Hashing to Prime in Zero-Knowledge

Thomas Groß
School of Computing, Newcastle University, United Kingdom

Keywords:
Primality Testing, Prime Hashing, RSA, Prime Encoding, Zero-Knowledge Argument

Abstract:
We establish a set of zero-knowledge arguments that allow for the hashing of a committed secret a-bit input x to a committed secret $(k+1)$-bit prime number p_x. The zero-knowledge arguments can convince a verifier that a commitment indeed is the correctly generated prime number derived from x with a soundness error probability of at most $2^{-k} + 2^{-t}$ dependent on the number of zero-knowledge argument rounds k and the number of primality bases t to establish primality. Our constructions offer a range of contributions including enabling dynamic encodings for prime-based accumulator (Barić and Pfitzmann, 1997; Camenisch and Lysyanskaya, 2002), signature (Groß, 2015) and attribute-based credential schemes (Camenisch and Groß, 2008) allowing to reduce these schemes' public key size and setup requirements considerably and rendering them extensible. While our new primality zero-knowledge arguments are of independent interest, we also show improvements on proving that a secret number is the product of two secret safe primes significantly more efficient than previously known results (Camenisch and Michels, 1999), with applications to setting up secure special RSA moduli.

1 INTRODUCTION

Hashing to a prime number is a foundational cryptographic function that enables the computation of a $(k+1)$-bit prime number p_x from an a-bit input x. It finds its applications in a wide range of higher-level constructions that rely on division-intractable encoding, such as in private information retrieval (Cachin et al., 1999), verifiable random functions (Micali et al., 1999) or verifiable computing (Ozdemir et al., 2020).

While the hashing-to-prime primitives have been found generally useful, a number of cryptographic constructions was barred from their benefits: Schemes that rely on prime or division-intractable encoding while seeking to convince verifiers in zero-knowledge of their faithful protocol execution. Such schemes include credential schemes (Camenisch and Groß, 2012), special-purpose signature schemes (Groß, 2015), as well as dynamic accumulators and related revocation systems (Barić and Pfitzmann, 1997; Camenisch and Lysyanskaya, 2002)). These constructions suffer from cumbersome join and elaborate setup protocols, which bear large public key sizes, and thereby limit their practical applications.

In this paper, we aim at creating zero-knowledge arguments that convince a verifier that a committed secret a-bit input x was deterministically hashed to a committed secret $(k+1)$-bit prime number p_x.

Clearly, this goal is feasible as all NP-languages are provable in zero-knowledge (Goldreich et al., 1991). What we are interested in is establishing efficient, flexible and practical constructions that can lift the limitations from a wide range of other constructions.

1.1 High-Level Concept

We pursue two different concepts to hash-to-prime zero-knowledge arguments. One concept operates on the principle of elimination, the other concept on the principle of recursive construction. Both constructions have in common that they establish that a hidden integer p_x is a prime derived from input x. That is, the prover convinces a verifier of a zero-knowledge predicate $p_x = \mathsf{hashToPrime}(x)$ between two commitments C_x and C_{p_x} on x and prime p_x, respectively.

Hash-to-Prime by Elimination. Upon receiving an a-bit bitstring x, the prover computes a

$(k+1)$-bit $y_i = \mathsf{PRG}_{\mathcal{H}(x)}(i)$, iterating over an index i starting from 1 until y_u is a probable prime. The prover commits to x and all $(y_i)_{i=1}^{u}$ and engages with a verifier in a zero-knowledge argument showing (i) that all computations were executed correctly, (ii) that the final committed output $p_x := y_u$ passes a primality test. (iii) that all committed intermediate candidates $(y_i)_{i=1}^{u-1}$ are composites. The number of eliminated composites u is public knowledge.

Hash-to-Prime by Recursive Construction. Upon receiving an a-bit bitstring x, the prover establishes a first prime p_0 as $2^{\ell_{n_0}} h_0 + n_0$ where $h_0 := \mathcal{H}_{Q_0,z_0}(x)$ and n_0 is number of iterations till the first prime is reached. Then, the prover continues to build a Pocklington sequence (p_{j-1}, a_j, r_j) where each r_j is given by $r_j := 2^{\ell_{n_j}} \cdot \mathcal{H}_{Q_j,z_j}(x) + n_j$ for the current Pocklington step j. In establishing each step, the prover tests in sequence for primality integers $y_{j,i} := p_{j-1} \cdot r_{j,i} + 1$ till one is found to be prime. Then, the prover commits to x, all $(y_{j,i})$, r_j, a_j. The prover makes the integers $n_{j,i}$ public. Then the prover engages with a verifier in a zero-knowledge argument yielding that the base value is prime and the preceding candidates composite, followed by that for each Pocklington step (i) the prover knows a Pocklington witness (p_{j-1}, a_j, r_j), (ii) the Pocklington criterion is fulfilled such that the subsequent number p_j is prime, (iii) the intermediate candidates $(y_{j,i})$ leading up to p_j are composites. The integer n_j of each step is public knowledge.

1.2 Our Contributions

We offer new zero-knowledge arguments for hashing to prime with both (i) elimination and (ii) recursive construction methods. This inquiry yields a range of modular, reusable zero-knowledge predicates for primality criteria, pseudo-random number generators, and square hashes to extend zero-knowledge specification languages in the Camenisch-Stadler framework (Camenisch and Stadler, 1997) and the UC framework (Camenisch et al., 2011). We gain a new efficient zero-knowledge argument that an integer is a product of two safe primes as an immediate consequence of the new predicates. We further provide a comprehensive complexity analysis that illustrates the trade-offs for different application scenarios. These contributions impact especially prime-encoded credential schemes and signature schemes by lifting the encoding to bit-strings, by enabling dynamic changes of the encoding dictionaries, and by reducing public key size and setup requirements. They enable identity-based accumulators.

2 RELATED WORK

Hashing to prime has been investigated and used as a primitive in a range of constructions. For instance, Cachin, Micali, and Stadler (Cachin et al., 1999) created a function PrimeSeq, which would produce a prime p_x from an input x with a deterministic algorithm in the context of private information retrieval. Micali, Rabin, and Vadhan (Micali et al., 1999) used an adaptation of that primitive in their first construction of a verifiable random function. Both are examples of hash-to-prime by elimination.

Ozdemir et al. (Ozdemir et al., 2020) included a construction for division-intractable encoding through a hash-to-prime primitive. They, however, chose a recursive, constructive method by establishing a sequence of primes and Pocklington witnesses. They do this in the context of SNARKs and verifiable computing. Both former approaches are not applicable as zero-knowledge arguments discrete-log based signature and zero-knowledge proof systems.

This work is also related to and borrows techniques from Camenisch and Michels' general zero-knowledge arguments on the primality of a secret integer and the composition of a special RSA modulus from two safe primes (Camenisch and Michels, 1999).

3 PRELIMINARIES

We assume a group $\mathbb{G}$ with prime order Q and two generators g and h, $\langle g \rangle = \langle h \rangle = \mathbb{G}$, for which the discrete logarithm $\log_g h$ is not known. We assume a commitment scheme in group $\mathbb{G}$ which commits messages m with commitments of the structure $C_m := g^m h^{r_m}$, where r_m is chosen uniformly at random, $r_m \in_R \mathbb{Z}_Q$.

We assume the setup for a *Square Hash* (SQH/U) (Etzel et al., 1999) family of functions from $\mathbb{Z}_Q$ to $\mathbb{Z}_Q$ as: $\{\mathcal{H}_{Q,z} : \mathbb{Z}_Q \longrightarrow \mathbb{Z}_Q | z \in \mathbb{Z}_Q\}$ and $\{\mathcal{H}_{Q,z,b} : \mathbb{Z}_Q \longrightarrow \mathbb{Z}_Q | z, b \in \mathbb{Z}_Q\}$ where the

functions $\mathcal{H}_{Q,z}$ and $\mathcal{H}_{Q,z,b}$ are defined as:

$$\begin{aligned} \text{SQH:} \quad & \mathcal{H}_{Q,z}(x) && \equiv (x+z)^2 \pmod{Q} \\ \text{SQHU:} \quad & \mathcal{H}_{Q,z,b}(x) && \equiv (x+z)^2 + b \pmod{Q} \end{aligned}$$

Theorem 1 (Square Hash (Etzel et al., 1999))**.** *The family SQH is Δ-universal. The family SQHU is strongly universal.*

We assume a setup for the Naor-Reingold PRG (Naor and Reingold, 1997). An instance is generated with a key $\langle Q, P, \mathfrak{g}, \vec{a}\rangle$ with prime Q, prime P dividing $Q-1$, $\mathfrak{g}$ a generator of order P, $\mathfrak{G} := \langle \mathfrak{g} \rangle \subset \mathbb{Z}_Q$, and $\vec{a}$ a sequence of $k+1$ elements of $\mathbb{Z}_P$. For an a-bit input x with bits $x_1, \ldots, x_n$, $\mathsf{PRG}_{a_0}(x) := f_{Q,P,\mathfrak{g},\vec{a}}$ is defined as:

$$f_{Q,P,\mathfrak{g},\vec{a}} := (\mathfrak{g}^{a_0})^{\prod_{x_i=1} a_i} \pmod{Q}.$$

3.1 Known Primality Criteria

We introduce in turn primality criteria related to (i) Lehmann's and Solovay-Strassen's tests, (ii) Miller's test, and (iii) Pocklington's test.

