Hindawi
Journal of Computer Networks and Communications
Volume 2017, Article ID 6575907, 15 pages
https://doi.org/10.1155/2017/6575907

Research Article

Symmetric Blind Decryption with Perfect Secrecy

Juha Partala

Physiological Signal Analysis Team, The Center for Machine Vision and Signal Analysis, University of Oulu, Oulu, Finland

Correspondence should be addressed to Juha Partala; juha.partala@oulu.fi

Received 12 April 2017; Accepted 2 July 2017; Published 16 August 2017

Academic Editor: Ziba Eslami

A blind decryption scheme enables a user to query decryptions from a decryption server without revealing information about the plain-text message. Such schemes are useful, for example, for the implementation of privacy-preserving encrypted file storages and payment systems. In terms of functionality, blind decryption is close to oblivious transfer. For noiseless channels, information-theoretically secure oblivious transfer is impossible. However, in this paper, we show that this is not the case for blind decryption. We formulate a definition of perfect secrecy of symmetric blind decryption for the following setting: at most one of the scheme participants is a passive adversary (honest-but-curious). We also devise a symmetric blind decryption scheme based on modular arithmetic on a ring $\mathbb{Z}_{p^2}$, where p is a prime, and show that it satisfies our notion of perfect secrecy.

1. Introduction

Over the past 15 years, data has moved from local storage to centralized data warehouses in the cloud. The accessibility of large amounts of personal data through a public network has given rise to many security and privacy issues [1]. Fortunately, such issues have generally been taken seriously. For example, in many countries, ethical and legal requirements have been imposed on guaranteeing the confidentiality of medical records [2, 3]. However, the implementation of privacy technologies is nontrivial, especially if the data storage has been outsourced to a cloud operator. Sensitive information can often be inferred from simple access patterns either by outsiders or by the operator of the storage. For example, being able to observe a medical doctor to access the medical record of a patient can leak sensitive information. Therefore, such access patterns should be kept hidden both from outsiders and from the party that is administering the records. Oblivious databases [4] and privacy-preserving encrypted file systems [5] are examples of technologies that can be used to hide the access information from the administrator. For such systems, the decryption of data is typically handled by a central decryption server. Such systems can be conveniently implemented using *blind decryption schemes* [6]. Blind decryption is a versatile primitive. It can be used as a building block for many privacy-critical applications, such as privacy-preserving payment systems [7], key escrow systems, oblivious transfer protocols [8], privacy-preserving systems for digital rights management [9, 10], and private information retrieval [11]. A blind decryption scheme consists of an encryption scheme together with a blind decryption protocol intended to decrypt messages in a privacy-preserving fashion. The meaning of "blind decryption" can be easily described based on the following scenario depicted in Figure 1. Suppose that Alice has obtained several encrypted messages from an encryptor. Alice is entitled to choose and decrypt exactly one of those messages. Suppose that the decryption key k is stored on a decryption server and Alice wishes to have the server decrypt the message for her in such a way that neither the encryptor nor the decryptor learns the message chosen by Alice.

There are suggestions for practical blind decryption based on public-key cryptography [5, 6, 12–14]. It is also possible to implement the blind decryption functionality with other protocols such as secure multiparty computation [15]. However, the resulting schemes would be computationally demanding. For many applications, symmetric primitives are sufficient and computationally more efficient. In addition, they can provide secrecy that is not based on computational assumptions. Oblivious transfer schemes [16, 17] deliver the same functionality directly between the sender and the receiver without the decryption server. However, for noiseless

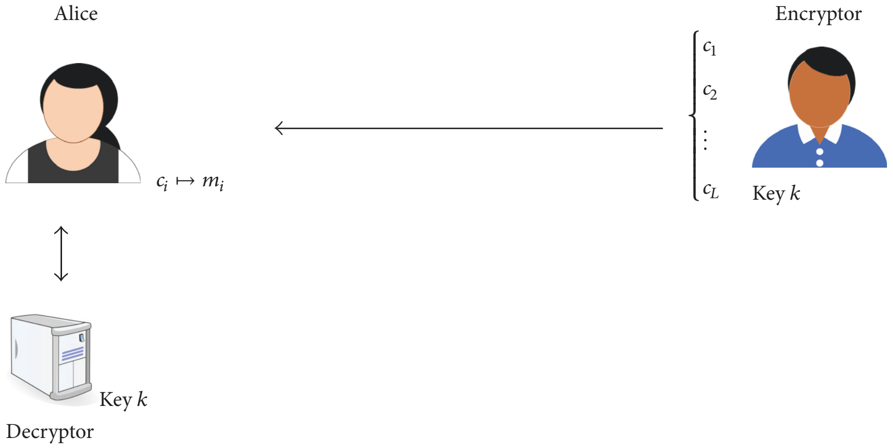


FIGURE 1: Blind decryption. Alice has obtained L ciphertexts from an encryptor and is entitled to choose exactly one of those for decryption. Alice interacts with a decryptor that shares a key k with the encryptor to transform the ciphertext message c_i into a plain-text message m_i. Neither the encryptor nor the decryptor learns the plain-text message chosen by Alice.

channels, information-theoretically secure oblivious transfer is impossible [18]. In addition, blind decryption schemes do not seem to exist, such that the privacy of the user is based on information-theoretic security. Our work aims to fill this shortage. In this paper, we give a meaningful definition of perfect secrecy for the blind decryption scenario. In particular, we formulate perfect secrecy of symmetric blind decryption in a setting in which at most one of the participants is an adversary but adhering to the protocol (at most one of the participants is honest-but-curious). We also propose a symmetric key blind decryption scheme SymmetricBlind which satisfies our definition. The scheme is based on modular arithmetic on a ring $\mathbb{Z}_{p^2}$, where p is a prime. Our main contribution is theoretical. Perfect secrecy requires the key to be changed for each decryption. Therefore, many existing applications of blind decryption which are built on the public-key case in the computational security model are not directly applicable. However, for the first time, we are able to give a meaningful definition of perfect secrecy of blind decryption and to show that blind decryption is possible in the information-theoretic security model. Additional research is needed to show which applications are possible in this model.

The paper is organized as follows. In Section 2, we describe work that is related to ours. Section 3 discusses the fundamental definitions and the preliminaries for the rest of the paper. In Section 4, we formulate three perfect secrecy properties that the blind decryption scheme needs to satisfy. In Section 5, we give a description of a symmetric blind decryption scheme SymmetricBlind. In Section 6, we show that the devised scheme satisfies our definition of perfect secrecy. Finally, Section 7 considers future work and Section 8 provides the conclusion.

2. Related Work

Chaum was the first to consider blindness in the context of digital signatures and privacy-preserving payment systems [7]. He described the first public-key blind signature scheme [19] by utilizing the properties of RSA encryption [20]. The scheme can be also used for encryption and can be therefore considered as the first blind decryption scheme. In the early articles, blind decryption is referred to as "blind decoding." Discrete logarithm based blind signature schemes were suggested in [21–24]. Sakurai and Yamane were the first to consider public-key blind decryption based on the discrete logarithm problem [6]. Their method was based on ElGamal Cryptosystem [25] and related to the blind signature of Camenisch et al. [24]. The method was later applied for the implementation of a key escrow system [12]. Mambo et al. were the first to consider blind decryption that is secure against chosen plain-text attacks by signing the ciphertext messages [26]. The resulting scheme is not capable of public-key encryption, since a secret signing key is required. Green described the first public-key blind decryption scheme [5] that is secure against adaptive chosen ciphertext attacks (IND-CCA2) using bilinear groups. The security of these constructions has been considered computationally either in the random oracle model [11] or using computational indistinguishability and infeasibility assumptions [5].

Oblivious transfer protocols are symmetric primitives that offer functionality similar to blind decryption. For oblivious transfer, there are two participants: a sender and a receiver. For the original definition of oblivious transfer, the sender transmits a message which the receiver gets with probability 1/2. The sender remains oblivious as to whether the receiver actually got the message. This form of oblivious transfer was introduced by Rabin [16]. The concept was later extended by Even et al. [17]. For $\binom{2}{1}$-oblivious transfer, the receiver can choose one from two messages without the sender knowing which of the messages was chosen. A related concept that can be considered as a further generalization is *all-or-nothing disclosure of secrets* [27] for which Alice is willing to disclose at most one secret from a set to Bob without Bob learning information about the rest of the secrets. Alice must not learn which secret Bob chose. Adaptive queries were considered by Naor and Pinkas [28]. They also considered active adversaries and provided security definitions related to the simulatability of the receivers.

Camenisch, Neven, and Shelat extended the work of Naor and Pinkas by defining *simulatable* oblivious transfer [29] and providing practical constructions for such a scheme. There are other suggestions for oblivious transfer based on problems in bilinear groups [30], groups of composite order [31], and the Diffie-Hellman problem [32–37]. These schemes are based on computational assumptions. It is impossible to achieve information-theoretic security for both of the parties using noiseless channels [18]. However, it is possible using noisy channels such as discrete memoryless channels [38] or a trusted initializer (shown by Rivest in 1999; see "unconditionally secure commitment and oblivious transfer schemes using private channels and a trusted initializer"). For the computational security setting, the functionality of oblivious transfer can be also implemented with public-key blind decryption using the method of Dodis et al. [39].

General *multiparty computation* protocols can be also applied to implement blind decryption capabilities. Secure multiparty computation was originally introduced by Yao [40] for the two-party case. The general case for $n \geq 2$ is due to Goldreich et al. [41]. However, secure multiparty computation protocols are computationally intensive in comparison to blind decryption and oblivious transfer.

3. Preliminaries

3.1. Notation. For the set of integers modulo n, we denote $\mathbb{Z}_n = \{[0], [1], \ldots, [n-1]\}$ and equate a congruence class with its least nonnegative representative. That is, we consider $\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$. By the notation $x \bmod n$ we mean the unique $i \in \{0, 1, \ldots, n-1\}$ such that $i \equiv x \pmod{n}$. We denote the uniform distribution on a set X by $U(X)$. If a random variable Z is uniformly distributed on a set X, we denote it by $Z \sim U(X)$. When an element x is sampled from $U(X)$, we denote it by $x \leftarrow U(X)$.

