A Generalized Distributed RSA Key Generation

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

[1]AMIS
{chihyun, glen}@maicoin.com
[2]Division of Biostatistics, NYU Grossman School of Medicine
Ting-Fang.Lee@nyulangone.org

Abstract

In this paper, we propose a novel bi-primality test to determine whether $N = pq$ is the product of two primes on any RSA modulus in which we relaxed the restriction, $p \equiv q \equiv 3 \pmod 4$, that was assumed in most of current bi-primality tests. Our bi-primality test is generalized from Lucas primality test to the bi-prime case. Our test always accepts when p and q are both prime, and otherwise accepts with probability at most $1/2$. In addition, we also prove that the Boneh-Franklin's bi-primality test accepts composite with probability at most $1/4$ instead of $1/2$, if we add an additional condition $\gcd(N, p + q - 1) = 1$. Moreover, we design a multiparty protocol against of static semi-honest adversaries in the hybrid model and provide a security proof. We then implement the proposed protocol and run in a single thread on a laptop which turned out with average 224 seconds execution time, given that N is around 2048-bit.

1 Introduction

The RSA cryptosystem [40] is one of the original and commonly used public-key cryptosystems. In most applications, two large distinct primes p and q, are initially generated as secrets and the public-key, $N = pq$, is the product of the two different primes. However, it may lead to single point of attack. This problem can be resolved by multi-party computation (abbrev. MPC) which allows participants jointly computing a function using all parties' inputs while each party can still keep their own input private. In this scenario, the MPC technology enables parties to generate public key N without knowing the two primes p and q. Many cryptographic protocols and primitives such as threshold homomorphics encryption [24,28], time-lock puzzle [1,33,41], accumulators [5,8,31], and VDFs [7,15,20,29,37,42] need such feature. In particular, when n parties are given, of which any $t < n$ can be corrupted by an adversary, and we want a secure protocol that outputs a random and valid RSA modulus $N = pq$ where p, q are primes of a given size, while the adversary learns nothing but N from the protocol. It is often a challenge to keep p and q private while generating such modulus.

Two primality tests were widely used to tackle this problem. One is the modified Miller-Rabin primality test [2, 17] that uses generic MPC methods to generate the prime factors by testing random candidate numbers individually for primality. It firstly takes two integers, $p \equiv 3 \pmod 4$ and N such that $p|N$. We assume $p \in [2^{\kappa_1 - 1}, 2^{\kappa_1}]$ and $N \in [2^{2\kappa_1 - 2}, 2^{2\kappa_1}]$. To examine if p is a prime, the test follows the steps:

1. Randomly choose v in $\mathbb{Z}_N$.
2. Compute $\gamma_p = v^{(p-1)/2} \pmod N$.
3. If $\gamma_p \equiv \pm 1 \mod p$, then p is a *probably prime*, else p is a composite number.

However, using MPC technology to efficiently compute $\gamma_p \pmod p$ while keeps p private remaining a challenge. Boneh and Franklin [9] proposed the generalized Miller-Rabin test to avoid computational modular with secret moduli. It takes two integers $p \equiv q \equiv 3 \pmod 4$ with $N = pq$. The test proceeds as the following:

1. randomly choose $g \in \mathbb{Z}_N$ with $\left[\frac{g}{N}\right] = 1$.

2. Compute $g^{(p-1)(q-1)/4} = \gamma_N \pmod N$.

3. If $\gamma_N \equiv \pm 1 \pmod N$, then N is a *Probably bi-prime*, else N is not.

4. $\gcd(N, p+q-1) = 1$[1]

The disadvantage of this test is that the two inputs, p and q, both need to be primes to pass the test while the modified Miller-Rabin test can collect qualified primes one at a time.

It is noteworthy that both Miller-Rabin and Boheh-Franklin's tests can only support the case that $p \equiv q \equiv 3 \pmod 4$ which allows $(p-1)/2$ and $(p-1)(q-1)/4$ to be both odd integers to further simplify the computation. The Miller-Rabin test, in the worst case, may accept a composite with probability $1/4$ [11,39] while it is known that the average case behavior has an upper bound [16,17]. The Boneh-Fanklin's test always accepts when both p and q are prime, and otherwise accepts with probability at most $1/2$; while there is no known average error [9].

There are two parts of such protocols: a) Prime Candidate Sieving: participants generate potential biprime N that does not divide by a prime less than a pre-determined integer B; and b) Biprimality testing: the candidate N is repeatedly tested by a biprimality test. If N is not a biprime, then start over the process.

1.1 Our contribution

Our paper will be focusing on studying the biprimality testing. We illustrate that, in the worst scenario, the Boneh and Fanklin's [9] biprimality test accepts a composite with probability $1/4$ instead of $1/2$ when assuming $\gcd(N, p+q-1) = 1$. More precisely, we have:

Theorem 1 (Revised Boneh-Franklin)**.** *Let $p \equiv q \equiv 3 \pmod 4$, $\gcd\big(pq, (p-1)(q-1)\big) = 1$, and $e := (p-1)(q-1)/4$. Assume that $N := pq$. Consider a subgroup of $\mathbb{Z}_N^\times$,*

$$G(N) := \left\{ g \in \mathbb{Z}_N^\times \;\middle|\; \left[\frac{g}{N}\right] = 1 \right\},$$

which has a subgroup

$$\mathrm{BF}(N, e) := \left\{ g \in \mathbb{Z}_N^\times \mid g^e \equiv \pm 1 \pmod N \right\}.$$

If p, q are both primes, then we have $|\mathrm{BF}(N,e)| = |G(N)|$. For the other cases, we have $|\mathrm{BF}(N,e)| \leq |G(N)|/4$.

This theorem bypasses the exceptional case in [9, Lemma 2] and improves the efficiency in planning the computation since the statistical error is down to 4^{-s} after executing the test s times.

As far as we know, the current distributed RSA protocols can only generate the primes p, q with $p \equiv q \equiv 3 \pmod 4$. We relax the restriction to arbitrary odd primes in the distributed RSA protocols through a generalized Lucas biprimality test (cf. Theorem 4), and prove that the test accepts with probability at most $1/2$ when p and q are not both primes. In addition, we define a natural functionality for generating RSA key, and design protocols for distributed generation of RSA moduli that are secure against static semi-honest corruption. The protocol is statistically secure when assuming access to a functionality for secure multiplication.

We first compare with Boneh-Franklin's test. Theoretically, their efficiency is slightly better than our test. We next explain that our protocol is expexted polynomial time. Lastly, implemented our semi-honest protocol using our biprimality test, for 3 and 4 participants, and the bit length of N 2048, and so on. Details on the implementation and results can be found in Section 5. The results clearly show that our protocol is practical assuming honest majority.

[1] This test can be replaced of by the test [9, Section 4.1] on the group $\big(\mathbb{Z}_N[x]/(x^2+1)\big)^\times/\mathbb{Z}_N^\times$. However, its cost is higher [23, Figure 3.4].

Table 1: Comparison of the biprimality tests

	Probability of false positive under the worst case	p, q restriction	p, q can be generated separately
Our Test	1/2	Any primes	No
Boneh-Fanklin	1/2	$p \equiv q \equiv 3 \pmod 4$	No
Boneh-Fanklin*	1/4	$p \equiv q \equiv 3 \pmod 4$	No
Miller-Rabin	1/4	$p \equiv q \equiv 3 \pmod 4$	Yes

Boneh-Fanklin*: the Revised Boneh-Fanklin.

1.2 Technical Overview

We follow the blueprint of the protocols in [9], and [13] but replace with our primality testing. Given integers D and $N = pq$, we extend the original Lucas primality test to the biprime case, so we study the cardinality of the set

$$\mathrm{LPBP}(D, N, e) := \left\{ (P, Q) \;\middle|\; \begin{array}{l} 0 \le P, Q < N, P^2 - 4Q = D \pmod N, \\ \gcd(Q, N) = 1, N \text{ is } \mathrm{lpbp}(P, Q) \end{array} \right\}.$$

Here N is $\mathrm{lpbp}(P, Q)$, Lucas pseudo-bi-prime (cf. Definition 2), means that N is not product of two distinct primes, but can pass the test. The size of this set can determine the difficulty of verifying if N is a bi-prime. We prove that when N is not square-free, then $|\mathrm{LPBP}(D, N, e)| \le N/3$ for all D. If N is square-free and a product of at least three distinct primes, then $|\mathrm{LPBP}(D, N, e)|$ is approximately N for some "unlucky" D, which indicates that for arbitrary pairs P, Q can pass the test with overwhelming probability. To resolve this issue, we expand this set to a product space (D, P, Q) with D satisfying $\left[\frac{-D}{p}\right] = \left[\frac{-D}{q}\right] = -1$ where $\left[\frac{\cdot}{\cdot}\right]$ is the Jacobi symbol. Naturally, we would consider $D \in \mathbb{Z}_N^\times$. However, in the security proof, the simulator might require D to be factorized for processing simulation to avoid the bias of the view of ideal/real world. It is challenging to factorize D when it is randomly selected large integer. Therefore, the above discussion suggests us that D should be a prime. By Dirichlet's theorem on arithmetic progressions (cf. Theorem 2), we can construct an interval that contains a good portion of such D. The probability of randomly sampling qualified D is approximately 1/4. Validating if the sampled D is qualified during the sampling process may leak some information of N. The leakage can be proved that it is negligible by assuming the hardness of factoring.

1.3 Related work

Boneh and Franklin [9] first proposed the distributed RSA modulus generation. They provided an efficient distributed biprime test protocol which can test if $N = pq$ is a biprime without needing to know information about p and q and is secure in semi-honest model against an honest majority. Frankel, MacKenzie and Yung [22] improved Boneh and Franklin's protocol to achieve malicious security in honest majority. Poupard and Stern [38] then proposed two party malicious secure protocol using OT; however, their protocol can leak some bits of honest users' information.

Gilboa [26] revised Boneh and Franklin's protocol and presented a two party semi-honest protocol. They also proposed three secure multiplications using OT, homomorphic encryption, and oblivious polynomial evaluation. Hazay et al. [28] then added zero knowledge into each step of Gilboa's secure multiplication that is based on homomorphic encryption to achieve two party malicious security. This was the first dishonest majority malicious secure protocol without leakage. Frederiksen et al. [23] implement two party protocol secure in malicious model using Gilboa's secure multiplication protocol based on OT which they check at the last step instead of adding zero knowledge proof to each step. This significantly enhanced the efficiency while allowing slightly leakage in honest party's input.

Malkin et al. [34] introduced sieving which randomly generate prime candidates and validate those candidates using trial division. Chen et al. [13] followed Malkin et al. [34] to incorporate Chinese Reminder Theorem to enhance the efficiency in generating prime candidates. They then introduce secure multiplication

based on homomorphic encryption and semi-honest aggregator such that the protocol is scaleable [14]. The approach proposed by Guilhem et al. [18] also employs the Chinese Remainder Theorem for sieving. What distinguishes it from [13] is that they first generate multiplicative sharings and then transform them into additive sharings through semi-honest multiplication, thereby reducing communication costs.

On the other hand, Algesheimer et al. [2] proposed a distributed primality test based on Miller–Rabin primality test that achieves semi-honest security against a dishonest majority. The advantage of such distributed Miller-Rabin primality test is that it can test if p and q are primes separately instead of needing p and q pass the test simultaneously like in Boneh-Franklin's test [9]. This can largely reduce the iterations, but the cost of each iteration is higher than Boneh-Franklin's test. Damgård and Mikkelsen [17] utilized the replicated secret sharing in the case of three parties to improved efficiency, secure in malicious model against an honest majority in comparison with Algesheimer et al. [2]. Burkhardt at el. [12] proposed a protocol which followed Damgård and Mikkelsen's idea and used a different test along with shamir secret sharing. Their protocol achieved efficient distributed RSA key generation, with more than 3 parties and without set-up assumptions.

2 Preliminaries

Basic notations. Let $\mathbf{P}$ be the set of all primes, $\mathbb{N}$ be the nature numbers, and $\mathbb{Z}$ be the ring of integers. For a finite set S, $|S|$ means the cardinality of S. Let $\mathbb{Z}_N$ be the additive group of order order N, and $\mathbb{Z}_N^\times$ be the multiplicative group in $\mathbb{Z}_N$. Moreover, $|\mathbb{Z}_N^\times| = \phi(N)$, where ϕ is the Euler's totient function. For an interval $\mathcal{I}$, we set $\mathbf{P}(\mathcal{I}) := \{p \in \mathbf{P} \mid p \in \mathcal{I}\}$. The greatest common divisor of two positive integers x and $y \in \mathbb{N}$ is denoted by $\gcd(x, y)$. For an integer N, we denote $\epsilon_D(N)$ to be the Jacobi symbol $\left[\frac{D}{N}\right]$. For secret sharings, we adapt the following notation and conventions: $[\alpha]_\beta$ will denote the secure additive sharing of value α in the integer domain $\mathbb{Z}_\beta$ (i.e. Each of the participants, $\{\mathcal{P}_i\}_{i=1}^k$, has their own secret $\alpha_i \in \mathbb{Z}_\beta$ such that $\sum_{i=1}^k \alpha_i \equiv \alpha \pmod{\beta}$).

2.1 Some Mathematical results

In this section, we recall some facts used here. The Chinese remainder theorem asserts that let $\mathbf{m} = (m_j)_{j=1}^\ell$ be a vector of pairwise-coprime positive integers and $\mathbf{a} = (a_j)_{j=1}^\ell$ be a vector of numbers such that $0 \le a_j < m_j$ for all $1 \le j \le \ell$. Set $M := \prod_{j=1}^\ell m_j$. Then there exists a unique y with $0 \le y < M$ such that the system $y \equiv a_j \pmod{m_j}$ holds for all $1 \le j \le \ell$.

Given such $\mathbf{m}$ and $\mathbf{a}$, we have the following algorithm to find the unique y:

Protocol 1 CRT-Reconstruct

Inputs: Vectors $\mathbf{m}$ and $\mathbf{a}$.

