Beyond the 1/2 Bound: On the Theory and Practice of Biprimality Tests

ChihYun Chuang[1], IHung Hsu[1], and TingFang Lee[2]

[1] AMIS, Taipei, Taiwan
{chihyun,glen}@maicoin.com
[2] Division of Biostatistics, NYU Langone Health
Ting-Fang.Lee@nyulangone.org

Abstract. The Boneh-Franklin (BF) biprimality test, a cornerstone of distributed RSA key generation, has a universally accepted worst-case soundness error of 1/2. We show that this two-decade-old bound is not tight and present a collection of results that refine and generalize this fundamental test. Our contributions are threefold: **(1) A Tight Soundness Bound for the BF Test:** Our primary contribution is a rigorous proof that the worst-case acceptance probability of the BF test for a non-RSA modulus is at most 1/4, not the long-accepted 1/2. By constructing cases that meet this bound, we establish that it is tight. This fundamental result allows existing protocols to halve the required iterations, thereby improving the efficiency of verifying valid moduli while maintaining the same security level. **(2) A Generalized Test for Universal Applicability and a Nuanced Verdict:** We introduce a versatile Lucas-sequence-based test that resolves the long-standing limitation of the BF test to Blum integers. While we prove its soundness error is theoretically and empirically superior in the vast majority of cases, our analysis, based on a performance model parameterized by simulation, indicates a trade-off. For the specific task of generating Blum integers, the exceptional local computation speed of the BF test suggests it maintains a practical advantage in latency-sensitive environments. This nuanced finding underscores the interplay between theoretical soundness error and real-world performance. **(3) New Protocols for an Expanded Design Space:** We construct new distributed verification protocols that realize our foundational insights. Our Lucas-based protocol provides the first efficient, provably secure solution for generating all standard RSA moduli in the semi-honest model, alongside a maliciously secure variant for Blum integers. This fills a gap for applications requiring generality and, combined with our comparative analysis, provides a clearer and more complete toolkit for protocol designers. Overall, our work refines the theoretical bounds of biprimality testing and provides a more efficient and versatile foundation for distributed RSA key generation.

Table of Contents

Beyond the 1/2 Bound: On the Theory and Practice of Biprimality Tests 1
ChihYun Chuang, IHung Hsu, and TingFang Lee

1 Introduction . . . 3
- 1.1 Our Contributions . . . 4
- 1.2 Technical Overview . . . 6

2 Preliminaries . . . 7
- 2.1 Boneh-Franklin Biprimality Test. . . . 8

3 Soundness Analysis of Biprimality Tests . . . 8
- 3.1 Refined Analysis of the Boneh-Franklin Test . . . 9
- 3.2 Soundness Analysis of the Lucas Test . . . 12

4 The Proposed Lucas Biprimality Protocols . . . 24
- 4.1 MPC Building Blocks and the Hybrid Model . . . 24
- 4.2 The Semi-Honest Setting . . . 25
- 4.3 The Malicious Setting . . . 29
- 4.4 A Leakage-Free Variant via Constrained Prime Generation . . . 33

5 Performance Modeling and Simulation-Based Analysis . . . 35
- 5.1 Theoretical and Empirical Comparison of Soundness Errors . . . 36
- 5.2 Computational Cost Comparison of Three Tests . . . 38
- 5.3 Performance Modeling and Evaluation . . . 39

6 Conclusion . . . 42

7 Mathematical Tools and Supplementary Proofs and Protocols . . . 46
- 7.1 The CRT and its applications . . . 46
- 7.2 The Jacobi Symbol and Related Consequences . . . 48
- 7.3 Lucas Pseudo-primes . . . 49
- 7.4 Three Biprimality Tests: Variant Miller-Rabin, Boneh-Franklin, Lucas Tests . . . 50
- 7.5 The Security Model . . . 52
- 7.6 Missing Proofs of Section 3.2 . . . 53
- 7.7 A Proof of Theorem 3 . . . 56
- 7.8 A security proof of Protocol π_{Leg} . . . 64
- 7.9 Some Lemmas of Statistical Distance . . . 66
- 7.10 Security Proof against Malicious Adversaries . . . 71
- 7.11 Missing Functionalities, Protocols and Algorithms . . . 75
- 7.12 Three RSA Moduli Protocols in the Semi-honest Model . . . 79
- 7.13 Adaptation of Chen's Biprime Test Protocol . . . 81
- 7.14 AND Gate Cost Analysis of the π_{VerifyBI} Circuit . . . 85

1 Introduction

The security of foundational public-key cryptosystems like RSA [38] rests on the elegant complexities of number theory. A central challenge within this paradigm is generating the required secret primes, p and q, without a single point of failure. Multi-Party Computation (MPC) offers a robust solution by allowing multiple parties to collectively compute the public modulus $N = pq$ from their private inputs while preserving confidentiality. This technique has become a critical building block for numerous advanced cryptographic primitives, such as threshold homomorphic encryption [24,26], time-lock puzzles [1,32,39], accumulators [6, 9, 31], and verifiable delay functions [8, 17, 21, 27, 36, 41]. However, this distributed approach introduces a fundamental mathematical problem: how to efficiently verify number-theoretic properties, such as the biprimality of N, when the underlying factors must remain secret.

The primary objective in distributed RSA modulus generation is to design a secure protocol for n parties, resilient against up to $t < n$ colluding adversaries. The protocol should output a random and valid RSA modulus $N = pq$, where p, q are distinct primes of a specified size, such that an adversary learns nothing beyond N, ensuring the privacy of p and q. Such protocols typically involve two phases: **(a) Prime Candidate Sieving**: participants generate a potential RSA modulus N that does not divide by a prime less than a predetermined integer $p_{\min}$; and **(b) Biprimality test**: the candidate N is repeatedly tested by a biprimality test. If N is rejected by the biprimality test, then the process starts over. Current state-of-the-art sieving techniques often employ the Chinese Remainder Theorem (CRT) to efficiently generate candidates N free of small prime factors [15, 40]. For biprimality testing, variants of the Miller-Rabin primality test (cf. [14, Section 3.2]) and Boneh-Franklin (BF) biprimality test (cf. Theorem 1) have been commonly used.

Among these, the BF test [10] has become a foundational building block in numerous state-of-the-art protocols for distributed RSA generation [10,16,23,40], largely due to its conceptual simplicity and efficiency per iteration in an MPC setting. However, its widespread adoption has been blocked by a persistent efficiency bottleneck rooted in its theoretical understanding. The worst-case soundness error of the test is universally accepted to be $1/2$, forcing protocols to perform a large number of costly iterations (e.g., 80 iterations for 2^{-80} security) to ensure soundness. This raises a fundamental question: is this $1/2$ bound the true theoretical limit of the test, making such a high iteration count an unavoidable cost for verifying an RSA modulus?

This inefficiency becomes more apparent when compared to the Miller-Rabin primality test. The Miller-Rabin test exhibits a strictly lower worst-case soundness error of $1/4$ [13, 37]. Furthermore, subsequent studies [14, 19] have formally derived its average-case properties, utilizing these bounds to propose a significantly reduced iteration count for achieving a target security level. However, for applications requiring provable security guarantees, relying on average-case results can be problematic. An often unstated issue is that the average-case soundness of Miller-Rabin tests is proven under the assumption that primes are

chosen uniformly at random. This assumption is not met by many efficient distributed RSA generation algorithms [10, 15, 16, 18, 23, 40], which produce primes from non-uniform distributions (e.g., sums of uniform variables) to achieve MPC efficiency. Therefore, with provable security as the foremost priority, utilizing the worst-case soundness error to establish the number of verification rounds remains an essential requirement.

In light of these challenges, although the Miller-Rabin test exhibits a lower worst-case soundness error compared to the BF test, its distributed protocols are significantly more complex. This inherent complexity is precisely why prior works have relied on average-case results to boost its practical efficiency. Furthermore, in the realm of biprimality testing, the standard BF test is strictly confined to generating Blum integers; determining whether this limitation can be removed remains a compelling open question. Motivated by these observations, our work sets out to explore the true theoretical limits of the BF test and to investigate whether the strict limitation to Blum integers can be overcome.

Related Work The paradigm of distributed RSA moduli generation was established in the seminal work of Boneh and Franklin [10]. They presented an efficient protocol for verifying if a modulus $N = pq$ is biprime without disclosing the factors p and q. Their construction is secure against semi-honest adversaries, assuming an honest majority. The test guarantees perfect completeness, always accepting a valid RSA modulus, while ensuring a soundness error of at most $1/2$. The paper detailed two distinct biprimality tests, primarily differing in their core algebraic checks: one verifies that $\gcd(pq, (p-1)(q-1)) = 1$, whereas the other relies on exponentiation within the quotient group $\left(\mathbb{Z}_N[x]/(x^2+1)\right)^{\times}/\mathbb{Z}_N^{\times}$.

The mainstream approach in subsequent literature has largely centered on the variant that checks the condition $\gcd(pq, p+q-1) = 1$. A pivotal contribution came from Algesheimer et al. [2], who developed a distributed Miller-Rabin test secure against a dishonest majority. This direction, leveraging the Miller-Rabin framework for biprimality testing, has since been advanced by several other studies [14, 18].

Separately, another body of work has investigated the average-case error probability of these tests. Assuming a uniformly random input, Damgård et al. [19] derived an upper bound on this error for the Miller-Rabin test, and Einsele et al. [20] did so for Lucas-based tests. More recently, Burkhardt et al. [14] provided a comprehensive survey of the state-of-the-art, while also proposing improvements in RSA candidate sieving and enhanced security models.

1.1 Our Contributions

Our work provides a precise analysis of the worst-case soundness error bound for the existing BF test using a precise counting formula, introduces a novel testing method applicable to more general RSA moduli, and conducts a rigorous comparative evaluation that clarifies the practical trade-offs between these methods. Consequently, similar to the Miller-Rabin test, the BF test now only requires $s/2$

iterations to achieve a security level of 2^{-s}. Furthermore, our analysis reveals that in the specific scenario of generating Blum integers, the BF test remains the superior choice. Our primary contributions are:

1. A Precise Counting Formula for Biprimality Tests. Our foundational contribution is a precise characterization of the algebraic structure of false witnesses for biprimality tests proposed by Boneh-Franklin. This counting formula yields two immediate and significant consequences:

A Tighter Bound for the Boneh-Franklin Test. We prove that the worst-case soundness error for the BF test is 1/4, not the long-accepted 1/2. This theoretical result directly enhances the efficiency of numerous existing protocols. For instance, the protocol against malicious adversaries of Chen et al. [15], our tighter bound reduces the required number of iterations from approximately $2.5s$ to $\lceil 1.475s \rceil$ to achieve a failure probability of less than 2^{-s} (cf. Section 7.13).

The First Direct, Quantitative Comparison of Biprimality Tests. Our framework facilitates the first direct, quantitative comparison of the soundness of different biprimality tests, including the BF test, the variant Miller-Rabin test, and our proposed Lucas test. This enables the deep performance analysis presented in Section 5, which was previously not possible.

2. A Novel, Generalized Test with Widespread Soundness Superiority. Inspired by our counting strategy, we introduce a versatile biprimality test based on Lucas sequences. This test has two key advantages:

A Provable and Empirically Widespread Soundness Advantage. Enabled by our framework, we formally prove that for a significant class of non-RSA moduli (cf. Lemma 9), our Lucas test has a strictly lower soundness error. Our empirical results (cf. Table 3) strongly confirm that this theoretical advantage holds in over 99.99% of tested cases, establishing the compelling hypothesis that the Lucas test, by requiring fewer average rejection rounds, could overcome its marginal local computation cost.

Relaxed Prime Constraints. A crucial advantage is that our test operates without the common $p \equiv q \equiv 3 \pmod 4$ restriction. This broadens its applicability to all standard RSA moduli, including those required for advanced schemes such as the Joye-Libert cryptosystem [29] and KMOV cryptosystem [12], filling a gap in the literature.

3. A Rigorous Performance Verdict and Secure Protocols. To definitively test the aforementioned hypothesis, a key contribution of our work is the development of a holistic performance model that precisely characterizes the performance trade-offs (cf. Section 5.3). This model indicates a crucial insight:

when instantiated with conservative, empirically-grounded parameters, the extreme local efficiency of the BF test suggests it maintains a practical advantage in most realistic scenarios for generating Blum integers.

Finally, we construct secure and practical protocols for our Lucas test. We provide both an efficient controllable method and a general MPC-based approach to securely generate the required parameter D. Building on these solutions, we construct a Lucas-based protocol for the semi-honest model that ensures privacy for all standard RSA moduli (cf. Theorem 3), and adapt it to the malicious model for Blum integers (cf. Theorem 4). To validate our claims, our implementation is publicly available for reproducibility [3].

1.2 Technical Overview

We now provide a high-level roadmap of the technical framework underlying our results. Fundamentally, establishing the worst-case soundness of a biprimality test reduces to computing the exact ratio of passing witnesses to all valid candidates. To tackle this algebraic counting problem within specific multiplicative groups associated with $\mathbb{Z}_N$ and its quadratic algebras, we develop a unified framework that systematically reduces the problem scale. Specifically, we utilize the CRT to decompose the global counting problem into local prime-power components, resolve the base cases, and subsequently apply Hensel lifting to obtain the exact solutions. By applying this approach to the BF test's Sylow 2-subgroup structure, we reveal that its theoretical worst-case configuration is structurally impossible, tightening the bound to 1/4 (Section 3.1). We then deploy this exact machinery to rigorously quantify the false witnesses of our proposed Lucas test (Section 3.2).

Translating the Lucas test into a secure protocol (Section 4) introduces a unique challenge: jointly sampling a public parameter D that satisfies the hidden constraint $(\frac{-D}{p}) = -1$ without leaking the secret prime p. To overcome this, we design two distinct parameter-generation strategies. For arbitrary RSA moduli, we propose an interactive rejection-sampling approach that guarantees a negligible abort probability while strictly bounding the adversarial factoring advantage by a constant factor $2^{\text{LeakBound}}$. Alternatively, we construct a perfectly leakage-free variant that shifts this burden to the MPC prime generation phase. By embedding specific quadratic non-residue constraints (e.g., $p \equiv a \pmod{4D}$) directly into the distributed sieving process, the Legendre symbol condition is satisfied implicitly by construction.

In the proof against semi-honest adversaries, a novel trapdoor mechanism (Section 4.2) within the quadratic algebra is introduced: the simulator predetermines the exponentiation outcome and reverse-engineers the public Lucas parameters (P, Q) to match, ensuring indistinguishability. As for the model against malicious adversaries, we adapt the commit-and-prove framework of Chen et al. [15]. It (Section 4.3) combines semi-honest execution with a Schnorr-

[3] https://github.com/getamis/alice/blob/master/biprimalitytest/README.md

like zero-knowledge proof to enforce honest behavior, achieving a provable soundness bound of $\gamma := \frac{5}{8} + \frac{0.625}{p_{\min}-3}$ per round.

2 Preliminaries

Basic notations. Let $\mathbb{N}$ denote the set of natural numbers, and let $\mathbb{Z}$ be the ring of integers. For a finite set S, $|S|$ denotes its cardinality. Let $\mathbb{Z}_N$ denote the additive group of integers modulo N, and let $\mathbb{Z}_N^\times$ be its multiplicative group of units. The order of this multiplicative group is $|\mathbb{Z}_N^\times| = \phi(N)$, where ϕ is Euler's totient function. The greatest common divisor of two positive integers $x, y \in \mathbb{N}$ is denoted by $\gcd(x, y)$. Let $[\![a]\!]_m$ denote the additive sharing of a value a over the domain $\mathbb{Z}_m$. In this scheme, each participant $\mathcal{P}_i$ (for $i \in \{1, \ldots, n\}$) holds a secret share $\mathfrak{a}_i \in \mathbb{Z}_m$ such that $\sum_{i=1}^n \mathfrak{a}_i \equiv a \pmod m$. Similarly, $[\![a]\!]_\mathbb{Z}$ denotes the secure additive sharing of a over the integers $\mathbb{Z}$, where each participant holds a share $\mathfrak{a}_i \in \mathbb{Z}$ satisfying $\sum_{i=1}^n \mathfrak{a}_i = a$. For a finite set S, the notation $a \leftarrow S$ indicates that a is sampled uniformly at random from S.

To formally evaluate the biprimality tests, we define the following key parameters and sets. Given two odd positive integers p, q, set $e_4 (= e_4(p, q)) := \frac{1}{4}\big(p + \left(\frac{-1}{p}\right)\big)\big(q + \left(\frac{-1}{q}\right)\big)$. In particular, if $p \equiv q \equiv 3 \pmod 4$, then $e_4 = (p-1)(q-1)/4$. Here $\left(\frac{\cdot}{\cdot}\right)$ is the Jacobi symbol (cf. Section 7.2). For odd integers p, q, we set

$$\mathrm{BF}(N, e_4) := \left\{ g \in \mathbb{Z}_N^\times \mid g^{e_4} \equiv \pm 1 \pmod N \right\}, G(N) := \left\{ g \in \mathbb{Z}_N^\times \;\middle|\; \left(\frac{g}{N}\right) = 1 \right\},$$

$$\mathcal{Z}^\epsilon(D, N) := \left\{ (P, Q) \;\middle|\; \begin{array}{l} P^2 - 4Q = D \pmod N, \\ \left(\frac{Q}{N}\right) = \epsilon, \gcd(Q, N) = 1, \\ 0 \le P, Q < N \end{array} \right\},$$

$$\mathrm{LPBP}(D, N, e_4) := \left\{ (P, Q) \;\middle|\; \begin{array}{l} 0 \le P, Q < N, \\ \gcd(Q, N) = 1, \\ P^2 - 4Q = D \pmod N, \\ (\alpha\beta^{-1})^{e_4} \equiv \pm 1 \pmod{N\mathcal{O}_D} \end{array} \right\}.$$

For $\epsilon \in \{\pm 1\}$, we define the union $\mathcal{Z}(D, N) = \mathcal{Z}^{+1}(D, N) \cup \mathcal{Z}^{-1}(D, N)$. In the definitions above, α and β are the two distinct roots of the quadratic polynomial $x^2 - Px + Q$, and $\mathcal{O}_D := \mathbb{Z}[x]/\langle x^2 - Px + Q \rangle$ denotes the quadratic algebra. When $p \equiv q \equiv 3 \pmod 4$, the set $\mathrm{BF}(N, e_4)$ forms a subgroup of $G(N)$. Under the same conditions, $\mathrm{LPBP}(D, N, e_4)$ is a subset of $\mathcal{Z}^{+1}(D, N)$ (see Proposition 2). Assuming $p \equiv q \equiv 3 \pmod 4$ and D is a perfect square, we are primarily interested in evaluating two quantities: $\beta_{\mathrm{Lucas}}(D, N, e_4) := \frac{|\mathrm{LPBP}(D,N,e_4)|}{|\mathcal{Z}^{+1}(D,N)|}$ and $\beta_{\mathrm{BF}}(N, e_4) := \frac{|\mathrm{BF}(N,e_4)|}{|G(N)|}$. These two ratios represent the soundness errors of their respective biprimality tests; they quantify the probability that a randomly selected candidate passes the test when $N = pq$ is a composite, non-RSA modulus. Note that Propositions 1 and 2 imply that $\beta_{\mathrm{Lucas}}(D, N, e_4)$ is independent

of the specific choice of the perfect square D, provided $p \equiv q \equiv 3 \pmod 4$. For simplicity, we adopt the shorthand notation $\beta_{\text{Lucas}}(N, e_4) \coloneqq \beta_{\text{Lucas}}(1, N, e_4)$.

2.1 Boneh-Franklin Biprimality Test.

A core component of distributed RSA key generation is the Boneh-Franklin (BF) biprimality test [10]. It provides an efficient method for parties to verify that a shared candidate modulus N is the product of two distinct primes, p and q, without revealing these secret factors. The core logic of a single, idealized round of the test is captured in Algorithm 1. The challenge of computing this value securely in an MPC setting is addressed in Protocol 6.

Algorithm 1: Boneh-Franklin Biprimality Test (Single Round)

Input: Odd integer $N = pq$, where $p, q \equiv 3 \pmod 4$ and $\gcd(N, p + q - 1) = 1$.

1 Select an integer $g \in \mathbb{Z}_N^\times$ with $(\frac{g}{N}) = 1$;
2 **if** $g^{e_4} \equiv \pm 1 \pmod N$ **then**
  return true;
end
3 **return false**;

3 Soundness Analysis of Biprimality Tests

Before delving into the specific counting formulas for the Boneh-Franklin and Lucas tests, we provide a unified intuition for our proofs. Determining the worst-case upper bound, specifically maximizing the possible ratio $|H|/|G|$, for a composite, non-RSA modulus N is fundamentally equivalent to studying the behavior of the Sylow 2-subgroups within the underlying algebraic structures of these tests.

By decomposing the relevant set via the CRT, one might naively assume that the absolute worst-case scenario occurs when N is square-free and composed of exactly three distinct prime factors, seemingly yielding a soundness error of $1/2$. However, our proofs reveal a fundamental algebraic barrier: the specific input conditions of both tests strictly force this worst-case $1/2$ to be unattainable.

- **In the Boneh-Franklin Test:** The condition $N \equiv 1 \pmod 4$ strictly implies that any false modulus composed of three prime factors must contain at least one factor $p_i \equiv 1 \pmod 4$. This inevitably expands the Sylow 2-subgroup of $p_i - 1$, strictly depressing the upper bound to $1/4$.
- **In the Lucas Test:** The analysis is remarkably parallel, despite operating in a quadratic algebra. Driven by the constraint $\left(\frac{-D}{p}\right) = \left(\frac{-D}{q}\right) = -1$, a similar algebraic contradiction prevents all prime factors from taking the minimal Sylow 2-subgroup size.

This shared algebraic phenomenon, where structural congruences force an expanded Sylow 2-subgroup, elegantly unifies both proofs and explains why the worst-case soundness error is strictly bounded at $1/4$ rather than $1/2$.

3.1 Refined Analysis of the Boneh-Franklin Test

In this context, the set of all potential witnesses, G, is the set $G(N) = \{g \in \mathbb{Z}_N^\times \mid \left(\frac{g}{N}\right) = 1\}$, while the set of passing witnesses, H, is $\mathrm{BF}(N, e_4) = \{g \in \mathbb{Z}_N^\times \mid g^{e_4} \equiv \pm 1 \pmod{N}\}$. We demonstrate that, contrary to the long-held belief of a $1/2$ worst-case soundness, a belief based on analyses that went no further than proving $\mathrm{BF}(N, e_4) \subset G(N)$, the tightest bound is in fact $1/4$. We adapt this general counting framework to refine the long-standing soundness bound of the BF test, and then apply this canonical approach to rigorously establish the soundness of our proposed Lucas test.

The Precise Counting Formula The precise counting formulas of $\mathrm{BF}(N, e_4)$ and $G(N)$ are given below. First, to derive the formula for $|\mathrm{BF}(N, e_4)|$, our proof relies on a fundamental result for counting roots of unity in cyclic groups.

Lemma 1. [5, Lemma 2.1] *Let G be a cyclic group and d an integer. There are exactly $\gcd(d, |G|)$ dth-roots of 1 in G.*

By applying this result to the cyclic subgroups of $\mathbb{Z}_N^\times$ that arise from the CRT, we obtain the following precise formula.

Lemma 2. *Let $p \equiv q \equiv 3 \pmod 4$ with $\gcd\left(pq, e_4\right) = 1$. Assume that $N \coloneqq pq = \prod_{i=1}^{s} p_i^{r_i}$, where p_i is prime for all i, then we have*

$$|\mathrm{BF}(N, e_4)| = 2 \cdot \prod_{i=1}^{s} \gcd(e_4, d_i).$$

Here $p_i - 1 = 2^{k_i} d_i$ with $2 \nmid d_i$ for all $1 \le i \le s$.

Proof. Since e_4 is odd, we have

$$|\{g \in \mathbb{Z}_N^\times \mid g^{e_4} \equiv 1 \pmod{N}\}| = |\{g \in \mathbb{Z}_N^\times \mid g^{e_4} \equiv -1 \pmod{N}\}|$$

by the bijective map $g \mapsto -g$, which implies that

$$|\mathrm{BF}(N, e_4)| = 2 \cdot |\{g \in \mathbb{Z}_N^\times \mid g^{e_4} \equiv 1 \pmod{N}\}|.$$

According to Lemma 12, we reduce the problem to count the cardinality of e_4-th roots of 1 in $(\mathbb{Z}/p_i^{r_i}\mathbb{Z})^\times$ which are cyclic groups for all i [28, Theorem 3, Chapter 4], since N is odd. Combining this fact, and Lemma 1, one has the number of e_4-th roots of 1 in the group $\left(\mathbb{Z}/p_i^{r_i}\mathbb{Z}\right)^\times$ is

$$\begin{aligned} \gcd(e_4, \phi(p_i^{r_i})) &= \gcd(e_4, p_i - 1), \text{ since } \gcd(e_4, N) = 1, \\ &= \gcd(e_4, d_i), \text{ since } e_4 \text{ is odd.} \end{aligned}$$

The above discussion implies the desired result. □

Lemma 3. *Let* $N = \prod_{i=1}^{s} p_i^{r_i}$ *be a positive odd integer. Then, the cardinality of* $G(N)$ *is given by:*

$$|G(N)| = \begin{cases} \phi(N), & \text{if } N \text{ is a perfect square;} \\ \phi(N)/2, & \text{if } N \text{ is not a perfect square.} \end{cases}$$

Proof. The proof is based on the fact that if $\gcd(a, N) = 1$, then for exactly half of such values of a, the Jacobi symbol $\left(\frac{a}{N}\right) = 1$. A complete proof is given in Section 7.2.

A Tighter Worst-Case Bound of 1/4 Our proof of a tighter bound for non-RSA moduli relies on a case-by-case analysis based on the number of prime factors and their exponents.

Theorem 1 (Boneh-Franklin biprimality test). *Let* N *be an odd integer given as the product of two integers* $p, q \equiv 3 \pmod 4$*, and let* $e_4 = (p-1)(q-1)/4$ *be the test exponent, with* $\gcd(N, e_4) = 1$*. Then we have*

Completeness: *If* N *is an RSA modulus, then* $\mathrm{BF}(N, e_4) = G(N)$.
Soundness: *If* $N \neq 9$ *is a non-RSA modulus, then* $|\mathrm{BF}(N, e_4)| \leq |G(N)|/4$.

Proof. We now apply the counting strategy outlined in our introduction. The proof hinges on a structural analysis of the non-RSA modulus N, particularly when it has three or more prime factors.
Case 1: N is a valid RSA modulus (p, q are distinct primes). Recall that $p_i - 1 = 2^{k_i} d_i$ with odd d_i for all i using the same notation as in Lemma 2. First, consider the case p, q are distinct primes, which implies $e_4 = d_1 d_2$ and $k_1 = k_2 = 1$. From Lemma 3, the proof of this case is completed by $\mathrm{BF}(N, e_4) \subset G(N)$, and the following equality:

$$|\mathrm{BF}(N, e_4)| = 2 \gcd(e_4, d_1) \cdot \gcd(e_4, d_2) = 2 d_1 d_2 = \phi(N)/2 = |G(N)|.$$

Case 2: N is not an RSA modulus. Before analyzing each case, we first emphasize that by Lemma 2 and Lemma 3, for all $N = \prod_{i=1}^{s} p_i^{r_i}$, the soundness error β_{BF} is given by

$$\beta_{\mathrm{BF}}(N, e_4) = \begin{cases} \dfrac{2 \prod_{i=1}^{s} \gcd(e_4, d_i)}{\prod_{i=1}^{s} p_i^{r_i - 1}(p_i - 1)} \leq \dfrac{2^{1 - \sum k_i}}{\prod_{i=1}^{s} p_i^{r_i - 1}}, & \text{if } N \text{ is a square;} \\ \dfrac{2 \prod_{i=1}^{s} \gcd(e_4, d_i)}{(\prod_{i=1}^{s} p_i^{r_i - 1}(p_i - 1))/2} \leq \dfrac{2^{2 - \sum k_i}}{\prod_{i=1}^{s} p_i^{r_i - 1}}, & \text{otherwise.} \end{cases} \tag{1}$$

Subcase 2.1: N is a perfect square. A perfect square N implies that its prime factorization $N = \prod p_i^{r_i}$ has $r_i \geq 2$ for all $1 \leq i \leq s$. The upper bound for this expression occurs when the number of prime factors s is minimized (i.e., $s = 1$), and the values of p_i and r_i in the denominator are as small as possible (i.e.,

$r_1 = 2$, $p_1 = 7$ or $r_1 = 4$, $p_1 = 3$). Hence, we have an upper bound is $1/p_1 < 1/7$, since the theorem explicitly excludes the case $p = q = 3$.
Subcase 2.2: N is square-free (with $s \geq 3$ distinct prime factors). Consider the case $s = 3$. We first establish the modular properties of the prime factors $\{p_1, p_2, p_3\}$. From the condition of this theorem, $N = pq$ with $p, q \equiv 3 \pmod 4$, which implies $N = p_1 p_2 p_3 \equiv 1 \pmod 4$, so there exists at least one p_i of them to be congruent to 1 (mod 4) for some $1 \leq i \leq 3$. For such p_i with $p_i \equiv 1 \pmod 4$, its corresponding exponent k_i is at least 2. Thus, for our set of three prime factors, the sum of the exponents is bounded by $\sum_{i=1}^{3} k_i \geq 1 + 1 + 2 = 4$, which gives the desired result for $s = 3$:

$$\beta_{\mathrm{BF}}(N, e_4) \leq 2^{2-\sum k_i} \leq 2^{2-4} = 1/4.$$

For cases where $s = 4$, the argument is more direct. Since $k_i \geq 1$ for any odd prime p_i, the sum of exponents is bounded by $\sum_{i=1}^{s} k_i \geq s$. Therefore, the general bound is:

$$\beta_{\mathrm{BF}}(N, e_4) \leq 2^{2-\sum k_i} \leq 2^{2-s} \leq 1/4.$$

For any $s \geq 5$, the bound becomes strictly tighter (e.g., $2^{2-5} = 1/8$ or smaller). Thus, the worst-case probability for any $s \geq 3$ is $1/4$.
Subcase 2.3: N is neither a perfect square nor square-free. This case is defined by two conditions: (1) at least one exponent r_i is odd (non-perfect-square), and (2) at least one exponent $r_j \geq 2$ (non-square-free). First, consider the case $s = 1$, then the formula of β_{BF} gives us

$$\beta_{\mathrm{BF}}(N, e_4) \leq 2^{2-1} \cdot p_1^{1-r_1} \leq 2 \cdot 3^{1-3} = 2/9.$$

Next, if $s \geq 2$, then without loss of generality, we may assume $N = p_1^2 \prod_{i=2}^{s} p_i^{r_i}$. Recall the estimate of β_{BF} is given in the inequality (1). To maximize β_{BF}, we use minimal parameters. If $p_1 \geq 5$, the formula yields $\beta_{\mathrm{BF}}(N, e_4) \leq p_1^{-1} 2^{2-s} \leq 1/5$. We set $p_1 = 3$. For the resulting case $N = 3^2 p_2^{r_2}$, if $r_2 \geq 2$, the error is bounded by $(1/3) \cdot (1/p_2) \leq 1/15$, since $p_2 \geq 5$. Thus, we consider $r_2 = 1$. The assumption $p \equiv q \equiv 3 \pmod 4$ implies $p_2 \equiv 1 \pmod 4$, which dictates $k_2 \geq 2$. Consequently, the maximum error is exactly $2^{2-3} \cdot 3^{-1} = 1/6$, which occurs when $p_2 \equiv 5 \pmod 8$ (e.g., $p_2 = 5$).

Having analyzed all possible structures for a non-RSA modulus N, we have shown that the soundness error is at most $1/4$ in every case (excluding $N = 9$). This completes the proof. □

Conditions for the Worst-Case Scenario Based on the proof of Theorem 1, we establish the necessary and sufficient conditions for the worst-case scenario to occur.

Corollary 1. *Assume that the assumption of Theorem 1 holds. The equality* $|\mathrm{BF}(N, e_4)| = |G(N)|/4$ *is true if and only if* $N = \prod_{i=1}^{s} p_i^{r_i}$ *has one of the two structures described in Table 1, up to the symmetry of the inputs p and q.*

Table 1. Necessary and Sufficient Conditions for the Worst-Case Soundness Error

Case	Structure	Prime Factor Conditions	GCD Conditions
$s = 3$	$p = p_1 p_2$ $q = p_3$	$p_1 \equiv 5 \pmod 8$ $p_2, p_3 \equiv 3 \pmod 4$	$\gcd(e_4, p_1 - 1) = \frac{p_1 - 1}{4}$ $\gcd(e_4, p_i - 1) = \frac{p_i - 1}{2}$ for $i \in \{2, 3\}$
$s = 4$	$p = p_1 p_2 p_3$ $q = p_4$	$p_i \equiv 3 \pmod 4$ for all $i \in \{1, \ldots, 4\}$	$\gcd(e_4, p_i - 1) = \frac{p_i - 1}{2}$ for all $i \in \{1, \ldots, 4\}$

Proof. From the proof of Theorem 1, the maximum value can occur only if N is square-free with $s = 3$ or 4 and the equality holds throughout the estimate (1). □

Remark 1 (Necessity of the 3 (mod 4) *Condition).* The structural constraint $p \equiv q \equiv 3 \pmod 4$ is absolutely strictly necessary for the soundness of the Boneh-Franklin test. If this condition is relaxed (e.g., allowing $p \equiv q \equiv 1 \pmod 4$), the test can completely fail. Specifically, via the CRT, we can construct an infinite family of composite moduli $N = p_1(p_2 p_3)$ where the test accepts *every* element $g \in \mathbb{Z}_N^\times$ (yielding a soundness error of exactly 1). The formal construction of these catastrophic counterexamples is detailed in Lemma 14. This highlights a vulnerability if the congruence check is omitted in practical MPC implementations (cf. Appendix 7.13).

Tightness of the Bound. To explain that the 1/4 soundness bound is tight, we construct infinitely many pairs (p, q) that satisfies the conditions of the first worst-case scenario ($s = 3$) from Table 1. Note that Dirichlet's theorem on arithmetic progressions ensures that there are infinitely many primes p' such that $p' \equiv 23 \pmod{420}$. We set $p = p_1 p_2$ with $p_1 = 5, p_2 = 3$, and $q = p_3 = p'$. These satisfy the prime factor conditions in Table 1. We now verify the GCD conditions. The term $e_4 = (p-1)(q-1)/4 = 7(p'-1)/2$. For example, to check the condition for $p_1 = 5$, we compute $\gcd(e_4, p_1 - 1) = \gcd(7(p'-1)/2, 4)$. Since $p' \equiv 23 \pmod{420}$, $p' - 1 = 22 + 420k$, which means $(p'-1)/2 = 11 + 210k$ is odd. Therefore, $\gcd(7(p'-1)/2, 4) = 1$, which equals $(p_1 - 1)/4$. A similar check confirms the other conditions.

Furthermore, this construction ensures that $\gcd(N, e_4) = 1$. Since $p' \equiv 23 \pmod{420}$, we have $p' \equiv 2 \pmod 3$ and $p' \equiv 3 \pmod 5$. Consequently, neither 3 nor 5 divides $p' - 1$, guaranteeing that $\gcd(15p', e_4) = 1$.

3.2 Soundness Analysis of the Lucas Test

We first present the standard Lucas test for prime numbers, and also provide an algorithmic representation to give the reader a clearer understanding of the verification method in our proposed test.

Let P and Q be integers, $D \coloneqq P^2 - 4Q$, and $\mathcal{O}_D$ be the associated quadratic ring. If a prime p with $p \nmid 2QD$, then we have

$$(\alpha\beta^{-1})^{p - \left(\frac{D}{p}\right)} \equiv 1 \pmod{p\mathcal{O}_D}.$$

Here α, β are two distinct roots of the polynomial $x^2 - Px + Q$ (cf. [5]). This thus provides a primality test. Inspired by this property, we have designed a novel biprimality test.

Before detailing the algorithm, we provide the intuition behind its construction. To transition from a single-prime test to a biprimality test for an RSA modulus $N = pq$, we naturally look for an exponent that simultaneously annihilates the underlying algebraic groups for both prime factors. By constructing the index $e_4 = \left[p - \left(\frac{D}{p}\right)\right]\left[q - \left(\frac{D}{q}\right)\right]/4$, we guarantee that it relates to a common multiple of the respective group orders modulo p and modulo q. Consequently, by CRT, the congruence naturally holds modulo $N = pq$.

