LIKE – Lattice Isomorphism-based Non-Interactive Key Exchange via Group Actions

Alessandro Budroni[1], Jesús-Javier Chi-Domínguez[1], Mukul Kulkarni[1]

[1]Cryptography Research Centre, Technology Innovation Institute, Abu Dhabi, UAE
{alessandro.budroni,jesus.dominguez,mukul.kulkarni}@tii.ae

Abstract. We propose a new Diffie-Hellman-like Non-Interactive Key Exchange that uses the Lattice Isomorphisms as a building block. Our proposal also relies on a group action structure, implying a similar security setup as in the Commutative Supersingular Isogeny Diffie-Hellman (CSIDH) protocol where Kuperberg's algorithm applies. We short label our scheme as LIKE. As with the original Diffie-Hellman protocol, our proposed scheme is also passively secure. We provide a proof-of-concept constant-time C-code implementation of LIKE, and conservatively propose LIKE-1, LIKE-3, and LIKE-5 instances with equivalent asymptotic Kuperberg's algorithm complexity than CSIDH-4096, CSIDH-6144, and CSIDH-8192. Our experiments illustrate that LIKE-1 is about 3.8x faster than CTIDH-512 (the current fastest variant of CSIDH-512), and it is about 641.271x faster than CSIDH-4096 when deriving shared keys (while LIKE-1 key generation is about 2.16x faster than CSIDH-4096); oppositely, LIKE-1 public keys are 32.25x larger than CSIDH-4096.

Keywords: Post-Quantum Cryptography · NIKE · Lattice Isomorphism Problem · Group Action

1 Introduction

The advent of quantum computation revealed that the computationally hard mathematical problems employed in public-key cryptography today, could be solved efficiently, thus undermining the security of cryptographic protocols. In Shor's seminal work [53], he introduced a quantum algorithm which solves the Discrete Logarithm Problem (DLP) over finite fieds and elliptic curves, and the Integer Factorization Problem (IFP) in polynomial time. This breakthrough result motivated the researchers to design cryptographic schemes which are secure even in the presence of adversaries with quantum resources, commonly referred to as post-quantum cryptography (PQC); and also to develop better attacks and perform cryptanalysis of cryptographic schemes using quantum computing [38,20,39,12]. Since then, the research community started to look for new quantum-secure hard problems to replace DLP and IFP.

Post-quantum cryptography has come a long way since its inception and arguably will be the focus of cryptographic research in coming years as highlighted by the NIST post-quantum cryptography standardization process [44]. PQC research is rapidly developing quantum-secure alternatives to not only many existing cryptographic primitives such as public-key encryption schemes [50], digital signatures [48,41,29,34,9], key encapsulation mechanisms (KEMs) [2,5,43,52,3], proof-of-knowledge (PoK) systems [28,27,24,54,26,10]; but also additionally building advanced cryptographic primitives such as fully homomorphic encryption (FHE) [15,13], functional encryption (FE) [51,11] etc. Another exciting aspect of PQC is – the new designs are based on a diverse set of assumptions including code-based cryptography [7,32,33], lattice-based cryptography [4,42,50], isogeny-based cryptography [5,28,8].

Motivation Non-interactive key exchange (NIKE) is one of the most useful cryptographic primitives which is embedded in modern communication over the internet. Informally, NIKE scheme allows two parties that know each other's public key, to agree on a shared secret without requiring any interaction. The Diffie-Hellman key exchange protocol [23], based on the conjectured hardness of discrete logarithm problem (DLP), is probably the most known instance of a NIKE. Surprisingly, designing quantum-secure NIKE scheme has been a challenging task with only a few post-quantum NIKE scheme [6,37] being designed till date to the best of our knowledge. Instead, the PQC is focused on design of key-encapsulation mechanisms (KEMs) as a solution for establishing a shared secret. Both KEMs and NIKE output a shared, pseudo-random key as a result of the local computation by the parties. The main difference between KEMs and NIKE is that NIKE scheme "derive" the shared secret by *combining the public keys with local secret information*, whereas in KEMs, one of the parties "encrypts" a message using other party's public key and derives the shared secret as part of the output of the encryption process, in addition to a ciphertext. This ciphertext is sent to the other party (this is the only interaction between the parties), which then "decrypts" the ciphertext and derives the same shared secret as a result of the decryption process. As a consequence, KEMs are generally more complex to design and implement.

In practice, this is undesirable since many existing real-world applications use authenticated Diffie-Hellman key exchange protocol, and replacing it with post-quantum KEMs can lead to major re-designing of infrastructure. Additionally, deploying complex schemes also increases the risk of implementation errors by developers who may not (and need not) be experts in cryptography. Another important aspect is, due to the nature of the underlying assumptions (such as noisy decoding is hard computational problem), many of the existing KEMs need an additional reconciliation step to ensure that parties agree on the shared secret. In fact this is true also of the NIKE scheme presented in [37] based on lattice-based cryptography. This motivates the following question

Is it possible to construct a Diffie-Hellman-like quantum secure non-interactive key exchange scheme that is efficient and easy to implement?

As discussed next, we advance towards the construction of such schemes by exploring the connections between lattice based computationally hard problems and group actions on hard homogeneous spaces[1]. Looking ahead, we propose a Lattice Isomorphism-based Key Exchange (LIKE) scheme based on the group action related to computation of lattice isomorphism. We believe that the conceptual simplicity, its similarity to the classical Diffie-Hellman Key Exchange, and efficient group action computation are the attractive features of our proposal. Similar to original Diffie-Hellman proposal our construction is also passively secure. We also implement a simple proof-of-concept (PoC) of our proposed scheme.

1.1 Our results / Contribution

We propose a new Diffie-Hellman-like NIKE that uses group actions over quadratic forms derived from lattices as building blocks. More specifically, we exploit the connections between lattice isomorphism problem (LIP) and group actions. Our construction is simple and conceptually close to the well-known Diffie-Hellman key exchange protocol. This similarity in the structure of the scheme can lead to a smoother transition and adaption of post-quantum cryptographic solutions. Additionally, this also minimizes the re-design of other protocols which use key exchange as means to establish secure connection.

Our main idea stems from the framework of [22] which generalizes the group-theoretic computational hard problems, like discrete logarithm problem (DLP) and computational Diffie-Hellman problem (CDH) based on actions of specific groups on certain sets. The second important ingredient is the lattice isomorphism problem (LIP) which has been studied recently by Ducas and van Woerden [24]. We define a novel group action based on the conjectured hardness of the LIP, along with analogues of DLP, CDH, and decisional Diffie-Hellman (DDH) problems related to quadratic forms of isomorphic lattices. We then propose a simple key exchange scheme based on the conjectured hardness of these problems.

Theorem 1 (Informal). *There exists an efficient, post-quantum secure non-interactive key exchange scheme based on the (conjectured) hardness of lattice isomorphism problem and analogues of DLP, CDH, and DDH in the quadratic form setting.*

Background on assumptions We first present some background on the assumptions we use to prove the security of our scheme.

Lattice Isomorphism Problem. The *lattice isomorphism problem* (LIP) is a computational problem in which, given two isomorphic lattices $\mathcal{L}, \mathcal{L}'$ the goal is to

[1] Group action is mathematical generalization of computations performed in Diffie-Hellman like scheme. See Subsection 1.1 and Subsection 2.3 for details.

find the isomorphism between them. Recently, Ducas and van Woerden [24] studied this problem in the quadratic form setting, where the lattices are represented by the quadratic form $Q := B^T B$ where B is a basis of $\mathcal{L}$. The computational problem translates into finding a unimodular matrix U such that $Q' = U^\intercal Q U$, where Q and Q' are the quadratic forms of $\mathcal{L}$ and $\mathcal{L}'$ respectively. In [24], the authors conjectured this problem to be $2^{\Theta(n)}$ hard, where n is the dimension of the lattice. In addition, they presented a security reduction that connects the hardness of LIP to the hardness of the Shortest Independent Vector Problem on a given lattice. In fact, the authors of [24] show an average-case to worst-case reduction, where the average-case instances of the problem are computed using a unimodular matrix U sampled according to the *Gaussian form distribution* (See Definition 3). Looking ahead, in our construction we build our LIP instances from unimodular matrices of the form $W := U^a$, where U is sampled from the Gaussian form distribution and a is sampled uniform randomly from $\mathbb{Z}_N$ for some large $N \in \mathbb{N}$. Despite our best efforts we cannot prove reduction from our LIP instances to worst-case (or average-case) LIP, we therefore rely on the following conjecture:

Conjecture 1. Solving the search LIP for given $Q' := W^\intercal Q W \in [Q]$ where W is a unimodular matrix of the form $W := U^a$, for uniform random $a \in \mathbb{Z}_N$ and U is the unimodular matrix for some $Q'' := U^\intercal Q U$ sampled from Gaussian form distribution, is computationally hard problem even in the presence of quantum adversaries. [2]

Group actions. The hardness of the Discrete logarithms problem (DLP) and computational Diffie-Hellman problem (CDH) are well-studied assumptions serving as bedrock of the public-key cryptography for decades. In [22], Couveignes generalizesd these problems by representing them in terms of a more general algebraic framework. A *group action* is a map between a given group and set. The group is said to *act* on the set if the map satisfies certain properties (See Definition 5 for details). In this framework DLP and CDH are seen as specific instances of two more general problems called *vectorization* and *parallelization* problems respectively. Informally, let $(G, \star)$ be a group and X be a set. Let *group action* be a map $\alpha : G \times X \to X$. The vectorization problem consists of, given $x, x' \in X$, finding the unique group element g such that $\alpha(g, x) = x'$. On the other hand, the parallelization problem consists of, given $x := \alpha(g, y)$, $x' := \alpha(h, y)$, and $y \in X$, finding $z := \alpha((g \star h), y)$. One can see that DLP and CDH are particular instances of vectorization and parallelization respectively.

In this work, we define a group action based on quadratic forms of isomorphic lattices. We then define the analogues of the DLP, CDH and decisional Diffie-Hellman problem (DDH) in this setting. To the best of our knowledge, this is the first work connecting LIP to group actions. The security of the scheme then relies on the following conjecture

Conjecture 2. (Informal) The vectorization and parallelization problems are computationally hard when the group action is instantiated with additive group

[2] Clearly the problem is identical to the average-case sLIP when $a = 1$.

$(\mathbb{Z}_N, +)$ and equivalence class of isomorphic quadratic forms ($[Q]$) as underlying G and X respectively, and group action is defined as:

$$(a \in \mathbb{Z}_n, Q' \in [Q]) \to Q'' := {U^a}^\intercal Q' U^a,$$

where U is a unimodular matrix.

