Security and **Privacy** in Post-Quantum World

Dr. Veronika Kuchta
Research Fellow
Monash University
Australia

Outline

Post-Quantum Cryptography

- Motivation for lattice-based cryptography
- Lattice-Based Ring CT
- Lattice-Based Zero-Knowledge Proofs

Quantum Random Oracle Security Proof

Motivation for Lattice-Based Cryptography

Post-Quantum Cryptography

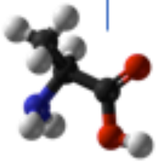
Development in quantum computing

copyrightTamiko Thiel
Source: Wikiquote

1982 — Idea by Richard Feynman proposed

......

1998 — First 2-qubit quantum computer realized



Molecule of alanine used in NMR implementation of quantum computing.

2000 — 7-qubit quantum computer realized

Science News — from research organizations

Los Alamos Scientists Shed New Light On Quantum Computation

Date: January 5, 2001
Source: Los Alamos National Lab
Summary: Scientists at the Department of Energy's Los Alamos National Laboratory and the University of Queensland's Centre for Quantum Computer Technology in Australia have made an advance in the quest for a functional quantum computer by exploiting currently existing technology in a novel and unexpected way. A functional quantum computer could solve certain large mathematical problems and crack secret codes at speeds faster than today's fastest supercomputers.
Share:

2006 — 12-qubit quantum computer realized

2011 — 14-qubit quantum computer realized

Phys Rev Lett. 2011 Apr 1;106(13):130506. Epub 2011 Mar 31.

14-Qubit entanglement: creation and coherence.

Monz T, Schindler P, Barreiro JT, Chwalla M, Nigg D, Coish WA, Harlander M, Hänsel W, Hennrich M, Blatt R.

Author information

Abstract
We report the creation of Greenberger-Horne-Zeilinger states with up to 14 qubits. By investigating the coherence of up to 8 ions ov
we observe a decay proportional to the square of the number of qubits. The observed decay agrees with a theoretical model which a
a system affected by correlated, Gaussian phase noise. This model holds for the majority of current experimental systems develope
quantum computation and quantum metrology.

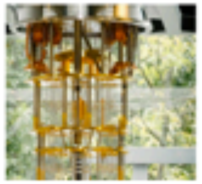
© 2011 American Physical Society

PMID: 21517367 DOI: 10.1103/PhysRevLett.106.130506

2017 — 49/50-qubit quantum computer realized

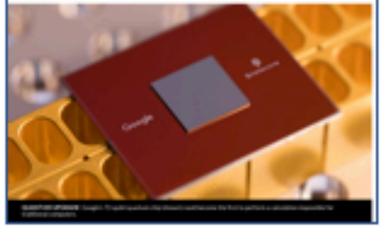
MIT Technology Review

IBM Raises the Bar with a 50-Qubit Quantum Computer



2018 — 72-qubit quantum computer realized

Google moves toward quantum supremacy with 72-qubit computer



2019

NewScientist

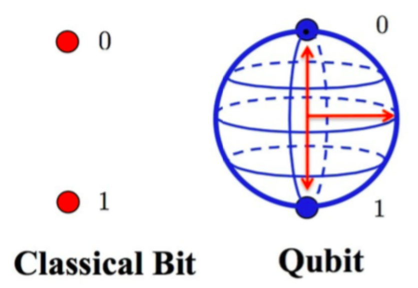
IBM unveils its first commercial quantum computer

Post-Quantum Cryptography

- Once a quantum computer (QC) will be available for the daily use, it will break RSA
- Quantum supremacy (defined by US scientist John Preskill) = ability of QC to perform computations faster than classical computers.
- NIST (US) initiated PQC standardization process to solicit, evaluate and standardize one or more quantum-resistant public-key cryptosystems:
- How do we secure our internet data (stored, transmitted via the Internet)?
- There are several post-quantum candidates which look into this question:
    - Lattice-based cryptography
    - Code-based cryptography
    - Symmetric primitives
    - Isogeny-based cryptography
    - Multi-variate cryptography

Post-Quantum Cryptography

Post-Q.	Security	Efficiency	Compactness	Applications
Lattice-based	**High** Worse-case	**High** Signing + Verification +	**Medium** Signature Size + Pub-Key Size +	**High** BLISS, FHE…
Code-based	**High**	**Medium** Signing - Verification +	**Medium** Signature Size + Pub-Key Size -	Low None.
Multivariate-based	**High**	**High** Signing + Verification +	**Medium** Signature Size + Pub-Key Size -	**Medium** Only DS: Rainbow.
Hash-based	**High**	Low Signing - Verification -	**High** Signature Size + Pub-Key Size +	Low None.
Isogeny-based	**High**	Low Signing - Verification -	**Medium** Signature Size - Pub-Key Size +	Low None.



Lattice-Based Cryptography

Motivation: Efficiency

Popular cryptosystems are relatively inefficient;

For security level 2^n :

RSA -- key length $O(n^3)$, computation $O(n^6)$.
ECC -- key length $O(n)$, computation $O(n^2)$.

Structured (`Ring based') Lattices -- key length and computation $O(n)$ asymptotically, as n grows towards infinity.

In Practice, for typical security parameter $n \approx 100$, with best current schemes, typically have:

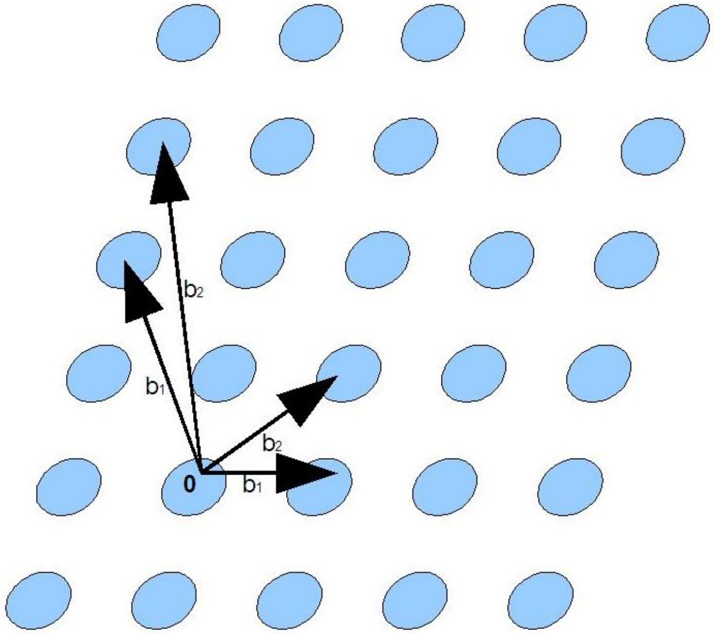
Structured Lattice crypto: Computation ≈ 100 times faster than RSA
Structured Lattice crypto: ciphertext/key length $\approx$ RSA key/ciphertext

