the above analysis, we show that, if the FT w.r.t. EGS Functionality is satisfied, but the FT w. r.t. EGS Obfuscator is NOT satisfied, then it contradicts the RR&ACVBP w.r.t. dependent oracle $T(C) = Sign_{sk_{\mathrm{ID}}}$ and restricted dependent oracle $\mathcal{R}(C) = Enroll_{MK}^{[id \neq \mathrm{ID}]}$. This ends the proof.

The following propositions are easy to verify, hence we omit the proofs.

Proposition 1. The proposed obfuscator Obf_{EGS} is rerandomizable w.r.t. the dependent oracle $T(C) = Sign_{sk_{\mathrm{ID}}}$ and the restricted dependent oracle $\mathcal{R}(C) = Enroll_{MK}^{[id \neq \mathrm{ID}]}$.

Proposition 2. The group signature scheme in [45] satisfies the following security properties with regard to the proposed EGS functionality or the proposed obfuscator Obf_{EGS}:

1. FT w.r.t. the proposed EGS functionality.
2. FA w.r.t. the proposed EGS functionality.
3. FT w.r.t. the proposed obfuscator Obf_{EGS}.
4. FA w.r.t. the proposed obfuscator Obf_{EGS}.

Related Studies and Comparison

The work of Barak et al. [30] initiated the theoretical investigation of obfuscators and has been a landmark in the research of obfuscation. The main result is that **general-purpose** obfuscation is impossible even under rather weak security definitions. This result is extended in many publications, such as the impossibility of obfuscation with auxiliary input [32], the impossibility of approximate obfuscation [31], the impossibility of efficient best-possible obfuscation [33], and the impossibility of restricted circuit classes [29]. Because of the difficulties or even impossibilities of various obfuscations, it is challenging to find a secure obfuscator even for a special functionality.

Fortunately, some positive results were obtained besides these negative results. It was shown by Canetti that under a very strong Diffie-Hellman assumption, point functions can be

obfuscated [51]. Further work from Wee [52] relaxes the assumptions required for obfuscation. Lynn et al. [53] gave several provable obfuscations for complex access control functionalities in the random oracle model. Moreover, Hofheinz et al. [42] provided some specific examples also with theoretical importance. One example is that we can easily transform an asymmetric encryption scheme into an obfuscatable symmetric encryption scheme, another example is that we can easily transform digital signature scheme into an obfuscatable MAC (Message Authentication Code).

However, these positive results mainly serve as theoretical illustrations. For instance, because the speed of the encryption algorithm in an asymmetric encryption scheme is usually much slower than that of a traditional block cipher, the strongly obfuscatable symmetric encryption scheme [42] is not suitable for practice. It is similar for the obfuscatable MAC [42], because the speed of verification algorithm in a digital signature scheme is usually much slower than that of using a keyed-Hash Message Authentication Code (HMAC).

Besides these positive results for theoretical study, some obfuscatable cryptographic functionalities and corresponding obfuscators for these cryptographic functionalities with acceptable runtime costs are introduced in recent years. In Table 8, we list the functionalities and obfuscators to the extent of our knowledge, and the last line is the proposed scheme in this paper. Furthermore, we provide a comparison on the security notions of obfuscation for different signature-related schemes in Table 9.

From Table 8, we can see that the proposed obfuscator is the first obfuscator for group oriented security schemes. Furthermore, as shown in Table 9, ACVBP w.r.t. DOs **and RDOs** introduced in this paper is the first security notion to fulfill the security requirement to protect a signature related scheme against collusion attacks. This new security notion should also be used to capture the security requirements of obfuscation of identity-based cryptosystems, forward-secure cryptosystem, key-insulated cryptosystem and threshold cryptosystem.

As we have mentioned at the end of section 2.1, there are two general-purpose obfuscators (in [43,44,58]) proposed in 2013 and 2014.

The first one is constructed for indistinguishability obfuscation that supports all polynomial-size circuits, which were given by Garg et al. [43] and strengthened by Barak et al. [58]. However, it uses a weak security notion of obfuscation, i.e. the indistinguishability obfuscation [29] which says that, for any pair of circuits C_0 and C_1 that agree on all inputs $C_0(x) = C_1(x)$, it should be hard to distinguish the obfuscation of C_0 from that of C_1. The new security notion used in this paper, i.e. the ACVBP w.r.t. DOs and RDOs, is stronger than the indistinguishability obfuscation.

The second general-purpose obfuscator is capable of obfuscating all polynomial size circuits [44]. The obfuscator, which uses graded encoding schemes is proven that the obfuscator exposes no more information than the program's black-box functionality, and achieves virtual black-box security, in the generic graded encoded scheme model. The obfuscator is obtained by developing techniques used to obfuscate d-CNF formulas in [59], and applying them to permutation branching programs. This yields an obfuscator for circuits in the complexity class NC1 and the obfuscator can be extended to a more powerful one for any polynomial-size circuit by using the homomorphic encryption technique. However, the complexity and expansion rate of homomorphic encryption are too large to be applied in practical applications.

Remark 10

NC1 denotes the class of decision problems decidable by uniform boolean circuits with a polynomial number of gates of at most two inputs and depth O(logn), or the class of decision

Table 8. A comparison of highly relative studies.

