Short (resp. Fast) CCA2-Fully-Anonymous Group Signatures using IND-CPA-Encrypted Escrows

Victor K. Wei[1]

Department of Information Engineering,
The Chinese University of Hong Kong,
Shatin, Hong Kong
{kwwei}@ie.cuhk.edu.hk

Abstract. In the newest and strongest security models for group signatures [7, 10, 41], attackers are given the capability to query an Open Oracle, $\mathcal{OO}$, in order to obtain the signer identity of the queried signature. This oracle mirrors the Decryption Oracle in security experiments involving encryption schemes, and the security notion of CCA2-full-anonymity for group signatures mirrors the security notion of IND-CCA2-security for encryption schemes. Most group signatures escrows the signer identity to a TTP called the *Open Authority (OA)* by encrypting the signer identity to OA. Methods to efficiently instantiate $O(1)$-sized CCA2-fully-anonymous group signatures using IND-CCA2-secure encryptions, such as the Cramer-Shoup scheme or the twin encryption scheme, exist [7, 10, 41, 49]. However, it has long been suspected that IND-CCA2-secure encryption to OA is an overkill, and that CCA2-fully-anonymous group signature can be constructed using only IND-CPA-secure encryptions. Here, we settle this issue in the positive by constructing CCA2-fully-anonymous group signatures from IND-CPA-secure encryptions for the OA, without ever using IND-CCA2-secure encryptions. Our technique uses a single ElGamal or similar encryption plus Dodis and Yampolskiy [35]'s VRF (Verifiable Random Function). The VRF provides a sound signature with zero-knowledge in both the signer secret and the signer identity, while it simultaneously defends active $\mathcal{OO}$-query attacks. The benefits of our theoretical advance is improved efficiency. Instantiations in pairings result in the shortest CCA2-fully-anonymous group signature at 11 rational points or ≈ 1870 bits for 170-bit curves. It is 27% shorter (and slightly faster) than the previous fastest [12, 41] at 15 rational points. Instantiations in the strong RSA framework result in the fastest CCA2-fully-anonymous group signature at 4 multi-base exponentiations for 1024-bit RSA. It is 25% faster than the previous fastest at 5 multi-base exponentiations [3, 20, 41].

1 Introduction

Chaum and van Heyst pioneered the study of group signatures [28]. Early group signatures grow in size proportional to the number of members and/or the group manager picks user secrete keys [29, 30, 23]. After progressive advances, contemporary state-of-the-art group signatures require $O(1)$ size, users choose their own keys and are free from framing by managers, coalition-resistant, in addition to the basic anonymity and soundness requirements [3, 20, 7, 10, 41].

In the newest and strongest security models for group signatures [7, 10, 41], attackers are given the capability to query an Open Oracle, $\mathcal{OO}$, in order to obtain the signer identity of the queried signature. This oracle mirrors the Decryption Oracle in security experiments involving encryption schemes, and the security notion of CCA2-full-anonymity for group signatures mirrors the security notion of IND-CCA2-security for encryption schemes. Most group signatures escrows the signer identity to a TTP called the *Open Authority (OA)* by encrypting the signer identity to OA. Method to efficiently instantiate $O(1)$-sized CCA2-fully-anonymous group signatures using IND-CCA2-secure encryptions, such as the Cramer-Shoup scheme or the twin encryption scheme, exist [7, 10, 41, 49]. However, it has long been suspected that IND-CCA2-secure encryption to OA is an overkill, and that CCA2-fully-anonymous group signature can be constructed using only IND-CPA-secure encryptions. Here, we settle this issue in the positive by constructing CCA2-fully-anonymous group signatures from only IND-CPA-secure encryptions without ever converting them to IND-CCA2 encryptions. The benefits of this theoretical advance is efficiency improvements. We explain below.

Boneh, Boyen, and Shacham constructed a short group signature [12] using pairings. However, their scheme had only CPA-full-anonymity. Their security is reduced to the DLDH (Decisional Linear Diffie-Hellman)

Assumption and the q-SDH (Strong Diffie-Hellman) Assumption. Nguyen and Safavi-Naini [49] constructed additional CPA-fully-anonymous and CCA2-fully-anonymous group signatures in pairings. Their security is reduced to the DBDH (Decisional Bilinear Diffie-Hellman) Assumption and the q-SDH Assumption. Due to the use of pairings, [12]'s signatures are the shortest group signatures to date. Its length is about 9 rational points, for 170-bits curves from contemporary pairings technology. However, the signature verification costs one expensive pairings and a few multi-base exponentiations online.

The well-known method of twin encryption can be used to modify [12]'s CPA-fully-anonymous group signature to CCA-fully-anonymous ones [47, 41, 49]. The resulting group signatures remain the shortest to date versus other CCA-fully-anonymous group signature, roughly 15 rational points. The online signature verification complexity is one pairings plus 8 multi-base exponentiations. The pairings is expensive to compute. In comparison, an ordinary (non-group) IND-CCA2-secure RSA signature is 2048 bits long for the 1024-bit RSA framework. The fastest CCA2-fully-anonymous group signature which can be instantiated by existing methods [3, 20, 41] stands at 5 online multi-base exponentiations for the 1024-bit RSA framework.

Instantiating our theoretical advance in pairings, we obtain an even shorter CCA2-fully-anonymous group signature, at roughly 11 rational points for 170-bit curves. That is a 27% improvement. The complexity is also improved slightly. The cost remains one online pairings but plus only 5 multi-base exponentiations. Instantiating in the RSA framework, our new CCA2-fully-anonymous group signature costs only 4 online multi-base exponentiations. That is 25% faster than the previous record.

Our **Contributions** are

1. We make the theoretical advancement of constructing CCA2-fully-anonymous group signatures using only CPA-secure encryptions to escrow the signer identity to the OA (Open Authority). Our technique uses Dodis and Yampolskiy [35]'s Verifiable Random Function (VRF) as a sound signature which is zero-knowledge about both the signer secret and the signer identity, while simultaneously defends active attacks via Open Oracle queries.
2. The benefits of our theoretical advancements includes significant improvements in the efficiency of CCA2-fully-anonymous group signatures.
   (a) Instantiating in pairings we obtain the new shortest CCA2-fully-anonymous group signature. It is 27% faster than the previous fastest [12, 41, 49], reducing from 15 rational points to 11 (at $11\lambda_s = 1870$ bits). 170-bit curves and 1024-bit RSA are considered contemporary security standards. Note that an ordinary (non-group) IND-CCA2-secure RSA encryption costs 2048 bits.
   (b) Instantiating in the strong RSA framework, we obtain the new fastest CCA2-fully-anonymous group signature. It is 25% faster than the previous fastest [3, 21, 41], reducing from 5 to 4 multi-base exponentiations for online signature verification. Our scheme also improves the bandwidth cost somewhat.

Related results The mainstream group signature proceeds as follows: A member/user joins a group by presenting its public key, prove knowledge of its corresponding secrete key, and obtains a certificate from the Group Manager who serves as a kind of CA (Certificate Authority). The group signature consists of an NIZK (non-interactive zero-knowledge) proof of simultaneous knowledge of a valid certificate (i.e. certified public key), of its corresponding secrete key, of proper encryption (escrowing) of the signer identity to the OA. The results of [3, 21, 7, 10, 41] represent the mainstream group signature well. The group membership certificate is (A, e) satisfying $A^e h_1^{x_1} h_2^{x_2} h_3^{x_3} = h_0$ where group trapdoor is the factoring of a safe product N, the user sk-pk pair it $((x_1, x_2, x_3), (h_1^{x_1}, h_2^{x_2}, h_3^{x_3}))$. Note x_1 is user generated, $x2$ is jointly generated by user and group, and x_3 is generated by group to meet three kinds of principle needs. The group signature is the NIZK (or SPK, for Signature Proof of Knowledge [24]):

$$\sigma = SPK\{(A, e, x_1, x_2, x_3) : A^e h_1^{x_1} h_2^{x_2} h_3^{x_3} = h_0 \in QR_N \ \wedge \ \mathsf{ctxt} = \mathsf{Enc}(\mathsf{pk}_{OA}, A, h_1^{x_1}, \rho)\}(M) \tag{1}$$

Instantiations in pairings are represented by [44, 58, 12, 49]:

$$\sigma = SPK\{(A, e, x_1, x_2, x_3) : A^{e+\gamma} h_1^{x_1} h_2^{x_2} h_3^{x_3} = h_0 \in \mathbb{G}_1 \ \wedge \ \mathsf{ctxt} = \mathsf{Enc}(\mathsf{pk}_{OA}, \theta(A, h_1^{x_1}), \rho)\}(M) \tag{2}$$

where group sk-pk pair is $(\gamma, u^\gamma \in \mathbb{G}_2)$, and θ implements [24]'s *shadow encryption.* Incidentally [10]'s generic group signature excludes shadow encryption which we find eminently useful in group signatures from pairings.

Variations: Ring signatures [51, 2, 57] and the early [31] are essentially group signatures without the OA. Direct Digital Attestation (DAA) is essentially the above plus a linking tag [16, 19] of the form h_{new}^x where x is a user secrete key. The linking tag has also been used in traceable (group) signature [39], linkable group/ring signatures [45, 46, 42, 19, 37]. The Group Manager and the OA are the same in some group signatures [39, 5, 37]. The results in [44, 40] made connections between group signature and traitor tracing. Membership revocation which require all members to alter their certificate is optimized in [6, 20, 48]. If the group size is too large to make this approach efficient, then the CRL (Certificate Revocation List) approach to membership revocation is advanced in [39].

The intuition of our results: In order to support CA2-full-anonymity, [7, 41, 10] specified the use of IND-CCA2-secure encryption to escrow the signer identity. However, the state-of-the-art IND-CCA2-secure encryptions are typically twice as expensive as IND-CPA-secure ones. Examples: twin encryption [47]; the new three-round OAEP doubles the RSA ciphertext size [9, 50]; the Cramer-Shoup ciphertext[32] is twice as long as the ElGamal ciphertext.

It has long been suspected that IND-CCA2-secure encryptions should be an overkill for the group signature. Observe that, in querying the Open Oracle $\mathcal{OO}$, the Adversary $\mathcal{A}$ must present a valid group signature. Therefore $\mathcal{OO}$ need not be as powerful as the Decryption Oracle. We set out to exploit this intuition. We observe that the most threatening attacks query $\mathcal{OO}$ with (1) a signature signed by a corrupted user, (2) with a $\mathcal{SO}$ (Signing Oracle) output, or (3) with a successfully forged signature. The recent tecnology of VRF (Verifiable Random Function) [43] and [35]'s practical instantiation of it can be used to construct a group signature where the first two kinds of threatening queries can be answered, and the unforgeability (non-frameability) of the group signature itself means the third kind cannot happen. And we are successful.

