Custodian-Hiding Verifiable Encryption

Joseph K. Liu[1], Victor K. Wei[1], and Duncan S. Wong[2]*

[1] Department of Information Engineering
The Chinese University of Hong Kong
Shatin, Hong Kong
{ksliu9,kwwei}@ie.cuhk.edu.hk

[2] Department of Computer Science
City University of Hong Kong
Kowloon, Hong Kong
duncan@cityu.edu.hk

Abstract. In a verifiable encryption, an asymmetrically encrypted ciphertext can be publicly verified to be decipherable by a designated receiver while maintaining the semantic security of the message [2, 6, 9]. In this paper, we introduce *Custodian-Hiding Verifiable Encryption*, where it can be publicly verified that there exists at least one custodian (user), out of a designated group of n custodians (users), who can decrypt the message, while the semantic security of the message and the anonymity of the actual decryptor are maintained. Our scheme is proven secure in the random oracle model. We also introduce two extensions to decryption by a subset of more than one user.

Keywords: Verifiable Encryption, Publicly Verifiable, Anonymity

23th August, 2004

1 Introduction

We introduce the new paradigm *custodian-hiding verifiable encryption, CH-VE.* It allows a sender to verifiably encrypt a message to a group of receivers in a way that only one of them is able to decrypt it. In addition, any public verifier can ascertain this fact while he knows nothing about the plaintext and cannot compute the identity of the actual decryptor. Before formal definitions, we give a few motivating applications of CH-VE.

First consider the following scenario. Alice wants to send a public-key encrypted message to Bob, who works for ABC Company. For some security reason the company gateway system does not allow the message in unless it is for a company employee. However, Bob does not wish to divulge his private key. Without knowing Bob's private key, how can the gateway ensure the message is intended for a company employee? Furthermore, Alice and Bob do not want

* The work described in this paper was fully supported by a grant from CityU (Project No. 9360087).

the company gateway to know that Bob is the actual recipient. By knowing only the public information of the company employees, the gateway system has to determine if the encrypted incoming data is for a company employee without being able to identify the actual recipient. In addition, other employees of the company should not be involved in the secret communication between Alice and Bob and should be totally unaware of the entire process.

Another scenario is about key escrow. In a key escrow, Alice encrypts a secret key under the public key of a custodian and sends to an organization or a government this ciphertext together with a proof that the ciphertext is indeed an encryption of her own secret key. In order to increase the level of trust, n custodians maybe used instead of one single custodian. The key is shared among a particular set of t custodians so that only by these t custodians cooperating together can decrypt the secret of Alice while other subset of custodians cannot do so. It is more secure as the organization or the government does not know which particular t custodians know the secret. It takes an exponential time to find out which those t custodians are if t is equal to half of n.

In this paper, we present solutions to the above problems as well as some other application problems. Below, we briefly review verifiable encryption before introducing custodian-hiding verifiable encryption.

Verifiable Encryption. A verifiable encryption [20, 2, 6, 4, 9] allows a prover to encrypt a message and sends to a receiver such that the ciphertext is publicly verifiable. That is, any verifier can ensure the ciphertext can be decrypted by the receiver yet knowing nothing about the plaintext. There are numerous applications of verifiable encryption. For example, in a publicly verifiable secret sharing scheme [20], a dealer shares a secret with several parties such that another third party can verify that the sharing was done correctly. This can be done by verifiably encrypting each shares under the public key of the corresponding party and proves to the third party that the ciphertext encrypt the correct shares. Another scenario is in a fair exchange environment [2], in which both parties want to exchange some information such that either each party obtain the other's data, or neither party does. One approach is to let both parties verifiably encrypt their data to each other under the public key of a trusted party and then to reveal their data. If one party refuses to do so, the other can go to the trusted party to obtain the required data. Verifiable encryption can be also applied in revokable anonymous credential [7]. When the administration organization issues a credential, it verifiably encrypts enough information under the public key of the anonymity revocation manager, so that later if the identity of the credential owner needs to be revealed, this information can be decrypted.

Custodian-Hiding Verifiable Encryption (CH-VE). In a Custodian-Hiding Verifiable Encryption (CH-VE), a *Prover* is to send a public-key encrypted message to one among n *Custodians* through a *Verifier*. The Prover and the Verifier agree upon a group of n public keys, and then conduct an interactive protocol such that, if the Verifier is satisfied and relays a ciphertext from Prover to the

n decryptors, at least one of the decryptor can recover the message. Furthermore, the message is semantically secure to the Verifier, and the identity of the actual decryptor is anonymous to the Verifier. CH-VE can solve the motivating applications earlier.

Receiver-Oblivious Transfer. CH-VE can be also regarded as a form of "*Receiver-Oblivious Transfer*". It is a dual to the following equivalent formulation of (interactive) *Oblivious Transfer (OT)* [18, 15, 5, 11–13, 1, 17]:

1. Verifier sends auxiliary encryption parameters.
2. Sender encrypts n messages to n public keys.
3. Verifier processes and then relays n ciphertexts to n Decryptors, at most one Decryptor can recover its message.

CH-VE does the following:

1. Sender sends n ciphertexts.
2. Verifier challenges. If satisfied with responses, relays ciphertexts to n Decryptors, at least one Decryptor can recover its message.

In OT, Sender is oblivious of the identity of the capable Decryptor. In CH-VE, Verifier is oblivious of the identity of the capable Decryptor. From this perspective, we consider CH-VE as a dual paradigm to the important fundamental paradigm of OT.

We also introduce two extensions. In the first extension a *targeted* subset of t custodians (out of n) jointly recover the message. In the second extension, any member of a *targeted* subset of t custodians can recover the encrypted message. Both extensions preserve the anonymity of the targeted subset.

These extensions can be useful in the following scenario. Bob belongs to a cluster of t members in a group of n members. For example, the cluster can be a small unit in a temporarily formed task force for a special mission. Our extension schemes can be used to transmit confidential messages to the unit, or to unit members, while keeping non-unit members of the task force and the security gateway of the task force at bay.

Contributions: We introduce a new paradigm: Custodian-Hiding Verifiable Encryption (CH-VE), which is an extension of (ordinary) Verifiable Encryption (VE). It retains the basic properties of VE: Message is encrypted to a designated decryptor/custodian in such a way that a public third party can verify that fact while knowing nothing about the plaintext. CH-VE adds the following basic anonymizing property: The decryptor/custodian is anonymized, in such a way that it is indistinguishable/hidden among a designated group of n decryptors/custodians. The public verifier can ensure one of the n decryptors/custodians can actually decrypt the message, but the verifier cannot identify the actual decryptor.

We present formal security models and definitions of security notions of CH-VE. The models are very strong models, formulating the anonymity in terms of IND-CCA2 games, with random oracle, decryptor oracle, and colluder oracle.

We present two constructions of CH-VE for DL homomorphic image. The cut-and-choose methodology is used [2, 6]. The schemes are proven secure in the random oracle model. The security is reduced to that of the hashing and the underlying encryption. The second of our constructions is secure against some Decryptors colluding with Verifier.

Our schemes support perfectly separability [16, 8], where Decryptors can use different encryption functions.

We also introduce the new paradigm of Custodian-Hiding Group Verifiable Encryption (CH-GVE). The first one requires a *particular* subset of t custodians out of n custodians to work jointly in decrypting the message while other subsets cannot. The second extension allows any member of a *particular* subset of t custodians to decrypt the message while members outside this subset cannot. Both extensions preserve the anonymity of the targeted decipherers.

The CH-VE is different from Group VE of [6]. The latter can verify encryption to a group of n decryptors where any subset of decryptors satisfying an access structure can recover the message. However, the Verifier always knows the entire access structure, and therefore knows which subset of decryptors can recover. There is no anonymizing of the designated decryptors. In CH-VE, the identities of the designated decryptors who can jointly recover is anonymized, and uncomputable by the Verifier.

Organization: This paper is organized as follows. We describe some related work in Sec. 2. This is followed by our security model specified in Sec. 3. Our basic schemes are described in Sec. 4. The paper is concluded in Sec. 5.

2 Related Work

Verifiable Encryption (VE) was first introduced by Stadler [20] in 1996 for the use of publicly verifiable secret sharing scheme [10]. The VE is based on ElGamal's public key system [14]. It allows a public verifier to determine if a ciphertext contains the discrete logarithm of a given value without decrypting it. The scheme uses the cut-and-choose methodology. Later, Asokan, et al. [2] presented a very general cut-and-choose based VE for encryption of pre-image of any homomorphic one-way function. Their scheme also provides *perfect separability* in such a way that the scheme can take any type of encryption algorithm and encryption key associated to the receiver. Camenisch, et al. [6] proposed another VE which is also perfectly separable and is proven secure without relying on random oracle. It is not limited to homomorphic one-way function but generalized to any boolean relation. Bao [4] proposed a deterministic VE for discrete logarithm without using the cut-and-choose methodology. Camenisch, et al. [9] propose another VE for discrete logarithm without using cut-and-choose methodology and achieved provable security under chosen-ciphertext-attack security model. Ateniese [3] also propose an efficient VE for digital signatures.

In all the above schemes the verifier knows the identity of the receiver. An anonymous verifiable encryption scheme proposed by Camenisch, et al. [7] hides the identity of the receiver from the verifier. Their scheme requires the prover to know the private key of the receiver.

