Direct Range Proofs for Paillier Cryptosystem and Their Applications

Zhikang Xie
The University of Hong Kong
zkxiecs@gmail.com

Mengling Liu
The Hong Kong Polytechnic University
mengling.liu@connect.polyu.hk

Haiyang Xue
Singapore Management University
haiyangxc@gmail.com

Man Ho Au
The Hong Kong Polytechnic University
mhaau@polyu.edu.hk

Robert H. Deng
Singapore Management University
robertdeng@smu.edu.sg

Siu-Ming Yiu
The University of Hong Kong
smyiu@cs.hku.hk

ABSTRACT

The Paillier cryptosystem is renowned for its applications in electronic voting, threshold ECDSA, multi-party computation, and more, largely due to its additive homomorphism. In these applications, range proofs for the Paillier cryptosystem are crucial for maintaining security, because of the mismatch between the message space in the Paillier system and the operation space in application scenarios.

In this paper, we present novel range proofs for the Paillier cryptosystem, specifically aimed at optimizing those for both Paillier plaintext and affine operation. We interpret encryptions and affine operations as commitments over integers, as opposed to solely over $\mathbb{Z}_N$. Consequently, we propose direct range proof for the updated cryptosystem, thereby eliminating the need for auxiliary integer commitments as required by the current state-of-the-art. Our work yields significant improvements: In the range proof for Paillier plaintext, our approach reduces communication overheads by approximately 60%, and computational overheads by 30% and 10% for the prover and verifier, respectively. In the range proof for Paillier affine operation, our method reduces the bandwidth by 70%, and computational overheads by 50% and 30% for the prover and verifier, respectively. Furthermore, we demonstrate that our techniques can be utilized to improve the performance of threshold ECDSA and the DCR-based instantiation of the Naor-Yung CCA2 paradigm.

KEYWORDS

Paillier cryptosystem; Range proof; Multiplicative-to-Additive function; Threshold ECDSA; Naor-Yung CCA2; Sigma protocol

1 INTRODUCTION

The Paillier encryption scheme introduced by [38] is a prototypical example of additively homomorphic encryption, and has also been standardized by ISO/IEC [1]. The additive homomorphism enables one to compute the encryption of the sum of two plaintexts directly from their respective ciphertexts without the need for prior decryption. Besides, given an encrypted message, the ciphertext of its multiplication with a known number can also be computed publicly. The additive homomorphism of Paillier encryption makes it well-suited for applications such as threshold ECDSA [32, 33], electronic voting [13, 30], private auction [35], general multiparty computation [16], and etcetera. Denote the encryption of message m as $\mathsf{Enc}(m)$, then the additive homomorphism of Paillier encryption can be represented as

$$\mathsf{Enc}(m_1) \cdot \mathsf{Enc}(m_2) = \mathsf{Enc}(m_1 + m_2),$$
$$\mathsf{Enc}(m)^{\mu} = \mathsf{Enc}(\mu \cdot m).$$

1.1 Necessity of Range Proof for Paillier

In scenarios where Paillier encryption is utilized, it is often necessary to ensure some value like the plaintext is in a desired interval. In these cases, a range proof is essential, which is a specific type of zero-knowledge proofs, allowing a prover to convince a verifier that a committed value falls within a certain range while revealing nothing else. We show the critical role of range proofs in the scenarios of threshold ECDSA and Naor-Yung paradigm as examples.

Threshold ECDSA. Threshold ECDSA [26] is an extension of ECDSA that distributes the power of signing into a threshold group of parties. It can enhance the security and robustness by preventing the problem of single-point-of-failure. Consequently, it serves as a crucial component in the forthcoming standardization process for Multi-Party Threshold Cryptography being conducted by NIST [37]. Existing solutions (e.g., [6, 33, 41]) identify a basic building block, Multiplicative-to-Additive functionality (denoted by MtA henceforth), which allows two parties to securely compute A and B from their private inputs a and b respectively, ensuring $A + B = ab \bmod q$, where q denotes the order of the underlying elliptic curve in ECDSA.

MtA can be constructed by leveraging the homomorphic properties of Paillier encryption. Nonetheless, a significant challenge arises from the mismatch between the desired domain $\mathbb{Z}_q$ for homomorphic operations, and the message space $\mathbb{Z}_N$ of Paillier encryption. Ensuring correctness and security requires that no reduction modulo N occurs for all operations throughout the entire MtA protocol. Hence, range proofs are necessary to validate this condition. The absence of range proofs exposes MtA to concrete attacks, as detailed in [33, Section 6.2].

Naor-Yung Paradigm. The CCA2 (Chosen-Ciphertext Attacks) security is recognized as the standard security notion for public key encryption schemes. Naor and Yung [36] proposed a generic method that can transform a public key encryption scheme, initially secure only against Chosen-Plaintext Attacks (CPA), to one that achieves CCA2 security. The core idea is that the receiver holds two key pairs, and the sender encrypts the message under both public

keys and provides a zero-knowledge proof of plaintext equivalence in the resulting ciphertexts.

Nevertheless, situation becomes complicated when the framework is applied to the CPA-secure Paillier encryption. This stems from the distinct RSA moduli in the two pairs of keys, which could be leveraged by a malicious adversary to prove false statements, as shown in [17]. This problem can be overcome by integrating an auxiliary proof, which demonstrates that the plaintext is smaller than each respective RSA modulus, employing well-established range proof techniques, as suggested by [17].

1.2 Range Proof Methods for Paillier

In this paper, we identify and consider two types of range proofs for Paillier cryptosystem. The first type is for Paillier plaintext, aiming to prove that the plaintext under a given ciphertext falls within a certain range. The second type is for Paillier affine operation. Given a Paillier encryption $C_b = \mathsf{Enc}(b)$, a Paillier affine operation involves computing $(C_b)^a \cdot \mathsf{Enc}(A)$ with private input a and A, resulting in the encryption of $ab + A$. Here, the focus is on demonstrating that the values of a and A lie within certain ranges. For simplicity, this section primarily focuses on the first type for Paillier plaintext, as the principles for the second type involving Paillier affine operations are similarly applied.

The main obstacle to building range proofs for Paillier plaintext is that the Paillier encryption inherently supports proving relations only over $\mathbb{Z}_N$ rather than $\mathbb{Z}$. This limitation is problematic in the context of range proofs. For instance, the Lagrange's 4-square theorem shows that an integer is non-negative if and only if it can be represented as a sum of 4 squares, but a 4-square decomposition over $\mathbb{Z}_N$ does not suffice to establish non-negativity [34]. Typical solutions for this issue include the range proof from Lindell and Nof [33], and that from Devevey *et al.* [18], both of which consist of two main steps: first, leveraging tools that allow to prove relations over $\mathbb{Z}$; and second, applying range proof techniques (suitable over $\mathbb{Z}$) to these tools.

Lindell and Nof [33] employ an auxiliary integer commitment scheme [12, 25] to force the prover to argue over $\mathbb{Z}$. To demonstrate the range of underlying plaintext, the prover firstly commits to the same plaintext using an integer commitment, and then provides a proof of equivalence, demonstrating that the commitment indeed corresponds to the same value. Secondly, the CFT range proof [8], which has been refined by [2, Section 1.2.3], is applied to the commitment to demonstrate the range of the committed message, which is indeed the plaintext.

Borrowing the idea from [9], Devevey *et al.* [18] first interpret Paillier ciphertexts as bounded integer commitments which allow to prove relations over $\mathbb{Z}$. Next, the 3-square-decomposition-based range proof [28], developed from the 4-square decomposition method [34] and Boudot's technique [2], is adopted to prove the range of plaintext.

The range proofs for Paillier plaintext and affine operation introduced by Lindell and Nof [33] have slack, which means that the soundness range is larger than the range specified in the statement. Fortunately, in applications where such slack is acceptable, range proofs with slack are favored due to their significant efficiency compared to exact ones. However, this approach introduces an auxiliary integer commitment scheme to serve as a bridge, resulting in additional communication and computational overheads. Conversely, the range proof for Paillier plaintext from Devevey *et al.* [18] is exact. A drawback of this method is the inefficiency: additional commitments are needed (e.g., those for the 3 decomposed numbers), which considerably increases the proof size and computational cost compared with range proofs that have slack. Besides, their scheme does not accommodate affine operations. Furthermore, to interpret Paillier ciphertexts as bounded integer commitments, they introduce modifications to the original Paillier encryption. As a result, the homomorphic properties are only partially retained. The compatibility of this approach with applications that require both Paillier's homomorphic features and range proofs, such as the instantiation of MtA, is still unknown. Table 1 presents a comparison among different range proof methods for the Paillier cryptosystem.

Table 1: Comparison of Range Proofs for Paillier

Methods	Plaintext		Affine Operation		Exact	Direct
	Comp.	Commu.	Comp.	Commu.		
LN [33]	9	1.22	13.5	2.12	×	×
DLP [18]	36.5	4.37	×	×	✓	×
Ours	6	0.47	8	0.60	×	✓

Notes: We use $|x|$ to denote the bit length of element x. The unit for computation (comp.) is an exponentiation operation within $\mathbb{Z}_N^*$ with exponent of bit length $|N|$, while that for communication (commu.) is kibibytes (KiB). The comparison is based on a typical choice of parameters. Specifically, the statistical and soundness parameters are 80 and 128, respectively. $|N| = 3072$. As to range proofs for plaintext, the range is $[0, B]$ with $|B| = 256$. For affine operation $(\mathsf{Enc}(b))^a \cdot \mathsf{Enc}(A)$, the ranges for a and A are $[0, B_1]$ and $[0, B_2]$, with $|B_1| = 256$ and $|B_2| = 800$, respectively. Here we don't even account for the costs of 3-square decomposition in [18], which are necessary in practice and will bring additional computational overheads. When a range proof scheme does not require extra commitments, we refer to it as 'direct'.

1.3 Motivation and Contributions

Range proofs for Paillier are essential building blocks for important applications, such as constructing MtA in threshold ECDSA and the instantiation of the Naor-Yung CCA2 paradigm, as previously discussed. Thus, investigating methods to enhance their performance is valuable, as such improvements will directly benefit the applications that rely on them. As indicated in Table 1, Lindell and Nof [33] have proposed a more efficient range proof for Paillier plaintext compared to that of Devevey *et al.* [18], by introducing a degree of slack. Fortunately, this level of slack is acceptable in several applications, including the two discussed before.

Nevertheless, the approach by Lindell and Nof still requires auxiliary integer commitments to bridge the gap between Paillier encryption and CFT range proof. This situation prompts an important question: Is it possible to conduct range proofs directly on the Paillier cryptosystem, particularly for plaintext and affine operations, without relying on any auxiliary commitments?

Motivated by the above, this paper focuses on investigating how to design direct range proofs for Paillier cryptosystem, aiming to

improve efficiency and further benefit applications built on them. The contributions of this paper are summarized as follows:

(1) Firstly, we propose a new commitment scheme over (vectors of) integers based on Paillier cryptosystem alongside a modified version of Paillier encryption scheme, which have unique and advantageous properties. Specifically, our slight modifications to the Paillier encryption enable an encrypting operation to be interpreted as committing to the plaintext over integers $\mathbb{Z}$ under our proposed commitment scheme, rather than being limited to $\mathbb{Z}_N$. This opens the door for proving relations over integers and further designing direct range proof for the plaintext. As we will discuss later, an affine operation within the Paillier framework will also align with our commitment scheme, specifically in terms of committing to an affine operation over $\mathbb{Z}\times\mathbb{Z}$. It is also important to highlight that our modified Paillier encryption remains compatible with the existing decryption algorithm, eliminating the need for any updates to the decryption process.

(2) Secondly, we introduce more efficient range proofs for the Paillier cryptosystem. To achieve this, we first design a direct range proof for our commitment scheme. Then, due to the connection among our modified Paillier encryption, affine operations, and our commitment scheme, the range proof for the commitment can be easily applied to giving direct range proofs for Paillier plaintexts and affine operations. Additionally, we provide a comprehensive analysis regarding its correctness and security. The security of our proof is based on the strong RSA assumption, and we emphasize that, although our approach is inspired by the concept of CFT proof [8], the task is far from straightforward, since we have to operate within a different mathematical structure $\mathbb{Z}^*_{N^2}$. Moreover, to ensure that our commitment scheme is compatible with the encryption scheme, we introduce an additional component $(1+N)$, which is not present in the original integer commitment scheme.

(3) Furthermore, we implement our schemes in GO. The benchmark shows that our range proofs outperform state-of-the-art ones, in both computation and communication, for both Paillier plaintext and affine operation.

(4) Finally, we show improvements to practical applications by our techniques, specifically threshold ECDSA and Naor-Yung CCA2 paradigm, that rely on the Paillier cryptosystem and related range proofs. Notably, our proposed MtA protocol outperforms the state-of-the-art one based on the Paillier cryptosystem. Besides, we show that our instantiation of Naor-Yung CCA2 paradigm also has advantages in computation and ciphertext size.

1.4 Technical Overview

In this study, we focus on strong RSA modulus $N = pq$, which means $p = 2p' + 1, q = 2q' + 1$ for distinct primes p, p', q, q'. Recall that our goal is to develop direct range proofs for Paillier cryptosystem, particularly concerning plaintext and affine operation. At the heart of our proposed solution is an innovative commitment scheme tailored for (vectors of) integers, coupled with a modified version of the Paillier encryption scheme. This commitment scheme is designed to enable direct proof of the value's range. Furthermore, with our modified Paillier encryption, both encryption and affine operation can be treated as commitment actions, and thus the corresponding ranges can be directly proven.

Integer Commitment & Encryption. We begin by addressing the direct range proof for Paillier plaintext. Subsequently, the range proof for affine operations can be viewed as an extension of this initial approach. The core idea involves treating a Paillier ciphertext as a commitment over the integers $\mathbb{Z}$, rather than solely over $\mathbb{Z}_N$. Recall that a typical Paillier ciphertext in $\mathbb{Z}^*_{N^2}$ is of the form

$$C = (1+N)^m \cdot r^N \bmod N^2, \tag{1}$$

where $m \in \mathbb{Z}_N$ is the plaintext, and $r \in \mathbb{Z}^*_N$ is the randomness.

Roughly, we may require an integer commitment scheme where the commitments are also within $\mathbb{Z}^*_{N^2}$ so that they could also be treated as ciphertexts. However, for a typical integer commitment scheme [25] that uses the quadratic residue group QR_N with hidden order, the commitments lie in $\mathbb{Z}^*_N$. Specifically, let h and g be random generators of QR_N, where both the discrete logarithm of h in base g and that of g in base h are unknown. The integer commitment of $m \in \mathbb{Z}$ is

$$c = h^m \cdot g^r \bmod N,$$

for some randomness $r \in \mathbb{Z}_N$. To modify this for use within $\mathbb{Z}^*_{N^2}$, we lift h and g to generators of the $2N$-th residues over $\mathbb{Z}^*_{N^2}$, denoted by $2\mathrm{NR}_{N^2}$, which is another hidden order group. The integer commitment for $m \in \mathbb{Z}$ is updated accordingly to

$$c = h^m \cdot g^r \bmod N^2, \tag{2}$$

where $r \in \mathbb{Z}_N$ is the randomness. Intuitively, this adaptation is well-formed due to the isomorphism between QR_N and $2\mathrm{NR}_{N^2}$. Refer to Fact 1 for details.

However, the adaptation remains insufficient because, obviously, such a commitment scheme lacks the capability for decryption. Thus, a Paillier ciphertext cannot be interpreted as this type of commitment. To address this, we relax the statistical hiding requirement to computational security by incorporating the component $(1+N)$ into h. Specifically, let $y = h(1+N) \bmod N^2$, and define the commitment as

$$c = y^m \cdot g^r \bmod N^2, \tag{3}$$

which possesses the characteristics of both a ciphertext (referred to as modified Paillier encryption) and an integer commitment. Specifically, it retains the nature of an integer commitment similar to Equation (2) under the Decisional Composite Residuosity (DCR) assumption, while the inclusion of the $(1+N)$ component introduces the capability for decryption, similar to the original Paillier encryption scheme as shown in Equation (1).

Assume that Alice, who is in possession of the public key but not the secret key, encrypts a message m using this framework. In doing so, she effectively commits to the value of m within the set of integers $\mathbb{Z}$ rather than within the ring $\mathbb{Z}_N$, which opens the door for directly proving the range of m without the need for auxiliary integer commitment.

Direct Range Proof. Now we are left with the task of designing the corresponding range proof for our commitment, as well as demonstrating its correctness and security. We follow the idea of range proof for the integer commitment scheme based on QR_N [12], which is developed from the CFT technique [8]. We emphasize that this task is far from straightforward, given that we must navigate a different mathematical framework within $\mathbb{Z}^*_{N^2}$. Additionally, we

have to carefully deal with the $(1+N)$ component, especially when the commitment is maliciously generated. We successfully reduce the security of range proof to the Paillier root assumption, which holds under the normal strong RSA assumption. Refer to Section 4.1 for details.

Generalized Commitment. Our integer commitment scheme and corresponding range proof can be readily extended to those for vector of integers. The generalization is beneficial for scenarios where one wants to commit a series of integers at a time. Suppose that the length of the vectors is l. Unlike the prior integer commitment where a single y is used, here we need l similar y-elements. Specifically, the i-th y-element is computed as $y_i = h_i(1+N)^{\beta_i} \bmod N^2$, where h_i is randomly chosen from $2\mathsf{NR}_{N^2}$ and β_i is any value from $\mathbb{Z}_N$ (it can be either known or not to the committer). Consequentially, the commitment of $\mathbf{m} = (m_1, \cdots, m_l) \in \mathbb{Z}^l$ is defined as

$$c = g^r \prod_{i=1}^{l} y_i^{m_i} \bmod N^2, \tag{4}$$

where $r \in \mathbb{Z}_N$ is the randomness. Besides, the range proof technique used for integer commitments can also be easily generalized accordingly.