Lattice Isomorphism-based Key Exchange (LIKE) Assume Alice and Bob are provided a public representative Q of an equivalence class $[Q]$. They locally sample unimodular matrices U, U' and compute $Q_a = (U)^\intercal Q\,(U)$ and $Q_b = (U')^\intercal Q\,(U')$ respectively. Alice then sends Q_a to Bob, and Bob sends Q_b to Alice. Both of them then locally compute $Q_{ab} = (U)^\intercal Q_b\,(U)$ and $Q_{ba} = (U')^\intercal Q_a\,(U')$. The shared secrets are Q_{ab} and Q_{ba}. Given that search version of LIP (hereafter referred as sLIP) is computationally hard, neither Q_a nor Q_b leak information about U and U' respectively. However, this simple construction lacks of correctness since matrix multiplication is not commutative. In fact, $UU' \neq U'U$ in general and, therefore $Q_{ab} \neq Q_{ba}$. Our solution to achieve correctness consists of restricting the group from where the private keys U, U' are sampled to a subgroup of the unimodular matrices in which commutativity is guaranteed. Let U be a unimodular matrix and let $\langle U \rangle$ be the multiplicative group generated by it. For any pair of positive integers $a, b \in \mathbb{Z}$ we have that $U^a, U^b \in \langle U \rangle$, and $U^a U^b = U^{a+b} = U^{b+a} = U^b U^a$. With this solution to achieve commutativity in private keys, we introduced the framework of group actions to quadratic forms.

Informally, let Q, Q' be two equivalent quadratic forms (representing two isomorphic lattices) and let U be a unimodular matrix such that $Q' = (U^a)^\intercal Q\,(U^a)$, for some positive integer a. The Discrete Logarithm Problem on Quadratic Forms (DLP-QF) aims to find a given Q, Q' and U. Similarly, one defines CDH and DDH on quadratic forms (See Section 3 and Definition 15, Definition 16, and Definition 17 for details). Based on the assumption that these problems are hard to solve, the Lattice Isomorphism (non-interactive) Key Exchange (LIKE) protocol described in Figure 1 follows. One can see that the shared secrets coincide as follows

$$Q_{ab} = (U^a)^\intercal Q_b\,(U^a) = \left(U^{a+b}\right)^\intercal Q\,\left(U^{a+b}\right) = \left(U^b\right)^\intercal Q_a\,\left(U^b\right) = Q_{ba}.$$

Our group action is easy to compute and involves only matrix multiplications and matrix exponentiations, both of these operations can be performed efficiently. To compare the efficiency of our protocol against other post-quantum NIKE schemes such as CTIDH-512 (the current fastest CSIDH-like variant) and CSIDH-4096, we implemented a constant-time proof-of-concept of our scheme. Despite being non-optimized (for exmaple, we use the simple school-book matrix multiplication), our implementation shows a clear advantage in speed for key-derivation against the other protocols (see Table 1).

1.2 Related Work

Isogeny-based primitives are currently in the eye to building a NIKE; they ensure (sometimes) significantly shorter keys than the other quantum-secure primitives.

Alice **Public parameters:** security parameter $\lambda \in \mathbb{Z}$, quadratic form Q, unimodular matrix U **Bob**

Key generation

Sample $a \xleftarrow{\$} [\![0 \, .. \, 2^\lambda - 1]\!]$
Compute $Q_a \leftarrow (U^a)^\mathsf{T} Q \, (U^a)$

Sample $b \xleftarrow{\$} [\![0 \, .. \, 2^\lambda - 1]\!]$
Compute $Q_b \leftarrow \left(U^b\right)^\mathsf{T} Q \left(U^b\right)$

$\xrightarrow{\quad Q_a \quad}$

$\xleftarrow{\quad Q_b \quad}$

Shared secret derivation

Derive $Q_{ab} \leftarrow (U^a)^\mathsf{T} Q_b \, (U^a)$

Derive $Q_{ba} \leftarrow \left(U^b\right)^\mathsf{T} Q_a \left(U^b\right)$

Fig. 1: Informal description of LIKE

	Public-key size		Running time (clock cycles)	
	CTIDH-512	CSIDH-4096	CTIDH-512	CSIDH-4096
LIKE-1	258x larger	32.25x larger	3.82x faster	641.271x faster
	CSIDH-6144		LIKE-1	
LIKE-3	48.25x larger		3.4x slower	
	CSIDH-8192		LIKE-1	
LIKE-5	64.25x larger		8.2x slower	

Table 1: Speedup and size factors concerning CTIDH-512 [6] and CSIDH-[4096/6144/8192] [19] compared with LIKE-1, LIKE-3, and LIKE-5. We compare the efficiency of our proposal of LIKE-3 and LIKE-5 with LIKE-1 to illustrate the impact of increasing the security parameters.

Nevertheless, they come with a considerable latency that penalizes them when compared with, for example, lattice-based primitives. In 2018, Castryck et al. presented the first quantum-resistant NIKE based on group actions and isogenies, named CSIDH [17]. In 2021, Banegas et al. [6] significantly improved the efficiency of CSIDH's group action by moving to a different private keyspace and combining it with a Matryoshka trick on the isogeny computations; they called their proposal CTIDH and illustrated a 2x speedup factor compared to CSIDH. Following the path of isogenies, recently, Leroux proposed pSIDH as a new NIKE relying on the suborder to ideal problem [40].

In 2018, Bor de Kock described a post-quantum NIKE based on ring-Learning With Errors [37]. In 2019, Ji, Qiao, Song, and Yun analyzed post-quantum primitives falling into group actions on 3-tensors [35]; their security relies on the

3-tensor isomorphism problem, studied by Futorny, Grochow, and Sergeichuk in [31]. Tang, Duong, Joux, Plantard, Qiao, and Susilo recently presented a signature scheme based on the 3-tensor isomorphism problem [55]. Lastly, Ducas and van Woerden proposed a lattice-based KEM centered on the Lattice Isomorphism Problem (LIP) [24].

1.3 Organization of the Paper

The paper is organized as follows: We present the preliminaries and notation in Section 2, followed by the background on lattice isomorphism along with the related definitions and lemmas in Subsection 2.1. We then present some important properties of unimodular matrices in Subsection 2.2, and background, important definitions, and lemmas related to group action in Subsection 2.3. The definition of NIKE is given in Subsection 2.4. Our Section 3 focuses on the connections between quadratic forms and group action based on QF along with DH like assumptions. The main construction and security proof of LIKE is presented in Section 4, followed by the cryptanalysis of our assumptions in Section 5. We present the experimental data and implementation details in Section 6 and conclude with the conclusion in Section 7.

2 Preliminaries and Notation

Let $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$ and $\mathbb{R}$ denote the sets of natural, integer, rational and real numbers respectively. We denote vectors in boldface (e.g. $\mathbf{x}$) and treat them as column vectors by default. We denote matrices by uppercase letters (e.g. M). For a vector $\mathbf{x}$ in $\mathbb{R}^n$, define the ℓ_2 norm as $\|\mathbf{x}\|_2 := \left(\sum_{i\in[n]} |x_i|^2\right)^{\frac{1}{2}}$, where $|x_i|$ is the absolute value of the i^{th} component of $\mathbf{x}$. We write $\|\mathbf{x}\|$ to denote ℓ_2 norm for simplicity. For a matrix B with columns $\mathbf{b}_1, \ldots, \mathbf{b}_n$, we denote its Gram-Schmidt orthogonalization by B^* with columns $\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*$. We also denote the matrix norm of B by $\|B\| := \max_i \|\mathbf{b}_i\|$.

The set of all $n \times n$ invertible matrices with entries in ring $\mathcal{R}$ is denoted by $\mathcal{GL}_n(\mathcal{R}) := \{M \in \mathcal{R}^{n\times n} : \det(M) \neq 0\}$. Similarly, the set of matrices with determinant 1 and entries in ring $\mathcal{R}$ is denoted by $\mathcal{SL}_n(\mathcal{R}) := \{U \in \mathcal{R}^{n\times n} : \det(U) = 1\} \subset \mathcal{GL}_n(\mathcal{R})$.

The set of all *orthonormal* matrices with entries in field $\mathbb{F}$ is denoted by $\mathcal{O}_n(\mathbb{F}) := \{O \in \mathbb{F}^{n\times n} : OO^T = O^TO = I_n \text{ and } \|\mathbf{o}_i\| = 1\, \forall i \in [n]\}$ where I_n is $n \times n$ identity matrix. A square matrix O is called *orthonormal* if and only if its transpose O^T is also its inverse and each column vector $\mathbf{o}_1, \ldots, \mathbf{o}_n$ has norm exactly equal to 1. A matrix $S \in \mathbb{R}^{n\times n}$ is called *symmetric positive definite* if $S = S^T$ and $\mathbf{x}^T S\mathbf{x} > 0$ for all $\mathbf{x} \in \mathbb{R}^n \setminus \{\mathbf{0}\}$. The set of all *symmetric positive definite* matrices over $\mathbb{R}$ is denoted by $\mathcal{S}_n^{>0}$.

2.1 Lattice Isomorphism and Quadratic Form

A full-rank n-dimensional lattice $\mathcal{L} = \mathcal{L}(B) := B \cdot \mathbb{Z}^n$ is generated by taking all the possible integer combinations of the (linearly independent) columns of a basis $B \in \mathbb{R}^{n\times n}$. Denote with $\lambda_1(\mathcal{L}(B)) = \min_{\mathbf{x}\in\mathcal{L}\setminus\{\mathbf{0}\}}\|\mathbf{x}\|$ the length of a shortest non-zero vector of $\mathcal{L}$, and let $\mathsf{gh}(\mathcal{L}(B))$ denote the *Gaussian Heuristic* estimate for $\lambda_1(\mathcal{L}(B))$ defined as:

$$\mathsf{gh}(\mathcal{L}(B)) = \sqrt{\frac{n}{2\pi e}} \cdot \det(B)^{1/n}.$$

Two bases B and B' generate the *same* lattice if and only if $\exists\, U \in \mathcal{GL}_n(\mathbb{Z})$ such that $B' = BU$. Two lattice $\mathcal{L}$, $\mathcal{L}'$ are *isomorphic* if there exists an orthonormal transformation $O \in \mathcal{O}_n(\mathbb{R})$ such that $\mathcal{L}' = O \cdot \mathcal{L}$.

Definition 1 (Search Lattice Isomorphism Problem (sLIP)). *Given two isomorphic lattices $\mathcal{L}$, $\mathcal{L}' \subset \mathbb{R}^n$ find an orthonormal transform $O \in \mathcal{O}_n(\mathbb{R})$ such that $\mathcal{L}' = O \cdot \mathcal{L}$.*

The above problem can be rephrased as follows. Given the bases B, $B' \in \mathcal{GL}_n(\mathbb{R})$ for $\mathcal{L}$ and $\mathcal{L}'$ respectively, find $O \in \mathcal{O}_n(\mathbb{R})$ *along with* $U \in \mathcal{GL}_n(\mathbb{Z})$ such that $B' = OBU$. In practice, the real-valued entries of basis and orthonormal matrices can be inconvenient to represent and result in inefficient computations. However, this can be eased by considering an equivalent problem to the LIP by taking the quadratic form of B, a.k.a Gram matrix $Q := B^T B$.