Comparisons with Group Verifiable Encryption (GVE) [**6**]**:** Camenishch and Damgård proposed two schemes in [6]. The first one is a VE scheme which is targeted for one decryptor only. The second one is a Group VE (GVE) scheme which is targeted for any t decryptors. They allow the prover to choose an access structure Γ for a group of n receivers such that *any* subset of t members can jointly recover the message. The access structure can be instantiated by a secret sharing scheme. That is, the prover divides the message m into n pieces of shares $m_1, \ldots, m_n$ such that any t of them are enough to reconstruct m. Then he encrypts m_i using the encryption function of user i, for $i = 1, \ldots, n$, and sends all ciphertext to the verifier. It is clear that the message m can be reconstructed if any t users decrypt their corresponding ciphertext to get the shares.

In summary, their GVE scheme is a threshold version of their VE scheme by using a secret-sharing scheme to share the escrowed information such that any qualified subset (under the access structure) of custodians/decryptors can jointly recover the information. But the Verifier knows the identities of each subset of custodians who can recover the information.

Our proposed CH-VE basic scheme hides the identity of the one *targeted* custodian among n possible custodians. Others $n-1$ custodians cannot decrypt the message. The identity of the "trusted" or "designated" custodian is anonymized to n possible custodians. Our extensions, the CH-GVE schemes deploy some threshold-sharing technique of the escrowed secret (plaintext message) to t possible custodians and hiding the identity of the t "trusted" or "designated" custodians among n possible custodians (in certain ways).

In general, all our proposed scheme allows the escrowed information to be shared out among a specific subgroup of t custodians. The Verifier is assured that there exists t among the total population of n custodians that can recover the escrowed information, but the Verifier cannot compute the identities of these t custodians.

3 Security Model

In this section we define the security model to be used.

An CH-VE scheme is a tuple $(\mathcal{G}, P, V, R)$. Components specified below.

A PPT (polynomial probabilistic time) algorithm $\mathcal{G}(1^{\lambda_s})$, on input security parameter λ_s, generates n pairs of encryption functions E_i (with public key PK_i) and decryption functions D_i (with private key SK_i), for $i = 1, \ldots, n$, which are secure against chosen-ciphertext attack [19].

There is a two-party PPT protocol between P (Prover) and V (Verifier), a PPT algorithm R (Recovery Algorithm), also known as the Decryption Algorithm, and a homomorphic one-way function f. P accepts as inputs the security parameter λ_s, some appropriate binary string m (a *message*), n public-key encryption functions $\{E_i\}_{1 \le i \le n}$ which are secure against chosen-ciphertext attack [19], and an integer $\pi \in \{1, \cdots, n\}$ (the index of the actual decryptor). V accepts as inputs λ_s, f and $\{E_i\}_{1 \le i \le n}$ only. If the protocol completes without early termination, V outputs two finite binary strings $d = f(m)$ (homomorphic image) and $\mathcal{C}$ (ciphertext). We call it successful. Otherwise, V outputs Reject.

The Recovery Algorithm $R : (d, \mathcal{C}, D_i) \mapsto \{m', NULL\}$ accepts as inputs the homomorphic image, the ciphertext and one of the n decryption functions corresponding to $\{E_i\}_{1 \le i \le n}$ and outputs either a finite string m' or the $NULL$ string.

We formulate the security definition more precisely as below:

Definition 1. *The CH-VE scheme is* complete *if V always outputs homomorphic image d and ciphertext $\mathcal{C}$ such that $R(\mathcal{C}, D_\pi) = m$ and $d = f(m)$ for some $1 \le \pi \le n$, for arbitrary input m, homomorphic image d and ciphertext $\mathcal{C}$ satisfying $d = f(m)$, whenever both P and V are honest.*

Definition 2. *The CH-VE scheme is* sound *if, whenever V completes a protocol run without early termination and outputs homomorphic image d and ciphertext $\mathcal{C}$, then with overwhelming probability $f(R(\mathcal{C}, D_\pi)) = d$ for some π, $1 \le \pi \le n$.*

Next we define some oracles we are going to use in this model:

Prover Oracle $\mathcal{PO}$: Upon query, it interacts with the querier in the role of the Prover P.

Decryption Oracle $\mathcal{DO}$: Upon input a ciphertext, decrypt it or output Reject on invalid ciphertext.

Colluder Oracle $\mathcal{CO}$: Upon input a public key generated by $\mathcal{G}$, output its corresponding secret key.

We define a game, **Game D**, between Adversary $\mathcal{A}$ and Simulator $\mathcal{S}$:

1. *Setup Phase:* Algorithm $\mathcal{G}$ is invoked to generate n key pairs. The public keys $\mathsf{PK}_1, \cdots, \mathsf{PK}_n$, are published.
2. *Probe-1 Phase:* $\mathcal{A}$ queries $\mathcal{DO}$, $\mathcal{PO}$, and $\mathcal{CO}$.
3. *Gauntlet Phase:* $\mathcal{A}$ generates message m_1, decryptor identity π_1, $1 \le \pi_1 \le n$, and give them to $\mathcal{S}$. $\mathcal{S}$ generates message m_0, decryptor identity π_0, and $b_G \in_R \{0,1\}$; computes and sends to $\mathcal{A}$ the CH-VE encryption, consisting of homomorphic image d and ciphertext $\mathcal{C}$, of message m_{b_G} targeted for decryptor π_{b_G}.
4. *Probe-2 Phase:* $\mathcal{A}$ interacts, as Verifier, with $\mathcal{S}$ as an honest Prover while querying $\mathcal{DO}$, $\mathcal{PO}$, and $\mathcal{CO}$. Except $\mathcal{A}$ cannot query $\mathcal{DO}$ with the gauntlet ciphertext $\mathcal{C}$.
5. *Output Phase:* $\mathcal{A}$ outputs its estimate $\hat{b}_G$ of b_G

We say $\mathcal{A}$ wins the Game if $\hat{b}_G = b_G$. There are two sub-games: **Game D-C:** $\pi_0 = \pi_1$ and $\mathcal{S}$ chooses m_0 randomly. The *advantage* of $\mathcal{A}$ is its probability of winning the game, minus 1/2. **Game D-A:** $m_0 = m_1$, $\mathcal{S}$ choses π_0 randomly from $\{1, \cdots, n\} \setminus \{\pi_1\}$. The *advantage* of $\mathcal{A}$ is its probability of winning the game, minus the probability of winning the game by random guessing. The latter probability equals $\frac{1}{2}(1 + \frac{q_C}{n-1})$ where q_C is the number of Colluder Oracle queries. We require $q_C \le n - 1$, and $\mathcal{A}$ cannot query $\mathcal{CO}$ with PK_{π_1}.

Definition 3. (Zero Knowledge) *The CH-VE scheme is* zero-knowledge *if no PPT adversary $\mathcal{A}$ can win Game D-C or Game D-A with non-negligibly advantage. It is* no-colluder zero-knowledge *if no PPT adversary $\mathcal{A}$ can win Game D-C or Game D-A with non-negligibly advantage without making any query to the Colluder Oracle.*

Definition 4. *A CH-VE scheme is* secure *if it is complete, sound, and zero-knowledge. It is* no-colluder secure *if it is complete, sound, and no-colluder zero-knowledge.*

Remark: In Game D-C and when $b_G = 0$, the honest prover $\mathcal{S}$ does not use m_1 at all. This fact can be used to prove that our zero knowledge implies the zero knowledge in Asokan, et al.[2], p.599, l.13.

4 Secure Custodian Hiding Verifiable Encryption (CH-VE) Schemes

We specify two schemes in this section. The first scheme is a no-colluder-secure CH-VE scheme while the second one is secure even with the existence of colluders. Both schemes use the 3-choice cut-and-choose methodology. In each of N cut-and-choose rounds, a three-move protocol (commit, challenge, respond) is conducted between Prover P and Verifier V. V flips a three-way coin to issue one of three possible challenges. Depending on the challenge, P provides suitable response. If all cut-and-choose rounds are satisfactory, V outputs a image d and a ciphertext $\mathcal{C}$. Otherwise, it aborts. Each receiver i attempts to decipher using its own asymmetric decryption function D_i, $1 \leq i \leq n$. At least one receiver will succeed.

4.1 A no-colluder-secure CH-VE scheme

Let (E_i, D_i), $1 \leq i \leq n$, be secure public-key encryption and decryption functions generated by $\mathcal{G}$. Let π index the targeted receiver. Let p, q be large primes, $q \mid p-1$, and $g \in F_p$, order(g)=q. Let the security parameter λ_s be as large as $|q|$. Let f be defined by $x \to g^x$ which is an instantiation of the one-way group homomorphism from $\mathbb{Z}_q$ to $<g>$. Let $m \in \mathbb{Z}_q$ be a message. Let N be the number of cut-and-choose rounds. Let $H_1 : \{0,1\}^* \to \{0,1\}^{\lambda_s}$ and $H_2 : \{0,1\}^* \to \mathbb{Z}_q$ be some statistically independent and cryptographically strong hash functions. Sometimes, we may pass in an element in $\mathbb{Z}_q$ for encryption and we implicitly assume that certain appropriate encoding method is applied.

When the Prover P computes any probabilistic public-key encryption function, P needs to send the corresponding coin flip sequence to the Verifier V and the sequence is to be carried on wherever the original message goes. We do not explicitly specify such in the following.

Sym(n) denotes the symmetric group of order n. It consists of all permutations on n objects. We instantiate a one-way homomorphic mapping of m, typically used in VE (verifiable encryption) literature [2,6,9] by discrete exponentiation g^m, and assume DLP (discrete logarithm problem) to be secure. We do so throughout the paper.

