# Quantum Money from Abelian Group Actions

**Mark Zhandry**
NTT Research, Sunnyvale, CA, USA

## Abstract

We give a construction of public key quantum money, and even a strengthened version called quantum lightning, from abelian group actions, which can in turn be constructed from suitable isogenies over elliptic curves. We prove security in the generic group model for group actions under a plausible computational assumption, and develop a general toolkit for proving quantum security in this model. Along the way, we explore knowledge assumptions and algebraic group actions in the quantum setting, finding significant limitations of these assumptions/models compared to generic group actions.

**2012 ACM Subject Classification** Theory of computation → Quantum complexity theory

**Keywords and phrases** Quantum Money, Cryptographic Group Actions, Isogenies

**Digital Object Identifier** 10.4230/LIPIcs.ITCS.2024.101

**Related Version** *Full Version*: `https://eprint.iacr.org/2023/1097/`

**Acknowledgements** We thank Hart Montgomery for many helpful discussions about isogenies.

## 1 Introduction

Quantum money, first envisioned by Wiesner [52], is a system of money where banknotes are quantum states. By the no-cloning theorem, such banknotes cannot be copied, leading to un-counterfeitable currency. A critical feature of quantum money, identified by [1], is *public verification*, allowing anyone to verify while only the mint can create new banknotes. Such public key quantum money is an important central object in the study of quantum protocols, but unfortunately convincing constructions have remained elusive. See Section 1.5 for a more thorough discussion of prior work in the area.

We construct public key quantum money from abelian group actions, which can be instantiated by suitable isogenies over ordinary elliptic curves. Group actions, and the isogenies they abstract, are one of the leading contenders for post-quantum secure cryptosystems. Our construction could plausibly even be quantum lightning, a strengthening of quantum money with additional applications. Our construction is arguably the first time group actions have been used to solve a classically-impossible cryptographic task that could not already be solved using other standard tools like LWE. Our construction is sketched in Section 1.1 below, and given in detail in Section 3.

While our main construction can be instantiated on a clean abelian group action – often referred to as an "effective group action" (EGA) – many isogeny-based group actions diverge from this convenient abstraction. We therefore provide an alternative candidate scheme which can be instantiated on so-called "restricted effective group actions" (REGAs); see the Full Version [57]. We prove the quantum lightning security of our protocols in the generic group action model – a black box model for group actions – assuming a new but natural strengthening of the discrete log assumption on group actions. Note that generic group actions cannot be used to give unconditional quantum hardness results, so some additional computational assumption is necessary. In order to prove our result, we develop a new toolkit for quantum generic group action proofs; see Section 4. We believe ours is the first proof of security in the quantum generic group action model.


© Mark Zhandry;
licensed under Creative Commons License CC-BY 4.0

Along the way, we explore knowledge assumptions and algebraic group actions in the quantum setting, finding significant limitations of these assumptions/models compared to generic group actions. Specifically, unlike the classical setting where knowledge assumptions typically hold unconditionally against generic attacks, we explain why such statements likely do not hold quantumly. In the specific case of group actions, we indeed show an efficient generic attack on an analog of the "knowledge of exponent" assumption. This potentially casts doubt on quantum knowledge assumptions in general. We do give a more complex definition that avoids our attack, but it is unclear if the assumption is sound and more analysis is needed. For completeness, we give an alternative proof of security for our construction under this new knowledge assumption, which avoids generic group actions.

We also discuss an algebraic model for group actions, which can be seen as a variant of the knowledge of exponent assumption. Unlike the classical setting where algebraic models live "between" the fully generic and standard models, we find that the algebraic group action model is likely incomparable to the generic group action model, and security proofs in the model are potentially problematic. As these issues do not appear for generic group actions, we therefore propose that generic group actions are the preferred quantum idealized model for analyzing cryptosystems, instead of the algebraic group action model as argued for in [23]. See the Full Version [57] for details.

In the Full Version [57], we include a discussion of possible generalizations. In particular, we propose the notion of a *quantum* group action where the set elements are quantum states instead of bit strings. We discuss how instantiating our scheme on quantum group actions is closely related to failed approaches for building quantum money from LWE, but different in key ways that seem to allow our scheme to remain secure while the related LWE approaches failed.

## 1.1 Our Construction

### 1.1.1 Abelian Group Actions

We will use additive group notation for abelian groups. An abelian group action consists of an abelian group $\mathbb{G}$ and a set $\mathcal{X}$, such that $\mathbb{G}$ "acts" on $\mathcal{X}$ through the binary relation $* : \mathbb{G} \times \mathcal{X} \to \mathcal{X}$ with the property that $g*(h*x) = (g+h)*x$ for all $g, h \in \mathbb{G}, x \in \mathcal{X}$. We will also assume a *regular* group action, which means that for every $x \in \mathcal{X}$, the map $g \mapsto g * x$ is a bijection.

The main group actions used in cryptography are those arising from isogenies over elliptic curves. For example, see [19, 46, 15, 9, 20]. Group action cryptosystems rely at a minimum on the assumed hardness of discrete logarithms: given $x, y = g * x \in \mathcal{X}$, finding $g$. For isogeny-based actions, this corresponds to the hard problem of computing isogenies between elliptic curves. Other hard problems are possible, such as analogs of computational/decisional Diffie-Hellman, and more.

### 1.1.2 The QFT

Our quantum money scheme will utilize the quantum Fourier transform (QFT) over general abelian groups. This is a quantum procedure that maps

$$|g\rangle \mapsto \frac{1}{\sqrt{|\mathbb{G}|}} \sum_{h \in \mathbb{G}} \chi(g,h)|h\rangle \ .$$

Here, $\chi$ is some potentially complex phase term. In the case of $\mathbb{G}$ being the additive group $\mathbb{Z}_N$, $\chi(g,h)$ is defined as $e^{i2\pi gh/N}$, with a slightly more complicated definition for non-cyclic

groups[1]. The main property we need from $\chi$ (besides making the QFT unitary) is that it is *bilinear*, in the sense that $\chi(g, h_1 + h_2) = \chi(g, h_1) \cdot \chi(g, h_2)$. It is also symmetric: $\chi(g, h) = \chi(h, g)$.

### 1.1.3 Our Quantum Money Scheme

Our quantum money scheme is as follows; see Section 3 for additional details.

- Gen: initialize a register in the state $\frac{1}{\sqrt{|\mathbb{G}|}} \sum_{g \in \mathbb{G}} |g\rangle$, which can be computed by applying the QFT to $|0\rangle$. Let $x \in \mathcal{X}$ be arbitrary. Then by computing the group action in superposition, compute $\frac{1}{\sqrt{|\mathbb{G}|}} \sum_{h \in \mathbb{G}} |g\rangle |g * x\rangle$. Next, apply the QFT over $\mathbb{G}$ to the first register. The result is:

$$\frac{1}{|\mathbb{G}|} \sum_{g,h \in \mathbb{G}} \chi(g, h) |h\rangle |g * x\rangle = \frac{1}{\sqrt{|\mathbb{G}|}} \sum_{h} |h\rangle |\mathbb{G}^h * x\rangle$$

  Here, $|\mathbb{G}^h * x\rangle$ is the state $\frac{1}{\sqrt{|\mathbb{G}|}} \sum_{g \in \mathbb{G}} \chi(g, h) |g * x\rangle$. Note that $|\mathbb{G}^h * x\rangle$ is, up to an overall phase, independent of $x$.
  Now measure $h$, in which case the second register collapses to $|\mathbb{G}^h * x\rangle$. Output $h$ as the serial number, and $|\mathbb{G}^h * x\rangle$ as the money state.
- To verify a banknote \$, we do the following[2]: Initialize a new register in the state $|\phi\rangle := \frac{1}{\sqrt{|\mathbb{G}|}} \sum_{u \in \mathbb{G}} |u\rangle$. Then apply the map $(u, y) \mapsto (u, (-u) * y)$ to the joint system $|\phi\rangle \times \$$ [3]. In the case where \$ is the honest banknote $|\mathbb{G}^h * x\rangle$, the result is

$$\begin{aligned}
\frac{1}{|\mathbb{G}|} \sum_{u \in \mathbb{G}} |u\rangle \sum_{g \in \mathbb{G}} \chi(g, h) |(g - u) * x\rangle &= \frac{1}{|\mathbb{G}|} \sum_{u \in \mathbb{G}} |u\rangle \sum_{g' \in \mathbb{G}} \chi(g' + u, h) |g' * x\rangle \\
&= \frac{1}{|\mathbb{G}|} \sum_{u \in \mathbb{G}} \chi(u, h) |u\rangle \sum_{g \in \mathbb{G}} \chi(g', h) |g' * x\rangle \\
&= \left( \frac{1}{\sqrt{|\mathbb{G}|}} \sum_{u \in \mathbb{G}} \chi(u, h) \right) |\mathbb{G}^h * x\rangle
\end{aligned}$$

  where we used the substitution $h' = g - u$. Thus we see that this process preserves the honest banknte state $|\mathbb{G}^h * x\rangle$. Moreover, if we apply the inverse QFT to the first register, the result for honest banknotes is $|h\rangle$, and for any state orthogonal to the honest banknote, the result of the inverse QFT will be something orthogonal to $|h\rangle$. Thus by measuring this register and checking if the result is $h$, we can distinguish the honest banknote state from any other state.

---

[1] Remember that the group operation is $+$, so $gh$ in the exponent is not the group operation, but instead multiplication in the ring $\mathbb{Z}_N$.

[2] In an initial version of this work, we had a more complicated verification. The simplified version here was pointed out to us by Jake Doliskani.

[3] Note that we used the "minimal" oracle here for the group action computation, having $(-u) * y$ replace $y$, instead of being written to a response register as in the standard quantum oracle. However, since the computation $y \mapsto (-u) * y$ is efficiently reversible (by $y \mapsto u * y$), we can easily implement the minimal oracle efficiently by first computing $|(-u) * y\rangle$, then uncomputing $|y\rangle$ using the efficient inverse, and finally swapping in $|(-u) * y\rangle$.

### 1.1.4 An instantiation using REGAs

For some isogeny-based group actions such as CSIDH [15], the operation $*$ is only efficiently computable for a very small set $S \subseteq \mathbb{G}$ of group elements. Such group actions are called "restricted effective group actions" (REGAs) [3]. Above, however, we see that we need to compute the group action on all possible elements in $\mathbb{G}$, both for minting and for verification. We therefore give a variant of the construction above which only uses the ability to compute $*$ for elements in $S$. We show that we are still able to sample $|\mathbb{G}^h * x\rangle$, but now the serial number has the form $\mathbf{A}^T h + \mathbf{e} \bmod N$ for a known matrix $\mathbf{A}$ and a "small" $\mathbf{e} \in \mathbb{Z}^n$ [4]. Under plausible assumptions, the serial number actually hides $h$ [5]. We nevertheless show that we can use such a noisy serial number for verification. For details, see the Full Version [57]. The security of our alternate scheme is essentially equivalent to the main scheme.