3.2. Symmetric Encryption. A symmetric encryption scheme $\mathsf{SE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with key space $\mathscr{K}$, plain-text space $\mathscr{M}$, and ciphertext space $\mathscr{C}$ consists of three algorithms:

(1) The key generation algorithm $\mathsf{Gen}(s)$: on inputting a security parameter s, Gen outputs a key $k \in \mathscr{K}$

(2) The encryption algorithm $\mathsf{Enc}(k, m)$: on inputting a key $k \in \mathscr{K}$ and a message $m \in \mathscr{M}$, Enc outputs a ciphertext $c \in \mathscr{C}$

(3) The decryption algorithm $\mathsf{Dec}(k, m)$: on inputting a key $k \in \mathscr{K}$ and a ciphertext $c \in \mathscr{C}$, Dec outputs a message $m \in \mathscr{M}$ such that $m = \mathsf{Dec}(k, \mathsf{Enc}(k, m))$

3.3. Blind Decryption. Blind decryption has been considered in the literature for the asymmetric case. However, in this paper, we are interested in the symmetric case that is easily adapted from the asymmetric one [5]. A symmetric blind decryption scheme $\mathsf{BlindDecyption}$ consists of a symmetric encryption scheme $\mathsf{SE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ and a two-party protocol $\mathsf{BlindDec}$. The protocol $\mathsf{BlindDec}$ is conducted between an honest user Alice and the decryption server which we shall call the decryptor. The protocol enables Alice, who is in possession of a ciphertext c, to finish the protocol with the correct decryption of c. As a result of running $\mathsf{BlindDec}$, Alice on inputting a ciphertext $c = \mathsf{Enc}(k, m) \in \mathscr{C}$ outputs either the message $m \in \mathscr{M}$ or an error message $\perp$. The decryptor, on inputting the key $k \in \mathscr{K}$, outputs nothing or an error message $\perp$. To be secure, the exchanged messages must not leak information to malicious users (the *leak-freeness property* [8]). The property can be formalized based on computational indistinguishability. For every adversary, there has to be a simulator so that the following two games are well defined. For the first game, a probabilistic polynomial time (PPT) adversary A can choose any number L of ciphertexts c_i for $i \in \{1, 2, \ldots, L\}$. It is then given the correct decryptions by executing $\mathsf{BlindDec}$ with the decryptor. Finally, A outputs the plain-text message and ciphertext pairs (m_i, c_i) for $i \in \{1, 2, \ldots, L\}$. For the second game, a simulator S chooses any number L of ciphertexts c_i for $i \in \{1, 2, \ldots, L\}$. In this game, the plain-text messages are obtained by querying a trusted party. $\mathsf{BlindDecyption}$ is *leak-free* if for every PPT adversary A there is a simulator S such that for every PPT distinguisher D the probability of distinguishing between these two games is negligible [5].

Another important property for secure blind decryption is the *blindness property*. It formalizes the idea that the decryptor must not learn anything about the actual plain-text message. This can be formalized by giving a PPT algorithm D the possibility to choose two ciphertexts c_1, c_2 and giving it oracle access to two instances of $\mathsf{BlindDec}$ based on these choices. If the probability of distinguishing these two instances is negligible for every PPT algorithm D, then $\mathsf{BlindDecyption}$ satisfies *ciphertext blindness*. For a formal and rigorous definition, see, for example, [5].

3.4. Perfect Secrecy. The notion of perfect secrecy is due to Shannon [42]. Let $\mathsf{SE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an encryption scheme with key space $\mathscr{K}$, plain-text space $\mathscr{M}$, and ciphertext space $\mathscr{C}$. Let K denote a random variable on the key space induced by Gen. SE satisfies perfect secrecy if, for every random variable M on the plain-text space, every plain-text $m \in \mathscr{M}$, and every ciphertext $c \in \mathscr{C}$,

$$\Pr[M = m \mid c = \mathsf{Enc}(K, M)] = \Pr[M = m]. \quad (1)$$

Equivalently, SE satisfies perfect secrecy if and only if, for every random variable M on the plain-text space, every plain-text messages $m_1, m_2 \in \mathscr{M}$, and every ciphertext $c \in \mathscr{C}$,

$$\begin{aligned} &\Pr[c = \mathsf{Enc}(K, M) \mid M = m_1] \\ &\quad = \Pr[c = \mathsf{Enc}(K, M) \mid M = m_2]. \end{aligned} \quad (2)$$

4. Perfect Secrecy for Symmetric Blind Decryption

In this section, we formulate a condition for the perfect secrecy of blind decryption. Instead of computational indistinguishability, we consider secrecy of symmetric blind decryption based on the information observed by the parties. In the following, let $\mathsf{SE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ together with $\mathsf{BlindDec}$ be a symmetric blind decryption scheme with key space $\mathscr{K}$, plain-text space $\mathscr{M}$, and ciphertext space $\mathscr{C}$.

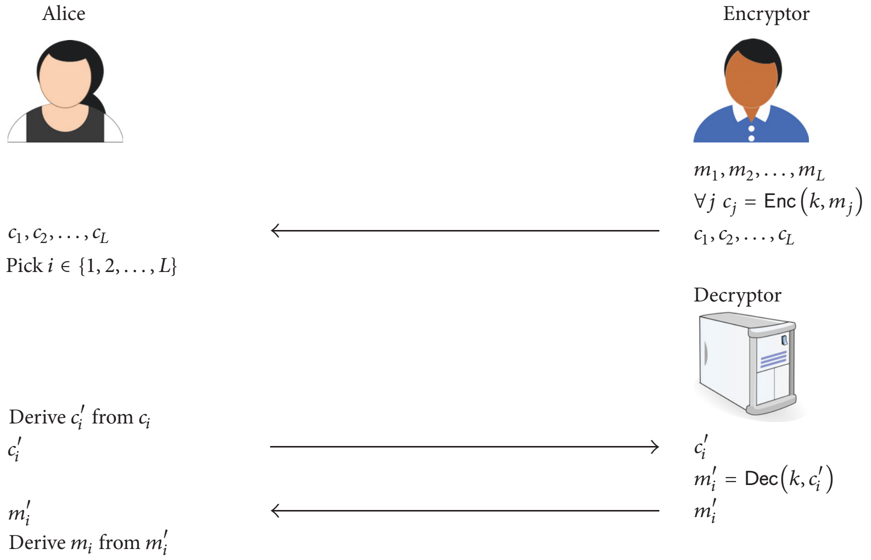


Figure 2: The general blind decryption scenario. Alice chooses a ciphertext c_i and derives a related ciphertext c_i' that she transmits to the decryptor. The decryptor responds with the corresponding plain-text message m_i' from which Alice can recover m_i.

4.1. The Scenario. For the sake of clarity, we do not consider active adversaries. We assume that the parties adhere to the blind decryption protocol and only observe the flow of messages (and possibly deduce information from those messages). Active adversaries could, for example, induce errors to the protocol messages. Such adversarial scenarios are left for future work. In addition, we do not consider the case where the decryptor is colluding with either Alice or the encryptor against the other. Such a case is equivalent to the oblivious transfer scenario and information-theoretic security is impossible for noiseless channels [18]. However, we note that such collusion scenarios are important for certain applications and need to be investigated in the future. We do consider the case where the adversary is impersonating one of the parties, which is a paramount requirement for many applications. For clarity, we also restrict ourselves to the case where Alice decrypts a single message $m \in \mathscr{M}$. Similar to the one-time pad, we assume that a new key is derived after every decryption. However, in our case, there could be several ciphertexts $c_1, c_2, \ldots, c_L$ encrypted under the same key. Nevertheless, once Alice has decrypted one of the messages, we consider that particular key used and a new key, and a new set of ciphertexts is generated.

The scenario is the following. The encryptor chooses a set of L plain-text messages m_i for $i \in \{1, 2, \ldots, L\}$. He encrypts those messages under a key k to obtain ciphertext messages $c_j = \mathsf{Enc}(k, m_j)$ for $j \in \{1, 2, \ldots, L\}$ that he transmits to Alice. Alice chooses one of those messages c_i. To hide the actual ciphertext c_i, we assume that there is a ciphertext transformation space $\mathscr{C}' \subseteq \mathscr{C}$ so that Alice can derive a related ciphertext message $c_i' \in \mathscr{C}'$ that she transmits to the decryptor. The decryptor responds with its decryption $m_i' \in \mathscr{M}$ which Alice transforms to the correct plain-text message m_i. The general scenario has been depicted in Figure 2. The used variables have been collected into Notations for easier reference.

4.2. Security Requirements. As described in Section 3.2, the scheme has to satisfy the following property.

4.2.1. Leak-Freeness. Outsiders must not learn information about the plain-text messages by observing the exchanges.

The easiest way to provide leak-freeness against outsiders is to protect each exchange with an encryption scheme that satisfies perfect secrecy. However, leakage also needs to be addressed considering the protocol participants. Considering each individual party, we can divide leak-freeness as follows.

(1) Leak-Freeness against the Encryptor. Honest-but-curious encryptor must not learn information about the plain-text message obtained by Alice at the end of the protocol by observing the blind decryption messages. The situation is depicted in Figure 3.

(2) Leak-Freeness against Alice. This property ensures that, after obtaining m_i, Alice does not learn information about the remaining $L - 1$ plain texts m_j for $j \neq i$. The situation is depicted in Figure 4.

In contrast to computational security, we cannot define leak-freeness as a distinguishing problem. Instead, we shall consider the probability distributions regarding the exchanged elements. We also want to prevent decryptor from deducing information about the plain-text message m_i.

4.2.2. Blindness against the Decryptor. This property ensures that an honest-but-curious decryption server does not learn the message Alice wants to decrypt. The situation is depicted in Figure 5.