Lattice-Based Cryptography

Definition: An n dimensional (full-rank) lattice $L(B)$ is the set of all integer linear combinations of some basis set of linearly independent vectors $\vec{b}_1, \dots \vec{b}_n \in \mathbb{R}^n$:

$$L(B) \coloneqq \{c_1\vec{b}_1 + c_2\vec{b}_2 + \cdots + c_n\vec{b}_n : c_i \in \mathbb{Z}, i = 1, \dots, n\}.$$

Call a $n\times n$ matrix $B = (\vec{b}_1, \dots \vec{b}_n)$ a basis for $L(B)$.



Example: in 2 dimensions, i.e. $n = 2$:

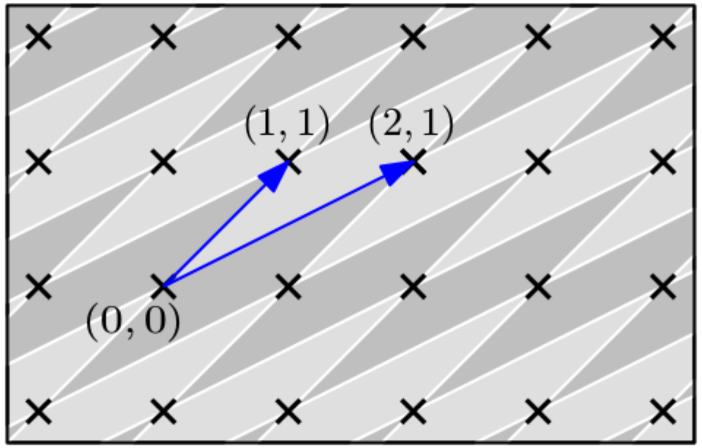
$$\vec{b}_1 = \begin{bmatrix}1\\0\end{bmatrix}, \qquad \vec{b}_2 = \begin{bmatrix}1.2\\1\end{bmatrix}$$

$$\vec{b}_1' = \begin{bmatrix}-0.6\\2\end{bmatrix}, \qquad \vec{b}_2' = \begin{bmatrix}-0.3\\3\end{bmatrix}$$

Lattice-Based Cryptography

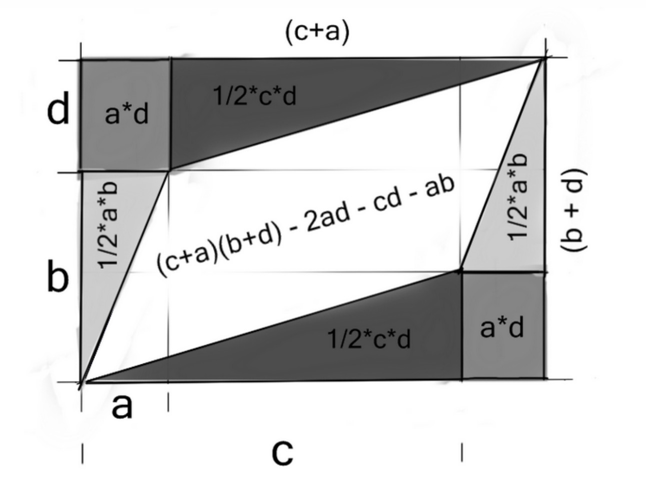
Definition: For an n-dimensional lattice basis $B = (\vec{b}_1, \dots \vec{b}_n) \in \mathbb{R}^{n \times n}$, the fundamental parallelepiped of B, denoted $P(B)$, is the set of all real-valued $[0,1)$-linear combinations of some basis set of linearly independent vectors $(\vec{b}_1, \dots \vec{b}_n) \in \mathbb{R}^n$:

$$P(B) \coloneqq \{c_1\vec{b}_1 + c_2\vec{b}_2 + \cdots + c_n\vec{b}_n \colon 0 \le c_i < 1, i = 1, \dots, n\}$$



For an n-dimensional lattice $L(B)$ the determinant of $L(B)$ is the n-dim. volume of the $P(B)$

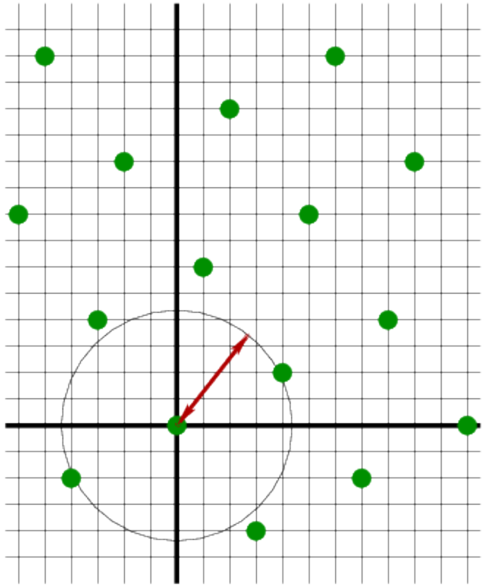
Example: 2-dim $B = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$



Lattice-Based Cryptography

For cryptographic security, need computationally hard lattice problems. Many problems related to geometry of lattices seem to be hard.

The most basic geometric quantity about a lattice is its minimum (aka Minkowski first minimum).



Definition: For an n-dim. lattice L it's minimum $\lambda(L)$ is the length of the shortest non-zero vector of L: $\lambda(L) = \min(\|\vec{b}\| \,:\, \vec{b} \in L \setminus 0)$.

For any n-dim. lattice L holds: $\lambda(L) \leq \sqrt{n} \cdot \det L^{\frac{1}{n}}$.

Lattice-Based Cryptography

Ajtai's Random q-ary perp Lattice: Given an integer q and a uniformly random matrix $A \in \mathbb{Z}_q^{n\times m}$, the q-ary perp lattice $L_q^{\perp}(A) = \{\vec{v} \in \mathbb{Z}^m : A \cdot \vec{v} = \vec{0} \bmod q\}$.

Lattice-based problems.