[**Protocol Between P and V (Encryption)** (Illustrated in Figure 1)]

1. P computes $d = g^m \bmod p$ and sends d to V.
2. Repeat the following steps N times in parallel.
   a. (*Commitment*) P randomly picks $s \in_R \mathbb{Z}_q$, $r_i \in_R \{0,1\}^{\lambda_s}$ for $1 \le i \le n$, and $\phi \in_R \mathrm{Sym}(n)$. P computes

$$\begin{aligned} \lambda &= E_1(r_1)||\cdots||E_n(r_n) \\ \gamma &= (g^{H_2(r_{\phi(1)})} \bmod p,\ \cdots,\ g^{H_2(r_{\phi(n)})} \bmod p) \\ \alpha' &= E_1(s)||\cdots||E_n(s) \\ \alpha &= H_1(\alpha') \\ \beta &= g^{H_2(r_\pi)s} \bmod p \\ \theta &= H_1(\lambda||\gamma||\alpha||\beta) \end{aligned}$$

   P sends θ to V.

   b. (*Challenge*) V picks $b \in_R \{1,2,3\}$ and sends b to P.

   c. (*Response*)
   - Case $b=1$, P sends $r_1,\cdots,r_n$, γ, α and β to V
   - Case $b=2$, P sends λ, γ, and s to V.
   - Case $b=3$, P sends λ, γ, α', and $s' = H_2(r_\pi)s + m \bmod q$ to V.

   d. (*Verification* by V)
   - Case $b=1$:
     - Verify that $r_1, \cdots, r_n$ are distinct.
     - Verify that there exists a unique permutation $\delta \in \mathrm{Sym}(n)$ such that $\gamma = (g^{H_2(r_{\delta(1)})} \bmod p,\ \cdots,\ g^{H_2(r_{\delta(n)})} \bmod p)$
     - Verify that $\theta = H_1(\hat{\lambda}||\gamma||\alpha||\beta)$ where $\hat{\lambda} = E_1(r_1)||\cdots||E_n(r_n)$.

     Continue only if all verifications succeed.
   - Case $b=2$:
     - Denote $\gamma = (\gamma_1,\cdots,\gamma_n)$.
     - Compute $\tilde{\alpha} = H_1(E_1(s)\ ||\ \cdots\ ||\ E_n(s))$ and $\beta_i = \gamma_i^s \bmod p$, for $i = 1,\cdots,n$.
     - Verify that $\theta = H_1(\lambda||\gamma||\tilde{\alpha}||\beta_i)$ for exactly one index $i \in \{1,\cdots,n\}$.

     Continue only if the verification succeeds.
   - Case $b=3$:
     - Compute $\beta' = g^{s'}/d \bmod p$
     - Verify that $\theta = H_1(\lambda||\gamma||H_1(\alpha')||\beta')$

     Continue only if the verification succeeds.
3. (*Output*) V terminates if any verification fails in any of the N cut-and-choose Rounds. Otherwise, it outputs d and the four-tuple sequences $(\alpha', \lambda, \beta', s')$ for all Case-(b=3) Rounds to all n receivers as the ciphertext, $\mathcal{C}$.

Prover P **Verifier V**

$d = g^m \bmod p$

$$\xrightarrow{\quad d \quad}$$

Repeat N times:

$s \in_R \mathbb{Z}_q$

$r_i \in_R \{0,1\}^{\lambda_s}$, $1 \le i \le n$

$\phi \in_R \mathrm{Sym}(n)$

$\lambda = E_1(r_1)||\cdots||E_n(r_n)$

$\gamma = (g^{H_2(r_{\phi(1)})} \bmod p,\ \cdots,\ g^{H_2(r_{\phi(n)})} \bmod p)$

$\alpha' = E_1(s)||\cdots||E_n(s)$

$\alpha = H_1(\alpha')$

$\beta = g^{H_2(r_\pi)s} \bmod p$

$\theta = H_1(\lambda||\gamma||\alpha||\beta)$

$$\xrightarrow{\quad \theta \quad}$$

$b \in_R \{1,2,3\}$

$$\xleftarrow{\quad b \quad}$$

Case $b = 1$

$$\xrightarrow{\quad r_1, \cdots, r_n, \gamma, \alpha, \beta \quad}$$

$r_i \overset{?}{\neq} r_j$, $\forall i,j \in \{1,\cdots,n\}, i \neq j$

$\overset{?}{\exists}\ \delta \in \mathrm{Sym}(n)$, $\gamma = (g^{H_2(r_{\delta(1)})}, \cdots, g^{H_2(r_{\delta(n)})})$

$\hat{\lambda} = E_1(r_1)||\cdots||E_n(r_n)$

$\theta \overset{?}{=} (\hat{\lambda}||\gamma||\alpha||\beta)$

Case $b = 2$

$$\xrightarrow{\quad \lambda, \gamma, s \quad}$$

$(\gamma_1, \cdots, \gamma_n) \leftarrow \gamma$

$\tilde{\alpha} = H_1(E_1(s)||\cdots||E_n(s))$

$\beta_i = \gamma_i^s \bmod p$, $1 \le i \le n$

$\overset{?}{\exists}\ i \in \{1,\cdots,n\}$, $\theta = H_1(\lambda||\gamma||\tilde{\alpha}||\beta_i)$

Case $b = 3$

$s' = H_2(r_\pi)s + m \mod q$

$$\xrightarrow{\quad \lambda, \gamma, \alpha', s' \quad}$$

$\beta' = g^{s'}/d \bmod p$

$\theta \overset{?}{=} H_1(\lambda||\gamma||H_1(\alpha')||\beta')$

If no rejection in all N rounds,
output d and $(\alpha', \lambda, \beta', s')$ for all Case-($b = 3$) rounds

Fig. 1. A custodian-hiding verifiable encryption scheme.

Recovery Algorithm R.

Denote $\bar{\lambda}_1||\cdots||\bar{\lambda}_n = \lambda$ and $\bar{\alpha'}_1||\cdots||\bar{\alpha'}_n = \alpha'$. For d and each four-tuple sequence $(\alpha', \lambda, \beta', s')$, each receiver i, $1 \le i \le n$, independently performs the following steps.

1. Compute $r_i = E_i^{-1}(\bar{\lambda}_i)$ and $s = E_i^{-1}(\bar{\alpha'}_i)$.
2. Compute $m' = s' - H_2(r_i)s \bmod q$.

3. Verify that $g^{s'} = g^{m'}\beta' \bmod p$. If the verification succeeds, then receiver i is the targeted decypherer and it outputs the decrypted message m' and halts. Otherwise, the receiver repeats the steps for another four-tuple sequence.

4.2 Security Analysis

Theorem 1. *Our CH-VE scheme above is no-colluder secure under the random oracle model, provided all encryptions are IND-CCA2 secure.*

Proof in the Appendix. **Practical Security and Performance:** We recommend N to be approximately $80-100$ which is equivalent to about 2^{-64} and it should be sufficient for most applications. We have in mind OAEP for the component encryptions and elliptic curve is used as the homomorphic one-way function. Let the length of each encryption be 1024 bits. The communication bandwidth is about 4096 bits for each round. If there are 100 rounds, the total bandwidth is about $50n$ kbytes and the size of the ciphertext is about $17n$ kbytes, where n is the number of custodians. Our bandwidth is larger than the VE scheme of [2], for preserving the anonymity of the actual decryptor.

4.3 A Secure Custodian Hiding Verifiable Encryption (CH-VE) Scheme (with colluders)

The previous CH-VE scheme is not secure if Adversary has a colluder. In that case Adversary obtains s from Colluder, then exhaustively tests out $g^{m_b} \overset{?}{=} g^s g^{H_2(r_i)}$ to distinguish m_0 from m_1. In this section, we modify the above scheme to provide security even in the existence of colluders. We use the same notation as described in the first scheme unless otherwise stated.

[**Protocol Between** P **and** V **(Encryption).** Let $L = \log n$]

1. P computes $d = g^m \bmod p$ and sends d to V.
2. Repeat the following steps N times in parallel.
   a. (*Commitment*) P randomly picks $s \in_R \mathbb{Z}_q$, $r_i^{(j)} \in_R \{0,1\}^{\lambda_s}$ for $1 \le i \le n$, $1 \le j \le L$ and $\phi_j \in_R \mathrm{Sym}(n)$ for $1 \le j \le L$. P computes

$$\begin{aligned}\lambda^{(j)} &= E_1(r_1^{(j)})||\cdots||E_n(r_n^{(j)})\\ \gamma^{(j)} &= (g^{H_2(r^{(j)}_{\phi_j(1)})} \bmod p,\ \cdots,\ g^{H_2(r^{(j)}_{\phi_j(n)})} \bmod p)\\ \alpha' &= E_1(s)||\cdots||E_n(s), \quad \alpha = H_1(\alpha')\\ \beta^{(j)} &= g^{H_2(r_\pi^{(j)})s} \bmod p\\ \theta^{(j)} &= H_1(\lambda^{(j)}||\gamma^{(j)}||\alpha||\beta^{(1)}\ \cdots\beta^{(L)})\end{aligned}$$

   and sends $\theta^{(j)}$ to V, for $1 \le j \le L$.

   b. (*Challenge*) V picks $b \in_R \{1,2,3\}$ and sends b to P.

c. (*Response*)
  - Case $b = 1$, P sends $r_1^{(j)}, \cdots, r_n^{(j)}$, $\gamma^{(j)}$, α and $\beta^{(j)}$ to V, for $1 \le j \le L$.
  - Case $b = 2$, P sends $\lambda^{(j)}$, $\gamma^{(j)}$, and s to V, for $1 \le j \le L$.
  - Case $b = 3$, P sends $\lambda^{(j)}$, $\gamma^{(j)}$, α', and $s' = (H_2(r_\pi^{(1)}) + \ldots + H_2(r_\pi^{(L)}))s + m \bmod q$ to V, for $1 \le j \le L$.

d. (*Verification* by V)
  - Case $b = 1$:
    - Verify that $r_1^{(j)}, \cdots, r_n^{(j)}$ are distinct, for $1 \le j \le L$.
    - Verify that there exists a unique permutation $\delta \in \mathrm{Sym}(n)$ s.t. $\gamma^{(j)} = (g^{H_2(r_{\delta(1)}^{(j)})} \bmod p, \cdots, g^{H_2(r_{\delta(n)}^{(j)})} \bmod p)$ for $1 \le j \le L$.
    - Verify that $\theta^{(j)} = H_1(\hat{\lambda}^{(j)} || \gamma^{(j)} || \alpha || \beta^{(1)} \cdots \beta^{(L)})$ where $\hat{\lambda^{(j)}} = E_1(r_1^{(j)}) || \cdots || E_n(r_n^{(j)})$, for $1 \le j \le L$.