Output: The value y satisfying $y \equiv a_j \pmod{m_j}$ for all $1 \le j \le \ell$.

1. Compute $M = \prod_{j=1}^\ell m_j$.
2. Compute $x_j := M/m_j$ and find the inverse b_j of x_j in $\mathbb{Z}_{m_j}$. Output $y := \sum_{j=1}^\ell a_j b_j x_j \pmod{M}$.

Next, we introduce Dirichlet's theorem on arithmetic progressions assuming generalized Riemann hypothesis (abbrev. GRH). For a given Dirichlet character $\chi \pmod{q}$ and a complex number s and $\Re(s) > 1$, its L-function is defined by

$$L(s, \chi) := \sum_{n \ge 1} \chi(n) n^{-s},$$

which can be extended by analytic continuation to the complex plane.

Roughly speaking, GRH says that if $L(s,\chi)=0$ and $\Re(s)>0$, then $\Re(s)=1/2$ (i.e. all non-trivial roots are on the line $\Re(s)=1/2$). For two positive integers with $\gcd(a,q)=1$, let

$$\pi(x;q,a):=\sum_{\substack{p\in\mathbf{P}([2,x))\\ p\equiv a \pmod q}} 1.$$

The asymptotic formula of $\pi(x;q,a)$ is given below.

Theorem 2. *[35, Corollary 13.8] Suppose that* $\gcd(a,q)=1$. *Then for* $x\geq 2$,

$$\pi(x;q,a)=\frac{\mathrm{li}(x)}{\phi(q)}+O\left(x^{1/2}\ln x\right).$$

Here $\mathrm{li}(x):=\displaystyle\int_2^x \frac{du}{\ln u}$.

Therefore, one has

Corollary 1. *Let* $N=pq$ *be the product of two distinct odd integers such that neither* p *nor* q *are perfect squares, and assume GRH. Then for all* $x\geq 2$, *we have*

$$\left[\sum_{D\in\mathbf{P}([2,x]):\left[\frac{-D}{p}\right]=\left[\frac{-D}{q}\right]=-1} 1\right]\Bigg/\left[\sum_{\substack{D\in\mathbf{P}([2,x]):\\ (D,N)=1}} 1\right]=\psi(p,q)+O\big(x^{-1/2}(\ln x)^2\big).$$

Here $\psi(p,q):=\begin{cases}1/2, \text{ if } p=p'k_1^2 \text{ and } q=p'k_2^2, \text{ for some prime } p';\\ 1/4, \text{ otherwise.}\end{cases}$

Proof. When N is given and be a product of two distinct odd integers p,q, Lemma 7 tells us

$$\big|\{a\in\mathbb{Z}_N^\times \mid \epsilon_{-a}(p)=-1 \text{ and } \epsilon_{-a}(q)=-1\}\big|=\phi(N)\cdot\psi(p,q).$$

Then

$$\begin{aligned}&\sum_{D\in\mathbf{P}([2,x]):\left[\frac{-D}{p}\right]=\left[\frac{-D}{q}\right]=-1} 1\\ =&\sum_{t\in\mathbb{Z}_N^\times:\left[\frac{-t}{p}\right]=\left[\frac{-t}{q}\right]=-1}\pi(x;N,t)\\ =&\psi(p,q)\cdot\mathrm{li}(x)+O\big(x^{1/2}\ln x\big),\end{aligned}$$

and

$$\sum_{\substack{D\in\mathbf{P}([2,x]):\\ (D,N)=1}} 1=\mathrm{li}(x)+O\big(x^{1/2}\ln x\big)$$

The proof is concluded by $\mathrm{li}(x)=\frac{x}{\ln x}+O\left(\frac{x}{(\ln x)^2}\right)$. □

We can derive that, given an interval, the size of the set of primes that satisfy any quadratic values are almost the same.

Corollary 2. *Let* $N=\prod_{i=1}^s p_i$ *be an odd square-free integer, and assume GRH. For all* $x\geq 2$, *and any* $\epsilon_i,\epsilon_i'\in\{-1,1\}$, *where* $1\leq i\leq s$, *we have*

$$\begin{aligned}&\frac{\left|\left\{p\in\mathbf{P}([2,x])\mid\left[\frac{p}{p_i}\right]=\epsilon_i \text{ for all } 1\leq i\leq s\right\}\right|}{\left|\left\{p\in\mathbf{P}([2,x])\mid\left[\frac{p}{p_i}\right]=\epsilon_i' \text{ for all } 1\leq i\leq s\right\}\right|}\\ =&1+O\big(x^{-1/2}(\ln x)^2\big).\end{aligned}$$

Proof. For any $\{\epsilon_i\}_{i=1}^s$, we have

$$\begin{aligned}
&\left|\left\{p\in\mathbf{P}([2,x])\mid \left[\frac{p}{p_i}\right]=\epsilon_i \text{ for all } 1\le i\le s\right\}\right| \\
=&\sum_{t\in\mathbb{Z}_N^\times:\left[\frac{t}{p_i}\right]=\epsilon_i,\ \forall i} \pi(x;N,t) \\
=&\left(\sum_{t\in\mathbb{Z}_N^\times:\left[\frac{t}{p_i}\right]=\epsilon_i,\ \forall i} 1\right)\left(\frac{\mathrm{li}(x)}{\phi(N)}+O(x^{1/2}\ln x)\right) \\
=&\frac{\mathrm{li}(x)}{2^s}+O(x^{1/2}\ln x).
\end{aligned}$$

□

Below is a fact from group theorem [3, Lemma 2.1].

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

2.2 Lucas pseudo-primes

We introduce Lucas sequence and some results [3]. Let P and Q be integers and $D:=P^2-4Q$. The Lucas sequence (U_k,V_k) that is associated with the parameters P,Q are defined as, for $k\ge 0$,

$$\begin{cases} U_{k+2}=PU_{k+1}-QU_k; \\ V_{k+2}=PV_{k+1}-QV_k, \end{cases}$$

with the initial conditions

$$\begin{cases} U_0=0, U_1=1; \\ V_0=2, V_1=P. \end{cases}$$

It is well known that $U_{p-\epsilon_D(p)}\equiv 0 \pmod p$ for any prime $p\nmid 2QD$. A composite integer N that is relatively prime to $2QD$ and satisfies $U_{N-\epsilon_D(N)}\equiv 0 \pmod N$ is called a **Lucas pseudo-prime** with respect to P and Q.

For the Lucas sequence [3, 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 ring of integers of a quadratic field $\mathbb{Q}(\sqrt{D})$. If $N\nmid 2QD$, we set $\tau:=\alpha\beta^{-1}$. Then we have, for $k\in\mathbb{N}$,

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

Given an element $u+v\sqrt{D}\in\mathbb{Q}(\sqrt{D})$, the norm map is given by $\mathbf{N}(u+v\sqrt{D})=u^2-v^2D\in\mathbb{Q}$. When $x\in\mathcal{O}_D$, the norm $\mathbf{N}(x)\in\mathbb{Z}$. Consider the multiplicative group of norm 1 elements denoted by $\widehat{(\mathcal{O}_D/N)}$ 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 $\mathcal{O}_D\to\mathcal{O}_D/N$.

For completion, we also recall two facts [3, Therorem 3.1 & Proposition 3.2.] about the group $\widehat{(\mathcal{O}_D/N)}$:

Proposition 1. *Let $p\nmid 2D$ be a prime number and $r\ge 1$ be an integer. The group $\widehat{(\mathcal{O}_D/p^r)}$ is cyclic of the order $p^{r-1}\left(p-\epsilon_D(p)\right)$.*

Proposition 2. *Let D be a non-square integer and $N := \prod_{i=1}^{s} p_i^{r_i}$ 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*

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

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 units in $\mathcal{O}_D/N$. Moreover, we have

$$\begin{aligned} |\mathcal{Z}(D, N, e)| &:= \left| \left\{ (P, Q) \;\middle|\; \begin{array}{l} P^2 - 4Q = D \pmod N, \\ \gcd(Q, N) = 1, 0 \le P, Q < N \end{array} \right\} \right| \\ &= \prod_{i=1}^{s} p_i^{r_i - 1}(p_i - \epsilon_D(p_i) - 1). \end{aligned}$$

2.3 The Security Model

In this paper, we are interested in static semi-honest adversities. **Static** means that the adversary is restricted to choose a set of parties to corrupt before the protocol execution starts and cannot change this set after. **Semi-honest adversaries** run the protocol honestly, but try to learn as much as possible from the message received from other parties. Here, We adapt the definition in [36, Definition 7.5.1] stated as below.

Let $f : (\{0,1\}^*)^n \to (\{0,1\}^*)^n$ be an $n-$ary functionality, where $f_i(x_1, \ldots, x_n)$ denotes the i-th element of $f(x_1, \ldots, x_n)$. For $I = \{i_1, \ldots i_t\} \subset \{1, \ldots, n\}$, we let $f_I(x_1, \ldots, x_n)$ denote the subsequence $f_{i_1}(x_1, \ldots, x_n), \ldots, f_{i_t}(x_1, \ldots, x_n)$. Let Π be an n-party protocol for computing f. The view of the $i-$th party during an execution of Π on $\boldsymbol{x} = (x_1, \ldots, x_n)$, denoted $\mathrm{VIEW}_i^{\Pi}(\boldsymbol{x})$, is $(x_i, r_i, m_{i_1}, \ldots, m_{i_\ell})$, where r_i represents the outcome of the $i-$th party's internal coin tosses, and m_{i_j} represents the $j-$th message it has received. For $I = \{i_1, \ldots, i_t\}$, we let $\mathrm{VIEW}_I^{\Pi}(\boldsymbol{x}) := (I, \mathrm{VIEW}_{i_1}^{\Pi}(\boldsymbol{x}), \ldots, \mathrm{VIEW}_{i_t}^{\Pi}(\boldsymbol{x}))$.

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

$$\begin{aligned} &\{(S(I, (x_{i_1}, \ldots, x_{i_t}), f_I(\boldsymbol{x})), f(\boldsymbol{x}))\}_{\boldsymbol{x} \in (\{0,1\}^*)^n} \\ \stackrel{\mathrm{c}}{\equiv} &\{(\ \mathrm{VIEW}_I^{\Pi}(\boldsymbol{x}),\ \mathrm{OUTPUT}^{\Pi}(\boldsymbol{x}))\}_{\boldsymbol{x} \in (\{0,1\}^*)^n}. \end{aligned}$$

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

3 Two Biprimality Testings

In this section, we will discuss two biprimality tests. Let $N = pq$ be the product of two odd integers. One is adding the condition, $\gcd\big(N, (p-1)(q-1)\big)$, to Boneh-Franklin biprimality test. The probability of the worst case scenario is reduced from $1/2$ to $1/4$. The another one is the proposed Lucas biprimality test that generates N where N is the product of arbitrary two primes. Our goal is to count the number of pseudo-bi-primes and calculate the proportion of pseudo-bi-prime in various scenarios.

3.1 Revisit Boneh-Franklin Biprimality Testing

We will illustrate another proof of Boneh-Franklin biprimality testing with an additional condition. Let p, q be two odd positive integers. Assume that $N := p \cdot q = \prod_{i=1}^{s} p_i^{r_i}$ and $e := (p-1) \cdot (q-1)/4$. We use the

following notations:

$$\begin{cases} e = 2^k d, & \text{where } 2 \nmid d; \\ p_i - 1 = 2^{k_i} d_i \text{ for all } 1 \le i \le s, & \text{where } 2 \nmid d_i. \end{cases}$$

In the case of $p \equiv q \equiv 3 \pmod 4$, we have $k = 0$. For such N, e, we consider

$$\mathrm{BF}(N, e) := \{g \in \mathbb{Z}_N^\times \mid g^e \equiv \pm 1 \pmod N\},$$

which is a subgroup of

$$G(N) := \left\{ g \in \mathbb{Z}_N^\times \;\middle|\; \left[\frac{g}{N}\right] = 1 \right\}.$$

Lemma 2. *Given the assumptions in the Theorem 1. Then we have*

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

Proof. Since $e = \dfrac{(p-1)(q-1)}{4}$ is odd, we have

$$\begin{aligned} &|\{g \in \mathbb{Z}_N^\times \mid g^e \equiv 1 \pmod N\}| \\ =&|\{g \in \mathbb{Z}_N^\times \mid g^e \equiv -1 \pmod N\}| \end{aligned}$$

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

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

According to the Chinese Remainder Theorem, we reduce the problem to count the cardinality of $\mathrm{BF}(p_i^{r_i}, e)$ which are cyclic groups for all i [30, Theorem 3, Chapter4], since N is odd. Combining this fact and Lemma 1, one has the number of e-th roots of 1 in the group $(\mathbb{Z}/p_i^{r_i}\mathbb{Z})^\times$ is

$$|\mathrm{BF}(p_i^{r_i}, e)| = \gcd(d, p_i^{r_i - 1}(p_i - 1)) = \gcd(d, d_i).$$

The above discussion implies that

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

□

Proof of Theorem 1. Note that $p \equiv q \equiv 3 \pmod 4$ implies that $k = 0$. At first, consider the case p, q are both primes. Meanwhile, we also have $e = d_1 d_2$ and $k_1 = k_2 = 1$. The proof of this case is complete by the following equality:

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

For any N, e, we have

$$\begin{aligned} |\mathrm{BF}(N, e)| &= 2 \cdot \prod_{i=1}^{s} \gcd(d, d_i) \\ &\le 2^{-k_1 - \ldots - k_s + 2} \left(\prod_{i=1}^{s} p_i^{r_i - 1} \right)^{-1} \left(\frac{\phi(N)}{2} \right). \end{aligned}$$

Secondly, considering the case $p = p_1$ and $q = p_2^{r_2}$, where $r_2 \ge 3$ because of $q \equiv 3 \pmod 4$, we have $|\mathrm{BF}(N, e)| \le \frac{\phi(N)}{18}$. The minimal value $1/4$ occurs at the case $s = 3$ and $r_i = 1$ for all $1 \le i \le 3$. For this case, because $p \equiv q \equiv 3 \pmod 4$, two elements of the set $\{p_1, p_2, p_3\}$ are 3 module 4 and one of it is 1 module 4, which gives the bound $2^{-k_1 - k_2 - k_3 + 2} \le 2^{-2}$. For all $s \ge 4$, one has $2^{-k_1 - \ldots - k_s + 2} \le 2^{-2}$, because $s_i \ge 1$ for all $1 \le i \le s$. This concludes the proof. □

3.2 A Lucas Biprimality Testing

We generalize the idea of Lucas primality testing to the bi-prime case. As Boneh-Franklin test, our test is also a distributed biprimality test. In the beginning, we state an analogous congruence condition for verifying bi-prime as below.