Return back to our goal, if a Paillier affine operation can be interpreted as committing over $\mathbb{Z}\times\mathbb{Z}$, then we will have a chance to apply direct range proof to it. An insightful observation is that an affine operation can indeed be directly interpreted as our generalized commitment operation, specifically with a vector length of $l = 2$. Assuming Alice gets a ciphertext C from a peer, and computes an affine operation denoted by $C^a[h(1+N)]^A g^r \bmod N^2$ with private input (a, A). Without knowing the secret key, she actually commits to (a, A) in $\mathbb{Z}\times\mathbb{Z}$, with $y_1 = C$ and $y_2 = y = h(1+N) \bmod N^2$.

1.5 Discussion

Other Applications. Our range proofs and schemes are also applicable to other scenarios, e.g., the multiparty computations SPDZ [15, 16], SPD$\mathbb{Z}_{2^k}$ [11], and the e-voting systems [14]. Our Paillier-based MtA could be also used to build more efficient three-party TLS handshake [42, Section 4.1] by integrating it into their protocol.

Limitations. We would like to point out some limitations of our techniques. Firstly, in our range proofs shown in Section 4, the prover must not have control over the public parameters, and specifically, knowledge of the factorization of the employed modulus N, or the related discrete logarithms, is not allowed. When the situation arises where the prover possesses such secrets, we still need an auxiliary integer commitment with public parameters prepared by the verifier, which will increase the complexity.

Secondly, a shared shortcoming of our proofs and those of Lindell and Nof [33] is their dependence on the strong RSA assumption. However, it is possible to adapt the technique proposed by Couteau *et al.* [10] to our context, making our proofs based on the RSA assumption, which is generally considered to be more standard. We leave it as future work.

Finally, within the scope of the range proof for Paillier plaintext, it may be challenging to instantiate our method into a non-interactive version in the standard model by correlation-intractable hash functions such as [5], compared to the method proposed by Devevey *et al.* [18]. This is mainly due to the fact that the soundness of our range proof is based on certain cryptographic assumptions, instead of achieving statistical soundness. The range proof from Lindell and Nof [33] also has such problem.

1.6 Related Work

Other Range Proof Methods for Paillier. Except for the methods by [18, 33] discussed before, one may use other tools to conduct range proofs for Paillier, such as zk-SNARK proof systems. These tools can produce efficient range proofs for commitments based on prime order groups, as shown in [4, 29]. However, they seem not to be suitable for Paillier or the applications we focus on. For example, it is costly to use zk-SNARKs based on arithmetic circuits over prime fields, since they require large circuits to represent Paillier encryption and related relations, such as the Paillier knowledge-of-plaintext circuit, which has 80 million gates for N of length 2048 [31]. Very recently, [27] proposed a range proof method for Paillier plaintext which is a zk-SNARK using arithmetic circuits directly working in $\mathbb{Z}_{N^2}$. However, this method was designed for the batch proof, where a prover wants to prove that multiple Paillier plaintexts lie in the same range. When applied to a single range proof task, it would incur heavy communication and computation overheads, due to the parallel repetitions for reducing soundness error. Similarly, the batch proof proposed by [39] also incurs significant costs when applied to scenarios involving only a small number of plaintexts.

Threshold ECDSA. The main task of threshold ECDSA is to design a secure multi-party computation of non-linear operations specified in ECDSA signature, as outlined in [32]. Existing solutions [7, 20, 21, 32, 33, 40] leverage the Multiplicative-to-Additive (MtA) functionality to deal with the problem. Current MtAs are derived either from Oblivious Transfer (OT) [20, 21] or homomorphic encryption schemes like Paillier encryption [32, 33], Castagnos-Laguillaumie (CL) encryption [7], and Joye-Libert (JL) encryption [40]. Among these, Paillier-based MtAs are the most popular and favored by industry due to their optimal balance between computational and communication costs. In contrast, OT-based MtAs demand substantial bandwidth, and CL-based MtAs incur high computational costs. Concurrently, a key limitation of JL-based MtAs is that the proof for the correctness of the JL modulus remains under-explored, necessitating the assumption of a trusted dealer.

Naor-Yung Paradigm. The Naor-Yung paradigm refers to the generic transformation method introduced by [36], which can transform a CPA-secure public key encryption scheme to a CCA2-secure one. In this method, the receiver will have two key pairs, and the sender is required to encrypt the message with both public keys, then compute a non-interactive zero-knowledge proof with simulation soundness for the equality of respective plaintexts. Based on the Paillier encryption, Fouque and Pointcheval [24] propose an instantiation of Naor-Yung paradigm (with additional threshold property). However, Devevey *et al.* [17] point out that the distinct RSA moduli in the two pairs of keys can be leveraged by a malicious adversary to prove false statements, thus invalidating the claims of security in [24]. Fortunately, this problem can be fixed by including an additional proof for that the underlying plaintext is smaller than each RSA modulus, suggested also by [17].

2 PRELIMINARIES

2.1 Notations and Mathematics

In this paper, abbreviation PPT stands for Probabilistic Polynomial Time. We use κ to denote the security parameter. For a finite set X, $x \leftarrow X$ represents uniformly sampling an element x from X. We leverage $|x|$ to denote the bit length of some element x, while $|X|$ to represent the number of contained elements for some finite set X. For $a, b \in \mathbb{Z}$ and $a \leq b$, we use $[a, b]$ to denote the set of $\{a, a+1, \cdots, b-1, b\}$.

Let $N = pq$ be RSA modulus. In this paper, we will only consider safe primes p and q, that is, $p = 2p' + 1$ and $q = 2q' + 1$ where p' and q' are also primes. It is easy to see that p and q are Blum primes, namely, $p = q = 3 \bmod 4$, when p' and q' are odd primes. Besides, $\varphi(\cdot)$ and $\lambda(\cdot)$ refer to the Euler and Carmichael function, respectively. By convention, we need that $|p| = |q|$ and p, q, p', q' are distinct primes, and we have $\gcd(N, \varphi(N)) = 1$.

We define the following three multiplicative groups and give some facts which are essential throughout the entire paper:

$$\mathsf{2NR}_{N^2} = \left\{ x \in \mathbb{Z}^*_{N^2} : \exists a \in \mathbb{Z}^*_{N^2}, x = a^{2N} \bmod N^2 \right\}.$$

$$\mathsf{QR}_{N^2} = \left\{ x \in \mathbb{Z}^*_{N^2} : \exists a \in \mathbb{Z}^*_{N^2}, x = a^{2} \bmod N^2 \right\}.$$

$$\mathsf{QR}_{N} = \left\{ x \in \mathbb{Z}^*_{N} : \exists a \in \mathbb{Z}^*_{N}, x = a^{2} \bmod N \right\}.$$

Fact 1. *The function $f(x) = x^N \bmod N^2$ is a 1-1 mapping from QR_N to $\mathsf{2NR}_{N^2}$.*

Fact 2. *$\mathsf{2NR}_{N^2}$ is a cyclic group of order $p'q'$, and for $g \leftarrow \mathsf{2NR}_{N^2}$, g is a generator with overwhelming probability $\frac{\varphi(p'q')}{p'q'} = \left(1 - \frac{1}{p'}\right)\left(1 - \frac{1}{q'}\right)$. Besides, we can sample a uniform element g from $\mathsf{2NR}_{N^2}$ by picking $a \leftarrow \mathbb{Z}^*_{N^2}$ and setting $g = a^{2N} \bmod N^2$.*

Fact 3. *QR_{N^2} is a cyclic group of order $p'q'N$, and for $g \leftarrow \mathsf{QR}_{N^2}$, g is a generator with overwhelming probability $\frac{\varphi(p'q'N)}{p'q'N} = \left(1 - \frac{1}{p'}\right)\left(1 - \frac{1}{q'}\right)\left(1 - \frac{1}{p}\right)\left(1 - \frac{1}{q}\right)$. Additionally, we can sample a uniform element g from QR_{N^2} by selecting $a \leftarrow \mathbb{Z}^*_{N^2}$ and setting $g = a^2 \bmod N^2$.*

Fact 4. *$1 + N \in \mathsf{QR}_{N^2}$, $1 + N \notin \mathsf{2NR}_{N^2}$, and the order of $1 + N$ modulo N^2 is N.*

Fact 5. *For $r \leftarrow \mathbb{Z}_N$, if g is a generator of $\mathsf{2NR}_{N^2}$, then $g^r \bmod N^2$ is statistically close to a uniform variable over $\mathsf{2NR}_{N^2}$.*

Fact 6. *Computing a non-zero multiple of $p'q'$ is equivalent to factoring N.*

Fact 7. *Finding a non-trivial square root of 1 modulo N is equivalent to factoring N.*

Fact 8. *If $a^2 = 1 \bmod N^2, a \in \mathsf{QR}_{N^2}$ (resp. $a^2 = 1 \bmod N, a \in \mathsf{QR}_N$), then $a = 1 \bmod N$.*

Now we give a concise illustration why the above facts are correct. Fact 1 comes from that each N-th residue modulo N^2 has exactly one root in $\mathbb{Z}^*_N$ [38, Section 2]. Fact 2 can be induced from Fact 1 and that QR_N is a cyclic group of order $p'q'$. Fact 3 is directly from [3, Section 2.1]. For Fact 4, that $1 + N \in \mathsf{QR}_{N^2}$ and that the order of $1 + N$ modulo N^2 is N also come directly from [3, Section 2.1]. Besides, $1 + N \notin \mathsf{2NR}_{N^2}$ since the order of $1 + N$ modulo N^2 does not divide the order of $\mathsf{2NR}_{N^2}$. Fact 5 is correct since $N \bmod p'q'$ is negligible in $p'q'$ (note that $p'q'$ is the order of $\mathsf{2NR}_{N^2}$). Fact 6 and 7 can be derived from the proof for [10, Fact 4 of Proposition 1]. Finally, we explain Fact 8. From $a^2 = 1 \bmod N^2$ we immediately get $a^2 = 1 \bmod N$. In $\mathbb{Z}^*_N$, since any square has exactly one incongruent square root also being square (see [10, Fact 2 of Proposition 1]), 1 is a square, and $1^2 = 1 \bmod N$, we have that $a = 1 \bmod N$.

2.2 Assumptions

Definition 2.1 (DCR Assumption). The Decisional Composite Residuosity (DCR) assumption assumes that, given $N = pq$ for safe primes p, q, it is hard to distinguish a uniformly random N-th residue modulo N^2 from a uniformly random element in $\mathbb{Z}^*_{N^2}$. Specifically, it is hard to distinguish x and y, where $a \leftarrow \mathbb{Z}^*_N$, $x = a^N \bmod N^2$, and $y \leftarrow \mathbb{Z}^*_{N^2}$.

Definition 2.2 (DL Assumption). In this paper, we consider the discrete logarithm (DL) assumption over $\mathsf{2NR}_{N^2}$ (we may omit $\mathsf{2NR}_{N^2}$ from time to time when there is no ambiguity). Concretely, given $N = pq$ for safe primes p, q and random $g, h \leftarrow \mathsf{2NR}_{N^2}$, it is hard to find an integer α such that $g^\alpha = h \bmod N^2$ for any PPT algorithm. Note that the DL problem over $\mathsf{2NR}_{N^2}$ is not easier than that over QR_N and please refer to Appendix A for the detailed proof.

Definition 2.3 (Strong RSA Assumption). Given $N = pq$ for safe primes p, q and a random $T \leftarrow \mathbb{Z}^*_N$, it is hard to find an e-th root a modulo N, namely, $a^e = T \bmod N$, for any PPT algorithm with an exponent $e > 1$ of its choice.

Definition 2.4 (Paillier Root Assumption). Given $N = pq$ for safe primes p, q and a random $T \leftarrow \mathsf{2NR}_{N^2}$, it is hard to find an e-th root a modulo N^2, namely, $a^e = T \bmod N^2$, for any PPT algorithm with an exponent $e > 1$ and $\gcd(e, N) = 1$ of its choice.

We show that the Paillier root assumption holds under the strong RSA assumption. Please refer to Appendix B for the detailed proof.

2.3 Commitment

Definition 2.5 (Commitment). A commitment scheme is defined by a tuple of algorithms $(\mathsf{Setup}, \mathsf{Commit}, \mathsf{Verify})$.

- $\mathsf{Setup}(1^\kappa)$. The setup algorithm takes as input a security parameter $\kappa \in \mathbb{N}$ and returns the public parameter pp (it implicitly defines the message space $\mathcal{M}$).
- $\mathsf{Commit}(pp, m)$. The committing algorithm takes as input the public parameter pp and a message $m \in \mathcal{M}$, then outputs a commitment c along with an opening o.
- $\mathsf{Verify}(pp, c, o)$. The verification algorithm takes as input the public parameter pp, a commitment c, and an opening o, then outputs 1 if o is a valid opening of c, or 0 otherwise.

The correctness of a commitment scheme requires that for any $pp \leftarrow \mathsf{Setup}(1^\kappa)$, for any $m \in \mathcal{M}$, for any $(c, o) \leftarrow \mathsf{Commit}(pp, m)$, it holds that $\mathsf{Verify}(pp, c, o) = 1$.

Setup: On receiving (setup) from P_1 and P_2

- Store and send (setup-complete) to P_1 and P_2.

Trans: On receiving (input, sid, $a \in \mathbb{Z}_q$) from P_1, (input, sid, $b \in \mathbb{Z}_q$) from P_2, where sid has not been used, if (setup-complete) exists:

- Sample $A \leftarrow \mathbb{Z}_q$ and compute $B = ab - A \bmod q$.
- Send (output-1, sid, A) to P_1.
- Send (output-2, sid, B) to P_2.

Figure 1: Multiplicative-to-Additive Functionality

Typical security properties are statistical hiding and computational binding or computational hiding and statistical binding. In this paper, we will consider computational hiding and computational binding, which are defined as follows.

- Hiding. For any $m_1, m_2 \in \mathcal{M}$, their commitments are computationally indistinguishable.
- Binding. No efficient adversary can open a commitment c to two valid and different openings o and o', except with negligible probability.

2.4 Sigma Protocol

Σ-protocol is a special zero-knowledge proof defined as follows.

Definition 2.6 (Sigma Protocol). Let $\mathcal{R} \subseteq \mathcal{X} \times \mathcal{W}$ be some effective binary relation, where $\mathcal{X}$, $\mathcal{W}$ and $\mathcal{R}$ are efficiently recognizable finite sets. Elements in $\mathcal{X}$ are named as statements. If $(x, w) \in \mathcal{R}$, w is called a witness for x. A Σ-protocol for $\mathcal{R}$ is a pair $(\mathcal{P}, \mathcal{V})$, standing for prover and verifier respectively. The form of a Σ-protocol should be 3-move:

- $\mathcal{P}$ with input $(x, w) \in \mathcal{R}$ computes a message d, called the commitment, and sends d to $\mathcal{V}$.
- $\mathcal{V}$ picks a challenge e at random from a finite challenge space $\mathcal{E}$, and sends e to $\mathcal{P}$.
- $\mathcal{P}$ computes a response z, and sends z to $\mathcal{V}$.

Upon receiving the response from $\mathcal{P}$, $\mathcal{V}$ either accepts or rejects the proof. Besides, the tuple (d, e, z) is called a conversation.

The completeness requires that for any $(x, w) \in \mathcal{R}$, if $\mathcal{P}$ and $\mathcal{V}$ runs the protocol honestly, $\mathcal{V}$ will always accept the proof.

The security of a Σ-protocol is defined as below.

- Honest-Verifier Zero-Knowledge (HVZK). There is a PPT simulator that on input $(x, e) \in \mathcal{X} \times \mathcal{E}$ can output (d, z) such that (d, e, z) is an accepting conversation for x. Furthermore, for any true statement, the simulated conversation is indistinguishable from a real one (with the same challenge).
- Special soundness (Proof of Knowledge, PoK). There is a PPT knowledge extractor that, for any statement x, on input two accepting conversation (d, e, z) and (d, e', z') with $e \neq e'$, can output a witness w' s.t. $(x, w') \in \mathcal{R}$.

2.5 Multiplicative-to-Additive Functionality

The multiplicative-to-additive functionality $\mathcal{F}_{\mathsf{MtA}}$ runs between two parties P_1 and P_2 as illustrated in Figure 1. It is parameterized by some integer q, which stands for the prime order of the elliptic curve utilized in ECDSA in this paper. The Setup phase only needs to be conducted once, then P_1 and P_2 can run Trans many times. Each call of Trans, P_1 and P_2 will obtain A and B from their private input a and b, respectively, satisfying $A + B = ab \bmod q$.

2.6 Public Key Encryption

Definition 2.7 (PKE). A public key encryption (PKE) scheme is defined by a tuple of algorithms (KGen, Enc, Dec).

- $\mathsf{KGen}(1^\kappa)$. The key generation algorithm takes as input a security parameter $\kappa \in \mathbb{N}$ and returns a pair of public key and secret key (pk, sk).
- $\mathsf{Enc}(pk, m)$. The encryption algorithm takes as input a public key pk and a message $m \in \mathcal{M}$ where $\mathcal{M}$ denotes the message space, then returns a ciphertext C of m.
- $\mathsf{Dec}(sk, C)$. The decryption algorithm takes as input a secret key sk and a ciphertext C, then returns the corresponding plaintext m.

The correctness of a PKE scheme requires that for any $(pk, sk) \leftarrow \mathsf{KGen}(1^\kappa)$, for any $m \in \mathcal{M}$, for any $C \leftarrow \mathsf{Enc}(pk, m)$, it holds that $\mathsf{Dec}(sk, C) = m$.