Note that, the quadratic form Q is symmetric by definition. Moreover, since B is a basis (and thus full-rank), Q is actually *symmetric positive definite*. Recall that, since $\mathcal{L}(B) := B \cdot \mathbb{Z}^n$, every lattice vector in $\mathcal{L}$ can be written as $B\mathbf{x}$, where $\mathbf{x} \in \mathbb{Z}^n$. In the quadratic form setting each lattice vector $B\mathbf{x}$ is represented by its integral basis coefficient $\mathbf{x} \in \mathbb{Z}^n$. The norm of vector $\mathbf{x}$ can be naturally defined in the quadratic form as $\|\mathbf{x}\|_Q^2 := \mathbf{x}^T Q \mathbf{x}$. Similarly, the inner product with respect to Q can be defined as $\langle \mathbf{x}, \mathbf{y}\rangle_Q := \mathbf{x}^T Q \mathbf{y}$. We extend also the notation for the shortest vector norm and heuristic to quadratic forms. Specifically, we define

$$\lambda_1(Q) := \min_{\mathbf{x}\in\mathbb{Z}^n\setminus\{0\}} \|\mathbf{x}\|_Q,$$

and *gaussian heuristic* (heuristic estimate of $\lambda_1(Q)$) as

$$\mathsf{gh}(Q) \approx (\det(Q))^{1/2n} \cdot \sqrt{\frac{n}{2\pi e}}.$$

In general, the i^{th} minimum distance $\lambda_i(Q)$ is the *smallest* radius $r > 0$, such that $\{\mathbf{x} \in \mathbb{Z}^n \quad | \quad \|\mathbf{x}\|_Q \le r\}$ contains i linearly independent vectors.

We can now rephrase the LIP problem in terms of quadratic forms. For $\mathcal{L}, \mathcal{L}'$ isomorphic lattices with respective basis B, B', we have that $B' = OBU$ where $O \in \mathcal{O}_n(\mathbb{R})$ is orthonormal and $U \in \mathcal{GL}_n(\mathbb{Z})$ is unimodular, then we have,

$$Q' := {B'}^T B' \ = U^T B^T O^T O B U \ = U^T B^T B U \ = U^t Q U$$

where, $Q := B^T B$ is the quadratic form of B. We call Q, Q' equivalent if such $U \in \mathcal{GL}_n(\mathbb{Z})$ exists. We also denote the equivalence class by $[Q]$.

The following definition is referred to as the *worst-case* sLIP in quadratic form formulation [24].

Definition 2 (wc − sLIP$^{\boldsymbol{Q}}$, [24, Definition 2.2]). *For a quadratic form $Q \in \mathcal{S}_n^{>0}$, the problem* $\mathrm{wc} - \mathrm{sLIP}^Q$ *is, given any quadratic form $Q' \in [Q]$, to find a unimodular $U \in \mathcal{GL}_n(\mathbb{Z})$ such that $Q' = U^T Q U$.*

Ducas and van Woerden provide a polynomial time algorithm **Extract** that, on input a set of n linearly independent vectors Y and a quadratic form Q, returns a pair (Q', U) such that $Q' = U^{\mathsf{T}} Q U$ [24, Lemma 3.1]. They also show that Q' is independent from the input class representative Q [24, Lemma 3.2].

Discrete Gaussians and Sampling For any quadratic form $Q \in \mathcal{S}_n^{>0}$, the Gaussian function on $\mathbb{R}^n$ with parameter $s > 0$ and center $\mathbf{c}$ is defined by

$$\forall \mathbf{x} \in \mathbb{R}^n,\ \rho_{Q,s,\mathbf{c}}(\mathbf{x}) := \exp\left(-\pi \|\mathbf{x} - \mathbf{c}\|_Q^2 / s^2\right).$$

The discreet Gaussian distribution is obtained by restricting the continuous gaussian distribution to a discreet lattice. In the quadratic form setting, the underlying lattice will always be $\mathbb{Z}^n$, but with the geometry induced by the quadratic form. For any quadratic form $Q \in \mathcal{S}_n^{>0}$, parameter $s > 0$ and center $\mathbf{c}$, the discreet Gaussian distribution $\mathcal{D}_{Q,s,\mathbf{c}}$ is defined as

$$\Pr_{X \sim \mathcal{D}_{Q,s,\mathbf{c}}}[X = \mathbf{x}] := \begin{cases} \frac{\rho_{Q,s,\mathbf{c}}(\mathbf{x})}{\rho_{Q,s,\mathbf{c}}(\mathbb{Z}^n)} & \text{if } \mathbf{x} \in \mathbb{Z}^n, \\ 0 & \text{otherwise} \end{cases}.$$

If the center $\mathbf{c} = \mathbf{0}$, then we omit it.

Brakerski *et al.* [14, Lemma 2.3] showed how to sample from the discrete gaussian distribution distribution efficiently.

Definition 3 (Gaussian form distribution, [24, Definition 3.3]). *Given a quadratic form equivalence class $[Q] \subset \mathcal{S}_n^{>0}$, the Gaussian form distribution $\mathcal{D}_s([Q])$ over $[Q]$ with parameter $s > 0$ is defined algorithmically as follows:*

1. *Fix a representative $Q \in [Q]$.*
2. *Sample n vectors $(\mathbf{y}_1, \mathbf{y}_2, \dots \mathbf{y}_n) := Y$ from $\mathcal{D}_{Q,s}$. Repeat until linearly independent.*
3. *$(R, U) \leftarrow$ **Extract**(Q, Y).*
4. *Return R.*

Definition 4 (ac − sLIP$_{\boldsymbol{s}}^{\boldsymbol{Q}}$, [24, Definition 3.7]). *For a quadratic form $Q \in \mathcal{S}_n^{>0}$ and $s > 0$ the problem* $\mathrm{ac} - \mathrm{sLIP}_s^Q$ *is, given a quadratic form sampled as $Q' \leftarrow \mathcal{D}_s([Q])$, to find a unimodular $U \in \mathcal{GL}_n(\mathbb{Z})$ such that $Q' = U^T Q U$.*

In [24], the authors show that the worst-case and average-case problems are equivalent (via reduction from worst-case to average-case). We report the relevant lemma stating such reduction.

Lemma 1 (ac − sLIP$_{\boldsymbol{s}}^{\boldsymbol{Q}}$ $\geq$ wc − sLIP$^{\boldsymbol{Q}}$ for large $\boldsymbol{s}$, [24, Lemma 3.9]). *Given an oracle that solves* $\mathrm{ac} - \mathrm{sLIP}_s^Q$ *for some $s \geq 2^{\Theta(n)} \cdot \lambda_n([Q])$ in time T_0 with probability $\varepsilon > 0$, we can solve* $\mathrm{wc} - \mathrm{sLIP}^Q$ *with probability at least ε in time $T + \mathrm{poly}(n, \log s)$.*

For smaller values of s the authors of [24] give a reduction based on stronger lattice reduction algorithms.

Lemma 2 **($\mathbf{ac-sLIP}_s^Q \geq \mathbf{wc-sLIP}^Q$, [24, Lemma 3.10]).** *Given an oracle that solves* $\mathrm{ac}-\mathrm{sLIP}_s^Q$ *for some* $s \geq \lambda_n(Q)$ *in time* T_0 *with probability* $\varepsilon > 0$*, we can solve* $\mathrm{wc}-\mathrm{sLIP}^Q$ *with probability at least* $\frac{1}{2}$ *in time*

$$T = \frac{1}{\varepsilon}(T_0 + \mathrm{poly}(n, \log s)) + C\left(n, \frac{s}{\lambda_n(Q) \cdot \sqrt{\ln(2n+4)/\pi}}\right),$$

where $C(n, f)$ *is the cost of solving the Shortest Independent Vector Problem (SIVP,[50]) within approximation factor of* f*.*

2.2 Properties of Unimodular Matrices

We give here some useful properties of unimodular matrices. Let n be a positive integer, and $\mathbb{Z}_p$ be the integers modulo a prime number p. We have that

$$\#\mathcal{GL}_n(\mathbb{Z}_p) = (p^n - 1) \cdot (p^n - p) \cdot \left(p^n - p^2\right) \cdots \left(p^n - p^{n-1}\right).$$

Since the determinant function $\det\colon \mathcal{GL}_n(\mathbb{Z}_p) \to \mathbb{Z}_p^*$ is a surjective homomorphism with kernel being the subgroup $\mathcal{SL}_n(\mathbb{Z}_p) \subset \mathcal{GL}_n(\mathbb{Z}_p)$, by the Fundamental Isomorphism Theorem, we have that the quotient group $\mathcal{GL}_n(\mathbb{Z}_p)/\mathcal{SL}_n(\mathbb{Z}_p)$ is isomorphic to $\mathbb{Z}_p^*$ and hence $\#\mathcal{GL}_n(\mathbb{Z}_p) = (p-1) \cdot \#\mathcal{SL}_n(\mathbb{Z}_p)$. In other words,

$$\#\mathcal{SL}_n(\mathbb{Z}_p) = \frac{(p^n - 1) \cdot (p^n - p) \cdot \left(p^n - p^2\right) \cdots \left(p^n - p^{n-1}\right)}{p-1}.$$

Moreover, the number of unimodular matrices (non-singular matrices with determinant ± 1) over $\mathbb{Z}_p$ becomes $\frac{2 \cdot (p^n - 1) \cdot (p^n - p) \cdot \left(p^n - p^2\right) \cdots \left(p^n - p^{n-1}\right)}{p-1}$. On the other hand, the group $\mathcal{GL}_n(\mathbb{Z}_p)$ is generated by the following two matrices [56]:

$$\sigma_1 = -\left(\begin{array}{c|c} \mathbf{0} & 1 \\ \hline \mathbf{Id}_{n-1} & \mathbf{0} \end{array}\right) \quad \text{and} \quad \sigma_2 = \left(\begin{array}{c|c} \begin{matrix} 1 & 1 \\ 0 & 1 \end{matrix} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{Id}_{n-2} \end{array}\right).$$

If n is even, then $-\sigma_1$, and σ_2 generate the whole $\mathcal{GL}_n(\mathbb{Z}_p)$; otherwise, $-\sigma_1$, and σ_2 only generate $\mathcal{SL}_n(\mathbb{Z}_p)$, and $\mathcal{GL}_n(\mathbb{Z}_p)$ is isomorphic to $\mathcal{SL}_n(\mathbb{Z}_p) \times \langle -\mathrm{Id}_n \rangle$.

2.3 Group Actions

We now present some definitions and computationally hard problems related to mathematical objects and functions which are used as building blocks for our construction and security proofs.

Definition 5. *Let $(G, \star)$ be a group with identity element e, and let X be a set. A group action of G on X is a map from $G \times X$ to X, where the image of a pair (g, x) is denoted with $g \circ x$ such that:*

- *(identity): $e \circ x = x$, for all $x \in X$;*
- *(compatibility): $(g_1 \star g_2) \circ x = g_1 \circ (g_2 \circ x)$, for all $g_1, g_2 \in G$ and for all $x \in X$.*

We say that G *acts* on X if there exist a group action of G on X.

A group action is said to be *regular* if the following two properties hold:

- *(transitive)*: for each $x, y \in X$, there exists a $g \in G$ such that $g \circ x = y$;
- *(free)*: if, for $g \in G$, there exists a $x \in X$ such that $x = g \circ x$, then g is the identity.

