A Probabilistic Public Key Encryption Switching Protocol for Secure Cloud Storage Applications

Radhakrishna Bhat*, N R Sunitha, S S Iyengar, *Life Fellow, IEEE*

Abstract—The high demand for user-centric applications such as secure cloud storage laid the foundation for the development of user-centric security protocols with multiple security features in recent years. But, the current state-of-art techniques primarily emphasized only one type of security feature i.e., either homomorphism or non-malleability. In order to fill this gap and provide a common platform for both homomorphic and non-malleable cloud applications, we have introduced a new public key-based probabilistic *encryption switching* (i.e., homomorphism to/from non-malleability property switching during the encryption phase without changing the underlying security structure) scheme by introducing a novel Contiguous Chain Bit Pair Encryption (CC-BPE) and Discrete Chain Bit Pair Encryption (DC-BPE) techniques for plaintext bits encryption and using quadratic residuosity-based trapdoor function of Freeman et al. [13] for intermediate ciphertext connections. The proposed scheme generates $\mathcal{O}(m+2 \log N)$ bits of ciphertext where $m \in \mathbb{N}$ and $m < n$, $n \in \mathbb{N}$ is the plaintext size, N is the RSA composite. This security extension would be helpful to cover both homomorphism and non-malleability cloud applications. The superior performance of the proposed scheme has been tested in comparison to existing methods and is reported in this paper.

Index Terms—Probabilistic encryption, public key cryptosystem, quadratic residuosity assumption, encryption switching protocol, homomorphic encryption, non-malleability, secure cloud storage and retrieval.

1 Introduction

The invention of public key cryptography has shown a new direction to several asymmetric privacy-preserving techniques such as information-hiding, private information retrieval, oblivious transfer. The main goal of any public key cryptography is to achieve secure asymmetric communication without prior communication/sharing as contrary to symmetric key cryptography.

1.1 Motivation and Background

It is evident from recent cloud storage and retrieval applications [14], [18], [30] that there is a need to extend security capabilities to accommodate both homomorphism and non-malleability under a single umbrella. Motivated by this, many security techniques such as Encryption Switching Protocol (formally known as ESP) have been proposed to provide unique solution to cover both homomorphism and non-malleability applications.

In asymmetric key cryptography, the bijective trapdoor function mappings are basically used during the encryption process. In order to overcome the problem of the secret-sharing over the insecure channel using symmetric encryption, the concept of asymmetric encryption was proposed using bijective trapdoor one-way function mappings. There are two notable drawbacks in such cryptosystems. First, the security of these cryptosystem completely depends on the underlying hardness assumption(s) (not on the underlying mapping function). Second, existing cryptosystems clearly failed to achieve efficient encryption switching between the homomorphic and the non-malleability properties without altering the underlying structure.

Generally, there are four major concerns in any asymmetric key constructions. i) reasonable ciphertext expansion (i.e., the ratio of ciphertext size to plaintext size) ii) possible operation on the ciphertexts (such as homomorphic property) iii) possible selection of the type of plaintext and size of the plaintext space iv) level of security (such as chosen ciphertext security). Although most of the existing number-theoretic asymmetric encryptions [10], [15], [22], [27] naturally impose a restriction on the generation of small ciphertexts (less than the plaintext) due to the existence of number-theoretic modular operations (like addition or multiplication), such schemes enjoy very useful properties such as homomorphism (partial or full) and cover many useful privacy-preserving extensions such as oblivious transfer, private information retrieval, oblivious RAM etc.

In fact, the security of most of the number-theoretic state-of-art schemes (both deterministic and probabilistic) completely depends upon the underlying intractability assumption (such as integer factorization, quadratic residuosity, phi-hiding, composite residuosity etc.) instead of the underlying bijective functions. It is a fact that most of these schemes (except lattice-based schemes up to some extent) are not one-way functions; they are just trapdoor one-way

- *Radhakrishna Bhat is with the Department of Computer Science and Engineering, Manipal Institute of Technology (MIT), Manipal Academy of Higher Education (MAHE), Manipal, Karnataka India 576104.*
  E-mail: rsb567@gmail.com
  ***Author Contribution:** Conceptualization, Methodology, Modeling, Implementation, Interpretation and validation of results, Manuscript writing and review.*
- *N R Sunitha is with the Department of Computer Science and Engineering, Siddaganga Institute of Technology, Tumakuru, India 572103.*
  E-mail: nrsunithasit@gmail.com
  ***Author Contribution:** Supervision, Resources, Validation, Manuscript review.*
- *S S Iyengar is with School of Computing and Information Science, Florida International University, Florida, Miami, USA.*
  E-mail: iyengar@cis.fiu.edu
  ***Author Contribution:** Resources, Validation, Manuscript review.*

functions. It is intuitive that the security of these trapdoor one-way function schemes relies on hiding the trapdoor information but not on the one-wayness of the underlying mapping function. This motivates us to find an alternative scheme that depends on the one-wayness of the underlying mapping function.

Although these existing schemes exhibit many useful properties they continue to use relatively weak trapdoor one-way functions. This move has naturally raise the following questions on some of the security properties (such as homomorphism) and mapping types (such as bijective and injective types) adopted.

1.1.1 Basic Questions On the Bijective Function-Based Encryption

It is quite natural question that why some lossy trapdoor functions such as surjective functions are not used as the underlying mapping functions and decrease the adversarial probability further? What are the fundamental barriers to stop using lossy trapdoor function-based asymmetric encryptions?

Following investigations on the existing schemes have clearly shown the right answers to the above questions.

- *Breaking the underlying intractability assumption (or compromising the trapdoor information itself) reveals complete plaintext*: The interesting fact is that revealing the trapdoor information (or just by stealing) itself is sufficient to reveal the complete plaintext. One more interesting fact is that if the trapdoor information is lost then the plaintext (not even a part of it) cannot be recovered forever. This security monopoly may be too risky for many privacy-critical user-centric applications such as patient record storage, patent storage on the cloud. Therefore, this clearly shows the need for some kind of partial dependency model where the security equally depends on both trapdoor information and mapping function during the encryption process.
- *Existence of trapdoor one-way functions (not a real one-way function) creates the security monopoly*: It is the well-accepted fact that the existence of true one-way functions does not useful to asymmetric encryption process. Therefore, all the asymmetric encryption schemes have completely dependent on the trapdoor information with the underlying mapping (injective or bijective) function(s). However, this creates a serious security threat to some of the privacy-critical applications such as military, stock, patent etc. Therefore, the existing schemes have clearly failed to break this security monopoly on the trapdoor information.
- *Security parameter dependent plaintext selection in public key encryption*: On the other hand, most of the existing number-theoretic block encryptions have continued to use security parameter dependent plaintext space to achieve almost practical ciphertext expansion factor and therefore their plaintext space always depends on the security parameter. This is one of the fundamental problems for most of the existing legacy infrastructures since they need to rearrange/recalculate their stored information (which is generally difficult for large commercial databases) to the respective encryption system. The security parameter dependent plaintext selection always creates an additional computational burden on the protocol to recalculate/transform the stored information from one format to another for every protocol type. Moreover, the probability distributions of all the elements of such plaintext space are different. This non-uniform distribution of the plaintext generally creates a backdoor to the adversary to compromise the system with very little effort.

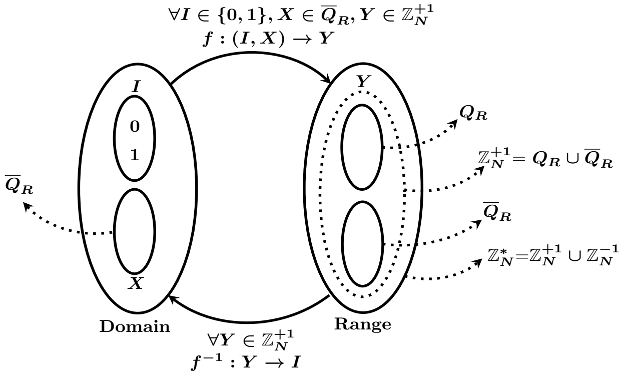

$\forall I \in \{0,1\}, X \in \overline{Q}_R, Y \in \mathbb{Z}_N^{+1}$
$f : (I, X) \rightarrow Y$
I
0
1
X
Y
Q_R
$\mathbb{Z}_N^{+1} = Q_R \cup \overline{Q}_R$
$\overline{Q}_R$
$\mathbb{Z}_N^{*} = \mathbb{Z}_N^{+1} \cup \mathbb{Z}_N^{-1}$
$\overline{Q}_R$
Domain
Range
$\forall Y \in \mathbb{Z}_N^{+1}$
$f^{-1} : Y \rightarrow I$

Fig. 1: The quadratic residuosity-based probabilistic encryption

The drawback of this fundamental security principle has created the necessity of constructing the security parameter independent bit-level encryption system with some useful properties such as *homomorphism* (i.e., method of carrying out particular operations on the ciphertexts). Several researchers have really motivated and successfully presented these security parameter independent encryptions. However, even those security parameter independent schemes have clearly failed to achieve the reasonable ciphertext expansion.

1.1.2 Basic Questions On the Probabilistic Homomorphic Encryption

- *Deterministic property mapping and probabilistic ciphertext mapping in homomorphic probabilistic encryption*: Most of the probabilistic encryption schemes have retained the homomorphic property from deterministic encryptions. It is worth adopting the homomorphic property in several privacy-critical applications including secure online voting. These homomorphic probabilistic encryption schemes have involved both deterministic property mapping and probabilistic ciphertext mapping to achieve both homomorphism and *semantic security*. In any quadratic residuosity-based probabilistic encryption, for instance, bit 0 is always mapped to quadratic residue ciphertext and bit 1 is always mapped to quadratic non-residue ciphertext. Therefore, these quadratic residuosity-based probabilistic encryption schemes involve deterministic mapping of their plaintext into a particular quadratic residuosity property output as shown in Fig. 1.
Let Q_R and $\overline{Q}_R$ be the quadratic residue set and quadratic non-residue set with *Jacobi symbol* 1 respectively. Let f be a plaintext to a ciphertext mapping function. For all bit $I \in \{0, 1\}$ and for all random input $X \in \overline{Q}_R$, the mapping of the bit $f : (I, X) \rightarrow Y$ into quadratic residuosity property ciphertext is always deterministic. Similarly, the inverse mapping of the ciphertext $f^{-1} : Y \rightarrow I$ into plaintext is also deterministic. This bijective mapping always assures the correctness of the inverse mapping to

get back the plaintext. But, the ciphertext value for each encryption instance is always probabilistic. Although these schemes have provided better security solutions (against line tappers) over the deterministic encryption schemes, these schemes have failed to provide chosen ciphertext attacks (both adaptive and non-adaptive) due to the existence of malleability property in these encryptions.

- *Inability of probabilistic homomorphic encryptions to prevent data modifications*: The property of homomorphism is possible due to the commutative property of quadratic residuosity numbers under multiplication. Even though the homomorphic property guarantees very useful operations on ciphertexts, there is a chance that the stored data can be modified. Most of the probabilistic homomorphic schemes are malleable by default (at least at the time when the underlying encryption exhibits homomorphism) and therefore cannot provide the desired security to the stored information. Thus, only semantic security (i.e., Chosen Plaintext Attack (CPA) security) is not sufficient to stop data modifications (even if the data in the encoded form). Therefore, the introduction of desired security notion called Chosen Ciphertext Attack (CCA) security has grabbed most attention and has now almost become the de-facto for the new security constructions to prevent several active attacks.

Therefore, there is a strong need for a combination of surjective and/or bijective mapping functions in the construction of a new encryption scheme to overcome the above problems. Motivated by this, we have proposed a new encryption model which combines both surjective and/or bijective mapping functions to support efficient encryption switching between the malleability and the non-malleability properties (Note: one property per encryption. One must re-encrypt (or switch) to get other property) during the encryption phase. The proposed model successfully covers both homomorphic and non-malleability applications with minimum computation overhead between homomorphic and non-malleability mapping.

1.2 A New Composite Function-Based Encryption Model

This paper proposes a new surjective and/or bijective composite function-based encryption model to answer the following question:

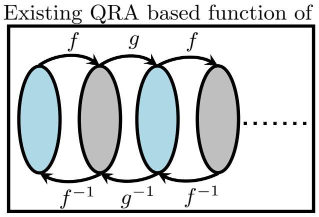
> *"Can we have a probabilistic public key-based encryption switching protocol that can efficiently switch between homomorphism and non-malleability using the surjective and/or bijective composite functions ?"*

In order to find out an efficient answer to the above question, we have proposed the following composite trapdoor functions.

Surjective and Freeman et al. function [13] combinations: For all $I \in \{0,1\}^2$ and for all $X \in \mathbb{Z}_N^{+1}$, the quadratic residuosity-based surjective function is $f : (I, X) \rightarrow Y$ where $Y \in \mathbb{Z}_N^{+1}$. For any two $i_1, i_2 \in I$, and for any $x \in X$ there exists $y \in Y$ such that $f : (i_1, X) \rightarrow y$ and $f : (i_2, X) \rightarrow y$. Therefore, the function f is surjective.

