Property Preserving Symmetric Encryption

Omkant Pandey[1] and Yannis Rouselakis[2]

[1] Microsoft, Redmond, USA and Microsoft Research, Bangalore, India
omkantp@microsoft.com
[2] The University of Texas at Austin
jrous@cs.utexas.edu

Abstract. Processing on encrypted data is a subject of rich investigation. Several new and exotic encryption schemes, supporting a diverse set of features, have been developed for this purpose. We consider encryption schemes that are suitable for applications such as data clustering on encrypted data. In such applications, the processing algorithm needs to learn certain properties about the encrypted data to make decisions. Often these decisions depend upon multiple data items, which might have been encrypted individually and independently. Current encryption schemes do not capture this setting where computation must be done on multiple ciphertexts to make a decision.

In this work, we seek encryption schemes which allow *public* computation of a pre-specified property P about the encrypted messages. That is, such schemes have an associated property P of fixed arity k, and a publicly computable algorithm Test, such that $\mathsf{Test}(ct_1, \ldots, ct_k) = P(m_1, \ldots, m_k)$, where ct_i is an encryption of m_i for $i = 1, \ldots, k$. Further, this requirement holds even if the ciphertexts $ct_1, \ldots, ct_k$ were generated individually and independently. We call such schemes *property preserving encryption schemes*. Property preserving encryption (PPEnc) makes most sense in the symmetric setting due to the requirement that Test is publicly computable.

In this work, we present a thorough investigation of property preserving symmetric encryption. We start by formalizing several meaningful notions of security for PPEnc. Somewhat surprisingly, we show that there exists a hierarchy of security notions for PPEnc, indexed by integers $\eta \in \mathbb{N}$, which does not collapse. We also present a symmetric PPEnc scheme for encrypting vectors in $\mathbb{Z}_N$ of polynomial length. This construction supports the orthogonality property: for every two vectors $(\vec{x}, \vec{y})$ it is possible to *publicly* learn whether $\vec{x} \cdot \vec{y} = 0 \mod p$. Our scheme is based on bilinear groups of composite order.

1 Introduction

This paper introduces the notion of *property preserving* encryption schemes. The idea is that it should be possible to *publicly* learn the properties of a massive data set, by only looking at the *encrypted* data elements. For simplicity, we model properties as boolean functions P defined over the space $\mathcal{M}^k$ for a fixed natural number $k \in \mathbb{N}$. The simplest way to capture this idea is by requiring a public algorithm, Test, such that $\forall(m_1,\ldots,m_k) \in \mathcal{M}^k$:

$$P(m_1,\ldots,m_k) = \mathsf{Test}(ct_1,\ldots,ct_k)$$

where ct_i is the encryption of m_i for every $i \in [k]$. An important observation is that the idea makes most sense only for symmetric encryption schemes, which will be the main focus of this work.[1]

Property preserving encryption represents great promise, particularly for developing *private* algorithms for data classification. Of particular interest are the applications that deal with *streaming* data. For example, consider the recipient of a data stream, who receives data-elements arriving one at a time: $m_1, m_2, \ldots$ and so on. The recipient would like to encrypt each of these elements, as they arrive, and store[2] the resulting ciphertexts on an untrusted computing facility, e.g., a public *cloud* [21,33]. The recipient can then instruct the cloud to classify and organize this data—e.g., using data clustering techniques [30,28], for the target application. Current encryption schemes fall short of dealing with this situation. This holds true even for the exotic class of schemes such as predicate encryption [31], functional encryption [15], and fully homomorphic encryption [39,24].

Order Preserving Symmetric Encryption. Property preserving encryption is directly inspired by the recent work of Boldyreva, Chenette, Lee, and O'Neill on *order preserving (symmetric) encryption* [10]. Informally speaking, an encryption scheme is order preserving if the ciphertexts preserve the order of the plaintexts; that is, if m_1, m_2 are two plaintexts integers and $m_1 \geq m_2$, then $ct_1 \geq ct_2$, where ct_1, ct_2 are encryptions of m_1, m_2 respectively. Boldyreva et al. show that order-preserving schemes cannot satisfy the usual "indistinguishability" based notions. In fact, as noted in [10,11], formulating a reasonable notion of security for order preserving encryption is a subtle and involved task. The starting point of our work was to understand the source of this difficulty, and how it affects other properties.

For this purpose, we start by generalizing the idea of preserving the order as follows. First, we do not restrict ourselves only to the ordering relation, and consider arbitrary properties. Second, we do not necessarily require the *same* relation on plaintexts and ciphertexts—e.g., the *greater than or equal to* operation. Instead, we only require a public algorithm to test this relation: $\mathsf{Test}(ct_1, ct_2) = 1$ *if and only* $m_1 \geq m_2$. With these generalizations, it turns out that there exist nontrivial properties for which we can satisfy indistinguishability-based security notions. This results in very strong and robust security guarantees.

[1] For asymmetric (or public-key) encryption, the encrypted message might be recoverable for most properties of interest, simply by using Test and the encryption algorithm. See also section 1.2 for further discussion.

[2] We note that this model is similar to the model considered by Gennaro and Rohatgi [23] for digital signatures. In particular, it is different from the "streaming algorithms" model where the stream cannot be stored, and the computations must be done in a single pass over a small sample of the stream[2,29].

1.1 Our Contribution

It quickly becomes apparent that property preserving encryption is a new notion that requires a thorough investigation. This is the focus of the current work. We present a summary of our results here.

Notions of Security. We start by defining three indistinguishability based notions of security: *(1) find-then-guess* (FtG), *(2) left-or-right* (LoR), and *(3) selective real-versus-random* (sRvR). These notions are directly based upon the work of Bellare, Desai, Jokipii, and Rogaway [5] for defining security of symmetric encryption.

In FtG-security the adversary first participates in a "find" stage in which he receives encryptions of many (adaptively) chosen messages. The adversary then selects two challenges (m_0^*, m_1^*), and receives an encryption of one of them. The adversary is supposed to "guess" which message was encrypted. In LoR-security, the adversary adaptively chooses many pairs of messages $(m_1^0, m_1^1), (m_2^0, m_2^1), \ldots$, and receives encryptions of messages $m_1^b, m_2^b, \ldots$, for a fixed bit b. The adversary is supposed to guess b. In property preserving encryption, the adversary is allowed to learn the value of the property P on various subsets of messages. Therefore we enforce the following "equality pattern" condition (assume P to be binary): in FtG game, we require that for every message m_i that was encrypted, $P(m_0^*, m_i) = P(m_1^*, m_i)$; likewise, in LoR game, we require that for every two indices (i, j): $P(m_i^0, m_j^0) = P(m_i^1, m_j^1)$.

For standard symmetric encryption, these two notions are proven equivalent using a simple hybrid experiment [5]. Quite surprisingly, we show that in case of property preserving encryption, the FtG-security is much weaker than LoR-security. There exist natural properties for which FtG can leak much more about the encrypted messages than LoR. This proof also highlights that in fact FtG is a rather subtle notion: there is a hierarchy of FtG definitions indexed by a natural number $\eta \in \mathbb{N}$, denoted FtG$^\eta$, which lie between FtG and LoR. Roughly speaking, the FtG$^\eta$ notion is like the FtG notion except that the adversary submits at most η pairs of challenges instead of just one: $(m_{0,1}^*, m_{1,1}^*), \ldots, (m_{0,\eta}^*, m_{1,\eta}^*)$. We go on to show that FtG$^\eta$ is weaker than FtG$^{\eta+1}$.