The security of a PKE scheme can be described by a game played between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$. This game is presented as follows:

- **Setup.** For a security parameter κ, $\mathcal{C}$ runs $\mathsf{KGen}(1^\kappa)$ to obtain a key pair (pk, sk) and sends pk to $\mathcal{A}$.
- **Phase 1.** In this phase, $\mathcal{A}$ is allowed to make decryption queries adaptively. For a query on some ciphertext C, $\mathcal{C}$ invokes $m \leftarrow \mathsf{Dec}(sk, C)$ and sends m to $\mathcal{A}$.
- **Challenge.** $\mathcal{A}$ outputs two distinct messages $m_0, m_1 \in \mathcal{M}$. $\mathcal{C}$ picks a bit $b \leftarrow \{0, 1\}$, calls $C^* \leftarrow \mathsf{Enc}(pk, m_b)$, and then sends C^* to $\mathcal{A}$.
- **Phase 2.** This phase is the same as **Phase 1**, except that decryption query on C^* is not allowed.
- **Guess.** $\mathcal{A}$ outputs a guess b' of b and wins if $b' = b$.

Then the advantage of $\mathcal{A}$ in breaking the indistinguishability under chosen-ciphertext attacks (IND-CCA2) is defined as

$$\mathrm{Adv}^{\text{IND-CCA2}}_{\mathcal{A}}(\kappa) = \left| \Pr[b = b'] - \frac{1}{2} \right|,$$

and the scheme is said to be IND-CCA2 secure if this advantage is negligible in κ for any efficient adversary $\mathcal{A}$.

The indistinguishability under chosen-plaintext attacks (IND-CPA) of a PKE scheme can be defined in a similar manner. In this case, the game between $\mathcal{C}$ and $\mathcal{A}$ is the same as before, but with the restriction that $\mathcal{A}$ is not allowed to make any decryption queries, i.e., there is no more **Phase 1** and **Phase 2**. Besides, the advantage of $\mathcal{A}$, which should be negligible for any efficient adversary, is defined as

$$\mathrm{Adv}^{\text{IND-CPA}}_{\mathcal{A}}(\kappa) = \left| \Pr[b = b'] - \frac{1}{2} \right|.$$

3 PAILLIER CRYPTOSYSTEM REVISITED

In this section, we introduce our commitment (named as Paillier commitment) and modified Paillier encryption. The Paillier commitment allows a committer to commit to some value over (vectors of) integers, while by our modified Paillier encryption, an encryption operation or affine operation will correspond to a committing operation in the Paillier commitment.

3.1 Paillier Commitment

Let $l \in \mathbb{N}$ be a predetermined value which defines the length of committed vectors. Our Paillier commitment scheme PaillCom is shown as follows.

- Setup(1^κ). Randomly generate distinct safe primes $p = 2p'+1$ and $q = 2q'+1$. Compute the corresponding RSA modulus $N = pq$. Pick $a \leftarrow \mathbb{Z}^*_{N^2}$ and compute $g = a^{2N} \bmod N^2$. Sample $\alpha_1, \cdots, \alpha_l \leftarrow \mathbb{Z}_N$, and then compute $y_i = g^{\alpha_i}(1+N)^{\beta_i} \bmod N^2$ for $i \in [1, l]$ (where $\beta_i \in \mathbb{Z}_N$ can be either known or not to the committer, depending on the application scenarios). Set pp as $(N, g, y_1, \cdots, y_l)$.
- Commit($pp, \mathbf{m}$). For $pp = (N, g, y_1, \cdots, y_l)$ and a vector $\mathbf{m} = (m_1, \cdots, m_l) \in \mathbb{Z}^l$, choose $r \leftarrow \mathbb{Z}_N$ and then compute
$$c = g^r \prod_{i=1}^{l} y_i^{m_i} \bmod N^2.$$
Finally, return c as the commitment and $(\mathbf{m}, r)$ as the corresponding opening.
- Verify($pp, c, \mathbf{m}, r$). For $pp = (N, g, y_1, \cdots, y_l)$ and a vector $\mathbf{m} = (m_1, \cdots, m_l)$, output 1 when
$$c = \pm g^r \prod_{i=1}^{l} y_i^{m_i} \bmod N^2,$$
or 0 otherwise.

Note that if and only if two openings $(\mathbf{m}, r)$ and $(\mathbf{m}', r')$ satisfy $\mathbf{m} \neq \mathbf{m}'$ (over $\mathbb{Z}^l$), we call them different openings. The Verify algorithm accepts the negative sign, since the knowledge-extractability discussed later cannot rule out the possibility of sign change.

Theorem 3.1. *If the DCR, factoring and DL assumptions hold,* PaillCom *is a commitment scheme over ordered integers with computational hiding and binding.*

The detailed proof of Theorem 3.1 is postponed to Appendix G and we give a sketch here. Firstly, the correctness is obvious. Secondly, $g^r \bmod N^2$ is statistically close to a uniformly random variable from $2\mathsf{NR}_{N^2}$, which is computationally indistinguishable from a uniformly random variable from QR_{N^2} under the DCR assumption (to see this, we only need to square the target element in the DCR problem instance). Since $y_i \in \mathsf{QR}_{N^2}$ for $i \in [1, l]$, the commitment c is computationally indistinguishable from a uniformly random variable from QR_{N^2}, regardless of the underlying committed message, which ensures the hiding property. Thirdly, for the binding property, in Appendix G we show that given two different and valid openings, we can either attack the DL assumption or find a non-zero multiple of $p'q'$ which will lead to the factorization of N according to Fact 6.

3.2 Modified Paillier Encryption

Our modified Paillier encryption scheme is denoted as MPaill. In our scheme, $\widetilde{\mathsf{Enc}}$ is an alternative way to compute ciphertexts. Looking ahead, when an encryptor would not conduct direct range proof for the plaintext, he/she can select to use $\widetilde{\mathsf{Enc}}$, so that the computation of $(1+N)^m \bmod N^2$ can be replaced by $1+mN \bmod N^2$ in order to reduce the computational costs. The concrete description of MPaill is as follows.

- KGen(1^κ). Randomly generate safe primes $p = 2p'+1$ and $q = 2q'+1$. Compute $\lambda = \lambda(N) = 2p'q'$, and its inverse modulo N, denoted by λ^{-1}. Choose $a \leftarrow \mathbb{Z}^*_{N^2}$ and $\alpha \leftarrow \mathbb{Z}_N$, then compute $g = a^{2N} \bmod N^2$ and set $y = g^\alpha(1+N) \bmod N^2$. Return the public and secret key pair
$$pk = (N, g, y), sk = (N, \lambda, \lambda^{-1}).$$
- Enc(pk, m). For a public key $pk = (N, g, y)$ and a message $m \in \mathbb{Z}_N$, sample $r \leftarrow \mathbb{Z}_N$ and compute the ciphertext as
$$C = y^m g^r \bmod N^2.$$
- $\widetilde{\mathsf{Enc}}(pk, m)$. For a public key $pk = (N, g, y)$ and a message $m \in \mathbb{Z}_N$, sample $r \leftarrow \mathbb{Z}_N$ and compute the ciphertext as
$$C = (1+N)^m g^r \bmod N^2.$$
- Dec(sk, C). For a secret key $sk = (N, \lambda, \lambda^{-1})$ and a ciphertext C, compute the plaintext as
$$m = \frac{C^\lambda \bmod N^2 - 1}{N} \cdot \lambda^{-1} \bmod N.$$

Theorem 3.2. *The* MPaill *scheme is correct, and satisfies IND-CPA security under the DCR assumption.*

Please refer to Appendix H for the detailed proof of Theorem 3.2 and we only give a sketch here. Firstly, for correctness, our scheme follows the decryption idea of the original Paillier encryption. In the original scheme, by adding the exponent of $\lambda(N)$, the randomness part $r^N \bmod N^2$ can be eliminated since the order of the group formed by all N-th residues modulo N^2 is $\lambda(N)$. Besides, the information of m can be fully preserved because the order of $1+N$ modulo N^2 is N, which is co-prime to the exponent $\lambda(N)$. In our scheme, the exponent $\lambda(N)$ can also be used to eliminate the part $g^{\alpha m+r} \bmod N^2 \in 2\mathsf{NR}_{N^2}$ or $g^r \bmod N^2 \in 2\mathsf{NR}_{N^2}$, because $2\mathsf{NR}_{N^2}$ is a subgroup of the N-th residue group. Secondly, the proof of IND-CPA security is similar to that of the hiding property of PaillCom (see the proof of Theorem 3.1). Apart from the full proof of this theorem, we also discuss various variations of the encryption algorithm in Appendix H.

Encryption Operation. In MPaill scheme, the public key $pk = (N, g, y)$ can be treated as the public parameter pp of PaillCom with length $l = 1$. Specifically, we have $y_1 = y$, $\alpha_1 = \alpha$ and $\beta_1 = 1$. When someone, say Alice, is not the owner of the corresponding secret key and encrypts a message m under pk to get a ciphertext C via Enc, she actually acts as a committer who commits to m over $\mathbb{Z}$ according to PaillCom.

Affine Operation. Assume a ciphertext C_b (generated from either Enc or $\widetilde{\mathsf{Enc}}$) of underlying plaintext b under public key $pk = (N, g, y)$, the affine operation can be denoted as $(C_b)^a \cdot \mathsf{Enc}_{pk}(A) = (C_b)^a y^A g^r \bmod N^2$ with private input a and A (for some randomness r), which will result in an encryption of $ab+A$. If this operation is performed by someone, say Alice, who is not the owner of the corresponding secret key and not the encryptor of C_b, then she actually acts as a committer who commits to (a, A) over $\mathbb{Z} \times \mathbb{Z}$ according to PaillCom. In this situation, the length l is equal to 2 and the public parameter pp is set as $(N, g, y_1, y_2) = (N, g, C_b, y)$.

4 RANGE PROOFS

From the discussion of previous section, we have that in MPaill, an encryption operation or affine operation can be related to a committing operation under PaillCom. Thus, if there is a range proof method for PaillCom, it can be easily applied to provide range proofs for Paillier plaintext and affine operation. In other words, we are left with the task of designing a range proof for PaillCom, which is the goal of this section.

4.1 Range Proof for Paillier Commitment

Suppose that $pp = (N, g, y_1, \cdots, y_l)$ is the public parameter of PaillCom and a prover would like to prove the knowledge of an opening $(\mathbf{m} = (m_1, \cdots, m_l), r)$ of commitment c with $m_i \in [0, B_i]$ for each $i \in [1, l]$. Formally, we design a Σ-protocol $\mathsf{ZK}_{\mathsf{PaillCom}}$ between a prover $\mathcal{P}$ and verifier $\mathcal{V}$ for the relation

$$\mathcal{R}_{\mathsf{PaillCom}} = \{(c; \mathbf{m}, r) \ : c = \pm g^r \prod_{i=1}^{l} y_i^{m_i} \bmod N^2, \\ \forall i \in [1, l], m_i \in [0, B_i]\}.$$

We use s and t to represent the statistical and soundness parameters respectively, and the protocol works as follows.

- $\mathcal{P}$ chooses $u_i \leftarrow [0, 2^{s+t}B_i]$ for every $i \in [1, l]$, and picks $v \leftarrow [0, 2^{s+t}N]$. Next, $\mathcal{P}$ computes $d = g^v \prod_{i=1}^{l} y_i^{u_i} \bmod N^2$ and sends it to $\mathcal{V}$.
- $\mathcal{V}$ selects $e \leftarrow [0, 2^t]$ and sends it to $\mathcal{P}$.
- $\mathcal{P}$ computes $z_i = em_i + u_i$ (over $\mathbb{Z}$) for every $i \in [1, l]$, and $z_r = er + v$ (over $\mathbb{Z}$), then sends them to $\mathcal{V}$.
- $\mathcal{V}$ accepts the proof if both of the conditions hold:
  (1) $g^{z_r} \prod_{i=1}^{l} y_i^{z_i} = c^e d \bmod N^2$.
  (2) $z_i \in [0, 2^{s+t}B_i]$ for every $i \in [1, l]$.

Theorem 4.1. *$\mathsf{ZK}_{\mathsf{PaillCom}}$ is a Σ-protocol for the relation $\mathcal{R}_{\mathsf{PaillCom}}$ which provides completeness, HVZK and PoK under the factoring assumption and Paillier root assumption (it holds under the strong RSA assumption).*

The proof of Theorem 4.1 is the most challenging part of this paper, and we have deferred it to Section 4.2.

Remark. Here, we reiterate that the range proof for PaillCom can be applied to provide range proofs for both Paillier plaintext and Paillier affine operation, due to the discussion in Section 3, regarding the relationship between MPaill and PaillCom. Furthermore, since the range proof for PaillCom does not require any additional auxiliary commitments, the resulting range proofs for Paillier plaintext and affine operation are also direct range proofs.

4.2 Proof of Theorem 4.1

Completeness. Firstly, we have $d = g^v \prod_{i=1}^{l} y_i^{u_i} \bmod N^2$, $z_i = em_i + u_i$ for each $i \in [1, l]$, and $z_r = er + v$, from the protocol. Thus, $g^{z_r} \prod_{i=1}^{l} y_i^{z_i} = g^{er+v} \prod_{i=1}^{l} y_i^{em_i+u_i} = \left(g^r \prod_{i=1}^{l} y_i^{m_i}\right)^e g^v \prod_{i=1}^{l} y_i^{u_i} = c^e d \bmod N^2$. Next, a legitimate $\mathcal{P}$ with message $\mathbf{m} = (m_1, \cdots, m_l)$ where $m_i \in [0, B_i]$ for every $i \in [1, l]$ fails to convince $\mathcal{V}$ when there is some index j such that $z_j = em_j + u_j \geq 2^{s+t}B_j + 1$, or equivalently, $u_j \geq 2^{s+t}B_j - em_j + 1$. For any $i \in [1, l]$, the probability that $u_i \geq 2^{s+t}B_i - em_i + 1$ is

$$\frac{2^{s+t}B_i - (2^{s+t}B_i - em_i + 1) + 1}{2^{s+t}B_i + 1} = \frac{em_i}{2^{s+t}B_i + 1} \leq \frac{2^t B_i}{2^{s+t}B_i} = 2^{-s}.$$

Then the probability that there is such index j is at most $l \cdot 2^{-s}$, which is negligible.

HVZK. We construct a simulator $\mathcal{S}$ that works as follows on input c and e. $\mathcal{S}$ picks $z_i \leftarrow [0, 2^{s+t}B_i]$ for each $i \in [1, l]$ and $z_r \leftarrow [0, 2^{s+t}N]$, then sets $d = g^{z_r} \prod_{i=1}^{l} y_i^{z_i} c^{-e} \bmod N^2$. It is clear that $\mathcal{S}$ always outputs an accepting conversation. Next, we argue that the simulated conversation has the right distribution when e is uniformly distributed over $[0, 2^t]$. This comes from the fact that the distributions of $\{z_i\}_{i\in[1,l]}$ and z_r are statistically indistinguishable from the real distributions, and d is uniquely determined by $g^{z_r} \prod_{i=1}^{l} y_i^{z_i} = c^e d \bmod N^2$ when $\{z_i\}_{i\in[1,l]}$, z_r and e are fixed.

Proof Sketch of PoK. We will first prove the PoK in the special case where $l = 1$, then extend this to the PoK for any polynomial-bounded value of l. Additionally, here we provide a proof sketch for the PoK when $l = 1$.

The general idea is to use two accepting conversations denoted as $(d, e, (z_1, z_r))$ and $(d, e', (z_1', z_r'))$ with $e \neq e'$ (obtained from standard rewind technique), to extract a witness (within proper interval) or solve some hard problem (factoring or strong RSA). Let $\Delta e = e - e'$ (without loss of generality, assume $\Delta e > 0$), $\Delta z_1 = z_1 - z_1'$ and $\Delta z_r = z_r - z_r'$, the starting point is trying to extract a witness by moving Δe to the left-side of the following equation

$$g^{\Delta z_r} y_1^{\Delta z_1} = c^{\Delta e} \bmod N^2, \tag{5}$$

which is obtained from the definition of verification. The approach to accomplish this task is to raise both sides of the equation to the power of the inverse of Δe modulo $\varphi(N^2)$, noting that we are working over $\mathbb{Z}^*_{N^2}$. However, there are two main obstacles: first, Δe is not necessarily co-prime to $\varphi(N^2) = 4p'q'pq$, meaning its inverse might not exist; second, even if the inverse exists, we do not know its value (since we do not know $\varphi(N^2)$). These obstacles lead to the following case analysis.

Case 1.1: $\Delta e \mid \Delta z_r, \Delta e \mid \Delta z_1, \Delta e$ is odd. In this case, we can easily extract a witness by raising both sides of Equation (5) to the power of the inverse of Δe modulo $\varphi(N^2)$. That Δe is odd ensures the existence of inverse, while the condition of $\Delta e \mid \Delta z_r \wedge \Delta e \mid \Delta z_1$ allows us to move Δe without knowing its inverse: let $\widetilde{\Delta e}$ be its inverse, we can compute an integer division of $\Delta z_r / \Delta e$ to replace $\Delta z_r \cdot \widetilde{\Delta e}$, since $\Delta z_r / \Delta e = \Delta z_r \cdot \widetilde{\Delta e} \bmod \varphi(N^2)$. We briefly explain why this is true. Since $\Delta e \mid \Delta z_r$, we have $\Delta z_r = k \cdot \Delta e$ for some integer k, then we have $\Delta z_r \cdot \widetilde{\Delta e} = k \cdot \Delta e \widetilde{\Delta e} \bmod \varphi(N^2) = k \bmod \varphi(N^2) = (\Delta z_r / \Delta e \text{ in } \mathbb{Z}) \bmod \varphi(N^2)$. Likewise, we can compute $\Delta z_1 / \Delta e$ instead of $\Delta z_1 \cdot \widetilde{\Delta e}$.