Theorem 2 ((Kranakis, 2013))**.** *An odd integer $n > 1$ is prime if and only if*

$$\forall a \in \mathbb{Z}_n^* : \quad a^{(n-1)/2} \equiv \pm 1 \pmod{n} \text{ and} \quad (1)$$

$$\exists a \in \mathbb{Z}_n^* : \quad a^{(n-1)/2} \equiv -1 \pmod{n}. \quad (2)$$

Theorem 3 (Miller, adapted from (Kranakis, 2013))**.** *For any odd integer $n > 1$ write $n-1 = 2^e u$, with u odd. Then, n is prime if and only if*

$$(\forall a \in \mathbb{Z}_n^*) : \quad a^u \not\equiv 1 \pmod{n} \implies \exists k < e : \left(a^{2^k u} \equiv -1 \pmod{n}\right).$$

Definition 1. *We call an odd integer $n > 1$ in Theorem 3 which fulfills both clauses with respect to base a, a strong probable prime to base a. We call a composite n fulfilling those clauses a strong pseudoprime to base a, extending naturally to the case of pseudoprimes to several bases $(a_j)_{j=1}^t$. (Pomerance et al., 1980; Jaeschke, 1993) We write:*

$$\mathsf{spsp}\left((a_j)_{j=1}^t, n\right).$$

Theorem 4 (Pocklington, in an adaptation by (Brillhart et al., 1975; Ozdemir et al., 2020))**.** *Let p be a prime, and $r < p$ and a be positive integers. Define $p' := p \cdot r + 1$. Pocklington's criterion states that if $a^{pr} \equiv 1 \pmod{p'}$ and $\gcd(a^r - 1, p') = 1$, then p' is prime. In this case, we say that (p, r, a) is a* Pocklington witness *for p'.*

3.2 Known Zero-Knowledge Proofs

We use the Camenisch-Stadler notation (Camenisch and Stadler, 1997) to express known discrete-log based proofs of representation. Therein, we use these techniques, for instance, as employed by Camenisch and Groß (Camenisch and Groß, 2008; Camenisch and Groß, 2012), to prove the following predicate.

$\underline{(\mu \not\equiv \pm 1)}$**:** Shows that given committed value is neither one nor minus one.

$$\begin{aligned} S_{\neg\pm 1} := PK\{(\alpha, \rho, \sigma, \psi, \varsigma, \varpi) : \\ D = g^\mu h^\rho \wedge \\ g = (D/g)^\sigma h^\psi \wedge g = (gD)^\varsigma h^\varpi \\ \}. \end{aligned}$$

$\underline{(\gcd(\mu, \nu) = 1)}$**:** To show that two values μ and ν are coprime, we show that $(\gcd(\mu, \nu) = 1)$, which is true if and only if there exist integers α and β such that Bézout's identity is $\gcd(\mu, \nu) = 1 = \alpha\mu + \beta\nu$:

$$\begin{aligned} S_{\gcd} := PK\{ (\mu, \nu, \rho_\mu, \rho_\nu, \alpha, \beta, \rho) : \\ C_\mu = g^\mu h^{\rho_\mu} \wedge C_\nu = g^\nu h^{\rho_\nu} \wedge \\ g = C_\mu^\alpha C_\nu^\beta h^\rho \\ \}. \end{aligned}$$

$\underline{(\mu > 0)}$**:** We can show with a predicate $(\mu > 0)$ that an integer μ is greater than zero, proving knowledge of four integers χ_1, χ_2, χ_3, χ_4 such that $\mu = \sum_{i=1}^4 \chi_i^2$:

$$\begin{aligned} S_{>0} := PK\{ \left((\chi_i, \rho_{\chi_i})_{i=1}^4, \rho\right) : \\ (C_{\chi_i} = g^{\chi_i} h^{\rho_{\chi_i}})_{i=1}^4 \wedge \quad C_\mu = \prod_{i=1}^4 \left(C_{\chi_i}^{\chi_i}\right) h^\rho \\ \}. \end{aligned}$$

$\underline{\left(a^b \equiv d \pmod{n}\right)}$**:** Camenisch and Michels (Camenisch and Michels, 1999) introduced statistical zero-knowledge arguments for secret exponentiation we will adapt in two forms:

- $\left(a^b \equiv d \pmod{n}\right)$: signifies a secret modular exponentiation with secrets a, b, d, and n, realized with computing a committed square-and-multiply $\pmod{n}$ with respect to the committed bit-representation of b.

$(a^b = d)$: means a secret exponentiation over the integers with secrets a, b, and d, which can be easily adapted from the former.

Both forms have in common that they produce intermediary commitments C_{v_i} to either $a^{2^i} \pmod n$ in the former case or C_{v_i} to a^{2^i} over the integers in the latter case.

Theorem 5 (Secret ModExp (Camenisch and Michels, 1999, p. 114))**.** *Let c_a, c_b, c_d, and c_n be commitments on integers a, b, d, and n and let $c_{b_0}, \ldots, c_{b_{\ell-1}}, c_{v_1}, \ldots, c_{v_{\ell_b-1}}, c_{u_0}, \ldots, c_{u_{\ell_b-2}}$ be auxiliary commitments. Then assuming computing discrete logarithms in G is infeasible, the protocol $S_\uparrow$ is a statistical zero-knowledge argument that the equation $a^b \equiv d \pmod n$ holds. The soundness error probability is 2^{-k}.*

$\underline{(a_i \in \mathbb{Z}_n)_{i=1}^k}$**:** This proof predicate governs the joint generation of group elements $a_i \in \mathbb{Z}_n$, where the verifier may not be privy of n, showing that the bases were generated correctly with randomness from both prover and verifier. This predicate is part of the primality protocol (Camenisch and Michels, 1999). The full version contains a modular construction.

$\underline{(\mu \in \mathsf{primes}_{\mathsf{L}}(t))}$**:** Camenisch and Michels offered a PK predicate we call $\mathsf{primes}_{\mathsf{L}}(t)$ proving that a committed number is prime using a predicate S_p for a secret execution of Lehmann's primality test (Lehmann, 1982).

Theorem 6 (Primality (Camenisch and Michels, 1999, p. 118))**.** *Assume computing discrete logarithms in G is infeasible. Then, [the protocol $\mathsf{primes}_{\mathsf{L}}(t)$ on a commitment c_n] is a statistical zero-knowledge argument that the integer committed to by c_n is a prime. The soundness error probability is at most $2^{-k} + 2^{-t}$.*

$\underline{(\mu \in \mathsf{composites}())}$**:** This predicate proves that a number μ is a composite by proving knowledge of an integer $a \in \mathbb{Z}_n^*$ that is a Lehmann compositeness witness such that $a^{(n-1)/2} \not\equiv \pm 1 \pmod n$. We include details of this protocol in the full version. It holds:

Theorem 7. *If the discrete-logarithm problem is hard in group G, then the protocol* $\mathsf{composites}()$ *on commitment C_n is a statistical zero-knowledge argument that the integer n committed to by C_n is a composite. The soundness error probability is at most 2^{-k}.*

$\underline{(\mu = \mathcal{H}_{\zeta,\beta}(\nu))}$**:** This predicate proves that a committed value μ is the output of a square hash SQHU $\mathcal{H}_{Q,z,b}$ with key (ζ, β) on committed input ν. A corresponding predicate $(\mu = \mathcal{H}_\zeta(\nu))$ for SQH $\mathcal{H}_{Q,z}$ follows trivially. When evaluated in $\mathbb{Z}_Q$, we omit parameter Q.

$$
\begin{aligned}
S_{SQHU} := PK\{\, (\mu, \rho_\mu, \nu, \rho_\nu, \zeta, \rho_\zeta, \bar{\mu}, \rho_{\bar{\mu}}, \beta, \rho_\beta, \gamma, \delta, \eta) : \\
C_\mu = g^\mu h^{\rho_\mu} \wedge C_\nu = g^\nu h^{\rho_\nu} \wedge \\
C_\zeta = g^\zeta h^{\rho_\zeta} \wedge C_{\bar{\mu}} = g^{\bar{\mu}} h^{\rho_{\bar{\mu}}} \wedge \\
C_{\bar{\mu}} = C_\zeta C_\nu h^\gamma \wedge C_{\hat{\mu}} = C_{\bar{\mu}}^{\bar{\mu}} h^\delta \wedge \\
C_\beta = g^\beta h^{\rho_\beta} \wedge C_\mu = C_{\hat{\mu}} C_\beta h^\eta \\
\}.
\end{aligned}
$$

$\underline{(\mu = \mathsf{PRG}_\xi(\nu))}$**:** This predicate proves that μ is the output of pseudo-random generator PRG on seed ξ and input ν. We can prove predicate of the Naor-Reingold PRG in discrete-logarithm-based zero-knowledge proofs efficiently that the relation $\mu = f_{Q,P,\mathfrak{g},\vec{a}}(\nu)$ is fulfilled in $\mathfrak{G}$. We include a detailed construction of this predicate in the full version of this paper.