In the computational security setting, there can be multiple applications of the blind decryption protocol for a fixed key. In our case, we want a fresh key for every decryption to achieve perfect secrecy. Therefore, we formulate leak-freeness

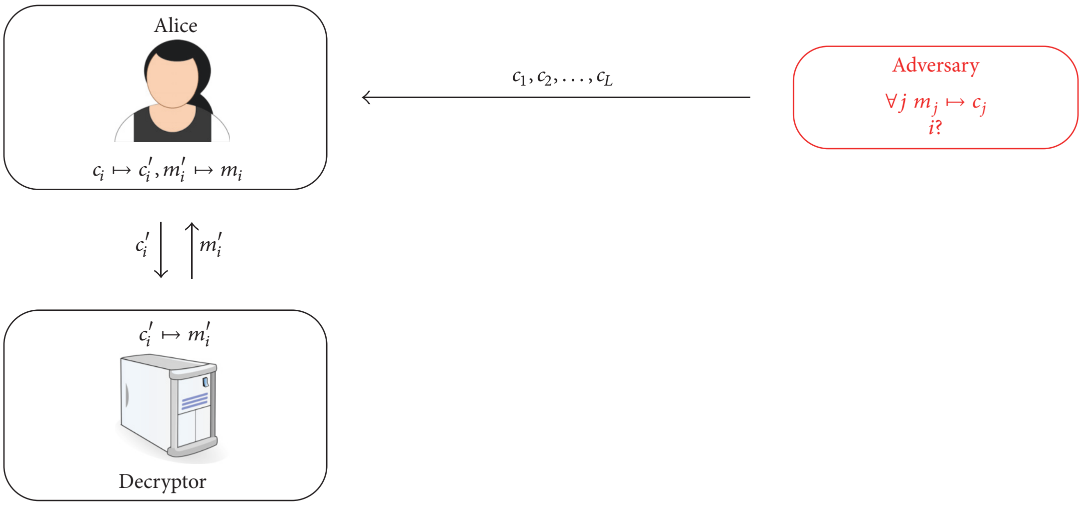


Figure 3: Malicious encryptor. The adversary attempts to learn which message was chosen by Alice.

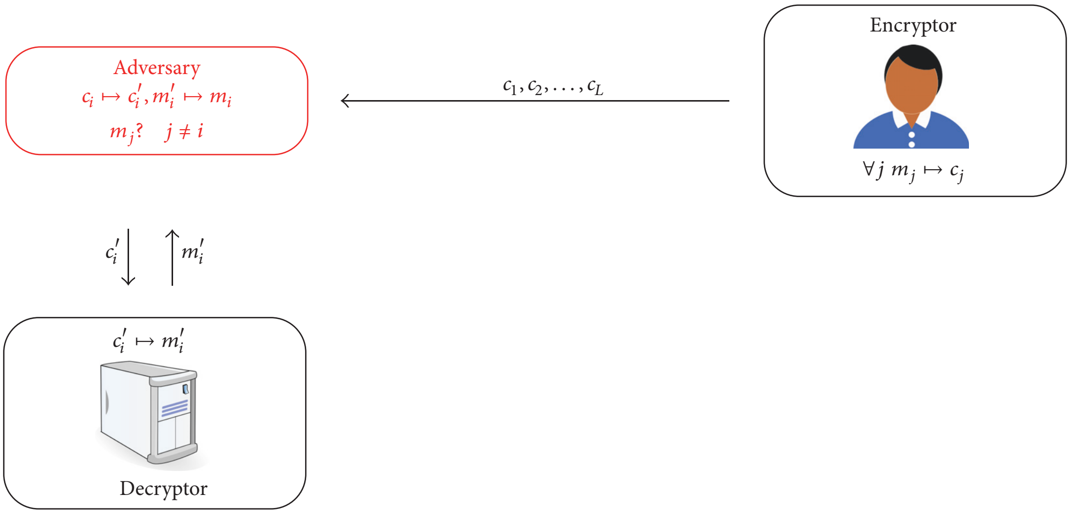


Figure 4: Malicious Alice. The adversary attempts to decrypt additional messages.

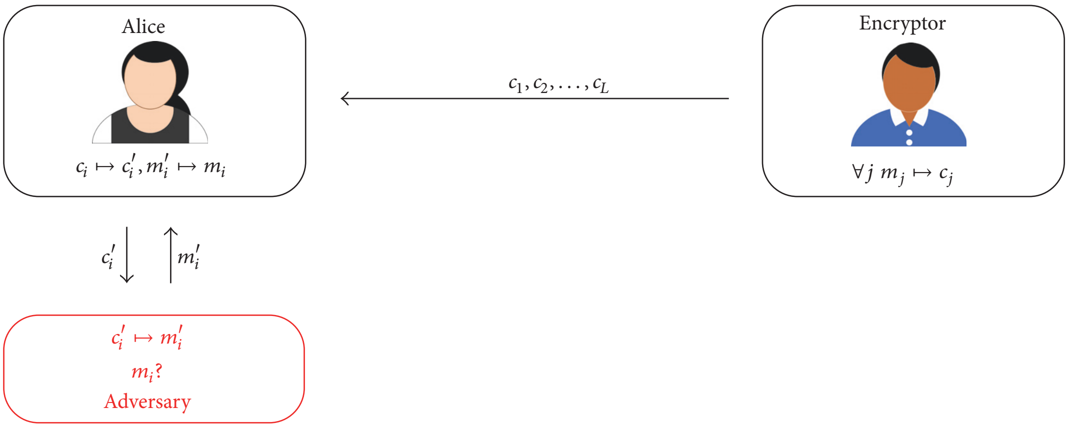


Figure 5: Malicious decryptor. The adversary attempts to learn the plain-text message that Alice obtains.

and blindness for a single decryption. However, as was described before, we want to be able to encrypt multiple messages with the same key. For example, in privacy-preserving payment systems, blind decryption is used to enable Alice to choose one (only one) item from a selection of items. This results in a scenario in which there are L plain-text and ciphertext pairs (m_j, c_j) for $j \in \{1, 2, \dots, L\}$ but there is only a single application of $\mathsf{BlindDec}$.

In the following section, we formulate these conditions based on information. Note that these conditions also provide secrecy against observers that are not participants of the scheme, since the information possessed by such observers is a proper subset of that of any of the participants. The following notation is used. Let K denote the random variable of blind decryption keys on the key space $\mathscr{K}$ induced by Gen. Let M_j for $j \in \{1, 2, \dots, L\}$ denote the random variables corresponding to the choice of m_i for $j \in \{1, 2, \dots, L\}$ by the encryptor and let M denote the random variable corresponding to the plain-text m Alice obtains at the end of the scheme. Following the standard practice [43], we assume that K is independent of M and M_j for every $j \in \{1, 2, \dots, L\}$. Let C' denote the random variable on the ciphertext transformation space $\mathscr{C}'$ for the ciphertext message c' that Alice discloses to the decryptor. Finally, let M' denote the random variable corresponding to the message m' that the decryptor responds with. These variables have been collected into Notations.

4.3. Perfect Leak-Freeness against the Encryptor. We shall first formulate leak-freeness against the encryptor. The blind decryption protocol messages c' and m' should not disclose any information about m_i to the encryptor. Equivalently, the messages should not leak information about i that was chosen by Alice even if the encryptor knows the key k and the right plain-text messages m_j for $j \in \{1, 2, \dots, L\}$.

Definition 1 (perfect leak-freeness against encryptor). A symmetric blind decryption scheme is *perfectly leak-free against the encryptor* for a single decryption of a maximum of L messages if, for every random variable M, M_j for $j \in \{1, 2, \dots, L\}$ on the plain-text space and every $m, m', m_j \in \mathscr{M}$ for $j \in \{1, 2, \dots, L\}$ and every $c' \in \mathscr{C}'$,

$$\begin{aligned}\Pr\left[M = m \mid C' = c', M' = m', M_1 = m_1, \dots, M_L = m_L\right] \\ = \Pr\left[M = m \mid M_1 = m_1, \dots, M_L = m_L\right].\end{aligned} \tag{3}$$

Our definition states that an honest-but-curious encryptor can equally easily guess the plain-text message Alice wanted to be decrypted with or without information provided by the blind decryption protocol messages c' and m'. Note that, in the normal scenario, $M = M_i$ for some $i \in \{1, 2, \dots, L\}$. However, we do not want to restrict the definition to such a case. For example, there could be homomorphic blind decryption schemes for which certain operations could be permitted on the ciphertexts. Note also that the encryptor inherently possesses more information about m than an outsider, since m is dependent on $m_1, m_2, \dots, m_L$.

4.4. Perfect Leak-Freeness against Alice. In order to be practical, the scheme needs to ensure that Alice is not able to decrypt messages. Therefore, we need to ensure that Alice obtains neither the decryption key nor any information about the decryptions of $c_1, c_2, \dots, c_L$ without interacting with the decryptor. In addition, after a single application of $\mathsf{BlindDec}$, Alice must not have any information about the remaining $L - 1$ messages. To make the requirement precise, we require that the observation of a single plain-text and ciphertext pair (m_1, c_1) does not leak any information about the decryption of another ciphertext c_2. The property is, in fact, a property of the encryption scheme.

Definition 2 (perfect leak-freeness against Alice). A symmetric encryption scheme SE satisfies *perfect leak-freeness against Alice* for a single decryption if, for every random variable M_1, M_2 on the plain-text space, every $m_1, m_2, m \in \mathscr{M}$, and every $c_1, c_2 \in \mathscr{C}$, such that $c_1 \neq c_2$,

$$\begin{aligned}\Pr[c_1 = \mathsf{Enc}(K, M_1), c_2 = \mathsf{Enc}(K, M_2) \mid M_1 = m_1, M_2 \\ = m_2] = \Pr[c_1 = \mathsf{Enc}(K, M_1), c_2 \\ = \mathsf{Enc}(K, M_2) \mid M_1 = m_1, M_2 = m].\end{aligned} \tag{4}$$

The condition states that the probability of obtaining the ciphertext pair (c_1, c_2) is the same whether we encrypt (m_1, m_2) or (m_1, m). That is, observation of the ciphertexts c_1, c_2 does not yield information about the decryption of c_2 even if we know the decryption of c_1.

4.5. Perfect Blindness against the Decryptor. We still need to consider privacy against an honest-but-curious decryptor. It is reasonable to assume that $c_1, c_2, \dots, c_L$ have been delivered to Alice using a private channel. If the decryptor can observe c_j for $j \in \{1, 2, \dots, L\}$, it means that he knows the corresponding plain-text messages, since he is in possession of the blind decryption key. Therefore, it is natural to require that the ciphertexts be protected by a separate secure channel between Alice and the encryptor. For the blindness property, we want the server to learn nothing of the actual message m that Alice derives at the end of the blind decryption scheme. In this case, the decryptor knows the correct key k as well as the messages c' and m' exchanged with Alice.