Case 1.2: $\Delta e \mid \Delta z_r, \Delta e \mid \Delta z_1, \Delta e$ is even. In this case, Δe does not have an inverse modulo $\varphi(N^2)$. Instead, we consider the odd factor $\Delta e'$ of Δe ($\Delta e = 2^\rho \Delta e'$ for $\rho \geq 1$), which is invertible and divides both Δz_r and Δz_1. By this we will obtain $Z^{2^\rho} = 1 \bmod N^2$ for some computable Z, such that if $Z = \pm 1 \bmod N^2$, we successfully extract a witness, and if $Z \neq \pm 1 \bmod N$, Z is a non-trivial square root of 1 modulo N, which leads to the factorization of N from Fact 7. The remaining and tricky situation is when $Z \neq \pm 1 \bmod N^2$ and $Z = \pm 1 \bmod N$. This neither provides a witness nor leads to

factoring N. In the complete proof, we rule out the possibility of this situation, by utilizing the condition $Z^{2^\rho} = 1 \bmod N^2$.

Case 2: $\Delta e \nmid \Delta z_r \vee \Delta e \nmid \Delta z_1$. We show by contradiction that this case happens only with negligible probability, or otherwise, we can construct a simulator $\mathcal{B}$ to solve the strong RSA problem with non-negligible probability. Since the strong RSA problem is defined in $\mathbb{Z}_N^*$ while our scheme works over $\mathbb{Z}_{N^2}^*$, we define a similar problem that is compatible with our scheme and not easier. Roughly speaking, Paillier root problem asks for an x-th root a modulo N^2, given a random T from $2\mathsf{NR}_{N^2}$, i.e., $T = a^x \bmod N^2$. The requirements are $x > 1$ (non-trivial solution) and $\gcd(x, N) = 1$. Note that the second requirement is unique in Paillier root problem, for the reduction of the strong RSA problem to the Paillier root problem (in Appendix B we prove that the Paillier root problem is not easier).

In the proof, we simulates the parameter g as the problem instance T. By this we will obtain $T^\eta = g^\eta = P^{\Delta e} \bmod N^2$ where P is some computable value for $\mathcal{B}$ and $\eta \mid \Delta e$. To get a solution to the Paillier root problem, we only need to move η to the right-side of the equation (a case analysis is needed based on the parity of η, similar to case 1.1 and 1.2).

Additionally, there is still an obstacle: we must ensure that our obtained solution is non-trivial, which means that the exponent needs to be greater than 1. Therefore, in the full proof, we need to conduct a case analysis based on whether η equals Δe. Specifically, we demonstrate that the situation where η equals Δe occurs with a probability noticeably lower than 1.

PoK with $l = 1$. Assume that we get two accepting conversations denoted as $(d, e, (z_1, z_r))$ and $(d, e', (z_1', z_r'))$ with $e \neq e'$. From the definition of verification, we have

$$g^{z_r} y_1^{z_1} = c^e d \bmod N^2 \text{ and } g^{z_r'} y_1^{z_1'} = c^{e'} d \bmod N^2.$$

Define $\Delta e = e - e'$ (without loss of generality, assume that $\Delta e > 0$), $\Delta z_1 = z_1 - z_1'$, and $\Delta z_r = z_r - z_r'$, then we have

$$g^{\Delta z_r} y_1^{\Delta z_1} = c^{\Delta e} \bmod N^2. \tag{6}$$

Case 1: $\Delta e \mid \Delta z_r \wedge \Delta e \mid \Delta z_1$. This case is divided into two subcases based on the parity of Δe.

- Case 1.1: Δe is odd. In this case, Δe is co-prime to $\varphi(N^2) = 4p'q'pq$, since $\Delta e \ll p', q', p, q$. Let $\widetilde{\Delta e}$ be the inverse of Δe modulo $\varphi(N^2)$. By Equation (6), we have

$$\begin{aligned} c &= c^{\Delta e \cdot \widetilde{\Delta e}} \bmod N^2 \\ &= g^{\Delta z_r \cdot \widetilde{\Delta e}} y_1^{\Delta z_1 \cdot \widetilde{\Delta e}} \bmod N^2 \\ &= g^{\Delta z_r / \Delta e} y_1^{\Delta z_1 / \Delta e} \bmod N^2, \end{aligned}$$

  which indicates that $(\Delta z_1/\Delta e, \Delta z_r/\Delta e)$ is a valid opening.
- Case 1.2: Δe is even. Let $\Delta e = 2^\rho \Delta e'$ for an odd $\Delta e'$ and $\rho \geq 1$. By this, $\Delta e'$ is co-prime to $\varphi(N^2) = 4p'q'pq$, since $\Delta e' < \Delta e \ll p', q', p, q$. Let $\widetilde{\Delta e'}$ be the inverse of $\Delta e'$ modulo $\varphi(N^2)$. By Equation (6), we have

$$\begin{aligned} c^{2^\rho} &= c^{\Delta e/\Delta e'} \bmod N^2 = c^{\Delta e \cdot \widetilde{\Delta e'}} \bmod N^2 \\ &= g^{\Delta z_r \cdot \widetilde{\Delta e'}} y_1^{\Delta z_1 \cdot \widetilde{\Delta e'}} \bmod N^2 \\ &= g^{\Delta z_r / \Delta e'} y_1^{\Delta z_1 / \Delta e'} \bmod N^2. \end{aligned}$$

  Define $Z = c^{-1} g^{\Delta z_r/\Delta e} y_1^{\Delta z_1/\Delta e} \bmod N^2$ and we get

$$\begin{aligned} Z^{2^\rho} &= c^{-2^\rho} g^{2^\rho \Delta z_r/\Delta e} y_1^{2^\rho \Delta z_1/\Delta e} \bmod N^2 \\ &= c^{-2^\rho} g^{\Delta z_r/\Delta e'} y_1^{\Delta z_1/\Delta e'} \bmod N^2 \\ &= c^{-2^\rho} c^{2^\rho} \bmod N^2 = 1 \bmod N^2. \end{aligned}$$

  From Fact 8, we have

$$Z^2 = 1 \bmod N.$$

  - If $Z = \pm 1 \bmod N$. In this case, our goal is to rule out the situation where $Z \neq \pm 1 \bmod N^2$, which neither provides a witness nor leads to factorization. Let $Z = \pm 1 + kN$ for some $k \in \mathbb{Z}$, then $Z^{2^\rho} = (\pm 1 + kN)^{2^\rho} = 1 \pm 2^\rho kN \bmod N^2$. Since $Z^{2^\rho} = 1 \bmod N^2$, we have $2^\rho kN = 0 \bmod N^2$. From the fact that $\gcd(2^\rho, N^2) = 1$ ($2^\rho \leq \Delta e \ll p, q$), necessarily $kN = 0 \bmod N^2$. Thus, $Z = \pm 1 \bmod N^2$. Thus, from the definition of Z, we get a valid opening $(\Delta z_1/\Delta e, \Delta z_r/\Delta e)$.
  - If $Z \neq \pm 1 \bmod N$. In this case, Z is a non-trivial square root of 1 modulo N. From Fact 7, this gives a factorization of N.

Case 2: $\Delta e \nmid \Delta z_r \vee \Delta e \nmid \Delta z_1$. We show by contradiction that this case happens only with negligible probability. If not, we construct a simulator $\mathcal{B}$ to solve the Paillier root problem with non-negligible probability based on the two accepting conversations. Given a Paillier root problem instance (N, T) where $T \leftarrow 2\mathsf{NR}_{N^2}$, $\mathcal{B}$ sets $g = T$, selects $\alpha_1 \leftarrow \mathbb{Z}_N$ and computes $y_1 = g^{\alpha_1}(1+N)^{\beta_1} \bmod N^2$ for any known $\beta_1 \in \mathbb{Z}_N$. The public parameter of PaillCom (with $l = 1$) is set as $pp = (N, g, y_1)$.

Since $y_1 = g^{\alpha_1}(1+N)^{\beta_1} \bmod N^2$, Equation (6) can be transformed into

$$g^{\alpha_1 \Delta z_1 + \Delta z_r}(1+N)^{\beta_1 \Delta z_1} = c^{\Delta e} \bmod N^2. \tag{7}$$

Denote the inverse of Δe modulo N as Δe^{-1} (note that the order of $(1+N)$ is N, and $\gcd(\Delta e, N) = 1$ since $\Delta e \ll p, q$), then define $\tilde{c}$ as

$$\tilde{c} = \frac{c}{(1+N)^{\beta_1 \Delta z_1 \cdot \Delta e^{-1}}} \bmod N^2, \tag{8}$$

and according to Equation (7) we can get

$$g^{\alpha_1 \Delta z_1 + \Delta z_r} = \tilde{c}^{\Delta e} \bmod N^2. \tag{9}$$

Let $\eta = \gcd(\Delta e, \alpha_1 \Delta z_1 + \Delta z_r)$, then by extended Euclidean algorithm we can find $\gamma, \delta \in \mathbb{Z}$ such that

$$\gamma \Delta e + \delta(\alpha_1 \Delta z_1 + \Delta z_r) = \eta.$$

Thus, from Equation (9),

$$g^\eta = g^{\gamma \Delta e + \delta(\alpha_1 \Delta z_1 + \Delta z_r)} = \left(g^\gamma \tilde{c}^\delta\right)^{\Delta e} \bmod N^2. \tag{10}$$

- Case 2.1: $\Delta e \nmid \alpha_1 \Delta z_1 + \Delta z_r$. Recall that $\eta = \gcd(\Delta e, \alpha_1 \Delta z_1 + \Delta z_r)$, so in this case, $\Delta e/\eta > 1$ (otherwise if $\Delta e = \eta$ then we have $\Delta e \mid \alpha_1 \Delta z_1 + \Delta z_r$).
  When η is odd, it is co-prime to $\varphi(N^2) = 4p'q'pq$, since $\eta < \Delta e \ll p', q', p, q$. Similar to the discussion of case 1.1, from Equation (10) we have

$$\left(g^\gamma \tilde{c}^\delta\right)^{\Delta e/\eta} = g = T \bmod N^2.$$

Also because $\gcd(\Delta e/\eta, N) = 1$ (since $\Delta e/\eta \leq \Delta e \ll p, q$) and $\Delta e/\eta > 1$, we get that $(g^{\gamma}\tilde{c}^{\delta}, \Delta e/\eta)$ is a solution to the Paillier root problem.
When η is even, let $\eta = 2^{\rho}\eta'$ for an odd η' and $\rho \geq 1$, we define $Z = g^{-1}\left(g^{\gamma}\tilde{c}^{\delta}\right)^{\Delta e/\eta} \bmod N^2$. Similar to the discussion of case 1.2, from Equation (10),

$$Z^{2^{\rho}} = 1 \bmod N^2.$$

Also similar to the discussion of case 1.2, we conclude that $Z = \pm 1 \bmod N^2$, otherwise, Z will lead to factoring N. If $Z = 1 \bmod N^2$, it is easy to see that $(g^{\gamma}\tilde{c}^{\delta}, \Delta e/\eta)$ gives a solution to the Paillier root problem. If $Z = -1 \bmod N^2$, we have

$$-\left(g^{\gamma}\tilde{c}^{\delta}\right)^{\Delta e/\eta} = g = T \bmod N^2.$$

In this case, $\Delta e/\eta$ must be odd. If not the case, since $-1 \notin \mathrm{QR}_{N^2}$, $\left(g^{\gamma}\tilde{c}^{\delta}\right)^{\Delta e/\eta} \in \mathrm{QR}_{N^2}$ and $g \in \mathrm{QR}_{N^2}$, the above equation will not hold. Thus,

$$\left(-g^{\gamma}\tilde{c}^{\delta}\right)^{\Delta e/\eta} = g = T \bmod N^2,$$

and $(-g^{\gamma}\tilde{c}^{\delta}, \Delta e/\eta)$ gives a solution to the Paillier root problem.

- Case 2.2: $\Delta e \mid \alpha_1 \Delta z_1 + \Delta z_r$. Let $\tilde{p}$ be some prime factor of Δe satisfying the following property: suppose that ν is the largest integer such that $\tilde{p}^{\nu} \mid \Delta e$, then $\tilde{p}^{\nu} \nmid \Delta z_1 \vee \tilde{p}^{\nu} \nmid \Delta z_r$. Such a $\tilde{p}$ must exist due to the condition $\Delta e \nmid \Delta z_r \vee \Delta e \nmid \Delta z_1$ that defines case 2.
  If $\tilde{p}^{\nu} \mid \Delta z_1$, then from the condition $\Delta e \mid \alpha_1 \Delta z_1 + \Delta z_r$ defining case 2.2, we have that $\tilde{p}^{\nu} \mid \Delta z_r$, which is a contradiction. Thus, we have $\tilde{p}^{\nu} \nmid \Delta z_1$.
  Recall that $\alpha_1 \in \mathbb{Z}_N$ and $N = pq = (2p'+1)(2q'+1) = 4p'q' + 2p' + 2q' + 1$. Define $\alpha_1' = \alpha_1 \bmod p'q'$, then $\alpha_1 = \alpha_1' + \zeta(p'q')$ for some entirely random $\zeta \in [0, 3]$ (note that α_1' lies in $[0, 2p' + 2q']$ only with negligible probability, so we can just omit the possible case that $\zeta = 4$). We consider the probability of (a necessary condition of case 2.2)

  $$\tilde{p}^{\nu} \mid \alpha_1 \Delta z_1 + \Delta z_r,$$

  or equivalently,

  $$\tilde{p}^{\nu} \mid \zeta(p'q')\Delta z_1 + \alpha_1' \Delta z_1 + \Delta z_r. \tag{11}$$

  Suppose there are two distinct values of ζ, e.g., $\zeta_1 > \zeta_2$, that can make Equation (11) hold. Then define $\Delta\zeta = \zeta_1 - \zeta_2 \in [1, 3]$, and we have $\tilde{p}^{\nu} \mid \Delta\zeta(p'q')\Delta z_1$. Since $\gcd(\tilde{p}^{\nu}, p'q') = 1$ ($\tilde{p}^{\nu} \leq \Delta e \ll p', q'$) and $\tilde{p}^{\nu} \nmid \Delta z_1$, we get $\tilde{p} \mid \Delta\zeta$ and then $\tilde{p} = 2$ ($\Delta\zeta$ can only be 2) or $\tilde{p} = 3$ ($\Delta\zeta$ can only be 3). For both of the above cases, there are at most two distinct values of ζ such that the Equation (11) holds. Thus, we conclude that case 2.2 happens with probability at most $2/4 = 1/2$.

Note that if we get a valid opening in case 1, $\Delta z_1/\Delta e$ lies in the range $[-2^{s+t}B_1, 2^{s+t}B_1]$, since $z_1, z_1' \in [0, 2^{s+t}B_1]$ and $\Delta e \geq 1$, which illustrates why an accepted proof ensures our previous claimed range (with slack).

PoK for General l. By verification phase, we get $g^{\Delta z_r}\prod_{i=1}^{l} y_i^{\Delta z_i} = c^{\Delta e} \bmod N^2$ from two accepting conversations $(d, e, (\{z_i\}_{i\in[1,l]}, z_r))$ and $(d, e', (\{z_i'\}_{i\in[1,l]}, z_r'))$ with $e \neq e'$, where we define $\Delta e = e - e'$ (without loss of generality, assume that $\Delta e > 0$), $\Delta z_i = z_i - z_i'$ for $i \in [1, l]$ and $\Delta z_r = z_r - z_r'$. From the generation of y_i-elements: $y_i = g^{\alpha_i}(1+N)^{\beta_i} \bmod N^2$, the above equation is equivalent to

$$g^{\Delta z_r + \sum_{i=1}^{l}\alpha_i \Delta z_i}(1+N)^{\sum_{i=1}^{l}\beta_i \Delta z_i} = c^{\Delta e} \bmod N^2.$$

Denote the inverse of Δe modulo N as Δe^{-1} (note that $\gcd(\Delta e, N) = 1$ since $\Delta e \ll p, q$), then define $\tilde{c}$ as

$$\tilde{c} = \frac{c}{(1+N)^{(\sum_{i=1}^{l}\beta_i \Delta z_i)\Delta e^{-1}}} \bmod N^2.$$

Combining the above two equations, we can get

$$g^{\Delta z_r + \sum_{i=1}^{l}\alpha_i \Delta z_i} = \tilde{c}^{\Delta e} \bmod N^2.$$

Equivalently, for any $k \in [1, l]$, we have

$$g^{\alpha_k \Delta z_k + (\sum_{i\in[1,l]-\{k\}}\alpha_i \Delta z_i + \Delta z_r)} = \tilde{c}^{\Delta e} \bmod N^2.$$

Under the factoring and Paillier root assumptions, Δe divides both Δz_k and $\sum_{i\in[1,l]-\{k\}}\alpha_i \Delta z_i + \Delta z_r$ with overwhelming probability for any $k \in [1, l]$, from the analysis for PoK with $l = 1$ (compare it with Equation (9)). Then it can be induced that Δe divides every Δz_i along with Δz_r.

- If Δe is odd, by a similar argument in the analysis for special case $l = 1$, we can immediately get a valid opening $((\Delta z_i/\Delta e)_{i\in[1,l]}, \Delta z_r/\Delta e)$.
- If Δe is even, define $\Delta e = 2^{\rho}\Delta e'$ for an odd $\Delta e'$ and $\rho \geq 1$. Let $Z = c^{-1}g^{\Delta z_r/\Delta e}\prod_{i=1}^{l} y_i^{\Delta z_i/\Delta e} \bmod N^2$. Also from a similar argument in the special case analysis, we have that $Z^{2^{\rho}} = 1 \bmod N^2$, which will lead to the factorization of N or give a valid opening $((\Delta z_i/\Delta e)_{i\in[1,l]}, \Delta z_r/\Delta e)$.