Definition 6 (Principal Homogeneous Spaces (PHS)). *Let $(G, \star)$ be an abelian group and let X be a set equipped with a regular group action of G. Then X is said to be a Principal Homogeneous Space.*

From now on, we will assume that the group operation $\star$ of the group G is efficient to compute.

Definition 7 (Vectorization Problem). *Let X be a PHS under a group $(G, \star)$. Given $x, x' \in X$, compute the unique $g \in G$ such that $x' = g \circ x$.*

Definition 8 (Parallelization Problem). *Let X be a PHS under a group $(G, \star)$. Given $y, x, x' \in X$ such that $x = g \circ y$ and $x' = h \circ y$, compute $z \in X$ such that $z = (g \star h) \circ y$.*

Definition 9 (Hard Homogeneous Space (HHS)). *Let X be a PHS under a group $(G, \star)$. If the group action $\circ$ is efficiently computable but the vectorization and the parallelization problems are computationally hard to solve, then we say that X is a Hard Homogeneous Space.*

Looking ahead, our construction uses a specific type of group action (see Section 3) and the security relies on the hardness of solving *Hidden Shift Problem* defined below.

Definition 10 (Hidden Shift Problem). *Let $(G, \star)$ be a group. The* Hidden Shift Problem *is to find $s \in G$, given two permutations $f_0, f_1 : G \to G$ such that for all $x \in G$, $f_1(x) = f_0(x + s)$.*

In the above, it is assumed that such s exists for given f_0, f_1. For $G = \mathbb{Z}_N$ and some set X with the associated group action $\circ$, the Vectorization Problem (Definition 7) becomes an instance of the Hidden Shift Problem over G. First, define $f_0 \colon g \to g \circ x$ and $f_1 \colon g \to g \circ x'$. Then,

$$f_1(g) = g \circ x' = g \circ (s \circ x) = (g + s) \circ x = f_0(g + s)$$

is a shifted version of f_0. Finding s reduces to solving the Hidden Shift Problem over G.

The hidden shift problem (Definition 10) is a special case of Hidden Subgroup Problem (a well-studied computation problem, See Definition 12) on a related group $\overline{G}$ (the G-dihedral group) [3].

We define the G-dihedral group and the Hidden Subgroup Problem (HSP) below.

Definition 11 (G-dihedral group, [25]). *Let $G = \mathbb{Z}_N$ be the additive group of integers modulo N. The G-dihedral group of order $2N$ is a regular N-sided polygons symmetry group, including rotations and flips. We denote the G-dihedral group by $\overline{G}$. More precisely, $\overline{G}$ coincides with the semidirect product $G \rtimes \mathbb{Z}_2$ determined by the relation*

$$\rtimes \colon \; ((g_1, z_1), (g_2, z_2)) \mapsto ((g_1 + \phi(z_1)(g_2), z_1 + z_2))$$

where ϕ is an homomorphism defined as

$$\begin{aligned} \phi \colon \mathbb{Z}_2 &\to Aut(G) \\ z &\mapsto \phi_z \colon g \mapsto (-1)^z g, \end{aligned}$$

and $Aut(G)$ denotes the group of automorphisms on G.

Let H be a subgroup of G. We say that a function $f : G \to X$ *hides* the subgroup H if, for all $g_1, g_2 \in G$, $f(g_1) = f(g_2)$ if and only if $g_1 H = g_2 H$. In the following, we assume f can be computed efficiently.

Definition 12 (Hidden Subgroup Problem). *Let G be a group, $H \subseteq G$ be a subgroup and X be a set. Given a function $f : G \to X$ that hides H, the Hidden Subgroup Problem (HSP) is to find a generator of H.*

The reduction from the hidden *shift* problem over G to the hidden *subgroup* problem (HSP) over $\overline{G}$ is done by converting the image $f_z(g)$ for $g \in G$ and $z \in \mathbb{Z}_2$ to the element $\overline{g} := (g, z) \in G \rtimes \mathbb{Z}_2$.

[3] For a detailed read, we encourage the readers to check [38, §2] and [49,25].

2.4 Key Exchange Protocols

We define the non-interactive key exchange scheme in the public key setting following the formal definitions given in [16], and [30].

Definition 13 (NIKE). *A non-interactive key exchange (NIKE) scheme is a tuple of algorithms*

$$\mathsf{NIKE} = (\mathsf{NIKE.Setup}, \mathsf{NIKE.KeyGen}, \mathsf{NIKE.SharedKey})$$

together with a shared keyspace $\mathcal{K}$, where

- $\mathsf{pp} \leftarrow \mathsf{NIKE.Setup}(1^\lambda)$*: a setup algorithm takes the security parameter λ as input and outputs the public parameters* pp*.*
- $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{NIKE.KeyGen}(\mathsf{pp})$*: a probabilistic polynomial time* (PPT) *algorithm taking on input public parameters and returning a pair of public and secret keys. Any user should be able to generate its own pair of keys from the public parameters.*
- $k \leftarrow \mathsf{NIKE.SharedKey}(\mathsf{pp}, \mathsf{pk}_1, \mathsf{sk}_2)$*: given the public key of one party* pk_1*, and the secret key of another party* sk_2*, along with the public parameters* pp *as input, this algorithm returns a shared key $k \in \mathcal{K} \bigcup \{\perp\}$ among the two parties.*

Correctness: a NIKE scheme provides correctness if, for all honestly generated public parameters pp, we get

$$\mathsf{NIKE.SharedKey}(\mathsf{pp}, \mathsf{pk}_1, \mathsf{sk}_2) = \mathsf{NIKE.SharedKey}(\mathsf{pp}, \mathsf{pk}_2, \mathsf{sk}_1)$$

where $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{NIKE.KeyGen}(\mathsf{pp})$ are honestly generated public and secret keys for $i \in \{1, 2\}$.

Security: let $\mathsf{pp} \leftarrow \mathsf{NIKE.Setup}(1^\lambda)$, and $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{NIKE.KeyGen}(\mathsf{pp})$ for $i \in \{1, 2\}$. We say that a NIKE scheme is (passively) secure if any PPT adversary $\mathcal{A}$ cannot distinguish between the following to games:

- Game_0: the adversary $\mathcal{A}$ receives a shared key
$$k \leftarrow \mathsf{NIKE.SharedKey}(\mathsf{pp}, \mathsf{pk}_1, \mathsf{sk}_2)$$
along with the public parameters pp, and pk_1, pk_2.
- Game_1: the adversary $\mathcal{A}$ receives a random key $k \leftarrow \mathcal{K}$ along with the public parameters pp, and pk_1, pk_2.

3 Diffie-Hellman over Quadratic Forms

Let $Q \in \mathcal{S}_n^{>0}$ be a quadratic form of dimension n, and let $U \in \mathcal{GL}_n(\mathbb{Z}_q)$ be an unimodular matrix. Let N be the smallest positive integer such that U^N is the identity matrix Id. Such an N always exists because $\mathbb{Z}_q$ is finite and, if U is idempotent, then $U = \mathrm{Id}$[4]. In the following, we will assume N to be prime or a power of a prime.

Definition 14 (Quadratic Form Group Action). *Consider the additive group $\mathbb{Z}_N$ of integers modulo N. Define the set*

$$\mathcal{X}_{U,Q} = \left\{ (U^g)^\intercal Q\,(U^g) : g \in \mathbb{Z}_N \right\}$$

and the map

$$\begin{aligned} \alpha\colon \mathbb{Z}_N \times \mathcal{X}_{U,Q} &\to \mathcal{X}_{U,Q} \\ (g,x) &\mapsto \alpha(g,x) \coloneqq (U^g)^\intercal x\,(U^g)\,, \end{aligned}$$

where U^g denotes the matrix obtained by raising U to the power of g, and, by convention, $U^0 = \mathrm{Id}$.

Proposition 1. *For every choice of $U \in \mathcal{GL}_n(\mathbb{Z}_q)$, the map α from Definition 14 is a group action of $\mathbb{Z}_N$ on $\mathcal{X}_{U,Q}$.*

Proof. Let $U \in \mathcal{GL}_n(\mathbb{Z}_q)$. We need to prove that both the identity and compatibility properties of group actions hold. The first one is true since $\alpha(0,x) = \left(U^0\right)^\intercal x\left(U^0\right) = x$. Given $g_1, g_2 \in \mathbb{Z}_N$, we have that

$$\alpha(g_1, \alpha(g_2, x)) = \left(U^{g_1+g_2}\right)^\intercal x\left(U^{g_1+g_2}\right) = \alpha(g_1+g_2, x),$$

so the compatibility property holds too.

Remark 1. Note that, thanks to the commutativity of $\mathbb{Z}_N$, we have that

$$\begin{aligned} \alpha(g_1, \alpha(g_2, x)) &= \alpha(g_1, (U^{g_2})^\intercal x (U^{g_2})) = (U^{g_1+g_2})^\intercal x (U^{g_1+g_2}) \\ &= \alpha(g_2, (U^{g_1})^\intercal x (U^{g_1})) = \alpha(g_2, \alpha(g_1, x)) \end{aligned}$$

Proposition 2. *The group action defined in Definition 14 is regular.*

Proof. We start proving the transitive property by construction. Let $g_1, g_2' \in \mathbb{Z}_N$ such that $x = (U^{g_1})^\intercal Q\,(U^{g_1})$ and $x' = (U^{g_2})^\intercal Q\,(U^{g_2})$. Consequently, $x' = (U^{g_2-g_1})^\intercal x\,(U^{g_2-g_1})$. We prove now the free property. If $g = 0$, then $U^g = \mathrm{Id}$ and $x = (\mathrm{Id})x(\mathrm{Id}) = x$. On the other hand, if $x = \alpha(g,x)$, then $x = (U^g)^\intercal x\,(U^g) = (U^g)^\intercal (U^g)^\intercal x\,(U^g)\,(U^g)$, and therefore $(U^g)\,(U^g) = U^g$. Since the the identity matrix is the only non-singular idempotent matrix in $\mathbb{Z}_q$, we have that $U^g = \mathrm{Id}$. Hence, $g = 0$ coincides with the identity of $\mathbb{Z}_N$.

[4] For a unimodular idempotent matrix $U \in \mathcal{GL}_n(\mathbb{Z}_q)$, we have that $\mathrm{Id} = UU^{-1} = U^2U^{-1} = U$.

Proposition 2 implies that $\mathcal{X}_{U,Q}$ is a Principal Homogeneous Space. The group operation cost consists simply of an addition of integers modulo N. The group action consists of raising the matrix U to a power $g \in \mathbb{Z}_N$, then perform 2 matrix multiplications. With an analogous approach to the one of Joye-Yen for modular exponentiation [36], and given that a matrix squaring has the same cost of matrix multiplication, the overall cost is reduced to be $2(\log_2(N)+1)$ matrix multiplications. For $N = O(q^n)$, for some $q > 1$ linear in n, the time complexity to perform the group operation is reduced to be $2(n\log_2(q)+1)$ matrix multiplications. Using, for example, the school book algorithm for matrix multiplication with a cost of n^3 scalar multiplications, the overall cost of the group action is therefore polynomial in n.