Definition 3 (perfect ciphertext blindness against the decryptor). A symmetric blind decryption scheme satisfies *perfect ciphertext blindness against the decryptor* if, for every random variable M on the plain-text space, every $m, m' \in \mathscr{M}$, and every $c' \in \mathscr{C}'$

$$\Pr\left[M = m \mid C' = c', M' = m'\right] = \Pr[M = m]. \tag{5}$$

The condition states that it is equally easy to guess the correct plain-text message with and without the information possessed by the decryptor. Note that we have assumed that $c_1, c_1, \dots, c_L$ have been delivered to Alice in perfect secrecy.

4.6. Perfect Secrecy for Symmetric Blind Decryption. Finally, we can state our definition of perfect secrecy based on the properties defined above.

Definition 4 (perfect secrecy of blind decryption). A symmetric blind decryption scheme consisting of a symmetric encryption scheme SE and a blind decryption protocol BlindDec satisfies perfect secrecy for symmetric blind decryption for a single decryption of a maximum of L messages against a single honest-but-curious party if the scheme is perfectly leak-free against the encryptor for a maximum of L messages, SE is leak-free against Alice, and the scheme satisfies perfect ciphertext blindness against the decryptor.

5. A Concrete Blind Decryption Scheme

We shall now devise a blind decryption scheme SymmetricBlind that satisfies Definition 4. We shall implement our scheme using two tiers of symmetric encryption. For the outer tier, we apply a scheme that satisfies ordinary perfect secrecy. Let that scheme be denoted by SE. The outer encryption scheme will hide information about $c_1, c_2, \ldots, c_L$ from the decryptor and also provide secrecy for c' and m' against the encryptor. To achieve perfect blindness and leak-freeness against Alice, we design an inner tier encryption scheme called 2PAD that satisfies a useful transformation property which enables us to construct a blind decryption protocol BlindDec. To sum up, our final construction will consist of two tiers of encryption and a protocol for Alice to query a single decryption from the decryptor. The general overview of the scheme is depicted in Figure 6. It would be possible to implement some of the required privacy properties with multiple applications of the one-time pad. For example, if $c_i = m_i \oplus k_i$, Alice could hide the plain-text message from the decryptor by querying for the decryption of $c_i' = c_i \oplus k'$, where k' is only known to Alice. The correct plain-text message would be obtained from $m_i' = c_i' \oplus k_i = c_i \oplus k' \oplus k_i$ by computing $m_i' \oplus k' = c_i \oplus k_i = m_i$. However, such a protocol would leak i to the decryptor, since i would be needed for decryption. In addition, for a single decryption, the decryptor would have to maintain a set of L keys which would quickly grow to an unmanageable size as L grows. In contrast, the optimal key size for single decryption would be $2|m_i|$, where $|m_i|$ is the bit length of m_i, assuming that each plain-text message is of the same bit length. Therefore, simply applying the one-time pad is not sufficient.

In the following, we first describe our inner encryption scheme 2PAD that will provide perfect leak-freeness against Alice, as well as the required message transformation property. Then, we proceed to the description of a blind decryption protocol utilizing this scheme. Finally, we combine the inner encryption scheme with an outer encryption scheme that satisfies ordinary perfect secrecy and describe the complete blind decryption scheme.

5.1. The Inner Encryption Scheme. We shall first construct an inner encryption scheme called 2PAD with some useful properties. Our inner scheme is based on modular arithmetic on the ring $\mathbb{Z}_{p^2}$, where $p \geq 5$ is a prime. Our plain-text space is $\mathbb{Z}_p$ and every $m \in \mathbb{Z}_p$ is mapped to the ciphertext space $\mathbb{Z}_{p^2}$. To satisfy Definition 2, we want to add an amount of randomness that is at least twice the binary length of m in the encryption operation. Therefore, the keys of 2PAD will consist of pairs $(x_k, y_k) \in \mathbb{Z}_p \times \mathbb{Z}_p$. Let $b \in \mathbb{Z}_{p^2}$. Then,

$$b \equiv pz' + z'' \pmod{p^2}, \tag{6}$$

where $z', z'' \in \mathbb{Z}_p$. Therefore, we can essentially represent b with two elements of $\mathbb{Z}_p$. Using such a representation, we encrypt a single message $m \in \mathbb{Z}_p$ by first sampling a random element $z \leftarrow U(\mathbb{Z}_p \setminus \{0\})$ and setting $b := (pm + z) \bmod p^2$. Then, we add the key (x_k, y_k) by computing

$$\begin{aligned} c &:= \left(px_k b^2 + py_k b + b\right) \bmod\ p^2 \\ &= px_k z^2 + py_k z + pm + z \end{aligned} \tag{7}$$

which is the ciphertext message. To enable blinding, Alice needs to be able to transform c into another ciphertext c'. The encryption operation entails such a transformation property that follows from the congruence

$$px_k b'^2 + py_k b' + b' \equiv px_k b^2 + py_k b + b' \pmod{p^2} \tag{8}$$

for every $x_k, y_k \in \mathbb{Z}_p$ and $b, b' \in \mathbb{Z}_{p^2}$ such that $b \equiv b'$ (mod p). Let m_1 be a plain text and let $c_1 = px_k b^2 + py_k b + b$ be its encryption with $b = (pm_1 + z) \bmod\ p^2$. Let c_2 now be any ciphertext under the same key (x_k, y_k) such that $c_2 \equiv c_1 \equiv z \pmod{p}$ and let m_2 be the corresponding plain text. Since $c_2 \equiv c_1 \equiv z \pmod{p}$, we have $c_2 = px_k b'^2 + py_k b' + b'$, where $b' = (pm_2 + z) \bmod\ p^2$. Now, by (8),

$$\begin{aligned} c_2 &\equiv px_k b'^2 + py_k b' + b' \equiv px_k b^2 + py_k b + b' \\ &\equiv c_1 - b + b' \pmod{p^2}, \end{aligned} \tag{9}$$

from which

$$c_2 \equiv c_1 - pm_1 + pm_2 \pmod{p^2}, \tag{10}$$

which enables us to compute m_2 using c_2, m_1, c_1 without the key (x_k, y_k). Namely, if we know a plain text m_1 and its encryption $c_1 = px_k z^2 + py_k z + pm_1 + z$, we know the decryption m_2 of c_2 for every $c_2 \equiv c_1 \pmod{p}$. The plain text m_2 can be computed by the transformation algorithm Map in Algorithm 1.

Let $z \equiv c_1 \equiv c_2 \pmod{p}$. The algorithm works because

$$\begin{aligned} &\frac{(c_2 - c_1 + pm_1)}{p} \\ &= \frac{\left(px_k z^2 + py_k z + pm_2 + z - px_k z^2 - py_k z - pm_1 - z + pm_1\right)}{p} \\ &= \frac{(pm_2)}{p} = m_2. \end{aligned} \tag{11}$$

In order to query the decryptor, Alice can transform a ciphertext c into any c' such that $c' \equiv c \pmod{p}$. The Map algorithm can transform the corresponding plain text m' to the decryption m of c.

Decryption is straightforward knowing the key (x_k, y_y). Its operation, as well as the complete encryption scheme, is described below.

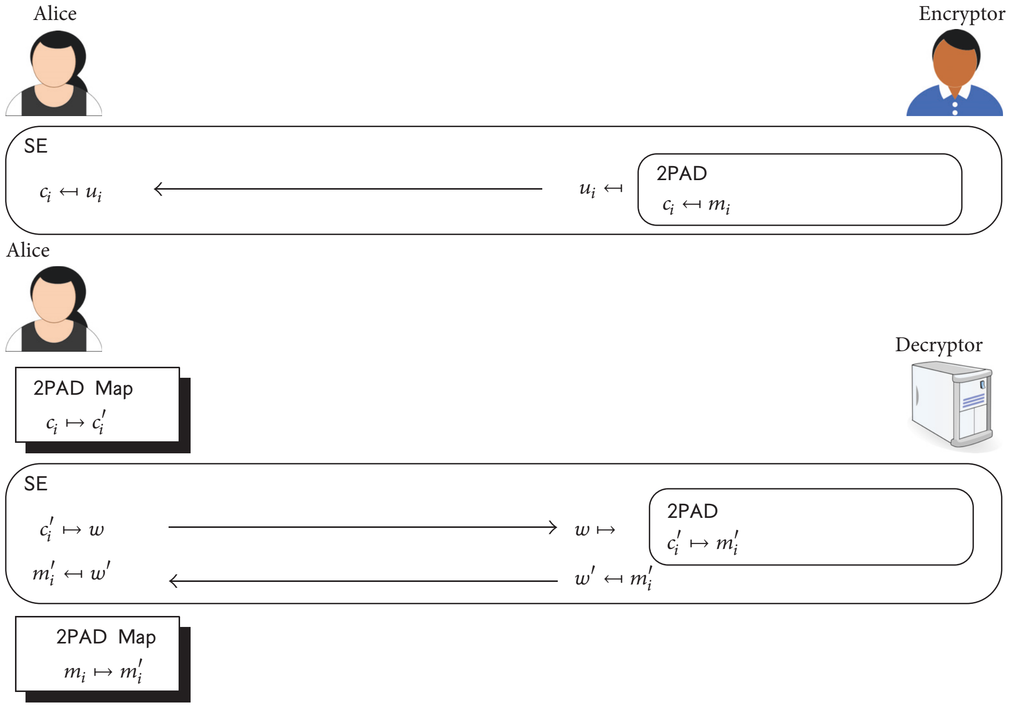

Alice
Encryptor
SE
$c_i \leftarrow\!\shortmid u_i$
$u_i \leftarrow\!\shortmid$
2PAD
$c_i \leftarrow\!\shortmid m_i$
Alice
Decryptor
2PAD Map
$c_i \mapsto c_i'$
SE
$c_i' \mapsto w$
$w \mapsto$
2PAD
$c_i' \mapsto m_i'$
$m_i' \leftarrow\!\shortmid w'$
$w' \leftarrow\!\shortmid m_i'$
2PAD Map
$m_i \mapsto m_i'$

FIGURE 6: General overview of SymmetricBlind. Two tiers of encryption are applied. The outer tier (SE) satisfies ordinary perfect secrecy. The inner tier (2PAD) provides perfect leak-freeness against Alice and has a transformation property enabling perfect blindness against the decryptor.

```
procedure Map($c_1, m_1, c_2$)
    If $c_1 \not\equiv c_2 \pmod p$ output $\perp$
    $m_2 := ((c_2 - c_1 + pm_1)/p) \bmod p$
    output $m_2$
end procedure
```

ALGORITHM 1

Definition 5 (2PAD). The symmetric encryption scheme

$$2\mathsf{PAD} = \left(\mathsf{Gen}_{2\mathsf{PAD}}, \mathsf{Enc}_{2\mathsf{PAD}}, \mathsf{Dec}_{2\mathsf{PAD}}\right) \tag{12}$$

consists of Algorithms 2, 3, and 4.