4 SECURITY REQUIREMENTS

Definition 2 (Requirements (Menezes et al., 2018))**.** *We expect the following properties of our interactive proof systems.*

Completeness: *An interactive proof system is* complete *if, given an honest prover and an honest verifier, the protocol succeeds with overwhelming probability.*

Soundness: *An interactive proof system is* sound *if there exists an exists an expected polynomial-time algorithm M with the following property: if a dishonest prover can with non-negligible probability successfully execute the protocol with the verifier, then M can be used to extract from this prover knowledge which with overwhelming probability allows successful future protocol executions.*

Zero-Knowledge: *A proof of knowledge has the* zero-knowledge *property if there exists an expected polynomial-time algorithm (*simulator*) which can produce, upon input of the assertions to be proven but without interacting with the real prover, transcripts indistinguishable from those resulting from interaction with the real prover.*

We will specify the protocols in the Camenisch-Stadler framework (Camenisch

Table 1: Complexity of known PK predicates for k rounds

	Computations (mexp)		Communication (ge/bits)
	Prover	Verifier	
$(\mu \not\equiv \pm 1)$	$1+3k$	$3k$	$1+3k$ ge $+\,6\log Q$ bits
$(\gcd(\mu,\nu)=1)$	$2+3k$	$3k$	$2+3k$ ge $+\,7\log Q$ bits
$(\mu > 0)$	$5+5k$	$5k$	$5+5k$ ge $+\,5\log Q$ bits
$\left(a^b \equiv d \pmod n\right)$ $\left(a^b = d\right)$	$3\ell_b + (7\ell_b)k$	$(7\ell_b)k$	$3\ell_b$ ge $+\,(14\ell_b \log Q + 4\ell_b\epsilon\ell)k$ bits
$(\mu \in \mathsf{primes}_{\mathsf{L}}(t))$	$2t\log n + (7t\log n)k$	$(7t\log n)k$	$2t\log n$ ge $+\,(14t\log n\log Q + 4t\log n 2\epsilon\ell)k$ bits
$(\mu \in \mathsf{composites}_{\mathsf{L}}())$	$2\log n + (11+7\log n)k$	$(11+7\log n)k$	$2\log n$ ge $+\,(14\log n\log Q + 4\ell_b\epsilon\ell)k$ bits

Note: k = number of PK rounds; mexp = multi-base exponentiations; ge = group elements from $\mathbb{G}$

and Stadler, 1997), which naturally extends to UC proofs in the Camenisch, Krenn and Shoup's corresponding UC framework (Camenisch et al., 2011). We take natural composition of discrete-logarithm based zero-knowledge proofs including their completeness and zero-knowledge properties for granted and focus on the soundness property and soundness error probability quantification.

5 CONSTRUCTIONS

5.1 Miller Primality Predicate

We can execute a deterministic Miller test on a careful selection of bases $(a_j)_{j=1}^t \in \mathbb{Z}_n^*$ to enable a deterministic primality test for small positive integers n, with fixed bitlength $\dot{\ell}_n$ and a soundness error of $\varepsilon_{\mathsf{S}} = 0$. The latter is based on research on strong pseudoprimes to several bases (Pomerance et al., 1980; Jaeschke, 1993).

Prover and Verifier agree on the variant of test to run by establishing the parameters $\dot{\ell}_n$, t, and $(a_j)_{j=1}^t$ as follows: $\left(\mu \in \mathsf{primes}_{\mathsf{M}}(\dot{\ell}_n, (a_j)_{j=1}^t)\right)$: Prover and Verifier agree on bitlength $\dot{\ell}_n$. t is set dependent on a valid fixed base set $(a_j)_{j=1}^t$ such that the least n^* with $\mathsf{spsp}\left((a_j)_{j=1}^t, n^*\right)$ is greater than $2^{\dot{\ell}_n}$.

To establish PK predicate $\mathsf{primes}_{\mathsf{M}}(t)$, let us begin by proving that an odd integer $n > 1$ has the form $n = 2^e u + 1$, with u odd.

1. The prover efficiently finds e such that $n-1 = 2^e u$ with u odd.
2. The prover commits to n in the form of $C_n := g^n h^{r_n}$, to u in the form $C_u := g^u h^{r_u}$, and to constant 2 in the form $C_2 := g^2 h^{r_2}$.
3. The prover computes the strong pseudo-prime equation for the selected bases $(a_j)_{j=1}^t$:
   (a) $d_j := a_j^u \pmod n$.
   (b) If $d_j \not\equiv 1 \pmod n$, the prover finds the $k < e$ such that
   $$d'_j := a_j^{2^k u} \pmod n \wedge d'_j \equiv -1 \pmod n.$$
4. The prover commits to d_j and d'_j in the form of $C_{d_j} := g^{d_j} h^{r_{d_j}}$ and $C_{d'_j} := g^{d'_j} h^{r_{d'_j}}$.
5. The prover sends all commitments to the verifier. Then the prover runs the following protocol with the verifier sequentially for k times:

$$
\begin{aligned}
PK\{\ & (\nu, \alpha, \beta, \mu, \gamma, \upsilon, \varepsilon, \\
& (\varrho_j, \psi_j, \varpi_j, \upsilon_j, \xi_j, \kappa_j, \eta_j, \Delta_j, \zeta_j, \tau_j, \varsigma_j, \delta_j, \\
& \pi_j, \delta'_j, \pi'_j, \vartheta_j, \vartheta'_j)_{j=1}^t, \chi, \big) : \\
& C_n = g^\nu h^\alpha \wedge (\nu > 0) - 2^{\dot{\ell}_n} < \nu < 2^{\dot{\ell}_n} \wedge && (3)\\
& C_2 = g^2 h^\beta \wedge C_u = g^\mu h^\gamma \wedge && (4)\\
& \left(C_{a_j} = g^{\varrho_j} h^{\varpi_j}\right)_{j=1}^t \wedge && (5)\\
& C_{2^e} = g^\upsilon h^{\psi_j} \wedge (2^\varepsilon = \upsilon) \wedge && (6)\\
& C_n / g = C_{2^e}^\mu h^\chi \wedge (\gcd(\mu, 2) = 1) \wedge && (7)\\
& \left(C_{2^{k_j}} = g^{\upsilon_j} h^{\xi_j} \wedge (2^{\kappa_j} = \upsilon_j)\right)_{j=1}^t \wedge && (8)\\
& \left(C_{\Delta_j} = g^{\Delta_j} h^{\zeta_j}\right)_{j=1}^t \wedge && (9)\\
& \left(C_{\Delta_j} = g^\varepsilon / g^{\kappa_j} h^{\eta_j} \wedge (\Delta_j > 0)\right)_{j=1}^t \wedge && (10)\\
& \left(C_{2^{k_j} u} = g^{\mu_j} h^{\tau_j}\right)_{j=1}^t \wedge && (11)\\
& \left(C_{2^{k_j} u} = C_{2^{k_j}}^\mu h^{\varsigma_j}\right)_{j=1}^t \wedge && (12)\\
& \left(C_{d_j} = g^{\delta_j} h^{\pi_j} \wedge C_{d'_j} = g^{\delta'_j} h^{\pi'_j}\right)_{j=1}^t \wedge && (13)\\
& \left(\varrho_j^\mu \equiv \delta_j \pmod \nu\right)_{j=1}^t \wedge && (14)\\
& \left(\varrho_j^{\mu_j} \equiv \delta'_j \pmod \nu\right)_{j=1}^t \wedge && (15)\\
& \left(C_{d_j} / g = h^{\vartheta_j} \vee C_{d'_j} g = h^{\vartheta'_j}\right)_{j=1}^t && (16)\\
\}.&
\end{aligned}
$$

Table 2: Overview of ZK primality arguments

Name	Th.	§	Type	Form p_x	t	Witnesses
Lehmann	2		Monte-Carlo	n/a	$-\log_2 \varepsilon_S$	$(a_j)_{j=1}^t \in_R \mathbb{Z}_n^*$
Det. Miller	3	§5.1	Deterministic	$< \dot{\ell}_n$	fixed t for $\dot{\ell}_n$	fixed $(a_j)_{j=1}^t$
Pocklington	4	§5.2	Constructive/Recursive	$p \cdot r + i$	$\approx \log_2(k+1)$	(p_{i-1}, r_i, a_i)

ε_S: *Soundness error*

Clause 3 establishes the knowledge of candidate n, that n is positive and fulfills the length restriction to $\dot{\ell}_n$. Clause 5 proves knowledge of the bases $(a_j)_{j=1}^t$ committed to in C_{a_j} for $j = 1, \ldots, t$; in the deterministic case, these bases are known publicly. Clause 4 shows the representation to commitments to 2 and u as foundation for the decomposition of $n - 1 = 2^e u$. The following clauses 6 thru 7 establish the composition of $n - 1 = 2^e u$, where the second clause of Line 7 yields that u is indeed odd. Subsequent clauses establish the deterministic Miller test for fixed known bases $(a_j)_{j=1}^t$. First, we prove the correct representation of $C_{2^{k_j}}$, which proceeds similarly to the proof for C_{2^e} in Line 6. Secondly, we establish the difference between 2^e and 2^{k_j} and prove that this difference is greater than zero in Clause 6. The clauses on Line 12 establish the knowledge and structure of commitment $C_{2^{k_j}u}$. The clauses on Line 13 establish the knowledge of the results of the Miller test d_j and d'_j. Clauses 14 and 15 establish the relations of the Miller test $a_j^u \equiv d_j \pmod n$ and $a_j^{2^{k_j}u} \equiv d'_j \pmod n$, where secret μ_j represents $2^{k_j}u$. The final clauses on Line 16 establish the different cases of strong probable prime test, that is, either $d_j = 1$, entailing $a_j^u \equiv 1 \pmod n$, or $d'_j = -1$, entailing $a_j^{2^{k_j}u} \equiv -1 \pmod n$. The proof sketch for the following theorem is included in the appendix.