Functionality	Year	Base scheme(s) or building component(s)	Complexity Assumptions
Re-encryption [35]	2007	The linear encryption scheme (in [46]).	DLIN
Encrypted signature [34]	2010	The linear encryption scheme (in [46]), and Water's signature scheme (in [54]).	DBDH, DLIN
Encrypted verifiable Encrypted signature [36]	2011	The linear encryption scheme (in [46]), and Water's signature scheme (in [54]).	Exponent l-weak DH, DBDH, DLIN
Two-step oblivious signature [38]	2012	The linear encryption scheme (in [46]), Water's signature scheme (in [54]), and Pedersen's VSS protocol (in [55]).	SDHI, DLIN
Functional re-encryption [37]	2012	Re-Encryption (in [35])	SXDH
Encrypted Blind Signature [49]	2013	Schnorr's Blind Signature, and the linear encryption scheme (in [46]).	DH
Encrypted Proxy Signatures [50]	2013	Tightly Structure-Preserving Signatures (in [56]), and the linear encryption scheme (in [46]).	DLIN, DBDH
Conditional Re-encryption with Keyword Search [39]	2013	A modified version of ElGamal encryption.	DBDH
Encrypted Verifiably Encrypted Signatures [41]	2014	The linear encryption scheme (in [46]), and Water's signature scheme (in [54]).	CDH, AgExt, DLIN
Re-encryption, Functional Re-encryption, and Multi-hop Re-encryption [40]	2014	Regev's encryption scheme (in [57])	DLWE
Encrypted Group Signature (**this paper**)		The linear encryption scheme (in [46]), and a group signature scheme (in [45]).	CDH, SD, HSDH, DLIN

Acronyms used in the last column are explained as follows:

- Decisional Linear (DLIN);
- Decisional Bilinear Diffie-Hellman (DBDH);
- Strong Diffie Hellman Indistingshuishability (SDHI);
- Symmetric External Diffie-Hellman (SXDH);
- Diffie-Hellman (DH);
- Computational Diffie-Hellman (CDH);
- Aggregate Extraction (AgExt);
- Decisional Learning with Errors (DLWE);
- Subgroup Decision (SD);
- Hidden Strong Diffie-Hellman (HSDH).

doi:10.1371/journal.pone.0131550.t008

problems solvable in time O(logn) on a parallel computer with a polynomial number of processors. Many cryptographic functionalities are out of this class.

Remark 11

The impossibility results in [29,30,32] do not extend to idealized models, such as the random oracle model, the generic group model, and particularly the generic graded encoding model which is used in [44], hence the "generic purpose obfuscator" does not contradict these impossibility results.

Hence, although there have been some important advances in general-purpose obfuscation, such as in [43,44,58], there is no practically general approach for designing obfuscators under the security notion used in this paper. Therefore, it is a challenging work to find an obfuscatable encrypted group signature (EGS) functionality and design a corresponding efficient obfuscator.

Table 9. A comparison on the security notions of obfuscation for different signature-related schemes.

Reference No.	The main Security notion for the obfuscator	The dependent oracle(s)	The restricted dependent oracle(s)	The scheme-related security notion(s) w.r.t Obfuscator
[34]	ACVBP w.r.t. DOs	Sign	N/A	EU w.r.t. ES Functionality
[36]	ACVBP w.r.t. DOs	Sign	N/A	/
[38]	ACVBP w.r.t. DOs	Sign	N/A	/
[49]	ACVBP w.r.t. DOs	Sign	N/A	Blindness w.r.t. EBS Obfuscator; One-more Unforgeability w.r.t. EBS Obfuscator
[50]	ACVBP w.r.t. DOs	Sign	N/A	EU w.r.t. ES Functionality
[41]	ACVBP w.r.t. DOs	Sign	N/A	EU w.r.t. EVES Obfuscator; Opacity w.r.t. EVES Obfuscator
This paper	ACVBP w.r.t. DOs **and RDOs**	Sign	Enroll	FT w.r.t. EGS Obfuscator; FA w.r.t. EGS Obfuscator

Acronyms used in the above table are explained as follows

- Dependent Oracle (DO)
- Restricted Dependent Oracle (RDO)
- Existential Unforgeability (EU)
- Encrypted Blind Signature (EBS)
- Encrypted Verifiably Encrypted Signatures (EVES)

doi:10.1371/journal.pone.0131550.t009

Discussions

In this section, we first introduce possible applications and extensions of the proposed technique. Then, the rationale behind the obfuscatable sign-then-encrypt functionalities are investigated. Finally, the contribution of our findings is discussed at the last subsection.

6.1. Possible Applications and Extensions

Group signature schemes are applicable in many practical applications, such as in social networks [3,4], medical information systems [5–7], VANets [8,9], electronic voting [10], WSNs [11], electronic cash [12,13], and cloud computing [14–18]. These studies make great contributions for protecting security of information systems and privacy of users against various attacks. However, these applications are rather complicated. Therefore, to illustrate the applicability of the proposed technique, two simple examples are provided in section 6.1.1 and 6.1.2.

Moreover, the proposed technique can be adapted to identity-based cryptography and key-insulated cryptography. These extensions are introduced in section 6.1.3.

6.1.1. An Application in cloud computing. Group signature technique is perfectly suited for privacy-preserving security schemes in cloud computing. The proposed application is inspired by [18]. The application scenario for encrypted group signature schemes is a single company hosting a private cloud for its employees, e.g. providing access to file sharing or printing services. Usually, there is no need to identify the employee who has uploaded a certain file, and in some cases this may even be a confidential information (e.g. for labor unions within the company). However, in the event of uploading a file illegally, the company's management may have a severe interest in finding the responsible employee, regardless of potential reasons for preserving anonymity. The application is illustrated in Fig 8.

6.1.2. An Application in mobile healthcare social networks. The second application is a privacy-preserving emergency call scheme for mobile healthcare social networks. It is adapted from [4].