Our final indistinguishability based notion, is an adaptation of the "real-or-random" security presented in [5]. Informally, in this game the attacker submits adaptively chosen messages that form the *real* sequence of messages to an encryption oracle. The oracle either only encrypts the real message sequence or a *random* message sequence. As usual, we want that the adversary should not know which is the case. Adopting this notion to the setting of property-preserving encryption is slightly tricky, due to the equality pattern condition. When returning encryptions of a random sequence, it should be ensured that the random sequence will have the equality pattern of the real sequence. Since the real sequence is chosen adaptively based on the ciphertexts seen so far, the equality pattern of the real sequence "evolves" during the entire experiment. One way to deal with this situation is to require the adversary to select its equality pattern ξ (a binary vector) at the beginning of the game. This choice is motivated by the work on selective security for identity based encryption [18,19]. We require that the encryptions of real sequence with equality pattern ξ, look indistinguishable from a random sequence with the same pattern ξ. The resulting notion is called the *selective real-versus-random* security denoted by sRvR, and is proven equivalent to the selective version of LoR-security, denoted sLoR. The summary of relationships between these security notions is presented in figure 1.

$$\begin{tikzcd}
\textsc{FtG} \arrow[d, phantom, sloped, "\equiv"] & & & & & & \\
\textsc{FtG}^1 \arrow[r, shift right, "/" marking] \arrow[rrrdd, bend right, "/" marking] & \textsc{FtG}^2 \arrow[l, shift right] \arrow[r, shift right, "/" marking] \arrow[rrdd, "/" marking] & \cdots \arrow[l, shift right] \arrow[r, shift right, "/" marking] & \textsc{FtG}^\eta \arrow[l, shift right] \arrow[r, shift right, "/" marking] \arrow[dd, "/" marking] & \textsc{FtG}^{\eta+1} \arrow[l, shift right] \arrow[r, shift right, "/" marking] \arrow[ldd, "/" marking] & \cdots \arrow[l, shift right] \arrow[r, shift right, "/" marking] & \textsc{LoR} \arrow[l, shift right] \arrow[llldd, bend left] \\
& & & & & & \\
& & & \mathsf{s}\textsc{LoR} \arrow[d, leftrightarrow] & & & \\
& & & \mathsf{s}\textsc{RvR} & & &
\end{tikzcd}$$

Fig. 1. Relations between all security notions. Solid arrows denote implications for all properties. Cut arrows denote that there exist some properties for which the implication is false.

Our Constructions. We seek interesting properties for which provably secure constructions satisfying our security notions can be obtained. We present constructions that preserve, according to our notion, the orthogonality of encrypted vectors. More formally, let p be a prime number; we construct a property preserving scheme for $P : \mathbb{Z}_p^n \times \mathbb{Z}_p^n \to \{0, 1\}$ such that: $P(\vec{u}, \vec{v}) = 0$ if $\vec{u} \cdot \vec{v} = 0 \mod p$ and 1 otherwise.

First we observe a general approach for constructing property preserving encryption from symmetric predicate-encryption that satisfy two essential properties: (1) predicate privacy in the multi-challenge model, and (2) security in the standard model (as opposed to the *selective* models as defined in [18,19]). Shen, Shi, and Waters [41] formulated the notion of predicate privacy in symmetric encryption, and presented a construction for orthogonality testing. However, their construction is secure only in the selective-security model. At present, there are no known constructions satisfying the two requirements.

We present a new, direct construction, for preserving orthogonality. Our construction is based on composite order groups with bilinear pairings. We prove that our construction satisfies the LoR-security in the generic group model [44]; a provably secure construction in the standard model is left as an important open problem.

1.2 Related Work

Other than the works of Boldyreva et al. [10,11], the work of Bellare, Ristenpart, Rogaway, and Stegers [8] on format preserving encryption is also a related concept which ensures that the ciphertext has the same *format* as the plaintext.

Encryption schemes supporting keyword search on encrypted data are very relevant to our work. They were considered by Song, Wagner, and Perrig in the symmetric setting [45], and by Boneh, Di Crescenzo, Ostrovsky, and Persiano [14] in the public-key setting. We can view these works as testing for the equality property for a fixed keyword(s). Equality tests in symmetric setting are related to oblivious RAM techniques [37]; in the public-key setting they are related to anonymous Identity Based Encryption (IBE) [14,1,17]. Subsequent works developed schemes for complex queries such as conjunctive and range queries [25,16,42], and more efficient constructions [22].

Bellare, Boldyreva, and O'Neill [4] investigated the notion of deterministic encryption to allow search in sub-linear time. These schemes provide meaningful security guarantee only when messages are drawn from high min-entropy distributions. Subsequent works further refined this notion and provided new constructions [7,12,36].

Another notion, closely related to our work, is predicate encryption, introduced by Katz, Sahai and Waters [31], and further generalized to functional encryption [15]. In predicate encryption, messages are encrypted using a set of attributes S, and secret keys can be derived for predicates f, say K_f. A message m encrypted using S can be decrypted using K_f if and only if $f(S) = 1$. The principal difference between our notion and predicate encryption is that the latter only tests *unary* property, i.e., f works only on a single ciphertext. In contrast, property-preserving encryption is required to deal with multiple ciphertexts each generated individually and independently. Predicate encryption is a generalization of previous works on attribute-based encryption [40], further developed in [27,9,20,38,26]. Subsequent works provided improved constructions under a variety of cryptographic assumptions [31,43,41,34].

Our study of relationships between security notions of encryption schemes is inspired by initial works of Bellare, Desai, Jokipii, and Rogaway [5], and Bellare, Desai, Pointcheval, and Rogaway [6]; it has been pursued in many subsequent works since then such as [3,32], as well as previously mentioned works on deterministic encryption.

Somewhat tangentially related to our work is the notion of fully homomorphic encryption (FHE) [39], first realized by Gentry [24]. While FHE allows processing arbitrary computations on any number of ciphertexts, the resulting output is encrypted, and therefore not useful for evaluating properties.

2 Property Preserving Encryption

Standard Notation. We write $s \overset{\$}{\leftarrow} S$ to mean that s is picked uniformly at random from the set S. When multiple elements $x, y, z, \ldots$ are picked uniformly at random from S, we write $x, y, z, \ldots \overset{\$}{\leftarrow} S$. Symbols $\neg, \wedge$, and $\oplus$ denote the standard boolean operations: NOT, AND, and XOR, respectively. The set of natural numbers is denoted by $\mathbb{N}$; for $n \in \mathbb{N}$, we write by $[n]$ the set $\{1, 2, \ldots, n\}$. We will often refer to a vector directly by writing its components in order as either

$(a_1, a_2, \dots, a_n)$ or $\{a_i\}_{i=1}^n$. The security parameter is denoted by $\lambda \in \mathbb{N}$, and a function negligible in λ is denoted by $\mathsf{negl}(\lambda)$. All algorithms are assumed to have λ as an implicit input, and run in time polynomial in λ.

Property Preserving Encryption. A property-preserving symmetric encryption scheme, is just like a normal symmetric encryption scheme except that it has an associated property P and a test algorithm, Test. Algorithm Test is a *publicly computable* polynomial time algorithm which operates on ciphertexts. The goal of Test algorithm is to test if the property P is satisfied on the underlying messages of the input ciphertexts. The formal definition of *symmetric* property-preserving encryption is given below; we allow some public-parameters in the system so that Test algorithm can properly operate on the ciphertexts.

