University of Wollongong
Research Online

Faculty of Informatics - Papers (Archive)

Faculty of Engineering and Information Sciences

2005

Privacy-enhanced Internet storage: Non-interactive publicly verifiable 1-out-of-n encryption schemes

Willy Susilo
University of Wollongong, wsusilo@uow.edu.au

Fangguo Zhang
Sun Yat-Sen University of Medical Sciences

Yi Mu
University of Wollongong, ymu@uow.edu.au

Follow this and additional works at: https://ro.uow.edu.au/infopapers

 Part of the Physical Sciences and Mathematics Commons

Recommended Citation

Susilo, Willy; Zhang, Fangguo; and Mu, Yi: Privacy-enhanced Internet storage: Non-interactive publicly verifiable 1-out-of-n encryption schemes 2005.
https://ro.uow.edu.au/infopapers/2858

Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: research-pubs@uow.edu.au

Privacy-enhanced Internet storage: Non-interactive publicly verifiable 1-out-of-n encryption schemes

Abstract

One of the main important uses of Internet is its ability to connect people through the use of email or Internet storage. However, it is often desirable to limit the use of email or Internet storage due to organization's restriction, avoiding spams, etc. In particular, emails with multimedia contents will take up a lot of spaces. In this paper, we propose cryptographic schemes that can be used to stop unwanted messages to be stored in the Internet server. We refer this technique as *privacy enhancement for Internet storage*, since the Internet server will *not* learn any information directed to its users, other than performing its task to deliver or stop the messages. Firstly, we describe a notion of *non-interactive publicly verifiable 1-out-of-n encryption* by proposing a model together with its security requirements. Then, we extend this notion to a *publicly verifiable ring-to-1-out-of- n encryption*, that provides *sender anonymity*. We note that the previously known interactive versions of the publicly verifiable 1-out-of-n encryption cannot be used to construct publicly verifiable ring-to-1-out-of-n encryption.

Disciplines

Physical Sciences and Mathematics

Publication Details

Susilo, W., Zhang, F. & Mu, Y. (2005). Privacy-Enhanced Internet Storage: Non-Interactive publicly Verifiable 1-out-of-n Encryption Schemes. International Journal of Computer Science and Network Security, 5 (12), 99-114.

This journal article is available at Research Online: https://ro.uow.edu.au/infopapers/2858

Privacy-Enhanced Internet Storage: Non-Interactive Publicly Verifiable 1-out-of-n Encryption Schemes

Willy Susilo[1], Fangguo Zhang[2] and Yi MU[1]
[1]School of IT and Computer Science,
University of Wollongong, Wollongong, NSW 2522, Australia
Email: {wsusilo, ymu}@uow.edu.au
[2] Department of Electronics and Communication Engineering
Sun Yat-Sen University, Guangzhou 510275, P.R. China
Email: isdzhfg@zsu.edu.cn

Abstract

One of the main important uses of Internet is its ability to connect people through the use of email or Internet storage. However, it is often desirable to limit the use of email or Internet storage due to organization's restriction, avoiding spams, etc. In particular, emails with multimedia contents will take up a lot of spaces. In this paper, we propose cryptographic schemes that can be used to stop unwanted messages to be stored in the Internet server. We refer this technique as *privacy enhancement for Internet storage*, since the Internet server will *not* learn any information directed to its users, other than performing its task to deliver or stop the messages. Firstly, we describe a notion of *non-interactive publicly verifiable 1-out-of-n encryption* by proposing a model together with its security requirements. Then, we extend this notion to a *publicly verifiable ring-to-1-out-of-n encryption*, that provides *sender anonymity*. We note that the previously known *interactive* versions of the publicly verifiable 1-out-of-n encryption cannot be used to construct publicly verifiable ring-to-1-out-of-n encryption.

Keywords: Internet storage, non-interactive, publicly verifiable 1-out-of-n encryption, publicly verifiable ring-to-1-out-of n encryption, signature of knowledge

1 Introduction

The public Internet can be considered as a world wide computer network, that is, a network that interconnects millions of computing devices throughout the world. Most of these devices are traditional desktop PCs, Windows/Linux/Unix based workstations, laptops, tablet PCs, handheld devices and so forth. Among them, there are servers that store and transmit information such as web pages and email messages.

One of the most important uses of Internet is its ability to connect people through the use of email or Internet storage. However, it is often desirable to limit the use of email or Internet storage due to organization's restriction, avoiding spams, etc.

Consider a situation where there is an Internet storage that can be used to store messages directed to a group of users, Γ. In this situation, it would be desirable to install a secure gateway $\mathcal{G}$ to stop all messages that are not directed to its group members in Γ. $\mathcal{G}$ will act as a mediator between a sender and a receiver of a message in group Γ. When Alice wants to send a public-key encrypted message to Bob, who is a member of Γ, then $\mathcal{G}$ must be able to check that the message is directed to a group member in Γ and store the message in the Internet storage. However, the problem is $\mathcal{G}$ does not hold Bob's secret key. This problem has been considered in [7] to create an anonymous ad hoc group. In this scenario, by knowing only the public information of the group members in Γ, $\mathcal{G}$ must determine if the encrypted incoming data is for a group member in Γ without being able to identify the actual recipient. Moreover, the other member in Γ, together with $\mathcal{G}$ itself, must not be able to read the message directed to Bob. In [7] (or subsequently revised in [8]), this problem has been considered as publicly verifiable 1-out-of-n encryption (PV1nE) scheme, and they proposed an interactive protocol for PV1nE in [7].