Algorithm 2: The Proposed Lucas Biprimality Test

Input: Odd integers p, q where $N = pq$ and $\gcd(N, e_4) = 1$, with $e_4 = [p + (\frac{-1}{p})][q + (\frac{-1}{q})]/4$.

1 Select an integer D such that $(\frac{-D}{p}) = (\frac{-D}{q}) = -1$;

2 Randomly select $P \in \mathbb{Z}_N$ such that $(\frac{Q}{N}) = 1$, where $Q = (P^2 - D)/4$;

3 **if** $(\alpha\beta^{-1})^{e_4} \equiv \pm 1 \pmod{N\mathcal{O}_D}$ **then**
  return true;
end

4 **return false**;

In the Boneh-Franklin test, verification relies solely on the multiplicative group $\mathbb{Z}_N^\times$, where false witnesses only need to align with a static period $p_i - 1$. The Lucas test, however, evaluates the roots of a quadratic polynomial $x^2 - Px + Q$, inextricably linking addition (trace P) and multiplication (norm Q). Consequently, the underlying test period shifts to $p_i - \left(\frac{D}{p_i}\right)$. Because the Legendre symbol $\left(\frac{D}{p_i}\right) \in \{\pm 1\}$ introduces a dynamic sign fluctuation, satisfying this quadratic period imposes strictly tighter arithmetic constraints than the static $p_i - 1$. This drastically reduces the chance of coincidental alignment by prime factors, giving the Lucas test a much stronger restrictive power against false witnesses.

By analogy with the study of the Boneh-Franklin test, we investigate sets $\mathcal{Z}^{+1}(D, N)$ and $\mathrm{LPBP}(D, N, e_4)$ comprising the elements that fulfill Steps (1) and (2) of Algorithm 2, respectively. In the following propositions, we determine their cardinalities to evaluate the soundness of the proposed Lucas test.

Our approach begins by establishing several key counting lemmas for the sets $\mathcal{Z}^{\epsilon}(D, p^r)$. We first analyze the base case for prime moduli ($r = 1$) and then use Hensel's Lemma to generalize these results to prime powers ($r \geq 2$). These building blocks are then combined via the CRT to prove the main counting formula in Proposition 1, which is the cornerstone for the soundness analysis of the theorem.

Phase 1: Local Analysis Modulo p We begin with the technical core involving character sums.

Lemma 4. *Let p be an odd prime and $D \not\equiv 0 \pmod{p}$. Then*

$$\sum_{i=1}^{(p-1)/2} \left(\frac{i^2+D}{p}\right) = \frac{-1-\left(\frac{D}{p}\right)}{2}.$$

Proof. This is a classical result in number theory, often presented as an exercise in standard textbooks (e.g., [28, Chapter 5, Exercise 8]). The logic rests on applying Euler's criterion to represent the Jacobi symbol as a power, followed by a binomial expansion. The inner sums vanish over the cyclic group $\mathbb{Z}_p^\times$ except for specific exponents. For completeness, the full algebraic derivation is provided in Lemma 19. □

We apply the same rigorous counting strategy to our proposed Lucas biprimality test. In this context, the set of all potential witnesses, G, is $\mathcal{Z}^{+1}(D,N)$, and the set of passing witnesses, H, is $\mathrm{LPBP}(D,N,e_4)$. Subsequently, we analyze the worst-case upper bound of their quotient.

Lemma 5. *Let p be an odd prime, and D be an element of $\mathbb{Z}_p^\times$, then for $\epsilon \in \{\pm 1\}$,*

$$|\mathcal{Z}^{\epsilon}(D,p^r)| = \begin{cases} (1+\epsilon)\,2^{-1}p^{r-1}\left(p-\left(\frac{D}{p}\right)-1\right), & \text{if } 2 \mid r; \\ 2^{-1}p^{r-1}\left(p-\left(\frac{D}{p}\right)-1-\epsilon\right), & \text{if } 2 \nmid r. \end{cases}$$

Proof. The proof is structured based on the parity of the exponent r.

Case 1: r is even. In this case, for any Q such that $\gcd(Q,p^r)=1$, the Jacobi symbol $\left(\frac{Q}{p^r}\right) = (\frac{Q}{p})^r$ is always 1. This implies that $|\mathcal{Z}^{+1}(D,p^r)| = |\mathcal{Z}(D,p^r)|$ and $|\mathcal{Z}^{-1}(D,p^r)| = 0$. We can therefore focus on computing the cardinality of the full set $\mathcal{Z}(D,p^r)$.

To compute the cardinality of the full set $|\mathcal{Z}(D,p^k)|$, we first derive a general formula for any exponent $k \geq 1$. For the base case $k=1$, we count the solutions to $P^2 \equiv D+4Q \pmod{p}$ where $0 < Q < p$. Since the map $Q \mapsto D+4Q$ is a bijection, as Q iterates over $\mathbb{Z}_p \setminus \{0\}$, the value $D+4Q$ iterates over $\mathbb{Z}_p \setminus \{D\}$. The equation has two distinct roots if $D+4Q$ is a non-zero quadratic residue. Since there are $(p-1)/2$ such residues, and we must subtract the case where the excluded value D is a residue, there are $(p-1)/2 - (1+(\frac{D}{p}))/2$ such values of Q. Accounting for the two roots per residue and the single root at $Q = -D/4$, the total cardinality is:

$$\left(\frac{p-1}{2} - \frac{1+\left(\frac{D}{p}\right)}{2}\right)\cdot 2 + 1 = p - \left(\frac{D}{p}\right) - 1.$$

For a general exponent $k \geq 2$, we lift the solutions from the base case. As shown in Lemma 7, which relies on Hensel's Lemma, shows $|\mathcal{Z}(D,p^k)| = p^{k-1}(p-(\frac{D}{p})-1)$. Now, we apply this general formula to our specific case where the exponent

is an even number r. This matches the formula in the lemma statement since $(1+\epsilon)/2$ is 1 for $\epsilon = 1$ and 0 for $\epsilon = -1$.

Case 2: r is odd. The proof for this case is more involved and proceeds in three steps for the base case $r = 1$.

Step 1: Relating $|\mathcal{Z}^{+1}|$ to helper sets. We first relate the problem of counting pairs (P, Q) to counting solutions of a simpler quadratic form. By assuming $Q = Q'^2$, the set $\mathcal{Z}^{+1}(D, p)$ becomes equivalent to:

$$\left\{(P, Q') \;\middle|\; \begin{array}{l}(P/2)^2 \equiv (Q')^2 + D/4 \pmod p, \gcd(Q', p) = 1, \\ 0 \le P < p, 1 \le Q' \le (p-1)/2.\end{array}\right\}.$$

To analyze this, we define two auxiliary sets S^ϵ for $\epsilon \in \{\pm 1\}$:

$$S^\epsilon = \left\{1 \le i \le \frac{p-1}{2} \;\middle|\; \left(\frac{i^2 + D/4}{p}\right) = \epsilon\right\},$$

excluding any i where $i^2 \equiv -D/4 \pmod p$.

Step 2: Deriving a system of equations. We now establish a system of two linear equations for $|S^{+1}|$ and $|S^{-1}|$. The first equation is derived by summing the Legendre symbol over all possibilities, leveraging Lemma 4:

$$\sum_{i=1}^{(p-1)/2} \left(\frac{i^2 + D/4}{p}\right) = |S^{+1}| - |S^{-1}| = \frac{-1 - \left(\frac{D/4}{p}\right)}{2} = \frac{-1 - \left(\frac{D}{p}\right)}{2}.$$

This gives us the relation:

$$|S^{-1}| = |S^{+1}| + \frac{1 + \left(\frac{D}{p}\right)}{2}. \tag{2}$$

The second equation comes from a direct count of the total number of terms:

$$\begin{aligned}|S^{+1}| + |S^{-1}| =& \left|\left\{1 \le i \le \frac{p-1}{2}\right\}\right| - \left|\left\{1 \le i \le \frac{p-1}{2} \;\middle|\; i^2 \equiv -D/4 \pmod p\right\}\right| \\ =& \frac{p-1}{2} - \frac{1 + \left(\frac{-D}{p}\right)}{2}.\end{aligned} \tag{3}$$

The final term accounts for the case where $i^2 \equiv -D/4 \pmod p$ might have a solution.

Step 3: Solving for $|\mathcal{Z}^\epsilon(D, p)|$. Solving the system of linear equations (2) and (3) yields the cardinality of $|S^{+1}|$. We then relate this back to $|\mathcal{Z}^{+1}(D, p)|$. For each $i \in S^{+1}$, there are two distinct solutions for P in $(P/2)^2 \equiv i^2 + D/4 \pmod p$.

An additional solution for P exists if $i = 0$ is a valid choice, which happens if $-D/4$ is a quadratic residue. This relationship is captured by:

$$|\mathcal{Z}^{+1}(D,p)| = 2 \cdot |S^{+1}| + \frac{1 + \left(\frac{-D}{p}\right)}{2}. \tag{4}$$

Combining these three equations gives the cardinality for $\epsilon = 1$:

$$\left|\mathcal{Z}^{+1}(D,p)\right| = \frac{p - \left(\frac{D}{p}\right) - 2}{2}.$$

The cardinality for $\epsilon = -1$ follows from $|\mathcal{Z}^{-1}| = |\mathcal{Z}| - |\mathcal{Z}^{+1}|$:

$$\left|\mathcal{Z}^{-1}(D,p)\right| = \left(p - \left(\frac{D}{p}\right) - 1\right) - \left(\frac{p - \left(\frac{D}{p}\right) - 2}{2}\right) = \frac{p - \left(\frac{D}{p}\right)}{2}.$$

For the general case where $r \geq 3$ is odd, the result is obtained by lifting the solutions from $\mathbb{Z}_p$ to $\mathbb{Z}_{p^r}$ using Hensel's Lemma, as detailed in Lemma 8. This yields the factor of p^{r-1}. □

Phase 2: Lifting to Prime Powers via Hensel's Lemma We lift the local results to prime powers p^r. The core intuition behind the following results lies in the lifting process from $\mathbb{Z}_p$ to $\mathbb{Z}_{p^r}$. For any element $x \in \mathbb{Z}_{p^r}$ expressed in the form $x = a + tp$ (where $0 \leq a < p$), the membership conditions, specifically the quadratic character, are primarily determined by the residue $a \pmod p$. Since the variable t can take any of the p^{r-1} possible values without altering the base residue modulo p, each solution found in the prime case naturally scales by a factor of p^{r-1} when extended to prime powers. We first recall Hensel's Lemma, which provides the standard method for this lifting, and then apply it to rigorously determine the cardinality of $\mathcal{Z}(D,p^r)$.

Lemma 6 (Hensel's Lemma). [34, Theorem 2.23 or 2.24] *Let $f(x)$ be a polynomial with integer coefficients. If p is a prime number and a is an integer such that $f(a) \equiv 0 \pmod{p^j}$, and $f'(a) \not\equiv 0 \pmod p$ then, there exists an integer $t \pmod p$ such that $f(a + tp^j) \equiv 0 \pmod{p^{j+1}}$.*

Lemma 7. *Let D be an integer and p be an odd prime with $\gcd(p, 2D) = 1$. Then we have $|\mathcal{Z}(D,p^r)| = p^{r-1}\left(p - \left(\frac{D}{p}\right) - 1\right)$.*

Proof. When $r = 1$, in the beginning proof of Lemma 5 gives us $|\mathcal{Z}(D,p)| = p - \left(\frac{D}{p}\right) - 1$. When $r \geq 2$, we first consider the case where $Q \equiv -D/4 \pmod p$, and need to compute the cardinality of the set

$$\left\{(P,Q) \;\middle|\; \begin{array}{l} P^2 \equiv 0 \pmod p, \\ \gcd(Q,p^r) = 1, 0 \leq P, Q < p^r \end{array}\right\}. \tag{5}$$

The number of solution (P,Q) is p^{r-1}, which form is $(P,Q) = \big(tp, ((tp)^2 - D)/4\big)$, where $t \in \mathbb{Z}_{p^{r-1}}$.

For the case $Q \not\equiv -D/4 \pmod{p}$, we consider $Q = a + tp$, where $t \in \mathbb{Z}_{p^{r-1}}$ and

$$a \in \mathcal{T} \coloneqq \left\{ Q \not\equiv -D/4 \in \mathbb{Z}_p^\times \;\middle|\; x^2 \equiv D + 4Q \mod p \text{ is solvable} \right\}.$$

For each $Q \in \mathbb{Z}_{p^r}$ with $Q \not\equiv -D/4 \pmod{p}$, if $f_Q(x) \coloneqq x^2 - 4Q - D \equiv 0 \pmod{p}$ has m solutions in $\mathbb{Z}_p$, and $f_Q'(a) \not\equiv 0 \pmod{p}$ for all $a \in \mathcal{T}$. Therefore, by Lemma 6, $f_Q(x) \equiv 0 \pmod{p^r}$ also has m solutions in $\mathbb{Z}_{p^r}$. Since $t \in \mathbb{Z}_{p^{r-1}}$ is arbitrary, then the number of solutions for this case is $p^{r-1}(|\mathcal{Z}(D,p)| - 1)$. Therefore, the total number of solutions is $p^{r-1}(|\mathcal{Z}(D,p)| - 1) + p^{r-1} = p^{r-1}(|\mathcal{Z}(D,p)|)$. □

Lemma 8. *Let p be an odd prime, and D be an element of $\mathbb{Z}_p^\times$. Then we have, for any odd $r \geq 1$ and $\epsilon \in \{\pm 1\}$,*

$$|\mathcal{Z}^\epsilon(D,p^r)| = p^{r-1} \cdot |\mathcal{Z}^\epsilon(D,p)|.$$

Proof. When $r = 1$, the desired result have been proved in the proof of Lemma 5. It suffices to prove the formula for the case $\epsilon = 1$. The result for $\epsilon = -1$ can then be derived from the identity $|\mathcal{Z}^{-1}(D,p^r)| = |\mathcal{Z}(D,p^r)| - |\mathcal{Z}^{+1}(D,p^r)|$, using the already established formula for $|\mathcal{Z}(D,p^r)|$ from Lemma 7.

We now proceed to prove the case for $\epsilon = 1$. When $\left(\frac{-D/4}{p}\right) = 1$ holds, one has $\big(tp, ((tp)^2 - D)/4\big) \in \mathcal{Z}^{+1}(D,p^r)$ for $t \in \mathbb{Z}_{p^{r-1}}$. The number of solutions lifted from the singular case where $Q \equiv -D/4 \pmod{p}$ is p^{r-1}. Using the same trick as in the Lemma 7, express Q as $a + tp$, where $t \in \mathbb{Z}_{p^{r-1}}$, and

$$a \in \left\{ Q \not\equiv -D/4 \in \mathbb{Z}_p^\times \;\middle|\; x^2 \equiv D + 4Q \mod p \text{ is solvable }, \left(\frac{Q}{p}\right) = 1 \right\}.$$

Notice that $\left(\frac{Q}{p^r}\right) = \left(\frac{Q}{p}\right) = 1$, since r is odd. Therefore, For each $Q \in \mathbb{Z}_{p^r}$, if $x^2 \equiv D + 4Q \mod p$ and $\left(\frac{Q}{p}\right) = 1$ has m solutions, then $x^2 \equiv D + 4Q \mod p^r$ and $\left(\frac{Q}{p^r}\right) = 1$ also has m solutions by Lemma 6. Since $t \in \mathbb{Z}_{p^{r-1}}$ is arbitrary, then the number of solutions of $\mathcal{Z}^{+1}(D,p^r)$ is $p^{r-1}\big(|\mathcal{Z}^{+1}(D,p)| - 1\big)$. For the case where $\left(\frac{-D/4}{p}\right) = -1$, there are no $(P,Q) \in \mathcal{Z}^{+1}(D,p^r)$ with $Q \equiv -D/4 \pmod{p}$. Consequently, the number of solutions of $|\mathcal{Z}^{+1}(D,p^r)|$ is given by $p^{r-1}|\mathcal{Z}^{+1}(D,p)|$ following the same reasoning as above. □

Phase 3: The Global Case (Proof of Proposition 1) Finally, we apply the CRT to obtain Proposition 1.

Proposition 1. *Let D be an integer and $N \coloneqq \prod_{i=1}^{s} p_i^{r_i}$ be a positive integer with $\gcd(N, 2D) = 1$. For each prime factor p_i, let $A_i \coloneqq p_i - \left(\frac{D}{p_i}\right) - 1$. Further, let $S_1 \coloneqq \{i \mid r_i \text{ is odd}\}$. Then:*

$$|\mathcal{Z}^{+1}(D,N)| = \begin{cases} \displaystyle\prod_{i=1}^{s} p_i^{r_i-1}\left(p_i - \left(\frac{D}{p_i}\right) - 1\right), & \text{if } N \text{ is square;} \\ \displaystyle\frac{1}{2}\left(\prod_{i=1}^{s} p_i^{r_i-1}\right)\left(\prod_{i=1}^{s} A_i + (-1)^{|S_1|}\prod_{i\in S_1^c} A_i\right), & \text{otherwise.} \end{cases}$$

where S_1^c is the complement of S_1, i.e., the set of indices i for which r_i is even.

Proof. We provide an elementary proof using the CRT and mathematical induction.

Case 1: N is a perfect square. Applying CRT (cf. Lemma 13), we only consider the case $\mathcal{Z}(D, p_i^{r_i}) = \mathcal{Z}^{+1}(D, p_i^{r_i})$. Lemma 5 gives us $|\mathcal{Z}(D, p_i^{r_i})| = p_i^{r_i-1}A_i$. The total cardinality of $\mathcal{Z}(D,N)$ is given by the product of the cardinalities of the individual components.

Case 2: N is not a perfect square. If N is not a perfect square, at least one exponent r_i must be odd, meaning the set S_1 is non-empty. From the CRT decomposition (cf. Lemma 13), we have:

$$|\mathcal{Z}^{+1}(D,N)| = \left(\prod_{i\in S_1^c} |\mathcal{Z}^{+1}(D, p_i^{r_i})|\right) \cdot \left(\sum_{\epsilon_1\cdots\epsilon_{|S_1|}=1} \prod_{j\in S_1} |\mathcal{Z}^{\epsilon_j}(D, p_j^{r_j})|\right).$$

The first term, over the set S_1^c (even exponents), simplifies as in Case 1 to $\prod_{i\in S_1^c} p_i^{r_i-1}(p_i - (\frac{D}{p_i}) - 1)$. The main difficulty lies in finding a closed form for the summation term over S_1. Let $k = |S_1|$. We will prove the following identity by induction on k:

$$\sum_{\epsilon_1\cdots\epsilon_k=1} \prod_{i=1}^{k} |\mathcal{Z}^{\epsilon_i}(D, p_i^{r_i})| = \frac{1}{2}\left(\prod_{i=1}^{k} p_i^{r_i-1}\right)\left(\prod_{i=1}^{k} A_i + (-1)^k\right).$$

This proof can be concluded through mathematical induction on the cardinality of $|S_1| = k$. When $|S_1| = 1$, the condition $\epsilon_1 = 1$ is fixed. Lemma 5 says that the sum holds for $k = 1$. Assuming that $|S_1| = k$, the equality is satisfied, and from Lemma 7,

$$|\mathcal{Z}^{-1}| = |\mathcal{Z}| - |\mathcal{Z}^{+1}| = \prod_{i=1}^{k} p_i^{r_i-1}A_i - |\mathcal{Z}^{+1}| = \frac{1}{2}\left(\prod_{i=1}^{k} p_i^{r_i-1}\right)\left(\prod_{i=1}^{k} A_i - (-1)^k\right).$$

When $|S_1| = k+1$, we have

$$\begin{aligned}
&\sum_{\epsilon_1\cdot\ldots\cdot\epsilon_{k+1}=1}\prod_{i=1}^{k+1}|\mathcal{Z}^{\epsilon_i}(D,p_i^{r_i})| \\
=\quad &|\mathcal{Z}^{+1}(D,p_{k+1}^{r_{k+1}})|\cdot\sum_{\epsilon_1\cdot\ldots\cdot\epsilon_k=1}\prod_{i=1}^{k}|\mathcal{Z}^{\epsilon_i}(D,p_i^{r_i})|+|\mathcal{Z}^{-1}(D,p_{k+1}^{r_{k+1}})| \\
&\cdot\sum_{\epsilon_1\cdot\ldots\cdot\epsilon_k=-1}\prod_{i=1}^{k}|\mathcal{Z}^{\epsilon_i}(D,p_i^{r_i})| \\
=&\left[\frac{p_{k+1}^{r_{k+1}-1}(A_{k+1}-1)}{2}\right]\left[\frac{1}{2}\left(\prod_{i=1}^{k}p_i^{r_i-1}\right)\left(\prod_{i=1}^{k}A_i+(-1)^k\right)\right] \\
&+\left[\frac{p_{k+1}^{r_{k+1}-1}(A_{k+1}+1)}{2}\right]\left[\frac{1}{2}\left(\prod_{i=1}^{k}p_i^{r_i-1}\right)\left(\prod_{i=1}^{k}A_i-(-1)^k\right)\right] \\
=&\frac{1}{4}\left(\prod_{i=1}^{k+1}p_i^{r_i-1}\right)\left[(A_{k+1}-1)\left(\prod_{i=1}^{k}A_i+(-1)^k\right)+(A_{k+1}+1)\left(\prod_{i=1}^{k}A_i-(-1)^k\right)\right] \\
=&\frac{1}{4}\left(\prod_{i=1}^{k+1}p_i^{r_i-1}\right)\Big[\left(A_{k+1}\textstyle\prod A_i+A_{k+1}(-1)^k-\prod A_i-(-1)^k\right) \\
&\qquad\qquad+\left(A_{k+1}\textstyle\prod A_i-A_{k+1}(-1)^k+\prod A_i-(-1)^k\right)\Big] \\
=&\frac{1}{4}\left(\prod_{i=1}^{k+1}p_i^{r_i-1}\right)\left[2\cdot\prod_{i=1}^{k+1}A_i-2\cdot(-1)^k\right]=\frac{1}{2}\left(\prod_{i=1}^{k+1}p_i^{r_i-1}\right)\left(\prod_{i=1}^{k+1}A_i+(-1)^{k+1}\right).
\end{aligned}$$

By the principle of mathematical induction, the identity holds. Thus, the identity for the summation over S_1 holds by induction. Finally, we combine the result for the odd-exponent factors for S_1 and the the even-exponent factors for S_1^c to obtain the desired formula stated in the proposition. This completes the proof. □

Remark 2. The result in Case 2 can be derived more concisely using the properties of character sums. Specifically, by using the indicator function $\frac{1+\prod_{i\in S_1}\left(\frac{Q}{p_i}\right)}{2}$, the sum over $\mathcal{Z}(D,N)$ can be expanded as:

$$\sum_{Q}\frac{1+\prod\left(\frac{Q}{p_i}\right)}{2}=\frac{1}{2}\left(|\mathcal{Z}(D,N)|+\prod_{i\in S_1}\left(\sum_{Q_i\in\mathcal{Z}_i}\left(\frac{Q}{p_i}\right)\right)\right).$$

Since $\sum_{Q_i\in\mathcal{Z}_i}\left(\frac{Q}{p_i}\right)=|\mathcal{Z}_i^{+1}|-|\mathcal{Z}_i^{-1}|=(-1)\cdot p_i^{r_i-1}$, the formula follows immediately.

Phase 4: Passing Witnesses and the Soundness Bound We study the cardinality of the set LPBP and then prove it is a subset of $\mathcal{Z}^{+1}$.

Proposition 2. *Let p, q be positive odd integers, $N = pq = \prod_{i=1}^{s} p_i^{r_i}$, and D be an integer in $\mathbb{Z}$ with $\gcd(2D, N) = 1$, and $\left(\frac{-D}{p}\right) = \left(\frac{-D}{q}\right) = -1$. Then we have the following:*

1. *The set $\mathrm{LPBP}(D, N, e_4)$ is a subset of $\mathcal{Z}^{+1}(D, N)$.*
2. *Furthermore, assuming $\gcd(N, e_4) = 1$, its cardinality is given by*

$$|\mathrm{LPBP}(D, N, e_4)| = \prod_{i=1}^{s} \left(\gcd(e_4, d_i') - 1 \right) + \prod_{i=1}^{s} \gcd(e_4, d_i'),$$

where the decomposition is now $p_i - \left(\frac{D}{p_i}\right) = 2^{k_i'} d_i'$ with $2 \nmid d_i'$ for all $1 \le i \le s$.

Proof. We first establish Claim 1 by leveraging the properties of the standard involution and the norm map in the quadratic algebra $\mathcal{O}_D/N\mathcal{O}_D$. Alternatively, this can also be proven by directly expanding $(P + \sqrt{D})^{e_4}$.

Let $(P, Q) \in \mathrm{LPBP}(D, N, e_4)$, which implies $(\alpha\beta^{-1})^{e_4} \equiv \pm 1 \pmod{N\mathcal{O}_D}$. We must show that $\left(\frac{Q}{N}\right) = 1$. Since e_4 is odd and $Q = \alpha\beta$, we have $\left(\frac{Q}{N}\right) = \left(\frac{Q^{e_4}}{N}\right) = \left(\frac{(\alpha\beta)^{e_4}}{N}\right)$.

Recall that α, β can be formally represented as $(P \pm \sqrt{D})/2$. Let σ denote the standard involution on $\mathcal{O}_D/N\mathcal{O}_D$ mapping $\sqrt{D} \mapsto -\sqrt{D}$, so $\sigma(\alpha) = \beta$.

Case A: $(\alpha\beta^{-1})^{e_4} \equiv 1 \pmod{N\mathcal{O}_D}$**.**
This implies $\alpha^{e_4} \equiv \beta^{e_4} \pmod{N\mathcal{O}_D}$. i.e., $\alpha^{e_4} \equiv A \in \mathbb{Z}_N$. Its norm satisfies $Q^{e_4} \equiv \alpha^{e_4}\beta^{e_4} \equiv A^2 \pmod{N}$. Therefore, $\left(\frac{Q^{e_4}}{N}\right) = \left(\frac{A^2}{N}\right) = 1$.

Case B: $(\alpha\beta^{-1})^{e_4} \equiv -1 \pmod{N\mathcal{O}_D}$**.**
This implies $\alpha^{e_4} \equiv -\beta^{e_4} \pmod{N\mathcal{O}_D}$. Thus, α^{e_4} is anti-invariant under σ (i.e., $\sigma(\alpha^{e_4}) \equiv -\alpha^{e_4}$). For any element $Y = u + v\sqrt{D} \in \mathcal{O}_D/N\mathcal{O}_D$, the condition $\sigma(Y) = -Y$ implies $u - v\sqrt{D} \equiv -u - v\sqrt{D}$, which forces $2u \equiv 0$. Since N is odd, $u \equiv 0$. Thus, the scalar part of α^{e_4} must vanish, meaning $\alpha^{e_4} \equiv B\sqrt{D} \pmod{N\mathcal{O}_D}$ for some $B \in \mathbb{Z}_N$. Its norm is $Q^{e_4} \equiv (B\sqrt{D})(-B\sqrt{D}) \equiv -B^2 D \pmod{N}$. Evaluating the Jacobi symbol yields:

$$\left(\frac{Q^{e_4}}{N}\right) = \left(\frac{-B^2 D}{N}\right) = \left(\frac{-D}{N}\right).$$

By the premise $\left(\frac{-D}{p}\right) = \left(\frac{-D}{q}\right) = -1$, we have $\left(\frac{-D}{N}\right) = (-1)(-1) = 1$. Thus, $\left(\frac{Q^{e_4}}{N}\right) = 1$. In both cases, $\left(\frac{Q}{N}\right) = 1$, which completes the proof of Claim 1.

Proof of Claim 2. To determine the cardinality of LPBP, we follow the methodology detailed in [5, Section 1.4]. By the CRT, the global counting problem reduces to evaluating the local sets $\mathrm{LPBP}(D, p_i^{r_i}, e_4)$. These local sets are partitioned into two disjoint cases based on the value of $(\alpha\beta^{-1})^{e_4} \pmod{p_i^{r_i}\mathcal{O}_D}$.

Case 1: $(\alpha\beta^{-1})^{e_4} \equiv 1 \pmod{p_i^{r_i}\mathcal{O}_D}$.
By Proposition 3, the number of e_4-th roots of 1 in $(\widehat{\mathcal{O}_D/p_i^{r_i}})$ is

$$\begin{aligned} d &= \gcd\left(e_4, p_i^{r_i-1}(p_i - (\tfrac{D}{p_i}))\right) \\ &= \gcd(e_4, p_i - (\tfrac{D}{p_i})) && \text{(since } \gcd(e_4, N) = 1\text{)} \\ &= \gcd(e_4, d_i'). && \text{(since } e_4 \text{ is odd)} \end{aligned}$$

Excluding roots where $\alpha\beta^{-1} - 1$ is not invertible modulo p_i, the local cardinality for this case is exactly $\gcd(e_4, d_i') - 1$.
Case 2: $(\alpha\beta^{-1})^{e_4} \equiv -1 \pmod{p_i^{r_i}\mathcal{O}_D}$.
In this case, the condition for the invertibility of $1 - (\alpha\beta^{-1}) \pmod{p_i}$ is always satisfied. Thus, the local cardinality for this case is simply $\gcd(e_4, d_i')$.

Aggregating these local results via the CRT requires observing that the global congruence $(\alpha\beta^{-1})^{e_4} \equiv \pm 1 \pmod{N\mathcal{O}_D}$ forces the local congruences to be either $+1$ for all prime factors or -1 for all prime factors simultaneously. Summing the products of these two strictly disjoint global scenarios yields the closed-form expression in the proposition. □

Finally, the soundness error, denoted β_{Lucas}, is estimated as follows.

Theorem 2. *Let N be an odd integer given as a product of two odd integers p, q. Let e_4 be the test exponent with $\gcd(N, e_4) = 1$. Let D be an integer such that $\gcd(2D, N) = 1$ and $\left(\frac{-D}{p}\right) = \left(\frac{-D}{q}\right) = -1$. The proposed Lucas test has the following properties:*

Completeness: *If N is an RSA modulus, then $\text{LPBP}(D, N, e_4) = \mathcal{Z}^{+1}(D, N)$.*
Soundness: *If N is non-RSA and its minimal prime factor $p_{\min} \geq 11$, the soundness error is bounded by:*

$$\beta_{\text{Lucas}}(D, N, e_4) < \frac{1}{4} + \frac{1.25}{p_{\min} - 3}.$$

Proof. Follow the same methodology of Theorem 1.
Case 1: N is a valid RSA modulus (p and q are distinct primes). Our goal is to prove that $\text{LPBP}(D, N, e_4) = \mathcal{Z}^{+1}(D, N)$. Since Proposition 2 already establishes that $\text{LPBP}(D, N, e_4) \subseteq \mathcal{Z}^{+1}(D, N)$, it suffices to show that their cardinalities are equal.

We use the notation $p_1 = p, p_2 = q$ and the decomposition $p_i - (\frac{D}{p_i}) = 2^{k_i'} d_i'$. The condition of the theorem premises $\left(\frac{-D}{p_i}\right) = -1$ implies the identity $p_i - \left(\frac{D}{p_i}\right) = p_i + \left(\frac{-1}{p_i}\right)$. Notice that $p_i + \left(\frac{-1}{p_i}\right) = 2d_i'$. Substituting this directly into the definition of e_4 gives $e_4 = \frac{1}{4}\left[p + \left(\frac{-1}{p}\right)\right]\left[q + \left(\frac{-1}{q}\right)\right] = d_1' d_2'$. This identity immediately implies that $\gcd(e_4, d_1') = d_1'$ and $\gcd(e_4, d_2') = d_2'$. Now, we have

$$\begin{aligned} |\text{LPBP}(D, N, e_4)| &= \big(\gcd(e_4, d_1') - 1\big)\big(\gcd(e_4, d_2') - 1\big) + \gcd(e_4, d_1')\gcd(e_4, d_2') \\ &= (d_1' - 1)(d_2' - 1) + d_1' d_2' = 2d_1' d_2' - d_1' - d_2' + 1. \end{aligned}$$

Next, Proposition 1 says that

$$|\mathcal{Z}^{+1}(D,N)| = ((2d_1'-1)(2d_2'-1)+1)/2 = 2d_1'd_2' - d_1' - d_2' + 1.$$

Case 2: Non-RSA Moduli. For the remaining cases, considering $N = \prod_i p_i^{r_i}$, applying Proposition 1 and Proposition 2 yields the following upper bound for the soundness error β_{Lucas}:

$$\begin{aligned} &\beta_{\text{Lucas}}(N, e_4) \qquad (6)\\ &\leq \begin{cases} \dfrac{2^{1-\sum k_i}\prod_{i=1}^{s}(p_i-1)}{\prod_{i=1}^{s} p_i^{r_i-1}(p_i-2)}, & \text{if } N \text{ is a square;} \\ \dfrac{2^{2-\sum k_i}\prod_{i=1}^{s}(p_i-1)}{\left(\prod_{i=1}^{s} p_i^{r_i-1}\right)\left(\prod_{i=1}^{s}(p_i-2) - \prod_{i\in S_1^c}(p_i-2)\right)}, & \text{otherwise.} \end{cases} \end{aligned}$$

Here, we recall that $S_1^c = \{i \mid r_i \text{ is even}\}$ is the set of indices corresponding to even exponents in the prime factorization of N.
Subcase 2.1: N is a perfect square. The equation (6) implies that for all $p_i \geq 11$,

$$\begin{aligned} &\beta_{\text{Lucas}}(D, N, e_4) \leq \left(\frac{2}{\prod_{i=1}^{s} p_i^{r_i-1}}\right)\left(\frac{\prod_{i=1}^{s} 2^{-k_i}(p_i-1)}{\prod_{i=1}^{s}(p_i-2)}\right) \\ &\leq \left(\frac{2}{\prod_{i=1}^{s} p_i^{r_i-1}}\right)\left(\prod_{i=1}^{s}\left(\frac{1}{2} + \frac{1}{2(p_i-2)}\right)\right) \leq \left(\frac{2}{11}\right)\left(\frac{1}{2}+\frac{1}{18}\right) = \frac{10}{99}. \end{aligned}$$

Subcase 2.2: N is square-free. When N is square-free. Consider the case $s = 3$. Then there exists one of $\{p_1, p_2, p_3\}$ is $4 \mid p_i - \left(\frac{D}{p_i}\right)$. If not, for all $1 \leq i \leq 3$, $p_i - \left(\frac{D}{p_i}\right) = 2d_i'$ with odd d_i' hold (i.e. $k_i' = 1$), which is equivalent to $p_i \equiv -\left(\frac{D}{p_i}\right) \pmod 4$. Let's assume, without loss of generality, that the inputs are $p = p_1$ and $q = p_2p_3$. The conditions of the theorem implies that $\left(\frac{-D}{p}\right) = \left(\frac{-D}{q}\right) = -1$. From $\left(\frac{-D}{q}\right) = -1$, we have $\left(\frac{D}{q}\right) = -\left(\frac{-1}{q}\right)$. Our contradiction assumption for p_2, p_3 means $p_2 \equiv -(\frac{D}{p_2}) \pmod 4$ and $p_3 \equiv -(\frac{D}{p_3}) \pmod 4$. Multiplying these congruences gives $q = p_2p_3 \equiv (\frac{D}{p_2})(\frac{D}{p_3}) = (\frac{D}{q}) \pmod 4$. So we have derived $q \equiv (\frac{D}{q}) \pmod 4$. Substituting in the identity $\left(\frac{D}{q}\right) = -\left(\frac{-1}{q}\right)$, we get:

$$q + \left(\frac{-1}{q}\right) \equiv 0 \pmod 4,$$

which gives us a contradiction, because $q + \left(\frac{-1}{q}\right) \equiv 2 \pmod 4$ for any odd integers $q > 1$. Hence, our initial assumption that all $k_i' = 1$ must be false. Therefore, at least one $k_i' \geq 2$, which implies $\sum_{i=1}^{3} k_i' \geq 4$. Therefore, applying

Lemma 20 and equation (6), we obtain that

$$\begin{aligned}\beta_{\text{Lucas}}(D,N,e_4) <& \frac{1}{4}\left[\frac{\prod_{i=1}^{3}(p_i-1)}{\prod_{i=1}^{3}(p_i-2)-1}\right] < \frac{1}{4}\left[\frac{(p_{\min}-1)^3}{(p_{\min}-2)^3-1}\right] \\ \leq& \frac{1}{4}\left[\frac{(p_{\min}-1)^4}{(p_{\min}-2)^4-1}\right], \text{ for all } p_{\min} \geq 5.\end{aligned}$$

Similarly, as $s = 4$, we also have

$$\beta_{\text{Lucas}}(D,N,e_4) < \frac{1}{4}\left[\frac{(p_{\min}-1)^4}{(p_{\min}-2)^4-1}\right].$$

When $s \geq 5$, applying the following fact

$$\frac{\prod_{i=1}^{s}(p_i-1)}{\prod_{i=1}^{s}(p_i-2)-1} < \left[\frac{\prod_{i=1}^{4}(p_i-1)}{\prod_{i=1}^{4}(p_i-2)-1}\right]\left[\frac{\prod_{i=5}^{s}(p_i-1)}{\prod_{i=5}^{s}(p_i-2)-1}\right],$$

and Lemma 21 with $j = 5$, we arrive that, for $s \geq 5$,

$$\beta_{\text{Lucas}}(D,N,e_4) \leq 2^{2-k_1-\ldots-k_s}\frac{\prod_{i=1}^{s}(p_i-1)}{\prod_{i=1}^{s}(p_i-2)-1} < \frac{1}{4}\left[\frac{(p_{\min}-1)^4}{(p_{\min}-2)^4-1}\right].$$

Thus, for any $s \geq 3$, the soundness error is bounded by:

$$\beta_{\text{Lucas}}(D,N,e_4) < \frac{1}{4}\left[\frac{(p_{\min}-1)^4}{(p_{\min}-2)^4-1}\right] < \frac{1}{4} + \frac{1.25}{p_{\min}-3}, \text{ (by Lemma 22).}$$

Subcase 2.3: N is neither a perfect square nor square-free. This case is defined by two conditions: (1) at least one exponent r_i is odd, which means $|S_1| \geq 1$ and (2) at least one exponent $r_j \geq 2$. Note that

$$\begin{aligned}&\prod_{i\in S_1^c}\left(p_i - \left(\frac{D}{p_i}\right) - 1\right)\left(\prod_{i\in S_1}\left(p_i - \left(\frac{D}{p_i}\right) - 1\right) + (-1)^{|S_1|}\right) \\ &\geq \prod_{i\in S_1^c}(p_i-2)\left(\prod_{i\in S_1}(p_i-2) - 1\right).\end{aligned}$$

Equation 6 shows that for all $p_i \geq 11$,

$$\begin{aligned}\beta_{\text{Lucas}}(D,N,e_4) &\leq \left[\frac{4}{\prod_{i=1}^{s} p_i^{r_i-1}}\right]\left[\frac{\prod_{i=1}^{s} 2^{-k_i}(p_i-1)}{\prod_{i=1}^{s}(p_i-2) - \prod_{i\in S_1^c}(p_i-2)}\right] \\ \leq \left[\frac{4}{\prod_{i=1}^{s} p_i^{r_i-1}}\right]&\left[\frac{\prod_{i=1}^{s}\left(\frac{1}{2} + \frac{1}{2(p_i-2)}\right)}{1-\left(\prod_{i\in S_1}(p_i-2)\right)^{-1}}\right] \leq \left[\frac{4}{11}\right]\left[\frac{\frac{1}{2}+\frac{1}{18}}{1-9^{-1}}\right] = \frac{5}{22}.\end{aligned}$$

Having analyzed all possible structures for a non-RSA modulus N, we have shown that the soundness error is bounded as claimed. □

Remark 3. A final point regarding the premise $\left(\frac{-D}{p}\right) = -1$: this condition is not met if the input p is a perfect square. This poses no issue for our protocol for two key reasons. First, from a practical standpoint, the probability of a large integer chosen from the sum of uniform distributions being a perfect square is negligible. Second, and more importantly, if the input p is a perfect square, it cannot be a prime number, which means the resulting modulus $N = pq$ is an invalid RSA modulus. The failure to satisfy this premise is thus a feature, not a bug, as it is one way the protocol can reject invalid inputs.

