Property Preserving Symmetric Encryption Revisited

Sanjit Chatterjee[1] and M. Prem Laxman Das[2]

[1] Department of Computer Science and Automation, Indian Institute of Science
sanjit@csa.iisc.ernet.in
[2] Society for Electronic Transactions and Security, Chennai
prem@setsindia.net

Abstract. At EUROCRYPT 2012 Pandey and Rouselakis introduced the notion of property preserving symmetric encryption. Such encryption schemes may be used for checking for a property on plaintexts by running a public test on the corresponding ciphertexts. It is claimed that they hold great promise in designing private algorithms for data classification. The main contributions of their work, the authors say, are a thorough investigation of property preserving symmetric encryption and consists of two main parts. On the definitional front the paper formalizes several notions of security, establish a separation between the weaker find-then-guess and the stronger left-or-right security notions and show that there exists a hierarchy of find-then-guess notions which do not collapse. The other main contribution is a concrete left-or-right secure construction for orthogonality testing.

In this work our primary focus is a critical analysis of property preserving symmetric encryption on both these fronts – security definition and provably secure construction. The separation results of Pandey-Rouselakis are conditioned on the assumed existence of a find-then-guess secure encryption of a quadratic residue based property. We observe that this property is captured by testing equality of encryption of one-bit messages and suggest a very simple and efficient scheme for testing equality. We show that the two security notions, find-then-guess and left-or-right, effectively collapse for the equality property. On the other hand, the separation results easily generalize for the equality property. Based on these results we contextualize the question of whether the separation is an artifact or indicate some real difference between the notions of find-then-guess and left-or-right for property preserving encryption. Next we cryptanalyze the scheme for testing orthogonality described in the Pandey and Rouselakis work, which was claimed to be secure in the strongest left-or-right model. We demonstrate a simple and elegant attack on the scheme which establishes that it is not even the weakest selective find-then-guess secure.

Finally, we show that given a find-then-guess secure orthogonality-preserving encryption of vectors of length 2n, there exists left-or-right secure orthogonality-preserving encryption of vectors of length n. This result gives further credence to our already established evidence that the find-then-guess is indeed a meaningful notion of security for property-preserving encryption.

Keywords: symmetric key, property preserving encryption, predicate private encryption, bilinear pairings

1 Introduction

The question of constructing useful cryptographic schemes for securing data in the cloud [21] has attracted a lot of research during the last decade. Notions like order preserving encryption [8, 10, 9, 11], attribute-based encryption [27, 25, 22], functional encryption [17, 1, 16, 15, 6, 26] and format preserving encryption [7] are useful for this purpose. The notions of IBE [13, 20, 12] and public key encryption with keyword search [18, 38, 14, 40] deal with testing of equality. Homomorphic encryption too [23, 41, 24] plays an important role in cloud security. These schemes aim to achieve data privacy, user privacy, secure computation on encrypted data, etc., on the cloud.

At EUROCRYPT 2012 Pandey and Rouselakis [34] defined the notion of *property preserving symmetric encryption* (PPEnc), which they claimed, can be used for data clustering [28]. A PPEnc scheme is a collection of four algorithms, namely, Setup, Encrypt, Decrypt and Test. The authors also considered a simpler notion called property preserving tag scheme PPTag, where there is no decryption algorithm. According to [34], the notion of PPEnc is most useful in the symmetric key setting. So the standard approach, according to [34], is to use a semantic secure symmetric key encryption scheme to encrypt the "payload" message while the encryption algorithm of PPTag is used to create a "tag" that is used as one of the inputs to the Test to publicly check whether the message satisfies the property or not. In fact a similar approach was taken in [30, 37]. Hence the question essentially boils down to constructing a secure PPTag scheme. Motivated by Bellare et.al., [4, 5] Pandey-Rouselakis define several security notions for property preserving encryption such as find-then-guess (FtG) and the left-or-right (LoR) security. However, unlike Bellare et al. [4] who showed FtG $\longrightarrow$ LoR in the ordinary symmetric key setting, Pandey and Rouselakis claim that there is a separation between FtG and LoR notions and an hierarchy among the FtG classes that does not collapse. They use a property based on quadratic residues, called P_{qr} to establish the separation results. Finally the paper proposes a scheme for achieving orthogonality, which is claimed to be LoR secure in the generic bilinear group model.

The notions of predicate encryption and inner product encryption in both public and secret key setting [30, 39, 37, 32, 31, 33] are closely related to the notion of property preserving encryption. While Okamoto and Takashima [32] give a public fully secure predicate encryption scheme, [31, 1] consider the notion of predicate privacy in public as well as private key setting. In [16] the authors consider the subspace membership predicates and use this to provide predicate privacy for orthogonality testing in the public key setting. In [34], the authors also claim that property preserving encryption is a generalization of order preserving encryption of Boldyreva et.al., [8, 10, 9, 11].

Property preserving encryption has a direct connection with predicate private encryption [37]. Unlike predicate private encryption, a PPTag scheme does not distinguish between predicates and attributes. A PPTag scheme may be easily constructed from a predicate-only scheme by concatenating the ciphertext and the token for a given message. If one starts from a full secure predicate-only scheme, one obtains a LoR secure PPTag scheme [34, 1].

Property preserving symmetric encryption is an interesting new concept with potential practical application for outsourcing computation and it is related to several other primitives like order preserving encryption and predicate encryption. Hence it is imperative that this notion should be critically examined from the definitional perspective. In particular, the separation results of [34] are surprising given the equivalence of the two notions in the symmetric key setting [4]. Furthermore, because of the separation, designers working on the problem of constructing property preserving encryption for various properties should now aim at the strongest LoR notion which may take considerably more resources than a scheme that achieves FtG security. Similarly, it is equally important to cryptanalyze the proposed provably secure construction in order to assess the concrete security guarantee for a particular property of interest, such as orthogonality. This motivates us for a critical evaluation of the Pandey-Rouselakis work.

Our Contributions. Pandey-Rouselakis formulated the notion of property preserving encryption and its security notions in the abstract setting of a general k-ary property. Based on the assumed existence of a PPEnc for the binary property called quadratic residuosity, they established their separation results. Finally, they proposed LoR secure construction for the binary property of orthogonality.

In §3, we perform a systematic analysis of the Pandey-Rouselakis separation results. As no concrete construction was suggested to validate the separation results, we first attempt to build such a scheme. Here the first observation is that the quadratic residuosity property used in the separation results of [34], is captured by a property-preserving test of equality. Hence we focus on equality property and show that one-time pad is sufficient to achieve FtG security for equality-preserving encryption of one-bit messages. Furthermore, the two notions of FtG and LoR security in fact collapse in such a *deterministic* setting. This result is then generalized for equality testing of n-bit messages where we show a pseudo-random permutation is sufficient to achieve the strongest LoR security. So, on one hand we can easily generalize the separation results of [34] for the equality property, on the other we show that in concrete terms the two notions of FtG and LoR effectively collapse for the same property. Thus contextualized, we note that the question of whether the separation results of [34] actually indicate any real world difference between the two notions of FtG and LoR security for property-preserving encryption still remains open.

In §4 we cryptanalyze the security of Pandey and Rouselakis scheme for testing orthogonality. We show that the PPEnc scheme given in [34, Sec. 5] for testing orthogonality property is not even weakest selective find-then-guess secure. This falsifies the claim [34, Theorem 5.1] that the proposed construction is strongest left-or-right secure in the generic group model.

While the (crypt)analysis of the security notions and provably secure construction of property-preserving encryption from [34] are the primary contribution of our work, as a secondary contribution in §5, we look at the relation of FtG and LoR in the context of orthogonality property. We show that given an FtG secure orthogonality-preserving encryption of vectors of length $2n$, there exists LoR secure orthogonality-preserving encryption of vectors of length n. This result gives further credence to our already established evidence that the FtG notion is a meaningful notion of security for property-preserving encryption.

We draw our conclusion in §6. Some of the detailed proofs are provided in the Appendix.

2 Preliminaries and Notions

In this section, we recall the basic definition of property preserving encryption and notions of its security from [34].

As in [34], we too model any k-ary property on $\mathcal{M}$ as a Boolean function on $\mathcal{M}^k$. One of the main properties considered is orthogonality, which depends on computing inner products in finite dimensional vector spaces over finite fields. Let $v = (v_1, \ldots, v_n)$ and $w = (w_1, \ldots, w_n)$ be vectors over a finite field $\mathbb{F}_q$. The inner product between them is defined as $v \cdot w = v_1 w_1 + \ldots + v_n w_n \pmod{q}$. These vectors are *orthogonal* if $v \cdot w = 0$.

Definition 1 *A property preserving encryption scheme for the k-ary property P is a collection of four probabilistic polynomial time (PPT) algorithms, which are defined as follows:*

1. Setup(λ)*: This takes as input the security parameter λ and outputs the message space ($\mathcal{M}$), public parameters (PP) and the secret key (SK).*

2. $\mathsf{Encrypt}(PP, SK, m)$*: This algorithm outputs the ciphertext CT corresponding to the message m, using the secret key SK and public parameter PP.*
3. $\mathsf{Decrypt}(PP, SK, CT)$*: This algorithm outputs the plaintext message m.*
4. $\mathsf{Test}(CT_1, \ldots, CT_k, PP)$*: This is a public algorithm that takes as inputs ciphertexts corresponding to messages $m_1, \ldots, m_k$ and outputs a bit.*

These set of four algorithms must satisfy the standard correctness requirement. In addition, if the Test *algorithm outputs 1 then, except with negligible probability, one has $P(m_1, \ldots, m_k) = 1$.*

A related notion of PPTag scheme was also defined. Informally, such a scheme does not have any decrypt module.