## 1.2 The security of our scheme

We do not know how to base the security of our schemes on any standard assumptions on isogenies. However, we are able to prove the security of our scheme in a black box model for group actions called the generic group action model (GGAM), an analog of the generic group model [48, 37] adapted to group actions. Generic models for group actions have been considered previously [38, 10, 39, 23]. While the model is motivated by post-quantum security, to the best of our knowledge ours is the first time the model has been used to actually prove security against quantum attacks.

The challenge with the quantum GGAM is that the query complexity of computing discrete logarithms is actually polynomial [24]. This means we cannot rely on query complexity alone to justify hardness, and must additionally make computational assumptions. This is in contrast to the classical setting, where the generic group (action) model allows for unconditional proofs of security by analyzing query complexity alone. In fact, most if not all generic group model proofs from the classical setting are unconditional query complexity proofs. This means that proofs in the quantum GGAM will look very different than classical proofs in the GGM/GGAM; in particular, proofs will require a reduction from the underlying hard problem. At the same time, in order to take advantage of the generic oracle setting, it would seem that quantum query complexity arguments are still needed. But a priori, it may not be obvious how to leverage query complexity in any useful way, given the preceding discussion.

### 1.2.1 Our Framework

In Section 4, we develop a new framework to help in the task of proving quantum hardness results relative to generic group actions. To illustrate our ideas, we consider the following warm-up task. An important feature in some isogeny-based group actions are twists, which allow for computing computing "negations": computing $(-g) * x$ from $g * x$. An interesting

[4] Here, we are interpreting $h$ a vector in $\mathbb{Z}_N^n$ for some $n, N$, which is possible since $\mathbb{G}$ is abelian.

[5] This is the search Learning with Errors (search LWE) problem [42] which is widely believed to be hard for *random* $\mathbf{A}$. In our case, $\mathbf{A}$ is a fixed matrix that depends on the group action, and LWE may or may not be hard for this $\mathbf{A}$. However, if LWE is easy for this $\mathbf{A}$, then we in fact have a plain group action. Indeed, a variant of Regev's quantum reduction between LWE and Short Integer Solution (SIS) [42], outlined by [54], shows that if LWE can be solved relative to $\mathbf{A}$, then SIS can be solved for $\mathbf{A}$ as well. It is straightforward to adapt this reduction to solve the Inhomogenous SIS (ISIS) problem, which then allows for computing the group action for all of $\mathbb{G}$. In this case we would have a clean group action and would not need this alternate construction.

question is whether this additional structure makes computing discrete logarithms easier. Here, we show that for generic group actions, such negations are unlikely to make discrete logarithms any easier than in group actions without negations. Concretely, we will show that discrete logarithms are generically hard, assuming a plausible computational assumption on some group action where such negation queries are *not* permitted.

Suppose toward contradiction that there was a generic adversary which could utilize negation queries to solve discrete logarithms. Let $(*, \mathbb{G}, \mathcal{X})$ be a plain group action where negation queries are not allowed. We will define a new group action $(\star, \mathbb{G}, \mathcal{X}')$ as follows. First sample a random injection $\Pi : \mathcal{X}^2 \to \{0,1\}^m$ whose inputs are *pairs* of set elements. Then define $\mathcal{X}'$ as the image under $\Pi$ of pairs of the form $(g * x, (-g) * x)$. $\star$ acts in the natural way: $g \star \Pi(y,z) = \Pi(g * y, (-g) * z)$.

Our reduction will sample a $\Pi$ [6] and run the generic adversary on the new group action, using its knowledge of $\Pi$ and its inverse to implement the action $\star$. Notice now that our reduction also has the ability to compute negations: given $\Pi(y,z)$ where $y = g * x$ and $z = (-g) * x$, the negation of $\Pi(y,z)$ is exactly the element $\Pi(z,y)$ obtained by swapping $y$ and $z$. Thus, our reduction is able to simulate the negation queries, even though the underlying group action does not support efficient negations. This is our main idea, though there are a couple lingering issues to sort out:

- The reduction cannot perfectly simulate $(\star, \mathbb{G}, \mathcal{X}')$. The issue is that there are elements $\Pi(y,z)$ where $y, z$ do not have the form $y = g * x, z = (-g) * x$ for some $g$. In the group action $(\star, \mathbb{G}, \mathcal{X}')$, these elements will be identified as invalid set elements. On the other hand, while our reduction can carry out the correct computation on $y, z$ of the correct form, it will be unable to distinguish such $y, z$ from ones of the incorrect form, and will act on these elements even though they are incorrect. As such, there will be elements that are not in $\mathcal{X}'$ that the reduction will nevertheless falsely identify as valid set elements. We resolve this problem by choosing the images of $\Pi$ to be somewhat sparse, by setting the output length $m$ sufficiently large. Our reduction only provides the adversary elements corresponding to valid $y, z$, and we can show, roughly, that the adversary has a negligible chance of computing elements in the image of $\Pi$ that correspond to invalid $y, z$. This follows from standard query complexity arguments. Thus, we are able to simulate with negligible error the correct group action $(\star, \mathbb{G}, \mathcal{X}')$.
- We have not yet specified what problem the reduction actually solves. The problem we would like to solve is the plain discrete logarithm on $(*, \mathbb{G}, \mathcal{X})$, where the reduction is given $g * x$, and must compute $g$. However, it is unclear what challenge the reduction should give to the adversary. The natural approach is to try to give the adversary $\Pi(g * x, (-g) * x)$, which is just the discrete log instance relative to $(\star, \mathbb{G}, \mathcal{X}')$ with the same solution $g$; the reduction can then simply output whatever the adversary outputs. However, this requires the reduction to know $(-g) * x$, which is presumably hard to compute given just $g * x$ (remember that negation queries are not allowed on $(*, \mathbb{G}, \mathcal{X})$). Our solution is to simply use a slight strengthening of discrete logarithms, where the adversary is given $(g * x, (-g) * x)$ and must compute $g$. Under the assumed hardness of this strengthened discrete log problem (again, in ordinary group actions where negations are presumed hard), we can complete the reduction and prove the generic hardness of discrete logarithms in the presence of negation queries.

[6] A random injection is exponentially large and cannot be sampled efficiently. Instead, the reduction will actually efficiently simulate a random injection $\Pi$ using known techniques. For the purposes of our discussion here, we can ignore this issue.

### 1.2.2 The security of our money scheme

We now turn to using our framework to prove the security of our quantum money scheme in the GGAM. Inspired by our negation example above, we will simulate a generic group action $(\star, \mathbb{G}, \mathcal{X}')$ using an injection $\Pi$ applied to a vector of set elements. Our goal will be to use two banknotes with the same serial number relative to $(\star, \mathbb{G}, \mathcal{X}')$ in order to break some distinguishing problem relative to $(*, \mathbb{G}, \mathcal{X})$. Any quantum money adversary yields such a pair of banknotes, and so if the distinguishing problem is hard, then there can be no such efficient quantum money adversary. This argument in fact shows the scheme attains the stronger notion of quantum lightning [54], which has additional applications.

Concretely, our starting assumption gives the adversary $y = u * x$ for a random $u$, and then allows the adversary a single quantum query to $z \mapsto v * z$ for an unknown $v$, where either $v$ is random or $v = 2u$. The adversary then has to tell whether $v = 2u$ or not. It is straightforward to prove this assumption is true in the classical GGAM. In fact, it is a quantum analog of the classical group-based problem of distinguishing $g^a, g^b$ from $g^a, g^{a^2}$ for a group generator $g$, a widely used Diffie-Hellman-like assumption. Under this analogy, $g$ plays the role of $x$, $a$ plays the role of $u$, and $b$ plays the role of $v$. The main difference from the classical assumption (besides being over group actions instead of groups) is that, instead of receiving $g^b$ or $g^{a^2}$, the adversary receives $h^b$ or $h^{a^2}$ for an adversarially chosen $h$.

Our idea is to have $\mathcal{X}'$ be elements of the form $\Pi(g * x, g * y)$ where $y = u * x$ is the challenge given by the assumption. Let $X = \Pi(x, y) \in \mathcal{X}'$. Now consider the output of a successful adversary, which is two copies of the banknote $|\mathbb{G}^h \star X\rangle$ relative to $(\star, \mathbb{G}, \mathcal{X}')$ for some serial number $h$. Now consider applying the following process to, say, the first copy: map any element $\Pi(z_1, z_2)$ in the range of $\Pi$ to $\Pi(z_2, v * z_1)$, where we compute $v * z_1$ from $z_1$ using the challenge oracle. We then observe that if $v = 2u$, this process preserves the banknote:

$$
\begin{aligned}
|\mathbb{G}^h \star X'\rangle &= \frac{1}{\sqrt{|\mathbb{G}|}} \sum_{g \in \mathbb{G}} \chi(g,h) |g \star \Pi(x,y)\rangle = \frac{1}{\sqrt{|\mathbb{G}|}} \sum_{g \in \mathbb{G}} \chi(g,h) |\Pi(g * x, g * y)\rangle \\
&\mapsto \frac{1}{\sqrt{|\mathbb{G}|}} \sum_{g \in \mathbb{G}} \chi(g,h) |\Pi(g * y, (g+2u) * x)\rangle \\
&= \frac{1}{\sqrt{|\mathbb{G}|}} \sum_{g \in \mathbb{G}} \chi(g,h) |\Pi((g+u) * x, (g+2u) * x)\rangle \\
&= \chi(-u,h) \frac{1}{\sqrt{|\mathbb{G}|}} \sum_{g' \in \mathbb{G}} \chi(g',h) |\Pi(g' * x, g' * y)\rangle = \chi(-u,h) |\mathbb{G}^h \star X'\rangle
\end{aligned}
$$

Above, we used the substitution $g' = g + u$.

On the other hand, if $v \neq 2u$, then the transformation will produce a state whose support is not even on $\mathcal{X}'$. In particular, the transformed state would be orthogonal to the original state. So our reduction will apply the above transformation to one copy of $|\mathbb{G}^h \star X\rangle$, leaving the other as is. Then it will perform the SWAP test on the two states. If $v = 2u$, the states will be identical and the SWAP test will accept. If $v \neq 2u$, the states will be orthogonal, and the swap test will accept only with probability $1/2$. Thus, we achieve a distinguishing advantage between the two cases, contradicting the assumption.

We believe our proof gives convincing evidence that our scheme should be secure on a suitable group action, perhaps even those based on isogenies over elliptic curves. However, our underlying assumption is new, and needs further cryptanalysis. One limitation of our assumption is that it is interactive, requiring a (quantum) oracle query to the challenger.

One may hope instead to use a non-interactive assumption. We do not know how to make non-interactive assumptions work, in general. In particular, if we do not have an oracle that can transform the input for us, it seems like we are limited to strategies that only permute the inputs to $\Pi$, like in our negation-query example. But since the scheme has to be efficient, the inputs to $\Pi$ can only consist of polynomial-length vectors of set elements. Any permutation on a polynomial-length set must have smooth order. On the other hand, the only permutations on $\mathcal{X}'$ which preserve $|\mathbb{G}^h \star X\rangle$ seem to have order that divides $|\mathbb{G}|$. Thus, if, say, the order of $\mathbb{G}$ were a large prime, it does not seem that permuting the inputs to $\Pi$ alone will be able to preserve $|\mathbb{G}^h \star X\rangle$.