In a different scenario, Alice does not want her identity to be revealed. Instead, she would like to send a message to Bob on behalf of a group, which can be verified. Suppose Alice is a worker in an insurance company who would like to send a message to Bob, then the message is considered to *belong* to the company instead of being sent by Alice. The situation becomes more complex than the original scenario mentioned earlier, since $\mathcal{G}$ cannot perform an interactive protocol with Alice, and hence, the notion of *non-interactive* PV1nE is essential and required. We note that this scenario is essential, especially if we would like to protect Alice's privacy. Without having to assume that an anonymous routing (eg MIX-nets) exists (as in [9]), then the existence of non-interactive PV1nE is essential.

In a non-interactive PV1nE, a `Prover` (or `Sender`, respectively) wishes to send a public-key encrypted message to a `Receiver` through a `Verifier`. The `Prover` arbitrarily forms a group Γ that consists of the `Receiver` together with other people that belong to the same group as the `Receiver`. Then, the `Prover` conducts a special public-key encryption for the group of receivers in such a way that the public verifier can be sure that the message can be decrypted by one of the receivers in the group. It is also required that 1) the `Verifier` cannot read the message; 2) the `Verifier` cannot identify to whom the message is designated to; and 3) the `Verifier` does not need to perform an interactive protocol with the `Prover` to check the validity of the message.

Our Contribution

The scheme used in [7] is based on the *cut-and-choose methodology* [11] and hence, interactions between sender and verifier are required. In this paper, we firstly provide a notion of a Non-Interactive Publicly Verifiable 1-out-of-n Encryption Scheme

(PV1nE), and propose a non-interactive scheme *without* employing a cut-and-choose technique that satisfies our model. Then, we extend this notion to create a Publicly Verifiable Ring-to-1-out-of-n Encryption (PVRTE) Scheme. We shall point out that the interactive version of PV1nE *cannot* be used to generate a PVRTE scheme. We also provide a generic construction of PVRTE schemes from any PV1nE schemes. We provide a complete security proof for our schemes.

The rest of this paper is organized as follows. In the next section, we will review some related work in this area. In Section 3, we will provide some cryptographic tools that will be used throughout this paper. We review the signature of knowledge that was proposed in [4] in this section. In section 4, we extend the notion of signature of knowledge to construct new signatures of knowledge, that will be used to construct our schemes. In Section 5, we will present a security model for non-interactive PV1nE scheme, together with a concrete scheme that satisfies this model. In Section VI, we extend the notion of non-interactive PV1nE scheme to PVRTE schemes. Section 7 concludes the paper.

2 Related Work

The concept of verifiable encryption was introduced by Stadler in [13]. This concept was introduced to construct a verifiable secret sharing scheme, where the partial shares of the secret sharing scheme can be verified by anyone publicly. Verifiable encryption can be achieved by using cut-and-choose methodology as demonstrated in [2]. Camenisch et. al. extended the notion of verifiable encryption to verifiable threshold encryption [5] such that a verifier can make sure that a minimum number of t arbitrary receivers are required to work together to recover a message.

The notion of verifiable encryption scheme has been extended to 1-out-of-n verifiable encryption scheme (PV1nE) in [7]. In a PV1nE scheme, a sender can arbitrarily form a group of n receivers, and prepare an encrypted message that can be recovered by at least one targeted group member, in such a way that a public verifier can be sure that the encrypted message can be recovered by at least one of the group member in the targeted group, but the verifier does not know anything about the identity of the targeted receiver. The scheme that was presented in [7] is an interactive scheme is based on the cut-and-choose methodology [11]. During the public verification, the verifier will contact the prover to verify the correctness of the message *interactively*. Hence, the verifier must know who the prover (or the sender, resp.) is.

In [12], the definition of *ring signatures* was formalized and an efficient scheme based on RSA was proposed. A ring signature scheme is based on trapdoor one-way permutations and an ideal block cipher that is regarded as a perfectly random permutation. A ring signature scheme allows a signer who knows at least one secret information (or trapdoor information) to produce a sequence of n random permutations and form them into a ring. This signature can be used to convince any third party that one of the people in the group (who knows the trapdoor information) has

authenticated the message on behalf of the group. The authentication provides *signer ambiguity*, in the sense that no one can identify who has actually signed the message.

In [1], a method to construct a ring signature from different types of public keys, such as these for integer factoring based schemes and discrete log based schemes, was proposed. The proposed scheme is more efficient than [12]. The formal security definition of a ring signature is also given in [1].

2.1 Notations

Throughout this paper, we will use the following notations. The ring of integers modulo a number p is denoted by Z_p, and the multiplicative subgroup of integers relatively prime to p, by Z_p^*. Let $||$ denote a binary operator that concatenates two bit strings as inputs. The inputs will be converted to its binary representation where its length is determined by a security parameter ℓ.