Efficiency discussions: Take [12]'s CPA-fully-anonymous group signature, which is the shortest group signature prior to the current result. The signature size is about $9\lambda_s$ bits, where $\lambda_s = 170$-bit curve is considered the standard contemporary technology. To break down the bandwidth cost: $5\lambda_s$ bits are for the linear encryption to escrow the signer identity, $3\lambda_s$ bits are for proof of certified public key and its corresponding secrete key, and λ_s bits are for transmitting the challenge of the proof system. [12] does not have non-frameability, which they termed exculpability. Non-frameability costs extra λ_s bits to support. Conversion to twin encryption doubles the encryption cost from $5\lambda_s$ to $10\lambda_s$ bits. Then CCA2-full-anonymity is gained. But the increase in bandwidth cost is a significant percentage: 50%. By using IND-CPA encryption, we save the 50% percent immediately. The overhead caused by using VRF does not eat up the entire gain. Some additional but hard-wrought optimization helped a bit. Our new shortest CCA2-fully-anonymous group signature stands at $11\lambda_s$ bits, and is 27% shorter than the previous shortest. Similar savings when we instantiate in the Strong RSA framework to break the speed record.

2 Preliminaries

2.1 Pairings, intractability assumptions

We summarize needed intractability assumptions. A *pairing* is a mapping $\hat{\mathbf{e}} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_3$ satisfying $\hat{\mathbf{e}}(g^a, h^b) = \hat{\mathbf{e}}(g, h)^{ab}$. Consult Boneh and Franklin [13] for further details. Pairings were initially viewed as a weakness of elliptic curve discrete logarithm when Joux and Nguyen [38] first exhibited GDH (Gap Diffie-Hellman) groups. Then Boneh and Franklin found its applications to post-certification [13] and a wide variety of other areas. The web page *Pairing-Based Crypto Lounge*, maintained by Paulo Barreto, contains a wealth of news and archival information.

Below, let G_a and G_b be groups. We use the convention that the *XYZ Assumption* is that no PPA algorithm has a non-negligible advantage over arbitrary guessing in solving solve a random instance of the *XYZ Problem*. Also, the XYZ(G_a,G_b) Problem becomes the *XYZ Problem in* G_a when $G_a = G_b$.

The *DDH(G_a, G_b) Problem* is given g, $g^x \in G_a$, $h \in G_b$, distinguish $h^x \in G_b$ from random, where $0 < x < \text{order}(G_a) \leq \text{order}(G_b)$

The *Decisional Linear Diffie-Hellman Problem for groups G_a and G_b*, denoted the *DLDH(G_a, G_b) Problem*, is, given g_1, g_2, g_1^x, $g_2^y \in G_a$, $h \in G_b$, distinguish $h^z \in G_b$ from random, where $0 < x, y, z < \text{order}(G_a) < \text{order}(G_b)$.

The *q-SDH(G_a, G_b) Problem* is, given $g, g^{x'} \in G_a$ and $h^{x^i} \in G_b$ for $0 \leq i \leq q$, compute some $(e, h^{1/(e+x')} \in G_b)$, where $0 < x, x' <$ order$(G_a) \leq$ order(G_b), $x' = x \in Z$. The *q-SDDH(G_a, G_b) Problem* is, given $g, g^{x'} \in G_a$ and $h^{x^i} \in G_b$ for $0 \leq i \leq q$ and R,distinguish $h^{1/(R+x')} \in G_b$ from random, where where $0 < x, x' <$ order$(G_a) \leq$ order(G_b), $x' = x \in Z$. Note "SDH" stands for Strong Diffie-Hellman and "SDDH" stands for Strong Decisional Diffie-Hellman. Note the q-SDDH Problem specializes to the q-DDHI Problem [11, 12, 35] when $R = 0$. We will need the following Lemma:

Lemma 1 *The q-SDDH(G_a, G_b) Assumption implies the DDH(G_a, G_b) Assumption.*

Proof Sketch: Given the q-SDDH Problem with test value τ, convert it to the DDH Problem instance g, g^x, τ, $h\tau^{-e}$ which equals τ^x if $\tau = h^{1/(e+x')}$. □

Let N be the product of two primes. The *Strong RSA Problem* is, given N and random $u \in QR_N$, compute some non-trivial $(e, u^{1/e})$.

2.2 Zero-knowledge proofs with known discrete logarithm bases

We will use literature results on proving statements about discrete logarithms, suc has (1) proof of knowledge of a discrete logarithm modulo a prime [52] or modulo a composite [36, 33]; (2) proof of knowledge of equality of representations modulo two (possibly different) prime [27] or composite [22] moduli; (3) proof that a commitment opens to the product of two other committed values [22]; (4) proof of range: that a committed value lies in a given integer interval [26, 22, 15]; (5) proof of the disjunction or conjunction of the previous [31].

We use mainly the notations from [24]. For example, $SPK\{(A, e, x) : A^e h^x = h_0 \in QR_N\}(M)$ means proof of knowledge of secrete values A, e, and x satisfying the relation $A^d h^x = h_0 \in QR_N$. All used symbols/values that are not explicitly denoted as secrete values are assumed non-secrete and must be made known to the verifier. These values include, in this example, h, h_0, N.

2.3 Verifiable random functions (VRF)

A Verifiable Random Function (VRF) [43] is a pseudo-random function that provides a non-interactive proof for the correctness of its output. Given an input R and the secrete sk, it is complete to compute the function value $f_{\mathsf{sk}}(R)$ and a universally verifiable proof of correctness π. The output reveals zero-knowledge about the pair (R, sk). The VRF finds many applications in protocol designs [35]. Dodis and Yamploskiy [35] proposed the VRF which will be used in this paper: $f_{\mathsf{sk}}(R) = \mathbf{g}^{1/(\mathsf{sk}+R)}$.

3 Security Model

We follow mainly the security model for group signature in [10, 7]. For simplicity we use the stagic group model [7]. But we include applicable concepts from the more advanced dynamic group model [10] as well.

3.1 Syntax

A *group signature* is a tuple (Init, OKg, GKg, UKg, Join, Iss, GSig, GVf, Open, Judge) where:

- Init: $1^{\lambda_s} \mapsto \mathsf{param}$. On input the security parameter 1^{λ_s}, generates system-wide public parameters param, CA (Group Manager) sk-pk pair $(u, u^\gamma) \in Z_{q_1} \times \mathbb{G}_2$, Open Manager (OA) sk-pk pair $(\mathsf{sk}_{OA}, \mathsf{pk}_{OA})$, a number of users with sk-pk pairs $(x_U, h^{x_U}) \in Z_{q_1} \times \mathbb{G}_1$, for user U, certificate cert_U for user U, ..., etc.
- GSig : $(\mathsf{pk}, \mathsf{sk}, \mathsf{cert}, M) \mapsto \sigma$. On input the keys, certificates and message, outputs a signature σ.
- GVf: $(\mathsf{ca}, \mathsf{oa}, M, \sigma) \mapsto 0$ or 1. On input the message and signature, outputs 1 for valid signature and 0 for invalid signature.
- Open: $(\mathsf{ca}, x_{\mathsf{oa}}, \mathsf{reg}, M, \sigma) \mapsto (i, \omega)$. The OA with key x_{oa} has read access to reg. On input a valid signature σ for message M for ca, output identity i for the corresponding signer, and ω is the proof of this claim. Output $i = \perp$ if no such member is found.

- Judge: $(\mathsf{ca}, \mathsf{id}, \mathsf{oa}, M, \sigma, \omega) \mapsto 0$ or 1. It checks if the proof ω is a valid proof that id is the real signer of σ for message M under ca, oa. Outputs 1 for valid and 0 for invalid.

Remarks: Here we use $(\mathsf{param}, \mathsf{ca})$ to denote *gpk* in [10]'s original syntax. We also split the GKg in [10] into Init, OKg and GKg. It is because we want to emphasize that group managers (CA) and open authorities (OA) are identity based.

Correctness: The *Verification Correctness* means honestly signed signatures should pass verification with overwhelming probability. The *Opening Correctness* means honestly signed signatures should honestly open to the actual signer. The group signature is *correct* if has verification correctness and opening correctness.

3.2 Attacker tools: oracles

We have the following oracles for the adversary to query:

- The *Random Oracle* $\mathcal{H}$: Ordinary random oracle.
- The *User Corruption Oracle* $\mathcal{UCO}$: gets user secrete key.
- The *Group Corruption Oracle* $\mathcal{GCO}$: gets manager secret key.
- The *Escrow Corruption Oracle* $\mathcal{ECO}$: gets OA's secret key.
- The *Signing Oracle* $\mathcal{SO}$: $(U, M) \to \sigma$.
- The *Decisional Open Oracle* $\mathcal{DOO}$: $(\sigma, U) \to 1$ or 0 for U *generated the signature* σ or not.

We use the *static adversary* in this paper, where the Adversary $\mathcal{A}$ corrupts users (resp. the group manager, the OA) only at the beginning of the security Experiments, and the Simulator $\mathcal{S}$ knows which users are corrupted. Issues concerning adaptive adversaries, reset adversaries, or UC (universal composability) adversaries are left to future research.

3.3 Security notions and experiments

3.3.1 Anonymity

Experiment Anon

1. *Initialization Phase*: Simulator $\mathcal{S}$ initializes.
2. *Probe-1 Phase*: $\mathcal{A}$ queries $\mathcal{GCO}$ and makes q_U queries to $\mathcal{UCO}$. Then it queries $\mathcal{H}$, $\mathcal{SO}$, and $\mathcal{DOO}$ in arbitrary interleaf.
3. *Gauntlet Phase*: $\mathcal{A}$ selects two uncorrupted users, U_0 and U_1, (called *gauntlet users* generated in the Initialization Phase, a message M, and gives them to $\mathcal{S}$. Then $\mathcal{S}$ randomly chooses $b \in \{0, 1\}$ and returns the *gauntlet signature* $\sigma = \mathsf{GSig}(\mathsf{sk}_{U_b}, \mathsf{cert}_{U_b}, M)$.
4. *Probe-2 Phase*: $\mathcal{A}$ queries $calH$, $\mathcal{SO}$, and $\mathcal{DOO}$ in arbitrary interleaf, except it cannot query the gauntlet signature to $\mathcal{DOO}$.
5. *End Game*: $\mathcal{A}$ delivers an estimate $\hat{b} \in \{0, 1\}$ of b.