γ −Shortest Vector Problem (γ-SVP): Given a basis B for n −dim lattice, find $\vec{b} \in L$ such that:

$$0 < \|\vec{b}\| < \gamma \cdot \lambda(L).$$

Small Integer Solution Problem $SIS_{q,m,n,\beta}$**:** Given n and a matrix A sampled uniformly in $\mathbb{Z}_q^{n\times m}$, find $\vec{v} \in \mathbb{Z}^m \setminus \{0\}$ such that $A \cdot \vec{v} = \vec{0} \bmod q$ and $\|\vec{v}\| \leq \beta$

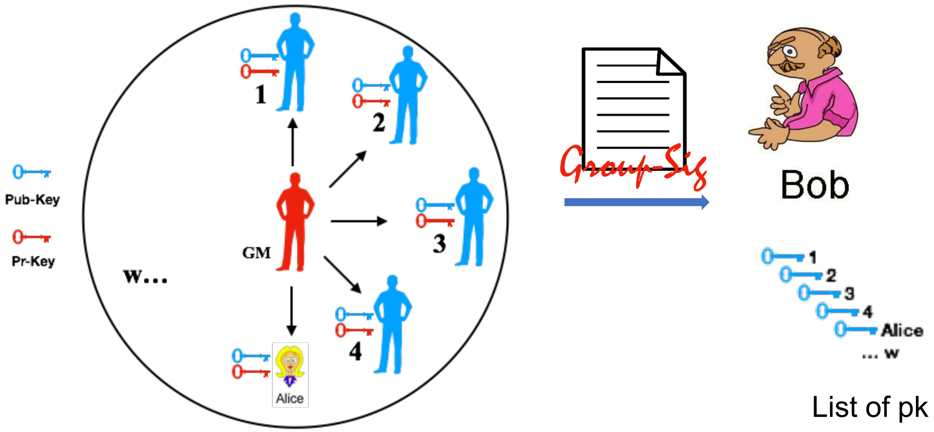
Search-LWE Problem: Given q, n, m, α, a matrix $A \hookleftarrow U(\mathbb{Z}_q^{m\times n})$ and $\vec{y} = A \cdot \vec{s} + \vec{e} \bmod q$ (with $\vec{e} \hookleftarrow \chi_{\alpha q}^m$ and $\vec{s} \hookleftarrow U(\mathbb{Z}_q^n)$, find $\vec{s}$.

Lattice-Based RingCT

Lattice-Based RingCT [ACISP'19]

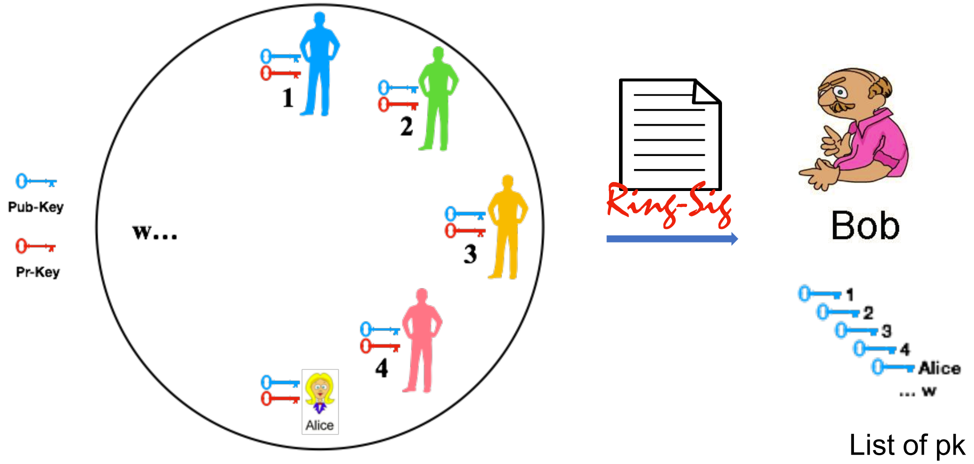
A **group signature** scheme allow a signer (Alice) as a member of a group to anonymously sign a message on behalf of the group with w users.

A group manager (GM) is in charge of establishing pairs of (**public key**, **secret key**) = (pk, sk).



A **ring signature** scheme allow a signer (Alice) to anonymously sign a message on behalf of the group with w users.

No GM is needed.



Lattice-Based RingCT [ACISP'19]

A ring signature has the following properties:

- All the properties of a digital signature,
- *Anonymity*: the identity of Alice cannot be determined,
- *Spontaneity*: any ring of users can be used as a group,
- *non-Linkability*: given two messages and their signatures, no one can tell if the signatures were from the same signer or not,
- *non-Framebility*: no set of users can forge a signature for a non-participating ring member.

Example:

Cryptocurrencies like Bytecoin (BCN) 2012, ShadowCoin,

Monero 2016 (based on Liu's PhD thesis and paper); Ring CT v 1.0 and v 2.0.

Lattice-Based RingCT [ACISP'19]

LRCT Scheme:

- BLISS (Bimodal Lattice Signature Scheme)
- Post-quantum cryptography
- Five polynomial time algorithms

Correctness is satisfied

Version-1: Single-Input Single-Output (SISO) wallets. (ACISP2018)

Version-2: Multiple-Input Multiple-Output (MIMO) wallets. (ACISP2019)

MIMO.LRCT	Description
Setup	Creates the public parameters
KeyGen	Generates the public keys
Mint	Produces the coins
Spend	Transfers input wallets to output wallets
Verify	Verifies transactions

Accounts - Wallets		
	Public "act"	**Private "ask"**
User	Public-Key	Private-Key
Coin	Coin	Coin-key

Input Wallet (*IW*)		
	Public "act"	**Private "ask"**
User	$\mathbf{a}_{(in)}^{(k)}$	$\mathbf{S}_{(in)}^{(k)}$
Coin	$\mathbf{cn}_{(in)}^{(k)}$	$\mathbf{ck}_{(in)}^{(k)}$

Output Wallet (*OW*)		
	Public "act"	**Private "ask"**
User	$\mathbf{a}_{(out)}^{(j)}$	$\mathbf{S}_{(out)}^{(j)}$
Coin	$\mathbf{cn}_{(out)}^{(j)}$	$\mathbf{ck}_{(out)}^{(j)}$