For all $Y \in \mathbb{Z}_N^{+1}$, the modified quadratic residuosity-based function of Freeman et al. is $g : Y \rightarrow Z$ where $Z \in Q_R$. For all $y \in \mathbb{Z}_N^{+1}$, $g : y \rightarrow z$ is defined as $g(y)$=y^2. The mapping $g(y)$ looses the position (i.e., [1,$\frac{N}{2}$] or [$\frac{N}{2}+1$,$N-1$]) of the input y in the ciphertext.

f= Proposed QRA based surjective function
g= Existing QRA based function of Freeman et al.



f^{-1}= Proposed QRA based inverse surjective function
g^{-1}= Existing QRA based inverse function of Freeman et al.

Fig. 2: The proposed composite mapping functions

$\forall \mathcal{M} \in \{0,1\}^n, n \in \mathbb{N}$
$\mathcal{PK}^{\mathcal{H}}$=Homomorphic public key
$\mathcal{PK}^{\mathcal{N}}$=Non-malleable public key
$\mathcal{SK}$=Private key
$\mathcal{C}^{\mathcal{H}}$=Homomorphic cipher
$\mathcal{C}^{\mathcal{N}}$=Non-malleable cipher

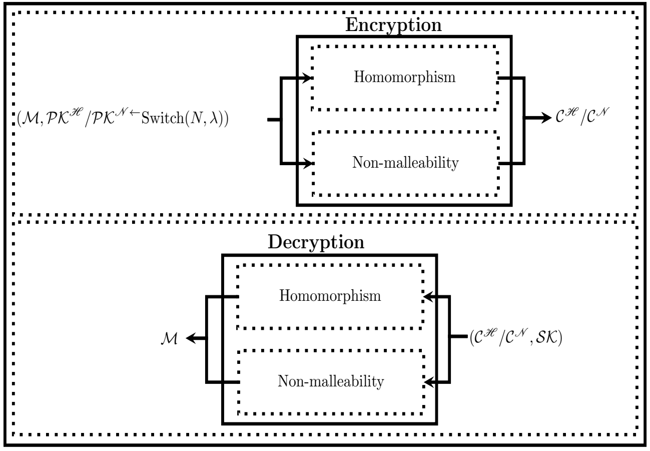

Encryption
$(\mathcal{M}, \mathcal{PK}^{\mathcal{H}}/\mathcal{PK}^{\mathcal{N}} \leftarrow$Switch$(N,\lambda))$
Homomorphism
Non-malleability
$\mathcal{C}^{\mathcal{H}}/\mathcal{C}^{\mathcal{N}}$
Decryption
$\mathcal{M}$
Homomorphism
Non-malleability
$(\mathcal{C}^{\mathcal{H}}/\mathcal{C}^{\mathcal{N}}, \mathcal{SK})$

Fig. 3: The encryption switching mechanism of the proposed scheme

In order to connect the above two functions, we have introduced a novel Plaintext Bit Selection Methods (PBSMs) and Plaintext Bit Connection Methods (PBCMs) for plaintext encryption (described in Section 3). The unique combination of i) proposed Quadratic Residuosity Assumption (QRA) based surjective functions ii) QRA-based functions of Freeman et al. iii) newly proposed plaintext bit selection and plaintext bit connection methods collectively achieve unique ciphertext generation through the composition of functions as shown in Fig. 2. Similarly, the unique combination of the respective inverse surjective function, the respective inverse function of Freeman et al., plaintext bit selection and plaintext bit connection methods collectively achieve unique plaintext retrieval through the composition of the respective inverse surjective and lossy functions. The surjective property of the proposed function f is nullified through the appropriate plaintext bit selection and plaintext bit connection methods.

In this paper, we have introduced a new probabilistic public key-based encryption switching scheme (i.e., switch between homomorphism and non-malleability during enryption phase without changing the underlying structure) in which the encryption receives the message and the switchable public key (either homomorphic public key or non-malleable public key) and outputs the respective ciphertext as shown in Fig. 3. By carefully selecting the appropriate combination of public key and the proposed composite functions, the proposed model exhibits encryption switching from homomorphic property scheme to/from non-malleability property scheme.

Let $\mathcal{PK}^{\mathscr{H}}$ and $\mathcal{PK}^{\mathscr{N}}$ are the homomorphic property supported public key component (i.e., $\mathcal{PK}^{\mathscr{H}}$) and the non-malleability property supported public key component (i.e., $\mathcal{PK}^{\mathscr{N}}$) respectively. The heart of this proposed model lies on the random generation of the specific property supported public key components and the appropriate surjective/bijective function selection during encryption process. For each encryption instance, these property specific public key components are randomly generated through a switching function called $Switch(\cdot)$ which receives the RSA composite N and a switching flag $\lambda \in \{0,1\}$ (0 indicates homomorphic, 1 indicates the non-malleability) and provides the respective property public key components to the encryption process. Finally, encryption generates a specific property supported ciphertext (either $\mathcal{C}^{\mathscr{H}}$ or $\mathcal{C}^{\mathscr{N}}$). If the switching algorithm generates the homomorphic public key components with λ=0 then the ciphertext generated after encryption is always supports homomorphic operations on it. If the switching algorithm generates the non-malleable public key components with λ=1 then the ciphertext generated after encryption is always supports the non-malleability property. The homomorphic version of the proposed model is semantically secure under quadratic residuosity assumption and higher degree residue assumptions whereas the non-malleability version of the proposed model is provably secure against passive attacks under standard integer factorization assumption.

The basic advantage of this model is that there is no change in the underlying encryption process and hence no prior exchange of any information between the communicating parties. The proposed model is completely asymmetric and therefore can be widely deployed in the presence of insecure communication channels and untrusted storage environments. This model successfully covers both homomorphic and non-malleability applications but one at a time.

The proposed encryption switching model has the following notable features.

- *Novel tricks*: The proposed model involves novel crypto (i.e., the proposed composite functions) and non-crypto techniques (i.e., the proposed PBSMs and PBCMs) to overcome several security drawbacks of the existing systems.
- *Probabilistic encryption*: The proposed model essentially involves the probabilistic encryption in which every ciphertext generated is always a result of the randomized operations of its plaintext.
- *One-wayness of the encryption functions*: As contrary to the existing systems, the one-wayness of the proposed model partially dependents on the underlying intractability assumption whereas the remaining dependency is on the underlying composite functions. But, every ciphertext will be uniquely decrypted to the intended plaintext.
- *Semantic security*: The proposed model is semantically secure if the underlying quadratic residuosity assumption is semantically secure.
- *Encryption switching*: Along with the semantic security, the proposed model supports encryption switching i.e., at the given instance, the model can behave either as the probabilistic homomorphic encryption or the probabilistic non-malleable encryption.
  - Homomorphic property: The probabilistic homomorphic encryption version of the proposed model supports the homomorphic multiplication operations on the ciphertexts.
  - Non-malleability property: The probabilistic non-malleable encryption version of the proposed model supports the non-malleability support on the ciphertexts.
- *Efficiency*: Compared with the most practical encryptions, the proposed model is almost comparable with respect to the ciphertext expansion and the running time. The encryption and key generation times of the proposed model are better than the number-theoretic encryptions such as RSA whereas the decryption time is slower than the existing encryptions. For any k bit plaintext, the proposed model generates around k+2 log N bits ciphertexts where $k \in \{0,1\}^*$, N is the RSA composite.
- *Security parameter independent plaintext space*: The proposed model involves any n-bit binary string as a plaintext. Therefore, the plaintext space is totally independent of the security parameter (Note: plaintext size $n \leq log\ N$ in case of RSA encryption where N is the RSA composite number).

1.3 Related Work

In order to overcome the secret-sharing over the insecure channel in symmetric encryption, the concept of asymmetric encryption has been formally discussed by Diffie and Hellman [10].

The seminal work of Diffie and Hellman [10] using discrete-log-problem (DLP) has been fulfilled the thirst of sharing the secure data over the insecure channel using the trapdoor-based one-way function. Various improvements including ElGamal McCurley [19], [12] on [10] have been achieved several cryptographic milestones to make the asymmetric mode of encryption more realistic and application friendly. But this family of schemes suffer from two major drawbacks. First, the plaintext space is restricted to a specific type and size. Second, the homomorphic property of these schemes is not a suitable candidate for high security applications since there exists a variety of adversaries to attack such systems.

The first success of practical result for efficient asymmetric encryption is constructed by Rivest et al. [27] popularly known as RSA. The generalized version of RSA has been constructed by Rabin [26] using the square root modulo composite number problem. Further, comprehensive research [17], [28], [31] has been carried out on these schemes to find more practical and secure systems. This class of systems has successfully achieved ciphertext size equal to plaintext size but fundamentally suffer from malleablity attacks.

In order to withstand against *line tapper*, Goldwasser-Micali [15] systematically presented the first probabilistic bit-level security with the relaxed notion of security called "semantic security" using quadratic residuosity as the underlying primitive. But, this scheme has no support to achieve a reasonable ciphertext expansion factor and also no support for non-malleability [2] feature. Several research efforts including Park and Won [23], Vanstone [29], Benaloh [3], Naccache and Stern [20], Okamoto [21] have been carried out to reduce the communication cost in these type of encryption. Notably, Blum-Goldwasser [4] have almost achieved the efficient communication cost for all large plaintext still no support has been provided for a non-malleability feature.

One more class of probabilistic asymmetric encryption has been introduced by Paillier [22] using composite residuosity problem. The ciphertext size of this scheme is twice the size of the plaintext. Several cryptographers including Cramer and Shoup [8], Damgard-Jurik [9] have put their efforts to provide communication efficient and secure schemes. Unfortunately, theis class of encryptions has also failed to provide efficient encryption switching.

In order to construct homomorphic and Chosen Ciphertext Attack (CCA) secure encryptions, many cryptographers [1], [5], [6], [16], [24], [25] have constructed almost optimal results using a variety of cryptographic primitives. But, the fundamental design requires multiple structures to provide homomorphism and CCA security.

To the best of our knowledge, no existing probabilistic schemes show (at the basic construction) the reasonable expansion factor with the efficient encryption switching capabilities. In fact, many secure Cloud storage and retrieval efforts have been proposed [14], [18], [30] using existing security techniques. But, today's cloud technologies looking towards an encryption switching protocol that supports both homomorphism and non-malleability at the protocol level with a minimum computational overhead. Though there exists some encryption switching protocols such as encryption switching protocol presented by Couteau et al. [7], there are several notable drawbacks as mentioned below.

- The encryption switching developed in [7] depends upon several security assumptions such as decisional composite residuosity, decisional Diffie-Hellman, and quadratic residuosity whereas our proposed scheme depends upon a single quadratic residuosity assumption.
- In [7], plaintext space is limited to a multiplicative group $\mathbb{Z}_N^*$ whereas plaintext space in our proposed scheme free from $\mathbb{Z}_N^*$.
- Most importantly, in [7], the encryption switching happens between two different cryptosystems whereas in our proposed scheme encryption switching happens within the same cryptosystem without altering the fundamental design.

2 Preliminaries and Notations

2.1 Notations

Let $[i] \triangleq \{1, 2, \cdot\cdot, i\}$ and $[i, j]$ is the process of selecting all the elements from i to j iteratively. Let $N \in \{0, 1\}^k$ be the RSA composite modulus with large distinct prime factors $p \equiv q \equiv 3 \pmod 4$ and Q_R denotes the quadratic residue modulo N set with *Jacobi Symbol* ($\mathcal{JS}$) 1 and $\overline{Q}_R$ denotes the quadratic non-residue modulo N set with Jacobi symbol -1. Let $\mathbb{Z}_N^{+1}$=($Q_R \cup \overline{Q}_R$) be a set of all the elements modulo N with Jacobi symbol 1 and $\mathbb{Z}_N^{-1}$ be a set of all the elements modulo N with Jacobi symbol -1. Let $\mathcal{LS}$ be a *Lagendre Symbol*. Let the plaintext be $\mathcal{M} \in \{0, 1\}^n$ where n=$\{2i : i \in \mathbb{N}\}$ is the plaintext size. Let the notation $< A, B >$ denote the ciphertext set in which $A \in \mathbb{Z}_N^{+1}$ and $B \in \{0, 1\}^l$ where $l < n$. Let p^{QR} be the quadratic residuosity assumption probability and p^R be the single fair coin toss probability. Let r, s, t, w be the public key components.

2.2 Preliminaries

2.2.1 Quadratic Residuosity

For each $y \in \mathbb{Z}_N^*$, if $x^2 \equiv y \pmod N$ (where $x \in \mathbb{Z}_N^{+1}$) then $y \in Q_R$ otherwise $y \in \overline{Q}_R$ or $y \in \mathbb{Z}_N^{-1}$.

2.2.2 Quadratic Residuosity Predicate ($\mathcal{PR}$)

For all $x \in \mathbb{Z}_N^*$, $\mathcal{PR}$ is a function to return a boolean value (0 or 1) to indicate whether "x" is Q_R if $\mathcal{PR}_{p,q}(x)$=1 or $\overline{Q}_R$ if $\mathcal{PR}_{p,q}(x)$=0.

