Quantum Money from Abelian Group Actions

Mark Zhandry
mzhandry@gmail.com
NTT Research

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.

1 Introduction

Quantum money, first envisioned by Wiesner [Wie83], 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 [Aar09], 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.

This Work. 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 Section 6 for details. 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.

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 [DHK$^+$23]. See Section 5 for details.

We conclude in Section 7 with 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

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 [Cou06, RS06, CLM$^+$18, BKV19, DFK$^+$23]. 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.

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, χ 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 χ (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: $\chi(g,h)=\chi(h,g)$.

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 $|(-u)*y\rangle$.

An instantiation using REGAs. For some isogeny-based group actions such as CSIDH [CLM+18], 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) [ADMP20]. 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 Section 6. 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 [Sho97, Mau05] adapted to group actions. Generic models for group actions have been considered previously [MZ22, BGZ23, OZ23, DHK+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 [EH00]. 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.

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 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

[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 [Reg05] 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) [Reg05], outlined by [Zha19b], 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.

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 Π 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 Π [6] and run the generic adversary on the new group action, using its knowledge of Π 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 Π 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 Π 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 Π using known techniques. For the purposes of our discussion here, we can ignore this issue.

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 Π 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 [Zha19b], 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 Π 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 Π, like in our negation-query example. But since the scheme has to be efficient, the inputs to Π 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 Π alone will be able to preserve $|\mathbb{G}^h \star X\rangle$.

1.3 On Knowledge Assumptions and Algebraic Group Actions

In Section 5, we show a different approach to justifying the security of our scheme, by adapting certain knowledge assumptions [LMZ23] to the setting of group actions. Despite some high-level similarities to [LMZ23], 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 [LMZ23] 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 [LMZ23] 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.

Algebraic Group Actions. The Algebraic Group Model (AGM) [FKL18] 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 [DHK+23, OZ23] have

[7] There are some caveats to this classical claim; see [Zha22] for discussion.

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 Section 5, 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 Section 5, 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 Section 7, 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 [Reg05], where we can actually prove that the discrete log problem is hard under 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 [LZ19, LMZ23]. 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 Section 7. 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

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 [Lut10]. Some partial solutions have been proposed, e.g. [BS20a, RZ21]. The dream solution, however, is known as *public key* quantum money [Aar09]. 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 [Aar09, AC12, FGH$^+$12, Kan18, Zha19b, KSS21, KLS22, LMZ23], they mostly either (1) have been subsequently broken (e.g. [Aar09, AC12, Zha19b, KLS22] which were broken by [LAF$^+$10, CPDDF$^+$19, Rob21, LMZ23]), or (2) rely on new cryptographic building blocks that have received little attention from the cryptographic community (e.g. [FGH$^+$12, Kan18, KSS21] from problems on knots or quaternion algebras). The two exceptions are:

- Building on a suggestion of [BDS16], [Zha19b] 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 [JLS21]. Yet the post-quantum study of iO is much less thorough. While some post-quantum proposals have been made [GGH15, BGMZ18, BDGM20, WW21], their post-quantum hardness is not well-understood.

- [LMZ23] 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.

Cryptography from group actions and isogenies. Isogenies were first proposed for use in post-quantum cryptography by Couveignes [Cou06] and Rostovtsev and Stolbunov [RS06]. Isogenies give a Diffie-Hellman-like structure, but importantly are immune to Shor's algorithm for discrete logarithms [Sho94] 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. [CJS14, DJP14, CLM$^+$18, BKV19, CK20, DM20, Pei20, BS20b, ADMP20, AMR22, MZ22, MM22, CD23, BGZ23, Rob23]).

Certain isogenies such as the original proposals of [Cou06, RS06] as well as CSIDH and its variants [CLM$^+$18, DFK$^+$23] can be abstracted as abelian group actions. However, many other isogenies (such as SIDH [DJP14] and OSIDH [CK20]) 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. [Sti05]), currently all such examples have been broken (e.g. [Shp08]). 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 [AMR22], who build a proof of quantumness [BCM+18]. We note that proofs of quantumness can also be achieved under several "standard" cryptographic tools, such as LWE [BCM+18] or under certain assumptions on hash functions [YZ22]. In contrast, no prior quantum money protocol could be based on similar standard building blocks. We also note that [AMR22] 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 [LMZ23], who build quantum money from walkable invariants, and propose an instantiation using isogenies over super-singular 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.

Relation to [LMZ23]. Aside from using isogenies, our construction has some conceptual similarities to [LMZ23], 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 [LMZ23] 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 $y = g * x$, 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

[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 [Pan23]. 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 [Sho94]. Since we always assume a quantum computer in this work, we can therefore treat CSIDH as a REGA.

group elements; this is accomplished using an analog of the swap test. [LMZ23] 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, [LMZ23] 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 [LMZ23], 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 $g = (g_1, \dots, g_k)$ where $g_j \in \mathbb{Z}_{n_j}$. Then define $\chi : \mathbb{G}^2 \to \mathbb{C}$ by

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

Observe the following:

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

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

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

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

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

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* [AC12], which are essentially the setting where there is only ever a single valid banknote produced by the mint. As shown in [AC12], 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 σ 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 Gen are accepted by 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 2.1. Consider a QPT adversary $\mathcal{A}$, which takes as input a serial number σ 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:

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

Definition 2.2. Consider a QPT adversary $\mathcal{B}$, which takes as input the security parameter λ, and outputs a serial number σ 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 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 [Zha19b].

Remark 2.3. One limitation of quantum lightning as defined above is that it cannot hold against non-uniform attackers with quantum advice, as such attackers could have $\sigma, \$_1, \$_2$ hard-coded in their advice. The situation is analogous to the case of collision resistance, where unkeyed hash functions cannot be secure against non-uniform attackers. This limitation be remedied by either insisting on only uniform attackers or attackers with classical advice. Alternatively, one can work in a trusted setup model, where a trusted third party generates a common reference string that is then inputted into $\mathsf{Gen}, \mathsf{Ver}$. A third option is to use the "human ignorance" approach [Rog06], in which we would formalize security proofs as explicitly transforming a quantum lightning adversary into an adversary for some other task, the latter adversary existing but is presumably unknown to human knowledge. We will largely ignore these issues throughout this work, but occasionally make brief remarks about what the various approaches would look like.

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^λ, 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 Recog which recognizes elements in $\mathcal{X}_\lambda$. That is, for any λ 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 $y = g * x_\lambda$.

Cryptographic group actions. At a minimum, a cryptographically useful group action will satisfy the following discrete log assumption:

Assumption 2.4. 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 λ such that

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

We will also consider stronger assumptions. One assumption we consider is the analog of DDH for group actions:

Assumption 2.5. The *decisional Diffie-Hellman assumption* (DDH) holds on a group action $(\mathbb{G}, \mathcal{X}, *)$ if, for all QPT adversaries $\mathcal{A}$, there exists a negligible λ such that

$$|\Pr[\mathcal{A}(a * x_\lambda, b * x_\lambda, c * x_\lambda) = 1 : a, b, c \leftarrow \mathcal{G}_\lambda] - \Pr[\mathcal{A}(a * x_\lambda, b * x_\lambda, (a + b) * x_\lambda) = 1 : a, b \leftarrow \mathcal{G}_\lambda]| \leq \mathsf{negl}(\lambda) \ .$$

Remark 2.6. For simplicity, we model the group actions as being deterministically computed from the security parameter. We could alternatively imagine the group actions being probabilistic, in which case they would be set up by some probabilistic procedure. The parameters would then be part of a common reference string that is supplied to all parties, including the adversary.

3 Our Quantum Lightning Scheme

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

Construction 3.1. 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_λ 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 $\mathsf{QFT}_{\mathbb{G}_\lambda}^{-1}$ to $\mathcal{H}$.
  - Measure $\mathcal{H}$, obtaining a group element h'. Accept if and only if $h' = h$.

Remark 3.2. If using a probabilistic setup of the group action, there are two options. The first is to have Gen set up the group action, and have the parameters be included in the serial number. The second is to have a trusted third party set up the group action, and publish the parameters in a common reference string (CRS). If the goal is only quantum money security, then the former option is always possible, since the security experiment uses an honestly generated serial number. If the goal is quantum lightning security, the former option may not be possible, as the adversary computes the serial number; it may be that there are bad choices of parameters for the group action (and hence the CRS inside the serial number) which make it easy to forge banknotes. Therefore, for quantum lightning security, we would expect using a trusted setup to generate a CRS containing the group action parameters.

[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 $\mathcal{M}$.

3.1 Accepting States of the Verifier

Above we showed that honest banknote states are accepted by the verifier. We now prove that, roughly, honest banknote states are the *only* states accepted by the verifier, with overwhelming probability.

Theorem 3.3. *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 $|\mathbb{G}_\lambda^h * x_\lambda\rangle$. In particular, honest banknotes are accepted with probability 1. The remainder of this subsection is devoted to proving Theorem 3.3.

Lemma 3.4. *For $h'\neq h$, $\langle\mathbb{G}_\lambda^{h'} * x_\lambda|\mathbb{G}_\lambda^h * x_\lambda\rangle=0$*

Proof.

$$\begin{aligned}\langle\mathbb{G}_\lambda^{h'} * x_\lambda|\mathbb{G}_\lambda^h * x_\lambda\rangle &= \frac{1}{|\mathbb{G}_\lambda|}\sum_{g,g'\in\mathbb{G}_\lambda}\chi(g',h')^{-1}\chi(g,h)\langle g' * x_\lambda|g * x_\lambda\rangle\\ &= \frac{1}{|\mathbb{G}_\lambda|}\sum_{g\in\mathbb{G}_\lambda}\chi(g,h')^{-1}\chi(g,h)=\frac{1}{|\mathbb{G}_\lambda|}\sum_{g\in\mathbb{G}_\lambda}\chi(g,h-h')=0\end{aligned}$$

□

Let $|\psi\rangle$ be a a state with support on $\mathcal{X}$. Since the $|\mathbb{G}^{h'} * x_\lambda\rangle$ are orthogonal and the number of h' equals the size of $\mathcal{X}$, the set $\{|\mathbb{G}_\lambda^{h'} * x_\lambda\rangle\}_{h'}$ forms a basis for the set of states with support on $\mathcal{X}$. We can then write $|\psi\rangle=\sum_{h'}\alpha_{h'}|\mathbb{G}_\lambda^{h'} * x_\lambda\rangle$ where $\sum_{h'}\|\alpha_{h'}\|^2=1$. We then have $\|\alpha_h\|^2=\|\langle\psi|\mathbb{G}_\lambda^h * x_\lambda\rangle\|^2$.

We now consider applying the verification algorithm to the state $|\psi\rangle_\mathcal{M}$. After initializing $\mathcal{H}$ to $\frac{1}{\sqrt{|\mathbb{G}_\lambda|}}\sum_{u\in\mathbb{G}_\lambda}|u\rangle_\mathcal{H}$ and applying the map $(u,y)\mapsto(u,(-u) * y)$, the just state of $\mathcal{H},\mathcal{M}$ is:

$$\begin{aligned}&\frac{1}{\sqrt{|\mathbb{G}_\lambda|}}\sum_u|u\rangle_\mathcal{H}\sum_{h'}\alpha_{h'}\frac{1}{\sqrt{|G_\lambda|}}\sum_g\chi(g,h')|(g-u) * x_\lambda\rangle_\mathcal{M}\\ &=\frac{1}{\sqrt{|\mathbb{G}_\lambda|}}\sum_u|u\rangle_\mathcal{H}\sum_{h'}\alpha_{h'}\frac{1}{\sqrt{|G_\lambda|}}\sum_{g'}\chi(g'+u,h')|g' * x_\lambda\rangle_\mathcal{M}\\ &=\sum_{h'}\alpha_{h'}\frac{1}{\sqrt{|\mathbb{G}_\lambda|}}\sum_u\chi(u,h')|u\rangle_\mathcal{H}\frac{1}{\sqrt{|G_\lambda|}}\sum_{g'}\chi(g',h')|g' * x_\lambda\rangle_\mathcal{M}\\ &=\sum_{h'}\alpha_{h'}\frac{1}{\sqrt{|\mathbb{G}_\lambda|}}\sum_u\chi(u,h')|u\rangle_\mathcal{H}|\mathbb{G}^{h'} * x_\lambda\rangle_\mathcal{M}\end{aligned}$$

where above we used the substitution $g'=g-u$. Now when we apply the inverse QFT to $\mathcal{H}$, the resulting state is:

$$\sum_{h'}\alpha_{h'}|h'\rangle_\mathcal{H}|\mathbb{G}^{h'} * x_\lambda\rangle_\mathcal{M}$$

We now measure $\mathcal{H}$, which produces outcome h' with probability $|\alpha_{h'}|^2$, and the register $\mathcal{M}$ collapses to the state $|\mathbb{G}^{h'} * x_\lambda\rangle$. We have that the verifier accepts if $h'=h$, which occurs with probability $\|\alpha_h\|^2=\|\langle\psi|\mathbb{G}_\lambda^h * x_\lambda\rangle\|^2$ as desired. In this case, the register $\mathcal{M}$ collapses to $|\mathbb{G}^h * x_\lambda\rangle$, as desired. This completes the proof of Theorem 3.3. □

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 will be used in Section 5 to break a certain natural knowledge assumption.

