Isogeny Based Cryptography
Quantum Money

Dintyala Rahul Bhardwaj

Supervised by Prof. Venkata Koppula and Prof. Surjeet Kour

1 Introduction

"We are in a race against time to deploy post-quantum cryptography before quantum computers arrive" - Bernstein and Lange

Post-quantum cryptography is essential because it addresses the imminent threat posed by quantum computers, which could potentially break widely used cryptographic algorithms like RSA (Rivest-Shamir-Adleman) and ECC (Elliptic Curve Cryptography). Quantum algorithms, such as Shor's algorithm, can efficiently factor large integers and compute discrete logarithms, undermining the security of current encryption methods. As advancements in quantum computing accelerate, transitioning to quantum-resistant algorithms is critical to safeguard sensitive data, ensure privacy, and maintain trust in digital communication systems. Without this transition, the integrity of financial transactions, personal communications, and national security could be at risk.

The focus of this project will be to study one of the prominent candidates in post-quantum cryptography, known as "isogeny-based cryptography."

Isogenies are morphisms in the category of elliptic curves. The foundation of isogeny-based cryptography traces back to the emergence of elliptic curve cryptography in the 1980s by Miller[Mil86] and Koblitz[Kob87], who proposed the integration of elliptic curves into the Diffie-Hellman key exchange protocol. In the early 2000s, the field witnessed significant advancements with the introduction of two pivotal concepts: pairing-based cryptography (PBCs) stemming from Joux's[Jou04] exploration of one-round tripartite DDH, and isogeny-based cryptography originating from the research efforts of Couveignes[Cou06], Teske[Tes06], Rostovtsev and Stolbunov [RS06]. Initially, isogeny-based cryptography lagged behind ECCs and PBCs until the late 2010s when the threat of quantum computers, capable of nullifying the latter, became apparent. Isogeny-based cryptography has demonstrated superior resilience against the cryptographic capabilities of quantum computers.

This project aims to find, via group actions, good abstractions and hard assumptions for isogenies (along the lines of the work done by Alamati et al. in [Ala+20]) as well as building cryptography on the basis of cryptographic group actions

2 Work So Far

The M.Tech Project Part I focused on reviewing cryptographic primitives built using group action assumptions.

We began with the seminal work by Couveignes [Cou06] which introduces the idea of hard homogeneous spaces created using regular group actions and uses isogenies to give a candidate hard homogeneous space. We moved on to more contemporary work on Cryptographic Group Actions by Alamati *et al.*[Ala+20], which introduces a framework based on group actions to simplify the use of various isogeny-based assumptions. This framework builds on the earlier works of Brassard and Yung, and Couveignes, by defining group actions with hardness assumptions tailored to isogeny-based constructions like CSIDH and CSI-FiSh. Alamati *et al.*[Ala+20] also demonstrate the versatility of their framework by using it to construct several cryptographic primitives that were not previously derived from isogeny-based assumptions. These include smooth projective hashing, dual-mode public-key encryption (PKE), two-message statistically sender-private oblivious transfer (OT), and Naor-Reingold style pseudorandom functions (PRFs).

As part of our work, we reviewed the work on a Trapdoor Claw-Free Functions from Group Actions by Alamati *et al.*in [AMR22]. This paper introduces a novel family of trapdoor claw-free functions (TCFs) based on isogeny-based group actions leveraging the extending LHS assumption and explores their potential applications in quantum cryptography.

A crucial aspect of constructing cryptographic primitives using generic group actions is ensuring the existence of a group action model in which the security proofs are valid. Consequently, much of our focus in the latter half of the semester has been dedicated to understanding these group action models. Specifically, we examined the frameworks proposed by Zhandry in [Zha24] and by Duman et al. in [Dum+23], analyzing their respective strengths and limitations.

Additionally, we explored the quantum money scheme introduced by Zhandry in [Zha24], as well as the concept of Quantum State Group Actions, developed in subsequent work by Mutreja and Zhandry in [MZ24b]. These investigations have provided valuable insights into the evolving interplay between group actions and quantum cryptographic protocols.