Theorem 3. *Let p and q be distinct two primes, and $N = pq$. Let D be an integer satisfying $\epsilon_{-D}(p) = -1$, $\epsilon_{-D}(q) = -1$, and $e := \big(p - \epsilon_D(p)\big)\big(q - \epsilon_D(q)\big)/2$. If $\gcd(N, QD) = 1$, then we have*

$$U_e \equiv 0 \pmod{N}. \tag{1}$$

Proof. According to Chinese Remainder Theorem, we only need to prove that $U_e \equiv 0 \pmod{p}$. If $\epsilon_D(p) = 1$, then the two roots, α and β, of the polynomial $x^2 - Px + Q$ both belong to $\mathbb{Z}_p^\times$. Note that the condition $\gcd(N, QD) = 1$ implies that $\alpha\beta \neq 0$ and $\alpha \neq \beta$. Therefore, $\alpha^e \equiv \big(\alpha^{p-1}\big)^{(q-\epsilon_D(q))/2} \equiv 1 \pmod{p}$. Similarly, we also have $\beta^e \equiv 1 \pmod{p}$, which implies that $p \mid U_e$. For the case $\epsilon_D(p) = -1$, we have that two roots α, β belong to the quadratic field $\mathbb{F}_{p^2}$ over the finite field $\mathbb{Z}_p$. Because the Frobinus map gives us $\alpha^p = \beta$ and $\beta^p = \alpha$, we have

$$\alpha^e = \alpha^{(p(q-\epsilon_D(q))-\epsilon_D(p)(q-\epsilon_D(q)))/2} = (\alpha\beta)^{(q-\epsilon_D(q))/2}.$$

Similarly, we also have $\beta^e = (\alpha\beta)^{(q-\epsilon_D(q))/2}$. Hence, the proof is complete by the equality $\alpha^e - \beta^e = 0$ and $\alpha - \beta \neq 0$ in $\mathbb{F}_{p^2}$. □

Notice that the condition, $\epsilon_{-D}(p) = -1$ is equivalent to

$$\epsilon_D(p) = -\epsilon_{-1}(p) = \begin{cases} 1, & \text{if } p \equiv 3 \pmod{4}; \\ -1, & \text{if } p \equiv 1 \pmod{4}. \end{cases} \tag{2}$$

It implies that the value of $\epsilon_D(p)$ is independent of D.

Definition 2. *For convenience, a composite number N with $\gcd(N, 2QD) = 1$ is not the product of two different primes, and satisfies (1), which is called a* **Lucas pseudo-bi-prime** *with respect to P and Q. For short, we write N is an $\mathrm{lpbp}(P, Q)$.*

Remark 1. *Using the same argument, we have, for any prime $p \mid N$,*

$$V_e = \alpha^e + \beta^e = \begin{cases} 2 \mod p, & \text{if } \epsilon_D(p) = 1; \\ 2 \cdot Q^{(N/p)+1} \mod p, & \text{if } \epsilon_D(p) = -1, \end{cases}$$

which implies that if $\epsilon_D(p) = \epsilon_D(q) = 1$, then $V_e \equiv 2 \pmod{N}$. For more general N, taking $Q = \pm 1$, then $V_e \equiv 2 \pmod{N}$ always holds.

Remark 2. *When $p \equiv q \equiv 3 \pmod{4}$, one has $\epsilon_D(p) = \epsilon_D(q) = 1$, which implies that D has square roots in $\mathbb{Z}_N^\times$, and $\alpha\beta^{-1} \in \mathbb{Z}_N^\times$ for such P, Q. Now, we have the congruence conditions $(\alpha\beta^{-1})^e \equiv 1 \pmod{N}$ and $\alpha^e + \beta^e \equiv 2 \pmod{N}$, which implies that $\alpha^e \equiv \beta^e \equiv 1 \pmod{N}$. Since $\alpha = \frac{P+\sqrt{D}}{2}$. Therefore, running over all $D \in \mathbb{Z}_N^\times$ and $P \in \mathbb{Z}_N$, it motivates to study the group*

$$\{g \in \mathbb{Z}_N^\times \mid g^e \equiv 1 \pmod{N}\}.$$

Since $e = (p-1)(q-1)/2$ is an even number, then it is naturally to consider its subgroup

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

The above discussion gives us a relation between Lucas pseudo-bi-prime with the Boneh-Franklin's considering group.

Next, we sketch the proof of counting the cardinality of the set $\mathrm{LPBP}(D, N, e)$, which follows the same method in [3, Section 4].

Proposition 3. *Given $N = p \cdot q := \prod_i^s p_i^{r_i}$, where p_i is odd prime for all i. Let D be an integer satisfying $\epsilon_{-D}(p) = -1$, $\epsilon_{-D}(q) = -1$, and $e = \big(p - \epsilon_D(p)\big)\big(q - \epsilon_D(q)\big)/2$. Assume that $\gcd(N, e) = 1$ and set*

$$\begin{cases} e = 2^k d, & \text{where } 2 \nmid d; \\ p_i - \epsilon_D(p_i) = 2^{k_i} d_i \text{ for all } 1 \le i \le s, & \text{where } 2 \nmid d_i. \end{cases}$$

For such integer D, the size

$$\begin{aligned} &|\mathrm{LPBP}(D, N, e)| \\ :=& \left| \left\{ (P, Q) \;\middle|\; \begin{array}{l} 0 \le P, Q < N, P^2 - 4Q = D \pmod N, \\ \gcd(Q, N) = 1, N \text{ is } \mathrm{lpbp}(P, Q) \end{array} \right\} \right| \\ =& \prod_{i=1}^{s} \left(2 \gcd(d, d_i) - 1\right). \end{aligned}$$

Sketch of Proof. According to the Chinese Remainder Theorem and Proposition 3, we reduce the problem to compute the cardinality of $\mathrm{LPBP}(D, p_i^{r_i}, e)$. Proposition 2 implies that $\mathrm{LPBP}(D, p_i^{r_i}, e)$ equals

$$\{\tau \in \widehat{\left(\mathcal{O}_D/p_i^{r_i}\right)} \mid 1 - \tau \in \left(\mathcal{O}_D/p_i^{r_i}\right)^\times, \tau^e = 1\}.$$

The number of e-th roots of 1 in the group $\widehat{\left(\mathcal{O}_D/p_i^{r_i}\right)}$ is given by

$$\gcd(e, p_i^{r_i - 1}(p_i - \epsilon_D(p_i))) = 2 \gcd(d, d_i).$$

Meanwhile, if $1 - \tau \notin \left(\mathcal{O}_D/p_i^{r_i}\right)^\times$ with $\tau \in \widehat{\left(\mathcal{O}_D/p_i^{r_i}\right)}$ then $\tau \equiv 1 \pmod{p_i^{r_i}}$. Hence, we derive

$$\mathrm{LPBP}(D, p_i^{r_i}, e) = 2 \gcd(d, d_i) - 1.$$

□

Remark 3. *Proposition 3 also gives another proof of Theorem 3 which makes an extra assumption $(N, e) = 1$. Assume $p \neq q$ are both prime. Then the definition of e gives us*

$$4d = 2^{k_1 + k_2} d_1 d_2,$$

which implies that $k_1 = k_2 = 1$ and $d = d_1 d_2$. Using Proposition 2 and similar argument in Proposition 3, one has

$$\begin{aligned} &\left| \left\{ (P, Q) \;\middle|\; \begin{array}{l} 0 \le P, Q < N, P^2 - 4Q = D \pmod N, \\ \gcd(Q, N) = 1, N \text{ satisfies (1)} \end{array} \right\} \right| \\ &= \prod_{i=1}^{s} \left(2 \gcd(d, d_i) - 1\right) = (2d_1 - 1)(2d_2 - 1) \\ &= |\mathcal{Z}(D, N, e)|, \end{aligned}$$

which implies the desired result.

Corollary 3. *The notations and assumptions are given in the Proposition 2 and Proposition 3.*

1. *If $4 \mid p_i - \epsilon_D(p_i)$. Then*
$$|\mathrm{LPBP}(D, p_i^{r_i}, e)| \le \frac{|\mathcal{Z}(D, p_i^{r_i}, e)|}{2}.$$
2. *Let p be the smallest prime such that $p^2 \mid N$. Then*
$$|\mathrm{LPBP}(D, N, e)| \le \frac{|\mathcal{Z}(D, N, e)|}{p} \le \frac{N}{p}.$$

Proof. The proof is completed by observing the p_i part of $\mathcal{Z}(D,N,e)$ and LPBP. That is

$$\begin{aligned} &p_i^{r_i-1}(p_i-\epsilon_D(p_i)-1) = p_i^{r_i-1}\big(2^{k_i}d_i-1\big) \\ \geq &2\cdot p_i^{r_i-1}\cdot(2^{k_i-1}\gcd(d,d_i)-1). \end{aligned}$$

□

When N is square-free and product of at least three primes, Proposition 3 implies that $\mathrm{LPBP}(D,N,e)=N$ is likely to happen. To tackle this challenge, according to Corollary 3, we maybe consider to change the value of $\epsilon_D(p_i)$ by selecting different values of D in order to satisfy $4 \mid p_i-\epsilon_D(p_i)$ with fixing the value $\epsilon_D(p)$ and $\epsilon_D(q)$. Based on this observation, we extend the original LPBP to

$$\begin{aligned} \mathrm{LPBP}(x,N,e) &:= \bigcup_{\substack{D\in\mathbf{P}([2,x]),\\ \epsilon_{-D}(p)=\epsilon_{-D}(q)=-1}} \mathrm{LPBP}(D,N,e) \\ &= \left\{ (D,P,Q)\in\mathbf{P}([2,x])\times\mathbb{Z}_N\times\mathbb{Z}_N^{\times} \;\middle|\; \begin{array}{l} \epsilon_{-D}(p)=-1,\epsilon_{-D}(q)=-1, \\ P^2-4Q=D \pmod N, \\ N \text{ is lpbp}(P,Q) \end{array} \right\} \end{aligned}$$

Proposition 3 implies that

$$\begin{aligned} &\big|\mathrm{LPBP}(x,N,e)\big| \\ &= \sum_{\substack{D\in\mathbf{P}([2,x]),\\ \epsilon_{-D}(p)=\epsilon_{-D}(q)=-1}} \prod_{i=1}^{s}\big(2\gcd(d,d_i)-1\big). \end{aligned} \tag{3}$$

On the other hand,

$$\begin{aligned} &|\mathcal{Z}(x,N,e)| \\ :=&\left|\left\{ (D,P,Q)\in\mathbf{P}([2,x])\times\mathbb{Z}_N\times\mathbb{Z}_N^{\times} \;\middle|\; \begin{array}{l} \epsilon_{-D}(p)=-1,\epsilon_{-D}(q)=-1, \\ P^2-4Q=D \pmod N \end{array} \right\}\right| \\ =&\sum_{\substack{D\in\mathbf{P}([2,x]),\\ \epsilon_{-D}(p)=\epsilon_{-D}(q)=-1}} |\mathcal{Z}(D,N,e)|. \end{aligned} \tag{4}$$

Now, we arrive at the Lucas biprimary test as below.

Theorem 4. *Let $N=pq$ be the product two distinct odd integers p,q, and assume GRH. Set $e=\frac{1}{2}\Big(p+\epsilon_{-1}(p)\Big)\Big(q+\epsilon_{-1}(q)\Big)$. Assume that $\gcd\big(N,e\big)=1$. Then for any $x\geq 2$, we have*

$$\frac{\big|\mathrm{LPBP}(x,N,e)\big|}{|\mathcal{Z}(x,N,e)|} \leq \max\left\{\frac{1}{2},\frac{4}{13}+O\left(\frac{(\ln x)^2}{\sqrt{x}}\right)\right\}.$$

Proof. The case N is not square-free. The second inequality of Corollary 3, and equations (3), (4) imply that the upper bound of the consider quotient is $\frac{1}{3}$.