## 1.3 On Knowledge Assumptions and Algebraic Group Actions

In the Full Version [57], we show a different approach to justifying the security of our scheme, by adapting certain knowledge assumptions [32] to the setting of group actions. Despite some high-level similarities to [32], the underlying details are somewhat different. The advantage of this route is that it gives a standard-model security proof (albeit, using a non-standard knowledge definition) rather than a generic model proof.

However, we find significant issues with using knowledge assumptions quantumly, that appear not to have been observed before. In particular, the straightforward way to adapt the knowledge assumptions of [32] to group actions actually results in *false* assumptions, as we demonstrate. Interestingly, our attack on the assumption is entirely generic. This is quite surprising, as in the classical setting, knowledge assumptions generally trivially hold against generic attacks.

Concretely, we show how to construct a superposition over $\mathcal{X}$ where the underlying discrete logarithms are hidden, even to the algorithm creating the superposition. To accomplish this, we observe that any set element $x$ can be seen as a superposition over all possible banknotes $|\mathbb{G}^h * x\rangle$; the superposition is uniform up to individual phases. Then we show a procedure to compute, given $|\mathbb{G}^h * x\rangle$, the serial number $h$. This allows us to apply individual phases to the various banknotes in the superposition. Certain phases will simply map $x$ to another set element $y$. But other phases will map $x$ to a uniform superposition (up to phases) over $\mathcal{X}$. Call this state $|\psi\rangle$.

Any meaningful knowledge assumption, and in particular the result of adapting [32] to group actions, would imply that if we were to measure $|\psi\rangle$ to get a set element $y$, then we must also "know" $g$ such that $y = g * x$. However, measuring $|\psi\rangle$ simply gives a uniform set element, importantly without any side information about $y$. As such, under the discrete log assumption, computing such a $g$ is hard.

We resolve this particular problem by re-framing knowledge assumptions as follows: instead of saying that any algorithm $A$ which produces a set element $y$ must know $g$ such that $y = g * x$, we say that for any such $A$ solving some task $T$, there is another algorithm $B$ that also solves $T$ such that $B$ knows $g$, even if $A$ would not. Thus, even if the original $A$ is constructed in such a way that it does not know $g$, at least $B$ does, and we can apply any security arguments to $B$ instead of $A$. We demonstrate that this assumption, together with an appropriate generalization of the discrete log assumption, are enough to prove the security of our scheme. However, we are somewhat skeptical of our new knowledge assumption, and it certainly needs more cryptanalysis.

### 1.3.1 Algebraic Group Actions

The Algebraic Group Model (AGM) [26] is an important classical model for studying group-based cryptosystems. It is considered a refinement of the generic group model, meaning that a proof in the model is "at least as" convincing as a proof in the generic group model[7], potentially even more convincing. A couple of recent works [23, 39] have considered the group action analog, the Algebraic Group Action Model (AGAM). Here, any time an adversary outputs a set element $y$, it must "explain" $y$ in terms of one of its input set elements $x_1, \ldots, x_n$ by providing a group element $g$ such that $y = g * x_i$.

The AGM can be seen as an idealized model version of the knowledge of exponent assumption, and likewise the AGAM can be seen as an idealized model version of an appropriate knowledge assumption on group actions. After all, a knowledge assumption would say that any time the adversary outputs a $y$, it must "know" how it derived $y$ from its inputs. The AGM/AGAM simply require the adversary to actually output this knowledge.

In the Full Version [57], we explore the AGAM in the presence of quantum attackers. We do not prove any formal results, but discuss why, unfortunately, the quantum AGAM appears problematic. For starters, given our attack on quantum knowledge assumptions, we are skeptical about the soundness of the quantum AGAM. In particular, our attack indicates that it is unlikely that the AGAM is a refinement of the generic group action model; rather they are likely incomparable.

Another problem we observe with the AGAM is that it requires the adversary to both solve some task, and also produce some extra information, namely the explanation $g$ of any output element $y$. Classically, if the adversary is able to both solve the task and produce this extra information (which would follow from an appropriate knowledge assumption), then the adversary can do both simultaneously, as required by the AGM/AGAM. However, quantumly, even if we believe the adversary can separately solve the task *or* produce the extra information (provided we believe the knowledge assumption), it may be impossible to do both simulteneously, as required by the AGAM.

This issue manifests in the following way: suppose the output is actually a superposition. Then the information $g$ will be entangled with the superposition, meaning the AGAM adversary's output will actually be a different state than if it did not output $g$. For example, if an AGAM adversary had to output a banknote $|\mathbb{G}^h * x\rangle$ (say, as part of the quantum money/lightning experiment), then if it also "explained" the banknote by outputting a group element $g$, the entanglement with $g$ would actually cause the banknote state to fail verification. It therefore unclear how to interpret such an adversary. Does it actually break the scheme, even if it does not pass verification? In the Full Version [57], we go into more details about this issue as well as pointing out several other issues with the AGAM.

We note that these issues are not present in the generic group action model. Thus, despite classically being a "worse" model than the algebraic model, we propose for the quantum setting that the generic group action model is actually *preferred* to the AGAM.

## 1.4 Further Discussion

In the Full Version [57], we generalize group actions to *quantum* group actions, which replace classical set elements with quantum states, but otherwise behave mostly the same as standard group actions. We give a simple quantum group action based on the Learning with Errors (LWE) problem [42], where we can actually prove that the discrete log problem is hard under

[7] There are some caveats to this classical claim; see [56] for discussion.

LWE. Despite this promising result, we expect that the LWE-based quantum group action will be of limited use. In particular, if we instantiate our quantum money construction over this group, the construction is *insecure*. The reason is that, in this group action, it is impossible to recognize the quantum states of the set. Our security proof crucially relies on such recognition in order to characterize states accepted by the verifier. Moreover, without recognition, there is an attack which fools the verifier with dishonest – and importantly, clonable – banknotes that are different from the honest ones, breaking security.

Interestingly, we explain that this failed instantiation is actually *equivalent* to a folklore approach toward building quantum money from lattices, which has been more-or-less shown impossible to make secure [33, 32]. The *only* missing piece in the folklore approach has been how to efficiently verify honest banknotes. Under our equivalence, this missing piece exactly maps to the problem of recognizing set elements in our quantum group action. For details, see the Full Version [57]. We believe this adds to the confidence of our proposal, since in group actions based on isogenies it is possible to recognize set elements, presumably without otherwise compromising hardness.

## 1.5 Related Work

### 1.5.1 Public key quantum money

In Wiesner's original scheme, the mint is required to verify banknotes, meaning the mint must be involved in any transaction. The involvement of the mint also leads to potential attacks [34]. Some partial solutions have been proposed, e.g. [6, 45]. The dream solution, however, is known as *public key* quantum money [1]. Here, anyone can verify the banknote, while only the mint can create them.

Unlike Wiesner's scheme which is well-understood, secure public key quantum money has remained elusive. While there have been many proposals for public key quantum money [1, 2, 25, 29, 54, 30, 31, 32], they mostly either (1) have been subsequently broken (e.g. [1, 2, 54, 31] which were broken by [35, 18, 44, 32]), or (2) rely on new cryptographic building blocks that have received little attention from the cryptographic community (e.g. [25, 29, 30] from problems on knots or quaternion algebras). The two exceptions are:

- Building on a suggestion of [7], [54] proved that quantum money can be built from post-quantum indistinguishability obfuscation (iO). iO has received considerable attention and even has a convincing *pre-quantum* instaniation [28]. Yet the post-quantum study of iO is much less thorough. While some post-quantum proposals have been made [27, 5, 13, 51], their post-quantum hardness is not well-understood.
- [32] construct quantum money from isogenies over super-singular elliptic curves. While isogenies have garnered significant attention from cryptographers, there is a crucial missing piece to their proposal: generating uniform superpositions over super-singular curves, which is currently unknown how to do. This is closely related to the major open question of obliviously sampling super-singular elliptic curves.

In light of the above, the existence of public key quantum money is largely considered open.

### 1.5.2 Cryptography from group actions and isogenies

Isogenies were first proposed for use in post-quantum cryptography by Couveignes [19] and Rostovtsev and Stolbunov [46]. Isogenies give a Diffie-Hellman-like structure, but importantly are immune to Shor's algorithm for discrete logarithms [47] due to a more restricted structure. This restricted structure, while helping preserve security against quantum attacks, also

makes the design of cryptosystems based on them more complex. Thus, significant effort has gone into building secure classical cryptosystems from isogenies and understanding their post-quantum security (e.g. [16, 21, 15, 9, 17, 22, 41, 11, 3, 4, 38, 36, 14, 10, 43]).

Certain isogenies such as the original proposals of [19, 46] as well as CSIDH and its variants [15, 20] can be abstracted as abelian group actions. However, many other isogenies (such as SIDH [21] and OSIDH [17]) cannot be abstracted as abelian group actions. Even among abelian group actions, we must distinguish between "effective group actions" (EGAs) and *restricted* EGAs (REGAs). The former satisfies the notion of a clean group action, whereas in the latter, the group action can only be efficiently computed for a certain small set of group elements. CSIDH could plausibly be a EGA at certain concrete security parameters, though asymptotically it only achieves quasi-polynomial security[8]. Our alternate construction also works on REGAs, which can plausibly be instantiated even asymptotically by CSIDH using a quantum computer[9].

While some non-isogeny abelian group actions have been proposed (e.g. [50]), currently all such examples have been broken (e.g. [49]). For this reason, group actions are largely considered synonymous with isogenies, though this may change if more secure group actions are found.

The vast majority of the isogeny and group action literature has focused on post-quantum cryptography – classical protocols that are immune to quantum attacks. To the best of our knowledge, only two prior works have used isogenies/group actions to build quantum protocols for tasks that are *impossible* classically. The first is [4], who build a proof of quantumness [12]. We note that proofs of quantumness can also be achieved under several "standard" cryptographic tools, such as LWE [12] or under certain assumptions on hash functions [53]. In contrast, no prior quantum money protocol could be based on similar standard building blocks. We also note that [4] currently has no known asymptotic instantiation with better-than-quasi-polynomial security, as it requires a clean group action (EGA). The second quantum protocol based on isogenies is that of [32], who build quantum money from walkable invariants, and propose an instantiation using isogenies over supersingular elliptic curves. However, such isogenies cannot be described as abelian group actions, and even more importantly their proposal is incomplete, as discussed above. Thus, ours is arguably the first application of group actions or isogenies to obtain classically impossible tasks that could not already be achieved under standard tools.

### 1.5.3 Relation to [32]

Aside from using isogenies, our construction has some conceptual similarities to [32], though also crucial differences that allow us to specify a complete protocol, and our idealized-model analysis is completely new. Here, we give a brief overview of the similarities and differences.