Theorem 3.5. *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 3.3 shows 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.

A Shoup-style generic group action. There have been several different proposals for how to define generic group actions [MZ22, DHK+23, BGZ23, OZ23]. Here, we briefly give a definition in the style of Shoup [Sho97]. To help disambiguate between the different models, we will adapt terminology from [Zha22] 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 λ, 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_λ.
- 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 ℓ 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 [DHK+23]. However, in the quantum setting, considering the query complexity alone is insufficient, as discrete logarithms can be solved in polynomial query complexity [EH00]. 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 On other Styles of Generic Group Actions

Other styles of generic group action are possible. For example, [MZ22] consider a similar model except where the group itself is also hidden behind an oracle. We might call this the *Random Group, Random Set Representation* model. It is also possible to consider a version that is akin to Maurer's [Mau05] generic group model, where instead of random labels for every element one only receives handles. This is the kind of model considered in [BGZ23, OZ23]. Following the terminology of [Zha22], this can be called the Type Safe model. We note that it does not make much sense to consider the group as an idealized object while allowing complete access to the set. Since quantum algorithms can solve discrete logarithms, they can essentially "undo" the labelling of the group. This means any such "Random Group Representation" effectively is just a standard-model group action, defeating the purpose of considering an idealized model.

Here, we discuss why these alternate models come with limitations. First we observe that hiding the group behind an oracle puts more idealized constraints on the adversary, and is therefore a less accurate modeling of group actions.

Worse is the case of Type Safe models. In the classical generic group setting, as first proved in [JS08] and clarified in [Zha22], when it comes to proving security, the Type Safe and Random Representation models can usually be treated as equivalent[12]. This equivalence would carry over to the classical setting for generic group actions. However, we observe that the equivalence proved in [JS08, Zha22] does *not* hold in the quantum setting. This observation was first made, but not elaborated on, by [HYY23].

In more detail, one direction of the equivalence — converting an adversary in the Type Safe model into one in the Random Representation model — is trivial, both classically and quantumly. We just use the random labels from the random representation model as the handles for the Type Safe adversary. For the other direction, the classical proof will construct a Type Safe adversary out of a Random Representation adversary by choosing the random labels itself. The challenge is that the Random Representation adversary will expect identical labels on certain related queries, namely if it computes the same element $g * x_\lambda$ in multiple ways. To account for this, the Type Safe adversary maintains a table of all the queries made so far, and the labels generated for those queries. Then if it ever needs to output an element that was already produced, it can use the table to make sure it uses the same label.

In the quantum setting, maintaining this table is problematic, as it requires recording the queries made by the adversary. Quantum queries cannot be recorded without perturbing them, and if the adversary detects any disturbance it may abort and refuse to work. Such an adversary would break the classical reduction. We note that sometimes it is possible to record quantum queries [Zha19a], but the recording has to be done in careful ways that limit applications. In particular, such query recording is usually done on random oracles, and there has so far been no techniques for recording

[12] This is not the case when using the models to prove *impossibility* results, where even classical there is a major difference between the two models.

queries for complicated structures like group action oracles.

Thus, 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.

Algebraic Group Action Model. In Section 5, we consider a different idealized model called the Algebraic Group Action Model, the quantum and group action version of the classical Algebraic Group Model (AGM) [FKL18]. 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 [Zha22]). However, in Section 5, 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 equivalence between Random Representation and Type Safe models does not appear to hold 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.

Challenges with the quantum GGAM. The problem with the quantum GGAM, as observed by [DHK+23], is that we cannot hope for unconditional security results, as the discrete logarithm is easy if we only count quantum query complexity. [DHK+23] take the approach of instead considering the Algebraic Group Action Model (AGAM). We discuss the pitfalls of this approach in Section 5. 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.

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 Γ. 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 Π. We will then give the adversary $\Pi(\Gamma(y_0))$, ad quantum query access to $\mathcal{O}'$.

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 $c_1, \dots, c_k$ whose GCD is 1, and starting set elements $y_1, \dots, y_k$. 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 $P((z_1, \dots, z_k), h) = ((c_1 h) * z_1, \cdots, (c_k h) * z_k)$. 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 $(z_1, \dots, z_k)$, which will depend on the application.

Correctness of the Simulation.

Lemma 4.1. *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, Π are random injections, with $\mathcal{O}, \mathcal{O}'$ being derived from them as above. The probabilities are over the random choice of L, Π 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 Π is a random injection. But since Γ 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 Π 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 Π on inputs z that are in the image of Γ; for all other inputs z, Π remains unspecified. Observe that **Hybrid 1** never needs to evaluate Π on z outside of the image of Γ, since the oracle $\mathcal{O}$ will anyway reject in these cases. Let $S \subseteq \{0,1\}^{m'}$ be the set of images of Π sampled so far.

Now imagine simulating the rest of Π. Let $T \subset \{0,1\}^{m'}$ be the set of images of Π for $z \in \mathcal{Y}'$ that are not in the image of Γ. 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 (ℓ, 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 (ℓ, f) for $\ell \in T$ is at most $|T|/2^{m'} \leq 2^{-t}$. Then via standard results in quantum query complexity [BBBV97], 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 4.2 ([Zha21])**.** *Random injections with quantum query access can be simulated efficiently.*

With Lemmas 4.1 and 4.2 in hand, we now turn to security proofs in the GGAM.

4.3 Group Actions with Twists

In group actions based on isogenies, it is possible to compute a "twist", which maps $g * x_\lambda \mapsto (-g) * x_\lambda$. It is straightforward to update our notion of group action and generic group action to incorporate twists. Let $\mathsf{GGAM}^{\pm}_{\mathbb{G},m}$ denote the generic group action relative to group $\mathbb{G}$ with label length m. Such twists effectively allow for the dihedral group to act on the set $\mathcal{X}_\lambda$. An important question is whether having this larger (non-abelian) group act on $\mathcal{X}_\lambda$ can be damaging for security. Here, we show that, at least generically, the existence of twists plausibly has little impact on security.

Assumptions with Negation. We consider variants of standard assumptions on group actions where additional "negation" elements are given out. For example:

Assumption 4.3. The *discrete log assumption with negation* (DLog$^{\pm}$) holds on a group action $(\mathbb{G}, \mathcal{X}, *)$ if, for all QPT adversaries $\mathcal{A}$, there exists a negligible λ such that

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

Assumption 4.4. The *computational Diffie-Hellman assumption with negation* (CDH$^{\pm}$) holds on a group action $(\mathbb{G}, \mathcal{X}, *)$ if, for all QPT adversaries $\mathcal{A}$, there exists a negligible λ such that

$$\Pr\left[\mathcal{A}\left(\substack{a*x_\lambda,b*x_\lambda,\\(-a)*x_\lambda,(-b)*x_\lambda}\right) = (a+b) * x_\lambda : a,b \leftarrow \mathcal{G}_\lambda\right] \leq \mathsf{negl}(\lambda) \ .$$

Assumption 4.5. The *decisional Diffie-Hellman assumption with negation* (DDH$^{\pm}$) holds on a group action $(\mathbb{G}, \mathcal{X}, *)$ if, for all QPT adversaries $\mathcal{A}$, there exists a negligible λ such that

$$\begin{aligned}&\Bigg|\Pr\left[\mathcal{A}\left(\substack{a*x_\lambda,b*x_\lambda,c*x_\lambda,\\(-a)*x_\lambda,(-b)*x_\lambda,(-c)*x_\lambda}\right) = 1 : a,b,c \leftarrow \mathcal{G}_\lambda\right]\\ &- \Pr\left[\mathcal{A}\left(\substack{a*x_\lambda,b*x_\lambda,(a+b)*x_\lambda,\\(-a)*x_\lambda,*(-b)*x_\lambda,(-a-b)*x_\lambda}\right) = 1 : a,b \leftarrow \mathcal{G}_\lambda\right]\Bigg| \leq \mathsf{negl}(\lambda) \ .\end{aligned}$$

Note that the $^{\pm}$ versions od DLog, CDH, DDH imply their ordinary counterparts. Moreover, the assumptions are *equivalent* to the ordinary versions on group actions with twists. Also, note that, while [MZ22] prove the equivalence of ordinary DLog and CDH, their proof does not necessarily apply to the $^{\pm}$ versions, and an equivalence between these versions may be incomparable since it would start from a stronger property, but also reach a stronger conclusion.

Our Result. We now show that, the negation assumptions allow us to lift to security under twists, generically.

Theorem 4.6. *Let $(\mathbb{G}, \mathcal{X}, *)$ be a group with $\mathcal{X} \subseteq \{0,1\}^m$ such that DLog$^{\pm}$ (resp. CDH$^{\pm}$, DDH$^{\pm}$) holds. Let $m' \geq 2m + \omega(\log \lambda)$. Then DLog$^{\pm}$ (resp. CDH$^{\pm}$, DDH$^{\pm}$) hold in $\mathsf{GGAM}^{\pm}_{\mathbb{G},m'}$, the GGAM with twists relative to group $\mathbb{G}$ and with label length m'.*

Proof. We prove the case of DDH, the other proofs being nearly identical. Let $\mathcal{A}^{\mathsf{GGAM}^{\pm}_{\mathbb{G},m}}$ be a supposed adversary for DDH$^{\pm}$ in $\mathsf{GGAM}^{\pm}_{\mathbb{G},m}$, the GGAM with twists and with label length m'. Let ϵ be the distinguishing advantage of $\mathcal{A}$, and q the polynomial number of queries. We construct a new adversary $\mathcal{B}$ for DDH$^{\pm}$ in the group action $(\mathbb{G}, \mathcal{X}, *)$ as follows.

- $\mathcal{B}$, on input $u^+, v^+, w^+, u^-, v^-, w^-$, will chose a random injective function Π from $\{0,1\}^{2m} \to \{0,1\}^{m'}$. To make $\mathcal{B}$ efficient, we will actually use Lemma 4.2 to efficiently simulate Π. For simplicity in the following proof, we will treat $\mathcal{B}$ as actually using a true random injection.
- $\mathcal{B}$ will compute $X = \Pi(x_\lambda, x_\lambda), U = \Pi(u^+, u^-), V = \Pi(v^+, v^-), W = \Pi(w^+, w^-)$.
- $\mathcal{B}$ will then run $\mathcal{A}(X, U, V, W)$ [13], simulating its queries as follows:
  - For queries to the group action (ℓ, g), $\mathcal{B}$ simulates the query by computing $(z_1, z_2) \leftarrow \Pi^{-1}(\ell)$, and then returning $\Pi(g * z_1, (-g) * z_2)$. For superposition queries, $\mathcal{B}$ simply runs this computation in superposition. Note that if we let $\Gamma(g) = (g * x_\lambda, (-g) * x_\lambda)$, then $\mathcal{B}$ simulates these queries exactly as prescribed above in our general framework, for constants $c_1 = 1, c_2 = -1$ and $y_1 = y_2 = x_\lambda$.
  - When $\mathcal{A}$ makes a twist query on label ℓ, $\mathcal{B}$ computes $(z_1, z_2) \leftarrow \Pi^{-1}(\ell)$, and then computes $\ell' = \Pi(z_2, z_1)$ and responds with ℓ'. For superposition queries, $\mathcal{B}$ simply runs this computation in superposition. Observe that the twist of $\Pi(\Gamma(g))$ is exactly $\Pi(\Gamma(-g))$.
- $\mathcal{B}$ then outputs whatever $\mathcal{A}$ outputs.