The case $N=\prod_{i=1}^{s}p_i$ is square-free with $p=\prod_{i=1}^{k-1}p_i$ and $q=\prod_{i=k}^{s}p_i$. Let

$$\begin{aligned} &\mathcal{S}(D,\epsilon_1,\ldots,\epsilon_{k-1};\epsilon_k,\ldots,\epsilon_s) \\ :=&\Big|\Big\{D\in\mathbf{P}([2,x]):\Big[\tfrac{-D}{p_i}\Big]=\epsilon_i,\forall i\in\{1,\ldots,s\}\Big\}\Big|, \end{aligned}$$

where $\epsilon_i \in \{-1, 1\}$ for all i. Suppose $s = 3$. Without loss of generality, we assume $p = p_1$ and $q = p_2 p_3$. Since $\epsilon_{-D}(p_2 p_3) = -1$, one has

$$\begin{aligned}
&|\mathrm{LPBP}(x, N, e)| \\
=&\mathcal{S}_{(D,-1;1,-1)} \left(\prod_{i=1}^{3} \left(2\gcd(d_i, d) - 1 \right) \right) && (5) \\
+&\mathcal{S}_{(D,-1;-1,1)} \left(\prod_{i=1}^{3} \left(2\gcd(d_i, d) - 1 \right) \right), && (6)
\end{aligned}$$

and

$$\begin{aligned}
&\left| \left\{ (D,P,Q) \in \mathbf{P}([2,x]) \times \mathbb{Z}_N \times \mathbb{Z}_N^{\times} \;\middle|\; \begin{array}{l} \left[\frac{-D}{p}\right] = \left[\frac{-D}{q}\right] = -1, \\ P^2 - 4Q = D \pmod N \end{array} \right\} \right|. \\
=&\mathcal{S}_{(D,-1;1,-1)} \left(\prod_{i=1}^{3} \left(2^{k_i} d_i - 1 \right) \right) && (7) \\
+&\mathcal{S}_{(D,-1;-1,1)} \left(\prod_{i=1}^{3} \left(2^{k_i} d_i - 1 \right) \right). && (8)
\end{aligned}$$

It is easy to see that only one of $p_i - \epsilon_D(p_i)$ is divided by 4 (i.e. $k_i \geq 2$). According to Corollary 3, there are two such cases. The Case 1: $(5) \leq (7)/2$ and $(6) \leq (8)/2$ or the Case 2^2: $(5) \leq (7)$ and $(6) \leq (8)/4$. Then we have

$$\frac{(5)+(6)}{(7)+(8)} \leq \begin{cases} \frac{(5)+(6)}{2\left((5)+(6)\right)} = \frac{1}{2}, & \text{for Case 1;} \\ \frac{(5)+(6)}{(5)+4(6)} = \frac{2}{5} + O\left(\frac{(\ln x)^2}{\sqrt{x}}\right), & \text{for Case 2.} \end{cases}$$

The last estimation of the Case 2 comes from Corollary 2. For $s \geq 4$, the proof is similar. The upper bound is still $1/2$ and the second largest is $\frac{4}{13} + O\left(\frac{(\ln x)^2}{\sqrt{x}}\right)$. More details can be found in subsection A.2.

□

Remark 4. *For $x \geq q$, the bound of the error term for $\pi(x; q, a)$ with $\gcd(a, q) = 1$ is given by $\left(\frac{1}{8\pi\phi(q)} + \frac{1}{6\pi}\right)\sqrt{x}\ln x + (0.184 \ln q + 12969.946)\sqrt{x}$ [21]. Applying this bound, we can deduce that the lower bound of x such that*

$$\frac{4}{13} + O\left(\frac{(\ln x)^2}{\sqrt{x}}\right) < \frac{1}{2}.$$

Remark 5. *If we consider $D \in \mathbb{Z}_N^{\times}$, then we use the same argument as the above theorem, which gives the upper bound is exactly $1/2$.*

Remark 6. *If we relax GRH, this theorem still holds. However, the error term $O\left(\frac{(\ln x)^2}{\sqrt{x}}\right)$ should be replaced by a suitable error function, which depends on the asymptotic formula of Dirichlet's theorem on arithmetic progressions.*

4 A Protocol of Distributed Generation of RSA Moduli

4.1 The Distributed Biprime-Sampling

A common way in sampling biprime is to uniformly random select p and q from the set of primes in a given range. However, the efficiency of sampling according to this uniform biprime distribution is poor in

[2] In fact, this case does not exist, because it violates $\epsilon_{-D}(q) = -1$.

a multiparty computation context. Most previous works chose other distributions instead such as 1) Boneh and Fanklin [9] considered a generation given by using a private distributed computation the n participants compute $N = (p_1 + \ldots + p_n)(q_1 + \ldots + q_n)$ and hope N is not divisible by any prime less than some bound B. The running time is expected polynomial; and 2) Chen et al. [13] considered another modulus-sampling functionality called CRT-Sample, which runs in strict polynomial-time. Conditioning on success, the outputs of two distributions are statistically indistinguishable. Chen et al.'s method demonstrates superior computational complexity for computing N compared to the former, which involves direct computation of N followed by trial division. Instead, the latter utilizes the Chinese Remainder Theorem to obtain N, which is co-prime to small prime numbers.

To illustrate our modified biprime-sampling functionality, we first introduce the definitions in Chen et al. [13, Definition 3.3-3.5, 4.3].

Definition 3 (Primorial Number)**.** *The i^{th} primorial number is defined to be the product of the first i prime numbers.*

Definition 4 ((κ_1, n)-Near-Primorial Vector)**.** *Let ℓ be the largest number such that the ℓ^{th} primorial number is less than $2^{\kappa_1 - \log n}$, and let $\mathbf{m}$ be a vector of length ℓ such that $m_1 = 2$ and $m_2, \ldots, m_\ell$ are the odd factors of the ℓ^{th} primorial number, in ascending order. $\mathbf{m}$ is the unique (κ_1, n)-near-primorial vector.*

Definition 5 ((κ_1, n)-Compatible Parameter Set)**.** *Let ℓ' be the smallest number such that the ℓ'^{th} primorial number is larger than $2^{2\kappa_1}$, and let $\mathbf{m}$ be a vector of length ℓ' such that $m_1 = 2$ and $m_2, \ldots, m'_\ell$ are the odd factors of the ℓ'^{th} primorial number, in ascending order. $(\mathbf{m}, \ell', \ell, M)$ is the* (κ_1, n)-Compatible Parameter Set *if $\ell < \ell'$ and $(m_1, m_2, \ldots, m_\ell)$ is the (κ_1, n)-Near-primorial vector, and $M = \prod_{i=1}^{\ell} m_i$.*

Definition 6 ($\mathbf{m}$-Coprimality)**.** *Let $\mathbf{m}$ be a vector of integers. An integer x is $\mathbf{m}$-coprime if and only if it is not divisible by any m_i for all $1 \leq i \leq |m|$. Here $|m|$ is the length of the vector $\mathbf{m}$.*

Both protocols of Chen et al. and Boneh et al. only need a few modifications to apply the ideal functionality, Functionality 1. We display Chen et al.'s protocol, and use similar sampling method with theirs. Comparing to their method, our sampled p and q can be arbitrary odd numbers. In addition, our protocol may seem to provide extra information of $p, q \pmod 4$, but it is actually not since the information is known in Chen et al.'s protocol.

Functionality 1 $\mathcal{F}_{\text{SamplePrime}}(\kappa_1, n, B)$

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

Outputs: Uniformly sample integer $p_i, q_i \in [0, 2^{\kappa_1 - \log n})$ for $i \in \{1, \ldots, n\}$ such that

- $p_1 \equiv q_1 \equiv 1 \pmod 2$;
- $2 \mid p_i$ and $2 \mid q_i$, for all $2 \leq i \leq n$;
- for any $p' \in \mathbf{P}([2, B])$, $p' \nmid N$, where $p := \sum_{i=1}^{n} p_i$, $q := \sum_{i=1}^{n} q_i$, and $N = pq$.

Each party receives
$(N, [p]_N, [q]_N, \{p_i \pmod 4, q_i \pmod 4\}_{i=1}^n)$.

The probability of randomly sampling a prime from $[0, 2^{\kappa_1 - \log n})$ is low which costs a lot of computation time of our protocol. A way to improve this is that each party locally perform division on N to ensure p and q do not divide by the primes within a trial division bound B. We then execute biprime test to the rest of N. DeBruijn [10] demonstrated the correlation between the probability that p is a prime and the trial division bound B is

$$\Pr(p \in \mathbf{P} | \text{ trial division up to } B) \sim 2.57 \left(\frac{\ln B}{\kappa_1} \right).$$

Here the notation $\sim$ is asymptotically equivalent.

As Boneh-Franklin [9, Lemma 2.1] and Chen et al. [13, Lemma 3.7] results, the knolwedge of $n-1$ integer shares of the factors p and q does not give the adversary any meaningful advantage in factoring biprimes from the distribution of Functionality 1. Applying the same argument in [9, Lemma 2.1], we have an analogous result by replacing the condition $p \equiv q \equiv 3 \pmod 4$ to $p \equiv q \equiv 1 \pmod 2$.

Lemma 3. *Let $\mathbb{Z}_N^{(2)}$ be the set of RSA moduli $N = pq$ that can be output by Functionality 1 with $n < \log N$. Suppose there exists a polynomial time algorithm $\mathcal{A}$ that given a random $N \in \mathbb{Z}_N^{(2)}$ chosen from the distribution of Functionality 1 and the shares (p_i, q_i) of $n-1$ parties, factors N with probability at least $1/n^d$. Then there exists an expected polynomial time algorithm $\mathcal{B}$ that factors $1/4k^3n^d$ (resp. $1/4k^3n^d - \mathrm{negl}(\kappa_1)$) the integers in $\mathbb{Z}_N^{(2)}$.*

We now introduce a realization against semi-honest adversaries of Functionality 1. The protocol is to build a series of congruence equations $x \equiv a_k \pmod{m_k}$ where a_k satisfies $\gcd(a_k, m_k) = 1$. When $\prod_{k=1}^{\ell} m_k$ reaches a pre-determined bits, we construct x such that $\gcd(x, m_k) = 1$ for all $1 \le k \le \ell$ through Chinese Reminder Theorem. p and q can be obtained by repeating the process twice. To calculate $N = pq$, we expand the congruence linear system $m_{k'}$ such that $x \equiv p \pmod{m_{k'}}$ and $x \equiv q \pmod{m_{k'}}$ for $\ell < k' \le \ell'$ until $N < \prod_{k=1}^{\ell'} m_k$.

The following protocol is a sub-protocol in [13, Protocol 4.4] and the proof is given in Theorem 4.5.

Protocol 2 CRT-Sample (κ_1, n)

Inputs: Each party has input $\perp$.

Outputs: p_i, q_i, $\{p_i \pmod 4, q_i \pmod 4\}_{i=1}^n$, and N.

1. Let $(\mathbf{m}, \ell', \ell, M)$ be the (κ_1, n)-Compatible parameter set. For each party $\mathcal{P}_i$ samples

$$\begin{cases} p_{i,1} = q_{i,1} = 1 & \text{if } i = 1; \\ p_{i,1} = q_{i,1} = 0 & \text{if } 2 \le i \le n, \end{cases}$$

   $0 \le p_{i,j}, q_{i,j} < m_j$ for all $2 \le j \le \ell$.

2. Each party performs Functionality 7 to obtain $N_{i,j}$ such that

$$\sum_{i=1}^{n} N_{i,j} = \left(\sum_{i=1}^{n} p_{i,j}\right)\left(\sum_{i=1}^{n} q_{i,j}\right) \pmod{m_j},$$

   for all $2 \le j \le \ell$.

3. $\mathcal{P}_i$ broadcasts $N_{i,j}$ for all $2 \le j \le \ell$ and locally checks $\sum_{i=1}^n N_{i,j} \not\equiv 0 \pmod{m_j}$ for all $2 \le j \le \ell$. Let $\mathbf{J} := \{2 \le j \le \ell | \sum_{i=1}^n N_{i,j} \equiv 0 \pmod{m_j}\}$.

   - For $j' \in \mathbf{J}$: discard $p_{i,j'}$ and $q_{i,j'}$, and repeat steps 1 and 2 to reselect $p_{i,j'}$ and $q_{i,j'}$ until $\sum_{i=1}^n N_{i,j'} \not\equiv 0 \pmod{m_j}$.
   - For $j \in \{2, 3, \dots \ell\} \setminus \mathbf{J}$: keep $p_{i,j}$ and $q_{i,j}$.

4. Each party compute the lifting p_i, q_i such that $p_i \equiv p_{i,j} \pmod{m_j}$, and $q_i \equiv q_{i,j} \pmod{m_j}$ for all $1 \le j \le \ell$. Computes for $\ell + 1 \le j \le \ell'$, $p_{i,j} := p_i \pmod{m_j}$ and $q_{i,j} := q_i \pmod{m_j}$. Perform the Functionality 7 to obtain $N_{i,j}$ such that

$$\sum_{i=1}^{n} N_{i,j} := \left(\sum_{i=1}^{n} p_{i,j}\right)\left(\sum_{i=1}^{n} q_{i,j}\right) \pmod{m_j},$$

   for all $\ell + 1 \le j \le \ell'$, and broadcasts $N_{i,j}$ for $\ell + 1 \le j \le \ell'$ and $p_i \pmod 4$, $q_i \pmod 4$.

5. Each party reconstructs a new N by Protocol 1 such that $N \equiv \sum_{i=1}^{n} N_{i,j} \pmod{m_j}$ for all $1 \le j \le \ell'$.

This output N of protocol satisfies $\gcd(N, p_i) = 1$ for all $1 \le i \le \ell$. which might deviate from B in Functionality 1. If $B < p_\ell$, then N naturally meets the requirements of Functionality 1. When $B > p_\ell$, one can check all primes p', $p_\ell < p' < B$, do not divide N, since everyone receives N after executing the protocol. Therefore, the requirements of Functionality 1 can be fulfilled through the calculation of the last party.

4.2 Distributed Lucas Biprimality Testing

We will design the distributed testing protocol based on the Lucas biprimality test theorem. The biprimality testing functionality is described as follows:

Functionality 2 $\mathcal{F}_{\text{LucasBiprime}}(x, n)$

Inputs: Each party $\mathcal{P}_i$ has public numbers $N = pq$, $p \pmod 4$, $q \pmod 4$, and shares $[p]_N$ and $[q]_N$.

Outputs: Sample prime numbers $D_j \in [2, x]$ with $\epsilon_{-D_j}(N) = 1$ until $\epsilon_{-D_j}(p) = -1$. Let κ' be the number of trial attempts.

Each party $\mathcal{P}_i$ receives $\begin{cases} (1, \{D_j\}_{j=1}^{\kappa'}, \{\epsilon_{-D_j}(p)\}_{j=1}^{\kappa'}), & \text{if} \mathcal{L}(D_{\kappa'}, p, q) = 1; \\ (0, \{D_j\}_{j=1}^{\kappa'}, \{\epsilon_{-D_j}(p)\}_{j=1}^{\kappa'}, \{p_i, q_i\}_{i=1}^{n}), & \text{otherwise.} \end{cases}$

Here $\mathcal{L}(D_j, p, q) \in \{0, 1\}$ represents the result of Lucas biprimality test with inputs D_j, p, q.