Definition 2 *A property preserving tag scheme (*PPTag*) for the k-ary property P is a collection of three probabilistic polynomial time (PPT) algorithms, which are defined as follows:*

1. $\mathsf{Setup}(\lambda)$*: This takes as input the security parameter λ and outputs the message space ($\mathcal{M}$), public parameters (PP) and the secret key (SK).*
2. $\mathsf{Encrypt}(PP, SK, m)$*: This algorithm outputs the ciphertext CT corresponding to the message m, using the secret key SK and public parameter PP.*
3. $\mathsf{Test}(CT_1, \ldots, CT_k, PP)$*: This is a public algorithm that takes as inputs ciphertexts corresponding to messages $m_1, \ldots, m_k$ and outputs a bit.*

This set of algorithms must satisfy the standard correctness requirement. If the Test *algorithm outputs 1 then, except with negligible probability, one has $P(m_1, \ldots, m_k) = 1$.*

Remark 1 *In [34], the authors suggest the following strategy while designing a secure property preserving encryption scheme. The actual "payload" message is encrypted using an IND-CPA secure symmetric encryption scheme. For testing the property, a tag is constructed for each message using a* PPTag *scheme. This idea is similar to the predicate-only encryption schemes [30, 37].*

2.1 Security Notions

Inspired by the study of security notions of symmetric key encryption by Bellare et al. [4], Pandey and Rouselakis [34] propose several notions of security for property-preserving symmetric encryption. These notions are defined by taking into account the specific nature of PPEnc. Here we informally describe the two notions of security for such schemes which are most relevant to our work. For more details refer to [34].

Definition 3 *For a k-ary property P, any two sequences $X = (x_1, \ldots, x_n)$ and $Y = (y_1, \ldots, y_n)$ of inputs are said to have the same equality pattern if*

$$P(x_{i_1}, \ldots, x_{i_k}) = P(y_{i_1}, \ldots, y_{i_k}),\ \forall\ (i_1, \ldots, i_k) \in [n]^k.$$

Find-then-Guess Security (FtG). In this game the adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, plays the following game $\mathsf{Game}^{\mathsf{FtG}}_{\Pi,\mathcal{A},\lambda}$. After the Setup phase, in $\mathcal{A}_1$, the adversary first adaptively queries the encryption oracle for messages $(m_1, \ldots, m_t)$. Then the adversary outputs the challenge messages (m_0^*, m_1^*). In $\mathcal{A}_2$, after the challenger returns the ciphertext c_b^* where $b \in_R \{0,1\}$, the adversary again adaptively queries $(m_{t+1}, \ldots, m_q)$. The adversary wins the game if s/he can correctly predict the bit b. In order to ensure that the adversary cannot trivially win the game, the adversarial queries must satisfy the *extra* condition that the equality patterns of $(m_1, \ldots, m_t, m_0^*, m_{t+1}, \ldots, m_q)$ and $(m_1, \ldots, m_t, m_1^*, m_{t+1}, \ldots, m_q)$ are the same. The game is formally defined in [34, Sec. 3]. The advantage of the adversary is formally defined as follows.

Definition 4 *Let $\Pi = \mathsf{Setup}, \mathsf{Encrypt}, \mathsf{Decrypt}, \mathsf{Test}$ be a symmetric key property preserving encryption scheme (Defn. 1). Then Π is said to be FtG secure if there exists a negligible function $n(\cdot)$ such that for all PPT FtG adversaries $\mathcal{A}$ as above and for all $\lambda \in \mathbb{N}$ sufficiently large, the advantage of $\mathcal{A}$ in the $\mathsf{Game}^{\mathsf{FtG}}_{\Pi,\mathcal{A},\lambda}$ is negligible:*

$$\mathsf{Adv}^{\mathsf{FtG}}_{\Pi,\mathcal{A},\lambda} = \left| P\left[\mathsf{Game}^{\mathsf{FtG}}_{\Pi,\mathcal{A},\lambda}(1) = 1\right] - P\left[\mathsf{Game}^{\mathsf{FtG}}_{\Pi,\mathcal{A},\lambda}(0) = 1\right]\right| \leq n(\lambda).$$

The authors further introduce a hierarchy in the FtG notion depending on the number of challenge queries. In particular, any adversary playing the FtG^η game, for $\eta \in \mathbb{N}$, is allowed to make η many challenge queries interleaved between encryption oracle queries. A *selective* FtG notion may be defined in the usual way, following [12], where the adversary outputs the challenge messages even before receiving the public parameters.

Left-or-Right Security (LoR). In this game the adversary $\mathcal{A}$ plays the following game $\mathsf{Game}^{\mathsf{LoR}}_{\Pi,\mathcal{A},\lambda}$. After setup, the adversary $\mathcal{A}$ makes q encryption queries, where each query is of the form $(m_0^{(i)}, m_1^{(i)})$. The queries are such that $(m_0^{(1)}, \ldots, m_0^{(q)})$ and $(m_1^{(1)}, \ldots, m_1^{(q)})$ have the same equality pattern. The challenger returns the encryption of $m_b^{(i)}$ for each i where $b \in_R \{0,1\}$ is chosen at the beginning of the game. At the end, the adversary has to output a guess b' of b and wins if $b' = b$. The game is formally defined in [34, Sec. 3]. The definition of adversarial advantage is as follows.

Definition 5 *Let $\Pi = \mathsf{Setup}, \mathsf{Encrypt}, \mathsf{Decrypt}, \mathsf{Test}$ be a symmetric key property preserving encryption scheme (Defn. 1). Then Π is said to be LoR secure if there exists a negligible function $n(\cdot)$ such that for all PPT LoR adversaries $\mathcal{A}$ as above and for all $\lambda \in \mathbb{N}$ sufficiently large, the advantage of $\mathcal{A}$ in the $\mathsf{Game}^{\mathsf{LoR}}_{\Pi,\mathcal{A},\lambda}$ is negligible:*

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

3 Security Notions: A Closer Look

We recall the separation results obtained by Pandey-Rouselakis [34] regarding the above security notions. In [34, Theorem 4.1], they state that $\mathsf{FtG} \nrightarrow \mathsf{LoR}$. Further in [34, Theorem 4.4], they claim that $\mathsf{FtG}^\eta \nrightarrow \mathsf{FtG}^{\eta+1}$.

Let $\mathcal{QR}_p$ (resp. $\mathcal{QNR}_p$) be the set of quadratic residues (res. quadratic non-residues) in $\mathbb{Z}_p^*$ for some prime p. Both the separation results in [34] are established through a PPEnc scheme for the quadratic residuosity property P_{qr} defined as follows:

$$P_{qr}(m_1, m_2) = \begin{cases} 1 \text{ if } m_1 \cdot m_2 \in \mathcal{QR}_p \\ 0 \text{ if } m_1 \cdot m_2 \in \mathcal{QNR}_p \end{cases} \tag{1}$$

Assuming that there *exists* an FtG secure PPEnc scheme Π for P_{qr}; Pandey-Rouselakis constructs another scheme Π' which they show is FtG but not LoR secure. Naturally the separation is conditioned on the fact that there exists such an FtG secure scheme Π. However, no such construction was known or suggested in [34].

3.1 Property Preserving Encryption for Equality

As already observed, in the symmetric key setting the actual payload message can be efficiently encrypted using a secure symmetric key encryption scheme. So the problem of constructing a PPEnc for P_{qr} is reduced to the problem of constructing a PPTag scheme for the same property.

Claim 1 *To construct a* PPTag *scheme for the property* P_{qr}*; it suffices to construct a* PPTag *scheme for equality where the message space is* $\mathcal{M} = \{0, 1\}$.

Proof. The argument is quite straightforward. A "sign" function $\mathcal{S}$ was used by [34] to define P_{qr} where $\mathcal{S}(m) = 0$ if $m \in \mathcal{QR}_p$; else $\mathcal{S}(m) = 1$. In other words, P_{qr} divides the message space $\mathcal{M} = \mathbb{Z}_p^*$ into 2 equivalence classes. Given any message in $\mathbb{Z}_p^*$ one can efficiently determine $\mathcal{S}(m)$ and then use the PPTag scheme for equality over the message space $\{0, 1\}$ to encrypt $\mathcal{S}(m)$. Testing whether the product of two messages x and y is a quadratic residue or not is now reduced to the task of testing whether $\mathcal{S}(x)$ and $\mathcal{S}(y)$ are equal or not.

Remark 2 *The property* P_{qr} *used in [34] is a particular instance of a larger class of property* $\mathcal{P}$*. In particular, the property* $\mathcal{P}$ *induces an equivalence relation on a set* $\mathcal{M}$ *such that there exists an efficient algorithm to determine the class in which a given element lies. Another example of such property is to test, given two integers* m *and* n*, whether their difference is divisible by a fixed prime* p*. It is easy to see that a* PPTag *scheme for such a property* $\mathcal{P}$ *can be realized by any* PPTag *scheme for equality. Note, however, that there do exist equivalence relations for which the question of membership testing is not known to be easy.*

Given the above result we henceforth focus on the question of constructing an FtG secure PPTag or, more generally, a PPEnc scheme for equality. Interestingly, the one-time pad turns out to be sufficient to realize such a construction.

PPEnc ***for equality over*** $\{0, 1\}$. We describe a PPEnc scheme for testing equality over message space $\{0, 1\}$.

1. $\mathsf{Setup}(1^\lambda)$: Set $SK = t$, where $t \in_R \{0, 1\}$.
2. $\mathsf{Encrypt}(SK, m)$: $CT(m) = t \oplus m$.
3. $\mathsf{Decrypt}(SK, C)$: $m' = C \oplus t$.
4. $\mathsf{Test}(CT_1, CT_2)$: Return 1 if and only if $CT_1 = CT_2$.

It is well-known that as a symmetric key encryption scheme the above construction (or any deterministic encryption scheme) is not FtG secure in the sense of [4] but it is as a PPEnc as the following claim shows.

Claim 2 *The above construction is an* FtG *secure* PPEnc *for one-bit messages.*

Proof. The key idea is that an FtG adversary $\mathcal{A}$ is restricted by the equality pattern. If $\mathcal{A}$ makes the challenge query as $(0,1)$ then s/he cannot make any encryption oracle query. Hence, the one-time pad ensures the challenge bit is information theoretically hidden from $\mathcal{A}$. On the other hand, if the challenge query is of the form $(0,0)$ or $(1,1)$ then there is no non-trivial information for $\mathcal{A}$ to learn either from the encryption queries or from the challenge. □

Deterministic PPEnc ***for equality over*** $\{0,1\}$***.*** The above result further leads us to the following interesting consequence. Let $E : \mathcal{K} \times \{0,1\} \longrightarrow \{C_0, C_1\}$ be a deterministic encryption scheme.