3 Hard Homogeneous Spaces [Cou06]

The objective of this section is to introduce group actions, for their own sake and as a means of constructing cryptographic primitives. We begin by defining group actions.

Definition 3.1 (Group Action). *A group $\mathbb{G}$ is said to act on a set $\mathcal{X}$ if there is a map $\star : \mathbb{G} \times \mathcal{X} \to \mathcal{X}$ that satisfies the following two properties :*

- *If e is the identity element of $\mathbb{G}$, then for any $x \in \mathcal{X}$, we have $e \star x = x$.*
- *For any $g, h \in \mathbb{G}$ and any $x \in \mathcal{X}$, we have $(gh) \star x = g \star (h \star x)$.*

◊

Based on the additional structure in the group action, the following definitions are given :

Definition 3.2. *A group action $(\mathbb{G}, \mathcal{X}, \star)$ is said to be :*

1. ***transitive** if for every $x_1, x_2 \in \mathcal{X}$, there exist a group element $g \in \mathbb{G}$ such that $x_2 = g \star x_1$. For such a transitive group action, the set $\mathcal{X}$ is called a homogeneous space for G.*
2. ***faithful** if for each group element $g \in \mathbb{G}$, either G is the identity element or there exists a set element $x \in \mathcal{X}$ such that $x \neq g \star x$.*
3. ***free** if for each group element $g \in \mathbb{G}$, g is the identity element if and only if there exists some set element $x \in \mathcal{X}$ such that $x = g \star x$.*
4. ***regular** if it is both free and transitive.*

◊

We concern ourselves with regular group actions. Regularity of a group action induces a natural bijection between $\mathbb{G}$ and $\mathcal{X}$, $g \mapsto g \star x$. So if $\mathbb{G}$ (or $\mathcal{X}$) is finite, $|\mathcal{X}| = |\mathbb{G}|$.

Let $\mathbb{G}$ be a commutative group, $\mathcal{X}$ be a set, and $\star$ be a reular group action. The for any $x_1, x_2 \in \mathcal{X}$, there exists a unique $g \in \mathbb{G}$ such that $x_2 = g \star x_1$. Borrowing the notation from [Cou06], let $g := \delta(x_2, x_1)$. We also have that $(\exists x \in \mathcal{X}, g \star x = x) \implies g = e$.

The process of utilizing a mathematical construct, such as group actions, in the design of cryptographic primitives necessitates that certain operations be computationally efficient for practical implementation, while others must be inherently difficult to guarantee intractability and security.

The following are *easy*:

1. For $\mathbb{G}$:
   (a) Given a string g, decide if it represents an element in $\mathbb{G}$.
   (b) Given strings g_1, g_2 representing two elements in $\mathbb{G}$, compute $g_1 g_2, g_1^{-1}$ and decide if $g_1 = g_2$.
   (c) Find a random element in $\mathbb{G}$ with uniform probability.
2. For $\mathcal{X}$:
   (a) Given a string x, decide if h represents an element in $\mathcal{X}$.
   (b) Given $x_1, x_2 \in \mathcal{X}$, decide if $x_1 = x_2$.
3. For $\star$: Given $g \in \mathbb{G}$ and $x \in \mathcal{X}$ compute $g \star x$.

Definition 3.3. *Let $\mathbb{G}$ be an abelian group and $(\mathbb{G}, \mathcal{X}, \star)$ be. Then the following problems are defined :*

1. ***Vectorisation Problem**: Given x_1 and $x_2 \in \mathcal{X}$, find $\delta(x_2, x_1)$.*
2. ***Parallelisation Problem**: Given x_1, x_2, and $x_3 \in \mathcal{X}$, find the unique x_4 such that $\delta(x_2, x_1) = \delta(x_4, x_3)$.*
3. ***Parallel Testing Problem**: Given x_1, x_2, x_3, and $x_4 \in \mathcal{X}$ decide whether $\delta(x_2, x_1) = \delta(x_4, x_3)$.*