4 The Proposed Lucas Biprimality Protocols

Distributed biprimality tests are essential components of distributed RSA key generation. The joint generation of an RSA modulus can be simplified into two main steps: 1) candidate modulus generation, where potential prime factors p and q are sampled in secret-shared form and their product $N = p \cdot q$ is securely computed using multi-party computation algorithms; and 2) RSA modulus verification, where participants verify that N is an RSA modulus without disclosing the underlying secret factors. In such distributed environments, sensitive values are maintained in secret-shared form among n participants $\mathcal{P}_1, \ldots, \mathcal{P}_n$, and to facilitate the description of our protocols, we recall that $[a]_m$ denotes the additive sharing of a value a over the finite group $\mathbb{Z}_m$, where each participant $\mathcal{P}_i$ holds a secret share $\mathfrak{a}_i \in \mathbb{Z}_m$ such that $\sum_{i=1}^{n} \mathfrak{a}_i \equiv a \pmod{m}$, and similarly, $[a]_{\mathbb{Z}}$ represents additive sharing over the integers satisfying $\sum_{i=1}^{n} \mathfrak{a}_i = a$ in $\mathbb{Z}$.

We propose two protocols based on Theorem 2. The first achieves security against semi-honest adversaries for all standard RSA moduli, while the second provides security with abort against malicious adversaries for the case of Blum integers. Both protocols are proven secure in the standard stand-alone model.

4.1 MPC Building Blocks and the Hybrid Model

Our protocols are analyzed in the *ideal-protocol hybrid model*, where parties have access to ideal functionalities that perform specific sub-tasks securely. For standard operations, we assume access to the following building blocks, whose formal specification boxes are provided in Appendix 7.11:

- **Additive Zero-Sharing ($\mathcal{F}_{\mathbf{Zero}}$):** Used to generate a set of secret shares that sum to zero, allowing parties to mask their private inputs during joint computations.
- **Coin-Tossing ($\mathcal{F}_{\mathbf{CT}}$):** Provides all participants with a source of common, unbiased public randomness, which is essential for sampling public test parameters.
- **Commitment ($\mathcal{F}_{\mathbf{Com}}$):** A standard primitive used in the malicious setting to ensure that parties are bound to their secret choices before the opening phase, preventing adaptive cheating.

– **Secure Multiplication and Shuffle ($\mathcal{F}_{\mathbf{ModMul}}, \mathcal{F}_{\mathbf{Shuffle}}$):** These facilitate the secure evaluation of products and exponents over $\mathbb{Z}_N$ and its quadratic algebras without revealing individual shares.

By abstracting these foundational primitives, we focus our analysis on the high-level logic of the Lucas biprimality verification.

4.2 The Semi-Honest Setting

To translate our Lucas test into a secure protocol, we must address a challenge that does not exist in the original Boneh-Franklin setting: the selection of the parameter D. For non-Blum integers, the protocol must securely find a D satisfying both $\left(\frac{-D}{N}\right) = 1$ and $\left(\frac{-D}{p}\right) = -1$. A naive interactive search for D would leak information about the secret prime p with each tested candidate. This challenge exposes a fundamental trade-off between generality, efficiency, and perfect (leakage-free) security. Consequently, this section explores the two viable pathways that navigate this trade-off: one offering perfect security in controlled environments, and another providing a broadly applicable solution that accepts bounded leakage as a deliberate engineering compromise. We then present our main protocol, π^{S}_{RSA}, which formalizes the latter approach.

Navigating the Security-Generality Trade-off

A Leakage-Free but Constrained Approach. Our primary and most efficient solution mirrors the elegant approach for Blum integers where $D = 1$ is pre-selected. The core idea is to pre-fix a public value for D and impose congruence constraints on p and q during their generation. This guarantees that the conditions are met automatically, circumventing any interactive and potentially leaky search. For instance, by choosing a small prime D (e.g., $D = 3$) and ensuring primes are generated such that $p \equiv a \pmod{4D}$ and $q \equiv b \pmod{4D}$ for appropriate non-residues a, b, this leakage problem is completely avoided (cf. Section 4.4).

A General Approach with Bounded Leakage. For applications requiring full generality where pre-constraining primes is infeasible, a direct trade-off emerges between achieving perfect security and maintaining practical efficiency. Guided by the principle of practical viability, our work focuses on a strategy that deliberately navigates this compromise. We propose capping the interactive search for a suitable prime D at a constant number of trials, $\mathsf{LeakBound}$ (e.g., 7). Under the standard heuristic that the Legendre symbol behaves like a random coin flip, the probability of failing to find a suitable D for a valid prime after $\mathsf{LeakBound}$ independent trials is approximately $(1/2)^{\mathsf{LeakBound}}$. This design offers a twofold benefit: it guarantees a strictly bounded and tolerable abort probability ($< 2^{-\mathsf{LeakBound}}$) for valid moduli while retaining universal applicability. The explicit cost of this generality is a bounded information leakage of $\mathsf{LeakBound}$

bits about the secret p. This quantifiable leakage is reflected directly in the security proof (cf. Lemma 25), which incurs a loss factor of $2^{\mathsf{LeakBound}}$ and bounds the adversarial advantage by $2^{\mathsf{LeakBound}}\epsilon$, where ϵ is the advantage of factoring N.

The Protocol and Its Security We now present our main semi-honest protocol, $\pi^{\mathcal{S}}_{\mathrm{RSA}}$, which formalizes the generally applicable, bounded-leakage approach. We first define its target functionality, which captures the protocol's behavior, including the bounded leakage. To facilitate the secure computation of the exponent, we assume the input shares $[p]_{\mathbb{Z}}$ and $[q]_{\mathbb{Z}}$ possess a specific structure modulo 4, a standard technique first introduced by Boneh and Franklin [10].

Functionality 1 $\mathcal{F}^{\mathcal{S}}_{\mathrm{RSA}}(n, \mathsf{LeakBound})$

Inputs: Each party $\mathcal{P}_i$ has a public number $N = pq$, $p \pmod 4$, $q \pmod 4$, shares $[p]_{\mathbb{Z}}$ and $[q]_{\mathbb{Z}}$, where each share satisfies $\mathfrak{p}_1 \equiv p \pmod 4$, $\mathfrak{q}_1 \equiv q \pmod 4$, and $\mathfrak{p}_i \equiv \mathfrak{q}_i \equiv 0 \pmod 4$ for all $2 \leq i \leq n$.

Outputs:
If $p \equiv q \equiv 3 \pmod 4$:

- If $p \neq q$ are both primes and $\gcd(N, e_4) = 1$, then each party receives $(\texttt{RSAModulus}, \emptyset)$.
- Otherwise, each party receives $(\texttt{NonRSAModulus}, \{\mathfrak{p}_i, \mathfrak{q}_i\}_{i=1}^n)$.

Else:
Let $S_{\mathrm{Leak}} \coloneqq \{D_k \in [3, D_{\min}] \mid (\frac{-D_k}{N}) = 1 \text{ and } D_k \text{ is prime}\}$, where $D_{\min}$ is the minimal odd prime satisfying $(\frac{-D_{\min}}{p}) = -1$ and $(\frac{-D_{\min}}{N}) = 1$.

- If $|S_{\mathrm{Leak}}| > \mathsf{LeakBound}$, then each party receives $(\texttt{AbortLeakBound}, \{\mathfrak{p}_i, \mathfrak{q}_i\}_{i=1}^n)$.
- If $|S_{\mathrm{Leak}}| \leq \mathsf{LeakBound}$, and $p \neq q$ are both primes such that $\gcd(N, e_4) = 1$, then each party receives $\big(\texttt{RSAModulus}, \{(\frac{D_k}{p})\}_{D_k \in S_{\mathrm{Leak}}}\big)$.
- Otherwise, each party receives $(\texttt{NonRSAModulus}, \{\mathfrak{p}_i, \mathfrak{q}_i\}_{i=1}^n)$.

The realization of this functionality, Protocol 1, consists of the bounded search for D followed by κ iterations of the core exponentiation test. To compute the necessary Legendre symbols, it relies on an ideal functionality $\mathcal{F}_{\mathrm{Leg}}$, whose realization is detailed in Appendix 7.8.

Protocol 1 Lucas Biprimality Test $\pi^{\mathcal{S}}_{\mathrm{RSA}}(n, \kappa, \mathsf{LeakBound})$

Inputs: Each party $\mathcal{P}_i$ has $p \pmod 4$, $q \pmod 4$, N and $[p]_{\mathbb{Z}}, [q]_{\mathbb{Z}}$ with the structure defined in Functionality 1.

Outputs: $\left(\texttt{RSAModulus}, \left\{\left(\frac{D_k}{p}\right)\right\}_{D_k \in S_{\text{Leak}}}\right)$, $(\texttt{NonRSAModulus}, \{\mathfrak{p}_i, \mathfrak{q}_i\}_{i=1}^n)$, or $(\texttt{AbortLeakBound}, \{\mathfrak{p}_i, \mathfrak{q}_i\}_{i=1}^n)$.

Select an appropriate positive integer D:

1. If $p \equiv q \equiv 3 \pmod 4$, parties set $D = 1$, $S_{\text{Leak}} \coloneqq \emptyset$, and proceed to Step 5.
2. Let D_k be the k-th prime in the sequence $(3, 5, 7, \dots)$. If $\left(\frac{-D_k}{N}\right) \neq 1$, increment k and repeat Step 2. Otherwise, proceed to Step 3.
3. Party $\mathcal{P}_i$ adds D_k to S_{Leak}. If $|S_{\text{Leak}}| > \mathsf{LeakBound}$, the party broadcasts its shares and returns $(\texttt{AbortLeakBound}, \{\mathfrak{p}_i, \mathfrak{q}_i\}_{i=1}^n)$. Otherwise, it sends $([p]_{\mathbb{Z}}$, $p \pmod 4$, $D_k)$ to the functionality $\mathcal{F}_{\text{Leg}}$ to obtain $\left(\frac{-D_k}{p}\right)$.
4. If $\left(\frac{-D_k}{p}\right) = -1$, parties set $D = D_k$ and proceed to Step 5. Else, parties increment k and return to Step 2.

Exponential verification: For $j = 1, \dots, \kappa$:

5. Sample $P_j \leftarrow \mathbb{Z}_N$ and set $Q_j \coloneqq (P_j^2 - D) \cdot 4^{-1}$. If $\gcd(N, Q_j) \neq 1$, abort by broadcasting shares.
6. If $\left(\frac{Q_j}{N}\right) \neq 1$, they restart from Step 5 (via rejection sampling).
7. The parties securely compute $u_j = (\alpha_j \beta_j^{-1})^{e_4}$ using $\mathcal{F}_{\text{Shuffle}}$, where α_j, β_j are roots of $x^2 - P_j x + Q_j$.
8. All parties check $u_j \equiv \pm 1 \pmod{N\mathcal{O}_D}$. If the check fails, they abort by broadcasting shares.

GCD Test:

9. Parties securely compute a randomized multiple of e_4, denoted z, using $\mathcal{F}_{\text{ModMul}}$.
10. Parties open z and check if $\gcd(N, z) = 1$. If the check fails, they abort by broadcasting shares.

If all verifications pass, the protocol outputs $\left(\texttt{RSAModulus}, \left\{\left(\frac{D_k}{p}\right)\right\}_{D_k \in S_{\text{Leak}}}\right)$.

The following theorem formally establishes the privacy of our proposed protocol against semi-honest adversaries within the hybrid model.

Theorem 3. *Let p and q be odd integers, $N = pq$, and let $p_{\min}$ be the minimal prime factor of N. Let $\kappa, \mathsf{LeakBound} > 0$ be security parameters. The inputs to $\mathcal{P}_i$ are given as*

$$(N, [p]_{\mathbb{Z}}, [q]_{\mathbb{Z}}),$$

*where each share satisfies $\mathfrak{p}_1 \equiv p \pmod 4$, $\mathfrak{q}_1 \equiv q \pmod 4$, and $\mathfrak{p}_i \equiv \mathfrak{q}_i \equiv 0 \pmod 4$ for all $2 \leq i \leq n$. If $p_{min} \geq 11$, then the Protocol $\pi_{\text{RSA}}^{\mathcal{S}}$ $(n-1)$-*privately computes *the functionality $\mathcal{F}_{\text{RSA}}^{\mathcal{S}}$ in the $\mathcal{F}_{\text{Leg}}$, $\mathcal{F}_{\text{Shuffle}}$, $\mathcal{F}_{\text{ModMul}}$-hybrid model.*

Proof (Sketch). We prove the theorem for the case of a semi-honest adversary, as the protocol essentially implements a secure MPC version of Algorithm 2. The privacy of this protocol is established via the standard simulation paradigm. The core intuition relies on a novel trapdoor mechanism: because the Lucas test operates in the quadratic algebra $\mathcal{O}_D/N\mathcal{O}_D$, the simulator can pre-determine the test's outcome (i.e., fixing $(\alpha\beta^{-1})^{e_4}$) and subsequently reverse-engineer the public parameters (P, Q) to match. This allows the simulator to generate an indistinguishable transcript without knowing the underlying secret factors.

Correctness. The correctness of the protocol follows directly from the properties of the underlying Lucas biprimality test, as established in Theorem 2. Specifically:

1. $|S_{Leak}| > \mathsf{LeakBound}$**.** The protocol aborts during the parameter search phase and directly outputs `AbortLeakBound`.
2. $|S_{Leak}| \leq \mathsf{LeakBound}$**,** $p \neq q$ **are primes, and** $\gcd(N, e_4) = 1$**.** These inputs satisfy the completeness conditions of the Lucas and GCD tests. Consequently, the protocol passes all verifications and outputs `RSAModulus` with overwhelming probability.
3. $|S_{Leak}| \leq \mathsf{LeakBound}$**, and either** N **is not biprime or** $\gcd(N, e_4) \neq 1$**.** By Theorem 2, the protocol falsely passes the κ verification rounds to output `RSAModulus` with a negligible probability bounded by $\left(\frac{1}{4} + \frac{1.25}{p_{\min}-3}\right)^{\kappa}$. Thus, it successfully rejects the candidate and outputs `NonRSAModulus` with overwhelming probability.

Privacy. For privacy, we sketch the construction of a simulator $\mathcal{S}$. When the functionality's output is `NonRSAModulus` or `AbortLeakBound`, the simulation is straightforward, since the output includes all honest parties' shares $\{\mathfrak{p}_i, \mathfrak{q}_i\}_{i=1}^{n}$. We thus focus on the challenging case where the output is `RSAModulus`. The core task for the simulator is to generate a valid transcript, particularly the values $\{u_j\}_{j=1}^{\kappa}$, without knowledge of the honest parties' secret shares.

The simulation hinges on a trapdoor mechanism to construct the public Lucas parameters P_j, Q_j. The goal is to fix the value of $u_j \equiv (\alpha_j\beta_j^{-1})^{e_4} \pmod{N\mathcal{O}_D}$, which can be either 1 or -1, before determining P_j and Q_j.

To achieve this, the simulator proceeds as follows:

1. It randomly samples $v_j, w_j \in \mathbb{Z}_N$ (with $\gcd(v_j^2 - w_j^2 D, N) = 1$) and a bit $b_j \in \{0, 1\}$.
2. Let $a_j = \frac{v_j + w_j\sqrt{D}}{v_j - w_j\sqrt{D}}$. Exploiting the property (cf. Section 7.7)

$$(a_j^2(-1)^{b_j})^{e_4} \equiv (-1)^{b_j} \pmod{N\mathcal{O}_D},$$

the simulator sets the element $\alpha_j\beta_j^{-1} \coloneqq a_j^2(-1)^{b_j}$. This allows the simulator to know the value of u_j (which will be $(-1)^{b_j}$) in advance.

3. With the value of $\alpha_j \beta_j^{-1}$ fixed, the simulator can uniquely determine the corresponding public parameters (P_j, Q_j), since α_j and β_j are the roots of the polynomial $x^2 - P_j x + Q_j = 0$ and the parameters must satisfy the constraint $P_j^2 - 4Q_j \equiv D \pmod{N}$. These relations provide the trapdoor to generate the public transcript values.

Finally, the indistinguishability of the simulated transcript is guaranteed by Proposition 4, which asserts that the distribution of the P_j values generated by the simulator is identical to that of a real-world execution. Since the simulator can produce a valid transcript without any secret information, the protocol achieves strict privacy against semi-honest adversaries. □

4.3 The Malicious Setting

This section addresses the challenge of securing it against malicious adversaries. Transitioning from the semi-honest setting requires significant strengthening. We adopt the established commit-and-prove paradigm of Chen et al., which is directly applicable because the core algebraic structure of the Lucas test, a single multiplicative group exponentiation, is readily amenable to the Schnorr-like zero-knowledge proofs used to enforce honest execution.

Our goal here is not to present a fully optimized or universal protocol, but rather to provide a crucial proof-of-concept. We aim to demonstrate that the core algebraic structure of the Lucas test is compatible with such malicious-security transformations. To this end, we focus our analysis on the foundational case of Blum integers, for which $D = 1$. By constructing a secure variant for this domain, we establish that the Lucas test is not inherently limited to the semi-honest model. The generalization of this protocol to all RSA moduli and its concrete performance optimization are left as important directions for future work.

Our formal analysis will employ the biprimality test functionality $\mathcal{F}_{\mathrm{BI}}^{\mathcal{M}}$, as introduced by Chen et al. [15, Functionality 4.2].

Functionality 2 $\mathcal{F}_{\mathrm{BI}}^{\mathcal{M}}(n)$

Inputs: Each party $\mathcal{P}_i$ has a public number N, shares $[p]_{\mathbb{Z}}$ and $[q]_{\mathbb{Z}}$ with $\mathfrak{p}_i, \mathfrak{q}_i \geq 0$.

Outputs:
If all the following conditions are satisfied, then $\mathcal{F}_{\mathrm{BI}}^{\mathcal{M}}$ sends the message `BlumInteger` to the adversary $\mathcal{S}$:

1. All parties agree on the value of N;
2. $N = p \cdot q$;
3. $p \neq q$ are both primes;
4. $p \equiv q \equiv 3 \pmod{4}$;

5. $\gcd(N, e_4) = 1$;
6. $\mathfrak{p}_i \geq 0$ and $\mathfrak{q}_i \geq 0$ for all i.

If $\mathcal{S}$ responds with `proceed`, then output `BlumInteger` to all parties. If $\mathcal{S}$ responds with `cheat`, or if any of the previous conditions are false, then output $\{(\mathfrak{p}_i, \mathfrak{q}_i)\}_{i=1}^n$ directly to $\mathcal{S}$, and output `NonBlumInteger` to all parties.

Compared to the protocol by Chen et al., our approach incorporates an explicit check for $p \equiv q \equiv 3 \pmod 4$. Furthermore, we relax the requirement for p and q; instead of needing them to be confined by a fixed upper bound M (i.e., $0 < \mathfrak{p}_i, \mathfrak{q}_i < M$), our analysis only assumes $\mathfrak{p}_i, \mathfrak{q}_i > 0$. While this broader condition on p, q might theoretically include small primes if $p_{\min}$ is not enforced, this is not a concern in our setting as the target functionality is assumed to output p, q forming a Blum integer. Thus, an explicit upper bound M is unnecessary for the security of our protocol. In addition, we have some recommendations for the protocol by Chen et al. Details can be found in Section 7.13.

The overall single-round soundness error of the proposed Lucas test, which we denote by $\beta_{\text{Lucas}}^{\mathcal{M}}$, combines two factors: the soundness of the underlying Lucas test, bounded by $\beta_{\text{Lucas}} \coloneqq \frac{1}{4} + \frac{1.25}{p_{\min}-3}$ (Theorem 2), and the 1/2 soundness of the Schnorr-like zero-knowledge verification. This yields a total error of: $\beta_{\text{Lucas}}^{\mathcal{M}} \leq \beta_{\text{Lucas}} + (1 - \beta_{\text{Lucas}}) \cdot \frac{1}{2} = \frac{5}{8} + \frac{0.625}{p_{\min}-3}$. Consequently, to achieve an overall soundness error of $2^{-\kappa}$, the protocol must be iterated $\lceil \kappa / \log_2({\beta_{\text{Lucas}}^{\mathcal{M}}}^{-1}) \rceil = \left\lceil \left(\log_2 \frac{8p_{\min}-24}{5p_{\min}-10} \right)^{-1} \kappa \right\rceil$ times. For instance, setting $p_{\min} = 233$ requires $\lceil 1.489\kappa \rceil$ iterations to achieve the target security of $2^{-\kappa}$.

Our protocol is constructed in the hybrid model, relying on several standard ideal functionalities to perform fundamental cryptographic tasks. For general-purpose secure computation, we utilize $\mathcal{F}_{\text{ComCompute}}$ [15], which allows parties to first commit to their private inputs and later securely evaluate an agreed-upon public function on these committed values. To generate shared secrets that sum to zero, a common requirement for masking, we employ $\mathcal{F}_{\text{Zero}}$ [15]. Finally, our protocol also makes use of standard functionalities for commitment ($\mathcal{F}_{\text{Com}}$) and coin-tossing ($\mathcal{F}_{\text{CT}}$) to, respectively, ensure parties are bound to their choices and to generate public randomness [15].

Protocol 2 Malicious Lucas Biprimality Test $\pi_{\text{BI}}^{\mathcal{M}}(n, \kappa, p_{\min})$

Inputs: Each party $\mathcal{P}_i$ has N and $[p]_{\mathbb{Z}}, [q]_{\mathbb{Z}}$ with $0 \leq \mathfrak{p}_i, \mathfrak{q}_i$, where each share satisfies $\mathfrak{p}_1 \equiv \mathfrak{q}_1 \equiv 3 \pmod 4$, and $\mathfrak{p}_i \equiv \mathfrak{q}_i \equiv 0 \pmod 4$ for all $2 \leq i \leq n$.

Outputs: `BlumInteger` or `NonBlumInteger`.
Commitment Phase:

1. Let $\kappa_{\min} \coloneqq \left\lceil \left(\log_2 \frac{8p_{\min}-24}{5p_{\min}-10}\right)^{-1} \kappa \right\rceil$. Party $\mathcal{P}_i$ samples $\tau_{i,j} \leftarrow \mathbb{Z}_{2^{2\kappa-1}n^3 N\kappa_{\min}}$ for $1 \le j \le \kappa_{\min}$, and sends $\big(\texttt{commit}, i, (\mathfrak{p}_i, \mathfrak{q}_i, \{\tau_{i,j}\}_{j=1}^{\kappa_{\min}})\big)$ to functionality $\mathcal{F}_{\text{ComCompute}}$.
2. Each party $\mathcal{P}_i$ sends `sample` to $\mathcal{F}_{\text{Zero}}(2^{\kappa-3}nN)$ and receives r_i in response.

Lucas Test Phase:

3. Parties invoke $\mathcal{F}_{\text{CT}}(\mathcal{Z}^{+1}(1, N))$ to obtain $\{(P_j, Q_j)\}_{j=1}^{\kappa_{\min}}$.
4. Party $\mathcal{P}_1$ sets $y_{1,j} \coloneqq (\alpha_j\beta_j^{-1})^{r_1+(\mathfrak{p}_1(\frac{-1}{q})+\mathfrak{q}_1(\frac{-1}{p})+6)/4}$, and the other parties set $y_{i,j} \coloneqq (\alpha_j\beta_j^{-1})^{r_i+(\mathfrak{p}_i(\frac{-1}{q})+\mathfrak{q}_i(\frac{-1}{p}))/4}$ for all $2 \le i \le n$, $1 \le j \le \kappa_{\min}$. Here α_j and β_j are two roots of the polynomial $x^2 - P_j x + Q_j$. Party $\mathcal{P}_i$ sends $(\texttt{commit}, i, \{y_{i,j}\}_{j=1}^{\kappa_{\min}}, \{1, \ldots, n\})$ to $\mathcal{F}_{\text{Com}}$.
5. Party $\mathcal{P}_i$ sends $(\texttt{decommit}, i)$ to $\mathcal{F}_{\text{Com}}$ and receives $\{y_{i',j}\}_{j=1}^{\kappa_{\min}}$ for $i' \ne i$.
6. The parties output `NonBlumInteger` and halt if there exists $1 \le j \le \kappa_{\min}$ such that
$$(\alpha_j\beta_j^{-1})^{(N-5)/4} \cdot \prod_{i=1}^{n} y_{i,j} \not\equiv \pm 1 \pmod N.$$

Zero-Knowledge Verification Phase :

7. For $1 \le j \le \kappa_{\min}$, each party $\mathcal{P}_i$ computes $\gamma_{i,j} = (\alpha_j\beta_j^{-1})^{\tau_{i,j}} \pmod N$, and broadcasts $\{\gamma_{i,j}\}_{j=1}^{\kappa_{\min}}$.
8. All parties send `flip` to $\mathcal{F}_{\text{CT}}(\{0,1\}^{\kappa_{\min}})$ and then obtain an agreed-upon random bit vector $\vec{c} = (c_j)$ of length $\kappa_{\min}$.
9. For $1 \le j \le \kappa_{\min}$, the party $\mathcal{P}_1$ computes
$$\zeta_{1,j} = \tau_{1,j} + c_j \cdot \big(r_1 - (\mathfrak{p}_1 + \mathfrak{q}_1 - 6)/4\big),$$
and every other party $\mathcal{P}_i$ for $2 \le i \le n$ computes
$$\zeta_{i,j} = \tau_{i,j} + c_j \cdot \big(r_i - (\mathfrak{p}_i + \mathfrak{q}_i)/4\big).$$
They all broadcast the values they have computed to one another.
10. The parties halt and output `NonBlumInteger` if there exists any $1 \le j \le \kappa_{\min}$ such that
$$\prod_{i=1}^{n} (\alpha_j\beta_j^{-1})^{\zeta_{i,j}} \not\equiv \prod_{i=1}^{n} \gamma_{i,j} \cdot y_{i,j}^{c_j} \pmod N.$$
11. Let C be a circuit computing $\pi_{\text{VerifyBI}}(N, \vec{c}, \{\cdot, \cdot, \cdot, \zeta_{i,*}\}_{i\in\{1,\ldots,n\}})$; that is, let it be a circuit representation of Algorithm π_{VerifyBI} with the public values N, $\vec{c}$, and ζ hardcoded. The parties send $(\texttt{compute}, 1, \{1, \ldots, n\}, C)$ to $\mathcal{F}_{\text{ComCompute}}$, and in response they all receive z or `VerifyFail`. If receive `VerifyFail`, or if $\mathcal{F}_{\text{ComCompute}}$ aborts, then the parties halt and output `NonBlumInteger`.
12. The parties halt and output `BlumInteger` if $\gcd(z, N) = 1$, or halt and output `NonBlumInteger` otherwise.

While a full implementation and concrete benchmarking of the malicious protocol are beyond the scope of this work, we briefly analyze its performance characteristics. The dominant overhead stems from the generic secure computation within the $\mathcal{F}_{\text{ComCompute}}$ functionality of the monolithic π_{VerifyBI} circuit (cf. Algorithm 3), which holistically validates all relationships between secret shares and the public transcript.

The π_{VerifyBI} circuit is a comprehensive verification component executed via secure computation. It performs three main tasks:

1. **Modulus Validation:** Confirms that the secret shares $(\mathfrak{p}_i, \mathfrak{q}_i)$ correctly reconstruct the public modulus $N = (\sum \mathfrak{p}_i)(\sum \mathfrak{q}_i)$ as a Blum integer.
2. **Transcript Validation:** Validates the consistency of the zero-knowledge proof transcript by verifying Schnorr-like relations between the parties' commitments $(\tau_{i,j})$, challenges $(\vec{c})$, and responses $(\zeta_{i,j})$.
3. **GCD Output:** If all checks pass, it computes $z = r \cdot (p + q - 1) \pmod{N}$, which is used in a final GCD test to verify the coprimality of the exponent e_4 and N.

The circuit's complexity is primarily driven by large integer arithmetic. For n parties and $\kappa_{\min}$ iterations, the main computational components are: two large multiplications (one for modulus composition and one to compute z), and approximately $O(n \cdot \kappa_{\min} \cdot \ell)$ arithmetic operations to validate the zero-knowledge responses, where ℓ is the bit-length of N. These validation operations are performed over the large domain $\mathbb{Z}_{2^{2\kappa-1}n^3 N \kappa_{\min}}$. Given that these computations are realized using a generic MPC framework, the concrete costs–particularly in terms of communication rounds and total data transferred–are expected to be substantial. A detailed breakdown of the circuit's AND gate size is provided in Appendix 7.14.

We posit that these costs could be dramatically reduced by designing a dedicated, more efficient zero-knowledge proof system for the specific algebraic relations in Algorithm 3, rather than relying on generic MPC. We leave such concrete optimizations as a significant and important direction for future work.

Theorem 4. *Let $\kappa > 0$ be a security parameter. The inputs to all participants $\{\mathcal{P}_i\}_{i=1}^n$ are given as $(N, [p]_{\mathbb{Z}}, [q]_{\mathbb{Z}})$. If $p_{min} \geq 11$, then the Protocol $\pi_{\text{BI}}^{\mathcal{M}}$ $(n-1)$-*securely computes *the functionality $\mathcal{F}_{\text{BI}}^{\mathcal{M}}$* with abort *in the $\mathcal{F}_{\text{ComCompute}}$, $\mathcal{F}_{\text{Zero}}$, $\mathcal{F}_{\text{CT}}$, $\mathcal{F}_{\text{Com}}$-hybrid model. Here $p_{\min}$ is the minimal prime factor of N.*

Proof (Sketch). The proof demonstrates that Protocol $\pi_{\text{BI}}^{\mathcal{M}}$ securely computes the functionality $\mathcal{F}_{\text{BI}}^{\mathcal{M}}$ against a malicious adversary, $\mathcal{A}$, in the hybrid model. To achieve this malicious security, we rely on the established commit-and-prove paradigm via a Schnorr-like zero-knowledge proof. The core intuition for the simulation is that the simulator works backwards: it simulates the zero-knowledge responses prior to receiving the random challenges. This seamlessly forces the transcript to evaluate as a valid Blum integer check without requiring the underlying secret shares.

Case 1: The functionality's output is `BlumInteger` (Success Case). In this case, the simulator must generate a full, consistent protocol transcript without knowing the honest parties' secret shares. The simulator programs the ideal coin-tossing functionality $\mathcal{F}_{\text{CT}}$ to output Lucas parameters P_j using the same trapdoor technique as in the semi-honest case (cf. Theorem 3), which allows it to know the outcome of each test in advance.

To complete the transcript, the simulator generates the remaining values. To generate the values $y_{i,j}$ for the honest parties ($\overline{\mathcal{P}^*}$), the simulator uses a common technique to satisfy the protocol's verification equation. For all but one designated honest party $\mathcal{P}_{i'}$, it simulates their values $y_{i,j}$ by raising the base $(\alpha_j \beta_j^{-1})$ to a fresh, large random exponent. The value for the final party, $y_{i',j}$, is then uniquely determined by solving the global verification equation to match the predetermined outcome $(-1)^{b_j}$. This ensures the full set of values is internally consistent. For the zero-knowledge proof, the tuple $(\zeta_{i,j}, c_j, \gamma_{i,j})$ is simulated using a standard technique reminiscent of a Schnorr protocol. The simulator works backward: it first randomly samples the challenges $\{c_j\}$ and the responses $\{\zeta_{i,j}\}$ for the honest parties, and then computes the corresponding commitments $\{\gamma_{i,j}\}$, where each element is defined as $\gamma_{i,j} \coloneqq (\alpha_j \beta_j^{-1})^{\zeta_{i,j}} \cdot y_{i,j}^{-c_j}$. Notably, the simulator then programs the ideal functionality $\mathcal{F}_{\text{CT}}$ to output these exact challenges $\{c_j\}$. The indistinguishability of this simulation from a real execution is formally guaranteed. The simulated values $y_{i,j}$ are shown to be statistically close to the real ones by Proposition 6 and Proposition 8. Similarly, Proposition 7 ensures that the distribution of the simulated tuple $(\zeta_{i,j}, c_j, \gamma_{i,j})$ is indistinguishable from its real-world counterpart.

Case 2: The functionality's output is `NonBlumInteger` (Abort Case). In this case, we only need to argue for correctness. That is, if the inputs provided by the adversary do not form a valid Blum integer, the protocol will abort with overwhelming probability. The protocol's correctness against a malicious adversary builds upon the semi-honest case. The underlying Lucas test, as per Theorem 2, already ensures that a non-Blum integer N passes the primality check in Step 7 with a negligible probability of at most $\left(\frac{1}{4} + \frac{1.25}{p_{\min}-3}\right)^{\kappa_{\min}}$. Furthermore, the protocol is strengthened by the zero-knowledge proof in Step 10, which detects malicious deviations. As described, even if an adversary crafts inputs to pass the check in Step 7 by introducing a non-zero offset $\delta_{1,j} \neq 0$, they are challenged in Step 10. As the adversary must commit to their values before receiving the random challenge c_j, the probability of passing the Step 10 check with a non-zero offset is at most $1/2$. The combination of these independent checks ensures that the overall probability of an adversary successfully deceiving the protocol with an invalid modulus is negligible, specifically bounded by $2^{-\kappa}$. □

4.4 A Leakage-Free Variant via Constrained Prime Generation

To eliminate the information leakage from the interactive search for parameter D in the general Lucas protocol, we propose a more constrained but highly

efficient alternative. This method reverses the generation logic: instead of finding a suitable D for given primes, we first fix a small, public odd prime D. We then impose specific congruence constraints on the prime generation process itself. This ensures that any resulting primes p and q automatically satisfy the required conditions, $\left(\frac{-D}{p}\right) = \left(\frac{-D}{q}\right) = -1$, by construction. This approach completely obviates the need for a potentially leaky, interactive search for D, albeit at the cost of sacrificing some generality in prime selection.