$\mathcal{A}$ *wins* Experiment Anon if $\hat{b} = b$. $\mathcal{A}$'s *advantage* is his probability of winning minus $1/2$.

Definition 1. *A group signature is* fully-CCA2-anonymous *if no PPT algorithm has a non-negligible advantage in Experiment Anon.*

Remark: The most dangerous threat summarized by Experiment Anon is the Adversary $\mathcal{A}$'s ability to query $\mathcal{DOO}(\sigma, U)$ in the Probe-2 Phase, even with $U = U_1$ or U_2. The ability to withstand such attacks highlight the strength of the full-CCA2-anonymity. Note that making queries $\mathcal{DOO}(\sigma, U)$ for every U generated in the Initialization Phase implements a full-fledged Open Oracle. Therefore there is no dilution in our definition of the full-CCA2-anonymity from those in the literature [12, 41, 49]. The treatment of queries $\mathcal{DOO}(\sigma, U')$ where σ is a valid signature by an illegitimate user U' (i.e. an user who is not generated in the Initialization Phase but is concocted by the Adversary) is an interesting issue for future research. In this paper, $\mathcal{DOO}$ outputs 0 in our static group model.

3.3.2 Full Traceability.

Experiment FT

1. *Initialization Phase*: $\mathcal{S}$ initializes. $\mathcal{A}$ makes q_U queries to $\mathcal{UCO}$.
2. *Probe Phase*: $\mathcal{A}$ queries $\mathcal{H}$, $\mathcal{SO}$, $\mathcal{DOO}$ in arbitrary interleaf.
3. *Delivery Phase*: $\mathcal{A}$ delivers a signature σ which is not an $\mathcal{SO}$ output.

$\mathcal{A}$ *wins* Experiment FT if $\mathsf{GVf}(\sigma) = 1$ and $\mathsf{Open}(\mathsf{sk}_{OA}, \sigma)$ does not output a user corrupted in the Initialization Phase by $\mathcal{A}$. $\mathcal{A}$'s *advantage* is his probability of winning.

Definition 2. *A group signature is* fully traceable *if no PPT algorithm has a non-negligible advantage in Experiment FT.*

3.3.3 Non-frameability.

Experiment NF

1. *Initialization Phase*: $\mathcal{S}$ initializes. $\mathcal{A}$ queries $\mathcal{GCO}$ and $\mathcal{ECO}$, and makes q_U queries to $\mathcal{UCO}$.
2. *Probe Phase*: $\mathcal{A}$ queries $\mathcal{H}$, $\mathcal{SO}$, $\mathcal{DOO}$ in arbitrary interleaf.
3. *Delivery Phase*: $\mathcal{A}$ delivers a valid σ along with a proof π that it opens to U, an uncorrupted user.

$\mathcal{A}$ *wins* Experiment NF if $\mathsf{Judge}(\sigma, U, \pi) = 1$. Its *advantage* is its probability of winning.

Definition 3. *A group signature is* non-frameable *if no PPT algorithm has a non-negligible advantage in Experiment NF.*

3.3.4 Security In summary,

Definition 4. *A group signature is a* secure CCA2-fully anonymous group signature *if it is CCA2-fully-anonymous, fully traceable, and non-frameable.*

4 Construction

We construct CCA2-fully-anonymous group signatures using only IND-CPA-secure encryptions to escrow the signer identity to the OA.

4.1 Generic construction: Protocol CCAGS-gen

The generic CCA2-fully-anonymous group signature construction uses an IND-CPA-secure public-key encryption Enc is as follows:

$$\begin{aligned}&SPK\{(\mathsf{cert}, \mathsf{pk}_U, \mathsf{sk}_U, \rho) : \\ &\quad \mathsf{cert} \text{ is valid on } (\mathsf{pk}_U, \mathsf{sk}_U) \ \wedge\ \mathsf{ctxt} = \mathsf{Enc}(\mathsf{pk}_U, \mathsf{pk}_{OA}, \rho) \ \wedge\ S = \mathsf{VRF}(R, \mathsf{sk}_U)\}(M, \mathsf{param}, R)\end{aligned} \tag{3}$$

where VRF is a Verifiable Random Function (VRF) [43, 37, 35, 18].

4.2 Instantiation in pairings: Protocol CCAGS-SDH($\mathbb{G}_1$, G_S).

We instantiate the generic construction above in the pairings framework. Let $\hat{\mathbf{e}} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_3$ be a pairing where order($\mathbb{G}_1$) = order($\mathbb{G}_2$) = order($\mathbb{G}_3$) = q_1. The OA's sk-pk is $(x_E, \mathbf{g}_E^{x_E}) \in Z_{q_1} \times \mathbb{G}_3$. The CA (GM)'s sk-pk pair is $(\gamma, u^\gamma) \in Z_{q_1} \times \mathbb{G}_2$. The certificate on user sk-pk pair $(x, h^x) \in Z_{q_1} \times \mathbb{G}_1$ is (A, e) satisfying $A^{e+\gamma} h^x = h_0$. All discrete logarithm bases are fairly generated. We use the CPA-secure ElGamal encryption to escrow signer identity to OA. Assume all discrete logarithm bases are fairly generated, e.g. $g_i = \mathcal{H}('g', i)$, $h_i = \mathcal{H}('h', i)$, $\mathbf{g}_i = \mathcal{H}('\mathbf{g}', i)$.

Let G_S be a known-order group slightly larger than z_{q_1}. We have in mind $G_S = Z_p$ with a generator $\mathbf{g}$ whose order at least as large as q_1. This choice reduces the signature's bandwidth. But the more complicated technique of proving equality of discrete logarithm between two groups is required [15]. Another choice is $G_S = \mathbb{G}_3$. This choice is easier to understand but the signature bandwidth is increased significantly because one element of $\mathbb{G}_3$ costs a multiple number of bits to transmit than one element of $\mathbb{G}_1$. The signature is

$$\begin{aligned} &SPK\{(A,e,x,\rho) : A^{e+\gamma}h^x = h_0 \in \mathbb{G}_1 \\ &\quad \wedge\ \kappa_1 = g_E^{\rho}\ \wedge\ \kappa_2 = \hat{\mathbf{e}}(h^x, g_{E,2})\mathbf{g}_E^{x_E\rho}\ \wedge\ S = \mathbf{g}_S^{1/(R+x)} \in G_S\}(M, \mathsf{param}, R) \end{aligned} \tag{4}$$

Using linear encryption let $g_1 = g_3^{1/a}$, $g_2 = g_3^{1/b}$, where $\mathsf{sk}_{OA} = (a,b)$. In instantiation details, the commitments are

$$\begin{gathered} T_1 = g_1^{s_1},\quad T_2 = g_2^{s_2},\quad T_3 = Ag_3^{s_1+s_2} \in \mathbb{G}_1, \\ \tilde{T}_3 = \hat{\mathbf{e}}(T_3^{-1}, u^{\gamma})\hat{\mathbf{e}}(h_0, u) = \hat{\mathbf{e}}(T_3, u)^{e}\hat{\mathbf{e}}(h, u)^{x}\hat{\mathbf{e}}(g_3, u^{\gamma})^{-(s_1+s_2)}\hat{\mathbf{e}}(g_3, u)^{-s_3}, \qquad (5) \\ T_4 = \mathbf{g}_S^{1/(R+x)},\quad T_5 = 1 = T_4^{R+x}\mathbf{g}_S^{-1} \\ D_1 = g_1^{r_1},\quad D_2 = g_2^{r_2}, \\ D_3 = \hat{\mathbf{e}}(T_3, u)^{r_e}\hat{\mathbf{e}}(h, u)^{r_x}\hat{\mathbf{e}}(g_3, u^{\gamma})^{-(r_1+r_2)}\hat{\mathbf{e}}(g_3, u)^{-r_3}\tilde{T}_3^{c}, \\ D_4 = \mathbf{g}_S^{r_4},\quad D_5 = T_4^{r_x} \qquad (6) \end{gathered}$$

The challenge is

$$c = \mathcal{H}(\mathsf{param}, R, M, T_1, T_2, T_3, T_4, D_1, D_2, D_3, D_4, D_5). \tag{7}$$

The responses are

$$\begin{gathered} z_1 = r_1 - cs_1,\quad z_2 = r_2 - cs_2,\quad z_e = r_e - ce, \\ z_x = r_x - cx,\quad z_3 = r_3 - cs_3,\quad z_4 = r_4 - c/(R+x) \end{gathered} \tag{8}$$

where $s_3 = e(s_1 + s_2)$. The signature is

$$\sigma = (\mathsf{param}, R, M, T_1, T_2, T_3, T_4, c, z_1, z_2, z_3, z_4, z_e, z_x). \tag{9}$$

param and the nonce R can be considered known or pre-protocoled to GVf and omitted from the transmission.

Protocol $\mathsf{GVf}(\sigma)$: Parse σ and compute

$$\begin{gathered} D_1 = g_1^{z_1}T_1^{c},\quad D_2 = g_2^{z_2}T_2^{c}, \\ \tilde{T}_3 = \hat{\mathbf{e}}(T_3^{-1}, u^{\gamma})\hat{\mathbf{e}}(h_0, u),\quad D_3 = \hat{\mathbf{e}}(T_3, u)^{z_e}\hat{\mathbf{e}}(h, u)^{z_x}\hat{\mathbf{e}}(g_3, u^{\gamma})^{-(z_1+z_2)}\hat{\mathbf{e}}(g_3, u)^{-z_3}\tilde{T}_3^{c}, \\ D_4 = \mathbf{g}_S^{z_4}T_4^{c},\quad D_5 = T_4^{z_x}(T_5\mathbf{g}_S T_4^{-R})^{c} \end{gathered} \tag{10}$$

Protocol $\mathsf{Open}(\sigma, \mathsf{sk}_{OA})$: Compute $A = T_1^{-a}T_2^{-b}T_3$ where $\mathsf{sk}_{OA} = (a,b)$. Also publish a proof-of-opening:

$$PK\{(a,b) : A = T_1^{-a}T_2^{-b}T_3\ \wedge\ g_1^{a} = g_2^{b} = g_3\} \tag{11}$$

Protocol Judge verifies the proof-of-opening.

The **security analysis** is in the following Theorem, whose proof is sketched in Appendix B.