We give now a reformulation of the *Vectorization* and *Parallelization* problems for the $\mathcal{X}_{U,Q}$ setting case respectively. These can be seen as the Discrete Logarithm Problem (DLP) and Computational Diffie Hellman Problem (CDHP) adapted to our study case. We will assume $U \in \mathcal{GL}_n(\mathbb{F}_q)$ unimodular and $Q \in \mathcal{S}_n^{>0}$ to be public.

Definition 15 (DLP on Quadratic Forms (DLP-QF)). *Given $Q_a \in \mathcal{X}_{U,Q}$, with $Q_a = (U^a)^\intercal Q\,(U^a)$, for some secret $a \in \mathbb{Z}_N$, find a.*

Definition 16 (CDHP on Quadratic Forms (CDHP-QF)). *Given two elements $Q_a, Q_b \in \mathcal{X}_{U,Q}$, with $Q_a = (U^a)^\intercal Q\,(U^a)$ and $Q_b = \left(U^b\right)^\intercal Q\left(U^b\right)$, for some secret $a, b \in \mathbb{Z}_N$, find $Q_s = \left(U^{a+b}\right)^\intercal Q\left(U^{a+b}\right)$.*

Conjecture 3. The set $\mathcal{X}_{U,Q}$ is a Hard Homogeneus Space, that is, Vectorization (DLP-QF) and Parallelization (CDHP-QF) problems are computationally hard.

We introduce another computational problem that can be seen as the analogous of the Decisional Diffie-Hellman Problem (DDHP) to our setting.

Definition 17 (DDHP on Quadratic Forms (DDHP-QF)). *The Decisional Diffie-Hellman Problem on Quadratic Forms is to distinguish with non-negligible advantage between the distributions*

$$\left((U^a)^\intercal Q(U^a), (U^b)^\intercal Q(U^b), (U^{a+b})^\intercal Q(U^{a+b})\right)$$

and

$$\left((U^a)^\intercal Q(U^a), (U^b)^\intercal Q(U^b), (U^c)^\intercal Q(U^c)\right),$$

where a, b, c are chosen uniformly at random from $\mathbb{Z}_N$.

Conjecture 4. We conjecture that the Decisional Diffie-Hellman Problem on Quadratic Forms is computationally hard.

4 A New Non-Interactive Key Exchange

From the analysis in Section 3, we build the following non-interactive key exchange protocol. Figure 2 gives an explicit description of our proposal.

Public parameters: Equivalence class $[Q]$, security parameter $\lambda \in \mathbb{Z}$,
$U \in \mathcal{GL}(\mathbb{Z}^n)$ unimodular and $N \in \mathbb{Z}_q$ such that $U^N = \mathrm{Id}$ and $N > 2^\lambda$

Alice		**Bob**
	Key generation	
Sample $a \xleftarrow{\$} [\![0 \mathinner{..} 2^\lambda - 1]\!]$		Sample $b \xleftarrow{\$} [\![0 \mathinner{..} 2^\lambda - 1]\!]$
Compute $U_A \leftarrow U^a$		Compute $U_B \leftarrow U^b$
Calculate $Q_A \leftarrow {U_A}^\intercal Q U_A$	$\xrightarrow{Q_A}$	Calculate $Q_B \leftarrow {U_B}^\intercal Q U_B$
	$\xleftarrow{Q_B}$	
	Shared secret derivation	
Derive $Q_{AB} \leftarrow {U_A}^\intercal Q_B U_A$		Derive $Q_{BA} \leftarrow {U_B}^\intercal Q_A U_B$

Fig. 2: Lattice Isomorphism based Key Exchange. We assume U comes from a sample $Q' \leftarrow \mathcal{D}_s([Q])$ satisfying $Q' = U^\intercal Q U$, and $N = O(q^n)$.

Setup. Let q be a power of a prime number p, n be a positive integer, and $N = O(q^n)$. Let $Q \in \mathcal{S}_n^{>0}$ and $U \in \mathcal{GL}_n(\mathbb{Z}_q)$ unimodular be public such that $U^N = \mathrm{Id}$.

Key Generation. Both the public keys and the shared keys are elements of the set $\left\{(U^g)^\intercal Q\,(U^g) : g \in \mathbb{Z}_N\right\}$. Each party samples a random secret residue $d \in \mathbb{Z}_N$, and compute its public key as $Q_d = \left(U^d\right)^\intercal Q\left(U^d\right)$.

Key Derivation. Let Alice and Bob be the two parties of the key exchange, and let Q_a and Q_b be their public keys respectively. Upon receiving Bob's public key, Alice computes her shared key as $Q_{ab} = (U^a)^\intercal Q_b\,(U^a)$. In parallel, Bob computes his shared key Q_{ba}. Due to the commutativity of $\mathbb{Z}_n$ (see Remark 1), we have that $Q_{ab} = Q_{ba}$.

Theorem 2. *Let*
$$\mathcal{X}_{U,Q} = \left\{(U^g)^\intercal Q\,(U^g) : g \in \mathbb{Z}_N\right\}$$
be a hard homogeneous space, and $\alpha \colon \mathbb{Z}_N \times \mathcal{X}_{U,Q} \to \mathcal{X}_{U,Q}$ *defined as* $(g, x) \mapsto \alpha(g, x) := (U^g)^\intercal x\,(U^g)$*, be a map, where* U^g *denotes the matrix obtained by raising* U *to the power of* g*, and, by convention,* $U^0 = \mathrm{Id}$*. If Conjecture 1 and Conjecture 4 hold true, then the scheme presented in Figure 2 is a passively secure non-interactive key exchange (NIKE) scheme.*

The correctness of the shared keys $Q_{AB} = Q_{BA}$ follows from the fact that the U_A and U_B commute.

Proof (Proof of security Theorem 2). In order to prove the security of the scheme we need to show that any PPT(quantum) adversary cannot distinguish between the shared key $k := Q_{AB} = Q_{BA}$ and a given uniform random matrix $Q' \in [Q]$

when given along with the public parameters U, Q, s. This follows from Conjecture 4 since $(U_A U_B) = U^{a+b}$ is indistinguishable from U^c. We also additionally need to show that, the public values Q_A (resp. Q_B) do not leak any information about the secret key a (resp. b) and secret unimodular matrices $U_A := U^a$ (resp. $U_B := U^b$).

Note that recovering a (resp. b) from the public key Q_A (resp. Q_B) is exactly the *vectorization problem* which is computationally hard assuming $\mathcal{X}_{U,Q}$ is a hard homogeneous space (See Conjecture 3 and Definition 15). Similarly, recovering the shared key Q_{AB} from the public keys Q_A and Q_B is exactly the *parallelization problem* which is computationally hard assuming $\mathcal{X}_{U,Q}$ is a hard homogeneous space (See Conjecture 3 and Definition 16).

Finally, recovering the secret unimodular matrices U_A (resp. U_B) from the public keys Q_A (resp. Q_B) is computationally hard assuming our LIP instances to be hard, see Conjecture 1. This completes the proof of Theorem 2.

In section 5 we give details related to the conjectured hardness of the different computational problems mentioned above.

4.1 Public parameters setting

From Subsection 2.2, we have ${\sigma_1}^{n \cdot 2^{(n \bmod 2)}} = \mathrm{Id}$ and ${\sigma_2}^p = \mathrm{Id}$ on $\mathcal{GL}_n(\mathbb{Z}_p)$. We suggest to set

$$Q = {\sigma_2}^{\mathsf{T}} \sigma_2,$$

which has order $O(p)$. Heuristically, we noticed there are Q's having order $p+1$ for some prime values of p. It seems the remaining cases satisfy Q has order divisible by $p-1$. We propose to sample U coming from a sample $Q' \leftarrow \mathcal{D}_s([Q])$ satisfying $Q' = U^{\mathsf{T}} Q U$. The order of U is expected to be $O(p^n)$, so it is easy to find a suitable unimodular matrix U.

Computational group action cost. Let a be a random positive integer smaller than order $N \approx p^n$ of U. Then, computing $Q_a = (U^a)^{\mathsf{T}} Q\,(U^a)$ requires to calculate $V = U^a$ and perform two matrix multiplications. A matrix multiplication approximately costs $O(n^{2.8074})$ field operations employing Strassen's algorithm. Now, raising U to a requires $2\log_2(N) \approx 2n\log_2(p)$ matrix multiplications for a constant-time implementation. In practice, we fix a security parameter λ and assume a is a random number of 2λ bits to reduce the matrix exponentiation cost from $2n\log_2(p)$ into 4λ matrix multiplications. Table 2 summarizes the costs and bits concerning our NIKE proposal.

5 Cryptanalysis

This section lists potential attacks on our LIKE proposal and discusses public-key validation by comparing it with CSIDH public-key validation.

	Private key	Public key	Shared secret
Runtime	$O(4\lambda n^{2.8074})$	$O(2n^{2.8074})$	$O(2n^{2.8074})$
Bitlength	$n^2 \log_2(p)$	$\frac{n(n+1)\log_2(p)}{2}$	$\frac{n(n+1)\log_2(p)}{2}$

Table 2: Assuming a is fixed, we set as private key the unimodular matrix U^a instead of a. We increase the private key size to make faster our group action. Public keys ($x_a = a * x$) and shared secrets ($b * x_a$) requires two matrix multiplications, and they correspond with symmetric matrix implying we can store a lesser number of coefficients than n^2. We assume a has exactly 2λ bits.

5.1 Key-recovery attacks

We present here the known approaches to perform a key-recovery attack to our protocol. The general setting is, given a public key $Q_a = (U^a)^{\mathsf{T}} Q\,(U^a)$, find either a or U^a. We present in this section the known classical and quantum approaches to perform this attack.

Bruteforce One computes $Q_c = U^{c\mathsf{T}} Q U^c$, for every $0 < c < 2^\lambda$. If $Q_a = Q_c$, then one sets $a = c$. The time complexity is $O(2^\lambda)$ and the space complexity is $\mathcal{O}(n^2)$.

Meet-in-the-Middle attack. One can perform a *meet-in-the-middle* style attack to retrieve the exponent a. The idea is to look for $a_1, a_2 < 2^{\lambda/2}$ such that $a = a_1 + a_2 2^{\lambda/2}$. Let $h_{\lambda/2} : \mathcal{S}_n^{>0} \to \{0,1\}^{\lambda/2}$ be a hash map. One stores the following table in memory as linked list

$$\mathcal{T} = \left\{ \left(h_{\lambda/2} \left((U^{c_1})^{\mathsf{T}} Q\,(U^{c_1}) \right), c_1 \right) : 0 \le c_1 < 2^{\lambda/2} \right\}.$$

Then one computes, for every $0 \le c_2 < 2^{\lambda/2}$, the binary string

$$h_{c_2} := h_{\lambda/2} \left(\left(U^{-c_2 2^{\lambda/2}} \right)^{\mathsf{T}} Q_a \left(U^{-c_2 2^{\lambda/2}} \right) \right).$$

If the memory cell of the table indexed h_{c_2} is not empty, one checks, for each corresponding element c_1, whether the following equation holds

$$\left(U^{-c_2 2^{\lambda/2}} \right)^{\mathsf{T}} Q_a \left(U^{-c_2 2^{\lambda/2}} \right) = (U^{c_1})^{\mathsf{T}} Q\,(U^{c_1}).$$

In case of success, then $a_1 = c_1$ and $a_2 = c_2$. Both space and time complexity required to perform such attack are of the order $O(2^{\lambda/2})$.