We now prove security via a sequence of hybrids.

Hybrid 0. . Here, we run $\mathcal{A}^{\mathcal{O}}(X, U = a * X, V = b * X, W = c * X)$ for a random injection L, where $X = L(0)$, a, b, c are uniform in $\mathbb{G}_\lambda$, and $*$ denotes the action defined by $\mathcal{O}$. Let p_0 be the probability $\mathcal{A}$ outputs 1.

Hybrid 1. Here, $\mathcal{B}$ is given $u^+, v^+, w^+, u^-, v^-, w^- = a * y, b * y, c * y, (-a)^y, (-b) * y, (-c) * y$, and simulates $\mathcal{A}$ as described above. Let p_1 be the probability $\mathcal{A}$ (and hence $\mathcal{B}$) outputs 1. Observe that $X, U, V, W = L(0), L(a), L(b), L(c)$, where L is the implicit labeling function $L(g) = \Pi(g * x_\lambda, (-g) * x_\lambda)$. Since $\mathcal{B}$ simulates twist queries by mapping $L(g) \mapsto L(-g)$, $\mathcal{B}$ correctly simulates the view of $\mathcal{A}$ in **Hybrid 0**, except that $\mathcal{O}'$ and the twist oracle operate on values $\Pi(z_1, z_2)$ that might not be in the image of L. But we can invoke Lemma 4.1 to conclude that $|p_0 - p_1| \leq O(q \times 2^{2m-m'}) = q \times \mathsf{negl}(\lambda) = \mathsf{negl}(\lambda)$.

[13] Recall that in the definition of DDH, the adversary is only given U, V, W. However, in the generic group action model, we additionally give all parties the starting point X.

Hybrid 2. Here, $\mathcal{B}$ is $u^+, v^+, w^+, u^-, v^-, w^- = a * y, b * y, (a+b) * y, (-a)^y, (-b) * y, (-a-b) * y$, and simulates $\mathcal{A}$ as described above. Let p_2 be the probability $\mathcal{A}$ (and hence $\mathcal{B}$) outputs 1. By Assumption 4.5, $|p_1 - p_2| \leq \mathsf{negl}(\lambda)$.

Hybrid 3. Now we run $\mathcal{A}^{\mathcal{O}}(X, U = a * X, V = b * X, W = (a+b) * X)$. Let p_3 be the probability $\mathcal{A}$ outputs 1. By a similar argument for going from **Hybrid 0** to **Hybrid 1**, we conclude that $|p_2 - p_3| \leq \mathsf{negl}(\lambda)$ is negligible. Piecing everything together, we have that $\epsilon = |p_0 - p_3| \leq \mathsf{negl}(\lambda)$, thereby proving DDH$^{\pm}$ holds in $\mathsf{GGAM}^{\pm}_{\mathbb{G},m'}$. □

4.4 Computing Banknotes With Complementary Serial Numbers

Here, we prove that it is hard in generic group action to compute two banknotes for our scheme with "complementary" serial numbers that sum to zero.

Theorem 4.7. *Let $(\mathbb{G}, \mathcal{X}, *)$ be a group with $\mathcal{X} \subseteq \{0,1\}^m$ such that DDH holds (Assumption 2.5). Let $m' \geq 4m + \omega(\log \lambda)$. Let $(\mathsf{Gen}^{\mathsf{GGAM}_{\mathbb{G},m'}}, \mathsf{Ver}^{\mathsf{GGAM}_{\mathbb{G},m'}})$ be the quantum money construction from Construction 3.1, using the generic group action $\mathsf{GGAM}_{\mathbb{G},m'}$. Consider a QPT adversary $\mathcal{B}^{\mathsf{GGAM}_{\mathbb{G},m'}}$ making queries to $\mathsf{GGAM}_{\mathbb{G},m'}$, which takes as input the security parameter λ, and outputs a serial number $h \in \mathbb{G}_\lambda$ and two potentially entangled states $\$_1, \$_2$, which it tries to pass off as two banknnotes. For all such $\mathcal{B}$, there exists a negligible $\mathsf{negl}(\lambda)$ such that the following holds:*

$$\Pr\left[\mathsf{Ver}^{\mathsf{GGAM}_{\mathbb{G},m'}}(h, \$_1) = \mathsf{Ver}^{\mathsf{GGAM}_{\mathbb{G},m'}}(-h, \$_2) = 1 : (h, \$_1, \$_2) \leftarrow \mathcal{B}^{\mathsf{GGAM}_{\mathbb{G},m'}}(1^\lambda)\right] \leq \mathsf{negl}(\lambda) \ .$$

Notice that the statement above is *almost* the statement that $(\mathsf{Gen}^{\mathsf{GGAM}_{\mathbb{G},m'}}, \mathsf{Ver}^{\mathsf{GGAM}_{\mathbb{G},m'}})$ is a quantum lightning scheme, except that the second banknote is verified with respect to $-h$ instead of h. Theorem 4.7 is therefore not quite enough to prove the security of our scheme, since it could be the case that it is possible to output many banknotes with the same serial number, even if it is impossible to output two with complementary numbers. We give a different proof below in Section 4.5 based on a stronger assumption which proves our scheme quantum lightning. We use the result here as a warm-up to our later result, which is based on a more complex assumption. Moreover, Theorem 4.7 lets us prove that it is generically hard to output the uniform superposition $\frac{1}{\sqrt{|\mathbb{G}_\lambda|}} \sum_{g \in \mathbb{G}_\lambda} |L(g)\rangle$, which is just the banknote $|\mathbb{G}^0_\lambda * L(0)\rangle$ with serial number 0. We state and prove this fact before proving Theorem 4.7.

Corollary 4.8. *Let $(\mathbb{G}, \mathcal{X}, *)$ be a group with $\mathcal{X} \subseteq \{0,1\}^m$ such that DDH holds (Assumption 2.5). Let $m' \geq 4m + \omega(\log \lambda)$. Let L be the labeling function for the generic group action $\mathsf{GGAM}_{\mathbb{G},m'}$. Then for any QPT adversary $\mathcal{A}^{\mathsf{GGAM}_{\mathbb{G},m'}}$ making queries to $\mathsf{GGAM}_{\mathbb{G},m'}$ which outputs a state ρ, there exists a negligible $\mathsf{negl}(\lambda)$ such that $\langle \mathbb{G}^0_\lambda * L(0) | \rho | \mathbb{G}^0_\lambda * L(0) \rangle \leq \mathsf{negl}(\lambda)$.*

Proof. Consider an adversary $\mathcal{A}^{\mathsf{GGAM}_{\mathbb{G},m'}}$ outputting a mixed state ρ and let $\epsilon = \langle \mathbb{G}^0_\lambda * L(0) | \rho | \mathbb{G}^0_\lambda * L(0) \rangle \leq \mathsf{negl}(\lambda)$. We will assume for simplicity that we can project exactly onto $|\mathbb{G}^0_\lambda * L(0)\rangle$; using our verifier from Section 3 introduces negligible error that is easily accounted for. By applying this projection to ρ, we have that $\mathcal{A}^{\mathsf{GGAM}_{\mathbb{G},m'}}$ outputs $|\mathbb{G}^0_\lambda * L(0)\rangle$ with probability ϵ. We will therefore assume we have the state $|\mathbb{G}^0_\lambda * L(0)\rangle$.

Apply in superposition the map $|x\rangle \mapsto |x, x\rangle$. Now we have the state

$$\frac{1}{\sqrt{|\mathbb{G}_\lambda|}} \sum_{g \in \mathbb{G}_\lambda} |L(g), L(g)\rangle$$

We can equivalently write this state as:

$$\frac{1}{\sqrt{|\mathbb{G}_\lambda|}}\sum_{h\in\mathbb{G}_\lambda}|\mathbb{G}_\lambda^h * L(0)\rangle|\mathbb{G}_\lambda^{-h} * L(0)\rangle$$

We therefore apply our algorithm Findh from Theorem 3.5 to the first register. The output will be a random serial number h, and the state will collapse to $|\mathbb{G}_\lambda^h * L(0)\rangle|\mathbb{G}_\lambda^{-h} * L(0)\rangle$. We output this, which solves the problem in Theorem 4.7. Thus, we conclude that ϵ must be negligible. □

We now turn to proving Theorem 4.7.

Proof. Consider an adversary $\mathcal{B}^{\mathsf{GGAM}_{\mathbb{G},m'}}$, and define:

$$\epsilon := \Pr\left[\mathsf{Ver}^{\mathsf{GGAM}_{\mathbb{G},m'}}(h,\$_1)=\mathsf{Ver}^{\mathsf{GGAM}_{\mathbb{G},m'}}(-h,\$_2)=1:(h,\$_1,\$_2)\leftarrow\mathcal{B}^{\mathsf{GGAM}_{\mathbb{G},m'}}(1^\lambda)\right]$$

We will assume that $\mathsf{Ver}^{\mathsf{GGAM}_{\mathbb{G},m'}}(h,\$)$ projects onto the correct banknote $|\mathbb{G}_\lambda^h * L(0)\rangle$; this assumption only introduces a negligible error which is easily accounted for. Therefore, with probability ϵ, $\mathcal{B}$ outputs h and exactly the states $|\mathbb{G}_\lambda^h * L(0)\rangle, |\mathbb{G}_\lambda^{-h} * L(0)\rangle$.

We now construct an adversary $\mathcal{A}$ for DDH on the group action $(\mathbb{G},\mathcal{X},*)$. $\mathcal{A}$, on input (u,v,w), will choose a random injection $\Pi:\{0,1\}^{4m}\to\{0,1\}^{m'}$. It will then compute $X=\Pi(x_\lambda,u,v,w)$. $\mathcal{A}$ will then run $\mathcal{B}(X)$, simulating its queries (ℓ,g) to the group action as follows: compute $(z_1,z_2,z_3,z_4)\leftarrow\Pi^{-1}(\ell)$, and then return $\Pi(g*z_1,(-g)*z_2,g*z_3,(-g)*z_4)$. For superposition queries, $\mathcal{A}$ simply runs this computation in superposition. Note that if we let $\Gamma(g)=(g*x_\lambda,(-g)*u,g*v,(-g)*w)$, then $\mathcal{A}$ simulates these queries exactly as prescribed above in our general framework, for constants $c_1=1,c_2=-1,c_3=1,c_4=-1$ and $(y_1,y_2,y_3,y_4)=(x_\lambda,u,v,w)$.

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 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 $\ell\mapsto\Pi^{-1}(\ell)=(z_1,z_2,z_3,z_4)$
  - Now map $(z_1,z_2,z_3,z_4)\mapsto\ell'=\Pi(z_2,z_1,z_4,z_3)$.

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

By applying Lemma 4.1, we can conclude that $\$_1,\$_2$ are actually superpositions over elements of the form $L(g)=\Pi(g*z_1,(-g)*z_2,g*z_3,(-g)*z_4)$ for varying g. Then using our characterization of the accepting states of Ver, we see that both runs of Ver simulteneously accept with probability ϵ, and in this case $\$'_1=|\mathbb{G}_\lambda^h * L(0)\rangle, \$'_2=|\mathbb{G}_\lambda^{-h} * L(0)\rangle$.

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

- $u=a*x_\lambda, v=b*x_\lambda, w=(a+b)*x_\lambda$. Let $\ell=L(g)=\Pi(g*z_1,(-g)*z_2,g*z_3,(-g)*z_4)=\Pi(g*x_\lambda,(a-g)*x_\lambda,(b+g)*x_\lambda,(a+b-g)*x_\lambda)$, which maps to $\ell'=\Pi((a-g)*x_\lambda,g*x_\lambda,(a+b-g)*x_\lambda,(b+g)*x_\lambda)=L(a-g)$.