Specifically, for a fixed odd prime D, let a be a quadratic non-residue modulo D. If we consider a prime p such that $p \equiv 1 \pmod 4$ and $p \equiv a \pmod D$, then we can verify that the required condition $\left(\frac{-D}{p}\right) = -1$ holds as follows:

$$\left(\frac{-D}{p}\right) = \left(\frac{-1}{p}\right)\left(\frac{D}{p}\right) = \left(\frac{D}{p}\right) = \left(\frac{p}{D}\right)(-1)^{(D-1)(p-1)/4} = \left(\frac{a}{D}\right) = -1.$$

Impact on Prime Density. A crucial consideration is whether constraining p to specific residue classes modulo $4D$ significantly hinders the prime generation process. We analyze this by comparing the density of primes within the constrained arithmetic progressions to that within a standard progression, such as $\{x \in \mathbb{N} \mid x \equiv 1 \pmod 4\}$.

Let $\pi(x; a, d)$ be the count of primes $p \leq x$ such that $p \equiv a \pmod d$. Let $S(x; a, d) \coloneqq \{a + nd < x \mid n \in \mathbb{N}\}$ be the set of numbers in the progression up to x; its size is approximately $|S(x; a, d)| = \lceil x/d \rceil$. The density of primes within this progression is approximately the ratio $\pi(x; a, d)/|S(x; a, d)|$. By the Dirichlet's theorem (i.e. $\pi(x; a, d) \approx x/(\ln x\phi(d))$ for $\gcd(a, d) = 1$), this density is approximately:

$$\frac{\pi(x; a, d)}{|S(x; a, d)|} \approx \frac{x/(\phi(d) \ln x)}{x/d} = \frac{d}{\phi(d) \ln x}$$

Now, we compare the two cases:

1. For the $a = 1, d = 4$, the density of primes within this progression is approximately $\frac{2}{\ln X}$.
2. For the case an odd prime D, $d = 4D$, and $\gcd(a, 4D) = 1$, the density of primes within this progression is approximately $\frac{2D}{(D-1)\ln X}$.

Therefore, the ratio is $\frac{2D/(D-1)}{2} = \frac{D}{D-1} \geq 1$. Our analysis shows that valid progressions modulo $4D$ have a slightly higher prime density than those modulo 4. Consequently, restricting the prime search to these specific residue classes $a \pmod{4D}$ slightly improves the probability of a random candidate being prime.

Realizing Constrained Share Generation in MPC. We now sketch the realization of Functionality 3. To generate secret-shared primes $p = \sum \mathfrak{p}_i$ and $q = \sum \mathfrak{q}_i$ satisfying target congruences modulo a value d, we employ a standard MPC technique: a designated party $\mathcal{P}_1$ sets its share to the target residue (e.g., $\mathfrak{p}_1 \equiv a \pmod d$), while all other parties $\mathcal{P}_i$ $(i \neq 1)$ ensure their shares are multiples of

the modulus ($\mathfrak{p}_i \equiv 0 \pmod{d}$). While commonly used for conditions like $p, q \equiv 3 \pmod 4$, our protocol leverages this principle for the more general congruences $p \equiv a \pmod{4D}$ and $q \equiv b \pmod{4D}$, where a, b are quadratic non-residues modulo D. All shares are sampled within $[0, 2^\ell)$, and assuming $2^\ell \gg 4D$, this method is efficient and the resulting distribution is statistically close to uniform.

Functionality 3 Constrained Share Generation and Product Computation

Inputs:

- **Common Inputs:** A public odd prime D, target residues $a, b \in \mathbb{Z}_{4D}$, and a bit-length parameter ℓ.
- **Party Inputs:** Each party $\mathcal{P}_i$ implicitly provides randomness.

Functionality:

1. The functionality internally generates shares $(\mathfrak{p}_i, \mathfrak{q}_i)$ for each party $\mathcal{P}_i$ $(i = 1, \ldots, n)$, ensuring each $\mathfrak{p}_i, \mathfrak{q}_i \in [0, 2^\ell)$.
2. It guarantees that the sums $p = \sum \mathfrak{p}_i$ and $q = \sum \mathfrak{q}_i$ satisfy the target congruences $p \equiv a \pmod{4D}$ and $q \equiv b \pmod{4D}$.
3. It computes the product $N = p \cdot q$.
4. Each party $\mathcal{P}_i$ receives its individual shares $(\mathfrak{p}_i, \mathfrak{q}_i)$ as private output and N as common output.

This functionality is realized by composing two standard MPC building blocks. First, parties execute the constrained share generation protocol as previously described, whereby each party $\mathcal{P}_i$ privately obtains its shares $(\mathfrak{p}_i, \mathfrak{q}_i)$. Second, all parties engage in a standard secure multiplication protocol on their respective shares to compute the product $N = (\sum \mathfrak{p}_i) \cdot (\sum \mathfrak{q}_i)$. The resulting value N is then opened and becomes the common output for all parties.

5 Performance Modeling and Simulation-Based Analysis

Having established the broader applicability of our Lucas-based test, this section aims to illuminate the factors governing practical performance in the common scenario of generating Blum integers. Our goal is not to declare a universal winner, but to explore the nuanced trade-offs between theoretical soundness and computational efficiency. To this end, our evaluation unfolds in three stages: we first establish the soundness landscape (Section 5.1), then filter out the variant Miller-Rabin (vMR) test, which, despite a comparable worst-case soundness, is shown to be non-competitive due to its significant protocol overhead (Section 5.2), and finally resolve the subtle trade-off between the BF and Lucas tests using a holistic performance model (Section 5.3).

5.1 Theoretical and Empirical Comparison of Soundness Errors

To substantiate the soundness advantage outlined in Section 3, this section provides a detailed theoretical and empirical analysis. We begin by deriving the precise algebraic ratio of the two soundness errors, which we then use to establish the conditions for superiority before validating our findings with experimental data.

Algebraic Ratio Analysis. We begin our theoretical analysis by formalizing the comparison between the Boneh-Franklin and Lucas tests. We analyze the ratio of the two soundness errors, $\beta_{\text{Lucas}}/\beta_{\text{BF}}$, for a given modulus N. Let $\mathbf{1}_{\mathbb{P}}(\cdot)$ be the indicator function of positive integers. Based on the counting formulas established in Proposition 1 and Lemma 2, this ratio can be expressed as:

$$\frac{\beta_{\text{Lucas}}(N, e_4)}{\beta_{\text{BF}}(N, e_4)} = \left(\frac{\prod_{i=1}^{s}(\gcd(e_4, d_i) - 1) + \prod_{i=1}^{s} \gcd(e_4, d_i)}{2 \prod_{i=1}^{s} \gcd(e_4, d_i)} \right) \cdot \left(\frac{\prod_{i=1}^{s}(p_i - 1)}{\prod_{i \in S_1^c}(p_i - 2) \left(\prod_{i \in S_1}(p_i - 2) + \mathbf{1}_{\mathbb{P}}(|S_1|)(-1)^{|S_1|} \right)} \right). \quad (7)$$

The first term in (7), involving the GCDs, is always less than or equal to 1/2 if any $\gcd(e_4, d_i) = 1$, and approaches 1 only when all $\gcd(e_4, d_i)$ are large. The second term, involving the prime factors, is typically slightly greater than 1. Our theoretical findings regarding this ratio are summarized in Table 2.

Table 2. Theoretical Comparison of Soundness Errors

Case Analysis	GCD Conditions	Other Conditions	Conclusion
Favorable Case (Lemma 9)	$\exists i$ s.t. $\gcd(d_i, e_4) = 1$	$p_{\min} \geq 467, \log_2(N) \leq 2048$	$\beta_{\text{Lucas}} < \beta_{\text{BF}}$
Worst-Case Scenario (Lemma 10)	$\forall i, d_i \mid e_4$	$\forall i, p_i \equiv 3 \pmod 4, p_i > 3$	$\beta_{\text{Lucas}} \geq \beta_{\text{BF}}$

The algebraic structure of this ratio reveals a clear dichotomy, which we formalize in the following two lemmas. Lemma 9 demonstrates that the Lucas test is superior in the common case where at least one GCD term is trivial.

Lemma 9. *Let p, q be two positive integers with $p \equiv q \equiv 3 \pmod 4$, and $N = \prod_{i=1}^{s} p_i^{r_i}$ be the product of p, q such that there exists $\gcd(d_i, e_4) = 1$ for some $1 \leq i \leq s$. Here $p_i - 1 = 2^{k_i} d_i$ with $2 \nmid d_i$ for all $1 \leq i \leq s$. Assume that $p_{\min}$ is the minimal prime factor of N. For all non-RSA moduli N with $2s < p_{\min} - 3$, we have:*

$$\beta_{\text{Lucas}}(N, e_4) < \beta_{\text{BF}}(N, e_4).$$

Proof. If there exists $1 \leq i \leq s$ such that $\gcd(e_4, d_i) = 1$, then the ratio in (7) is dominated by the first term. Observe that Bernoulli's inequality implies that:

$$\prod_{i \in S} \left(1 - \frac{1}{p_i - 1}\right) + \mathbf{1}_{\mathbb{P}}(|S_1|)(-1)^{|S_1|} \prod_{i \in S_1^c} \left(1 - \frac{1}{p_i - 1}\right) \prod_{i \in S_1} \frac{1}{p_i - 1}$$
$$> \left(1 - \frac{1}{p_{\min} - 1}\right)^s - \frac{1}{p_{\min} - 1} > 1 - \frac{s+1}{p_{\min} - 1}.$$

This estimation implies that if $1 - \frac{s+1}{p_{\min}-1} > \frac{1}{2}$, which is equivalent to $2s < p_{\min} - 3$, then $\beta_{\text{Lucas}}(N, e_4) < \beta_{\text{BF}}(N, e_4)$. □

As a concrete consequence, when $p_{\min} = 467$ (consistent with our experimental parameters), it can be ensured that for N satisfying the conditions of Lemma 9 and having a bit length of up to 2048 bits, the Lucas test always provides a tighter bound than the BF test.

Conversely, Lemma 10 identifies the rare, adversarial conditions under which the Boneh-Franklin test might perform as well as or better than the Lucas test.

Lemma 10. *Let p, q be two positive integers with $p \equiv q \equiv 3 \pmod 4$, and $N = \prod_{i=1}^{s} p_i^{r_i}$ be the product of p, q such that $p_i > 3$, $\gcd(e_4, p_i - 1) = d_i$, and $p_i \equiv 3 \pmod 4$ for all $1 \leq i \leq s$. For any non-perfect square N except for $s = |S_1| = 2$, we have $\beta_{\text{Lucas}}(N, e_4) > \beta_{\text{BF}}(N, e_4)$. In particular, if $s = |S_1| = 2$, then $\beta_{\text{Lucas}}(e_4) = \beta_{\text{BF}}(N, e_4)$.*

Proof. Let $p_i - 1 = 2^{k_i} d_i$. The assumptions imply $k_i = 1$ and $\gcd(e_4, d_i) = d_i$ for all i. The ratio of soundness errors is:

$$\frac{\beta_{\text{Lucas}}(N, e_4)}{\beta_{\text{BF}}(N, e_4)} = \frac{\prod_{i=1}^{s}(p_i - 1 - 2) + \prod_{i=1}^{s}(p_i - 1)}{2\left(\prod_{i=1}^{s}(p_i - 2) + \mathbf{1}_{\mathbb{P}}(|S_1|)(-1)^{|S_1|} \prod_{i \in S_1^c}(p_i - 2)\right)}.$$

Letting $x_i \coloneqq p_i - 2 > 1$, the condition $\beta_{\text{Lucas}} > \beta_{\text{BF}}$ is equivalent to proving $e_{s-2} + e_{s-4} + \ldots > \mathbf{1}_{\mathbb{P}}(|S_1|)(-1)^{|S_1|} \prod_{i \in S_1^c} x_i$, where e_j is the elementary symmetric polynomial.

For $s \geq 3$, we analyze by cases on $|S_1^c|$. The cases where $|S_1^c| \leq s - 2$ hold directly, as e_{s-2} is strictly greater than the magnitude of the RHS. The non-trivial case $|S_1^c| = s - 1$ is impossible under the lemma's premises: it implies $|S_1| = 1$, which would force $N = \prod p_i^{r_i} \equiv 3 \pmod 4$ since all $p_i \equiv 3 \pmod 4$. This contradicts the premise $N = pq \equiv 1 \pmod 4$. Thus, the inequality holds for all valid non-RSA moduli. □

Empirical Validation To empirically validate our theoretical analysis, we conducted an extensive search over all non-RSA moduli $N = pq$ with p, q up to $1,440,003$. The results are summarized in Table 3.

As predicted by our theoretical analysis, the data show that in over 99.99% of cases, the Lucas test offers a strictly lower soundness error than the BF test

Table 3. Pairwise comparison of soundness errors across three tests.

Method	$\beta = \frac{\beta_{\text{BF}}}{\beta_{\text{MR}}}$	$\beta = \frac{\beta_{\text{BF}}}{\beta_{\text{Lucas}}}$	$\beta = \frac{\beta_{\text{Lucas}}}{\beta_{\text{MR}}}$
$\beta < 1$	0.08%	$< 0.01\%$	54.26%
$\beta = 1$	54.18%	0%	0%
$\beta > 1$	45.74%	$> 99.99\%$	45.74%

Count of non-RSA moduli $N = pq$ (with $3 \leq p < q \leq 1440003$, $p \equiv q \equiv 3 \pmod 4$, etc.) satisfying the ratio conditions. See main text for full details.

($\beta_{\text{BF}}/\beta_{\text{Lucas}} > 1$). The absence of observed cases where $\beta = 1$ is also consistent with our theory; since our experiment enforces $p_{\min} \geq 541$, any boundary case would require an extremely large and specifically structured modulus $N > p_{\min}^3 \approx 1.58 \times 10^8$, which was not encountered in our search space. This empirical result provides overwhelming evidence that the "Favorable Case" identified in our theory is not a mere special condition, but the prevailing reality.

5.2 Computational Cost Comparison of Three Tests

To evaluate the efficiency of our protocol, we compare its per-iteration cost against the BF protocol as presented in [23] and the vMR test from [14], focusing on the common case of generating Blum integers. In this setting, our Lucas test is nearly identical in cost to the BF test; both operate in the same group ($\mathbb{Z}_N^\times$ since $D = 1$) and rely on a single, lightweight MPC shuffle protocol.

Both tests, however, are demonstrably more efficient than the vMR. As detailed in Table 4, the vMR protocol is substantially more complex. A detailed analysis of local computation reveals that the two $|N|/2$-bit exponentiations required by vMR are, in aggregate, theoretically faster than the single $|N|$-bit exponentiation in our test and the BF test. However, this local performance gain is rendered insignificant by the protocol's overall cost. The vMR test's reliance on multiple heavyweight MPC subroutines, including the expensive "Divisible" macro, creates a communication and computation bottleneck that far outweighs any local efficiency advantage. While the worst-case soundness errors of our Lucas test and the vMR test are practically indistinguishable for realistic parameters (both converging to 1/4), the deciding factor becomes protocol efficiency. Here, our test's simplicity and lower overhead make it the unequivocal alternative to the vMR for applications requiring provable security. This result allows us to focus the final, most nuanced performance analysis on the head-to-head competition between our Lucas test and the BF test. For applications requiring provable worst-case security, both the BF test and our Lucas test represent demonstrably more efficient alternatives to the vMR protocol.

Table 4. Single Execution Comparison for Three Tests

Method	BF test	vMR test	Proposed test
Basis selection	$g \leftarrow \mathbb{Z}_N$ s.t. $(\frac{g}{N}) = 1$	$v \leftarrow \mathbb{Z}_N$	$P \leftarrow \mathbb{Z}_N$ s.t. $(\frac{P^2-1}{N}) = 1$
Exponential test (local)	$\mathcal{P}_1 : g^{\frac{N-(\mathtt{p}_i+\mathtt{q}_i)+1}{4}}$ $\mathcal{P}_i : g^{\frac{-\mathtt{p}_i-\mathtt{q}_i}{4}}$	$\mathcal{P}_1 : v^{\frac{\mathtt{f}_i-1}{2}}$ $\mathcal{P}_i : v^{\frac{\mathtt{f}_i}{2}}$	$\mathcal{P}_1 :$ $(\frac{p+1}{p-1})^{\frac{N-(\mathtt{p}_i+\mathtt{q}_i)+1}{4}}$ $\mathcal{P}_i : (\frac{p+1}{p-1})^{\frac{-\mathtt{p}_i-\mathtt{q}_i}{4}}$
Exponential test (MPC)	$g^{e_4} \leftarrow$ Shuffle	Mul-to-Add $[v^{\frac{f-1}{2}}] \leftarrow$ Divisible $[y_{+1}y_{-1}] \leftarrow$ Mult	$(\alpha\beta^{-1})^{e_4} \leftarrow$ Shuffle
GCD test (MPC)	$[r] \leftarrow$ Sample $[r(p+q-1)] \leftarrow$ Mult	*None*[1]	$[r] \leftarrow$ Sample $[r(p+q-1)] \leftarrow$ Mult

Note: The calculations of the vMR test need to be executed for both $f = p$ and $f = q$. **Basis selection** refers to the conditions of the basis for exponential calculations. In **Exponential test (MPC)**, the Shuffle protocol outputs the product of shares. Mul-to-Add is the conversion of multiplicative to additive shares. The output of Divisible $y_{\pm 1}$ indicates if $v^{(f\pm 1)/2} \equiv 0 \pmod f$. Mult denotes MPC multiplication. Sample outputs a random element.
[1] The Variant Miller-Rabin test needs to confirm that $p \neq q$.

5.3 Performance Modeling and Evaluation

Now, we resolve the subtle performance trade-off between our Lucas test and the BF test. We employ a holistic performance model, clarifying that this evaluation is a heuristic analysis, not a definitive benchmark. The model is parameterized by local computation costs from a single-machine simulation to provide a conservative estimate of performance trends.

A Holistic Performance Model. To deliver a conclusive verdict, our evaluation distinguishes between provable security (the number of iterations needed, governed by worst-case soundness) and practical efficiency (the average time to reject invalid candidates, which dominates the key generation process). Our tighter 1/4 bound for the BF test significantly closes the gap in the first aspect, making their required iteration counts far more comparable. This shifts the focus to the second aspect: practical rejection speed.

The dominant factor in real-world performance is the speed at which the numerous non-RSA moduli are filtered out. The metric is thus the expected time to reject an invalid modulus, which reveals a crucial trade-off: the theoretical soundness of a test must be weighed against its per-iteration cost. To quantify this, we introduce a performance model that integrates these components. The expected time to reject an invalid modulus N, denoted $E[T(N)]$, can be

expressed as:

$$E\left[T(N)\right] = (1 - \beta(N))^{-1} \times (T_{\text{local}} + T_{\text{net}}) \tag{8}$$

where $\beta(N)$ is the single-round soundness error, T_{local} is the local computation time, and T_{net} is the network latency. This model provides a key insight: since network latency (T_{net}) constitutes a large and often dominant overhead for both protocols, even a small, consistent advantage in local computation time (T_{local}) can have a decisive impact on the overall average rejection speed.

Benchmarking Local Computation Cost. To obtain a precise, noise-free parameterization of T_{local} for our model, we deliberately employ a single-machine simulation to isolate computational costs from the highly variable effects of network latency. Our SageMath implementation simulated 2, 3, and 4-party protocols for 2048-bit and 3072-bit moduli. All tests ran single-threaded on a MacBook Pro (Apple M2, 16GB RAM). The resulting average execution times, which represent our empirical values for T_{local}, are presented in Table 5.

Table 5. The mean ± standard deviation of execution time (in milliseconds) for our methods and the competing method.

	N : 2048 bits		N : 3072 bits	
	Proposed test	Boneh-Franklin	Proposed test	Boneh-Franklin
$n = 2$	3.199 ± 0.251	3.145 ± 0.124	12.320 ± 0.462	12.253 ± 0.574
$n = 3$	4.55 ± 0.35	4.367 ± 0.253	17.310 ± 0.906	16.907 ± 0.588
$n = 4$	6.151 ± 0.329	5.992 ± 0.404	21.386 ± 1.006	21.300 ± 1.146

Threshold Analysis and Typical-Case Distribution. With the local computation costs (T_{local}) benchmarked, we can now use our performance model to deliver a final verdict. To do so, we analyze the condition $E\left[T_{\text{Lucas}}(N)\right] < E\left[T_{\text{BF}}(N)\right]$, which, after rearranging, shows that the Lucas test outperforms the BF test precisely when:

$$\beta_{\text{BF}}(N) > \frac{T_{\text{local,Lucas}} - T_{\text{local,BF}}}{(T_{\text{net}} + T_{\text{local,Lucas}}) - (T_{\text{net}} + T_{\text{local,BF}}) \left(\frac{\beta_{\text{Lucas}}(N)}{\beta_{\text{BF}}(N)}\right)}.$$

To parameterize the model, we first establish the constants. For the network cost, we adopt the standard heuristic of $T_{\text{net}} = 100$ ms for a typical WAN scenario. For the soundness error ratio, we maintain our conservative assumption of $\beta_{\text{Lucas}}/\beta_{\text{BF}} = 1/2$, which minimizes the soundness gap in our model. Next, we incorporate our benchmark results from Table 3. To determine a realistic threshold from our data, we identify the case with the smallest observed local computation time difference, which occurs for the $n = 2$, 2048-bit modulus: $T_{\text{local,Lucas}} - T_{\text{local,BF}} = 0.054$ ms. Substituting these empirical values into our model yields a critical threshold of $\beta_{\text{BF}} > 0.00105$.

However, to construct an even more rigorous "stress-test" scenario, we deliberately adopt a threshold that is an order of magnitude more favorable to the Lucas test. We therefore select a significantly lower threshold of 0.0002. This choice ensures that our final verdict is based on conditions that are exceptionally biased against the BF test. This leads to the condition that the BF test is slower only if:

$$\beta_{\mathrm{BF}} > (5000)^{-1} = 0.0002. \tag{9}$$

The conclusion that the Boneh-Franklin test maintains a performance advantage hinges on the observation that its soundness error, β_{BF}, is typically far smaller than this threshold. We support this observation with both experimental data and a compelling theoretical heuristic. Our experiments reveal that over 87% of non-RSA moduli within our testable range yield a β_{BF} value that does not satisfy this inequality. Our search covered all non-RSA moduli $N = pq$ constructed from factors $7 \leq p, q \leq 1,440,003$, where the minimal prime factor of any tested N exceeds 877.

The Rarity of Malicious Alignment. This empirical result is not a coincidence but is underpinned by a robust theoretical explanation. For β_{BF} to approach its worst-case bound, a form of rare "arithmetic alignment" must occur. Specifically, for a non-RSA modulus $N = \prod p_i^{r_i}$, the terms $\gcd(e_4, p_i - 1)$ must be simultaneously large for many prime factors p_i. This requires an improbable alignment between the prime factors of each $p_i - 1$ and those of the globally constructed value e_4.

An excellent analogy for such an alignment is the existence of Carmichael numbers. By Korselt's criterion, a square-free composite N is a Carmichael number if and only if it satisfies the stringent divisibility condition $p - 1 \mid N - 1$ for every prime factor p of N. As shown by Erdős [22], Carmichael numbers are extraordinarily rare, confirming their vanishingly small density.

The condition for a large β_{BF} is conceptually similar, though weaker. The value of e_4 is determined by the adversary's inputs (e.g., $p = p_1p_2, q = p_3p_4$), resulting in an expression like $e_4 = \frac{(p_1p_2-1)(p_3p_4-1)}{4}$. For $\gcd(e_4, p_i-1)$ to be large, the prime factorization of $p_i - 1$ must fortuitously overlap in a significant way with the prime factorization of e_4. While these values are algebraically related, their relationship is complex and non-obvious. There is no a priori reason to expect a structural alignment that systematically produces large GCDs.

Just as the stringent condition $p - 1 \mid N - 1$ makes Carmichael numbers exceedingly rare, we argue that the analogous requirement for multiple, simultaneous large GCDs makes the moduli N that maximize β_{BF} equally exceptional. This provides a strong theoretical basis for our empirical findings: for any typical non-RSA modulus, which lacks this malicious arithmetic alignment, the GCD terms are overwhelmingly likely to be small, suppressing the value of β_{BF} far below the threshold.

Final Verdict. Although the Lucas test has superior soundness, its expected gain from saving iterations is too infrequent to offset the certain, accumulated cost

Table 6. A Comparative Overview of Three Biprimality Tests

Feature	**Boneh-Franklin**	**Miller-Rabin**	**Lucas-based Test**
Worst-Case Soundness[a]	$1/2 \to 1/4$	$1/4$	$1/4 + \frac{1.25}{p_{\min}-3}$
Exceptional Cases	$p = q = 3$	$p, q \leq 9$	$p_{\min} < 11$
Extra Assumption	$\gcd(pq, e_4) = 1$	p, q equal-length	$\gcd(pq, e_4) = 1$
Empirical Soundness	*3*	*2*	*1*
MPC Efficiency[*]	*1*	*3*	*1*
Local Computation[*]	*1*	*3*	*2*
Leakage	No	No	None / Bounded[b]
RSA Moduli Type	Blum only	Blum only	Arbitrary

[*] Rankings where 1 is best. Comparisons are in the semi-honest model.
[a] The worst-case soundness error, excluding the exceptional cases listed below.
[b] The protocol is completely leak-free for Blum integers, while for non-Blum integers, it offers both leak-free and bounded-leakage variants (see Section 4.2).

of its slightly higher local computation per round, making the BF test faster in practice. It confirms that the cryptographic worst-case scenarios, while theoretically necessary to analyze, correspond to numbers with an extreme degree of arithmetic alignment that are, like Carmichael numbers, statistically insignificant in practice.

We therefore conclude with high confidence that the BF test performs better under realistic network conditions. This verdict is robust and even strengthens in lower-latency environments, where the BF test's slight T_{local} advantage is amplified, making its practical superiority all the more decisive.

6 Conclusion

In this work, we have fundamentally revisited the Boneh-Franklin (BF) biprimality test. A key contribution of this work was a rigorous proof tightening its worst-case soundness bound from 1/2 to a tight 1/4, a result of immediate practical significance that halves the required iterations for many existing protocols. We also introduced a more versatile test based on Lucas sequences, which provides the first efficient and provably secure solution for generating all standard RSA moduli in a distributed setting, not just Blum integers.

Our comparative analysis, however, reveals a key and subtle insight. Through a holistic performance model, we demonstrated that for the common case of generating Blum integers, the extreme local computation efficiency of the BF test often outweighs the superior empirical soundness of the Lucas test. This

provides a data-driven verdict for practitioners: while our Lucas test expands the design space, the newly-fortified BF test remains the superior practical choice in its original domain. This work thus provides a clearer toolkit for designers and opens avenues for future research. A particularly compelling direction is the potential creation of a distributed analogue to the highly robust Baillie-PSW test. By pairing our provably secure Lucas protocol with the BF test within an MPC framework, it is possible to construct a composite biprimality test of exceptional strength and efficiency. Verifying this possibility remains an intriguing open question.

Acknowledgements

We thank the anonymous reviewers for their valuable feedback and Yu-Te Lin (AMIS) for supporting this research. We also thank Gemini 2.5 Pro for its assistance in refining the language and enhancing the fluency of the manuscript.

References

1. Abadi, A., Ristea, D., Murdoch, S.J.: Delegated time-lock puzzle. arXiv preprint arXiv:2308.01280 (2023) 1
2. Algesheimer, J., Camenisch, J., Shoup, V.: Efficient computation modulo a shared secret with application to the generation of shared safe - prime products. In: Yung, M. (ed.) Advances in Cryptology — CRYPTO 2002. pp. 417 – 432. Springer Berlin Heidelberg, Berlin, Heidelberg (2002) 1
3. Apostol, T.M.: Introduction to Analytic Number Theory. Springer-Verlag, New York (1976). https://doi.org/10.1007/978-1-4757-5579-4 7.2
4. Apostol, T.M.: Introduction to analytic number theory. Springer Science & Business Media (2013) 1, 2, 15, 16
5. Arnault, F.: The Rabin-Monier theorem for Lucas pseudoprimes. Math. Comput. **66**, 869–881 (04 1997). https://doi.org/10.1090/S0025-5718-97-00836-3 1, 3.2, 4, 7.3, 7.3, 3
6. Benaloh, J., de Mare, M., Accumulators, O.W.: A decentralized alternative to digital signatures. In: Advances in Cryptology-Proceedings of Eurocrypt. vol. 93 (1994) 1
7. Benaloh, J.: Secret sharing homomorphisms: keeping shares of a secret secret. In: Proceedings on Advances in cryptology—CRYPTO '86. vol. LNCS 263, pp. 251–260 (01 1987) 7.11
8. Boneh, D., Bonneau, J., Bünz, B., Fisch, B.: Verifiable delay functions. In: Annual international cryptology conference. pp. 757–788. Springer (2018) 1
9. Boneh, D., Bünz, B., Fisch, B.: Batching techniques for accumulators with applications to IOPs and stateless blockchains. In: Advances in Cryptology–CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2019, Proceedings, Part I 39. pp. 561–586. Springer (2019) 1
10. Boneh, D., Franklin, M.: Efficient generation of shared RSA keys. Journal of the ACM **48** (12 2001). https://doi.org/10.1145/502090.502094 1, 1, 2.1, 4.2, 7.11, 7.12, 7.12

11. Boneh, D., Shoup, V.: A Graduate Course in Applied Cryptography. Stanford University (2023), available at https://crypto.stanford.edu/~dabo/cryptobook/ 5
12. Boudabra, M., Nitaj, A.: A new RSA variant based on elliptic curves. Cryptography (2023). https://doi.org/10.3390/cryptography7030037 1.1
13. Buhler, J., Stevenhagen, P.: Algorithmic number theory. Lattices, number fields, curves and cryptography. Reprint of the 2008 hardback ed. Cambridge University Press (01 2011) 1
14. Burkhardt, J., Damgård, I., Frederiksen, T., Ghosh, S., Orlandi, C.: Improved distributed RSA key generation using the Miller-Rabin test. CCS '23: Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security pp. 2501–2515 (2023). https://doi.org/10.1145/3576915.3623163 1, 1, 5.2, 7, 7.4, 7.12, 7.12, 7.12, 7
15. Chen, M., Doerner, J., Kondi, Y., Lee, E., Rosefield, S., Shelat, A., Cohen, R.: Multiparty generation of an RSA modulus. Journal of Cryptology **35** (04 2022). https://doi.org/10.1007/s00145-021-09395-y 1, 1.1, 1.2, 4.3, 4.3, 4, 7.13, 7.14, 7.14
16. Chen, M., Hazay, C., Ishai, Y., Kashnikov, Y., Micciancio, D., Riviere, T., Shelat, A., Venkitasubramaniam, M., Wang, R.: Diogenes: Lightweight scalable RSA modulus generation with a dishonest majority. In: 2021 IEEE Symposium on Security and Privacy (SP). pp. 590–607. IEEE (2021) 1
17. Chvojka, P.: Private coin verifiable delay function. Cryptology ePrint Archive (2023) 1
18. Damgård, I., Mikkelsen, G.: Efficient, robust and constant-round distributed RSA key generation. In: TCC 2010. LNCS, vol. 5978, pp. 183–200. Springer (2010), dOI: 10.1007/978-3-642-11799-2_12 1, 1, 7.4
19. Damgård, I., Landrock, P., Pomerance, C.: Average case error estimates for the strong probable prime test. Mathematics of Computation - Math. Comput. **61**, 177–177 (09 1993). https://doi.org/10.2307/2152945 1, 1
20. Einsele, S., Paterson, K.: Average case error estimates of the strong Lucas test. Designs, Codes and Cryptography pp. 1–38 (01 2024). https://doi.org/10.1007/s10623-023-01347-w 1
21. Ephraim, N., Freitag, C., Komargodski, I., Pass, R.: Continuous verifiable delay functions. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 125–154. Springer (2020) 1
22. Erdős, P.: On pseudoprimes and Carmichael numbers. Publ. Math. Debrecen. 4 **(3-4)**, 201–206 (2022). https://doi.org/10.5486/PMD.1956.4.3-4.16 5.3
23. Frederiksen, T., Lindell, Y., Osheter, V., Pinkas, B.: Fast distributed RSA key generation for semi-honest and malicious adversaries. In: CRYPTO 2018. LNCS, vol. 10992, pp. 331–361. Springer (2018). https://doi.org/10.1007/978-3-319-96881-0_12 1, 5.2, 7
24. Friedman, O., Marmor, A., Mutzari, D., Scaly, Y.C., Spiizer, Y., Yanai, A.: Tiresias: Large scale, maliciously secure threshold Paillier. Cryptology ePrint Archive (2023) 1
25. Grassi, L., Rechberger, C., Rotaru, D., Scholl, P., Smart, N.: MPC-friendly symmetric key primitives. In: CCS '16: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. pp. 430–443 (10 2016). https://doi.org/10.1145/2976749.2978332 7.8
26. Hazay, C., Mikkelsen, G.L., Rabin, T., Toft, T., Nicolosi, A.A.: Efficient RSA key generation and threshold Paillier in the two-party setting. Journal of Cryptology **32**, 265–323 (2019) 1

27. Hoffmann, C., Hubáček, P., Kamath, C., Krňák, T.: (verifiable) delay functions from Lucas sequences. Cryptology ePrint Archive (2023) 1
28. Ireland, K., Rosen, M.I.: A classical introduction to modern number theory, vol. 84. Springer Science & Business Media (01 1990) 3.1, 4, 11, 7.6
29. Joye, M., Libert, B.: Efficient cryptosystems from 2 k-th power residue symbols. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 76–92. Springer (2013) 1.1
30. Katz, J.: On achieving the" best of both worlds" in secure multiparty computation. In: Proceedings of the thirty-ninth annual ACM symposium on Theory of computing. pp. 11–20 (2007) 7.5
31. Khedr, W.I., Khater, H.M., Mohamed, E.R.: Cryptographic accumulator-based scheme for critical data integrity verification in cloud storage. IEEE Access **7**, 65635–65651 (2019) 1
32. Malavolta, G., Thyagarajan, S.A.K.: Homomorphic time-lock puzzles and applications. In: Annual International Cryptology Conference. pp. 620–649. Springer (2019) 1
33. Monier, L.: Evaluation and comparison of two efficient probabilistic primality testing algorithms. Theoretical Computer Science **12**, 97–108 (09 1980). https://doi.org/10.1016/0304-3975(80)90007-9 7.4
34. Niven, I., Zuckerman, H.S., Montgomery, H.L.: An introduction to the theory of numbers. Wiley, New York, fifth edition. edn. (1991) 6, 7.6
35. Oded, G.: Foundations of Cryptography: Volume 2, Basic Applications. Cambridge University Press, USA, 1st edn. (2009) 7.5, 7.5
36. Pietrzak, K.: Simple verifiable delay functions. In: 10th innovations in theoretical computer science conference (itcs 2019). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik (2018) 1
37. Rabin, M.: Probabilistic algorithm for testing primality. Journal of Number Theory **12**, 128–138 (02 1980). https://doi.org/10.1016/0022-314X(80)90084-0 1
38. Rivest, R., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM **26**, 96–99 (01 1983). https://doi.org/10.1145/359340.359342 1
39. Rivest, R.L., Shamir, A., Wagner, D.A.: Time-lock puzzles and timed-release crypto. 1996 Technical Report (1996) 1
40. Delpech de Saint Guilhem, C., Makri, E., Rotaru, D., Tanguy, T.: The return of Eratosthenes: Secure generation of RSA moduli using distributed sieving. In: ACM CCS 2021. pp. 594–609 (2021) 1
41. Wesolowski, B.: Efficient verifiable delay functions. In: Advances in Cryptology–EUROCRYPT 2019: 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19–23, 2019, Proceedings, Part III 38. pp. 379–407. Springer (2019) 1

Appendix 7 Mathematical Tools and Supplementary Proofs and Protocols

This appendix provides supplementary material supporting our main results, organized as follows:

- **Mathematical Preliminaries:** We begin by reviewing the necessary mathematical background on Lucas sequences (Section 7.3), the Chinese Remainder Theorem (CRT) (Section 7.1), the Jacobi symbol (Section 7.2), and the key lemmas on statistical distance (Section 7.9) that underpin our security analyses.
- **Supplementary and Security Proofs:** We provide the missing technical lemmas and algebraic derivations that support the soundness analysis of our Lucas test (Section 7.6). Furthermore, we present the complete simulation-based security proofs for our protocols against semi-honest adversaries (Theorem 3 in Section 7.7) and malicious adversaries (Theorem 6 in Section 7.10).
- **Comparative and Supplementary Materials:** To provide broader context, we detail the analysis of the Miller-Rabin test (Section 7.4), reformulate protocols using the unified macro framework from Burkhardt et al. [14] (Section 7.12), and provide proofs for auxiliary components like the Legendre symbol protocol (Section 7.8) and our formal security model (Section 7.5).
- **Revisions and Extensions:** Finally, we offer a revised version of the Chen et al. protocol (Section 7.13) with a detailed AND gate cost analysis of its core π_{VerifyBI} circuit (Section 7.14).