Definition 2.1. *A* symmetric property-preserving encryption *scheme, with plaintext - space $\mathcal{M}$, consists of four probabilistic polynomial-time algorithms $\Pi = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Test})$ and an associated property $P : \mathcal{M}^k \to \{0, 1\}$, such that:*

$\mathsf{Setup}(1^\lambda) \to (pp, sk)$:
: *This is a randomized algorithm, which on input a security parameter $\lambda \in \mathbb{N}$, outputs a secret-key sk, and public-parameters pp.*

$\mathsf{Enc}(pp, sk, m) \to ct$:
: *The (possibly randomized) encryption algorithm takes as input pp, sk, and the plaintext m; it outputs a ciphertext ct.*

$\mathsf{Dec}(pp, sk, ct) \to m$:
: *The decryption algorithm takes as input pp, sk, and the ciphertext ct; it outputs the plaintext message m.*

$\mathsf{Test}(pp, ct_1, \dots, ct_k) \to \{0, 1\}$:
: *The testing algorithm takes as input the public parameters pp, and k ciphertexts $ct_1, \dots, ct_k$; it outputs a bit.*

We require that for all possible outputs (pp, sk) of algorithm Setup*, and every $m \in \mathcal{M}$, it holds that $\mathsf{Dec}(pp, sk, \mathsf{Enc}(pp, sk, m)) = m$. Further, we also require that there exist a negligible function $\mathsf{negl}(\cdot)$ such that $\forall (m_1, \dots, m_k) \in \mathcal{M}^k$:*

$$\Pr\left[\begin{array}{l}\mathsf{Test}(pp, ct_1, \dots, ct_k) \\ = P(m_1, m_2, \dots, m_k)\end{array}\middle|\begin{array}{l}(pp, sk) \leftarrow \mathsf{Setup}(1^\lambda) \\ \forall i \in [k] : ct_i \leftarrow \mathsf{Enc}(pp, sk, m_i)\end{array}\right] \geq 1 - \mathsf{negl}(\lambda)$$

where the probability is taken over the randomness of all algorithms.

3 Security Notions

We follow the approach of Bellare, Desai, Jokipii, and Rogaway [5], and present three different definitions. We will start by considering the two simplest variants, each of which is obtained by modifying definitions in [5] to accommodate the equality pattern. To do this, we introduce some notation.

Notation. Let $\Pi = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Test})$ be a symmetric property-preserving encryption scheme with plaintext space $\mathcal{M}$. Let P be a k-ary property defined over $\mathcal{M}$ for some fixed positive integer $k \in \mathbb{N}$: $P : \mathcal{M}^k \to \{0,1\}$. For a bit b, let the "Left-Right Oracle" be defined as the following function: $\mathsf{LR}(m_0, m_1, b) = m_b$. Let $X = (x_1, \ldots, x_n) \in \mathcal{M}^n$ and $Y = (y_1, \ldots, y_n) \in \mathcal{M}^n$ be two message sequences of polynomial length $n = n(\lambda)$. We say that X and Y have the *same* equality pattern for property P, if and only if: $\forall (i_1, \ldots, i_k) \in [n]^k,\ P(x_{i_1}, \ldots, x_{i_k}) = P(y_{i_1}, \ldots, y_{i_k})$.

It will be convenient to formally define the equality pattern of a sequence X. For integers n, k, let $I_1, \ldots, I_{n^k}$ be *all* sequences of indices $(i_1, \ldots, i_k) \in [n]^k$ in the *lexicographic* order.[3] The equality pattern of a sequence $X \in \mathcal{M}^n$ w.r.t. property $P : \mathcal{M}^k \to \{0,1\}$ is a binary vector of length n^k, denoted by $\mathsf{Eqp}(X) := (b_1, \ldots, b_{n^k})$, such that $b_j = P(X_{I_j})$. Here X_{I_j} denotes the projection of X on j^{th}-sequence I_j, for $j \in [n^k]$.

Find-then-Guess Security. The simplest indistinguishability based definition is the "find-then-guess" security. Informally, adversary $\mathcal{A}$ participates in a game, in which first it is given access to an encryption oracle. $\mathcal{A}$ can ask polynomially many encryption queries by adaptively choosing and sending plaintexts $m \in \mathcal{M}$. This is called the "find" stage; at some point, $\mathcal{A}$ produces two equal-length messages (m_0^*, m_1^*). At this point, $\mathcal{A}$ is given a challenge ciphertext ct, which is an encryption of m_b for a random bit b. $\mathcal{A}$ can make more queries to the encryption oracle after receiving ct. At some point, $\mathcal{A}$ outputs a bit b' (as its guess of b), and the game ends. The output of the game is b'.

For convenience, we split $\mathcal{A}$, into two algorithms denoted $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2)$. Algorithm $\mathcal{A}_1$ participates in the "find" stage and outputs (m_0^*, m_1^*) and some state information st (which includes public-parameters). Algorithm $\mathcal{A}_2$ represents the actions of $\mathcal{A}$ after the find stage—$\mathcal{A}_2$ receives the challenge ciphertext ct, and the state information st, and outputs the bit b'. Formally, this game is captured by a random process, denoted $\mathsf{Game}^{\text{FTG}}_{\Pi,\mathcal{A},\lambda}(b)$, which appears in table 1. For succinctness, we adopt the convention that sk includes the public-parameters pp, and we write $\mathsf{Enc}_{sk}(m)$ to mean $\mathsf{Enc}(pp, sk, m)$.

Let the queries of $\mathcal{A}_1$ to the encryption oracle be $(m_1, \ldots, m_\ell)$, and the queries of $\mathcal{A}_2$ be $(m_{\ell+1}, \ldots, m_n)$. We say that $\mathcal{A}$ is a *valid* FTG-adversary if sequences X_0 and X_1 have the same equality pattern, where $X_0 = (m_1, \ldots, m_\ell, m_0^*, m_{\ell+1}, \ldots, m_n)$ and $X_1 = (m_1, \ldots, m_\ell, m_1^*, m_{\ell+1}, \ldots, m_n)$; that is $\mathsf{Eqp}(X_0) = \mathsf{Eqp}(X_1)$. Define the advantage of a valid FTG-adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ as follows:

$$\mathsf{Adv}^{\text{FTG}}_{\Pi,\mathcal{A},\lambda} = \left| \Pr\left[\mathsf{Game}^{\text{FTG}}_{\Pi,\mathcal{A},\lambda}(1) = 1\right] - \Pr\left[\mathsf{Game}^{\text{FTG}}_{\Pi,\mathcal{A},\lambda}(0) = 1\right] \right|$$

[3] Equivalently, every sequence is an ordered multi-set of $[n]^k$. Note that multi-set is important since the property is defined for sequences of the form $P(m, \ldots, m)$. Likewise, order is important since changing the message-order may change the value of P.