The walkable invariant framework of [32] is very general, but here we describe a special case of it that would apply to certain group actions, in order to illustrate the differences with our scheme. Consider a group action that is *not* regular, so that the set $\mathcal{X}$ is partitioned into many distinct orbits. For $x, y$ in the same orbit there will exist a unique $g$ such that

[8] With the state-of-the-art, evaluating CSIDH as an EGA would require time approximately $2^{\sqrt[3]{n}}$ on a quantum computer, while the best quantum attack is time $2^{\sqrt{n}}$. For a thorough discussion, see [40]. By setting $n = \log^3(\lambda)$, one gets polynomial-time evaluation and the best attack taking time $\lambda^{\sqrt{\log(\lambda)}}$.

[9] In order for CSIDH to be a REGA, one needs to compute the structure of the group. While this is hard classically, it is easy with a quantum computer using Shor's algorithm [47]. Since we always assume a quantum computer in this work, we can therefore treat CSIDH as a REGA.

$y = g * x$, but for $x, y$ in different orbits, there will not exist any group element mapping between them. We will also assume the ability to generate a uniform superposition over $\mathcal{X}$. We finally assume an "invariant", a unique label for each orbit which can be efficiently computed from any element in the orbit.

The minting process generates the uniform superposition over $\mathcal{X}$, and then measures the invariant, which becomes the serial number. The state then collapses to a uniform superposition over a single orbit, which becomes the banknote. This superposition can then be verified as follows. First check that the banknote has support on the right orbit by re-computing the invariant. Then check that the state is in uniform superposition by checking that the state is preserved under action by random group elements; this is accomplished using an analog of the swap test. [32] prove the security of their scheme under the certain assumptions which, when mapped to the group action setting above, correspond to the discrete log assumption and a knowledge assumption very similar to ours.

Unfortunately, there are no known instantiations of suitable group actions for their scheme. One possibility is to use the set of ordinary elliptic curves as the set, the number of points on the curve as the invariant, and orbits being sets of curves with the same number of points. Isogenies between curves are then the action[10], which do not change the number of points on the curve. The problem is that in general curves, it is not possible to efficiently compute the action, since the degree of the isogenies will be too high. The action *can* be computed on smooth-degree isogenies, but these are rare and there is no known way to compute a uniform superposition over curves supporting smooth-degree isogenies. For reasons we will not get into here, [32] propose using instead supersingular curves with non-smooth order, but again these are rare and there is no known way to generate a uniform superposition over such curves.

We resolve the issues with instantiating [32], without needing the ability to compute uniform superpositions over the set. Our key insight is that, if we can compute the group action efficiently (say because we are using isogenies of smooth degree), then this is enough to sample states that *are* uniform over a given orbit, except for certain phase terms: namely the states $|\mathbb{G}^h * x\rangle$ for uniform $h$. Then, rather than the serial number indicating which orbit we are in (which is now useless since we are in a single orbit), the serial number is a description of the phase terms, namely $h$.

## 2 Preliminaries

Here we give our notation and definitions. We assume the reader is familiar with the basics of quantum computation.

### 2.1 Quantum Fourier Transform over Abelian Groups

Let $\mathbb{G}$ be an abelian group, which we will denote additively. We here define our notation for the quantum Fourier transform over $\mathbb{G}$. Write $\mathbb{G} = \mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2} \times \mathbb{Z}_{n_k}$ where $\mathbb{Z}_{n_j}$ are the additive cyclic groups on $n_j$ elements, and associate elements $g \in \mathbb{G}$ with tuples $g = (g_1, \ldots, g_k)$ where $g_j \in \mathbb{Z}_{n_j}$. Then define $\chi : \mathbb{G}^2 \to \mathbb{C}$ by

$$\chi_{\mathbb{G}}(g, h) = \prod_{j=1}^{k} e^{i2\pi g_j h_j / n_j}$$

[10] It is not a proper group action since different orbits will be acted on by different groups.

Observe the following:

$$\chi_{\mathbb{G}}(g,h) = \chi_{\mathbb{G}}(h,g) \qquad \chi_{\mathbb{G}}(g_1+g_2,h) = \chi_{\mathbb{G}}(g_1,h) \times \chi_{\mathbb{G}}(g_2,h)$$

$$\chi_{\mathbb{G}}(-g,h) = \chi_{\mathbb{G}}(g,h)^{-1} \qquad \sum_{g\in\mathbb{G}} \chi_{\mathbb{G}}(g,h) = \begin{cases} |\mathbb{G}| & \text{if } h = 1_{\mathbb{G}} \\ 0 & \text{if } h \neq 1_{\mathbb{G}} \end{cases}$$

The quantum Fourier transform (QFT) over $\mathbb{G}$ is the unitary $\mathsf{QFT}_{\mathbb{G}}$ defined as

$$\mathsf{QFT}_{\mathbb{G}}|g\rangle = \frac{1}{\sqrt{|\mathbb{G}|}} \sum_{h\in\mathbb{G}} \chi(g,h)|h\rangle \ .$$

Observe that $\mathsf{QFT}_{\mathbb{G}} = \mathsf{QFT}_{\mathbb{Z}_{n_1}} \otimes \cdots \otimes \mathsf{QFT}_{\mathbb{Z}_{n_k}}$. Therefore, since the standard QFT corresponds to $\mathsf{QFT}_{\mathbb{Z}_{n_j}}$ and can be implemented efficiently, so can $\mathsf{QFT}_{\mathbb{G}}$.

From this point on, we will only work with a single group, so we will drop the sub-script and simply write $\chi(g,h), \mathsf{QFT}$, etc.

## 2.2 Quantum Money and Quantum Lightning

Here we define quantum money and quantum lightning. In the case of quantum money, we focus on *mini-schemes* [2], which are essentially the setting where there is only ever a single valid banknote produced by the mint. As shown in [2], such mini-schemes can be upgraded generically to full quantum money schemes using digital signatures.

### Syntax

Both quantum money mini-schemes and quantum lightning share the same syntax:

- $\mathsf{Gen}(1^\lambda)$ is a quantum polynomial-time (QPT) algorithm that takes as input the security parameter (written in unary) which samples a classical serial number $\sigma$ and quantum banknote \$.
- $\mathsf{Ver}(\sigma, \$)$ takes as input the serial number and a supposed banknote, and either accepts or rejects, denoted by 1 and 0 respectively.

### Correctness

Both quantum money mini-schemes and quantum lightning have the same correctness requirement, namely that valid banknotes produced by $\mathsf{Gen}$ are accepted by $\mathsf{Ver}$. Concretely, there exists a negligible function $\mathsf{negl}(\lambda)$ such that

$$\Pr[\mathsf{Ver}(\sigma,\$) = 1 : (\sigma,\$) \leftarrow \mathsf{Gen}(1^\lambda)] \geq 1 - \mathsf{negl}(\lambda) \ .$$

### Security

We now discuss the security requirements, which differ between quantum money and quantum lightning.

▶ **Definition 1.** *Consider a QPT adversary $\mathcal{A}$, which takes as input a serial number $\sigma$ and banknote \$, and outputs two potentially entangled states $\$_1, \$_2$, which it tries to pass off as two banknnotes.* $(\mathsf{Gen}, \mathsf{Ver})$ *is a secure* quantum money mini-scheme *if, for all such $\mathcal{A}$, there exists a negligible $\mathsf{negl}(\lambda)$ such that the following holds:*

$$\Pr\left[\mathsf{Ver}(\sigma,\$_1) = \mathsf{Ver}(\sigma,\$_2) = 1 : \begin{smallmatrix}(\sigma,\$)\leftarrow\mathsf{Gen}(1^\lambda)\\ (\$_1,\$_2)\leftarrow\mathcal{A}(\sigma,\$)\end{smallmatrix}\right] \leq \mathsf{negl}(\lambda) \ .$$

▶ **Definition 2.** *Consider a QPT adversary $\mathcal{B}$, which takes as input the security parameter $\lambda$, and outputs a serial number $\sigma$ and two potentially entangled states $\$_1, \$_2$, which it tries to pass off as two banknnotes.* $(\mathsf{Gen}, \mathsf{Ver})$ *is a secure* quantum lightning *scheme if, for all such $\mathcal{B}$, there exists a negligible* $\mathsf{negl}(\lambda)$ *such that the following holds:*

$$\Pr\left[\mathsf{Ver}(\sigma, \$_1) = \mathsf{Ver}(\sigma, \$_2) = 1 : (\sigma, \$_1, \$_2) \leftarrow \mathcal{B}(1^\lambda)\right] \leq \mathsf{negl}(\lambda) \ .$$

Quantum lightning trivially implies quantum money: any quantum money adversary $\mathcal{A}$ can be converted into a quantum lightning adversary $\mathcal{B}$ by having $\mathcal{B}$ run both $\mathsf{Gen}$ and $\mathcal{A}$. But quantum lightning is potentially stronger, as it means that even if the serial number is chosen adversarially, it remains hard to devise two valid banknotes. This in particular means there is some security against the mint, which yields a number of additional applications, as discussed by [54].

## 2.3 Group Actions

An (abelian) group action consists of a family of (abelian) groups $\mathbb{G} = (\mathbb{G}_\lambda)_\lambda$ (written additively), a family of sets $\mathcal{X} = (\mathcal{X}_\lambda)_\lambda$, and a binary operation $* : \mathbb{G}_\lambda \times \mathcal{X}_\lambda \to \mathcal{X}_\lambda$ satisfying the following properties:

- **Identity:** If $0 \in \mathbb{G}_\lambda$ is the identity element, then $0 * x = x$ for any $x \in \mathcal{X}_\lambda$.
- **Compatibility:** For all $g, h \in \mathbb{G}_\lambda$ and $x \in \mathcal{X}_\lambda$, $(g + h) * x = g * (h * x)$.

We will additionally require the following properties:

- **Efficiently computable:** There is a QPT procedure $\mathsf{Construct}$ which, on input $1^\lambda$, outputs a description of $\mathbb{G}_\lambda$ and an element $x_\lambda \in \mathcal{X}_\lambda$. The operation $*$ is also computable by a QPT algorithm.
- **Efficiently Recognizable:** There is a QPT procedure $\mathsf{Recog}$ which recognizes elements in $\mathcal{X}_\lambda$. That is, for any $\lambda$ and any string $y$ (not necessarily in $\mathcal{X}_\lambda$), $\mathsf{Recog}(1^\lambda, y)$ accepts $y$ with overwhelming probability if $y \in \mathcal{X}_\lambda$, and rejects with overwhelming probability if $y \notin \mathcal{X}_\lambda$.
- **Regular:** For every $y \in \mathcal{X}_\lambda$, there is exactly one $g \in \mathbb{G}_\lambda$ such that $y = g * x_\lambda$.

### 2.3.1 Cryptographic group actions

At a minimum, a cryptographically useful group action will satisfy the discrete log assumption:

▶ **Assumption 3.** The *discrete log assumption* (DLog) holds on a group action $(\mathbb{G}, \mathcal{X}, *)$ if, for all QPT adversaries $\mathcal{A}$, there exists a negligible $\lambda$ such that

$$\Pr[\mathcal{A}(g * x_\lambda) = g : g \leftarrow \mathbb{G}_\lambda] \leq \mathsf{negl}(\lambda) \ .$$

# 3 Our Quantum Lightning Scheme

Here, we give our basic quantum lightning construction, which assumes a cryptographic group action.