7.1 The CRT and its applications

We start by illustrating the CRT and demonstrating some of its applications.

Lemma 11. [28, Theorem 3, Chapter 4] *Let $N = \prod_{i=1}^{s} p_i^{r_i}$ be the prime decomposition of N. Then we have the isomorphism f_{CRT} defined by*

$$\begin{aligned} f_{CRT} : \quad \mathbb{Z}_N^\times \quad &\longrightarrow \mathbb{Z}_{p_1^{r_1}}^\times \times \cdots \times \mathbb{Z}_{p_s^{r_s}}^\times \\ x \pmod N &\mapsto \big(x \pmod{p_1^{r_1}}, \ldots, x \pmod{p_s^{r_s}}\big). \end{aligned}$$

Here $\mathbb{Z}_{p_i^{r_i}}^\times$ is a cyclic group of order $p_i^{r_i-1}(p_i-1)$. $\mathbb{Z}_{2^a}^\times$ is cyclic of order 1 and 2 for $a = 1$ and 2, respectively. If $a \geq 3$, then it is the product of two cyclic groups, one of order 2, the other of order 2^{a-2}.

Lemma 12. *Let $p \equiv q \equiv 3 \pmod 4$, and $\gcd\big(pq, e_4\big) = 1$, where e_4 is equal to $(p-1)(q-1)/4$. Assume that $N \coloneqq pq = \prod_{i=1}^{s} p_i^{r_i}$. Then*

$$|\mathrm{BF}(N, e_4)| = \prod_{i=1}^{s} |\mathrm{BF}(p_i^{r_i}, e_4)|.$$

Proof. Through the map f_{CRT}, we only need to prove that $x^{e_4} \equiv 1 \pmod{N}$ if and only $x^{e_4} \equiv 1 \pmod{p_i^{r_i}}$ for all $1 \leq i \leq s$. This is directly proven due to the fact $N \mid x^{e_4} - 1$ if and only if $p_i^{r_i} \mid x^{e_4} - 1$ for all $1 \leq i \leq s$. □

Lemma 13. *Let D be an integer and $N \coloneqq \prod_{i=1}^{s} p_i^{r_i}$ be a positive integer with $\gcd(N, 2D) = 1$. Then, for any $\epsilon \in \{-1, 1\}$, one has*

$$|\mathcal{Z}^{\epsilon}(D, N)| = \sum_{\epsilon_1 \cdots \epsilon_s = \epsilon} \prod_{i=1}^{s} |\mathcal{Z}^{\epsilon_i}(D, p_i^{r_i})|.$$

Proof. It is clear that $P^2 - 4Q \equiv D \pmod{N}$ is solvable, then $P^2 - 4Q \equiv D \pmod{p_i^{r_i}}$ for all $1 \leq i \leq s$. Conversely, CRT says that if $P^2 - 4Q \equiv D \pmod{p_i^{r_i}}$ for all $1 \leq i \leq s$, then $P^2 - 4Q \equiv D \pmod{N}$. Moreover, $\gcd(Q, N) = 1$ if and only if $\gcd(Q, p_i^{r_i}) = 1$ for all $1 \leq i \leq s$. Consider the condition of the Jacobi symbol. As for the case N is non-square, given a fixed set $\epsilon_1, \ldots, \epsilon_s$ such that $\epsilon_1 \cdots \epsilon_s = \epsilon$, where $\epsilon_i \in \{-1, 1\}$ for all $1 \leq i \leq s$, the map f_{CRT} guarantees that for any Q_i with $\left(\frac{Q_i}{p_i^{r_i}}\right) = \epsilon_i$ for all $1 \leq i \leq s$, there exists a $Q \equiv Q_i \pmod{p_i^{r_i}}$ such that $\left(\frac{Q}{N}\right) = \epsilon$. The case square N is straightforward, since $\left(\frac{Q}{N}\right) = 1$ for all $\gcd(Q, N) = 1$. The proof is complete by the above discussion. □

Finally, we use the CRT to construct an infinite family of counterexamples to the BF test for the case where the condition $p \equiv q \equiv 3 \pmod{4}$ is relaxed.

Lemma 14. *Let p_1, p_2, p_3 be distinct prime numbers with $p_3 = (p_2 - 1)p' + 1$, and $p' \mid p_1 - 1$. Here p' is a prime number. Assume that $p = p_1$, $q = p_2 p_3$, $e_4 = (p-1)(q-1)/4$, and $N = pq$. Then for all $g \in \mathbb{Z}_N^{\times}$, we have*

$$g^{e_4} \equiv 1 \pmod{N}.$$

Proof. Since $p' \mid p - 1$, then for any $g \in \mathbb{Z}_N^{\times}$, we have

$$g^{e_4} \equiv g^{(p-1)(p_2 p_3 - 1)/4} \equiv 1 \pmod{p},$$

$$g^{e_4} \equiv g^{(p-1)(p_2-1)((p_2-1)p' + 1 + p')/4} \equiv 1 \pmod{p_2},$$

and

$$g^{e_4} \equiv g^{p'(p_2-1)((p-1)/p')((p_2-1)p' + 1 + p')/4} \equiv 1 \pmod{p_3}.$$

The above equalities, and Lemma 11 implies the desired result. □

To satisfy the $\gcd(N, e_4) = 1$ constraints, we first fix $a = p_2 - 1$ and choose a prime p' such that both p_2 and $p_3 = a \cdot p' + 1$ are prime. As long as there exists a pair (a, p') satisfying the above conditions, then by Dirichlet's theorem, there are infinitely many primes of the form $p_1 = 4kp' + 1$ such that: $p' \mid p_1 - 1$, $\gcd(a, p_1) = 1$, and $\gcd(p_1, ap' + 1 + p') = 1$, because a, p' have been fixed, which implies that the desired result $\gcd(N, e_4) = 1$. As a concrete example, we may simply choose $p_2 = 5$ and $p' = 3$.

7.2 The Jacobi Symbol and Related Consequences

This section lists the properties and some results of the Jacobi symbol.

Definition 1 (Legendre symbol). [4, Section 9.2, Theorem 9.2] *Let p be an odd prime. We define Legendre symbol $\left(\frac{a}{p}\right)$ as follows:*

$$\left(\frac{a}{p}\right) = a^{\frac{p-1}{2}} \pmod{p} = \begin{cases} 0 & \text{if } a \equiv 0 \pmod{p}; \\ +1 & \text{if } a \text{ is a quadratic residue modulo } p; \\ -1 & \text{if } a \text{ is a quadratic nonresidue modulo } p. \end{cases}$$

Definition 2 (Jacobi symbol). [4, Section 9.7] *If N is a positive odd integer with prime factorization $N = \prod_{i=1}^{s} p_i^{r_i}$, then the Jacobi symbol $\left(\frac{a}{N}\right)$ is defined for all integers a by the equation*

$$\left(\frac{a}{N}\right) = \prod_{i=1}^{s} \left(\frac{a}{p_i}\right)^{r_i},$$

where $\left(\frac{a}{p_i}\right)$ is the Legendre symbol.

Lemma 15. [4, Theorem 9.9, Theorem 9.10] *If N_1 and N_2 are odd positive integers, we have*

(a) $\left(\frac{ab}{N_1}\right) = \left(\frac{a}{N_1}\right)\left(\frac{b}{N_1}\right)$;
(b) $\left(\frac{a}{N_1 N_2}\right) = \left(\frac{a}{N_1}\right)\left(\frac{a}{N_2}\right)$;
(c) $\left(\frac{a}{N_1}\right) = \left(\frac{b}{N_1}\right)$ *whenever* $a \equiv b \pmod{N_1}$;
(d) $\left(\frac{c^2 b}{N_1}\right) = \left(\frac{b}{N_1}\right)$ *whenever* $\gcd(c, N_1) = 1$;
(e) $\left(\frac{-1}{N_1}\right) = (-1)^{(N_1-1)/2}$;
(f) $\left(\frac{-2}{N_1}\right) = (-1)^{(N_1^2-1)/8}$.

Lemma 16 (Reciprocity law for Jacobi symbols). [4, Theorem 9.11] *If N_1 and N_2 are two coprime odd numbers, then*

$$\left(\frac{N_1}{N_2}\right)\left(\frac{N_2}{N_1}\right) = (-1)^{\frac{(N_1-1)(N_2-1)}{4}}.$$

Lemma 17 (Restatement of Lemma 3). *Let $N = \prod_{i=1}^{s} p_i^{r_i}$ be a positive odd integer. Then, the cardinality of $G(N)$ is given by:*

$$|G(N)| = \begin{cases} \phi(N) & \text{if } N \text{ is a perfect square;} \\ \phi(N)/2 & \text{if } N \text{ is not a perfect square.} \end{cases}$$

Proof (Proof of Lemma 3). First, we prove the case where N is a perfect square. The proof is complete by the below observation

$$|G(N)| = \left|\left\{a \in \mathbb{Z}_N \middle| \left(\frac{a}{N}\right) = 1\right\}\right| = \left|\mathbb{Z}_N^\times\right| = \phi(N).$$

For the case where N is not a perfect square, we claim that there exists an $a \in \mathbb{Z}_N^\times$ such that $\left(\frac{a}{N}\right) = -1$. Without loss of generality, we assume r_1 is odd, which means that there exists $1 \leq a' < p_1$ such that $\left(\frac{a'}{p_1}\right) = -1$ (cf. [3, Theorem 9.1]). CRT implies that the integer a can be constructed by the following system of congruence equations. That is $a \equiv 1 \pmod{p_i}$ for all $i \geq 2$, and $a \equiv a' \pmod{p_1}$. Notice that

$$\left(\frac{a}{N}\right) \sum_{x \in \mathbb{Z}_N^\times} \left(\frac{x}{N}\right) = \sum_{x \in \mathbb{Z}_N^\times} \left(\frac{ax}{N}\right) = \sum_{x \in \mathbb{Z}_N^\times} \left(\frac{x}{N}\right),$$

which gives that $\sum_{x \in \mathbb{Z}_N^\times} \left(\frac{x}{N}\right) = 0$. Therefore, this equality gives us that the size of $G(n)$ is half of $\mathbb{Z}_N^\times$. □

7.3 Lucas Pseudo-primes

We recall Lucas sequence and some results [5]. Let P and Q be integers and $D \coloneqq P^2 - 4Q$. The Lucas sequence (U_k, V_k) that is associated with the parameters P, Q are defined as, for $k \geq 0$,

$$\begin{cases} U_{k+2} = PU_{k+1} - QU_k; \\ V_{k+2} = PV_{k+1} - QV_k, \end{cases} \quad \text{with} \quad \begin{cases} U_0 = 0, U_1 = 1; \\ V_0 = 2, V_1 = P. \end{cases}$$

Based on the primality criterion established (cf. [5, Theorem 1.2]), if p is a prime number such that $p \nmid 2QD$, then it holds that

$$U_{p-\left(\frac{D}{p}\right)} \equiv 0 \pmod{p}.$$

For the Lucas sequence [5, Section 3], (U_k, V_k) associated with P, Q and $P^2 - 4Q \neq 0$, we have the general formula: for all $k \in \mathbb{N}$,

$$U_k = \frac{\alpha^k - \beta^k}{\alpha - \beta},\ V_k = \alpha^k + \beta^k,$$

where α, β are two distinct roots of the polynomial $x^2 - Px + Q$. Let $\mathcal{O}_D$ be the quadratic order of discriminant D, and $\tau \coloneqq \alpha\beta^{-1}$. If $N \nmid 2QD$, then we have, for $k \in \mathbb{N}$,

$$N \mid U_k \text{ if and only if } \tau^k \equiv 1 \pmod{N\mathcal{O}_D}.$$

For any element $u + v\sqrt{D} \in \mathcal{O}_D$ (where $u, v \in \mathbb{Z}$), the norm map is given by $\mathbf{N}(u + v\sqrt{D}) = u^2 - v^2 D \in \mathbb{Z}$. Consider the multiplicative group of norm 1

elements denoted by $(\widehat{\mathcal{O}_D/N\mathcal{O}_D})$ in a free $\mathbb{Z}/N\mathbb{Z}$-algebra of rank 2. This group is the image of the set

$$\{x \in \mathcal{O}_D \mid \mathbf{N}(x) \equiv 1 \pmod{N}\}$$

by the canonical map $\mathcal{O}_D \to \mathcal{O}_D/N\mathcal{O}_D$.

Proposition 3. [5, Proposition 3.2] *Let D be a non-square integer and $N \coloneqq \prod_{i=1}^{s} p_i^{r_i}$ be a positive integer with $\gcd(N, 2D) = 1$. Then, for all integers P, there exists an integer Q, uniquely determined modulo N, such that $P^2 - 4Q \equiv D \pmod{N}$. Moreover, the set of integers P such that*

$$\begin{cases} 0 \leq P < N; \\ \gcd(P^2 - D, N) = \gcd(Q, N) = 1, \end{cases}$$

is in a one-to-one correspondence with the elements τ in $\left(\widehat{\mathcal{O}_D/N}\right)$ such that $\tau - 1$ is a unit in $\mathcal{O}_D/N$. Moreover, we have

$$|\mathcal{Z}(D, N)| = \prod_{i=1}^{s} p_i^{r_i - 1} \left(p_i - \left(\frac{D}{p_i}\right) - 1\right).$$

7.4 Three Biprimality Tests: Variant Miller-Rabin, Boneh-Franklin, Lucas Tests

We begin by recalling the variant Miller-Rabin test [14]. We then conduct a comparative analysis of the Boneh-Franklin test, the variant Miller-Rabin test, and our proposed Lucas test. By deriving explicit formulas for their respective soundness errors, we evaluate which test is superior in identifying non-RSA moduli. For this analysis, consider $N = pq$ with $p \equiv q \equiv 3 \pmod{4}$, and let $f \in \{p, q\}$. The algorithm for the variant Miller-Rabin test is as follows:

1. Uniformly sample an element $v \in \mathbb{Z}_N^{\times}$ [4] (i.e., in [14], v is chosen in $\mathbb{Z}_N$).
2. Compute $\gamma = v^{\frac{f-1}{2}} \pmod{N}$.
3. If $\gamma \equiv \pm 1 \pmod{f}$, then output **probably prime**. Otherwise output **composite**.

The biprimality test proposed in [14, 18] applies the variant Miller-Rabin test separately to $f \in \{p, q\}$. Therefore, for any $N = pq$ with $p \equiv q \equiv 3 \pmod{4}$ and $\gcd(N, e_4) = 1$ the probability that N passes the process is (cf. Lemma 18)

$$\beta_{\mathrm{MR}}(p) \coloneqq \frac{|\mathrm{MR}(p)|}{\phi(p)} = 2 \left(\prod_{p_i \mid p} \frac{\gcd(d_i, \frac{p-1}{2})}{p_i^{r_i - 1}(p_i - 1)}\right).$$

[4] We narrow the selection range of v from $\mathbb{Z}_N$ to $\mathbb{Z}_N^{\times}$ because an element $v \in \mathbb{Z}_N \backslash \mathbb{Z}_N^{\times}$ will let the test output composite even when f is prime.

Here $\mathrm{MR}(p) \coloneqq \{g \in \mathbb{Z}_p^\times \mid g^{\frac{p-1}{2}} \equiv \pm 1 \pmod p\}$. In particular, when $p = q$ is prime, such an RSA modulus candidate p, q will always pass this algorithm's test with 100% certainty. Therefore, we recommend incorporating a check to verify whether N is a perfect square to exclude this case.

For completeness, we provide the formula for the number of witnesses in the variant Miller-Rabin test. The proof follows a logic similar to Theorem 1 and represents a special case of [33, Proposition 1].

Lemma 18. *Let* $p = \prod_{i=1}^s p_i^{r_i} \equiv 3 \pmod 4$. *Then*

$$|\mathrm{MR}(p)| = 2 \prod_{i=1}^{s} \gcd\left((p-1)/2, d_i\right).$$

Proof. Since $(p-1)/2$ is odd, we have

$$|g \in \mathbb{Z}_p^\times \mid g^{(p-1)/2} \equiv 1 \pmod p| = |g \in \mathbb{Z}_p^\times \mid g^{(p-1)/2} \equiv -1 \pmod p|,$$

due to the bijective map $g \mapsto -g$. This implies that

$$|\mathrm{MR}(p)| = 2 \cdot |g \in \mathbb{Z}_p^\times \mid g^{(p-1)/2} \equiv 1 \pmod p|.$$

Similar to the proof of Lemma 2, we determine the number of $(p-1)/2$-th roots of unity in the group $(\mathbb{Z}/p_i^{r_i}\mathbb{Z})^\times$ using the CRT. Since p is odd, $(\mathbb{Z}/p_i^{r_i}\mathbb{Z})^\times$ is a cyclic group. Combining this fact with $\gcd(p, (p-1)/2) = 1$ and applying Lemma 1, the number of $(p-1)/2$-th roots of unity in each local group $(\mathbb{Z}/p_i^{r_i}\mathbb{Z})^\times$ is:

$$\gcd((p-1)/2, p_i^{r_i-1}(p_i-1)) = \gcd((p-1)/2, d_i).$$

The result follows from the multiplicativity property of the CRT. □

We provide the formulas to compare the tests pairwise and analyze their soundness ratios across three scenarios. These formulas form the basis for the comparative data presented in Table 3. Let $\mathbf{1}_{\mathbb{P}}(\cdot)$ be the indicator function of positive integers (i.e., $\mathbf{1}_{\mathbb{P}}(0) = 0$).

- **Variant Miller-Rabin vs. Boneh-Franklin Test:**

$$\frac{\beta_{\mathrm{BF}}(N, e_4)}{\beta_{\mathrm{MR}}(p)\beta_{\mathrm{MR}}(q)} = \left(\frac{1}{\mathbf{1}_{\mathbb{P}}(\sqrt{N}) + 1}\right) \left(\prod_{\substack{p_i \mid p \\ p_i \nmid q}} \frac{\gcd(e_4, d_i)}{\gcd(d_i, \frac{p-1}{2})}\right)$$
$$\cdot \left(\prod_{\substack{p_i \mid q \\ p_i \nmid p}} \frac{\gcd(e_4, d_i)}{\gcd(d_i, \frac{q-1}{2})}\right) \left(\prod_{p_i \mid \gcd(p,q)} \frac{(p_i - 1)\gcd(e_4, d_i)}{p_i \gcd(d_i, \frac{p-1}{2}) \gcd\left(d_i, \frac{q-1}{2}\right)}\right).$$

- **Lucas Test vs. Boneh-Franklin Test**

$$\frac{\beta_{\mathrm{Lucas}}(N, e_4)}{\beta_{\mathrm{BF}}(N, e_4)} = \left(\frac{\prod_{i=1}^s (\gcd(e_4, d_i) - 1) + \prod_{i=1}^s \gcd(e_4, d_i)}{2\prod_{i=1}^s \gcd(e_4, d_i)}\right)$$
$$\cdot \left(\frac{\prod_{i=1}^s (p_i - 1)}{\prod_{i \in S_1^c}(p_i - 2)\left(\prod_{i \in S_1}(p_i - 2) + \mathbf{1}_{\mathbb{P}}(|S_1|)(-1)^{|S_1|}\right)}\right).$$

- **Lucas Test vs. Variant Miller-Rabin Test**

$$\frac{\beta_{\text{Lucas}}(N,e_4)}{\beta_{\text{MR}}(p)\beta_{\text{MR}}(q)} = \left(\frac{\prod_{i=1}^{s}(\gcd(e_4,d_i)-1)+\prod_{i=1}^{s}\gcd(e_4,d_i)}{2\prod_{p_i|p}\gcd(\frac{p-1}{2},d_i)\prod_{p_i|q}\gcd(\frac{q-1}{2},d_i)}\right) \cdot\left(\frac{\left(\prod_{i=1}^{s}(p_i-1)\right)\left(\prod_{p_i|\gcd(p,q)}(1-p_i^{-1})\right)}{\prod_{i\in S_1^c}(p_i-2)\left(\prod_{i\in S_1}(p_i-2)+\mathbf{1}_{\mathbb{P}}(|S_1|)(-1)^{|S_1|}\right)}\right).$$

Here S_1^c is the complement of S_1, i.e., the set of indices i for which r_i is even.

7.5 The Security Model

Our security analysis considers static and rushing adversaries. In the static model, the adversary must choose the set of corrupted parties before the protocol execution commences and cannot alter this selection. Conversely, a rushing adversary can delay messages from corrupted parties within a given round until all messages from honest parties in that round have been received. We assume a standard communication model where n parties exchange messages in synchronized rounds via pairwise private and authenticated channels, and additionally have access to a broadcast channel. In the following definitions, we use $[n] \coloneqq \{1,\ldots,n\}$ and boldface symbols (e.g., $\boldsymbol{x}$) to denote vectors for conciseness.

Semi-honest adversaries follow the protocol specifications honestly but attempt to learn as much information as possible from the messages they receive from other parties. For this model, we adopt the definition provided in Goldreich [35, Definition 7.5.1], detailed as follows. Let $f : (\{0,1\}^*)^n \to (\{0,1\}^*)^n$ be an n-ary functionality, where $f_i(\boldsymbol{x})$ denotes the i-th element of $f(\boldsymbol{x})$. For $I \subseteq [n]$, let $f_I(\boldsymbol{x})$ denote the subsequence of $f(\boldsymbol{x})$ indexed by I. Let Π be an n-party protocol for computing f. The view of party $\mathcal{P}_i$ during an execution on input $\boldsymbol{x}$, denoted $\text{VIEW}_i^{\Pi}(\boldsymbol{x})$, is $(x_i, r_i, m_{i,1}, \ldots, m_{i,\ell})$, where r_i and $m_{i,j}$ are $\mathcal{P}_i$'s internal coins and j-th received message, respectively. For $I = \{i_1,\ldots,i_t\}$, let $\text{VIEW}_I^{\Pi}(\boldsymbol{x}) \coloneqq (I, \text{VIEW}_{i_1}^{\Pi}(\boldsymbol{x}), \ldots, \text{VIEW}_{i_t}^{\Pi}(\boldsymbol{x}))$.

Definition 3. *We say that Π t-privately computes f if there exists a probabilistic polynomial-time algorithm, denoted S, such that for every $I \subseteq \{1,\ldots,n\}$ with $|I| \le t$, it holds that*

$$\{(S(I,(x_{i_1},\ldots,x_{i_t}),f_I(\boldsymbol{x})),f(\boldsymbol{x}))\}_{\boldsymbol{x}\in(\{0,1\}^*)^n} \overset{\text{c}}{\equiv} \{(\ \text{VIEW}_I^{\Pi}(\boldsymbol{x}),\ \text{OUTPUT}^{\Pi}(\boldsymbol{x}))\}_{\boldsymbol{x}\in(\{0,1\}^*)^n}.$$

Here $\text{OUTPUT}^{\Pi}(\boldsymbol{x})$ denotes the output sequence of all parties during the execution represented in $\text{VIEW}_I^{\Pi}(\boldsymbol{x})$, and $\overset{\text{c}}{\equiv}$ is computationally indistinguishable of two distribution ensembles.

We consider security with abort against malicious adversaries according to the definition presented by Katz [30, Definition 1], a definition grounded in the framework of Goldreich [35]. In the malicious adversarial model, corrupted parties may deviate from the protocol specification in an arbitrary manner.

The Real Model. At the beginning of a real execution of a protocol, each party $\mathcal{P}_i$ holds the security parameter 1^λ and its input x_i. Let $f : (\{0,1\}^*)^n \to (\{0,1\}^*)^n$ be an $n-$ary functionality. The adversary $\mathcal{A}$ takes as input 1^λ, the set $I \subset \{1,\ldots,n\}$ of corrupted parties, the inputs of the corrupted parties, and an auxiliary input **aux**. The interaction of $\mathcal{A}$ with a protocol Π defines a random variable $\mathrm{REAL}_{\Pi,\mathcal{A},I}(\lambda, \vec{x}, \mathbf{aux})$ whose value is determined by the coin tosses of the adversary and the honest players. This random variable contains the output of the adversary (which may be an arbitrary function of its view) as well as the outputs of the uncorrupted parties. We let $\mathrm{REAL}_{\Pi,\mathcal{A},I}$ denote the distribution ensemble $\{\mathrm{REAL}_{\Pi,\mathcal{A},I}(\lambda, \vec{x}, \mathbf{aux})\}_{k\in\mathbb{N},(\vec{x},\mathbf{aux})\in\{0,1\}^*}$.

The ideal model. Here the parties interact with a trusted party implementing f. Each honest party $\mathcal{P}_i$ holds an input x_i as before; the adversary $\mathcal{A}'$ is again given 1^λ, the set I of corrupted parties, the inputs of all the corrupted parties, and an auxiliary input **aux**. Each honest party $\mathcal{P}_i$ sets $x'_i = x_i$ and sends x'_i to the trusted party; each corrupted party $\mathcal{P}_j$ sends an arbitrary input x'_j to the trusted party as directed by $\mathcal{A}'$. In case some corrupted party $\mathcal{P}_j$ does not send an input, x'_j is set to a default value. The trusted party computes $(y_1,\ldots,y_n) \leftarrow f(x'_1,\ldots,x'_n)$, choosing a uniformly random tape for f in case it is randomized. Then, the trusted party sends the outputs $\{y_i\}_{i\in I}$ to $\mathcal{A}'$ first. The adversary can then decide whether to abort the trusted party, or whether to allow it to continue. In the former case, the trusted party sends the special symbol $\perp$ to all honest parties as their output, where $\perp$ is assumed not to lie in the range of f. In the latter case, the trusted party sends the correct output y_i to each honest party $\mathcal{P}_i$.

The interaction of $\mathcal{A}'$ with the trusted party defines a random variable $\mathrm{IDEAL}_{f,\mathcal{A}',I}(\lambda, \vec{x}, \mathbf{aux})$ whose value is determined by the random coins of the adversary and those used by the trusted party in evaluating f. This random variable contains the output of $\mathcal{A}'$ (which may be an arbitrary function of its view) as well as the outputs of the uncorrupted parties. We let $\mathrm{IDEAL}_{f,\mathcal{A}',I}$ denote the distribution ensemble $\{\mathrm{IDEAL}_{f,\mathcal{A}',I}(\lambda, \vec{x}, \mathbf{aux})\}_{\lambda\in\mathbb{N},(\vec{x},\mathbf{aux})\in\{0,1\}^*}$.

Definition 4. *Let f be an n-party randomized functionality, and Π be an n-party protocol. Then Π* t-securely computes *f* with abort *if for any PPT adversary $\mathcal{A}$ there exists a PPT adversary $\mathcal{A}'$ such that for any $I \subseteq \{1,\ldots,n\}$ with $|I| \le t$:*

$$REAL_{\Pi,\mathcal{A},I} \stackrel{\mathrm{c}}{\equiv} IDEAL_{f,\mathcal{A}',I}.$$

7.6 Missing Proofs of Section 3.2

Lemma 19 (Restatement of Lemma 4). *Let p be an odd prime and $D \not\equiv 0 \pmod p$. Then*

$$\sum_{i=1}^{(p-1)/2} \left(\frac{i^2+D}{p}\right) = \frac{-1-\left(\frac{D}{p}\right)}{2}.$$

Proof. This is a classical result in number theory, often presented as an exercise in standard textbooks (e.g., [28, Chapter 5, Exercise 8]). For completeness, we provide a proof below. First, we prove that

$$\sum_{i=1}^{p}\left(\frac{i^2+D}{p}\right)=-1.$$

According to Euler's criterion (cf. [34, Theorem 3.1]), the above considering sum can be written as

$$\sum_{i=1}^{p}\left(\frac{i^2+D}{p}\right)\equiv\sum_{i=1}^{p}(i^2+D)^{\frac{p-1}{2}}\equiv\sum_{\ell=0}^{\frac{p-1}{2}}\binom{\frac{p-1}{2}}{\ell}D^{\ell}\sum_{i=1}^{p}i^{p-1-2\ell}\pmod p.$$

Recall that since $\mathbb{Z}_p^{\times}$ is a cyclic group, there exists a generator g, which induces that

$$\sum_{i=1}^{p-1}i^k\pmod p=\sum_{i=0}^{p-2}g^{ik}\pmod p=\begin{cases}0, \text{ if } p-1\nmid k;\\ -1, \text{ if } p-1\mid k.\end{cases}$$

In our expression, the exponent is $k=p-1-2\ell$, so we have

- **For $\ell=0$:** The exponent is $p-1$. The term is

$$\sum_{i=1}^{p}i^{p-1}\equiv\sum_{i=1}^{p-1}i^{p-1}\equiv-1\pmod p.$$

- **For $0<\ell<(p-1)/2$:** The exponent $p-1-2\ell$ is not a non-zero multiple of $p-1$, thus the inner sum

$$\sum_{i=0}^{p}i^{p-1-2\ell}=\sum_{i=0}^{p-1}i^{p-1-2\ell}\equiv0\pmod p.$$

- **For $\ell=(p-1)/2$:** The exponent is 0. The term is

$$\sum_{i=1}^{p}i^0\equiv0\pmod p.$$

Summing the only non-vanishing term, we find that

$$\sum_{i=0}^{p-1}\left(\frac{i^2+D}{p}\right)\equiv-1\pmod p.$$

Notice that

$$\left|\sum_{i=1}^{p}\left(\frac{i^2+D}{p}\right)\right|\le p,$$

which implies that $\sum_{i=1}^{p}\left(\frac{i^2+D}{p}\right) = -1$ or $p-1$. However, if $\sum_{i=1}^{p}\left(\frac{i^2+D}{p}\right) = p-1$, then we must have $p-1$ terms equal to 1 and exactly 1 term $a^2 + D \equiv 0 \pmod{p}$ with $a \equiv -a \pmod{p}$, which implies that $a \equiv 0 \pmod{p}$, since p is odd. Therefore, one has $D \equiv 0 \pmod{p}$, which gives us a contradiction. The proof is completed by the above fact and the following observation.

$$2\sum_{i=1}^{\frac{p-1}{2}}\left(\frac{i^2+D}{p}\right) = \sum_{i=1}^{p-1}\left(\frac{i^2+D}{p}\right) = -1 - \left(\frac{D}{p}\right).$$

□

Lemma 20. *Let $p_i > 3$ be distinct primes and $s \geq 1$. Then*

$$\frac{\prod_{i=1}^{s}(p_i-1)}{\prod_{i=1}^{s}(p_i-2)-1} \leq \frac{(p_{\min}-1)^s}{(p_{\min}-2)^s-1}.$$

Here $p_{\min} \coloneqq \min_{1\leq i\leq s}\{p_i\}$.

Proof. Observe that

$$\frac{\prod_{i=1}^{s}(p_i-1)}{\prod_{i=1}^{s}(p_i-2)-1} = \left(\frac{\prod_{i=1}^{s}(p_i-1)}{\prod_{i=1}^{s}(p_i-2)}\right)\left(\frac{\prod_{i=1}^{s}(p_i-2)}{\prod_{i=1}^{s}(p_i-2)-1}\right).$$

Since $(p_i-1)/(p_i-2)$ is a decreasing function for p_i, we have

$$\prod_{i=1}^{s}\left(\frac{p_i-1}{p_i-2}\right) \leq \frac{(p_{\min}-1)^s}{(p_{\min}-2)^s}.$$

The proof is completed by the facts that $x/(x-1)$ is decreasing and $\prod_{i=1}^{s}(p_i - 2) \geq (p_{\min}-2)^s$. □

Lemma 21. *Let $p_i \geq 5$ be distinct primes and $s \geq 1$. Then for any $1 \leq j \leq s$,*

$$\prod_{i=j}^{s}(p_i-1) \leq \prod_{i=j}^{s}2(p_i-2) - 2^{s-j+1}.$$

Proof. For all $p_i \geq 5$, we have

$$\prod_{i=j}^{s}(p_i-1) + 2^{s-j+1} \leq \prod_{i=j}^{s}\big((p_i-1)+2\big) = \prod_{i=j}^{s}(p_i+1) \leq \prod_{i=j}^{s}2(p_i-2).$$

□

Lemma 22. *For any integer $p_{\min} \geq 5$, the following inequality holds:*

$$\frac{1}{4}\left[\frac{(p_{\min}-1)^4}{(p_{\min}-2)^4-1}\right] < \frac{1}{4} + \frac{1.25}{p_{\min}-3}.$$

Proof. Let $x = p_{\min} - 2$. Thus, we have the equality:

$$\frac{1}{4}\left[\frac{(x+1)^4}{x^4-1}\right] = \frac{1}{4} + \frac{1}{4}\left(\frac{4}{x-1} + \frac{2}{x^2+1}\right).$$

For any real number x, it follows that $x^2 + 1 > x^2 - 1$, which implies

$$\begin{aligned}\frac{1}{4} + \frac{1}{4}\left(\frac{4}{x-1} + \frac{2}{x^2+1}\right) &< \frac{1}{4} + \frac{1}{4}\left(\frac{4}{x-1} + \frac{2}{x^2-1}\right) \\ &= \frac{1}{4} + \frac{1}{4}\left(\frac{4}{x-1} + \frac{1}{x-1} - \frac{1}{x+1}\right) \\ &< \frac{1}{4} + \frac{1}{4}\left[\frac{5}{x-1}\right].\end{aligned}$$

The proof is complete. □

7.7 A Proof of Theorem 3

We now construct a complete security proof for our proposed Lucas-based biprimality test protocol against semi-honest adversaries.

Theorem 5. *Let p and q be odd integers, $N = pq$, and let $p_{\min}$ be the minimal prime factor of N. Let $\kappa, \mathsf{LeakBound} > 0$ be security parameters. The inputs to $\mathcal{P}_i$ are given as*

$$(N, [p]_{\mathbb{Z}}, [q]_{\mathbb{Z}}),$$

where each share satisfies $\mathfrak{p}_1 \equiv p \pmod 4$, $\mathfrak{q}_1 \equiv q \pmod 4$, and $\mathfrak{p}_i \equiv \mathfrak{q}_i \equiv 0 \pmod 4$ for all $2 \le i \le n$. If $p_{min} \ge 11$, then the Protocol $\pi^{\mathcal{S}}_{\mathrm{RSA}}$ $(n-1)$-$\mathsf{privately\ computes}$ the functionality $\mathcal{F}^{\mathcal{S}}_{\mathrm{RSA}}$ in the $\mathcal{F}_{\mathrm{Leg}}$, $\mathcal{F}_{\mathrm{Shuffle}}$, $\mathcal{F}_{\mathrm{ModMul}}$-hybrid model.

Proof. The proof proceeds in two main parts. First, we establish the protocol's correctness, showing that honest parties with valid inputs complete the protocol successfully, while invalid inputs are rejected with overwhelming probability. Second, we prove privacy by constructing a simulator $\mathcal{S}$ that can generate a transcript indistinguishable from a real execution without knowing the honest parties' secrets. The core of the privacy proof relies on a trapdoor mechanism for generating the public parameters, and its indistinguishability is formally established through a series of number-theoretic lemmas (Lemma 23 to Proposition 4).

Correctness. We first consider the parameter search phase. If the protocol fails to find a suitable parameter D_k within $\mathsf{LeakBound}$ iterations (i.e., $|S_{\mathrm{Leak}}| > \mathsf{LeakBound}$), it immediately aborts and outputs `AbortLeakBound`, identically matching the functionality $\mathcal{F}^{\mathcal{S}}_{\mathrm{RSA}}$.