Theorem 8. *Assuming that the discrete logarithm problem is hard in $\mathbb{G}$ and that the least n^* with $\mathsf{spsp}((a_j)_{j=1}^t, n^*)$ is greater than $2^{\dot{\ell}_n}$, then the protocol $\left(\mu \in \mathsf{primes}_{\mathsf{M}}(\dot{\ell}_n, (a_j)_{j=1}^t)\right)$ is a zero-knowledge argument that the committed integer μ is prime. The soundness error probability is 2^{-k}.*

5.2 Pocklington Primality Witness

The predicate $\nu_j \in \mathsf{primes}_{\mathsf{P}}(\nu_{j-1}, \varrho_j, \chi_j)$ convinces a verifier that Pocklington's criterion is fulfilled for p_j based on a secret Pocklington witness (p_{j-1}, r_j, a_j) such that:

$$a_j^{p_{j-1}r} \equiv 1 \pmod{p_j} \wedge \gcd(a_j^{r_j} - 1, p_j) = 1.$$

We assume that a commitment to p_{j-1} is given as $C_{p_{j-1}} = g^{p_{j-1}} h^{r_{p_{j-1}}}$ and a commitment to r_j is given as $C_{r_j} = g^{r_j} h^{r_{r_j}}$ where $r_{p_{j-1}}, r_{r_j} \in_R \mathbb{Z}_Q$.

1. The prover searches for a positive integer a_j such that Pocklington's criterion is fulfilled for p_{j-1}, r_j, a_j and p_j:
$$a_j^{p_{j-1}r} \equiv 1 \pmod{p_j} \wedge \gcd(a_j^{r_j} - 1, p_j) = 1.$$
If p_j is prime, such an a_j will exist.
2. The prover commits to a_j with the structure $C_{a_j} = g^{a_j} h^{r_{a_j}}$ with $r_{a_j} \in_R \mathbb{Z}_Q$.
3. Then the prover engages with the verifier in the following zero-knowledge proof k times:

$$\begin{aligned}
PK\{\, &(\nu_j, \nu_{j-1}, \varrho_j, \chi_j, \delta_j, \Delta_j, \alpha, \beta, \gamma, \varepsilon, \zeta, \eta, \kappa, \xi, \tau, \pi, \psi) : \\
&C_{p_{j-1}} = g^{\nu_{j-1}} h^{\alpha} \wedge C_{p_j} = g^{\nu_j} h^{\beta} \wedge && (17)\\
&C_{a_j} = g^{\varrho_j} h^{\gamma} \wedge C_{r_j} = d^{\chi_j} h^{\varepsilon} \wedge && (18)\\
&-2^{\ell} < \varrho_j, \chi_j, \nu_j < 2^{\ell} \wedge && (19)\\
&C_{\Delta_j} = g^{\Delta_j} h^{\zeta} \wedge C_{\Delta_j} = g^{\nu_{j-1}} / g^{\chi_j} h^{\eta} \wedge && (20)\\
&(\varrho_j > 0) \wedge (\chi_j > 0) \wedge (\Delta_j > 0) \wedge && (21)\\
&C_{p_j} / g = C_{p_{j-1}}^{\chi_j} h^{\kappa} \wedge && (22)\\
&C_{d_j} = g^{\delta_j} h^{\xi} \wedge && (23)\\
&\left(\varrho_j^{\nu_{j-1}\chi_j} \equiv \delta_j \pmod{\nu_j}\right) \wedge && (24)\\
&C_{d_j} / g = h^{\tau} \wedge && (25)\\
&C_{\lambda_j} = d^{\lambda_j} h^{\pi} \wedge C_{\lambda_j} g = C_{a_j}^{\chi_j} h^{\psi} \wedge && (26)\\
&(\gcd(\lambda_j, \nu_j) = 1) && (27)\\
\}.&
\end{aligned}$$

We assume that the length of p_{j-1} has been established by the corresponding primality zero-knowledge argument. Clauses 17 establish the knowledge of the successive Pocklington primes p_{j-1} and p_j. Clauses 18 proves the knowledge of the remainder of the Pocklington witness a_j and r_j. The clauses 19 thru 21 prove the length restrictions on p_j, r_j, and a_j, where Clause 21 ensures that a_j and r_j are indeed positive integers and that $r < p_{j-1}$ as required by Theorem 4.

Clauses 23 thru 27 prove the Pocklington criterion itself. Clause 23 proves knowledge of the committed result d_j of the Pocklington congruence. Clause 24 shows the structure of the key

Pocklington congruence: $a_j^{p_{j-1}r} \equiv 1 \pmod{p_j}$. Subsequently, Clause 25 shows that the result is indeed congruent to 1. Clause 26 shows the commitment to the term $a_j^{r_j} - 1$ and the subsequent Clause 27 yields the coprimality with p_j. The proof sketch for this theorem is in the Appendix.

Theorem 9. *Assuming that ν_{j-1} has been established to be prime and assuming that the discrete logarithm problem is hard in $\mathbb{G}$, then the protocol $(\nu_j \in \mathsf{primes}_{\mathsf{P}}(\nu_{j-1}, \varrho_j, \chi_j))$ is a zero-knowledge argument that the committed integer ν_j is prime. The soundness error probability is 2^{-k}.*

5.3 Special RSA Modulus

Before we turn to the main contribution of this paper, we show how the Pocklington Witness predicate presented in Section 5.2 directly offers a more efficient zero-knowledge argument that a number n is a product of two safe primes. For that, by proving knowledge of a Pocklington witness we can assert the structure of the constituent safe primes, improving on computational and communication complexity as well as soundness error probability of earlier methods (Camenisch and Michels, 1999).

1. The Prover computes two safe primes $p := 2\tilde{p} + 1$ and $q := 2\tilde{q} + 1$, creating the RSA modulus $n := pq$.
2. The prover searches for two bases a_p and a_q that complete Pocklington witnesses for the primality of p and q such that

$$a_p^{2\tilde{p}} \equiv 1 \pmod{p} \text{ and } \gcd(a_p^2 - 1, p) = 1$$
$$a_q^{2\tilde{q}} \equiv 1 \pmod{q} \text{ and } \gcd(a_q^2 - 1, q) = 1$$

3. We assume a commitment on integer n be given as $C_n = g^n h^{r_n}$. The prover commits to p, q as well as a_p and a_q with $C_p := g^p h^{r_p}$, $C_q := g^q h^{r_q}$, $C_{\tilde{p}} := g^{(p-1)/2} h^{r_{\tilde{p}}}$, $C_{\tilde{q}} := g^{(q-1)/2} h^{r_{\tilde{q}}}$, $C_{a_p} := g^{a_p} h^{r_{a_p}}$, $C_{a_q} := g^{a_q} h^{r_{a_q}}$, where the corresponding randomnesses r_p, r_q, $r_{\tilde{p}}$, $r_{\tilde{q}}$, r_{a_p}, and $r_{a_q} \in_R \mathbb{Z}_Q$. Furthermore, the prover computes all commitments prescribed by $\tilde{p} \in \mathsf{primes}_{\mathsf{L}}(t)$, $\tilde{q} \in \mathsf{primes}_{\mathsf{L}}(t)$, $p \in \mathsf{primes}_{\mathsf{P}}(\tilde{p}, 2, a_p)$, $q \in \mathsf{primes}_{\mathsf{P}}(\tilde{q}, 2, a_q)$.
4. Then the prover and the verifier engage in the following zero-knowledge protocol k times.