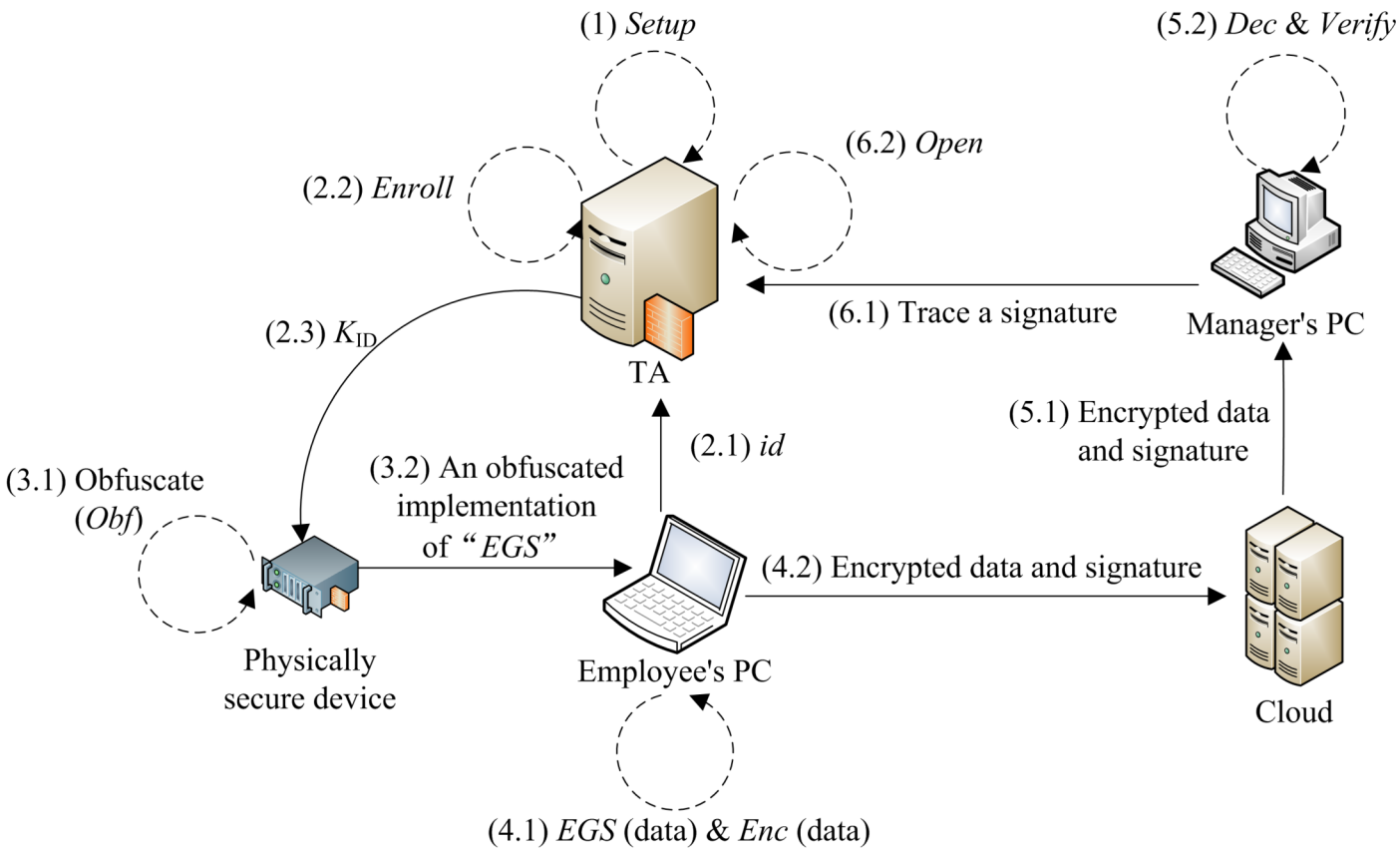


Fig 8. An application in cloud computing.

doi:10.1371/journal.pone.0131550.g008

As illustrated in Fig 9, a privacy-preserving emergency call system enables patients in life-threatening emergencies to accurately and fast transmit emergency data to the nearby helpers via mobile healthcare social networks. Once an emergency happens, the personal digital assistant (PDA) of the patient runs the emergency call procedure to collect the emergency data including patient health record, patient physiological condition, as well as the current emergency location. The emergency call procedure then generates an emergency call with the emergency data inside and epidemically disseminates it to every user in the patient's neighborhood. If a physician happens to be nearby, the PEC ensures the time used to notify the physician of the emergency to be the shortest.

In an emergency situation, a patient must reveal his/her information to the nearby users in order to ask for their instant help. However, with privacy concerns, the patients would preserve the identity privacy and prevent their transactions being linked to their unique identities. On the other hand, a TA must be able to trace the emergency call and identify the corresponding patient. In this way, any malicious attacker who has generated a bogus emergency call would be detected and punished. Details of the application are listed as follows and illustrated in Fig 10.

(1) Initialization Phase:

At the beginning, the Trusted Authority (TA) uses the algorithm *Setup* to generate system parameters, public values, the master enrollment key *MK*, and the group manager's tracing key *TK*.