Theorem 2. *Let $\hat{\mathbf{e}} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_3$ be a pairing and $G_S = \langle \mathbf{g}_S \rangle$ be a known-order group at least as big as $\mathbb{G}_1$. Let q_S (resp. q_U, q_H) be the number of Signing Oracle (resp. User Corruption Oracle, Random Oracle) queries. Protocol CCAGS-SDH($\mathbb{G}_1$, G_S) is a group signature which is, assuming the random oracle (RO) model,*

1. corect*;*
2. CCA2-fully-anonymous *provided the q_S-SDDH($\mathbb{G}_1$,G_S) Assumption and the DLDH($\mathbb{G}_1$) Assumption both hold;*

3. full traceable *provided the q_U-SDH($\mathbb{G}_1$) Assumption or the q_S-SDH($\mathbb{G}_1$, G_S) Assumption holds;*

4. non-frameable *provided the Discrete Logarithm Assumption holds in $\mathbb{G}_1$.*

In summary, the CCAGS-SDH group signature is a secure CCA2-fully-anonymous *group signature provided the q_S-SDDH($\mathbb{G}_1$,G_S) Assumption and the DLDH($\mathbb{G}_1$) both hold in the RO model.*

Corollary 3 *The CCAGS-SDH($\mathbb{G}_1$, $\mathbb{G}_3$) group signature is a* secure CCA2-fully-anonymous group signature *provided the q_S-SDDH($\mathbb{G}_1$, $\mathbb{G}_3$) Assumption and the DLDH($\mathbb{G}_1$) Assumption both hold.*

Efficiency discussions: The CCAGS-SDH($\mathbb{G}_1$, $\mathbb{G}_3$) group signature has a bandwidth of approximately what it costs to transmit 10 rational points in $\mathbb{G}_1$ and one point in $\mathbb{G}_3$. The bandwidth of one point in $\mathbb{G}_3$ can be made as small as two in $\mathbb{G}_1$. The CCAGS-SDH($\mathbb{G}_1$, G_S) signature's bandwidth costs about $10\lambda_s$ plus the bandwidth of one G_S element, which is slightly larger than λ_s. Using pairings with 170-bit curves, the total signature bandwidth is approximately 1870 bits. In comparison, the shortest CCA2-fully-anonymous group signature that uses IND-CCA2 encryption costs $15\lambda_s$ bits, see our Section 5. Our theoretical advance of being able to use CPA-encryption saves about 27%. Note an IND-CCA2-secure RSA encryption in the 1024-bit RSA framework cost at least 2048 bits. Following convention, we do not count published system parameters param, pre-protocoled nonce R, and the message M in comparing lengths. They are either included or excluded from both sides of the comparison. Many protocols use nonces (or should use this technology) even if not explicited stated so. All of GSig's computations can be offline pre-computed. The online computation of GVf consists of 5 multi-base exponentiations and one pairings. Note the one pairing is the very compressed $\hat{\mathbf{e}}(T_3, u)^{z_e}\hat{\mathbf{e}}(T_3^{-1}, u^\gamma)^c = \hat{\mathbf{e}}(T_3, u^{z_e}u^{-\gamma c})$, using a technique from [12]. The pairing is expensive to compute. The previous shortest CCA2-fully-anonymous group signature costs 1 pairing and 8 multi-base exponentiations online.

4.3 Instantiation in strong RSA: Protocol CCAGS-SRSA(QR_N, G_S)

We instantiate in the strong RSA framework. The result is the fastest CCA2-fully anonymous group signature. Given security parameter 1^{λ_s}, initializes additional security parameters $\epsilon > 1$, λ_p, λ_1, λ_2, γ_1, γ_2 satisfying $\lambda_2 > 4\ell_p$, $\lambda_1 > \epsilon(\lambda_2 + \lambda_s) + 2$, $\gamma_2 > \lambda_1$, $\gamma_1 > \epsilon(\gamma_2 + \lambda_s) + 2$. Note ℓ_p sets the size of the modulus of the RSA framework, ϵ controls the tightness of the zero-knowledge [3]. Generate as product N of two ℓ_p-bit safe primes p and q, i.e. $p = 2p' + 1$, $q = 2q' + 1$, p' and q' are both primes. Generate a known-order group $G_S = \langle \mathbf{g} \rangle$, where order$(G_S) = p_s$ which is a $(\gamma_1 + 1)$-bit prime. The group sk-pk pair is $((p, q), N)$. Generates a user list UL which is initially empty. Let all discrete logarithm bases be fairly generated, e.g. $g_i = \mathcal{H}('g', i) \in QR_N$, $h_i = \mathcal{H}('h', i) \in QR_N$, $\mathbf{g}_i = \mathcal{H}('\mathbf{g}', i) \in G_S$. We adopt the flexible notation $\mathcal{H}$ which is a full-domain collision-resistant hash functin mapping from a union of mixed domains to a union of mixed ranges. Ambiguity should not arise from the context. Denote the intervals $\Lambda =]2^{\lambda_1} - 2^{\lambda_2}, 2^{\lambda_1} + 2^{\lambda_2}[$ and $\Gamma =]2^{\gamma_1} - 2^{\gamma_2}, 2^{\gamma_1} + 2^{\gamma_2}[$.

The OA's sk-pk is $(x_E, g_E^{x_E})$. The CA (GM)'s sk-pk pair is $((p, q), N)$. The certificate on user public key h^x is (A, e) satisfying $A^e h^x = h_0 \in QR_N$, where $e \in \Lambda$ is a prime and $x \in \Gamma$. We follow the procedures in [21] where $x = 2^{\lambda_1} + 2(x' + x'')$ where $x', x'' \in] - 2^{\lambda_2 - 2}, 2^{\lambda_2 - 2}[$, x' is generated by the user in privace, x'' is randomly generated by the GM and given to the user when the user join the group. All discrete logarithm bases are fairly generated, e.g $g_i = \mathcal{H}('g', i)$. We will use the CPA-secure ElGamal encryption to escrow signer identity to OA. The signature is

$$\begin{aligned} &SPK\{(A, e, x, \rho) : x_2, x_2 \in \Lambda \ \wedge \ e \in \Gamma \\ &\quad \wedge \ A^e h^x = h_0 \ \wedge \ \kappa_1 = g_E^\rho \ \wedge \ \kappa_2 = A g_E^{\rho x_E} \ \wedge \ S = \mathbf{g}_S^{1/(R+x)} \in G_S\}(M, \mathsf{param}, R) \end{aligned} \tag{12}$$

Below, $g_0 = g_E$, $g_A = g_E^{s_E}$. In further instantiation details, the commitments are (omitted range proofs for simplicity)

$$\begin{aligned} T_0 = g_0^{s_0}, \ \ T_A = A g_A^{s_0}, \ \ T_2 = g_{2,1}^e g_{2,2}^{s_0}, \ \ T_3 = \mathbf{g}_S^{1/(R+x)} \in G_S, \\ D_0 = g_0^{r_0}, \ \ D_A = T_0^{r_e} h^{r_x} g_A^{-r_1}, \ \ D_1 = T_0^{r_e} g_0^{-r_1}, \ \ D_2 = g_{2,1}^{r_e} g_{2,2}^{r_0}, \ \ D_3 = T_3^{r_x}. \end{aligned} \tag{13}$$

Note $h_0 = T_0^e h^x g_A^{-s_1}$, $s_1 = es_0$, $1 = T_0^e g_0^{-s_1}$. The challenge is

$$c = \mathcal{H}(\mathsf{param}, R, M, T_0, T_A, T_2, T_3, D_0, D_A, D_1, D_2, D_3). \tag{14}$$

The responses are

$$z_0 = r_0 - cs_0, \quad z_1 = r_1 - cs_1, \quad z_e = r_e - ce, \quad z_x = r_x - cx, \tag{15}$$

The signature is

$$\sigma = (\mathsf{param}, R, M, T_0, T_A, T_2, T_3, c, z_0, z_1, z_e, z_x). \tag{16}$$

The system parameters param and the nonce R are considered known or pre-protocoled to GVf and can be omitted from the transmission of the signature.

Protocol $\mathsf{GVf}(\sigma)$: Parse σ and compute

$$D_0 = g_0^{z_0} T_0^c, \quad D_A = T_A^{z_e} h^{z_x} g_A^{-z_1} h_0^c, \quad D_1 = T_0^{z_0} g_1^{-z_1}, \quad D_2 = g_{2,1}^{z_e} g_{2,2}^{z_0} T_2^c, \quad D_3 = T_3^{z_x} (\mathbf{g}_S T_3^{-R})^c \tag{17}$$

Verify the received challenge c equals to that computed by Equation (14).

Protocol $\mathsf{Open}(\sigma, \mathsf{sk}_{OA})$: Compute $A = T_0^{-x_E} T_A$ where $\mathsf{sk}_{OA} = x_E$. Also publish a proof-of-opening:

$$PK\{x_E : A = T_0^{-x_E} T_A \ \wedge \ g_0^{x_E} = g_A\} \tag{18}$$

Protocol Judge verifies the proof-of-opening.

Security Analysis of Protocol CCAGS-SRSA(QR_N, G_S) is in the following Theorem.

Theorem 4. *Protocol CCAGS-SRSA(QR_N, G_S) is a group signature which, assuming the RO model, is*

1. correct*;*
2. CCA2-fully-anonymous *provided the q_S-SDDH(QR_N,G_S) Assmption and the DDH(QR_N) Assumption both hold;*
3. full traceable *provided the q_S-SDH(QR_N, G_S) Assumption or the Strong RSA Assumption holds;*
4. non-frameable *provided the Discrete Logarithm Assumption holds in QR_N.*

In summary Protocol CCAGS-SRSA(QR_N, G_S) is a secure CCA2-fully-anonymous *group signature provided the q_S-SDDH(QR_N, G_S) Assumption and the DDH(QR_N) Assumption both hold in the RO model.*

Efficiency discussions: The signature verification costs 4 multi-base exponentiations in QR_N. In comparison, the fastest CCA2-fully-anonymous group signature that uses IND-CCA2 encryption costs 5 multi-base exponentiations, see our Section 5. Our theoretical advancement of being able to use CPA-encryption saves about 25%. All of GSig's computations can be offline pre-computed. The online computation of GVf consists of one pairing and 5 multi-base exponentiations. Pairing are quite expensive to compute in general.

Proof is similar to that of Theorem 2 and is postponed to the full version of this paper. *Proof intuitions*: Since our construction adds additional relations to classic constructions [3, 20, 41], the soundness tends to be enhanced, but the anonymity faces pressure. About soundsness: An attacker must forge all components of the signature. The forging of each component usually reduces to an intractability assumption. Therefore full traceability (resp. non-frameability), which is a form of coalition-resistance unforgeability, typically reduces to the holding of all of several intractability assumptions.