Therefore, $|\mathbb{G}_\lambda^{-h} * L(0)\rangle$ maps to

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

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 $\frac{1}{2}(1-\epsilon)+\epsilon = \frac{1+\epsilon}{2}$.

- $u = a * x_\lambda, v = b * x_\lambda, w = c * x_\lambda$ with $c \neq a+b$. In this case, $\ell = L(g) = \Pi(g * x_\lambda, (a-g) * x_\lambda, (b+g) * x_\lambda, (c-g) * x_\lambda)$ maps to $\ell' = \Pi((a-g) * x_\lambda, g * x_\lambda, (c-g) * x_\lambda, (b+g) * x_\lambda)$. However, ℓ' is *not* equal to $L(g')$ for any g'. Indeed, in order for $\ell' = L(g')$, we get several equations:

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

The first two equations require that $g' = a - g$, while the last two require that $g' = c - b - g \neq a - g$. 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. □

4.5 Security of our Quantum Lightning Scheme

Here, we prove the generic security of our quantum lightning scheme (Construction 3.1). 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.

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_λ 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_λ 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 4.9. 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 λ such that

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

Above, M_c is the in-place (or "minimal") oracle mapping $y \mapsto c * y$, and M_c^1 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 4.10. 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 λ such that

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

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

The following lemma is straightforward:

Lemma 4.11. *If D2X/std holds on a group action $(\mathbb{G}, \mathcal{X}, *)$, then so does D2X/min*

Proof. We simply use the oracles S_c^1, S_{-c}^1 to simulate the oracle M_c^1 in the obvious way. □

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

Theorem 4.12. *Let $(\mathbb{G}, \mathcal{X}, *)$ be a group with $\mathcal{X} \subseteq \{0,1\}^m$ such that D2X/min holds (Assumption 4.9). 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 3.1, 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 ϵ be the probability that $\mathcal{B}$ wins. We will assume that $\mathsf{Ver}^{\mathsf{GGAM}_{\mathbb{G},m'}}(h,\$)$ projects onto the correct banknote $|\mathbb{G}_\lambda^h * L(0)\rangle$; this assumption only introduces a negligible error which is easily accounted for. Therefore, with probability ϵ, $\mathcal{B}$ outputs h and exactly two copies of the state $|\mathbb{G}_\lambda^h * L(0)\rangle$.

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 (ℓ, 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 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 4.1, 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 Ver, we see that both runs of Ver simulteneously accept with probability ϵ, 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, ℓ' 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. □

5 On Quantum Knowledge Assumptions and Algebraic Adversaries

In this section, we explore knowledge assumptions in the quantum setting, as well the algebraic model for group actions. We find significant issues with both settings. Nevertheless, we give a second security proof for our quantum lightning scheme (Construction 3.1), this time using knowledge assumptions.

5.1 The Knowledge of Group Element Assumption (KGEA)

Here, we discuss a new assumption that we define, called the Knowledge of Group Element Assumption (KGEA). This is an analog of the classical Knowledge of Exponent Assumption (KEA) [Dam92], but adapted for quantum adversaries and group actions. It can also be seen as an adaptation of the Knowledge of Path assumption of [LMZ23], specialized to group actions. Despite coming from plausible origins, however, we will see that the assumption is, in fact, false. This leads to concerns over the more general Knowledge of Path assumption. We give a candidate replacement assumption that avoids our attack, but more cryptanalysis is needed to understand the new assumption.

The Knowledge of Group Element Assumption (KGEA). This assumption states, informally, that any algorithm that produces a set element y must "know" g such that $y = g * x_\lambda$. Implicit in this assumption is the requirement that it is hard to obliviously sample set elements; we discuss later how to model security when oblivious sampling is possible. In the classical setting, the KGEA assumption would be formalized as follows:

Assumption 5.1. The *classical knowledge of group element assumption* (C-KGEA) holds on a group action $(\mathbb{G}, \mathcal{X}, *)$ if the following is true. For any probabilistic polynomial time (PPT) adversary $\mathcal{A}$, there exists a PPT "extractor" $\mathcal{E}$ and a negligible ϵ such that:

$$\Pr\left[y \in \mathcal{X} \wedge y \neq g * x_\lambda : {}^{y \leftarrow \mathcal{A}(1^\lambda; r)}_{g \leftarrow \mathcal{E}(1^\lambda, r)} \right] \leq \epsilon(\lambda) \ .$$

Above, r are the random coins given to $\mathcal{A}$, which are also given to $\mathcal{E}$, and the probability is taken over uniform r and any additional randomness of $\mathcal{E}$.

In other words, if $\mathcal{A}$ outputs any set element, it must "know" how to derive that set element from x_λ, since it can compute g such that $y = g * x_\lambda$ using $\mathcal{E}$ and its random coins. Note that once the random coins are fixed, $\mathcal{A}$ is deterministic.

As observed by [LMZ23], when moving to the quantum setting, the problem with Assumption 5.1 is that quantum algorithms do not have to flip random coins to generate randomness, and instead their output may be a measurement applied to a quantum state, the result being inherently randomized even if the quantum state is fixed. Thus there is no meaningful way to give the same random coins to $\mathcal{E}$.

The solution used in [LMZ23] is to, instead of giving $\mathcal{E}$ the same inputs as $\mathcal{A}$, give $\mathcal{E}$ the remaining state of $\mathcal{A}$ at the *end* of the computation. This requires some care, since an algorithm can of course forget any bit of information by simply throwing it away. A more sophisticated way to lose information is to perform other measurements on the state, say measuring in the Fourier basis. The solution in [LMZ23] is to require that $\mathcal{A}$ makes no measurements at all, *except* for measuring the final output. Note that the Principle of Delayed Measurement implies that it is always possible without loss of generality to move all measurements to the final output. Then $\mathcal{E}$ is given both the output and the remaining quantum state of $\mathcal{A}$, and tries to compute g. Note that in the classical setting, if we restrict to *reversible* $\mathcal{A}$, this formulation of giving $\mathcal{E}$ the final state of $\mathcal{A}$ is equivalent to given $\mathcal{E}$ the randomness, since the randomness can be computed by reversing $\mathcal{A}$. Similar to how we can assume a quantum $\mathcal{A}$ makes all its measurements at the end, in we can always assume without loss of generality that a classical $\mathcal{A}$ is reversible. Thus, in the classical setting these two definitions coincide. Adapting to our setting, this approach yields the following assumption:

Assumption 5.2. The *quantum knowledge of group element assumption* (Q-KGEA) holds on a group action $(\mathbb{G}, \mathcal{X}, *)$ if the following is true. For any quantum polynomial time (QPT) adversary $\mathcal{A}$ which performs no measurements except for its final output, there exists a QPT extractor $\mathcal{E}$ and negligible ϵ such that

$$\Pr\left[y \in \mathcal{X} \wedge y \neq g * x_\lambda : {}^{(y,|\psi\rangle)\leftarrow\mathcal{A}(1^\lambda)}_{g\leftarrow\mathcal{E}(y,|\psi\rangle)}\right] \leq \epsilon(\lambda) \ .$$

Above, y is considered as the output of $\mathcal{A}$, and the only measurements applied to $\mathcal{A}$ is the measurement of y to obtain the output.

Our Attack on Q-KGEA. Here, we show that Q-KGEA is *false*.

Theorem 5.3. *On any group action where the discrete logarithm assumption holds (Assumption 2.4), Q-KGEA (Assumption 5.2) does not hold.*

Proof. Our proof will use the Findh algorithm developed in Section 3.2. We first recall the functionality guaranteed by the algorithm. The algorithm takes as input the state $|\mathbb{G}_\lambda^h * x_\lambda\rangle = \frac{1}{\sqrt{|\mathbb{G}_\lambda|}} \sum_{g\in\mathbb{G}_\lambda} \chi(g,h)|g * x_\lambda\rangle$, and outputs h, while leaving $|\mathbb{G}_\lambda^h * x_\lambda\rangle$ intact. In other words, it maps $|\mathbb{G}_\lambda^h * x_\lambda\rangle \mapsto |\mathbb{G}_\lambda^h * x_\lambda\rangle|h\rangle$.

Now, recall that the $|\mathbb{G}_\lambda^h * x_\lambda\rangle$ form a basis. In particular, observe that $|x_\lambda\rangle = \frac{1}{\sqrt{|\mathbb{G}_\lambda|}} \sum_h |\mathbb{G}_\lambda^h * x_\lambda\rangle$. Therefore, we have that

$$\mathsf{Findh}|x_\lambda\rangle = \frac{1}{\sqrt{|\mathbb{G}_\lambda|}} \sum_h |\mathbb{G}_\lambda^h * x_\lambda\rangle|h\rangle$$

We can now apply an arbitrary h-dependent phase to the state, and then uncompute h. The result is that we have applied an arbitrary phase to whatever state we started from, but in the

Fourier domain of the group. That is, let $F : \mathbb{G} \mapsto \mathbb{R}$ be an arbitrary function. We can apply the phase $|h\rangle \mapsto e^{iF(h)}|h\rangle$, and then uncompute h. The result is that $|x_\lambda\rangle$ maps to

$$\frac{1}{\sqrt{|\mathbb{G}_\lambda|}} \sum_h e^{iF(h)} |\mathbb{G}_\lambda^h * x_\lambda\rangle = \frac{1}{|\mathbb{G}_\lambda|} \sum_g |g * x_\lambda\rangle \left(\sum_h \chi(g,h) e^{iF(h)} \right) \tag{5.1}$$

Now suppose we apply Q-KGEA to the algorithm producing this state. When we measure the register, all we get is a sample of $|g * x_\lambda\rangle$ according to some distribution, with no side information. The Q-KGEA assumption then implies an algorithm $\mathcal{E}$ which can recover g just given $|g * x_\lambda\rangle$. Therefore, if we can guarantee that measuring the state in Equation 5.1 gives a uniform choice of g, then $\mathcal{E}$ must be solving discrete logarithms, breaking Assumption 2.4.

It is not hard to devise a function F which makes the resulting sample g uniform. For example, if $\mathbb{G} = \mathbb{Z}_N$ for an odd integer N, we can let $F(h) = 2\pi h^2/N$. Then the probability of observing g is

$$\frac{1}{|\mathbb{G}_\lambda|^2} \times \left| \sum_h e^{i2\pi(gh+h^2)/N} \right|^2 = \frac{1}{|\mathbb{G}_\lambda|^2} \times |\mathbb{G}_\lambda| = \frac{1}{|\mathbb{G}_\lambda|}$$

as desired, where above we used the fact about quadratic Gauss sums that $\sum_h e^{i2\pi(gh+h^2)/N}$ is equal to $|\mathbb{G}_\lambda|^{-1/2}$, up to phase. □

Our Modified Knowledge Assumption. We propose a simple way to circumvent the attack above. Our basic observation is that, while the attack in Theorem 5.3 allows for obliviously sampling elements in arbitrary group actions, it does not appear useful for actually breaking cryptosystems. After all, all the attack is doing is sampling random set elements, which can anyway be sampled easily by choosing a random group element g and computing $g * x_\lambda$. Thus, while strictly speaking violating the knowledge assumption, the attack appears useless for actually breaking cryptosystems.

More generally, for "nice" cryptographic games (which we will define shortly), in particular games that only use the group action interface and do not themselves obliviously sample elements, it seems that giving the adversary the ability to obliviously sample elements is no help in breaking the game. We therefore postulate that, for any adversary $\mathcal{A}$ that wins such a nice game, there is a different adversary $\mathcal{A}'$ for which the KGEA assumption can be appled, yielding an extractor *for that* $\mathcal{A}'$. Thus, even if the original $\mathcal{A}$ can obliviously sample elements, we essentially assume that $\mathcal{A}'$ cannot, and therefore $\mathcal{E}$ is possible. We now make this intuition precise.

We first introduce the notion of generic group action games. Note that we will only be interested in *games* that are given by generic algorithms; we will always treat the adversary as non-generic.