3 Cryptographic Tools

In this section, we will review some cryptographic tools, together with proposing some new cryptographic primitives, that will be used throughout this paper.

Let q be a large prime and $p = 2q+1$ be also a prime. Let G be a finite cyclic group of prime order p. Let $g, h \in Z_p^*$ be two elements of order q. Let $\hat{g}$ be a generator of G such that computing discrete logarithms of any group element (apart from the identity element) with respect to one of the generators is infeasible. Let $H : \{0,1\}^* \rightarrow \{0,1\}^\ell$ denote a strong collision-resistant hash function.

3.1 Signature of Knowledge of Representation

The first signature of knowledge (SPK) was proposed in [4, 3]. We will use the following definition of signature of knowledge from [4].

Definition 1 *[4] A pair* $(c, s) \in \{0,1\}^\ell \times Z_q$ *satisfying*

$$c = H(S||V||m) \text{ with } S = g||y \text{ and } V = g^s y^c \pmod p$$

is a signature knowledge of the discrete logarithm of a group element y *to the base* g *of the message* $m \in \{0,1\}^*$ *and is denoted*

$$SPKLOG\{\alpha : y = g^\alpha\}(m).$$

An $SPKLOG\{\alpha : y = g^\alpha \bmod p\}(m)$ can be computed if the value (secret key) $\alpha = \log_g(y)$ is known, by selecting a random integer $r \in Z_q$ and computing $t = g^r \pmod p$ and then c and s according to

$$c = H(g||y||t||m)$$

and

$$s = r - c\alpha \pmod{q}.$$

This is also known as a non-interactive proof of the knowledge α.

3.2 Ring Signature Schemes

We adopt the notations proposed in [1] to define ring signature schemes. We note that the ring signature schemes are referred to 1-out-of-n in [1].

Definition 2 *[1] A ring signature scheme consists of three polynomial time algorithms*

- $(s_k, p_k) \leftarrow \mathcal{G}(1^\kappa)$*: A probabilistic algorithm that takes security parameter* κ *and outputs private key* s_k *and public key* p_k*.*
- $\sigma \leftarrow \mathcal{S}(m, s_k, L)$*: A probabilistic algorithm that takes a message* m*, a list* L *that contains public keys including the one that corresponds to* s_k *and outputs a signature* σ*.*
- $\{\texttt{True}$ *or* $\perp\} \leftarrow \mathcal{V}(m, \sigma, L)$*: A deterministic algorithm that takes a message* m *and a signature* σ*, and outputs either* True *or* $\perp$ *meaning* accept *or* reject*, respectively. It is required to have* $\texttt{True} \leftarrow \mathcal{V}(m, \mathcal{S}(m, s_k, L), L)$ *with an overwhelming probability.*

A ring that allows a mixture of factorization and discrete log based public keys has been constructed in [1].

4 A New Signature of Knowledge for Proving Equality of Discrete Logarithm and Double Discrete Logarithm

In this section, we extend the notion of signature of knowledge mentioned in the previous section to signature of knowledge for proving equality of discrete logarithm and double discrete logarithm.

Definition 3 *A signature of knowledge on* $m \in \{0,1\}^*$*, denoted by*

$$SPK2LOG\{\alpha : y = g^\alpha \wedge z = \hat{g}^{(h^\alpha)}\}(m),$$

is a signature of knowledge on equality proof that the knowledge of the discrete logarithm of y *to the base* $g \in Z_p^*$ *equals the double discrete logarithm of* z *to the base* $\hat{g} \in G$ *and* $h \in Z_p^*$*.*

This signature of knowledge is represented by $(c, s_1, \cdots, s_\ell, \hat{s}) \in \{0,1\}^k \times Z^\ell$, where $\ell \leq k$ be a security parameter, satisfying equation

$$c = H(m||y||g||\hat{g}||g^{\hat{s}}y^c||h||t_1|| \cdots ||t_\ell)$$

with

$$t_i = \begin{cases} \hat{g}^{(h^{s_i})} & if c[i] = 0. \\ z^{(h^{s_i})} & \text{otherwise.} \end{cases}$$

Note that $z = \hat{g}^{(h^\alpha)}$.
To compute the above signature of knowledge, three conditions must hold.

- The value of $x = \log_g(y)$ is known.
- The value of $x = \log_g(\log_h(z))$ is known.
- $\log_g(y) = \log_g(\log_h(z))$ holds.

We assume that there is an upper bound λ on the length of x, i.e. $0 \leq x < 2^\lambda$. The signature of knowledge is generated as follows.
Firstly, compute the values of

$$t_i^* = \hat{g}^{(h^{r_i})}$$

for $i = 1, \cdots, \ell$ with randomly chosen $r_i \in \{0, \cdots 2^{\epsilon\lambda} - 1\}$, where $\epsilon > 1$ be a constant. Then, select a random $r \in Z_q$ and compute c as

$$c = H(m||y||g||\hat{g}||g^r||h||t_1|| \cdots ||t_\ell)$$

Finally, we set the following values

$$\begin{cases} s_i \leftarrow r_i & \text{if } c[i] = 0, \\ s_i \leftarrow r_i - x & \text{otherwise.} \end{cases}$$

and compute

$$\hat{s} = r - cx \pmod{q}.$$

One can verify that the resulting tuple $(c, s_1, \cdots, s_\ell, \hat{s})$ satisfies the verification equation.