2.2.3 Quadratic Residuosity Assumption (QRA)

Decision of the quadratic residuosity of a number modulo N is intractable in polynomial time. That is, for all probabilistic polynomial time algorithm $\mathcal{G}$, there exists a negligible function $\mathcal{F}$ such that $|\mathbb{P}[\mathcal{G}(x_{Q_R},N)= 1] - \mathbb{P}[\mathcal{G}(x_{\overline{Q}_R},N)= 1]| \leq \mathcal{F}(k)$ where k is the security parameter, x_{Q_R} is in Q_R, $x_{\overline{Q}_R}$ is in $\overline{Q}_R$ and $\mathbb{P}$ is the probability finding function.

2.2.4 QRA-based trapdoor function of Freeman et al. [13] (TF)

For all random input $x \in \mathbb{Z}_N^*$ and the public key components $s \in \overline{Q}_R$ with Jacobi Symbol -1 and $t \in \overline{Q}_R$, the quadratic residuosity-based function described in [13] is

$$x^2 \cdot s^j \cdot t^h \pmod N \tag{1}$$

where j=0 if the Jacobi symbol of x is 1 otherwise j=1. Also, h=0 if $x \leq N/2$ otherwise h=1. If the value h of the input number is stored as a "trapdoor" for all random input $x \in \mathbb{Z}_N^{+1}$, then the modified function is

$$\mathcal{TD}(x) = (x^2 \equiv\ C \pmod N)) = (C, h_x) \tag{2}$$

where h_x is the 'h' value of x as discussed in Eq. 1 and the respective inverse is $\mathcal{TD}^{-1}(C, j$=0,h_x)=$\sqrt{C}$=x.

Probabilistic Encryption: For all given random $x \in \mathbb{Z}_N^{+1}$ and random $r \in \overline{Q}_R$, for all random $\delta \in \{0, 1\}$, the modified probabilistic trapdoor function of Eq. 2 is

$$\mathcal{TD}(x) = (x^2 \cdot r^\delta \equiv\ C \pmod N)) = (C, h_x) \tag{3}$$

and the respective inverse function is defined as $\mathcal{TD}^{-1}(C$,j=0,h_x) = $\sqrt[j,h_x]{C \cdot (r)^{-\delta}}$ =x where "$\sqrt{}$" is the quadratic root finding function under modulo N, "$\cdot$" is the modular multiplication operator, "$r^{-\delta}$ when δ=1" is the modular inverse modulo N.

3 A New Algebraic Framework

3.1 QRA-based Single Bit Encryption (SBE)

Let a bit $b \in \{0,1\}$. For all random input $x, y \in \mathbb{Z}_N^{+1}$ and random public key components $r, s \in \mathbb{Z}_N^{+1}$ with $\mathscr{PR}(r) \neq \mathscr{PR}(s)$, $w \in \mathbb{Z}_N^{-1}$, the single bit encryption $\mathcal{E}_s(b, N, x, y, r, s, w)$ is given as

$$\mathcal{E}_s = \begin{cases} \begin{array}{lll} \textbf{j,h} & \textbf{If } b=0 & \textbf{If } b=1 \end{array} \\ \left.\begin{array}{lll} 0,0 & x^2 \cdot r \equiv c_1 & x^2 \cdot r \equiv c_1 \\ 0,0 & y^2 \cdot w \equiv c_2 & y^2 \cdot s \equiv c_2 \end{array}\right\} \textbf{if } x \leq \mathrm{N}/2, y \leq \mathrm{N}/2 \\ \left.\begin{array}{lll} 0,0 & x^2 \cdot r \equiv c_1 & x^2 \cdot w \equiv c_1 \\ 0,1 & y^2 \cdot r \equiv c_2 & y^2 \cdot r \equiv c_2 \end{array}\right\} \textbf{if } x \leq \mathrm{N}/2, y > \mathrm{N}/2 \\ \left.\begin{array}{lll} 0,1 & x^2 \cdot w \equiv c_1 & x^2 \cdot s \equiv c_1 \\ 0,0 & y^2 \cdot s \equiv c_2 & y^2 \cdot s \equiv c_2 \end{array}\right\} \textbf{if } x > \mathrm{N}/2, y \leq \mathrm{N}/2 \\ \left.\begin{array}{lll} 0,1 & x^2 \cdot s \equiv c_1 & x^2 \cdot s \equiv c_1 \\ 0,1 & y^2 \cdot r \equiv c_2 & y^2 \cdot w \equiv c_2 \end{array}\right\} \textbf{if } x > \mathrm{N}/2, y > \mathrm{N}/2 \end{cases} \tag{4}$$

The inputs x,$y \in \mathbb{Z}_N^{+1}$ consist of their respective j,h values as described in Eq.1. Therefore, there are four j,h possible combinations (listed in the first column of Eq. 4) for any $x, y \in \mathbb{Z}_N^{+1}$ when j=0. Encryption of b is done using the correct pair of equations. For instance, if j_x=0,h_x=0 and j_y=0,h_y=1, bit b=0 is encrypted using the pair of equations defined in second row and second column of Eq. 4; similarly, bit b=1 is encrypted using second row and third column of Eq. 4.

The decryption of $\mathcal{E}_s$ to get back bit b involves the identification of the respective quadratic residuosity properties of the ciphertexts c_1 and c_2 as follows.

- *Step-1*: Find quadratic residuosity properties of the ciphertexts c_1 and c_2 as $\mathscr{PR}(c_1)$ and $\mathscr{PR}(c_2)$. Based on the quadratic residuosity properties of the ciphertexts, output b and (j,h) combinations of x, y.
- *Step-2*: Multiply respective public key inverses to the ciphertexts to get back x^2, y^2. Then, given x^2 and (j_x, h_x) values, find unique x as described in Eq. 2. Similarly, given y^2 and (j_y, h_y) values, find unique y as described in Eq. 2.

3.2 QRA-based Bit Pair Encryption (BPE)

Let (a, b) be a bit-pair where $a, b \in \{0,1\}$ (in which a is the first bit and b is the second bit). For all random input $x \in \mathbb{Z}_N^{+1}$ and random public key components $r, s \in \mathbb{Z}_N^{+1}$ with $\mathscr{PR}(r) \neq \mathscr{PR}(s)$, random $t \in \overline{Q}_R$, the probabilistic encryption $\mathcal{E}((a,b), N, x, r, s, t)$ of the bit pair is

$$\mathcal{E} = \begin{cases} x \cdot r \cdot r \ \equiv \ y \ (\text{mod } N) \ \textbf{if } a=0, b=0 \\ x \cdot r \cdot s \ \equiv \ y \ (\text{mod } N) \ \textbf{if } a=0, b=1 \\ x \cdot t \ \ \ \ \equiv \ y \ (\text{mod } N) \ \textbf{if } a=1, b=0 \\ x \cdot s \cdot s \ \equiv \ y \ (\text{mod } N) \ \textbf{if } a=1, b=1 \end{cases} = y \tag{5}$$

For any $x \in \mathbb{Z}_N^{+1}$, the unique combinations of $r, s \in \mathbb{Z}_N^{+1}$ and $t \in \overline{Q}_R$ are given in Table 1. Since there is no pre-agreement in public key encryptions, fix any one of the combinations given in Table 1 for encryption. For convenience, let $r \in \overline{Q}_R$, $x, s \in Q_R$, and the first combination of the above table is used for the encryption of the bit-pair.

TABLE 1: Unique combinations of $r, s \in \mathbb{Z}_N^{+1}$ with $\mathscr{PR}(r) \neq \mathscr{PR}(s)$, $t \in \overline{Q}_R$ for the given $x \in \mathbb{Z}_N^{+1}$.

a b	**Comb-1**	**Comb-2**	**Comb-3**	**Comb-4**
0 0	$x \cdot r \cdot r$	$x \cdot r \cdot r$	$x \cdot r \cdot s$	$x \cdot t$
0 1	$x \cdot r \cdot s$	$x \cdot t$	$x \cdot r \cdot r$	$x \cdot r \cdot r$
1 0	$x \cdot t$	$x \cdot r \cdot s$	$x \cdot s \cdot s$	$x \cdot s \cdot s$
1 1	$x \cdot s \cdot s$	$x \cdot s \cdot s$	$x \cdot t$	$x \cdot r \cdot s$

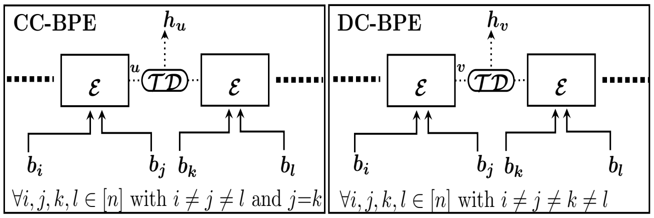


Fig. 4: Contiguous and discrete chain BPE encryptions.

Decryption: Given the ciphertext y, the second bit b (assume that the second bit b has been received by some other function) and the private key p,q, the decryption function outputs first bit a and input x as

- *Step-1*: Find the quadratic residuosity of the ciphertext y as $\mathscr{PR}(y)$.
- *Step-2*: Given $\mathscr{PR}(y)$ and second bit b, find the first bit a and input x as

$$\begin{aligned} \mathcal{E}^{\text{-}1} &= \begin{cases} a=0 \text{ and } y \cdot r^{-1} \cdot r^{-1} \equiv \ x \ \textbf{if } b=0, y \in Q_R \\ a=0 \text{ and } y \cdot r^{-1} \cdot s^{-1} \equiv \ x \ \textbf{if } b=1, y \in \overline{Q}_R \\ a=1 \text{ and } y \cdot t^{-1} \qquad \equiv \ x \ \textbf{if } b=0, y \in \overline{Q}_R \\ a=1 \text{ and } y \cdot s^{-1} \cdot s^{-1} \equiv \ x \ \textbf{if } b=1, y \in Q_R \end{cases} \\ &= \ (x, a) \end{aligned} \tag{6}$$

3.3 Contiguous Chain Bit Pair Encryption (CC-BPE)

Let l bit plaintext be P=$\{b_1, b_2, \cdots, b_l\}$. For all random input $x \in \mathbb{Z}_N^{+1}$ and random public key components $r, s \in \mathbb{Z}_N^{+1}$ with $\mathscr{PR}(r) \neq \mathscr{PR}(s)$ and random $t \in \overline{Q}_R$, the contiguous chain encryption $\mathcal{E}_{con}(P, N, x, r, s, t)$ as shown in Fig. 4 is

$$\begin{aligned} \mathcal{E}_{con} \quad &= \mathcal{E}_i((b_{d=l\text{-}c}, b_l), \mathcal{TD}_{i\text{-}1}(\mathcal{E}_{i\text{-}1}((b_{d\text{-}c}, b_d), \mathcal{TD}_{i\text{-}2}(\mathcal{E}_{i\text{-}2}) \\ &\quad))) \\ &= < y_1, y_2 = \{h_{u_1}, h_{u_2}, \cdot\cdot, h_{u_{(l\text{-}2)}}\} > \\ &= \mathcal{C}_1 \end{aligned} \tag{7}$$

where $c \in [l], 3 \leq i < l, \mathcal{E}(\cdot)$ is the BPE encryption described in Eq. 5, $\mathcal{TD}(\cdot)$ is the injective function described in Eq. 2 and $< y_1, y_2 >$ is the ciphertext set where $y_1 \in \{0,1\}^k$, y_2=$\{h_{u_1}$, h_{u_2}, $\cdot\cdot$,$h_{u_{(l\text{-}2)}}\}$. Each $u_j \in \mathbb{Z}_N^{+1}$, $j \in [1, l\text{-}1]$, is the intermediate ciphertext coming out of each $\mathcal{E}$ and each h_{u_j} is the "h" value of it. The respective decryption (i.e., $\mathcal{E}_{con}^{\text{-}1}$) of CC-BPE is simply the inverse function of $\mathcal{E}_{con}$. For instance, consider the ordered subset ($\mathcal{M}^{'} \subseteq \mathcal{M} \times \mathcal{M}$)=$\{(b_2$, $b_4)$, $(b_4$, $b_6)$,$\cdot\cdot\cdot$, $(b_{n\text{-}2}$,$b_n)\}$. The encryption of the plaintext $\mathcal{M}^{'}$ using the contiguous chain encryption $\mathcal{E}_{con}(\mathcal{M}^{'}, N, x, r, s, t)$ is given as

$$\mathcal{E}_{con} = \mathcal{E}_i((b_{d=n\text{-}2}, b_n), \mathcal{TD}_{i\text{-}1}(\mathcal{E}_{i\text{-}1}(b_{d\text{-}c}, b_d, \mathcal{TD}_{i\text{-}2}(\mathcal{E}_{i\text{-}2}))))$$

where c=2.