Conditioned on successfully finding a valid D_k ($|S_{\mathrm{Leak}}| \le \mathsf{LeakBound}$), we analyze the remaining execution based on the input N: Assuming primes $p < q$

with $\gcd(N, e_4) = 1$, these p, q do not output $(\texttt{NonRSAModulus}, \{p_i, q_i\}_{i\in[n]})$ with overwhelming probability. Note that for any $1 \le j \le \kappa$,

$$\begin{aligned}
&\Pr[\gcd(Q_j, N) = 1] = \Pr[(P_j^2 - D)/4 \in \mathbb{Z}_N^\times] \\
=&1 - \Pr[(P_j^2 - D)/4 \in (\mathbb{Z}_N \setminus \mathbb{Z}_N^\times)] \\
=&1 - \Pr[P_j^2 \equiv D \pmod p \text{ or } P_j^2 \equiv D \pmod q] \\
\ge&1 - \big(\Pr[P_j^2 \equiv D \pmod p] + \Pr[P_j^2 \equiv D \pmod q]\big) \\
\ge&1 - \left(\frac{2q}{N} + \frac{2p}{N}\right) \ge 1 - \frac{4}{p},
\end{aligned} \tag{10}$$

which implies that such p, q will pass all tests in step 5 with overwhelming probability(cf. Remark 4). For the check of step 8, by Theorem 2, we have $u_j \equiv \pm 1 \pmod{N\mathcal{O}_D}$ for all $1 \le j \le \kappa$. Using a similar argument, we may assume $r \in \mathbb{Z}_N^\times$ which implies

$$\gcd(N, z) = \gcd(N, e_4) = 1.$$

The output of $\pi_{\mathrm{RSA}}^{\mathcal{S}}$ is $\left(\texttt{RSAModulus}, \left\{\left(\frac{D_k}{p}\right)\right\}_{D_k\in S_{\mathrm{Leak}}}\right)$. In the case where $\gcd(N, e_4) \ne 1$, we have $\gcd(N, z) > 1$, and both $\pi_{\mathrm{RSA}}^{\mathcal{S}}$ and $\mathcal{F}_{\mathrm{RSA}}^{\mathcal{S}}$ output $(\texttt{NonRSAModulus}, \{\mathfrak{p}_i, \mathfrak{q}_i\}_{i=1}^n)$. If p, q are not distinct primes and $\gcd(N, e_4) = 1$, the exponential test pass probability is at most $\frac{1}{4} + \frac{1.25}{p_{\min}-3}$ (see Theorem 2). Thus, $\pi_{\mathrm{RSA}}^{\mathcal{S}}$ outputs $(\texttt{RSAModulus}, \{(\frac{D_k}{p})\}_{D_k\in S_{\mathrm{Leak}}})$ with probability at most $(\frac{1}{4} + \frac{1.25}{p_{\min}-3})^\kappa$.

Privacy. Let $\mathcal{P}^*$ be the set of corrupt parties. We show that a simulator $\mathcal{S}$ can be constructed to simulate the transcript of $\pi_{\mathrm{RSA}}^{\mathcal{S}}$. If the input of $\mathcal{S}$ is

$$(\mathcal{P}^*, N, \{\mathfrak{p}_i, \mathfrak{q}_i\}_{i\in\mathcal{P}^*}, \texttt{out}, \{\mathfrak{p}_i, \mathfrak{q}_i\}_{i=1}^n),$$

where the output flag is $\texttt{out} \in \{\texttt{NonRSAModulus}, \texttt{AbortLeakBound}\}$, then $\mathcal{S}$ only needs to follow the honest parties' strategy to simulate the view of the protocol. Therefore, we consider the case $\mathcal{S}$ is given the input

$$\left(\mathcal{P}^*, N, \{\mathfrak{p}_i, \mathfrak{q}_i\}_{i\in\mathcal{P}^*}, \texttt{RSAModulus}, \left\{\left(\frac{D_k}{p}\right)\right\}_{D_k\in S_{\mathrm{Leak}}}\right).$$

1: For all $1 \le j \le \kappa$, $\mathcal{S}$ randomly samples $v_j, w_j \in \mathbb{Z}_N$ with $\gcd(v_j^2 - w_j^2 D, N) = 1$, $b_j \in \{0, 1\}$, and sets $a_j = \frac{v_j + w_j\sqrt{D}}{v_j - w_j\sqrt{D}}$, $P_j' \in \mathbb{Z}_N$ such that the two roots of polynomial $x^2 - P_j'x + Q_j'$ are $\beta_j' \coloneqq \frac{\sqrt{D}}{a_j^2\cdot(-1)^{b_j}-1}$ and $\alpha_j' \coloneqq \beta_j' + \sqrt{D}$.
2: The simulator $\mathcal{S}$ randomly generates $z' \in \mathbb{Z}_N$, and it's additive shares $[z']_N$.
3: The simulator $\mathcal{S}$ outputs

$$\big(\mathcal{P}^*,\ N, \{\mathfrak{p}_i, \mathfrak{q}_i\}_{i\in\mathcal{P}^*}, \{P_j', (-1)^{b_j}\}_{i=1}^\kappa, [z']_N, \{z_i'\}_{i=1}^n\big).$$

First, we demonstrate that the expression for P'_j simplifies to a value in $\mathbb{Z}_N$ under the given conditions. Let $a_j = \frac{v_j + w_j\sqrt{D}}{v_j - w_j\sqrt{D}}$. The simulator sets P'_j such that

$$P'_j = \frac{2\sqrt{D}}{a_j^2(-1)^{b_j} - 1} + \sqrt{D}.$$

We analyze this expression by considering the two cases for $b_j \in \{0, 1\}$.
Case 1: $b_j = 0$**.** The expression becomes:

$$\begin{aligned}
P'_j = \frac{2\sqrt{D}}{a_j^2 - 1} + \sqrt{D} &= \frac{2\sqrt{D}}{\left(\frac{v_j + w_j\sqrt{D}}{v_j - w_j\sqrt{D}}\right)^2 - 1} + \sqrt{D} \\
&= \frac{2\sqrt{D}(v_j - w_j\sqrt{D})^2}{(v_j + w_j\sqrt{D})^2 - (v_j - w_j\sqrt{D})^2} + \sqrt{D} \\
&= \frac{2\sqrt{D}(v_j - w_j\sqrt{D})^2}{4v_j w_j\sqrt{D}} + \sqrt{D} = \frac{(v_j - w_j\sqrt{D})^2}{2v_j w_j} + \frac{2v_j w_j\sqrt{D}}{2v_j w_j} \\
&= \frac{v_j^2 - 2v_j w_j\sqrt{D} + w_j^2 D + 2v_j w_j\sqrt{D}}{2v_j w_j} = \frac{v_j^2 + w_j^2 D}{2v_j w_j}.
\end{aligned}$$

Case 2: $b_j = 1$**.** The expression becomes:

$$\begin{aligned}
P'_j = \frac{2\sqrt{D}}{-a_j^2 - 1} + \sqrt{D} &= \frac{-2\sqrt{D}}{\left(\frac{v_j + w_j\sqrt{D}}{v_j - w_j\sqrt{D}}\right)^2 + 1} + \sqrt{D} \\
&= \frac{-2\sqrt{D}(v_j - w_j\sqrt{D})^2}{(v_j + w_j\sqrt{D})^2 + (v_j - w_j\sqrt{D})^2} + \sqrt{D} \\
&= \frac{-2\sqrt{D}(v_j - w_j\sqrt{D})^2}{2(v_j^2 + w_j^2 D)} + \sqrt{D} \\
&= \frac{-\sqrt{D}(v_j^2 - 2v_j w_j\sqrt{D} + w_j^2 D)}{v_j^2 + w_j^2 D} + \frac{\sqrt{D}(v_j^2 + w_j^2 D)}{v_j^2 + w_j^2 D} \\
&= \frac{-\sqrt{D}v_j^2 + 2v_j w_j D - w_j^2 D\sqrt{D} + \sqrt{D}v_j^2 + w_j^2 D\sqrt{D}}{v_j^2 + w_j^2 D} = \frac{2v_j w_j D}{v_j^2 + w_j^2 D}.
\end{aligned}$$

These two results can be unified into a single expression. Let $X = \frac{v_j^2 + w_j^2 D}{2v_j w_j}$. Then for $b_j = 0$, $P'_j = X$. For $b_j = 1$, $P'_j = D \cdot X^{-1}$. This corresponds exactly to $X^{1-2b_j} D^{b_j}$. Since v_j, w_j are chosen such that $v_j^2 - w_j^2 D \in \mathbb{Z}_N^\times$, it implies that v_j, w_j are not both zero-divisors, ensuring that $2v_j w_j$ is invertible with overwhelming probability. Thus, $P'_j \in \mathbb{Z}_N$.

Secondly, we show that the distribution of $(P'_j, (-1)^{b_j})$ generated by the simulator is indistinguishable from the distribution of the real-world transcript $(P_j, u_j) = (P_j, (\alpha_j \beta_j^{-1})^{e_4})$. Note that

$$(\alpha'_j \beta'^{-1}_j)^{e_4} = ((\beta'_j + \sqrt{D})\beta'^{-1}_j)^{e_4} = (a_j^2(-1)^{b_j})^{e_4}.$$

Due to the symmetry between p and q, we only need to consider proving

$$(a_j^2)^{e_4} \equiv 1 \pmod{p\mathcal{O}_D}.$$

Since p, q are odd primes and e_4 is odd, we have

1. If $\left(\frac{D}{p}\right) = -1$, we have, recalling that $\mathbf{N}$ is the norm map from $\mathcal{O}_D$ to $\mathbb{Z}$,

$$\left(\frac{v_j + w_j\sqrt{D}}{v_j - w_j\sqrt{D}}\right)^{2e_4} \equiv \left(\mathbf{N}\left(\frac{v_j + w_j\sqrt{D}}{v_j - w_j\sqrt{D}}\right)\right)^{(q+(\frac{-1}{q}))/2} \equiv 1 \pmod{p\mathcal{O}_D}.$$

2. If $\left(\frac{D}{p}\right) = 1$ (i.e., $\sqrt{D} \in \mathbb{Z}_p^\times$), Euler theorem says that

$$\left(\frac{v_j + w_j\sqrt{D}}{v_j - w_j\sqrt{D}}\right)^{(p+(\frac{-1}{p}))(q+(\frac{-1}{q}))/2} \equiv 1 \pmod{p}.$$

Therefore, $(\alpha'_j\beta'^{-1}_j)^{e_4} \equiv (-1)^{b_j} \pmod{N\mathcal{O}_D}$ by CRT. Note that the distribution of P'_j produced by the simulator $\mathcal{S}$ at the step 1. Proposition 4 says that the distributions of P_j and P'_j are identical. Lastly, $\gcd(N, e_4) = 1$ implies that $(p(\frac{-1}{q}) + q(\frac{-1}{p}) + (\frac{-1}{N})) \in \mathbb{Z}_N^\times$, and $z \equiv r(p(\frac{-1}{q}) + q(\frac{-1}{p}) + (\frac{-1}{N})) \pmod{N}$ is uniformly distributed in $\mathbb{Z}_N$. We conclude that the joint distribution of the outputs generated by $\mathcal{S}$ and $\mathcal{F}_{\mathrm{RSA}}^{\mathcal{S}}$, and of the view and output of an execution $\pi_{\mathrm{RSA}}^{\mathcal{S}}$ are indistinguishable. □

Next, for an integer m, if $\sqrt{D} \in \mathbb{Z}_m^\times$, then $\pmod m$ refers to the module $m\mathbb{Z}$; otherwise, if $\sqrt{D} \notin \mathbb{Z}_m^\times$, $\pmod m$ refers to the module $m\mathcal{O}_D$. To investigate the distribution of P_j and P'_j, we will examine the relationship between $S_{\mathrm{real}(m,b)}$ and $S_{\mathrm{ideal}(m,b)}$ given an odd integer m and $b \in \{0, 1\}$. Here

$$S_{\mathrm{real}(m,b)} \coloneqq \left\{ P \in \mathbb{Z}_m \;\middle|\; \left(\frac{(P^2 - D)/4}{m}\right) = (-1)^b \right\}, \text{ and}$$

$$S_{\mathrm{ideal}(m,b)} \coloneqq \left\{ \frac{2\sqrt{D}}{a^2(-1)^b - 1} + \sqrt{D} \;\middle|\; a = \frac{v + w\sqrt{D}}{v - w\sqrt{D}}, v, w \in \mathbb{Z}_m, \right.$$
$$\left. v^2 - w^2 D \in \mathbb{Z}_m^\times, a^2(-1)^b \not\equiv 1 \pmod{m} \right\}.$$

Then we have

Lemma 23. *If p is an odd prime, and D is an integer with $\left(\frac{-D}{p}\right) = -1$, then we have $S_{\mathrm{real}(p,b)} = S_{\mathrm{ideal}(p,b)}$ for $b \in \{0, 1\}$.*

Proof. For any $P' \in S_{\text{ideal}(p,b)}$, we have

$$\begin{aligned}(P'^2 - D)/4 =& \left[\frac{\sqrt{D}}{a^2(-1)^b - 1}\right]^2 + \frac{D}{a^2(-1)^b - 1} \\ =& D\left[\frac{(v^2 - w^2D)^2(-1)^b}{\left((v + w\sqrt{D})^2(-1)^b - (v - w\sqrt{D})^2\right)^2}\right]\end{aligned}$$

Therefore,

$$\left(\frac{(P'^2 - D)/4}{p}\right) = \begin{cases} \left(\frac{1/(v^2w^2)}{p}\right) = 1 & \text{, if } b = 0; \\ \left(\frac{-D/(v^2+w^2D)^2}{p}\right) = -1 & \text{, if } b = 1. \end{cases}$$

We derive $\left(\frac{(P'^2-D)/4}{p}\right) = (-1)^b$ and $S_{\text{real}(p,b)} \supseteq S_{\text{ideal}(p,b)}$. On the other hand, let P be an element in $S_{\text{real}(p,b)}$. We assume that there exists a belonging the set

$$\left\{\frac{v + w\sqrt{D}}{v - w\sqrt{D}}\middle|\ v, w \in \mathbb{Z}_p, v^2 - w^2D \in \mathbb{Z}_{p^\times}, \right. \\ \left. (v + w\sqrt{D})^2 \equiv (-1)^b(v - w\sqrt{D})^2 \pmod p\right\}$$

such that $a^2(-1)^b = \frac{P+\sqrt{D}}{P-\sqrt{D}} \not\equiv 1 \pmod p$. Then we have

$$P \equiv \frac{2\sqrt{D}}{\frac{P+\sqrt{D}}{P-\sqrt{D}} - 1} + \sqrt{D} \equiv \frac{2\sqrt{D}}{a^2(-1)^b - 1} + \sqrt{D} \pmod p,$$

which implies $S_{\text{real}}(p, b) \subseteq S_{\text{ideal}}(p, b)$. To prove the assumption, we split it into two cases.

Case1: $\left(\frac{D}{p}\right) = 1$ (i.e., $\sqrt{D} \in \mathbb{Z}_p^\times$).

Note that $\left(\frac{P^2-D}{p}\right) = (-1)^b$, because of P is an element in $S_{\text{real}}(p, b)$. Since the condition in Lemma gives $\left(\frac{-D}{p}\right) = -1$, we have $\left(\frac{-1}{p}\right) = -1$. Then one has

$$\left(\frac{(-1)^b(P + \sqrt{D})/(P - \sqrt{D})}{p}\right) = \left(\frac{(-1)^b(P + \sqrt{D})^2/(P^2 - D)}{p}\right)$$

$$= \left(\frac{(-1)^b(P^2 - D)}{p}\right) = \left(\frac{-1}{p}\right)^b (-1)^b = (-1)^b(-1)^b = 1.$$

There exists $t \in \mathbb{Z}_p^\times$ such that $t^2 \equiv (-1)^b \frac{P+\sqrt{D}}{P-\sqrt{D}} \pmod p$. Consider the case $t \not\equiv 1 \pmod p$, we take $(v, w) = (\frac{t+1}{t-1}\sqrt{D}, 1)$ and then $a^2 \equiv \left[\frac{v+w\sqrt{D}}{v-w\sqrt{D}}\right]^2 \equiv (-1)^b \frac{P+\sqrt{D}}{P-\sqrt{D}} \pmod p$. If $t = 1$, we set $(v, w) = (1, 0)$, then $a^2 = 1$.

Case 2: $\left(\frac{D}{p}\right) = -1$.
If $b = 0$, we take $(v, w) = (P + \sqrt{P^2 - D}, 1)$. If $b = 1$, we must further distinguish based on P. For $P \not\equiv 0 \pmod{p}$, we take $(v, w) = \left(\frac{D+\sqrt{D(D-P^2)}}{P}, 1\right)$. For the special case $P \equiv 0 \pmod{p}$, which is a valid element in $S_{\text{real}(p,1)}$ since $\left(\frac{-D}{p}\right) = -1$, the previous choice is undefined. Here, we instead take $(v, w) = (0, 1)$.

These choices for v are well-defined in $\mathbb{Z}_p$ since for $b = 0$, we have $\left(\frac{P^2-D}{p}\right) = 1$, and for $b = 1$, $\left(\frac{D(D-P^2)}{p}\right) = \left(\frac{-D}{p}\right)\left(\frac{P^2-D}{p}\right) = (-1)(-1) = 1$.

Let $a = \frac{v+w\sqrt{D}}{v-w\sqrt{D}}$. The target identity $a^2(-1)^b = \frac{P+\sqrt{D}}{P-\sqrt{D}}$ is algebraically equivalent to the quadratic relation $v^2 - 2Pv + D = 0$ for $b = 0$, and $Pv^2 - 2Dv + PD = 0$ for $b = 1$. A direct substitution confirms that our choices of (v, w) in each case satisfy the corresponding equation (i.e. for $P = 0, b = 1$, the choice $(0, 1)$ satisfies $-2Dv = 0$). Thus, one has $a^2 \equiv (-1)^b \frac{P+\sqrt{D}}{P-\sqrt{D}} \pmod{p}$. □

Assume that p is an odd prime and $D \in \mathbb{Z}_p^\times$. Let

$$G \coloneqq \left\{ (a, b) \;\middle|\; a, b \in \mathbb{Z}_p, a^2 - b^2 D \in \mathbb{Z}_p^\times \right\}.$$

Given $g_1 = (a_1, b_1), g_2 = (a_2, b_2) \in G$, define $g_1 * g_2 = (a_1a_2 + b_1b_2D, a_1b_2 + b_1a_2)$. Then G is a group with the identity $(1, 0)$, and its inverse of $g = (a, b)$ is $\big(a/(a^2 - b^2D), -b/(a^2 - b^2D)\big)$. Let $\mathcal{O}_D/p\mathcal{O}_D = \mathbb{Z}_p[x]/(x^2 - D)$ be the formal quotient ring. For any element $a + b\sqrt{D}$ with $a^2 - b^2D \in \mathbb{Z}_p^\times$, the element is invertible with inverse $(a - b\sqrt{D})(a^2 - b^2D)^{-1}$. We use fraction notation $\frac{u}{v}$ to denote $u \cdot v^{-1}$ in this ring. Let

$$H \coloneqq \left\{ \frac{a + b\sqrt{D}}{a - b\sqrt{D}} \in \mathcal{O}_D/p\mathcal{O}_D \;\middle|\; a, b \in \mathbb{Z}_p, a^2 - b^2D \in \mathbb{Z}_p^\times \right\},$$

which forms a multiplicative group. The inverse of any $h = \frac{a+b\sqrt{D}}{a-b\sqrt{D}} \in H$ is $\frac{a-b\sqrt{D}}{a+b\sqrt{D}}$, and the identity is 1.

Lemma 24. *Let p be an odd prime, and $D \in \mathbb{Z}_p^\times$. Consider a group homomorphism $f : G \to H$ defined by*

$$g = (a, b) \in G \mapsto \left(\frac{a + b\sqrt{D}}{a - b\sqrt{D}}\right)^2 \in H.$$

Then the set of $f(g)$ forms a subgroup of H, and $|\ker(f)| = 2p - 2$.

Proof. It is straightforward to verify that f is a group homomorphism, that is, for any $(a_1, b_1), (a_2, b_2) \in G$:

$$\begin{aligned} f(a_1, b_1)f(a_2, b_2) &= \left(\frac{a_1a_2 + b_1b_2D + (a_1b_2 + a_2b_1)\sqrt{D}}{a_1a_2 + b_1b_2D - (a_1b_2 + a_2b_1)\sqrt{D}}\right)^2 \\ &= f((a_1, b_1) * (a_2, b_2)). \end{aligned}$$

We now compute its kernel. An element $g = (a, b) \in G$ is in $\ker(f)$ if and only if $f(g) = 1$, which means:

$$\left(\frac{a + b\sqrt{D}}{a - b\sqrt{D}}\right)^2 = 1 \implies \frac{a + b\sqrt{D}}{a - b\sqrt{D}} = \pm 1.$$

We consider the two cases:

- If the ratio is $+1$, then $a + b\sqrt{D} = a - b\sqrt{D}$, which implies $2b\sqrt{D} = 0$, so $b = 0$. For $(a, 0)$ to be in G, we require $a^2 \in \mathbb{Z}_p^\times$, giving $p - 1$ choices for a.
- If the ratio is -1, then $a + b\sqrt{D} = -a + b\sqrt{D}$, which implies $2a = 0$, so $a = 0$. For $(0, b)$ to be in G, we require $-b^2D \in \mathbb{Z}_p^\times$, which means $b \in \mathbb{Z}_p^\times$ (since D is invertible). This gives another $p - 1$ choices for b.

The kernel is the disjoint union of these two sets of elements. Thus, $|\ker(f)| = (p - 1) + (p - 1) = 2p - 2$. □

Proposition 4. *If $N = pq$ is an odd RSA modulus, and D is an integer with $\left(\frac{-D}{p}\right) = \left(\frac{-D}{q}\right) = -1$, then we have $S_{\text{real}(N,0)} = S_{\text{ideal}(N,0)} \cup S_{\text{ideal}(N,1)}$. Furthermore, uniformly sampling $v, w \in \mathbb{Z}_N$ and $b \in \{0, 1\}$ subject to $v^2 - w^2D \in \mathbb{Z}_N^\times$ and $a^2(-1)^b \not\equiv 1 \pmod N$ (where $a = \frac{v+w\sqrt{D}}{v-w\sqrt{D}}$) is equivalent to randomly selecting from the set $S_{\text{real}(N,0)}$.*

Proof. The proof consists of two parts. First, we establish the equality of the sets. Second, we analyze the distribution of the elements.

Part 1: Equality of Sets. According to the CRT, an element is defined by its residues modulo p and q. Thus, we have

$$S_{\text{real}(N,0)} = \left(S_{\text{real}(p,0)} \times S_{\text{real}(q,0)}\right) \cup \left(S_{\text{real}(p,1)} \times S_{\text{real}(q,1)}\right).$$

Similarly, the definition of $S_{\text{ideal}(N,b)}$ decomposes as

$$S_{\text{ideal}(N,0)} = S_{\text{ideal}(p,0)} \times S_{\text{ideal}(q,0)}, \text{ and } \quad S_{\text{ideal}(N,1)} = S_{\text{ideal}(p,1)} \times S_{\text{ideal}(q,1)}.$$

By Lemma 23, we have the local equality $S_{\text{real}(p,b)} = S_{\text{ideal}(p,b)}$ for each prime factor. Substituting these equalities back into the decomposition of $S_{\text{real}(N,0)}$ gives

$$S_{\text{real}(N,0)} = S_{\text{ideal}(N,0)} \cup S_{\text{ideal}(N,1)}.$$

This proves the claimed set equality.

Part 2: Uniformity of Distribution. Our goal is to prove that uniformly sampling (v, w) and b from the valid domain

$$\mathcal{D}_N \coloneqq \left\{ ((v,w),b) \in T_N \times \{0,1\} \;\middle|\; a^2(-1)^b \not\equiv 1 \pmod{N} \right\},$$

where $T_N \coloneqq \{(v,w) \mid v^2 - w^2 D \in \mathbb{Z}_N^\times\}$ and $a = \frac{v+w\sqrt{D}}{v-w\sqrt{D}}$, results in a value P that is uniformly distributed over $S_{\text{real}(N,0)}$.

To do this, we define a map Φ from this valid input space to the output space:

$$\Phi : \mathcal{D}_N \to S_{\text{real}(N,0)}$$

where $\Phi(((v,w),b))$ is the corresponding output P. To prove that the output distribution is uniform, it suffices to show that the size of the preimage, $|\Phi^{-1}(P)|$, is a constant for every element P in the output set $S_{\text{real}(N,0)}$.

Let $P \in S_{\text{real}(N,0)}$ be an arbitrary element. From Part 1, we know that P must belong to either $S_{\text{ideal}(N,0)}$ or $S_{\text{ideal}(N,1)}$. We consider these two cases.

- **Case 1:** $P \in S_{\text{ideal}(N,0)}$**.** Such a P must be generated with the choice $b = 0$. By the CRT, P corresponds to a unique pair (P_p, P_q) where $P_p \in S_{\text{ideal}(p,0)}$ and $P_q \in S_{\text{ideal}(q,0)}$. According to Lemma 24, the number of pairs $(v_p, w_p) \in T_p$ that map to P_p is exactly $|\ker(f_p)| = 2p - 2$. Similarly, the number of pairs $(v_q, w_q) \in T_q$ that map to P_q is $2q - 2$. Since $T_N \cong T_p \times T_q$, the total number of pairs $(v, w) \in T_N$ that generate P is the product $(2p-2)(2q-2)$. Therefore, the size of the preimage for P is

$$|\Phi^{-1}(P)| = (2p-2)(2q-2).$$

- **Case 2:** $P \in S_{\text{ideal}(N,1)}$**.** Such a P must be generated with the choice $b = 1$. A parallel argument holds. The element P corresponds to a unique pair (P_p, P_q) where $P_p \in S_{\text{ideal}(p,1)}$ and $P_q \in S_{\text{ideal}(q,1)}$. By the same reasoning from Lemma 24, the number of pairs in T_N that generate this P is also $(2p-2)(2q-2)$. Thus, the size of the preimage is

$$|\Phi^{-1}(P)| = (2p-2)(2q-2).$$

In both cases, we have shown that for any element P in the output set $S_{\text{real}(N,0)}$, the number of valid inputs in $\mathcal{D}_N$ that map to it is the constant value $(2p-2)(2q-2)$. This implies that if an input is chosen uniformly at random from $\mathcal{D}_N$, every possible output $P \in S_{\text{real}(N,0)}$ has an equal probability of being generated. Therefore, the sampling process is equivalent to selecting an element uniformly at random from $S_{\text{real}(N,0)}$. □

Remark 4. In the practical scenario (e.g., [15]), distributed RSA moduli protocols generate $p = \sum_{i=1}^{n} \mathfrak{p}_i$ and $q = \sum_{i=1}^{n} \mathfrak{q}_i$, where $\mathfrak{p}_i$ and $\mathfrak{q}_i$ are uniformly sampled from $[0, 2^{\ell - \log_2 n}]$, with ℓ being the security parameter. This implies $\max\{p, q\}$ is at most ℓ-bits and

$$\mathbb{P}[\min\{p,q\} \text{ is larger than } (\ell - \log_2 n - 80)\text{-bits }] \geq 1 - 2^{-80n}.$$

Lemma 25. *Let $N = pq$ be the product of two different primes and* LeakBound *be a positive integer. Suppose there exists a polynomial time algorithm $\mathcal{A}$ that given 1) a sequence of primes $\{D_j\}_{j=1}^{LeakBound}$ satisfying $\left(\frac{-D_j}{N}\right) = 1$; and 2) $\left(\frac{-D_j}{p}\right) = \epsilon_j$ for all $1 \leq j \leq$ LeakBound, factors N. Then there exists a polynomial time algorithm $\mathcal{B}$ that factors N.*

Proof. Given any $\{\epsilon_j\}_{j=1}^{\mathsf{LeakBound}}$ with $\left\{\left(\frac{-D_j}{p}\right) = \epsilon_j\right\}_{j=1}^{\mathsf{LeakBound}}$, we can use the algorithm $\mathcal{A}$ to factor N, which is repeated at most $2^{\mathsf{LeakBound}}$ due to a total of $2^{\mathsf{LeakBound}}$ possibilities for $\left\{\left(\frac{-D_j}{p}\right) = \epsilon_j\right\}_{j=1}^{\mathsf{LeakBound}}$ with arbitrary $\epsilon_j \in \{-1, 1\}$. □

7.8 A security proof of Protocol $\pi_{\mathbf{Leg}}$

For completeness, we provide protocol π_{Leg}, which securely realizes Functionality $\mathcal{F}_{\mathrm{Leg}}$ (cf. Proposition 5). A similar protocol for computing the Legendre symbol was proposed in [25], but a detailed security proof was not provided.

Functionality 4 $\mathcal{F}_{\mathrm{Leg}}(n)$

Inputs: Each party $\mathcal{P}_i$ has a share $[p]_{\mathbb{Z}}$, $p \pmod 4$, and a prime D with $\gcd(D, p) = 1$.

Outputs: Each party $\mathcal{P}_i$ receives the value $\left(\frac{-D}{p}\right)$.

Protocol 3 Legendre symbol $\pi_{\mathrm{Leg}}(n)$

Inputs: Each party $\mathcal{P}_i$ has $[p]_{\mathbb{Z}}$, $p \pmod 4$, and a prime D with $\gcd(D, p) = 1$.

Outputs: $\left(\frac{-D}{p}\right)$.

1. Each party randomly sample $\mathfrak{s}_i \in \mathbb{Z}_D$ sends $(\mathfrak{s}_i, \mathfrak{s}_i, D)$ to $\mathcal{F}_{\mathrm{ModMul}}$ to obtain $[s^2]_D$.
2. Each party sends $([s^2]_D, \mathfrak{p}_i \pmod D), D)$ to $\mathcal{F}_{\mathrm{ModMul}}$ to obtain $[s^2 p]_D$.
3. Each party opens $[s^2 p]_D$. If $\gcd(s^2 p, D) \neq 1$, then restarts to the step 1. Otherwise, output

$$\begin{cases} -\left(\frac{s^2 p}{D}\right), & \text{if } p \equiv 3 \pmod 4 \text{ and } D \equiv 1 \pmod 4; \\ \left(\frac{s^2 p}{D}\right), & \text{otherwise.} \end{cases}$$

Proposition 5. *Protocol* π_{Leg} $(n-1)$-privately computes *the functionality* $\mathcal{F}_{\mathrm{Leg}}$ *in* $\mathcal{F}_{\mathrm{ModMul}}$*-hybrid model.*

Proof. We omit the straightforward proof of correctness and focus on the proof of privacy. We construct the simulator $\mathcal{S}$ to simulate the transcript of π_{Leg}. Suppose $\mathcal{S}$ is given input

$$\left(\mathcal{P}^*, \{\mathfrak{p}_i\}_{i\in\mathcal{P}^*}, p \pmod 4, D, \left(\frac{-D}{p}\right)\right).$$

1: $\mathcal{S}$ uniformly samples $s \in \mathbb{Z}_D^\times$ and $\mathfrak{s}_i \in \mathbb{Z}_D$ for $i \in \{1,\dots,n\}$ such that $\sum_{i=1}^n \mathfrak{s}_i \equiv s \pmod D$.
2: $\mathcal{S}$ uniformly samples $\mathfrak{s}'_i \in \mathbb{Z}_D$ for $i \in \{1,\dots,n\}$ such that $\sum_{i=1}^n \mathfrak{s}'_i \equiv s^2 \pmod D$.
3: $\mathcal{S}$ uniformly samples $r \in \mathbb{Z}_D^\times$ such that

$$\left(\frac{r}{D}\right) = \begin{cases} -\left(\frac{-D}{p}\right), & \text{if } p \equiv 3 \pmod 4 \text{ and } D \equiv 1 \pmod 4; \\ \left(\frac{-D}{p}\right), & \text{otherwise.} \end{cases}$$

4: $\mathcal{S}$ uniformly samples $\mathfrak{r}_i \in \mathbb{Z}_D$ for $i \in \{1,\dots,n\}$ such that $\sum_{i=1}^n \mathfrak{r}_i \equiv r \pmod D$.
5: $\mathcal{S}$ outputs

$$\begin{aligned}(&\{\mathfrak{p}_i\}_{i\in\mathcal{P}^*}, p \pmod 4, D,\\ &\{\mathfrak{s}_i\}_{i\in\mathcal{P}^*}, \{\mathfrak{s}'_i\}_{i\in\mathcal{P}^*}, \{\mathfrak{r}_i\}_{i\in\mathcal{P}^*}, \{\mathfrak{r}_i\}_{i\in\{1,\dots,n\}\setminus\mathcal{P}^*})\end{aligned}$$

Because $\mathcal{F}_{\mathrm{Leg}}$ is a deterministic function, we only need to prove

$$\begin{aligned}&\left\{\mathcal{S}\left(\mathcal{P}^*, \{\mathfrak{p}_i\}_{i\in\mathcal{P}^*}, p \pmod 4, D, \left(\frac{-D}{p}\right)\right)\right\}\\ \overset{c}{\equiv}&\{\mathrm{view}^{\pi_{\mathrm{Leg}}}_{\mathcal{P}^*}(\mathcal{P}^*, \{\mathfrak{p}_i\}_{i\in\mathcal{P}^*}, p \pmod 4, D)\}\end{aligned}$$

for any $\mathcal{P}^* \subseteq \{1,\dots,n\}, |\mathcal{P}^*| \le n-1, \{\mathfrak{p}_i\}_{i=1}^n$ and prime D. In the beginning, fixed any $\{\mathfrak{p}_i\}_{i=1}^n$ and D, we claim that the output of

$$\mathcal{S}\left(\mathcal{P}^*, \{\mathfrak{p}_i\}_{i\in\mathcal{P}^*}, p \pmod 4, D, \left(\frac{-D}{p}\right)\right)$$

and the view

$$\mathrm{view}^{\pi_{\mathrm{Leg}}}_{\mathcal{P}^*}(\mathcal{P}^*, \{\mathfrak{p}_i\}_{i\in\mathcal{P}^*}, p \pmod 4, D)$$

are identical. Observe that

$$\begin{aligned}\left(\frac{p}{D}\right) &= \left(\frac{D}{p}\right)\cdot(-1)^{\frac{p-1}{2}\frac{D-1}{2}} = \left(\frac{-D}{p}\right)\cdot\left(\frac{-1}{p}\right)\cdot(-1)^{\frac{p-1}{2}\frac{D-1}{2}}\\ &= \left(\frac{-D}{p}\right)\cdot(-1)^{\frac{p-1}{2}}\cdot(-1)^{\frac{p-1}{2}\frac{D-1}{2}}\end{aligned}$$

implies that $\left(\frac{p}{D}\right) = \left(\frac{r}{D}\right)$. The facts that D is a prime, and s is uniformly randomly chosen from $\mathbb{Z}_D^\times$, which gives us the identical distribution between $\{s^2 p \mid s \in \mathbb{Z}_D^\times\}$ with $\left\{r \in \mathbb{Z}_D^\times \mid \left(\frac{r}{D}\right) = (-1)^{\frac{(p-1)(D+1)}{4}} \left(\frac{-D}{p}\right)\right\}$. Due to $|\mathcal{P}^*| < n$, $\mathfrak{s}_i, \mathfrak{s}'_i$ in the $\text{view}_{\mathcal{P}^*}^{\pi_{\text{Leg}}}(\mathcal{P}^*, \{\mathfrak{p}_i\}_{i\in\mathcal{P}^*}, p \pmod 4, D)$ and

$$\mathcal{S}\left(\mathcal{P}^*, \{\mathfrak{p}_i\}_{i\in\mathcal{P}^*}, p \pmod 4, D, \left(\frac{-D}{p}\right)\right)$$

are both independently and uniformly distributed in $\mathbb{Z}_D$. Let $\mathfrak{z}_i$ denote the uniformly random share of $[s^2 p]_D$ received by party $\mathcal{P}_i$ from $\mathcal{F}_{\text{ModMul}}$ in the real execution. We conclude that for any $\mathcal{P}^* \subseteq \{1, \ldots, n\}, |\mathcal{P}^*| \leq n-1, \{\mathfrak{p}_i\}_{i=1}^n$, and prime D

$$\begin{aligned}
&\left\{\mathcal{S}\left(\mathcal{P}^*, \{\mathfrak{p}_i\}_{i\in\mathcal{P}^*}, p \pmod 4, D, \left(\frac{-D}{p}\right)\right)\right\} \\
&\overset{c}{\equiv}\left(\{\mathfrak{p}_i\}_{i\in\mathcal{P}^*}, p \pmod 4, D, \{\mathfrak{s}_i\}_{i\in\mathcal{P}^*}, \{\mathfrak{s}'_i\}_{i\in\mathcal{P}^*}, \{\mathfrak{r}_i\}_{i\in\mathcal{P}^*}, \{\mathfrak{r}_i\}_{i\in\{1,\ldots,n\}\setminus\mathcal{P}^*}\right) \\
&\overset{c}{\equiv}\left(\{\mathfrak{p}_i\}_{i\in\mathcal{P}^*}, p \pmod 4, D, \{\mathfrak{s}_i\}_{i\in\mathcal{P}^*}, \{\mathfrak{s}'_i\}_{i\in\mathcal{P}^*}, \{\mathfrak{z}_i\}_{i\in\mathcal{P}^*}, \{\mathfrak{z}_i\}_{i\in\{1,\ldots,n\}\setminus\mathcal{P}^*}\right) \\
&\overset{c}{\equiv}\left\{\text{view}_{\mathcal{P}^*}^{\pi_{\text{Leg}}}(\mathcal{P}^*, \{\mathfrak{p}_i\}_{i\in\mathcal{P}^*}, p \pmod 4, D)\right\}.
\end{aligned}$$

□

7.9 Some Lemmas of Statistical Distance

In this paper, we use the following statistical distance to prove Proposition 6, 7 for our malicious model, showing that the real view and the ideal view are indistinguishable.