5 Discussions and Conclusions

We describe how to extend our CCA2-fully-anonymous group signatures, Protocols CCAGS-SDH and CCAGS-SRSA, to support other group signature features, such as revocation [6, 20, 12, 39, 48], identity-based users and/or OA [48, 56], trapdoor-freeness [4, 49], anonymous authentication in *ad hoc* groups [34], ring signatures [31, 51], tracing-by-linking (TbL) group signatures [55, 54, 18].

5.1 Revocation

Revocation is to prevent blacklisted group members from generating valid signatures. It is a challenge to revoke anonymous memberships, but the literature contains solutions. There are two major approaches. In the *dynamic accumulator* approach [6, 20, 12, 48], the GM alters its public key and publishes certain information to the general membership in order to blacklist one member. Each remaining member must alter their group membership certificate according to the published information in order to continue to be able to generate valid group signatures – its old certificate is no longer valid against the GM's new public key. The blacklisted member cannot alter its certificate into one that will be able to generate additional signatures valid against the GM's new public key.

In the other approach, called *traceable signature* by [39] and called *Verifier Local Revocation (VLR)* by [14], the members never modify their originally issued group membership certificates. Instead, the GM publishes a CRL (Certificte Revocation List) containing information about the entirety of blacklisted members. The Verifier checks otherwise-valid signatures against the CRL for revoked members. Either the dynamic accumulator approach or the VRL approach has relative efficiencies and drawbacks, in addition to related synchronization and latency issues [6, 20, 12, 39, 14, 48].

In the following, we extend [39]'s revocation mechanism to our group signatures. In the CCAGS-SRSA group signature, when a user joins the group in the $(\mathsf{Join}, \mathsf{Iss})$ interactive protocol with the GM, it presents the user public key $h^{x'}$, prove knowledge of its discrete logarithm x', and is issued a certificate (A, e) satisfying $A^e h^x = h_0$ by the GM, where $x = 2^{\lambda_1} + 2x' + 2x''$ where $x', x'' \in]-2^{-\lambda_2-2}, 2^{-\lambda_2-2}[$ and $e \in \Gamma$ is a prime. In order to support VLR, the $(\mathsf{Join}, \mathsf{Iss})$ interactive protocol is modified as follows:

Protocol $(\mathsf{Join}, \mathsf{Iss})$

1. The user presents its public key $(h_1^{x_1'}, h_2^{x_2'})$, and a verifiable encryption to OA of x_2';
2. prove knowledge of the discrete logarithms x_1' and x_2', prove the range x_1' and $x_2' \in]-2^{-\lambda_2-2}, 2^{-\lambda_2-2}[$, prove correctness of the verifiable encryption;
3. The GM verifies the proofs; randomly generates x_1'' and $x_2'' \in]-2^{-\lambda_2-2}, 2^{-\lambda_2-2}[$, generates a certificate (A, e) satisfying $A^e h_1^{x_1} h_2^{x_2} = h_0$, where $e \in \Gamma$ is a prime.
4. Upon receiving A, e, x_1'', x_2'', the user verifies and completes.

The signature is, called **Protocol CCAGS-VRF**,

$$\sigma = SPK\{(A, e, x_1, x_2) : A^e h_1^{x_1} h_2^{x_2} = h_0 \ \wedge \ x_1, x_2 \in \Lambda \ \wedge \ e \in \Gamma \ \wedge \ S = \mathbf{g}^{1/(R+x_2)}\}(M) \tag{19}$$

The remaining instantiation details are similar to those of Protocol CCAGS-SRSA and omitted. Protocol CCAGS-VRF is formulated only in the strong RSA framework above. But it can be easily converted to the pairings framework. Use the parameters from Protocol CCAGS-SDH. Protocol CCAGS-VRF-SDH has the following signature:

$$\sigma = SPK\{(A, e, x_1, x_2) : A^{e+\gamma} h_1^{x_1} h_2^{x_2} = h_0 \in \mathbb{G}_1 \ \wedge \ S = \mathbf{g}^{1/(R+x_2)} \in \mathbb{G}_3\}(M) \tag{20}$$

To open a signature σ, the OA checks (R, S) against every x_2 in its database for a match $S = \mathbf{g}^{1/(R+x_2)}$. When a match is found, the member who owns x_2 is the signer. To revoke that member, the OA publishes its x_2 to the CRL. Then GVf checks an otherwise valid signature σ's S against the CRL, where $S = \mathbf{g}^{1/(R+x_2)}$ indicates a blacklisted user.

Effieincy discussions. The above CCA2-fully-anonymous group signature has highly efficient signature generation and signature verification. The tradeoff is that the signature opening protocol becomes inefficient: it requires an exhaustive search of x_2 for every user in its database. In scenarios where signature opening is infrequent or offline, the above scheme achieves significant online signature generation and signature verification. Overall, it achieves a tradeoff of online efficiency of frequently-used protocols against offline or infrequent signature opening protocol.

5.2 Identity-based

In identity-based encryption (IBE) [53, 13], the public encryption key is the decryptor's identity (or a hashed value of it). The decryption key is derived from the identity by a TTP called the *key extractor*. The IBE can be viewed as a *post-certified* public key cryptosystem. It achieves various convenience and security tradeoffs versus the more common *pre-certified* public key cryptosystem where each individual must have its public key certified by the CA (Certificate Authority) prior to conducting authenticated secure interaction with another entity. Identity-based identification (IBI) and identity-based signatures (IBS) are also important research topics [53, 8]. Identity-based group signatures, where the user and/or the OA are identity-based, have also aroused research interests [56, 48].

Protocol CCAGS-VRF above can be easily modified to support identity-based user and identity-based OA. To support **identity-based OA**: First note that Protocol CCAGS-VRF does not verifiably encrypt the signer's identity in the signature itself, in the style of state-of-the-art group signatures in [7, 10, 41]. Therefore, not modification to Protocol CCAGS-VRF's GSig is needed to support identity-based OA. Rather, the attention to modification should be directed to (Join, Iss). There, a user to join the group must verifiably encrypt its second secrete key x_2' to the OA. The state-of-the-art verifiable encryption to an identity-based decryptor is the general-purpose verfiable encryption of Camenisch and Damgard [17]. It can be used here. Its complexity is $O(\lambda_s)$.

To support **identity-based group members**: Let $\mathcal{R}$ denote the relation specifying members' identity-based sk-id pairs. Protocol CCAGS-VRF can be modified as follows to support identity-based group members:

$$\sigma = SPK\{(A, e, x_1, x_2, \mathsf{sk}, \mathsf{id}) : A^{e+\gamma} h_1^{x_1} h_2^{x_2} = h_0 \in \mathbb{G}_1 \ \wedge \ S = \mathbf{g}^{1/(R+x_2)} \in \mathbb{G}_3 \ \wedge \ \}(M) \qquad (21)$$

5.3 Trapdoor-free group signatures

5.4 Anonymous authentication in *ad hoc* groups

5.5 Ring signatures

5.6 Tracing-by-linking (TbL) group signatures

5.7 Conclusion:

We prove one can construct CCA2-fully-anonymous group signatures by using only IND-CPA-secure encryptions to escrow the signer identity to the OA (Open Authority). The benefits of this theoretical advance significantly improves the efficiency. Instantiating in pairings, we construct the new shortest CCA2-fully-anonymous group signature which is 27% shorter than the previous shortest. Instantiating in the strong RSA framework, we construct the new fastest CCA2-fully-anonymous group signature which is 25% faster than the previous fastest.

References

1. M. Abe and S. Fehr. Adaptively secure feldman vss and applications to universally-composable threshold cryptography. In *CRYPTO '05*, volume 3152 of *LNCS*, pages 317–334. Springer-Verlag, 2005.
2. M. Abe, M. Ohkubo, and K. Suzuki. 1-out-of-n signatures from a variety of keys. In *Proc. ASIACRYPT 2002*, pages 415–432. Springer-Verlag, 2002. Lecture Notes in Computer Science No. 2501.
3. G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik. A practical and provably secure coalition-resistant group signature scheme. In *Proc. CRYPTO 2000*, pages 255–270. Springer-Verlag, 2000. Lecture Notes in Computer Science No. 1880.
4. G. Ateniese and B. de Medeiros. Efficient group signatures without trapdoors. In *ASIACRYPT 2003*, volume 2894 of *LNCS*, pages 246–268. Springer-Verlag, 2003.
5. G. Ateniese, D. Song, and G. Tsudik. Quasi-efficient revocation in group signatures. In *Financial Crypt. 2002*, volume 2357 of *LNCS*, pages 183–197. Springer-Verlag, 2002.
6. N. Baric and B. Pfitzmann. Collision-free accumulators and fail-stop signature schemes without trees. In *EUROCRYPT 97*, volume 1233 of *LNCS*, page 480C494. Springer-Verlag, 1997.