▶ **Construction 4.** Let $\mathsf{Gen}, \mathsf{Ver}$ be the following QPT procedures:

- $\mathsf{Gen}(1^\lambda)$: Initialize quantum registers $\mathcal{S}$ (for serial number) and $\mathcal{M}$ (for money) to states $|0\rangle_\mathcal{S}$ and $|0\rangle_\mathcal{M}$, respectively. Then do the following:
  - Apply $\mathsf{QFT}_{\mathbb{G}_\lambda}$ to $\mathcal{S}$, yielding the joint state $\frac{1}{\sqrt{|\mathbb{G}_\lambda|}} \sum_{g \in \mathbb{G}_\lambda} |g\rangle_\mathcal{S} |0\rangle_\mathcal{M}$.

- Apply in superposition the map $|g\rangle_{\mathcal{S}}|y\rangle_{\mathcal{M}} \mapsto |g\rangle_{\mathcal{S}}|y \oplus (g*x_\lambda)\rangle_{\mathcal{M}}$. Here, $x_\lambda$ is an arbitrary set element. The joint state of the system $\mathcal{S}\otimes\mathcal{M}$ is then $\frac{1}{\sqrt{|\mathbb{G}_\lambda|}}\sum_{g\in\mathbb{G}_\lambda}|g\rangle_{\mathcal{S}}|g*x_\lambda\rangle_{\mathcal{M}}$.
- Apply $\mathsf{QFT}_{\mathbb{G}_\lambda}$ to $\mathcal{S}$ again, yielding $\frac{1}{|\mathbb{G}_\lambda|}\sum_{g,h\in\mathbb{G}_\lambda}\chi(g,h)|h\rangle_{\mathcal{S}}|g*x_\lambda\rangle_{\mathcal{M}}$
- Measure $\mathcal{S}$, giving the serial number $\sigma := h$. The $\mathcal{M}$ register then collapses to the banknote $\$ = |\mathbb{G}_\lambda^h * x_\lambda\rangle := \frac{1}{\sqrt{|\mathcal{G}_\lambda|}}\sum_{g\in\mathbb{G}_\lambda}\chi(g,h)|g*x_\lambda\rangle_{\mathcal{M}}$. Output $(\sigma,\$)$.

- $\mathsf{Ver}(\sigma,\$)$ : First verify that the support of $\$$ is contained in $\mathcal{X}_\lambda$, by applying the assumed algorithm for recognizing $\mathcal{X}_\lambda$ in superposition. Then do the following:
  - Initialize a new register $\mathcal{H}$ to $\frac{1}{\sqrt{|\mathbb{G}_\lambda|}}\sum_{u\in\mathbb{G}_\lambda}|u\rangle_{\mathcal{H}}$
  - Apply in superposition the map $|u\rangle_{\mathcal{H}}|y\rangle_{\mathcal{M}} \mapsto |u\rangle_{\mathcal{S}}|(-u)*y\rangle_{\mathcal{M}}$ [11].
  - Apply $\mathsf{QFT}^{-1}_{\mathbb{G}_\lambda}$ to $\mathcal{H}$.
  - Measure $\mathcal{H}$, obtaining a group element $h'$. Accept if and only if $h' = h$.

## 3.1 Accepting States of the Verifier

Above we showed that honest banknote states are accepted by the verifier. The following stronger statement also holds, which is proved in the Full Version [57].

▶ **Theorem 5.** *Let $|\psi\rangle$ be a state over $\mathcal{M}$. Then $\Pr[\mathsf{Ver}(h,|\psi\rangle) = 1] = \|\langle\psi|\mathbb{G}_\lambda^h * x_\lambda\rangle\|^2$. Moreover, if verification accepts, the resulting state is exactly $|\mathbb{G}_\lambda^h * x_\lambda\rangle$.*

In other words, we can treat $\mathsf{Ver}(h,|\psi\rangle)$ as projecting exactly onto $|\mathbb{G}_\lambda^h * x_\lambda\rangle$.

## 3.2 Computing the Serial Number

Here, we show that, given a valid banknote $\$ = |\mathbb{G}_\lambda^h * x_\lambda\rangle$ with unknown serial number $h$, it is possible to efficiently compute $h$. This result is not needed for understanding the construction or its security, but is used in the Full Version [57] to break a certain natural knowledge assumption.

▶ **Theorem 6.** *There exists a QPT algorithm* Findh *such that, on input $|\mathbb{G}_\lambda^h * x_\lambda\rangle$, outputs $h$ with probability* 1.

**Proof.** We originally had a much more complicated algorithm Findh (and one that had a negligible correctness error). We thank Jake Doliskani for pointing our a much simpler version using phase kickback.

Indeed, by simply modifying Ver to output the measurement result $h'$ instead of testing whether or not $h' = h$, we immediately obtain such a Findh. The proof of Theorem 5 readily adapts to show that Findh indeed outputs $h$ with probability 1 for the input state $|\mathbb{G}_\lambda^h * x_\lambda\rangle$. ◀

# 4 A Quantum Toolkit for Generic Group Actions

Here, we recall a definition of the generic group action model (GGAM), and show how to use it to give quantum security proofs.

[11] Note that we used the "minimal" oracle here for the group action computation, having $(-u)*y$ replace $y$, instead of being written to a response register as in the standard quantum oracle. However, since the computation $y \mapsto (-u)*y$ is efficiently reversible (by $y \mapsto u*y$), we can easily implement the minimal oracle efficiently by first computing $|(-u)*y\rangle_{\mathcal{M}'}$ in a new register $\mathcal{M}'$, then uncomputing $|y\rangle_{\mathcal{M}}$ using the efficient inverse (so it now contains $|0\rangle_{\mathcal{M}}$), and finally swapping $\mathcal{M}'$ with $\mathcal{M}$.

## 4.1 A Shoup-style generic group action

There have been several different proposals for how to define generic group actions [38, 23, 10, 39]. Here, we briefly give a definition in the style of Shoup [48]. To help disambiguate between the different models, we will adapt terminology from [56] and refer to ours as the *Random Set Representation* model.

We first fix a (family of) groups $\mathbb{G} = (\mathbb{G}_\lambda)_\lambda$. We also fix a length function $m : \mathbb{Z} \to \mathbb{Z}$ with the property that $m(\lambda) \geq \log_2 |\mathbb{G}_\lambda|$. We call $m$ the *label length*. In this model, for a given security parameter $\lambda$, a random injection $L : \mathbb{G}_\lambda \to \{0,1\}^m$ is chosen, where $m = m(\lambda)$. Think of $L(g)$ as representing $g * x_\lambda$; we call $L$ the labeling function. $\mathcal{X}_\lambda$ will then be the image of $\mathbb{G}_\lambda$ under$L$. All parties are then given the following:

- As input, all parties receive the string $L(0)$, where $0 \in \mathbb{G}_\lambda$ is the identity. $L(0)$ represents $x_\lambda$.
- All parties can then make "group action" queries. For classical algorithms, such a query takes the form $(\ell, g) \in \{0,1\}^m \times \mathbb{G}_\lambda$. The response to the query is $L(g + L^{-1}(\ell))$; if $\ell$ is not in the image of $L$, then the response to the query is $\perp$. For quantum algorithms, we follow the usual convention for modeling superposition queries to classical functions, and have the query perform the map:

$$\sum_{\ell,g,\ell'} \alpha_{\ell,g,\ell'} |\ell, g, \ell'\rangle \mapsto \sum_{\ell,g,\ell'} \alpha_{\ell,g,\ell'} |\ell, g, \ell' \oplus L(g + L^{-1}(\ell))\rangle$$

  The set $\mathcal{X}_\lambda$ will be interpreted as the image of $\mathbb{G}_\lambda$ under $L$. Note that group action queries allow for testing membership in $\mathcal{X}_\lambda$: $\mathcal{X}_\lambda$ are exactly the set of strings where the group action query does not output $\perp$.

We call the oracle above $\mathsf{GGAM}_{\mathbb{G},m}$. In the classical setting, we usually consider queries to the oracle to have unit cost while computation outside the oracle queries is free. If following this convention, our model essentially corresponds to the model considered in [23]. However, in the quantum setting, considering the query complexity alone is insufficient, as discrete logarithms can be solved in polynomial query complexity [24]. Therefore, it is necessary to consider the total cost of an algorithm as including both the queries (unit cost per query) and the computation outside the queries.

### 4.1.1 On other Styles of Generic Group Actions

Other styles of generic group action are possible, and have been considered in the literature (e.g. [38, 10, 39]). In the Full Version [57], we explain why these models all come with limitations that seem to not apply to our model. In particular, based on our current understanding, the Random Set Representation model defined above seems to be "at least as good" as any other model for group actions in the quantum setting, and may in fact be "better" than the other models. For this reason, we focus on the Random Set Representation model. We leave exploring the exact relationship between the models as an interesting open question.

### 4.1.2 Algebraic Group Action Model

In the Full Version, we also consider a different idealized model called the Algebraic Group Action Model, the quantum and group action version of the classical Algebraic Group Model (AGM) [26]. In the classical world, this model is "between" the Type Safe model and the standard model, in the sense that security in the algebraic model implies security in the Type Safe model (which in turn often implies security in the Random Representation model,

per [56]). However, in the Full Version, we explain that the quantum analog of this model is actually problematic, and the proof of "between-ness" does not hold quantumly, for similar reasons as to why the Random Set Representation model appears superior to the other generic group action models quantumly. As such, it seems that the (Random Representation) generic group action model actually *better* captures available attacks than the algebraic group action model.

## 4.2 Our Framework for Quantum GGAM Security Proofs

### 4.2.1 Challenges with the quantum GGAM

The problem with the quantum GGAM, as observed by [23], is that we cannot hope for unconditional security results, as the discrete logarithm is easy if we only count quantum query complexity. [23] take the approach of instead considering the Algebraic Group Action Model (AGAM). We discuss the pitfalls of this approach in the Full Version [57]. Here we instead observe that we can recover a meaningful model by counting both queries and computational cost. However, because we cannot hope to prove unconditional query complexity lower bounds, we must instead resort to making computational assumptions and giving reduction-style arguments. This means arguments in the quantum GGAM will look very different that proofs in the classical GGM. To the best of our knowledge, there have been no prior security proofs in the quantum GGAM. We therefore develop some new tools and techniques for giving such proofs, including a proof of security of our quantum money scheme.

### 4.2.2 Our Abstract Framework

We first give a very abstract framework, which we will then apply the framework to the GGAM.

Let $\mathcal{Y}$ be a set, and $\mathcal{F}$ be a family of functions $f : \mathcal{Y} \to \mathcal{Y}$. Let $y_0 \in \mathcal{Y}$ be a specific starting element in $\mathcal{Y}$. Consider a random injection $L : \mathcal{Y} \to \{0,1\}^{m'}$, and consider the oracle $\mathcal{O}$ which maps $\mathcal{O}(L(y), f) = L(f(y))$; $\mathcal{O}$ outputs $\perp$ on any string that is not in the image of $L$. We will give the adversary $L(y_0)$ and also superposition access to $\mathcal{O}$.