SISO: $k = 1$ *and* $j = 1$

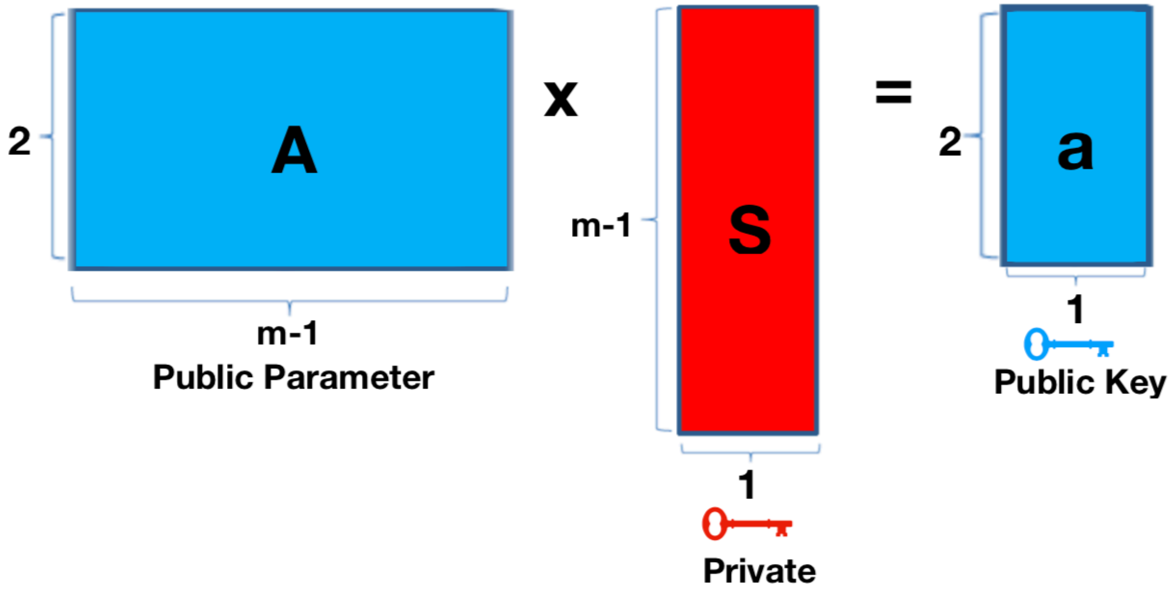
MIMO: $k > 1$ *and* $j > 1$

Lattice-Based RingCT [ACISP'19]

Algorithm 1 MIMO.L2RS.KeyGen - Key-pair Generation $(\mathbf{a}, \mathbf{S})$

Input: Pub-Param: $\mathbf{A} \in \mathcal{R}_q^{2\times(m-1)}$.
Output: $(\mathbf{a}, \mathbf{S})$, being the public-key and the private-key, respectively.

1: **procedure** MIMO.L2RS.KeyGen($\mathbf{A}$)
2: Let $\mathbf{S}^T = (\mathbf{s}_1, \ldots, \mathbf{s}_{m-1}) \in \mathcal{R}_q^{1\times(m-1)}$, where $\mathbf{s}_i \leftarrow (-2^\gamma, 2^\gamma)^n$, for $1 \le i \le m-1$
3: Compute $\mathbf{a} = (\mathbf{a}_1, \mathbf{a}_2)^T = \mathbf{A} \cdot \mathbf{S} \bmod q \in \mathcal{R}_q^2$.
4: **return** $(\mathbf{a}, \mathbf{S})$.



Lattice-Based RingCT [ACISP'19]

Algorithm 4 MIMO.LRCT.Mint

Input: $(\mathbf{A} \in \mathcal{R}_q^{2\times(m-1)}, \$ \in [0, 2^{\ell_\$ - 1}])$, being the public parameter $\mathbf{A}$ and the amount $\$$.

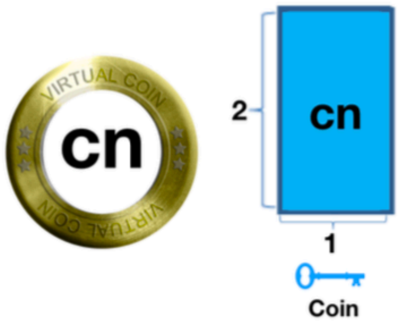
Output: $(\mathbf{cn}, \mathbf{ck})$, where they are the coin and the coin key, respectively.

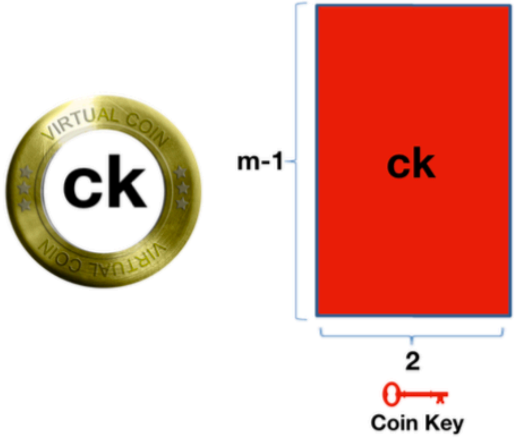
procedure MIMO.LRCT.MINT($\mathbf{A}, \$$)

  Let $\mathbf{ck}^T = (\mathbf{ck}_1, \ldots, \mathbf{ck}_{m-1}) \in \mathcal{R}_q^{1\times(m-1)}$ with $\mathbf{ck}_i \leftarrow (-2^\gamma, 2^\gamma)^n$, for $1 \le i \le m-1$

  $\mathbf{cn} = \mathsf{Com}_{\mathbf{A}}(\$, \mathbf{ck}) = \mathbf{A} \cdot \mathbf{ck} + \overline{\$} \bmod q \in \mathcal{R}_q^2$ with $\overline{\$} = (0, \$)^T \in \mathcal{R}_q^{1\times 2}$

  return $(\mathbf{cn}, \mathbf{ck})$





Lattice-Based RingCT [ACISP'19]

MIMO. LRCT. Spend protocol


Alice

1. Determines the amount $\mathbf{\$}_{in}$ to spend: $N_{in}\ of\ IW$
2. Determines the **Bob**'s wallets $N_{out}\ of\ OW$, using **Bob**'s *pk*
3. Proves balance, $\sum \mathbf{\$}_{in} = \sum \mathbf{\$}_{out}$ → amount preservation
4. Verifies $\mathbf{\$}_{out}$ → range preservation (PoK_{Range})
5. Securely sends $\mathbf{ck}_{out}$ and $\mathbf{\$}_{out}$ to **Bob**
6. Creates the List of the Ring Signature → adding $N_{in}\ of\ IW$
7. Signs the transaction *TX* with **(sk,ck)** → SigGen (PoK)
8. Sets $TX = \{\mu, IW, OW\}$, $Sig = \{PoK, PoK_{Range}\}$
9. Outputs TX, Sig and *Linking Tags*