Claim 3 *If E is* FtG *secure* PPEnc *scheme for equality then it is* LoR *secure.*

Proof. Let $\mathcal{A}$ be a valid LoR adversary for E. We will construct a valid FtG adversary $\mathcal{B}$ for E, which is playing the FtG game with its own challenger $\mathcal{C}$ by internally running $\mathcal{A}$.

Observe that $\mathcal{A}$ has to respect the equality pattern and hence can only make queries from the following disjoint sets: $S_1 = \{(0,0),(1,1)\}$ and $S_2 = \{(0,1),(1,0)\}$. If $\mathcal{A}$ makes queries from the set S_1, then $\mathsf{FtG} \longrightarrow \mathsf{LoR}$ holds trivially.

Now let us analyze the case when $\mathcal{A}$ makes queries from $S_2 = \{(0,1),(1,0)\}$. Let's, without loss of generality, assume that $\mathcal{A}$'s first query is $(0,1)$. $\mathcal{B}$ sets the same message $(0,1)$ as its own FtG challenge query, forwards it to $\mathcal{C}$. In response $\mathcal{C}$ provides a challenge ciphertext C_b to $\mathcal{B}$, $b \in \{0,1\}$ by encrypting $\beta \in_R \{0,1\}$ using the encryption function E as per the rule of the FtG game. $\mathcal{B}$ forwards the same C_b to $\mathcal{A}$. Note that by the definition of FtG security, $\mathcal{B}$ cannot make any other query to $\mathcal{C}$. However, if $\mathcal{A}$ repeats the same query $(0,1)$, then $\mathcal{B}$ simply forwards the same ciphertext C_b. If $\mathcal{A}$ queries the other valid message pair $(1,0)$, then $\mathcal{B}$ returns ciphertext C_{1-b}. When $\mathcal{A}$ outputs a bit as its guess and halts, then $\mathcal{B}$ outputs the same bit to $\mathcal{C}$ and halts.

The simulation of $\mathcal{A}$'s environment by $\mathcal{B}$ is perfect. In fact, after the first query, $\mathcal{A}$ can on its own generate the response for all other queries it is going to make. Now the FtG security of E ensures that the encryption of 1 is indistinguishable from the encryption of 0. Hence, the advantage of $\mathcal{B}$ is the same as that of $\mathcal{A}$ and the two notions actually collapse. □

As a consequence we note that the one-time pad construction of PPEnc achieves LoR security. However, it's well-known that the same is not even FtG secure as standard symmetric key encryption scheme. Thus there exists binary property preserving encryption scheme secure in the strong LoR sense of property preserving encryption but does not even achieve FtG security as a standard symmetric key encryption scheme.

Based on our previous observations we suggest the following direct construction of LoR secure PPEnc for equality testing on $\mathcal{M} = \{0,1\}^n$. A PPTag can be obtained by dropping the $\mathsf{Decrypt}$ algorithm from the description.[3]

[3] Similar construction for testing equality in the context of authenticated encryption and searchable encryption schemes was suggested earlier by Rogaway-Shrimpton [36] and Amanatidis et al. [2]. Their constructions used deterministic MAC which is modeled as a PRF.

PPEnc **for Equality Using a PRP.** We describe a scheme Π to test for equality of strings of length n.[4] Let $\{\mathcal{F}\}_n$ be a pseudo-random permutation (PRP) family and an element $F \in \{\mathcal{F}\}_n$ is defined as $F : \{0,1\}^n \times \{0,1\}^n \longrightarrow \{0,1\}^n$.

1. $\mathsf{Setup}(1^\lambda)$: Set a random n-bit binary string K as the secret key SK.
2. $\mathsf{Encrypt}(SK, m)$: $CT_m = F_K(m)$.
3. $\mathsf{Decrypt}(SK, CT)$: Return $F_K^{-1}(CT)$.
4. $\mathsf{Test}(CT_1, CT_2)$: Return 1 if and only if $CT_1 = CT_2$.

Claim 4 *If the underlying PRP family is secure, then Π is* LoR *secure.*

Proof. (Sketch) The claim is established through a simple hybrid argument. Let the adversary for the LoR game $\mathcal{A}$ set $(m^*_{0,1}, m^*_{1,1}), \ldots, (m^*_{0,t}, m^*_{1,t})$ as challenges. We claim that the games $\text{Game}_0 : m^*_{0,1}, \ldots, m^*_{0,t}$ and $\text{Game}_1 : m^*_{1,1}, \ldots, m^*_{1,t}$ are indistinguishable. We note that, by the security of the PRP, the Game_0 is indistinguishable from a game where the challenger computes the response from a random permutation. Similarly, challenges output in Game_1 will be indistinguishable from the output of a random permutation. □

3.2 Separation Between FtG and LoR Notions for Equality

After establishing the existence of natural $\mathsf{PPEnc}/\mathsf{PPTag}$ scheme for equality testing satisfying LoR security (and, hence, FtG security), we now generalize the result of [34] (Theorem 4.1) to show that the separation holds for the equality property and need not necessarily be restricted to small number of equivalence classes. Let $\mathcal{M}$ be the message space. Suppose $z = \lceil \log_2 |\mathcal{M}| \rceil$ so that every element $m \in \mathcal{M}$ can be represented by a bit string of length z. Note that z (and not $\mathcal{M}$) is a polynomial in the security parameter. Let $\Pi = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Test})$ be any FtG secure PPTag scheme for equality. From this scheme we construct another scheme $\Pi' = (\mathsf{Setup}', \mathsf{Enc}', \mathsf{Test}')$ for realizing the same property. The construction uses a PRF family $\mathcal{F} : \{0,1\}^\kappa \times \{0,1\}^z \longrightarrow \{0,1\}^z$.[5]

1. $\mathsf{Setup}'(\lambda)$: Calls Setup of Π to obtain (PP, SK) and chooses $k \in_R \{0,1\}^\kappa$ (as the key for the PRF). The algorithm outputs PP as the public parameters for Π' and sets the secret key as $SK' = (SK, k)$.
2. $\mathsf{Enc}'(PP, SK', m)$: While encrypting $m \in \mathcal{M}$, the encryption algorithm of Π is used to obtain $ct = \mathsf{Enc}(PP, SK, m)$. Then choose a bit $b \in_R \{0,1\}$. The ciphertext of Π' is computed as
$$CT = \begin{cases} (ct, b, F_k(m)), & \text{if } b = 0, \\ (ct, b, F_k(m) \oplus m), & \text{otherwise.} \end{cases}$$
3. $\mathsf{Test}'(CT_1, CT_2, PP)$: Given $CT_1 = (ct_1, b_1, t_1)$ and $CT_2 = (ct_2, b_2, t_2)$, the algorithm outputs $\mathsf{Test}(PP, ct_1, ct_2)$.

[4] For the case of PPTag there is no need to decrypt and hence the construction can be extended to arbitrary length messages by the use of a CRHF H with n-bit digests.

[5] The PRF can be replaced by a set of $|\mathcal{M}|$ random bit strings when $|\mathcal{M}|$ is *small* (i.e., polynomial in the security parameter). On the other hand, for arbitrary length messages one can use a collision resistant hash function (CRHF) H to first map the message to a digest of z-bit and then apply the PRF on the digest.

The following two lemma generalizes the result of [34] (we provide the detailed proofs in the Appendix A) and together establish that the separation result for FtG and LoR holds for the equality property.

Lemma 5 *If the scheme Π is FtG secure and $\mathcal{F}$ is a secure PRF then Π' constructed as above is also FtG secure. In particular, $\epsilon_{\Pi'} \leq \epsilon_{\Pi} + 2\epsilon_{\mathcal{F}}$ where ϵ_X denotes the advantage in the corresponding security game for the primitive $X \in \{\Pi, \mathcal{F}\}$.*

Lemma 6 *There is a LoR adversary for the scheme Π' with non-negligible advantage.*

Remark 3 *We point out an interesting consequence of the above separation result. Shen-Shi-Waters [37] proposed two security notions, the single challenge and full challenge security for predicate private encryption (see [37] for the definitions of security). The strategy outlined in Lemma 5 and Lemma 6 in the context of PPTag can be adapted to establish a similar separation between single challenge and full challenge security of predicate encryption. Suppose we are given a single challenge secure predicate private scheme for equality, called Ψ. From that we construct another scheme Ψ' where the only changes are in the Setup and $\mathsf{Encrypt}$ as described in the context of Π' above. In particular, the encryption algorithm of Ψ' outputs a ciphertext of Ψ together with either $(b, F_k(m))$ or $(b, F_k(m) \oplus m)$ depending upon whether $b = 0$ or $b = 1$. A similar argument as in the case of PPTag above shows that Ψ' is single challenge secure but not full secure.*

Hierarchy Among FtG Classes: We briefly comment on the separation result for the hierarchy among FtG classes given in [34]. The equality property is used to establish this result. We start with a scheme Π which is FtG^{η} secure. Then we derive a scheme Π' which is not $\mathsf{FtG}^{\eta+1}$ secure. We follow the same notations used while proving the previous separation result. Encryption algorithm[6] of Π' chooses $b \in_R \{1, \ldots, \eta\}$ and returns

$$\Pi'.CT(m) = \begin{cases} (\Pi.CT(m), b, F_k(m, b)), & \text{if } 1 \leq b \leq \eta \\ (\Pi.CT(m), b, F_k(m, 1) \oplus \ldots \oplus F_k(m, \eta) \oplus m), & \text{if } b = \eta + 1. \end{cases}$$

The derived scheme Π' is not $\mathsf{FtG}^{\eta+1}$ secure, but FtG^{η} secure. The proof of FtG^{η} security is similar to the previous separation result, argued using a sequence of hybrids. For completeness, in Appendix B we provide the argument for the case where there are polynomial (in security parameter) many messages. Note that [34] does not provide an explicit proof for their hierarchy result for the property P_{qr} in the paper.

3.3 Contextualizing the Separation

Based on our above observations we argue that the Pandey-Rouselakis separation results of [34, Theorem 4.1,Theorem 4.4] give only a partial answer to the question of the relation between FtG and LoR in the context of property-preserving encryption.

To better understand the real world difference between FtG and LoR for PPEnc it is worth studying them in the context of concrete natural properties. First, look at the unary

[6] For simplicity we assume that the length of m and b together is z-bit. If necessary one can use a CRHF to achieve this.

properties. It is suggested [34] that for any unary property P, one can trivially obtain a PPTag by providing $P(m)$ in the clear as part of the ciphertext. (It is assumed that the actual message will be encrypted by a semantically secure symmetric key encryption scheme.) We note that in such a scenario, the two notions FtG and LoR actually collapse. In fact, there is no non-trivial information to be gained by the adversary by participating in the FtG or LoR security game.

Next, look at the binary property of equality. We showed that equality captures the quadratic residuosity property P_{qr} used in [34] and that any deterministic one-bit encryption scheme that is FtG-secure is sufficient to construct LoR secure PPTag for P_{qr}. In general a PRP is sufficient to construct an LoR secure PPEnc for equality. Our results seem to indicate that when it comes to the question of constructing a practical scheme for P_{qr} (or equality property, in general), then effectively there is no real difference between FtG and LoR notions of security. In Section 5 we will see that for the orthogonality property any FtG secure PPEnc for vectors of length $2n$ gives a LoR secure PPEnc for length n which provides further evidence that FtG is a meaningful notion of security for PPEnc.

We end this section with the following open question: is there a "natural" construction of a scheme for testing equality or, for that matter, any other natural property, which is FtG secure but not LoR secure. Resolving this question will shed further light into the usefulness of the hierarchy of the security notions introduced in [34].

4 Attack on Pandey and Rouselakis Scheme for Orthogonality

In [34], a PPTag scheme for testing the orthogonality property of two vectors is proposed. The construction is in the composite order bilinear group setting and claimed to achieve LoR security in the generic group model. The security claim is established in [34, Theorem 5.1] with a precise bound on the adversarial advantage. Here, we first reproduce the PPTag scheme described in [34] and then discuss its insecurity. We show that this scheme is not even selective FtG secure.

- **Setup**(λ, n). Pick two different primes p and q uniformly in the range $(2^{\lambda-1}, 2^{\lambda})$ where λ is the security parameter. Let $\mathbb{G}$ and $\mathbb{G}_T$ be two groups of order $N = pq$ such that there is an efficiently computable bilinear map $e : \mathbb{G} \times \mathbb{G} \longrightarrow \mathbb{G}_T$. Select a vector $(\gamma_1, \ldots, \gamma_n) \in \mathbb{Z}_q$ such that $\sum_{i=1}^{n} \gamma_i^2 = \delta^2 \pmod{q}$. Let g_0 (resp. g_1) be a generator of a subgroup of order p (resp. q) of $\mathbb{G}$. Set the message space as $\mathcal{M} = (\mathbb{Z}_N^* \bigcup \{0\})^n$. Set
$$PP = \langle n, N, G, G_T, e \rangle, \quad SK = \langle g_0, g_1, \{\gamma_i\}, \delta \rangle,$$
- **Encrypt**(PP, SK, M). On input a message $M = (m_1, \ldots, m_n)$, select two random elements ϕ and ψ from $\mathbb{Z}_N$. The ciphertext is computed as
$$CT = (ct_0, \{ct_i\}_{i=1}^n) = \left(g_1^{\psi\delta}, \{g_0^{\phi m_i} \cdot g_1^{\psi\gamma_i}\}_{i=1}^n \right).$$
- **Test**$(PP, CT^{(1)}, CT^{(2)})$. When two ciphertexts $CT^{(1)} = (ct_0^{(1)}, \{ct_i^{(1)}\}_{i=1}^n)$ and $CT^{(2)} = (ct_0^{(2)}, \{ct_i^{(2)}\}_{i=1}^n)$ are input, the algorithm outputs 1 if and only if:
$$\prod_{i=1}^{n} e(ct_i^{(1)}, ct_i^{(2)}) = e(ct_0^{(1)}, ct_0^{(2)}).$$

We show that the construction of [34, Section 5] is not even FtG secure, and hence, by implication, cannot be LoR secure. This contradicts the claim of [34, Theorem 5.1]. In fact, the PPTag scheme of [34] does not even satisfy the weaker *selective* notion of FtG security.

Intuition: Recall that in Setup, the user chooses secret key components $\gamma_1, \ldots, \gamma_n, \delta \in \mathbb{Z}_q$ such that $\delta^2 = \gamma_1^2 + \ldots + \gamma_n^2 \bmod q$. Now observe that for such values, we have

$$\delta^2 = \gamma_1(\gamma_1 + \gamma_2) + \gamma_2(\gamma_2 - \gamma_1) + \gamma_3^2 + \ldots + \gamma_n^2 \bmod q. \tag{2}$$

4.1 Attack for $n = 2$ case

The above intuition (Eqn. 2) immediately gives an attack on the PPTag scheme of [34] for testing orthogonality. We first describe the attack for the case $n = 2$ with concrete challenges and queries.

(i) $\mathcal{A}$ outputs the challenge vectors $\boldsymbol{w}_0^* = (0, 1)$ and $\boldsymbol{w}_1^* = (1, 0)$.

(ii) In the FtG game $\mathcal{A}$ receives the public parameter PP from its challenger $\mathcal{S}$.

(iii) $\mathcal{A}$ asks for encryption of $\boldsymbol{v}_1 = (1, 1)$ and obtains $(C_0, C_1, C_2) = (g_1^{\psi\delta}, g_0^{\phi} g_1^{\psi\gamma_1}, g_0^{\phi} g_1^{\psi\gamma_2})$, where $\phi, \psi \in_R \mathbb{Z}_N$ are chosen by the challenger (unknown to $\mathcal{A}$). From the obtained ciphertext (C_0, C_1, C_2), $\mathcal{A}$ computes the following:

$$\xi = (\xi_0, \xi_1, \xi_2) = (C_0, C_1 \cdot C_2, C_2/C_1) = (g_1^{\psi\delta}, g_0^{2\phi} g_1^{\psi(\gamma_1+\gamma_2)}, g_1^{\psi(\gamma_2-\gamma_1)}). \tag{3}$$

(iv) The challenger returns the encryption $C_{\boldsymbol{w}_b^*}$, for a bit $b \in_R \{0, 1\}$. We shall denote the ciphertext of $\boldsymbol{w}_b^* = (m_1, m_2)$ by $C_{\boldsymbol{w}_b^*}$ where

$$C_{\boldsymbol{w}_b^*} = (\zeta_0, \zeta_1, \zeta_2) = (g_1^{\psi_1\delta}, g_0^{\phi_1 m_1} g_1^{\psi_1\gamma_1}, g_0^{\phi_1 m_2} g_1^{\psi_1\gamma_2}). \tag{4}$$

(v) $\mathcal{A}$ runs the Test algorithm with inputs $(\xi, C_{\boldsymbol{w}_b^*})$ and returns $b' = 0$ if $\mathrm{Test}(\xi, C_{\boldsymbol{w}_b})$ returns 1. Otherwise $\mathcal{A}$ returns $b' = 1$.

Analysis. It is easy to verify that $e(\zeta_0, \xi_0) = e(g_1, g_1)^{\psi_1\psi\delta^2}$. Similarly we verify that $e(\zeta_1, \xi_1) \cdot e(\zeta_2, \xi_2) = e(g_0, g_0)^{2m_1\phi_1\phi} \cdot e(g_1, g_1)^{\psi_1\psi\delta^2}$. Hence, Test outputs 1 with high probability only when $m_1 = 0$, i.e., when $\boldsymbol{w}_0^*$ was encrypted by the challenger.

4.2 Attack for General n

Here, we discuss the attack on [34] scheme for orthogonality testing when the vectors are of any length. The same intuition (Eqn. 2) works here too.

Proposition 7 *The* PPTag *scheme proposed in [34] for testing orthogonality is not even* ***selective FtG*** *secure.*

Proof. We establish the claim in terms of the following attack game between the adversary and the challenger.

(i) $\mathcal{A}$ outputs two n-dimensional vectors $\overrightarrow{m}_0^*, \overrightarrow{m}_1^*$ as the challenge messages where $n \ll N$. The challenges are of the form $\overrightarrow{m}_0^* = (m_1, m_0, 1, \ldots, 1)$ and $\overrightarrow{m}_1^* = (m_1, m_1, 1, \ldots, 1)$, where $m_1 \neq m_0$ are from $\mathbb{Z}_N^*$.

(ii) $\mathcal{A}$ receives the public parameter PP from challenger.

(iii) $\mathcal{A}$ queries $Q = ((m_1 + m_0)/2, (m_0 - m_1)/2, 1, \ldots, 1, -(n-3))$. Observe that Q is not orthogonal to either of the challenge messages $(\overrightarrow{m}_0^*, \overrightarrow{m}_1^*)$ and hence, is a valid query. $\mathcal{S}$ responds with

$$CT_Q = \left(g_1^{\psi\delta}, g_0^{\phi(m_1+m_0)/2} g_1^{\psi\gamma_1}, g_0^{\phi(m_0-m_1)/2} g_1^{\psi\gamma_2}, g_0^{\phi} g_1^{\psi\gamma_3}, \ldots, g_0^{\phi} g_1^{\psi\gamma_{n-1}}, g_0^{-(n-3)\phi} g_1^{\psi\gamma_n} \right)$$

for some $\psi, \phi \in_R \mathbb{Z}_N$. Given CT_Q, $\mathcal{A}$ takes the product and ratio of the second and third components of the ciphertext to obtain $g_0^{m_0\phi} g_1^{\psi(\gamma_1+\gamma_2)}$ and $g_0^{-m_1\phi} g_1^{\psi(\gamma_2-\gamma_1)}$, respectively. $\mathcal{A}$ now computes the *pseudo-ciphertext* for $\overrightarrow{m}' = (m_0, -m_1, 1, \ldots, 1, -(n-3))$ as

$$CT_Q' = (g_1^{\psi\delta}, g_0^{m_0\phi} g_1^{\psi(\gamma_1+\gamma_2)}, g_0^{-m_1\phi} g_1^{\psi(\gamma_2-\gamma_1)}, g_0^{\phi} g_1^{\psi\gamma_3}, \ldots, g_0^{\phi} g_1^{\psi\gamma_{n-1}}, g_0^{-(n-3)\phi} g_1^{\psi\gamma_n}).$$

Note that the message vector $\overrightarrow{m}'$ is orthogonal to $\overrightarrow{m}_0^*$ but not to $\overrightarrow{m}_1^*$. As in our previous attack for length 2 vectors, the pseudo-ciphertext for $\overrightarrow{m}'$ can be used to distinguish the challenge messages.

(iv) $\mathcal{A}$ now asks for the challenge ciphertext. Suppose that the challenger responds with

$$CT_b = \left(g_1^{\tilde{\psi}\delta}, g_0^{m_1\tilde{\phi}} g_1^{\gamma_1\tilde{\psi}}, g_0^{m_b\tilde{\phi}} g_1^{\gamma_2\tilde{\psi}}, g_0^{\tilde{\phi}} g_1^{\gamma_3\tilde{\psi}}, \cdots, g_0^{\tilde{\phi}} g_1^{\gamma_n\tilde{\psi}} \right),$$

where $b \in_R \{0,1\}$ is chosen by $\mathcal{S}$ and $\tilde{\phi}, \tilde{\psi} \in_R \mathbb{Z}_N$.

(v) $\mathcal{A}$ runs the Test algorithm on CT_Q' and CT_b. This amounts to computing:

$$\begin{aligned} A &= e(g_1^{\psi\delta}, g_1^{\tilde{\psi}\delta}) \quad \text{and} \\ B &= e(g_0^{m_0\phi} g_1^{\psi(\gamma_1+\gamma_2)}, g_0^{m_1\tilde{\phi}} g_1^{\gamma_1\tilde{\psi}}) \cdot e(g_0^{-m_1\phi} g_1^{\psi(\gamma_2-\gamma_1)}, g_0^{m_b\tilde{\phi}} g_1^{\gamma_2\tilde{\psi}}) \\ &\prod_{i=3}^{n-1} e(g_0^{\phi} g_1^{\psi\gamma_i}, g_0^{\tilde{\phi}} g_1^{\gamma_i\tilde{\psi}}) \cdot e(g_0^{-(n-3)\phi} g_1^{\psi\gamma_n}, g_0^{\tilde{\phi}} g_1^{\gamma_n\tilde{\psi}}). \end{aligned}$$

If $A = B$ then $\mathcal{A}$ outputs $b' = 0$, otherwise $\mathcal{A}$ outputs $b' = 1$.

Analysis. We see that $A = B$ implies $m_b = m_0$, except with negligible probability. Hence, the adversary wins the selective FtG game with overwhelming probability of success. □

Remark 4 *It would have been illustrating to see where exactly the proof of Theorem 5.1 in [34] fails. Unfortunately no such proof is provided by the authors of [34].*

5 Orthogonality: Relation Between FtG and LoR and with Equality

We show that it is possible to construct a LoR secure scheme from a FtG secure scheme for orthogonality. This result provides evidence that FtG is a meaningful notion for property preserving encryption. In particular, we show that LoR security for a scheme for testing orthogonality of n length vectors is implied by FtG security of the scheme for testing orthogonality of $2n$ length vectors. Next, we show that in the property preserving scenario, orthogonality implies equality.

5.1 FtG_{2n} implies LoR_n

Motivated by the relationship between the single challenge and full security in the predicate private case for orthogonality, we examine the relationship between FtG and LoR security for that property. It has been shown by Shen, Shi and Waters [37, Theorem 2.8] that a single challenge secure predicate-only private scheme Π_{2n} for testing orthogonality of vectors of length $2n$ may be used to construct one achieving full security for n length vectors Π_n. Inspired by their technique we derive a similar result for property preserving orthogonality testing.

Let Θ_{2n} be a FtG secure PPTag encryption scheme for testing orthogonality of vectors of length $2n$. We construct a PPTag scheme Θ_n for testing orthogonality of vectors of length n as follows. In the following we assume that the underlying field on which the vectors are defined does not have characteristic 2 (this is required in the security argument).

1. $\Theta_n{\cdot}\mathsf{Setup}(\lambda)$: The public parameters and the secret key are the same as the corresponding ones of Θ_{2n}.
2. $\Theta_n \cdot \mathsf{Encrypt}(PP, SK, x)$: $\mathsf{Enc}_n(x) = \mathsf{Enc}_{2n}(x||x)$.
3. $\Theta_n \cdot \mathsf{Test}(CT_1, CT_2, PP)$: The test is carried out using that of the Θ_{2n} scheme as $\mathsf{Test}_n(CT_1, CT_2) = 1$ if and only if $\mathsf{Test}_{2n}(CT_1, CT_2) = 1$.

Next, we show that Θ_n is LoR secure. The proof proceeds via a sequence of hybrids. Any adversary who can distinguish two adjacent games can break the FtG security of Θ_{2n}.

Theorem 8 *The scheme Θ_{2n} is FtG secure implies the derived scheme Θ_n is LoR secure.*

Proof. Recall that we assumed the underlying field on which the vectors are defined does not have characteristic 2. For $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$, the quantity $x||y := (x_1, \ldots, x_n, y_1, \ldots, y_n)$. We observe that $x \cdot y = 0$ if and only if $(x||x) \cdot (y||y) = 0$. The encoding which maps x to $x||x$ is used for proving LoR security via a hybrid argument.

Let $\mathcal{A}$ be a valid LoR adversary for Θ_n. The adversary $\mathcal{A}$ sets $(x_0^{(1)}, x_1^{(1)}), \ldots, (x_0^{(q)}, x_1^{(q)})$ as challenges to the challenger $\mathcal{C}$. The challenger fixes a random bit b and returns encryption of $x_b^{(i)},\ 1 \le i \le q$. The adversary outputs a bit b' at the end of the game and wins if $b = b'$.

We prove that the distributions of the ciphertexts of the left and right side messages are indistinguishable. That is, the adversary $\mathcal{A}$ can not distinguish Game_0 and Game_1 of Table 1. The proof proceeds via a sequence of hybrid games. We tabulate the sequence of hybrids in Table 1. In Game_C, the value α is chosen at random from the underlying field. We mention that a sequence of intermediate games is defined between two consecutive games for proving indistinguishability, where only one ciphertext is changed. One such sequence between Game_A and Game_B is given in Table 1.

We first argue that Game_0 and Game_A are indistinguishable. Consider an intermediate game, called $\mathsf{Game}_{0,1}$, defined as

$$x_0^{(1)}||0, x_0^{(2)}||x_0^{(2)} \ldots, x_0^{(s)}||x_0^{(q)}.$$

Notice that this game differs from Game_0 only in the first component. We claim that Game_0 and $\mathsf{Game}_{0,1}$ are indistinguishable. For, suppose $\mathcal{A}$ can distinguish them. Setting $(x_0^{(1)}||x_0^{(1)}, x_0^{(1)}||0)$ as challenge messages and querying the rest of the elements, adversary

Table 1. Sequence of Hybrids (Left) and Intermediate Games between Game_A and Game_B (Right)

$\mathsf{Game}_0 : x_0^{(1)}\Vert x_0^{(1)}, \ldots, x_0^{(q)}\Vert x_0^{(q)}$
$\mathsf{Game}_A : x_0^{(1)}\Vert 0, \ldots, x_0^{(q)}\Vert 0$
$\mathsf{Game}_B : x_0^{(1)}\Vert \alpha x_1^{(1)}, \ldots, x_0^{(q)}\Vert \alpha x_1^{(q)}$
$\mathsf{Game}_C : 0\Vert \alpha x_1^{(1)}, \ldots, 0\Vert \alpha x_1^{(q)}$
$\mathsf{Game}_D : x_1^{(1)}\Vert \alpha x_1^{(1)}, \ldots, x_1^{(q)}\Vert \alpha x_1^{(q)}$
$\mathsf{Game}_1 : x_1^{(1)}\Vert x_1^{(1)}, \ldots, x_1^{(q)}\Vert x_1^{(q)}$

$\mathsf{Game}_A : x_0^{(1)}\Vert 0, x_0^{(2)}\Vert 0, \ldots, x_0^{(q)}\Vert 0$
$\mathsf{Game}_{A,1} : x_0^{(1)}\Vert \alpha x_1^{(1)}, x_0^{(2)}\Vert 0, \ldots, x_0^{(q)}\Vert 0$
$\mathsf{Game}_{A,2} : x_0^{(1)}\Vert \alpha x_1^{(1)}, x_0^{(2)}\Vert \alpha x_1^{(2)}, x_0^{(3)}\Vert 0, \ldots, x_0^{(q)}\Vert 0$
$\vdots$
$\mathsf{Game}_B : x_0^{(1)}\Vert \alpha x_1^{(1)}, \ldots, x_0^{(q)}\Vert \alpha x_1^{(q)}$

$\mathcal{A}$ becomes a valid FtG adversary for Θ_{2n}. We proceed by defining a sequence of games where any two consecutive games vary exactly at one component. Similar argument would show that Game_B and Game_C are indistinguishable. Game_C and Game_D too may similarly be shown to be indistinguishable.

Recall that Game_B was defined using a random parameter α. Even though, say for example $(x_0^{(1)}||0) \cdot (x_0^{(2)}||0) \neq 0$ holds, it may so happen that $(x_0^{(1)}||x_1^{(1)}) \cdot (x_0^{(2)}||x_1^{(2)}) = 0$. Thus, a random choice of α ensures that setting $(x_0^{(1)}||0, x_0^{(1)}||\alpha x_1^{(1)})$ as the challenge and the rest of the elements as queries one gets a valid FtG adversary for Θ_{2n}. This argument shows that Game_A and Game_B are indistinguishable. Similar argument shows that Game_D and Game_1 are indistinguishable. □

5.2 A Direct Test for Equality from Orthogonality

Katz et al. [30] suggested a simple encoding to test for equality using inner product: create a ciphertext for $\overrightarrow{I} = (1, I)$ and a token for $\overrightarrow{J} = (-J, 1)$. Now the inner product of $\overrightarrow{I}$ and $\overrightarrow{J}$ is 0 if and only if $I = J$. This encoding does not directly work in the PPEnc setting as there is no separate token and the Test is performed only the ciphertext. Nevertheless, we show that one can construct a scheme for testing equality property, given a scheme for testing orthogonality of vectors. The new scheme inherits the same security as that of the underlying orthogonality testing scheme. Note that this result is of theoretical interest, but of little practical value as we already have much more efficient scheme for testing equality. The result formally establishes a connection between orthogonality and equality for property preserving encryption.

The setting is as follows. Let the message space be $\mathbb{F}_q$, where the finite field is assumed to contain $i = \sqrt{-1}$. Examples of fields which contain i are $\mathbb{F}_{2^n}$; $\mathbb{F}_p$, where $p \equiv 1 \pmod 4$; or extensions of the form $\mathbb{F}_q$ which contain i. The square root of -1 may be given explicitly or may be computed using Tonelli-Shanks algorithm [3, Chapter 7].

We encode any $x \in \mathbb{F}_q$ as a vector in $\mathbb{F}_q^5$ given by $x \mapsto v_x = (x^2 + 1, ix^2, ix, ix, i)$ (in characteristic 2 fields $m \mapsto v_m = (m + 1, m, 1)$). The mapping $m \mapsto v_m$ is one-to-one. Observe that, elements x and y are equal if and only if $v_x \cdot v_y = 0$. We now describe a scheme Π' for testing for equality, given a scheme Π for testing for orthogonality of vectors of length 5 over $\mathbb{F}_q$.

1. $\mathsf{Setup}(\lambda)$: The public parameters and secret key for Π' are those of Π.
2. $\mathsf{Encrypt}(PP, SK, m)$: While encrypting $m \in \mathbb{F}_q$, the encryption algorithm first computes the encoding v_m corresponding to m. Then the ciphertext corresponding to m is $CT(m) = \Pi.ct(v_m)$.

3. $\mathsf{Test}(CT_1, CT_2, PP)$: Same as that of Π.

Lemma 9 *If Π is* FtG *(respectively* LoR*) secure then so is Π', correspondingly.*

Proof. We describe the FtG case as the LoR case may be similarly handled. Suppose Π' is not FtG secure, with $\mathcal{A}_{\Pi'}$ a valid adversary. We construct $\mathcal{A}_{\Pi}$, a FtG adversary for scheme Π, which internally runs $\mathcal{A}_{\Pi'}$. Whenever $\mathcal{A}_{\Pi'}$ makes an encryption query m, the adversary $\mathcal{A}_{\Pi}$ forwards v_m to the challenger $\mathcal{B}_{\Pi}$. On receiving the ciphertext, it forwards it to $\mathcal{A}_{\Pi'}$. When $\mathcal{A}_{\Pi'}$ sets (m_0^*, m_1^*) as challenge, the adversary $\mathcal{A}_{\Pi}$ forwards $(v_{m_0^*}, v_{m_1^*})$ to the challenger. On receiving the encryption of one of the two vectors $\mathcal{A}_{\Pi}$ forwards it to $\mathcal{A}_{\Pi'}$. The other queries made by $\mathcal{A}_{\Pi'}$ may be handled similarly. When $\mathcal{A}_{\Pi'}$ outputs a bit b' and halts, so does $\mathcal{A}_{\Pi}$. This is a perfect simulation and $\mathcal{A}_{\Pi}$ wins with the same advantage as that of $\mathcal{A}_{\Pi'}$. □

6 Concluding Remarks

In the preface of their highly acclaimed book, Katz and Lindell [29] spell out three principles of modern cryptography: the central role of definitions, importance of formal and precise assumptions and the possibility of rigorous proofs of security. Our work of revisiting the Pandey and Rouselakis [34] can be regarded as an exploration of the first and the third principles mentioned above.

On the definitional front, we revisit the FtG vs. LoR and the FtG hierarchy separation results in [34]. To do that we show that equality property captures the property P_{qr} used in the separation results and provide a simple construction for the equality property to demonstrate that the separation results are non-vacuous. Based on the security attributes of our construction and its generalization we raised the pertinent question of whether the separation results actually indicate any real world difference between the two notions of security. Continuing further our exploration of the relation between FtG and LoR notions, we see that a LoR-secure scheme may be constructed from a so-called weaker FtG-secure one for orthogonality. We demonstrate a simple attack on the PPTag scheme for testing orthogonality from [34] refuting the claim that the scheme is provably secure.

Our study indicates that a more detailed analysis is required regarding proper notion of security for property preserving symmetric encryption and also underlines the importance of cryptanalysis in provable security.

References

1. Shashank Agrawal, Shweta Agrawal, Saikrishna Badrinarayanan, Abishek Kumarasubramanian, Manoj Prabhakaran, and Amit Sahai. Function Private Functional Encryption and Property Preserving Encryption : New Definitions and Positive Results. Cryptology ePrint Archive, Report 2013/744, 2013. http://eprint.iacr.org/.
2. Georgios Amanatidis, Alexandra Boldyreva, and Adam O'Neill. Provably-secure schemes for basic query support in outsourced databases. In Steve Barker and Gail-Joon Ahn, editors, *Data and Applications Security XXI, 21st Annual IFIP WG 11.3 Working Conference on Data and Applications Security, Redondo Beach, CA, USA, July 8-11, 2007, Proceedings*, volume 4602 of *Lecture Notes in Computer Science*, pages 14–30. Springer, 2007.
3. E. Bach and J.O. Shallit. *Algorithmic Number Theory.* Foundations of computing. MIT Press, 1996.
4. Mihir Bellare, Anand Desai, E. Jokipii, and Phillip Rogaway. A concrete security treatment of symmetric encryption. In *FOCS*, pages 394–403. IEEE Computer Society, 1997.

5. Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations among notions of security for public-key encryption schemes. In Hugo Krawczyk, editor, *CRYPTO*, volume 1462 of *Lecture Notes in Computer Science*, pages 26–45. Springer, 1998.
6. Mihir Bellare and Adam O'Neill. Semantically-secure functional encryption: Possibility results, impossibility results and the quest for a general definition. In Michel Abdalla, Cristina Nita-Rotaru, and Ricardo Dahab, editors, *CANS*, volume 8257 of *Lecture Notes in Computer Science*, pages 218–234. Springer, 2013.
7. Mihir Bellare, Thomas Ristenpart, Phillip Rogaway, and Till Stegers. Format-preserving encryption. In Michael J. Jacobson Jr., Vincent Rijmen, and Reihaneh Safavi-Naini, editors, *Selected Areas in Cryptography*, volume 5867 of *Lecture Notes in Computer Science*, pages 295–312. Springer, 2009.
8. Alexandra Boldyreva, Nathan Chenette, Younho Lee, and Adam O'Neill. Order-preserving symmetric encryption. In Antoine Joux, editor, *EUROCRYPT*, volume 5479 of *Lecture Notes in Computer Science*, pages 224–241. Springer, 2009.
9. Alexandra Boldyreva, Nathan Chenette, Younho Lee, and Adam ONeill. Order-preserving symmetric encryption. Cryptology ePrint Archive, Report 2012/624, 2012. http://eprint.iacr.org/.
10. Alexandra Boldyreva, Nathan Chenette, and Adam O'Neill. Order-preserving encryption revisited: Improved security analysis and alternative solutions. In Phillip Rogaway, editor, *CRYPTO*, volume 6841 of *Lecture Notes in Computer Science*, pages 578–595. Springer, 2011.
11. Alexandra Boldyreva, Nathan Chenette, and Adam ONeill. Order-preserving encryption revisited: Improved security analysis and alternative solutions. Cryptology ePrint Archive, Report 2012/625, 2012. http://eprint.iacr.org/.
12. Dan Boneh and Xavier Boyen. Efficient selective identity-based encryption without random oracles. *J. Cryptology*, 24(4):659–693, 2011.
13. Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity based encryption with constant size ciphertext. In Ronald Cramer, editor, *EUROCRYPT*, volume 3494 of *Lecture Notes in Computer Science*, pages 440–456. Springer, 2005.
14. Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. Public key encryption with keyword search. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT*, volume 3027 of *Lecture Notes in Computer Science*, pages 506–522. Springer, 2004.
15. Dan Boneh, Ananth Raghunathan, and Gil Segev. Function-private identity-based encryption: Hiding the function in functional encryption. In Canetti and Garay [19], pages 461–478.
16. Dan Boneh, Ananth Raghunathan, and Gil Segev. Function-private subspace-membership encryption and its applications. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT (1)*, volume 8269 of *Lecture Notes in Computer Science*, pages 255–275. Springer, 2013.
17. Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: Definitions and challenges. In Yuval Ishai, editor, *TCC*, volume 6597 of *Lecture Notes in Computer Science*, pages 253–273. Springer, 2011.
18. Dan Boneh and Brent Waters. Conjunctive, subset, and range queries on encrypted data. In Salil P. Vadhan, editor, *TCC*, volume 4392 of *Lecture Notes in Computer Science*, pages 535–554. Springer, 2007.
19. Ran Canetti and Juan A. Garay, editors. *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II*, volume 8043 of *Lecture Notes in Computer Science*. Springer, 2013.
20. Angelo De Caro, Vincenzo Iovino, and Giuseppe Persiano. Fully Secure Anonymous HIBE and Secret-Key Anonymous IBE with Short Ciphertexts. In Marc Joye, Atsuko Miyaji, and Akira Otsuka, editors, *Pairing*, volume 6487 of *Lecture Notes in Computer Science*, pages 347–366. Springer, 2010.
21. Mache Creeger. Cloud computing: An overview. *ACM Queue*, 7(5):2, 2009.
22. Sanjam Garg, Craig Gentry, Shai Halevi, Amit Sahai, and Brent Waters. Attribute-based encryption for circuits from multilinear maps. In Canetti and Garay [19], pages 479–499.
23. Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *STOC*, pages 169–178. ACM, 2009.
24. Craig Gentry, Shai Halevi, and Nigel P. Smart. Fully homomorphic encryption with polylog overhead. In Pointcheval and Johansson [35], pages 465–482.
25. Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In Ran Canetti and Juan A. Garay, editors, *CRYPTO (1)*, volume 8042 of *Lecture Notes in Computer Science*, pages 75–92. Springer, 2013.
26. Shafi Goldwasser, S. Dov Gordon, Vipul Goyal, Abhishek Jain, Jonathan Katz, Feng-Hao Liu, Amit Sahai, Elaine Shi, and Hong-Sheng Zhou. Multi-input functional encryption. In Phong Q. Nguyen and

Elisabeth Oswald, editors, *EUROCRYPT*, volume 8441 of *Lecture Notes in Computer Science*, pages 578–602. Springer, 2014.

27. Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM Conference on Computer and Communications Security*, pages 89–98. ACM, 2006.
28. Sudipto Guha, Adam Meyerson, Nina Mishra, Rajeev Motwani, and Liadan O'Callaghan. Clustering data streams: Theory and practice. *IEEE Trans. Knowl. Data Eng.*, 15(3):515–528, 2003.
29. J. Katz and Y. Lindell. *Introduction to Modern Cryptography.* Chapman and Hall/CRC, 2008.
30. Jonathan Katz, Amit Sahai, and Brent Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In Nigel P. Smart, editor, *EUROCRYPT*, volume 4965 of *Lecture Notes in Computer Science*, pages 146–162. Springer, 2008.
31. Yutaka Kawai and Katsuyuki Takashima. Predicate- and attribute-hiding inner product encryption in a public key setting. In Zhenfu Cao and Fangguo Zhang, editors, *Pairing*, volume 8365 of *Lecture Notes in Computer Science*, pages 113–130. Springer, 2013.
32. Tatsuaki Okamoto and Katsuyuki Takashima. Fully secure unbounded inner-product and attribute-based encryption. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT*, volume 7658 of *Lecture Notes in Computer Science*, pages 349–366. Springer, 2012.
33. Tatsuaki Okamoto and Katsuyuki Takashima. Efficient (hierarchical) inner-product encryption tightly reduced from the decisional linear assumption. *IEICE Transactions*, 96-A(1):42–52, 2013.
34. Omkant Pandey and Yannis Rouselakis. Property Preserving Symmetric Encryption. In Pointcheval and Johansson [35], pages 375–391.
35. David Pointcheval and Thomas Johansson, editors. *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*, volume 7237 of *Lecture Notes in Computer Science*. Springer, 2012.
36. Phillip Rogaway and Thomas Shrimpton. A provable-security treatment of the key-wrap problem. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, volume 4004 of *Lecture Notes in Computer Science*, pages 373–390. Springer, 2006.
37. Emily Shen, Elaine Shi, and Brent Waters. Predicate Privacy in Encryption Systems. In Omer Reingold, editor, *TCC*, volume 5444 of *Lecture Notes in Computer Science*, pages 457–473. Springer, 2009.
38. Elaine Shi, John Bethencourt, Hubert T.-H. Chan, Dawn Xiaodong Song, and Adrian Perrig. Multi-dimensional range query over encrypted data. In *IEEE Symposium on Security and Privacy*, pages 350–364. IEEE Computer Society, 2007.
39. Elaine Shi and Brent Waters. Delegating capabilities in predicate encryption systems. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *ICALP (2)*, volume 5126 of *Lecture Notes in Computer Science*, pages 560–578. Springer, 2008.
40. Dawn Xiaodong Song, David Wagner, and Adrian Perrig. Practical techniques for searches on encrypted data. In *IEEE Symposium on Security and Privacy*, pages 44–55. IEEE Computer Society, 2000.
41. Marten van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers. In Henri Gilbert, editor, *EUROCRYPT*, volume 6110 of *Lecture Notes in Computer Science*, pages 24–43. Springer, 2010.

Appendix A

We give the proof of Lemma 5 and 6 in this Appendix. We first argue the separation result for polynomial size message space case and use it to prove the general case.

A.1 Separation Result for Polynomial Size Message Space

Let $\mathcal{M} = \{1, \ldots, l\}$ be the message space. We give an argument for the separation result $\mathsf{FtG} \not\rightarrow \mathsf{LoR}$ for the equality property in the case where l is polynomial in the security parameter.

Set $z = \lceil \log_2 l \rceil$. We assume that l is polynomial in the security parameter. Let Π be a FtG secure PPTag scheme for equality. From this scheme we construct another scheme Π' for realizing the same property as follows. Let the set of l binary strings of length z, denoted by $\{\alpha_i \mid 1 \leq i \leq l\}$, which represent the integers 1 through l.

1. $\mathsf{Setup}(\lambda)$: The public parameters for Π' include those of Π. The secret key of Π' comprises of those of Π and a set of randomly chosen binary strings $\{t_i \mid 1 \leq i \leq l\}$, where each t_i is of length z.
2. $\mathsf{Encrypt}(PP, SK, m)$: While encrypting $m \in \mathcal{M}$, the encryption oracle of Π is used to obtain ct. Then the algorithm chooses a random bit b. The ciphertext now comprises of

$$CT = \begin{cases} (ct, b, t_l), & \text{if } b = 0, \\ (ct, b, t_l \oplus \alpha_s), & \text{otherwise.} \end{cases}$$

3. $\mathsf{Test}(CT_1, CT_2, PP)$: Same as that of Π, where only the relevant parts of the ciphertexts are used.

We first prove that Π' is not LoR secure. The same proof holds also for Lemma 6.

Lemma 10 *There is a* LoR *adversary for the scheme* Π' *with non-negligible advantage.*

Proof. A valid LoR adversary sets as u challenges the same pair of the form (m_0, m_1), with $m_0 \neq m_1$. Equality pattern is clearly preserved between the left and right sequences. If the challenger outputs two ciphertexts for which the b-values are distinct, the adversary can win with advantage $1 - 2^{-u+1}$. □

We now prove that Π' is FtG secure.

Theorem 11 *The scheme* Π' *is not* FtG *secure implies* Π *is not* FtG *secure.*

Proof. Consider a valid FtG adversary for Π', denoted by $\mathcal{A}$. Two cases arise with respect to the challenges, which we describe below.

Case 1 : The challenge messages m_0^* and m_1^* are equal al.
Case 2 : The challenge messages m_0^* and m_1^* are different. In this case, the adversary can not make an encryption query for these two messages.

We describe how a FtG adversary $\mathcal{B}$ for Π, with same advantage as that of $\mathcal{A}$ and which internally uses $\mathcal{A}$, can be constructed.

1. The algorithm $\mathcal{B}$ receives the public parameters from Π. It also initializes an empty table T. Apart from these, note that the $\alpha_1, \ldots, \alpha_l$ values are known to $\mathcal{A}$.
2. Whenever $\mathcal{A}$ makes an encryption query i, the algorithm $\mathcal{B}$ receives and forwards it to the simulator of Π. On receiving ct, the algorithm $\mathcal{B}$ whether the same query was made earlier or not.
   (a) If the query is made for the first time, then it chooses $t \in \{0,1\}^z$. It also sets $t_i = t$ and updates the table T with $\{(i, t_i)\}$.
   (b) Else, it reuses t_i.

Thus, the t_i value is determined by $\mathcal{B}$. It then chooses a random bit b and forwards

$$CT = \begin{cases} (ct, b, t_i), & \text{if } b = 0, \\ (ct, b, t_i \oplus \alpha_i), & \text{if } b = 1. \end{cases}$$

respectively, to $\mathcal{A}$.

3. After a certain number of queries $\mathcal{A}$ outputs the challenge $(m_0^*, m_1^*, state)$. The algorithm $\mathcal{B}$ forwards the challenge to the simulator of Π and gets ct^*. If the challenge messages are equal (**Case 1**), then (ct^*, b, val) may be computed by $\mathcal{B}$ in the two sub cases where the t-value is known or unknown. If the challenge messages are different (**Case 2**), then both these have not been queried previously. The valid Π^* ciphertext may be computed by $\mathcal{B}$ in the obvious way.
4. All the subsequent queries which $\mathcal{A}$ makes can be handled similarly by $\mathcal{B}$. When $\mathcal{A}$ outputs a bit b' and halts, so does $\mathcal{B}$.

Notice that the ciphertexts which $\mathcal{B}$ computes for forwarding to $\mathcal{A}$ are properly distributed. In other words, the algorithm $\mathcal{B}$ is a perfect simulation. Clearly, the algorithm $\mathcal{B}$ is PPT.

Algorithm $\mathcal{B}$ outputs whatever is given to it by $\mathcal{A}$. Hence, advantage of $\mathcal{B}$ is equal to that of $\mathcal{A}$. □

A.2 Proof of Lemma 5

Recall that in the FtG game $\mathcal{A}$ makes a polynomial number of encryption oracle query m_i, $1 \leq i \leq q$, and a single challenge query (m_0^*, m_1^*) maintaining the equality pattern. Two cases arise depending upon whether the challenge messages m_0^* and m_1^* are equal or not. If $m_0^* = m_1^*$ then it is easy to see that any advantage of $\mathcal{A}$ against Π' translates into the same advantage against Π. Hence, we consider the case when $m_0^* \neq m_1^*$. Note that in this case none of the queries to the encryption oracle m_i is equal to m_b^*, for $b \in \{0, 1\}$. Otherwise, the equality pattern of the two sequences will be different allowing $\mathcal{A}$ to trivially distinguish.

Let Game_0 correspond to the adversarial queries $(m_1, \ldots, m_i, m_0^*, m_{i+1}, \ldots, m_q)$ while Game_1 correspond to the adversarial queries $(m_1, \ldots, m_i, m_1^*, m_{i+1}, \ldots, m_q)$. Suppose $\mathcal{A}$ can distinguish whether it is playing Game_0 or Game_1 with a non-negligible advantage $\epsilon_{\Pi'}$. The proof will proceed through a hybrid argument. Given an adversary $\mathcal{A}$ against Π' we construct a series of four games and then show that if $\mathcal{A}$ can distinguish between any two consecutive games then we can construct either a PRF adversary against $\mathcal{F}$ or an FtG adversary against Π.

Game_0 The challenger runs the Setup algorithm of Π' and gives the PP to $\mathcal{A}$ and keeps the secret key $SK' = (SK, k)$ to itself. The challenger computes the ciphertext corresponding to $(m_1, \ldots, m_i, m_0^*, m_{i+1}, \ldots, m_q)$ using SK' as per the encryption algorithm of Π' and give them to $\mathcal{A}$.

Game_A The challenger runs the Setup algorithm of Π and gives the PP to $\mathcal{A}$ and keeps the secret key SK of Π to itself. Note that the challenger does not generate the PRF key k; instead it will maintain a table $\mathbb{T} = \langle x_i, y_i \rangle$ where x_i and y_i are two z-bit strings. The first entry in each row of $\mathbb{T}$ corresponds to the messages queried by $\mathcal{A}$ while the second entry is a random bit-string. The table is initially empty. Whenever $\mathcal{A}$ makes an encryption query for a message x, the challenger first checks whether there is a corresponding entry in $\mathbb{T}$. If not, it chooses a random z-bit string y and enters (x, y) in the table $\mathbb{T}$ sorted according to the first entry. $\mathcal{A}$ makes encryption queries for $(m_1, \ldots, m_i, m_0^*, m_{i+1}, \ldots, m_q)$. To answer the query of $\mathcal{A}$ for a message, say x, the challenger computes the ciphertext of Π on x and then uses the corresponding random string y from the entry (x, y) in $\mathbb{T}$ to create a ciphertext of Π'. Note that $\mathcal{A}$ makes at most q encryption oracle queries and a single challenge query. So the size of $\mathbb{T}$ is $O(q)$ and hence the challenger can consistently respond to all the queries of $\mathcal{A}$.

Claim 12 *If $\mathcal{A}$ can decide with a non-negligible advantage whether it is playing Game_0 or Game_A then we can construct a PRF distinguisher with the same advantage.*

Recall that in the PRF security game we are provided with an oracle which is either a function from the PRF family or a random function. In the former case the challenger will be playing Game_0 while in the latter case it'll be playing Game_A. Hence, any advantage of $\mathcal{A}$ in distinguishing between the two games translate into the same advantage of the challenger in breaking the PRF security.

Game_1 (resp. Game_B) will be identical to Game_0 (resp. Game_A) except the fact that $\mathcal{A}$ now queries with $(m_1, \ldots, m_i, m_1^*, m_{i+1}, \ldots, m_q)$. An identical argument as in the claim above establishes that any advantage of $\mathcal{A}$ in deciding whether it is playing Game_1 or Game_B translates into the same PRF advantage for the challenger.

Note that the only difference in Game_A and Game_B is in the challenge ciphertext (corresponding to m_0^* and m_1^*). The challenge is computed by calling the encryption algorithm of Π and appending either a random bit string or a one-time encryption of m_b^* (using that random string). Hence, an adversary distinguishing between Game_A and Game_B can be converted into an adversary breaking the FtG security of Π. As there are only polynomial many queries, this case is the same as the one where there are only small (polynomial in λ) number of messages. This case can be easily handled by using random strings. We have already given the analysis in the previous subsection.

Appendix B

Here, we examine the hierarchy among FtG classes for equality property. In particular, for this property we show that $\mathsf{FtG}^\eta \nrightarrow \mathsf{FtG}^{\eta+1}$. As before, we start with any PPTag scheme Π which is FtG^η secure and derive another PPTag scheme Π' which is FtG^η secure but not $\mathsf{FtG}^{\eta+1}$ secure. The case of polynomial many equivalence classes is dealt with using random strings here. We begin by describing Π'. Let $\mathcal{M} = \{1, \ldots, l\}$, be the messages. The values $\{\alpha_1, \ldots, \alpha_l\}$ are publicly computable bit representations of the messages.

1. $\mathsf{Setup}(\lambda)$: The public parameters for Π' are precisely those of Π. The secret key of Π' comprises of those of Π and a set of $l \times \eta$ randomly chosen z-bit integers represented as a matrix $((t_{i,j}))$, where $1 \leq i \leq l$ and $1 \leq j \leq \eta$. For $1 \leq k \leq l$, set $T_k = t_{k,1} \oplus \ldots \oplus t_{k,\eta} \oplus \alpha_k$.
2. Encrypt(PP, SK, m): While encrypting $m \in \mathcal{M}$, the encryption oracle of Π is used to obtain ct. Then the algorithm chooses a random integer $b \in [1, \eta + 1]$. The ciphertext now comprises of $CT = (ct, b, val)$, where

$$val = \begin{cases} t_{m,b}, & \text{if } 1 \leq b \leq \eta, \\ T_m, & \text{o.w.} \end{cases}$$

3. Test(CT_1, CT_2, PP): Same as that of $\varPi$, where only the relevant parts of the ciphertexts are used.

We next prove that $\varPi'$ is FtG^{η} secure, but not $\mathsf{FtG}^{\eta+1}$ secure, thus proving hierarchy among FtG classes for the equivalence relation P.

Theorem 13 *The scheme $\varPi'$ is not $\mathsf{FtG}^{\eta+1}$ secure but is FtG^{η} secure.*

Proof. We first show that if the scheme $\varPi'$ is not FtG^{η} secure then $\varPi$ is not FtG^{η} secure. Let $\mathcal{A}_{\varPi}$ be a FtG^{η} adversary which internally runs $\mathcal{A}_{\varPi'}$, a FtG^{η} adversary for $\varPi'$, while interacting with the challenger $\mathcal{B}_{\varPi}$. Let $\{(m^*_{0,i}, m^*_{1,i}) \mid m^*_{0,i} \in C_{x_i},\ m^*_{1,i} \in C_{y_i},\ 1 \leq i \leq \eta\}$ be all the challenge pairs set (adaptively) by $\mathcal{A}_{\varPi'}$.

The adversary $\mathcal{A}_{\varPi}$ initializes an empty array L for storing pairs of integers between 1 and l. It also initializes $\mathcal{T} = ((t_{i,j}))$, a $l \times (\eta + 1)$ empty matrix for storing z-bit strings. We describe how $\mathcal{A}_{\varPi}$ interacts with $\mathcal{B}_{\varPi}$ while internally running $\mathcal{A}_{\varPi'}$.

A. When $\mathcal{A}_{\varPi'}$ makes an encryption query, say m, the adversary $\mathcal{A}_{\varPi}$ forwards it to $\mathcal{B}_{\varPi}$ and obtains the ciphertext *ct*. If the m-th row in $\mathcal{T}$ is empty, it fills values $t_{m,i}$ for $1 \leq i \leq \eta$ with random z-bit strings and sets $t_{m,\eta+1} = t_{m,1} \oplus \ldots t_{m,\eta} + \alpha_m$. It then chooses $b \in [1, \eta + 1]$ at random and returns $(ct, b, t_{m,b})$ to $\mathcal{A}_{\varPi'}$.

B. When $\mathcal{A}_{\varPi'}$ makes the i-th challenge query $(m^*_{0,i}, m^*_{1,i})$, the adversary $\mathcal{A}_{\varPi}$ forwards it to $\mathcal{B}_{\varPi}$ and obtains the ciphertext *ct* of the left or the right message. Then $\mathcal{A}_{\varPi}$ sets $x_i = m^*_{0,i}$ and $y_i = m^*_{1,i}$. It then determines if (x_i, y_i) is present in the list L. Two cases arise.

Case 1. Suppose $x_i = y_i$. This case may be handled by $\mathcal{A}_{\varPi}$ in a fashion similar to handling encryption queries. The $\mathcal{A}_{\diamond}$ chooses $b \in [0, \eta + 1]$ at random and sets $val = t_{x_i,b}$.

Case 2. Suppose $x_i \neq y_i$. Then $\mathcal{A}_{\varPi}$ scans the list L and sees if either (x_i, y_i) or (y_i, x_i) is an entry. If neither is, then $\mathcal{A}_{\varPi}$ chooses a bit β and $b \in [0, \eta + 1]$ at random. If $\beta = 0$ it updates $t_{x_i,b}$ with a random z-bit string and if $\beta = 1$ it updates $t_{y_i,b}$ with a random z-bit string *val*, else sets $val = t_{x_i,b}$. If (x_i, y_i) is already an entry in L, but (y_i, x_i) is not, it checks which of the two rows in $\mathcal{T}$ is non-empty, with out loss of generality say x_i. It chooses a $b \in [0, \eta + 1]$ at random. If $t_{x_i,b}$ is not already updated, it sets this entry to a random z-bit string *val*, else sets $val = t_{x_i,b}$. We next consider the case where (y_i, x_i) is an entry in L, but (x_i, y_i) is not. It checks which of the two rows in $\mathcal{T}$ is non-empty, with out loss of generality say x_i. It chooses a $b \in [0, \eta + 1]$ at random. If $t_{y_i,b}$ is not already updated, it sets this entry to a random z-bit string *val*, else sets $val = t_{x_i,b}$. If both these pairs already occur in the list L, the adversary $\mathcal{A}_{\varPi}$, it chooses a bit β at random. If $\beta = 0$, it processes the row x_i in $\mathcal{T}$ in the usual fashion, else it processes row y_i.

Then $\mathcal{A}_{\varPi}$ returns (ct, b, val) to $\mathcal{A}_{\varPi'}$.

C. When $\mathcal{A}_{\varPi'}$ outputs a bit b' and halts, so does $\mathcal{A}_{\varPi}$.

This is a perfect simulation.

Next, we describe a strategy for $\mathsf{FtG}^{\eta+1}$ adversary. The adversary sets all left challenge messages to be the same fixed value and all the right ones another, such that they are unequal. Observe that he can not query the encryption oracle on these two messages. If all the $\eta+1$ values occur in the b component of the responses of the challenger, the adversary can set $\eta+1$ relations among the $\eta+1$ unknowns, including the α value of the challenger's choices. From this system, the class chosen by the challenger may be determined by the adversary. The event that all the values of b occur in challenger's responses happens with probability $\eta!/\left((\eta+1)^{(\eta+1)}\right)$. $\square$