Briefly, a generic group action game is given by an interactive algorithm ("challenger") Ch. Ch is limited to only performing group action computations that are "generic" and only interacts with the group action through oracles implementing the group action interface. Specifically, a generic algorithm is an oracle-aided algorithm $\mathcal{B}$ that has access to oracles $\mathsf{GA} = (\mathsf{Start}, \mathsf{Act}, \mathsf{Mem})$. Here, Start is the oracle that takes as input the empty query, and outputs a string $\tilde{x}$ representing x_λ. Act is the oracle that takes as input a group element $g \in \mathbb{G}_\lambda$ and a string $\tilde{y}$ representing a set element y, and outputs a string $\tilde{z}$ representing $z = g * x$. Finally, Mem is a membership testing oracle, that tests is a given string $\tilde{x}$ represents an actual set element. From a generic game, we obtain a standard model game by implementing the oracles $\mathsf{Start}, \mathsf{Act}, \mathsf{Mem}$ with the algorithms for an actual group action: Start outputs the actual set element x_λ, Act is the group action $*$, and

Mem is the membership tester for the set $\mathcal{X}_\lambda$. For a concrete group action $(\mathbb{G}, \mathcal{X}, *)$, we denote this standard-model game by $\mathsf{Ch}^{(\mathbb{G},\mathcal{X},*)}$.

For any algorithm $\mathcal{A}$, we say the algorithm $\delta(\lambda)$-breaks $\mathsf{Ch}^{(\mathbb{G},\mathcal{X},*)}$ if $\mathsf{Ch}^{(\mathbb{G},\mathcal{X},*)}(1^\lambda)$ outputs 1 with probability at least $\delta(\lambda)$ when interacting with $\mathcal{A}$.

We say that Ch is one-round if it sends a single classical string to $\mathcal{A}$, and then receives a single quantum message from $\mathcal{A}$, before deciding if $\mathcal{A}$ wins.

We now give our modified KGEA assumption.

Assumption 5.4. The *quantum modified knowledge of group element assumption* (Q-mKGEA) holds on a group action $(\mathbb{G}, \mathcal{X}, *)$ if the following is true. Consider a one-round generic group action game Ch and any quantum polynomial time (QPT) adversary $\mathcal{A}$ that $1-\delta$-breaks $\mathsf{Ch}^{(\mathbb{G},\mathcal{X},*)}$ for a negligible δ. Write the message from $\mathcal{A}$ to $\mathsf{Ch}^{(\mathbb{G},\mathcal{X},*)}$ as $\rho_{1,2}$, as a joint system over two registers $1, 2$. Consider measuring the first register, to obtain a set element y. Denote this as $(y, |\psi\rangle) \leftarrow \mathcal{A}'(1^\lambda) \Leftrightarrow \mathsf{Ch}^{(\mathbb{G},\mathcal{X},*)}(1^\lambda)$. Then for all such $\delta, \mathcal{A}, \mathsf{Ch}$, there exists another negligible δ', a QPT $\mathcal{A}'$ that also $1-\delta'$-breaks $\mathsf{Ch}^{(\mathbb{G},\mathcal{X},*)}$, and moreover there exists a QPT extractor $\mathcal{E}$ and negligible ϵ such that

$$\Pr\left[y \in \mathcal{X} \wedge y \neq g * x_\lambda : \substack{(y,|\psi\rangle)\leftarrow\mathcal{A}'(1^\lambda)\Leftrightarrow\mathsf{Ch}^{(\mathbb{G},\mathcal{X},*)}(1^\lambda) \\ g\leftarrow\mathcal{E}(y,|\psi\rangle)}\right] \leq \epsilon(\lambda) \ .$$

Intuitively, this assumption says that if $\mathcal{A}$ wins some game, we might not be able to apply the KGEA extractor to it. However, there is some other $\mathcal{A}'$ that also wins the game, and that we *can* apply the KGEA extractor to.

Remark 5.5. Our solution with Assumption 5.4 also resolves the problem that, for group actions based on isogenies over elliiptic curves, it is *classically* possible to sample certain set element obliviously, thus violating the plain KGEA assumption. A different remedy used in [LMZ23] explicitly assumes a probabilistic classical procedure $S()$ for obliviously sampling set elements, and modifies the KGEA assumption so that the extractor either outputs (1) an explanation relative to x_λ *or* (2) an explanation relative to some input y together with the random coins r that are fed into S so that $y = S(r)$. This approach works, but is not robust, in the sense that if another sampling procedure is found, it would contradict even the modified assumption. Moreover, our attack in Theorem 5.3 shows that, when specialized to group actions, even this approach fails, since there is a quantum procedure for sampling elements that has no randomness at all, and therefore can not be explained. Our solution is robust to new sampling procedures being found as well as our quantum sampler. Nevertheless, more cryptanalysis is needed to understand if the assumption is sound.

5.2 Quantum Lightning Security Using Q-mKGEA

Here, we give an alternative and incomparable proof of security of our quantum lightning construction to the proof given in Section 4. Our proof here does not require generic group actions, but instead requires our Q-mKGEA assumption. Thus, it achieves a trade-off by giving a standard-model justification, but the computational assumption is more suspect.

The Discrete Log Assumption, with Help. We now define a strengthening of the Discrete Log assumption (Assumption 2.4), which allows the adversary limited query access to a computational Diffie Hellman (CDH) oracle.

Assumption 5.6. We say that the *Discrete Log with a single minimal CDH query* assumption (DLog/1-minCDH) assumption holds if the following is true. For any QPT adversary $\mathcal{A}$ playing the following game, parameterized by λ, there is a negligible ϵ such that $\mathcal{A}$ wins with probability at most $\epsilon(\lambda)$:

- The challenger, on input λ, chooses a random $g \in \mathbb{G}_\lambda$. It sends λ to $\mathcal{A}$
- $\mathcal{A}$ submits a superposition query $\sum_{y\in\mathcal{X},z\in\{0,1\}^*} \alpha_{y,z}|y,z\rangle$. Here, y is a set element that forms the query, and z is the internal state of the adversary when making the query. The challenger responds with $\sum_{y\in\mathcal{X},z\in\{0,1\}^*} \alpha_{y,z}|(-g)*y,z\rangle$ [14].
- The challenger sends $g*x$ to $\mathcal{A}$.
- $\mathcal{A}$ outputs a guess g' for g. It wins if $g'=g$.

Note that Assumption 5.6 uses a "minimal" oracle for the CDH oracle, meaning is replaces y with $(-g)*y$. This is only a possibility because $y \mapsto (-g)*y$ is reversible; otherwise the query would not be unitary. The minimal oracle, however, is somewhat non-standard. So we here define a slightly different assumption which uses "standard" oracles:

Assumption 5.7. We say that the *Discrete Log with a double standard CDH query* assumption (DLog/2-stdCDH) assumption holds if the following is true. For any QPT adversary $\mathcal{A}$ playing the following game, parameterized by λ, there is a negligible ϵ such that $\mathcal{A}$ wins with probability at most $\epsilon(\lambda)$:

- The challenger, on input λ, chooses a random $g \in \mathbb{G}_\lambda$. It sends λ to $\mathcal{A}$.
- $\mathcal{A}$ submits a superposition query $\sum_{y\in\mathcal{X},w,z\in\{0,1\}^*} \alpha_{y,w,z}|y,w,z\rangle$. Here, y is a set element that forms the query, w is a string that forms the response register, and z is the internal state of the adversary when making the query. The challenger responds with $\sum_{y\in\mathcal{X},w,z\in\{0,1\}^*} \alpha_{y,w,z}|y,w\oplus[(-g)*y],z\rangle$.
- $\mathcal{A}$ submits a second superposition query $\sum_{y\in\mathcal{X},w,z\in\{0,1\}^*} \alpha_{y,w,z}|y,w,z\rangle$. The challenger responds with $\sum_{y\in\mathcal{X},w,z\in\{0,1\}^*} \alpha_{y,w,z}|y,w\oplus[g*y],z\rangle$.
- The challenger sends $g*x$ to $\mathcal{A}$.
- $\mathcal{A}$ outputs a guess g' for g. It wins if $g'=g$.

Lemma 5.8. *If DLog/2-stdCDH (Assumption 5.7) holds in a group action, then so does DLog/1-minCDH (Assumption 5.6).*

Proof. Like the proof of Lemma 4.11, Lemma 5.8 follows by using the two standard oracle queries to simulate a single minimal oracle query. □

From this point forward, we will use DLog/1-minCDH as our assumption; Lemma 5.8 then shows that we could have instead used DLog/2-stdCDH.

[14] Note that this operation is unitary and efficiently computable since $y \mapsto (-g)*y$ is efficiently computable and efficiently reversible given g.

The security proof. We are now ready to formally state and prove security.

Theorem 5.9. *Assuming Q-mKGEA (Assumption 5.4) and DLog/1-minCDH (Assumption 5.6) both hold on a group action $(\mathbb{G}, \mathcal{X}, *)$, then Construction 3.1 is a quantum lightning scheme.*

Remark 5.10. Before proving Theorem 4.12, we briefly discuss how to handle the case of non-uniform attackers, since in this setting quantum lightning is insecure without some modifications. Note that even against non-uniform attackers, DLog/1-minCDH still plausibly holds. However, Q-KGEA certainly does not, as a non-uniform attacker may have a y hard-coded for which it does not know the discrete log with x_λ. As discussed in Section 2, there are several possibilities.

- The first is to restrict to non-uniform attackers that only have classical advice. While classical advice does not appear to be useful in breaking Construction 3.1, it still allows for breaking Q-KGEA; thus while our scheme may be secure in this setting, the security proof would be vacuous.

- The second is to use a probabilistically generated group action, and define Q-KGEA and DLog/1-minCDH accordingly. For quantum money security, it would suffice to have Gen create the parameters of the group action and then include them in the serial number, since the serial number is generated honestly. For quantum lightning security, we would instead need the parameters to be generated by a trusted third party and then placed in a common random string (CRS).

- The final option is to use the human ignorance approach [Rog06], where we explicitly state our security theorem as transforming a quantum lightning adversary into a Q-KGEA adversary; while such Q-KGEA adversaries exist in the non-uniform setting without a CRS, they are presumably unknown to human knowledge. As a consequence, a quantum lightning attacker, while existing, would likewise be unknown to human knowledge.

For simplicity, state and prove Theorem 4.12 in the uniform setting; either probabilistically generating the group action or using human ignorance would require straightforward modifications.

We now are ready to prove Theorem 4.12.

Proof. Consider a QPT quantum lightning adversary $\mathcal{A}$ which breaks security with non-negligible success probability ϵ. Since an adversary can always tell if it succeeded by running Ver, we can run $\mathcal{A}'$ multiple times to boost the probability of a successful break. In particular, we can run $\mathcal{A}'$ for $\lambda\epsilon$, and at except with probability $1 - 2^{-\Theta(\lambda)}$, at least one of the runs will succeed. This allows us to conclude without loss of generality that $\mathcal{A}'$ has success probability $1 - 2^{-\Theta(\lambda)}$. We can then invoke Q-mKGEA (Assumption 5.4) to arrive at an adversary $\mathcal{A}$ which also breaks quantum lightning security with high success probability.

By Theorem 3.3, we know that if $\mathcal{A}$ outputs a serial number h, the states outputted are exponentially close to two copies of $|\mathbb{G}_\lambda^h * x_\lambda\rangle$.

For simplicity in the following proof, we will assume the probability of passing verification is actually 1; it is straightforward to adapt the proof to the case of negligible error.

Next, we purify $\mathcal{A}$, and assume that before measurement, $\mathcal{A}$ outputs a pure state $|\psi\rangle$. By our assumption that the success probability is 1, $|\psi\rangle$ will have the form

$$|\psi\rangle = \sum_h \alpha_h |\phi_h\rangle |\mathbb{G}_\lambda^h * x_\lambda\rangle |\mathbb{G}_\lambda^h * x_\lambda\rangle = \frac{1}{|\mathbb{G}_\lambda|} \sum_h \alpha_h |\phi_h\rangle \chi(h, g_1 + g_2) |g_1 * x\rangle_{\mathcal{M}_1} |g_2 * x\rangle_{\mathcal{M}_2} \ .$$

Above, $|\phi_h\rangle$ are arbitrary normalized states representing whatever state the adversary contains after outputting its banknotes, and $\sum_h \|\alpha_h\|^2 = 1$.

Now consider the adversary $\mathcal{B}$ which first constructs $|\psi\rangle$, and then measures the register $\mathcal{M}_2$ to obtain $y_2 = g_2 * x$.