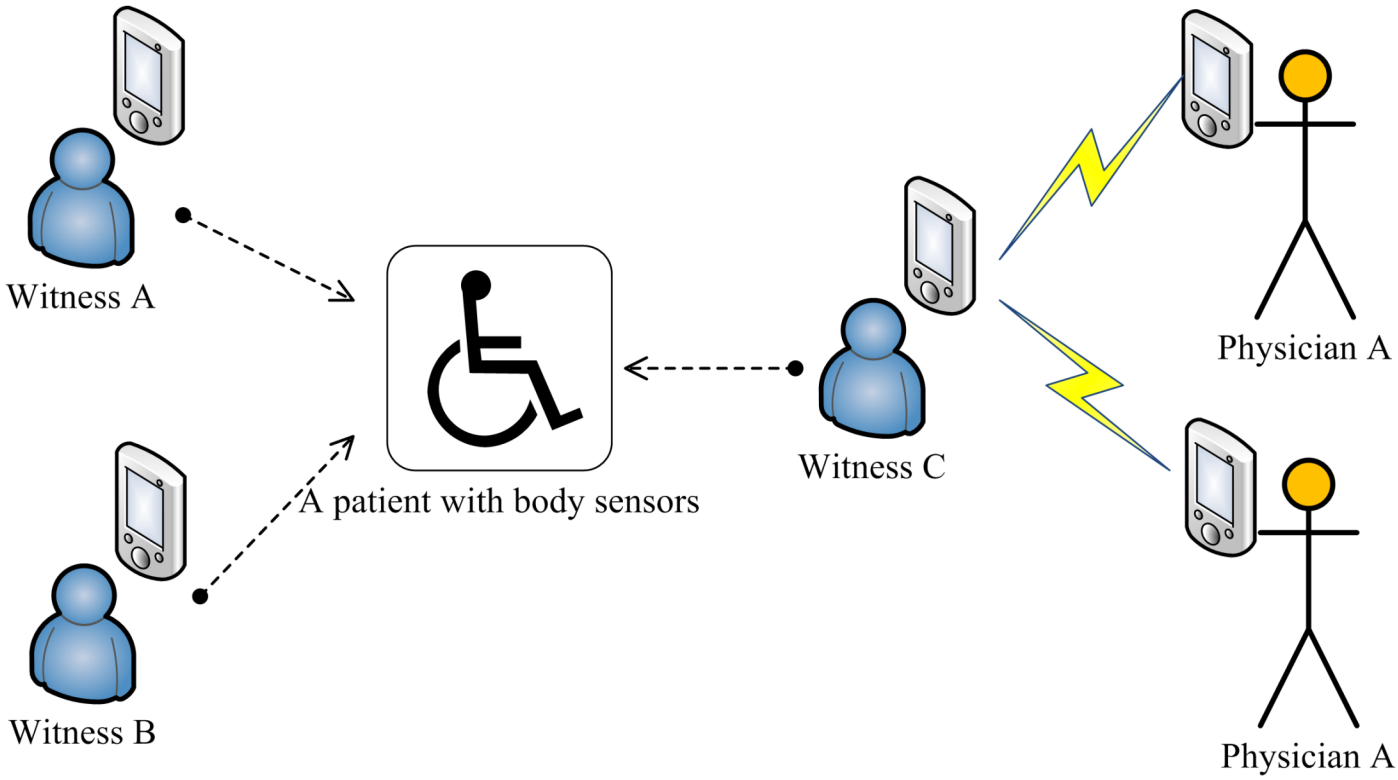


Fig 9. A decentralized emergency response scheme.

doi:10.1371/journal.pone.0131550.g009

(2) Registration Phase:

When a patient who needs medical services for possible emergency situations joins the system, TA uses the algorithm *Enroll* to issue a secret signing key *sk* and delivers the key to the patient through a secure channel.

(3) Obfuscation phase:

The patient uses the obfuscator Obf_{EGS} to generate an obfuscated implementation of encrypted group signature functionality EGS_{pub,sk,PK_e} in his/her personal computer. Then the patient transfers the obfuscated implementation to his/her PDA.

(4) Emergency call generation phase:

The emergency call generation is started by a detection of the abnormal physiological condition from body sensors. This condition can be pre-implanted into the patient's PDA with the instructions of the medical professionals.

Let patient n_i denote a user who has an emergency situation. The patient n_i's PDA generates an emergency call according to the following steps.

(4.1) General information collection phase: The PDA intelligently collects the following general information:

- Location (LOC): It contains the emergency location information which can be measured by a global positioning system (GPS) of the patient PDA.

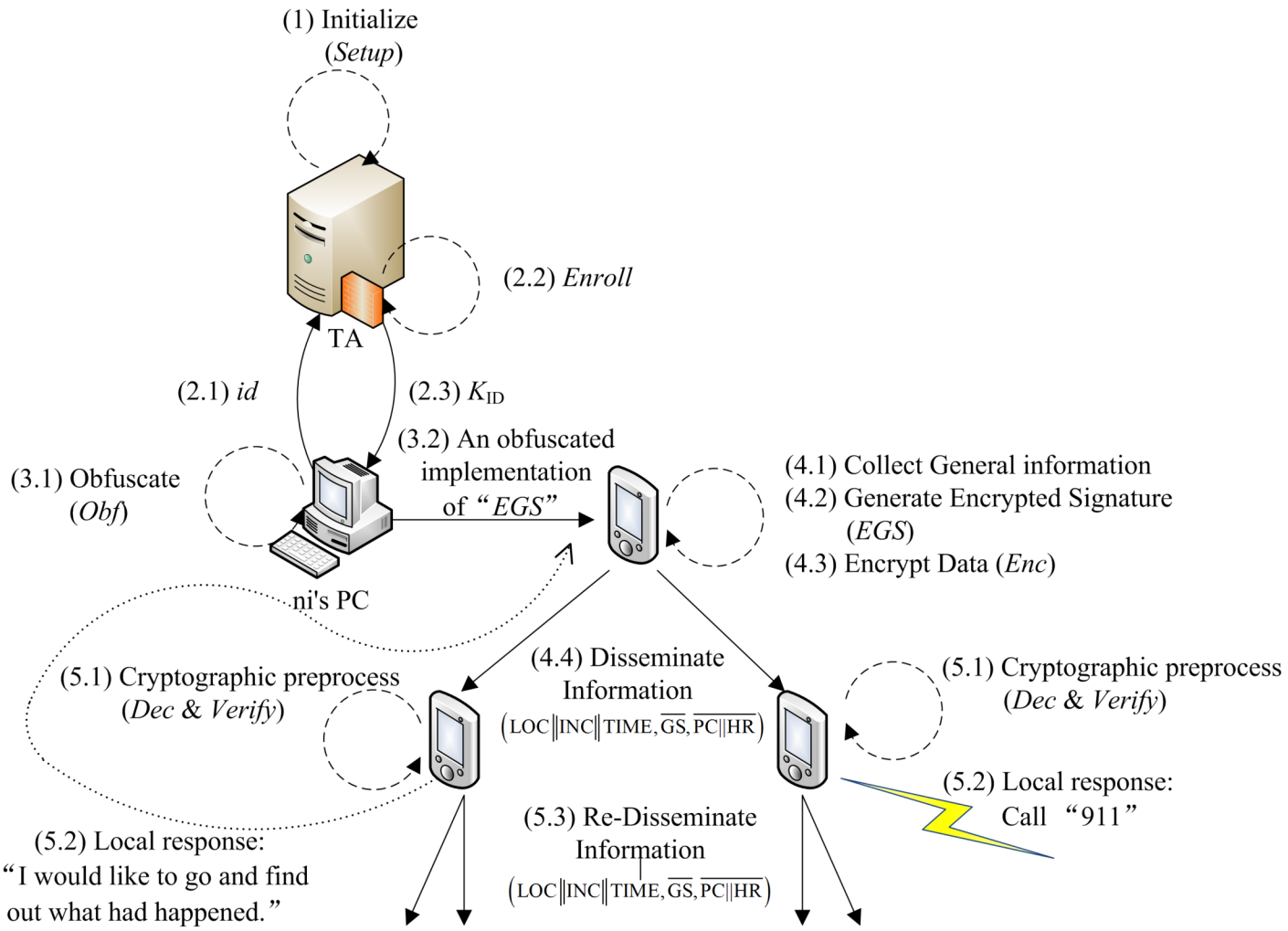


Fig 10. An application in a privacy-preserving emergency call system based on mobile social network.

doi:10.1371/journal.pone.0131550.g010

- Incident (INC): It contains a general description of the environment where the emergency occurs.
- Time (TIME): It contains the exact time when the emergency occurs.