Memory-limited scenario. As mentioned before, the MitM procedure has a space complexity of $2^{\lambda/2}$ cells of memory. So, assuming we have a maximum number of memory cells w, we trade space for time by ranging over a w-subset in the table generation phase. At the same time, the enumeration phase will compare $2^{\lambda/2}$ elements vs. the w-subset to find a (possible) collision. If there is no collision, we repeat the procedure with the next w-subset, and so on until getting a collision. This memory-limited approach has a runtime of about

$$\frac{2^{\lambda/2}}{w}\left(w + 2^{\lambda/2}\right) = 2^{\lambda/2} + \frac{2^{\lambda}}{w} = O\left(\frac{2^{\lambda}}{w}\right) \text{ operations.}$$

On the other hand, van Oorschot and Wiener provided an algorithm to find such collision [46], which is unique over $\langle U \rangle$. More precisely, we can apply the golden collision search procedure at the cost of

$$\frac{2.5\sqrt{8\left(2^{\lambda/2}\right)^{3}}}{\sqrt{w}} \approx 7.2\frac{2^{3\lambda/2}}{\sqrt{w}} \text{ operations.}$$

That is, van Oorschot and Wiener procedure becomes cheaper than MitM when having a memory-limit; this is also the case for SIDH [1,21] and CSIDH [19].

Pohlig-Hellam attack. Let us assume we have a unimodular matrix $U \in \mathcal{GL}_n(\mathbb{Z}_p)$ of order $N = N_1 N_2$. Fix a public quadratic form Q. The idea is to emphasize the importance of commutativity and group structure in the operations to apply the Chinese Remainder Theorem (CRT) [5]. For example, finding s from $V_s = U^s$ reduces the problem into solving it in small subgroups of sizes N_1 and N_2 as follows:

$$V_s^{N_2} = \left(U^{N_2}\right)^s, \tag{1}$$

$$V_s^{N_1} = \left(U^{N_1}\right)^s. \tag{2}$$

Solving Equation 1 and Equation 2 give $s_1 = s \bmod N_1$ and finds $s_2 = s \bmod N_2$, respectively. Then, CRT reconstructs s in terms of s_1 and s_2. However, our construction keeps $V_s = U^s$ secret and makes public $Q_s = {V_s}^{\mathsf{T}} Q V_s$, which is crucial to argue why (we believe) CRT does not reduce security. Let us analyse the following two equations

$$Q_s^{N_2} = \left({V_s}^{\mathsf{T}} Q V_s\right)^{N_2}, \tag{3}$$

$$Q_s^{N_1} = \left({V_s}^{\mathsf{T}} Q V_s\right)^{N_1}. \tag{4}$$

Now, Q_s has a different order N_s than N and $Q_s^e \neq \left(V_s^{e\mathsf{T}} Q^e V_s^e\right)$ for $i = 1, 2$ and any $e \in \mathbb{Z} \setminus \{0, \pm 1\}$. It is worth highlighting, Q_s^e does not belong to $\mathcal{X}_{U,Q}$.

[5] This is also the case for CSIDH as in both cases we work on an unstructured set [17].

Furthermore, neither Equation 3 nor Equation 4 reduce the order of U, Q and Q_s at the same time. So, it seems there is no way to reduce into small subgroups of order dividing N.

Hidden-Shift Problem. One can retrieve the secret key a from the public key $Q_a = (U^a)^{\mathsf{T}} Q\,(U^a)$ by solving an Hidden Shift Problem instance (see Definition 10). Bonnetain and Schrottenloher provide in [12] concrete complexities of three algorithms for solving the HSP over $G = \mathbb{Z}_N$. We next list the main algorithms that solve the HSP, Table 3 presents their complexities:

1. A generic procedure relying on Kuperberg's algorithm [38].
2. An approach based on the Regev's work [49].
3. A revised Kuperberg's algorithm [39].

	Classical		Quantum	
	time	memory	memory	queries
[12, §3.2]	$1.8\sqrt{n} + 4.3$	$1.8\sqrt{n} + 2.3$	$1.8\sqrt{n} + 2.3$	$1.8\sqrt{n} + 4.3$
[12, §3.3]	$0.291n + \log_2(n) + 3$	$0.291n$	$\log_2(n)$	$2\log_2(n) + 3$
[12, §3.4]	$4\sqrt{2n/3} + \log_2(n) + 3$	$\sqrt{2n/3}$	$\log_2(n)$	$\sqrt{2n/3} + \log_2(n) + 3$

Table 3: Classical and Quantum complexity of algorithms that solve HSP [12, Table 4]

Solving sLIP approach. Another possible approach for a key recovery attack is to solve the underlying sLIP instance in the public key. Let $Q_a = (U^a)^{\mathsf{T}} Q\,(U^a)$ be a public key. Assume that, through the use of an LIP solver, one obtains U^a. This would be enough to retrieve the private shared key by multiplying U^a in both sides of the other's party public key as follows

$$Q_{ab} = (U^a)^{\mathsf{T}} Q_b\,(U^a) = \left(U^{a+b}\right)^{\mathsf{T}} Q\,\left(U^{a+b}\right).$$

However, it is not guaranteed that the returned unimodular matrix by the LIP solver would be exactly U^a. Indeed, any matrix of the form $W := VU^a$, where $V \in \mathrm{Aut}(Q)$ is a solution of the underlying LIP instance, and the probability of getting $W := U^a$ depends on the size of $\mathrm{Aut}(Q)$. Conservatively, we assume that given any solution W of the underlying LIP instance, it is easy to retrieve U^a.

Our public keys are constructed by sampling U coming from a sample $Q' \leftarrow \mathcal{D}_s([Q])$ where $Q' = U^{\mathsf{T}} Q U$, and then raising U to the power of a random integer a when computing Q_a. We were not able to find a reduction from these LIP case instances to the average-case or worst-case sLIP. Intuitively, raising U to a power amplifies the underlying Gaussian distribution. Furthermore, $\mathrm{ac}-\mathrm{sLIP}^Q_s$ instances can be seen as a particular case of our instances for $a = 1$. Given that an average-to-worst-case reduction exists ([24, Lemma 3.9]), we conjecture that these instances as computationally hard to solve (Conjecture 1).

Typically, algorithms for solving sLIP consist of enumerating short vectors. In this paper, we follow the complexity conjectured by Ducas and van Woerden [24, sec 7.3]. Define the primal-dual gap to the Gaussian Heuristic as

$$\mathsf{gap}(Q) = \max\left\{\frac{\mathsf{gh}(Q)}{\lambda_1(Q)}, \frac{\mathsf{gh}(Q^{-1})}{\lambda_1(Q^{-1})}\right\}.$$

For any class of quadratic forms $[Q]$ of dimension n such that $\mathsf{gap}(Q) \leq \mathsf{poly}(n)$, $\mathrm{wc}-\mathrm{sLIP}^Q$ is $2^{\Theta(n)}$-hard.

Note that retrieving U^a would allow us to solve the underlying CDHP-QF instance too. Indeed, one exploits the commutativity of the multiplicative group $\langle U\rangle$ to apply Shor's algorithm to the pair (U, U^a) and retrieve a.

5.2 Public-key validation: a comparison with CSIDH

Let assume we receive a public key $Q' = V^\mathsf{T} Q V$ as in Figure 2. Now, [24, Section 7] gives as fingerprint $\mathrm{ari}(Q) \coloneqq \{\det(Q), \gcd(Q), \mathrm{par}(Q), [Q]_{\mathbb{Q}}, \left([Q]_{\mathbb{Z}_p}\right)_p\}$ that ensure an efficient procedure to decide whether two quadratic forms cannot be equivalent. Now, we say Q' is a valid public key if there is an integer s such that $V = U^s$. An honest entity, let's say Alice, will share with us a valid public key Q', but if Alice is not honest, she could cheat us by sampling a random unimodular matrix V that does not belong to $\langle U\rangle$. Ideally, we need a public-key validation to check whether $V \in \langle U\rangle$. As far as we know, the unstructured set $\mathcal{X}$ does not leak information whenever $Q' = V^\mathsf{T} Q V$ with $V \notin \langle U\rangle$ is or not received. The non-commutativity of matrix multiplications implies different shared secret for $V \notin \langle U\rangle$.

In summary, we do not have public-key validation for our proposal, being a disadvantage compared with CSIDH-like instantiations [18,19,6]. However, we want to address that validating public keys on some recent large CSIDH-like instantiations is easy to "cheat". Let's start with the original CSIDH-like of 512 bits to hint at why there is an issue. The private keyspace has 256 bits. The public keyspace comprises all supersingular curves over a 512-bits prime field, which is a 256-bits set. So, validation falls to verifying supersingularity of curves.

Recently, Chávez-Saab et al. suggested to reduce the private keyspace from 256 to 221 bits to get a faster group action evaluation [19], which was also taken into consideration in [6]. The public keyspace remains the same, implying that $2^{256}/2^{221} = 2^{35}$ public keys do not come from a private key. We can easily select one of those 2^{35} keys by sampling outside the private-key range, and still correctly pass the key validation. That issue extrapolates to large CSIDH-like instantiations since the private keyspace has at most 512 bits, while the number of supersingular curves is thousands of bits (≥ 1024 bits). Consequently, we can generate fake-valid public keys with preimage out from the private-key space due to the significant difference in the size of public and private keyspaces. It is worth highlighting that even using fake-valid public keys ensures the same shared secret due to the commutativity of CSIDH-like schemes.

We do not think this (minor) issue on the public-key validation compromises security for CSIDH-like constructions. In fact, further analysis is required, and it is out of the scope of this work.

6 Experiments and implementation

This section focuses on the performance of our LIKE proposal. We provided a constant-time proof-of-concept of LIKE in the C-language. Given a security parameter λ, we implement

- schoolbook matrix multiplication at the cost of n^3 field multiplications;
- matrices exponentiation through a constant-time Montgomery ladder [36]. We assume integer exponents of 2λ-bits. Then, each matrix exponentiation has a cost of 4λ matrix multiplications; and
- the computation of $V^{\mathsf{T}}QV$ by calculating $R = QV$ (one matrix multiplication), and then $V^{\mathsf{T}}R$ utilizing the symmetric property (a saving of 50% concerning one matrix multiplication).

Since LIP is $2^{\Theta(n)}$-hard, we set as matrix dimension $n = \lambda$. To a fair comparison, we follow the suggestions in [19] concerning the group sizes; that is, We work with the same group sizes N as in [19] and conservatively to choose:

- LIKE-1: $N = \#\langle U\rangle$ of 2048-bits (equivalent to CSIDH-4096),
- LIKE-3: $N = \#\langle U\rangle$ of 3072-bits (equivalent to CSIDH-6144),
- LIKE-5: $N = \#\langle U\rangle$ of 4096-bits (equivalent to CSIDH-8192)

to address close NIST security Level 1, 3, and 5, respectively. Table 4 lists the cost of the revised Kuperberg's algorithm concerning our instantiations.