This functionality is expected polynomial time, because of Dirichlet Theorem. More precisely, randomly sample D in $\mathbf{P}([2, x])$ has almost a $1/4$ chance of obtaining the desired D (cf. Corollary 1).

We design a protocol to realize Functionality 2. According to Theorem 4, we need to find D that satisfies $\epsilon_{-D}(p) = \epsilon_{-D}(q) = -1$ (cf. Functionality 3) without leaking p and q. Next, to validate $(\alpha\beta^{-1})^e \equiv 1 \pmod{N\mathcal{O}_D}$, parties decide P, Q together that satisfy $P^2 - 4Q \equiv D \pmod N$ where α and β are roots of $x^2 - Px + Q$. We introduce shuffle method to allow n participants calculate $(\alpha\beta^{-1})^e$ together while achieving $n - 1$ privacy. Lastly, the security proof can be found in Theorem 4.4.

Protocol 3 Lucas Biprimality test(x, n)

Inputs: Each party $\mathcal{P}_i$ has odd integers $[p]_N$, $[q]_N$, $p \pmod 4$, $q \pmod 4$, and N.

Outputs: $(1, \{D_j\}_{j=1}^{\kappa'}, \{\epsilon_{-D_j}(p)\}_{j=1}^{\kappa'})$ or
$(0, \{D_j\}_{j=1}^{\kappa'}, \{\epsilon_{-D_j}(p)\}_{j=1}^{\kappa'}, \{p_i, q_i\}_{i=1}^{n})$.

1. Parties agree on random primes $D_j \in [2, x]$ such that $\epsilon_{-D_j}(N) = 1$. Send $(N, [p]_N, p \pmod 4, D_j)$ to Functionality 3 to obtain $\epsilon_{-D_j}(p)$. Repeat until $\epsilon_{-D_j}(p) = -1$.

2. Let κ' be the number of trial attempts in step 1 and let $D := D_{\kappa'}$. Parties agree on a random $P \in \mathbb{Z}_N^\times$. Computes $Q := (P^2 - D)/4 \in \mathbb{Z}_N$. Verify $\gcd(N, Q) = 1$. If $\gcd(N, Q) \neq 1$ then each broadcasts p_i, q_i and outputs $(0, \{D_j\}_{j=1}^{\kappa'}, \{\epsilon_{-D_j}(p)\}_{j=1}^{\kappa'}, \{p_i, q_i\}_{i=1}^{n})$

3. Each party computes $\epsilon_D(p) = -\epsilon_{-1}(p)$ and $\epsilon_D(q) = -\epsilon_{-1}(q)$. Set $y_1 := (\alpha\beta^{-1})^{\frac{N - p_1\epsilon_D(q) - q_1\epsilon_D(p) + \epsilon_D(N)}{2}} \in (\mathcal{O}_D/N)^\times$ and $y_i := (\alpha\beta^{-1})^{\frac{-p_i\epsilon_D(q) - q_i\epsilon_D(p)}{2}} \in (\mathcal{O}_D/N)^\times$ for all $2 \le i \le n$, where α and β are two roots of the polynomial $x^2 - Px + Q$. Each party sends y_i to Functionality 5 to obtain u. They then check

$$u \equiv 1 \pmod{N\mathcal{O}_D}.$$

If the check fails then broadcasts p_i, q_i and return $(0, \{D_j\}_{j=1}^{\kappa'}, \{\epsilon_{-D_j}(p)\}_{j=1}^{\kappa'}, \{p_i, q_i\}_{i=1}^{n})$. Otherwise, return $(1, \{D_j\}_{j=1}^{\kappa'}, \{\epsilon_{-D_j}(p)\}_{j=1}^{\kappa'})$.

The definition of the Legendre symbol functionality in the above protocol is given as below.

Functionality 3 $\mathcal{F}_{\mathrm{Leg}}(x, n)$

Inputs: Each party $\mathcal{P}_i$ has, shares p_i, $p \pmod 4$, and a prime $D \in [2, x]$ with $\gcd(D, p) = 1$, where $p := \sum_{i=1}^{n} p_i$.

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

Let D be a prime number. In order to realization above functionality for computing quadratic symbol $\epsilon_p(D)$ with hiding p, we adapt the similar strategy appearing in [27, Figure 7]. Specifically, all parties agree on a random $s \in \mathbb{Z}_D^{\times}$, and then compute the values $s^2 p$ together. Finally, all parties can compute $\epsilon_{s^2 p}(D)$ by themselves. The security proof will given in Theorem 7.

Protocol 4 Legendre symbol $\pi_{\mathrm{Leg}}(x, n)$

Inputs: Each party $\mathcal{P}_i$ has p_i, $p \pmod 4$, and a prime number $D \in [2, x]$ with $\gcd(D, p) = 1$, where $p := \sum_{i=1}^{n} p_i$.

Outputs: $\epsilon_{-D}(p)$.

1. Each party randomly sample $s_i \in \mathbb{Z}_D$ sends (s_i, s_i, D) to functionality 7 to obtain $[s^2]_D$.
2. Each party sends $([s^2]_D, p_i \pmod D, D)$ to functionality 7 to obtain $[s^2 p]_D$.
3. Each party open $[s^2 p]_D$. If $\gcd(s^2 p, D) \neq 1$, then restart to the step 1. Otherwise, outputs
$$\begin{cases} -\epsilon_{s^2 p}(D), & \text{if } p \equiv 3 \pmod 4 \text{ and } D \equiv 1 \pmod 4; \\ \epsilon_{s^2 p}(D), & \text{otherwise.} \end{cases}$$

4.3 Distributed generation of RSA moduli

In this section, we introduce the major work in this study, the functionality of RSA moduli and its realization.

Functionality 4 $\mathcal{F}_{\mathrm{RSAGen}}(x, \kappa_1, \kappa_2, \kappa_3, n)$

Inputs: Each party $\mathcal{P}_i$ input $\perp$.

Outputs: Each party $\mathcal{P}_i$ uniformly samples $p_i, q_i \in [0, 2^{\kappa_1 - \log n})$ such that

- $p_1 \equiv q_1 \equiv 1 \pmod 2$;
- $p_i \equiv q_i \equiv 0 \pmod 2$, for all $2 \leq i \leq n$.

Set $p := \sum_{i=1}^{n} p_i$, $q := \sum_{i=1}^{n} q_i$, $N = pq$ and $e = (p + \epsilon_{-1}(p))(q + \epsilon_{-1}(q))/2$ (ref. (2)). Sample prime numbers $D_j \in [2, x]$ such that $\epsilon_{-D_j}(N) = 1$ until there are κ_3 values of D_j for which $\epsilon_{-D_j}(p) = -1$. Let κ'' be the total trial attempts. If $\kappa'' > \kappa_2$, then output "overleak" and halt. Else if p, q are both primes, each party $\mathcal{P}_i$ receives $\big(1, \{D_j\}_{j=1}^{\kappa''}, \{\epsilon_{-D_j}(p)\}_{j=1}^{\kappa''}, p_i, q_i, \{p_i \pmod 4, q_i \pmod 4\}_{i=1}^{n}, N\big)$. Otherwise, $\big(0, \{p_i, q_i\}_{i=1}^{n}\big)$.

In this functionality, we need to find D such that $\epsilon_{-D}(p) = -1$. Lemma 6 suggests that there are very rare p such that it is a square number, which implies that sampling such D is efficient. Next, a collection of the values for quadratic symbols $\big\{\epsilon_{-D_j}(p) = \epsilon_j\big\}_{j=1}^{\kappa''}$ with the restriction $\epsilon_{-D_j}(N) = 1$ for all $1 \leq j \leq \kappa''$ will be calculated for all public information D_j. When κ'' is less than a fixed number κ_2, we believe that this information reveal knowledge about p or q, which is negligible. Note that given the information of $p \pmod 4$

and D_j, one can efficiently compute $\epsilon_p(D_j)$ form $\epsilon_{-D_j}(p)$. Naively, for these $\{D_j\}_{j=1}^{\kappa''}$ and an interval $[2,x]$, one has

$$\begin{aligned}&\left|\{p \in \mathbf{P} \mid p \in [2,x], \epsilon_p(D_j) = \epsilon_j \text{ for all } 1 \le j \le \kappa'\}\right| \\ =&\frac{x}{2^{\kappa''}\ln x} + O\left(\sqrt{x}\ln x\right) \ge \frac{x}{2^{\kappa_2}\ln x} + O\left(\sqrt{x}\ln x\right).\end{aligned}$$

It implies that if x is large enough and κ_2 is fixed, then the leakage seems to be controllable. In fact, one has

Lemma 4. *Let $N = pq$ be the product of two different primes and κ_2 be a positive integer. Suppose there exists a polynomial time algorithm $\mathcal{A}$ that given 1) randomly sampling D_j are prime with $\epsilon_{-D_j}(N) = 1$ for all $1 \le j \le \kappa_2$; and 2) $\epsilon_p(D_j) = \epsilon_j$ for all $1 \le j \le \kappa_2$, factors N. Then there exists a polynomial time algorithm $\mathcal{B}$ that factors N.*

Proof. Given any $\{\epsilon_j\}_{j=1}^{\kappa_2}$ with $\{\epsilon_p(D_j) = \epsilon_j\}_{j=1}^{\kappa_2}$, we can use the algorithm $\mathcal{A}$ to factor N, which is repeated at most 2^{κ_2} due to a total of 2^{κ_2} possibilities for $\{\epsilon_p(D_j) = \epsilon_j\}_{j=1}^{\kappa_2}$ with arbitrary $\epsilon_j \in \{-1,1\}$. □

Our protocol concept involves initially executing a sieve to obtain candidates for $[p]_N$, $[q]_N$, and $N = pq$ in a way that ensures N is not divisible by any prime less than or equal to m_ℓ. Next, in order to perform the Lucas biprimality test κ_3 times, we first check whether $\gcd(N, \epsilon_{-1}(p)q + \epsilon_{-1}(q)p - \epsilon_{-1}(N)) = 1$. If the checking succeeds κ_3 times, then the generation is successful. Otherwise, each party discloses their initially chosen p_i and q_i. During the execution of the κ_3 Lucas biprimality tests, we also check that the total count of computed $\epsilon_{-D_j}(p)$ does not exceed the threshold of κ_2. If it does, each party reveals p_i and q_i.

Protocol 5 Distributed Biprime Sampling $\pi_{\mathrm{RSAGen}}(x, \kappa_1, \kappa_2, \kappa_3, n)$

Inputs: Each party inputs $\perp$.

Outputs:

1. Let $\mathbf{m}$ be the (κ_1, n)-near-primorial vector and m_ℓ be the last element of $\mathbf{m}$. Each party performs Functionality 1 with parameter (κ_1, n, m_ℓ) to obtain $[p]_N$, $[q]_N$, N, $\{p_i \pmod 4\}_{i=1}^n$, and $\{q_i \pmod 4\}_{i=1}^n$.
2. Each party sends $(N, p \pmod 4, q \pmod 4, [p]_N, [q]_N)$ to Functionality 6 to obtain b. If $b \neq 1$ then each party broadcasts p_i, q_i and outputs $(0, \{p_i, q_i\}_{i=1}^n)$.
3. Each party sequentially performs Functionality 2 κ_3 times with parameter(x, n):
   - Once the output is $(0, \{p_i, q_i\}_{i=1}^n)$. Output $(0, \{p_i, q_i\}_{i=1}^n)$ and halt.
   - Let κ'' be the current number of trial attempts. Once $\kappa'' > \kappa_2$, then output "overleak" and halt.

   Let $(\{D_j\}_{j=1}^{\kappa''}, \{\epsilon_{-D_j}(p)\}_{j=1}^{\kappa''})$ be the collection of outputs obtained from Functionality 2. Each party outputs $(1, \{D_j\}_{j=1}^{\kappa''}, \{\epsilon_{-D_j}(p)\}_{j=1}^{\kappa''}, [p]_N, [q]_N, \{p_i \pmod 4, q_i \pmod 4\}_{i=1}^n, N)$.

We employ some related functionalities and protocols in the rest of this section. The functionality below is to ensure that participants can learn $\prod_i y_i$ without revealing their own y_i.

Functionality 5 $\mathcal{F}_{\mathrm{Shuffle}}(n)$

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

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

The following protocol [4] is $n-1$ privacy that realize the above functionality. 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 6 **Shuffle**(n)

Inputs: Each party $\mathcal{P}_i$ has $y_i \in (\mathcal{O}_D/N)^\times$.

Outputs: $\prod_{i=1}^n y_i \in (\mathcal{O}_D/N)^\times$.

1. Each party $\mathcal{P}_i$ randomly chooses $x_{i,j} \in (\mathcal{O}_D/N)^\times$ for all $1 \le j \le n$ such that $\prod_{j=1}^n x_{i,j} = 1$ (i.e. randomly chooses $x_{i,j}$ for $1 \le j \le n-1$ and $x_{i,n}^{-1} := \prod_{j=1}^{n-1} x_{i,j}$). Set $y_{i,1} := x_{i,1} \cdot y_i$ and $y_{i,j} := x_{i,j}$ for all $2 \le j \le n$. Send $y_{i,j}$ to the party $\mathcal{P}_j$ for all $1 \le j \ne i \le n$.
2. Each party $\mathcal{P}_i$ computes $z_i := \prod_{j=1}^n y_{j,i}$. Broadcast z_i to the other party $\mathcal{P}_j$.
3. Outputs $z := \prod_{i=1}^n z_i$.

This functionality allows participants to calculate $\gcd(N, e)$ without learning $e := (p+\epsilon_{-1}(p))(q+\epsilon_{-1}(q))$.