Definition 5. [11, Definition 3.5] *Let X, Y be two random vectors that takes values in a finite set S. The statistical distance between X and Y is defined as*

$$\text{SD}(X, Y) = \frac{1}{2}\sum_{s\in S} |\Pr(X = s) - \Pr(Y = s)|.$$

Here, we will study the statistical distance between the distributions of the sum of two random variables modulo a positive integer A. More formally, let $X = (X_1, \ldots, X_n)$ and $Y = (Y_1, \ldots, Y_n)$ be random vectors defined on a probability space. We define $X + Y \pmod A$ to mean the component-wise modulo $\big(X_1 + Y_1 \pmod A, \ldots, X_n + Y_n \pmod A\big)$. In addition, when S is a finite set, we define U_S to be the random variable uniformly distributed over S.

Lemma 26. *Suppose $X = (X_1, \ldots, X_n)$, $Y = (Y_1, \ldots, Y_n)$ are random vectors with X_i, Y_i are all independent random variable on the finite set S. Then*

$$\text{SD}(X, Y) \leq \sum_{i=1}^{n} \text{SD}(X_i, Y_i).$$

Proof. This is a standard application of the hybrid argument. For $i \in \{1, \ldots, n-1\}$, let H_i be the hybrid distributions defined as follows: $H_1 \coloneqq (Y_1, X_2, \ldots, X_n)$, $H_2 \coloneqq (Y_1, Y_2, X_3, \ldots, X_n)$, $\ldots$, $H_{n-1} \coloneqq (Y_1, \ldots, Y_{n-1}, X_n)$. By triangle inequality, we have:

$$\begin{aligned}\mathrm{SD}(X, Y) &\leq \mathrm{SD}(X, H_1) + \mathrm{SD}(H_1, H_2) + \cdots + \mathrm{SD}(H_{n-1}, Y) \\ &= \sum_{i=1}^{n} \mathrm{SD}(X_i, Y_i).\end{aligned}$$

□

Lemma 27. *Let X, Z be independent random variables on the set $\{0, \ldots, A-1\}$, and U be the uniformly random variable on the set $\{0, \ldots, A-1\}$. Then*

$$\mathrm{SD}(X + Z \pmod A, U \pmod A) \leq \mathrm{SD}(X, U).$$

Proof. Note that the sum of two distributions $U + Z \pmod A$ is still U for any distribution Z. Therefore, we only need to prove that

$$\mathrm{SD}(X + Z \pmod A, U + Z \pmod A) \leq \mathrm{SD}(X, U).$$

From the definition of SD, one has

$$\begin{aligned}&\mathrm{SD}(X + Z \pmod A, U + Z \pmod A) \\ =&\frac{1}{2} \sum_z \left| \sum_x \Pr(X = x) \Pr(Z = z - x) - \sum_x \Pr(U = x) \Pr(Z = z - x) \right| \\ \leq&\frac{1}{2} \sum_z \sum_x |\Pr(X = x) - \Pr(U = x)| \Pr(Z = z - x) \\ =&\frac{1}{2} \sum_x |\Pr(X = x) - \Pr(U = x)| \sum_z \Pr(Z = z - x) \\ =&\mathrm{SD}(X \pmod A, U \pmod A).\end{aligned}$$

□

Lemma 28. *Let $B > A > 0$ be positive integers. Write $2B + 1 = QA + r$, where $0 \leq r < A$. Then the set*

$$\{x \pmod A \mid -B \leq x \leq B\} = \bigcup_{i=0}^{A-1} [i].$$

Here $[i]$ is equivalence class of i of modulo A. Furthermore, there exist $A - r$ equivalence classes $[i]$ whose cardinality is Q, and the remaining r equivalence classes have cardinality $Q + 1$.

Proof. The set $\{x \pmod A \mid -B \leq x \leq B\}$ is Q copies of a complete residue system modulo A. And there are an additional r distinct residue classes that will each have one more element. □

Lemma 29. *Assume that n, A, B are positive integers with $A < 2B+1$. Let $X = (X_1, \ldots, X_n)$ be the random vector with i.i.d. $X_i = U_B \coloneqq U_{\{x \pmod A | -B \le x \le B\}}$, $Z = (Z_1, \ldots, Z_n)$ be a random vector with a random variable Z_i on the set $\{0, \ldots, A-1\}$, and $U = (U_1, \ldots, U_n)$ be a random vector with i.i.d. $U_i = U_{\{0,1,\ldots,A-1\}}$ for all $1 \le i \le n$. Then*

$$\mathrm{SD}\big(X + Z \pmod A, U\big) \le \frac{n}{4\lfloor (2B+1)/A \rfloor}.$$

Proof. If $A \mid (2B+1)$, then Lemma 28 says that U_B is the uniform random variable on the set $\{0, 1, \ldots, A-1\}$. Therefore, the $\mathrm{SD}\big(X + Z \pmod A, U\big) = 0$, since for all $1 \le i \le n$, $X_i + Z_i$ is also the uniform variable on the set $\{0, \ldots, A-1\}$. Therefore, we only need to consider the case $A \nmid (2B+1)$. Write $2B + 1 = QA + r$ with $0 < r < A$. In particular, when $n = 1$, Lemma 27 says that

$$\mathrm{SD}(X_i + Z_i \pmod A, U_i) \le \mathrm{SD}(X_i, U_i).$$

Now, by Lemma 28, we have:

$$\begin{aligned}
\mathrm{SD}(X_i, U_i) &= \frac{1}{2}\left[r\left|\frac{Q+1}{2B+1} - \frac{1}{A}\right| + (A-r)\left|\frac{Q}{2B+1} - \frac{1}{A}\right|\right] \\
&= \frac{1}{2}\left[r\frac{A(Q+1) - (QA+r)}{(QA+r)A} + (A-r)\frac{(QA+r) - AQ}{(QA+r)A}\right] \\
&= \frac{1}{2(QA+r)A}\left[r(A-r) + (A-r)r\right] = \frac{r(A-r)}{(QA+r)A}.
\end{aligned}$$

To bound this expression, we use simple inequalities. The numerator is maximized when r is close to $A/2$, so $r(A-r) \le A^2/4$. The denominator is minimized when $r = 0$, so $(QA+r)A \ge QA^2$. Combining these gives:

$$\mathrm{SD}(X_i, U_i) = \frac{r(A-r)}{(QA+r)A} \le \frac{A^2/4}{QA^2} = \frac{1}{4Q}.$$

Therefore, Lemma 26 implies that

$$\mathrm{SD}(X + Z \pmod A, U) \le \frac{n}{4Q}.$$

□

We note that while we opted for an elementary arithmetic proof for Lemma 29 to keep the exposition self-contained, the exact same bound can also be elegantly derived using discrete Fourier analysis over the finite group $\mathbb{Z}_A$.

Proposition 6. *Assume that n, A, B are positive integers with $A < 2B+1$. Let $X = (X_1, \ldots, X_n)$ be a random vector representing the output of $\mathcal{F}_{Zero}(n, B)$, and $U = (U_1, \ldots, U_{n-1}, -\sum_{i=1}^{n-1} U_i)$ be a random vector with $U_i = U_{\{0,1,\ldots,A-1\}}$*

are i.i.d. for all $1 \le i \le n-1$. *For any vector* $s = (s_1, \ldots, s_n)$, *where* $s_i \in \{0, 1 \ldots, A-1\}$ *for all* $1 \le i \le n$. *Then*

$$\mathrm{SD}\big(X + s \pmod A, U + s \pmod A\big) \le \frac{n-1}{4\lfloor (2B+1)/A \rfloor}.$$

Here $\lfloor \cdot \rfloor$ *is the floor function.*

Proof. Note that the definition of $\mathcal{F}_{\mathrm{Zero}}(n, B)$ implies $\sum_{i=1}^{n} X_i = 0$, and thus $X_n = -\sum_{i=1}^{n-1} X_i$. For each $i \in [n-1]$, the random variable X_i can be represented as $Z_i + U_{B,i}$, where $U_{B,i}$ are independent uniformly distributed random variables on $\{-B, \ldots, B\}$.

Let $f : \mathbb{Z}_A^{n-1} \to \mathbb{Z}_A^n$ be the injective map defined by $(x_1, \ldots, x_{n-1}) \mapsto (x_1, \ldots, x_{n-1}, -\sum_{i=1}^{n-1} x_i)$. By the translation invariance of statistical distance (Lemma 27) and its properties under injective mappings, we derive:

$$\begin{aligned}
&\mathrm{SD}(X + s \pmod A, U + s \pmod A) \\
=&\mathrm{SD}(X \pmod A, U) \\
=&\mathrm{SD}\big(f(X_1 \pmod A, \ldots, X_{n-1} \pmod A), f(U_1, \ldots, U_{n-1})\big) \\
=&\mathrm{SD}\big((X_1 \pmod A, \ldots, X_{n-1} \pmod A), (U_1, \ldots, U_{n-1})\big) \\
=&\mathrm{SD}\big(((Z_1 + U_{B,1} \pmod A, \ldots, Z_{n-1} + U_{B,n-1} \pmod A), (U_1, \ldots, U_{n-1})\big) \\
\le&\frac{n-1}{4\lfloor (2B+1)/A \rfloor},
\end{aligned}$$

where the final inequality follows from Lemma 29. □

Proposition 7. *Let* A, B, n *be positive integers with* $B > n$ *for all* $n \ge 1$. *Assume that* $X = (X_1, \ldots, X_n)$, $X' = (X'_1, \ldots, X'_n)$ *are* n*-dimensional random vectors, where* $X_i = X'_i = U_{\{0,1,\ldots,AB-1\}}$ *are i.i.d.. For any* n*-dimensional vector* $s = (s_i)$ *with* $-A \le s_i \le A$ *for all* $1 \le i \le n$, *the statistical distance between* X *and* $X' - s$ *is at most* $\frac{n}{B}$.

Proof. If $0 \le s_n \le A$, by the definition of statistical distance, we have

$$\begin{aligned}
&\mathrm{SD}(X, X'-s)) \\
=&\sum_{\substack{x=(x_i)\\ x_i\in\mathbb{Z}}} \frac{1}{2}\left|\Pr[X=x]-\Pr[X'=s+x]\right| \\
=&\sum_{x=(x_1,\ldots,x_{n-1})} \frac{1}{2}\Bigg(\sum_{x_n=-s_n}^{-1} + \sum_{x_n=0}^{AB-s_n-1} \\
&\qquad + \sum_{x_n=AB-s_n}^{AB-1}\left|\prod_{i=1}^{n}\Pr[X_i=x_i]-\prod_{i=1}^{n}\Pr[X'_i=s_i+x_i]\right|\Bigg) \\
=&\frac{1}{2}\sum_{x=(x_1,\ldots,x_{n-1})}\Bigg(\frac{s_n}{AB}\prod_{i=1}^{n-1}\Pr[X'_i=s_i+x_i]+\frac{s_n}{AB}\prod_{i=1}^{n-1}\Pr[X_i=x_i] \\
&\qquad + \frac{AB-s_n}{AB}\left|\prod_{i=1}^{n-1}\Pr[X_i=x_i]-\prod_{i=1}^{n-1}\Pr[X'_i=s_i+x_i]\right|\Bigg) \\
=&\frac{|s_n|}{AB}+\left(1-\frac{|s_n|}{AB}\right)\sum_{x=(x_1,\ldots,x_{n-1})}\frac{1}{2}\left(\left|\prod_{i=1}^{n-1}\Pr[X_i=x_i]-\prod_{i=1}^{n-1}\Pr[X'_i=s_i+x_i]\right|\right)
\end{aligned}$$

Similarly, if $-A \le s_n < 0$ we have

$$\begin{aligned}
&\mathrm{SD}(X, X'-s)) \\
=&\frac{|s_n|}{AB}+\left(1-\frac{|s_n|}{AB}\right)\sum_{x=(x_1,\ldots,x_{n-1})}\frac{1}{2}\left(\left|\prod_{i=1}^{n-1}\Pr[X_i=x_i]-\prod_{i=1}^{n-1}\Pr[X'_i=s_i+x_i]\right|\right)
\end{aligned}$$

Therefore, for computing $\mathrm{SD}(X, X'-s))$, we can consider the recursive sequence F_n defined as follows, $F_1 = \frac{|s_1|}{AB}$, and $F_n = \frac{|s_n|}{AB} + \left(1-\frac{|s_n|}{AB}\right)F_{n-1}$. It is easy to obtain the closed form is

$$F_n = 1-\prod_{i=1}^{n}\left(1-\frac{|s_i|}{AB}\right) = \sum_{i=1}^{n}\frac{|s_i|}{AB} - \sum_{1\le i<j\le n}\frac{|s_i||s_j|}{(AB)^2} + \ldots - (-1)^n\frac{\prod_{i=1}^{n}|s_i|}{(AB)^n}.$$

To bound this expression, we apply Bernoulli's inequality, which states that for any set of values $\{x_i\}_{i=1}^n$ where $x_i \in [0,1]$, the following holds:

$$\prod_{i=1}^{n}(1-x_i) \ge 1-\sum_{i=1}^{n}x_i.$$

Applying this inequality with $x_i = |s_i|/AB$, we get:

$$F_n = 1-\prod_{i=1}^{n}\left(1-\frac{|s_i|}{AB}\right) \le 1-\left(1-\sum_{i=1}^{n}\frac{|s_i|}{AB}\right) = \sum_{i=1}^{n}\frac{|s_i|}{AB}.$$

Finally, because $|s_i| \leq A$ for all $1 \leq i \leq n$, we obtain the final bound:

$$\mathrm{SD}(X, X' - s)) = F_n \leq \sum_{i=1}^{n} \frac{|s_i|}{AB} \leq \sum_{i=1}^{n} \frac{A}{AB} = \frac{n}{B}.$$

□

Proposition 8. *Let A, B, n be positive integers with $B > n$ for all $n \geq 1$. Assume that $X = (X_1, \ldots, X_n)$, $X' = (X'_1, \ldots, X'_n)$ are n-dimensional random vectors, where $X_i = U_{\{0,1,\ldots,A-1\}}$ and $X'_i = U_{\{0,1,\ldots,B-1\}}$ are i.i.d.. The statistical distance between $X \pmod A$ and $X' \pmod A$ is at most $\frac{n}{4\lfloor B/A \rfloor}$.*

Proof. Using the same proof strategy as in Lemma 29, we only briefly outline the main idea of the proof. Let $B = QA + r$ with $0 \leq r < A$, by Lemma 26 the definition of statistical distance, we have

$$\mathrm{SD}\big(X, X'\big) \leq n \cdot \mathrm{SD}(X_1, X'_1) = \frac{nr(A-r)}{(QA+r)A} \leq \frac{n}{4Q}.$$

□

7.10 Security Proof against Malicious Adversaries

Theorem 6. *Let $\kappa > 0$ be a security parameter. The inputs to all participants $\{\mathcal{P}_i\}_{i=1}^n$ are given as $(N, [p]_{\mathbb{Z}}, [q]_{\mathbb{Z}})$. If $p_{min} \geq 11$, then the Protocol $\pi_{\mathrm{BI}}^{\mathcal{M}}$ $(n-1)$-*securely computes *the functionality $\mathcal{F}_{\mathrm{BI}}^{\mathcal{M}}$* with abort *in the $\mathcal{F}_{\mathrm{ComCompute}}$, $\mathcal{F}_{\mathrm{Zero}}$, $\mathcal{F}_{\mathrm{CT}}$, $\mathcal{F}_{\mathrm{Com}}$-hybrid model. Here $p_{\min}$ is the minimal prime factor of N.*

Proof. Let $\mathcal{A}$ be a real-world adversary, and let $\mathcal{P}^*$ denote the set of corrupted parties, and $\overline{\mathcal{P}^*}$ denote the set of honest parties. We construct the simulator $\mathcal{S}$ that operates as follows:

1. $\mathcal{S}$ simulates $\mathcal{F}_{\mathrm{ComCompute}}$ to obtain $(\mathtt{commit}, i, (\mathfrak{p}'_i, \mathfrak{q}'_i, \{\tau_{i,j}\}_{j=1}^{\kappa_{\min}}))$ for each $i \in \mathcal{P}^*$, where the parameter $\kappa_{\min}$ is defined as:

$$\kappa_{\min} \coloneqq \left\lceil \kappa \cdot \left(\log_2 \frac{8p_{\min} - 24}{5p_{\min} - 10} \right)^{-1} \right\rceil. \tag{11}$$

2. $\mathcal{S}$ sends $\{(N, \mathfrak{p}'_i, \mathfrak{q}'_i)\}_{i \in \mathcal{P}^*}$ to $\mathcal{F}_{\mathrm{BI}}^{\mathcal{M}}$ to obtain either `BlumInteger` or an error output $E \coloneqq (\mathtt{NonBlumInteger}, \{\mathfrak{p}_i, \mathfrak{q}_i\}_{i \in [n]})$. If E is received, $\mathcal{S}$ follows the honest parties' strategies in $\pi_{\mathrm{BI}}^{\mathcal{M}}$ using shares $\{\mathfrak{p}_i, \mathfrak{q}_i\}_{i \notin \mathcal{P}^*}$. Otherwise, $\mathcal{S}$ proceeds as follows.
3. The simulator $\mathcal{S}$ follows the strategy of functionality $\mathcal{F}_{\mathrm{Zero}}(2^{\kappa-3}nN)$, samples $\{r_i\}_{i=1}^n$, and sends $\{r_i\}_{i \in \mathcal{P}^*}$ to $\mathcal{A}$.

4. For each $1 \le j \le \kappa_{\min}$, the simulator $\mathcal{S}$ randomly samples $v_j, w_j \in \mathbb{Z}_N$ such that $\gcd(v_j^2 - w_j^2 D, N) = 1$, and $b_j \in \{0,1\}$. It then sets $a_j = \frac{v_j + w_j\sqrt{D}}{v_j - w_j\sqrt{D}}$, $P_j \in \mathbb{Z}_N$ such that the two roots of the polynomial $x^2 - P_j x + Q_j$ are

$$\beta_j \coloneqq \frac{\sqrt{D}}{a_j^2 \cdot (-1)^{b_j} - 1} \quad \text{and} \quad \alpha_j \coloneqq \beta_j + \sqrt{D}.$$

   The simulator $\mathcal{S}$ sends (P_j, Q_j) to $\mathcal{A}$ for each $1 \le j \le \kappa_{\min}$.
5. The simulator $\mathcal{S}$ simulates $\mathcal{F}_{\mathrm{Com}}$ to receive $(\mathtt{commit}, i, \{y_{i,j}\}_{j=1}^{\kappa_{\min}}, \{1, \ldots, n\})$ from $\mathcal{A}$ for each $i \in \mathcal{P}^*$.
6. Let $y'_{i,j}$ denote the value of, for $i \in \mathcal{P}^*$, (noting that $\left(\frac{-1}{p}\right) = \left(\frac{-1}{q}\right) = -1$ for Blum integers):

$$\begin{cases} (\alpha_j \beta_j^{-1})^{r_i + (\mathfrak{p}'_i(\frac{-1}{q}) + \mathfrak{q}'_i(\frac{-1}{p}) + 6)/4}, & \text{if } i = 1; \\ (\alpha_j \beta_j^{-1})^{r_i + (\mathfrak{p}'_i(\frac{-1}{q}) + \mathfrak{q}'_i(\frac{-1}{p}))/4}, & \text{if } 2 \le i \le n. \end{cases}$$

   Let i' be the minimal number such that $\mathcal{P}_{i'} \in \overline{\mathcal{P}^*}$. The simulator $\mathcal{S}$ randomly samples $r'_i \leftarrow \mathbb{Z}_{2^{\kappa-2}nN}$ and sets $y_{i,j} \coloneqq (\alpha_j \beta_j^{-1})^{r'_i}$ for each $i \in \overline{\mathcal{P}^*} \setminus \{i'\}$ and $1 \le j \le \kappa_{\min}$. Additionally, for $1 \le j \le \kappa_{\min}$, $\mathcal{S}$ sets $y_{i',j}$ be the value such that

$$(\alpha_j \beta_j^{-1})^{(N-5)/4} \prod_{i \in \overline{\mathcal{P}^*}} y_{i,j} \prod_{i \in \mathcal{P}^*} y'_{i,j} \equiv (-1)^{b_j} \pmod N,$$

   and sends $(\mathtt{decommitted}, i, \{y_{i,j}\}_{j=1}^{\kappa_{\min}})$ to $\mathcal{A}$ for each $i \in \overline{\mathcal{P}^*}$.
7. The simulator $\mathcal{S}$ checks

$$(\alpha_j \beta_j^{-1})^{(N-5)/4} \prod_{i=1}^{n} y_{i,j} \equiv \pm 1 \pmod N$$

   for each $1 \le j \le \kappa_{\min}$. If any check fails, then $\mathcal{S}$ sends $\mathtt{cheat}$ to the functionality $\mathcal{F}_{\mathrm{BI}}^{\mathcal{M}}$ and outputs the output of $\mathcal{A}$.
8. For each $1 \le j \le \kappa_{\min}$, the simulator $\mathcal{S}$ randomly samples $c_j \leftarrow \{0,1\}$ and $\zeta_{i,j} \leftarrow \mathbb{Z}_{2^{2\kappa-1}n^3 N \kappa_{\min}}$ for each $i \in \overline{\mathcal{P}^*}$. It sends $\{\gamma_{i,j}\}_{j=1}^{\kappa_{\min}} \coloneqq \{(\alpha_j \beta_j^{-1})^{\zeta_{i,j}} \cdot y_{i,j}^{-c_j}\}_{j=1}^{\kappa_{\min}}$ for each $i \in \overline{\mathcal{P}^*}$ to $\mathcal{A}$ and receives $\{\gamma_{i,j}\}_{j=1}^{\kappa_{\min}}$ from $\mathcal{A}$ for each $i \in \mathcal{P}^*$.
9. The simulator $\mathcal{S}$ simulates $\mathcal{F}_{\mathrm{CT}}$, and sends $\{c_j\}_{j=1}^{\kappa_{\min}}$ to $\mathcal{A}$.
10. The simulator $\mathcal{S}$ sends $\{\zeta_{i,j}\}_{j=1}^{\kappa_{\min}}$ to $\mathcal{A}$ for each $i \in \overline{\mathcal{P}^*}$ and receives $\{\zeta_{i,j}\}_{j=1}^{\kappa_{\min}}$ for each $i \in \mathcal{P}^*$ from $\mathcal{A}$.
11. The simulator $\mathcal{S}$ checks

$$\prod_{i=1}^{n} (\alpha_j \beta_j^{-1})^{\zeta_{i,j}} \equiv \prod_{i=1}^{n} \gamma_{i,j} \cdot y_{i,j}^{c_j} \pmod N$$

   for each $1 \le j \le \kappa_{\min}$. If any check fails, then $\mathcal{S}$ sends $\mathtt{cheat}$ to $\mathcal{F}_{\mathrm{BI}}^{\mathcal{M}}$and outputs the output of $\mathcal{A}$.

12. The simulator $\mathcal{S}$ simulates $\mathcal{F}_{\text{ComCompute}}$ to obtain $(\texttt{compute}, 1, \{1, \ldots, n\}, C)$, if C is the circuit described in $\pi_{\text{BI}}^{\mathcal{M}}$ step 11 and $(\tau_{i,j}, \zeta_{i,j}, c_j, \mathfrak{p}'_i, \mathfrak{q}'_i)$ satisfy the equations

$$\begin{cases} \tau_{i,j} = \zeta_{i,j} + c_j \cdot \big(r_i - (\mathfrak{p}'_i + \mathfrak{q}'_i - 6)/4\big), & \text{if } i = 1; \\ \tau_{i,j} = \zeta_{i,j} + c_j \cdot \big((r_i - (\mathfrak{p}'_i + \mathfrak{q}'_i)/4\big), & \text{if } 2 \le i \le n, \end{cases}$$

for each $i \in \mathcal{P}^*$ and $1 \le j \le \kappa_{\min}$, then randomly samples $z \leftarrow \mathbb{Z}_N$ and sends $(\texttt{result}, 1, z)$ to $\mathcal{A}$. Otherwise, $\mathcal{S}$ sends `cheat` to $\mathcal{F}_{\text{BI}}^{\mathcal{M}}$ and outputs the output of $\mathcal{A}$.

13. The simulator $\mathcal{S}$ sends `proceed` to $\mathcal{F}_{\text{BI}}^{\mathcal{M}}$ and outputs the output of $\mathcal{A}$.

We proceed with the proof by considering two cases, depending on the value obtained by the simulator $\mathcal{S}$ from $\mathcal{F}_{\text{BI}}^{\mathcal{M}}$ in Step 2.

Case 1: $\mathcal{S}$ obtains `BlumInteger` from $\mathcal{F}_{\text{BI}}^{\mathcal{M}}$.

To demonstrate the indistinguishability of the real and ideal world distributions, we employ the following hybrid worlds, where $\mathcal{H}_i$ denotes the joint distribution of the simulator's output and $\mathcal{F}_{\text{BI}}^{\mathcal{M}}$'s output.

Hybrid 0: The ideal world $\mathcal{H}_0$.

Hybrid 1: This hybrid world $\mathcal{H}_1$ is the same as $\mathcal{H}_0$, except that:

(1) The simulator $\mathcal{S}_1$ is given the honest parties' shares $\{\mathfrak{p}_i, \mathfrak{q}_i\}_{i \in \overline{\mathcal{P}^*}}$ as auxiliary input.

(2) In Step 4, $\mathcal{S}_1$ randomly samples (P_j, Q_j) from $\mathcal{Z}^{+1}(1, N)$ for each $1 \le j \le \kappa_{\min}$.

(3) In Step 6, the values $y_{i,j}$ are randomly sampled from $\mathbb{Z}_N^\times$. Let us define the reconstructed prime shares as $\tilde{p} = \sum_{i \in \mathcal{P}^*} \mathfrak{p}'_i + \sum_{i \in \overline{\mathcal{P}^*}} \mathfrak{p}_i$ and $\tilde{q} = \sum_{i \in \mathcal{P}^*} \mathfrak{q}'_i + \sum_{i \in \overline{\mathcal{P}^*}} \mathfrak{q}_i$. The values $y_{i,j}$ are sampled such that

$$(\alpha_j \beta_j^{-1})^{(N-5)/4} \prod_{i \in \overline{\mathcal{P}^*}} y_{i,j} \prod_{i \in \mathcal{P}^*} y'_{i,j} \equiv (\alpha_j \beta_j^{-1})^{(\tilde{p}-1)(\tilde{q}-1)/4} \pmod{N}.$$

(4) In Step 7, the check is replaced by, for each $1 \le j \le \kappa_{\min}$,

$$(\alpha_j \beta_j^{-1})^{(N-5)/4} \prod_{i=1}^{n} y_{i,j} \equiv (\alpha_j \beta_j^{-1})^{(\tilde{p}-1)(\tilde{q}-1)/4} \pmod{N},$$

where $\tilde{p}$ and $\tilde{q}$ are the reconstructed prime shares as defined in Step 6.

Hybrid 2: This hybrid world $\mathcal{H}_2$ is the same as $\mathcal{H}_1$, except that the values $y_{i,j}$ in Step 6 are defined as follows:

$$\begin{cases} (\alpha_j \beta_j^{-1})^{r_i + (\mathfrak{p}'_i(\frac{-1}{q}) + \mathfrak{q}'_i(\frac{-1}{p}) + 6)/4}, & \text{if } i = 1, \\ (\alpha_j \beta_j^{-1})^{r_i + (\mathfrak{p}'_i(\frac{-1}{q}) + \mathfrak{q}'_i(\frac{-1}{p}))/4}, & \text{if } 2 \le i \le n, \end{cases}$$

for each $1 \le j \le \kappa_{\min}$.

Hybrid 3: This hybrid world $\mathcal{H}_3$ is the same as $\mathcal{H}_2$, except that in Step 8, the simulator $\mathcal{S}_3$ randomly samples $\tau_{i,j} \leftarrow \mathbb{Z}_{2^{2\kappa-1}n^3 N\kappa_{\min}}$ and sets $\gamma_{i,j} \coloneqq (\alpha_j\beta_j^{-1})^{\tau_{i,j}}$ and

$$\zeta_{i,j} \coloneqq \begin{cases} \tau_{i,j} - c_j \cdot \big((r_i - (\mathfrak{p}_i + \mathfrak{q}_i - 6)/4\big), & \text{if } i = 1, \\ \tau_{i,j} - c_j \cdot \big((r_i - (\mathfrak{p}_i + \mathfrak{q}_i)/4\big), & \text{if } 2 \le i \le n \end{cases}$$

for each $1 \le j \le \kappa_{\min}$ and each $i \in \overline{\mathcal{P}^*}$.

Hybrid 4: The real world $\mathcal{H}_4$.

$\mathcal{H}_0 \overset{\text{c}}{\equiv} \mathcal{H}_1$: The distributions of $\mathcal{H}_0$ and $\mathcal{H}_1$ are identical; an analogous proof establishing this indistinguishability is provided in the privacy proof of Theorem 3.

$\mathcal{H}_1 \overset{\text{c}}{\equiv} \mathcal{H}_2$: We first observe that since N is an RSA modulus, $(\alpha_j\beta_j^{-1})^{e_4} \equiv \pm 1 \pmod N$ for $1 \le j \le \kappa_{\min}$. This implies that the order of $\alpha_j\beta_j^{-1}$ is at most $2e_4$ for all j. To bound the statistical distance SD between $\mathcal{H}_1$ and $\mathcal{H}_2$, we employ Proposition 6 with $A = 2e_4$ and $B = 2^{\kappa-3}nN$, and Proposition 8 with $A = 2e_4$ and $B = 2^{\kappa-2}nN$, yielding:

$$\mathrm{SD}(\mathcal{H}_1, \mathcal{H}_2) \le \frac{n}{4\lfloor(2^{\kappa-2}nN + 1)/(2e_4)\rfloor} + \frac{n}{4\lfloor(2^{\kappa-2}nN)/(2e_4)\rfloor} \le \frac{1}{2^{\kappa+1}}.$$

$\mathcal{H}_2 \overset{\text{c}}{\equiv} \mathcal{H}_3$: Proposition 7 demonstrates that the statistical distance between the distributions of $\zeta_{i,j}$ generated in **Hybrid 2** and **Hybrid 3** is at most $\frac{n\kappa_{\min}}{2^{\kappa+1}n\kappa_{\min}} < \frac{1}{2^{\kappa+1}}$, by setting $A = 2^{\kappa-2}n^2N$ and $B = 2^{\kappa+1}n\kappa_{\min}$. We note that $-A \le c_j\big(r_i - (\mathfrak{p}_i + \mathfrak{q}_i)/4\big) \le A$ for all $1 \le i \le n$ and $1 \le j \le \kappa_{\min}$, given that $\mathfrak{p}_i, \mathfrak{q}_i \ge 0$ and $N = (\sum_{i=1}^n \mathfrak{p}_i) \cdot (\sum_{i=1}^n \mathfrak{q}_i)$. Consequently, the joint distributions of $(\gamma_{i,j}, c_j, \zeta_{i,j})$ generated in **Hybrid 2** and **Hybrid 3** are indistinguishable, and

$$\mathrm{SD}(\mathcal{H}_2, \mathcal{H}_3) \le \frac{1}{2^{\kappa+1}}.$$

$\mathcal{H}_3 \overset{\text{c}}{\equiv} \mathcal{H}_4$: Note that the view of the real-world adversary $\mathcal{A}$ in **Hybrid 3** and in the real world is identical. Additionally, $\mathcal{S}_3$ sends `cheat` to $\mathcal{F}_{\text{BI}}^{\mathcal{M}}$ if and only if the honest parties output 0. Hence, the distributions of $\mathcal{H}_3$ and $\mathcal{H}_4$ are identical.

Case 2: $\mathcal{S}$ obtains $(\texttt{NonBlumInteger}, \{\mathfrak{p}_i, \mathfrak{q}_i\}_{i=1}^n)$ from $\mathcal{F}_{\text{BI}}^{\mathcal{M}}$.

Since $\mathcal{S}$ possesses the honest parties' shares and faithfully follows $\pi_{\text{BI}}^{\mathcal{M}}$, it suffices to prove that the parties output `NonBlumInteger` except with negligible probability in the real execution. Note that all check conditions of $\mathcal{F}_{\text{BI}}^{\mathcal{M}}$, except condition 3 (i.e., that $p \ne q$ and both p and q are primes), are directly checked in $\pi_{\text{BI}}^{\mathcal{M}}$. It remains to show that the parties output `NonBlumInteger` in a real execution except with negligible probability if $(\sum_{i\in\mathcal{P}^*} \mathfrak{p}'_i + \sum_{i\in\overline{\mathcal{P}^*}} \mathfrak{p}_i) \cdot (\sum_{i\in\mathcal{P}^*} \mathfrak{q}'_i + \sum_{i\in\overline{\mathcal{P}^*}} \mathfrak{q}_i)$ is not a Blum integer.

We now consider the probability that $\mathcal{A}$ passes both Step 7 and Step 10. Consider the check in Step 10. Let $\delta_{1,j}$ and $\delta_{2,j}$ be the offsets such that

$$(\alpha_j\beta_j^{-1})^{(N-5)/4}\prod_{i\in\overline{\mathcal{P}^*}} y_{i,j}\prod_{i\in\mathcal{P}^*} y'_{i,j}\cdot(\alpha_j\beta_j^{-1})^{\delta_{1,j}} \equiv \pm 1 \pmod N,$$

$$(\alpha_j\beta_j^{-1})^{\delta_{2,j}}\cdot\prod_{i=1}^{n}\gamma_{i,j} \equiv \prod_{i=1}^{n}(\alpha_j\beta_j^{-1})^{\tau_{i,j}} \pmod N$$

for all $1 \le j \le \kappa_{\min}$.

Combining this with the check on $\zeta_{i,j}$ in $\mathcal{F}_{\mathrm{BI}}^{\mathcal{M}}$, we have

$$(\alpha_j\beta_j^{-1})^{\delta_{2,j}} \equiv (\alpha_j\beta_j^{-1})^{c_j\cdot\delta_{1,j}} \pmod N$$

for all $1 \le j \le \kappa_{\min}$. Since the adversary $\mathcal{A}$ must choose $\delta_{1,j}$ and $\delta_{2,j}$ before it learns c_j, for each j, the probability that adversary $\mathcal{A}$ passes Step 10 in $\pi_{\mathrm{BI}}^{\mathcal{M}}$ with $\delta_{1,j} \neq 0$ is at most $\frac{1}{2}$. Using Theorem 2, the probability that $\mathcal{A}$ must set non-zero $\delta_{1,j}$ is $\frac{1}{4} + \frac{1.25}{p_{\min}-3}$ for each j. Therefore, the probability that $\mathcal{A}$ passes Step 7 and Step 10 is at most

$$\frac{1}{4} + \frac{1.25}{p_{\min}-3} + \left(1 - \left(\frac{1}{4} + \frac{1.25}{p_{\min}-3}\right)\right)\cdot\frac{1}{2} = \frac{5}{8} + \frac{0.625}{p_{\min}-3}$$

for each j. We have

$$\begin{aligned}&\Pr[\pi_{\mathrm{BI}}^{\mathcal{M}} \text{ outputs } \texttt{BlumInteger} \mid N \text{ is not a Blum integer}]\\ &\le \left(\frac{5}{8} + \frac{0.625}{p_{\min}-3}\right)^{\kappa_{\min}} \le \frac{1}{2^{\kappa}}.\end{aligned} \tag{12}$$

Finally, considering **Case 1** (which utilizes the triangle inequality) and **Case 2**, for any non-uniform probabilistic polynomial-time distinguisher $\mathcal{D}$, it follows that:

$$\begin{aligned}&\Big|\Pr\Big[\mathcal{D}\big(\{\mathrm{REAL}_{\pi_{\mathrm{BI}}^{\mathcal{M}},\mathcal{A},\mathcal{P}^*}(\lambda,\{(\mathfrak{p}_i,\mathfrak{q}_i,N)\}_{i=1}^n,\mathbf{aux})\}_{\lambda,\mathfrak{p}_i,\mathfrak{q}_i,N,\mathbf{aux}},\mathbf{aux}\big)\Big]\\ &-\Pr\Big[\mathcal{D}\big(\{\mathrm{IDEAL}_{\mathcal{F}_{\mathrm{BI}}^{\mathcal{M}},\mathcal{S},\overline{\mathcal{P}^*}}(\lambda,\{(\mathfrak{p}_i,\mathfrak{q}_i,N)\}_{i=1}^n,\mathbf{aux})\}_{\lambda,\mathfrak{p}_i,\mathfrak{q}_i,N,\mathbf{aux}},\mathbf{aux}\big)\Big]\Big| \le \frac{1}{2^{\kappa}}.\end{aligned}$$

□

7.11 Missing Functionalities, Protocols and Algorithms

The functionality describes that each party $\mathcal{P}_i$ has two shares, $\mathfrak{x}_i$ and $\mathfrak{y}_i$, the functionality outputs $\mathfrak{z}_i$ where $[z]_N = [xy]_N$ and assigns to $\mathcal{P}_i$.