    Continue only if all verifications succeed.
  - Case $b = 2$:
    - Denote $\gamma^{(j)} = (\gamma_1^{(j)}, \cdots, \gamma_n^{(j)})$.
    - Compute $\tilde{\alpha} = H_1(E_1(s) \,||\, \cdots \,||\, E_n(s))$ and $\beta_i^{(j)} = (\gamma_i^{(j)})^s \bmod p$, for $1 \le i \le n$, $1 \le j \le L$.
    - Verify that $\theta^{(j)} = H_1(\lambda^{(j)} || \gamma^{(j)} || \tilde{\alpha} || \beta_{i_1}^{(1)} \cdots \beta_{i_L}^{(L)})$ for exactly one index $i_\ell \in \{1, \cdots, n\}$, $1 \le \ell \le L$, $1 \le j \le L$.

    Continue only if the verification succeeds.
  - Case $b = 3$:
    - Compute $\beta' = g^{s'}/d \bmod p$
    - Verify that $\theta^{(j)} = H_1(\lambda^{(j)} || \gamma^{(j)} || H_1(\alpha') || \beta')$ for $1 \le j \le L$.

    Continue only if the verification succeeds.

3. (*Output*) V terminates if any verification fails in any of the N cut-and-choose Rounds. Otherwise, it outputs d and the four-tuple sequences $(\alpha', \lambda^{(j)}, \beta', s')$ for $1 \le j \le L$ and for all Case-($b = 3$) Rounds to all n receivers as the ciphertext, $\mathcal{C}$.

Recovery Algorithm R. Denote $\bar{\lambda}_1^{(j)} || \cdots || \bar{\lambda}_n^{(j)} = \lambda^{(j)}$ and $\bar{\alpha'}_1 || \cdots || \bar{\alpha'}_n = \alpha'$. For d and each four-tuple sequence $(\alpha', \lambda^{(j)}, \beta', s')$, $1 \le j \le L$, each receiver i, $1 \le i \le n$, independently performs the following steps.

1. Compute $r_i^{(j)} = E_i^{-1}(\bar{\lambda}_i^{(j)})$ and $s = E_i^{-1}(\bar{\alpha'}_i)$.
2. Compute $m' = s' - (H_2(r_i^{(1)}) + \ldots + H_2(r_i^{(L)}))s \bmod q$.
3. Verify that $g^{s'} = g^{m'}\beta' \bmod p$. If the verification succeeds, then receiver i is the targeted decypherer and it outputs the decrypted message m' and halts. Otherwise, the receiver repeats the steps for another four-tuple sequence.

Theorem 2. *The above CH-VE scheme is secure under the random oracle model, provided that all encryptions are IND-CCA2 secure.*

A sketch of the proof is in the Appendix.

4.4 Extensions

We present extensions of our CH-VE schemes to CH-GVE (Custodian-Hiding Group Verifiable Encryption) schemes.

Verifiable (t,t,n) Encryption for Anonymous Ad Hoc Groups In our basic VE-VE scheme, only a single targeted member can decrypt the message. Here we make an extension such that a targeted t-member subset of the ad hoc group of n receivers can jointly recover the message. On the notation of (t,t,n), symbol 'n' represents that P spontaneously forms a group of n receivers; the second symbol 't' represents that t targeted members of the group can recover a message; and the first symbol 't' means that all the t targeted members need to work jointly to recover the message. By using similar notation, we propose another extension shortly which allows *any* member of a targeted t-member subset of the ad hoc group of n receivers to recover the message. Hence the notation of the second extension is $(1,t,n)$.

Below is a (t,t,n) CH-GVE scheme.

Encryption. Here we use $\pi_1,\cdots,\pi_t$ to index the targeted receivers, where $t<n$ and $\pi_1,\cdots,\pi_t \in \{1,\cdots,n\}$ are distinct. The encryption algorithm is similar to the basic scheme described in Sec. 4, with the following modifications.

1. P also sends t to V before Commitment.
2. (Commitment) Compute as before, except
$$\beta = (g^{H_2(r_{\pi_1})} \cdot g^{H_2(r_{\pi_2})} \cdot \ldots \cdot g^{H_2(r_{\pi_t})})^s \bmod p$$
3. (Response) Compute as before, except that in Case b=3:
$$s' = (H_2(r_{\pi_1}) + \cdots + H_2(r_{\pi_t}))s + m \bmod q$$
4. (Verification)
   (a) Case b=1: Same as before.
   (b) Case b=2: Process γ, $\tilde{\alpha}$, β_i as before. Verify
$$\theta = H_1(\lambda \mid\mid \gamma \mid\mid \tilde{\alpha} \mid\mid \beta_{i_1} \cdots \beta_{i_t})$$
   for a unique t-element subset $\{i_1,\cdots,i_t\} \subset \{1,\cdots,n\}$.
   (c) Case b=3: No change.

Decryption. Same as before, except that t targeted decypherers jointly compute
$$m' = s' - (H_2(r_{\pi_1}) + \cdots + H_2(r_{\pi_t}))s \bmod q$$

$(1, t, n)$ CH-GVE scheme The $(1, t, n)$ CH-GVE scheme allows *any* one in a targeted set of t receivers to recover the encrypted message, and then the receivers are anonymized.

Encryption. Let $\pi_1, \cdots, \pi_t$ be the index of t targeted receivers. The encryption algorithm is similar to before, but with small modifications:

1. P also sends t to V before Commitment.
2. (Commitment) Same as before except
$$\beta = g^{H_2(r_{\pi_1})s} \bmod p \,||\cdots||\, g^{H_2(r_{\pi_t})s} \bmod p$$
3. (Response) Same as before except in Case b=3, replace the original s' with
$$s'_i = H_2(r_{\pi_i})s + m \bmod q$$
for $i = 1, \cdots, t$.
4. (Verification) Same as before except in
   (a) Case b=2, verify that
   $$\theta = H_1(\lambda \,||\, \gamma \,||\, \tilde{\alpha} \,||\, (\beta_{i_1}||\cdots||\beta_{i_t}))$$
   for a unique t-member ordered tuples $\{i_1, \cdots, i_t\} \subset \{1, \cdots, n\}$.
   (b) Case $b = 3$, compute
   $$\beta' = g^{s'_1}/d \bmod p \;||\; \cdots \;||\; g^{s'_t}/d \bmod p$$
5. (Output) Same as before except replacing the original s' with $s'_1, \cdots, s'_t$.

Decryption
- Denote $\bar{\beta}'_1 \;||\; \cdots \;||\; \bar{\beta}'_t = \beta'$.
- Step 2 is modified as: receiver i computes $m'_{i,j} = s'_j - H_2(r_i)s \bmod q$ for $j = 1, \cdots, t$.
- Step 3 is modified as: receiver i checks if $g^{s'_j} \overset{?}{=} g^{m'_{i,j}} \bar{\beta}'_j \bmod p$ for $j = 1, \ldots, t$. If one of them equal, then receiver i is one of the targeted decypherers.

5 Concluding Remarks

In this paper, we propose a new paradigm of Custodian-Hiding Verifiable Encryption (CH-VE) which allows the prover to specify any set of n receivers and send an encrypted message such that the verifier can make sure that the encrypted message can be decrypted by at least one of the receivers. Yet the verifier knows nothing about the identity of the actual decryptor. The complexity of our proposing scheme is linear in the size of the receiver group. We give two instantiations of CH-VE with different level of security.

We believe that other intriguing and efficient CH-VE schemes and various security models can be attained. Other variants and features may also be constructed. For example, it would be interesting to construct a general verifiable (k, t, n) encryption scheme.