Definition 3.1 (FtG Security). *Let $\Pi = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Test})$ be a symmetric property-preserving encryption scheme with plaintext space $\mathcal{M}$ and associated property $P : \mathcal{M}^k \to \{0,1\}$ for a fixed positive integer $k \in \mathbb{N}$. We say that Π is* FtG*-secure, if there exists a negligible function* $\mathsf{negl}(\cdot)$ *such that for all probabilistic polynomial time* valid FtG*-adversaries $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, and for all sufficiently large $\lambda \in \mathbb{N}$, the advantage of $\mathcal{A}$ in game $\mathsf{Game}^{\mathrm{FtG}}_{\Pi,\mathcal{A},\lambda}(b)$ is at most* $\mathsf{negl}(\lambda)$*. That is, $\mathsf{Adv}^{\mathrm{FtG}}_{\Pi,\mathcal{A},\lambda} \leq \mathsf{negl}(\lambda)$.*

Table 1. Security games for defining the three notions—FtG, LoR, and sRvR

```
$\mathsf{Game}^{\mathrm{FtG}}_{\Pi,\mathcal{A},\lambda}(b)$

$(pp, sk) \leftarrow \mathsf{Setup}(1^\lambda)$
$(m_0^*, m_1^*, st) \leftarrow \mathcal{A}_1^{\mathsf{Enc}_{sk}(\cdot)}(pp)$
$ct^* \leftarrow \mathsf{Enc}_{sk}(m_b^*)$
$b' \leftarrow \mathcal{A}_2^{\mathsf{Enc}_{sk}(\cdot)}(st, ct^*)$
return $b'$
```

```
$\mathsf{Game}^{\mathrm{LoR}}_{\Pi,\mathcal{A},\lambda}(b)$

$(pp, sk) \leftarrow \mathsf{Setup}(1^\lambda)$
$b' \leftarrow \mathcal{A}^{\mathsf{Enc}_{sk}(\mathsf{LR}(\cdot,\cdot,b))}(pp)$
return $b'$
```

```
$\mathsf{Game}^{\mathrm{sRvR}}_{\Pi,\mathcal{A},\lambda}(b)$

$(pp, sk) \leftarrow \mathsf{Setup}(1^\lambda)$
$(\xi, st) \leftarrow \mathcal{A}_1(pp)$
$Z \xleftarrow{\$} \mathcal{S}(\xi)$
$b' \leftarrow \mathcal{A}_2^{\mathsf{Enc}_{sk}(\mathsf{LR}(\cdot,Z,b))}(pp)$
return $b'$
```

Left-or-Right Security. Define left-or-right *encryption* oracle, denoted by $\mathsf{Enc}(pp, sk, \mathsf{LR}(\cdot,\cdot,b))$, which behaves as follows. On input a pair of equal-length messages $(m_0, m_1) \in \mathcal{M}^2$, the oracle obtains message $\mathsf{LR}(m_0, m_1, b) = m_b$, and then outputs a ciphertext by computing $\mathsf{Enc}(pp, sk, m_b)$. Once again, we drop pp from the notation for succinctness, and denote this oracle by $\mathsf{Enc}_{sk}(\mathsf{LR}(\cdot,\cdot,b))$.

In this security definition, $\mathcal{A}$ participates in a game in which he gets access to $\mathsf{Enc}_{sk}(\mathsf{LR}(\cdot,\cdot,b))$ for a random b. Throughout the execution of the game, $\mathcal{A}$ adaptively submits the queries of the form (m_i^0, m_i^1) to the encryption oracle and receives $ct_i = \mathsf{Enc}_{sk}(m_i^b)$ for $i = 1, \ldots, n$ where $n = n(\lambda)$ is an arbitrary polynomial. At some point, $\mathcal{A}$ outputs a bit b' (as its guess of b), and the game ends. The output of the game is b'. Formally, this game is captured by a random process, denoted $\mathsf{Game}^{\mathrm{LoR}}_{\Pi,\mathcal{A},\lambda}(b)$, which appears in table 1. Let the queries of $\mathcal{A}$ to the oracle be $\left\{(m_i^0, m_i^1)\right\}_{i=1}^n$, and let $X_0 = (m_1^0, \ldots, m_n^0)$ and $X_1 = (m_1^1, \ldots, m_n^1)$. We say that $\mathcal{A}$ is a *valid* LoR-adversary if sequences X_0 and X_1 have the same equality pattern; that is $\mathsf{Eqp}(X_0) = \mathsf{Eqp}(X_1)$. The advantage of a valid LoR-adversary $\mathcal{A}$ is defined as before:

$$\mathsf{Adv}^{\mathrm{LoR}}_{\Pi,\mathcal{A},\lambda} = \left|\Pr\left[\mathsf{Game}^{\mathrm{LoR}}_{\Pi,\mathcal{A},\lambda}(1) = 1\right] - \Pr\left[\mathsf{Game}^{\mathrm{LoR}}_{\Pi,\mathcal{A},\lambda}(0) = 1\right]\right|$$

Definition 3.2 (LoR Security). *Let $\Pi = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Test})$ be a symmetric property-preserving encryption scheme with plaintext space $\mathcal{M}$ and associated property $P : \mathcal{M}^k \to \{0,1\}$ for a fixed positive integer $k \in \mathbb{N}$. We say that Π is* LoR*-secure, if there exists a negligible function* $\mathsf{negl}(\cdot)$ *such that for all probabilistic polynomial time* valid LoR*-adversaries $\mathcal{A}$, and for all sufficiently large $\lambda \in \mathbb{N}$, the advantage of $\mathcal{A}$ in game $\mathsf{Game}^{\mathrm{LoR}}_{\Pi,\mathcal{A},\lambda}(b)$ is at most* $\mathsf{negl}(\lambda)$*. That is, $\mathsf{Adv}^{\mathrm{LoR}}_{\Pi,\mathcal{A},\lambda} \leq \mathsf{negl}(\lambda)$.*

We note that in their work on symmetric-key predicate encryption, Shen, Shi, and Waters [41] called the FTG-security as the "single-challenge" security, and the LOR-security as the "full-security."

Real-versus-Random Security. Another interesting notion considered in [5] is that of "real-or-random" security, where the attacker instead of giving two sequences gives only one, called the *real*, sequence). In return, it either receives the encryption of the messages from real sequence, or the encryption of *random* messages (which form the *random* sequence). As discussed earlier, adopting this notion to the setting of property-preserving encryption is slightly tricky.

Recalling briefly, the real sequence allows the adversary $\mathcal{A}$ to learn its equality pattern; and therefore indistinguishability makes sense only if a random sequence with the same equality pattern is selected. However, if the real sequence is selected adaptively, its equality pattern also evolves adaptively; but since $\mathcal{A}$ must receive encryptions "on-the-fly," providing encryptions of random messages that "in-the-end" would have the same equality pattern as the real sequence may not always be possible. It is for this reason that defining a meaningful "simulation-based" definition is difficult in this setting.

Nevertheless, a meaningful definition can still be achieved if we do not allow the adversary to *adaptively* evolve the security pattern of the real sequence. That is, we consider a *static* or *selective* setting, where the $\mathcal{A}$ "announces" the equality pattern that the real sequence will have at the beginning of the game (on input the public-parameters). This is much like the the selective-ID model of [18,19].[4]

The *selective* real-versus-random security denoted by sRVR, considers a game that is identical to the game in LOR-security except for the following difference. The adversary is a pair of algorithms $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ such that $\mathcal{A}_1$ on input the public-parameters, outputs a binary vector ξ of length polynomial in λ, and a state information st (which includes public-parameters). Vector ξ represents an equality-pattern and fixes an integer $n \in \mathbb{N}$. A random sequence $Z = (z_1, \ldots, z_n) \in \mathcal{M}^n$ is chosen such that $\mathsf{Eqp}(Z) = \xi$. $\mathcal{A}$ is given access to an encryption oracle which accepts queries of the form $m \in \mathcal{M}$; upon i^{th}-query m_i, the oracle returns the value of $\mathsf{Enc}_{sk}(\mathsf{LR}(m_i, z_i, b))$. We slightly abuse the notation, and denote this special oracle by $\mathsf{Enc}_{sk}(\mathsf{LR}(\cdot, Z, b))$. This game is formally captured by a random process, denoted $\mathsf{Game}^{\text{sRvR}}_{\Pi,\mathcal{A},\lambda}(b)$, which appears in table 1. Denote by $\mathcal{S}(\xi)$ the set of all message-sequences whose equality pattern is ξ.