◊

Remark 3.4. *Note that if vectorisation is easy, then so is parallelisation. If parallelisation is easy, then so is parallel testing. We have no reason to believe necessarily, the converse of either of these implications.* ◊

Definition 3.5 (Hard Homogeneous Space)**.** *A homogeneous space for which all the easy assumptions are true and vectorisation and parallelisation problems are hard is called a hard homogeneous space (HHS).* ◊

Definition 3.6 (Very Hard Homogeneous Space)**.** *A homogeneous space for which all the easy assumptions are true and Parallel Testing Problem is hard, is called a Very Hard Homogeneous Space (VHHS).* ◊

Example 3.7. *Let $\mathbb{G} = \langle g \rangle$ be a cyclic group of order n. Let $\mathrm{Aut}(\mathbb{G})$ be the set of automorphisms of $\mathbb{G}$. Note that any homomorphism ϕ from $\mathbb{G} \to \mathbb{G}$ is entirely described by $\phi(g)$. If $\phi(g) = g^k$. Then ϕ is an automorphism if $\{mk \pmod{n}\}_{k=1}^{n} = \mathbb{Z}_n$, which happens if and only if $\gcd(k, n) = 1$. Consider the map $\Psi : \mathrm{Aut}(\mathbb{G}) \to \mathbb{Z}_n^*$ that maps $(g \mapsto g^c) \mapsto c$. This map is clearly an isomorphism.*

Let $\mathfrak{g}$ be the set of generators of $\mathbb{G}$. Then $\mathrm{Aut}(\mathbb{G})$ *acts regularly on $\mathfrak{g}$ through the action* $\star : \mathrm{Aut}(\mathbb{G}) \times \mathfrak{g} \to \mathfrak{g}$, $((g \mapsto g^c), h) \mapsto h^c$.

$\Diamond$

3.1 *Key Exchange*

Consider the following Key-Exchange Protocol

Construction 3.8 (Key Exchange via. Hard Homogeneous Spaces)**.** *Let $\mathbb{G}$ be an abelian group and $\mathcal{X}$ be a hard homogeneous space for $\mathbb{G}$ with respect to the action $\star$. Alice and Bob use $(\mathbb{G}, \mathcal{X}, \star)$ to derive a shared key as follows :*

1. *Alice samples $x_0 \leftarrow \mathcal{X}$, $g_1 \leftarrow \mathbb{G}$ and sends $(x_0, x_1 = g_1 \star x_0)$ to Bob.*
2. *Bob samples $g_2 \leftarrow \mathbb{G}$ and sends $x_2 = g_2 \star x_0$ to Alice.*
3. *Alice computes $g_1 \star x_2$ and Bob computes $g_2 \star x_1$.*

$\Diamond$

The correctness for this procotol is clear since $g_1 \star x_2 = g_1 \star (g_2 \star x_0) = (g_1 g_2) \star x_0 = (g_2 g_1) \star x_0 = g_2 \star (g_1 \star x_0) = g_2 \star x_1$.

An eavesdropper can learn $(x_0, x_1 = g_1 \star x_0, x_2 = g_2 \star x_0)$. If an adversary $\mathcal{A}$ can solve the parallelization problem, then $\mathcal{A}(x_0, x_1, x_2) = x_3$ such that $\delta(x_3, x_2) = \delta(x_1, x_0) = g_1$. Since $x_3 = g_1 \star x_2 = g_1 \star (g_2 \star x_0) = (g_1 g_2) \star x_0$, which is the required key, this key-exchange protocol is only as safe as the parallelisation problem is hard. We capture this in the following definition.