Bob

Lattice-Based RingCT [ACISP'19]

Range Proof

Alice

$\mathbf{cn}_{in} = A \cdot \mathbf{ck} + 10$ ➡

Spend: $\mathbf{cn}_{out-1} = A \cdot \mathbf{ck} + 5 = \mathbf{Com}_A(5, \mathbf{ck})$

Change: $\mathbf{cn}_{out-2} = A \cdot \mathbf{ck} + 5 = \mathbf{Com}_A(5, \mathbf{ck})$

- Proves balance, $\sum \$_{in} = \sum \$_{out} \rightarrow$ amount preservation

$$\mathbf{cn}_{in} - (\mathbf{cn}_{out-1} + \mathbf{cn}_{out-2}) = \mathbf{Com}_A(0, \mathbf{ck})$$

Bob

$\mathbf{cn}_{in} = A \cdot \mathbf{ck} + 10$ ➡

Spend: $\mathbf{cn}'_{out-1} = A \cdot \mathbf{ck} + 11 = \mathbf{Com}_A(11, \mathbf{ck})$

Change: $\mathbf{cn}'_{out-2} = A \cdot \mathbf{ck} - 1 = \mathbf{Com}_A(-1, \mathbf{ck})$

- Proves balance, $\sum \$_{in} = \sum \$_{out} \rightarrow$ amount preservation

$$\mathbf{cn}_{in} - (\mathbf{cn}'_{out-1} + \mathbf{cn}'_{out-2}) = \mathbf{Com}_A(0, \mathbf{ck})$$

Lattice-Based
Zero-Knowledge Proofs

Lattice-Based Zero-Knowledge Proofs

Background: Schnorr Protocol

Prover		Verifier
$(x = \log_g h)$		
$u \in_R \mathbb{Z}_n$ $a \leftarrow g^u$	$\xrightarrow{\quad a \quad}$	
		$c \in_R \{0,1\}$
	$\xleftarrow{\quad c \quad}$	
$r \leftarrow_n \begin{cases} u, & \text{if } c = 0 \\ u + x, & \text{if } c = 1 \end{cases}$	$\xrightarrow{\quad r \quad}$	$g^r \stackrel{?}{=} \begin{cases} a, & \text{if } c = 0 \\ ah, & \text{if } c = 1 \end{cases}$

ZKP is useful tool for proving something about a secret is true while minimizing leakage of information on the secret ([GMR85]).

ZKP has been extensively investigated and generalized to cover almost any imaginable scenario! For instance, how to prove in ZK that:

- **Anonymous authentication**: I know a secret key that corresponds to one of N public keys of a group, without identifying which key.
- **Anonymous credentials**: I know a signature from an authority on my driver's license (containing my name, address, age,...) but I just want to prove to an alcohol merchant that I am over 18, without leaking who I am.

To handle such general situations, need to generalize definition (and construction!) of ZK.

Lattice-Based Zero-Knowledge Proofs

Generalizing the definition of ZK to any relation R:

- Let $R = \{(v; w)\} \subseteq V \times W$ be a relation (e.g. $R = \{(v = (g, h); w = x): h = g^x\}$ in Schnorr).
- Let $v \in V$ be the common public input to P and V (e.g. $h \in < g >$ in Schnorr).
- Let $w \in W$ be a witness private input to P (e.g. x such that $h = g^x$ in Schnorr).
- Let L_R be language corresponding to R, i.e. set of $v \in V$ for which there exists a witness $w \in W$ with $(v; w) \in R$. (e.g. set $< g >$ in Schnorr)

Goal: For a given relation R and v, prove in ZK that I know a witness w such that $(v; w) \in R$.

Lattice-Based Zero-Knowledge Proofs

General definition of Zero-Knowledge Proof to any relation R.

Completeness: If P and V follow protocol, V's test will always pass.

Soundness: There exists an efficient (probabilistic polynomial time) algorithm (witness extractor) that given any malicious prover P^* that passes with non-negligible probability the honest verifier's test on input v, can extract a witness w such that $(v; w) \in R$.

Zero-Knowledge: The exists an efficient (expected polynomial time) algorithm (simulator) that given any malicious verifier V^*, can simulate protocol messages received by V^* from P on input v with a computationally indistinguishable distribution.

Definition (Commitment Scheme): The formal definition of a commitment scheme is given as follows. A commitment scheme consists of the following three algorithms:

$KeyGen$: is a probabilistic polynomial-time (PPT) algorithm that outputs a commitment key ck and a definition of message space $\mathcal{M}_{ck}$.

Com: is a PPT algorithm that on input the commitment key ck and a message $\mu \in \mathcal{M}_{ck}$ outputs values C, r, where C is the commitment on μ and $r \in \mathcal{R}_{ck}$ is the corresponding randomness sampled from randomness space $\mathcal{R}_{ck}$.

$Open$: is a deterministic algorithm that on input ck, a message μ and values C, r opens the commitment to the value μ.

Homomorphic commitment: A homomorphic commitment scheme is a non-interactive commitment scheme such that the following property holds:

$$Com_{ck}(a, r_a) + Com_{ck}(b, r_b) = Com_{ck}(a + b, r_a + r_b)$$
$$\zeta \cdot Com_{ck}(a, r_a) = Com_{ck}(\zeta a, \zeta r_a)$$

Lattice-Based Zero-Knowledge Proof for Integer Relations (Designs, Codes and Cryptography, (to appear))

Definition (Challenge Space): Let $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^n+1)$ for $q \in \mathbb{Z}^+$. Let $HW(f)$ denote the Hamming weight of the elements $f \in Z[X]$ and $p \leq q/2$ then the challenge space $\mathcal{CH}^n_{\omega,p}$ is defined as follows:

$$\mathcal{CH}^n_{\omega,p} = \{f \in \mathbb{Z}[X] : \deg(f) = n-1 \wedge HW(f) = \omega \wedge \|f\|_\infty = p\}, \qquad and \quad \Delta\mathcal{CH}^n_{\omega,p} = \mathcal{CH}^n_{\omega,p} - \mathcal{CH}^n_{\omega,p}$$

Lattice-Based Commitment

If the M-LWE problem is hard then the commitment scheme is computationally hiding.

If M-SIS problem is hard, then our commitments scheme is computationally binding with respect to the relaxation factor d.