We say that $\mathcal{A}$ is a valid sRVR-adversary if the sequence of messages queried by $\mathcal{A}$, denoted $M \in \mathcal{M}^n$ is such that $\mathsf{Eqp}(M) = \xi$. Define the advantage $\mathsf{Adv}^{\text{sRvR}}_{\Pi,\mathcal{A},\lambda}$ and the sRVR-security of Π for a valid sRVR-adversary $\mathcal{A}$, analogous to $\mathsf{Adv}^{\text{LoR}}_{\Pi,\mathcal{A},\lambda}$ and LOR-security by replacing the word LOR with sRVR.

Remarks on the Hierarchy. As noted earlier, we show that there is a hierarchy of security notions that does not collapse. The security notion FTG$^{\eta}$ is

[4] The fact that in our model, the public-parameters *pp* are given *before* $\mathcal{A}$ decides the equality pattern does not make our model necessarily better. Indeed, *pp* are irrelevant since we are dealing with symmetric encryption; in particular, *pp* can be included simply as part of the ciphertext.

identical to FtG except that the adversary has multiple find stages, and sends exactly η pairs of challenges. Likewise, the sRvR notion reduces to the selective variant of the LoR notion, denoted sLoR: the only difference is that in sLoR definition, $\mathcal{A}$ announces the security pattern ξ of the two sequences before seeing any encryptions. Due to space constraints, the formal definitions of FtG$^{\eta}$, sLoR are given in the full version.

4 Relations among Security Notions

In this section, we will establish relationships between various notions security for symmetric property-preserving encryption (PPEnc). The main result of this section is that FtG$^{\eta}$ does not imply FtG$^{\eta+1}$. We will start with the simpler case that FtG-security does not imply LoR-security—not even the selective variants sLoR and sRvR. All other implications are rather trivial.

Informally, for a symmetric PPEnc Π for a property P, we say that LoR-security implies FtG-security, denoted LoR $\rightarrow$ FtG, to mean the following statement: "If Π satisfies LoR-security (i.e., definition 3.2) then it also satisfies FtG-security (i.e., definition 3.1)." In [5], it was shown that, for an *ordinary* symmetric encryption scheme, FtG-security and LoR-security, are in fact equivalent (up to a polynomial degradation in security). Which means that FtG implies LoR, and vice-versa. The same proof shows that FtG$^{\eta+1}$ $\rightarrow$ FtG$^{\eta}$ for every $\eta \in \mathbb{N}$.

4.1 LoR vs. FtG

First off, it is trivial to see that LoR implies FtG. In case of an ordinary[5] scheme, to simulate the FtG-game for an attacker, a simulator participates in an LoR game. To answer encryption queries of $\mathcal{A}$ (in "find" stage and after the challenge ciphertext) which consist of a single message $m \in \mathcal{M}$, the simulator can simply send a query of the form $(m, m) \in \mathcal{M}^2$ to its left-or-right-encryption oracle, and give the answer to $\mathcal{A}$. The challenge-query (m_0^*, m_1^*) can be used directly. This strategy also applies to our setting of symmetric PPEnc, with *no* change. The key observation is that the sequences sent by the simulator to the outside oracle have the same equality pattern, simply because $\mathcal{A}$ is a valid FtG-adversary. This proof is omitted, and we conclude that LoR $\rightarrow$ FtG for all P.

To prove the other direction, i.e., FtG $\rightarrow$ LoR, a simple hybrid experiment is used in [5] in which the left sequence is converted into the right sequence by changing one message at a time. While this works for an ordinary encryption scheme, this approach breaks down in case of PPEnc. In particular, in the i-th hybrid, as we change the encryption of i-th "left" message to the corresponding right message, the equality pattern may change. It might even be true that the right-sequence is not "reachable" from the left-sequence for every property P by changing one message at a time. In this case we say that the two sequences belong in different equivalence classes.

[5] That is, it is not necessarily a property-preserving encryption scheme.

Proving the Separation. To separate FtG from LoR, our goal is to think of a property P (preferably, a natural property) and an encryption scheme Π such that: P divides its message space in only a small number of equivalence classes, and Π leaks the "identity" of the equivalence class at the end of the security game. This will not break FtG-security, but by choosing two sequences with same equality pattern but different equivalence classes, LoR-security can be broken.

We will use quadratic residuosity to construct a property. For a prime number p, define by $\mathcal{QR}_p$ and $\mathcal{QNR}_p$ the set of quadratic residues and quadratic non-residues respectively in $\mathbb{Z}_p^*$. It will be convenient to define the following "sign" function $\mathcal{J}$, which outputs whether a message $m \in \mathbb{Z}_p^*$ is a quadratic residue or not:[6] if $m \in \mathcal{QR}_p$ then $\mathcal{J}(m) = 0$, otherwise (i.e., $m \in \mathcal{QNR}_p$), $\mathcal{J}(m) = 1$. For any two messages $(x, y) \in \mathbb{Z}_p^* \times \mathbb{Z}_p^*$, we define the following binary property:

$$P_{\mathrm{qr}}(x, y) = \begin{cases} 1 \text{ if } x \cdot y \in \mathcal{QR}_p \\ 0 \text{ if } x \cdot y \in \mathcal{QNR}_p \end{cases}$$

We now prove the following theorem.

Theorem 4.1 (FtG $\not\rightarrow$ LoR). *Suppose there exists a* FtG*-secure property-preserving symmetric encryption scheme Π for property P_{qr} and plaintext-space $\mathcal{M} = \mathbb{Z}_p^*$. Then there exists another property-preserving symmetric encryption scheme Π^* for property P_{qr} and plaintext space $\mathcal{M}$ such that Π^* is* FtG*-secure, but it is not* LoR*-secure.*

Proof. The key-idea in our proof is that the property P_{qr} puts a nice structure on the equality pattern of adversary's queries. We will use a one-time pad to hide crucial information about this structure in the ciphertext, which can be recovered in the LoR-game but not in the FtG-game.

Let $\Pi = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Test})$. We construct a new scheme $\Pi^* = (\mathsf{Setup}^*$, Enc^*, Dec^*, $\mathsf{Test}^*)$, whose algorithms are defined as follows.

1. The Setup^* algorithm calls $\mathsf{Setup} \rightarrow (pp, sk)$, it then picks a uniformly random bit $t \overset{\$}{\leftarrow} \{0, 1\}$. It outputs pp as the public-parameters and the secret-key is set to the pair $sk^* = (sk, t)$. The bit t will be used as a one-time pad.
2. Algorithm Enc^* encrypts an input $m \in \mathbb{Z}_p^*$ as follows. It calls $\mathsf{Enc}(pp, sk, m) \rightarrow ct$. Then it selects a uniformly random bit $b \overset{\$}{\leftarrow} \{0, 1\}$. If $b = 0$ the output ciphertext is $ct^* = (ct, b, t)$; otherwise, $b = 1$ and the ciphertext is $ct^* = (ct, b, t \oplus \mathcal{J}(m))$. Namely if $b = 0$ the ciphertext reveals the one-time pad, otherwise the XOR of the pad with the residuosity sign. Compactly, the ciphertext is $ct^* = (\mathsf{Enc}(pp, sk, m), b, t \oplus (b \wedge \mathcal{J}(m)))$.
3. The decryption algorithm, on input (ct, b, c) outputs $\mathsf{Dec}(pp, sk, ct)$. The test algorithm on input (ct_1, b_1, c_1) and (ct_2, b_2, c_2) outputs $\mathsf{Test}(pp, ct_1, ct_2)$.