Functionality 6 $\mathcal{F}_{\mathrm{GCD}}(n)$

Inputs: Each party $\mathcal{P}_i$ has $N = pq$, $p \pmod 4$, $q \pmod 4$ and shares $[p]_N$ and $[q]_N$.

Outputs: Each party $\mathcal{P}_i$ receives $\gcd\big(N, (p+\epsilon_{-1}(p))(q+\epsilon_{-1}(q))\big)$.

To confirm whether $\gcd\big(N, p\epsilon_{-1}(q) + q\epsilon_{-1}(p) + \epsilon_{-1}(N)\big)$ equals 1 without revealing $[p]_N$ and $[q]_N$, each party first generates a random mask $[r]_N$ in $\mathbb{Z}_N$. They then use functionality to calculate $[t]_N := [r \cdot \big(p\epsilon_{-1}(q) + q\epsilon_{-1}(p) + \epsilon_{-1}(N)\big)]_N$. After revealing t, each participant can locally calculate $\gcd(N, t)$.

Protocol 7 **GCD test**(n)

Inputs: Each party $\mathcal{P}_i$ has $N = pq$, $p \pmod 4$, $q \pmod 4$ and shares $[p]_N$ and $[q]_N$.

Outputs: Each party $\mathcal{P}_i$ receives $\gcd\big(N, (p+\epsilon_{-1}(p))(q+\epsilon_{-1}(q))\big)$.

1. Party $\mathcal{P}_i$ sets z_i to be
$$\begin{cases} p_i\epsilon_{-1}(q) + q_i\epsilon_{-1}(p) + \epsilon_{-1}(N), & \text{if } i = 1; \\ p_i\epsilon_{-1}(q) + q_i\epsilon_{-1}(p), & \text{if } 2 \le i \le n. \end{cases}$$
Each party randomly generates $r_i \in \mathbb{Z}_N$. Send (z_i, r_i, N) to Functionality 7 to obtain $[t]_N$.
2. Each party opens $[t]_N$ and outputs $\gcd(N, t)$.

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

Functionality 7 **Modular Multiplication**(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 $z_i \in \mathbb{Z}_N$ for all $1 \le i \le n$.

4.4 Security Proofs

Theorem 5. *The Protocol 5 is securely compute functionality $\mathcal{F}_{\text{RSAGen}}$ in the $\mathcal{F}_{\text{LucasBiprime}}, \mathcal{F}_{\text{GCD}}, \mathcal{F}_{\text{SamplePrime}}$-hybrid model in the presence of static semi-honest adversary corrupt up to $n-1$ parties.*

Proof. Let $\mathcal{P}^*$ be the set of corrupt parties. We show that a simulator $\mathcal{S}$ can be constructed for simulating the transcript of Protocol 5. If $\mathcal{S}$ is given input $(\mathcal{P}^*, \perp, (0, \{p_i, q_i\}_{i=1}^n))$[3], then $\mathcal{S}$ knows all of p_i and q_i such that $\mathcal{S}$ can simply follow the protocol. Additionally, we do not consider the event that $\mathcal{S}$ is given input $(\mathcal{P}^*, \perp, \text{“overleak”})$. By Corollary 1, the probability that each D_j results in $\epsilon_{-D_j}(p) = -1$ is $\frac{1}{2}$. We can choose an appropriate κ_2 based on κ_3 to ensure that the Functionality 4 outputs “overleak” with a negligible probability. When $\mathcal{S}$ is given input

$$(\mathcal{P}^*, \perp, (1, \{D_j\}_{j=1}^{\kappa''}, \{\epsilon_{-D_j}(p)\}_{j=1}^{\kappa''}, \{p_i, q_i\}_{i\in\mathcal{P}^*}, \{p_i \pmod 4, q_i \pmod 4\}_{i=1}^n, N)).$$

The adversary $\mathcal{S}$ outputs

$$(\mathcal{P}^*, \perp, N, \{p_i, q_i\}_{i\in\mathcal{P}^*}, \{p_i \pmod 4, q_i \pmod 4\}_{i=1}^n, 1, (1, \{D_j\}_{j=1}^{\kappa''}, \{\epsilon_{-D_j}(p)\}_{j=1}^{\kappa''})).$$

The output distribution of $\mathcal{S}$ is identical with the joint distribution $\{\text{view}_{\mathcal{P}^*}^{\pi_{\text{RSAGen}}}(\perp), \mathcal{F}_{\text{RSAGen}}(\perp)\}$. □

Theorem 6. *The Protocol 3 is securely compute functionality $\mathcal{F}_{\text{LucasBiprime}}$ in the $\mathcal{F}_{\text{Shuffle}}, \mathcal{F}_{\text{Leg}}$-hybrid model in the presence of static semi-honest adversary corrupt up to $n-1$ parties.*

Proof. Let $\mathcal{P}^*$ be the set of corrupt parties. We show that a simulator $\mathcal{S}$ can be contructed for simulating the transcript of Protocol 3. If the input of $\mathcal{S}$ is

$$(\mathcal{P}^*, N, \{p_i, q_i\}_{i\in\mathcal{P}^*}, p \pmod 4, q \pmod 4, 0, \{D_j\}_{j=1}^{\kappa'}, \{\epsilon_{-D_j}(p)\}_{j=1}^{\kappa'}, \{p_i, q_i\}_{i=1}^n),$$

then $\mathcal{S}$ only need to honestly follow the protocol. Therefore, we consider the case $\mathcal{S}$ is given the input

$$(\mathcal{P}^*, N, \{p_i, q_i\}_{i\in\mathcal{P}^*}, p \pmod 4, q \pmod 4, 1, \{D_j\}_{j=1}^{\kappa'}, \{\epsilon_{-D_j}(p)\}_{j=1}^{\kappa'}).$$

1: Let $D := D_{\kappa'}$. $\mathcal{S}$ randomly samples $P \in \mathbb{Z}_N^*$ until $\gcd(Q, N) = 1$, where $Q := (P^2 - D)/4 \pmod N$.

2: The adversary $\mathcal{S}$ outputs

$$(\mathcal{P}^*, N, \{p_i, q_i\}_{i\in\mathcal{P}^*}, p \pmod 4, q \pmod 4, \{D_j\}_{j=1}^{\kappa'}, \{\epsilon_{-D_j}(p)\}_{j=1}^{\kappa'}, P, 1).$$

Since the output of $\mathcal{F}_{\text{LucasBiprime}}$ is $(1, \{D_j\}_{j=1}^{\kappa'}, \{\epsilon_{-D_j}(p)\}_{j=1}^{\kappa'})$. We have $u \equiv 1 \pmod N$, where u is the value generated by step 3 of protocol 3. Therefore, the joint distribution of the outputs generated by $\mathcal{S}$ and $\mathcal{F}_{\text{LucasBiprime}}$ and the joint distribution of the view and output of an execution Protocol 3 are identical. Additionally, we show that $\mathcal{S}$ runs in expect polynomial time. Using the Lemma 5, we have $\Pr[Q \in \mathbb{Z}_N^*] > \gamma$ for any fixed D such that $\gcd(N, 2D) = 1$, the probability is taken over the randomness of P. □

Theorem 7. *The Protocol 4 is securely compute functionality $\mathcal{F}_{\text{Leg}}$ in $\mathcal{F}_{\text{ModMul}}$-hybrid model in the presence of static semi-honest adversary corrupt up to $n-1$ parties.*

Proof. We construct the simulator $\mathcal{S}$ to simulate the transcript of π_{Leg}. Suppose $\mathcal{S}$ is given input $(\mathcal{P}^*, \{p_i\}_{i\in\mathcal{P}^*}, p \pmod 4, D, \epsilon_{-D}(p))$.

[3]In Definition 1, the second part of the simulator's input represents the input from corrupt parties, denoted as $\perp$ in $\mathcal{F}_{\text{RSAGen}}$.

1: $\mathcal{S}$ uniformly samples $s \in \mathbb{Z}_D^\times$ and $s_i \in \mathbb{Z}_D$ for $i \in \{1, \ldots, n\}$ such that $\sum_{i=1}^n s_i \equiv s \pmod{D}$.

2: $\mathcal{S}$ uniformly samples $s_i' \in \mathbb{Z}_D$ for $i \in \{1, \ldots, n\}$ such that $\sum_{i=1}^n s_i' \equiv s^2 \pmod{D}$.

3: $\mathcal{S}$ uniformly samples $r \in \mathbb{Z}_D^\times$ such that $\epsilon_r(D) = \begin{cases} -\epsilon_{-D}(p), & \text{if } p \equiv 3 \pmod 4 \text{ and } D \equiv 1 \pmod 4 \\ \epsilon_{-D}(p), & \text{otherwise.} \end{cases}$

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

5: $\mathcal{S}$ outputs

$$\begin{aligned}&(\{p_i\}_{i\in\mathcal{P}^*}, p \pmod 4, D, \{s_i\}_{i\in\mathcal{P}^*}, \{s_i'\}_{i\in\mathcal{P}^*},\\ &\qquad \{r_i\}_{i\in\mathcal{P}^*}, \{r_i\}_{i\in\{1,\ldots,n\}\setminus\mathcal{P}^*})\end{aligned}$$

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

$$\begin{aligned}&\{\mathcal{S}\big(\mathcal{P}^*, \{p_i\}_{i\in\mathcal{P}^*}, p \pmod 4, D, \epsilon_{-D}(p)\big)\}\\ \overset{c}{\equiv}&\{\text{view}_{\mathcal{P}^*}^{\pi_{\text{Leg}}}(\mathcal{P}^*, \{p_i\}_{i\in\mathcal{P}^*}, p \pmod 4, D)\}\end{aligned}$$

for any $\mathcal{P}^* \subseteq \{1, \ldots, n\}, |\mathcal{P}^*| \leq n-1, \{p_i \geq 0\}_{i=1}^n$ and $D \in \mathbf{P}([a, b])$. In the beginning, fixed any $\{p_i\}_{i=1}^n$ and D, we claim that the output of

$$\mathcal{S}\big(\mathcal{P}^*, \{p_i\}_{i\in\mathcal{P}^*}, p \pmod 4, D, \epsilon_{-D}(p)\big)$$

and the view of

$$\pi_{\text{Leg}_{\mathcal{P}^*}}(\mathcal{P}^*, \{p_i\}_{i\in\mathcal{P}^*}, p \pmod 4, D)$$

are identical. Observe that

$$\begin{aligned}\epsilon_p(D) &= \epsilon_D(p) \cdot (-1)^{\frac{p-1}{2}\frac{D-1}{2}}\\ &= \epsilon_{-D}(p) \cdot \epsilon_{-1}(p) \cdot (-1)^{\frac{p-1}{2}\frac{D-1}{2}}\\ &= \epsilon_{-D}(p) \cdot (-1)^{\frac{p-1}{2}} \cdot (-1)^{\frac{p-1}{2}\frac{D-1}{2}}\end{aligned}$$

implies that $\epsilon_p(D) = \epsilon_r(D)$. The facts D is a prime, and s is uniformly randomly chosen from $\mathbb{Z}_D^\times$ give us the identical distribution between $\{s^2 p \mid s \in \mathbb{Z}_D^\times\}$ with $\{r \mid r \in \mathbb{Z}_D^\times\}$. Due to $|\mathcal{P}^*| < n$, the s_i, s_i' in the $\text{view}_{\mathcal{P}^*}^{\pi_{\text{Leg}}}(\mathcal{P}^*, N, \{p_i\}_{i\in\mathcal{P}^*}, p \pmod 4, D)$ and $\mathcal{S}\big(\mathcal{P}^*, N, \{p_i\}_{i\in\mathcal{P}^*}, p \pmod 4, D, \epsilon_{-D}(p)\big)$ are both independently and uniformly distributed in $\mathbb{Z}_D$. We conclude that for any $\mathcal{P}^* \subseteq \{1, \ldots, n\}, |\mathcal{P}^*| \leq n-1, \{p_i\}_{i=1}^n$, and $D \in \mathbf{P}([2, x])$

$$\begin{aligned}&\{\mathcal{S}\big(\mathcal{P}^*, \{p_i\}_{i\in\mathcal{P}^*}, p \pmod 4, D, \epsilon_{-D}(p)\big)\}\\ \equiv&(\{p_i\}_{i\in\mathcal{P}^*}, p \pmod 4, D, \{s_i\}_{i\in\mathcal{P}^*}, \{s_i'\}_{i\in\mathcal{P}^*},\\ &\qquad \{r_i\}_{i\in\mathcal{P}^*}, \{r_i\}_{i\in\{1,\ldots,n\}\setminus\mathcal{P}^*})\\ \equiv&(\{p_i\}_{i\in\mathcal{P}^*}, p \pmod 4, D, \{s_i\}_{i\in\mathcal{P}^*}, \{s_i'\}_{i\in\mathcal{P}^*},\\ &\qquad \{s^2 p_i\}_{i\in\mathcal{P}^*}, \{s^2 p_i\}_{i\in\{1,\ldots,n\}\setminus\mathcal{P}^*})\\ \equiv&\{\text{view}_{\mathcal{P}^*}^{\pi_{\text{Leg}}}(\mathcal{P}^*, \{p_i\}_{i\in\mathcal{P}^*}, p \pmod 4, D)\}.\end{aligned}$$

□

5 Implementation, Benchmarks, and Evaluation

Our experiment consists of two parts:

- **Sieve:** Utilize CRT Sampling to generate two candidates p and q, and obtain N. We then verify that all primes that are smaller than B can not divide N. The MPC multiplication were using the secret-sharing that was proposed by Gennaro et al. [25, Figure 2] assuming the honest majority.

- **Lucas biprimality test:** The parameters of Lucas sequence, P, D, were both generated by a assigned party according to the protocol. Verifying the congruence condition of the Lucas biprimality test is similar to checking the condition of the quotient ring $\mathcal{O}_D/N$ in the protocol.