We set $q = 32771$ as a 16-bit prime number. Table 5 illustrates private and public keys sizes, and Table 6 draws our experiments. All our experiments were run on a machine with 2.70GHz Intel Core i7-7500U CPU, 16 GB of RAM and running Ubuntu 20.04. We used `gcc 9.4.0` and `clang 10.0.0`. We replicated the experiments of CITDH-512 [6] and CSIDH-4096 [19] by using their respective public repositories [6]. Our code is freely available at https://archive.org/details/like-c.

Advantages: Concerning key derivation, LIKE has better performance than any large CSIDH instantiation. For example, LIKE-1 is about 3.82x faster than CTIDH-512. Also LIKE running time (and also key sizes) increase by a linear factor; LIKE-3 is 3.40x slower than LIKE-3, while LIKE-5 is 8.2x slower than LIKE-1. Additionally, LIKE-1 key generation is about 2.16x faster than CSIDH-4096.

[6] We compared with the fastest CSIDH-style that use dummy operations and two torsion points [45]

	Classical		Quantum	
N	time	memory	memory	queries
256	63.256	13.064	8.000	24.064
2048	161.802	36.950	11.000	50.950
3072	195.604	45.255	11.585	59.840
4096	224.023	52.256	12.000	67.256

Table 4: Classical and Quantum complexity of the revised Kuperberg's algorithms to solve HSP over $\mathbb{Z}_N$ [12, §3.4]. All the entries are presented after taking $\log_2$. Recall, $N = \#\langle U \rangle$ for our case, and it is compared with a CSIDH-$(2\log_2 N)$ instance.

Group bitlength	Private key	Public Key	NIST security
2048	32.768 KB	16.512 KB	Level 1
3072	73.728 KB	37.056 KB	Level 3
4096	131.072 KB	65.792 KB	Level 5

Table 5: Sizes concerning the subgroup generated by U (with coefficients in $\mathbb{Z}_q$ being q a 16-bits prime number). Matrix dimensions are $n = 128$, $n = 192$, $n = 256$ for NIST security Level 1, 3, and 5, respectively. All group sizes are given in $\log_2$ base.

Scheme	Key generation	Key Agreement	NIST security
CTIDH-512 [6]	139.509609	144.022198	Level 1
CSIDH-4096 [19]	24184.504	24184.504	Level 1
LIKE-1	11197.063	37.713	Level 1
LIKE-3	57163.334	128.051	Level 3
LIKE-5	183199.131	307.322	Level 5

Table 6: Million of clock cycles. Matrix dimensions are $n = 128$, $n = 192$, $n = 256$ corresponding to LIKE-1, LIKE-3, and LIKE-5, respectively. Asymptotic speaking, CTIDH-512 has smaller quantum security according to [12,47,19].

Disadvantages: Public CSIDH-keys are smaller than public LIKE-keys. More precise, CSIDH-4096 has public keys of 512 bytes (LIKE-1 keys are 32.25 larger), CSIDH-6144 of 768 bytes (LIKE-3 keys are 48.25 larger), and CSIDH-8192 of 1024 bytes (LIKE-5 keys are 64.25 larger).

Remark 2. We point out that our implementation should be taken as a proof-of-concept to hint the performance of our LIKE proposal. We leave as future work an optimized constant-time implementation of LIKE protocol with the appropriated unimodular public unimodular U.

7 Conclusions

We presented a new efficient NIKE scheme based on lattice isomorphisms and group actions. The hardness assumption on which the security is based are analogues to the ones of other quantum-secure NIKE schemes such as CTIDH and CSIDH. Our non-optimized constant-time implementation shows a clear advantage in key derivation against these schemes.

Future Research Directions. Our implementation has a lot of room for improvements. For example, we use school-book matrix multiplication in key derivation. However, one can exploit the fact that quadratic forms are symmetric, and that the two other multiplicands are transpose to each other, in order to design a much more efficient dedicate algorithm. In addition, more efficient algorithms such as Strassen algorithm can be employed in combination with the above remark, and in key generation.

To make our scheme even more competitive, one would want to reduce key sizes. A deeper study on the hardness of solving the underlying LIP instance of our public keys could lead to a reduction of the matrix sizes. Another hypothetical approach could be to add a ring/module structure to LIP, analogously to what has been already done with Learning With Errors, and then extend it to LIKE. Finally, it remains to investigate about authenticated key exchange in the LIKE framework.

Acknowledgments. We thank Professor Damien Stehlé for his insightful comments on an early version of this work, and Samuel Jaques for his thoughts on quantum aspects related to this paper. Finally, we greatly thank Victor Mateu for his advices and support.

References

1. Adj, G., Cervantes-Vázquez, D., Chi-Domínguez, J., Menezes, A., Rodríguez-Henríquez, F.: On the Cost of Computing Isogenies Between Supersingular Elliptic Curves. In: Cid, C., Jr., M.J.J. (eds.) Selected Areas in Cryptography - SAC 2018 - 25th International Conference, Calgary, AB, Canada, August 15-17, 2018, Revised Selected Papers. Lecture Notes in Computer Science, vol. 11349, pp. 322–343. Springer (2018). https://doi.org/10.1007/978-3-030-10970-7_15, https://doi.org/10.1007/978-3-030-10970-7_15

2. Aguilar Melchor, C., Aragon, N., Barreto, P., Bettaieb, S., Bidoux, L., Blazy, O., Deneuville, J.C., Gaborit, P., Ghosh, S., Gueron, S., Güneysu, T., Misoczki, R., Persichetti, E., Sendrier, N., Tillich, J.P., Vasseur, V., Zémor, G.: BIKE: Bit Flipping Key Encapsulation. NIST Post-Quantum Cryptography Standardization Project (Round 3) (2020)
3. Aguilar Melchor, C., Aragon, N., Bettaieb, S., Bidoux, L., Blazy, O., Bos, J., Deneuville, J.C., Dion, A., Gaborit, P., Lacan, J., Persichetti, E., Robert, J.M., Véron, P., Zémor, G.: Hamming Quasi-Cyclic (HQC). NIST Post-Quantum Cryptography Standardization Project (Round 3) (2020)
4. Ajtai, M.: Generating hard instances of lattice problems (extended abstract). In: 28th ACM STOC. pp. 99–108. ACM Press (May 1996). https://doi.org/10.1145/237814.237838
5. Azarderakhsh, R., Campagna, M., Costello, C., Feo, L.D., Hess, B., Jalali, A., Jao, D., Koziel, B., LaMacchia, B., Longa, P., Naehrig, M., Pereira, G., Renes, J., Soukharev, V., Urbanik, D.: Supersingular isogeny key encapsulation. third round candidate of the nist's post-quantum cryptography standardization process (2020), https://sike.org/
6. Banegas, G., Bernstein, D.J., Campos, F., Chou, T., Lange, T., Meyer, M., Smith, B., Sotáková, J.: CTIDH: faster constant-time CSIDH. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2021**(4), 351–387 (2021). https://doi.org/10.46586/tches.v2021.i4.351-387, https://doi.org/10.46586/tches.v2021.i4.351-387
7. Berlekamp, E., McEliece, R., Van Tilborg, H.: On the inherent intractability of certain coding problems (corresp.). IEEE Transactions on Information Theory **24**(3), 384–386 (1978)
8. Beullens, W., Disson, L., Pedersen, R., Vercauteren, F.: CSI-RAShi: Distributed Key Generation for CSIDH. In: Cheon, J.H., Tillich, J. (eds.) Post-Quantum Cryptography - 12th International Workshop, PQCrypto 2021, Daejeon, South Korea, July 20-22, 2021, Proceedings. Lecture Notes in Computer Science, vol. 12841, pp. 257–276. Springer (2021). https://doi.org/10.1007/978-3-030-81293-5_14, https://doi.org/10.1007/978-3-030-81293-5_14
9. Beullens, W., Kleinjung, T., Vercauteren, F.: CSI-FiSh: Efficient Isogeny Based Signatures Through Class Group Computations. In: Galbraith, S.D., Moriai, S. (eds.) Advances in Cryptology - ASIACRYPT 2019 - 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8-12, 2019, Proceedings, Part I. Lecture Notes in Computer Science, vol. 11921, pp. 227–247. Springer (2019). https://doi.org/10.1007/978-3-030-34578-5_9, https://doi.org/10.1007/978-3-030-34578-5_9
10. Bidoux, L., Gaborit, P.: Shorter signatures from proofs of knowledge for the sd, mq, pkp and rsd problems (2022). https://doi.org/10.48550/ARXIV.2204.02915, https://arxiv.org/abs/2204.02915
11. Boneh, D., Sahai, A., Waters, B.: Functional encryption: Definitions and challenges. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 253–273. Springer, Heidelberg (Mar 2011). https://doi.org/10.1007/978-3-642-19571-6_16
12. Bonnetain, X., Schrottenloher, A.: Quantum Security Analysis of CSIDH. In: Canteaut, A., Ishai, Y. (eds.) Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part II. Lecture Notes in Computer Science, vol. 12106, pp. 493–522. Springer (2020). https://doi.org/10.1007/978-3-030-45724-2_17, https://doi.org/10.1007/978-3-030-45724-2_17

13. Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (Leveled) fully homomorphic encryption without bootstrapping. In: Goldwasser, S. (ed.) ITCS 2012. pp. 309–325. ACM (Jan 2012). https://doi.org/10.1145/2090236.2090262
14. Brakerski, Z., Langlois, A., Peikert, C., Regev, O., Stehlé, D.: Classical hardness of learning with errors. In: Boneh, D., Roughgarden, T., Feigenbaum, J. (eds.) 45th ACM STOC. pp. 575–584. ACM Press (Jun 2013). https://doi.org/10.1145/2488608.2488680
15. Brakerski, Z., Vaikuntanathan, V.: Efficient fully homomorphic encryption from (standard) LWE. In: Ostrovsky, R. (ed.) 52nd FOCS. pp. 97–106. IEEE Computer Society Press (Oct 2011). https://doi.org/10.1109/FOCS.2011.12
16. Cash, D., Kiltz, E., Shoup, V.: The twin Diffie-Hellman problem and applications. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 127–145. Springer, Heidelberg (Apr 2008). https://doi.org/10.1007/978-3-540-78967-3_8
17. Castryck, W., Lange, T., Martindale, C., Panny, L., Renes, J.: CSIDH: An efficient post-quantum commutative group action. In: Peyrin, T., Galbraith, S. (eds.) ASIACRYPT 2018, Part III. LNCS, vol. 11274, pp. 395–427. Springer, Heidelberg (Dec 2018). https://doi.org/10.1007/978-3-030-03332-3_15
18. Castryck, W., Lange, T., Martindale, C., Panny, L., Renes, J.: CSIDH: an efficient post-quantum commutative group action. In: Peyrin, T., Galbraith, S.D. (eds.) Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part III. Lecture Notes in Computer Science, vol. 11274, pp. 395–427. Springer (2018). https://doi.org/10.1007/978-3-030-03332-3_15, https://doi.org/10.1007/978-3-030-03332-3_15
19. Chávez-Saab, J., Chi-Domínguez, J., Jaques, S., Rodríguez-Henríquez, F.: The SQALE of CSIDH: sublinear Vélu quantum-resistant isogeny action with low exponents. J. Cryptogr. Eng. (2021). https://doi.org/10.1007/s13389-021-00271-w, https://doi.org/10.1007/s13389-021-00271-w, https://doi.org/10.1007/s13389-021-00271-w
20. Childs, A., Jao, D., Soukharev, V.: Constructing elliptic curve isogenies in quantum subexponential time. Journal of Mathematical Cryptology **8**(1), 1–29 (2014). https://doi.org/doi:10.1515/jmc-2012-0016, https://doi.org/10.1515/jmc-2012-0016
21. Costello, C., Longa, P., Naehrig, M., Renes, J., Virdia, F.: Improved Classical Cryptanalysis of SIKE in Practice. In: Kiayias, A., Kohlweiss, M., Wallden, P., Zikas, V. (eds.) Public-Key Cryptography - PKC 2020 - 23rd IACR International Conference on Practice and Theory of Public-Key Cryptography, Edinburgh, UK, May 4-7, 2020, Proceedings, Part II. Lecture Notes in Computer Science, vol. 12111, pp. 505–534. Springer (2020). https://doi.org/10.1007/978-3-030-45388-6_18, https://doi.org/10.1007/978-3-030-45388-6_18
22. Couveignes, J.M.: Hard homogeneous spaces. Cryptology ePrint Archive, Paper 2006/291 (2006), https://eprint.iacr.org/2006/291, https://eprint.iacr.org/2006/291
23. Diffie, W., Hellman, M.E.: New Directions in Cryptography. IEEE Transaction on Information Theory **22**(6), 644–654 (1976)
24. Ducas, L., van Woerden, W.: On the lattice isomorphism problem, quadratic forms, remarkable lattices, and cryptography. Cryptology ePrint Archive, Report 2021/1332 (2021), https://ia.cr/2021/1332
25. Ettinger, M., Høyer, P.: On quantum algorithms for noncommutative hidden subgroups. Adv. Appl. Math. **25**(3), 239–251 (2000). https://doi.org/10.1006/aama.2000.0699, https://doi.org/10.1006/aama.2000.0699

26. Feneuil, T., Joux, A., Rivain, M.: Syndrome decoding in the head: Shorter signatures from zero-knowledge proofs. Cryptology ePrint Archive, Paper 2022/188 (2022), https://eprint.iacr.org/2022/188, https://eprint.iacr.org/2022/188
27. Feo, L.D., Dobson, S., Galbraith, S.D., Zobernig, L.: SIDH Proof of Knowledge. IACR Cryptol. ePrint Arch. p. 1023 (2021), https://eprint.iacr.org/2021/1023
28. Feo, L.D., Jao, D., Plût, J.: Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. J. Mathematical Cryptology **8**(3), 209–247 (2014)
29. Feo, L.D., Kohel, D., Leroux, A., Petit, C., Wesolowski, B.: SQISign: Compact Post-quantum Signatures from Quaternions and Isogenies. In: Moriai, S., Wang, H. (eds.) Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part I. Lecture Notes in Computer Science, vol. 12491, pp. 64–93. Springer (2020). https://doi.org/10.1007/978-3-030-64837-4_3, https://doi.org/10.1007/978-3-030-64837-4_3
30. Freire, E.S.V., Hofheinz, D., Kiltz, E., Paterson, K.G.: Non-interactive key exchange. In: Kurosawa, K., Hanaoka, G. (eds.) PKC 2013. LNCS, vol. 7778, pp. 254–271. Springer, Heidelberg (Feb / Mar 2013). https://doi.org/10.1007/978-3-642-36362-7_17
31. Futorny, V., Grochow, J.A., Sergeichuk, V.V.: Wildness for tensors. Linear Algebra and its Applications **566**, 212–244 (2019). https://doi.org/https://doi.org/10.1016/j.laa.2018.12.022, https://www.sciencedirect.com/science/article/pii/S0024379518305937
32. Gabidulin, E.M.: Theory of codes with maximum rank distance. Problemy peredachi informatsii **21**(1), 3–16 (1985)
33. Gaborit, P., Zémor, G.: On the hardness of the decoding and the minimum distance problems for rank codes. IEEE Transactions on Information Theory **62**(12), 7245–7252 (2016)
34. Galbraith, S.D., Petit, C., Silva, J.: Identification Protocols and Signature Schemes Based on Supersingular Isogeny Problems. J. Cryptol. **33**(1), 130–175 (2020). https://doi.org/10.1007/s00145-019-09316-0, https://doi.org/10.1007/s00145-019-09316-0
35. Ji, Z., Qiao, Y., Song, F., Yun, A.: General Linear Group Action on Tensors: A Candidate for Post-quantum Cryptography. In: Hofheinz, D., Rosen, A. (eds.) Theory of Cryptography - 17th International Conference, TCC 2019, Nuremberg, Germany, December 1-5, 2019, Proceedings, Part I. Lecture Notes in Computer Science, vol. 11891, pp. 251–281. Springer (2019). https://doi.org/10.1007/978-3-030-36030-6_11, https://doi.org/10.1007/978-3-030-36030-6_11
36. Joye, M., Yen, S.: The montgomery powering ladder. In: Jr., B.S.K., Koç, Ç.K., Paar, C. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers. Lecture Notes in Computer Science, vol. 2523, pp. 291–302. Springer (2002). https://doi.org/10.1007/3-540-36400-5_22, https://doi.org/10.1007/3-540-36400-5_22
37. de Kock, B.B.: A non-interactive key exchange based on ring-learning with errors. Master's thesis, Eindhoven University of Technology (25 Jun 2018), accessed at: https://research.tue.nl/en/studentTheses/a-non-interactive-key-exchange-based-on-ring-learning-with-errors on 19 May 2021
38. Kuperberg, G.: A subexponential-time quantum algorithm for the dihedral hidden subgroup problem. SIAM J. Comput. **35**(1), 170–188 (2005). https://doi.org/10.1137/S0097539703436345, https://doi.org/10.1137/S0097539703436345

39. Kuperberg, G.: Another subexponential-time quantum algorithm for the dihedral hidden subgroup problem. In: Severini, S., Brandão, F.G.S.L. (eds.) 8th Conference on the Theory of Quantum Computation, Communication and Cryptography, TQC 2013, May 21-23, 2013, Guelph, Canada. LIPIcs, vol. 22, pp. 20–34. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2013). https://doi.org/10.4230/LIPIcs.TQC.2013.20, https://doi.org/10.4230/LIPIcs.TQC.2013.20
40. Leroux, A.: A New Isogeny Representation and Applications to Cryptography. IACR Cryptol. ePrint Arch. p. 1600 (2021), https://eprint.iacr.org/2021/1600
41. Lyubashevsky, V., Ducas, L., Kiltz, E., Lepoint, T., Schwabe, P., Seiler, G., Stehle, D., Bai, S.: CRYSTAL-DILITHIUM. NIST Post-Quantum Cryptography Standardization Project (Round 3) (2020)
42. Micciancio, D., Regev, O.: Worst-case to average-case reductions based on gaussian measures. SIAM Journal on Computing **37**(1), 267–302 (2007). https://doi.org/10.1137/S0097539705447360, https://doi.org/10.1137/S0097539705447360
43. Naehrig, M., Alkim, E., Bos, J., Ducas, L., Easterbrook, K., LaMacchia, B., Longa, P., Mironov, I., Nikolaenko, V., Peikert, C., Raghunathan, A., Stebila, D.: FrodoKEM: Learning with Errors Key Encapsulation. NIST Post-Quantum Cryptography Standardization Project (Round 3) (2020)
44. NIST: NIST Post-Quantum Cryptography Standardization Process. Third Round Candidates (2020), https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions
45. Onuki, H., Aikawa, Y., Yamazaki, T., Takagi, T.: (Short Paper) A Faster Constant-Time Algorithm of CSIDH Keeping Two Points. In: Attrapadung, N., Yagi, T. (eds.) Advances in Information and Computer Security - 14th International Workshop on Security, IWSEC 2019, Tokyo, Japan, August 28-30, 2019, Proceedings. Lecture Notes in Computer Science, vol. 11689, pp. 23–33. Springer (2019). https://doi.org/10.1007/978-3-030-26834-3_2, https://doi.org/10.1007/978-3-030-26834-3_2
46. van Oorschot, P.C., Wiener, M.J.: Parallel Collision Search with Cryptanalytic Applications. J. Cryptol. **12**(1), 1–28 (1999). https://doi.org/10.1007/PL00003816, https://doi.org/10.1007/PL00003816
47. Peikert, C.: He gives C-sieves on the CSIDH. In: Canteaut, A., Ishai, Y. (eds.) EUROCRYPT 2020, Part II. LNCS, vol. 12106, pp. 463–492. Springer, Heidelberg (May 2020). https://doi.org/10.1007/978-3-030-45724-2_16
48. Prest, T., Fouque, P.A., Hoffstein, J., Kirchner, P., Lyubashevsky, V., Pornin, T., Ricosset, T., Seiler, G., Whyte, W., Zhang, Z.: Falcon: Fast-Fourier Lattice-Based Compact Signatures over NTRU. NIST Post-Quantum Cryptography Standardization Project (Round 3) (2020)
49. Regev, O.: A subexponential time algorithm for the dihedral hidden subgroup problem with polynomial space. arXiv: Quantum Physics (2004)
50. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. J. ACM **56**(6) (Sep 2009). https://doi.org/10.1145/1568318.1568324, https://doi.org/10.1145/1568318.1568324
51. Sahai, A., Waters, B.R.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (May 2005). https://doi.org/10.1007/11426639_27
52. Schwabe, P., Avanzi, R., Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J.M., Seiler, G., Stehle, D.: CRYSTALS-KYBER. NIST Post-Quantum Cryptography Standardization Project (Round 3) (2020)

53. Shor, P.: Algorithms for Quantum Computation: Discrete Logarithms and Factoring. In: Proceedings 35th Annual Symposium on Foundations of Computer Science. pp. 124–134 (1994). https://doi.org/10.1109/SFCS.1994.365700
54. Stern, J.: A new identification scheme based on syndrome decoding. In: Annual International Cryptology Conference (CRYPTO) (1993)
55. Tang, G., Duong, D.H., Joux, A., Plantard, T., Qiao, Y., Susilo, W.: Practical Post-Quantum Signature Schemes from Isomorphism Problems of Trilinear Forms. IACR Cryptol. ePrint Arch. p. 267 (2022), available at https://eprint.iacr.org/2022/267
56. Trott, S.M.: A Pair of Generators for the Unimodular Group. Canadian Mathematical Bulletin **5**(3), 245–252 (1962). https://doi.org/10.4153/CMB-1962-024-x