It is easy to see to see that Π^* satisfies all the correctness properties if Π does. We have to show that Π^* is FtG-secure but not LoR-secure. This follows from lemmas 4.2 and 4.3. This completes the proof.

[6] This is essentially the Legendre symbol with -1 replaced by 0.

Lemma 4.2. *For every valid* FtG *adversary $\mathcal{A}$ for Π^*, there exists a valid* FtG *adversary $\mathcal{B}$ for Π such that for every $\lambda \in \mathbb{N}$,* $\mathsf{Adv}^{\mathrm{FtG}}_{\Pi^*,\mathcal{A},\lambda} = \mathsf{Adv}^{\mathrm{FtG}}_{\Pi,\mathcal{B},\lambda}$

Proof. We construct adversary (a.k.a. simulator) $\mathcal{B}$, using $\mathcal{A}$. However, before doing so, we first analyze the possible attack sequences for $\mathcal{A}$. Remember that $\mathcal{A}$ participates in an FtG-game against Π^*, and is denoted by $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$. Further, it must satisfy the equality-pattern condition.

According to the definition of the FtG game, $\mathcal{A}_1$ will query for the messages $m_1, m_2, \ldots, m_\ell$ (in the "find" phase), and output a *challenge* pair (m_0^*, m_1^*) along with some state information. Then $\mathcal{A}_2$, on input a ciphertext and the state, will query for the messages $m_{\ell+1}, m_{\ell+2}, \ldots, m_n$ (in phase 2) and output a guess. There are only two possible cases regarding the challenge pair:

Case 1: $\mathcal{J}(m_0^*) = \mathcal{J}(m_1^*)$. That is, either both messages are quadratic residues, or both are non-residues.

Case 2: $\mathcal{J}(m_0^*) \neq \mathcal{J}(m_1^*)$. That is, one message is a quadratic residue, and the other is a non-residue. Notice that in this case it holds that neither $\mathcal{A}_1$ nor $\mathcal{A}_2$ makes any queries to the encryption oracle. That is, no queries are made either in phase-1 or phase-2. Indeed, suppose that either $\mathcal{A}_1$ or $\mathcal{A}_2$ queries m and receives $ct = \mathsf{Enc}_{sk}(m)$. Then, by the properties of quadratic residues, we have that $P_{\mathrm{qr}}(m, m_0^*) \neq P_{\mathrm{qr}}(m, m_1^*)$. This violates the equality pattern condition since P_{qr} can be learned from ct and ct^* (which $\mathcal{A}_2$ receives).

Now, the adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ when participating in the FtG-game for Π, internally simulates the FtG-game for $\mathcal{A}$ (with scheme Π^*) as follows. $\mathcal{B}_1$ on input the public parameters of Π, forwards them to $\mathcal{A}_1$. $\mathcal{A}$ must follow one of the two cases above. Suppose that $\mathcal{A}$ follows Case-1. In this case, if $\mathcal{A}_1$ makes a single-message encryption query, $\mathcal{B}_1$ forwards this query to the outside encryption oracle, and gives $\mathcal{A}_1$ whatever the answer is. At some point, $\mathcal{A}_1$ outputs (m_0^*, m_1^*, st); then $\mathcal{B}_1$ also outputs this triplet and halts.

Algorithm $\mathcal{B}_2$ picks a uniformly random one-time pad $t \xleftarrow{\$} \{0,1\}$ and stores it. $\mathcal{B}_2$ receives a ciphertext ct' (and state st) as input. Note that ct' is a ciphertext of Π. To construct a ciphertext of Π^*, $\mathcal{B}_2$ picks a random bit b, and sets $ct^* = (ct', b, t)$ if $b = 0$; otherwise it sets $ct^* = (ct', b, (t \oplus \mathcal{J}(m_0^*))$. This is a correctly distributed ciphertext since $\mathcal{J}(m_0^*) = \mathcal{J}(m_1^*)$. $\mathcal{B}_2$ internally provides (ct^*, st) to $\mathcal{A}_2$. Encryption queries of $\mathcal{A}_2$ are answered by $\mathcal{B}_1$ using its encryption oracle. It is clear that the simulation is perfect.

If on the other hand $\mathcal{A}_1$ gives out at the beginning of the game a challenge pair that consists of a residue and a non residue, we are in case-2. This means that no encryption queries are made by $\mathcal{A}_1$, and none will be made by $\mathcal{A}_2$. So $\mathcal{B}_1$ also simply outputs this pair and the state information to outside experiment. Upon receiving a challenge ciphertext and state, it gives the following ciphertext to $\mathcal{A}_2$: (ct, b, c) where both b and c are uniformly random bits. The state information is also given to $\mathcal{A}_2$. In this case also the simulation is perfect, since irrespective of the value of b, c is distributed correctly as in a proper ciphertext (every value of c defines an implicit value for the one-time pad, which is information theoretically hidden since there are no other encryption queries made). This completes the proof.

Lemma 4.3. *There exists a valid polynomial-time* LoR *attacker on Π^* with advantage $1 - 2^{-n+1}$, where n is the number of queries it makes.*

Proof. The attacker proceeds as follows in the LoR-game. It sends queries such that the the left-sequence contains only quadratic-residues, while the right-sequence contains only quadratic-non-residues. Notice that this a valid pair of sequences since the equality patterns are the same with respect to property P_{qr}: the output of the property is always 1 for any pair of messages in each sequence. However if the length of each sequence is n, then with probability $q = 1 - 2 \cdot \left(\frac{1}{2}\right)^n = 1 - 2^{-n+1}$, there will be two ciphertexts (ct_1, b_1, c_1) and (ct_2, b_2, c_2) for which $b_1 \neq b_2$. In this case, the value $c_1 \oplus c_2$ reveals the residuosity-sign of one of the two streams. Since this sign is known to the attacker and it is different for the two streams, it compromises LoR-security. In the unlikely case when $b_1 = b_2$ for all ciphertexts, the attacker fails, say by outputting 0, giving us the required advantage.

Our next goal is to separate $\text{FtG}^{\eta+1}$ from FtG^{η}. The following theorem will be proven in the full version using the same property P_{qr}.

Theorem 4.4 ($\mathbf{FtG}^{\eta} \not\Rightarrow \mathbf{FtG}^{\eta+1}$). *Let $\eta \in \mathbb{N}$ be a fixed positive integer. Suppose there exists a FtG^{η}-secure property-preserving symmetric encryption scheme Π for property P_{qr} and plaintext-space $\mathcal{M} = \mathbb{Z}_p^*$. Then there exists another property-preserving symmetric encryption scheme Π^* for property P_{qr} and plaintext space $\mathcal{M}$ such that Π^* is FtG^{η}-secure, but it is not $\text{FtG}^{\eta+1}$-secure.*

5 Constructions of Property-Preserving Encryption

In this section, we present constructions of property preserving encryption (PPEnc) encryption scheme. Instead of constructing the full-fledged scheme, it suffices to construct a slightly weaker variant, called *property-preserving tag scheme* (PPTag). A PPTag scheme allows us to test the property Test, without having a decryption algorithm. We can get correct decryption by utilizing appropriately any IND − CPA secure *symmetric* encryption scheme. We refer the reader to [31,41] for this somewhat standard approach.