Note that if we get a valid opening, each $\Delta z_i/\Delta e$ lies in the range $[-2^{s+t}B_i, 2^{s+t}B_i]$, since $z_i, z_i' \in [0, 2^{s+t}B_i]$ and $\Delta e \geq 1$, which illustrates why an accepted proof ensures our previous claimed range (with slack).

5 COMPARISON

In this section, we conduct a comparative analysis of existing range proof schemes introduced by [33] and [18] for the Paillier cryptosystem against our proposed methods, both theoretically and experimentally.

We do not consider the generalized version of Paillier cryptosystem, and set $\zeta = 1$ (i.e., compute over $\mathbb{Z}_{N^2}$) when considering [18]. Besides, we consider the non-interactive versions of proofs obtained via Fiat-Shamir transformation in the random oracle model [23].

Besides, as indicated in Section 1.5, our range proofs are not applicable when the prover knows the factorization of the RSA modulus or the related discrete logarithms. In this case, an auxiliary integer commitment is required, which introduces additional overhead. Specifically, for our MtA protocol described in Section 6.1, we include an auxiliary integer commitment in π_b, as detailed in Step 1-(b), Phase Trans (despite this additional overhead, our MtA protocol still has advantages within Paillier-based constructions). A concrete range proof protocol in this case can be found in Appendix E, along with an independent performance evaluation.

Table 2: Theoretical Comparison of Range Proofs

(a) For Paillier Plaintext

Methods	Communication (bit)	Computation (E)		Exact
		Prove	Verify	
LN [33]	$3\|N\| + 2s + 3t + \|B\|$	$4.5 + \frac{2s+3t+2\|B\|}{\|N\|}$	$3.5 + \frac{2s+5.5t+\|B\|}{\|N\|}$	×
DLP [18]	$11\|N\| + 4s + 5t + 4\|B\|$	$20 + \frac{10s+15t+14\|B\|}{\|N\|}$	$12.5 + \frac{10s+20t+10\|B\|}{\|N\|}$	✓
Ours	$\|N\| + 2s + 3t + \|B\|$	$2.5 + \frac{5s+5t+2.5\|B\|}{\|N\|}$	$2.5 + \frac{5s+7.5t+2.5\|B\|}{\|N\|}$	×

(b) For Paillier Affine Operation

Methods	Communication (bit)	Computation (E)		Exact
		Prove	Verify	
LN [33]	$5\|N\| + 4s + 5t + \|B_1\| + \|B_2\|$	$6.5 + \frac{6.5s+7.5t+4.5\|B_1\|+2\|B_2\|}{\|N\|}$	$4.5 + \frac{6.5s+11t+3.5\|B_1\|+\|B_2\|}{\|N\|}$	×
Ours	$\|N\| + 3s + 4t + \|B_1\| + \|B_2\|$	$2.5 + \frac{7.5s+7.5t+2.5\|B_1\|+2.5\|B_2\|}{\|N\|}$	$2.5 + \frac{7.5s+10t+2.5\|B_1\|+2.5\|B_2\|}{\|N\|}$	×

Notes: E refers to the computational overheads of an exponentiation operation within $\mathbb{Z}_N^*$ with the exponent of bit length close to $|N|$. We use s and t to denote the statistical and soundness parameters, respectively. As to the range proof for Paillier plaintext, the range to be proven is $[0, B]$. For the Paillier affine operation, we assume the ranges to be proven are $[0, B_1]$ and $[0, B_2]$, respectively.

5.1 Theoretical Analysis

A detailed comparison of [33], [18], and our methods is presented in Table 2. The unit for communication is measured in bits. E symbolizes the computational overheads of an exponentiation operation within $\mathbb{Z}_N^*$, where the corresponding exponent has a bit length approximately equal to $|N|$. For an exponentiation in $\mathbb{Z}_N^*$ with a corresponding exponent of bit length L, the time cost is estimated to be $\frac{L}{|N|}$E. In cases where the exponentiation occurs in $\mathbb{Z}_{N^2}^*$, the time cost is projected to be approximately 2.5 times higher than that in $\mathbb{Z}_N^*$ for the same exponent bit length. For proofs concerning bounded Paillier plaintext, we consider the range to be proven as $[0, B]$. In the context of Paillier affine operations, we assume the ranges to be proven are $[0, B_1]$ and $[0, B_2]$, respectively. In alignment with our prior discussions, we use s and t to denote the statistical and soundness parameters, respectively.

As to proofs for bounded Paillier plaintext, Table 2 clearly indicates that the exact method proposed by [18] incurs significantly higher costs in terms of bandwidth and computation. Additionally, it is important to note that the table does not account for the computational overheads of the 3-square decomposition required to generate the range proof, which is also expensive in practice. Thus, when applications tolerate slack, opting for range proofs with slack is preferable over exact ones.

In the context of range proofs for both Paillier plaintext and Paillier affine operations, our approach offers a more compact solution in terms of bandwidth compared to the method proposed by [33]. This advantage becomes particularly pronounced in practical scenarios where the parameters s, t, and $|B|$ (or $|B_1|$, $|B_2|$) are relatively small compared to $|N|$. Moreover, under such typical parameters, our methods also demonstrate computational benefits over the approaches by [33], which will become more obvious in the subsequent experimental analysis.

Table 3: Experimental Comparison of Range Proofs

(a) For Paillier Plaintext

Methods	$\|B\|$ (bit)	Communication (KiB)	Computation (ms)	
			Prove	Verify
LN [33]	256	1.22	32.5	26.5
	512	1.25	33.5	27
	1024	1.32	36	28
DLP [18]	256	4.37	153	102
	512	4.49	161	108
	1024	4.74	177	119
Ours	256	0.47	21	22
	512	0.50	22.5	23.5
	1024	0.57	25.5	26

(b) For Paillier Affine Operation

Methods	Communication (KiB)	Computation (ms)	
		Prove	Verify
LN [33]	2.12	55.5	39.5
Ours	0.60	27	28

5.2 Experimental Analysis

We implement the above range proof schemes and execute a series of experiments. These implementations are developed using the Go language and conducted on a 24-inch iMac, equipped with an Apple M1 chip and 16 GB of RAM, running macOS Sonoma 14.2.1.

Our results are based on a standard choice of parameters, specifically $(s, t, |N|) = (80, 128, 3072)$. The unit for communication is kibibytes (KiB), while that for computation is millisecond (ms).

Table 4: Average Cost per Ciphertext Compared to TBM+ Range Proof with $|B| = 512$ bits

Methods	Communication (KiB)		Computation (ms)		
	Batch Size	Result	Batch Size	Prove	Verify
TBM+ [39]	20	2.88	30	74	77.5
	50	1.16	60	37	41
	80	0.73	90	25	28
	110	0.54	120	18.5	22
	140	0.42	150	15	19
Ours	$*$	0.50	$*$	22.5	23.5

Note: '$*$' denotes 'arbitrary'.

Furthermore, the results are rounded to the nearest 0.01 for bandwidth, while they are rounded to the nearest 0.5 when considering computational overheads. Our results are shown in Table 3.

Paillier Plaintext. In the experiments conducted, the sizes considered for B are 256, 512, and 1024 bits. In practical scenarios, it is often unnecessary to have larger ranges.

As shown in Table 3, it is evident that the exact range proof method presented in [18] incurs substantially higher communication and computation overheads. This observation is in concordance with the aforementioned argument that for applications which tolerate slack, adopting an exact range proof method is not the most advantageous strategy.

When comparing our method with that from [33], there are notable improvements in both bandwidth and computation. Specifically, our method reduces the communication overheads by approximately 60%. From the computational perspective, the cost of prover in our method shows an improvement ranging from 29% to 35%, while the cost of verifier is also reduced by 7% to 17%.

Comparison to Batch Proof. In practical scenarios, it may be necessary to prove the ranges of a series of plaintexts (under certain Paillier ciphertexts). In such cases, a trivial approach is to use our range proof for each plaintext. However, when the number of plaintexts to be proven is large, more efficient solutions exist. Specifically, we compare our approach with the TBM+ range proof from [39, Section 4.5]. TBM+ can be adapted to (batch-)prove the ranges of Paillier plaintexts due to the homomorphism of Paillier encryption. Essentially, it is a cut-and-choose range proof tailored for proving multiple witnesses in batch, requiring t parallel executions to achieve a soundness error of 2^{-t} [39, Appendix C]. In contrast, our range proof uses a large challenge space, achieving negligible soundness error in one-shot. As a result, when the batch size is small, the parallel executions required by TBM+ can become relatively costly.

For a concrete comparison, we make the following assumptions: (1) $(s, t, |N|) = (80, 128, 3072)$ as we set before; (2) $|B| = 512$ bits (we can get similar results for 256 and 1024 bits); (3) when the batch size is smaller than the length of D-element (in $\mathbb{Z}_{N^2}$), the non-interactive version of TBM+ by Fiat-Shamir transformation can be further optimized by replacing D-elements with shorter $\mathbf{t}$-elements in the proof. Accordingly, the verifier will compute the D-elements first and then check the hash validity. Please refer to [39, Section 4.5] for details. We conduct experiments to compare the average cost of our approach and TBM+, with the results shown in Table 4. From these results, we can observe that when the batch size is less than 90, our approach is more efficient in terms of both computation (for prover and verifier) and communication. If the batch size is less than 110, our approach is more efficient in terms of communication. Therefore, when conducting range proofs for more than about 110 Paillier ciphertexts, TBM+ is desirable, whereas for fewer than approximately 90, our approach is preferable. Thus, both range proofs have their respective application scenarios.

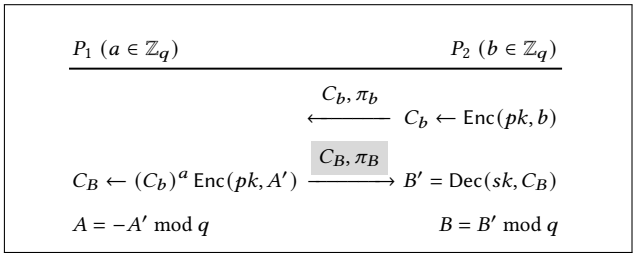

Figure 2: Illustration of MtA Protocol

Paillier Affine Operation. Since the range proof for Paillier affine operation serves as a critical component of the MtA protocol, here we conduct the experiments in alignment with the requirements of MtA. For a common MtA protocol that operates within $\mathbb{Z}_q$, with q of bit length 256, $|B_1|$ is equal to 256. Moreover, $|B_2|$ is determined by the equation $|B_2| = 2s + t + 2|B_1|$, which equates to 800.

Compared with [33], our range proof for Paillier affine operation highlights significant enhancements in efficiency. In terms of bandwidth, our approach reduces the cost by about 72%. Besides, the computational overheads are reduced by 51% for the prover, and 29% for the verifier, respectively.

6 APPLICATIONS

6.1 Multiplicative-to-Additive Protocol

We show that our techniques can be used to construct an MtA protocol, which has advantages over existing Paillier-based solution.

General Idea. Figure 2 gives a rough illustration how an MtA protocol can be constructed from additively homomorphic encryptions, such as Paillier encryption. Assume that the public and secret keys pair of P_2 is (pk, sk), and the RSA modulus is N. First, P_2 encrypts its private input b under its public key pk to obtain C_b. Next, with its private input a, P_1 computes an affine operation denoted by $C_B \leftarrow (C_b)^a \mathsf{Enc}(pk, A')$ for a randomly chosen A', which results in an encryption of $ab + A'$. Then, P_2 decrypts C_B with its secret key sk to get B'. Finally, P_1 gets $A = -A' \bmod q$, while P_2 obtains $B = B' \bmod q$. It is easy to check that $ab = A + B \bmod q$ holds. Apart from the ciphertexts, it is important to generate range proofs to ensure that no reduction modulo N occurs. Specifically, π_b is to bound the range of b, while π_B is to bound the ranges of a and A'.

Our Construction. Our protocol consists of two phases, Setup and Trans. P_1 and P_2 only need to conduct Setup phase once, then they can run Trans phase many times. We emphasize that even though our protocol overall follows the above general idea, there are few differences. Looking ahead, the affine operation by P_1 should be twice as large as the original one, i.e., $C_B \leftarrow (C_b)^{2a} \mathsf{Enc}(pk, 2A')$,

in case C_b is the negative of the encryption of b for malicious P_2. To cater to this, P_2 should additionally divide the decryption result by 2 in the final step.

- Setup. P_1 invokes PedCom.Setup (refer to Appendix C) to obtain its public parameter $pp = (\tilde{N}, \tilde{g}, \tilde{y})$. To ensure the validity of pp, P_1 generates 3 zero-knowledge proofs. The first proof demonstrates that $\tilde{N}$ is the product of two primes using the technique introduced by [6]. The second and third proofs utilize $\mathsf{ZK}_{\mathsf{QR}_N}$ and $\mathsf{ZK}_{\mathsf{DL}_N}$ from Appendix D to show that $\tilde{g}$ is a quadratic residue modulo $\tilde{N}$, and there exists an $\tilde{\alpha}$ such that $\tilde{g}^{\tilde{\alpha}} = \tilde{y} \bmod \tilde{N}$, respectively. P_2 calls MPaill.KGen to get its secret key sk and public key $pk = (N, g, y)$. Then P_2 computes a zero-knowledge proof for the statement that N is the product of two primes, as described by [6]. Besides, P_2 generates proofs to demonstrate that g is a $2N$-th residue modulo N^2 using $\mathsf{ZK}_{\mathsf{2NR}_{N^2}}$, and that there is an α such that $g^\alpha = y/(1+N) \bmod N^2$ via $\mathsf{ZK}_{\mathsf{DL}_{N^2}}$, both from Appendix D. Furthermore, it is crucial for each party to verify the proofs of the other party before proceeding with subsequent steps.
- Trans. For each run of this phase, P_1 and P_2 engage in the following interactive protocol with their private input $a \in \mathbb{Z}_q$ and $b \in \mathbb{Z}_q$, then finally receive A and B, respectively, such that $ab = A + B \bmod q$.
  1. P_2's message:
     (a) Compute $C_b \leftarrow \mathsf{MPaill.}\widetilde{\mathsf{Enc}}(pk, b)$.
     (b) Generate a proof π_b that b lies in the range $[0, q]$, via $\mathsf{ZK}_{\mathsf{MPaillPed}}$ shown in Appendix E. Note that P_2 should commit to b via PedCom under P_1's public parameter pp, which is included in π_b.
     (c) Send (C_b, π_b) to P_1.
  2. P_1's message and output:
     (a) Verify π_b, and abort when it fails.
     (b) Compute the affine operation

$$C_B \leftarrow \left(C_b \cdot y^{2^{s+t}q}\right)^{2a} \cdot \mathsf{MPaill.Enc}(pk, 2A') \bmod N^2,$$

for $A' \leftarrow [0, 2^{2s+t+1}q^2]$.
     (c) Calculate a range proof π_B for $2a \in [0, 2q]$ and $2A' \in [0, 2^{2s+t+2}q^2]$, via $\mathsf{ZK}_{\mathsf{PaillCom}}$.
     (d) Send (C_B, π_B) to P_2, output $A = -A' \bmod q$.
  3. P_2's output: Verify π_B, and when the verification is accepted, output $B = \mathsf{Dec}(sk, C_B)/2 \bmod q$.

Similar to [33], the above protocol is correct if there is no reduction modulo N during the Trans phase. By the range proofs π_b and π_B, the upper bound of the underlying plaintext $2a(b + 2^{s+t}q) + 2A'$ of C_B is

$$\begin{aligned} &2^{s+t} \cdot 2q(2^{s+t}q + 2^{s+t}q) + 2^{s+t} \cdot 2^{2s+t+2}q^2 \\ &= 2^{2s+2t+2}q^2 + 2^{3s+2t+2}q^2 < 2^{3s+2t+3}q^2, \end{aligned}$$

which is smaller than N for typical parameter settings, such as our choice $(s, t, |N|, |q|) = (80, 128, 3072, 256)$.

Theorem 6.1. *Given that* MPaill *is IND-CPA secure,* PedCom *has hiding property, and the involved* Σ*-protocols (including* $\mathsf{ZK}_{\mathsf{MPaillPed}}$, $\mathsf{ZK}_{\mathsf{PaillCom}}$, $\mathsf{ZK}_{\mathsf{QR}_N}$, *and etc.) are secure, the above protocol securely realizes* $\mathcal{F}_{\mathsf{MtA}}$ *in the presence of a malicious static adversary.*

Table 5: Comparison of MtAs from Paillier

Schemes	Communication (KiB)	Computation (ms)	
		P_1	P_2
LN [33]	4.84	103.5	110
Ours	3.34	78.5	96.5

The complete proof is deferred to Appendix I and we present a sketch here. The Setup phase can be easily simulated by the corresponding zero-knowledge simulators and we only need to deal with the Trans phase. There are two cases, i.e., the adversary $\mathcal{A}$ corrupts P_1 or P_2, and we need to construct a simulator $\mathcal{S}$ to simulate the view of $\mathcal{A}$ accordingly based on the public parameters and the output of $\mathcal{F}_{\mathsf{MtA}}$. (1) $\mathcal{S}$ simulates P_1 when P_2 is corrupted by the adversary. $\mathcal{S}$ receives the tuple (C_b, π_b) that $\mathcal{A}$ instructs P_2 to send with sid. If π_b is accepted, $\mathcal{S}$ can extract b via the knowledge extractor of $\mathsf{ZK}_{\mathsf{MPaillPed}}$ from Appendix E. Then $\mathcal{S}$ queries $\mathcal{F}_{\mathsf{MtA}}$ with (sid, b) and receives P_2's output (sid, B). Next, $\mathcal{S}$ picks $\xi \leftarrow [0, 2^{2s+t+2}q]$ and computes C_B as the encryption of $2B + \xi q$. Additionally, π_B can be simulated by the zero-knowledge simulator of $\mathsf{ZK}_{\mathsf{PaillCom}}$. (2) $\mathcal{S}$ simulates P_2 when P_1 is corrupted by the adversary. $\mathcal{S}$ generates C_b and the related Pedersen commitment in π_b with a same random value. It then computes the other parts of π_b via the zero-knowledge simulator of $\mathsf{ZK}_{\mathsf{MPaillPed}}$ from Appendix E.