Lucas sequence is a recursive sequence, the general term U_k can also be obtained through matrices operation.

$$\begin{bmatrix} U_k \\ U_{k+1} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -Q & P \end{bmatrix}^k \begin{bmatrix} 0 \\ 1 \end{bmatrix} \text{ for all } k \geq 0$$

We then verify $U_e \equiv 0 \pmod{N}$ (cf. (1)). However, matrices operation is less efficient than operating in $\mathcal{O}_D$. To optimize, we use sliding window method to calculate exponential in $\mathcal{O}_D/N$, since it is faster than $m-$ary method [32] in most scenarios[4]. We also optimized trial division less than B following the steps: 1) we obtain a product of multiple primes $K := \prod_{i=1}^{\ell} p_i$ and $K \leq 2^{64}$; and 2) we next check if $p_i \mid x$ where $x := N \pmod{K}$ to detect if $p_i \mid N$. This approach aggregates calculating the congruence of ℓ integers into calculating congruence of one large integer and ℓ integers that are less than 64 bits.

Effective. Chen et al. [13] has pointed out that CRT Sampling is an polynomial-time algorithm. We will focus on explaining that the proposed Lucas Biprimairy testing is expected polynomial-time algorithm. Two major factors will impact our effectiveness. One is to find a prime D such that $\epsilon_{-D}(p) = -1$ and $\epsilon_{-D}(q) = -1$. To overcome this, we can use an integer that passed certain times of Miller-Rabin test. Theorem 2 illustrates that the probability of D meets the conditions is $1/4$ when the bit length of D is large enough. To reduce the information leakage of $\epsilon_{-D}(p)$, we can verify $\epsilon_{-D}(N) = 1$, and then verify $\epsilon_{-D}(p) = -1$ or $\epsilon_{-D}(q) = -1$, since N and D are both public. The advantage is the information leakage of testing $\epsilon_{-D}(p)$ can be reduce to half.

Assume that we need to collect k different D such that $\epsilon_{-D}(p) = -1$ and $\epsilon_{-D}(N) = 1$ to achieve the error bound of 2^{-k}. Therefore, we can estimate how many κ_2 are needed through $\sum_{i=k}^{\kappa_2} \binom{\kappa_2}{i}/2^{\kappa_2}$. For example, if $k = 80$ and $\kappa_2 = 344$, the probability of generating a qualified D while total number of sampled D is less than 80 is 2^{-80}. In practice, we can set up an upper bound for κ_2 and restart the protocol when the leaking exceeds the upper bound.

The another one is that, given D, how difficult it is to find P, Q to meet the conditions $P^2 - 4Q = D \pmod{N}$ and $\gcd(Q, N) = 1$. By Proposition 2, we know that there is great probability of obtaining P, Q that meet the conditions. Moreover, we have

Lemma 5. *Let $0 < \gamma < 1$ be a real number. If $N = \prod_{i=1}^{s} p_i^{r_i}$ is $\mathbf{m}$-coprime with $p_{k+1}^{p_{k+1}} > N^{2/(1-\gamma)}$, where $|\mathbf{m}| = k$ and p_{k+1} is the $k+1$-th prime number. Then*

$$\frac{\left|\left\{(P,Q) \;\middle|\; \begin{array}{l} P^2 - 4Q = D \pmod{N}, \\ \gcd(Q,N) = 1, 0 \leq P, Q < N \end{array}\right\}\right|}{N} > \gamma.$$

Proof. If N is $\mathbf{m}$-coprime, then the upper bound of s is $\dfrac{\ln N}{\ln p_{k+1}}$. By Proposition 2, we have, for any N,

$$\begin{aligned} &\frac{\left|\left\{(P,Q) \;\middle|\; \begin{array}{l} P^2 - 4Q = D \pmod{N}, \\ \gcd(Q,N) = 1, 0 \leq P, Q < N \end{array}\right\}\right|}{N} \\ &\geq \prod_{i=1}^{s} \frac{p_i - 2}{p_i} \geq 1 - \sum_{i=1}^{s} \frac{2}{p_i} \geq 1 - \frac{2 \ln N}{p_{k+1} \ln p_{k+1}} \geq \gamma. \end{aligned}$$

□

Fixing the bit-length of N, we can observe from Lemma 5 that when we can ensure that N is not divisible by many prime numbers, i.e., when p_{k+1} is larger, α becomes larger. From Table 2, it is evident that N produced by Protocol 2 allows for $\gamma > 0.4$.

[4]We also try to implement the algorithm in [19], but it is still slower than the sliding window method.

κ_1	$k+1$	p_{k+1}	γ
1024	132	751	≈ 0.43
1536	183	1097	≈ 0.45

Table 2: The number of γ when N is sampled from protocol 2. Recall that the size of the number N is $2\kappa_1$ bits.

Comparison. In the scenario that $N = pq$ with $p \equiv q \equiv 3 \pmod 4$, Boneh et al's biprimary testing has better efficiency. The reasons are: 1) They only need to find a g that satisfies $\epsilon_g(N) = 1$. We need to first get a prime number D such that $\epsilon_{-D}(N) = 1$ and $\epsilon_{-D}(p) = -1$, and second to randomly select P that meets $\gcd(P^2 - D, N) = 1$. 2) The validation in Boneh et al.'s test is more effective since they only need to compute one exponential of g module N. However, we validate in a 2-rank $\mathbb{Z}_N$ module. 3) In the worst scenario, Boneh et al.'s test accepts a composite with probability 1/4 and our test is 1/2. In order to achieve the same error bound, we need to double the iterations.

Our method has no restrictions to p, q. Therefore, we only benchmarked an implementation of our (semi-honest) protocol under different parameters, κ_1, κ_2, κ_3, n, and B, in a single-threaded on an Apple M2 and 16GB LPDDR5 of RAM in the 13-inch (2022) macbook pro. The results are as following:

		Time			Average
		Average	Worst	Best	no. of D
2048 bits	n=3	224s	988s	8s	161
	n=4	396s	1443s	7s	167

Average no. of D: the average number of D generated; B=9013; κ_2=200, κ_3=80; the bit length of D is 80.

Average no. of D means the average the number of κ'' in Protocol 5. As our claim, if we want to obtain 80 desired $\epsilon_{-D}(p)$ satisfying $\epsilon_{-D}(N) = 1$, we need to produce about 160.

We fix the value of κ_2 which limits the iterations of calculating $\epsilon_{-D}(p)$ where D is a prime number. To boost the efficiency, we replace the interval $[2, x]$ with $[2^{80}, 2^{81}]$. Given any $N = pq$ and p, q are non-square, the experiment results suggests that, if we select D from a small interval and fix the value of κ_2, the efficiency can be enhanced and the security can be guaranteed.

6 Future work

There are two directions of potential future work. Firstly, we provided the protocol against static semi-honest adversaries in this paper. Development of protocols against malicious adversary is needed. The other direction is to optimize the efficiency. Good bounds on the average case behavior of our test (or Boneh-Franklin test) is important to discover. For instance, for each successfully executing Miller-Rabin protocol, the probability that p is prime is high when p is random and large [16]. If such bound of our test is known, the iterations can be largely reduced.

References

[1] A. Abadi, D. Ristea, and S. J. Murdoch. Delegated time-lock puzzle. *arXiv preprint arXiv:2308.01280*, 2023.

[2] J. Algesheimer, J. Camenisch, and V. Shoup. Efficient computation modulo a shared secret with application to the generation of shared safe - prime products. In M. Yung, editor, *Advances in Cryptology — CRYPTO 2002*, pages 417 – 432. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002.

[3] F. Arnault. The rabin-monier theorem for lucas pseudoprimes. *Math. Comput.*, 66:869–881, 04 1997.

[4] J. Benaloh. Secret sharing homomorphisms: keeping shares of a secret secret. volume LNCS 263, pages 251–260, 01 1987.

[5] J. Benaloh, M. de Mare, and O.-W. Accumulators. A decentralized alternative to digital signatures. In *Advances in Cryptology-Proceedings of Eurocrypt*, volume 93.

[6] V. Bonde and J. M. Siktar. On the combinatorics of placing balls into ordered bins, 2021.

[7] D. Boneh, J. Bonneau, B. Bünz, and B. Fisch. Verifiable delay functions. In *Annual international cryptology conference*, pages 757–788. Springer, 2018.

[8] D. Boneh, B. Bünz, and B. Fisch. 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*, pages 561–586. Springer, 2019.

[9] D. Boneh and M. Franklin. Efficient generation of shared rsa keys. *Journal of the ACM*, 48, 12 2001.

[10] N. Bruijn, de. On the number of uncancelled elements in the sieve of eratosthenes. *Proceedings of the Koninklijke Nederlandse Akademie van Wetenschappen: Series A: Mathematical Sciences*, 53(5-6):803–812, 1950.

[11] J. Buhler and P. Stevenhagen. *Algorithmic number theory. Lattices, number fields, curves and cryptography. Reprint of the 2008 hardback ed.* 01 2011.

[12] J. Burkhardt, I. Damgård, T. Frederiksen, S. Ghosh, and C. Orlandi. Improved distributed rsa key generation using the miller-rabin test. *Cryptology ePrint Archive*, 2023.

[13] M. Chen, J. Doerner, Y. Kondi, E. Lee, S. Rosefield, A. Shelat, and R. Cohen. Multiparty generation of an rsa modulus. *Journal of Cryptology*, 35, 04 2022.

[14] M. Chen, C. Hazay, Y. Ishai, Y. Kashnikov, D. Micciancio, T. Riviere, A. Shelat, M. Venkitasubramaniam, and R. Wang. Diogenes: Lightweight scalable rsa modulus generation with a dishonest majority. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 590–607. IEEE, 2021.

[15] P. Chvojka. Private coin verifiable delay function. *Cryptology ePrint Archive*, 2023.

[16] I. Damgård, P. Landrock, and C. Pomerance. Average case error estimates for the strong probable prime test. *Mathematics of Computation - Math. Comput.*, 61:177–177, 09 1993.

[17] I. Damgård and G. Mikkelsen. Efficient, robust and constant-round distributed rsa key generation. pages 183–200, 02 2010.

[18] C. Delpech de Saint Guilhem, E. Makri, D. Rotaru, and T. Tanguy. The return of eratosthenes: Secure generation of rsa moduli using distributed sieving. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 594–609, 2021.

[19] C. Doche and L. Habsieger. A tree-based approach for computing double-base chains. pages 433–446, 06 2008.

[20] N. Ephraim, C. Freitag, I. Komargodski, and R. Pass. Continuous verifiable delay functions. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 125–154. Springer, 2020.

[21] A.-M. Ernvall-Hytönen and N. Palojärvi. Explicit bound for the number of primes in arithmetic progressions assuming the generalized riemann hypothesis. *Mathematics of Computation*, 91(335):1317–1365, May 2022.

[22] Y. Frankel, P. D. MacKenzie, and M. Yung. Robust efficient distributed rsa-key generation. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 663–672, 1998.

[23] T. Frederiksen, Y. Lindell, V. Osheter, and B. Pinkas. *Fast Distributed RSA Key Generation for Semi-honest and Malicious Adversaries: 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19–23, 2018, Proceedings, Part II*, pages 331–361. 07 2018.

[24] O. Friedman, A. Marmor, D. Mutzari, Y. C. Scaly, Y. Spiizer, and A. Yanai. Tiresias: Large scale, maliciously secure threshold paillier. *Cryptology ePrint Archive*, 2023.

[25] R. Gennaro and M. Rabin. Simplified vss and fast-track multiparty computations with applications to threshold cryptography. *Proc. of 17th PODC*, 06 1998.

[26] N. Gilboa. Two party rsa key generation. In *Annual International Cryptology Conference*, pages 116–129. Springer, 1999.

[27] L. Grassi, C. Rechberger, D. Rotaru, P. Scholl, and N. Smart. Mpc-friendly symmetric key primitives. pages 430–443, 10 2016.

[28] C. Hazay, G. L. Mikkelsen, T. Rabin, T. Toft, and A. A. Nicolosi. Efficient rsa key generation and threshold paillier in the two-party setting. *Journal of Cryptology*, 32:265–323, 2019.

[29] C. Hoffmann, P. Hubáček, C. Kamath, and T. Krňák. (verifiable) delay functions from lucas sequences. *Cryptology ePrint Archive*, 2023.

[30] K. Ireland and M. I. Rosen. *A classical introduction to modern number theory*, volume 84. Springer Science & Business Media, 01 1990.

[31] W. I. Khedr, H. M. Khater, and E. R. Mohamed. Cryptographic accumulator-based scheme for critical data integrity verification in cloud storage. *IEEE Access*, 7:65635–65651, 2019.

[32] C. Koc. Analysis of sliding window techniques for exponentiation. *Computers & Mathematics with Applications*, 30:17–24, 11 1995.

[33] G. Malavolta and S. A. K. Thyagarajan. Homomorphic time-lock puzzles and applications. In *Annual International Cryptology Conference*, pages 620–649. Springer, 2019.

[34] M. Malkin, T. D. Wu, and D. Boneh. Experimenting with shared generation of rsa keys. In *NDSS*, 1999.

[35] H. L. Montgomery and R. C. Vaughan. *Multiplicative Number Theory I: Classical Theory*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2006.

[36] G. Oded. *Foundations of Cryptography: Volume 2, Basic Applications*. Cambridge University Press, USA, 1st edition, 2009.

[37] K. Pietrzak. Simple verifiable delay functions. In *10th innovations in theoretical computer science conference (itcs 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[38] G. Poupard and J. Stern. Generation of shared rsa keys by two parties. In *Advances in Cryptology—ASIACRYPT'98: International Conference on the Theory and Application of Cryptology and Information Security Beijing, China, October 18–22, 1998 Proceedings*, pages 11–24. Springer, 1998.

[39] M. Rabin. Probabilistic algorithm for testing primality. *Journal of Number Theory*, 12:128–138, 02 1980.

[40] R. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 26:96–99, 01 1983.

[41] R. L. Rivest, A. Shamir, and D. A. Wagner. Time-lock puzzles and timed-release crypto. 1996.