The plain-text and ciphertext spaces of 2PAD depend on the chosen prime p; the plain-text space is $\mathbb{Z}_p$, while the ciphertext space is $\mathbb{Z}_{p^2}$. Let us show the correctness of the scheme. That is,

$$\mathsf{Dec}_{2\mathsf{PAD}}\left(x_k, y_k, \mathsf{Enc}_{2\mathsf{PAD}}\left(x_k, y_k, m\right)\right) = m \tag{13}$$

for every key (x_k, y_k) and plain text m. Let $c = \mathsf{Enc}_{2\mathsf{PAD}}(x_k, y_k, m)$. Then one has

$$c = px_k b^2 + py_k b + b \equiv px_k z^2 + py_k z + pm + z \pmod{p^2} \tag{14}$$

and $c \bmod p = z$, where $z \in \mathbb{Z}_p$. Now,

$$\begin{aligned}\mathsf{Dec}_{2\mathsf{PAD}}\left(x_k, y_k, c\right) &= \frac{(t-z)}{p} \\ &= \frac{\left(p\left(-x_k\right)z^2 + p\left(-y_k\right)z + px_k z^2 + py_k z + pm + z - z\right)}{p} \\ &= \frac{(pm + z - z)}{p} = m.\end{aligned} \tag{15}$$

We shall later show that, given a single plain-text and ciphertext pair (m_1, c_1) and a ciphertext c_2 such that $c_2 \not\equiv c_1 \pmod p$, we still have information-theoretic security for c_2. That is, 2PAD satisfies perfect leak-freeness against Alice whenever $c_i \not\equiv c_j \pmod p$ for $i \neq j$. However, suppose that we have two plain-text and ciphertext pairs $(m_1, c_1), (m_2, c_2)$ such that $c_1 \not\equiv c_2 \pmod p$. We can show that the key x_k, y_k can be completely determined from such two pairs.

Proposition 6. *For every plain-text and ciphertext pair $(m_1, c_1), (m_2, c_2)$ such that $c_1 \not\equiv c_2 \pmod p$, there is a unique key (x_k, y_k) such that*

$$\begin{aligned}c_1 &= \mathsf{Enc}_{2\mathsf{PAD}}\left(x_k, y_k, m_1\right), \\ c_2 &= \mathsf{Enc}_{2\mathsf{PAD}}\left(x_k, y_k, m_2\right).\end{aligned} \tag{16}$$

```
procedure $\mathsf{Gen}_{\mathsf{2PAD}}(s)$                    ▷ $s$ determines the size for the plaintext space
    Choose a public prime $p$ such that $p \geq 5$ and $p \geq 2^s$
    $x_k \leftarrow U(\mathbb{Z}_p)$
    $y_k \leftarrow U(\mathbb{Z}_p)$
    output $(x_k, y_k)$
end procedure
```

Algorithm 2

```
procedure $\mathsf{Enc}_{\mathsf{2PAD}}(x_k, y_k, m)$    ▷ Input consists of a key $(x_k, y_k)$ and a message $m \in \mathbb{Z}_p$
    $z \leftarrow U(\mathbb{Z}_p \setminus \{0\})$
    $b \coloneqq (pm + z) \bmod p^2$
    $c \coloneqq (px_k b^2 + py_k b + b) \bmod p^2$
    output $c$
end procedure
```

Algorithm 3

```
procedure $\mathsf{Dec}_{\mathsf{2PAD}}(x_k, y_k, c)$    ▷ Input consists of a key $(x_k, y_k)$ and a ciphertext $c \in \mathbb{Z}_{p^2}$
    $z \coloneqq c \bmod p$
    $t \coloneqq (p(-x_k)z^2 + p(-y_k)z + c) \bmod p^2$
    $m \coloneqq ((t - z)/p) \bmod p$
    output $m$
end procedure
```

Algorithm 4

Proof. Let $z_1, z_2 \in \mathbb{Z}_p$ such that $z_1 \equiv c_1 \pmod{p}$ and $z_2 \equiv c_2 \pmod{p}$. Let also $v_1 = (c_1 - pm_1 - z_1)/p$ and $v_2 = (c_2 - pm_2 - z_2)/p$. Then, we have a system of two equations:

$$\begin{aligned} v_1 &= x_k z_1^2 + y_k z_1, \\ v_2 &= x_k z_2^2 + y_k z_2, \end{aligned} \tag{17}$$

where v_1, v_2, z_1, z_2 are known. Now, let

$$Z = \begin{pmatrix} z_1^2 & z_2^2 \\ z_1 & z_2 \end{pmatrix}. \tag{18}$$

Note that since $z_1, z_2 \not\equiv 0 \pmod{p}$ and $z_1 \not\equiv z_2 \pmod{p}$, we have $z_1^2 z_2 - z_1 z_2^2 \not\equiv 0$ (mod p) and Z is invertible modulo p. Therefore, the equation pair has a unique solution:

$$\begin{aligned} \begin{pmatrix} v_1 & v_2 \end{pmatrix} \cdot Z^{-1} &= \begin{pmatrix} x_k z_1^2 + y_k z_1 & x_k z_2^2 + y_k z_2 \end{pmatrix} \cdot Z^{-1} \\ &= \begin{pmatrix} x_k & y_k \end{pmatrix} \begin{pmatrix} z_1^2 & z_2^2 \\ z_1 & z_2 \end{pmatrix} \cdot Z^{-1} = \begin{pmatrix} x_k & y_k \end{pmatrix}. \end{aligned} \tag{19}$$

□

Due to the transformation algorithm Map, we require that if Bob sends L ciphertext messages $c_1, c_2, \ldots, c_L$ to Alice, we have $c_i \not\equiv c_j \pmod{p}$ for every $i \neq j$. Otherwise, it would be trivial for Alice to derive the decryptions of all of the ciphertexts from a single plain-text and ciphertext pair. Therefore, the maximum number of ciphertext messages under the same key is determined by $L \leq p - 1$.

5.2. Blind Decryption Protocol.

Next, we give a description of a blind decryption protocol based on the transformation algorithm Map.

Definition 7 ($\mathsf{BlindDec}$). Suppose that the encryptor and the decryptor share a key $(x_k, y_k) = \mathsf{Gen}_{\mathsf{2PAD}}(s)$ intended for a single decryption by Alice. Furthermore, let Alice have an encrypted message $c = \mathsf{Enc}_{\mathsf{2PAD}}(x_k, y_k, m)$ that is not known to the decryptor. Finally, suppose that the prime p is public knowledge. Let the protocol $\mathsf{BlindDec}$ be defined by the following exchange between Alice and the decryptor:

(1) Alice: compute $c' \coloneqq c \bmod p$ and transmit it to the decryptor

(2) Decryptor: reply with $m' = \mathsf{Dec}_{\mathsf{2PAD}}(x_k, y_k, c')$

(3) Alice: compute the plain-text message $m = \mathsf{Map}(c', m', c)$

Let us quickly check the correctness of BlindDec. Let $z \equiv c' \equiv c \pmod{p}$. Then, $c = px_k z^2 + py_k z + pm + z$, where m is the plain-text message. The decryptor replies with

$$\begin{aligned} m' &= \frac{\left(p\left(-x_k\right)z^2 + p\left(-y_k\right)z + z - z\right)}{p} \\ &= \left(-x_k\right)z^2 + \left(-y_k\right)z. \end{aligned} \tag{20}$$

But now Alice can compute

$$\begin{aligned} \mathsf{Map}\left(c', m', c\right) &= \frac{\left(c - z + pm'\right)}{p} \\ &= \frac{\left(px_k z^2 + py_k z + pm + z - z + pm'\right)}{p} \\ &= \frac{\left(px_k z^2 + py_k z + pm - px_k z^2 - py_k z\right)}{p} = \frac{(pm)}{p} \\ &= m, \end{aligned} \tag{21}$$

which is the correct plain-text message.

5.3. The Complete Blind Decryption Scheme. As was mentioned earlier, the communication between Alice and the encryptor has to be protected in order to prevent the decryptor from obtaining the plain-text messages corresponding to $c_1, c_2, \ldots, c_L$. If the decryptor can observe these ciphertext messages, it can freely decrypt all of them, since it knows the correct key. Therefore, we need to apply an outer encryption scheme that hides the ciphertext messages. The same solution is the easiest way to provide perfect leak-freeness against the encryptor, since it enables us to simplify the secrecy conditions. In our case, we want to protect both of these exchanges with an outer tier of encryption which provides perfect secrecy. Let $\mathsf{SE}_n = (\mathsf{Gen}_n, \mathsf{Enc}_n, \mathsf{Dec}_n)$ be any symmetric encryption scheme that satisfies perfect secrecy such that the plain-text and ciphertext space is $\mathbb{Z}_n$. We will be applying SE_n with both $n = p^2$ and $n = p$ together with 2PAD to provide the required leak-freeness and blindness properties. The outer tier is composed in the following way. Alice and the encryptor share a set of keys $k_1, k_2, \ldots, k_L$. The encryptor protects each ciphertext message by computing $u_j = \mathsf{Enc}_{p^2}(k_j, c_j)$ for $j \in \{1, 2, \ldots, L\}$. It sends $u_1, u_2, \ldots, u_L$ to Alice. Similarly, Alice and the decryptor share a pair of keys k_C, k_P that are used to protect c_i' and m_i'. Alice sends $w = \mathsf{Enc}_p(k_C, c')$ to the decryptor that responds with $w' = \mathsf{Enc}_p(k_P, m')$. The resulting scheme SymmetricBlind is defined as follows.

Definition 8 (SymmetricBlind). Let $\mathsf{SE}_n = (\mathsf{Gen}_n, \mathsf{Enc}_n, \mathsf{Dec}_n)$ be a symmetric encryption scheme such that the plain-text and ciphertext space is $\mathbb{Z}_n$ and let SE_n satisfy perfect secrecy. Let Alice and the encryptor share a set of keys $k_1, k_2, \ldots, k_L$. Let Alice and the decryptor share a pair of keys k_C, k_P intended for a single blind decryption by Alice. Also let the encryptor and the decryptor share a blind decryption key $(x_k, y_k) = \mathsf{Gen}_{2PAD}(s)$, where $2^s \geq L + 1$, which is intended for single blind decryption by Alice. SymmetricBlind is determined by Box 1.