Now consider a set $\mathcal{Y}' \subset \{0,1\}^s$, and suppose we have a not-necessarily-random injection $\Gamma : \mathcal{Y} \to \mathcal{Y}'$ (meaning $s \geq |\mathcal{Y}|$). We also have a procedure $P$ which is able to map $P(\Gamma(y), f) = \Gamma(f(y))$. However, unlike the oracle $\mathcal{O}$ considered above, this procedure $P$ may output value other than $\perp$ when given inputs that are not in the image of $\Gamma$. Our goal is to, nevertheless, simulate $\mathcal{O}$ using $P$.

Concretely, we will choose a random injection $\Pi : \{0,1\}^s \to \{0,1\}^{m'}$, and simulate $\mathcal{O}$ with the oracle $\mathcal{O}'(\Pi(z), f) = \Pi(P(z, f))$; $\mathcal{O}'$ will output $\perp$ on any input not in the image of $\Pi$. We will then give the adversary $\Pi(\Gamma(y_0))$, ad quantum query access to $\mathcal{O}'$.

### 4.2.3 Application to the GGAM

In our case, we will have $\mathcal{Y}$ be a group $\mathbb{G}_\lambda$. $\mathcal{F}$ will include for each $h \in \mathbb{G}_\lambda$ the map $g \mapsto h + g$. The distinguished element $y_0$ is just $0 \in \mathbb{G}_\lambda$. In this way, $\mathcal{O}$ becomes the generic group action oracle, with labeling function $L$. However, we also include extra operations in $\mathcal{F}$, the exact operations will depend on the application.

Our goal will be to simulate $\mathcal{O}$, the generic group action oracle with extra operations, using only a plain group action $(\mathbb{G}, \mathcal{X}, *)$. $(\mathbb{G}, \mathcal{X}, *)$ could be a standard-model group action, or perhaps a plain generic group action. We will assume $\mathcal{X}_\lambda \subseteq \{0,1\}^m$ for some polynomial $m = m(\lambda)$. This "base" group action will be the source of hardness. We will therefore make some hopefully simple and mild computational assumptions about $(\mathbb{G}, \mathcal{X}, *)$, and hope to derive useful hardness results about the expanded group action $\mathcal{O}$.

To do so, we will let $\mathcal{Y}' = \mathcal{X}_\lambda^{\otimes k}$ for some $k$. We will also choose some integers $c_1, \ldots, c_k$ whose GCD is 1, and starting set elements $y_1, \ldots, y_k$. Then define $\Gamma(g) = ((c_1 g) * y_1, (c_2 g) * y_2, \cdots, (c_k g) * y_k)$. Since the GCD of the $c_i$ is 1, the map $\Gamma(g)$ is injective.

For $f$ corresponding to adding group element $h$, we can set $P((z_1, \ldots, z_k), h) = ((c_1 h) * z_1, \cdots, (c_k h) * z_k)$. Note that this will have the correct effect, as $P(\Gamma(g), h) = \Gamma(h + g)$. For simulating other functions $f \in \mathcal{F}$, we will rely on other transformations to the vector $(z_1, \ldots, z_k)$, which will depend on the application.

### 4.2.4 Correctness of the Simulation

▶ **Lemma 7.** *Fix $y_0, \mathcal{Y}, \mathcal{Y}', \Gamma, \mathcal{F}$ as above. Assume $m' \geq s + t$ for some $t$. Then consider any quantum algorithm $\mathcal{A}$ which makes $q$ quantum queries to its oracle. Then:*

$$\left| \Pr\left[\mathcal{A}^{\mathcal{O}}(L(y_0)) = 1\right] - \Pr\left[\mathcal{A}^{\mathcal{O}'}(\Pi(\Gamma(y_0))) = 1\right] \right| < O(q \times 2^{-t/2})$$

*Above, $L, \Pi$ are random injections, with $\mathcal{O}, \mathcal{O}'$ being derived from them as above. The probabilities are over the random choice of $L, \Pi$ and the randomness of $\mathcal{A}$. Note that our order of quantifiers allows $\mathcal{A}$ to depend on $y_0, \mathcal{Y}, \mathcal{Y}', \Gamma, \mathcal{F}$.*

**Proof.** We prove security via a sequence of hybrids.

**Hybrid 0.** This is the case where we run $\mathcal{A}^{\mathcal{O}}(L(y_0))$ where $L : \mathcal{Y} \to \{0,1\}^{m'}$ is uniform random injection. Let $p_0$ be the probability of outputting 1.

**Hybrid 1.** Here, we run $\mathcal{A}^{\mathcal{O}}(L(y_0))$, except that we set $L$ to be the function $L(y) = \Pi(\Gamma(y))$, where $\Pi$ is a random injection. But since $\Gamma$ is an injection, this means $L$ is a random injection anyway, so the distribution of $L$ and hence $\mathcal{O}$ is identical to **Hybrid 0**. Therefore, if we let $p_1$ be the probability $p_0$ outputs 1 in **Hybrid 1**, we have $p_1 = p_0$. Observe that $L(y_0) = \Pi(\Gamma(y_0))$.

**Hybrid 2.** Here, we run $\mathcal{A}^{\mathcal{O}'}(\Pi(\Gamma(y_0)))$. Let $p_2$ be the probability of outputting 1. On all points that $\mathcal{O}$ accepts, $\mathcal{O}'$ behaves identically. Likewise, on any point that $\mathcal{O}'$ rejects, $\mathcal{O}'$ rejects as well. The only difference between this and **Hybrid 1** is that here, $\mathcal{O}'$ may accept elements that were rejected by $\mathcal{O}$, namely elements that are in the image of $\Pi$ but not in the image of $L = \Pi \circ \Gamma$. We will show that these potential changes are nevertheless undetectable except with small probability.

Consider running $\mathcal{A}^{\mathcal{O}}(L(y_0))$ where $L(y) = \Pi(\Gamma(y))$ as in **Hybrid 1**. However, we only sample $\Pi$ on inputs $z$ that are in the image of $\Gamma$; for all other inputs $z$, $\Pi$ remains unspecified. Observe that **Hybrid 1** never needs to evaluate $\Pi$ on $z$ outside of the image of $\Gamma$, since the oracle $\mathcal{O}$ will anyway reject in these cases. Let $S \subseteq \{0,1\}^{m'}$ be the set of images of $\Pi$ sampled so far.

Now imagine simulating the rest of $\Pi$. Let $T \subset \{0,1\}^{m'}$ be the set of images of $\Pi$ for $z \in \mathcal{Y}'$ that are not in the image of $\Gamma$. Observe that $T$ is a random subset of size $|\mathcal{Y}'| \setminus |\mathcal{Y}| \leq |\mathcal{Y}'| \leq 2^s$. We now observe that the only points where $\mathcal{O}$ and $\mathcal{O}'$ differ are on pairs $(\ell, f)$ for $\ell \in T$: for $\ell \in S$, the two faithfully compute the same function and are identical, while for $\ell \notin T \cup S$, both output $\perp$.

From here, concluding that $p_1$ and $p_2$ are close is a standard argument. The expected total query weight in **Hybrid 1** on points $(\ell, f)$ for $\ell \in T$ is at most $|T|/2^{m'} \leq 2^{-t}$. Then via standard results in quantum query complexity [8], the difference in acceptance probabilities $|p_1 - p_2|$ is at most $O(\sqrt{q^2 2^{-t}}) = O(q \times 2^{-t/2})$. Thus $|p_0 - p_2| \leq O(q \times 2^{-t/2})$, as desired. ◀

Next, we recall a lemma that shows that random injections can be simulated quantumly:

▶ **Lemma 8** ([55]). *Random injections with quantum query access can be simulated efficiently.*

With Lemmas 7 and 8 in hand, we now turn to security proofs in the GGAM.

## 4.3 Security of our Quantum Lightning Scheme

Here, we prove the generic security of our quantum lightning scheme (Construction 4). We do not know how to prove security under any standard group action-based assumption. We instead introduce a novel assumption that appears plausible, but needs extra cryptanalysis to be certain.

### 4.3.1 The Decisional 2x Assumption (D2X)

A classical "Diffie-Hellman Exponent" assumption is to distinguish $g^a, g^{a^2}$ from $g^a, g^b$ for uniform $a, b$. The group action equivalent would be to distinguish $a * x_\lambda, (2a) * x_\lambda$ from $a * x_\lambda, b * x_\lambda$ for uniform $a, b \in \mathbb{G}_\lambda$. Our assumption is based on this assumption. Hover, we need something a bit stronger. In particular, we need not just the set element $(2a) * x_\lambda$ or $b * x_\lambda$, but the ability to query on an *arbitrary* set element $y$ and receive $(2a) * y$ or $b * y$. In th classical group setting, this would correspond to receiving $g^a$, and then being able to query the function $h \mapsto h^{a^2}$ or $h \mapsto h^b$.

Note that if allowing arbitrary queries to this oracle, the problem is *easy* in many cases. In particular, suppose the order of $\mathbb{G}_\lambda$ is odd with order $2t - 1$. Then by querying the oracle $t$ times, we can compute $y_1 = (2a) * x_\lambda, y_2 = (2a) * y_1 = (4a) * x_\lambda, \cdots$, ultimately computing $y_t = (2ta) * x_\lambda = a * x_\lambda$. On the other hand, if the oracle maps $y \mapsto b * x_\lambda$ for a random $b$, then $y_t = (tb) * x_\lambda \neq a * x_\lambda$. This allows for distinguishing the two cases.

Therefore, we only allow a *single* query to the oracle. In this case, a single query does not appear sufficient for breaking the assumption. The adversary, on input $u = a * x_\lambda$, can send $u$ to the oracle, receiving $(3a) * x_\lambda$ or $(a + b) * x_\lambda$. Or it can send $x_\lambda$ to the oracle, receiving $(2a) * x_\lambda$ or $b * x_\lambda$. It can also act on these elements by known constants, computing either $(2a + c) * x_\lambda, (3a + d) * x_\lambda$, or $(b + c) * x_\lambda, (a + b + d) * x_\lambda$. It can also act on the original element $u$, and also on $x_\lambda$ by known constants, receiving $(a + e) * x_\lambda, f * x_\lambda$. Intuitively, it seems the only way the adversary can distinguish between these cases is to find constants $c, d, e, f$ that cause a collision between elements when the oracle acts by $2a$, but no collision when the oracle acts by $b$. However, for any constants $c, d, e, f$, the probability of a collision occurring in either case is negligible. Based on this intuitive argument, it is possible to prove that this assumption is generically hard against *classical* algorithms. We do not, however, know if there is a clever quantum algorithm that breaks the assumption. However, it seems plausible that there is no such efficient quantum algorithm.