Based on the above signature of knowledge, we further extend it to $SPK2LOG(1,n)\{\alpha : y = g^\alpha \wedge z = \hat{g}^{h_1^\alpha} \vee \hat{g}^{h_2^\alpha} \ldots \vee \hat{g}^{h_n^\alpha}\}(m)$ defined as follows.

Definition 4 *A signature of knowledge on $m \in \{0,1\}^*$, denoted by*

$$SPK2LOG(1,n)\{\alpha : y = g^\alpha \wedge z = \hat{g}^{h_1^\alpha} \vee \hat{g}^{h_2^\alpha} \ldots \vee \hat{g}^{h_n^\alpha}\}(m),$$

is a 1-out-of-n knowledge equality proof that the knowledge of the discrete logarithm of y to the base $g \in Z_p^*$ equals the double discrete logarithm of z to the base $\hat{g} \in G$ and one of $h_1, h_2, \ldots, h_n \in Z_p^*$.

This signature of knowledge is represented by $(\hat{s}, c_1, c_2, \cdots, c_n, s_1, s_2, \cdots, s_n)$, satisfying equation

$$\begin{aligned}\sum_{i=1}^{n} c_i &= H(m||y||g||\hat{g}||g^{\hat{s}} y^{\sum_{i=1}^{n} c_i}||h_1||\cdots||h_n||t_{11}||\cdots \\ &\quad ||t_{1\ell}||\cdots||t_{1\ell}||\cdots||t_{n\ell})\end{aligned}$$

where

$$t_{ji} = \begin{cases} \hat{g}^{(h_j^{s_{ji}})} & \text{if } c_j[i] = 0, \\ z^{(h_j^{s_{ji}})} & \text{otherwise.} \end{cases}$$

for $j = 1, \cdots, n$.
The signature of knowledge can only be computed if one of the valid α is known, where $\alpha = \log_{\hat{g}}(\log_{h_i} z)$ and $\alpha = \log_g(y)$. It can be computed as follows. Without losing generality, we assume that the prover knows α where $\alpha = \log_{\hat{g}}(\log_{h_1} z)$ and $\alpha = \log_g(y)$.

1. Firstly, select $n-1$ random numbers, $c_2, \cdots, c_n \in Z_q$.
2. Then, select ℓn random numbers, $t_{11}, \cdots, t_{1\ell}, \cdots, t_{n1}, \cdots, t_{n\ell} \in \{0, \cdots, 2^{\epsilon\lambda} - 1\}$.
3. Select a random number $r \in Z_q$.
4. For $j = 2, \cdots, n$, compute

$$t_{ji} = \begin{cases} \hat{g}^{h_j^{r_{ji}}} & \text{if } c_j[i] = 0, \\ z^{h_j^{r_{ji}}} & \text{otherwise.} \end{cases}$$

   for all $i = 1, \cdots, \ell$.
5. Compute

$$t_{1i} = \hat{g}^{(h^{r_{1i}})}$$

6. Compute

$$\begin{aligned} c_1 &= H(m||y||g||\hat{g}||g^r||h_1||\cdots||h_n|| \\ &\quad t_{11}||\cdots||t_{1\ell}||\cdots||t_{1\ell}||\cdots||t_{n\ell}) \\ &\quad - \sum_{i=2}^{n} c_i \pmod q \end{aligned}$$

7. Let

$$s_{1i} = \begin{cases} r_{1i} & \text{if } c_1[i] = 0, \\ r_{1i} - x_1 & \text{otherwise.} \end{cases}$$

8. Compute

$$\hat{s} = r - x_1 \sum_{i=1}^{n} c_i \pmod{q}$$

One can verify that the resulting tuple $(\hat{s}, c_1, c_2, \cdots, c_n, s_1, s_2, \cdots, s_n)$ satisfies the verification equation.

5 Non-Interactive Publicly Verifiable 1-out-of-n Encryption Scheme (PV1nE)

5.1 Model

A PV1nE scheme involves three entities, namely a `Prover` P (or `Sender`, respectively), a `Verifier` V and a `Receiver` R. There are three algorithms involved, namely a probabilistic algorithm: `Verifiable Encryption` (VE), a deterministic algorithm: `Verification` (Ver) and a deterministic algorithm: `Decryption` (Dec).

P accepts as inputs a security parameter k, a message $m \in \{0,1\}^*$, n public-key encryptions $\{E_i\}_{1 \le i \le n}$. By invoking `Verifiable Encryption` (VE) algorithm, it outputs a valid ciphertext $\mathcal{C}$ that can only be deciphered by one of the secret keys D_i associated with E_i, $1 \le i \le n$.