(4.2) Encrypted Signature generation phase: The PDA uses the obfuscated implementation EGS to generate an encrypted group signature on "LOC||INC||TIME". The encrypted group signature will be put into the "EGS" component.

(4.3) Data encryption phase: The PDA encrypts the patient n_i's physiological condition (PC) and health record (HR).

(4.4) Information dissemination phase: The PDA disseminates the Emergency Call Message $ECM = LOC||INC||TIME, \overline{GS}, \overline{PC \parallel HR}$ to the neighboring users and APs (Access Points). Note that $\overline{GS}, \overline{PC \parallel HR}$ denotes the ciphertexts of GS and PC||HR where $EGS = \overline{GS}$.

(5) Emergency call response phase:

User n_j receives the ECM from patient n_i and executes the following steps, where a PDA represents user n_j 's PDA:

(5.1) Cryptographic preprocess: First, the PDA decrypts the encrypted part of the message $(\overline{GS}, \overline{PC \parallel HR})$ and gets the plaintext GS,PC||HR. Second, the PDA verifies the group signature "GS" by using the verification algorithm *Verify*. If the verification passes, user n_j confirms the information "LOC||INC||TIME."

(5.2) Local response: As an emergency response, user n_j firstly makes a phone call (e.g., dialing 911) to report the emergency to the hospital/first-aid center. Then, the PDA executes the step (5.3).

(5.3) Information re-dissemination phase: The PDA checks the "TIME" component: If the time period from the emergency occurrence to the user n_j receiving it, is larger than the threshold value, then the emergency call is discarded. Otherwise, the PDA forwards the ECM to the neighboring users.

6.1.3. Extensions to identity-based signatures and key-insulated signatures. In a traditional (certification-based) public-key cryptosystem, the association between a user's identity and his/her public key is obtained through a digital certificate issued by a Certifying Authority (CA). The CA checks the credentials of a user before issuing a certificate to the user. If a signer wants to sign a message, first the signer obtains a digital certificate for his/her public key from a CA. The signer then signs the message using the private signing key and sends the signed message along with his/her certificate to the receiver. The receiver (verifier) first verifies the validity of the certificate by checking the certificate revocation list published by the CA, then the receiver verifies the signature using the public key in the certificate. If many CAs are involved between the signer and the verifier, then the entire certificate path has to be verified.

Hence, the process of certificate management requires high computational and storage efforts. To simplify the certificate management process, Shamir [60] introduced the concept of identity-based cryptosystem. In such cryptosystems the public key of a user is derived from his/her identity information and a private key is generated by a Trusted Authority (TA). The advantage of an identity-based cryptosystem is that it simplifies the key management process which is a heavy burden in the traditional certificate based cryptosystems. In these cryptosystems, the verifier can verify the signer's signature just by using his/her identity information. In general, an identity based cryptosystem has the following properties:

- user's public key is his/her identity (or derived from the identity).
- no requirement of public key directories
- the verification process of a signature requires only the signers' identity (along with some public system parameters)

These properties make identity-based cryptosystems advantageous over the traditional certification-based cryptosystems, as key distribution is far simplified. It needs a directory only for authenticated public system parameters of the KGC (Key Generation Center), which is clearly less burdensome than maintaining a public key directory for total users.

As a by-product of the paper, we found that one could easily transform the proposed functionality to an obfuscatable encrypted identity-based signature by applying the following steps.

1. The KGC sets up the group for all users.
2. The KGC broadcasts the tracing key as a public value.
3. When generating a signature, the signer appends his/her identity at the end of the signature.
4. Merge the verification algorithm and the opening algorithm together.

5. The obfuscators, and even the encryption key generation algorithm, the encryption algorithm and the decryption algorithm can be used in the identity-based scenario without modification.

Note that the main security definition (ACVBP w.r.t. DOs and RDOs) can be almost directly obtained from Section 4. Moreover, the security proof can be deducted from this paper easily. So we omit the details in this paper.

Furthermore, as suggested in [61], by identifying time periods with identities, any identity-based signature scheme yields a perfectly (but not necessarily strong) key-insulated signature scheme. Accordingly, by applying this technique to the above identity-based scheme, an obfuscatable key-insulated signature scheme and a corresponding obfuscator is acquired.

Hence, besides the encrypted group signature, this paper adds two more new items in the list of obfuscatable cryptographic functionalities, i.e., the encrypted identity-based signature and the encrypted key-insulated signature.

6.2. The Rationale Behind the Obfuscatable Sign-Then-Encrypt Functionalities

Intuitively, design a sign-then-encrypt functionality needs a signing algorithm and an asymmetric encryption algorithm which are "commutable". Hence, the obfuscator can encrypt the private signing key by using the encryption algorithm with the public encryption key. In fact, all the works which are listed in Table 9 follow the thread of the idea in [34]. We explain the idea formally as follows.

In a digital signature scheme, the signing algorithm $\mathcal{S}$ takes the secret signing key $sk \in \mathbb{S}_{SSk}$ and a message $M \in \mathbb{S}_{Msg}$ to return a signature (also sometimes called a tag) $\sigma \in \{0,1\}^* \cup \perp$. The algorithm may be randomized. We write $\mathcal{S}(sk, M)$ for the operation of running Sign on inputs sk, M and letting σ be the signature returned.