To start with, we note that for unary properties P, one can simply include the value of $P(m)$ in the ciphertext, to get a construction. Therefore, we focus on properties of higher arity. In the full version of the paper, we present a generic construction of PPTag for any binary property from adaptively fully secure predicate encryption[41]. The main idea of this construction is that the new encryption algorithm calls the encryption algorithm of the original predicate encryption scheme and the token generation algorithm, both with input the message m. The resulting ciphertext consists of a ciphertext part and a token part. A selectively fully secure scheme is given in [41], which is not sufficient for our LoR security definition. Therefore, we present an explicit PPEnc construction in the following section.

5.1 An Explicit Construction for Testing Orthogonality

This is a construction for testing orthogonality of two vectors. The plaintext space of our scheme is $\mathcal{M} = (\mathbb{Z}_N^* \cup \{0\})^n$ where $N = pq$ for two λ-bit primes p and q, $\mathbb{Z}_N^*$ is the set of invertible elements of $\mathbb{Z}_N$, and $n :\in \mathbb{N} \to \mathbb{N}$ polynomial in λ. [7] The associated property $P : \mathcal{M} \times \mathcal{M} \to \{0,1\}$ is such that: $P(\vec{u},\vec{v}) = 0$ if $\vec{u}\cdot\vec{v} = 0 \mod p$ and 1 otherwise. The algorithms of our scheme are the following:

- $\mathsf{Setup}(1^\lambda, n) \to (pp, sk)$: Pick two different prime numbers p, q uniformly in the range $[2^{\lambda-1}, 2^\lambda)$, where $\lambda \geq 3$. Pick a group $\mathbb{G}$ of order $N = pq$ with a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. Select two random generators g_0, g_1 for subgroups of order p and q respectively.

  Let $\mathcal{S}_n \overset{\text{def.}}{=} \{(x_1, \ldots, x_n) \in \mathbb{Z}_q^n | \sum_{i=1}^n x_i^2 \in \mathcal{QR}_q\}$ be a set of vectors with n components. Select a vector $\gamma = (\gamma_1, \ldots, \gamma_n)$ uniformly from the set $\mathcal{S}_n$. Finally, let $\delta \in \mathbb{Z}_q$ be such that $\delta^2 = \sum_{i=1}^n \gamma_i^2$ (pick one of the two at random), and $\mathbb{1}_\mathbb{G}$ be the identity element of $\mathbb{G}$. The parameters output by the algorithm are:

$$pp = (\lambda, n, N, \mathbb{G}, \mathbb{G}_T, e, \mathbb{1}_\mathbb{G}) \qquad sk = (g_0, g_1, \{\gamma_i\}_{i=1}^n, \delta)$$

- $\mathsf{Enc}(pp, sk, M) \to ct$: On input a message $M = (m_1, m_2, \ldots, m_n)$ the algorithm picks two random elements of $\mathbb{Z}_N$: $\phi, \psi \overset{\$}{\leftarrow} \mathbb{Z}_N$. It outputs the following ciphertext:

$$ct = (ct_0, \{ct_i\}_{i=1}^n) = \left(g_1^{\psi\delta}, \left\{g_0^{\phi m_i} \cdot g_1^{\psi\gamma_i}\right\}_{i=1}^n\right)$$

- $\mathsf{Test}(pp, ct^{(1)}, ct^{(2)}) \to \{0,1\}$: On input the two ciphertexts $ct^{(1)} = \left(ct_0^{(1)}, \left\{ct_i^{(1)}\right\}_{i=1}^n\right)$ and $ct^{(2)} = \left(ct_0^{(2)}, \left\{ct_i^{(2)}\right\}_{i=1}^n\right)$, the algorithm outputs 0 if and only if

$$\prod_{i=1}^n e\left(ct_i^{(1)}, ct_i^{(2)}\right) = e\left(ct_0^{(1)}, ct_0^{(2)}\right)$$

Correctness. Correctness is satisfied, except with negligible probability, due to the following:

$$\begin{aligned}\prod_{i=1}^n e\left(ct_i^{(1)}, ct_i^{(2)}\right) &= \prod_{i=1}^n e\left(g_0^{\phi^{(1)} m_i^{(1)}} \cdot g_1^{\psi^{(1)}\gamma_i}, g_0^{\phi^{(2)} m_i^{(2)}} \cdot g_1^{\psi^{(2)}\gamma_i}\right)\\ &= \prod_{i=1}^n e(g_0, g_0)^{\phi^{(1)}\phi^{(2)} m_i^{(1)} m_i^{(2)}} e(g_1, g_1)^{\psi^{(1)}\psi^{(2)}\gamma_i^2}\\ &= e(g_0, g_0)^{\phi^{(1)}\phi^{(2)}\vec{m}^{(1)}\cdot\vec{m}^{(2)}} e(g_1, g_1)^{\psi^{(1)}\psi^{(2)}\sum_i \gamma_i^2}\end{aligned}$$

[7] Since the factorization of N is not public, the plaintext space is not public. However if we assume that factoring is hard, any user that generates messages in $\mathbb{Z}_N$ will, except with negligible probability, generate a message in the correct plaintext space $\mathbb{Z}_N^* \cup \{0\}$.

$$\begin{aligned} e\left(ct_0^{(1)}, ct_0^{(2)}\right) &= e\left(g_1^{\psi^{(1)}\delta}, g_1^{\psi^{(2)}\delta}\right) \\ &= e\left(g_1, g_1\right)^{\psi^{(1)}\psi^{(2)}\delta^2} \\ &= e\left(g_1, g_1\right)^{\psi^{(1)}\psi^{(2)}\sum_i \gamma_i^2} \end{aligned}$$

In the full version, we prove that our construction satisfies LoR-security in the generic group model. We follow the terminology and proof ideas of [13] and [9]. We assume that the group elements of groups $\mathbb{G}$ and $\mathbb{G}_T$ are encoded by two random encodings $\psi, \psi_T : \mathbb{F}_N \to \{0,1\}^m$. These are injective functions that define the groups $\mathbb{G} = \{\psi(i) | i \in \mathbb{F}_N\}$ and $\mathbb{G}_T = \{\psi_T(i) | i \in \mathbb{F}_N\}$. We are also given functions to compute the group operations on $\mathbb{G}$ and $\mathbb{G}_T$ and a function that computes the non degenerate bilinear mapping e. Then, we prove the following theorem.

Theorem 5.1. *Let $\psi, \psi_T, \mathbb{G}, \mathbb{G}_T$ be as above, and let $\mathcal{A}$ be a generic algorithm, representing a valid* LoR*-adversary against the scheme described above. Further, suppose that $\mathcal{A}$ makes at most Q encryption queries, and at most W group operations and pairings counted together. Then the advantage of $\mathcal{A}$ in the* LoR*-game is at most $O\left((nQ + W)^2 \cdot 2^{-\lambda}\right)$.*

Acknowledgments. We are thankful to the Math Overflow online community, especially to the users Noam D. Elkies, GH, and Gerry Myerson, for their swift responses regarding sums of squares modulo a prime number [35], to Brent Waters for useful discussions about predicate encryption and to the anonymous reviewers for their insightful comments.