We will also allow the query to be quantum, and for technical reasons, we will use an *in place* oracle, meaning $\sum_g \alpha_g |g * x_\lambda\rangle \mapsto \sum_g \alpha_g |(2a + g) * x_\lambda\rangle$, as opposed to the using "standard" oracle which maps $\sum_{g,y} \alpha_{g,y} |g * x_\lambda, y\rangle \mapsto \sum_{g,y} \alpha_{g,y} |g * x_\lambda, y \oplus |(g + 2a) * x_\lambda\rangle$.

▶ **Assumption 9.** The Decisional 2X Assumption with minimal oracle (D2X/min) assumption holds on a group action $(\mathbb{G}, \mathcal{X}, *)$ if, for all QPT adversaries $\mathcal{A}$, there exists a negligible $\lambda$ such that

$$\left| \Pr\left[ \mathcal{A}^{M^1_{2a}}(a * x_\lambda) = 1 : a \leftarrow \mathcal{G}_\lambda \right] - \Pr\left[ \mathcal{A}^{M_b}(a * x_\lambda) = 1 : a, b \leftarrow \mathcal{G}_\lambda \right] \right| \leq \mathsf{negl}(\lambda) \ .$$

Above, $M_c$ is the in-place (or "minimal") oracle mapping $y \mapsto c * y$, and $M^1_c$ means the adversary can make only a single query to $M_c$.

If we insist on standard oracles, we can instead utilize the following assumption:

▶ **Assumption 10.** The Decisional 2X Assumption with standard oracle (D2X/std) assumption holds on a group action $(\mathbb{G}, \mathcal{X}, *)$ if, for all QPT adversaries $\mathcal{A}$, there exists a negligible $\lambda$ such that

$$\left|\Pr\left[\mathcal{A}^{S^1_{2a}, S^1_{-2a}}(a * x_\lambda) = 1 : a \leftarrow \mathcal{G}_\lambda\right] - \Pr\left[\mathcal{A}^{S^1_b, S^1_{-b}}(a * x_\lambda) = 1 : a, b \leftarrow \mathcal{G}_\lambda\right]\right| \leq \mathsf{negl}(\lambda) \ .$$

Above, $S_c$ is the standard oracle mapping $(y, z) \mapsto (y, z \oplus (c * y))$, and $S^1_c$ means the adversary can make only a single query to $S_c$.

The following lemma is straightforward:

▶ **Lemma 11.** *If D2X/std holds on a group action $(\mathbb{G}, \mathcal{X}, *)$, then so does D2X/min*

**Proof.** We simply use the oracles $S^1_c, S^1_{-c}$ to simulate the oracle $M^1_c$ in the obvious way. ◀

### 4.3.2 Our security proof

We now prove the generic security of our quantum lightning scheme.

▶ **Theorem 12.** *Let $(\mathbb{G}, \mathcal{X}, *)$ be a group with $\mathcal{X} \subseteq \{0,1\}^m$ such that D2X/min holds (Assumption 9). Let $m' \geq 2m + \omega(\log \lambda)$. Let $(\mathsf{Gen}^{\mathsf{GGAM}_{\mathbb{G},m'}}, \mathsf{Ver}^{\mathsf{GGAM}_{\mathbb{G},m'}})$ be the quantum money construction from Construction 4, using the generic group action $\mathsf{GGAM}_{\mathbb{G},m'}$. Then the quantum money construction is a secure quantum lightning scheme.*

**Proof.** Consider an adversary $\mathcal{B}^{\mathsf{GGAM}_{\mathbb{G},m'}}$ for quantum lightning security, and let $\epsilon$ be the probability that $\mathcal{B}$ wins. We will assume that $\mathsf{Ver}^{\mathsf{GGAM}_{\mathbb{G},m'}}(h, \$)$ projects onto the correct banknote $|\mathbb{G}^h_\lambda * L(0)\rangle$; this assumption only introduces a negligible error which is easily accounted for. Therefore, with probability $\epsilon$, $\mathcal{B}$ outputs $h$ and exactly two copies of the state $|\mathbb{G}^h_\lambda * L(0)\rangle$.

We now construct an adversary $\mathcal{A}$ for D2X/min on the group action $(\mathbb{G}, \mathcal{X}, *)$. $\mathcal{A}$, on input $u = a * x_\lambda$, will choose a random injection $\Pi : \{0,1\}^{2m} \to \{0,1\}^{m'}$. It will then compute $X = \Pi(x_\lambda, u)$. $\mathcal{A}$ will then run $\mathcal{B}(X)$, simulating its queries $(\ell, g)$ to the group action as follows: compute $(z_1, z_2) \leftarrow \Pi^{-1}(\ell)$, and then return $\Pi(g * z_1, g * z_2)$. For superposition queries, $\mathcal{A}$ simply runs this computation in superposition. Note that if we let $\Gamma(g) = (g * x_\lambda, g * u)$, then $\mathcal{A}$ simulates these queries exactly as prescribed above in our general framework, for constants $c_1 = c_2 = 1$ and $(y_1, y_2) = (x_\lambda, u)$.

Finally, when $\mathcal{B}$ produces serial number $h$ and banknotes $\$_1, \$_2$, $\mathcal{A}$ does the following:

- Run $\mathsf{Ver}^{\mathcal{O}'}(h, \$_1)$ and $\mathsf{Ver}^{\mathcal{O}'}(h, \$_2)$, answering the queries of $\mathsf{Ver}$ using the simulated group action oracle. If either run rejects, output a random bit. Otherwise, let $\$'_1, \$'_2$ be the resulting states of the verifier.
- In superposition, it applies the following map $\ell \mapsto \ell'$ to $\$'_2$:
  - First map $\ell \mapsto \Pi^{-1}(\ell) = (z_1, z_2)$
  - Use the oracle $M_c$ from the D2X/min assumption to replace $z_1$ with $z'_1 = c * z_1$, where $c = 2a$ or $b$.
  - Now map $(z'_1, z_2) \mapsto \ell' = \Pi(z_2, z'_1)$.

  Let $\$''_2$ be the result of this map.
- Apply the swap test to $\$'_1, \$''_2$, outputting whatever the swap test outputs.

By applying Lemma 7, we can conclude that $\$_1, \$_2$ are actually superpositions over elements of the form $L(g) = \Pi(g * z_1, g * z_2)$ for varying $g$. Then using our characterization of the accepting states of $\mathsf{Ver}$, we see that both runs of $\mathsf{Ver}$ simulteneously accept with probability $\epsilon$, and in this case $\$'_1 = \$'_2 = |\mathbb{G}_\lambda^h * L(0)\rangle, \$'_2$.

We must analyze the effect of the map $\ell \mapsto \ell'$ on $|\mathbb{G}_\lambda^h * L(0)\rangle$. We break into two cases:

- $M_c$ implements the action $y \mapsto c * y$ with $c = 2a$. Let $\ell = L(g) = \Pi(g * z_1, g * z_2) = \Pi(g * x_\lambda, (a+g) * x_\lambda)$, which maps to $\ell' = \Pi(g * x_\lambda, (2g) * x_\lambda = L(a+g)$.
  Therefore, $|\mathbb{G}_\lambda^h * L(0)\rangle$ maps to

$$\begin{aligned}
|\mathbb{G}_\lambda^h * L(0)\rangle &= \frac{1}{\sqrt{|\mathbb{G}_\lambda}} \sum_g \chi(g,h)|L(g)\rangle \\
&\mapsto \frac{1}{\sqrt{|\mathbb{G}_\lambda}} \sum_g \chi(g,h)|L(a+g)\rangle \\
&= \frac{1}{\sqrt{|\mathbb{G}_\lambda}} \sum_g \chi(g-a,h)|L(g)\rangle \\
&= \chi(a,-h)|\mathbb{G}_\lambda^h * L(0)\rangle
\end{aligned}$$

  Thus, in this case, $\mathcal{A}$ obtains two copies of $|\mathbb{G}_\lambda^h * L(0)\rangle$, which the swap test will accept with probability 1. Therefore, the probability $\mathcal{A}$ outputs 1 is $\frac{1}{2}(1-\epsilon) + \epsilon = \frac{1+\epsilon}{2}$.
- $M_c$ implements the action $y \mapsto c * y$ with $c = b$ for a random $b$. In this case, $\ell = L(g) = \Pi(g * x_\lambda, (a+g) * x_\lambda)$ maps to $\ell' = \Pi((a+g) * x_\lambda, (g+b) * x_\lambda)$. However, $\ell'$ is *not* equal to $L(g')$ for any $g$. Indeed, in order for $\ell' = L(g')$, we get several equations:

$$g' = a + g \quad a + g' = g + b$$

  The first equation requires that $g' = a+g$, while the last one requires that $g' = g+b-a \neq g+a$. Hence, the state $\$''_2$ has disjoint support from the state $|\mathbb{G}_\lambda^h * L(0)\rangle$, and hence is orthogonal to it. Therefore, the swap test will accept with probability exactly $1/2$. The overall probability $\mathcal{A}$ outputs 1 is therefore exactly $1/2$.

Thus, we see that $\mathcal{A}$ has advantage $\epsilon/2$ in distinguishing DDH, breaking the assumption. ◀

## References


1. Scott Aaronson. Quantum copy-protection and quantum money. In *Proceedings of the 2009 24th Annual IEEE Conference on Computational Complexity*, CCC '09, pages 229–242, Washington, DC, USA, 2009. IEEE Computer Society. doi:10.1109/CCC.2009.42.
2. Scott Aaronson and Paul Christiano. Quantum money from hidden subspaces. In Howard J. Karloff and Toniann Pitassi, editors, *44th ACM STOC*, pages 41–60. ACM Press, May 2012. doi:10.1145/2213977.2213983.
3. Navid Alamati, Luca De Feo, Hart Montgomery, and Sikhar Patranabis. Cryptographic group actions and applications. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part II*, volume 12492 of *LNCS*, pages 411–439. Springer, Heidelberg, December 2020. doi:10.1007/978-3-030-64834-3_14.
4. Navid Alamati, Giulio Malavolta, and Ahmadreza Rahimi. Candidate trapdoor claw-free functions from group actions with applications to quantum protocols. In Eike Kiltz and Vinod Vaikuntanathan, editors, *TCC 2022, Part I*, volume 13747 of *LNCS*, pages 266–293. Springer, Heidelberg, November 2022. doi:10.1007/978-3-031-22318-1_10.
5. James Bartusek, Jiaxin Guan, Fermi Ma, and Mark Zhandry. Return of GGH15: Provable security against zeroizing attacks. In Amos Beimel and Stefan Dziembowski, editors, *TCC 2018, Part II*, volume 11240 of *LNCS*, pages 544–574. Springer, Heidelberg, November 2018. doi:10.1007/978-3-030-03810-6_20.

6 Amit Behera and Or Sattath. Almost public quantum coins. Cryptology ePrint Archive, Report 2020/452, 2020. URL: https://eprint.iacr.org/2020/452.

7 Shalev Ben-David and Or Sattath. Quantum tokens for digital signatures, 2016. arXiv:1609.09047.

8 Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and weaknesses of quantum computing. *SIAM J. Comput.*, 26(5):1510–1523, October 1997.