References

1. B. Aiello, Y. Ishai, and O. Reingold. Priced oblivious transfer: how to sell digital goods. In *Proc. EUROCRYPT 2001*, pages 119–135. Springer-Verlag, 2001. LNCS No. 2045.
2. N. Asokan, V. Shoup, and M. Waidner. Optimistic fair exchange of digital signatures. In *Proc. EUROCRYPT 98*, pages 591–606. Springer-Verlag, 1998. LNCS No. 1403.
3. G. Ateniese. Verifiable encryption of digital signatures and applications. *ACM Transactions on Information and System Security*, 7(1):1–20, February 2004.
4. F. Bao. An efficient verifiable encryption scheme for encryption of discrete logarithms. In *Proc. Smart Card Research and Applications (CARDIS) 1998*, pages 213–220. Springer-Verlag, 2000. LNCS No. 1820.
5. G. Brassard, C. Crepeau, and J. Robert. Information theoretic reductions among disclose problem. In *Proc. 27th IEEE Symp. on Foundations of Comp. Science*, pages 168–173. IEEE, 1986.
6. 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. LNCS No. 1976.
7. J. Camenisch and A. Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocations. In *Proc. EUROCRYPT 2001*, pages 93–118. Springer-Verlag, 2001. LNCS No. 2045.
8. J. Camenisch and M. Michels. Separability and efficiency for generic group signature schemes. In *Proc. CRYPTO 99*, pages 413–430. Springer-Verlag, 1999. LNCS No. 1666.
9. J. Camenisch and V. Shoup. Practical verifiable encryption and decryption of discrete logarithms. LNCS No. 2729, 2003.
10. B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch. Verifiable secret sharing and achieving simultaneity in the presence of faults. In *Proc. 26th IEEE Symp. on Foundations of Comp. Science*, pages 383–395, Portland, 1985. IEEE.
11. C. Crepeau. Equivalence between two flavours of oblivious transfers. In *Proc. CRYPTO 87*, pages 350–354. Springer-Verlag, 1987. LNCS No. 293.
12. C. Crepeau and J. Kilian. Weakening security assumptions and oblivious transfer. In *Proc. CRYPTO 88*, pages 2–7. Springer-Verlag, 1988. LNCS No. 403.
13. B. den Boer. Oblivious transfer protecting secrecy. In *Proc. EUROCRYPT 90*, pages 31–46. Springer-Verlag, 1990. LNCS No. 473.
14. T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, IT-31(4):496–472, July 1985.
15. S. Even, O. Goldreich, and A. Lempel. A randomized protocol for signing contracts. In *Proc. CRYPTO 82*, pages 205–210. Springer-Verlag, 1982.
16. J. Kilian and E. Petrank. Identity escrow. In *Proc. CRYPTO 98*, pages 169–185. Springer-Verlag, 1998. LNCS No. 1642.
17. Y. Mu, J. Zhang, and V. Varadharajan. m out of n oblivious transfer. In *ACISP02*, pages 395–405. Springer-Verlag, 2002. LNCS No. 2384.
18. M. Rabin. How exchange secrets by oblivious transfer. Technical Report TR-81, Computer Science Laboratory, Harvard, 1981.
19. C. Rackoff and D. Simon. Noninteractive zero-knowledge proof of knowledge and chosen ciphertext attack. In *Proc. CRYPTO 91*, pages 433–444. Springer, 1991. LNCS No. 576.

20. M. Stadler. Publicly verifiable secret sharing. In *Proc. EUROCRYPT 96*, pages 191–199. Springer-Verlag, 1996. LNCS No. 1070.

A Proofs

In this Appendix, we prove Theorem 1 and then Theorem 2,

Completeness Clear.

Soundness Assume Prover P has an overwhelming probability of passing the verification in all cut-and-choose rounds. Then in each round, P must supply the following parameters in response to various challenge values generated by an honest V, and these parameters must pass all verifications in their respective Cases: $\check{\theta}$ (for Commitment); $\hat{r}_1, \cdots, \hat{r}_n, \hat{\gamma}, \hat{\alpha}, \hat{\beta}$ (for Case $b = 1$); $\tilde{\lambda}, \tilde{\gamma}, \tilde{s}$ (for Case $b = 2$); $\bar{\lambda}, \bar{\gamma}, \bar{\alpha}', \bar{s}'$ (for Case $b = 3$). We show below then there exists a unique decypherer. Furthermore, it recovers the message m accurately, for all $m \in \mathbb{Z}_q$.

Since H_1 is ideal, we have $\hat{\lambda} \mid\mid \hat{\gamma} \mid\mid \hat{\alpha} \mid\mid \hat{\beta} = \tilde{\lambda} \mid\mid \tilde{\gamma} \mid\mid \tilde{\alpha} \mid\mid \tilde{\beta} = \bar{\lambda} \mid\mid \bar{\gamma} \mid\mid \bar{\alpha} \mid\mid \bar{\beta}$ with overwhelming probability, where

$$\begin{aligned}
&\hat{\lambda} = E_1(\hat{r}_1) \mid\mid \cdots \mid\mid E_n(\hat{r}_n)\\
&\tilde{\alpha} = H_1(E_1(\tilde{s}) \mid\mid \cdots \mid\mid E_n(\tilde{s}))\\
&\tilde{\beta} = \tilde{\gamma}_\ell^{\tilde{s}}, \ell \text{ is the unique index found in Verification Case } b\text{=}2\\
&\bar{\alpha} = H_1(\bar{\alpha}')\\
&\bar{\beta} = g^{\bar{s}'}/d \bmod p
\end{aligned}$$

Combining and with overwhelming probability, we have

$$\begin{aligned}
\bar{\beta} = g^{\bar{s}'}/d = \tilde{\beta} = \tilde{\gamma}_\ell^{\tilde{s}} = \hat{\gamma}_\ell^{\tilde{s}} = g^{H_2(\hat{r}_{\delta(\ell)})\tilde{s}} \bmod p\\
\bar{s}' - \bar{m} = H_2(\hat{r}_{\delta(\ell)})\tilde{s} \bmod q
\end{aligned}$$

where $\delta \in \mathrm{Sym}(n)$ is the permutation from Case b=1, and $\bar{m}$ is such that $g^{\bar{m}} = d$.

Denote $\bar{\lambda} = \bar{\lambda_1} \mid\mid \ldots \mid\mid \bar{\lambda}_n$ and let $\bar{r}_i = E_i^{-1}(\bar{\lambda}_i)$, for $1 \leq i \leq n$. That $\bar{\lambda} = \hat{\lambda}$ implies $\bar{r}_{\delta(\ell)} = \hat{r}_{\delta(\ell)}$. Denote $\bar{\alpha}' = \bar{\alpha}_1' \mid\mid \ldots \mid\mid \bar{\alpha}_n'$ and let $\bar{s}_i = E_i^{-1}(\bar{\alpha}_i')$, for $1 \leq i \leq n$. That $\bar{\alpha} = \tilde{\alpha}$ implies $\bar{s}_{\delta(\ell)} = \tilde{s}$. Therefore,

$$\bar{m} = \bar{s}' - H_2(\bar{r}_{\delta(\ell)})\bar{s} = \bar{s}' - H_2(E_{\delta(\ell)}^{-1}(\bar{\lambda}_{\delta(\ell)}))E_{\delta(\ell)}^{-1}(\bar{\alpha}'_{\delta(\ell)}) \bmod q$$

Member $\delta(\ell)$, $1 \leq \delta(\ell) \leq n$, can decypher the message $\bar{m}$ satisfying $g^{s'} = g^{\bar{m}}\beta'$, where $\beta' = \bar{\beta}$. From Case b=2, δ is unique. From Case b=1, $\hat{r}_1, \cdots, \hat{r}_n$ are distinct and thus δ is unique. Therefore, the decypherer is unique with overwhelming probability.

We already have $g^{\bar{m}} = d$ above. In a Completeness proof, P is honest and $d = g^m$. Therefore, $\bar{m} = m$, the message recovery is accurate.

The above proves that if V satisfies with probability non-negligibly higher than 2/3 in an individual cut-and-choose round, then there exists a unique decypherer for that round. In our current cut-and-choose scheme, the unique decypherer from different rounds may differ.

Anonymity We prove that no PPT Adversary has non-negligile advantage in Game D-A.

We have $q_C = 0$ by Theorem assumptions. Simulating the Prover Oracle is easy: it involves only encrypting. Individual encryptions are IND-CCA2-secure by Theorem assumptions. Therefore their individual Decryption Oracles can be simulated.

In the following proof, we assume Adversary knows the value of π_0 and has a non-nogligible advantage in Game D-A. We will construct a Simulator from Adversary which can solve a hard problem. We adopt the simplifications $n = 2$ and $\pi = \pi_{b_G}$.

Assume the verifier V can compute the identity of the targeted decypherer with probability $1/n + \epsilon(k)$, where ϵ is a non-negligible function. We say that ϵ is non-negligible if there exists a polynomial ρ such that $\epsilon(k) > 1/\rho(k)$. Then V must solve one of the following problems with probability at least $1/n + \epsilon(k)$.

A. V can compute the identity (with probability at least $1/n + \epsilon(k)$) in Case $b = 1$ of an individual round.
B. V can compute the identity in Case $b = 2$ of an individual round.
C. V can compute the identity in Case $b = 3$ of an individual round.
D. V can compute the identity based on transcripts (commit, challenge, response) of multiple rounds, all of which are Cases $b = 3$.
E. V can compute the identity based on all N transcripts.

Problem A: Given $(r_1, \cdots, r_n, \gamma, \alpha, \beta)$ where $r_i \in \{0,1\}^k$, $1 \leq i \leq n$, $\gamma = (g^{H_2(r_{\phi(1)})}, \cdots, g^{H_2(r_{\phi(n)})})$, $\alpha = H_1(E_1(s)|| \cdots ||E_n(s))$, $\beta = g^{H_2(r_\pi)s}$ for some $\phi \in \mathrm{Sym}(n)$, $s \in F_q$, and $\pi \in \{1, \cdots, n\}$, find π.