Claim 5.11. *g_2 is uniform in $\mathbb{G}$.*

Proof. Consider additionally measuring $\mathcal{M}_1$ in the basis $\{|\mathbb{G}_\lambda^h * x_\lambda\rangle\}$. This this measurement is on a different register than the measurement on $\mathcal{M}_2$, measuring $\mathcal{M}_1$ does not affect the output distribution of $\mathcal{M}_2$ (though the results may be correlated). But the measurement on $\mathcal{M}_1$ determines h, and conditioned on h, $\mathcal{M}_2$ collapses to $|\mathbb{G}_\lambda^h * x_\lambda\rangle$. Regardless of what h is, measuring $|\mathbb{G}_\lambda^h * x_\lambda\rangle$ gives a uniformly random element in $\mathcal{X}$. Thus, even without measuring $\mathcal{M}_1$, the measurement of $\mathcal{M}_2$ gives a uniform element in $\mathcal{X}$. □

Therefore, after measuring $\mathcal{M}_2$, the state $|\psi\rangle$ then collapses to

$$|\psi_{g_2 * x_\lambda}\rangle := \frac{1}{\sqrt{|\mathbb{G}_\lambda|}} \sum_h \alpha_h |\phi_h\rangle \chi(h, g_1 + g_2) |g_1 * x\rangle_{\mathcal{M}_1} \quad .$$

Claim 5.12. *There is a QPT procedure* Map *such that* $\mathsf{Map}(g, |\psi_y\rangle) = |\psi_{g*y}\rangle$.

Proof. Map simply applies the map $y \mapsto (-g) * y$ to $\mathcal{M}_1$ in superposition. Then we have that:

$$\begin{aligned}\mathsf{Map}(g, |\psi_{g_2 * x_\lambda}\rangle) &= \frac{1}{\sqrt{|\mathbb{G}_\lambda|}} \sum_h \alpha_h |\phi_h\rangle \chi(h, g_1 + g_2) |(g_1 - g) * x\rangle_{\mathcal{M}_1} \\ &= \frac{1}{\sqrt{|\mathbb{G}_\lambda|}} \sum_h \alpha_h |\phi_h\rangle \chi(h, g_1' + g + g_2) |g_1' * x\rangle_{\mathcal{M}_1} = |\psi_{(g+g_2)*y}\rangle = |\psi_{g*(g_2*y)}\rangle\end{aligned}$$

Above we used the change of variables $g_1' = g_1 - g$. □

Now we invoke Q-KGEA (Assumption 5.2) on the adversary $\mathcal{B}$. Since $\mathcal{B}$ always outputs a valid set element, this means there is another QPT algorithm $\mathcal{E}$ such that

$$\Pr[\mathcal{E}(g_2 * x_\lambda, |\psi_{g_2 * x_\lambda}\rangle) = g_2] \geq 1 - \mathsf{negl}(\lambda)$$

Above, the probability is over $g_2 * x_\lambda$, as well as any randomness incurred when executing $\mathcal{E}$. We note by a simple random self-reduction that we can insist the above probability holds for *all* $g_2 * x_\lambda$, where the randomness is only over $\mathcal{E}$. Indeed, given $|\psi_{g_2 * x_\lambda}\rangle, g_2 * x_\lambda$, we can choose a random g and compute $g_2' * x_\lambda$ as $g * (g_2 * x_\lambda)$ where $g_2' = g + g_2$. Likewise, we can compute $|\psi_{g_2' * x_\lambda}\rangle$ as $\mathsf{Map}(g, |\psi_{g_2 * x_\lambda}\rangle)$. This gives a random instance on which to apply $\mathcal{E}$, giving g_2' with probability $1 - \mathsf{negl}(\lambda)$, regardless of g_2. Then we can compute $g_2 = g_2' - g$. We thus compute g_2 with overwhelming probability, even in the worst case. We will therefore assume without loss of generality that this is the case for $\mathcal{E}$.

For simplicity, we will actually assume that the probability is 1; it is straightforward to handle the case the probability is negligibly close to 1. By the Gentle Measurement Lemma [Win99], $\mathcal{E}$ can compute g_2 without altering the state $|\psi_{g_2 * x}\rangle$. Thus, by combining $\mathcal{B}$ and $\mathcal{E}$, we can compute both $|\psi_{g_2 * x}\rangle$ and g_2 with probability 1. We can then compute $\mathsf{Map}(-g_2, |\psi_{g_2 * x_\lambda}\rangle) = |\psi_{x_\lambda}\rangle$.

We now describe a new algorithm $\mathcal{C}$ which breaks DLog/1-minCDH (Assumption 5.6). $\mathcal{C}$ works as follows:

- It constructs $|\psi_{x_\lambda}\rangle$ as above.
- It makes its query to the DLog/1-minCDH challenger, setting $\mathcal{M}_1$ as the query register. This query simulates the operation $\mathsf{Map}(g, \cdot)$, where g is the group element chosen by the challenger. Thus, at the end of the query, $\mathcal{C}$ has $|\psi_{g*x_\lambda}\rangle$.
- Now upon receiving $g * x_\lambda$ from the challenger, run $\mathcal{E}(g * x_\lambda, |\psi_{g*x_\lambda}\rangle)$. By the guarantees of $\mathcal{E}$, the output will be g.

Thus we see that $\mathcal{C}$ breaks the DLog/1-minCDH assumption. This completes the security proof. □

5.3 Algebraic Group Actions.

Next we turn to the Algebraic Group Action Model (AGAM), considered by a couple recent works [DHK+23, OZ23]. This is an analog of the Algebraic Group Model (AGM) [FKL18], adapted to group actions and quantum attackers. This model considers algebraic adversaries, which are algorithms where, any time they produce a set element output, must also "explain" the output in terms of the set elements the adversary saw as input. That is, if the algebraic adversary has so far been given set elements $y_1, \ldots, y_\ell$, when it outputs a new element y, it must also output a group element $g \in \mathbb{G}_\lambda$ and index i such that $y = g * y_i$.

In the classical world, a common refrain is that the AGM is "between" the generic group model and standard model. As formalized by Zhandry [Zha22], this is true in a particular sense: any "nice" security game that is secure in the standard model is also secure in the AGM, and in turn any nice security game that is secure in the AGM is also secure in the appropriate generic group model. The statements also hold true for group actions, provided we still restrict to the classical world. Here, "nice" comes with some important restrictions. The game must be "single stage", meaning there is only a single adversary interacting with the challenger. Moreover, the game must be a "type safe" game, which for group actions informally means the algorithms can pass set elements around and perform group action computations on them as a black box, but cannot maniluate the individual bits of the set element representations.

We might expect, therefore, that the AGAM is also "between" the GGAM and the standard model quantumly. However, this appears not to be the case, or at least it does not follow from any obvious adaptation of existing work. There are at least three problems.

The first is closely related to the issue with knowledge assumptions explored above. After all, the motivation for the AGAM, following the motivation from the AGM, is that we would expect the only way to output set elements is to actually derive them from existing set elements via the group action, in which case we would seem to know how to explain the new elements in terms of existing elements. In the classical setting, you can indeed show that this is true generically. However, our attack on the Q-KGEA assumption (Theorem 5.3) shows that this is not true quantumly. Namely, it is possible to output a superposition of set element where one does not "know" how to derive those elements from input elements.

For the second issue, consider the security game for our quantum lightning scheme. Recall that the adversary must output some h along with two copies of $|\mathbb{G}^h * x_\lambda\rangle = \frac{1}{\sqrt{|\mathbb{G}_\lambda|}} \sum_g \chi(h, g)|g * x_\lambda\rangle$. An algebraic adversary would have to "explain" this state, meaning it must output two copies of

$$\frac{1}{\sqrt{|\mathbb{G}_\lambda|}} \sum_g \chi(h, g)|g * x_\lambda, g\rangle$$

But here, note that if the challenger tries to verify the banknote state, the verification will actually *fail*, since the state is entangled with g. Worse, observe that the state produced by the algebraic adversary is actually trivial to construct for any given h, by first constructing $\frac{1}{\sqrt{|\mathbb{G}_\lambda|}}\sum_g \chi(h,g)|g\rangle$ and then applying the group action operation. Thus, we see that the algebraic adversary can actually trivially produce two copies of the requisite state. This is in contrast to the actual banknote state $|\mathbb{G}^h * x_\lambda\rangle$, where it appears only possible to sample actual banknotes for a random h, but not produce a banknote for a given h; indeed the security of our scheme inherently relies on this difficult. That is, the state required of the algebraic adversary is trivial, whereas the state required by a standard-model adversary is presumably hard to construct. This is in contrast to the classical world, where the algebraic adversary's task is always at least as hard as the real-world adversary.

The third issue is the claim that any game which is secure in the classical AGM/AGAM is also secure in the classical GGM/GGAM. This claim, or at least the classical proof of it, does not hold quantumly. This is because the proof relies on the ability to view the adversary's queries to the group/group action oracle and extract information from them. Specifically, in the classical AGAM, the only way the adversary can obtain new set elements is to act on existing elements by querying the group action. By writing down the input set and group element as well as the output group element, we can remember how we derived all set elements. Importantly, for any set element we produce, we can trace that set element back to an input set element, and see that the output element was obtained via a sequence of actions by group elements on the original input element. By multiplying these group actions together, we can explain the output element in terms of the input set element.

This strategy, however, does not work quantumly. Consider for example the hardness assumption DLog/minCDH (Assumption 5.6). Here, the adversary can query on a superposition $\sum_y \alpha_y|y\rangle$ of set elements, and get the resulting superposition obtained by action of a secret group element $(-g)$: $\sum_y \alpha_y|(-g)*y\rangle$

In the AGAM, we would ask the adversary queries on $\sum_y \alpha_y|y,\mathtt{Explain}_y\rangle$, where $\mathsf{Explain}_y$ is an explanation of y in terms of the elements the adversary has seen so far. In the case of DLog/minCDH, the only element seen by the time the adversary must make its query is x_λ, and so $\mathsf{Explain}_y$ is the unique h such that $y = h * x_\lambda$. Thus, the adversary's query takes the form $\sum_h \alpha_{h*x_\lambda}|h*x_\lambda, h\rangle$. In response, it receives

$$|\phi_{\mathsf{AGAM}}\rangle = \sum_h \alpha_{h*x_\lambda}|(h-g)*x_\lambda, h\rangle$$

On the other hand, a generic adversary would have just

$$|\phi_{\mathsf{GGAM}}\rangle = \sum_h \alpha_{h*x_\lambda}|(h-g)*x_\lambda\rangle$$

While in the classical setting, having the extra information h about y does not cause problems (it can just be erased or ignored), this extra information is problematic quantumly. For example, it might be that having $|\phi_{\mathsf{GGAM}}\rangle$ allows for solving some task, whereas having $\sum_h \alpha_{h*x_\lambda}|(h-g)*x_\lambda, h\rangle$ does not. In such a case, we find that the task is hard in the AGAM, despite being easy in the GGAM and even in the standard model. In particular, if we want the AGAM to be "between" the GGAM and standard models, we would need to rule this situation out, meaning we would need a way to map the state $|\phi_{\mathsf{AGAM}}\rangle$ containing the explanation back to the state $|\phi_{\mathsf{GGM}}\rangle$ without the explanation. This mapping, in general, will be intractible, as it requires un-computing h from $|(h-g)*x_\lambda$.

Based on these issues, we see that the AGAM is probably *not* a reasonable model for quantum attacks, at least when the game is inherently quantum, as with the security of our quantum lightning scheme or with assumptions that allow quantum queries. On the other hand, the model might be reasonable for "classically stated" security games, such as ordinary discrete log or CDH. However, these problems do not arise at all for generic group actions. Therefore, based on this discussion, we posit that generic group actions should be the preferred method for analyzing cryptosystems and security games.

6 A Construction for REGAs

In this section, we give a construction for the case where the group action can only be computed efficiently for a small "base" set of group elements. Such group actions are known as "restricted effective group actions" (REGAs).

6.1 Some additional background

Before giving the construction, we here provide some additional background that will be necessary for understanding the construction.