$$\begin{aligned}
PK\{\, &(\mu, \nu, \tilde{\mu}, \tilde{\nu}, \varrho_p, \varrho_q, \alpha, \beta, \gamma, \varepsilon, \zeta, \eta, \kappa, \xi) : \\
&C_p = g^{\mu} h^{\alpha} \wedge C_q = g^{\nu} h^{\beta} \wedge && (28)\\
&C_{\tilde{p}} = g^{\tilde{\mu}} h^{\gamma} \wedge C_{\tilde{q}} = g^{\tilde{\nu}} h^{\delta} \wedge && (29)\\
&C_n = C_p^{\nu} h^{\varepsilon} \wedge && (30)\\
&C_p / (C_{\tilde{p}}^2 g) = h^{\zeta} \wedge C_q / (C_{\tilde{q}}^2 g) = h^{\eta} \wedge && (31)\\
&C_{a_p} = g^{\varrho_p} h^{\kappa} \wedge C_{a_q} = g^{\varrho_q} h^{\xi} \wedge && (32)\\
&(\tilde{\mu} \in \mathsf{primes}_{\mathsf{L}}(t)) \wedge && (33)\\
&(\tilde{\nu} \in \mathsf{primes}_{\mathsf{L}}(t)) \wedge && (34)\\
&(\mu \in \mathsf{primes}_{\mathsf{P}}(\tilde{\mu}, 2, \varrho_p)) \wedge && (35)\\
&(\nu \in \mathsf{primes}_{\mathsf{P}}(\tilde{\nu}, 2, \varrho_q)) && (36)\\
\}.&
\end{aligned}$$

The length of μ, ν, $\tilde{\mu}$, $\tilde{\nu}$, ϱ_p and ϱ_q is constrained by the corresponding $\mathsf{primes}_{\mathsf{L}}(t)$ and $\mathsf{primes}_{\mathsf{P}}(n, r, a)$ protocols. While the primality of $(p-1)/2$ and $(q-1)/2$ is proven by secret Lehmann primality tests in clauses 33 and 34. We proceed with proving the primality of the p and q with a Pocklington witness. The following theorem is proven in the Appendix.

Theorem 10. *Assuming that the discrete logarithm problem is hard in $\mathbb{G}$, then the protocol is a zero-knowledge argument that the integer n committed to in C_n is the product of two safe primes p and q, for which $(p-1)/2$ and $(q-1)/2$ are prime as well. The soundness error probability is at most $2^{-k} + 2^{-t}$.*

6 SECRET HASH-TO-PRIME

Definition 3 (Hash-to-Prime)**.**
$(\mu \in \mathsf{hashToPrime}_{\mathcal{F}}(\nu, t))$ is a zero-knowledge predicate stating that committed secret μ an element of the set of prime numbers derived according to a specified procedure $\mathcal{F}$ from committed secret input ν, using t steps.
$\left(\mu = \mathsf{hashToPrime}_{\mathcal{F}}^{\Upsilon}(\nu, t, u)\right)$ is a zero-knowledge proof predicate stating that committed secret μ is exactly the first prime number in sequence derived according to a specified procedure $\mathcal{F}$ from committed secret input ν, using t steps and having eliminated u composite candidates in sequence. The parameters t and u are public knowledge.

We offer two constructions for proving in a zero-knowledge argument that a committed $(k+1)$-bit prime p_x was generated via hashing from an a-bit input x: (i) by elimination (E) and (ii) by recursive construction (R). Both variants of the predicate $\left(\mu = \mathsf{hashToPrime}_{\mathcal{F}}^{\Upsilon}(\nu, t, u)\right)$

have in common that the prover is required to prove the compositeness of u eliminated candidates in a fixed order, enforcing that the first prime in sequence must be used.

6.1 Hash-to-Prime by Elimination

The idea of hashing-to-prime by elimination is that the prover hashes the input x which then seeds a PRG, evaluated in a deterministic sequence with known indices $i = 1, \ldots, u$ until the outcome y_u passes a test as a probable prime. To establish the predicates $(\mu \in \mathsf{hashToPrime}_{\mathsf{E}}(\nu, t))$ and $\left(\mu = \mathsf{hashToPrime}_{\mathsf{E}}^{\Upsilon}(\nu, t, u)\right)$, the prover runs a protocol with the verifier to establish a zero-knowledge argument (i) that the hash- and PRG-computations are executed correctly, (ii) that all committed eliminated candidates $(y_i)_{i=1}^{u-1}$ are composites $(*)$, and (iii) that the final committed value $p_x := y_u$ is prime, where $(*)$ is only executed for predicate $\left(\mu = \mathsf{hashToPrime}_{\mathsf{E}}^{\Upsilon}(\nu, t, u)\right)$.

For specified and setup pseudo-random number generator $\mathsf{PRG}()$, a hash function $\mathcal{H}_{Q,z,b}()$ and corresponding zero-knowledge predicates $(\mu = \mathsf{PRG}_{\chi}(\nu))$ and $(\mu = \mathcal{H}_{\zeta,\beta}(\nu))$, we establish the ZK predicates $(\mu \in \mathsf{hashToPrime}_{\mathsf{E}}(\nu, t))$ and $\left(\mu = \mathsf{hashToPrime}_{\mathsf{E}}^{\Upsilon}(\nu, t, u)\right)$ as follows, where $(*)$ marks the steps transforming the former to the latter:

1. The prover computes $y_i := \mathsf{PRG}_{\mathcal{H}_{z,b}(x)}(i)$ for $i = 1, \ldots, u$ testing each y_i with a Miller-Rabin test till y_u passes the primality test as a probable prime. The prover calls this y_u the outcome p_x.
2. the prover commits to the intermediary output of the hash function $\mathcal{H}_{z,b}()$ as $\bar{y}$, in the form of $C_{\bar{y}} = g^{\bar{y}} h^{r_{\bar{y}}}$ and $r_{\bar{y}} \in_R \mathbb{Z}_Q$.
3. The prover computes commitments on all eliminated composite values y_i as $C_{y_i} := g^{y_i} h^{r_{y_i}}$ with $i = 1, \ldots, u-1$ and r_{y_i} is $\in_R \mathbb{Z}_Q$. $(*)$ The prover commits to the determined prime $p_x = y_u$ in the form of $C_{p_x} := g^{p_x} h^{r_{p_x}}$, where $r_{p_x} \in_R \mathbb{Z}_Q$.
4. The prover sends all commitments, including ones of sub-ordinate predicates to the verifier.
5. Finally, the prover engages with the verifier in a zero-knowledge argument sequentially for k times:

$$
\begin{aligned}
PK\{\ & \left(\nu, \zeta, \beta, \mu, \rho, \rho', \bar{\mu}, \bar{\rho}, (\mu_i)_{i=1}^{u-1}\right) : \\
& C_x = g^{\nu} g^{\rho} \wedge C_{p_x} = g^{\mu} h^{\rho'} \wedge && (37)\\
& (C_{y_i} = g^{\mu_i} h^{\rho_i})_{i=1}^{u-1} \wedge && (38)\\
& \left(C_{\bar{y}_i} = g^{\bar{\mu}_i} h^{\bar{\rho}_i}\right)_{i=1}^{u} \wedge && (39)\\
& (\bar{\mu} = \mathcal{H}_{\zeta,\beta}(\nu)) \wedge && (40)\\
& (\mu = \mathsf{PRG}_{\mu_u}(u)) \wedge && (41)\\
& (\mu \in \mathsf{primes}_{\mathsf{L}}(t)) \wedge && (42)\\
& (\mu_i = \mathsf{PRG}_{\mu_i}(i))_{i=1}^{u-1}\,(*) \wedge && (43)\\
& (\mu_i \in \mathsf{composites}())_{i=1}^{u-1}\,(*) && (44)\\
\}.&
\end{aligned}
$$

The first three clauses 37 thru 39 establish the representation of the commitments. Clause 40 yields the correct computation of the hash function $\mathcal{H}_{Q,z,b}()$ with respect to the secret input x, while the following two clauses 43 and 41 give the correct computation of the PRG. Clause 42 shows the primality of the resulting output p_x. Clause 44 establishes the elimination of intermediary composites $(*)$. The size constraints are governed in the corresponding subordinate ZK predicates.

Theorem 11. *Assuming that the discrete logarithm and the decisional Diffie-Hellman problems are hard in $\mathbb{G}$. Then the protocol $(\mu \in \mathsf{hashToPrime}_{\mathsf{E}}(\nu, t))$ is a zero-knowledge argument that the committed integer μ is prime and was derived as hash-to-prime from committed integer ν. The protocol $\left(\mu = \mathsf{hashToPrime}_{\mathsf{E}}^{\Upsilon}(\nu, t, u)\right)$ is a zero-knowledge argument that the committed integer μ is exactly the first prime in sequence succeding u eliminated candidates derived as hash-to-prime from committed integer ν, where the integer u is publicly known. The soundness error probability is at most $2^{-k} + 2^{-t}$.*

6.2 Hash-to-Prime by Recursion

For a recursive construction, we draw inspiration from Ozdemir et al.'s approach to hashing to primes (Ozdemir et al., 2020) to establish predicates $(\mu \in \mathsf{hashToPrime}_{\mathsf{R}}(\nu, t))$ and $\left(\mu = \mathsf{hashToPrime}_{\mathsf{R}}^{\Upsilon}(\nu, t, (n_j)_{j=0}^{t})\right)$, $(*)$ marking the steps to transform the former to the latter. The recursion has has t steps, each doubling the size of the prime established, using a setup of collection of SQHUs $\left(\mathcal{H}_{Q_j,z_j,b_j}(\cdot)\right)_{j=0}^{t}$.