9 Ward Beullens, Thorsten Kleinjung, and Frederik Vercauteren. CSI-FiSh: Efficient isogeny based signatures through class group computations. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part I*, volume 11921 of *LNCS*, pages 227–247. Springer, Heidelberg, December 2019. doi:10.1007/978-3-030-34578-5_9.

10 Dan Boneh, Jiaxin Guan, and Mark Zhandry. A lower bound on the length of signatures based on group actions and generic isogenies. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 507–531. Springer, Heidelberg, April 2023. doi:10.1007/978-3-031-30589-4_18.

11 Xavier Bonnetain and André Schrottenloher. Quantum security analysis of CSIDH. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part II*, volume 12106 of *LNCS*, pages 493–522. Springer, Heidelberg, May 2020. doi:10.1007/978-3-030-45724-2_17.

12 Zvika Brakerski, Paul Christiano, Urmila Mahadev, Umesh V. Vazirani, and Thomas Vidick. A cryptographic test of quantumness and certifiable randomness from a single quantum device. In Mikkel Thorup, editor, *59th FOCS*, pages 320–331. IEEE Computer Society Press, October 2018. doi:10.1109/FOCS.2018.00038.

13 Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. Factoring and pairings are not necessary for iO: Circular-secure LWE suffices. Cryptology ePrint Archive, Report 2020/1024, 2020. URL: https://eprint.iacr.org/2020/1024.

14 Wouter Castryck and Thomas Decru. An efficient key recovery attack on SIDH. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 423–447. Springer, Heidelberg, April 2023. doi:10.1007/978-3-031-30589-4_15.

15 Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: An efficient post-quantum commutative group action. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part III*, volume 11274 of *LNCS*, pages 395–427. Springer, Heidelberg, December 2018. doi:10.1007/978-3-030-03332-3_15.

16 Andrew Childs, David Jao, and Vladimir Soukharev. Constructing elliptic curve isogenies in quantum subexponential time. *Journal of Mathematical Cryptology*, 8(1):1–29, 2014.

17 Leonardo Colò and David Kohel. Orienting supersingular isogeny graphs. *Journal of Mathematical Cryptology*, 14:414–437, October 2020. doi:10.1515/jmc-2019-0034.

18 Marta Conde Pena, Raul Durán Díaz, Jean-Charles Faugère, Luis Hernández Encinas, and Ludovic Perret. Non-quantum cryptanalysis of the noisy version of aaronson–christiano's quantum money scheme. *IET Information Security*, 13(4):362–366, 2019.

19 Jean-Marc Couveignes. Hard homogeneous spaces. Cryptology ePrint Archive, Report 2006/291, 2006. URL: https://eprint.iacr.org/2006/291.

20 Luca De Feo, Tako Boris Fouotsa, Péter Kutas, Antonin Leroux, Simon-Philipp Merz, Lorenz Panny, and Benjamin Wesolowski. SCALLOP: Scaling the CSI-FiSh. In Alexandra Boldyreva and Vladimir Kolesnikov, editors, *PKC 2023, Part I*, volume 13940 of *LNCS*, pages 345–375. Springer, Heidelberg, May 2023. doi:10.1007/978-3-031-31368-4_13.

21 Luca De Feo, David Jao, and Jérôme Plût. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. *Journal of Mathematical Cryptology*, 8(3):209–247, 2014.

22 Luca De Feo and Michael Meyer. Threshold schemes from isogeny assumptions. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020, Part II*, volume 12111 of *LNCS*, pages 187–212. Springer, Heidelberg, May 2020. doi:10.1007/978-3-030-45388-6_7.

23 Julien Duman, Dominik Hartmann, Eike Kiltz, Sabrina Kunzweiler, Jonas Lehmann, and Doreen Riepel. Generic models for group actions. In Alexandra Boldyreva and Vladimir

Kolesnikov, editors, *PKC 2023, Part I*, volume 13940 of *LNCS*, pages 406–435. Springer, Heidelberg, May 2023. doi:10.1007/978-3-031-31368-4_15.

24 Mark Ettinger and Peter Høyer. On quantum algorithms for noncommutative hidden subgroups. *Advances in Applied Mathematics*, 25(3):239–251, 2000. doi:10.1006/aama.2000.0699.

25 Edward Farhi, David Gosset, Avinatan Hassidim, Andrew Lutomirski, and Peter W. Shor. Quantum money from knots. In Shafi Goldwasser, editor, *ITCS 2012*, pages 276–289. ACM, January 2012. doi:10.1145/2090236.2090260.

26 Georg Fuchsbauer, Eike Kiltz, and Julian Loss. The algebraic group model and its applications. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part II*, volume 10992 of *LNCS*, pages 33–62. Springer, Heidelberg, August 2018. doi:10.1007/978-3-319-96881-0_2.

27 Craig Gentry, Sergey Gorbunov, and Shai Halevi. Graph-induced multilinear maps from lattices. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 498–527. Springer, Heidelberg, March 2015. doi:10.1007/978-3-662-46497-7_20.

28 Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from well-founded assumptions. In Samir Khuller and Virginia Vassilevska Williams, editors, *53rd ACM STOC*, pages 60–73. ACM Press, June 2021. doi:10.1145/3406325.3451093.

29 Daniel M. Kane. Quantum money from modular forms, 2018. arXiv:1809.05925.

30 Daniel M. Kane, Shahed Sharif, and Alice Silverberg. Quantum money from quaternion algebras. Cryptology ePrint Archive, Report 2021/1294, 2021. URL: https://eprint.iacr.org/2021/1294.

31 Andrey Boris Khesin, Jonathan Z Lu, and Peter W Shor. Publicly verifiable quantum money from random lattices, 2022. arXiv:2207.13135.

32 Jiahui Liu, Hart Montgomery, and Mark Zhandry. Another round of breaking and making quantum money: How to not build it from lattices, and more. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part I*, volume 14004 of *LNCS*, pages 611–638. Springer, Heidelberg, April 2023. doi:10.1007/978-3-031-30545-0_21.

33 Qipeng Liu and Mark Zhandry. Revisiting post-quantum Fiat-Shamir. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 326–355. Springer, Heidelberg, August 2019. doi:10.1007/978-3-030-26951-7_12.

34 Andrew Lutomirski. An online attack against wiesner's quantum money, 2010. arXiv:1010.0256.

35 Andrew Lutomirski, Scott Aaronson, Edward Farhi, David Gosset, Jonathan A. Kelner, Avinatan Hassidim, and Peter W. Shor. Breaking and making quantum money: Toward a new quantum cryptographic protocol. In Andrew Chi-Chih Yao, editor, *ICS 2010*, pages 20–31. Tsinghua University Press, January 2010.

36 Luciano Maino and Chloe Martindale. An attack on SIDH with arbitrary starting curve. Cryptology ePrint Archive, Report 2022/1026, 2022. URL: https://eprint.iacr.org/2022/1026.

37 Ueli M. Maurer. Abstract models of computation in cryptography (invited paper). In Nigel P. Smart, editor, *10th IMA International Conference on Cryptography and Coding*, volume 3796 of *LNCS*, pages 1–12. Springer, Heidelberg, December 2005.

38 Hart Montgomery and Mark Zhandry. Full quantum equivalence of group action DLog and CDH, and more. In Shweta Agrawal and Dongdai Lin, editors, *ASIACRYPT 2022, Part I*, volume 13791 of *LNCS*, pages 3–32. Springer, Heidelberg, December 2022. doi:10.1007/978-3-031-22963-3_1.

39 Emmanuela Orsini and Riccardo Zanotto. Simple two-round OT in the explicit isogeny model. Cryptology ePrint Archive, Report 2023/269, 2023. URL: https://eprint.iacr.org/2023/269.

40 Lorenz Panny. Csi-fish really isn't polynomial-time, 2023. URL: https://yx7.cc/blah/2023-04-14.html.

41 Chris Peikert. He gives C-sieves on the CSIDH. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part II*, volume 12106 of *LNCS*, pages 463–492. Springer, Heidelberg, May 2020. doi:10.1007/978-3-030-45724-2_16.

42 Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005. doi:10.1145/1060590.1060603.

43 Damien Robert. Breaking SIDH in polynomial time. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 472–503. Springer, Heidelberg, April 2023. doi:10.1007/978-3-031-30589-4_17.

44 Bhaskar Roberts. Security analysis of quantum lightning. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part II*, volume 12697 of *LNCS*, pages 562–567. Springer, Heidelberg, October 2021. doi:10.1007/978-3-030-77886-6_19.

45 Bhaskar Roberts and Mark Zhandry. Franchised quantum money. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part I*, volume 13090 of *LNCS*, pages 549–574. Springer, Heidelberg, December 2021. doi:10.1007/978-3-030-92062-3_19.

46 Alexander Rostovtsev and Anton Stolbunov. Public-Key Cryptosystem Based On Isogenies. Cryptology ePrint Archive, Report 2006/145, 2006. URL: https://eprint.iacr.org/2006/145.

47 Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *35th FOCS*, pages 124–134. IEEE Computer Society Press, November 1994. doi:10.1109/SFCS.1994.365700.

48 Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 256–266. Springer, Heidelberg, May 1997. doi:10.1007/3-540-69053-0_18.

49 Vladimir Shpilrain. Cryptanalysis of stickel's key exchange scheme. In Edward A. Hirsch, Alexander A. Razborov, Alexei Semenov, and Anatol Slissenko, editors, *Computer Science – Theory and Applications*, pages 283–288, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

50 E. Stickel. A new method for exchanging secret keys. In *Third International Conference on Information Technology and Applications (ICITA'05)*, volume 2, pages 426–430, 2005. doi:10.1109/ICITA.2005.33.

51 Hoeteck Wee and Daniel Wichs. Candidate obfuscation via oblivious LWE sampling. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part III*, volume 12698 of *LNCS*, pages 127–156. Springer, Heidelberg, October 2021. doi:10.1007/978-3-030-77883-5_5.

52 Stephen Wiesner. Conjugate coding. *SIGACT News*, 15(1):78–88, January 1983. doi:10.1145/1008908.1008920.

53 Takashi Yamakawa and Mark Zhandry. Verifiable quantum advantage without structure. In *63rd FOCS*, pages 69–74. IEEE Computer Society Press, October / November 2022. doi:10.1109/FOCS54457.2022.00014.

54 Mark Zhandry. Quantum lightning never strikes the same state twice. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part III*, volume 11478 of *LNCS*, pages 408–438. Springer, Heidelberg, May 2019. doi:10.1007/978-3-030-17659-4_14.

55 Mark Zhandry. Redeeming reset indifferentiability and applications to post-quantum security. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part I*, volume 13090 of *LNCS*, pages 518–548. Springer, Heidelberg, December 2021. doi:10.1007/978-3-030-92062-3_18.

56 Mark Zhandry. To label, or not to label (in generic groups). In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part III*, volume 13509 of *LNCS*, pages 66–96. Springer, Heidelberg, August 2022. doi:10.1007/978-3-031-15982-4_3.

57 Mark Zhandry. Quantum money from abelian group actions. Cryptology ePrint Archive, Paper 2023/1097, 2023. URL: https://eprint.iacr.org/2023/1097.