Comparison. We compare our MtA construction with the one presented by [33], which similarly utilizes the Paillier cryptosystem. Both constructions follow the general idea depicted in Figure 2. Our approach notably improves the performance of the parts associated with the gray box. In particular, π_B refers to the range proof for Paillier affine operation. By applying our direct range proof method, π_B becomes more compact, more efficient to generate and verify, which will benefit the communication, the computation of P_1 and P_2 in MtA protocol, respectively.

We present a comparison of the two schemes from the experimental aspect. It's important to note that the MtAs operate over $\mathbb{Z}_q$, for which we have assigned q as the order of secp256k1, resulting in $|q| = 256$. The other conventions and experimental environments align with those described in Section 5, thus we will not duplicate those details here. The comprehensive results are displayed in Table 5. Approximately, our scheme achieves a reduction in communication costs by 31%, lowers the computational overhead for P_1 by 24%, and reduces that for P_2 by 12%.

We also give an experimental comparison of the Setup phase, and the results are shown in Table 6 (the cost of related zero-knowledge proofs for the well-formedness). Even if it takes more time in our MtA, we emphasize that the Setup phase only needs to be executed once. Thus, the inefficiency is acceptable in practice.

Improvements to Threshold ECDSA. We demonstrate the improvements of integrating our MtA protocol into the Paillier-based threshold ECDSAs from [33, 41]. Specifically, we benchmark the distributed signing phase of improved and original schemes in a special two-party setting, with the results shown in Table 7. It can

Table 6: Comparison of Setup in MtAs from Paillier

Schemes	Communication (KiB)	Computation (ms)	
		P_1	P_2
LN [33]	435.24	6544	6392
Ours	435.24	10059	9746

Table 7: Comparison of Distributed Signing in Paillier-Based Threshold (2-out-of-n) ECDSAs

Threshold ECDSA	Communication (KiB)	Computation (ms)
[33]	23.09	896
Our MtA to [33]	17.09	756
[41]	5.19	211
Our MtA to [41]	3.69	180

be observed that the percentage improvement of the threshold ECDSAs is close to that of the MtA, due to the fact that as a building block, MtA dominates the overall complexity.

Additionally, we briefly compare the differences between Paillier-based and OT-based threshold ECDSAs (use [19] as an example). Roughly speaking, OT-based schemes have faster signing times at the expense of high communication cost. Specifically, [19] achieves a signing time of 10-20 ms in the 2-party setting with a communication cost of 99.4 KiB. On the other hand, when integrating our Paillier-based MtA in [41], we obtain a scheme with slower signing times (about 180 ms in 2-party) and a much lower communication cost (3-4 KiB in 2-party).

6.2 Naor-Yung CCA2

We show that our techniques can also be used to give an efficient DCR-based instantiation of Naor-Yung CCA2 paradigm.

Construction from Range Proof with Slack. Firstly recall the Naor-Yung paradigm. This framework, when applied to a CPA-secure PKE scheme, enables the transformation into a CCA-secure variant. In this paradigm, the recipient possesses two sets of key pairs, and the sender has to encrypt the message to send using both public keys, then compute a non-interactive zero-knowledge proof for the equality of respective plaintexts. Upon receiving these ciphertexts and the equality proof, the recipient must check the validity of this proof prior to decrypting any of the ciphertexts. However, applying this paradigm to the CPA-secure Paillier encryption introduces problems due to the distinct RSA moduli in the dual public keys, potentially leading to security issues, as identified by [17]. Fortunately, [17] also proposes a remedy for this issue through an auxiliary proof to show that the plaintext is smaller than each respective RSA modulus. This solution is straightforward when employing an exact range proof (also trivial when the slack does not introduce negative range). Our focus here is to demonstrate a resolution when using a range proof with slack that introduce negative range. Suppose that the CPA-secure Paillier encryption is

Table 8: Comparison of DCR-Based Naor-Yung CCA

Range Proof	$\|M\|$ (bits)	Ciphertext (KiB)	Computation (ms)	
			Sender	Receiver
LN [33]	256	3.10	85.5	62
	512	3.13	86.5	62.5
	1024	3.19	88.5	63.5
Ours	256	2.37	77	58
	512	2.40	80	59.5
	1024	2.47	85	62.5

denoted as $(\widehat{\mathsf{KGen}}, \widehat{\mathsf{Enc}}, \widehat{\mathsf{Dec}})$. Besides, if the completeness and zero-knowledge range of the related range proof is $[0, B]$, we assume the corresponding soundness range is $[-RB, RB]$ for some $R > 1$.

- $\mathsf{KGen}(1^\kappa)$. Invoke $\widehat{\mathsf{KGen}}(1^\kappa)$ twice to get two pairs of keys (pk_1, sk_1) and (pk_2, sk_2). Return (pk, sk) as
$$pk = (pk_1, pk_2), sk = (sk_1, sk_2).$$
Assume that the RSA moduli in pk_1 and pk_2 are N_1 and N_2 respectively, then define $N = \min\{N_1, N_2\}$. Furthermore, the message space of the scheme can be defined as $\mathcal{M} = [RM, (R+1)M]$, as long as $M < \frac{N}{2R}$.
- $\mathsf{Enc}(pk, m)$. For $pk = (pk_1, pk_2)$ and $m \in \mathcal{M}$, call $\widehat{\mathsf{Enc}}(pk_1, m-RM)$ and $\widehat{\mathsf{Enc}}(pk_2, m-RM)$ to get C_1 and C_2 respectively. Compute a non-interactive zero-knowledge proof π, for the equality of plaintext under C_1 and C_2, and that the plaintext is in the range $[0, M]$. Return $C = (C_1, C_2, \pi)$.
- $\mathsf{Dec}(sk, C)$. For $sk = (sk_1, sk_2)$ and $C = (C_1, C_2, \pi)$, firstly check the validity of π. If not, return $\perp$. Otherwise, call $m' = \widehat{\mathsf{Dec}}(sk_1, C_1)$, return $m = m' + RM$.

The proof π ensures the plaintext $m - RM$ lies in the range $[-RM, RM]$, equivalently, $m \in [0, 2RM] \subset [0, N]$, which fulfills the suggestion in [17] for constructing Naor-Yung CCA2 PKE based on the DCR assumption, to overcome the incompleteness of [24].

Instantiations. As for the range proof for Paillier plaintext, we can select the method from [33], or ours that is adapted from $\mathsf{ZK}_{\mathsf{PaillCom}}$. For both cases, we use s and t to represent the statistical and soundness parameters, respectively. Then R is equal to 2^{s+t}. Without loss of generality, we assume that the range proof is actually performed upon C_1. Since we use the Fiat-Shamir transformation [23] in the random oracle model to make the range proofs non-interactive, the requirement of Naor-Yung CCA2 for simulation soundness can be fulfilled (refer to [22] for a discussion of this property).

First we consider the range proof from [33], and the underlying CPA-secure scheme should be the original Paillier encryption Paill. In order for the sender proving the range of plaintext under C_1, the recipient should include the public parameter of a related Pedersen commitment in pk_1.

If we use our range proof method for plaintext, we need to set the CPA-secure scheme as our modified Paillier encryption MPaill. C_1 should be generated from MPaill.Enc so that the sender can apply our direct range proof to it. C_2 can be alternatively computed from $\mathsf{MPaill}.\widehat{\mathsf{Enc}}$ to further reduce the computational overheads. A concrete description of π is shown in $\mathsf{ZK}_{\mathsf{MPaillMPaill}}$, Appendix F.

Comparison. We compare the two instantiations discussed above from the experimental aspect. The results are displayed in Table 8, where we test different values of $|M|$, i.e., 256, 512 and 1024. Note that the size of the message space is $M + 1$, and our choices of $|M|$ is enough for actual use. The other conventions and experimental environments align with those described in Section 5, so we omit them here. Approximately, the instantiation from our range proof achieves a reduction in ciphertext by 23%, and also improves the computational overheads of encryption, compared with the one instantiated from the range proof introduced by [33].

REFERENCES

[1] 2019. ISO/IEC 18033-6:2019, IT Security techniques — Encryption algorithms — Part 6: Homomorphic encryption. (2019). https://www.iso.org/standard/67740.html

[2] Fabrice Boudot. 2000. Efficient proofs that a committed number lies in an interval. In *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 431–444.

[3] Emmanuel Bresson, Dario Catalano, and David Pointcheval. 2003. A simple public-key cryptosystem with a double trapdoor decryption mechanism and its applications. In *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 37–54.

[4] Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. 2018. Bulletproofs: Short proofs for confidential transactions and more. In *2018 IEEE symposium on security and privacy (SP)*. IEEE, 315–334.

[5] Ran Canetti, Yilei Chen, Justin Holmgren, Alex Lombardi, Guy N Rothblum, Ron D Rothblum, and Daniel Wichs. 2019. Fiat-Shamir: from practice to theory. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*. 1082–1090.

[6] Ran Canetti, Rosario Gennaro, Steven Goldfeder, Nikolaos Makriyannis, and Udi Peled. 2020. UC non-interactive, proactive, threshold ECDSA with identifiable aborts. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. 1769–1787.

[7] Guilhem Castagnos, Dario Catalano, Fabien Laguillaumie, Federico Savasta, and Ida Tucker. 2020. Bandwidth-efficient threshold EC-DSA. In *IACR International Conference on Public-Key Cryptography*. Springer, 266–296.

[8] Agnes Chan, Yair Frankel, and Yiannis Tsiounis. 1998. Easy come—easy go divisible cash. In *Advances in Cryptology—EUROCRYPT'98: International Conference on the Theory and Application of Cryptographic Techniques Espoo, Finland, May 31–June 4, 1998 Proceedings 17*. Springer, 561–575.

[9] Geoffroy Couteau, Michael Klooß, Huang Lin, and Michael Reichle. 2021. Efficient range proofs with transparent setup from bounded integer commitments. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 247–277.

[10] Geoffroy Couteau, Thomas Peters, and David Pointcheval. 2017. Removing the strong RSA assumption from arguments over the integers. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 321–350.

[11] Ronald Cramer, Ivan Damgård, Daniel Escudero, Peter Scholl, and Chaoping Xing. 2018. SPD$\mathbb{Z}_{2^k}$: Efficient MPC mod 2^k for Dishonest Majority. In *Advances in Cryptology–CRYPTO*.

[12] Ivan Damgård and Eiichiro Fujisaki. 2002. A statistically-hiding integer commitment scheme based on groups with hidden order. In *Advances in Cryptology—ASIACRYPT 2002: 8th International Conference on the Theory and Application of Cryptology and Information Security Queenstown, New Zealand, December 1–5, 2002 Proceedings 8*. Springer, 125–142.

[13] Ivan Damgård and Mads Jurik. 2001. A generalisation, a simplification and some applications of Paillier's probabilistic public-key system. In *Public Key Cryptography: 4th International Workshop on Practice and Theory in Public Key Cryptosystems, PKC 2001 Cheju Island, Korea, February 13–15, 2001 Proceedings 4*. Springer, 119–136.

[14] Ivan Damgård and Mads Jurik. 2002. Client/server tradeoffs for online elections. In *International Workshop on Public Key Cryptography*. Springer, 125–140.

[15] Ivan Damgård, Marcel Keller, Enrique Larraia, Valerio Pastro, Peter Scholl, and Nigel P Smart. 2013. Practical covertly secure MPC for dishonest majority–or: breaking the SPDZ limits. In *European Symposium on Research in Computer Security*. Springer, 1–18.

[16] Ivan Damgård, Valerio Pastro, Nigel Smart, and Sarah Zakarias. 2012. Multiparty computation from somewhat homomorphic encryption. In *Annual Cryptology Conference*. Springer, 643–662.

[17] Julien Devevey, Benoît Libert, Khoa Nguyen, Thomas Peters, and Moti Yung. 2021. Non-interactive CCA2-secure threshold cryptosystems: achieving adaptive security in the standard model without pairings. In *IACR International Conference on Public-Key Cryptography*. Springer, 659–690.

[18] Julien Devevey, Benoît Libert, and Thomas Peters. 2022. Rational Modular Encoding in the DCR Setting: Non-Interactive Range Proofs and Paillier-Based Naor-Yung in the Standard Model. In *IACR International Conference on Public-Key Cryptography*. Springer, 615–646.

[19] Jack Doerner, Yashvanth Kondi, Eysa Lee, et al. 2024. Threshold ECDSA in Three Rounds. In *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, 174–174.

[20] Jack Doerner, Yashvanth Kondi, Eysa Lee, and Abhi Shelat. 2018. Secure two-party threshold ECDSA from ECDSA assumptions. In *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 980–997.

[21] Jack Doerner, Yashvanth Kondi, Eysa Lee, and Abhi Shelat. 2019. Threshold ECDSA from ECDSA assumptions: The multiparty case. In *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 1051–1066.

[22] Sebastian Faust, Markulf Kohlweiss, Giorgia Azzurra Marson, and Daniele Venturi. 2012. On the non-malleability of the Fiat-Shamir transform. In *Progress in Cryptology-INDOCRYPT 2012: 13th International Conference on Cryptology in India, Kolkata, India, December 9-12, 2012. Proceedings 13*. Springer, 60–79.

[23] Amos Fiat and Adi Shamir. 1986. How to prove yourself: Practical solutions to identification and signature problems. In *Conference on the theory and application of cryptographic techniques*. Springer, 186–194.

[24] Pierre-Alain Fouque and David Pointcheval. 2001. Threshold cryptosystems secure against chosen-ciphertext attacks. In *Advances in Cryptology—ASIACRYPT 2001: 7th International Conference on the Theory and Application of Cryptology and Information Security Gold Coast, Australia, December 9–13, 2001 Proceedings 7*. Springer, 351–368.

[25] Eiichiro Fujisaki and Tatsuaki Okamoto. 1997. Statistical zero knowledge protocols to prove modular polynomial relations. In *Advances in Cryptology—CRYPTO'97: 17th Annual International Cryptology Conference Santa Barbara, California, USA August 17–21, 1997 Proceedings 17*. Springer, 16–30.

[26] Rosario Gennaro and Steven Goldfeder. 2018. Fast multiparty threshold ECDSA with fast trustless setup. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. 1179–1194.

[27] Borui Gong, Wang Fat Lau, Man Ho Au, Rupeng Yang, Haiyang Xue, and Lichun Li. 2024. Efficient Zero-Knowledge Arguments For Paillier Cryptosystem. In *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, 93–93.

[28] Jens Groth. 2005. Non-interactive zero-knowledge arguments for voting. In *Applied Cryptography and Network Security: Third International Conference, ACNS 2005, New York, NY, USA, June 7-10, 2005. Proceedings 3*. Springer, 467–482.

[29] Jens Groth. 2011. Efficient zero-knowledge arguments from two-tiered homomorphic commitments. In *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 431–448.

[30] Aggelos Kiayias and Moti Yung. 2004. The vector-ballot e-voting approach. In *International Conference on Financial Cryptography*. Springer, 72–89.

[31] Dimitris Kolonelos, Mary Maller, and Mikhail Volkhov. 2023. Zero-Knowledge Arguments for Subverted RSA Groups. In *IACR International Conference on Public-Key Cryptography*. Springer, 512–541.

[32] Yehuda Lindell. 2017. Fast secure two-party ECDSA signing. In *Advances in Cryptology–CRYPTO 2017: 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20–24, 2017, Proceedings, Part II 37*. Springer, 613–644.

[33] Yehuda Lindell and Ariel Nof. 2018. Fast secure multiparty ECDSA with practical distributed key generation and applications to cryptocurrency custody. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. 1837–1854.

[34] Helger Lipmaa. 2003. On diophantine complexity and statistical zero-knowledge arguments. In *Advances in Cryptology-ASIACRYPT 2003: 9th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, November 30–December 4, 2003. Proceedings 9*. Springer, 398–415.

[35] Helger Lipmaa, N Asokan, and Valtteri Niemi. 2003. Secure Vickrey auctions without threshold trust. In *Financial Cryptography: 6th International Conference, FC 2002 Southampton, Bermuda, March 2002 Revised Papers 6*. Springer, 87–101.

[36] Moni Naor and Moti Yung. 1990. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*. 427–437.

[37] NIST. 2024. Multi-Party Threshold Cryptography. (2024). https://csrc.nist.gov/Projects/threshold-cryptography

[38] Pascal Paillier. 1999. Public-key cryptosystems based on composite degree residuosity classes. In *International conference on the theory and applications of cryptographic techniques*. Springer, 223–238.

[39] Sri Aravinda Krishnan Thyagarajan, Adithya Bhat, Giulio Malavolta, Nico Döttling, Aniket Kate, and Dominique Schröder. 2020. Verifiable timed signatures made practical. In *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*. 1733–1750.

[40] Haiyang Xue, Man Ho Au, Mengling Liu, Kwan Yin Chan, Handong Cui, Xiang Xie, Tsz Hon Yuen, and Chengru Zhang. 2023. Efficient Multiplicative-to-Additive Function from Joye-Libert Cryptosystem and Its Application to Threshold ECDSA. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*. 2974–2988.

[41] Haiyang Xue, Man Ho Au, Xiang Xie, Tsz Hon Yuen, and Handong Cui. 2021. Efficient online-friendly two-party ECDSA signature. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. 558–573.
[42] Fan Zhang, Deepak Maram, Harjasleen Malvai, Steven Goldfeder, and Ari Juels. 2020. Deco: Liberating web data using decentralized oracles for tls. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. 1919–1938.