V accepts a ciphertext $\mathcal{C}$ together with all public keys $\{E_i\}_{1 \le i \le n}$. By invoking the `Verification` (Ver) algorithm, it outputs $\{\texttt{True}, \perp\}$. The output is `True` if $\mathcal{C}$ is *valid*, which means that it will be able to be decrypted by one of the secret keys D_i, $1 \le i \le n$, otherwise, it outputs $\perp$. When the output is $\perp$, then $\mathcal{C}$ is discarded (since it is tagged as 'invalid').

R accepts a ciphertext $\mathcal{C}$ and obtains the plaintext m by invoking the `Decryption` (Dec) algorithm.

Security Requirements

1. Probability of a prover P to produce an invalid ciphertext $\mathcal{C}$ that will pass the verification test is negligible. We require

$$\begin{aligned} \Pr \quad & \{Ver(\mathcal{C}, \{E_i\}_{1 \le i \le n}, \forall i) = \texttt{True} | \\ & \mathcal{C} \leftarrow VE(k, msg, E_j \notin \{E_i\}_{1 \le i \le n})\} \\ & = \epsilon. \end{aligned}$$

2. V will not have any knowledge about the plaintext msg after the verification test.

3. *Targeted Decipherability:*
   If both P and V are honest, at the execution of the `Decryption` algorithm, the

plaintext msg can always be obtained. We require

$$\begin{aligned} \Pr \quad & \{msg \text{ is } valid | \\ & C \leftarrow VE(k, msg, E_j \in \{E_i\}_{1 \leq i \leq n}), \\ & msg \leftarrow Decrypt(C, D_j)\} \\ & = 1. \end{aligned}$$

4. *Anonymity:*
   Having observed several C's, V cannot observe to whom a ciphertext is directed to.

5.2 Security Notions

In terms of security of PV1nE scheme, we need to consider two types of attackers, namely *outsider* and *insider* attacks. We call an attack to be an *insider* attack, if the attack is launched by an adversary who either compromises one of the player in the system, namely a receiver R_i, $i = 1, \cdots, n$, a sender or a gateway. We will describe the attacks launched by these players in more detail later. An outsider attack is an attack that is performed by an "outsider", who is not one of the player in the system. Formally, we define these attacks as follows.

Outsider Attacks

Let $\mathcal{A}$ be an outside attacker, whose running time is bounded by t, that is polynomial in a security parameter k. We require that

$$\begin{aligned} \mathtt{Pr}_{\mathcal{A}(t,k)}[m \quad | \quad & C \leftarrow VE(k, msg, E_j \notin \{E_i\}_{1 \leq i \leq n}), \\ & \mathtt{True} \leftarrow Ver(C, \mathcal{PK})] \leq \epsilon \end{aligned}$$

Insider Attacks

In the following, we define three different attacks launched by insiders. We relate these attacks with the security notions for the participants in the system.

Security for the Sender

Informally, the security for the sender is defined as follows. We require that if a message is encrypted and directed for user R_i, $1 \leq i \leq n$, then any other receiver R_j, $j \neq i$, will not be able to read the encrypted message. We require that

$$\begin{aligned} \mathtt{Pr}_{\mathcal{A}(t,k)}[m \quad | \quad & x_j, C \leftarrow VE(k, msg, E_i, i \neq j), \\ & \mathtt{True} \leftarrow Ver(C, \mathcal{PK})] \leq \epsilon \end{aligned}$$

Security for the Verifier

The main role of the verifier is to make sure that the encrypted message is directed to one of the receivers in the group. Hence, informally, the security for the verifier

is defined as follows. Consider an attacker $\mathcal{A}$ who would like to send an encrypted message to a person, R_z, $z \notin \{1, \cdots n\}$. His intention is to make the verifier believes that this message is intended to one of the receiver R_i, $1 \leq i \leq n$. We say this attack is successful, if the receiver believes that the message is directed to one of the R_i, $1 \leq i \leq n$. Intuitively, this attack is explained as follows. The attacker wants to "flood" the server with junk messages, so that at some stage, the server will collapse since the messages will be stored forever in the server, but no receiver will retrieve it. The success probability of this attack is bounded by

$$\begin{aligned} \mathtt{Succ}_{\mathcal{A}}(k) &= \Pr\{Ver(\mathcal{C}, \{E_i\}_{1\leq i\leq n}, \forall i) = \mathtt{True} | \\ &\quad \mathcal{C} \leftarrow VE(k, msg, E_j \notin \{E_i\}_{1\leq i\leq n}) \}. \end{aligned}$$

We note that this security notion is related to the first security requirement mentioned earlier. We also note that this attack is often referred to *Denial of Service* attack.

Security for the Receiver
In this attack, the attacker $\mathcal{A}$ controls the verifier and tries to *sabotage* encrypted message directed to a receiver R_j, $1 \leq j \leq n$. Intuitively, $\mathcal{A}$ would like to *reject* valid encrypted messages directed to R_j, but he will accept all other valid encrypted messages directed to different receivers. We formally define this security notion as "IND-PV1nE-SCCA" (SCCA = "Signer Chosen Ciphertext Attack") as follows.