7. M. Bellare, D. Micciancio, and B. Warinschi. Foundations of group signatures: formal definitions, simplified requirements and a construction based on general assumptions. In *EUROCRYPT'03*, volume 2656 of *LNCS*. Springer-Verlag, 2003.
8. M. Bellare, C. Namprempre, and G. Neven. Security proofs for identity-based identification and signature schemes. In *Proc. EUROCRYPT 2004*, pages 268–286. Springer-Verlag, 2004. Lecture Notes in Computer Science No. 3027.
9. M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proc.* 1*st ACM Conference on Computer and Communications Security*, pages 62–73. ACM Press, 1993.
10. M. Bellare, H. Shi, and C. Zhang. Foundations of group signature: The case of dynamic groups. Lecture Notes in Computer Science No. 3376, 2005.
11. D. Boneh and X. Boyen. Short signatures without random oracles. In *Proc. EUROCRYPT 2004*, pages 56–73. Springer-Verlag, 2004. Lecture Notes in Computer Science No. 3027.
12. D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In *Proc. CRYPTO 2004*, pages 41–55. Springer-Verlag, 2004. Lecture Notes in Computer Science No. 3152.
13. D. Boneh and M. Franklin. Identity-based encryption from the weil paring. In *Proc. CRYPTO 2001*, pages 213–229. Springer-Verlag, 2001. Lecture Notes in Computer Science No. 2139.
14. D. Boneh and H. Shacham. Group signatures with verifier-local revocation. In *ACM Conference on Computer and Communications Security 2004*, pages 168–177, 2004.
15. F. Boudot. Efficient proofs that a committed number lies in an interval. In *Eurocrypt'00*, pages 431–444, 2000.
16. E. Brickell, J. Camenisch, and L. Chen. Direct anonymous attestation. Cryptology ePrint Archive, Report 2004/205, 2004. http://eprint.iacr.org/.
17. J. Camenisch and I. Damgård. Verifiable encryption, group encryption, and their applications to separable group signatures and signature sharing schemes. In *Proc. ASIACRYPT 2000*, pages 331–345. Springer-Verlag, 2000. Lecture Notes in Computer Science No. 1976.
18. J. Camenisch, S. Hohenberger, and A. Lysyanskaya. Compact e-cash. In *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 302–321. Springer-Verlag, 2005. Also http://eprint.iacr.org/2005/060.
19. J. Camenisch and A. Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *Proc. EUROCRYPT 2001*, pages 93–118. Springer-Verlag, 2001. Lecture Notes in Computer Science No. 1294.
20. J. Camenisch and A. Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In *CRYPTO 2002*, volume 2442 of *LNCS*, pages 61–76. Springer-Verlag, 2002.
21. J. Camenisch and A. Lysyanskaya. A signature scheme with efficient protocols. In *SCN 2002*, volume 2576 of *LNCS*, pages 268–289. Springer-Verlag, 2003.
22. J. Camenisch and M. Michels. Separability and efficiency for generic group signature schemes. In *Proc. CRYPTO 99*, pages 413–430. Springer-Verlag, 1999. Lecture Notes in Computer Science No. 1666.
23. J. Camenisch and M. Stadler. Efficient group signature schemes for large groups. In *Proc. CRYPTO 97*, pages 410–424. Springer-Verlag, 1997. Lecture Notes in Computer Science No. 1294.
24. J. Camenisch and M. Stadler. Proof systems for general systems of discrete logarithms. ETH Technical Report No. 260, 1997. ftp://ftp.inf.ethz.ch/pub/publications/tech-reports/.
25. R. Canetti, R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin. Adaptive security for threshold cryptosystems. In *CRYPTO 1999*, volume 1666 of *LNCS*, pages 98–115. Springer-Verlag, 1999.
26. A. Chan, Y. Frankel, and Y. Tsiounis. Easy come - easy go divisible cash. In *EUROCRYPT '98*, volume 1403 of *LNCS*, pages 561–575. Springer-Verlag, 1998.
27. D. Chaum and T. P. Pedersen. Wallet databases with observers. In *EUROCRYPT '92*, volume 658 of *LNCS*, pages 89–105. Springer-Verlag, 1992.
28. D. Chaum and E. van Heyst. Group signatures. In *EUROCRYPT'91*, volume 547 of *LNCS*, pages 257–265. Springer-Verlag, 1991.
29. L. Chen and T. Pedersen. New group signature schemes. In *Proc. EUROCRYPT 94*, pages 171–181. Springer-Verlag, 1994. Lecture Notes in Computer Science No. 950.
30. L. Chen and T.P. Pedersen. On the efficiency of group signatures providing information-theoretic anonymity. In *Eurocrypt '95*, volume 921 of *LNCS*, pages 39–49. Springer-Verlag, 1995.
31. R. Cramer, I. Damgard, and B. Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *CRYPTO'94*, pages 174–187. Springer-Verlag, 1994.
32. R. Cramer and V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *Proc. CRYPTO 98*, pages 13–25. Springer-Verlag, 1998. Lecture Notes in Computer Science No. 1462.
33. I. Damgard and E. Fujisaki. An integer commitment scheme based on groups with hidden order. In *ASIACRYPT 2002*, volume 2501 of *LNCS*, pages 125–142. Springer-Verlag, 2004.
34. Y. Dodis, A. Kiayias, A. Nicolosi, and V. Shoup. Anonymous identification in ad hoc groups. In *Proc. EUROCRYPT 2004*, pages 609–626. Springer-Verlag, 2004. Lecture Notes in Computer Science No. 3027.

35. Y. Dodis and A. Yampolskiy. A verifiable random function with short proofs and keys. In *PKC 2005*, pages 416–431. Springer-Verlag, 2005. Lecture Notes in Computer Science No. 3386.
36. E. Fujisaki and T. Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. In *CRYPTO'97*, pages 16–30. Springer-Verlag, 1997.
37. S. Jarecki and V. Shmatikov. Handcuffing Big Brother: an abuse-resilient transaction escrow scheme. In *Eurocrypt 2004*, volume 3027 of *LNCS*, pages 590–608. Springer-Verlag, 2004.
38. A. Joux and K. Nguyen. Separating Decision Ddiffie-Hellman from Computational Diffie-Hellman in cryptographic groups. *J. of Cryptology*, 16(4):239–247, 2003.
39. A. Kiayias, Y. Tsiounis, and M. Yung. Traceable signatures. In *Eurocrypt 2004*, LNCS, pages 571–589. Springer-Verlag, 2004.
40. A. Kiayias and M. Yung. Extracting group signatures from traitor tracing schemes. In *Eurocrypt 2003*, LNCS, pages 630–648. Springer-Verlag, 2003.
41. A. Kiayias and M. Yung. Group signatures: provable security, efficient constructions, and anonymity from trapdoor-holders. Cryptology ePrint Archive, Report 2004/076, 2004. http://eprint.iacr.org/.
42. J. K. Liu, V. K. Wei, and D. S. Wong. Linkable and culpable ring signatures. *eprint*, 2004(027), 2004.
43. S. Micali, M. Rabin, and S. P. Vadhan. Verifiable random functions. In *FOCS '99*, pages 120–130. IEEE Press, 1999.
44. S. Mitsunari, R. Sakai, and M. Kasahara. A new traitor tracing. IEICE Trans. Fundamentals, E85-A(2):481-4, 2002.
45. T. Nakanishi, T. Fujiwara, and H. Watanabe. A linkable group signature and its application to secret voting. In *4th Int'l Symp. on Communicatin Theory and Appl.*, 1997.
46. T. Nakanishi, T. Fujiwara, and H. Watanabe. A linkable group signature and its application to secret voting. *Trans. of Information Processing Society of Japan*, 40(7):3085–3096, 1999.
47. M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *STOC 1990*, pages 427–437. ACM Press, 1990.
48. L. Nguyen. Accumulators from bilinear pairings and applications. In *CT-RSA 2005*, volume 3376 of *LNCS*, pages 275–292. Springer-Verlag, 2005.
49. L. Nguyen and R. Safavi-Naini. Efficient and provably secure trapdoor-free group signature schems from bilinear pairings. In *ASIACRYPT 2004*, pages 372–386, 2004. Lecture Notes in Computer Science No. 3329.
50. D. H. Phan and D. Pointcheval. OAEP 3-round: A generic and secure asymmetric encryption padding. In *ASIACRYPT 2004*, volume 3329 of *LNCS*, pages 63–77. Springer-Verlag, 2004.
51. R. L. Rivest, A. Shamir, and Y. Tauman. How to leak a secret. In *ASIACRYPT 2001*, pages 552–565. Springer-Verlag, 2001.
52. C. P. Schnorr. Efficient signature generation for smart cards. *J. of Cryptology*, 4(3):239–252, 2091.
53. A. Shamir. Identity-based cryptosystems and signature schemes. In *Proc. CRYPTO 84*, pages 47–53. Springer-Verlag, 1984. Lecture Notes in Computer Science No. 196.
54. I. Teranishi, J. Furukawa, and K. Sako. k-times anonymous authentication. In *Asiacrypt 2004*, volume 3329 of *LNCS*, pages 308–322. Springer-Verlag, 2004.
55. Victor K. Wei. Tracing-by-linking group signatures. Cryptology ePrint Archive, Report 2004/370, 2004. http://eprint.iacr.org/.
56. Victor K. Wei, Tsz Hon Yuen, and Fangguo Zhang. Group signature where group manager, members and open authority are identity-based. In Proc. ACISP 2005, To appear, 2005.
57. F. Zhang and K. Kim. ID-Based blind signature and ring signature from pairings. In *Proc. ASIACRYPT 2002*, pages 533–547. Springer-Verlag, 2002. Lecture Notes in Computer Science No. 2501.
58. F. Zhang, R. Safavi-Naini, and W. Susilo. An efficient signature scheme from bilinear pairings and its applications. In *Proc. PKC'2004*, pages 277–290. Springer-Verlag, 2004. Lecture Notes in Computer Science No. 2947.

A Previous state-of-the-art CCA2-fully-anonymous group signatures

For the purpose of comparing efficiencies against our new group signatures, we instantiate the previous state-of-the-art CCA2-fully-anonymous group signatures using the best known published techniques. As are typically the cases, instantiating in pairings results in the (previous) shortest [12, 41, 49] but it incurs expensive pairings computations to verify the signatures; and instantiating in the strong RSA framework results in group signaturew with the fastest signature verifications, but the signatures are not as short as those in pairings.

A.1 The previous shortest CCA2-fully-anonymous group signature: Protocol CCAGS-BBS. The literature [12, 41, 49] contains sufficient methods to efficiently instantiate a short CCA2-fully-anonymous group signature, even though no explicit instantiations are published. BOneh, Boyen, and Shacham [12] published the shortest CPA-fully-anonymous group signature to date. They used Linear Encryption, an IND-CPA encryption, to escrow the signer identity to the OA. Kiayias and Yung [41] described a method to convert a CPA-fully-anonymous group signature to a CCA2-fully-anonymous one. Their method is to convert the IND-CPA escrowing encryption to a twin encryption [47] which is IND-CCA2 secure.

In this Section, we apply [41]'s conversion to the Linear Encryption in [12] to obtain a CCA2-fully-anonymous group signature called Protocol CCAGS-BBS. We also optimize it to the best of our ability. The result represents the previous shortest CCA2-fully-anonymous group signature prior to our theoretical advance in the previous Section.