Alice		*Encryptor*
		Choose $m_1, m_2, \ldots, m_L$
		$\forall j$:
		$c_j = \mathsf{Enc}_{2PAD}(x_k, y_k, m_j)$
		such that
		$c_j \not\equiv c_{j'} \pmod{p}\ \forall j \neq j'$
		$\forall j : u_j = \mathsf{Enc}_{p^2}(k_j, c_j)$
$u_1, u_2, \ldots, u_L$	⟵	$u_1, u_2, \ldots, u_L$
$\forall j\ c_j = \mathsf{Dec}_{p^2}(k_j, u_j)$		
Pick i		
$c' = c_i \mod p$		
$w = \mathsf{Enc}_p(k_C, c')$		*Decryptor*
w	⟶	w
		$c' = \mathsf{Dec}_p(k_C, w)$
		$m' = \mathsf{Dec}_{2PAD}(x_k, y_k, c')$
		$w' = \mathsf{Enc}_p(k_P, m')$
w'	⟵	w'
$m' = \mathsf{Dec}_p(k_P, w')$		
$m_i = \mathsf{Map}(c', m', c_i)$		

Box 1

Note that we require that the parameter s determining the size of the plain-text space satisfy $2^s \geq L+1$ to ensure that the generated prime p satisfies $L \leq p-1$ and the scheme supports at least the encryption of L messages.

6. Security of SymmetricBlind

We shall now consider the security of SymmetricBlind. We proceed to show that the devised scheme satisfies the three conditions formulated in Section 4: perfect leak-freeness against the encryptor, perfect leak-freeness against Alice, and perfect blindness against the decryptor.

6.1. Perfect Leak-Freeness against the Encryptor

Proposition 9. SymmetricBlind *satisfies perfect leak-freeness against the encryptor for a single decryption of a maximum of $L \leq p - 1$ messages, where p is determined by* $\mathsf{Gen}_{2PAD}(s)$.

Proof. The claim follows directly from the observation that the encryptor sees only w and w'. By the description of SymmetricBlind, c' and m' are protected by encryption satisfying perfect secrecy and thus do not leak information to the encryptor. □

It is easy to see that the outer tier of encryption is necessary. Suppose that the outer encryption scheme was not applied. Then c' would leak $c_i \bmod p$ which would betray i to the encryptor.

6.2. Perfect Blindness against Decryptor. We shall now prove that the decryptor does not get information about the plain-text message.

Proposition 10. SymmetricBlind *satisfies perfect blindness against the decryptor for a single blind decryption.*

Proof. Since $c_1, c_2, \ldots, c_L$ are protected with perfect secrecy, we only need to show that

$$\Pr\left[M = m \mid C' = c', M' = m'\right] = \Pr\left[M = m\right], \qquad (22)$$

where C' and M' are the random variables associated with the messages c' and m', respectively. Let X, Y denote the random variables corresponding to the key elements $(x_k, y_k) \leftarrow \mathsf{Gen}(s)$, respectively. The reply m' from the decryptor is completely determined by the key (x_k, y_k) and the element $c' = c_i \bmod p$, since $m' = (-x_k)c'^2 + (-y_k)c'$. Therefore,

$$\begin{aligned} &\Pr\left[\mathrm{M} = m \mid C' = c', M' = m'\right] \\ &\quad = \Pr\left[M = m \mid X = x_k, Y = y_k, C' = c'\right]. \end{aligned} \qquad (23)$$

Let us consider C'. By the description of the scheme, we have $C' = C_i \bmod p$, where i is the chosen index of Alice. But, for every i, we have, by the description of $\mathsf{Enc_{2PAD}}$, that $C_i \bmod p \sim U(\mathbb{Z}_p \setminus \{0\})$. Therefore, C' is independent of X and Y and

$$\begin{aligned} &\Pr\left[M = m \mid X = x_k, Y = y_k, C' = z\right] \\ &\quad = \Pr\left[M = m \mid X = x_k, Y = y_k, C' = z'\right] \end{aligned} \qquad (24)$$

for every $z, z' \in \mathbb{Z}_p \setminus \{0\}$ and

$$\begin{aligned} &\Pr\left[M = m \mid X = x_k, Y = y_k\right] \\ &\quad = \sum_{z \in \mathbb{Z}_p \setminus \{0\}} \Pr\left[M = m \mid X = x_k, Y = y_k, \mathrm{C}' = z\right] \\ &\quad \cdot \Pr\left[C' = z \mid X = x_k, Y = y_k\right] = \frac{1}{p-1} \\ &\quad \cdot \sum_{z \in \mathbb{Z}_p \setminus \{0\}} \Pr\left[M = m \mid X = x_k, Y = y_k, C' = z\right] \\ &\quad = \Pr\left[M = m \mid X = x_k, Y = y_k, C' = z\right] \end{aligned} \qquad (25)$$

for any $z \in \mathbb{Z}_p$. By our assumption, M is independent of X and Y and therefore we have

$$\Pr\left[M = m \mid X = x_k \cap Y = y_k\right] = \Pr\left[M = m\right], \qquad (26)$$

which shows our claim. □

The proof shows that the decryptor (with the knowledge of the key (x_k, y_k) and c' and m') does not gain any information about the plain-text message m assuming that c_j for $j \in \{1, 2, \ldots, L\}$ have been delivered to Alice in perfect secrecy. Considering the secrecy against the decryptor, it would suffice to send c' without the additional level of encryption. However, the additional level is necessary to achieve leak-freeness against the encryptor.

6.3. Perfect Leak-Freeness against Alice. We shall now consider an honest-but-curious Alice and show that the observation of a single plain-text and ciphertext pair (m_1, c_1) does not yield information about the decryption of c_2 for $c_2 \not\equiv c_1 \pmod{p}$.

Proposition 11. SymmetricBlind *satisfies perfect leak-freeness against Alice for a single decryption of a maximum of $L \leq p-1$ ciphertexts.*

Proof. By the description of SymmetricBlind, the ciphertext messages $c_1, c_2, \ldots, c_L$ are of different congruence class modulo p. Let M_1, M_2 be random variables over the plain-text space $\mathbb{Z}_p$. Let X, Y denote the random variables corresponding to the key elements $(x_k, y_k) = \mathsf{Gen_{2PAD}}(s)$. We have to show that

$$\begin{aligned} &\Pr\left[c_1 = \mathsf{Enc_{2PAD}}\left(X, Y, M_1\right), c_2\right. \\ &\quad = \mathsf{Enc_{2PAD}}\left(X, Y, M_2\right) \mid M_1 = m_1, M_2 = m_2, c_1 \\ &\quad \left.\not\equiv c_2 \pmod{p}\right] = \Pr\left[c_1 = \mathsf{Enc_{2PAD}}\left(X, Y, M_1\right), c_2\right. \\ &\quad = \mathsf{Enc_{2PAD}}\left(X, Y, M_2\right) \mid M_1 = m_1, M_2 = m, c_1 \\ &\quad \left.\not\equiv c_2 \pmod{p}\right] \end{aligned} \qquad (27)$$

for every $m_1, m_2, m \in \{0, 1, 2, \ldots, p-1\}$ and $c_1, c_2 \in \mathbb{Z}_{p^2}$ such that $c_1 \not\equiv c_2 \pmod{p}$. Given a valid assignment for m_1, c_1 and c_2, it suffices to show that

$$\begin{aligned} &\Pr\left[c_1 = \mathsf{Enc_{2PAD}}\left(X, Y, M_1\right), c_2\right. \\ &\quad = \mathsf{Enc_{2PAD}}\left(X, Y, M_2\right) \mid M_1 = m, M_2 = m_2, c_1 \\ &\quad \left.\not\equiv c_2 \pmod{p}\right] = \frac{1}{p^2} \end{aligned} \qquad (28)$$

for every $m \in \mathbb{Z}_p$. By Proposition 6, for every plain-text and ciphertext pair $(m_1, c_1), (m, c_2)$ such that $c_1 \not\equiv c_2 \pmod{p}$, there is a unique key (x_k, y_k). Therefore,

$$\begin{aligned} &\Pr\left[c_1 = \mathsf{Enc_{2PAD}}\left(X, Y, M_1\right), c_2\right. \\ &\quad = \mathsf{Enc_{2PAD}}\left(X, Y, M_2\right) \mid M_1 = m_1, M_2 = m, c_1 \\ &\quad \left.\not\equiv c_2 \pmod{p}\right] = \Pr\left[X = x_k, Y = y_k\right]. \end{aligned} \qquad (29)$$

By the definition of $\mathsf{Gen_{2PAD}}$, X and Y are independent and we have

$$\begin{aligned} \Pr\left[X = x_k, Y = y_k\right] &= \Pr\left[X = x_k\right] \cdot \Pr\left[Y = y_k\right] \\ &= \frac{1}{p^2}. \end{aligned} \qquad (30)$$

□

We have now established the perfect secrecy of SymmetricBlind according to Definition 4.

Table 1: Parameter examples for SymmetricBlind.

p	Decryptor key length [bits]	Plain text length [bits]	Ciphertext length [bits]
5	12	3	5
7	12	3	6
11	16	4	7
23	20	5	10
101	28	7	14
1009	40	10	20
5003	52	13	25
20011	60	15	29
$2^{31} - 1$	124	31	62
$2^{61} - 1$	244	61	122
$2^{127} - 1$	508	127	254

6.4. The Parameters. An optimal encryption scheme, with plain-text space $\mathscr{M}$, that satisfies perfect leak-freeness against Alice for a single decryption needs $2\log_2|\mathscr{M}|$ bits of randomness for a key. 2PAD achieves exactly this bound, since the plain-text space is $\mathbb{Z}_p$ and a single key (x_k, y_k) contains $2\log_2 p$ bits of randomness. Assuming that messages and keys are represented by binary strings, we need $2\lceil\log_2 p\rceil$ bits of key to encrypt messages of length $\lfloor\log_2 p\rfloor$. For a single decryption with SymmetricBlind, the decryptor needs to store the key elements $x_k, y_k \in \mathbb{Z}_p$, as well as the keys k_C, k_P. The keys k_C, k_P are used to encrypt messages of $\mathbb{Z}_p$. Therefore, $\lceil\log_2 p\rceil$ bits for each of these keys suffice for perfect secrecy. In total, the decryptor needs to store key material of $4\lceil\log_2 p\rceil$ bits for a single decryption of a message of bit length $\lfloor\log_2 p\rfloor$. Since the ciphertext space is $\mathbb{Z}_{p^2}$, the ciphertext's length in bits is approximately twice the plain text's length. Depending on the length of the plain-text messages and the needed maximum number of encryptions $L \leq p-1$, we should therefore choose the smallest possible p, since its bit size has no effect on the security of the scheme. Table 1 lists some possible choices for p and the resulting key, plain text, and ciphertext lengths in bits. Note that for long plain-text messages the maximum number of messages L is practically unlimited.