Groups. Let $\mathbb{G}$ be a group (written additively), and N an integer such that $N \times g = 0$ for all $g \in \mathbb{G}$. $N = |\mathbb{G}|$ will do. Then $\mathbb{G}$ is a subgroup of $\mathbb{Z}_N^n$ for some positive integer n. Let W be the set of vectors in $\mathbb{Z}_N^n$ such that $\mathbf{w} \cdot g = 0 \bmod N$ for all $g \in \mathbb{G}$. W is then a group, and we can therefore consider the group $(\mathbb{Z}_N^n)/W$ defined using the equivalence relation $\sim$, where $\mathbf{u}_1 \sim \mathbf{u}_2$ if $\mathbf{u}_1 - \mathbf{u}_2 \in W$. $(\mathbb{Z}_N^n)/W$ is isomorphic to $\mathbb{G}$; let $\phi : \mathbb{G} \to (\mathbb{Z}_N^n)/W$ be an isomorphism. Note that for $g \in \mathbb{G} \subseteq \mathbb{Z}_N^n$ and $h \in \mathbb{G}$, $g \cdot \phi(h) \bmod N$ is well-defined by taking any representative $h' \in \phi(h)$ and computing $g \cdot h' \bmod N$.

Under this notation, we can re-define $\chi(g, h)$ as $e^{i2\pi g\cdot\phi(h)/N}$, which is equivalent to the definition in Section 2.

We associate $\mathbb{Z}_N$ with the interval $[-\lfloor (N-1)/2 \rfloor, \lceil (N-1)/2 \rceil]$ in the obvious way, and likewise associate $\mathbb{Z}_N^n$ with the hypercube $[-\lfloor (N-1)/2 \rfloor, \lceil (N-1)/2 \rceil]^n$. This gives rise to a notion of norm on $\mathbb{Z}_N^n$ by taking the norm in $\mathbb{Z}^n$.

Lemma 6.1. *Let $\mathbb{G}$ be a subgroup of $\mathbb{Z}_N$. Then the number of elements $g \in \mathbb{G}$ such that $|g| \geq N/4$ is exactly $|\mathbb{G}| + 1 - 2\lceil |\mathbb{G}|/4 \rceil$. In particular, if $\mathbb{G} \neq \{0\}$, then there is at least one element $g \in \mathbb{G}$ has $|g| \geq N/4$.*

Proof. First, it suffices to consider $|\mathbb{G}| = N$, in other words $\mathbb{G} = \mathbb{Z}_N$: we can then lift to $N = t|\mathbb{G}|$, where $\mathbb{G}$ is embedded into $\mathbb{Z}_N$ by multiplying each element in $\mathbb{G}$ by t (where multiplication is over the integers). Since N is also multiplied by t, this preserves the number of elements with $|g| \geq N/4$.

When $\mathbb{G} = \mathbb{Z}_N$, we are then simply asking for the number of elements in $[-\lfloor (|\mathbb{G}|-1)/2 \rfloor, \lceil (|\mathbb{G}|-1)/2 \rceil]$ with absolute value at least $|\mathbb{G}|/4$. In other words, it is the combined size of the intervals $[\lceil |\mathbb{G}|/4 \rceil, \lceil (|\mathbb{G}|-1)/2 \rceil]$ and $[-\lfloor (|\mathbb{G}|-1)/2 \rfloor, -\lceil |\mathbb{G}|/4 \rceil]$, giving a total of $(\lceil (|\mathbb{G}|-1)/2 \rceil - \lceil |\mathbb{G}|/4 \rceil + 1) + (\lfloor (|\mathbb{G}|-1)/2 \lfloor - \lceil |\mathbb{G}|/4 \rceil + 1) = |\mathbb{G}| + 1 - 2\lceil |\mathbb{G}|/4 \rceil$. □

Lemma 6.2. *Let $\mathbf{A} \in \mathbb{Z}_N^{n\times m}$ be a matrix. Let $\mathbb{G}$ be the subgroup of $\mathbb{Z}_N^n$ generated by the columns of $\mathbf{A}$. Let B, C be positive integers such that $8BCm < N$. Suppose there is a distribution $\mathcal{D}$*

on $[-B, B]^m$ *such that* $\mathbf{A} \cdot \mathbf{x}$ *for* $x \leftarrow \mathcal{D}$ *is negligibly close to uniform in* $\mathbb{G}$. *Then the function* $f : \mathbb{G} \times [-C, C] \to \mathbb{Z}_N^m$ *given by* $f(g, \mathbf{e}) = \mathbf{A}^T \cdot \phi(g) + \mathbf{e}$ *is injective.*

Proof. Note that $\mathbf{A}^T \cdot \phi(g)$ is well defined since it is independent of the representative of $\phi(g)$. Consider a potential collision in f: $\mathbf{A}^T \cdot \phi(g_1) + \mathbf{e}_1 = \mathbf{A}^T \cdot \phi(g_2) + \mathbf{e}_2$. By subtracting, this gives a non-zero pair $(g = g_1 - g_2, \mathbf{e} = \mathbf{e}_1 - \mathbf{e}_2)$ where $\mathbf{e} \in [-2C, 2C]$ such that $\mathbf{A}^T \cdot \phi(g) + \mathbf{e} = 0$ or equivalently $\mathbf{A}^T \cdot \phi(g) = -\mathbf{e}$. Now consider sampling $\mathbf{x} \leftarrow \mathcal{D}$, meaning $\mathbf{u} = \mathbf{A} \cdot \mathbf{x}$ is negligibly close to uniform in $\mathbb{G}$. Then $\mathbf{u}^T \cdot \phi(g) = \mathbf{x}^T \cdot \mathbf{A}^T \cdot \phi(g) = -\mathbf{x}^T \cdot \mathbf{e}$. On one hand, $\mathbf{u}^T \cdot \phi(g)$ is statistically close to uniform in a subgroup $\mathbb{G}'$ of $\mathbb{Z}_N$, and $\mathbb{G}'$ is different from $\{0\}$ since $g \neq 0$. By Lemma 6.1, the probability $|\mathbf{u}^T \cdot \phi(g)| \geq N/4$ is $|\mathbb{G}'| + 1 - 2\lceil |\mathbb{G}'|/4 \rceil > 0$ since $|\mathbb{G}'| \geq 2$. On the other hand, $|-\mathbf{x}^T \cdot \mathbf{e}| < 2mBC \leq N/4$ always. This means the distributions of $\mathbf{u}^T \cdot \phi(g)$ and $-\mathbf{x}^T \cdot \mathbf{e}$ must be non-negligibly far, a contradiction. □

Discrete Gaussians. The *discrete Gaussian distribution* is the distribution over $\mathbb{Z}$ defined as:

$$\Pr[x] = \mathcal{D}_\sigma(x) := C_\sigma e^{2\pi x^2/\sigma^2},$$

where C_σ is the normalization constant $C_\sigma = \sum_{x \in \mathbb{Z}} e^{2\pi x^2/\sigma^2}$, so that $\mathcal{D}_\sigma$ defined a probability distribution. We will also define a truncated variant, denoted

$$\mathcal{D}_{\sigma,B}(x) := \begin{cases} C_{\sigma,B} e^{2\pi x^2/\sigma^2} & \text{if } |x| \leq B \\ 0 & \text{otherwise} \end{cases},$$

where again $C_{\sigma,B}$ is an appropriately defined normalization constant. For large B, we can treat the truncated and un-truncated Gaussians as essentially the same distribution:

Fact 6.3. *For* $\sigma \geq \omega(\sqrt{\log \lambda})$ *and* $B \geq \sigma \times \omega(\sqrt{\log \lambda})$, *the distributions* $\mathcal{D}_\sigma$ *and* $\mathcal{D}_{\sigma,B}$ *are negligibly close*

For a vector $\mathbf{r} \in \mathbb{Z}^m$, we write $\mathcal{D}_{\sigma,B}(\mathbf{r}) = \prod_{i=1}^m \mathcal{D}_{\sigma,B}(r_i)$.
The *discrete Gaussian superposition* is the quantum state

$$|\mathcal{D}_\sigma\rangle := \sum_{x \in \mathbb{Z}} \sqrt{\mathcal{D}_\sigma(x)} |x\rangle$$

As we will generally need to restrict to finite-precision, we also consider the truncated variant

$$|\mathcal{D}_{\sigma,B}\rangle := \sum_{x \in [-B,B]} \sqrt{\mathcal{D}_{\sigma,B}(x)} |x\rangle$$

Again, for large enough B, we can treat the truncated and un-truncated Gaussian superpositions as essentially the same state:

Fact 6.4. *For* $\sigma \geq \omega(\sqrt{\log \lambda})$ *and* $B \geq \sigma \times \omega(\sqrt{\log \lambda})$, *the* $\| |\mathcal{D}_\sigma\rangle - |\mathcal{D}_{\sigma,B}\rangle \|$ *is negligible.*

By adapting classicsal lattice sampling algorithms, the states $|\mathcal{D}_{\sigma,B}\rangle$ can be efficiently constructed.

Fourier transform pairs. Fix an integer N. We will associate the set $\mathbb{Z}_N$ with the integers $[-\lfloor (N-1)/2 \rfloor, \lceil (N-1)/2 \rceil]$. Denote by QFT_N the Quantum Fourier Transform $\mathsf{QFT}_{\mathbb{Z}_N}$. We now recall some basic facts about quantum Fourier transforms.

$$\mathsf{QFT}_N^m \sum_{\mathbf{r}\in\mathbb{Z}_N^m:\mathbf{A}\cdot\mathbf{r}=\mathbf{s}} |\mathbf{r}\rangle = N^{m/2-n} \sum_{\mathbf{t}\in\mathbb{Z}_N^n} e^{i2\pi\mathbf{t}\cdot\mathbf{s}/N} |\mathbf{A}^T\cdot\mathbf{t}\rangle \text{ for } \mathbf{A}\in\mathbb{Z}_N^{n\times m}$$

$$\mathsf{QFT}_N^m \sum_{\mathbf{r}} \alpha_{\mathbf{r}}\beta_{\mathbf{r}}|\mathbf{r}\rangle = \frac{1}{N^{m/2}} \sum_{\mathbf{t},\mathbf{u}} \hat{\alpha}_{\mathbf{t}}\hat{\beta}_{\mathbf{u}}|\mathbf{u}+\mathbf{t}\rangle \text{ for } {\scriptstyle \sum_{\mathbf{t}}\hat{\alpha}_{\mathbf{t}}|\mathbf{t}\rangle=\mathsf{QFT}_N^m\sum_{\mathbf{r}}\alpha_{\mathbf{r}}|\mathbf{r}\rangle \atop \sum_{\mathbf{u}}\hat{\beta}_{\mathbf{u}}|\mathbf{u}\rangle=\mathsf{QFT}_N^m\sum_{\mathbf{r}}\beta_{\mathbf{r}}|\mathbf{r}\rangle}$$

$$\mathsf{QFT}_N|\mathcal{D}_{\sigma,\lfloor (N-1)/2\rfloor}\rangle \approx |\mathcal{D}_{N/\sigma,\lfloor (N-1)/2\rfloor}\rangle \text{ for } {\scriptstyle N\geq\sigma\times\omega(\sqrt{\log\lambda}) \atop \sigma\geq\omega(\sqrt{\log\lambda})}$$

Above, $\approx$ means the two states are negligibly close.

6.2 The Construction

Let $\mathbb{G}_\lambda, \mathcal{X}_\lambda, *$ be a REGA, and $\mathcal{T} = (g_1, \dots, g_m)$ a set such that $*$ can be efficiently computed for g_i and g_i^{-1}. We can associate $\mathbb{G}_\lambda$ with a subgroup of $\mathbb{Z}_N^n$ for some integers N, n. We can likewise associate the list $\mathcal{T}$ with the matrix $\mathbf{A} = (g_1, \cdots, g_m) \in \mathbb{Z}_N^{n\times m}$.

Since we can only compute the action of certain group elements, this will significantly complicate our construction. There are several issues that need to be resolved.