$\texttt{KeyGen}$: Create $(\mathbf{A}_1, \mathbf{A}_2) \in \mathcal{R}_q^{\nu \times m} \times \mathcal{R}_q^{n' \times m}$. Public parameters are:

$$\mathbf{A}_1 = [\mathbf{I}_\nu \| \mathbf{A}'_1], \quad \text{where} \quad \mathbf{A}'_1 \leftarrow_\$ \mathcal{R}_q^{\nu \times (m-\nu)}$$
$$\mathbf{A}_2 = [\mathbf{0}^{n' \times \nu} \| \mathbf{I}_{n'} \| \mathbf{A}'_2], \quad \text{where} \quad \mathbf{A}'_2 \leftarrow_\$ \mathcal{R}_q^{n' \times (m-\nu-n')}$$

Set the commitment key $ck = \mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix}$, which is used to commit to $\mathbf{x} \in \mathcal{R}_q^{n'}$.

$\texttt{Com}$: To commit to a message $\mathbf{x} \in \mathcal{R}_q^{n'}$, choose a random polynomial vector $\mathbf{r} \leftarrow_\$ \mathcal{U}(\{-\mathcal{B}, \ldots, \mathcal{B}\}^{mn})$ and output the commitment

$$\mathsf{C} := \texttt{Com}_{ck}(\mathbf{x}, \mathbf{r}) = \mathbf{A} \cdot \mathbf{r} + \mathbf{x} = \mathbf{A} \cdot \mathbf{r} + \texttt{enc}(\mathbf{x}), \text{ where } \texttt{enc}(\mathbf{x}) = \begin{bmatrix} \mathbf{0}^\nu \\ \mathbf{x} \end{bmatrix} \in \mathcal{R}_q^{\nu+n'}.$$

$\texttt{ROpen}$: A valid opening of a commitment C is a tuple consisting of $\mathbf{x} \in \mathcal{R}_q^{n'}$, $\mathbf{r} \in \mathcal{R}_q^m$ and $\mathsf{d} \in \Delta\mathcal{CH}^n_{\omega,p}$. The verifier checks that $\mathsf{d} \cdot \mathsf{C} = \mathbf{A} \cdot \mathbf{r} + \mathsf{d} \cdot \texttt{enc}(\mathbf{x})$, and that $\|\mathbf{r}\| \leq \beta$. Otherwise return 0.

Lattice-Based Zero-Knowledge Proof for Integer Relations

Constructions

1. Integer addition ZK protocol: Prove knowledge of $X, Y, Z \in \mathbb{Z}$ such that $X + Y = Z \in \mathbb{Z}$
2. Polynomial multiplication ZK protocol: Prove knowledge of polynomials $\mathcal{X}, \mathcal{Y}, \mathcal{Z} \in \mathcal{R}_q$ such that $\mathcal{X} \cdot \mathcal{Y} = \mathcal{Z}$.
3. Integer multiplication ZK protocol: Prove knowledge of integers $X, Y, Z \in \mathbb{Z}$ such that $X \cdot Y = Z$

Techniques:

- *One-shot proof:* The shortness of the extracted witness is one of the main challenges in lattice-based zero-knowledge proofs and arguments of knowledge.
  - Since most of the extraction techniques use multiplication by the inverse of challenge differences, this can be challenging when we deal with lattice-based proofs.
  - Solution: introduction of relaxed arguments of knowledge.
  - -> solving a system of equations of the form $V \cdot \vec{c} = \vec{y}$, where V is a Vandermonde matrix, and the entries of this matrix are the different powers of challenges.
  - The one-shot proof in CRYPTO'19 uses adjugate matrices instead of Vandermonde. → Use a challenge space with large challenges

Lattice-Based Zero-Knowledge Proof for Integer Relations

Techniques (cont.):

- For integer addition protocol: Motivated by [CRYPTO'18].
  - [CRYPTO'18] provides efficient integer relations protocol for integers of length $L \leq 2^{13}$.
  - However, for smaller integers, i.e. $L \in [2^4, 2^8]$ the [CRYPTO'18] approach can be outperformed by our protocol.
  - We use a chunking technique, applying on integers of length L and then perform the classical addition/multiplication algorithm on each chunk.

Parameter	Set 1	Set 2	[22]	Set 3	Set 4	[22]	Set 5	Set 6	[22]
Modulus q	2^{34}	2^{34}	2^{34}	2^{34}	2^{34}	2^{34}	2^{36}	2^{36}	2^{36}
Ring dim. n	2^7	2^7	2^7	2^8	2^8	2^8	2^9	2^9	2^9
L (Int. length)	2^5	2^5	2^5	2^6	2^6	2^6	2^7	2^7	2^7
$\widetilde{m} = \mathcal{O}(n)$	896	896	N/A	1024	1024	N/A	1024	1024	N/A
$\mathcal{B}_{IA}$	280	280	N/A	73	73	N/A	157	157	N/A
$\log(\beta'_{IA})$	≈ 33.12	≈ 26.62	N/A	≈ 31.27	≈ 24.78	N/A	≈ 32.38	≈ 25.9	N/A
Nr. of chunks k	4	8	1	4	16	1	16	32	1
Nr. of repet. t	1	1	≈ 137	1	1	1	1	1	137
Proof size	195.89KB	189KB	1.8MB	1.02MB	846.67MB	3.57MB	2.09MB	1.75MB	6.23MB

← Integer Addition Protocol

Integer Multiplication Protocol

↓

Parameter	Set 1	Set 2	[22]	Set 3	Set 4	[22]	Set 5	Set 6	[22]
Modulus q	2^{34}	2^{34}	2^{34}	2^{34}	2^{34}	2^{34}	2^{30}	2^{30}	2^{30}
Ring dim. n	2^7	2^7	2^7	2^8	2^8	2^8	2^9	2^9	2^9
L (Int. length)	2^5	2^5	2^5	2^6	2^6	2^6	2^7	2^7	2^7
$\widetilde{m}$	896	896	N/A	1024	1024	N/A	1024	1024	N/A
$\mathcal{B}_{IM}$	280	280	N/A	73	73	N/A	16	16	N/A
$\log(\beta_{IM})$	33.12	26.94	N/A	31.5	25.01	N/A	29.10	26.88	N/A
Nr. of repet. t	1	1	≈ 137	1	1	1	1	1	137
Nr. of chunks k	8	16	1	16	64	1	32	64	1
Proof size	255.27KB	239.96KB	2.8MB	848.97KB	704.55KB	5.66MB	2.14MB	1.76MB	9.08MB