A RELATION BETWEEN DL ASSUMPTIONS

A commonly used DL assumption works over QR_N. Specifically, given $N = pq$ for safe primes p, q and random $g, h \leftarrow \mathsf{QR}_N$, it is hard to find an integer α such that $g^\alpha = h \bmod N$ for any PPT algorithm.

Theorem A.1. *The DL problem over* $2\mathsf{NR}_{N^2}$ *is not easier than that over* QR_N.

Proof. If there is an algorithm $\mathcal{A}$ that can solve the DL problem over $2\mathsf{NR}_{N^2}$, we can construct an algorithm $\mathcal{B}$ to solve that over QR_N. Given a DL problem instance (N, g, h) over QR_N, $\mathcal{B}$ can feed $(N, g^N \bmod N^2, h^N \bmod N^2)$ to $\mathcal{A}$. Note that $g^N \bmod N^2$ and $h^N \bmod N^2$ are both uniformly distributed over $2\mathsf{NR}_{N^2}$ from Fact 1. When the solver outputs an answer denoted as α such that $(g^N)^\alpha = h^N \bmod N^2$, $\mathcal{B}$ can directly use α as a solution to the DL problem over QR_N, also from Fact 1. □

B PAILLIER ROOT ASSUMPTION

Definition B.1 (Strong RSA Assumption over Quadratic Residues). Given $N = pq$ for safe primes p, q and a random $T \leftarrow \mathsf{QR}_N$, it is hard to find an e-th root a modulo N, namely, $a^e = T \bmod N$, for any PPT algorithm with an exponent $e > 1$ of its choice.

Theorem B.2. *The strong RSA assumption over* QR_N *holds under the strong RSA assumption.*

Proof. Let $\mathsf{Adv}^{\mathsf{sRSA}}_{\mathcal{A}}$ (resp., $\mathsf{Adv}^{\mathsf{sQR}}_{\mathcal{A}}$) be the advantage of a PPT algorithm $\mathcal{A}$ solving strong RSA problem (resp., strong RSA problem over QR_N). We have that $\mathsf{Adv}^{\mathsf{sQR}}_{\mathcal{A}} \leq 4\mathsf{Adv}^{\mathsf{sRSA}}_{\mathcal{A}}$, since $\mathsf{QR}_N \subset \mathbb{Z}^*_N$ and $4|\mathsf{QR}_N| = |\mathbb{Z}^*_N|$. □

Theorem B.3. *The Paillier root assumption holds under the strong RSA assumption over* QR_N.

Proof. If there is an algorithm $\mathcal{A}$ that can solve the Paillier root problem, we can construct an algorithm $\mathcal{B}$ to solve the strong RSA problem over QR_N. Given a strong RSA problem instance (N, x) for a uniform variable $x \in \mathsf{QR}_N$, $\mathcal{B}$ computes $T = x^N \bmod N^2$. Since T is uniformly distributed over $2\mathsf{NR}_{N^2}$, $\mathcal{B}$ can directly invoke $\mathcal{A}$ with input (N, T). Suppose that $\mathcal{A}$ outputs (a, e) satisfying $a^e = T \bmod N^2$, $e > 1$ and $\gcd(e, N) = 1$. By extended Euclidean algorithm $\mathcal{B}$ can find $\gamma, \delta \in \mathbb{Z}$ such that $\gamma e + \delta N = 1$, and so

$$x = x^{\gamma e + \delta N} = x^{\gamma e} T^\delta = (x^\gamma a^\delta)^e \bmod N^2,$$

which also yields

$$x = (x^\gamma a^\delta)^e \bmod N.$$

Therefore, we get a solution $(x^\gamma a^\delta, e)$ to the strong RSA problem over QR_N. □

C PEDERSEN COMMITMENT

A widely used integer commitment is Pedersen commitment [25] (or named as RSA commitment to be distinguished from the Pedersen commitment over prime order groups). Pedersen commitment PedCom is defined by a tuple of algorithms (Setup, Commit, Verify).

- $\mathsf{Setup}(1^\lambda)$. Randomly generate safe primes $p = 2p' + 1$ and $q = 2q' + 1$, then compute the RSA modulus $N = pq$. Choose $a \leftarrow \mathbb{Z}^*_N$ and $\alpha \leftarrow \mathbb{Z}_N$, then compute $g = a^2 \bmod N$ and $y = g^\alpha \bmod N$. Return the public parameter as $pp = (N, g, y)$.
- $\mathsf{Commit}(pp, m)$. On input $pp = (N, g, y)$ and $m \in \mathbb{Z}$, sample $r \leftarrow \mathbb{Z}_N$ and compute $c = y^m g^r \bmod N$. Return c as the commitment and (m, r) as the corresponding opening.
- $\mathsf{Verify}(pp, c, m, r)$. For $pp = (N, g, y)$, if $c = \pm y^m g^r \bmod N$ output 1, otherwise output 0.

PedCom is an integer commitment scheme with perfect hiding and computational binding under the factoring and DL (over QR_N) assumptions.

D PROOFS OF PUBLIC PARAMETERS

Let $N = pq$ be an RSA modulus. All of the zero-knowledge proof protocols in this section have soundness error $1/2$. We emphasize that we can repeat each one t times to achieve a soundness error of 2^{-t}, and use Fiat-Shamir transformation [23] to get a non-interactive version.

D.1 Proof of Quadratic Residue

We consider the following relation:

$$\mathcal{R}_{\mathsf{QR}_N} = \{(N, g; a \in \mathbb{Z}^*_N) : g = a^2 \bmod N\}.$$

The protocol $\mathsf{ZK}_{\mathsf{QR}_N}$ between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$ for this relation is as follows.

- $\mathcal{P}$ selects $b \leftarrow \mathbb{Z}^*_N$, computes $d = b^2 \bmod N$, and then sends d to $\mathcal{V}$.
- $\mathcal{V}$ picks $e \leftarrow \{0, 1\}$ and sends it to $\mathcal{P}$.
- $\mathcal{P}$ computes $z = a^e b \bmod N$ and sends it to $\mathcal{V}$.
- $\mathcal{V}$ accepts the proof if and only if $z^2 = g^e d \bmod N$.

D.2 Proof of *2N*-th Residue

We consider the following relation:

$$\mathcal{R}_{2\mathsf{NR}_{N^2}} = \{(N, g; a \in \mathbb{Z}^*_N) : g = a^{2N} \bmod N^2\}.$$

The protocol $\mathsf{ZK}_{2\mathsf{NR}_{N^2}}$ between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$ for this relation is as follows.

- $\mathcal{P}$ selects $b \leftarrow \mathbb{Z}^*_N$, computes $d = b^{2N} \bmod N^2$, and then sends d to $\mathcal{V}$.
- $\mathcal{V}$ picks $e \leftarrow \{0, 1\}$ and sends it to $\mathcal{P}$.
- $\mathcal{P}$ computes $z = a^e b \bmod N$ and sends it to $\mathcal{V}$.
- $\mathcal{V}$ accepts the proof if and only if $z^{2N} = g^e d \bmod N^2$.

D.3 Proofs of Discrete Logarithms

Modulo N. First we consider the discrete logarithm relation in $\mathbb{Z}_N^*$:

$$\mathcal{R}_{\mathsf{DL}_N} = \{(N, g, h; \alpha) : g^\alpha = h \bmod N\}.$$

The protocol $\mathsf{ZK}_{\mathsf{DL}_N}$ between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$ for this relation is described as follows, where s refers to the statistical parameter.

- $\mathcal{P}$ picks $\beta \leftarrow [0, 2^s N]$, calculates $d = g^\beta \bmod N$, and sends d to $\mathcal{V}$.
- $\mathcal{V}$ selects $e \leftarrow \{0, 1\}$ and sends it to $\mathcal{P}$.
- $\mathcal{P}$ computes $z = e\alpha + \beta$ in $\mathbb{Z}$ and sends it to $\mathcal{V}$.
- $\mathcal{V}$ accepts the proof if and only if $g^z = h^e d \bmod N$.

Modulo N^2. Then we consider the discrete logarithm relation in $\mathbb{Z}_{N^2}^*$:

$$\mathcal{R}_{\mathsf{DL}_{N^2}} = \{(N, g, h; \alpha) : g^\alpha = h \bmod N^2\}.$$

The protocol $\mathsf{ZK}_{\mathsf{DL}_{N^2}}$ between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$ for this relation is described as follows, where s refers to the statistical parameter.

- $\mathcal{P}$ picks $\beta \leftarrow [0, 2^s N]$, computes $d = g^\beta \bmod N^2$, and sends d to $\mathcal{V}$.
- $\mathcal{V}$ selects $e \leftarrow \{0, 1\}$ and sends it to $\mathcal{P}$.
- $\mathcal{P}$ computes $z = e\alpha + \beta$ in $\mathbb{Z}$ and sends it to $\mathcal{V}$.
- $\mathcal{V}$ accepts the proof if and only if $g^z = h^e d \bmod N^2$.

E RANGE PROOF WHERE PROVER CONTROLS PAILLIER KEYS

In this section, we show how to conduct range proofs for plaintexts in MPaill scheme, when the prover controls the related keys (specifically, knows the factorization of the RSA modulus or the related discrete logarithms). In this case, an additional integer commitment PedCom whose public parameter is not controlled by the prover is required (e.g., generated by the verifier). Generally, to prove the range of some plaintext, the prover will commit to a same value via PedCom, then provide an equality proof (he/she indeed commits to a same value) as well as a range proof (conducted over PedCom).

Let $pk = (N, g, y)$ and $pp = (\tilde{N}, \tilde{g}, \tilde{y})$ be the public key of MPaill and public parameter of PedCom, respectively. We consider the scenario where a prover would like to prove the knowledge of the same message m under an encryption generated from MPaill.$\widetilde{\mathsf{Enc}}$, as well as a commitment generated from PedCom.Commit. Besides, m should be in a certain range, say, $[0, B]$.

In this case, the prover is allowed to know the order of $\mathsf{2NR}_{N^2}$, and the DL relations between g and $y/(1+N) \bmod N^2$, while not allowed to know the order of $\mathsf{QR}_{\tilde{N}}$, as well as the DL relations between $\tilde{g}$ and $\tilde{y}$.

Concretely, $\mathsf{ZK}_{\mathsf{MPaillPed}}$ is a Σ-protocol between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$ for the relation

$$\mathcal{R}_{\mathsf{MPaillPed}} = \{(C, \tilde{c}; m, \tilde{r}) : \exists r \text{ s.t. } C = \pm(1+N)^m g^r \bmod N^2, \\ \tilde{c} = \pm\tilde{y}^m \tilde{g}^{\tilde{r}} \bmod \tilde{N}, m \in [0, B]\}.$$

The protocol works as follows, where s and t represent the statistical and soundness parameters, respectively.

- $\mathcal{P}$ samples $u \leftarrow [0, 2^{s+t}B]$, $v \leftarrow [0, 2^{s+t}N]$ and $\tilde{v} \leftarrow [0, 2^{s+t}\tilde{N}]$, computes $D = (1+N)^u g^v \bmod N^2$ as well as $\tilde{d} = \tilde{y}^u \tilde{g}^{\tilde{v}} \bmod \tilde{N}$, and forwards $(D, \tilde{d})$ to $\mathcal{V}$.
- $\mathcal{V}$ picks $e \leftarrow [0, 2^t]$ and sends it to $\mathcal{P}$.
- $\mathcal{P}$ computes $z_m = em + u$, $z_r = er + v$ and $z_{\tilde{r}} = e\tilde{r} + \tilde{v}$, all over integers, then sends them to $\mathcal{V}$.
- $\mathcal{V}$ accepts the proof when all of the following conditions hold:
  (1) $(1+N)^{z_m} g^{z_r} = C^e D \bmod N^2$.
  (2) $\tilde{y}^{z_m} \tilde{g}^{z_{\tilde{r}}} = \tilde{c}^e \tilde{d} \bmod \tilde{N}$.
  (3) $z_m \in [0, 2^{s+t}B]$.

The completeness and HVZK are obvious, and thus we only demonstrate PoK here. Assume two accepting conversations denoted by $((D, \tilde{d}), e, (z_m, z_r, z_{\tilde{r}}))$ and $((D, \tilde{d}), e', (z'_m, z'_r, z'_{\tilde{r}}))$. Write $\Delta e = e - e'$, $\Delta z_m = z_m - z'_m$, $\Delta z_r = z_r - z'_r$, and $\Delta z_{\tilde{r}} = z_{\tilde{r}} - z'_{\tilde{r}}$. Without loss of generality, assume that $\Delta e > 0$.

From the proof-of-knowledge analysis for Pedersen commitment [12, Section 5.1], we have that under the strong RSA assumption, $\Delta e \mid \Delta z_m$ and $\Delta e \mid \Delta z_{\tilde{r}}$. Besides, we can extract an opening $(\Delta z_m/\Delta e, \Delta z_{\tilde{r}}/\Delta e)$ for Pedersen commitment, such that $\tilde{c} = \pm\tilde{y}^{\Delta z_m/\Delta e} \tilde{g}^{\Delta z_{\tilde{r}}/\Delta e} \bmod \tilde{N}$ and $\Delta z_m/\Delta e \in [-2^{s+t}B, 2^{s+t}B]$.

From the first condition of verification phase, we also have $(1+N)^{\Delta z_m} g^{\Delta z_r} = C^{\Delta e} \bmod N^2$. Since $\gcd(\Delta e, N) = 1$, by extended Euclidean algorithm we can find $\gamma, \delta \in \mathbb{Z}$ such that $\gamma\Delta e + \delta N = 1$. Thus, we have

$$\begin{aligned} C = C^{\gamma\Delta e + \delta N} &= (1+N)^{\gamma\Delta z_m} g^{\gamma\Delta z_r} C^{\delta N} \bmod N^2 \\ &= \pm(1+N)^{\gamma\Delta z_m} g^r \bmod N^2, \end{aligned}$$

for some randomness r, and the sign depends on whether C is a quadratic residue modulo N^2.

Finally, it is easy to check that $\gamma\Delta z_m = \Delta z_m/\Delta e \bmod N$ (γ is an inverse of Δe modulo N from $\gamma\Delta e + \delta N = 1$) and we have extracted an opening $(\Delta z_m/\Delta e, \Delta z_{\tilde{r}}/\Delta e)$ such that there is some randomness r, satisfying $C = \pm(1+N)^{\Delta z_m/\Delta e} g^r \bmod N^2$, $\tilde{c} = \pm\tilde{y}^{\Delta z_m/\Delta e} \tilde{g}^{\Delta z_{\tilde{r}}/\Delta e} \bmod \tilde{N}$ and $\Delta z_m/\Delta e \in [-2^{s+t}B, 2^{s+t}B]$.

Performance Evaluation. We conduct both theoretical and experimental analyses of the above protocol (settings consistent with those in Section 5), and the results are presented in Table 9. Note that the generation of $\tilde{c}$ should be included in the prover's computational overhead, and $\tilde{c}$ also needs to be included in the communication cost. From the results, we can observe that the range proof in this section incurs additional overhead compared to our direct range proof, while the cost is close to that of the LN range proof [33].

F BOUNDED EQUALITY PROOF

Let $pk_1 = (N_1, g_1, y_1)$ and $pk_2 = (N_2, g_2, y_2)$ be two public keys of MPaill. We consider the scenario where a prover would like to prove the knowledge of (m, r_1, r_2) such that $C_1 = \mathsf{Enc}(pk_1, m; r_1)$, $C_2 = \widetilde{\mathsf{Enc}}(pk_2, m; r_2)$, and m is in the range $[0, B]$.

In the situation we consider, the prover is not allowed to know the orders of $\mathsf{2NR}_{N_1^2}$ and $\mathsf{2NR}_{N_2^2}$, and the discrete logarithm relations between g_1 and $y_1/(1+N_1) \bmod N_1^2$.

Concretely, $\mathsf{ZK}_{\mathsf{MPaillMPaill}}$ is a Σ-protocol between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$ for the relation

$$\mathcal{R}_{\mathsf{MPaillMPaill}} = \{(C_1, C_2; m, r_1, r_2) : C_1 = \pm y_1^m g_1^{r_1} \bmod N_1^2, \\ C_2 = \pm(1+N_2)^m g_2^{r_2} \bmod N_2^2, m \in [0, B]\}.$$

Table 9: Performance Evaluation for the Range Proof when Prover Controls Paillier Keys

	Theoretical	Experimental	
	(E or bit)	$\|B\|$ (bit)	Result (ms or KiB)
Prove	$4.5+\frac{4.5s+4.5t+2\|B\|}{\|N\|}$	256	33.5
		512	34.5
		1024	36.5
Verify	$3.5+\frac{4.5s+8t+\|B\|}{\|N\|}$	256	27.5
		512	28
		1024	29
Communication	$3\|N\|+3s+4t+\|B\|$	256	1.25
		512	1.28
		1024	1.34

The protocol works as follows, where s and t represent the statistical and soundness parameters, respectively.

- $\mathcal{P}$ selects $u \leftarrow [0, 2^{s+t}B]$, $v_1 \leftarrow [0, 2^{s+t}N_1]$ and $v_2 \leftarrow [0, 2^{s+t}N_2]$, then computes $d_1 = y_1^u g_1^{v_1} \bmod N_1^2$ and $d_2 = (1+N_2)^u g_2^{v_2} \bmod N_2^2$. Next, $\mathcal{P}$ sends (d_1, d_2) to $\mathcal{V}$.
- $\mathcal{V}$ chooses $e \leftarrow [0, 2^t]$ and sends it to $\mathcal{P}$.
- $\mathcal{P}$ computes $z_m = em + u$, $z_1 = er_1 + v_1$ and $z_2 = er_2 + v_2$, all in $\mathbb{Z}$, then sends them to $\mathcal{V}$.
- $\mathcal{V}$ accepts the proof when all of the following conditions hold:
  (1) $y_1^{z_m} g_1^{z_1} = C_1^e d_1 \bmod N_1^2$.
  (2) $(1+N_2)^{z_m} g_2^{z_2} = C_2^e d_2 \bmod N_2^2$.
  (3) $z_m \in [0, 2^{s+t}B]$.