In an asymmetric encryption scheme, the encryption algorithm $\mathcal{E}$ takes the public encryption key $PK \in \mathbb{S}_{PEK}$ and a plaintext $M \in \mathbb{S}_{PT}$ to return a value called the ciphertext. The algorithm may be randomized. We write $C \leftarrow \mathcal{E}(PK, M)$ for the operation of running $\mathcal{E}$ on inputs PK, M and letting $C \in \mathbb{S}_{CT}$ be the ciphertext returned.

Suppose the encryption is semantic secure (IND-CPA). If (34) holds, it is expected that we can obtain a secure obfuscator of $\mathcal{E} \circ \mathcal{S}$.

$$\forall sk \in \mathbb{S}_{SSK}, \forall PK \in \mathbb{S}_{PEK}, \forall M \in \mathbb{S}_{Msg}, \mathcal{E}(PK, \mathcal{S}(sk, M)) = \mathcal{S}(\mathcal{E}(PK, sk), M) \tag{34}$$

Eq (34) implies that

$$\mathcal{E}(\mathbb{S}_{PEK}, \mathbb{S}_{SSK}) \subseteq \mathbb{S}_{SSK}$$

Usually, the signing algorithm and the encryption algorithm are randomized. Therefore, (34) could be relaxed. Consider the implicit usage of random variables in the encryption algorithm and signing algorithm, we denote the set of random variables in the encryption process as R_E and the set of random variables in the signing process as R_S. Let $\mathcal{E}_{R_E}$ and $\mathcal{S}_{R_S}$ be the corresponding encryption process and signing process, respectively. In order to satisfy the requirement of preserving functionality, (35) is sufficient.

$$\begin{aligned} &\forall sk \in \mathbb{S}_{SSK}, \forall PK \in \mathbb{S}_{PEK}, \forall M \in \mathbb{S}_{Msg}, \exists (R_E, R_E', R_S, R_S',), s.t. \\ &\mathcal{E}_{R_E}(PK, \mathcal{S}_{R_S}(sk, M)) = \mathcal{S}_{R_S'}(\mathcal{E}_{R_E'}(PK, sk), M) \end{aligned} \tag{35}$$

As to the security requirement, the formal security analysis hinges on the security model of

the signature scheme. Because there are many variations of digital signature schemes (e.g., group-oriented, identity-based, etc.), it is out of the scope of this paper.

6.3. The Contribution

The paper has introduced a new secure obfuscator for encrypted group signature and corresponding security notions.

Theoretically, six new security notions of the encrypted group signature functionality and its obfuscators are proposed. The notions are ACVBP w.r.t. w.r.t. Dependent Oracles (DOs) and Restricted Dependent Oracles (RDOs), rerandomizable w.r.t. dependent oracle set and restricted dependent oracle set, full-traceability w.r.t. EGS Functionality, full-anonymity w.r.t. EGS Functionality, full-traceability w.r.t. EGS Obfuscator, and full-anonymity w.r.t. EGS Obfuscator.

The most important one of the new security notions is ACVBP w.r.t. DOs and RDOs which describes the security requirement of protecting the output of the proposed obfuscator, i.e., the obfuscated implementation of encrypted group signature functionality against collision attacks from group members. The security notions fit for many other cryptographic schemes which collision attacks from users need to be considered, such as identity-based signature schemes, ring signature schemes, attribute-based signature schemes and key-insulated signature schemes.

Practically, as it was discussed in Section 5, a generic obfuscator (for various sign then encrypt functionalities) is hard to find. Therefore, the obfuscators for various sign then encrypt functionalities must be studied one by one. In [34], Hada proposed the first obfuscatable sign-then-encrypt functionality and a corresponding obfuscator. Since then, most of the research work has been done on proposing various sign-then-encrypt functionalities and obfuscators, such as obfuscators for oblivious signature, encrypted blind signature, encrypted proxy signature, and encrypted verifiably encrypted signature. Note that it is not a trivial work to find an obfuscatable cryptographic functionality. For example, many widely-used signature schemes have not been found any obfuscatable concrete scheme (even in the sign then encrypt form), such as identity-based signature schemes, attribute-based signature schemes and key-evolvement signature schemes (include forward-secure signature, key-insulated signature, and intrusion-resilient signature).

A special obfuscatable group signature functionality, i.e., the encrypted group signature, is proposed with a concrete scheme, and then a corresponding obfuscator is provided in this paper. The correctness and security of the proposed obfuscator are proven. Then the efficiency of the proposed encrypted group signature functionality and its obfuscator is analyzed. The results of this paper can be used as building blocks of privacy preserving security protocol of various emerging applications such as social networks, medical information systems, Vehicular Ad hoc Networks (VANets), electronic voting, Wireless Sensor Networks (WSNs), electronic cash, and especially cloud computing.

Finally, as by-products of this paper, besides the encrypted group signature, we add two more new items in the list of obfuscatable cryptographic functionalities, i.e., the encrypted identity-based signature and the encrypted key-insulated signature.

Conclusions and Future Work

Group signature technique is used in many privacy-preserving security schemes for social networks, cloud computing, VANets, WSNs, electronic voting and electronic cash. To provide a building block for these schemes in white-box attack contexts, we give an obfuscatable EGS functionality, and then provide an obfuscator for the proposed EGS functionality. We also

introduce a new security notion (ACVBP w.r.t. Dependent Oracles and Restricted Dependent Oracles) to capture the requirement of protecting the output of an obfuscator for EGS functionality against collision attacks from group members. Moreover, five other new security notions are also provided. We prove that the proposed obfuscator preserves EGS functionality and satisfies the proposed security notions. As a byproduct of this study, ACVBP w.r.t. Dependent Oracles and Restricted Dependent Oracles fits for many types of cryptosystems, such as identity-based cryptosystems, forward-secure cryptosystems, key-insulated cryptosystems and threshold cryptosystems. Finally, we have introduced two possible applications and two extensions of the proposed technique.

In the future, we plan to adopt the obfuscatable EGS functionality in practical security solutions for validation. By using the proposed obfuscatable EGS functionality as a building block, we also consider exploring novel approaches for designing privacy-preserving security schemes. Furthermore, we will try to explore the design strategy for generalized constructions of obfuscatable sign-then-encrypt functionalities.

Author Contributions

Wrote the paper: YS. Validated the algorithms and proof-read the manuscript: QZ. Reviewed the paper: HF QL. Provided applications: HF. Provided extensions: QL.

References

1. Chaum D, Van Heyst E. Group signatures; 1991. Springer. pp. 257–265.
2. Bellare M, Micciancio D, Warinschi B (2003) Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. Advances in Cryptology-Eurocrypt. 2003 2656: 614–629.
3. Yager RR, Reformat MZ (2013) Looking for Like-Minded Individuals in Social Networks Using Tagging and E Fuzzy Sets. Ieee Transactions on Fuzzy Systems 21: 672–687.
4. Xiaohui L, Rongxing L, Le C, Xiaodong L, Xuemin S (2011) PEC: A privacy-preserving emergency call scheme for mobile healthcare social networks. Communications and Networks, Journal of 13: 102–112.
5. Guo LK, Zhang C, Sun JY, Fang YG (2012) PAAS: A Privacy-Preserving Attribute-based Authentication System for eHealth Networks. 2012 Ieee 32nd International Conference on Distributed Computing Systems (Icdcs): 224–233.
6. Hsu CL, Lu CF (2012) A Security and Privacy Preserving E-Prescription System Based on Smart Cards. Journal of Medical Systems 36: 3637–3647. doi: 10.1007/s10916-012-9838-y PMID: 22407399
7. Zhang K, Liang X, Baura M, Lu R, Shen X (2014) PHDA: A priority based health data aggregation with privacy preservation for cloud assisted WBANs. Information Sciences 284: 130–141.
8. Chen YY, Wang YJ, Jan JK (2013) The design of speedy seamless safe messaging mechanism in VANET. International Journal of Computer Mathematics 90: 2614–2630.
9. Zhu XY, Jiang SR, Wang LM, Li H (2014) Efficient Privacy-Preserving Authentication for Vehicular Ad Hoc Networks. Ieee Transactions on Vehicular Technology 63: 907–919.
10. Tornos JL, Salazar JL, Piles JJ (2013) An eVoting platform for QoE evaluation. 2013 Ifip/Ieee International Symposium on Integrated Network Management (Im 2013): 1346–1351.
11. Zhu LH, Li M, Liao LJ (2012) Dynamic Group Signature Scheme Based Integrity Preserving Event Report Protocol in Wireless Sensor Networks. Sensor Letters 10: 1785–1791.
12. Chen MT, Fan CI, Juang WS, Yeh YC (2012) An Efficient Electronic Cash Scheme with Multiple Banks Using Group Signature. International Journal of Innovative Computing Information and Control 8: 4469–4482.
13. Lian B, Chen GL, Li JH (2014) Provably secure E-cash system with practical and efficient complete tracing. International Journal of Information Security 13: 271–289.
14. Liu XF, Zhang YQ, Wang BY, Yan JB (2013) Mona: Secure Multi-Owner Data Sharing for Dynamic Groups in the Cloud. Ieee Transactions on Parallel and Distributed Systems 24: 1182–1191.

15. Sookhak M, Talebian H, Ahmed E, Gani A, Khan MK (2014) A review on remote data auditing in single cloud server: Taxonomy and open issues. Journal of Network and Computer Applications 43: 121–141.
16. Wang B, Li H, Li M. Privacy-preserving public auditing for shared cloud data supporting group dynamics; 2013. IEEE. pp. 1946–1950.
17. Yu Y, Niu L, Yang GM, Mu Y, Susilo W (2014) On the security of auditing mechanisms for secure cloud storage. Future Generation Computer Systems-the International Journal of Grid Computing and Escience 30: 127–132.
18. Jensen M, Schäge S, Schwenk J. Towards an Anonymous Access Control and Accountability Scheme for Cloud Computing; 2010. IEEE. pp. 540–541.
19. Shi Y, Lin J, Zhang C (2011) A White-Box Encryption Algorithm for Computing with Mobile Agents. Journal of Internet Technology 12: 981–993.
20. Shi Y, Xiong GY (2013) An Undetachable Threshold Digital Signature Scheme Based on Conic Curves. Applied Mathematics & Information Sciences 7: 823–828.
21. Yum DH, Lee PJ (2012) Exact Formulae for Resilience in Random Key Predistribution Schemes. Ieee Transactions on Wireless Communications 11: 1638–1642.
22. Tague P, Li MY, Poovendran R (2009) Mitigation of Control Channel Jamming under Node Capture Attacks. Ieee Transactions on Mobile Computing 8: 1221–1234.
23. Nishimoto Y, Imaizumi H, Mita N (2009) Integrated Digital Rights Management for Mobile IPTV Using Broadcasting and Communications. Ieee Transactions on Broadcasting 55: 419–424.
24. Hwang SO (2009) Content and Service Protection for IPTV Ieee Transactions on Broadcasting 55: 686–686.
25. Razzaque MA, S A, Cheraghi S (2013) Security and Privacy in Vehicular Ad-Hoc Networks: Survey and the Road Ahead. In: Khan S, Khan Pathan A-S, editors. Wireless Networks and Security: Springer Berlin Heidelberg. pp. 107–132.
26. Yang W (2013) Security in Vehicular Ad Hoc Networks (VANETs). Wireless Network Security: Springer Berlin Heidelberg. pp. 95–128.
27. Mejri MN, Ben-Othman J, Hamdi M (2014) Survey on VANET security challenges and possible cryptographic solutions. Vehicular Communications 1: 53–66.
28. Elahi G, Yu E, Gu YX, Wajs A, Liu L Goal-Oriented Modeling and Analysis of White-Box Security: Toward an Ontology.
29. Barak B, Goldreich O, Impagliazzo R, Rudich S, Sahai A, et al. (2012) On the (Im)possibility of Obfuscating Programs. Journal of the Acm 59.
30. Barak B, Goldreich O, Impagliazzo R, Rudich S, Sahai A, et al. (2001) On the (im) possibility of obfuscating programs. Advances in Cryptology—CRYPTO 2001: Springer. pp. 1–18.
31. Bitansky N, Paneth O (2013) On the impossibility of approximate obfuscation and applications to resettable cryptography. Proceedings of the forty-fifth annual ACM symposium on Theory of computing: ACM. pp. 241–250.
32. Goldwasser S, Kalai YT (2005) On the impossibility of obfuscation with auxiliary input. 46th Annual Ieee Symposium on Foundations of Computer Science, Proceedings: 553–562.
33. Goldwasser S, Rothblum GN (2014) On Best-Possible Obfuscation. Journal of Cryptology 27: 480–505.
34. Hada S (2010) Secure Obfuscation for Encrypted Signatures. Advances in Cryptology—Eurocrypt 2010. pp. 92–112.
35. Hohenberger S, Rothblum GN, Shelat A, Vaikuntanathan V (2011) Securely Obfuscating Re-Encryption. Journal of Cryptology 24: 694–719.
36. Cheng R, Zhang B, Zhang FG (2011) Secure Obfuscation of Encrypted Verifiable Encrypted Signatures. Provable Security. pp. 188–203.
37. Chandran N, Chase M, Vaikuntanathan V (2012) Functional Re-encryption and Collusion-Resistant Obfuscation. In: Cramer R, editor. Theory of Cryptography: Springer Berlin Heidelberg. pp. 404–421.
38. Li C, Yuan Z, Mao M (2012) Secure Obfuscation of a Two-Step Oblivious Signature. Network Computing and Information Security 345: 680–688.
39. Cheng R, Zhang F (2013) Secure Obfuscation of Conditional Re-encryption with Keyword Search. Intelligent Networking and Collaborative Systems (INCoS), 2013 5th International Conference on: IEEE. pp. 30–37.
40. Chandran N, Chase M, Liu F-H, Nishimaki R, Xagawa K (2014) Re-encryption, Functional Re-encryption, and Multi-hop Re-encryption: A Framework for Achieving Obfuscation-Based Security and