Quantum Random Oracle Security Proof

Quantum Random Oracle Model [EUROCRYPT'20]

- Fujisaki-Okamoto (FO) transform for CPA → CCA security
  - Commonly used to strengthen CPA → CCA security for pub-key encryption
    - Start from a CPA secure pub key encryption scheme E
    - Get a CCA secure pub key enc scheme $E' = FO(E)$
  - Used by most NIST PQC pub-key encryption scheme candidates
  - We focus on the $FO^{\not\vdash} = U^{\not\vdash\circlearrowleft} \circ T$ (E) variant
    - using two hash functions (H, H'), modelled as Random Oracles
    - Focus on hash H used by U:
      - $c = Enc(m; H'(m))$, encapsulated key $K = H(m, c)$
  - Assume two (mild) properties on the CPA pub-key encryption scheme:
    - Det. Scheme $T(E)$ is η – injective for sufficiently negligible η
    - CPA scheme E has sufficiently negligible decryption failure probability δ

Quantum Random Oracle Model [EUROCRYPT'20]

- Security proofs in the Quantum Random Oracle Model (QROM)
  - Model hash functions used in FO transform as random oracles (q attack queries)

  - Quantum accessible random oracle O, modelled as a unitary map U_O:
    - $U_O|x\rangle|y\rangle \longmapsto |x\rangle|y \oplus O(x)\rangle$

  - Model QROM quantum attacker $\mathcal{A}^{|O\rangle}$ as a sequence of attack unitaries $\mathcal{A}_i$ interleaved with oracle queries to U_O, followed by a final measurement $\mathbb{M}$ to produce output:
    - $\mathcal{A}^{|O\rangle} := \mathbb{M} \circ \mathcal{A}_N \circ U_O \circ \mathcal{A}_{N-1} \circ U_O \circ \cdots U_O \circ \mathcal{A}_1$
      - $\mathcal{A}_i$ outputs i'th query to O

- **Prior FO QROM Security Proofs** (w/o strong "DS" properties): **square root adv. Loss**
  - $\boldsymbol{Adv(CCA) \leq \sqrt{q \cdot Adv(CPA)}}$ **(simplified)**
- **Our result (with FFC/injectivity properties):**

$$\boxed{\boldsymbol{Adv(CCA) \leq q^2 \cdot Adv(CPA)} \text{ (simplified) (no sq-root adv loss)}}$$

Background: One-Way To Hiding (OWTH) Lemma

- **Core tool in QROM CCA proofs: One-Way to Hiding (OWTH) Lemma [U14]**
  - Recall - FO use of H: $x^* \leftarrow \$$, $z = Enc_{pk}(x^*)$, encaps. key $K = H(x^*, c)$
  - Classical ROM argument: if $\mathcal{A}(pk, c)$ can distinguish K from random, $\mathcal{A}$ must query H at (x^*, c).

    →proof reduction can extract x from $\mathcal{A}$'s queries to H → break one-wayness of Enc.

  - OWTH [U14]: QROM variant of above
    - **Goal of $\mathcal{A}$:** Distinguish whether $O = H$ or $O = G$ -- G differs from H only at x^*
    - $x^*, y_H, y_G \leftarrow \$$ // $H(x^*) := y_H$, $G(x^*) \coloneqq y_G$
    - $\mathrm{Adv}_{\mathrm{OWTH}}(\mathcal{A}) := |\Pr[1 \leftarrow \mathcal{A}^{|H\rangle}(z^* = Enc(x^*), y_H, y_G)] - \Pr[1 \leftarrow \mathcal{A}^{|G\rangle}(z^* = Enc(x^*), y_H, y_G)]|$
    - **Goal of OWTH extractor algorithm $B^{|O'\rangle}$**: Given $z^* = Enc(x^*)$, use $\mathcal{A}$ to efficiently extract x^*
    - $Adv_{OW}(B) := \Pr[x^* \leftarrow B^{|O'\rangle}(z^* = Enc(x^*), y_H, y_G)]$
  - Original B strategy [U14], $|O'\rangle$= $|H\rangle$ ("single sided"): query-based extraction → measure a random query of A
    - [U14] OWTH bound: $Adv_{OWTH}(\mathcal{A}) \leq 2q \cdot \sqrt{Adv_{OW}(B)}$ -- square-root loss!
    - **Subsequent work [AHU18], [BH+19 - $|O'\rangle$=$|G\rangle$ and $|H\rangle$ ("double sided")]**: Improve on "random query", but still query-based extraction
    - [BH+19] bound: $Adv_{OWTH}(\mathcal{A}) \leq 2 \cdot \sqrt{Adv_{OW}(B)}$ -- **square-root loss remains**!

Background: One-Way To Hiding (OWTH) Lemma

- Q: Square-root loss in **query-based extraction** unavoidable?
- A: [PQCrypto'19] Impossibility Result -- **Yes**!
- **Main observation of [PQCrypto'19]**– quantum origin of square-root loss:
  - For q=1 query to O, **there exists** a quantum distinguisher A with
    - $Adv_{OWTH}(\mathcal{A}) = \sqrt{2 \cdot Adv_{OW}(B)}$, where B is the query-based extractor that measures $\mathcal{A}$'s query.

→Impossible to remove OWTH square-root loss with a **query-based extractor**

- **Our observation:** But, the above distinguisher suggests an alternative extraction method that can circumvent the square-root loss:
  - use a **measurement-based extractor**
    - **Extract knowledge of x^* from A's measurement,**
    - **rather than only from A's queries!**