Definition 5 (IND-PV1nE-SCCA). *Let $\mathcal{A}$ be an attacker whose running time is bounded by t, that is polynomial in a security parameter k. $\mathcal{A}$ controls the view of the verifier. We consider the following game:*

S1: *The* Setup *algorithm is run. A public parameters, p, q are generated. A set of private keys of the receivers x_i are generated, and the public keys $h_i = g^{x_i} \pmod p$, for $1 \leq i \leq n$ are published.*

S2: *$\mathcal{A}$ can query the encryption and decryption oracle for a message/ciphertext of his choice.*

S3: *$\mathcal{A}$ outputs a target message m^* and two receivers R_1, R_2, and sends it to the encryption oracle. The encryption oracle chooses $\beta \in \{1, 2\}$ uniformly at random and creates a target ciphertext C_i^* that is an encrypted version of m^* directed to the receiver R_β and returned it to $\mathcal{A}$.*

S4: *$\mathcal{A}$ can issue some other ciphertexts C_i and message m_i and ask the decryption oracle to decrypt the message. The restriction here is $m_i \neq m^*$.*

S5: *$\mathcal{A}$ outputs its guess $\beta' \in \{1, 2\}$.*

We define the attacker $\mathcal{A}$'s success by the probability $\mathtt{Succ}_{\mathcal{A}}^{\mathtt{IND-PV1nE-SCCA}}(\mathtt{k}) = \Pr[\beta = \beta']$. $PV1nE$ scheme is said to be IND-PV1nE-SCCA secure if $\mathtt{Succ}_{\mathcal{A}}^{\mathtt{IND-PV1nE-SCCA}}(\mathtt{k})$ is negligible in k.

5.3 The Scheme

In this section, we propose non-interactive PV1nE schemes based on the building blocks developed in the previous section. The scheme is as follows.

- `Setup`:

  Let q be a large prime and $p = 2q + 1$. Let G be a cyclic group of order p and g be an element of Z_p^* with order q. Each receiver R_i, $1 \leq i \leq n$, selects a random number $x_i \in Z_q$ and sets his public key to $h_i = g^{x_i} \pmod{p}$.

- `Verifiable Encryption` (VE):
  To encrypt a message $m \in \{0,1\}^*$ under the public key $h_i \in Z_p^*$ (that is one of $h_1, h_2, \ldots, h_n$), the `Prover` does the following:

  1. Selects a random element $\hat{g}$ from G;
  2. Computes $v = \hat{g}^m \in G$, $A = g^\alpha \bmod p$, $B = m^{-1} h_i^\alpha \bmod p$.
  3. Sets the ciphertext to be:

$$\begin{aligned} C \;=\; & [h_1, \ldots, h_n, \hat{g}, v, A, B, \\ & SPK2LOG(1,n)\{\alpha : A = g^\alpha \wedge \\ & v^B = \hat{g}^{h_1^\alpha} \vee \hat{g}^{h_2^\alpha} \ldots \vee \hat{g}^{h_n^\alpha}\}(v)]. \end{aligned}$$

- `Verification` (Ver):
  To verify a ciphertext, the `Verifier` tests whether

$$\begin{aligned} SPK2LOG(1,n)\{\alpha \;:\; & A = g^\alpha \wedge \\ & v^B = \hat{g}^{h_1^\alpha} \vee \hat{g}^{h_2^\alpha} \ldots \\ & \vee \hat{g}^{h_n^\alpha}\}(v) \end{aligned}$$

  is correct. If it is correct, then it outputs `True`. Otherwise, it outputs $\perp$.

- `Decryption` (Dec):
  A receiver R_i, $1 \leq i \leq n$, who holds the correct secret key D_i associated with E_i can decrypt the message by computing

$$m = A^{x_i}/B$$

  Then, R_i needs to verify whether $v \stackrel{?}{=} \hat{g}^m$ holds. If it holds with equality, then R_i obtains the plaintext m.

5.4 Security Analysis

Theorem 1 (Security against Outsider Attacks).
If there exists a polynomial time algorithm to decrypt a ciphertext without any knowledge of a secret key x_i, $1 \leq i \leq n$, then Decisional Diffie-Hellman problem can be solved in polynomial time.

Theorem 2 (Security for the Sender). *The probability of a polynomially bounded attacker $\mathcal{A}$ to decrypt a ciphertext directed to the receiver R_i, given a valid secret key of x_j, $j \neq i$, is negligible.*

Theorem 3 (Security for the Verifier).
There is no polynomially bounded attacker $\mathcal{A}$ who has $\mathrm{Succ}_{\mathcal{A}}(\mathtt{k}) \geq \epsilon$, for a polynomial time t, and can break the security for the verifier in the PV1nE scheme.

Theorem 4 (Security for the Receiver).
Our scheme is secure in the sense of IND-PV1nE-SCCA.

6 An Extension: The Sender from A Ring

Since our scheme is non-interactive, it allows the sender to be one of users in a ring [12], and hence, the sender's identity can be protected.