Definition 3.9. *Let $\mathbb{G}$ be an abelian group, $\mathcal{X}$ be a homogeneous space for $\mathbb{G}$ and $(\mathbb{G}, \mathcal{X}, \star)$ be a regular group action. We say that the Decisional HHS problem is hard for $(\mathbb{G}, \mathcal{X}, \star)$ if, for any p.p.t. adversary $\mathcal{A}$, the following quantity is negligible:*

$$\left| \Pr\left[\begin{array}{c} x_0 \leftarrow \mathcal{X}, g_1, g_2 \leftarrow \mathbb{G} \\ 0 \leftarrow \mathcal{A}(x_0, g_1 \star x_0, g_2 \star x_0, (g_1 g_2) \star x_0) \end{array} \right] - \Pr\left[\begin{array}{c} x_0 \leftarrow \mathcal{X}, g_1, g_2, g_3 \leftarrow \mathbb{G} \\ 0 \leftarrow \mathcal{A}(x_0, g_1 \star x_0, g_2 \star x_0, g_3 \star x_0) \end{array} \right] \right|$$

$\Diamond$

4 Cryptographic Group Actions [Ala+20]

4.1 *Effective Group Action (EGA)*

Definition 4.1 (Effective Group Action)**.** *A group action* $(\mathbb{G}, \mathcal{X}, \star)$ *is* effective *if:*

1. *$\mathbb{G}$ is finite, with efficient algorithms for:*
   (a) ***Membership testing:*** *Check if a bit string represents a valid element in $\mathbb{G}$.*
   (b) ***Equality testing:*** *Check if two bit strings represent the same element in $\mathbb{G}$.*
   (c) ***Sampling:*** *Sample an element $g \in \mathbb{G}$.*
   (d) ***Inversion:*** *Compute g^{-1} for any $g \in \mathbb{G}$.*
2. *$\mathcal{X}$ is finite, with efficient algorithms for:*
   (a) ***Membership testing:*** *Check if a bit string represents an element in $\mathcal{X}$.*
   (b) ***Unique representation:*** *Compute a canonical representation $\hat{x}$ for any $x \in \mathcal{X}$.*
3. *A known distinguished element $x_0 \in \mathcal{X}$ (origin).*
4. *An efficient algorithm exists for computing $g \star x$ given any $g \in \mathbb{G}$ and $x \in \mathcal{X}$.*

◊

Definition 4.2 (One-Way Group Action)**.** *A group action* $(\mathbb{G}, \mathcal{X}, \star)$ *is a one-way if the family of efficiently computable functions* $\{f_x : \mathbb{G} \to \mathcal{X}\}_{x \in \mathcal{X}}$ *is one-way, where* $f_x : g \mapsto g \star x$. ◊

Definition 4.3 (Weak Unpredictable Group Action)**.** *A group action* $(\mathbb{G}, \mathcal{X}, \star)$ *is weakly unpredictable if the family of efficienly computable permutations* $\{\pi_g : \mathcal{X} \to \mathcal{X}\}_{g \in \mathbb{G}}$ *is* ◊

4.2 *Restricted Effective Group Action (REGA)*

In the previous section, we made the assumption (or hoped) that for *any* $g \in \mathbb{G}$ and $x \in \mathcal{X}$, computing $g \star x$ is *easy*.

The group $\mathbb{G}$, the homogeneous space $\mathcal{X}$, and the corresponding group action $\star$ we will be working with in the isogeny-based cryptography setting will be non-trivial. Evaluating the group action efficiently for all $g \in \mathbb{G}$ and $x \in \mathcal{X}$ will not be possible.

Since every group has a set of generators, if one can evaluate efficiently the group action for that set of generators, then as long as the exponents are polynomial in the security parameter, the group action can be evaluated efficiently.

[ADMP2020] capture this limitation through their definition of a *Restricted Effective Group Action* (REGA).