In the following, we show in the random oracle model that if Problem A is easy, the discrete logarithm problem (DLP) is easy.

Lemma 3 *Suppose a PPT algorithm V, with H_1 and H_2 being random oracles, solves Problem A with probability at least $1/n + \epsilon(k)$. There exists a PPT algorithm $\mathcal{M}$, which invokes V and simulates the view of V by answering all its H_1-queries and H_2-queries, can compute the discrete logarithm problem (DLP) with probability at least $\frac{n}{n-1}\epsilon(k)$.*

We construct $\mathcal{M}$ as follows.

$\mathcal{M}$ = "On input $Y \in G$,
1. Randomly pick $r_1, \cdots, r_n \in_R \{0,1\}^k$ and $R_1, \cdots, R_n \in_R F_q$. Set the values of $H_2(r_i) = R_i$, for all $1 \leq i \leq n$.
2. Randomly pick $\alpha \in_R \{0,1\}^k$.
3. Set $\beta = Y$ and $\gamma = (g^{R_1}, \cdots, g^{R_n})$
4. Randomly generate secure asymmetric encryption functions $\bar{E}_1, \cdots, \bar{E}_n$ (whose decryption functions are generated by, and known to, $\mathcal{M}$).
5. Run V on corresponding inputs and reply all the queries of H_1 and H_2 in the following manner.

- For any H_2-query with input r_i, $1 \le i \le n$, R_i is replied.
- For a H_1-query with input $Z_1||\cdots||Z_n$, compute $s'_i \leftarrow \bar{E}_i^{-1}(Z_i)$, $1 \le i \le n$, and determine if $s'_1 = \cdots = s'_n$ and $Y = g^{R_\ell s'_1}$, for some $1 \le \ell \le n$. If they are true, output $R_\ell s'_1$ and halt. Otherwise, randomly pick an element from $\{0,1\}^k \setminus \{\alpha\}$ as the reply.
- For any other queries of H_1 and H_2, random numbers are generated in the corresponding range of H_1 and H_2 as the replies.
- For query consistency, for any query with an input value which has been received before, the same reply as the last time is returned.

6. Halt with no output if V stops."

Since V is a PPT, the complexity of $\mathcal{M}$ is also in polynomial time. Let $\mathbf{E}$ denote the event that V queries H_1 with $Z_1||\cdots||Z_n$ such that $s = \bar{E}_1^{-1}(Z_1) = \cdots = \bar{E}_n^{-1}(Z_n)$ and $Y = g^{R_\ell s}$ for some $1 \le \ell \le n$. Then

$$\begin{aligned}1/n + \epsilon(k) &\le \Pr[\text{V solves Problem A}]\\ &\le \Pr[\mathbf{E}]\Pr[\text{V solves Problem A}|\mathbf{E}] + \Pr[\bar{\mathbf{E}}]\Pr[\text{V solves Problem A}|\bar{\mathbf{E}}]\\ &\le \eta(k)\cdot 1 + (1-\eta(k))\cdot(1/n) = (1/n) + (1 - 1/n)\eta(k)\end{aligned}$$

where $\eta(k) = \Pr[\mathbf{E}]$. In the event $\mathbf{E}$, which has non-negligible probability $\eta(k) \ge n/(n-1)\cdot\epsilon(k)$, $\mathcal{M}$ obtains its DLP answer. This contradicts Theorem assumptions.

Remark: To see why V cannot do better than random guess in the event $\bar{\mathbf{E}}$, assume V mysteriously obtains a value X such that $\beta = g^X$. Notice that for each R_i, $1 \le i \le n$, there is a value s_i such that $X = R_i \cdot s_i$. In the event $\bar{\mathbf{E}}$, the outcome of the following n queries are yet to be generated by $\mathcal{M}$'s random tape: $H_1(\bar{E}_1(s_i)||\cdots||\bar{E}_n(s_i))$, $1 \le i \le n$. These outcomes are not yet generated by the time V returns its output to $\mathcal{M}$, and thus V essentially cannot do better than random guess, even if it mysteriously knows the discrete logarithm of β. The detailed proof is technical and omitted.

Problem B is equivalent to Problem *B'* below.

Problem B' : Given ℓ such that $\pi = \phi(\ell)$, $E_1(r_1)||\cdots||E_n(r_n)$, $g^{H_2(r_{\phi(1)})}$, $\cdots$, $g^{H_2(r_{\phi(n)})}$, s, and $g^{H_2(r_\pi)s}$, compute π. Note $\phi \in \text{Sym}(n)$ and $r_1, \cdots, r_n$ are unspecified.

Lemma 4 *Suppose a PPT V', after making q_{H_2} queries of H_2, computes Problem B' with probability $1/n + \epsilon(k)$. There exists a PPT $\mathcal{M}$ which invokes V' and answers all H_2-queries, can invert one of E_1, $\cdots$, E_n with probability at least $\frac{n}{n-1}\cdot\frac{1}{q_{H_2}}\cdot\epsilon(k)$.*

To compute at least one of the asymmetric inversions $E_1^{-1}(Z_1), \cdots, E_n^{-1}(Z_n)$, $\mathcal{M}$ randomly generates $\phi \in_R \text{Sym}(n), \pi, s, Y_1, \cdots, Y_n$, and invokes V' with inputs ℓ such that $\pi = \phi(\ell)$, $Z_1||\cdots||Z_n$, $g^{Y_{\phi(1)}}, \cdots, g^{Y_{\phi(n)}}$, s, and $g^{Y_\pi s}$.

Let $\mathbf{E}$ denote the event that V' queries H_2 with z satisfying $Z_i \leftarrow E_i(z)$ for some i, $1 \leq i \leq n$. Let $\Pr\{\mathbf{E}\} = \eta(k)$. Then

$$\begin{aligned} 1/n + \epsilon(k) &\leq \Pr\{V' \text{ solves}\} \\ &= \Pr\{\mathbf{E}\}\Pr\{V' \text{ solves } |\mathbf{E}\} + \Pr\{\bar{\mathbf{E}}\}\Pr\{V' \text{ solves } |\bar{\mathbf{E}}\} \\ &\leq \eta(k) \cdot 1 + (1 - \eta(k))(1/n) = (1/n) + (1 - 1/n)\eta(k) \end{aligned}$$

Note in the event of $\bar{\mathbf{E}}$, V' essentially cannot do better than random guess even if he knows all values $r_1, \cdots, r_n$ because the query outcomes $H_2(r_{\phi(1)}), \cdots H_2(r_{\phi(n)})$ are yet to be randomly generated by $\mathcal{M}$'s random tape by the time V' completes.

Note that $\mathcal{M}$ has to identify the occurence of event . This may be accomplished to test any given query z of H_2 such that $Z_i = E_i(z)$, for some i, $1 \leq i \leq n$. However, $\mathcal{M}$ may not be able to do this if a probabilistic public-key encryption function E_i's random tape for generating Z_i is unknown. $\mathcal{M}$ has to randomly pick one query out of q_{H_2} H_2-queries and hopes it is the right moment of event $\mathbf{E}$. Therefore, $\mathcal{M}$ succeeds in inverting one of the encryptions with probability at least $\frac{n\epsilon(k)}{(n-1)q_{H_2}}$. This contradicts the Theorem assumption that each E_i is secure against PPT adversaries.

Problem C: We re-iterate Problem C below.

Problem C: Given $\lambda_i = E_i(r_i)$, $1 \leq i \leq n$, $\gamma = g^{H_2(r_{\phi(1)})}|| \cdots ||g^{H_2(r_{\phi(n)})}$, $\alpha' = E_1(s)|| \cdots ||E_n(s)$, $s' = H_2(r_\pi)s + m$, $d = g^m$, $\theta = H_1(\lambda||\gamma||H_1(\alpha')||g^{s'}/d)$, $E_1, \cdots, E_n$, H_1, H_2, compute π. Note $\phi \in \mathrm{Sym}(n)$, s, and $r_i, \cdots, r_n$ are unspecified.

Problem C is reducible to inverting one of encryptions $E_1, \cdots, E_n$, under the random oracle model. We use a special form of the random oracle.

(*A formulation of the random oracle with n permutable back patches*) For asymmetric encryption functions $E_1, \cdots, E_n$, let $\mathcal{H}(E_1, \cdots, E_n)$ denote the random oracle which generates its outputs in the following way:

1. $\mathcal{H}$ randomly generates n distinct values $Z_1, \cdots, Z_n \in \{1, \cdots, q\}$.
2. $\mathcal{H}$ randomly generates n distinct values $Y_1, \cdots, Y_n \in \{1, \cdots, q\}$.
3. Let X be a query to $\mathcal{H}$. If X has never been queried before, $\mathcal{H}$ checks if $E_i(X) = Z_i$ for some i, $1 \leq i \leq n$. Case yes, randomly select a member Y from $\{Y_1, \cdots, Y_n\}$ which has never been selected before and set $\mathcal{H}(X) = Y$. Case no, set $\mathcal{H}(X)$ to a random member of $\{1, \cdots, q\} \backslash \{Y_1, \cdots, Y_n\}$ which has never been outputted by $\mathcal{H}$ before. On duplicated queries X, $\mathcal{H}$ maintains consistency with previous outputs.

Remark: The above remains a random oracle.