Using linear encryption let $g_1 = g_3^{1/a}$, $g_2 = g_3^{1/b}$, $g_6 = g_8^{1/a'}$, $g_7 = g_8^{1/b'}$, where $\mathsf{sk}_{OA} = (a, b, a', b')$. Using the parameters from Section 4.2, the CCAGS-BBS group signature is specified below. The commitments are

$$T_1 = g_1^{s_1}, \quad T_2 = g_2^{s_2}, \quad T_3 = A g_3^{s_1+s_2} \in \mathbb{G}_1,$$

$$\tilde{T}_3 = \hat{\mathbf{e}}(T_3^{-1}, u^\gamma)\hat{\mathbf{e}}(h_0, u) = \hat{\mathbf{e}}(T_3, u)^e \hat{\mathbf{e}}(h, u)^x \hat{\mathbf{e}}(g_3, u^\gamma)^{-(s_1+s_2)} \hat{\mathbf{e}}(g_3, u)^{-s_3}, \tag{22}$$

$$T_4 = 1 = T_1^e g_1^{-s_3}, \quad T_5 = 1 = T_2^e g_2^{-s_4}, \quad T_6 = g_6^{s_6}, \quad T_7 = g_7^{s_7}, \quad T_8 = A g_8^{s_6+s_7} \in \mathbb{G}_1,$$

$$\tilde{T}_8 = T_3^{-1} T_8 = g_3^{-s_1-s_2} g_8^{s_6+s_7}, \quad D_1 = g_1^{r_1}, \quad D_2 = g_2^{r_2},$$

$$D_3 = \hat{\mathbf{e}}(T_3, u)^{r_e} \hat{\mathbf{e}}(h, u)^{r_x} \hat{\mathbf{e}}(g_3, u^\gamma)^{-(r_1+r_2)} \hat{\mathbf{e}}(g_3, u)^{-r_3}, \quad D_4 = T_1^{r_e} g_1^{-r_3}, \quad D_5 = T_2^{r_e} g_2^{-r_4}, \tag{23}$$

$$D_6 = g_6^{r_6}, \quad D_7 = g_7^{r_7}, \quad D_8 = g_3^{-r_1-r_2} g_8^{r_6+r_7}$$

where $s_3 = es_1$, $s_4 = es_2$. The challenge is

$$c = \mathcal{H}(\mathsf{param}, R, M, T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8, D_1, D_2, D_3, D_4, D_5, D_6, D_7, D_8). \tag{24}$$

The responses are $z_i = r_i - cs_i$ for all indices i, where $s_e = e$, $s_x = x$. The signature is

$$\sigma = (\mathsf{param}, R, M, T_1, T_2, T_3, T_6, T_7, T_8, c, z_1, z_2, z_3, z_4, z_e, z_x, z_6, z_7). \tag{25}$$

Its bandwidth is $15\lambda_s$ bits, 36% higher than our CCAGS-SDH group signature which exploits our new theoretical advance.

Comparing against the CPA-fully-anonymous group signature in [12]: [12] cost $9\lambda_s$ bits, including $5\lambda_s$ for the linear encryption and 4 for the remainder. [12] did not have non-frameability. To add it costs extra λ_s. Converting from single encryption to twin encryption costs extra $5\lambda_s$ bits.

Note [49] instantiated twin ElGamal encryption in their CCA2-fully-anonymous group signature. Their instantiation's (the correct version in ePrint, GVf in their Asiacrypt'04 version cannot complete because of non-knowledge of a discrete logarithm base S_i) bandwidth costs at least $24\lambda_s$ bits includes $16\lambda_s$ specified in their paper plus $8\lambda_s$ for the *Proof* of the twin ElGamal encryption also specified in their paper. We counted that each $\mathbb{G}_3$ point costs at least twice the bandwidth of a $\mathbb{G}_1$ point. The security of Protocol CCAGS-BBS is analyzed in the Theorem below. Its proof consists of known proofs from [12, 41] and is therefore omitted.

Theorem 5. *The CCAGS-BBS group signature is, assuming the random oracle (RO) model,*

1. correct*;*
2. CCA2-fully-anonymous *provided the DLDF($\mathbb{G}_1$) Assumption holds.*
3. full traceable *provided the q_S-SDH($\mathbb{G}_1$) Assumption holds;*
4. non-frameable *provided the Discrete Logarithm Assumption holds in $\mathbb{G}_1$.*

In summary, the CCAGS-BBS group signature is a secure CCA2-fully-anonymous *group signature provided the DLDF($\mathbb{G}_1$) Assumption and the q_S-SDH Assumption both hold in the RO model.*

A.2 The previous fastest CCA2-fully-anonymous group signature: Protocol CCAGS-KY. Kiayias and Yung [41] described a generic method to convert CPA-fully-anonymous group signatures including [3, 21] to CCA2-fully-anonymous ones. The core technique [41] is the use of twin encryption [47]. We instantiate and optimize it below in the strong RSA framework. The result represents the previous fastest CCA2-fully-anonymous group signature prior to our advance in the previous Section.

We use the twin ElGamal encryption. The OA's public encryption key is pk_{OA}= $(g_{0,1}^{s_{E,1}}, g_{0,2}^{s_{E,2}})$. Its secrete decryption key is $\mathsf{sk}_{OA} = (s_{E,1}, s_{E,2})$. Initializations and system parameters are similar to those of Section 4.3. The group signature, which we call Protocol CCAGS-KY, is as follows. The commitments are

$$T_{0,1} = g_{0,1}^{s_{0,1}},\ \ T_{0,2} = g_{0,2}^{s_{0,2}},\ \ T_{A,1} = Ag_{A,1}^{s_{0,1}},\ \ T_{A,2} = Ag_{A,2}^{s_{0,2}},\ \ T_{2,1} = g_{2,1}^{e}g_{2,2}^{s_{0,1}},\ \ T_{2,2} = g_{2,3}^{e}g_{2,4}^{s_{0,2}},$$
$$T_3 = \mathbf{g}_S^{1/(R+x)},\ \ D_{0,1} = g_{0,1}^{r_{0,1}},\ \ D_{0,2} = g_{0,2}^{r_{0,2}},\ \ D_{A,1} = T_{0,1}^{r_e}h^{r_x}g_{A,1}^{-r_{1,1}},\ \ D_{A,2} = T_{0,2}^{r_e}h^{r_x}g_{A,2}^{-r_{1,2}},$$
$$D_{1,1} = T_{0,1}^{r_e}g_{0,1}^{-r_{1,1}},\ \ D_{1,2} = T_{0,2}^{r_e}g_{0,2}^{-r_{1,2}},\ \ D_{2,1} = g_{2,1}^{r_e}g_{2,2}^{r_{0,1}},\ \ D_{2,2} = g_{2,3}^{r_e}g_{2,4}^{r_{0,2}},\ \ D_3 = T_3^{r_x}.$$

Note $T_{A,1}T_{A,2}^{-1} = g_{A,1}^{s_{0,1}}g_{A,2}^{-s_{0,2}}$, $h_0 = T_{0,1}^{e}h^{x}g_{A,1}^{-s_{1,1}} = T_{0,2}^{e}h^{x}g_{A,2}^{-s_{1,2}}$, $s_{1,1} = es_{0,1}$, $s_{1,2} = es_{0,2}$, $1 = T_{0,1}^{e}g_{0,1}^{-s_{1,1}} = T_{0,2}^{e}g_{0,2}^{-s_{1,2}}$. The challenge is

$$c = \mathcal{H}(\mathsf{param}, R, M, T_{0,1}, T_{0,2}, T_{A,1}, T_{A,2}, T_{2,1}, T_{2,2}, T_3, D_{0,1}, D_{0,2}, D_{A,1}, D_{A,2}, D_{1,1}, D_{1,2}, D_{2,1}, D_{2,2}, D_3)(26)$$

The responses are $z_i = r_i - cs_i$ for all indices i, with $s_e = e$, $s_x = x$. The signature is

$$\sigma = (\mathsf{param}, R, M, T_{0,1}, T_{0,2}, T_{A,1}, T_{A,2}, T_{2,1}, T_{2,2}, T_3, c, z_{0,1}, z_{0,2}, z_{1,1}, z_{1,2}, z_3, z_x).$$

Protocol $\mathsf{GVf}(\sigma)$ computes

$$D_{0,1} = g_{0,1}^{z_{0,1}}T_{0,1}^{c},\ \ D_{0,2} = g_{0,2}^{z_{0,2}}T_{0,2}^{c},\ \ D_{A,1} = T_{0,1}^{z_e}h^{z_x}g_{A,1}^{-z_{1,1}}h_0^{c},\ \ D_{A,2} = T_{0,2}^{z_e}h^{z_x}g_{A,2}^{-z_{1,2}}h_0^{c},$$
$$D_{1,1} = T_{0,1}^{z_e}g_{0,1}^{-z_{1,1}},\ \ D_{1,2} = T_{0,2}^{z_e}g_{0,2}^{-z_{1,2}},\ \ D_{2,1} = g_{2,1}^{z_e}g_{2,2}^{z_{0,1}}T_{2,1}^{c},\ \ D_{2,2} = g_{2,3}^{z_e}g_{2,4}^{z_{0,2}}T_{2,2}^{c},\ \ D_3 = T_3^{z_x}(\mathbf{g}_S T_3^{-R})^{c}.$$

and verifies the received challenge c equals to that computed according to Equation (25).

Protocol CCAGS-KY's online complexity consists of GVf's 9 multi-base exponentiations. This complexity is 80% higher than our Protocol CCAGS-SRSA which exploits our new theoretical advance. Protocol CCAGS-KY's signature bandwidth is also higher than that of Protocol CCAGS-SRSA. The security reduction for Protocol CCAGS-KY is in the Theorem below. Its proof consists of known proofs from [41, 21, 3] and is therefore omitted.

Theorem 6. *The CCAGS-KY group signature is, assuming the RO model, (1)* correct*; (2)* CCA2-fully-anonymous *provided the DDH Assumption holds in QR_N; (3)* full traceable *provided the Strong RSA Assumption holds in QR_N; (4)* non-frameable *provided the Discrete Logarithm Assumption holds in QR_N. In summary the CCAGS-KY group signature is a* secure CCA2-fully-anonymous *group signature provided the DDH Assumption and the Strong RSA Assumption both hold in QR_N under the RO model.*

B Proof Sketch of Theorem 2

We use the *static attacker model*, where $\mathcal{A}$ corrupts q_U users at the beginning, and $\mathcal{S}$ knows which q_U users are corrupted at that time. Issues with active, concurrent, reset, or UC (Universal Composability) attackers are left to future research.

1. CCA2-full-anonymity The crucial issue is the simulation of the Decisional Open Oracle $\mathcal{DOO}(\sigma, U)$. Some IND-CCA2 encryptions utilize the WI (Witness Indistinguishability) to simulate their Decryption Oracles. We do not use WI. Instead we exploit the outstanding properties of the group signature. The attacker $\mathcal{A}$ must effectively query $\mathcal{DOO}$ with a valid signature σ. But attacker cannot generate a valid signature that opens to a legitimate user provided Non-Frameability holds. Therefore $\mathcal{DOO}$ has a way to detect and reject all queries from non-corrupted users.

If $\mathcal{A}$ has a non-negligible advantage in Experiment Anon, then $\mathcal{A}$ can disntinguish at least one of the following:

1. $\mathcal{A}$ can distinguish via the linear encryption of the certificate value A in the three commitments T_1, T_2, T_3. Then we will prove $\mathcal{A}$ can solve the DLDH($\mathbb{G}_1$) Problem.
2. $\mathcal{A}$ can distinguish via $T_4 = \mathbf{g}_S^{1/(R+x)}$ and known R. Then we will prove $\mathcal{A}$ can solve the q_S-SDDH($\mathbb{G}_1$, G_S) Problem.
3. $\mathcal{A}$ can distinguish via $T_5 = 1$ and $\mathbf{g}_S = T_4^{R+x}$ with known R. Then we will prove that $\mathcal{A}$ can solve the DDH($\mathbb{G}_1$, G_S) Problem.