References

1. Abdalla, M., Bellare, M., Catalano, D., Kiltz, E., Kohno, T., Lange, T., Malone-Lee, J., Neven, G., Paillier, P., Shi, H.: Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 205–222. Springer, Heidelberg (2005)
2. Alon, N., Matias, Y., Szegedy, M.: The space complexity of approximating the frequency moments. In: STOC, pp. 20–29 (1996)
3. Bellare, M., Boldyreva, A., Micali, S.: Public-Key Encryption in a Multi-user Setting: Security Proofs and Improvements. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 259–274. Springer, Heidelberg (2000)
4. Bellare, M., Boldyreva, A., O'Neill, A.: Deterministic and Efficiently Searchable Encryption. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 535–552. Springer, Heidelberg (2007)
5. Bellare, M., Desai, A., Jokipii, E., Rogaway, P.: A concrete security treatment of symmetric encryption. In: FOCS, pp. 394–403 (1997)
6. Bellare, M., Desai, A., Pointcheval, D., Rogaway, P.: Relations among Notions of Security for Public-Key Encryption Schemes. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 26–45. Springer, Heidelberg (1998)

7. Bellare, M., Fischlin, M., O'Neill, A., Ristenpart, T.: Deterministic Encryption: Definitional Equivalences and Constructions without Random Oracles. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 360–378. Springer, Heidelberg (2008)
8. Bellare, M., Ristenpart, T., Rogaway, P., Stegers, T.: Format-Preserving Encryption. In: Jacobson Jr., M.J., Rijmen, V., Safavi-Naini, R. (eds.) SAC 2009. LNCS, vol. 5867, pp. 295–312. Springer, Heidelberg (2009)
9. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: IEEE Symposium on Security and Privacy, pp. 321–334 (2007)
10. Boldyreva, A., Chenette, N., Lee, Y., O'Neill, A.: Order-Preserving Symmetric Encryption. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 224–241. Springer, Heidelberg (2009)
11. Boldyreva, A., Chenette, N., O'Neill, A.: Order-Preserving Encryption Revisited: Improved Security Analysis and Alternative Solutions. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 578–595. Springer, Heidelberg (2011)
12. Boldyreva, A., Fehr, S., O'Neill, A.: On Notions of Security for Deterministic Encryption, and Efficient Constructions without Random Oracles. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 335–359. Springer, Heidelberg (2008)
13. Boneh, D., Boyen, X., Goh, E.-J.: Hierarchical Identity Based Encryption with Constant Size Ciphertext. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 440–456. Springer, Heidelberg (2005)
14. Boneh, D., Di Crescenzo, G., Ostrovsky, R., Persiano, G.: Public Key Encryption with Keyword Search. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 506–522. Springer, Heidelberg (2004)
15. Boneh, D., Sahai, A., Waters, B.: Functional Encryption: Definitions and Challenges. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 253–273. Springer, Heidelberg (2011)
16. Boneh, D., Waters, B.: Conjunctive, Subset, and Range Queries on Encrypted Data. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 535–554. Springer, Heidelberg (2007)
17. Boyen, X., Waters, B.: Anonymous Hierarchical Identity-Based Encryption (Without Random Oracles). In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 290–307. Springer, Heidelberg (2006)
18. Canetti, R., Halevi, S., Katz, J.: A Forward-Secure Public-Key Encryption Scheme. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 255–271. Springer, Heidelberg (2003)
19. Canetti, R., Halevi, S., Katz, J.: Chosen-Ciphertext Security from Identity-Based Encryption. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 207–222. Springer, Heidelberg (2004)
20. Chase, M.: Multi-authority Attribute Based Encryption. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 515–534. Springer, Heidelberg (2007)
21. Creeger, M.: Cloud computing: An overview. Queue 7, 2:3–2:4 (2009)
22. Curtmola, R., Garay, J.A., Kamara, S., Ostrovsky, R.: Searchable symmetric encryption: improved definitions and efficient constructions. In: ACM Conference on Computer and Communications Security, pp. 79–88 (2006)
23. Gennaro, R., Rohatgi, P.: How to Sign Digital Streams. In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 180–197. Springer, Heidelberg (1997)
24. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: STOC, pp. 169–178 (2009)
25. Golle, P., Staddon, J., Waters, B.: Secure Conjunctive Keyword Search over Encrypted Data. In: Jakobsson, M., Yung, M., Zhou, J. (eds.) ACNS 2004. LNCS, vol. 3089, pp. 31–45. Springer, Heidelberg (2004)

26. Goyal, V., Jain, A., Pandey, O., Sahai, A.: Bounded Ciphertext Policy Attribute Based Encryption. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part II. LNCS, vol. 5126, pp. 579–591. Springer, Heidelberg (2008)
27. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: ACM Conference on Computer and Communications Security, pp. 89–98 (2006)
28. Guha, S., Meyerson, A., Mishra, N., Motwani, R., O'Callaghan, L.: Clustering data streams: Theory and practice. IEEE Trans. Knowl. Data Eng. 15(3), 515–528 (2003)
29. Henzinger, M., Raghavan, P., Rajagopalan, S.: Computing on data streams. Technical report, SRC Palo Alto, CA (1998)
30. Jain, A., Dubes, R.: Algorithms for Clustering Data. Prentice-Hall (1988)
31. Katz, J., Sahai, A., Waters, B.: Predicate Encryption Supporting Disjunctions, Polynomial Equations, and Inner Products. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 146–162. Springer, Heidelberg (2008)
32. Katz, J., Yung, M.: Complete characterization of security notions for probabilistic private-key encryption. In: STOC, pp. 245–254 (2000)
33. Klien, M.: Six Benefits of Cloud Computing. Internet Article (2010), `http://resource.onlinetech.com/the-six-benefits-of-cloud-computing/`
34. Lewko, A., Okamoto, T., Sahai, A., Takashima, K., Waters, B.: Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 62–91. Springer, Heidelberg (2010)
35. Math Overflow. Sum of squares modulo a prime (2011), `http://mathoverflow.net/questions/69576/sum-of-squares-modulo-a-prime`
36. O'Neill, A.: Deterministic public-key encryption revisited. Cryptology ePrint Archive, Report 2010/533 (2010)
37. Ostrovsky, R.: Efficient computation on oblivious rams. In: STOC, pp. 514–523 (1990)
38. Ostrovsky, R., Sahai, A., Waters, B.: Attribute-based encryption with non-monotonic access structures. In: ACM Conference on Computer and Communications Security, pp. 195–203 (2007)
39. Rivest, R., Adleman, L., Dertouzos, M.: On data banks and privacy homomorphisms. In: Foundations of Secure Computation, pp. 169–177 (1978)
40. Sahai, A., Waters, B.: Fuzzy Identity-Based Encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005)
41. Shen, E., Shi, E., Waters, B.: Predicate Privacy in Encryption Systems. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 457–473. Springer, Heidelberg (2009)
42. Shi, E., Bethencourt, J., Chan, H.T.-H., Song, D.X., Perrig, A.: Multi-dimensional range query over encrypted data. In: IEEE Symposium on Security and Privacy, pp. 350–364 (2007)
43. Shi, E., Waters, B.: Delegating Capabilities in Predicate Encryption Systems. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part II. LNCS, vol. 5126, pp. 560–578. Springer, Heidelberg (2008)
44. Shoup, V.: Lower Bounds for Discrete Logarithms and Related Problems. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 256–266. Springer, Heidelberg (1997)
45. Song, D.X., Wagner, D., Perrig, A.: Practical techniques for searches on encrypted data. In: IEEE Symposium on Security and Privacy, pp. 44–55 (2000)