Definition 4.4 (Restricted Effective Group Action)**.** *Let $(\mathbb{G}, \mathcal{X}, \star)$ be a group action with a not-necessarily minimal generating set $\mathbf{g} = \{g_1, \dots, g_n\}$ and $G = \langle \mathbf{g} \rangle$. The action is said to be* $\mathbf{g}$*-restricted effective if:*

1. *$\mathbb{G}$ is finite, and $n = poly(\log |\mathbb{G}|)$.*
2. *$\mathcal{X}$ is finite, with efficient algorithms for:*
   (a) ***Membership testing:*** *Check if a bit string represents an element in $\mathcal{X}$.*
   (b) ***Unique representation:*** *Compute a canonical string $\hat{x}$ for any $x \in \mathcal{X}$.*
3. *A known distinguished element $x_0 \in \mathcal{X}$ (origin).*
4. *There exists an efficient algorithm that, given any $i \in [n]$ and a bit string representation of $x \in \mathcal{X}$, computes $g_i \star x$ and $g_i^{-1} \star x$.*

◊

4.3 *Known-Order Effective Group Action (KEGA)*

[Ala+20] extend the Effective Group Action (EGA) model by assuming the group structure of $\mathbb{G}$ is explicitly known. By "known order," we mean that the group $\mathbb{G}$ has a known set of generators $\mathbf{g} = \{g_1, \dots, g_n\}$ along with their corresponding orders $(m_1, \dots, m_n)$. This is equivalent to expressing $\mathbb{G}$ as a direct sum decomposition $\mathbb{G} \cong \bigoplus_{i=1}^n \mathbb{Z}_{m_i}$.

A special case of this model is when $\mathbb{G}$ is cyclic, meaning $\mathbb{G} = \langle g \rangle \cong \mathbb{Z}/m\mathbb{Z}$. We define the lattice $\mathcal{L} = \bigoplus_{i=1}^n m_i\mathbb{Z}$, and the map $\phi : \mathbb{Z}^n/\mathcal{L} \to \mathbb{G}$, where $(a_1, \dots, a_n) \mapsto \prod_{i=1}^n g_i^{a_i}$. This mapping is an effective isomorphism, and its inverse corresponds to solving a generalized discrete logarithm problem.

If $(\mathbb{G}, \mathcal{X}, \star)$ is an instance of the EGA, it can be shown that $(\mathbb{Z}^n/\mathcal{L}, \mathcal{X}, \star)$ is also an EGA via the isomorphism ϕ. Consequently, $\mathbb{Z}^n/\mathcal{L}$ serves as a standard representation of the group $\mathbb{G}$.

Definition 4.5 (Known-Order Effective Group Action (KEGA) Model)**.** *A Known-Order Effective Group Action (KEGA) is an EGA $(\mathbb{Z}^n/\mathcal{L}, \mathcal{X}, \star)$, where the lattice $\mathcal{L}$ is determined by the tuple $(m_1, \ldots, m_n)$, representing the orders of the generators.* $\Diamond$

Remark 4.6. *Since for an abelian group $\mathbb{G}$, Shor's Algorithm and its generalization precisely compute an isomorphism $\mathbb{G} \cong \bigoplus_{i=1}^{n} \mathbb{Z}_{m_i}$, KEGA and abelian EGA are quantumly equivalent.* $\Diamond$

5 Candidate TCFs from Group Actions [AMR22]

Definition 5.1. *(Extended LHS assumption). Let $(\mathbb{G}, \mathbb{X}, \star)$ be an EGA, and let $n > \log|\mathbb{G}| + \omega(\log \lambda)$ be an integer. We say that extended LHS assumption holds over $(\mathbb{G}, \mathbb{X}, \star)$ if for any $\ell = \mathrm{poly}(\lambda)$ the following holds:*

$$\left(\mathbf{M}_i, \mathbf{m}_i, \mathbf{x}_i^{(\beta)}, \mathbf{y}_i^{(\beta)}\right)_{i\in[\ell],\beta\in\{0,1\}} \overset{c}{\approx} \left(\mathbf{M}_i, \mathbf{m}_i, \mathbf{u}_i^{(\beta)}, \mathbf{u}_i'^{(\beta)}\right)_{i\in[\ell],\beta\in\{0,1\}},$$