We start with establishing a first small prime p_0 based on the result of $\mathcal{H}_{Q_0,z_0,b_0}(x)$. From this

first prime, we recursively establish Pocklington steps with their corresponding proofs with predicate $\nu_j \in \mathsf{primes}_{\mathsf{P}}(\nu_{j-1}, \varrho_j, \chi_j)$, while each step roughly doubles the bitlength of the prime p_j. Finally, the prover convinces the verifier (i) that the initial value p_0 is prime with a deterministic Miller predicate, while showing that intermediate candidates were composite $(*)$, (ii) that, for each subsequent value p_j, it is prime with a Pocklington primality witness predicate $\nu_j \in \mathsf{primes}_{\mathsf{P}}(\nu_{j-1}, \varrho_j, \chi_j)$ relating it to the previous prime p_{j-1}, (iii) that, for each primes p_j, the candidates eliminated in finding r_j to complete p_j are composite $(*)$. For clarity, we shall explain the base case and the recursion step separately, even if the protocol is executed as one compound zero-knowledge argument.

Base Case $(j = 0)$**.** We establish the first prime p_0 with a bitlength arbitrarily set to 32 bits. This prime will be derived from input secret x as $p_0 := 2^{\ell_{n_0}} h_0 + n_0$ with $h_0 := \mathcal{H}_{Q_0,z_0,b_0}(x)$ and an integer counter n_0. The primality is established with a deterministic Miller primality predicate (cf. Section 5.1) and constant 3 bases.

1. The prover computes $\mathcal{H}_{Q_0,z_0,b_0}(x)$ and establishes the first integer n_0 in the sequence $1, \ldots, n_0$ such that $p_0 = 2^{\ell_{n_0}} \cdot h_0 + n_0$ is probable prime. The prover stores all intermediate values $y_{0,i} = 2^{\ell_{n_0}} h_0 + i$ for $1, \ldots, n_0 - 1$ that are composites. $(*)$
2. The prover commits to p_0 as $C_{p_0} = g^{p_0} h^{r_{p_1}}$ with $r_{p_0} \in_R \mathbb{Z}_Q$ and to the eliminated composites as $C_{y_{0,i}} = g^{y_{0,i}} h^{r_{y_{0,i}}}$ for $i = 1, \ldots, n_0 - 1$ and with $r_{y_{0,i}} \in_R \mathbb{Z}_Q$ $(*)$ and sends the commitments to the verifier.
3. Then the prover runs the following protocol with the verifier sequentially k times:

$$
\begin{aligned}
PK\{\, &\left(\mu, \alpha, \beta_0, \bar{\mu}, \bar{\alpha}, \nu_0, \rho_0, (\nu_{0,i}, \rho_{0,i}\gamma_{0,i},)_{i=1}^{u}, \gamma\right) : \\
&C_x = g^{\mu} h^{\alpha} \wedge C_{\bar{\mu}} = g^{\bar{\mu}} h^{\bar{\alpha}} \wedge && (45)\\
&C_{p_0} = g^{\nu_0} h^{\rho_0} \wedge && (46)\\
&\left(C_{y_{0,i}} = g^{\nu_{0,i}} h^{\rho_{0,i}}\right)_{i=1}^{n_0-1} \wedge && (47)\\
&(\bar{\mu} = \mathcal{H}_{\zeta_0,\beta_0}(\mu)) \wedge && (48)\\
&C_{p_0} = C_{\bar{\mu}}^{2^{\ell_{n_0}}} g^{n_0} h^{\gamma} \wedge && (49)\\
&\nu_0 \in \mathsf{primes}_{\mathsf{M}}(t) \wedge && (50)\\
&\left(C_{y_{0,i}} = C_{\bar{\mu}}^{2^{\ell_{n_0}}} g^{i} h^{\gamma_{0,i}}\right)_{i=1}^{n_0-1} (*) \wedge && (51)\\
&(\nu_i \in \mathsf{composites}())_{i=1}^{n_0-1} (*) && (52)\\
\}.&
\end{aligned}
$$

Recursion Step $(j - 1 \rightarrow j)$**.** The prover constructs the subsequent prime p_j of the form $p_{j-1} \cdot r_{j,i} + 1$ and established a Pocklington Witness zero-knowledge argument on is primality.

1. Given p_{j-1} and input x committed in C_x, the prover computes $y_{j,i} := p_{j-1} \cdot r_{j,i} + 1$ where the positive integer $r_{j,i} := 2^{\ell_{n_j}} \cdot \mathcal{H}_{Q_j,z_j,b_j}(x) + i$, iterating over $i = 1, \ldots, u$ till $p_j := y_{j,u}$ is a probably prime. The integer u is stored as n_j, the corresponding $r_{j,u}$ is called r_j.
2. The prover searches for a positive integer a_j such that Pocklington's criterion is fulfilled:
$$a_j^{p_{j-1} r} \equiv 1 \pmod{p_j} \wedge \gcd(a_j^{r_j} - 1, p_j) = 1.$$
3. The prover commits to p_j, a_j, r_j as well as all intermediate values $y_{j,i}$ for $i = 1, \ldots, u-1$ $(*)$. The commitments have the forms $C_{p_j} := g^{p_j} h^{r_{p_j}}$, $C_{a_j} := g^{a_j} h^{r_{a_j}}$, $C_{r_j} := g^{r_j} h^{r_{r_j}}$, $C_{y_{j,i}} := g^{y_{j,i}} h^{r_{y_{j,i}}}$, where r_{p_j}, r_{a_j}, r_{r_j}, and $(r_{y_{j,i}})_{i=1}^{u-1}$ are all $\in_R \mathbb{Z}_Q$.
4. The prover sends commitments to the verifier.
5. Then the prover engages with the verifier in the following protocol sequentially k times:

$$
\begin{aligned}
PK\{\, &\left(\bar{\mu}, \bar{\alpha}, \beta_j\, (\chi_{j,i}, \beta_{j,i}, \gamma_{j,i}, \psi_{j,i}, \varphi_{j,i}, \varepsilon_{j,i})_{i=1}^{u}, \varrho_j, \delta_j\right) : \\
&C_{\bar{\mu}} = g^{\bar{\mu}} h^{\bar{\alpha}} \wedge \left(\bar{\mu} = \mathcal{H}_{\zeta_j,\beta_j}(\mu)\right) \wedge && (53)\\
&\left(C_{r_{j,i}} = g^{\chi_{j,i}} h^{\beta_{j,i}}\right)_{i=1}^{u} \wedge && (54)\\
&\left(C_{r_{j,i}} = C_{\bar{\mu}}^{2^{\ell_{n_j}}} g^{i} h^{\gamma_{j,i}}\right)_{i=1}^{u} \wedge && (55)\\
&\left(C_{y_{j,i}} = g^{\psi_{j,i}} h^{\varphi_{j,i}}\right)_{i=1}^{u} \wedge && (56)\\
&C_{a_j} = g^{\varrho_j} h^{\delta_j} \wedge && (57)\\
&(\nu_j \in \mathsf{primes}_{\mathsf{P}}(\nu_{j-1}, \varrho_j, \chi_{j,i})) \wedge && (58)\\
&\left(C_{y_{j,i}}/g = C_{p_{j-1}}^{\chi_{j,i}} h^{\varepsilon_{j,i}}\right)_{i=1}^{u} (*) \wedge && (59)\\
&(\psi_{j,i} \in \mathsf{composites}(t))_{i=1}^{u-1} (*) && (60)\\
\}.&
\end{aligned}
$$

Theorem 12. *Assuming that the discrete logarithm problem is hard in* $\mathbb{G}$*. Then the protocol* $(\mu \in \mathsf{hashToPrime}_{\mathsf{R}}(\nu, t))$ *is a zero-knowledge argument that the committed integer* μ *is prime and was derived as hash-to-prime from committed integer* ν*. The protocol* $\left(\mu = \mathsf{hashToPrime}_{\mathsf{R}}^{\Upsilon}(\nu, t, (n_j)_{j=0}^{t})\right)$ *is a zero-knowledge argument that the committed integer* μ *is the prime created by choosing the first prime in sequence in each of the* t *steps succeeding* $(n_j)_{j=0}^{t}$ *eliminated candidates, derived as hash-to-prime from committed integer* ν*, where the integers* $(n_j)_{j=0}^{t}$ *are publicly known. The soundness error probability is* 2^{-k}*.*

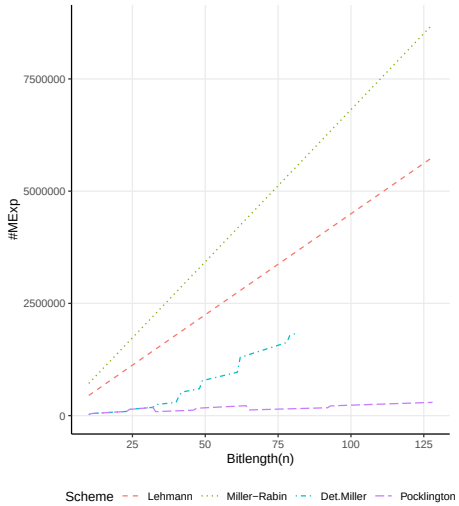

Figure 1: Number of multi-base exponentiations by bitlength of n for a primality ZK argument on small primes $n, |n| \leq 128$, with a fixed number of rounds $k = 80$ and a soundness error probability of at most $2^{-80} + 2^{-80}$