Functionality 5 Modular Multiplication $\mathcal{F}_{\text{ModMul}}(n)$

Inputs: Each party $\mathcal{P}_i$ has shares $[x]_N$, $[y]_N$ and N.

Outputs: Each party has shares of $[z]_N = [x \cdot y]_N$, with uniformly random $\mathfrak{z}_i \in \mathbb{Z}_N$ for all $1 \leq i \leq n$.

The functionality below ensures that participants can learn $\prod_i y_i$ without revealing their own y_i. In our setting, we define the finite group G utilized by the functionality $\mathcal{F}_{\text{Shuffle}}$ as the group of invertible elements in the formal quadratic ring $\mathcal{O}_D/N\mathcal{O}_D$:

$$G \coloneqq \left\{ a + b\sqrt{D} \in \mathcal{O}_D/N\mathcal{O}_D \mid a, b \in \mathbb{Z}_N, a^2 - b^2 D \in \mathbb{Z}_N^\times \right\}.$$

Note that the inverse of $x = v + w\sqrt{D} \in G$ is well-defined and given by $x^{-1} = (v - w\sqrt{D})(v^2 - w^2 D)^{-1} \in G$, and the identity is 1.

Functionality 6 $\mathcal{F}_{\text{Shuffle}}(n, G)$

Inputs: Each party $\mathcal{P}_i$ has y_i in a finite group G.

Outputs: Each party $\mathcal{P}_i$ receives $y \coloneqq \prod_{i=1}^{n} y_i \in G$.

In the following protocol [7], each party splits their own input y_i into $n-1$ partitions and randomly send one share to other parties to avoid revealing their own input y_i. Every party will calculate the product of all obtained shares $\prod_i z_i$ and publish it. Eventually, we have $\prod_{i=1}^n z_i = \prod_{i=1}^n y_i$.

Protocol 4 Shuffle $\pi_{\text{Shuffle}}(n)$

Inputs: Each party $\mathcal{P}_i$ has $y_i \in G$.

Outputs: $\prod_{i=1}^{n} y_i \in G$.

1. Each party $\mathcal{P}_i$ randomly chooses $x_{i,j} \in G$ for all $1 \leq j \leq n$ such that $\prod_{j=1}^{n} x_{i,j} = 1$ (i.e., randomly chooses $x_{i,j}$ for $1 \leq j \leq n-1$ and $x_{i,n}^{-1} \coloneqq \prod_{j=1}^{n-1} x_{i,j}$). Set $y_{i,1} \coloneqq x_{i,1} \cdot y_i$ and $y_{i,j} \coloneqq x_{i,j}$ for all $2 \leq j \leq n$. Send $y_{i,j}$ to the party $\mathcal{P}_j$ for all $1 \leq j \neq i \leq n$.

2. Each party $\mathcal{P}_i$ computes $z_i \coloneqq \prod_{j=1}^{n} y_{j,i}$. Broadcast z_i to the other party $\mathcal{P}_j$.

3. Outputs $z \coloneqq \prod_{i=1}^{n} z_i$.

If $G = \left\{ a + b\sqrt{D} \mid a, b \in \mathbb{Z}_N, a^2 - b^2 D \in \mathbb{Z}_N^\times \right\}$, the parties choose $x_{i,j}$ by randomly selecting $v_{i,j}, w_{i,j} \in \mathbb{Z}_N$ such that $v_{i,j}^2 - w_{i,j}^2 D \in \mathbb{Z}_N^\times$ and setting $x_{i,j} \coloneqq (v_{i,j} + w_{i,j}\sqrt{D})$. In our setting, all inputs are norm 1 (i.e., $y_i = \alpha_i \beta_i^{-1}$) elements of G.

Security. This protocol is a direct implementation of the secure product protocol from Boneh and Franklin [10], and is secure against semi-honest adversaries. The security stems from the masking provided by the random shares. From the perspective of any party $\mathcal{P}_i$, the value it computes, $z_i = \prod_{j=1}^{n} y_{j,i}$, is a product of n shares. One share, $y_{i,i} = x_{i,i} y_i$, contains its own masked secret, while the other $n-1$ shares are received from the other parties.

In a setting with at most $n-1$ corruptions, there is at least one honest party other than $\mathcal{P}_i$ (unless $\mathcal{P}_i$ is the sole honest party). Let this honest party be $\mathcal{P}_j$. The share $y_{j,i}$ sent from the honest party $\mathcal{P}_j$ to $\mathcal{P}_i$ is a uniformly random group element, since $x_{j,i}$ is chosen uniformly at random. Multiplying by a single uniformly random element from a group randomizes the entire product. Therefore, the computed value z_i is statistically independent of the secret inputs of other honest parties. Since the broadcast values $\{z_j\}$ are similarly randomized products, the view of any party reveals no information beyond what can be inferred from the final public product $\prod y_i$.

Functionality 7 $\mathcal{F}_{\text{ComCompute}}(n)$

Input Commitment: Upon receiving $(\texttt{commit}, \texttt{id}, x)$ from party $\mathcal{P}_i$, if $\texttt{id}$ is a fresh value, then store $(\texttt{value}, \texttt{id}, i, x)$ in memory, and send $(\texttt{committed}, \texttt{id}, i)$ to all other parties.

Computation: Upon receiving $(\texttt{compute}, \texttt{id}, \texttt{input-ids}, f)$ from all parties, where $\texttt{id}$ is a fresh, agreed upon value, and where $\texttt{input-ids}$ is a vector of IDs such that for every $1 \leq i \leq |\texttt{input-ids}|$ there exists in memory a record of the form $(\texttt{value}, \texttt{input-ids}_i, *, *)$, and where f is the description of a function that takes as input the values associated with the IDs in $\texttt{input-ids}$ and produces as output an n-tuple of values, if the parties disagree upon the function f or the vector $\texttt{input-ids}$, then abort, and otherwise:

1. Let x be a vector of the same length as $\texttt{input} - \texttt{ids}$ such that for each $1 \leq i \leq |\texttt{input-ids}|$, there is a record $(\texttt{value}, \texttt{input-ids}_i, *, v_i)$ in memory, and we set $x_i = v_i$.

2. Compute $(y_1, \ldots, y_n) \coloneqq f(x)$, and then send $(\texttt{result}, \texttt{id}, y_i)$ to each party $\mathcal{P}_i$.

Functionality 8 $\mathcal{F}_{\text{Zero}}(n, B)$

Inputs: Each party $\mathcal{P}_i$ has input `sample`.

Outputs:

1. Uniformly sample $x_{i,j} \leftarrow \{-B, -B+1, \ldots, B\}$ for $1 \leq i, j \leq n$ such that $x_{i,j} + x_{j,i} = 0$.
2. Each party $\mathcal{P}_i$ receives the value $r_i \coloneqq \sum_{j=1}^{n} x_{i,j}$.

Functionality 9 $\mathcal{F}_{\text{CT}}(n, S)$

Inputs: Each party $\mathcal{P}_i$ has input `flip`.

Outputs: Each party $\mathcal{P}_i$ receives the value c, where c is uniformly sampled from the set S.

Functionality 10 $\mathcal{F}_{\text{Com}}(n)$

Commit: On receiving $(\texttt{commit}, \texttt{id}, x, J)$ from party $\mathcal{P}_i$, where $J \subseteq \{1, \ldots n\}$ and $x \in \{0,1\}^*$, if `id` is a fresh value, then store $(\texttt{commitment}, \texttt{id}, x, J, i)$ in memory and send $(\texttt{committed}, \texttt{id}, i)$ to each party $\mathcal{P}_j$ for $j \in J$.

Decommit: On receiving $(\texttt{decommit}, \texttt{id})$ from $\mathcal{P}_i$, if a record of the form $(\texttt{commitment}, \texttt{id}, x, J, i)$ exists in memory, then send $(\texttt{decommitted}, \texttt{id}, x)$ to every party $\mathcal{P}_j$ for $j \in J$.

Algorithm π_{VerifyBI} details the verification circuit for our maliciously-secure Protocol 2. It is invoked by $\mathcal{F}_{\text{ComCompute}}$ to ensure protocol transcript consistency.

Algorithm 3 $\pi_{\text{VerifyBI}}(N, \vec{c}, \{\mathfrak{p}_i, \mathfrak{q}_i, \{\tau_{i,j}\}_{j=1}^{\kappa_{\min}}, \{\zeta_{i,j}\}_{j=1}^{\kappa_{\min}}\}_{i=1}^{n})$

1. Sample $r \leftarrow \mathbb{Z}_N$ and compute

$$z = r \cdot \left(-1 + \sum_{i=1}^{n} (\mathfrak{p}_i + \mathfrak{q}_i) \right) \pmod{N}.$$

2. Return z if and only if it holds that

$$\begin{aligned} N = & \left(\sum_{i=1}^{n} \mathfrak{p}_i \cdot \sum_{i=1}^{n} \mathfrak{q}_i \right) \\ & \wedge \sum_{i=1}^{n} \mathfrak{p}_i \equiv \sum_{i=1}^{n} \mathfrak{q}_i \equiv 3 \pmod{4} \\ & \wedge \big(\mathfrak{p}_i \geq 0\big) \wedge \big(\mathfrak{q}_i \geq 0\big) \text{ for all } 1 \leq i \leq n \\ & \wedge \Big(\tau_{1,j} = \zeta_{1,j} - c_j \cdot \big(r_1 - (\mathfrak{p}_1 + \mathfrak{q}_1 - 6)\big)/4\Big) \text{ for all } 1 \leq j \leq \kappa_{\min} \\ & \wedge \Big(\tau_{i,j} = \zeta_{i,j} - c_j \cdot \big(r_i - (\mathfrak{p}_i + \mathfrak{q}_i)/4\big)\Big) \text{ for all } 2 \leq i \leq n \text{ and } 1 \leq j \leq \kappa_{\min}. \end{aligned}$$

If any part of the above predicate does not hold, output `VerifyFail`.

7.12 Three RSA Moduli Protocols in the Semi-honest Model

In this section, we rewrite the Lucas test using macros from [14] to facilitate comparison with the Boneh-Franklin test [10] and Burkhardt's et al.'s [14] protocols. Here, we always assume $p \equiv q \equiv 3 \pmod{4}$. Finally, we note that an RSA modulus refers to N, which is the product of two distinct prime numbers. In contrast, a biprime refers to N being the product of any two prime numbers.

Protocol 5 Lucas Biprimality test type (n)

Inputs: Each party $\mathcal{P}_i$ has odd integers $[p]_{\mathbb{Z}}$, $[q]_{\mathbb{Z}}$, $D = 1$, and N.

Outputs:

1. Party $\mathcal{P}_1$ randomly chooses $0 \leq P < N$ such that $Q = (P^2 - D)/4$ and $\left(\frac{Q}{N}\right) = 1$. Send this P to the other parties.
2. Party $\mathcal{P}_1$ computes $v_1 \coloneqq g^{(N - \mathfrak{p}_1 - \mathfrak{q}_1 + 1)/4} \pmod{N}$, where $g \coloneqq \frac{P - \sqrt{D}}{P + \sqrt{D}}$. The other parties compute $v_i \coloneqq g^{-(\mathfrak{p}_i + \mathfrak{q}_i)/4} \pmod{N}$. Parties broadcast v_i to compute $v \coloneqq \prod_{i=1}^{n} v_i \pmod{N}$. They then check if

$$v = \prod_{i=1}^{n} v_i \equiv 1 \pmod{N}.$$

   If the test fails, output `NonBlumInteger`.

3. Parties verify $\gcd(N, e) = 1$ as follows:
   3.1 obtain $[r]_N \leftarrow \texttt{RandomSample}(\mathbb{Z}_N)$.
   3.2 compute $[p]_N \leftarrow \texttt{Int} - \texttt{to} - \texttt{mod}(\mathbb{Z}_N, [p]_{\mathbb{Z}})$ and $[q]_N \leftarrow \texttt{Int} - \texttt{to} - \texttt{mod}(\mathbb{Z}_N, [q]_{\mathbb{Z}})$.
   3.3 call $[b]_N \leftarrow \texttt{Mult}(\mathbb{Z}_N, [r]_N, [p]_N + [q]_N - 1)$.
   3.4 obtain $b \leftarrow \texttt{OpenAll}(\mathbb{Z}_N, [b]_N)$. If $b \neq 1$ then output `NonBlumInteger`. Otherwise, output `BlumInteger`.

Below is Boneh-Franklin protocol [10], as cited from the version in [14, FIGURE 7.1].

Protocol 6 Boneh-Franklin biprimality protocol(n)

Inputs: Each party has $[p]_{\mathbb{Z}}$, $[q]_{\mathbb{Z}}$ and N.

Outputs:

1. Party $\mathcal{P}_1$ randomly chooses $g \in \mathbb{Z}_N^{\times}$ and $\left(\frac{g}{N}\right) = 1$. Send this g to the other parties.
2. Party $\mathcal{P}_1$ computes $v_1 \coloneqq g^{(N - \mathfrak{p}_1 - \mathfrak{q}_1 + 1)/4} \pmod N$. The other parties compute $v_i \coloneqq g^{-(\mathfrak{p}_i + \mathfrak{q}_i)/4} \pmod N$. Parties broadcast v_i to compute $v \coloneqq \prod_{i=1}^{n} v_i \pmod N$. They then check if
$$v = \prod_{i=1}^{n} v_i \equiv \pm 1 \pmod N.$$
   If the test fails, output `NonBlumInteger`.
3. Parties verify $\gcd(N, e) = 1$ as follows:
   3.1 obtain $[r]_N \leftarrow \texttt{RandomSample}(\mathbb{Z}_N)$.
   3.2 compute $[p]_N \leftarrow \texttt{Int} - \texttt{to} - \texttt{mod}(\mathbb{Z}_N, [p]_{\mathbb{Z}})$ and $[q]_N \leftarrow \texttt{Int} - \texttt{to} - \texttt{mod}(\mathbb{Z}_N, [q]_{\mathbb{Z}})$.
   3.3 call $[b]_N \leftarrow \texttt{Mult}(\mathbb{Z}_N, [r]_N, [p]_N + [q]_N - 1)$.
   3.4 obtain $b \leftarrow \texttt{OpenAll}(\mathbb{Z}_N, [b]_N)$. If $b \neq 1$ then output `NonBlumInteger`. Otherwise, output `BlumInteger`.

Herein lies Burkhardt's protocol. For further details, please consult [14].

Protocol 7 Miller-Rabin biprimality protocol$(\kappa_{\mathrm{lenP}}, s, n)$

Inputs: Each party has $[p]_{\mathbb{Z}}$, $[q]_{\mathbb{Z}}$, P, Q and N. Here P and Q are primes satisfying $n^2 2^{2\kappa_P} < nP < Q$.

Outputs:

1. Let $G = \emptyset$, for $f \in \{p, q\}$:
   1.1 $\mathcal{P}_n$ uniformly samples $v \in \mathbb{Z}_N$ and broadcasts v.

1.2 Compute $<\gamma>_N$ as follows: Party $\mathcal{P}_1$ sets $\gamma_1 \coloneqq v^{\frac{f_1-1}{2}} \pmod N$. For $2 \leq i \leq n$, $\mathcal{P}_i$ sets $\gamma_i \coloneqq v^{\frac{f_i}{2}} \pmod N$.
1.3 Obtain $[\gamma]_N \leftarrow \texttt{Mul}-\texttt{to}-\texttt{Add}(\mathbb{Z}_N, <\gamma>_N)$.
1.4 Compute $[\gamma+1]_N$ and $[\gamma-1]_N$.
1.5 For $\delta \in \{\gamma+1, \gamma-1\}$, compute

$$[y_\delta]_Q \leftarrow \texttt{Divisible}(\kappa_{\mathrm{lenP}}, s, \mathbb{Z}_P, \mathbb{Z}_Q, [\delta]_N, [f]_{\mathbb{Z}}).$$

1.6 Compute $[y]_Q \leftarrow \texttt{Mult}(\mathbb{Z}_Q, [y_{\gamma+1}]_Q, [y_{\gamma-1}]_Q)$.
1.7 Reveal $y \leftarrow \texttt{OpenAll}(\mathbb{Z}_Q, [y]_Q)$.
1.8 If $y = 0$, set $G = G \cup \{f\}$.
2. If $G = \{p, q\}$ output `BlumInteger`, otherwise output `NonBlumInteger`.

The number of macros used in each test are summarized below.

Table 7. The number of macros in biprimality tests.

	#Random −sample	♯Int − to −mod	♯Mult	♯OpenAll	♯Mult − to -add
Boneh-Franklin [23]	1	2	1	1	0
Miller-Rabin [14]	≥ 2	4	≥ 6	≥ 4	2
Type-(I)	1	2	1	1	0

In addition to the aforementioned, Burkhardt's protocol includes other macros such as **Invert** and **Larger-domain**.

7.13 Adaptation of Chen's Biprime Test Protocol

This section offers several suggestions, notably an optimization for the number of iterations in the Chen et al. protocol [15, Protocol 5.2]. This refinement stems from our updated worst-case soundness error for the Boneh-Franklin test: while the original test yielded a false positive with at most 1/2 probability, our new analysis (cf. Theorem 1) reduces this probability to 1/4. Therefore, assuming the adversary always cheats and considering an honest execution would not have yielded a false positive, the probability of producing a positive outcome in the jth iteration (i.e., Steps 4–6 and 8–11) is upper-bounded by 5/8 (cf. the inequality (12)). Consequently, the probability that the adversary succeeds across all $\lceil 1.475s \rceil$ (i.e., the original result gives $2.5s$) iterations is at most $(5/8)^{1.475s} < 2^{-s}$. A brief description of our suggestion is provided below. A revising protocol from Chen et al. can be found in the end of this subsection.

1. **Lack of Congruence Verification for p and q.**
   The VarifyBiprime algorithm by Chen et al. [15, Algorithm 5.3] does not enforce the condition that both prime factors p and q of $N = pq$ are congruent to 3 (mod 4). This omission presents a vulnerability: an adversary could

select primes $p \equiv q \equiv 1 \pmod 4$, potentially causing specific variants of the Boneh-Franklin test (i.e., or its associated soundness analysis, which often assumes $p \equiv q \equiv 3 \pmod 4$) to behave unexpectedly or fail to provide its intended security guarantees. Indeed, as detailed in Lemma 14, infinitely many RSA moduli $N = pq$ with $p \equiv q \equiv 1 \pmod 4$ consistently pass certain Boneh-Franklin test configurations, even if those configurations are primarily analyzed for $3 \pmod 4$ primes.

To address this, one possible enhancement is to augment the verification procedure with an explicit check for $p \equiv q \equiv 3 \pmod 4$. An alternative approach, applicable if using encoding components (e.g., shares $\mathfrak{p}_i, \mathfrak{q}_i$ of p, q), could be to enforce constraints directly at the encoding stage, such as requiring primary shares $\mathfrak{p}_1, \mathfrak{q}_1 \equiv 3 \pmod 4$ while other shares $\mathfrak{p}_i, \mathfrak{q}_i$ for $i > 1$ satisfy $\mathfrak{p}_i, \mathfrak{q}_i \equiv 0 \pmod 4$.

2. **Mismatch Between Proof Structure and Commitment Definition.** The security proof follows a structure analogous to Schnorr's protocol. Consequently, the definition of the responses $\zeta_{i,j}$ in the challenge-response phase (cf. Step 10) must correctly incorporate the randomness r_i used in the commitment phase (cf. Step 4). Specifically, for commitments $\chi_{i,j}$ (e.g., $\gamma_j^{r_i-(p_i+q_i-6)/4}$ or $\gamma_j^{r_i-(p_i+q_i)/4}$), the corresponding responses must be defined using the actual exponent containing r_i (i.e., $\zeta_{i,j} = \tau_{i,j} + c_j(r_i - (p_i + q_i)/4)$ or $\tau_{i,j} + c_j(r_i - (p_i + q_i - 6)/4)$ respectively). A definition of $\zeta_{i,j}$ that omits r_i (e.g., based on $-(p_i + q_i - 6)/4$ or $-(p_i + q_i)/4$) would not faithfully represent a Schnorr-like proof of knowledge for $\chi_{i,j}$, even if the formal verification equation in Step 10 remains unchanged.

   Moreover, the introduction of r_i in the exponent necessitates an adjustment to the sample space of each blinding factor $\tau_{i,j}$. Since the quantity that $\tau_{i,j}$ must blind becomes either $c_j(r_i - (p_i + q_i)/4)$ or $c_j(r_i - (p_i + q_i - 6)/4)$, the dominant term is now r_i rather than the prime shares p_i and q_i. As a result, the distribution of $\tau_{i,j}$ must be chosen to statistically hide r_i, not merely the smaller shares. Accordingly, in our proposed protocol $\pi_{\mathrm{BI}}^{\mathcal{M}}$ and the corresponding functionality $\mathcal{F}_{\mathrm{BI}}^{\mathcal{M}}$, we remove the upper bound M and eliminate the assumption that $p_i, q_i \leq M$. In contrast, in our revised version of Chen et al.'s protocol (cf. Section 7.13), we retain the bound M and enlarge the population of $\tau_{i,j}$ from $\mathbb{Z}_{2^{s+1}M}$ to $\mathbb{Z}_{\lceil 1.475s \rceil \cdot n 2^{s+1}(n^2 2^{2\kappa+s-3}+M)}$ to accommodate the additional randomness introduced by r_i.

3. **Improper Dependency of Security Parameters.** Intuitively, the output of $\mathcal{F}_{\mathrm{Zero}}$ should depend on the number of participants n, since each party receives the sum of a matrix row, which aggregates inputs from all n participants. As the number of participants increases, the potential range of these row sums naturally grows. Consequently, both the sample space of $\mathcal{F}_{\mathrm{Zero}}$ and the choice of the security parameter $\tau_{i,j}$ should scale with n, rather than being determined solely by a fixed upper bound on the shared values.

   Indeed, by Proposition 6, ensuring the output of $\mathcal{F}_{\mathrm{Zero}}$ is statistically close to $U \coloneqq (U_1, \ldots, U_{n-1}, -\sum_{i=1}^{n-1} U_i)$, where each U_i is uniformly distributed

over $\mathbb{Z}_{\phi(N)}$ for $i \in [n-1]$, requires adjusting the population from $2^{2\kappa+s}$ to $n2^{2\kappa+s-3}$.

4. **Sampling considerations in Boneh-Franklin Simulation.**
   In the Boneh–Franklin test, all parties agree on a common base $g \in \mathbb{Z}_N^\times$ such that $\left(\frac{g}{N}\right) = 1$. Each participant then computes $\chi_i \equiv g^{x_i} \pmod N$ for $1 \le i \le n$, where the exponent x_i is deterministically derived from the party's share in the protocol.
   For the simulation in the security proof, Chen et al. [15] propose sampling each χ_i uniformly from $\mathbb{Z}_N^\times$. However, this approach treats χ_i as independent of g, which contradicts the actual protocol behavior where each χ_i must be a specific power of g. This dependency is essential: for example, if $g = 1$, then $\chi_i = 1$ for all $1 \le i \le n$, whereas a uniform sample from $\mathbb{Z}_N^\times$ would likely yield different values, possibly with $\left(\frac{\chi_i}{N}\right) = -1$, violating the protocol's correctness.
   To ensure a faithful simulation, we instead sample exponents r_i' uniformly from $\mathbb{Z}_{2^{2\kappa+s-2}n}$ and set $\chi_i \equiv g^{r_i'} \pmod N$ for $1 \le i \le n$. This produces values that are statistically close to uniform over the subgroup generated by g, while preserving the necessary algebraic relation to g.

Now, we present a modification of the protocol by Chen et al. [15, Protocol 5.2], building upon the enhancements detailed. For ease of comparison, the notation employed in this section adheres to that of [15].

Protocol 8 Adaptation of Chen's Biprime Test $\pi_{\text{Biprime}}(M, n)$

This protocol is parametrized by an integer M and the number of parties n. In addition, there is a statistical parameter s. The parties have access to the $\mathcal{F}_{\text{CT}}$, $\mathcal{F}_{\text{Com}}$, $\mathcal{F}_{\text{ComCompute}}$, and $\mathcal{F}_{\text{Zero}}$ functionalities.

Input Commitment:

1. Upon receiving input $(\texttt{check}-\texttt{biprimality}, \mathsf{sid}, N, p_i, q_i)$ from the environment, each party $\mathcal{P}_i$ for $i \in [n]$ samples $\boldsymbol{\tau}_{i,j} \leftarrow \mathbb{Z}_{\lceil 1.475s \rceil n 2^{s+1}(n^2 2^{2\kappa+s-3}+M)}$ for $j \in [\lceil 1.475s \rceil]$ and commits to these values, along with its shares of p and q, by sending $(\texttt{commit}, \mathsf{GenSID}(\mathsf{sid}, i), (p_i, q_i, \boldsymbol{\tau}_{i,*}))$ to $\mathcal{F}_{\text{ComCompute}}(n)$.

Boneh-Franklin Test:

2. Each party $\mathcal{P}_i$ for $i \in [n]$ sends $(\texttt{sample}, \mathsf{sid})$ to $\mathcal{F}_{\text{Zero}}(n, n2^{2\kappa+s-3})$ and receives $(\texttt{zero}-\texttt{share}, \mathsf{sid}, r_i)$ in response.
3. For $j \in [\lceil 1.475s \rceil]$, the parties invoke $\mathcal{F}_{\text{CT}}(n, \mathbb{J}_N)$, where $\mathbb{J}_N$ is the subdomain of $\mathbb{Z}_N^*$ that contains only values with Jacobi symbol 1. The parties define vector $\boldsymbol{\gamma}$ that contains the $\lceil 1.475s \rceil$ sampled values.
4. For every $j \in [\lceil 1.475s \rceil]$, party $\mathcal{P}_1$ computes

$$\boldsymbol{\chi}_{1,j} \coloneqq \boldsymbol{\gamma}_j^{r_1-(p_1+q_1-6)/4} \pmod N$$

   and every other party $\mathcal{P}_i$ for $i \in [2, n]$ computes

$$\boldsymbol{\chi}_{i,j} \coloneqq \boldsymbol{\gamma}_j^{r_i-(p_i+q_i)/4} \pmod N.$$

5. Every $\mathcal{P}_i$ for $i \in [n]$ sends $(\texttt{commit}, \mathsf{GenSID}(\mathsf{sid}, i), \boldsymbol{\chi}_{i,*}, [n])$ to $\mathcal{F}_{\mathrm{Com}}(n)$.
6. After being notified that all other parties are committed, each party $\mathcal{P}_i$ for $i \in [n]$ sends $(\texttt{decommit}, \mathsf{GenSID}(\mathsf{sid}, i))$ to $\mathcal{F}_{\mathrm{Com}}(n)$, and in response receives $\boldsymbol{\chi}_{i',*}$ from $\mathcal{F}_{\mathrm{Com}}(n)$ for $i' \in [n] \setminus \{i\}$.
7. The parties output $(\texttt{not} - \texttt{biprime}, \mathsf{sid})$ to the environment and halt if there exists $j \in [\lceil 1.475s \rceil]$ such that

$$\boldsymbol{\gamma}_j^{(N-5)/4} \cdot \prod_{i \in [n]} \boldsymbol{\chi}_{i,j} \not\equiv \pm 1 \pmod{N}.$$

Consistency Check and GCD Test:

8. For $j \in [\lceil 1.475s \rceil]$, each party $\mathcal{P}_i$ for $i \in [n]$ computes

$$\boldsymbol{\alpha}_{i,j} \coloneqq \boldsymbol{\gamma}_j^{\boldsymbol{\tau}_{i,j}} \pmod{N}.$$

   The parties all broadcast the values they have computed to one another.
9. The parties all send $(\texttt{flip}, \mathsf{sid})$ to $\mathcal{F}_{\mathrm{CT}}(n, \{0,1\}^{\lceil 1.475s \rceil})$ to obtain an agreed-upon random bit vector $\boldsymbol{c}$ of length $\lceil 1.475s \rceil$.
10. For $j \in [\lceil 1.475s \rceil]$, party $\mathcal{P}_1$ computes

$$\boldsymbol{\zeta}_{1,j} \coloneqq \boldsymbol{\tau}_{1,j} + \boldsymbol{c}_j \cdot \big(r_i - (p_1 + q_1 - 6)/4\big),$$

    and every other party $\mathcal{P}_i$ for $i \in [2, n]$ computes

$$\boldsymbol{\zeta}_{i,j} \coloneqq \boldsymbol{\tau}_{i,j} + \boldsymbol{c}_j \cdot \big(r_i - (p_i + q_i)/4\big).$$

    They all broadcast the values they have computed to one another.
11. The parties halt and output $(\texttt{not} - \texttt{biprime}, \mathsf{sid})$ if there exists any $j \in [\lceil 1.475s \rceil]$ such that

$$\prod_{i \in [n]} \boldsymbol{\gamma}_j^{\boldsymbol{\zeta}_{i,j}} \not\equiv \prod_{i \in [n]} \boldsymbol{\alpha}_{i,j} \cdot \boldsymbol{\chi}_{i,j}^{\boldsymbol{c}_j} \pmod{N}.$$

12. Let C be a circuit computing $\mathsf{VerifyBiprime}(N, M, \boldsymbol{c}, \{\cdot, \cdot, \cdot, \boldsymbol{\zeta}_{i,*}\}_{i \in [n]})$; that is, let it be a circuit representation of Algorithm $\mathsf{VerifyBiprime}$ with the public values $N, M, \boldsymbol{c}$, and $\boldsymbol{\zeta}$ hardcoded. The parties send $(\texttt{compute}, \mathsf{sid}, \{\mathsf{GenSID}(\mathsf{sid}, i)\}_{i \in [n]}, C)$ to $\mathcal{F}_{\mathrm{ComCompute}}(n)$, and in response they all receive $(\texttt{result}, \mathsf{sid}, z)$. If $z = \perp$, or if $\mathcal{F}_{\mathrm{ComCompute}}(n)$ aborts, then the parties halt and output $(\texttt{not} - \texttt{biprime}, \mathsf{sid})$.
13. The parties halt and output $(\texttt{biprime}, \mathsf{sid})$ to the environment if $\gcd(z, N) = 1$, or halt and output $(\texttt{not} - \texttt{biprime}, \mathsf{sid})$ otherwise.

Algorithm 4 $\mathsf{VerifyBiprime}(N, M, \boldsymbol{c}, \{(p_i, q_i, \boldsymbol{\tau}_{i,*}, \boldsymbol{\zeta}_{i,*})\}_{i \in [n]})$

---

1. Sample $r \leftarrow \mathbb{Z}_N$ and compute

$$z = r \cdot \left(-1 + \sum_{i \in [n]} (p_i + q_i) \right) \pmod{N}$$

2. Return z if and only if it holds that

$$\begin{aligned} N &= \sum_{i \in [n]} p_i \cdot \sum_{i \in [n]} q_i \\ &\wedge \sum_{i \in [n]} p_i \equiv \sum_{i \in [n]} q_i \equiv 3 \pmod{4} \\ &\wedge 0 \leq p_i < M \wedge 0 \leq q_i < M \text{ for all } i \in [n] \\ &\wedge \boldsymbol{\tau}_{1,j} = \boldsymbol{\zeta}_{1,j} - \boldsymbol{c}_j \cdot (r_1 - (p_1 + q_1 - 6))/4 \text{ for all } j \in [\lceil 1.475s \rceil] \\ &\wedge \boldsymbol{\tau}_{i,j} = \boldsymbol{\zeta}_{i,j} - \boldsymbol{c}_j \cdot (r_i - (p_i + q_i))/4 \text{ for all } i \in [2, n] \text{ and } j \in [\lceil 1.475s \rceil] \end{aligned}$$

If any part of the above predicate does not hold, output $\perp$.

7.14 AND Gate Cost Analysis of the π_{VerifyBI} Circuit

This section provides a detailed, quantitative analysis of the AND gate cost for the π_{VerifyBI} circuit (Algorithm 3), complementing the high-level discussion in the main body. Following the methodology of Chen et al. [15, Section 6.2], we measure complexity in the number of AND gates, using the cost functions for fundamental arithmetic operations outlined in [15, Table 6.7](e.g., integer multiplication, intmul(L), and modular multiplication, modmul(L)).

A gate-level analysis of the circuit implementing Algorithm 3 is derived by summing the costs of each verification step. The cost can be conceptually broken down into: (1) the cost of the final GCD test check, which is dominated by a single large modular multiplication; (2) the cost of verifying the modulus composition $N = (\sum p_i)(\sum q_i)$, which requires a large integer multiplication; and (3) the primary cost driver, the $\kappa_{\min}$ consistency checks $\tau_{i,j} = \zeta_{i,j} + c_j \cdot (p_i + q_i)/4$, whose complexity scales with the number of parties n, the number of iterations $\kappa_{\min}$, and the bit-lengths of the shares and blinding factors. A detailed, gate-level analysis aggregates these components into an approximate total circuit size, where ℓ denotes the bit-length of the modulus N:

$$\begin{aligned} \text{AND Gates} \approx\ & \text{modmul}(\ell) + \text{intmul}(\ell/2) + (4\kappa_{\min} \cdot n + \kappa_{\min}) \cdot 2\ell \\ & + \kappa_{\min} \cdot n(4\kappa - 7 + 7\log_2 n + \log_2 \kappa_{\min}) \\ & + 4n + \ell + 3 \end{aligned}$$

The terms in this formula correspond directly to the cryptographic operations within the algorithm. The table below provides a breakdown of the origins of the dominant terms.

Table 8. Breakdown of the AND Gate Cost Formula for the π_{VerifyBI} Circuit

Formula Term	**Source Operation**	**Explanation**
modmul(ℓ)	$z = r \cdot \left(-1 + \sum(p_i + q_i)\right) \pmod N$	Upper-bound cost for the asymmetric modular multiplication to compute z.
intmul($\ell/2$)	$N = \left(\sum p_i\right)\left(\sum q_i\right)$	Cost of the integer multiplication to verify the composition of N.
$O(\kappa_{\min} \cdot n \cdot \ell)$	$\tau_{i,j} = \zeta_{i,j} + c_j \cdot (p_i + q_i)/4$	Cost from arithmetic on share-length values, aggregated over all parties and iterations.
$O(\kappa_{\min}^2 \cdot n)$	$\tau_{i,j} = \zeta_{i,j} + c_j \cdot (p_i + q_i)/4$	Quadratic cost from operating on blinding factors (τ, ζ) whose bit-length is proportional to $\kappa_{\min}$.

This cost function is structurally similar to the one derived for the original protocol in Chen et al. [15], with slight modifications. The differences arise from two main sources. First, the number of iterations in our protocol is $\kappa_{\min}$ (which for a practical setting like $p_{\min} = 233$ is $\lceil 1.489\kappa \rceil$), whereas the original analysis uses a fixed factor of $2.5s$, where s corresponds to our security parameter κ. Second, our protocol incorporates several minor adjustments to the zero-knowledge checks (cf. Section 7.13), which slightly alter the lower-order terms. Despite these differences, the dominant, high-complexity terms—a large modular multiplication (modmul(ℓ)) and an integer multiplication (intmul($\ell/2$))—remain structurally the same. Therefore, the overall computational complexity is comparable to the original work, confirming the assertion in the main body that the costs are substantial.