Instantiations from Lattices. In: Krawczyk H, editor. Public-Key Cryptography—PKC 2014: Springer Berlin Heidelberg. pp. 95–112.

41. Nishimaki R, Xagawa K (2014) Verifiably encrypted signatures with short keys based on the decisional linear problem and obfuscation for encrypted VES. Designs, Codes and Cryptography: 1–38.
42. Hofheinz D, Malone-Lee J, Stam M (2010) Obfuscation for Cryptographic Purposes. Journal of Cryptology 23: 121–168.
43. Garg S, Gentry C, Halevi S, Raykova M, Sahai A, et al. (2013) Candidate Indistinguishability Obfuscation and Functional Encryption for all circuits (Extended Abstract). 2013 Ieee 54th Annual Symposium on Foundations of Computer Science (Focs). pp. 40–49.
44. Brakerski Z, Rothblum G (2014) Virtual Black-Box Obfuscation for All Circuits via Generic Graded Encoding. In: Lindell Y, editor. Theory of Cryptography: Springer Berlin Heidelberg. pp. 1–25.
45. Boyen X, Waters B (2007) Full-domain subgroup hiding and constant-size group signatures. Public Key Cryptography—PKC 2007. pp. 1–15.
46. Boneh D, Boyen X, Shacham H (2004) Short group signatures. Advancs in Cryptology—Crypto 2004, Proceedings. pp. 41–55.
47. Shen E, Shi E, Waters B (2009) Predicate Privacy in Encryption Systems. Theory of Cryptography, 6th Theory of Cryptography Conference, Tcc 2009 5444: 457–473.
48. Boyen X, Waters B (2006) Anonymous hierarchical identity-based encryption (Without random oracles). Advances in Cryptology—Crypto 2006, Proceedings 4117: 290–307.
49. Feng X, Yuan Z (2014) A Secure Obfuscator for Encrypted Blind Signature Functionality. Network and System Security: Springer International Publishing. pp. 311–322.
50. Wei X, Yuan Z, Li X, Feng X, Liu J. Secure Obfuscation for Tightly Structure-Preserving Encrypted Proxy Signatures; 2013. IEEE. pp. 589–593.
51. Canetti R (1997) Towards realizing random oracles: Hash functions that hide all partial information. Advances in Cryptology—Crypto'97, Proceedings. pp. 455–469.
52. Wee H (2005) On obfuscating point functions. Proceedings of the thirty-seventh annual ACM symposium on Theory of computing: ACM. pp. 523–532.
53. Lynn B, Prabhakaran M, Sahai A (2004) Positive results and techniques for obfuscation. Advances in Cryptology—Eurocrypt 2004, Proceedings. pp. 20–39.
54. Waters B (2005) Efficient identity-based encryption without random oracles. Advances in Cryptology—Eurocrypt 2005,Proceedings. pp. 114–127.
55. Chen L, Pedersen TP (1995) New group signature schemes. In: De Santis A, editor. Advances in Cryptology—EUROCRYPT'94: Springer Berlin Heidelberg. pp. 171–181.
56. Hofheinz D, Jager T (2012) Tightly Secure Signatures and Public-Key Encryption. In: Safavi-Naini R, Canetti R, editors. Advances in Cryptology—CRYPTO 2012: Springer Berlin Heidelberg. pp. 590–607.
57. Regev O (2009) On lattices, learning with errors, random linear codes, and cryptography. Journal of the ACM (JACM) 56: 34.
58. Barak B, Garg S, Kalai Y, Paneth O, Sahai A (2014) Protecting Obfuscation against Algebraic Attacks. In: Nguyen P, Oswald E, editors. Advances in Cryptology—EUROCRYPT 2014: Springer Berlin Heidelberg. pp. 221–238.
59. Brakerski Z, Rothblum GN (2014) Black-box obfuscation for d-CNFs. Proceedings of the 5th conference on Innovations in theoretical computer science. Princeton, New Jersey, USA: ACM. pp. 235–250.
60. Shamir A (1985) Identity-based cryptosystems and signature schemes. Advances in cryptology: Springer. pp. 47–53.
61. Dodis Y, Katz J, Xu S, Yung M (2002) Strong key-insulated signature schemes. Public Key Cryptography—PKC 2003: Springer. pp. 130–144.