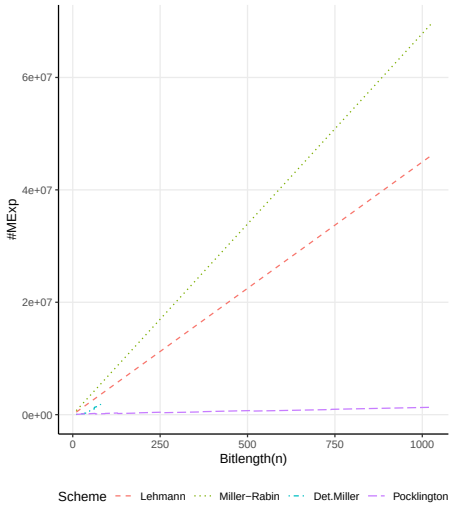

Figure 2: Number of multi-base exponentiations by bitlength of n for a primality ZK argument on primes n, with a fixed number of rounds $k = 80$ and a soundness error probability of at most $2^{-80} + 2^{-80}$

7 COMPLEXITY EVALUATION

We have analyzed the computation complexity of proving the primality of a secret integer n in number of multi-base exponentiations by the bitlength of n and the maximum soundness error probability ε_{S} allowed, which in turn determines the number of primality-test bases used t and the number of ZKP rounds k executed. All simulations are computed in the statistics software R.

7.1 Primality ZK Arguments

We have computed an simulation in R pitting the growth of the number of multi-base exponentiations for different primality predicates by bitlength of n against each other. Figure 1 displays this complexity analysis graphically for small primes; Figure 2 shows the secret primality proof for primes n up to a bitlength of $\ell_n = 1024$. Therein, we notice that the secret primality test with probabilistic Lehmann and Miller-Rabin tests $((\mu \in \mathsf{primes}_{\mathsf{L}}(t))$ and $(\mu \in \mathsf{primes}_{\mathsf{MR}}(t)))$ dominate the complexity, where the Lehmann test proposed by Camenisch and Michels (Camenisch and Michels, 1999) is the more efficient of the two for equal soundness error probabilities. Their complexity is largely dictated by the number of rounds t.

For the new primality predicates proposed in this work, we find that they excel at proving the primality of small primes. While the deterministic Miller test $((\mu \in \mathsf{primes}_{\mathsf{M}}(t)))$ realized here only allows proving the primality of primes of a max bitlength of 81, it can be computed with a small number of fixed based for each bitlength. The Pocklington witness sequence employs $(\nu_j \in \mathsf{primes}_{\mathsf{P}}(\nu_{j-1}, \varrho_j, \chi_j))$ recursively from a threshold bitlength of 32 bits yielding the recursion base with a deterministic Miller test $(\mu \in \mathsf{primes}_{\mathsf{M}}(t))$. Each recursion step doubles the bitlength of the intermediate prime, which leads to a logarithmic growth of the length t of the Pocklington sequence in the bitlength of n.

To put the recursive Pocklington sequence simulation on even footing with the other primality predicates, we included an estimate of the cost that each Pocklington step j needs to find by trial-and-error a new integer r_j to construct the subsequent prime p_j. Given ℓ as bitlength of that prime, we use as expected number of intermediary Miller-Rabin (MR) tests computed per recursion step $\frac{1-1/(\ln(2)\ell)}{1/(\ln(2)\ell)}$. We account for two multi-base exponentiations per MR trial.

7.2 SRSA ZK Arguments

We simulated the zero-knowledge argument that a number n is product of two safe primes, comparing Camenisch and Michels' method based on the Lehmann primality criterion and our new construction using a Pocklington witness from Section 5.3. Figure 3 shows the comparison in number of multi-base exponentiations by bitlength of the product n. In the figure, we can clearly see

Table 3: Computation complexity of constituent predicates for k rounds

	Computations (mexp)
$(\mu \in \mathsf{primes}_{\mathsf{L}}(t))$	$2t \log n + (7t \log n)k$
$(\mu \in \mathsf{primes}_{\mathsf{M}}(t))$	$4 + 6 \log n + (13 + 12t + (3t + 1)(7 \log n))k$
$(\nu_j \in \mathsf{primes}_{\mathsf{P}}(\nu_{j-1}, \varrho_j, \chi_j))$	$18 + 3 \log n + (29 + 7 \log n)k$

Note: mexp = multi-base exponentiations

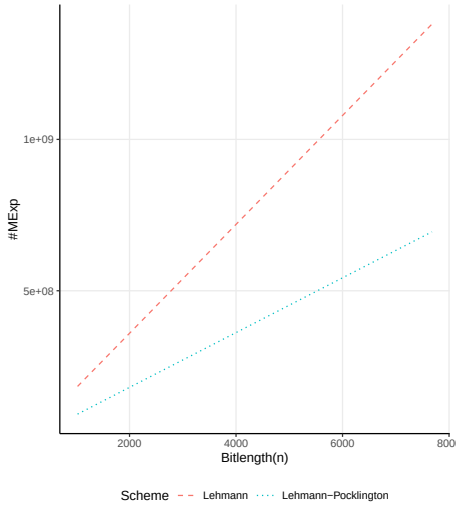

Figure 3: Number of multi-base exponentiations by bitlength of n for the proof that a number n is the product of two safe primes, with a fixed number of rounds $k = 80$ and a soundness error probability of at most $2^{-80} + 2^{-80}$.

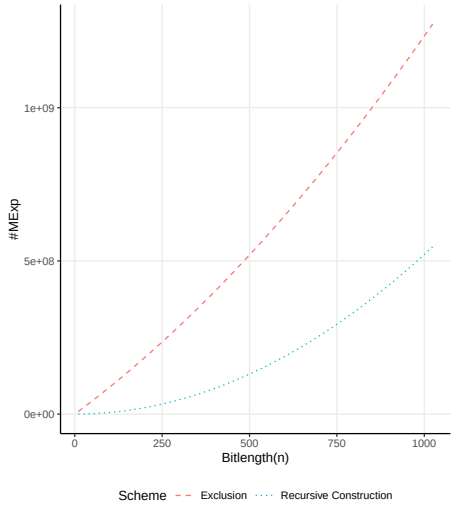

Figure 4: Expected number of multi-base exponentiations for prove predicate $(\mu = \mathsf{hashToPrime}(\nu))$ by bitlength of the output prime p_x

that new Pocklington-Lehmann ZK argument is more efficient and its complexity in mexp growing more slowly that the ZK argument by Camenisch and Michels. It is, thereby, more suitable to compute large special RSA moduli for strong key strengths. For instance, the number of mexp the CM-algorithm uses to establish a ZK argument that a 2048-bit number is a special RSA modulus is roughly the same as to make an equivalent ZK argument for a 4096-bit number with our new method, maximum soundness error probabilities being equal.

7.3 Hash-to-Prime ZK Arguments

We simulated the expected computational complexity of the hash-to-prime zero-knowledge arguments. We computed the expected number of multi-base exponentiations for making a zero-knowledge argument that a committed prime p_x is the outcome of a hash-to-prime operation on a committed input x. We evaluate that based on the bitlength of the output prime p_x. The simulation takes into account the expected number of eliminated prime candidates at each stage and thereby the expected number of calls to the $\mathsf{composite}()$ predicate. Figure 4 illustrates the outcome of the complexity simulation.

We observe in this analysis that the predicate $\left(\mu = \mathsf{hashToPrime}^{\Upsilon}_{\mathcal{F}}(\nu, t, u)\right)$ is considerably more efficient when realized by recursive construction (R) than by elimination (E). This efficiency comes with a trade-off in prime distribution. Whereas the elimination method yields primes that are statistically closely distributed to uniform at random, the recursive construction method yields primes with less entropy and not uniformly distributed. We note that the recursive construction method is quite efficient for hashing to primes with less than 256 bits.

8 CONCLUSION

We showed zero-knowledge arguments for deterministic Miller and Pocklington witness primality, as well as Special RSA modulus correctness. We constructed hash-to-prime zero-knowledge arguments by elimination and recursion.

ACKNOWLEDGMENT

The author would like to thank Ioannis Sfyrakis and Syh-Yuan Tan for the discussions on hashing-to-prime primitives. This work was funded by the ERC Starting Grant CASCAde (GA n°716980).

REFERENCES

Barić, N. and Pfitzmann, B. (1997). Collision-free accumulators and fail-stop signature schemes without trees. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 480–494. Springer.

Brillhart, J., Lehmer, D. H., and Selfridge, J. L. (1975). New primality criteria and factorizations of 2ˆ{m}±1. *Mathematics of computation*, 29(130):620–647.

Cachin, C., Micali, S., and Stadler, M. (1999). Computationally private information retrieval with polylogarithmic communication. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 402–414. Springer.

Camenisch, J. and Groß, T. (2008). Efficient attributes for anonymous credentials. In *Proceedings of the 15th ACM conference on Computer and communications security (CCS 2008)*, pages 345–356. ACM Press.

Camenisch, J. and Groß, T. (2012). Efficient attributes for anonymous credentials. *ACM Transactions on Information and System Security (TISSEC)*, 15(1):4:1–4:30.

Camenisch, J., Krenn, S., and Shoup, V. (2011). A framework for practical universally composable zero-knowledge protocols. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 449–467. Springer.

Camenisch, J. and Lysyanskaya, A. (2002). Dynamic accumulators and application to efficient revocation of anonymous credentials. In *Annual International Cryptology Conference*, pages 61–76. Springer.

Camenisch, J. and Michels, M. (1999). Proving in zero-knowledge that a number is the product of two safe primes. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 107–122. Springer.

Camenisch, J. and Stadler, M. (1997). Efficient group signature schemes for large groups. In *Annual International Cryptology Conference*, pages 410–424. Springer.

Etzel, M., Patel, S., and Ramzan, Z. (1999). Square hash: Fast message authentication via optimized universal hash functions. In *Annual International Cryptology Conference*, pages 234–251. Springer.

Goldreich, O., Micali, S., and Wigderson, A. (1991). Proofs that yield nothing but their validity or all languages in np have zero-knowledge proof systems. *Journal of the ACM (JACM)*, 38(3):690–728.

Groß, T. (2015). Signatures and efficient proofs on committed graphs and NP-statements. In *19th International Conference on Financial Cryptography and Data Security (FC 2015)*, pages 293–314.

Jaeschke, G. (1993). On strong pseudoprimes to several bases. *Mathematics of Computation*, 61(204):915–926.

Kranakis, E. (2013). *Primality and cryptography*. Springer-Verlag.

Lehmann, D. J. (1982). On primality tests. *SIAM Journal on Computing*, 11(2):374–375.

Menezes, A. J., Van Oorschot, P. C., and Vanstone, S. A. (2018). *Handbook of applied cryptography*. CRC press.

Micali, S., Rabin, M., and Vadhan, S. (1999). Verifiable random functions. In *40th annual symposium on foundations of computer science (cat. No. 99CB37039)*, pages 120–130. IEEE.

Naor, M. and Reingold, O. (1997). Number-theoretic constructions of efficient pseudo-random functions. In *Proceedings 38th Annual Symposium on Foundations of Computer Science*, pages 458–467. IEEE.

Ozdemir, A., Wahby, R., Whitehat, B., and Boneh, D. (2020). Scaling verifiable computation using efficient set accumulators. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 2075–2092.

Pomerance, C., Selfridge, J. L., and Wagstaff, S. S. (1980). The pseudoprimes to $25 \cdot 10^9$. *Mathematics of Computation*, 35(151):1003–1026.

APPENDIX

We offer proof sketches for the theorems presented in the paper. The full version of this paper contains the corresponding proofs.

Miller Primality Predicate. Let us consider the proof sketch for Theorem 8:

Proof Sketch. The proof is based on the zero-knowledge properties of the underlying predicates. Using standard techniques a knowledge extractor can extract integers for the secrets in the protocol. Given that the discrete-logarithm problem in $\mathbb{G}$ is assumed hard and provided that the $\log_g h$ is unknown, then the equations encoded on generator g hold in the exponent $\pmod{Q}$. The knowledge extractor gains the integer $\hat{n}$ and integers $\hat{u}$ and $\hat{e}$ for which it establishes the relation $\hat{n} - 1 = 2^{\hat{e}}\hat{u}$ and that $\hat{u}$ is odd. The knowledge extractor gains bases $(\hat{a}_j)_{j=1}^{t}$. We are interested which relations hold for these extracted secrets, especially the Miller primality relations $\hat{a}_j^{\hat{u}} \equiv \hat{d}_j \pmod{\hat{n}}$ and $\hat{a}_j^{2^{\hat{k}_j}\hat{u}} \equiv \hat{d}'_j \pmod{\hat{n}}$. That these relations hold in zero-knowledge follows from Theorem 5. $\hat{d}_j = 1$ and $\hat{d}'_j = -1$ is established with standard techniques. Finally, we have that $\hat{n} < 2^{\hat{\ell}_n} < n^*$, where n^* is the least integer such that $\mathsf{spsp}\left((\hat{a}_j)_{j=1}^{t}, n^*\right)$. Therefore, $\hat{n}$ fulfilling the established relations must be prime. The primality relation established by the extracted secrets are deterministic. Therefore the soundness error probability is 2^{-k}, gained from the number of zero-knowledge proof rounds k. □

Pocklington Primality Witness. The proof sketch for Theorem 9 is as follows:

Proof Sketch. With standard techniques the knowledge extractor extracts integers for the secrets in the protocol. Assuming the hardness of the discrete logarithm and that $\log_g h$ is unknown, equations encoded on g hold in the exponent $\pmod{Q}$. Especially, it gains $\hat{p}_{j-1}$, $\hat{r}_j$, $\hat{p}_j$, and $\hat{a}_j$. In the relations it is assured that $\hat{r}_j$ and $\hat{a}_j$ are positive and that $\hat{r}_j < \hat{p}_j$. Thereby, the conditions for the Pocklington criterion named in Theorem 4 are fulfilled. Two aspects remain to show: First, $\hat{a}^{\hat{p}\cdot\hat{r}} \equiv 1 \pmod{\hat{p}'_j}$, which follows from Theorem 5 and the standard comparison of $\hat{d}$ with 1. Second, $\gcd(\hat{a}^{\hat{r}_i} - 1, \hat{p}_i)$ is shown with the predicate $(\gcd(x, y) = 1)$. Provided that $\hat{p}_{j-1}$ is prime and that these two relations have been established, by $\hat{p}_j$ is prime by Theorem 4. The soundness error probability of 2^{-k} stems from the k rounds of the zero-knowledge proof. □

Special RSA Modulus. We sketch the proof for Theorem 10:

Proof Sketch. It is standard to construct knowledge extractors for the given protocol and to establish the relation between the secrets showing that the following relations between extracted integers hold hold:

$$\hat{p} = 2\tilde{\hat{p}} + 1, \hat{q} = 2\tilde{\hat{q}} + 1, \text{ and } \hat{n} = \hat{p}\hat{q}.$$

The primality of $(\hat{p} - 1)/2$ and $(\hat{q} - 1)/2$ is established as a zero-knowledge argument governed by Theorem 6, yielding a soundness error probability of $2^{-k} + 2^{-t}$ with t being the number of Lehmann primality bases employed. The primality of $\hat{p}$ and $\hat{q}$ is given by the Pocklington witness zero-knowledge argument established in Theorem 9. The latter proven with one base only per predicate and has a soundness error probability of 2^k. □

Secret Hash-to-Prime. Let us consider the proof sketch for the Theorems 11 and 12.

Proof Sketch. For the secrets derived by the knowledge extractor, the Theorems 6, 8, and 9 govern that the committed integer $\hat{p}_x$ is indeed prime with a primality soundness error probability of at most 2^{-t}. Furthermore, proof predicates $(\mu = \mathcal{H}_{\zeta,\beta}(\nu))$ and $(\mu = \mathsf{PRG}_{\xi}(\nu))$ govern that the outputs of hash function and PRG were computed correctly. It remains to show that the prime $\hat{p}_x$ is indeed the first prime in the sequence $\hat{y}_i$ established by the known indices $i = 1, \ldots, u$. For each $\hat{y}_i$ with $i = 1, \ldots, u - 1$, the predicate $(\mu \in \mathsf{composites}())$ establishes that $\hat{y}_i$ is composite by Theorem 7. By contradiction, if $\hat{y}_i$ were prime, it holds by Theorem 2, Clause 1, that $\forall a \in \mathbb{Z}_n^* : a^{(n-1)/2} \equiv \pm 1 \pmod{n}$. Hence, an extracted base $\hat{a} \in \mathbb{Z}_Q^*$ such that $\hat{a}^{(\hat{n}-1)/2} \not\equiv \pm 1 \pmod{\hat{n}}$ does not exist. Consequently, the sequence $(\hat{y}_i)_{i=1}^{u-1}$ does not contain a prime. Therefore, the overall soundness error probability of $(\mu = \mathsf{hashToPrime}_{\mathsf{E}}(\nu, t))$ is at most $2^{-k} + 2^{-t}$ and of $(\mu = \mathsf{hashToPrime}_{\mathsf{R}}(\nu, t))$ is 2^{-k}. We note that the predicates $\left(\mu = \mathsf{hashToPrime}_{\mathsf{E}}^{\Upsilon}(\nu, t, u)\right)$ and $\left(\mu = \mathsf{hashToPrime}_{\mathsf{R}}^{\Upsilon}(\nu, t, (n_j)_{j=0}^{t})\right)$ declare as public knowledge the number of eliminated prime candidates u and $(n_j)_{j=0}^{t}$, respectively, in the predicate specification. □