Lemma 5 *Suppose a PPT algorithm V solves Problem C with probability at least $1/n + \epsilon(k)$. There exists a PPT algorithm, $\mathcal{M}$, which invokes V and simulates the view of V by replacing its queries to H_2 by queries to $\mathcal{H}(t, E_1, \cdots, E_n)$, can compute (the discrete logarithm of) at least one of (distinct) $E_t^{-1}(Z_{t+1}), \cdots, E_n^{-1}(Z_n)$, with probability at least $\frac{n}{n-1}\epsilon(k)$. The complexity of $\mathcal{M}$ is in the same order as that of V.*

In the following we abbreviate $\mathcal{H} = \mathcal{H}(E_1, \cdots, E_n)$. Assume PPT algorithm V solves Problem C with probability $1/n + \epsilon(k)$. $\mathcal{M}$ randomly generates, on behalf of $\mathcal{H}$, the values Z_1, $\cdots$, Z_n, Y_1, $\cdots$, Y_n. Further, $\mathcal{M}$ randomly generates m, s, and ℓ with $1 \leq \ell \leq n$. Then $\mathcal{M}$ invokes $\mathcal{V}$ with inputs $\lambda_1 = Z_1$, $\cdots$, $\lambda_n = Z_n$, $\gamma = g^{Y_1}||\cdots||g^{Y_n}$, $\alpha' = E_1(s)||\cdots||E_n(s)$, $s' = Y_\ell s + m$, $d = g^m$, $\theta = H_1(\lambda||\gamma||H_1(\alpha')||g^{s'}/d)$. As $\mathcal{V}$ executes, $\mathcal{M}$ records its queries to $\mathcal{H}$, $\mathcal{M}$ simulates $\mathcal{H}$ in answering queries including flips coins, and $\mathcal{M}$ checks the eventual output from $\mathcal{V}$. Note $\mathcal{M}$ can simulate $\mathcal{H}$ in polynomial time.

Let $\bar{\mathbf{E}}$ denote the event $\mathcal{V}$ does not query $\mathcal{H}$ with any input X satisfying $E_i(X) = Z_i$ for any i with $1 \leq i \leq n$. In the event $\mathbf{E}$, V queries $\mathcal{H}$ with an input X satisfying $H_i(X) = Z_i$ for some i, $1 \leq i \leq n$. Then $\mathcal{M}$ has obtained the desired asymmetric decryption $X = E_i^{-1}(Z_i)$. In the event $\bar{\mathbf{E}}$, the correspondence between the Y_i's and the Z_i's in the third step of $\mathcal{H}$ specification has not yet been decided by the time $\mathcal{V}$ completes. The value of π such that there exists X with $E_\pi(X) = Z_\pi$ and $\mathcal{H}(X) = Y_\ell$ has yet to be decided with at least n equally probable candidates $\pi \in \{1, \cdots, n\}$. $\mathcal{M}$ will have to flip additional coins in order to choose π among the candidates. Therefore

$$\begin{aligned}\frac{1}{n} + \epsilon(k) &\leq \Pr\{V \text{ succeeds}\} \\ &= \Pr\{\mathbf{E}\}\Pr\{V \text{ succeeds}|\mathbf{E}\} + \Pr\{\bar{\mathbf{E}}\}\Pr\{V \text{ succeeds}|\bar{\mathbf{E}}\} \\ &\leq \eta \cdot 1 + (1-\eta)\frac{1}{n}\end{aligned}$$

where $\eta = \Pr\{\mathbf{E}\}$. $\mathcal{M}$'s probability of success is at least η and $\eta \geq \frac{n}{n-1}\epsilon(k)$.

Problem D: For simplicity, we prove for the scenario where there are two rounds with Case b=3. Other scenarios are similar. Problem D is reiterated below.

Problem D: Given g^m, $E_1, \cdots, E_n$, $\lambda = E_1(r_1)||\cdots||E_n(r_n)$, $\bar{\lambda} = E_1(\bar{r}_1)||\cdots|| E_n(\bar{r}_n)$, $\gamma = g^{H_2(r_{\phi(1)})} || \cdots || g^{H_2(r_{\phi(n)})}$, $\bar{\gamma} = g^{H_2(\bar{r}_{\bar{\phi}(1)})}||\cdots||g^{H_2(\bar{r}_{\bar{\phi}(n)})}$, $\alpha' = E_1(s)||\cdots||E_n(s)$, $\bar{\alpha}' = E_1(\bar{s})||\cdots||E_n(\bar{s})$, $s' = H_2(r_\pi)s + m$, $\bar{s}' = H_2(\bar{r}_\pi)\bar{s} + m$, $\theta = H_1(\lambda||\gamma||H_1(\alpha')||g^{s'}/d)$, and $\bar{\theta} = H_1(\bar{\lambda}||\bar{\gamma}||H_1(\bar{\alpha}')||g^{\bar{s}'}/d)$, compute π.

Problem D is reducible to inverting one of the asymmetric encryptions E_1, $\cdots$, E_n, under the random oracle model. Like Problem C, We use a special form of the random oracle.

(*A formulation of the random oracle in terms of* $2n$ *relations*) For asymmetric encryption functions E_1, $\cdots$, E_n, let $\mathcal{H}_D(E_1, \cdots, E_n)$ denote the random oracle which generates its outputs in the following way:

1. $\mathcal{H}_D$ randomly generates $2n$ distinct values Z_1, $\cdots$, Z_n, $\bar{Z}_1$, $\cdots$, $\bar{Z}_n$ such that $Z_i, \bar{Z}_i$ are in the range of E_i, for all i, $1 \leq i \leq n$.
2. $\mathcal{H}_D$ randomly generates $2n$ distinct values Y_1, $\cdots$, Y_n, $\bar{Y}_1$, $\cdots$, $\bar{Y}_n \in_R \mathbb{Z}_q$.
3. Let X be a query to $\mathcal{H}_D$. If X has never been queried before, $\mathcal{H}_D$ checks if $E_i(X) = Z_i$ for some i, $1 \leq i \leq n$. Case yes, randomly select a member Y from $\{Y_1, \cdots, Y_n\}$ which has never been selected before and set $\mathcal{H}_D(X) = Y$. Case no, set $\mathcal{H}_D(X)$ to a random member of $\{1, \cdots, q\} \setminus \{Y_1, \cdots, Y_n\}$

which has never been outputted by $\mathcal{H}_D$ before. Otherwise : $\mathcal{H}_D$ checks if $E_i(X) = \bar{Z}_i$ for some i, $1 \le i \le n$. If yes, randomly select a member Y from $\{\bar{Y}_1, \cdots, \bar{Y}_n\}$ which has never been selected before and set $\mathcal{H}_D(X) = Y$. If no, set $\mathcal{H}_D(X)$ to a random member of $\{1, \cdots, q\} \setminus \{Y_1, \cdots, Y_n, \bar{Y}_1, \cdots, \bar{Y}_n\}$ which has never been outputted by $\mathcal{H}_D$ before. On duplicated queries X, $\mathcal{H}_D$ maintains consistency with previous outputs.

Remark: The above remains a random oracle.

Lemma 6 *Suppose a PPT algorithm V solves Problem D with probability at least $1/n + \epsilon(k)$. There exists a PPT algorithm, $\mathcal{M}$, which invokes V and simulates the view of V by replacing its queries to H_2 by queries to $\mathcal{H}_D(E_1, \cdots, E_n)$, can compute (the discrete logarithm of) at least one of (distinct) $E_1^{-1}(Z_{t+1})$, $\cdots$, $E_n^{-1}(Z_n)$, with probability at least $\frac{n}{n-1}\epsilon(k)$. The complexity of $\mathcal{M}$ is in the same order as that of V.*

In the following we abbreviate $\mathcal{H}_D = \mathcal{H}_D(E_1, \cdots, E_n)$. Assume PPT algorithm V solves Problem D with probability $1/n + \epsilon(k)$. $\mathcal{M}$ randomly generates, on behalf of $\mathcal{H}_D$, the values Z_1, $\cdots$, Z_n, Y_1, $\cdots$, Y_n. and $\bar{Z}_1$, $\cdots$, $\bar{Z}_n$, $\bar{Y}_1$, $\cdots$, $\bar{Y}_n$. Further, $\mathcal{M}$ randomly generates m, s, $\bar{s}$, ℓ with $1 \le \ell \le n$. $\bar{\ell}$ with $1 \le \bar{\ell} \le n$. Then $\mathcal{M}$ invokes $\mathcal{V}$ with inputs g^m, E_1, $\cdots$, E_n, $\lambda = Z_1 || \cdots || Z_n$, $\bar{\lambda} = \bar{Z}_1 || \cdots || \bar{Z}_n$, $\gamma = g^{Y_1} || \cdots || g^{Y_n}$, $\bar{\gamma} = g^{\bar{Y}_1} || \cdots || g^{\bar{Y}_n}$, $\alpha' = E_1(s) || \cdots || E_n(s)$, $\bar{\alpha}' = E_1(\bar{s}) || \cdots || E_n(\bar{s})$, $s' = Y_\ell s + m$, $\bar{s}' = \bar{Y}_{\bar{\ell}} \bar{s} + m$, $\theta = H_1(\lambda || \gamma || H_1(\alpha') || g^{s'}/d)$, $\bar{\theta} = H_1(\bar{\lambda} || \bar{\gamma} || H_1(\bar{\alpha}') || g^{\bar{s}'}/d)$.

The rest of the proof is similar to that of the previous Lemma and omitted.

Problem E: This problem can be re-iterated as a collection of *Problem E(i)*, $1 \le i \le 3^N$:

$$\{\text{Find } \pi \text{ from } \mathcal{T}_i : m \leftarrow F_q; \mathcal{T}_i \leftarrow (i, P)\}$$

where P is a honest prover and $\mathcal{T}_i$ is a particular transcript formally specified as follows.

(Transcripts $\mathcal{T}_i$) Let $i = (b_1, \cdots, b_N)$ in ternary notation. Let

$$\{1, \cdots, N\} = \mathbf{A} \cup \mathbf{C} \cup \mathbf{F} = \{a_1, \cdots, a_{N_1}\} \cup \{c_1, \cdots, c_{N_2}\} \cup \{f_1, \cdots, f_{N_3}\}$$

where $N = N_1 + N_2 + N_3$ and $\mathbf{A}$, $\mathbf{C}$ and $\mathbf{F}$ are the sets of indices corresponding to Rounds with $b = 1, 2, 3$, respectively.

$$\begin{aligned}
\mathcal{T}_i \leftarrow P : \{&(E_1, \cdots, E_n) \leftarrow \mathcal{G}(1^k);\\
&d; \theta^{(i)}, 1 \le i \le N; (r_1^{(i)}, \cdots, r_n^{(i)}), i \in \mathbf{A};\\
&\gamma^{(i)}, 1 \le i \le n; \alpha^{(i)}, i \in \mathbf{A}; \beta^{(i)}, i \in \mathbf{A};\\
&\lambda^{(i)}, i \in \mathbf{C} \cup \mathbf{F}; s^{(i)}, i \in \mathbf{C}; \alpha'^{(i)}, i \in \mathbf{F}; s'^{(i)}, i \in \mathbf{F}\}
\end{aligned}$$

(*Problem E*) Solve Problem $E(1)$, $\cdots$, or $E(3^N)$.

Lemma 7 *Suppose a PPT algorithm V, with H_1 and H_2 being random oracles, achieves*

$$\max_{1 \le i \le 3^N} \Pr[V \textit{ solves problem } E(i)] \ge \frac{1}{n} + \epsilon(k)$$

for some non-negligible function ϵ. There exists a PPT algorithm $\mathcal{M}$, which invokes V and simulates the view of V by answering all its H_1-queries and H_2-queries, can solve at least one of the following problems: Discrete Logarithm Problem or inverting an encryptions, with probability at least $(n-1)/n \cdot \epsilon(k)$.

We construct $\mathcal{M}$ as follows.

$\mathcal{M}$ = "On

1. inputs $\{Y^{(i)} \in G : i \in \mathbf{A}\}$, for each $a \in \mathbf{A}$,
   (a) Randomly pick $r_1^{(a)}, \cdots, r_n^{(a)} \in_R \{0,1\}^k$ and $R_1^{(a)}, \cdots, R_n^{(a)} \in_R F_q$. Set the values of $H_2(r_i) = R_i^{(a)}$, for all $1 \le i \le n$.
   (b) Randomly pick $\alpha^{(a)} \in_R \{0,1\}^k$.
   (c) Set $\beta^{(a)} = Y^{(a)}$ and $\gamma^{(a)} = (g^{R_1^{(a)}}, \cdots, g^{R_n^{(a)}})$
   (d) Generate n secure public-key encryption functions $\bar{E}_1^{(a)}, \cdots, \bar{E}_n^{(a)}$ at random (whose decryption functions are generated by, and known to, $\mathcal{M}$).
   (e) Run V on corresponding inputs and reply all the queries of H_1 and H_2 in the following manner.
      – For any H_2-query with input $r_i^{(a)}$, $1 \le i \le n$, $R_i^{(a)}$ is replied.
      – For a H_1-query with input $Z_1^{(a)} || \cdots || Z_n^{(a)}$, compute the inversion $s_i'^{(a)} \leftarrow \bar{E}_i^{-1(a)}(Z_i^{(a)})$, $1 \le i \le n$, and determine if $s_1'^{(a)} = \cdots = s_n'^{(a)}$ and $Y^{(a)} = g^{R_\ell^{(a)} s_1'^{(a)}}$, for some $1 \le \ell \le n$. If they are true, output $R_\ell^{(a)} s_1'^{(a)}$ as the discrete logarithm of $Y^{(a)}$ and halt. Otherwise, randomly pick an element from $\{0,1\}^k \setminus \{\alpha^{(a)}\}$ as the reply.
      – For any other queries of H_1 and H_2, random numbers are generated in the corresponding range of H_1 and H_2 as the replies.
      – For query consistency, for any query with an input value which has been received before, the same reply as the last time is returned.
2. and inputs $(Z_1^{(a)}, \cdots, Z_n^{(a)})$, $a \in \mathbf{C}$", where each $Z_i^{(a)}$ is in the range of a secure public-key encryption function E_i, $1 \le i \le$, for each $a \in \mathbf{C}$, $\mathcal{M}$ sets $\lambda^{(a)} = Z_1^{(a)} || \cdots || Z_n^{(a)}$ and randomly generates $Y_i^{(a)} \in_R G$, $1 \le i \le n$ and $s^{(a)} \leftarrow F_q$, and set $\gamma^{(a)} = (Y_1^{(a)}, \cdots, Y_n^{(a)})$. Then $\mathcal{M}$ runs V on corresponding inputs and reply all the queries of H_1 and H_2 in the similar manner to the above. Besides ensuring randomness in replies and maintaining query consistency, $\mathcal{M}$ evaluates the input value of each H_2-query, denoted by $r^{(a)}$, and determine if $Z_i^{(a)} = E_i(r^{(a)})$, for some i, $1 \le i \le n$. If this is the case, $\mathcal{M}$ outputs $r^{(a)}$ as the plaintext of $Z_i^{(a)}$ and halts;

3. and inputs $(Z_1^{(a)}, \cdots, Z_n^{(a)})$, $a \in \mathbf{F}$", where each $Z_i^{(a)}$ is in the range of a secure public-key encryption function E_i, $1 \leq i \leq$, for each $a \in \mathbf{F}$, $\mathcal{M}$ prepares the appropriate inputs for V and invokes V and answering all the queries of H_1 and H_2 in the similar manner to Step 2 above.
4. Halt with no output if V stops."

If V does not make any queries to H_1 or to H_2 in any of the N rounds, then he can best randomly guess π because $\mathcal{M}$ has not decided on the value of π yet. If V makes any "qualified" query, then $\mathcal{M}$ succeeds.

Remark on Probabilistic Public-key Encryption Functions : In the formulation of $\mathcal{H}$ and $\mathcal{H}_D$ above, the functions are required to check if $E_i(X)$ is computed to Z_i for some i, $1 \leq i \leq n$. In general, this may not be feasible if E_i is probabilistic while the corresponding encryption coin flip sequence as well as E_i^{-1} are unknown. The corresponding coin flip sequence of E_i for yielding Z_i from X is also given. One may also consider the coin flip sequence to be part of the message X. For the case when coin flip sequences are not carried over to places where encryptions are not required, for example, computing $g^{H_2(r_{\phi(i)})}$, we can use the technique of the proof of Lemma 4 instead. Therefore for simplicity, we assume that all the underlying public key encryption functions used in most parts of our proof are deterministic.

Confidentiality We prove that no PPT Adversary has a non-negligible advantage in Game D-C. By Theorem assumptions, $q_C = 0$, $\pi_0 = \pi_1 = \pi$, and with overwhelming probability $m_0 \neq m_1$. Prover Oracle and Decryption Oracle can be simulated as before.

If a PPT adversary V can distinguish m_0 and m_1, then it can do so in one of the following five scenarios:

A. V can distinguish in an individual Round with $b = 1$.
B. V can distinguish in an individual Round with $b = 2$.
C. V can distinguish in an individual Round with $b = 3$.
D. V can distinguish in multiple Rounds all with $b = 3$.
E. V can distinguish based on transcripts from all N Rounds.

By our security model, V knows m_1 and $d = g^{m_1}$.

Problem A: V cannot possibly distinguish from knowing parameters $\theta = H_1(\lambda||\gamma||\alpha||\beta)$ and $r_1, \cdots, r_n$, γ, α, β, because these parameters are all unrelated to m_0 or m_1.

Problem B: V cannot possibly distinguish from knowing parameters $\theta = H_1(\lambda||\gamma||\alpha||\beta)$ and λ, γ, and s, because these parameters are all unrelated to m_0 or m_1.

Problem C: The equality $\theta = H_1(\lambda||\gamma||H_1(\alpha')||\beta')$ always holds by backpatching the random oracle. We further assis V by assuming he knows r_π. Then Problem C reduces to testing the equality $s' \overset{?}{=} H_2(r_\pi)s + m_1$ based on knowing s', r_π, m_1, and $E_i(s)$ for $1 \leq i \leq n$. If V can accomplish this with non-negligible advantage over random guessing, then V can violate the IND-CCA2 security of one of the encryptions.

Problem D: Problem D also reduces to violating the IND-CCA2 security of an encryptions. Details omitted.

Problem E: All rounds with b=1 or 2 do not involve m_0 or m_1. Therefore solving Problem E is reduced to solving Problem D. $\square$

Proof Sketch of Theorem 2. In order to win Game D-C, the Adversary computes m, s', obtains s from a colluder, but still needs to try an exponential number of combinations of $(g^{H_2(r_i^{(1)})}, \cdots, g^{H_2(r_i^{(L)})})$ to distinguish. The complexity of the Prover is polynomial. The complexity of the Decryptor is polynomial. The same argument can be applied to Game D-A as well. $\square$