where each of the terms above is distributed as

$$\begin{gathered}
\mathbf{w} \leftarrow \{0,1\}^n, \quad \mathbf{M}_i \leftarrow \mathbb{G}^{n\times n}, \quad \mathbf{m}_i \leftarrow \mathbb{G}^n, \quad \mathbf{x}_i^{(0)} \leftarrow \mathbb{X}^n, \\
\mathbf{t}_i \leftarrow \mathbb{G}^n, \quad \mathbf{u}_i^{(\beta)} \leftarrow \mathbb{X}^n, \quad \mathbf{u}_i^{(\beta)} \leftarrow \mathbb{X}^n, \quad (\beta \in \{0,1\}) \\
\mathbf{x}_i^{(1)} := [\mathbf{M}_i\mathbf{w}] \star \mathbf{x}_i^{(0)}, \quad \mathbf{y}_i^{(0)} := \mathbf{t}_i \star \mathbf{x}_i^{(0)}, \\
\mathbf{y}_i^{(1)} := [\mathbf{M}_i\mathbf{w} + \mathbf{m}_i \odot \mathbf{w}] \star \mathbf{y}_i^{(0)}.
\end{gathered}$$

$\Diamond$

Construction. Let n be the secret dimension of underlying extended LHS assumption, and let $B > 2n^3$ be an integer. We define a wTCF family as follows. Let $X = [B]^n$, and $Y = \left(\mathbb{X}^{2n}\right)^n$. Note that $X^n = ([B]^n)^n$ and Y will be the input and output space of our wTCF family, respectively. To generate a key-trapdoor pair, for each $i \in [n]$ and $\beta \in \{0,1\}$ sample

$$\mathbf{v}_i \leftarrow \{0,1\}^n, \quad \mathbf{M}_i^{(\beta)} \leftarrow \mathbb{G}^{n\times n}, \quad \mathbf{m}_i^{(\beta)} \leftarrow \mathbb{G}^n, \quad \mathbf{x}_i^{(0)} \leftarrow \mathbb{X}^n, \quad \mathbf{t}_i \leftarrow \mathbb{G}^n,$$

and set

$$\mathbf{x}_i^{(1)} := \left[\mathbf{M}_i^{(0)}\left(\mathbf{1} - \mathbf{v}_i\right) + \mathbf{M}_i^{(1)}\mathbf{v}_i\right] \star \mathbf{x}_i^{(0)}, \quad \mathbf{y}_i^{(0)} := \mathbf{t}_i \star \mathbf{x}_i^{(0)},$$

$$\mathbf{y}_i^{(1)} := \left[\mathbf{M}_i^{(0)}\left(\mathbf{1} - \mathbf{v}_i\right) + \mathbf{M}_i^{(1)}\mathbf{v}_i + \mathbf{m}_i^{(0)} \odot \left(\mathbf{1} - \mathbf{v}_i\right) + \mathbf{m}_i^{(1)} \odot \mathbf{v}_i\right] \star \mathbf{y}_i^{(0)}$$

where $\odot$ denotes component-wise product. Output (ek,td) where

$$\mathrm{td} = \left(\mathbf{v}_i, \mathbf{t}_i\right)_{i \in [n]}, \quad \mathrm{ek} \ = \left(\mathbf{M}_i^{(\beta)}, \mathbf{m}_i^{(\beta)}, \mathbf{x}_i^{(\beta)}, \mathbf{y}_i^{(\beta)}\right)_{i \in [n], \beta \in \{0,1\}}$$

To evaluate the function $f_{\mathrm{ek},b}$ on input $\left(\mathbf{s}_i\right)_{i \in [n]} \in \left([B]^n\right)^n$, output $\left(\overline{\mathbf{z}}_i, \mathbf{z}_i\right)$ for $i \in [n]$ where

$$\overline{\mathbf{z}}_i = \left[(1-b) \cdot \mathbf{M}_i^{(0)}\mathbf{1} + \left(\mathbf{M}_i^{(1)} - \mathbf{M}_i^{(0)}\right)\mathbf{s}_i\right] \star \mathbf{x}_i^{(b)}$$

$$\mathbf{z}_i = \left[(1-b) \cdot \mathbf{M}_i^{(0)}\mathbf{1} + \left(\mathbf{M}_i^{(1)} - \mathbf{M}_i^{(0)}\right)\mathbf{s}_i + (1-b) \cdot \mathbf{m}_i^{(0)} + \left(\mathbf{m}_i^{(1)} - \mathbf{m}_i^{(0)}\right) \odot \mathbf{s}_i\right] \star \mathbf{y}_i^{(b)}$$