[42] B. Wesolowski. 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*, pages 379–407. Springer, 2019.

A Appendix

A.1 Some math results

The following lemma explains that each participant randomly samples $p_i \in [0, M] \cap \mathbb{Z}$ then the probability of $\sum_{i=1}^{n} p_i$, which is square, is very low.

Lemma 6. *Let n be a fixed integer and $\{X_i\}_{i=1}^{n}$ be a collection of independent and identically distributed random variables from a discrete uniform distribution on $\{0, 1, \ldots, M\}$. Then*

$$\begin{aligned}&\sum_{k=0}^{[\sqrt{nM}]} \mathrm{P}(X_1 + \ldots + X_n = k^2)\\ <&\frac{n(\sqrt{n}+1)\binom{n}{[\frac{n}{2}]}}{(n-1)!}\frac{1}{\sqrt{M}} + O(1).\end{aligned}$$

Here $\binom{\cdot}{\cdot}$ is the binomial coefficient.

Proof. Note that the number of the set [6, Theorem 8]

$$A_k := \left\{(x_1, ..., x_n) \mid x_i \in [0, M] \text{ and } \sum_{i=1}^{n} x_i = k\right\},$$

which is a Balls-into-Bins problem, is

$$B_k := \sum_{t=1}^{n}(-1)^t \binom{n}{t}\binom{k - t(M+1) + n - 1}{n-1}.$$

Here the binomial coefficient stands for 0 if the upper index is less than the lower index, and the Stirling's formula says that $\ln n! = n \ln n + n + O(\ln n)$. Then we have

$$\begin{aligned}|B_k| <& n\binom{n}{[\frac{n}{2}]}\binom{nM + n - 1}{n-1}\\ =&\frac{n\binom{n}{[\frac{n}{2}]}}{(n-1)!}(nM + n - 1)^{n-1} + O(1).\end{aligned}$$

The number of $\sum_{i=1}^{nM} A_i = (M+1)^n$, the desired result comes from the above estimation and the following inequality

$$\sum_{k=0}^{[\sqrt{Mr}]} \mathrm{P}(X_1 + \ldots + X_n = k^2) < \frac{(\sqrt{Mn}+1)|B_k|}{(M+1)^n}.$$

□ □

Given a prime number p and an integer $x \in \mathbb{N}$, we set $\mathrm{ord}_p(x) := r$ with $p^r \mid x$ and $p^{r+1} \nmid x$.

Lemma 7. *Assume that neither r nor q are perfect square. Then*

$$\begin{aligned}&\left|\left\{D \in \mathbb{Z}_{rq}^{\times} : \left[\frac{-D}{r}\right] = -1, \left[\frac{-D}{q}\right] = -1\right\}\right|\\ =&\begin{cases}\phi(rq)/2, \text{ if } r = pk_1^2 \text{ and } q = pk_2^2, \text{ for some prime } p;\\ \frac{\phi(rq)}{4}, \text{ otherwise .}\end{cases}\end{aligned}$$

Proof. Note that for fixing $\epsilon_1, \epsilon_2 \in \{1, -1\}$,

$$\begin{aligned}&\left|\left\{D \in \mathbb{Z}^{\times}_{p^{\mathrm{ord}_p(rq)}} : \left[\frac{-D}{p^{\mathrm{ord}_p(r)}}\right] = \epsilon_1, \left[\frac{-D}{p^{\mathrm{ord}_p(q)}}\right] = \epsilon_2\right\}\right| \\ =&\begin{cases}\phi(p^{\mathrm{ord}_p(rq)}), \text{ if } 2 \mid \mathrm{ord}_p(r) \text{ and } 2 \mid \mathrm{ord}_p(q), \\ \frac{\phi(p^{\mathrm{ord}_p(rq)})}{2}, \text{ otherwise .}\end{cases}\end{aligned} \tag{9}$$

We only prove that the case $2 \nmid \mathrm{ord}_p(r)$ and $2 \nmid \mathrm{ord}_p(q)$, because the others are straightforward. For $x \in \mathbb{Z}^{\times}_{p^{\mathrm{ord}_p(rq)}}$, we can write it as $x_0 + x_1 p + x_2 p^2 + ... + x_k p^k$, where $k = \mathrm{ord}_p(r) + \mathrm{ord}_p(q) - 1$, $x_0 \in \mathbb{Z}_p^{\times}$, and $x_i \in \mathbb{Z}_p$ for all $1 \le i \le k$. Observe that x_0 is a quadratic residue of $\mathbb{Z}_p$ if and only if x belongs to the consider set (9). Therefore, in this case, we have

$$\begin{aligned}&\left|\left\{D \in \mathbb{Z}^{\times}_{p^{\mathrm{ord}_p(rq)}} : \left[\frac{-D}{p^{\mathrm{ord}_p(r)}}\right] = \epsilon_1, \left[\frac{-D}{p^{\mathrm{ord}_p(q)}}\right] = \epsilon_2\right\}\right| \\ =&\left|\left\{D \in \mathbb{Z}^{\times}_{p^{\mathrm{ord}_p(rq)}} : \left[\frac{-D}{p}\right] = \epsilon_1\right\}\right| (\text{i.e. } \epsilon_1 \text{ should equal } \epsilon_2) \\ =&\frac{\phi(p^{\mathrm{ord}_p(rq)})}{2}.\end{aligned}$$

Now, we assume that the p is the only one prime factor of r and q with $\mathrm{ord}_p(r), \mathrm{ord}_p(q)$ odd (i.e. $r = pk_1^2$ and $q = pk_2^2$). In this case, except for the prime factor of p, the other prime factors $\hat{p}$ of r and q contribute the cardinality $\phi(\hat{p}^{\mathrm{ord}_{\hat{p}}(rq)})$, which gives us the desired result $\phi(rq)/2$.

For the remainder case, there exists two primes $p' \neq p$ with $p \mid r$ and $p' \mid q$ such that $\mathrm{ord}_{p'}(q)$, and $\mathrm{ord}_p(r)$ both odd. For the other prime $\hat{p}|rq$, the quadratic value $\left[\frac{-D}{\hat{p}}\right]$ can be arbitrary, because we can choose suitable $y_p, y_{p'}$ with $-D \equiv y_p \pmod{p}$ and $-D \equiv y_{p'} \pmod{p'}$ such that $\left[\frac{-D}{r}\right] = -1$, and $\left[\frac{-D}{q}\right] = 1$. Therefore, the cardinality of the p' (resp. p) part of q (resp. r) is $\phi(p'^{\mathrm{ord}_{p'}(q)})/2$ (resp. $\phi(p^{\mathrm{ord}_p(r)})/2$), and the other prime $\hat{p}$ gives $\phi(\hat{p}^{\mathrm{ord}_{\hat{p}}(rq)})$. Combining the above discussion, the proof is complete. □

A.2 Proof of Theorem 4 for the case $s \ge 4$

We now consider the case $N = \prod_{i=1}^{s} p_i$ is square-free with $s \ge 4$, $p = \prod_{i=1}^{k-1} p_i$ and $q = \prod_{i=k}^{s} p_i$.

Let $(a_1, \ldots, a_s) \in \{1, -1\}^s$ be the sequence such that $a_i \equiv p_i \pmod{4}$ for all $1 \le i \le s$. Note that for any D' such that $\epsilon_{-D'}(p) = \epsilon_{-D'}(q) = -1$, by (2) we have

$$\epsilon_{D'}(p) = \prod_{i=1}^{k-1} \epsilon_{D'}(p_i) = \begin{cases}1, & \text{if } \prod_{i=1}^{k-1} a_i = -1; \\ -1, & \text{if } \prod_{i=1}^{k-1} a_i = 1.\end{cases}$$

The above equality implies $\prod_{i=1}^{k-1} \epsilon_{D'}(p_i) = -\prod_{i=1}^{k-1} a_i$. Similarly we have $\prod_{i=k}^{s} \epsilon_{D'}(p_i) = -\prod_{i=k}^{s} a_i$. Consider the sets

$$\mathcal{B} := \left\{(\epsilon_1, \ldots, \epsilon_s) \in \{\pm 1\}^s \middle| \prod_{i=1}^{k-1} \epsilon_i = -\prod_{i=1}^{k-1} a_i, \prod_{i=k}^{s} \epsilon_i = -\prod_{i=k}^{s} a_i\right\},$$

and $\mathcal{S}(D, \epsilon_1, \ldots, \epsilon_s)$ to be

$$\{D \in \mathbf{P}([2, x]) : \epsilon_D(p_i) = \epsilon_i, \ \forall i \in \{1, \ldots, s\}\},$$

where $\epsilon_i \in \{-1, 1\}$ for all i. One has

$$\begin{aligned} &|\mathrm{LPBP}(x, N, e)| \\ &= \sum_{\mathbf{b}\in\mathcal{B}} |\mathcal{S}(D,\mathbf{b})| \left(\prod_{i=1}^{s} \left(2\gcd(d_i, d) - 1\right) \right), \end{aligned} \tag{10}$$

and

$$\begin{aligned} &|\mathcal{Z}(x, N, e)|\,. \\ &= \sum_{\mathbf{b}\in\mathcal{B}} |\mathcal{S}\left(D, \mathbf{b}\right)| \left(\prod_{i=1}^{s} \left(2^{k_i} d_i - 1\right) \right) \\ &\geq \sum_{\mathbf{b}\in\mathcal{B}} |\mathcal{S}\left(D, \mathbf{b}\right)| \left(\prod_{i=1}^{s} \left(2^{k_i} \gcd(d_i, d) - 1\right) \right). \end{aligned} \tag{11}$$

Therefore, we need to know under what conditions k_i will be larger than 1. Note that

$$\begin{cases} k_i \geq 2, & \text{if } a_i = \epsilon_i; \\ k_i = 1, & \text{if } a_i = -\epsilon_i. \end{cases}$$

Now, we divide the proof into cases depending on the parity of $k-1$ and $s-k+1$.
Case 1: $k-1$ or $s-k+1$ is even.
Without loss of generality, we assume $k-1$ is even. We have $(-a_1, \ldots, -a_{k-1}, -a'_k, \ldots, -a'_s) \notin \mathcal{B}$ for any $(-a'_k, \ldots, -a'_s) \in \{\pm 1\}^{s-k-1}$ since

$$\prod_{i=1}^{k-1}(-a_i) = (-1)^{k-1} \prod_{i=1}^{k-1} a_i \neq -\Big(\prod_{i=1}^{k-1} a_i\Big).$$

Therefore, for any $\mathbf{b} = (\epsilon_1, \ldots, \epsilon_s) \in \mathcal{B}$, there exists $1 \leq i' \leq k-1$ such that $a_{i'} = \epsilon_{i'}$ (and hence $k_{i'} \geq 2$) and

$$\begin{aligned} &|\mathcal{S}\left(D, \mathbf{b}\right)| \left(\prod_{i=1}^{s} \left(2^{k_i} \gcd(d_i, d) - 1\right) \right) \\ &\geq 2 \cdot |\mathcal{S}\left(D, \mathbf{b}\right)| \left(\prod_{i=1}^{s} \left(2\gcd(d_i, d) - 1\right) \right). \end{aligned}$$

Therefore, we obtain

$$\frac{\left|\mathrm{LPBP}(x, N, e)\right|}{|\mathcal{Z}(x, N, e)|} \leq \frac{1}{2}.$$

Case 2: $k-1$ is odd and $s-k+1$ is odd.
Let $\mathbf{b}' := (-a_1, \ldots, -a_{k-1}, -a_k, \ldots, -a_s) \in \mathcal{B}$. For any $\mathbf{b} \in \mathcal{B} \setminus \{\mathbf{b}'\}$, there exist $1 \leq i', i'' \leq k-1$ or $k \leq i', i'' \leq s$ such that $a_{i'} = \epsilon_{i'}$ and $a_{i''} = \epsilon_{i''}$, we have

$$\begin{aligned} &|\mathcal{S}\left(D, \mathbf{b}\right)| \left(\prod_{i=1}^{s} \left(2^{k_i} \gcd(d_i, d) - 1\right) \right) \\ &\geq 4 \cdot |\mathcal{S}\left(D, \mathbf{b}\right)| \left(\prod_{i=1}^{s} \left(2\gcd(d_i, d) - 1\right) \right), \end{aligned}$$

which implies that

$$
\begin{aligned}
&|\mathcal{Z}(x,N,e)| \\
&\geq 4 \sum_{\mathbf{b}\in\mathcal{B}\setminus\{\mathbf{b}'\}} |\mathcal{S}\left(D,\mathbf{b}\right)| \left(\prod_{i=1}^{s}\left(2\gcd(d_i,d)-1\right)\right) \\
&+ |\mathcal{S}\left(D,\mathbf{b}'\right)| \left(\prod_{i=1}^{s}\left(2\gcd(d_i,d)-1\right)\right). \qquad (12)
\end{aligned}
$$

We can observe that the size of set $\mathcal{B}$ increases with the increase in s, so we only need to consider the case where $s = 4$. Taking $s = 4$ and with out loss of generality, assume $k - 1 = 1$. We have $\mathcal{B} \setminus \{\mathbf{b}'\} = \{(-a_1, -a_2, a_3, a_4), (-a_1, a_2, -a_3, a_4), (-a_1, a_2, a_3, -a_4)\}$ and $\mathbf{b}' = (-a_1, -a_2, -a_3, -a_4)$.

We conclude that:

$$
\begin{aligned}
&\frac{\left|\mathrm{LPBP}(x,N,e)\right|}{|\mathcal{Z}(x,N,e)|} \\
&\leq \frac{1+1+1+1}{4+4+4+1} + O\left(\frac{(\ln x)^2}{\sqrt{x}}\right) \\
&= \frac{4}{13} + O\left(\frac{(\ln x)^2}{\sqrt{x}}\right).
\end{aligned}
$$

The second inequality is derived from Corollary 2 and equations (11) and (12). □