Model
As motivated in Section 1, the PVRTE scheme can be used to provide sender's (or prover's) privacy. In PVRTE schemes, the identity of the prover is ambiguous. The prover can send an encrypted message on behalf of the group. To make the model clearer, we assume there are n_p eligible provers (or senders, resp.), denoted as P_i, $i = 1, \cdots, n_p$. The collection of provers is denoted as $\{P_i\}_{1 \leq i \leq n_p}$. There are n receivers in the group, denoted as R_i, $i = 1, \cdots, n$. The collection of receivers is denoted as $\{R_j\}_{1 \leq j \leq n}$.

Any P_i can send a message on behalf of the group $\{P_i\}_{1 \leq i \leq n_p}$. This encrypted message will then be verified by the verifier V.

V does not know about the identity of the prover P_i, but V can be assured that the encrypted message was send by $P_i \in \{P_j\}_{1 \leq j \leq n_p}$ and this encrypted message is intended to a receiver $R_i \in \{R_i\}_{1 \leq i \leq n}$. If V is assured with this fact, then the encrypted message is stored until it is retrieved by the designated verifier R_i. Finally, $R_i \in \{R_j\}_{1 \leq j \leq n}$ can retrieve the stored encrypted message by using his secret key.

Remarks:

1. We require that the verifier V does not know about the identity of the prover P_i, but V is assured that the message was sent by one of the $P_i \in \{P_j\}_{1 \leq j \leq n_p}$. Additionally, we also require that the verifier V does not know the designated receiver of the message, but V is assured that the encrypted message can be decrypted by one of the $R_i \in \{R_j\}_{1 \leq j \leq n}$.

2. Due to the requirement that V does not know about the identity of the prover P_i, then using an interactive protocol between V and P_i is not doable. The verification must only be done by V non-interactively, i.e. without the involvement of any P_i.

The Generic Construction for PVRTE Schemes

In this section, we provide a generic construction for PVRTE schemes from PV1nE schemes and ring signature schemes. For the ring signature schemes, we use the notation as mentioned in Section 2. We will also incorporate the following notations:

- A PV1nE scheme consists of three main algorithms, namely `Verifiable Encryption` (VE), `Verification` (Ver) and `Decryption` (Dec).
- A PVRTE scheme consists of three main algorithms, namely `Verifiable Ring Encryption` (VRE) and `Ring Verification` (RV).
- Let $\mathcal{SK}_{U_i}$ denote U_i's secret key, and $\mathcal{PK}_{U_i}$ denote U_i's public key.

The generic construction is as follows.

- $\text{VRE}(m, \{\mathcal{PK}_{\{P_j\}_{1\le j\le n_p}}\}, \mathcal{SK}_{P_i}, \mathcal{PK}_{\{R_i\}_{1\le i\le n}}) \stackrel{\text{def}}{=} \begin{cases} \gamma \leftarrow VE(m, \{\mathcal{PK}_{\{R_i\}_{1\le i\le n_p}}\}, \mathcal{PK}_V) \\ \eta \leftarrow \mathcal{S}(\gamma, \mathcal{SK}_{P_i}, \mathcal{PK}_{\{P_j\}_{1\le j\le n_p}}) \\ \textit{Output} : (\gamma, \eta) \end{cases}$

  The Verifiable Ring Signature on m is the double (γ, η).

- $RV(\gamma, \eta) \stackrel{\text{def}}{=} \begin{cases} \texttt{Test1} \leftarrow \mathcal{V}(\gamma, \eta, \mathcal{PK}_{\{P_j\}_{1\le j\le n}}) \\ \texttt{Test2} \leftarrow Ver(\gamma, \mathcal{PK}_{\{R_i\}_{1\le i\le n_p}}) \\ \textit{return}(\texttt{Test1 } \textit{and } \texttt{Test2}) \end{cases}$

  The result of this verification is either `True` or $\perp$.

- If it returns `True` in the preceding step, compute $Dec(\gamma)$.

We note that similar result can be obtained by employing a ring signcryption scheme. Since the main contribution of this paper is to demonstrate how to extend our PV1nE to PVRTE, then we will omit the detail of the ring signcryption scheme in this paper.

7 Conclusion

We presented a new scheme: *non-interactive publicly verifiable 1-out-of-n encryption*. Our scheme is based on the non-interactive signature based on proof of knowledge on equality of discrete logarithms and double discrete logarithms that was proposed in this paper. Our scheme can be easily extended to provide *sender anonymity*, that cannot be obtained using interactive schemes developed in the previous work. We showed how to achieve it by combining a ring signature scheme with our non-interactive publicly verifiable 1-out-of-n encryption scheme.

References

[1] M. Abe, M. Ohkubo, and K. Suzuki. 1-out-of-n Signatures from a Variety of Keys. *Advances in Cryptology - Asiacrypt 2002, Lecture Notes in Computer Science 2501*, pages 415 – 432, Springer-Verlag, Berlin, 2002.

[2] N. Asokan, V. Shoup, and M. Waidner. Optimistic fair exchange of digital signatures. *Advances in Cryptology-Eurocrypt 1998, Lecture Notes in Computer Science 1403*, pages 591 – 606, Springer-Verlag, Berlin, 1998.

[3] J. Camenisch. Efficient and generalized group signatures. *Advances in Cryptology - Eurocrypt '97, Lecture Notes in Computer Science 1233*, pages 465–479, Springer-Verlag, Berlin, 1997.

[4] J. Camenisch. Group signature schemes and payment systems based on the discrete logarithm problem. *PhD thesis, ETH Zürich*, 1998.

[5] J. Camenisch and I. B. Damgård. Verifiable encryption, group encryption, and their applications to separable group signatures and signature sharing schemes. *Asiacrypt 2000, Lecture Notes in Computer Science 1976*, pages 331–345, Springer-Verlag, Berlin, 2000.

[6] C. Dwork, M. Naor, and A. Sahai. Concurrent Zero-Knowledge. *Proc. 30th ACM Symposium on the Theory of Computing*, pages 409 – 418, 1998.

[7] J. K. Liu, V. K. Wei, and D. S. Wong. Verifiable encryption in anonymous ad hoc groups. *Cryptology ePrint Archive, Report 2004/028*, 2004.

[8] J. K. Liu, V. K. Wei, and D. S. Wong. Custodian-Hiding Verifiable Encryption. *Workshop on Information Security Applications (WISA 2004)*, 2004 (to appear).

[9] M. Naor. Deniable Ring Authentication. *Advances in Cryptology - Crypto 2002, Lecture Notes in Computer Science 2442*, pages 481 – 498, Springer-Verlag, Berlin, 2002.

[10] D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, 2000.

[11] M. O. Rabin. Digitalized signatures. *Foundations of Secure Computations, Academic Press*, 1978.

[12] R. L. Rivest, A. Shamir, and Y. Tauman. How to Leak a Secret. *Advances in Cryptology - Asiacrypt 2001, Lecture Notes in Computer Science 2248*, pages 552 – 565, Springer-Verlag, Berlin, 2001.

[13] M. Stadler. Publicly verifiable secret sharing. *Advances in Cryptology - Eurocrypt '96, Lecture Notes in Computer Science 1070*, pages 190–199, Springer-Verlag, Berlin, 1996.

Proof of Theorem 1.
Firstly, let us recall decisional Diffie-Hellman (DDH) Problem. Let G be a group of large prime of order q. Consider the following two distributions: the distribution R of random quadruple $(g, g_1, g_2, u_1) \in G^4$ and the distribution D of quadruples $(g, g_1, g_2, u_1) \in G^4$, where $g_1 = g^a$, $g_2 = g^b$ and $u_1 = g^{ab}$. An algorithm that solves the Diffie-Hellman decisional problem is a statistical test that can effectively distinguish these two distributions. That is, given a quadruple from one of the two distributions, it should output 0 or 1, and there should be a non-negligible difference between the probability that it outputs 1 given an input from R and the probability that it outputs 1 given an input from D. The Diffie-Hellman decisional problem is hard if there is no such polynomial statistical test. The Decisional Diffie-Hellman Assumption states that the Diffie-Hellman decisional problem is hard on groups of finite fields.

Now, we assume there exists a polynomial algorithm $\mathcal{A}$ that can decrypt a ciphertext generated by our scheme without any knowledge of the secret key x_i, $1 \leq i \leq n$. We will show that we can use this algorithm to solve an instance of Diffie-Hellman decisional problem in a polynomial time. To be more precise, the algorithm $\mathcal{A}$ accepts a ciphertext C and outputs m with a non-negligible probability. We construct an algorithm $\mathcal{B}$ that will use $\mathcal{A}$ to distinguish whether (g, g_1, g_2, u_1) is from D or D. The algorithm $\mathcal{B}$ is as follows.

1. Let $v = g_1$, $A = g_2$ and $B = u_1$.
2. Run algorithm $\mathcal{A}$ given a ciphertext constructed from (v, A, B).
3. Obtain the plaintext m' from $\mathcal{A}$.
4. Verify whether $g_1 \stackrel{?}{=} \hat{g}^{m'}$ holds. If this equation holds, then the given problem is from D. Otherwise, it is from R.

We note that our algorithm $\mathcal{B}$'s success probability will be the same as algorithm $\mathcal{A}$'s success probability. Hence, we obtain the contradiction.

Proof of Theorem 2 (sketch).
Part of the ciphertext B is composed from $m^{-1}h_i^{\alpha}$, for $h_i = g^{x_i} \pmod{p}$. This is a variant of ElGamal encryption scheme, that is secure in IND-CCA sense. If ElGamal is secure, then our scheme is secure in the same assumption. Hence, the probability of constructing an attacker who can decrypt a valid ciphertext without knowing the secret key is negligible. As in the proof of ElGamal scheme, if there exists an attacker $\mathcal{A}$ who can decrypt the ciphertext without knowing the correct secret key, we can construct a simulator $\mathcal{B}$ that can solve DDH problem in polynomial time. We omit the complete description of the proof in this paper since it is straightforward.

Proof of Theorem 3 (sketch).
An attacker $\mathcal{A}$ who can break the security for the verifier, can generate a valid sig-

nature of knowledge without knowing the correct α. This means this attacker can break the underlying signature of knowledge, that is believed to be hard.