Background: One-Way To Hiding (OWTH) Lemma

- How does the "square-root advantage" distinguisher work?
  - $\mathcal{A}$ makes a quantum query to O:
    - $\sum_{x'} \sqrt{p_{x'}}\,|x'\rangle|0\rangle = \sqrt{p_{x^*}}|x^*\rangle|0\rangle + \sum_{x' \neq x^*} \sqrt{p_{x'}}\,|x'\rangle|0\rangle$ → $\mathrm{Adv}_{\mathrm{OW}}(\mathrm{B}) = p_{x^*}$ (assume <<1).
  - The response $|\psi^O\rangle$ from O is either
    - $\mapsto |\psi^H\rangle := \sqrt{p_{x^*}}|x^*\rangle|y_H\rangle + \sqrt{1-p_{x^*}}\sum_{x' \neq x^*} \frac{\sqrt{p_{x'}}}{\sqrt{1-p_{x^*}}}|x'\rangle|H(x')\rangle$ **if $\boldsymbol{O = H}$**
    - $\mapsto |\psi^G\rangle := \sqrt{p_{x^*}}|x^*\rangle|y_G\rangle + \sqrt{1-p_{x^*}}\sum_{x' \neq x^*} \frac{\sqrt{p_{x'}}}{\sqrt{1-p_{x^*}}}|x'\rangle|H(x')\rangle$ **if $\boldsymbol{O = G}$**
  - To distinguish whether $|\psi^O\rangle$ is $|\psi^H\rangle$ or $|\psi^G\rangle$:
    - $\mathcal{A}$ makes a projective measurement on $|\psi^O\rangle$: $\mathbb{M}_v$ w.r.t. a **measurement vector** $|v\rangle$
    - $|v\rangle$:= vector in span($|\psi^H\rangle, |\psi^G\rangle$) at an angle of $\approx \frac{\pi}{4}$ from $|\psi^H\rangle$
    - $\mathbb{M}_v$ returns 1 with prob. $p^O := \|\text{proj. of } |\psi^O\rangle \text{ along } |v\rangle\|^2$

Our Idea: Measurement-Based Extraction

- Summary - our measurement-based extraction idea (assume 1 oracle query, optimal distinguisher $\mathcal{A}$) -- algorithm C:
  1. Run $\mathcal{A}_1^{|G\rangle}(z^* = Enc(x^*), y_H, y_G)$ to output oracle query
  2. Process the query with the oracle $U_{|G\rangle}$ // state → $|\psi^G\rangle$
  3. Let $\mathcal{A}$ perform its proj. meas. w.r.t. $|v\rangle$ // state → $|v\rangle$ with prob. $p_3 \approx \|proj_v(|\psi^G\rangle)\|^2 \approx \frac{1}{2}$
  4. Measure the input reg. and ret. result // state →$|x^*\rangle|\cdot\rangle$ with prob. $p_4 \approx \|proj_\delta(|v\rangle)\|^2 \approx \frac{1}{2}$

  Overall extraction success probability := $Adv_{OW}(C) = p_3 \cdot p_4 \approx \frac{1}{4}$

Our Idea: Measurement-Based Extraction

- Summary - our measurement-based extraction idea
  - (assume 1 oracle query, optimal distinguisher $\mathcal{A}$) -- algorithm C:

1. Run $\mathcal{A}_1^{|G\rangle}(z^* = Enc(x^*), y_H, y_G)$ to output oracle query
2. Process the query with the oracle $U_{|G\rangle}$ // state → $|\psi^G\rangle$
3. Let $\mathcal{A}$ perform its proj. meas. wrt $|v\rangle$ // state → $|v\rangle$ with prob. $p_3 \approx \|proj_v(|\psi^G\rangle)\|^2 \approx \frac{1}{2}$

   3.1 Run $\mathcal{A}_2$ -- pre-meas. unitary // rotates $|v\rangle$ to comp. basis st. $|1\rangle := \mathcal{A}_2|v\rangle$

   3.2 Run $\mathcal{A}$'s comp. basis out. **Meas.** $\mathbb{M}$ // state → $|1\rangle$ with prob. $\|proj_1(\mathcal{A}_2|\psi^G\rangle)\|^2 \approx \frac{1}{2}$.

   3.3 Run $\mathcal{A}_2^{-1}$-- **Rewind** back to query // rotates $|1\rangle$ back to $|v\rangle = \mathcal{A}_2^{-1}|1\rangle$
4. **Measure** the input reg. and ret. result // state →$|x^*\rangle|\cdot\rangle$ with prob. $p_4 \approx \|proj_\delta(|v\rangle)\|^2 \approx \frac{1}{2}$

Overall extraction success probability := $Adv_{OW}(C) = p_3 \cdot p_4 \approx \frac{1}{4}$

→ **"Measure-Rewind-Measure" (MRM) technique**

Our Idea: Measurement-Based Extraction

- **Comparison with prior OWTH results:**

OWTH Lemma	Adv(A) bound	Secret set size $\|S\|$	Extractor oracles	A's dist. event
Orig. [U14]	$2d\sqrt{Adv_{OW}}$	Arbitrary	$\|H\rangle$ **or** $\|G\rangle$	Arbitrary
Semi-Class. [AHU18]	$2\sqrt{d\ Adv_{OW}}$	Arbitrary	$(\|H\rangle \backslash S$ **or** $\|G\rangle \backslash S)$ **and** 1_S	Arbitrary
Orig. Double-Sided [BH+19]	$2\sqrt{Adv_{OW}}$	1	$\|H\rangle$ **and** $\|G\rangle$	Arbitrary
MRM	$4d\ Adv_{OW}$	Arbitrary	$\|H\rangle$ **and** $\|G\rangle$	1←A

d := A's oracle depth, Adv_{OW} := extractor's success probability,
S := set on which G, H differ, $|H\rangle \backslash S$:= restriction of $|H\rangle$ to complement(S), 1_S := indicator function of S

References

[ACISP'18] *Wilson Abel Alberto Torres, Ron Steinfeld, Amin Sakzad, Joseph K. Liu, Veronika Kuchta, Nandita Bhattacharjee, Man Ho Au, Jacob Cheng: " Post-Quantum One-Time Linkable Ring Signature and Application to Ring Confidential Transactions in Blockchain (Lattice RingCT v1.0)"*

[ACISP'19] *Wilson Abel Alberto Torres, Veronika Kuchta, Ron Steinfeld, Amin Sakzad, Joseph K. Liu, Jacob Cheng: "Lattice RingCT V2.0 with Multiple Input and Multiple Output Wallets."*

[CRYPTO'18] *B. Libert, S. Ling, K. Nguyen, and H. Wang, "Lattice-Based Zero-Knowledge Arguments for Integer Relations"*

[CRYPTO'19] *M. F. Esgin, R. Steinfeld, J. K. Liu, and D. Liu, "Lattice-based zero-knowledge proofs:New techniques for shorter and faster constructions and applications"*

[PQCrypto'19] *Jiang, H., Zhang, Z., Ma, Z.* "Tighter security proofs for generic key encapsulation mechanism in the quantum random oracle model".

[EUROCRYPT'20] *Veronika Kuchta, Amin Sakzad, Damien Stehlé, Ron Steinfeld, Shifeng Sun: "Measure-Rewind-Measure: Tighter Quantum Random Oracle Model Proofs for One-Way to Hiding and CCA Security"*