7. Future Work

There are two main drawbacks of the construction presented in this paper. First, we have not considered active adversaries. Similar to the one-time pad, we have only considered such adversaries that observe the flow of messages. For practical scenarios, we need to consider adversaries that actively induce errors into the protocol flow. However, such considerations are most naturally conducted in the computational infeasibility model which has been used, for instance, in [5]. In the active adversaries setting, it would also be natural to consider the security of the devised scheme in the framework of computational indistinguishability such that the truly random keys are exchanged with pseudorandom bit strings. In particular, the computationally hard version of our scheme yields efficient practical implementation. The computational security model is also more appealing considering applications due to the limitations induced by the information-theoretic model. For example, in the information-theoretic security model, private information retrieval requires an amount of communication that is at least the size of the database [44]. Similarly, in SymmetricBlind, a fresh key is needed for each decryption resulting in limitations regarding existing applications. For example, applications that require adaptive queries cannot be instantiated with SymmetricBlind, since a fresh key would be required for each query. We leave it for future research to consider SymmetricBlind and its possible generalizations and applications in the computational security model.

The second drawback is that we have only considered the case of a single adversary. While it does not make sense to consider a scenario where Alice is colluding with the encryptor against the decryptor, the scenario where the encryptor and the decryptor are colluding is an important one. For many scenarios, Alice cannot be certain whether the encryptor and the decryptor are in fact separate entities. However, if they are a single entity, the scenario is identical to oblivious transfer. We cannot achieve information-theoretic security in such a case [18]. For example, it is easy to see that our construction fails for colluding encryptor and decryptor. If that is the case, we effectively remove the outer layer of encryption, which means that $c' = c_i \bmod p$ leaks i to the adversary. To provide security against colluding encryptor and decryptor, we would need to detect such collusion or to turn to computational assumptions. We leave the question as an open problem for future research. Another interesting question for future work is to consider the case where we do not apply the outer layer of encryption from the encryptor to Alice. Thus far, we have defined perfect blindness so that the decryptor has absolutely no information about the plain-text message. However, we could relax the requirement so that, similar to leak-freeness against the encryptor, the information is conditioned on the plain texts $m_1, m_2, \ldots, m_L$. In other words, we could relax the requirement so that the decryptor may observe the selection (and the corresponding plain-text messages) given to Alice. Such relaxation is natural in the oblivious transfer case where the encryptor and the decryptor are the same entity. We could then define blindness as a property requiring only that the selection i be hidden. It is again easy to see that our scheme without the outer layer of encryption fails such a property. If $c_1, c_2, \ldots, c_L$ are not protected, then $c' = c_i \bmod p$ leaks i. Similarly, attempting to convert SymmetricBlind into an oblivious transfer scheme using the method of Dodis et al. is impossible, since SymmetricBlind requires that the parties be truly separate. The unification of encryptor and decryptor leaks i even in the computational security model [39]. We leave this consideration also for future work.

8. Conclusion

In this paper, we give a definition of perfect secrecy for symmetric blind decryption in the setting where one of the parties may be malicious but adhering to the protocol of the scheme. We consider neither active adversaries nor the setting where two of the participants are colluding against the third. We construct a symmetric blind decryption scheme SymmetricBlind and show that it satisfies our definition of perfect secrecy. The scheme is based on two layers of encryption, where the inner layer utilizes a novel encryption scheme 2PAD given in this paper. 2PAD is based on modular arithmetic with $\mathbb{Z}_{p^2}$ as the ciphertext space, $\mathbb{Z}_p$ as the plain-text space, and $\mathbb{Z}_p \times \mathbb{Z}_p$ as the key space, where $p \geq 5$ is a prime. The security of SymmetricBlind is shown information-theoretically and does not depend on the size of p. For a fixed blind decryption key, SymmetricBlind supports a single blind decryption from a selection of $L \leq p-1$ messages. For a single decryption of a message of bit length $\lfloor \log_2 p \rfloor$, the decryption server needs to store key material of $4\lceil \log_2 p \rceil$ bits.

Notations

Variables

$\mathscr{K}$:	Key space
$\mathscr{M}$:	Plain-text space
$\mathscr{C}$:	Ciphertext space
$\mathscr{C}'$:	Ciphertext transformation space
k:	Blind encryption/decryption key
L:	The number of messages encrypted under a single blind decryption key
$m_1, m_2, \ldots, m_L$:	Plain-text messages chosen by the encryptor
$c_1, c_2, \ldots, c_L$:	Ciphertext messages obtained by encrypting with the blind encryption key
c or c_i:	Ciphertext message chosen by Alice
c' or c_i':	Transformed ciphertext message chosen by Alice
m' or m_i':	Decryption of c' under the blind decryption key
m or m_i:	The plain-text message Alice obtains at the end of the scheme.

Random Variables

K:	Random variable on $\mathscr{K}$ induced by Gen
$M_1, M_2, \ldots, M_L$:	Random variables corresponding to the choice of $m_1, m_2, \ldots, m_L$ by the encryptor
C':	Random variable on $\mathscr{C}'$ induced by Alice using BlindDec
M':	Random variable on $\mathscr{M}$ induced by decryption of C' by the decryptor
M:	Random variable corresponding to the plain-text message m Alice obtains at the end of the scheme.

Disclosure

A preprint of a preliminary version of this manuscript can be found in [45].

Conflicts of Interest

The author declares that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

Financial support from Infotech Oulu Graduate School and the following foundations is gratefully acknowledged: Finnish Foundation for Technology Promotion, the Nokia Foundation, Tauno Tönning Foundation, Walter Ahslträm Foundation, and the Finnish Foundation for Economic and Technology Sciences (KAUTE).

References

[1] B. Thuraisingham, "Big data security and privacy," in *Proceedings of the 5th ACM Conference on Data and Application Security and Privacy (CODASPY '15)*, pp. 279-280, ACM, New York, NY, USA, March 2015.

[2] Office for Civil Rights, United State Department of Health and Human Services, Medical privacy. national standards of protect the privacy of personalhealth-information, 2013, http://www.hhs.gov/ocr/privacy/hipaa/administrative/privacyrule/index.html.

[3] European Parliament, Directive 95/46/EC of the European Parliament and of the Council of 24 october 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data, 1995, http://eur-lex.europa.eu/.

[4] S. Coull, M. Green, and S. Hohenberger, "Controlling access to an oblivious database using stateful anonymous credentials," in *Public key cryptography—PKC 2009*, S. Jarecki and G. Tsudik, Eds., vol. 5443 of *Lecture Notes in Computer Science*, pp. 501–520, Springer, Berlin, Germany, 2009.

[5] M. Green, "Secure blind decryption," in *Public Key Cryptography—PKC 2011*, D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, Eds., vol. 6571 of *Lecture Notes in Computer Science*, pp. 265–282, Springer Berlin Heidelberg, Berlin, Germany, 2011.

[6] K. Sakurai and Y. Yamane, "Blind decoding, blind undeniable signatures, and their applications to privacy protection," in *Information Hiding*, R. Anderson, Ed., vol. 1174 of *Lecture Notes in Computer Science*, pp. 257–264, Springer, Berlin, Germany, 1996.

[7] D. Chaum, "Blind signatures for untraceable payments," in *Advances in Cryptology*, D. Chaum, R. Rivest, and A. Sherman, Eds., pp. 199–203, Springer, Boston, Mass, USA, 1983.

[8] M. Green and S. Hohenberger, "Blind identity-based encryption and simulatable oblivious transfer," in *Advances in cryptology—ASIACRYPT 2007*, K. Kurosawa, Ed., vol. 4833 of *Lecture Notes in Computer Science*, pp. 265–282, Springer, Berlin, Germany, 2007.

[9] R. Perlman, C. Kaufman, and R. Perlner, "Privacy-preserving DRM," in *Proceedings of the 9th Symposium on Identity and Trust on the Internet (IDTRUST '10)*, pp. 69–83, Association for Computing Machinery, Gaithersburg, Md, USA, April 2010.

[10] L. L. Win, T. Thomas, and S. Emmanuel, "Privacy enabled digital rights management without trusted third party assumption," *IEEE Transactions on Multimedia*, vol. 14, no. 3, pp. 546–554, 2012.

[11] C. P. Schnorr and M. Jakobsson, "Security of signed ElGamal encryption," in *Advances in cryptology—ASIACRYPT 2000*, vol.

1976 of *Lecture Notes in Computer Science*, pp. 73–89, Springer, Berlin, Germany, 2000.

[12] K. Sakuraii, Y. Yamane, S. Miyazaki, and T. Inoue, "A key escrow system with protecting user's privacy by blind decoding," in *Information Security*, E. Okamoto, G. Davida, and M. Mambo, Eds., vol. 1396 of *Lecture Notes in Computer Science*, pp. 147–157, Springer, Berlin, Germany, 1998.

[13] Y. Sameshima, "A key escrow system of the RSA cryptosystem," in *Information Security*, E. Okamoto, G. Davida, and M. Mambo, Eds., vol. 1396 of *Lecture Notes in Computer Science*, pp. 135–146, Springer, Berlin, Germany, 1998.

[14] L. T. Phong and W. Ogata, "New identity-based blind signature and blind decryption scheme in the standard model," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E92.A, no. 8, pp. 1822–1835, 2009.

[15] A. C.-C. Yao, "How to generate and exchange secrets," in *Proceedings of the 27th Annual Symposium on Foundations of Computer Science*, pp. 162–167, Toronto, Canada, October 1986.