TABLE 2: Type of decryption

Chain	Encryption	Decryption
DC-BPE	Independent	Dependent
CC-BPE	Independent	Independent

TABLE 3: The proposed subset pairs and their respective chain pairs

Subset Pair	Chain Pair
($\mathcal{M}_1$,$\mathcal{M}_2$)	(DC-BPE,DC-BPE)
($\mathcal{M}_1$,$\mathcal{M}_3$)	(DC-BPE,CC-PBE)
($\mathcal{M}_1$,$\mathcal{M}_5$)	(CC-BPE,DC-PBE)
($\mathcal{M}_2$,$\mathcal{M}_4$)	(DC-BPE,DC-PBE)
($\mathcal{M}_2$,$\mathcal{M}_5$)	(DC-BPE,DC-PBE)
($\mathcal{M}_3$,$\mathcal{M}_4$)	(CC-BPE,CC-PBE)
($\mathcal{M}_3$,$\mathcal{M}_5$)	(CC-BPE,DC-PBE)
($\mathcal{M}_5$,$\mathcal{M}_5$)	(DC-BPE,DC-PBE)
($\mathcal{M}_5$,$\mathcal{M}_6$)	(DC-BPE,DC-PBE)
$\mathcal{M}_7$	CC-BPE

3.4 Discrete Chain Bit Pair Encryption (DC-BPE)

Let l bit plaintext be P=$\{b_1,\ b_2,\ \cdots,\ b_l\}$. For all random input $x \in \mathbb{Z}_N^{+1}$ and public key components $r, s \in \mathbb{Z}_N^{+1}$ with $\mathscr{PR}(r) \neq \mathscr{PR}(s)$ and $t \in \overline{Q}_R$, the discrete chain encryption $\mathcal{E}_{dis}(P, N, x, r, s, t)$ as shown in Fig. 4 is

$$\begin{aligned}\mathcal{E}_{dis} \quad &= \mathcal{E}_i((b_{d=l\text{-}c}, b_l), \mathcal{TD}_{i\text{-}1}(\mathcal{E}_{i\text{-}1}((b_{d\text{-}e\text{-}c}, b_{d\text{-}e}), \mathcal{TD}_{i\text{-}2}(\mathcal{E}_{i\text{-}2} \\ &\quad)))) \\ &=< y_3, y_4 = \{h_{v_1}, h_{v_2}, \cdot\cdot, h_{v(\frac{l}{2}\text{-}1)}\} > \\ &= \mathcal{C}_2\end{aligned} \tag{8}$$

where $c \in [l]$, $3 \leq i < l$, $\mathcal{E}(\cdot)$ is the BPE encryption as described in Eq. 5, $\mathcal{TD}(\cdot)$ is the modified trapdoor function described in Eq. 2 and $< y_3, y_4 >$ is the ciphertext set where $y_3 \in \{0,1\}^k$, y_4=$\{h_{v_1}$, h_{v_2}, $\cdot\cdot$,$h_{v_{(\frac{l}{2}\text{-}1)}}\}$. Each $v_j \in \mathbb{Z}_N^{+1}$, $j \in [1, l\text{-}1]$, is the intermediate ciphertext coming out of each $\mathcal{E}$ and each h_{v_j} is the "h" value of it. The respective decryption (i.e., $\mathcal{E}_{dis}^{\text{-}1}$) of DC-BPE requires an additional aid from other CC-BPE or DC-BPE chains. For instance, consider the ordered subset ($\mathcal{M}^{''} \subseteq \mathcal{M} \times \mathcal{M}$)= $\{$(b_1,b_2), (b_3,b_4), $\cdots$, ($b_{n\text{-}1}$,b_n)$\}$. The encryption of the plaintext $\mathcal{M}^{''}$ using the discrete chain encryption $\mathcal{E}_{dis}(\mathcal{M}, N, x, r, s, t)$ is given as

$$\begin{aligned}\mathcal{E}_{dis} \quad &= \mathcal{E}_i((b_{d=n\text{-}1}, b_n), \mathcal{TD}_{i\text{-}1}(\mathcal{E}_{i\text{-}1}((b_{d\text{-}2}, b_{d\text{-}1}), \mathcal{TD}_{i\text{-}2}(\mathcal{E}_{i\text{-}2} \\ &\quad))))\end{aligned}$$

where c=1, e=1.

3.5 Dependent/Independent Decryption

We call the decryption of DC-BPE chain as "dependent decryption" since every second bit of each BPE used in DC-BPE can be obtained (during decryption) from the corresponding CC-BPE or DC-BPE. We call the decryption of CC-BPE chain as "independent decryption" since every second bit of each succeeding BPE of CC-BPE is obtained (during decryption) by the preceding BPE of the same CC-BPE (Refer Table 2). Note that DC-BPE alone does not have the capability to get the second bits of its component BPEs whereas CC-BPE alone has the capability to get the second bits of its component BPEs. For instance, second bit of each BPE ($\mathcal{E}_{i\text{-}1}$) of CC-BPE of Eq. 7 is same as the first bit of each BPE ($\mathcal{E}_{i\text{-}2}$) when c=2. Also, second bit of each BPE ($\mathcal{E}_{i\text{-}1}$) of DC-BPE of Eq. 8 when c=1, e=1 is obtained by the respective BPE ($\mathcal{E}_i$) of CC-BPE of Eq. 7 when c=2.

3.6 Possible Subsets to Improve the Performance

For all n bit plaintext $\mathcal{M}$=$\{b_1$,b_2,$\cdots$,$b_n\}$, the possible ordered subsets (partial) of $\mathcal{M} \times \mathcal{M}$ are

$$\begin{aligned}\mathcal{M}_1 &= \{b_i : i = i + 2, i \in [1, n\text{-}1]\} \\ \mathcal{M}_2 &= \{b_i : i = i + 2, i \in [2, n]\} \\ \mathcal{M}_3 &= \{(b_i, b_{i+2}) : i = i + 2, i \in [2, n\text{-}2]\} \\ \mathcal{M}_4 &= \{(b_i, b_{i+2}) : i = i + 2, i \in [1, n\text{-}3]\} \\ \mathcal{M}_5 &= \{(b_i, b_{i+1}) : i = i + 2, i \in [1, n\text{-}1]\} \\ \mathcal{M}_6 &= \{(b_i, b_{i+1}) : i = i + 2, i \in [2, n\text{-}2]\} \\ \mathcal{M}_7 &= \{b_i : i = i + 1, i \in [1, n\text{-}1]\}\end{aligned} \tag{9}$$

and group a pair of above subsets in such a way that the concatenation of the bits of that subset pair should always equal to the plaintext $\mathcal{M}$. The possible pairs of such subsets are ($\mathcal{M}_1$,$\mathcal{M}_2$), ($\mathcal{M}_1$,$\mathcal{M}_3$), ($\mathcal{M}_1$,$\mathcal{M}_5$), ($\mathcal{M}_2$,$\mathcal{M}_4$), ($\mathcal{M}_2$,$\mathcal{M}_5$), ($\mathcal{M}_3$,$\mathcal{M}_4$), ($\mathcal{M}_3$,$\mathcal{M}_5$), ($\mathcal{M}_5$,$\mathcal{M}_5$), ($\mathcal{M}_5$,$\mathcal{M}_6$). In addition, a single subset $\mathcal{M}_7$ can also be used to encrypt the given plaintext (using contiguous chain bit-pair encryption). We use one of the pairs ($\mathcal{M}_1$,$\mathcal{M}_3$) throughout this paper to explain the proposed scheme.

3.7 Decryption Dependent Ciphers

We call y_1, y_2 of CC-BPE or y_3, y_4 of DC-BPE as "dependent ciphers" since they are completely dependent on each other during decryption.

Definition 1. *A Probabilistic Public Key Cryptosystem is a 3-tuple (*`KG`*,*`E`*,*`D`*) scheme consists of two probabilistic polynomial time (PPT) algorithms* `KG`*,* `E` *and a deterministic algorithm* `D` *described as follows.*

- *Key Generation (*`KG`*): Given a random security parameter k, algorithm generates a randomized public and private key pair ($\mathcal{PK}, \mathcal{SK}$).*
- *Encryption (*`E`*): Chooses a public key $\mathcal{PK} \overset{R}{\longleftarrow} \mathbb{Z}_N^*$ with certain quadratic residuosity property, a message $\mathcal{M} \in \mathbb{S}$ and generates a ciphertext $\mathcal{C}$=$\mathtt{E}(\mathcal{PK}, \mathcal{M})$.*
- *Decryption (*`D`*): Given the secret key $\mathcal{SK}$ and ciphertext $\mathcal{C}$, algorithm generates the same message $\mathcal{M}$ as $\mathcal{M}$=$\mathtt{D}(\mathcal{SK}, \mathcal{C})$.*

Indistinguishable Property: We say the ciphertexts are *indistinguishable* if any two ciphertexts $\mathcal{C}_1$=$\mathtt{E}(\mathcal{PK}_1, \mathcal{M})$ and $\mathcal{C}_2$=$\mathtt{E}(\mathcal{PK}_2, \mathcal{M})$ generated from `E` are computationally indistinguishable under the standard quadratic residuosity assumption (QRA) proposed in [15]. That is, for all PPT adversary $\mathcal{A}$, there exists a negligible function $\mathcal{F}$ such that $|\mathbb{P}[\mathcal{A}(\mathscr{PR}(\mathcal{PK}_1)$,$N)=\ 1] - \mathbb{P}[\mathcal{A}(\mathscr{PR}(\mathcal{PK}_2), N)=\ 1]| \leq \mathcal{F}(k)$ where k is the security parameter, $\mathscr{PR}$ is the quadratic residuosity predicate function and $\mathbb{P}$ is the probability finding function.

Correctness: We say that PKE satisfies *correctness* if for all $(\mathcal{PK}, \mathcal{SK}) \overset{R}{\longleftarrow}$ $\mathtt{KG}(1^k)$, $\mathbb{P}[\mathtt{D}(\mathcal{SK}, \mathtt{E}(\mathcal{PK}, \mathcal{M})) = \mathcal{M}]$=1 (where the randomness is taken over the internal coin tosses of algorithm `E`).

Definition 2. *The public-key encryption scheme described in Definition 1 is said to be semantically secure if for any PPT distinguisher $\mathcal{A}$ and any pair of messages $\mathcal{M}_0$, $\mathcal{M}_1$, given the* *public key $\mathcal{PK}$, the advantage for distinguishing $\mathcal{C}_0$ = $\mathtt{E}(\mathcal{PK}, \mathcal{M}_0)$ and $\mathcal{C}_1$ = $\mathtt{E}(\mathcal{PK}, \mathcal{M}_1)$ is negligible in security parameter. In other words, the above said scheme is semantically secure until underlying quadratic residuosity assumption is semantically secure.*

Definition 3. *Let $(G_1,\cdot)$, $(G_2,*)$ be groups. Let $\mathtt{E}$ be the probabilistic encryption algorithm and $\mathtt{D}$ be the decryption algorithm of an encryption scheme with plaintext set G_1 and ciphertext set G_2. The encryption scheme defined in Definition 1 is said to be group homomorphic if the encryption map $\mathtt{E} : G_1 \rightarrow G_2$ has the following property: $\forall \mathcal{M}_0, \mathcal{M}_1 \in G_1$, $\mathtt{D}(\mathtt{E}(\mathcal{M}_0 \cdot \mathcal{M}_1))$=$\mathtt{D}(\mathtt{E}(\mathcal{M}_0) \cdot \mathtt{E}(\mathcal{M}_1))$*

Definition 4. *[NM-CPA, NM-CCA1, NM-CCA2] Let NM-CPA, NM-CCA1, NM-CCA2 are non-malleable chosen plaintext attack, non-malleable chosen ciphertext attack1, non-malleable chosen ciphertext attack2 respectively. Let Π = $(\mathtt{KG}, \mathtt{E}, \mathtt{D})$ be an encryption scheme defined in Definition 1 and let C = (C_1, C_2) be an adversary consisting of a pair of algorithms. For atk $\in \{cpa, cca1, cca2\}$ and $k \in N$ define $Adv^{nm-atk}_{C,\pi}(k)$=$\mathbb{P}[Expt^{atk-1}_{C,\pi}(k) \Rightarrow 1]$-$\mathbb{P}[Expt^{atk-0}_{C,\pi}(k) \Rightarrow 1]$ where*

```
$Expt^{nm-atk-1}_{C,\pi}(k)$
$(\mathcal{PK}, \mathcal{SK}) \xleftarrow{R} \mathtt{KG}(1^k)$
$(\mathbb{S}, s) \xleftarrow{R} C_1^{\mathcal{O}_1}(\mathcal{PK})$
$\mathcal{M} \xleftarrow{R} \mathbb{S}$
$\mathcal{C}_1 \xleftarrow{R} \mathtt{E}(\mathcal{PK}, \mathcal{M})$
$(R, y) \xleftarrow{R} C_2^{\mathcal{O}_2}(s, \mathcal{C}_1)$
$x \xleftarrow{R} \mathtt{D}(\mathcal{SK}, y)$
if $\mathcal{M} == x$ then
    return 1
else
    return 0
end if
```

```
$Expt^{nm-atk-0}_{C,\pi}(k)$
$(\mathcal{PK}, \mathcal{SK}) \xleftarrow{R} \mathtt{KG}(1^k)$
$(\mathbb{S}, s) \xleftarrow{R} C_1^{\mathcal{O}_1}(\mathcal{PK})$
$\mathcal{M} \xleftarrow{R} \mathbb{S}$; $\widetilde{\mathcal{M}} \xleftarrow{R} \mathbb{S}$
$\mathcal{C}_1 \xleftarrow{R} \mathtt{E}(\mathcal{PK}, \widetilde{\mathcal{M}})$
$(R, \widetilde{y}) \xleftarrow{R} C_2^{\mathcal{O}_2}(s, \widetilde{\mathcal{C}}_1)$
$\widetilde{x} \xleftarrow{R} \mathtt{D}(\mathcal{SK}, \widetilde{y})$
if $\mathcal{M} == \widetilde{x}$ then
    return 1
else
    return 0
end if
```

and If atk = cpa then $\mathcal{O}_1(\cdot)$=ϵ and $\mathcal{O}_2(\cdot)$=ϵ. If atk = cca1 then $\mathcal{O}_1(\cdot)$=$\mathtt{D}(\mathcal{SK}, \cdot)$ and $\mathcal{O}_2(\cdot)$=ϵ. If atk = cca2 then $\mathcal{O}_1(\cdot)$=$\mathtt{D}(\mathcal{SK}, \cdot)$ and $\mathcal{O}_2(\cdot)$=$\mathtt{D}(\mathcal{SK}, \cdot)$.

4 New Probabilistic Single Structure Encryption Switching Protocol (sESP)

In this section, we propose quadratic residuosity-based asymmetric encryptions as defined in Definition 1. Let the plaintext $\mathcal{M}$=$\{b_1, b_2, \cdot\cdot\cdot, b_n\}$. We use the subset ($\mathcal{M}_1$,$\mathcal{M}_3$) of Table 3 to explain the proposed scheme and all the remaining subset pairs can also be encrypted in a similar fashion using their respective chains. The overall encryption process consists of two steps. In the first step, encrypt the subset $\mathcal{M}_1$ using CC-BPE chain and encrypt the subset $\mathcal{M}_3$ using DC-BPE chain. In the second step, encrypt the last bit b_n using the ciphertexts obtained from the first step as inputs to SBE function. The detailed description is as follows.

- **Key Generation ($\mathtt{KG}$):** Given the security parameter k, select the RSA composite modulus $N \in \{0,1\}^k$ with the large distinct prime factors p and q with $p \equiv q \equiv 3$ (mod 4). Choose the random numbers r,$s \in \mathbb{Z}^{+1}_N$ with $\mathscr{PR}(r) \neq \mathscr{PR}(s)$ and choose a random $t \in \overline{Q}_R$ and choose a random $w \in \mathbb{Z}^{-1}_N$. Also, choose a random input $x \in \mathbb{Z}^{+1}_N$. The public key is (N,x,r,s,t,w) and the private key is (p,q).
- **Encryption ($\mathtt{E}$):** For all plaintext $\mathcal{M}$ and the public key (N,x,r,s,t,w), the encryption $\mathtt{E}(\mathcal{M})$ is given as

$$\mathtt{E}(\mathcal{M}) = \begin{cases} \mathcal{E}_{con}(\mathcal{M}_1) =< Y_1, Y_2 = \{h_{u_1}, \cdot\cdot, h_{u_{(\frac{n}{2}\text{-}2)}}\} > \\ \qquad = \mathcal{C}_1 \\ \mathcal{E}_{dis}(\mathcal{M}_3) =< Y_3, Y_4 = \{h_{v_1}, \cdot\cdot, h_{v_{(\frac{n}{2}\text{-}1)}}\} > \\ \qquad = \mathcal{C}_2 \\ \qquad \text{and then do} \\ \mathcal{E}_s(b_n, N, Y_1, Y_3, r, s, w) = \{Z_1, Z_2\} \end{cases} \tag{10}$$

Therefore, the final ciphertexts are $\mathcal{C}_3$=$\{Z_1, Y_2\}$ and $\mathcal{C}_4$=$\{Z_2, Y_4\}$. The pictorial representation of the encryption process is given in Fig. 5.
- **Decryption ($\mathtt{D}$):** Given the ciphertexts ($\mathcal{C}_3$, $\mathcal{C}_4$) and the private key (p,q), the decryption $\mathtt{D}(\mathcal{C}_3, \mathcal{C}_4, p, q)$ is given as

$$\begin{aligned} &\begin{cases} \mathcal{E}^{\text{-}1}_s(Z_1, Z_2, p, q) = \{b_n, Y_1, Y_3\} \\ \qquad \text{and then do} \\ \mathcal{E}^{\text{-}1}_{con}(b_n, \mathcal{C}_1, p, q) = (\mathcal{M}_1 = \{b_2, b_4, b_6 \cdot\cdot, b_{n\text{-}2}\}, x) \\ \mathcal{E}^{\text{-}1}_{dis}(b_n, \mathcal{C}_2, p, q) = (\mathcal{M}_3 = \{b_1, b_3, b_5 \cdot\cdot, b_{n\text{-}1}\}, x) \end{cases} \\ &= \mathcal{M}_1 \cup \mathcal{M}_3, x \\ &= \mathcal{M}, x \end{aligned} \tag{11}$$

and the pictorial representation of the decryption process is given in Fig. 6.

4.1 Independent Decryption Scheme Using CC-BPE method

It is evident that no subset alone can be decrypted to generate all the plaintext bits except the subset $\mathcal{M}_5$ (Refer Eq. 9). Therefore, in order to reduce the dependency among the subsets, the single subset $\mathcal{M}_5$ can be encrypted as $\mathcal{E}_{con}(\mathcal{M}_5, N, x, r, s, t)$ using the CC-BPE encryption technique described in Eq. 7 as

$$\mathcal{E}_{con} = \mathcal{E}_i((b_{d=n\text{-}1}, b_n), \mathcal{TD}_{i\text{-}1}(\mathcal{E}_{i\text{-}1}((b_{d\text{-}1}, b_d), \mathcal{TD}_{i\text{-}2}(\mathcal{E}_{i\text{-}2})))) \tag{12}$$

Example: Consider N=133, p=19, q=7 and plaintext $\mathcal{M}$={1,1,0, 0,1,1,1,1} where $|\mathcal{M}|$=n=8. Consider $\mathcal{M}_1$={(1,0), (0,1), (1,1)} and $\mathcal{M}_3$={(1,1), (0,0), (1,1), (1,1)}. Let x=25, r=39, s=34, t=41, w=29. The complete encryption and decryption process is given in Table 4a and Table 4b.

4.2 Probabilistic Single Structure Encryption Switching Signature Scheme (sESPSig)

The proposed scheme of Section 4 can also be used for generating the digital signatures. The detailed description is as follows.

- **Key Generation ($\mathtt{KG}$):** Given security parameter k, select the RSA composite modulus $N \in \{0,1\}^k$ with the large prime factors p and q. Choose random quadratic residue $r \in Q_R$ and random quadratic non-residue $s \in \overline{Q}_R$. Also, choose random input $x \in \mathbb{Z}^{+1}_N$. The public key is (N,x,r,s) and the private key is (p,q).
- **Signature Creation ($\mathtt{D}$)** For any non-negative integer n, select two numbers (Y_1,Y_3)$\xleftarrow{R} \mathbb{Z}^{+1}_N$ and select two numbers $Y_2 \xleftarrow{R} \{0,1\}^{\frac{n}{2}\text{-}2}$, $y_4 \xleftarrow{R} \{0,1\}^{\frac{n}{2}\text{-}1}$. Given a random message

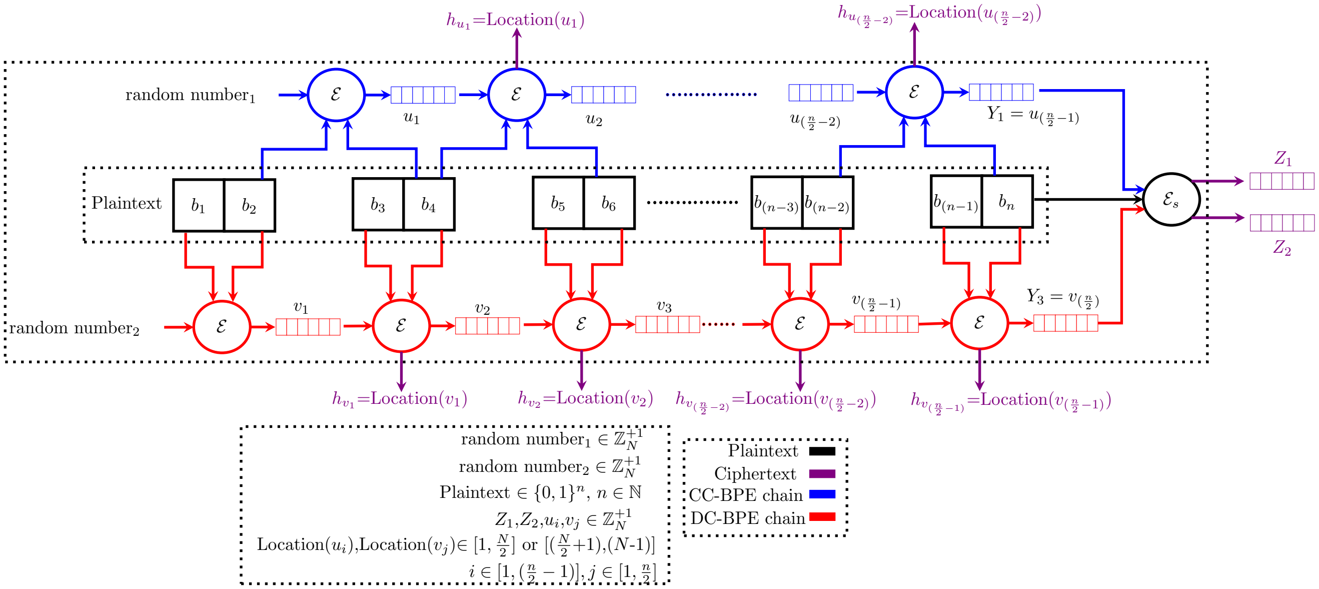

h_{u_1}=Location(u_1)
$h_{u_{(\frac{n}{2}-2)}}$=Location($u_{(\frac{n}{2}-2)}$)
random number$_1$
$\mathcal{E}$
u_1
u_2
$u_{(\frac{n}{2}-2)}$
$Y_1 = u_{(\frac{n}{2}-1)}$
Plaintext
b_1 b_2 b_3 b_4 b_5 b_6 $b_{(n-3)}$ $b_{(n-2)}$ $b_{(n-1)}$ b_n
$\mathcal{E}_s$
Z_1
Z_2
random number$_2$
v_1
v_2
v_3
$v_{(\frac{n}{2}-1)}$
$Y_3 = v_{(\frac{n}{2})}$
h_{v_1}=Location(v_1)
h_{v_2}=Location(v_2)
$h_{v_{(\frac{n}{2}-2)}}$=Location($v_{(\frac{n}{2}-2)}$)
$h_{v_{(\frac{n}{2}-1)}}$=Location($v_{(\frac{n}{2}-1)}$)
random number$_1$ $\in \mathbb{Z}_N^{+1}$
random number$_2$ $\in \mathbb{Z}_N^{+1}$
Plaintext $\in \{0,1\}^n$, $n \in \mathbb{N}$
Z_1,Z_2,u_i,v_j $\in \mathbb{Z}_N^{+1}$
Location(u_i),Location(v_j)$\in$ [1, $\frac{N}{2}$] or [($\frac{N}{2}$+1),(N-1)]
$i \in [1,(\frac{n}{2}-1)], j \in [1,\frac{n}{2}]$
Plaintext
Ciphertext
CC-BPE chain
DC-BPE chain

Fig. 5: Abstract view of encryption process.

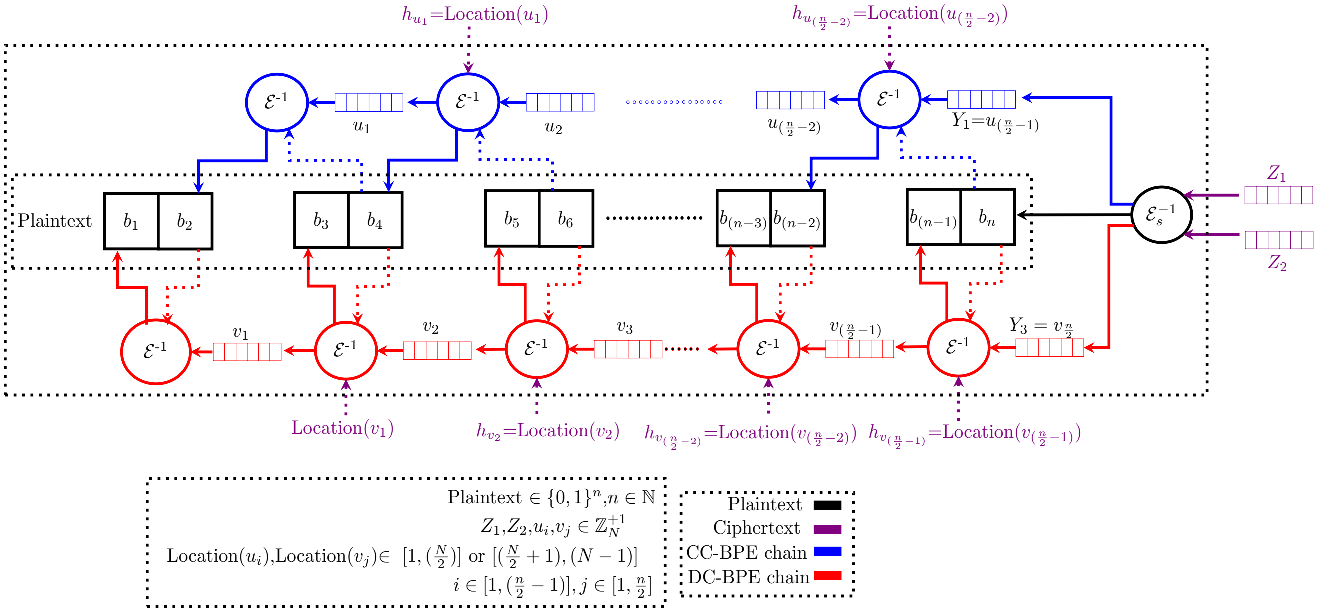

h_{u_1}=Location(u_1)
$h_{u_{(\frac{n}{2}-2)}}$=Location($u_{(\frac{n}{2}-2)}$)
$\mathcal{E}^{-1}$
u_1
u_2
$u_{(\frac{n}{2}-2)}$
Y_1=$u_{(\frac{n}{2}-1)}$
Plaintext
b_1 b_2 b_3 b_4 b_5 b_6 $b_{(n-3)}$ $b_{(n-2)}$ $b_{(n-1)}$ b_n
$\mathcal{E}_s^{-1}$
Z_1
Z_2
v_1
v_2
v_3
$v_{(\frac{n}{2}-1)}$
$Y_3 = v_{\frac{n}{2}}$
Location(v_1)
h_{v_2}=Location(v_2)
$h_{v_{(\frac{n}{2}-2)}}$=Location($v_{(\frac{n}{2}-2)}$)
$h_{v_{(\frac{n}{2}-1)}}$=Location($v_{(\frac{n}{2}-1)}$)
Plaintext $\in \{0,1\}^n$,$n \in \mathbb{N}$
Z_1,Z_2,u_i,v_j $\in \mathbb{Z}_N^{+1}$
Location(u_i),Location(v_j)$\in$ $[1,(\frac{N}{2})]$ or $[(\frac{N}{2}+1),(N-1)]$
$i \in [1,(\frac{n}{2}-1)], j \in [1,\frac{n}{2}]$
Plaintext
Ciphertext
CC-BPE chain
DC-BPE chain

Fig. 6: Abstract view of decryption process.

($\mathcal{C}_1$=\{Y_1,Y_2\}, $\mathcal{C}_2$=\{Y_3,y_4\}) and the private key (p,q), the signature creation algorithm D($\mathcal{C}_1$,$\mathcal{C}_2$) generates the signature $\mathcal{M}$ as

$$\begin{cases} b_n \xleftarrow{R} \{0,1\} \\ \mathcal{E}^{\text{-}1}_{con}(b_n,\mathcal{C}_1,p,q)=\{b_2,b_4,b_6\cdot\cdot,b_{n\text{-}2}\}=\mathcal{M}_1 \\ \mathcal{E}^{\text{-}1}_{dis}(b_n,\mathcal{C}_2,p,q)=\{b_1,b_3,b_5\cdot\cdot,b_{n\text{-}1}\}=\mathcal{M}_3 \end{cases} \tag{13}$$
$$=(\mathcal{M}_1\cup\mathcal{M}_3)=\mathcal{M}$$

where |$\mathcal{M}$|=n.

- **Verification (**E**)** Given the signature $\mathcal{M}$, message ($\mathcal{C}_1$, $\mathcal{C}_2$) and the public key (N,x,r,s), the verification algorithm finds E($\mathcal{M}$) and verifies whether E($\mathcal{M}$)=($\mathcal{C}_1$, $\mathcal{C}_2$) as

$$\begin{cases} \mathcal{E}_{con}(\mathcal{M}_1) =< Y_1, Y_2 = \{h_{u_1},\cdot\cdot,h_{u_{(\frac{n}{2}\text{-}2)}}\} >= \mathcal{C}_1 \\ \text{and} \\ \mathcal{E}_{dis}(\mathcal{M}_3) =< Y_3, Y_4 = \{h_{v_1},\cdot\cdot,h_{v_{(\frac{n}{2}\text{-}1)}}\} >= \mathcal{C}_2 \end{cases} \tag{14}$$

4.3 Level of Security

Since the proposed scheme uses quadratic residuosity properties, in order to reveal the plaintext $\mathcal{M}$, the adversary has to following i) find the properties of x, r, s ii) find the last bit from Eq. 4 iii) find the remaining bits from each BPE with p^{QR} probability and each TF with p^{QR} probability. The proposed scheme is secure until the underlying quadratic residuosity assumption is secure.

TABLE 4: A toy example of the proposed scheme

(a) A toy example of the encryption process

Encryption($\mathtt{E}(\mathcal{M})$)		
Step-1		**Step-2**
$\mathcal{E}_{dis}(\mathcal{M}_3, N, x, r, s, t)$	$\mathcal{E}_{con}(\mathcal{M}_1, N, x, r, s, t)$	$\mathcal{E}_s(b_n,Y_1,Y_3)$
$\mathcal{E}((1,1), 133, 25, 39, 34, 41)$=39 $\Downarrow$ $\mathcal{TD}$(39)=(58,0) $\Downarrow$ $\mathcal{E}((0,0), 133, 58, 39, 34, 41)$=39 $\Downarrow$ $\mathcal{TD}$(39)=(58,0) $\Downarrow$ $\mathcal{E}((1,1), 133, 58, 39, 34, 41)$=16 $\Downarrow$ $\mathcal{TD}$(16)=(123,0) $\Downarrow$ $\mathcal{E}((1,1), 133, 123, 39, 34, 41)$=11	$\mathcal{E}((1,0), 133, 25, 39, 34, 41)$=94 $\Downarrow$ $\mathcal{TD}$(94)=(58,1) $\Downarrow$ $\mathcal{E}((0,1), 133, 58, 39, 34, 41)$=34 $\Downarrow$ $\mathcal{TD}$(34)=(92,0) $\Downarrow$ $\mathcal{E}((1,1), 133, 92, 39, 34, 41)$=3	$3^2 \cdot 39$=85 11^2 · 34=124
$\mathcal{C}_2$={Y_3=11,Y_4=(0,0,0)}	$\mathcal{C}_1$={Y_1=3,Y_2=(1,0)}	Z_1=85 ,Z_2=124
Therefore, ciphertexts are $\mathcal{C}_3$={Z_1, Y_2}, $\mathcal{C}_4$={Z_2, Y_4}		

(b) A toy example of the decryption process

Decryption($\mathtt{D}(\mathcal{C}_3 = \{Z_1, Y_2\}, \mathcal{C}_4 = \{Z_2, Y_4\})$)		
Step-2		**Step-1**
$\mathcal{E}^{-1}_{dis}(b_n, \mathcal{C}_2, p, q)$	$\mathcal{E}^{-1}_{con}(b_n, \mathcal{C}_1, p, q)$	$\mathcal{E}^{-1}_s(Z_1, Z_2)$
$\mathcal{E}^{-1}$(1,11,19,7)=(123,1) $\Downarrow$ $\mathcal{TD}^{-1}(123, 0, 0) = 16$ $\Downarrow$ $\mathcal{E}^{-1}$(1,16,19,7)=(58,1) $\Downarrow$ $\mathcal{TD}^{-1}(58, 0, 0) = 39$ $\Downarrow$ $\mathcal{E}^{-1}$(0,39,19,7)=(58,0) $\Downarrow$ $\mathcal{TD}^{-1}(58, 0, 0) = 39$ $\Downarrow$ $\mathcal{E}^{-1}$(1,39,19,7,130,97)=(25,1)	$\mathcal{E}^{-1}$(1,3,19,7)=(92,1) $\Downarrow$ $\mathcal{TD}^{-1}(92, 0, 0) = 34$ $\Downarrow$ $\mathcal{E}^{-1}$(1,34,19,7)=(58,0) $\Downarrow$ $\mathcal{TD}^{-1}(58, 0, 1) = 94$ $\Downarrow$ $\mathcal{E}^{-1}$(0,94,19,7)=(25,1)	85$\in Q_R$ 124$\in \overline{Q}_R$
$\mathcal{M}_3$={(1,1),(0,0),(1,1),(1,1)}	$\mathcal{M}_1$={(1,0),(0,1),(1,1)}	b_n=1,Y_1=3 ,Y_3=11

TABLE 5: Performance comparison with the existing schemes

(a) Performance comparison for all $n \in \mathbb{N}$ bit plaintext schemes and $k \in \mathbb{N}$ bit plaintext schemes

Scheme	Year	Ciphertext	Ciphertext when n=k	Message Space	[‡]Homomorphism	[‡]Non-malleability
Goldwasser-Micali	1984	$n \cdot k$	k^2	Bit string	Yes	No
Blum-Goldwasser	1985	$n + k$	$2k$	Bit string	Yes	No
RSA	1978	k	k	Restricted to k	Yes	No
Rabin	1979	k	k	Restricted to k	Yes	No
Okamoto and Uchiyama	1998	$3k$	$3k$	Restricted to k	Yes	No
Paillier	1999	$2k$	$2k$	Restricted to k	Yes	No
Proposed Scheme	2021	$\boldsymbol{m}$**+**$\boldsymbol{2k}$	$\approx 2.99k$	**Bit string**	**Yes***	**Yes***

Note: $m < n$ and *Encryption Switching⇒ Exhibits both the properties but one at a time. Also, note that [‡]Homomorphism/[‡]Non-malleability here implies that the scheme which support these properties at their basic constructions.

(b) Performance of the proposed scheme when n=k

Scheme	Year	Multiplications		Multiplications per bit		Plaintext	Ciphertext
		Encrypt	Decrypt	Encrypt	Decrypt		
RSA	1978	17	$3k$+3	$17/k$	3/2	k	k
Paillier	1999	$3k$+1	$3k/2$	3	3	k	$2k$
El-gamal	2006	$3k$+1	$(3k/2)k$+1	3	3/2	k	k
Damgard-Jurik	2001	$(3k/2)$+5	$5k$+8	3/2	5	k	$2k$
Proposed Scheme	2021	**3k-1**	**3k-1**	$\boldsymbol{\approx 3 - (5/k)}$	$\boldsymbol{\approx 3 - (5/k)}$	k	$\approx 2.99k$

Note: Performance of the existing scheme is taken from Dumgard and Jurik [9]

In addition to basic security, the proposed scheme also provides additional benefit to reduce the plaintext revealing probability of the adversary further which is most suitable for secure cloud storage applications.

5 Performance Evaluation of the Proposed Scheme

5.0.1 Expansion factor

The generic expansion factor which is applicable for all the subset pairs mentioned in 9 is defined as f=(m+d log $N)/n$ where $m < n$ and d is the total number of ciphertexts generated from encryption process. In the proposed scheme, if the security parameter k=log N=512 and the plaintext size n=512 then m=($|Y_1|$ + $|Y_3|$)=509 and $d\cdot$ log N=2·log N. Therefore, the ciphertext expansion factor f=(509+ 2·log N)=2.99 (as shown in Table 5a). Similarly, if k=512, n=1024 then m=1021 and f=1.99. Also, if k=1024, n=2048 then m=2045 and f=1.99. Therefore, for all n=2k, f=1.99. Similarly, for all n=3k, f=1.66. For all n=4k, f=1.49. In general, for all n ≥20k, f ≤1. Even though the ciphertext size of the proposed scheme is little higher than the existing schemes (Refer Table 5), it can reach the ciphertext size as k for all large message with n ≥20k.

5.0.2 Execution time

Since there are $(n - 1)$ BPE functions involved in the encryption process of the proposed scheme and each BPE involves maximum of two modular multiplications, there are 2$(n - 1)$ multiplications. Along with BPE functions, there are $(n - 3)$ TF functions are also involved in the encryption process proposed scheme and each TF function involves one modular squaring. Also, there is one SBE function used in the encryption process proposed scheme and it has four modular multiplications. Therefore, in total,there are $2(n - 1)$+$(n - 3)$+4=($3n - 1$) modular multiplications involved in each encryption and decryption process. Since SBE involves constant number of multiplications (i.e., four), just ignore that. Each bit thus involves $(3n-5)/n$ number of

TABLE 6: Performance of all possible subset pairs of the plaintext $\mathcal{M}$

Pair	Multiplications		Per bit Multi.		Parallel Execution		m	d
	Encrypt	Decrypt	Encrypt	Decrypt	Encrypt	Decrypt		
($\mathcal{M}_1$,$\mathcal{M}_3$)	$3n{-}1$	$3n{-}1$	$\approx 3{-}(5/n)$	$\approx 3{-}(5/n)$	Yes	No	$n{-}3$	2
($\mathcal{M}_1$,$\mathcal{M}_2$)	$3n$	$3n$	$\approx 3{-}(8/n)$	$\approx 3{-}(8/n)$	Yes	Yes	$n{-}4$	4
($\mathcal{M}_2$,$\mathcal{M}_4$)*	$3n{-}1$	$3n{-}1$	$\approx 3{-}(5/n)$	$\approx 3{-}(5/n)$	Yes	No	$n{-}3$	2
($\mathcal{M}_3$,$\mathcal{M}_4$)	$3n{-}1$	$3n{-}1$	$\approx 3{-}(5/n)$	$\approx 3{-}(5/n)$	Yes	No	$n{-}3$	2
$\mathcal{M}_5$	$3n{-}1$	$3n{-}1$	$\approx 3{-}(5/n)$	$\approx 3{-}(5/n)$	No	No	$n{-}2$	2

*Note: For this pair, n=$\{3i : i \in \mathbb{N}\}$

modular multiplications during encryption and decryption process. Each encryption and decryption process of the proposed scheme involves $(3n - 5)/n$ number of modular multiplications plus $(n - 3)$ number of $\mathscr{PR}$ functions. Refer Table 5 for further details.

The performance of other subset pairs are tabulated in Table 6. The communication complexity is almost similar in all subset pairs. But, there is a difference in the capability of parallel execution. All the subset pairs except $\mathcal{M}_5$ can be encrypted independently (with separate chains) in encryption whereas the subset pair ($\mathcal{M}_1$,$\mathcal{M}_2$) can even be decrypted independently in decryption (Note: both $\mathcal{M}_1$,$\mathcal{M}_2$ are associated with CC-BPE chains. The CC-BPE chains have the capability of independent decryption). More precisely, during the encryption process, each chain (generally there are two chains involved) can be executed in parallel in ($\mathcal{M}_1$,$\mathcal{M}_3$)/($\mathcal{M}_1$,$\mathcal{M}_2$)/($\mathcal{M}_2$,$\mathcal{M}_4$)/($\mathcal{M}_3$,$\mathcal{M}_4$) subset pair. But, during the decryption process, one chain has to wait for the other chain in ($\mathcal{M}_1$,$\mathcal{M}_3$)/($\mathcal{M}_2$,$\mathcal{M}_4$)/($\mathcal{M}_3$,$\mathcal{M}_4$) subset pair. Therefore, only the scheme involves the subset pair ($\mathcal{M}_1$,$\mathcal{M}_2$) can be executed in parallel during encryption and decryption process. Also, there is no question of parallel execution in case of the subset $\mathcal{M}_5$, since there is only one contiguous chain involved.

5.0.3 Tamper proof facility

Due to the existence of the injective mapping, for all given plaintext $\mathcal{M}$ and public key (N, x, r, s, t, w), the encryption always produces a unique property ciphers $\mathcal{C}_3$,$\mathcal{C}_4$. Similarly, given the ciphers and private key (p, q), the decryption always produces unique plaintext $\mathcal{M}$ and input x as described in Eq. 11. Even a single bit change in $\mathcal{C}_3$,$\mathcal{C}_4$ does not produce unique plaintext $\mathcal{M}$ and input x. This great feature suits well for secure cloud storage applications where the user can verify the tampering in the stored data. Hence, the proposed scheme certainly provides inbuilt tamper proof feature to the decrypting party about the change in the stored ciphertexts.

5.0.4 Encryption switching

One of the useful feature of the proposed scheme is the ability to shift from homomorphism (in tern malleable) to/from non-malleability. This switching capacity will cover both homomorphic to non-malleable applications. As the best of our knowledge, no existing probabilistic encryption schemes (at their basic construction) provide this kind of switching feature without altering the underlying security structure.

The proposed scheme uses the injective function of Eq. 2. Suppose, if TF from Eq. 3 is chosen, then the adversarial probability can be further reduced. By carefully observing the quadratic residuosity properties of input and public key components, it is clear from Table 1 that there is a change to opt the random combination during encryption process and hence the adversarial probability can be further reduced if the combinations is chosen randomly during encryption process. But, adversarial probability cannot be reduced in the proposed scheme since it uses the fixed combination. Also, there is a possibility to choose any subset pair during encryption process. If the subset pair is selected uniformly at random during encryption, the adversary probability can be reduced further. There are 256 unique combinations of the equations of $\mathcal{E}_s$ that can be used to encrypt the last bit of the plaintext. If the purpose of the proposed scheme is to provide asymmetric encryption using the public key of other party then any one of 256 combinations can be used. But, if the purpose of the proposed scheme is to provide asymmetric encryption for secure user data storage on insecure cloud, then any one of the combinations can be selected at random. This random selection creates additional effort for the adversary to reveal the information.

5.0.5 Security extension by pre-storing ciphertext bits

The greatest advantage of the proposed scheme in secure cloud applications is to keep some of the ciphertext bits before storing on the insecure cloud. This additional feature greatly reduces the change of revealing the plaintext since the size of the ciphertext would be partially known and the locations of these bits are unknown to the cloud. In fact, this feature is unavailable in the existing encryption schemes.

6 Implementation and Results

Since the encryption process of the proposed scheme supports multi-threaded execution of encryption and decryption, the implementation of the encryption involves the multi-threaded execution of some of its parts. In fact, this multi-threading feature helps in reducing the overall execution time. Since $\mathcal{E}_{con}$ and $\mathcal{E}_{dis}$ are independent encryption chains used in encryption process (as shown in Fig. 5), these are executed with two concurrent threads. Further, each encryption chain ($\mathcal{E}_{con}$ or $\mathcal{E}_{dis}$) is executed in two steps. In the first step, given the plaintext bits, all the public key component multiplications (i.e., $(r \cdot r)$ or $(r \cdot s)$ or $(s \cdot s)$) for each bit pair are calculated in a single unit of time (please note that the unit time is the time required for a single multiplication). In particular, $(n - 1)$ multiplications from $\mathcal{E}_{dis}$ and $(n - 3)$ multiplications from $\mathcal{E}_{con}$ are concurrently executed in a single unit of time. In the second step, the remaining $((n-3)+2)$ modular multiplications are calculated sequentially. Therefore, the total time required to complete the encryption process is 1 unit from first step plus $(n-3$+2)

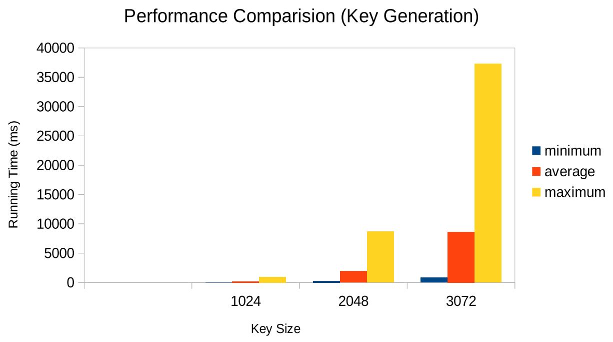


Fig. 7: The performance comparison (Key generation)

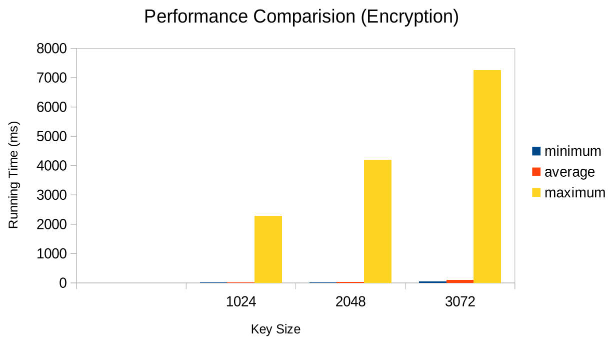


Fig. 8: The performance comparison (Encryption)

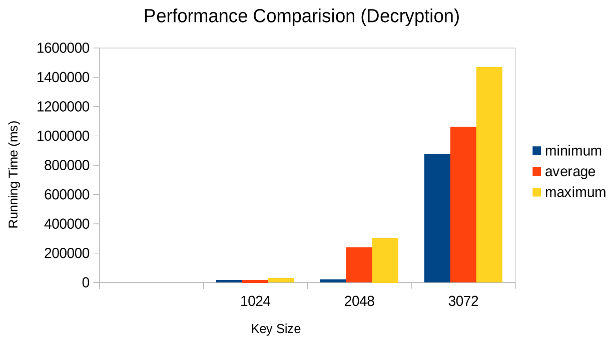


Fig. 9: The performance comparison (Decryption)

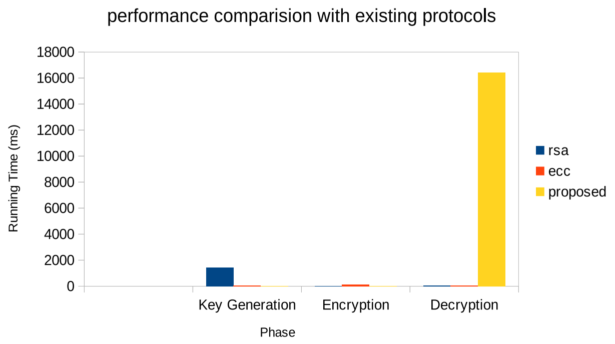


Fig. 10: The performance comparison of proposed scheme with existing protocols.

units from second step = n units. However, decryption process involves very less multi-threading facility compared to encryption. Also, the decryption chains are completely depend on each other and they cannot be executed in parallel. In any given unit of time, only two bit-pair decryptions (each from each chain) can be executed in parallel. Also, the major time consuming part in decryption is the calculation of the quadratic residuosity property and quadratic square roots for each bit pair. Since the whole decryption process involves $(\frac{n}{2}+2)$ inverse multiplications, $(\frac{n}{2}+2)$ quadratic residuosity property calculations and $(\frac{n}{2}-1)$ quadratic square root calculations, the total unit time required to get back the plaintext is $(\frac{n}{2}+2)$ inverse multiplication time plus $(\frac{n}{2}-1)$ quadratic square root calculation time. Therefore, decryption process comparatively takes more time than the encryption.

We have implemented the proposed scheme of Section 4 on the following hardware configurations: Intel Core i5-8265U CPU with 1.60GHz$*$8 processor, 64-bit Ubuntu operating system, 8GB RAM and software configurations: Java programming language on eclipse IDE, BigInteger package for large number generation. The running time performance of the proposed scheme in key generation, encryption and decryption processes are tabulated in Table 8. The pictorial representation of key generation, encryption and decryption process performances are shown in Fig. 7, Fig. 8 and Fig. 9 respectively.

The performance comparison of the proposed scheme with the existing security protocols as shown in Fig. 10 clearly shows that both key generation and encryption running times of the proposed scheme are better than RSA and ECC. The only time consuming part is the decryption. This slow running part is very much helpful for user-centric secure storage applications such as Healthcare record storage on untrusted Cloud. In fact, cloud has to invest huge amount of computation in order to reveal the information of the user data because of this slow running process. Therefore, the overall performance of the proposed scheme is reasonably well compared to the existing public key protocols.

7 A Tamper Evident Secure Storage and Retrieval Method on Insecure Cloud

It is intuitive that the proposed scheme provides a tamper evidence to the stored data when it is used to store the information securely over the untrusted cloud. Consider a scenario where there are only two entities namely *Alice* (client) and *Cloud* (server) in which *Alice* wants to store her private information securely over the *curious and untrusted* cloud as shown in Fig. 11. The proposed scheme effectively provides the solution to this scenario with highest security by choosing the non-malleable version (Refer DDN [11] for Non-malleability definition) of the proposed scheme. Let Alice selects her private message $\mathcal{M}$. Let *Alice* generates the (public,private) key pair and encrypts the message with her public key as $\mathtt{E}(\mathcal{M})$ and produces the ciphertexts $\mathcal{C}_3$=$\{Z_1, Y_2\}$, $\mathcal{C}_4$=$\{Z_2, Y_4\}$ as described in Eq. 10. Now, *Alice* stores the *dependent ciphers* Y_2, Y_4 on the untrusted cloud and

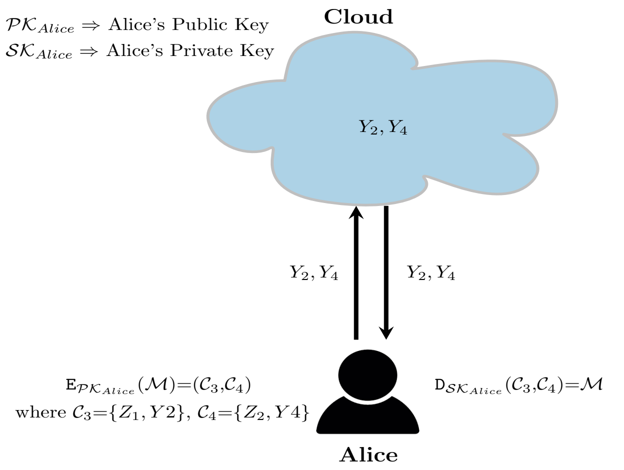

$\mathcal{PK}_{Alice}$ ⇒ Alice's Public Key
$\mathcal{SK}_{Alice}$ ⇒ Alice's Private Key
Cloud
Y_2, Y_4
Y_2, Y_4
Y_2, Y_4
$\mathtt{E}_{\mathcal{PK}_{Alice}}(\mathcal{M})$=($\mathcal{C}_3$,$\mathcal{C}_4$)
where $\mathcal{C}_3$={$Z_1, Y2$}, $\mathcal{C}_4$={$Z_2, Y4$}
$\mathtt{D}_{\mathcal{SK}_{Alice}}$($\mathcal{C}_3$,$\mathcal{C}_4$)=$\mathcal{M}$
Alice

Fig. 11: A secure cloud storage and retrieval

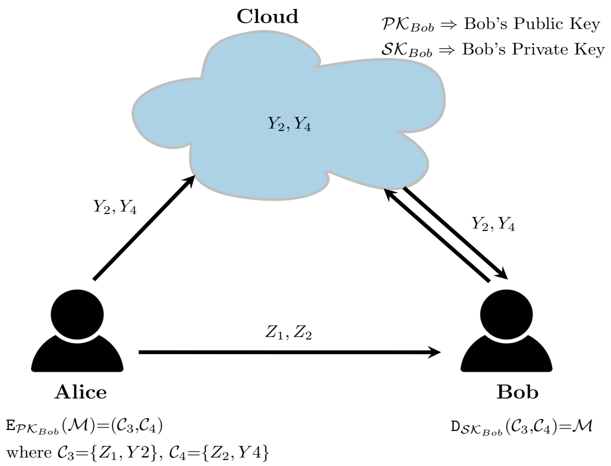

Cloud
$\mathcal{PK}_{Bob}$ ⇒ Bob's Public Key
$\mathcal{SK}_{Bob}$ ⇒ Bob's Private Key
Y_2, Y_4
Y_2, Y_4
Y_2, Y_4
Z_1, Z_2
Alice
Bob
$\mathtt{E}_{\mathcal{PK}_{Bob}}(\mathcal{M})$=($\mathcal{C}_3$,$\mathcal{C}_4$)
where $\mathcal{C}_3$={$Z_1, Y2$}, $\mathcal{C}_4$={$Z_2, Y4$}
$\mathtt{D}_{\mathcal{SK}_{Bob}}$($\mathcal{C}_3$,$\mathcal{C}_4$)=$\mathcal{M}$

Fig. 12: A secure-sharing using cloud storage

keeps Z_1, Z_2 with her. This method provides highest data security to Alice because of two reasons. First, each bit of the ciphers Y_2, Y_4 does not provide any information other than the location (i.e., whether it belongs to $[1, \frac{N}{2}]$ or $[\frac{N}{2}$+1,N-1]). Second, even a bit change in the ciphers Y_2, Y_4 will intimate *Alice* due to the existence of tamper proof support. This method also reduces the space overhead problem to Alice since $|Y_2|$+$|Y_4| \approx |\mathcal{M}|$ and $|Z_1|$=$|Z_2|$=k where k is the security parameter. To reveal the private message of Alice, the Cloud has the negligible probability due to the existence of several random functions and inability to access other dependent ciphers Z_1, Z_2.

Consider one more scenario where there are three participating entities namely, *Alice*, *Bob* and *Cloud* where Alice encrypts his private message using Bob's public key using Eq. 10 and stores the dependent ciphers (i.e., Y_2, Y_4) on the untrusted Cloud and sends the remaining dependent ciphers (i.e., Z_1, Z_2) to Bob as shown in Fig. 12. Finally, Bob downloads Y_2, Y_4 and decrypts Alice's message $\mathcal{M}$ using Eq. 11.

7.1 Patient-Cloud-Doctor Application

The above scenario is analogous to patient, doctor, cloud in which patient wants to send his private health record to his doctor using cloud as a storage media.

7.2 Author-Cloud-Editor Application

Using the above three party communication setting, assume a scenario where *Author* (Alice) wants to send his research paper to the *Editor-in-Chief* (Bob) for anonymous review by securely storing his paper at a common access point (*Cloud*). In this case, the author creates a hash of the author(s) details using the existing hashing algorithm. Then, using editor's public key, author encrypts his paper using Eq. 10 of the proposed scheme and stores the dependent ciphers (i.e., Y_2, Y_4) securely at a common access point (Cloud) and sends other dependent ciphers (i.e., Z_1, Z_2) to the editor as explained in the above scenario. The Editor, downloads the ciphertext from Cloud and decrypts the paper using Eq. 11 of the proposed scheme and initiate the review process in an anonymous way (Because, neither the paper contains author details nor cloud stores complete ciphertext. Only the hash contains that). When the paper is accepted, the author sends the author details to the editor and the editor generates a fresh hash of the received author details and verifies with the received hash. This method successfully hides author(s) details on both editor as well as cloud side and stores the data securely on the untrusted cloud.

8 Conclusion and Future Work

We have successfully presented the quadratic residuosity-based probabilistic encryption switching scheme along with some of its suitable secure storage applications. The overall performance of the proposed scheme including the bandwidth, encryption switching property are comparable with the existing schemes. Further, investigation is required on the extension of number of plaintext subsets of the plaintext and their effect on the overall performance. In addition, investigating per bit multiplication reduction and the chosen ciphertext security support is the future direction.

9 Acknowledgements

We thank Sartaj Sahni (university of Florida), Phoha (Syracuse University) and Sh Ching Chen (Florida International University) for their helpful comments.

References

[1] Jee Hea An, Yevgeniy Dodis, and Tal Rabin. On the security of joint signature and encryption. In *Advances in Cryptology — EUROCRYPT 2002*, pages 83–107, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[2] Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations among notions of security for public-key encryption schemes. In *Advances in Cryptology — CRYPTO '98*, pages 26–45, 1998.

[3] Josh Benaloh. Dense probabilistic encryption. 1997.

[4] Manuel Blum and Shafi Goldwasser. An efficient probabilistic public-key encryption scheme which hides all partial information. In *Advances in Cryptology*, pages 289–299. Springer Berlin Heidelberg, 1985.

[5] Ran Canetti, Hugo Krawczyk, and Jesper B. Nielsen. Relaxing chosen-ciphertext security. In *Advances in Cryptology - CRYPTO 2003*, pages 565–582, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

TABLE 7: Performance of the proposed encryption scheme (in milliseconds) for various key size

Type	n=1024			n=2048			n=3072		
	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg
Key Generation	31	940	175	260	8680	1956	818	37305	8630
Encryption	3.53	2283.31	5.18	15.08	4189.13	24.70	50.3	7241.37	98.361
Decryption	16420	29474	17448	19490.08	304515	237824.7	873067	1467392	1062166

Note: encryption has been executed with 10000 trials whereas the remaning are executed with 1000 trials for n=1024,2048,3072.

TABLE 8: Performance comparison of the proposed scheme (in milliseconds) with existing schemes.

Scheme	**Key Size**	**KeyGen**	**Encryption**	**Decryption**
RSA	1024	1432	4.28	48.5
ECC	168	65	140	67
Proposed	1024	31	3.53	16420

[6] Ran Canetti, Srinivasan Raghuraman, Silas Richelson, and Vinod Vaikuntanathan. Chosen-ciphertext secure fully homomorphic encryption. In *Public-Key Cryptography – PKC 2017*, pages 213–240, Berlin, Heidelberg, 2017. Springer Berlin Heidelberg.

[7] Geoffroy Couteau, Thomas Peters, and David Pointcheval. Encryption switching protocols. In *CRYPTO*, pages 308–338. Springer, 2016.

[8] Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In *Advances in Cryptology — EUROCRYPT 2002*, pages 45–64. Springer Berlin Heidelberg, 2002.

[9] Ivan Damgård and Mats Jurik. A generalisation, a simplification and some applications of paillier's probabilistic public-key system. In *PKC '01*, pages 119–136. Springer-Verlag, 2001.

[10] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Trans. Information Theory*, 22(6):644–654, 1976.

[11] Danny Dolev, Cynthia Dwork, and Moni Naor. Non-malleable cryptography. In *SIAM Journal on Computing*, pages 542–552, 2000.

[12] T. Elgamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Trans. Inf. Theor.*, 31(4):469–472, 2006.

[13] David Mandell Freeman, Oded Goldreich, Eike Kiltz, Alon Rosen, and Gil Segev. More constructions of lossy and correlation-secure trapdoor functions. Cryptology ePrint Archive, Report 2009/590, 2009. http://eprint.iacr.org/2009/590.

[14] Xiang Gao, Jia Yu, Wen-Ting Shen, Yan Chang, Shi-Bin Zhang, Ming Yang, and Bin Wu. Achieving low-entropy secure cloud data auditing with file and authenticator deduplication. *Information Sciences*, 546:177–191, 2021.

[15] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270 – 299, 1984.

[16] Jens Groth. Rerandomizable and replayable adaptive chosen ciphertext attack secure cryptosystems. In *Theory of Cryptography*, pages 152–170, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[17] Kenji Koyama, Ueli M. Maurer, Tatsuaki Okamoto, and Scott A. Vanstone. New public-key schemes based on elliptic curves over the ring zn. In *Advances in Cryptology — CRYPTO '91*, pages 252–266. Springer Berlin Heidelberg, 1992.

[18] Suhui Liu, Jiguo Yu, Yinhao Xiao, Zhiguo Wan, Shengling Wang, and Biwei Yan. Bc-sabe: Blockchain-aided searchable attribute-based encryption for cloud-iot. *IEEE Internet of Things Journal*, 7(9):7851–7867, 2020.

[19] Kevin S. McCurley. A key distribution system equivalent to factoring. *Journal of Cryptology*, 1(2):95–105, 1988.

[20] David Naccache and Jacques Stern. A new public key cryptosystem based on higher residues. In *Proceedings of the 5th ACM Conference on Computer and Communications Security*, CCS '98, pages 59–66. ACM, 1998.

[21] Tatsuaki Okamoto and Shigenori Uchiyama. A new public-key cryptosystem as secure as factoring. In *Advances in Cryptology — EUROCRYPT'98*, pages 308–318. Springer Berlin Heidelberg, 1998.

[22] Pascal Paillier. *Public-Key Cryptosystems Based on Composite Degree Residuosity Classes*, pages 223–238. Springer Berlin Heidelberg, 1999.

[23] S. Park and D. Won. A generalization of public-key residue cryptosystem. In *In Proceedings of 1993 Korean-Japan Joint Workshop on Information Security and Cryptology*, pages 202–206, 1993.

[24] Manoj Prabhakaran and Mike Rosulek. Rerandomizable rcca encryption. In *Advances in Cryptology - CRYPTO 2007*, pages 517–534, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[25] Manoj Prabhakaran and Mike Rosulek. Homomorphic encryption with cca security. In *Automata, Languages and Programming*, pages 667–678, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[26] M. O. Rabin. Digitalized signatures and public-key functions as intractable as factorization. Technical report, Massachusetts Institute of Technology, 1979.

[27] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978.

[28] Peter J. Smith and Michael J. J. Lennon. Luc: A new public key system. In *Proceedings of the IFIP TC11, Ninth International Conference on Information Security: Computer Security*, IFIP/Sec '93, pages 103–117, 1993.

[29] S. A. Vanstone and R. J. Zuccherato. Elliptic curve cryptosystems using curves of smooth order over the ring zn. *IEEE Transactions on Information Theory*, 43(4):1231–1237, 1997.

[30] Jibin Wang, Zhigang Zhao, Zhaogang Xu, Hu Zhang, Liang Li, and Ying Guo. I-sieve: An inline high performance deduplication system used in cloud storage. *Tsinghua Science and Technology*, 20(1):17–27, 2015.

[31] H C Williams. Some public key crypto-functions as intractable as factorization. In *Proceedings of CRYPTO 84 on Advances in Cryptology*, pages 66–72. Springer-Verlag New York, Inc., 1985.

Radhakrishna Bhat received his B.E. and integrated Ph.D. (M.Tech.+Ph.D) degrees in 2011 and 2020 from Visveswaraya Technological University (V.T.U), India. He is currently working as an Assistant Professor in the Department of Computer Science and Engineering, Manipal Institute of Technology, MAHE, Manipal, India. His research interests include information security and High Performance Computing.

N R Sunitha received her B.E. in Department of E and C, Gulbarga University, India. She received his M.S. from BITS, Pilanai, India. She is currently working as Professor, Dept. of CSE, Siddaganga Institute of Technology, India. She received her Ph.D. from Visvesvaraya Technological University, India. Her research interests include cryptography and network security and Industrial automation.

S S Iyengar is currently the Distinguished University Professor, Ryder Professor of Computer Science and Director of the School of Computing and Information Sciences at Florida International University (FIU), Miami. His research interests include High-Performance Algorithms, Biomedical Computing, Sensor Fusion, and Intelligent Systems for the last four decades. His research has been funded by the National Science Foundation (NSF), Defense Advanced Research Projects Agency (DARPA) and various state agencies and companies. He is a Fellow of IEEE, Fellow of ACM, Fellow of AAAS, Fellow of SDPS, Fellow of NAI.