The completeness and HVZK are obvious, and thus we only demonstrate PoK here. Suppose that we have two accepting conversations $((d_1, d_2), e, (z_m, z_1, z_2))$ and $((d_1, d_2), e', (z_m', z_1', z_2'))$. Write $\Delta e = e - e'$, $\Delta z_m = z_m - z_m'$, $\Delta z_1 = z_1 - z_1'$ and $\Delta z_2 = z_2 - z_2'$. Without loss of generality, assume that $\Delta e > 0$.

From the proof-of-knowledge analysis for PaillCom of length 1 (see Section 4.2), we have that under the Paillier root assumption, $\Delta e \mid \Delta z_m$ and $\Delta e \mid \Delta z_1$ with overwhelming probability. Moreover, we can extract an opening $(\Delta z_m/\Delta e, \Delta z_1/\Delta e)$ for PaillCom, s.t. $C_1 = \pm y_1^{\Delta z_m/\Delta e} g_1^{\Delta z_1/\Delta e} \bmod N_1^2$, and $\Delta z_m/\Delta e \in [-2^{s+t}B, 2^{s+t}B]$.

From the second condition of verification phase, we also have

$$(1+N_2)^{\Delta z_m} g_2^{\Delta z_2} = C_2^{\Delta e} \bmod N_2^2. \tag{12}$$

If $\Delta e \mid \Delta z_2$:

- If Δe is odd, it is co-prime to $\varphi(N_2^2)$, and from Equation (12) we get
$$(1+N_2)^{\Delta z_m/\Delta e} g_2^{\Delta z_2/\Delta e} = C_2 \bmod N_2^2,$$
which indicates that $(\Delta z_m/\Delta e, \Delta z_1/\Delta e, \Delta z_2/\Delta e)$ is a valid opening.
- If Δe is even, suppose that $\Delta e = 2^\rho \Delta e'$ for an odd $\Delta e'$ and $\rho \geq 1$. Let $Z = C_2^{-1}(1+N_2)^{\Delta z_m/\Delta e} g_2^{\Delta z_2/\Delta e} \bmod N_2^2$. Also from Equation (12) we have $Z^{2^\rho} = 1 \bmod N_2^2$. Similar to the proof of Theorem 4.1, $Z = \pm 1 \bmod N_2^2$, which indicates that $(\Delta z_m/\Delta e, \Delta z_1/\Delta e, \Delta z_2/\Delta e)$ is a valid opening.

As for the case of $\Delta e \nmid \Delta z_2$, we show by contradiction that it only happens with negligible probability. If not, we construct a simulator $\mathcal{B}$ to solve the Paillier root problem with non-negligible probability based on the two accepting conversations. Denote the inverse of Δe modulo N_2 as Δe^{-1} (note that $\gcd(\Delta e, N_2) = 1$), and define $\tilde{C_2}$ as

$$\tilde{C_2} = \frac{C_2}{(1+N_2)^{\Delta z_m \cdot \Delta e^{-1}}} \bmod N_2^2.$$

Then we have that $C_2 = \tilde{C_2} \cdot (1+N_2)^{\Delta z_m \cdot \Delta e^{-1}} \bmod N_2^2$, and bringing it into Equation (12) we have

$$g_2^{\Delta z_2} = \tilde{C_2}^{\Delta e} \bmod N_2^2.$$

Let $\eta = \gcd(\Delta e, \Delta z_2)$, then by extended Euclidean algorithm we can find $\gamma, \delta \in \mathbb{Z}$ such that

$$\gamma \Delta e + \delta \Delta z_2 = \eta.$$

Thus, we get

$$g_2^\eta = g_2^{\gamma \Delta e + \delta \Delta z_2} = \left(g_2^\gamma \tilde{C_2}^\delta\right)^{\Delta e} \bmod N_2^2. \tag{13}$$

Given a Paillier root problem instance (N, T) where $T \leftarrow 2\mathsf{NR}_{N^2}$, $\mathcal{B}$ sets $N_2 = N$ and $g_2 = T$. For pk_1 and y_2, $\mathcal{B}$ can generates them according to the real MPaill scheme.

- If η is odd, it is co-prime to $\varphi(N_2^2)$, and from Equation (13) we have
$$\left(g_2^\gamma \tilde{C_2}^\delta\right)^{\Delta e/\eta} = g_2 \bmod N_2^2 (= T \bmod N^2).$$
Also because $\gcd(\Delta e/\eta, N) = 1$ and $\Delta e/\eta > 1$ (otherwise $\Delta e \mid \Delta z_2$), we get a solution $(g_2^\gamma \tilde{C_2}^\delta, \Delta e/\eta)$ to the Paillier root problem.
- When $\eta = 2^\rho \eta'$ for an odd η' and $\rho \geq 1$, we can define $Z = g_2^{-1}\left(g_2^\gamma \tilde{C_2}^\delta\right)^{\Delta e/\eta} \bmod N_2^2$. From Equation (13),
$$Z^{2^\rho} = 1 \bmod N_2^2.$$
Similar to the proof of Theorem 4.1, $Z = \pm 1 \bmod N_2^2$. If $Z = 1 \bmod N_2^2$, i.e., $Z = 1 \bmod N^2$, it is easy to see that $(g_2^\gamma \tilde{C_2}^\delta, \Delta e/\eta)$ is a solution to the Paillier root problem. If $Z = -1 \bmod N_2^2$, we have
$$-\left(g_2^\gamma \tilde{C_2}^\delta\right)^{\Delta e/\eta} = g_2 \bmod N_2^2.$$
In this case, $\Delta e/\eta$ must be odd. If not the case, since $-1 \notin \mathrm{QR}_{N_2^2}$, $\left(g_2^\gamma \tilde{C_2}^\delta\right)^{\Delta e/\eta} \in \mathrm{QR}_{N_2^2}$ and $g_2 \in \mathrm{QR}_{N_2^2}$, the above equation will not hold. Thus,
$$\left(-g_2^\gamma \tilde{C_2}^\delta\right)^{\Delta e/\eta} = g_2 \bmod N_2^2 (= T \bmod N^2),$$
and $(-g_2^\gamma \tilde{C_2}^\delta, \Delta e/\eta)$ also gives a solution to the Paillier root problem, since $\gcd(\Delta e/\eta, N) = 1$ and $\Delta e/\eta > 1$.

G PROOF OF THEOREM 3.1

Correctness. It is obvious that this property holds.

Hiding. If there is an adversary $\mathcal{A}$ that can break this property with non-negligible advantage ε, we can construct a simulator $\mathcal{B}$ to solve the DCR assumption also with non-negligible advantage.

Given a DCR problem instance (N, T), $\mathcal{B}$ generates g and y-elements according to the real PaillCom scheme, then computes and sends to $\mathcal{A}$ the public parameter $pp = (N, g, y_1, \cdots, y_l)$.

On receiving two distinct messages $\mathbf{m}_0, \mathbf{m}_1 \in \mathbb{Z}^l$ from $\mathcal{A}$, $\mathcal{B}$ picks $b \leftarrow \{0, 1\}$, then for $\mathbf{m}_b = (m_1, \cdots, m_l)$, simulates and sends to $\mathcal{A}$ a commitment

$$c = T^2 \prod_{i=1}^{l} y_i^{m_i} \bmod N^2.$$

After $\mathcal{A}$ outputs b' as the guess of b, if $b' = b$, $\mathcal{B}$ outputs 1 to indicate that T is an N-th residue modulo N^2. If $b' \neq b$, $\mathcal{B}$ outputs 0 to indicate that T is a uniform element of $\mathbb{Z}^*_{N^2}$. Besides, we have the following observations.

- When T is an N-th residue modulo N^2, denoted as event $\mathsf{E_N}$, it is obvious that c is statistically close to a real commitment of $\mathbf{m}_b$.
- When T is a uniform element of $\mathbb{Z}^*_{N^2}$, denoted as event $\mathsf{E_R}$, $T^2 \bmod N^2$ is uniformly distributed over QR_{N^2}. Additionally, each $y_i \in \mathsf{QR}_{N^2}$. Thus, c is also uniformly distributed over QR_{N^2}, which indicates that c contains no information about $\mathbf{m}_b$ in this case.

Thus, the advantage of $\mathcal{B}$ solving the DCR problem is

$$\begin{aligned}\mathrm{Adv}_{\mathcal{B}} &= \Pr[\mathsf{E_N}] \cdot \Pr[b' = b \mid \mathsf{E_N}] + \Pr[\mathsf{E_R}] \cdot \Pr[b' \neq b \mid \mathsf{E_R}] - \frac{1}{2} \\ &= \frac{1}{2}\left(\varepsilon + \frac{1}{2}\right) + \frac{1}{2} \times \frac{1}{2} - \frac{1}{2} = \frac{\varepsilon}{2},\end{aligned}$$

which is non-negligible.

Binding. From the generation of $y_1, \cdots, y_l$ and verification algorithm, we have that $c = \pm g^{r+\sum_{i=1}^{l} \alpha_i m_i}(1+N)^{\sum_{i=1}^{l} \beta_i m_i} \bmod N^2$. Suppose that there is a committer who can output two different openings $(\mathbf{m} = (m_1, \cdots, m_l), r)$ and $(\mathbf{m}' = (m'_1, \cdots, m'_l), r')$ with non-negligible probability ε such that $\mathbf{m} \neq \mathbf{m}'$, we can get that $g^{r+\sum_{i=1}^{l} \alpha_i m_i}(1+N)^{\sum_{i=1}^{l} \beta_i m_i} = \pm g^{r'+\sum_{i=1}^{l} \alpha_i m'_i}(1+N)^{\sum_{i=1}^{l} \beta_i m'_i} \bmod N^2$. Write $\Delta r = r - r'$ and $\Delta m_i = m_i - m'_i$ for $i \in [1, l]$, then we have that $g^{\Delta r+\sum_{i=1}^{l} \alpha_i \Delta m_i} = \pm(1+N)^{-\sum_{i=1}^{l} \beta_i \Delta m_i} \bmod N^2$. Note that in $\mathbb{Z}^*_{N^2}$, the order of left part is a factor of $p'q'$, the order of right part is a factor of $2N$, and $\gcd(p'q', 2N) = 1$, so necessarily the order of both parts is 1, then we get

$$g^{\Delta r+\sum_{i=1}^{l} \alpha_i \Delta m_i} = 1 \bmod N^2. \tag{14}$$

(1) Case 1: $\Delta r + \sum_{i=1}^{l} \alpha_i \Delta m_i = 0$. We construct a simulator $\mathcal{B}$ to attack the DL assumption. Given a DL problem instance (N, g, h) over $2\mathsf{NR}_{N^2}$, $\mathcal{B}$ guesses an index $i^* \leftarrow [1, l]$, then computes $y_{i^*} = h(1+N)^{\beta_{i^*}} \bmod N^2$ (this implicitly sets $h = g^{\alpha_{i^*}} \bmod N^2$). Next, $\mathcal{B}$ generates the other y-elements, namely, y_i for $i \in [1, l] - \{i^*\}$, according to the real PaillCom scheme. The public parameter is set as $(N, g, y_1, \cdots, y_l)$. Note that the simulated PaillCom scheme is statistically close to a real scheme. After the committer outputs two different openings mentioned above, if $m_{i^*} = m'_{i^*}$, $\mathcal{B}$ aborts. From our constraint that $\mathbf{m} \neq \mathbf{m}'$, we know that $m_j \neq m'_j$ holds for at least one index $j \in [1, l]$, and $\mathcal{B}$ correctly guesses such an index with probability at least $1/l$. If $\mathcal{B}$ does not abort, which also means that $\Delta m_{i^*} = m_{i^*} - m'_{i^*} \neq 0$, then

$$\alpha_{i^*} = -\frac{\Delta r + \sum_{i \in [1,l]-\{i^*\}} \alpha_i \Delta m_i}{\Delta m_{i^*}}$$

is a solution to the DL problem. Besides, the advantage of $\mathcal{B}$ is ε/l, which is non-negligible.

(2) Case 2: $\Delta r + \sum_{i=1}^{l} \alpha_i \Delta m_i \neq 0$. We construct a simulator $\mathcal{B}$ to attack the factoring assumption. Given a factoring problem instance N, $\mathcal{B}$ generates $g, \{y_i\}_{i \in [1,l]}$ according to the real PaillCom scheme and sends the public parameter $pp = (N, g, y_1, \cdots, y_l)$ to the committer. Note that the values of all α-elements are picked by $\mathcal{B}$ and thus known to $\mathcal{B}$ in this case. Since g is a generator of $2\mathsf{NR}_{N^2}$ (whose order is $p'q'$) with overwhelming probability, we have $p'q' \mid (\Delta r + \sum_{i=1}^{l} \alpha_i \Delta m_i)$ from Equation (14). Thus, $\Delta r + \sum_{i=1}^{l} \alpha_i \Delta m_i$ is a non-zero multiple of $p'q'$, which can be used to factor N from Fact 6.

So far we have shown that PaillCom is a commitment scheme over vectors of integers with correctness, hiding and binding.

H PROOF OF THEOREM 3.2

The following proof is correct when we set α to be exactly 0. Thus, we will not describe the case when ciphertexts are generated through $\widetilde{\mathsf{Enc}}$.

Correctness. According to the scheme,

$$\begin{aligned}C^\lambda &= (y^m g^r)^\lambda \bmod N^2 \\ &= (g^{\alpha m}(1+N)^m g^r)^\lambda \bmod N^2 \\ &= (g^{p'q'})^{2(\alpha m + r)}(1+N)^{\lambda m} \bmod N^2 \\ &= (1+N)^{\lambda m} \bmod N^2 \\ &= 1 + (\lambda m \bmod N) \cdot N \bmod N^2.\end{aligned}$$

Since $1 + (\lambda m \bmod N) \cdot N$ is smaller than N^2, we have

$$\begin{aligned}&\frac{C^\lambda \bmod N^2 - 1}{N} \cdot \lambda^{-1} \\ &= \frac{1 + (\lambda m \bmod N) \cdot N - 1}{N} \cdot \lambda^{-1} \bmod N \\ &= (\lambda m \bmod N) \cdot \lambda^{-1} \bmod N \\ &= m \bmod N.\end{aligned}$$

IND-CPA security. The proof for this property can be seen as a special case ($l = \beta_1 = 1$) of that for the hiding property of PaillCom (detailed in Appendix G). Thus, we just omit the detailed proof here.

Remark. We would like to remark that, in the very original Paillier encryption scheme [38], the encryption of m is given by $C = y^m r^N \bmod N^2$, where y is a random element from $\mathbb{Z}^*_{N^2}$ whose order is a non-zero multiple of N, or equivalently, $y = h(1+N)^j \bmod N^2$ for a random N-th residue h, and a j relatively prime to N. This

was later simplified by setting $y = 1 + N$, as described in [13], resulting in a widely used Paillier ciphertext structure. Our scheme is structurally more similar to the most original encryption, but there are still differences which are intended to make the encryption consistent with our commitment. Note that our scheme and the simplified one are compatible with a same decryption process. For the most original one, an extra step is needed to eliminate j.

I PROOF OF THEOREM 6.1

We construct a simulator $\mathcal{S}$ to simulate the view of adversary.

In Setup phase, $\mathcal{S}$ samples a Pedersen public parameter and a modified Paillier public key from the corresponding spaces, respectively. Then it generates the zero-knowledge proofs for the well-formedness of them via related zero-knowledge simulators.

In Trans phase, $\mathcal{A}$ can corrupt P_1 or P_2, and $\mathcal{S}$ must simulate the view of corrupted party in each of the cases.

- $\mathcal{S}$ simulates P_1, when P_2 is corrupted by the adversary. $\mathcal{S}$ receives the tuple (C_b, π_b) that $\mathcal{A}$ instructs P_2 to send with sid. If π_b is accepted, $\mathcal{S}$ can extract b via the knowledge extractor of $\mathsf{ZK}_{\mathsf{MPaillPed}}$ from Appendix E. Then $\mathcal{S}$ queries $\mathcal{F}_{\mathsf{MtA}}$ with (sid, b) and receives P_2's output (sid, B). Next, $\mathcal{S}$ picks $\xi \leftarrow [0, 2^{2s+t+2}q]$ and computes C_B as the encryption of $2B + \xi q$. Additionally, π_B can be simulated by the zero-knowledge simulator of $\mathsf{ZK}_{\mathsf{PaillCom}}$. Note that the distribution of $2a(b + 2^{s+t}q) + 2A'$ is statistically close to that of $2B + \xi q$. Moreover, the distribution of the randomness of C_B in real scheme is statistically close to $\mathbb{Z}_{p'q'}$, which also holds in the simulation. Thus, the simulation is indistinguishable from a real execution from the point of view of $\mathcal{A}$.
- $\mathcal{S}$ simulates P_2, when P_1 is corrupted by the adversary. $\mathcal{S}$ generates C_b and the related Pedersen commitment in π_b with a same random value. It then computes the other parts of π_b via the zero-knowledge simulator of $\mathsf{ZK}_{\mathsf{MPaillPed}}$ from Appendix E. The IND-CPA security of MPaill and hiding property of PedCom guarantee that the simulated interaction is indistinguishable from a real one from the point of view of $\mathcal{A}$.

So far, we have shown that the proposed protocol securely computes $\mathcal{F}_{\mathsf{MtA}}$.