[16] M. O. Rabin, "How to exchange secrets with oblivious transfer," TR-81, Aiken Computation Lab, Harvard University, Cambridge, Mass, USA, 1981.

[17] S. Even, O. Goldreich, and A. Lempel, "A randomized protocol for signing contracts," *Communications of the Association for Computing Machinery*, vol. 28, no. 6, pp. 637–647, 1985.

[18] I. Damgård, J. Kilian, and L. Salvail, "On the (im)possibility of basing oblivious transfer and bit commitment on weakened security assumptions," in *Proceedings of the 17th International Conference on Theory and Application of Cryptographic Techniques (EUROCRYPT '99)*, vol. 1999, pp. 56–73, Springer, Berlin, Germany.

[19] D. Chaum, "Security without identification: transaction systems to make big brother obsolete," *Communications of the ACM*, vol. 28, no. 10, pp. 1030–1044, 1985.

[20] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the Association for Computing Machinery*, vol. 21, no. 2, pp. 120–126, 1978.

[21] D. Chaum and T. Pedersen, "Wallet databases with observers," in *Advances in Cryptology—CRYPTO' 92*, vol. 740 of *Lecture Notes in Computer Science*, pp. 89–105, Springer, Berlin, Germany, 1993.

[22] T. Okamoto, "Provable secure and practical identification schemes and corresponding signature schemes," in *Advances in Cryptology—CRYPTO '92*, E. Brickell, Ed., vol. 740 of *Lecture Notes in Computer Science*, pp. 31–53, Springer, Berlin, Germany, 1992.

[23] P. Horster, M. Michels, and H. Petersen, "Meta-Message recovery and Meta-Blind signature schemes based on the discrete logarithm problem and their applications," in *Advances in Cryptology—ASIACRYPT'94*, J. Pieprzyk and R. Safavi-Naini, Eds., vol. 917 of *Lecture Notes in Computer Science*, pp. 224–237, Springer, Berlin, Germany, 1995.

[24] J. L. Camenisch, J.-M. Piveteau, and M. A. Stadler, "Blind signatures based on the discrete logarithm problem," in *Advances in Cryptology—EUROCRYPT '94*, A. De Santis, Ed., vol. 950 of *Lecture Notes in Computer Science*, pp. 428–432, Springer, Berlin, Germany, 1995.

[25] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469–472, 1985.

[26] M. Mambo, K. Sakurai, and E. Okamoto, "How to utilize the transformability of digital signatures for solving the oracle problem," in *Advances in Cryptology—ASIACRYPT '96*, K. Kim and T. Matsumoto, Eds., vol. 1163 of *Lecture Notes in Computer Science*, pp. 322–333, Springer, Berlin, Germany, 1996.

[27] G. Brassard, C. Crépeau, and J.-M. Robert, "All-or-nothing disclosure of secrets," in *Proceedings of the Advances in Cryptology (CRYPTO '86)*, vol. 263, pp. 234–238, Springer, Santa Barbara, Cali, USA, 1987.

[28] M. Naor and B. Pinkas, "Oblivious transfer with adaptive queries," in *Advances in Cryptology—CRYPTO 99*, M. Wiener, Ed., vol. 1666 of *Lecture Notes in Computer Science*, pp. 573–590, Springer, Berlin, Germany, 1999.

[29] J. Camenisch, G. Neven, and a. shelat, "Simulatable adaptive oblivous transfer," in *Advances in Cryptology—EUROCRYPT 2007*, M. Naor, Ed., vol. 4515 of *Lecture Notes in Computer Science*, pp. 573–590, Springer, Berlin, Germany, 2007.

[30] M. Green and S. Hohenberger, "Universally composable adaptive oblivious transfer," in *Advances in Cryptology—ASIACRYPT 2008*, J. Pieprzyk, Ed., vol. 5350 of *Lecture Notes in Comput. Sci.*, pp. 179–197, Springer, Berlin, Germany, 2008.

[31] S. A. Jarecki and X. Liu, "Efficient oblivious pseudorandom function with applications to adaptive OT and secure computation of set intersection," in *Theory of Cryptography*, O. Reingold, Ed., vol. 5444 of *Lecture Notes in Computer Science*, pp. 577–594, Springer, Berlin, Germany, 2009.

[32] K. Kurosawa and R. Nojima, "Simple adaptive oblivious transfer without random oracle," in *Advances in cryptology—ASIACRYPT 2009*, M. Matsui, Ed., vol. 5912 of *Lecture Notes in Computer Science*, pp. 334–346, Springer, Berlin, Germany, 2009.

[33] K. Kurosawa, R. Nojima, and L. T. Phong, "Efficiency-improved fully simulatable adaptive OT under the DDH assumption," in *Proceedings of the 7th International Conference on Security and Cryptography for Networks (SCN '10)*, vol. 6280, pp. 172–181, Springer, Amalfi, Italy, September 2010.

[34] M. Green and S. Hohenberger, "Practical adaptive oblivious transfer from simple assumptions," in *Theory of Cryptography*, Y. Ishai, Ed., vol. 6597 of *Lecture Notes in Computer Science*, pp. 347–363, Springer, Berlin, Germany, 2011.

[35] K. Kurosawa, R. Nojima, and L. T. Phong, "Generic Fully Simulatable Adaptive Oblivious Transfer," in *Applied Cryptography and Network Security*, J. Lopez and G. Tsudik, Eds., vol. 6715 of *Lecture Notes in Computer Science*, pp. 274–291, Springer, Berlin, Germany, 2011.

[36] B. Zhang, H. Lipmaa, C. Wang, and K. Ren, "Practical fully simulatable oblivious transfer with sublinear communication," in *Financial Cryptography and Data Security*, A.-R. Sadeghi, Ed., vol. 7859 of *Lecture Notes in Computer Science*, pp. 78–95, Springer, Berlin, Germany, 2013.

[37] V. Guleria and R. Dutta, "Efficient adaptive oblivious transfer without q-type assumptions in UC framework," in *Information and Communications Security*, L. C. K. Hui, S. H. Qing, E. Shi, and S. M. Yiu, Eds., vol. 8958 of *Lecture Notes in Computer Science*, pp. 105–119, Springer International Publishing, New York, NY, USa, 2015.

[38] C. Crépeau, K. Morozov, and S. Wolf, "Efficient unconditional oblivious transfer from almost any noisy channel," in *Security in Communication Networks*, C. Blundo and S. Cimato, Eds., vol. 3352 of *Lecture Notes in Computer Science*, pp. 47–59, Springer, Berlin, Germany, 2005.

[39] Y. Dodis, S. Halevi, and T. Rabin, "A cryptographic solution to a game theoretic problem," in *Proceedings of the 20th Annual International Cryptology Conference (CRYPTO '00)*, M. Bellare,

Ed., vol. 1880 of *Lecture Notes in Computer Science*, pp. 112–130, Springer, Santa Barbara, Calif, USA, August 2000.

[40] A. C. Yao, "Protocols for secure computations," in *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science (SFCS '08)*, pp. 160–164, November 1982.

[41] O. Goldreich, S. Micali, and A. Wigderson, "How to play any mental game," in *Proceedings of the 19th Annual ACM Conference on Theory of Computing (STOC'87)*, pp. 218–229, IEEE Press, New York, NY, USA, May 1987.

[42] C. E. Shannon, "Communication theory of secrecy systems," *The Bell System Technical Journal*, vol. 28, no. 4, pp. 656–715, 1949.

[43] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*, Chapman & Hall, Boca Raton, Fla, USA, 2007.

[44] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, "Private information retrieval," *Journal of the ACM*, vol. 45, no. 6, pp. 965–982, 1998.

[45] J. Partala, Symmetric blind decryption with perfect secrecy, CoRR, 2015, http://arxiv.org/abs/1510.06231.

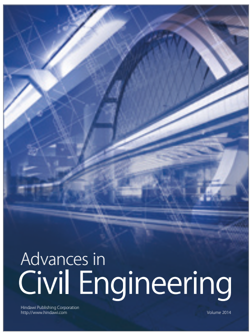
Journal of
Engineering

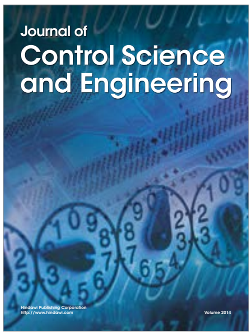
The Scientific World Journal

International Journal of
Rotating Machinery

Journal of
Sensors

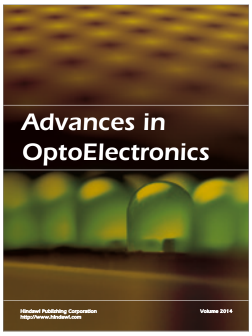
International Journal of
Distributed Sensor Networks


Advances in
Civil Engineering


Journal of
Control Science and Engineering

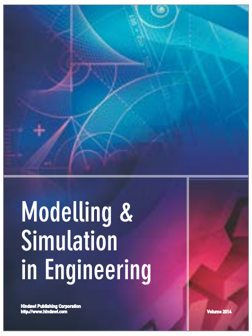
Journal of
Robotics

Journal of
Electrical and Computer Engineering

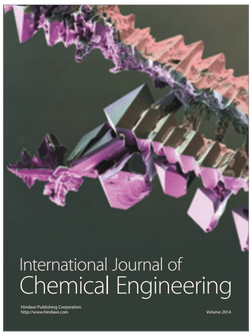

Advances in
OptoElectronics

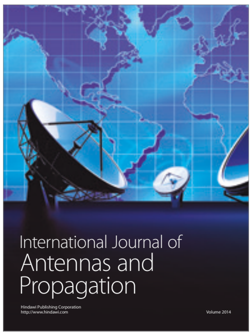
VLSI Design

International Journal of
Navigation and Observation

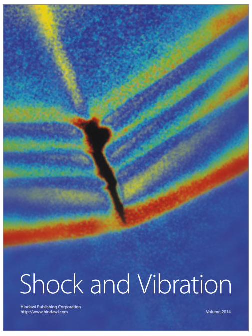

Modelling & Simulation in Engineering

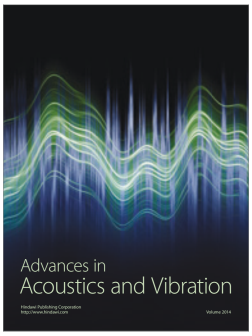
International Journal of
Aerospace Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration