BANK OF CANADA
BANQUE DU CANADA

Staff Working Paper/Document de travail du personnel—2023-33

Last updated: June 9, 2023

Privacy-Preserving Post-Quantum Credentials for Digital Payments

by Raza Ali Kazmi, Duc-Phong Le and Cyrus Minwalla

Information Technology Services Department
Bank of Canada
rkazmi@bankofcanada.ca, dple@bankofcanada.ca, cminwalla@bankofcanada.ca

DOI: https://doi.org/10.34989/swp-2023-33 | **ISSN** 1701-9397

Acknowledgements

We thank Han Du for reviewing this work and providing valuable feedback.

Abstract

Digital payments and decentralized systems enable the creation of new financial products and services for users. One core challenge in digital payments is the need to protect users from fraud and abuse while retaining privacy in individual transactions. We propose a pseudonymous credential scheme for use in payment systems to tackle this problem. The scheme is privacy-preserving, efficient for practical applications, and hardened against quantum computing attacks. We present a constant-round, interactive, zero-knowledge proof of knowledge (ZKPOK) that relies on a one-way function and an asymmetric encryption primitive—both of which need to support at most one homomorphic addition. The scheme is implemented with SWIFFT as a post-quantum one-way function and ring learning with errors as a post-quantum asymmetric encryption primitive, with the protocol deriving its quantum-hardness from the properties of the underlying primitives. We evaluate the performance of the ZKPOK instantiated with the chosen primitive and find that a memory footprint of 85 KB is needed to achieve 200 bits of security. Comparison reveals that our scheme is more efficient than equivalent, state-of-the-art post-quantum schemes. A practical and interactive credential mechanism was constructed from the proposed building blocks, in which users are issued pseudonymous credentials against their personally identifiable information that can be used to register with financial service providers without revealing personal information. The protocol is shown to be secure and free of information leakage, preserving the user's privacy regardless of the number of registrations.

Topics: Central bank research; Digital currencies and fintech; Payment clearing and settlement systems
JEL codes: E4, E42, G2, G21, O3, O31

Résumé

Les paiements numériques et les systèmes décentralisés permettent la création de nouveaux produits et services financiers. Un des principaux défis liés aux paiements numériques est la nécessité de protéger les utilisateurs contre la fraude et les abus tout en préservant la confidentialité des données associées à chaque transaction. Pour résoudre ce problème, nous proposons un mécanisme de pseudonymisation d'identifiants pour les systèmes de paiement. Ce mécanisme permet de préserver efficacement la confidentialité des données individuelles et offre une protection renforcée contre les attaques basées sur les technologies quantiques. Nous présentons une preuve de connaissance à divulgation nulle de connaissance (ZKPOK) interactive avec un nombre de tours constant. Elle se fonde sur une fonction de cryptage unidirectionnelle et sur une primitive de chiffrement asymétrique post-quantique, chacune devant supporter au plus une opération d'addition homomorphe. L'implantation de ce mécanisme de pseudonymisation est basée sur une fonction de cryptage unidirectionnelle SWIFFT et sur une primitive de chiffrement asymétrique d'apprentissage annulaire avec erreurs. Ces primitives de chiffrement donnent au protocole sa capacité de résistance aux attaques

quantiques. L'évaluation des performances du mécanisme ZKPOK, instancié et utilisant les primitives susmentionnées, nous montre que 85 kilooctets de mémoire sont requis pour obtenir un niveau de sécurité de 200 bits. Une étude comparative avec des dispositifs à la fine pointe nous montre que notre mécanisme ZKPOK est plus efficace que les mécanismes post-quantiques équivalents. Un mécanisme d'identifiants pratique et interactif a été créé à partir des composantes de base que nous proposons. Il fournit aux utilisateurs un pseudonyme associé aux renseignements permettant de les identifier, et ils peuvent ainsi s'enregistrer auprès de fournisseurs de services financiers sans révéler de données personnelles. Le protocole s'avère sûr et prévient toute fuite d'information, ce qui préserve la confidentialité de l'utilisateur, quel que soit le nombre d'enregistrements.

Sujets : Recherches menées par les banques centrales; Monnaies numériques et technologies financières; Systèmes de compensation et de règlement des paiements
Codes JEL : E4, E42, G2, G21, O3, O31

1 Introduction

New innovations such as open banking, smart contracts and decentralized finance reveal the increasing sophistication of financial instruments in the marketplace and speak to a broader digitization of finance. These innovations will empower individuals in previously unforeseen ways and usher in a new economic era. Participating in this new economy can be challenging for individuals, as institutions need to balance user access and privacy while remaining compliant with legal frameworks to prevent money laundering and terrorism financing. Registration can often be an awkward and painful process for users if a service is internet-based and the provider is unable to make physical touch-points available to verify traditional forms of government-issued identification. Furthermore, users are often required to share their personally identifiable information (PII) with multiple service providers, reducing how much control they possess over their own data and increasing the risk of a confidentiality breach under circumstances where a provider suffers a data breach. Intermediaries offering financial services to consumers are compelled by a strict legal and regulatory framework to know their customers. Traditionally, this requires service providers to record every user's PII data upon registration. However, doing so places strong legal, financial and ethical burdens on intermediaries to protect this "treasure trove" of user data against breaches and cyber attacks. To satisfy these conflicting demands, this paper proposes a pseudonymous credential scheme with the following qualities: (1) users can prove ownership of PII in a zero-knowledge fashion, and (2) intermediaries can work with the credential-issuing authority to uncover the identity of a user suspected of illicit activity. The first property would enable users to access services without revealing their personal information, while the second would allow intermediaries to satisfy their regulatory compliance burden while still offering services to said users.

1.1 Related Work

Pseudonymous credentials and privacy-preserving authentication schemes have a long history in cryptography and computer science. Early work by Chaum [1] in 1985 introduces blind signatures to generate pseudonymous credentials. Similarly seminal work by Damgärd [2] in 1988 develops anonymous credentials in the form of a commitment scheme, which negotiates a set of pseudonymous signatures for transactions with participating organizations. Both schemes are privacy-preserving, such that the pseudonymous credentials cannot be linked back to the user. However, Chaum's scheme requires a trusted authority to sign and validate credentials, thereby linking credentials and identifying the user. Damgärd's variant is more private, as even the trusted authority has no mechanism to link two pseudonymous credentials for the same user. Unfortunately, both the Chaum and Damgard schemes are not efficient for practical applications like financial services.

Subsequent credential schemes build on or refine Chaum and Damgärd's initial work. Chen [3] refines Chaum's scheme by replacing the verification mechanism (*cut-and-choose* protocol) with a discrete-logarithm-based blind signature primitive that retains the reliance on a trusted authority to sign each pseudonymous credential issued to the user. Lysanskaya et al. [4] extend Chaum's scheme by associating each user with a master key pair signed and validated by a central authority, and subsequent credentials are issued as one-time pseudonymous pairs that are cryptographically linked to the master secret key. This scheme allows the issuer to confirm keys as valid pairs without revealing the user's identity. Similarly, Brands [5] proposes a scheme to produce one-time signatures by controlling the attributes of the pseudonym. Camenisch et al. [6] propose a private pseudonymous credential scheme

in which a user enrolls at a specific organization via pseudonymous credentials, which are issued by a central authority once the user presents their real credentials. Key features of the scheme include protection against forgeability and the use of circular encryption primitives as an alternative to blind signatures to reveal any reuse of credentials. Camenisch et al. refine this work further [7], culminating in CL-Signatures [8] that create verifiable yet anonymous zero-knowledge proofs of knowledge from the strong RSA assumption. In all of the aforementioned schemes, reliance on an honest central authority is crucial to ensure the security of the protocol. In contrast, Blömer et al. [9, 10] construct practical credentials that are delegable and upgradable. Relying on Pointcheval-Sanders blind signatures [11] as a fundamental building block, the credentials are parameterized as a vector of attributes, wherein verification of one or more attributes can be zero-knowledge. In the schemes discussed so far, either the primitives are insecure against quantum computing attacks or, as in the case of Damgärd, the scheme is purely theoretical.

Quantum-hard pseudonymous credential schemes include work by Ben-Sasson et al. [12], who present an anonymous credential system built on ZK-STARKs and an interactive oracle proof construct that relies on probabilistic verification. This work was further refined into a practical, lightweight implementation called Aurora [13]. Malleable signatures introduced by Chase et al. [14] allow approved transformations to be performed on digitally signed messages while maintaining unforgeability of the original signature. The authors of [14] propose that the scheme is a natural fit for anonymous credentials. However, the construction relies on the availability of a succinct non-interactive argument (SNARG) and fully homomorphic encryption, and the scheme is neither efficient nor practical. Later work by Chase et al. [15] introduces upgrades to the ZKBoo protocol [16], adding ZKB++ as a post-quantum non-interactive zero-knowledge proof, and Fish and Picnic as two options for a lightweight post-quantum signature scheme based on symmetric encryption. Ligero [17], introduced by Ames et al., improves upon Chase's work by reducing the complexity requirement for ZKB++. Both Aurora and Ligero are provably secure from a post-quantum perspective and, as such, are suitable candidates for a performance comparison with the proposed protocol.

1.2 Contributions

This paper proposes a credential scheme that satisfies all five of the qualities described below:

1. An honest user can convince a verifier in polynomial time that a credential $\mathcal{C}$ corresponds to their **PII**.
2. No polynomial-time verifier can learn **PII** from a credential.
3. Any two credentials (even corresponding to the same **PII**) are computationally indistinguishable.
4. No polynomial-time malicious user can convince a verifier that a credential belongs to them except with negligible probability.
5. The issuer can revoke and learn the identity of a user from their credentials.

The credential mechanism can be instantiated by any one-way function and encryption scheme that supports at least one homomorphic addition. The scheme is secure against a polynomial-time quantum adversary, provided the underlying cryptographic primitives are secure against a quantum adversary.

At the heart of our scheme is a new, constant-round, computational zero-knowledge interactive proof for the language,

$$L_f = \{y \mid y = f(s) \text{ for } s \in \{0,1\}^*\},$$

where f is a one-way function that supports at least one homomorphic addition. The scheme is instantiated with SWIFFT, a provably secure post-quantum hash function [18]; and the ring learning with errors (RLWE) public-key cryptosystem [19], a provably secure post-quantum asymmetric encryption algorithm.

2 Building Blocks

2.1 Notations

Let $\mathbb{N}$ be the set of positive integers. For $k \in \mathbb{N}$, we set $[k] = \{1, \cdots, k\}$. We denote the set of all binary strings of length n by $\{0,1\}^n$. An element $s \in \{0,1\}^n$ is called a bit string, and $|s| = n$ denotes its length. Given two bit strings x and y of equal length, we denote their bitwise XOR by $x \oplus y$. For a finite set X, the notation $x \xleftarrow{\$} X$ indicates that x is selected uniformly at random from X.

A function $\texttt{negl} : \mathbb{N} \to \mathbb{R}^+ \cup \{0\}$ is *negligible* if for every positive polynomial, $poly(n)$, there exists a positive integer n_0 such that for all $n > n_0$, $\texttt{negl}(n) < 1/poly(n)$. A typical use of negligible functions is to indicate that the probability of success of some algorithm is too small to be amplified to a constant by a feasible (*i.e.*, polynomial) number of repetitions.

2.2 Computational Assumptions

Let $\Pi = (\texttt{KeyGen}, \texttt{Enc}, \texttt{Dec})$ be a CPA-secure public-key encryption scheme that supports at least one homomorphic addition and where `Keygen` and Enc are probabilistic polynomial-time algorithms and `Dec` is a deterministic polynomial-time algorithm. The `Keygen` takes a security parameter 1^λ and outputs a pair of public/private keys (pk, sk). `Enc` takes a public key and a message m from underlying plaintext space and outputs a ciphertext $c \leftarrow \texttt{Enc}_{pk}(m)$. `Dec` takes a private key sk and a ciphertext c, and outputs a message m or a symbol $\perp$ denoting failure. We define a language L_f for f ,

$$L_f = \{y \mid y = f(x) \text{ for } x \in \{0,1\}^*\},$$

where f is a one-way function that supports at least one homomorphic addition.

3 Constant-Round Zero-Knowledge Proof of Knowledge

In this section, we provide a constant-round computational zero-knowledge interactive proof of knowledge for the language L_f (Section 2.2) from any CPA-secure public-key encryption scheme Π that supports at least one homomorphic addition.

Theorem 1. *The Algorithm 1 is a (constant-round) computational zero-knowledge interactive proof of knowledge for L_f with an efficient (polynomial-time) prover.*

Proof. The protocol must be shown to satisfy the following four properties.

(a) **Completeness:** Suppose the statement is true i.e., $y \in L_f$. Then there exists an x such that $y = f(x)$. Note that ciphertexts arereceived by the prover is of form:

$$c_i = \begin{cases} Enc_{pk}(s_i), & \text{if } b_i = 0 \\ Enc_{pk}(x + s_i), & \text{otherwise.} \end{cases}$$

Common Input: $y \in L_f$.

1. Prover generates a public/private key-pair $(pk, sk) \leftarrow \texttt{Keygen}(1^\lambda)$.
2. Prover sends $(pk, Enc_{pk}(x))$ to the Verifier. Note x is the secret, such that $y = f(x)$.
3. Verifier computes for each $i \in [k]$, where $k \in O(poly(|y|))$.
   - $b_i \xleftarrow{\$} \{0,1\}$
   - $s_i \xleftarrow{\$} \{0,1\}^{|y|}$
   - Verifier computes ciphertexts

$$c_i = \begin{cases} Enc_{pk}(s_i), & \text{if } b_i = 0 \\ Enc_{pk}(x) + Enc_{pk}(s_i), & \text{otherwise.} \end{cases}$$

   - Verifier sends $(c_1, \ldots, c_k)$ to the prover.
4. Prover:
   - Using the private-key, sk computes $m_i = Dec_{sk}(c_i)$ for each $i \in [k]$.
   - Prover computes $f(m_1), \ldots, f(m_k)$.
   - Prover sends $(f(m_1), \ldots, f(m_k))$ to the verifier.
5. Verifier accepts the proof if and only if for all $i \in [k]$.

$$\Big(\big(b_i = 0 \wedge f(m_i) = f(s_i)\big) \vee \big(b_i = 1 \wedge y = f(m_i) \oplus f(s_i)\big)\Big) = 1.$$

Algorithm 1: Computational zero-knowledge proof for the language L_f

An honest prover can recover plaintexts from each ciphertext c_i and from the plaintexts can compute the quantity $f(x+s_i)$ or $f(s_i)$. The verifier knows b_i, s_i, $f(x)$ and can check using the homomorphic property of f if y is in the language. Therefore an honest verifier will be convinced that the input y is in the language.

(b) **Soundness:** Suppose $y \notin L_f$, *i.e.*, there does not exist a binary string x such that $y = f(x)$. Then, the only way a prover can deceive a verifier is to guess bits b_i for all $i \in [k]$ and send $f(s_i)$ if $b_i = 0$ and $y + f(s_i)$ otherwise. The probability of a verifier correctly guessing all randomly and independent chosen bits is 2^{-k}. Therefore, the protocol is sound.

(c) **Complexity** Clearly both the prover and verifier run in expected polynomial time.

(d) **Zero-Knowledge:** Let V^* denote a verifier (possibly malicious). What V^* learns by participating in the proof can be described by the following transcript:

$$(y, pk, Enc_{pk}(x), (c_1, \ldots, c_k), (f(m_1), \ldots, f(m_k))) .$$

For any $y \in L_f$, we denote the set of all possible transcripts that could be produced as a result of the prover and verifier carrying out the interactive proof as

$$\mathcal{T}(V^*, y).$$

We introduce a simulator $\mathcal{S}$ that can forge transcripts as perAlgorithm 2.
Let $P_{\mathcal{T}(V^*,y)}(T)$ denote the probability that a transcript $T \in \mathcal{T}(V^*, y)$ is produced by V^* taking part in the proof. Let $\mathcal{T}(\mathcal{S}(V^*), y)$ be the set of transcripts generated by the

Common Input: $y \in L_f$

1. Generate a public-private key pair $(pk', sk') \leftarrow \texttt{Keygen}(1^\lambda)$.
2. Compute $Enc_{pk'}(x')$ for any plaintext x' in the message space.
3. Concatenate y, pk' and $Enc_{pk'}$ onto the end of transcript $T_{\mathcal{S}}(y)$.
4. Call V^* with input y, pk', and $Enc_{pk'}(x')$ and obtain ciphertext $c'_1, \ldots, c'_k$.
5. Compute $m'_i = Dec_{sk}(c'_i)$ for $i \in [k]$.
6. Compute $f(m'_1), \ldots, f(m'_k)$.
7. Concatenate $(c'_1, \ldots, c'_k)$ and $(f(m'_1), \ldots, f(m'_k))$ onto the end of $T_{\mathcal{S}}(y)$.

Algorithm 2: Simulator $\mathcal{S}$ for forging transcripts

simulator $\mathcal{S}(V^*)$ on input y and given the oracle access to V^*. Let $P_{\mathcal{T}(\mathcal{S}(V^*),y)}(T)$ denote the probability that a transcript $T \in \mathcal{T}(V^*, y)$ is produced by $\mathcal{S}(\mathcal{V}^*)$.
To prove computational zero-knowledge, we need to illustrate that the probability distributions $P_{\mathcal{T}(V^*,y)}(T)$ and $P_{\mathcal{T}(\mathcal{S}(V^*),y)}(T)$ are computationally indistinguishable,

$$P_{\mathcal{T}(V^*,y)}(T) \approx_c P_{\mathcal{T}(\mathcal{S}(V^*),y)}(T).$$

The transcript generated in the interactive proof is

$$(y, pk, Enc_{pk}(x), (c_1, \ldots, c_k), (f(m_1), \ldots, f(m_k)))\,,$$

and the transcript generated by the simulator is

$$(y, pk', Enc_{pk'}(x'), (c'_1, \ldots, c'_k), (f(m'_1), \ldots, f(m'_k)))\,.$$

Note that both have the same common inputs y, the keys pk, and pk' are generated using the same key generation algorithm `KeyGen` with the same security parameter as an input. Moreover, it follows from semantic security of the encryption scheme that both encryptions $Enc_{pk'}(x')$ and $Enc_{pk'}(x')$ are computationally indistinguishable. Therefore,

$$(y, pk, Enc_{pk}(x)) \approx_c (y, pk', Enc_{pk'}(x')).$$

Each $i \in [k]$ ciphertexts c_i and C'_i are produced by V^* (on inputs $y, pk', Enc_{pk'}(x')$), but since $(y, pk, Enc_{pk}(x)) \approx_c (y, pk', Enc_{pk'}(x'))$. Therefore,

$$(c_1, \ldots, c_k) \approx_c (c_1{}', \ldots, c_k{}')$$

are also computationally indistinguishable.
Finally, $f(m_i)$'s and $f(m'_i)$'s are completely determined by c_i, $c_i{}'$ and f we have

$$(f(m_1), \ldots, f(m_k)) \approx_c (f(m'_1), \ldots, f(m'_k)))\,.$$

Therefore, the two probability distributions are computationally indistinguishable, *i.e.*,

$$P_{\mathcal{T}(V^*,y)}(T) \approx_c P_{\mathcal{T}(\mathcal{S}(V^*),y)}(T).$$

□

(e) **Proof of Knowledge:** To prove that Algorithm 1 is also a proof of knowledge, we first slightly modify this algorithm to Algorithm 6 (see Appendix A). The modified protocol is information-theoretically equivalent and has the same time complexity (up to a polynomial factor) as the original protocol. The reason for this modification is that it is much simpler to construct an extractor (Algorithm 3) for the modified version [1]. Let P^* be a (possibly malicious) prover that convinces the honest verifier with probability 1.

1. Initialize P^*: Copy fresh random bits to the prover's random tape and fill up the Auxiliary-Input tape with a witness x.
2. Run P^* to obtain $Enc_{pk}(x)$. The prover is now in state Q.
3. Compute for each $i \in [k]$,
   - $b_i \xleftarrow{\$} \{0,1\}$
   - $s_i \xleftarrow{\$} \{0,1\}^{|y|}$
   - $c_i = b_i \cdot Enc_{pk}(x) + Enc_{pk}(s_i)$
- Send $(c_1, \ldots, c_k)$ to P^* and obtain

$$(b_1x + s_1 + t_1, f(b_1x + s_1), \ldots, (b_1x + s_1 + t_1, f(b_kx + s_k))$$

4. Rewind P^* to state Q.
5. Compute for each $i \in [k]$,
   - $b_i \xleftarrow{\$} \{0,1\}$
   - $c'_i = b'_i \cdot Enc_{pk}(x) + Enc_{pk}(s_i)$
- Send $(c'_1, \ldots, c'_k)$ to P^* and get

$$\left(b'_1x + s_1 + t_1, f(b'_1x + s_1), \ldots, (b'_kx + s_k + t_k, f(b'_kx + s_k))\right).$$

6. Pick any $i \in [k]$ such that $b_i \neq b'_i$.
7. Output

$$x = \begin{cases} b'_ix + s_i + t_i - (b_ix + s_i + t_i), & \text{if } b_i = 0 \\ b_ix + s_i + t_i - (b'_ix + s_i + t_i), & \text{otherwise.} \end{cases}$$

Algorithm 3: Knowledge extractor $\mathcal{E}$

Note the extractor fails if for all $i \in [k]$, $b_i = b'_i$, which happens with probability 2^{-k}. Therefore, the knowledge error here is 2^{-k}.

3.1 Instantiation of Constant-Round ZKIP

In this section, the protocol is instantiated with cryptographic primitives that are provably secure against polynomial-time quantum adversaries and are efficient enough for practical application. Our protocol requires a one-way function and a semantically secure encryption that supports at least one homomorphic addition.

Lattice-based cryptography offers provable security against quantum attacks. Several works show [20] that lattice-based PKE can have performance competitive with those based

[1] The main difference is in step 4, the prover also sends n uniformly random strings to the verifier (see crefsec:appen).

on classical mechanisms like RSA or discrete logarithm. Among lattice-based PKEs, schemes based on learning with errors (LWE) [21] or its variants, such as RLWE, [19] are more efficient. The LWE problem was introduced by Regev in [21] and proven to be as secure as certain lattice problems under a quantum reduction. This problem can be seen as a generalization of the learning parity with noise (LPN) problem [22]. In the original Regev PKE scheme, which is based on LWE, the public and secret keys are very large (megabytes). A more efficient LWE-based PKE is Frodo [23, 24], an alternate in the NIST PQC competition [2]. Frodo's public key and ciphertext sizes are both approximately 11 KB each. Using structured lattices in the LWE problem can lead to more efficient schemes in terms of space. Both RLWE public-key cryptosystems [19, 25–27] and module learning with errors (MLWE) [28, 29][3] offer small public key and ciphertexts approximately $10\times$ smaller than plain LWE-based cryptosystems.

SWIFFT for One-Way Function SWIFFT [18] is a family of hash functions that are efficient, highly parallelizable and provably secure against quantum-safe adversaries. The throughput of SWIFFT is comparable to that of SHA-2, if not better, on modern computers [18]. This is considering that the parallelization of SWIFFT is not fully exploited [18]. The SWIFFT takes a binary string of length $m'n'$ as an input and outputs a binary string of length $n' \log_2 p$ (for appropriate parameters n', m' and p). One recommended choice for $m'n'$ is (see [18] for details)

$$n = 2^6, \quad m = 2^4, \quad p = 257.$$

For these concrete values the hash function takes an input of length 1024 bits and outputs a message digest of 528 bits, with throughput of 40 MB/s and 106 bits of security. [4] For parameters $n' = 2^7, m' = 2^4, p = 257$ SWIFFT maps 2048 bits to 1024 bits and provides 206 bits of security [18].

RLWE Cryptosystem for Encryption In [21], Regev introduces the LWE problem, which is proven secure under quantum attacks. It was demonstrated that solving a random LWE instance is as hard as solving certain worst-case instances of certain lattice problems. The RLWE problem is a variant of LWE that is defined over the ring of integers or polynomial rings [19, 25]. Due to its efficient public key and ciphertext in terms of space, a RLWE public-key cryptosystem is a viable option. The encryption scheme is additively homomorphic and provably semantically secure against quantum adversaries under worst-case lattice assumptions. In addition, the schemes have much smaller key sizes compared with most other post-quantum public key encryption schemes. For instance, NewHope [26, 30] offers both public key and ciphertext sizes are approximately 1 KB each for a 100+ bit security level.

3.2 Cost of the Zero-Knowledge Proof of Knowledge

The communication cost of the protocol is the total number of bits exchange between prover and verifier:

[2] https://csrc.nist.gov/projects/post-quantum-cryptography

[3] Kyber and Saber are both finalists in the NIST PQC competition.

[4] The implementation was tested on a 3.2GHz Intel Pentium 4, written in C, and compiled using gcc version 4.1.2 (compiler flags -O3) on a PC running under Linux kernel 2.6.18. The implementation can be found at https://github.com/micciancio/SWIFFT.

$$|pk| + |Enc(x)| + \sum_{i=1}^{k} |c_i| + \sum_{i=1}^{k} |f(m_i)|,$$

where k is the number of messages send by the verifier to the prover (see Algorithm 1). If we implement our protocol with SWIFFT and RLWE, then the size $Enc(x)$ is $n(1 + \log(q))$ [31]

$$|pk| + (k + 1) \cdot n(1 + \log(q)) + k|\text{SWIFFT}(m_i)|.$$

RLWE encryption provides at least 233 bits of security against quantum attacks for parameters $n = 1024, q = 12289$, and $|pk| = 1824$ bytes, respectively [30, 32]. Recall that for 106 and 206 bits of security, SWIFFT generates a message digest of 528 bits and 1024 bits, respectively. Therefore, to provide at least 206 bits of security against quantum attacks with soundness error 2^{-40}, the communication cost of our protocol is

$$\frac{1824 \cdot 8 + 41 \cdot (1024(1+\log(12289))) + 40 \cdot 1024}{8000} \approx 85 \text{ kB} .$$

A similar analysis shows that to achieve 101 bits of security against quantum attacks, we can set $n = 512, q = 12289$, and $|pk| = 928$ bytes []. This cost in communication is

$$\frac{928 \cdot 8 + 41 \cdot (512(1+\log(12289))) + 40 \cdot 528}{8000} \approx 42 \text{ kB} .$$

Table 1 provides a comparison between our proposed scheme and existing zero-knowledge proof protocols. All considered protocols are plausibly post-quantum secure.

3.3 Comparison with Other Schemes

Table 1 shows a comparison between communication cost of our and various other zero-knowledge proof systems that are plausibly secure against quantum attacks. However, the security and parameters of our scheme are based on primitives that are rigorously evaluated against known quantum attacks [30, 32], whereas other schemes in Table 1 have analyzed security against only classical adversaries. It may be that the communication cost for these proof systems increases when their security is analyzed against quantum attacks.

Name	Round(s)	Security	Soundness	Communication Cost
Ligero [17]	2	128 bit	unbounded adversaries	4000 kB
zk-STARK [12]	1	128 bit	polynomial-time adversaries	3200 kB
Aurora	1	128 bit	polynomial-time adversaries	130 kB
this work	3	206 bit	unbounded adversaries	85 kB
this work	3	101 bit	unbounded adversaries	42 kB

Table 1. Communication complexity of the ZKPOK.

A straightforward run-time comparison of schemes with those identified in Table 1 is not possible, as our scheme is procedural whereas others are circuit-based. Nevertheless, in Table 1 we have provided an asymptotic comparison between our proof system and these proof systems.

All schemes are transparent in that a trusted setup is not required. While the security of our protocol is based on the hardness of the LWE problem and can be proven in the standard

Name	Prover Time	Verifier Time	Operation Type
Ligero [17]	$O(N\log(N))$	$O(N)$	finite field
zk-STARK [12]	$O(N\log(N))$	$O(N\log(N)^2)$	finite field
Aurora	$O(N\log(N))$	$O(N)$	finite field
this work	$O(n\log(n))$	$O(n\log(n))$	finite quotient ring

Table 2. Computation time complexity of the ZKPOK. Here, N is the number of gates in the circuit and the cost is the number of field operations, whereas lowercase n is the degree of the cyclotomic polynomial $\phi(x)$ and the cost is the number of operations in quotient ring $\mathbf{F}_q[x]/\phi(x)$.

model, the security of other protocols is reduced to the collision resistance of hash functions in the random oracle model.

While our zero-knowledge proof of knowledge (ZKPoK) is interactive, requiring three rounds, other schemes are non-interactive. Although a non-interactive protocol requiring no synchronization between the prover and the verifier is useful when we need to broadcast a proof to more than one verifier, an interactive scheme is applicable in authentication/authorization scenarios controlling access to services. Furthermore, Pass [33] has shown that non-interactive zero-knowledge schemes do not preserve *deniability* that may be required in authentication systems.

4 Pseudonymous Credentials

4.1 Overview

We now take the underlying building blocks to develop a pseudonymous credential issuance and verification scheme. Figure 1 illustrates the enrollment and authentication workflow for the credential management scheme. Issuers are organizations authorized to issue credentials against government-issued identification or equivalent by conducting an approved Know Your Customer (KYC) process. Although the most typical form of identification is likely to be federal and provincial government-issued ID such as a driver's license, the space is evolving as third-party providers and digital identity providers, including self-sovereign identity, are gaining market share in this space. Note that an ID provider can also act as a credential issuer, potentially simplifying the KYC process. A credential data store with limited read access and privileged write access is deployed and operated by an arm's-length organization. A smaller pool of issuers are allowed to create, append, update, or revoke credentials on the store, whereas a larger pool of service providers can read and query credentials. Service providers may be established organizations offering financial products, emerging fintechs, payment service providers, or other entities as appropriate. Practical implementation for the store can take the form of a distributed database, a distributed ledger, or some other form of redundant, highly available service. Note that the credential data store is a platform owned and operated by an entity distinct from the pool of issuers and the pool of service providers to prevent collusion and information leakage. More broadly, the proposed scheme may be paired with a digital identity service to provide a holistic solution for all cases where government-issued identification is required for registration.

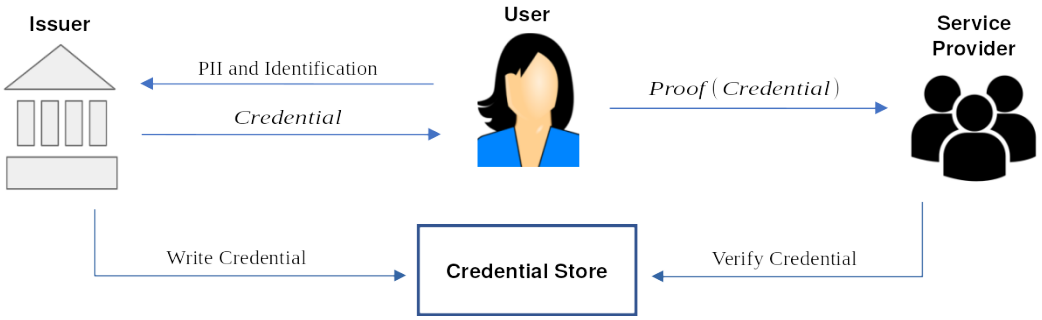

Figure 1. A high-level workflow capturing credential enrollment and verification to register a customer for a service with a service provider against a credential issued by an issuer.

4.2 High-Level Description

A high-level description of a credential enrollment and verification mechanism capturing the entities and relationships is depicted in Figure. 1. The user enrolls in the service by providing a government-issued identification document and associated PII to a credential issuer. The issuer verifies the user through an established verification process such as a KYC check, with appropriate steps taken to prevent identity theft and fraud. Once satisfied, the issuer generates a signed credential against the supplied PII data and updates the credential store. Upon success, the issuer transmits the credentials to the user.

The user may now use the credential to pseudonymously register with financial service providers. To do so, the user provides a handle to the pseudonymous credential and a corresponding zero-knowledge proof attesting to ownership of the credential to the service provider. Note that the attestation of a specific statement must already exist on the credential store prior to verification. The service provider can verify the pseudonymous credentials as being issued by a trusted issuer and the corresponding proof to validate the user.

Once credentials are verified, registration is complete and the user is successfully onboarded into the system. In case of malfeasance, the service provider can approach the issuer with the supplied pseudonymous credentials, who has sufficient information to link the pseudonymous credentials to PII data, revealing the user's identity.

4.3 Formal Definition

A credential mechanism is a procedure that takes $|\mathbf{PII}|$ as input and outputs a credential. We can assume without loss of generality that any **PII** can be converted to a binary string of length $|\mathbf{PII}|$. This operation is agnostic of the specific details of the PII, provided that the credential issuing organization is confident that the supplied documents belong to that individual user and claims made are truthful and valid, *i.e.*, the user has undergone a KYC process. The credentials are constructed with a property that users can prove to an organization some statement about their relationship with another institution anonymously. The mechanism will take as input the **PII** and a publicly known hash function H that supports at least one homomorphic addition.

Definition 1 (Credential Mechanism).
A credential mechanism is a pair of probabilistic polynomial-time algorithms (`CredentialIssue`*,* `CredentialVerification`*) such that:*

- `CredentialIssue` *takes a personal identification information* **PII** *as an input and outputs a credential* $\mathcal{C}$*. It satisfies the following properties:*
  1. *No polynomial-time malicious verifier can learn* **PII** *from* $\mathcal{C}$*.*
  2. *Any two credentials, even if linked to the same* **PII***, are computationally indistinguishable.*
  3. *There exists an efficient mechanism for the issuer to learn or revoke* $\mathcal{C}$ *from* **PII***.*

- `CredentialVerification` *satisfies the following properties:*
  4 *Honest users can convince a verifier in polynomial time that* $\mathcal{C}$ *is linked to their* **PII***.*
  5 *It is with negligible probability that a polynomial-time, malicious user can impersonate or forge a credential to convince the verifier of unauthorized ownership.*

4.4 Credential Mechanism Instantiation

The credential issuance and validation processes are instantiated as per Algorithm 4 and Algorithm 5 as follows:

Common Input: PII

1. User sends **PII** to Issuer.
2. Issuer conducts a KYC process to verify that user's **PII** is legitimate.
3. Issuer picks a binary string $x \xleftarrow{\$} \{0,1\}^l$. In practice $1024 \leq l \leq 2048$.
4. Issuer computes the credential $\mathcal{C} \leftarrow H(x)^a$.
5. Issuer stores (**PII**, x, $\mathcal{C}$) in its private database.
6. Issuer uploads the tuple (pk_I, $Sign_{sk_I}(\mathcal{C})$, $\mathcal{C}$) to the permissioned credential store or blockchain, where sk_I is an issuer's secret signature key.
7. Issuer sends $(x, \mathcal{C}, Sign_{pk_I}(\mathcal{C}))$ along with location metadata to User[b].

[a] $H()$ is a publicly known hash function.
[b] May use any efficient quantum-safe KEM to create a secure channel between issuer and user.

Algorithm 4: `CredentialIssuance`

Common Input: (Credential $\mathcal{C}$, $Sign(\mathcal{C})$)

1. The user supplies the block ID to the service provider to locate the credential.
2. The service provider accesses the permissioned store and extracts the credential $\mathcal{C}$ associated with the block ID. If the credential is valid, the protocol proceeds to the next step. If not, the request is rejected.
3. Using the zero-knowledge interactive proof (Algorithm 1), the service provider satisfies the constraint that the user is the rightful owner of the credential.

Algorithm 5: `CredentialVerification`

Proof of Protocol Correctness

Theorem 2. *Algorithm 4 and Algorithm 5 jointly define a credential mechanism.*

Proof.

- Algorithm 4 constructs a credential $\mathcal{C}$ by applying a hash function H to a uniformly random string. Therefore Algorithm 4 satisfies the first two properties of `CredentialIssuance` (definition 1). Clearly, given a block ID an issuer can recover the identity of the corresponding user efficiently.
- Algorithm 5 is essentially the zero-knowledge proof system (Algorithm 1) defined in Section 3. Therefore, by completeness property of the interactive proof the first property of `CredentialVerification` is satisfied (definition 1). The soundness property ensures that a malicious user can successfully impersonate only with negligible probability. The only other way a malicious user can forge a credential is to find an x' such that $H(x') = H(x)$ or to recover x from the encryption of x, both of which are infeasible for any polynomial-time malicious user, except with negligible probability.

Cost of Implementation Table 3 illustrates the communication cost of Algorithm 4 (`CredentialIssue`). The post-quantum cryptographic schemes of KHYBER [34], DILITHIUM [35], and FALCON [36] were chosen from the NIST Round 3 finalists [37]. Note that this cost does not include the size of **PII** or any associated metadata. In addition, the cost of Algorithm 5 (`CredentialVerification`) is essentially the cost of the zero-knowledge proof(Algorithm 1). If we instantiate Algorithm 1 with RLWE and SWIFFT (Sections 3 and 3.2), then the communication cost of `CredentialVerification` is computed to be 85 kB and 42 kB for 206 bits and 101 bits of security, respectively.

KEM	Signature Scheme	Hash	Security	Cost
KHYBER-512	DILITHIUM-512	SWIFFT-1024	101-bit	72 kB
KHYBER-1024	DILITHIUM-1024	SWIFFT-2048	206-bit	139 kB
KHYBER-512	FALCON-512	SWIFFT-1024	101-bit	37 kB
KHYBER-512	FALCON-1024	SWIFFT-2048	206-bit	68 kB

Table 3. Communication complexity of the credential issuance process (Algorithm 4).

4.5 Practical Verification Protocol

A practical instantiation of the credential verification protocol with defined primitives is illustrated in Fig. 2. The protocol is interactive and constructed using the primitives formally defined in Algorithm 1 and Algorithm 5.

The symbol $\oplus$ denotes a homomorphic addition, while x is the secret information that the prover demonstrates knowledge of via a ZKPOF to convince the verifier. The credential and the associated cryptographic material are stored in a unique block on a permissioned blockchain that can be accessed only by issuers and verifiers. As such, the prover supplies an identifier (ID_{Block}) such that the verifier can find the appropriate block in the chain and extract its contents. The remaining steps in the protocol execute the zero-knowledge proof of knowledge as originally described in Algorithm 1. Note that the verifier must generate the nonces b and s in a uniformly random fashion and keep them secret from the prover, otherwise the prover can cheat.

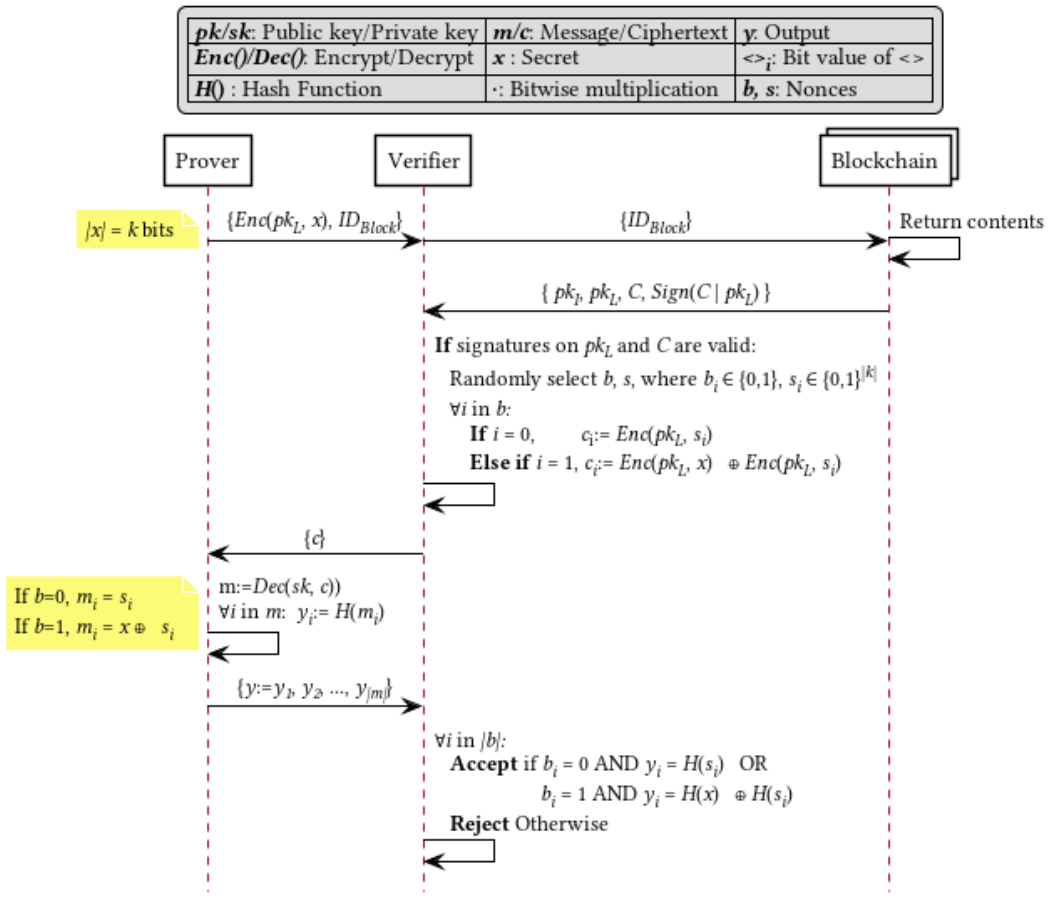

Figure 2. Practical instantiation of a three-round interactive verification protocol, where the prover is the user and the verifier is the service provider.

The described protocol implements single proof of knowledge against a single issued credential. It is straightforward to expand this scheme such that multiple, independent statements of knowledge linked to the same credentials are stored on the block. For instance, the block may store assertions that were validated during the KYC process, such as the age of majority (above 18) or residence in a particular locality, and each statement may be proven independently by the prover as per the requirements of the verifier's registration process, as verifiers may care about only a subset of those statements. However, in all cases the statement must be committed to the blockchain prior to attempting a proof for the verification to succeed.

The privacy guarantees of this protocol are strong. Due to the ZKPOK, the service provider (verifier) learns nothing about the user during the execution of the protocol. Furthermore, the credential issuance and validation processes are asynchronous, and the verification protocol does not require a live connection to the issuer. This ensures that the issuer remains unaware when the credential store is accessed by one or more service providers. Depending on the construction of the block, the verifier may learn details about the issuer,

including the organization name, the date of issuance, and any other associated attributes stored in the block metadata.

5 Conclusion

In this paper, we presented a credential issuance and verification scheme for use in payment systems that can be generalized to any system where registration is required. The building blocks are a constant-time, interactive, zero-knowledge proof relying on a one-way function and asymmetric encryption, both of which need to support only a single homomorphic addition. An instantiation of the ZKPOK based on SWIFFT and RLWE was analyzed and the protocol was formally demonstrated to be secure from post-quantum assumptions. Performance evaluation of the instantiated version indicates that the scheme has a ZKPOK memory footprint of 85 kB and a computational cost bounded by the RLWE encryption operation. A credential issuance and verification mechanism is constructed based on the proposed ZKPOK and a permissioned blockchain. The credential mechanism is shown to be private and secure against adversaries and incurs a communication cost of 68 kB for the issuance process and 85 kB for the verification process to achieve 206-bit equivalent security, if instantiated with efficient post-quantum primitives. Avenues of future work include the introduction of unlinkable credentials, a fixed number of authentication attempts (k-times anonymity), and proof that the zero-knowledge proof is a proof of knowledge, while retaining its zero-knowledge properties against a quantum polynomial-time verifier.

Bibliography

[1] David Chaum. Security without identification: Transaction systems to make Big Brother obsolete. *Communications of the ACM*, 28(10):1030–1044, 1985.

[2] Ivan Bjerre Damgård. Payment systems and credential mechanisms with provable security against abuse by individuals. In *Conference on the Theory and Application of Cryptography*, pages 328–335. Springer, 1988.

[3] Lidong Chen. Access with pseudonyms. In *International Conference on Cryptography: Policy and Algorithms*, pages 232–243. Springer, 1995.

[4] Anna Lysyanskaya, Ronald L Rivest, Amit Sahai, and Stefan Wolf. Pseudonym systems. In *International Workshop on Selected Areas in Cryptography*, pages 184–199. Springer, 1999.

[5] Stefan Brands. *Rethinking public key infrastructures and digital certificates: Building in privacy*. MIT Press, 2000.

[6] Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In Birgit Pfitzmann, editor, *Advances in Cryptology — EUROCRYPT 2001*, pages 93–118, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[7] Jan Camenisch and Anna Lysyanskaya. A signature scheme with efficient protocols. In Stelvio Cimato, Giuseppe Persiano, and Clemente Galdi, editors, *Security in Communication Networks*, pages 268–289, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[8] Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In Matt Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, pages 56–72, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[9] Johannes Blömer and Jan Bobolz. Delegatable attribute-based anonymous credentials from dynamically malleable signatures. In *International Conference on Applied Cryptography and Network Security*, pages 221–239. Springer, 2018.

[10] Johannes Blömer, Jan Bobolz, Denis Diemert, and Fabian Eidens. Updatable anonymous credentials and applications to incentive systems. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 1671–1685, 2019.

[11] David Pointcheval and Olivier Sanders. Short randomizable signatures. In Kazue Sako, editor, *Topics in Cryptology - CT-RSA 2016*, pages 111–126, Cham, 2016. Springer International Publishing.

[12] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Scalable, transparent, and post-quantum secure computational integrity. Cryptology ePrint Archive, Report 2018/046, 2018. https://ia.cr/2018/046.

[13] Eli Ben-Sasson, Alessandro Chiesa, Michael Riabzev, Nicholas Spooner, Madars Virza, and Nicholas P Ward. Aurora: Transparent succinct arguments for r1cs. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 103–128. Springer, 2019.

[14] Melissa Chase, Markulf Kohlweiss, Anna Lysyanskaya, and Sarah Meiklejohn. Malleable signatures: Complex unary transformations and delegatable anonymous credentials. Cryptology ePrint Archive, Report 2013/179, 2013. https://eprint.iacr.org/2013/179.

[15] Melissa Chase, David Derler, Steven Goldfeder, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, Daniel Slamanig, and Greg Zaverucha. Post-quantum zero-knowledge and signatures from symmetric-key primitives. In *Proceedings of the 2017*

ACM SIGSAC Conference on Computer and Communications Security, pages 1825–1842, 2017.

[16] Irene Giacomelli, Jesper Madsen, and Claudio Orlandi. Zkboo: Faster zero-knowledge for boolean circuits. In *25th {usenix} security symposium ({usenix} security 16)*, pages 1069–1083, 2016.

[17] Scott Ames, Carmit Hazay, Yuval Ishai, and Muthuramakrishnan Venkitasubramaniam. Ligero: Lightweight sublinear arguments without a trusted setup. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 2087–2104, 2017.

[18] Vadim Lyubashevsky, Daniele Micciancio, Chris Peikert, and Alon Rosen. Swifft: A modest proposal for fft hashing. In *Fast Software Encryption (FSE)*, pages 54–72. Springer, 2008.

[19] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. *Journal of the ACM*, 60:1–35, 2013.

[20] Chris Peikert. Lattice cryptography for the internet. In Michele Mosca, editor, *Post-Quantum Cryptography*, pages 197–219, Cham, 2014. Springer International Publishing.

[21] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6), September 2009.

[22] Avrim Blum, Merrick Furst, Michael Kearns, and Richard J Lipton. Cryptographic primitives based on hard learning problems. In *Annual International Cryptology Conference*, pages 278–291. Springer, 1993.

[23] Richard Lindner and Chris Peikert. Better key sizes (and attacks) for LWE-based encryption. In *Cryptographers' Track at the RSA Conference*, pages 319–339. Springer, 2011.

[24] Joppe Bos, Craig Costello, Léo Ducas, Ilya Mironov, Michael Naehrig, Valeria Nikolaenko, Ananth Raghunathan, and Douglas Stebila. Frodo: Take off the ring! Practical, quantum-secure key exchange from LWE. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 1006–1018, 2016.

[25] Damien Stehlé, Ron Steinfeld, Keisuke Tanaka, and Keita Xagawa. Efficient public key encryption based on ideal lattices. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 617–635. Springer, 2009.

[26] Joppe W Bos, Craig Costello, Michael Naehrig, and Douglas Stebila. Post-quantum key exchange for the TLS protocol from the ring learning with errors problem. In *2015 IEEE Symposium on Security and Privacy*, pages 553–570. IEEE, 2015.

[27] Oscar Reparaz, Ruan de Clercq, Sujoy Roy Sinha, Frederik Vercauteren, and Ingrid Verbauwhede. Additively Homomorphic Ring-LWE Masking. In *Post-Quantum Cryptography*, pages 233–244. Springer, 2016.

[28] Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. Crystals-kyber: A cca-secure module-lattice-based kem. In *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 353–367. IEEE, 2018.

[29] Jan-Pieter D'Anvers, Angshuman Karmakar, Sujoy Sinha Roy, and Frederik Vercauteren. Saber: Module-lwr based key exchange, cpa-secure encryption and cca-secure kem. In *International Conference on Cryptology in Africa*, pages 282–305. Springer, 2018.

[30] Newhope. https://newhopecrypto.org/index.shtml. Accessed: 2021-12-16.

[31] Christopher Peikert. Lattice cryptography for the internet. In *International Workshop on Post-Quantum Cryptography*, pages 197–219. Springer, 2014.

[32] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange: A new hope. In *Proceedings of the 25th USENIX Conference on Security Symposium*, SEC'16, page 327–343, USA, 2016. USENIX Association.
[33] Rafael Pass. On deniability in the common reference string and random oracle model. In *Annual International Cryptology Conference*, pages 316–337. Springer, 2003.
[34] Kyber. https://pq-crystals.org/kyber/index.shtml.
[35] Dilithium. https://pq-crystals.org/dilithium/index.shtml.
[36] Falcon. https://falcon-sign.info/.
[37] Nistpqcsubmission. https://csrc.nist.gov/Projects/post-quantum-cryptography/round-3-submissions.

A Modified Zero-Knowledge Interactive Proof

Common Input: $y \in L_f$.

1. Prover generates a public/private key pair $(pk, sk) \leftarrow \texttt{Keygen}(1^\lambda)$
2. Prover sends $(pk, Enc_{pk}(x))$ to the verifier.
3. Verifier computes for each $i \in [k]$, where $n \in O(poly(|y|))$:
   - $b_i \xleftarrow{\$} \{0,1\}$
   - $s_i \xleftarrow{\$} \{0,1\}^{|y|}$
   - Verifier computes ciphertexts

$$c_i = \begin{cases} Enc_{pk}(s_i) & \text{if } b_i = 0 \\ Enc_{pk}(x) + Enc_{pk}(s_i) & \text{if } b_i = 1 \end{cases}$$

   - Verifier sends $(c_1, \ldots, c_k)$ to the prover.
4. Prover computes the following. The prover is now in state Q.
   - $m_i = Dec_{sk}(c_i)$ for each $i \in [k]$.
   - $f(m_1), \ldots, f(m_k)$.
   - Prover sends $((m_1 + t_1, f(m_1)), \ldots, (m_k + t_k, f(m_k)))$ to the verifier.
5. Verifier accepts the proof if and only if for all $i \in [k]$.

$$\Big(\big(b_i = 0 \wedge f(m_i) = f(s_i)\big) \vee \big(b_i = 1 \wedge y = f(m_i) \oplus f(s_i)\big) \Big) = 1$$

Algorithm 6: Modified computational ZKIP for the language L_f

Remark

Note strings $m_1 + t_1, \ldots, m_k + t_k$ are distributed uniformly therefore do not provide any information to a verifier. Therefore, this proof system is equivalent to the proof system described in Algorithm 1.