To invert the function $f_{\mathrm{ek},b}$ on some value $\left(\overline{\mathbf{z}}_i, \mathbf{z}_i\right)_{i \in [n]}$, we recover each $\mathbf{s}_i$ (for $i \in [n]$) as follows. Observe that if $f_{\mathrm{ek},\,,b}\left(\left(\mathbf{s}_i\right)_{i \in [n]}\right) = \left(\overline{\mathbf{z}}_i, \mathbf{z}_i\right)_{i \in [n]}$ then the following relation holds for any $i \in [n]$:

$$\left(-\mathbf{t}_i - \mathbf{m}_i^{(0)}\right) \star \mathbf{z}_i = \left[\left(\mathbf{m}_i^{(1)} - \mathbf{m}_i^{(0)}\right) \odot \left(\mathbf{s}_i + b \cdot \mathbf{v}_i\right)\right] \star \overline{\mathbf{z}}_i.$$

Because the action is applied component-wise and each entry of $\mathbf{s}_i$ lies in $[B]$, one can recover each entry of $\mathbf{s}_i$ efficiently by a simple brute force, since both $\mathbf{v}_i$ and $\mathbf{t}_i$ are included in the trapdoor.

6 Quantum Money from Abelian Group Actions [Zha24]

7 Generic models for group actions [Dum+23]

8 Quantum State Group Actions [MZ24b]

9 Full quantum equivalence of group action DLog and CDH [MZ24a]

References

[Mil86] Victor S. Miller. "Use of Elliptic Curves in Cryptography". In: *Advances in Cryptology — CRYPTO '85 Proceedings*. Ed. by Hugh C. Williams. Berlin, Heidelberg: Springer Berlin Heidelberg, 1986, pp. 417–426. isbn: 978-3-540-39799-1.

[Kob87] Neal Koblitz. "Elliptic curve cryptosystems". In: *Mathematics of computation* 48.177 (1987), pp. 203–209.

[Jou04] Antoine Joux. "A one round protocol for tripartite Diffie–Hellman". In: *Journal of cryptology* 17.4 (2004), pp. 263–276.

[Cou06] Jean-Marc Couveignes. "Hard homogeneous spaces". In: *Cryptology ePrint Archive* (2006).

[RS06] Alexander Rostovtsev and Anton Stolbunov. "Public-key cryptosystem based on isogenies". In: *Cryptology ePrint Archive* (2006).

[Tes06] Edlyn Teske. "An elliptic curve trapdoor system". In: *Journal of cryptology* 19 (2006), pp. 115–133.

[Ala+20] Navid Alamati et al. "Cryptographic group actions and applications". In: *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 II 26*. Springer. 2020, pp. 411–439.

[AMR22] Navid Alamati, Giulio Malavolta, and Ahmadreza Rahimi. *Candidate Trapdoor Claw-Free Functions from Group Actions with Applications to Quantum Protocols*. Cryptology ePrint Archive, Paper 2022/1775. 2022. url: https://eprint.iacr.org/2022/1775.

[Dum+23] Julien Duman et al. “Generic models for group actions”. In: *IACR International Conference on Public-Key Cryptography*. Springer. 2023, pp. 406–435.

[MZ24a] Hart Montgomery and Mark Zhandry. “Full quantum equivalence of group action DLog and CDH, and more”. In: *Journal of Cryptology* 37.4 (2024), p. 39.

[MZ24b] Saachi Mutreja and Mark Zhandry. “Quantum state group actions”. In: *arXiv preprint arXiv:2410.08547* (2024).

[Zha24] Mark Zhandry. “Quantum Money from Abelian Group Actions”. en. In: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. DOI: 10.4230/LIPICS.ITCS.2024.101. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ITCS.2024.101.