For the first case above, that distinguishing via the linear encryption results in solving the Decision Linear Problem was proved in [12]. For the third case above, the q_S-SDDH(G_a, G_b) Assumption implies the DDH(G_a, G_b) Assumption by Lemma 1. Below, we prove the second case above.

Initialization. Simulator $\mathcal{S}$ is given a q_S-SDDH($\mathbb{G}_1$,G_S) Problem instance for each uncorrupted user. The problem instance consists of h, $h^{x'} \in G_1$, $\mathbf{g}^{x^i} \in G_S$, $0 \leq i \leq q_S$, $x' = x \in Z$, $0 < x' < x <$ order($\mathbb{G}_1$) $<$ order(G_S), R, and a test value in G_S which equals $\mathbf{g}^{1/(R+x)}$ or random. $\mathcal{S}$ generates group key pair, and generates the user key pair for each corrupted user to answer Adversaries $\mathcal{GCO}$ and uco queries at the beginning of Experiment Anon.

For each uncorrupted user U, $\mathcal{S}$ randomly generates R_i, $1 \leq i \leq q_S$, let $f_S(x) = \prod_{i=1}^{q_S}(R_i + x)$ and compute $S_i = g^{f_S(x)(R_i+x)^{-1}}$. $\mathcal{S}$ gives h^x as the user public key to U. $\mathcal{S}$ generates the OA's key pair. Assume the DLDH($\mathbb{G}_1$) Assumption so $\mathcal{A}$ cannot distinguish by exploiting the linear encryption, provided we can simulate $\mathcal{DOO}$ which we do below. $\mathcal{S}$ (and $\mathcal{A}$) knows the group secrete key to issue certificates.

This setup is similar to the single-inconsistent player (SIP) proof technique in [25, 1], for the static attacker model. A winner $\mathcal{A}$ of Experiment Anon will have a non-negligible probability of solving one of the q_S-SDDH Problem instances. The overall result will be that $\mathcal{A}$ can be simulated to solve a random instance of the q_S-SDDH Problem with non-negligible probability. Issues with multiple inconsistent players (MIP) is left to future research.

Simulating the Oracles. For corrupted users, $\mathcal{S}$ answers $\mathcal{SO}$ by using the user secrete key. For uncorrupted user U, $\mathcal{S}$ answers the Signing Oracle queries $\mathcal{SO}(U, M_i)$ with S_i, $1 \leq i \leq q_S$. For simplicity, we assume there is an equal number, q_S, of $\mathcal{SO}$ queries for each uncorrupted user. The other case can be proved similarly. The Decisional Open Oracle $\mathcal{DOO}(\sigma, U)$ is simulated as follows:

1. If U is a corrupted user, then $\mathcal{S}$ knows its secrete key sk_U and $\mathcal{S}$ uses it to test if $T_4 \stackrel{?}{=} \mathbf{g}_S^{1/(R+x)}$. $\mathcal{DOO}$ outputs 1 (0) if test is positive (negative).
2. If U is an uncorrupted user generated by $\mathcal{S}$ in Initialization, test if σ is an $\mathcal{SO}$ output for U. If so, output 1. Else, output 0.
3. If U is an user not generated by $\mathcal{S}$ in Initialization, output 0.

Intuition: Assuming non-frameability, then $\mathcal{A}$ cannot have generated a valid signature that opens to an uncorrupted user.

The Extraction. In the Gauntlet Phase of Experiment Anon, $\mathcal{A}$ presents two gauntlet users U_0 and U_1. Both are uncorrupted. $\mathcal{S}$ chooses $b \in \{0, 1\}$, and generates the gauntlet signature σ_g as follows:

1. Set T_4 to the test value prescribed in the q_S-SDDH Problem instance for user U_b.
2. Compute σ_g via the standard HVZK simulation as follows:
   (a) Randomly generate the challenge c, all responses z_i's, all secrets s_i's except $s_x = x$. (Note a uncorrupted user's public key h^x is prescribed from its q_S-SDDH Problem instance.)
   (b) Compute the T_i's by Equation (5) and compute the D_i's by Equation (10).
   (c) Use the RO to backpatch the hash output to c in Equation (7).

When $\mathcal{A}$ eventually returns $\hat{b}$ in Experiment Anon, $\mathcal{S}$ answers yes to the q_S-SDDH Problem instance with user U_b, if $\hat{b} = b$. $\mathcal{S}$ answers random to q_S-SDDH otherwise. If $\mathcal{A}$ has a non-negligible advantage in Experiment Anon, so does $\mathcal{S}$ in solving the q_S-SDDH Problem. □

2. Full traceability Full traceability is essentially coalition-resistance unforgeability. The attacker must forge all components. To forge T_4 is to solve a q_S-SDH Problem. The proof above for anonymity can be easily modified to prove this component here. Below, we reduce full traceability to the q_U-SDH Problem in the case where the q_S-SDH Assumption is not used.

Initialization and oracle simulations are similar to the above. Extraction is described below. Simulator $\mathcal{S}$ is given a q_U-SDH Problem instance u, u^γ, $g^{\gamma^i} \in G_1$, $0 \leq i \leq q_U$, $\hat{e}$. Then $\mathcal{S}$ randomly generates e_i, x_i, $1 \leq i \leq q_U$, let $f(\gamma) = \prod_{i=1}^{q_U}(e_i + \gamma)$ and compute $A_i = g^{f(\gamma)(e_i+\gamma)^{-1}(1-\alpha x_i)^{-1}}$ where random oracle (RO) is backpatched to $h = h_0^\alpha$. Gives (A_i, e_i, x_i), $1 \leq i \leq q_U$, to each corrupted user.

Assume $\mathcal{A}$ has a non-negligible advantage in Experiment FT. Fork-simulate $\mathcal{A}$ to extract witnesses $\bar{s}_1$, $bars_2$, $\hat{e}$, $\bar{x}$ satisfying

$$T_1 = g_1^{\bar{s}_1}, \quad T_2 = g_2^{\bar{s}_2}, \quad h_0 = (T_3 g_3^{-\bar{s}_1 - \bar{s}_2})^{\gamma+\hat{e}} h^{\bar{x}} g_3^{(\bar{s}_1+\bar{s}_2)\bar{e}-\bar{s}_3} \tag{27}$$

Backpatching the random oracle to $g_3 = \mathcal{H}('g', 3) = h_0^{\alpha_{g,3}}$, and $h = \mathcal{H}('h') = h_0^{\alpha_h}$, where $\alpha_{g,3}$ and α_h are chosen by $\mathcal{S}$ in Initialization. Denote $\delta = 1 - \alpha_{g,3}(\bar{s}_1 e + \bar{s}_2 e - \bar{s}_3)$ and $\hat{A} = (T_3 g_3^{-\bar{s}_1-\bar{s}_2})^{1/\delta}$, $\hat{x} = \bar{x}/\delta$, then $\hat{A}^{\hat{e}+\gamma} h^{\hat{x}} = h_0 = g^{f(\gamma)}$ Then

$$h_0^{1/(\hat{e}+\gamma)} = g^{f(\gamma/(\hat{e}+\gamma)} = g^{\sum_i=0^{q_U} \bar{f}_i \gamma_i} g^{\bar{f}_{-1}/(\hat{e}+\gamma)} \tag{28}$$

and we obtain $(g^{1/(\hat{\mathbf{e}}+\gamma)}, \hat{e})$ to solve the q_U-SDH Problem. □

3. Non-frameability The proof of non-frameability is similar to that of full traceability. Except now the Adversary $\mathcal{A}$ can corrupt the group trapdoor and the forgery must open to one of the users initialized in the beginning. The traceability attacker can generate an arbitrary user sk-pk pair with the obstacle being computing a certificate without the group trapdoor. The non-frameability attacker has the group trapdoor, but must forge a signature which opens to an legitimate user initialized with key pair $(\mathsf{sk}_U = x, \mathsf{pk}_U = h^x)$ where $\mathcal{A}$ is given h^x but not x.

If $\mathcal{A}$ has an advantage in Experiment NF, $\mathcal{A}$ can be rewound to extract the witness x, and solve a DLP. Oracle simulations are as before. The initialization and setup for extraction is straightforward. □

C Proof Sketch of Theorem 4.

Large parts of this proof is similar to that of Theorem 2. We focus only on the different parts.

1. CCA2-full-anonymity This is mostly similar to its counterpart in Theorem 2 and omitted. □

2. Full traceability *Simulating User Corruption Oracle* $\mathcal{UCO}$: Given a Strong RSA Problem instance (N, Z), Simulator $\mathcal{S}$ initializes N to be the group public key. Then $\mathcal{S}$ initializes the sk-pk pair and certificates of the q_U corrupted users as follow: Randomly generate x_i and primes within regulation range e_i, $1 \leq i \leq q_U$, and randomly generate r. Compute $h = Z^{e_1 \cdots e_{q_U}}$, $h_0 = h^r$. Then compute $A_i = h^{e_i^{-1}(-x_i+r)}$ and answers $\mathcal{A}$'s queries to $\mathcal{UCO}$ with (A_i, e_i, x_i) for each i, $1 \leq i \leq q_U$.

Simulating Signing Oracle $\mathcal{SO}(M, U)$: If U is corrupted, $\mathcal{S}$ knows its secretes and computes a signature. If U is uncorrupted, $\mathcal{S}$ uses the prescribed q_S-SDH Problem instances, one instance for each uncorrupted, to simulate.

The Extraction: Assume $\mathcal{A}$ has a non-negligible advantage in Experiment FT. Then fork simulate $\mathcal{A}$ to extract secretes $\bar{A}$, $\bar{e}$, $\bar{x}$ satisfying $\bar{A}^{\bar{e}} h^{\bar{x}} = h_0$. Noting $h^r = h_0$, we have $\bar{A}^{\bar{e}} = h^\delta = Z^{\delta e_1 \cdots e_{q_U}}$. Dividing both exponents by their GCD, we obtain $\bar{A}^a = Z^b$, with GCD(a,b)=1. Then $\alpha a + \beta b = 1 \in Z$ for some α and β. Then $Z = (A^\beta Z^\alpha)^a$ solves the strong RSA Problem. □

3. Non-frameability This part is also similar to Theorem 2 and omitted. □