- For both minting and verification of our original scheme, we needed the ability to apply the group action on *random* group elements, which is not possible in REGAs. Our solution, following typical applications of REGAs in the literature, is to choose our random group element as a "small" known combination of the base group elements $g = \sum_{i=1}^m r_i g_i$ where the r_i are small integers. Under mild assumptions, g will be uniform, and using the representation as a small combination of the g_i we can efficiently compute the action by g.
- Unfortunately, the r_i are now side-information entangled with g which is hard to un-compute. If the r_i are left around, it they will be entangled with the banknote which will break the correctness of the scheme. Our solution is to actually treat the r_i as teh group element, and perform the QFT on the r_i instead of on g. This results in a number of complications, one being that the serial number is actually now hidden, and a different quantity must be used as the serial number. This quantity also turns out to be noisy. Nevertheless, by careful analysis, we are able to show our scheme is correct, and explain how to adapt the security proof from Section 4.5 to our REGA scheme.

We now give the details. We will make the following assumption about the structure of $\mathcal{T}$, which is typical in the isogeny literature.

Assumption 6.5. There is a polynomial B and a distribution $\mathcal{D}^*$ on $[-B, B]^m$ such that for $\mathbf{x} \leftarrow \mathcal{D}$, $\sum_{i=1}^m x_i g_i = \mathbf{A}\cdot\mathbf{x}$ is statistically close to a uniform element in $\mathbb{G}$

Numerous examples of such $\mathcal{D}^*$ have been proposed, such as discrete Gaussians [DG19], or uniform vectors in small balls relative to different norms [CLM+18, NOTT20].

Let $C = N/8Bm$, which then satisfies the conditions of Lemma 6.2. Thus, for $\mathbf{e}$ with entries in $[-C, C]^m$, the map $(g, \mathbf{e}) \mapsto \mathbf{A}^T\cdot\phi(g) + \mathbf{e}$ is injective.

Let $\sigma \geq 16Bm/\epsilon \times \omega(\sqrt{\log \lambda})$ and $B' \geq \sigma \times \omega(\sqrt{\log \lambda})$ be polynomials. We will assume $N \geq 2B'$, which is always possible since we can take N to be arbitrarily large. We will also for simplicity assume N is even. This assumption is not necessary but will simplify some of the analysis, and is moreover without loss of generality since we can always make N larger by multiplying it by arbitrary factors.

Construction 6.6. $\mathsf{Gen}(1^\lambda)$: Initialize quantum registers $\mathcal{S}$ (for serial number) and $\mathcal{M}$ (for money) to states $|\mathcal{D}_{\sigma,B'}\rangle_{\mathcal{S}}^{\otimes m}$ and $|0\rangle_{\mathcal{M}}$, respectively. Then do the following:

- Apply in superposition the map $|\mathbf{r}\rangle_{\mathcal{S}}|y\rangle_{\mathcal{M}} \mapsto |\mathbf{r}\rangle_{\mathcal{S}}|y \oplus [(\sum_{i=1}^m r_i g_i) * x_\lambda])\rangle_{\mathcal{M}}$. The joint state of the system $\mathcal{S} \otimes \mathcal{M}$ is then

$$\sum_{\mathbf{r}\in\mathbb{Z}_N^m} \sqrt{\mathcal{D}_{\sigma',B}(\mathbf{r})}|\mathbf{r}\rangle_{\mathcal{S}}|(\sum_{i=1}^m r_i g_i) * x_\lambda\rangle_{\mathcal{M}} = \sum_{g\in\mathbb{G}_\lambda}\left(\sum_{\mathbf{r}\in\mathbb{Z}_N^m:\mathbf{A}\cdot\mathbf{r}=g} \sqrt{\mathcal{D}_{\sigma,B'}(\mathbf{r})}|\mathbf{r}\rangle_{\mathcal{S}}\right)|g * x_\lambda\rangle_{\mathcal{M}}$$

- Apply $\mathsf{QFT}_{\mathbb{Z}_N^m}$ to $\mathcal{S}$. Using the QFT rules given above, this yields the state negligibly close to:

$$\begin{aligned}&\frac{1}{N^n}\sum_{g\in\mathbb{G}_\lambda}\left(\sum_{\mathbf{s},\mathbf{e}\in\mathbb{Z}_N^n}\sqrt{\mathcal{D}_{N/\sigma,N/2-1}(\mathbf{e})}e^{i2\pi(g\cdot\mathbf{s})}|\mathbf{A}^T\cdot\mathbf{s}+\mathbf{e}\rangle_{\mathcal{S}}\right)|g * x_\lambda\rangle_{\mathcal{M}}\\ &=\frac{1}{|\mathbb{G}_\lambda|}\sum_{g\in\mathbb{G}_\lambda}\left(\sum_{h\in\mathbb{G}_\lambda,\mathbf{e}\in\mathbb{Z}_N^n}\sqrt{\mathcal{D}_{N/\sigma,N/2-1}(\mathbf{e})}e^{i2\pi(g\cdot\phi(h))}|\mathbf{A}^T\cdot\phi(h)+\mathbf{e}\rangle_{\mathcal{S}}\right)|g * x_\lambda\rangle_{\mathcal{M}}\\ &=\frac{1}{\sqrt{|\mathbb{G}_\lambda|}}\sum_{g\in\mathbb{G}_\lambda}\left(\frac{1}{\sqrt{|\mathbb{G}_\lambda|}}\sum_{h\in\mathbb{G}_\lambda,\mathbf{e}\in\mathbb{Z}_N^n}\sqrt{\mathcal{D}_{N/\sigma,N/2-1}(\mathbf{e})}\chi(g,h)|\mathbf{A}^T\cdot\phi(h)+\mathbf{e}\rangle_{\mathcal{S}}\right)|g * x_\lambda\rangle_{\mathcal{M}}\end{aligned}$$

- Measure $\mathcal{S}$, giving the serial number $\mathbf{t} := \mathbf{A}^T \cdot \phi(h) + \mathbf{e}$. $\mathbf{e}$ is distributed negligibly close to $\mathcal{D}_{N/\sigma}$, meaning with overwhelming probability each entry is in $[-N/16Bm, N/16Bm] = [-C/2, C/2] \subseteq [-C, C]$. This means, to within negligible error, $\mathbf{t}$ uniquely determines $\phi(h)$ and hence h. Therefore, the $\mathcal{M}$ register then collapses to a state negligibly close to

$$\frac{1}{\sqrt{|\mathbb{G}_\lambda|}}\sum_{g\in\mathbb{G}_\lambda}\chi(g,h)|g * x_\lambda\rangle_{\mathcal{M}} =: |\mathbb{G}_\lambda^h * x_\lambda\rangle$$

  Note that h is unknown. Output $(\mathbf{t}, |\mathbb{G}_\lambda^h * x_\lambda\rangle)$

$\mathsf{Ver}(\mathbf{t}, \$)$: 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 repeat the following λ times:

- Initialize a new register $\mathcal{H}$ to $(|0\rangle_{\mathcal{H}} + |1\rangle_{\mathcal{H}})/\sqrt{2}$.
- Choose a random element $\mathbf{x} \leftarrow \mathcal{D}^*$.
- Apply to $\mathcal{H} \otimes \mathcal{M}$ in superposition the map

$$\mathsf{Apply}|b\rangle_{\mathcal{H}}|y\rangle_{\mathcal{M}} \mapsto \begin{cases}|0\rangle_{\mathcal{H}}|y\rangle_{\mathcal{M}} & \text{if } b = 0\\ |1\rangle_{\mathcal{H}}|(-\sum_i x_i g_i) * y\rangle_{\mathcal{M}} & \text{if } b = 1\end{cases}$$

  Since the entries of $\mathbf{x}$ are bounded by B which is polynomial, this step is efficient.

- Measure $\mathcal{H}$ in the basis $B_{\mathbf{t},\mathbf{x}} := \{(|0\rangle_{\mathcal{H}} + e^{i2\pi\mathbf{x}^T\cdot\mathbf{t}/N}|1\rangle_{\mathcal{H}})/\sqrt{2}, (|0\rangle_{\mathcal{H}} - e^{i2\pi\mathbf{x}^T\cdot\mathbf{t}/N}|1\rangle_{\mathcal{H}})/\sqrt{2}\}$, giving a bit $b_u \in \{0,1\}$. Discard the $\mathcal{H}$ register.
- Accept if at least a fraction 7/8 of the $b_u = 0$ and the support of \$ is contained in $\mathcal{X}_\lambda$; otherwise reject.

6.3 Accepting States of the Verifier

We now analyze the correctness of the construction.

Theorem 6.7. *Let $|\psi\rangle$ be a state over $\mathcal{M}$. Then* $\Pr[\mathsf{Ver}(h, |\psi\rangle) = 1] = \|\langle\psi|\mathbb{G}_\lambda^h * x_\lambda\rangle\|^2(1 - 2^{-\Omega(\sqrt{\lambda})} + 2^{-\Omega(\sqrt{\lambda})}$.

Proof. For simplicity, we analyze the case of $|\psi\rangle = |\mathbb{G}_\lambda^{h'} * x_\lambda\rangle$, which form a basis for superpositions over $\mathcal{X}_\lambda$. In this case, Theorem 6.7 states that $|\mathbb{G}_\lambda^h * x_\lambda\rangle$ is accepted with probability $1 - 2^{\Omega(\sqrt{\lambda})}$, while $|\mathbb{G}_\lambda^{h'} * x_\lambda\rangle$ for $h' \neq h$ is accepted with probability $2^{\Omega(\sqrt{\lambda})}$. Linearity of the verifier allows us to extend to all possible states.

If we let $u = \mathbf{A}\cdot\mathbf{x} = \sum_i x_i g_i$, then by the same analysis as in Construction 3.1, we have that applying Apply to the state $|\mathbb{G}_\lambda^{h'} * x_\lambda\rangle$ results in the state

$$\begin{aligned}&\frac{1}{\sqrt{2}}\left(|0\rangle_{\mathcal{H}} + \chi(u,h')|1\rangle_{\mathcal{H}}\right)|\mathbb{G}_\lambda^{h'} * x_\lambda\rangle\\ &= \frac{1}{\sqrt{2}}\left(|0\rangle_{\mathcal{H}} + e^{i2\pi u\cdot\phi(h')/N}|1\rangle_{\mathcal{H}}\right)|\mathbb{G}_\lambda^{h'} * x_\lambda\rangle\\ &= \frac{1}{\sqrt{2}}\left(|0\rangle_{\mathcal{H}} + e^{i2\pi\mathbf{x}^T\cdot\mathbf{A}^T\cdot\phi(h')/N}|1\rangle_{\mathcal{H}}\right)|\mathbb{G}_\lambda^{h'} * x_\lambda\rangle\end{aligned}$$

Conditioned on sampling u, $\Pr[b_u = 0]$ is the inner product squared of $\left(|0\rangle_{\mathcal{H}} + e^{i2\pi\mathbf{x}^T\cdot\mathbf{A}^T\cdot\phi(h')/N}|1\rangle_{\mathcal{H}}\right)/\sqrt{2}$ with the basis state $\left(|0\rangle_{\mathcal{H}} + e^{i2\pi\mathbf{x}\cdot\mathbf{t}/N}|1\rangle_{\mathcal{H}}\right)/\sqrt{2}$. This is:

$$\begin{aligned}\Pr[b_u = 0] &= \frac{1}{4}\left\|1 + e^{i2\pi(\mathbf{x}^T\cdot\mathbf{A}^T\cdot\phi(h') - \mathbf{x}^T\cdot\mathbf{t})/N}\right\|^2\\ &= \frac{1}{2}\left(1 + \cos\left[2\pi(\mathbf{x}^T\cdot\mathbf{A}^T\cdot\phi(h') - \mathbf{x}^T\cdot(\mathbf{A}^T\cdot\phi(h) + \mathbf{e}))/N\right]\right)\\ &= \frac{1}{2}\left(1 + \cos\left[2\pi(\mathbf{x}^T\cdot\mathbf{A}^T\cdot\phi(h'-h) + \mathbf{x}^T\cdot\mathbf{e})/N\right]\right)\end{aligned}$$

In the case $h = h'$, $\Pr[b_u = 0] = \frac{1}{2}\left(1 + \cos\left[2\pi\mathbf{x}^T\cdot\mathbf{e}/N\right]\right)$. We have that $|2\pi\mathbf{x}^T\cdot\mathbf{e}/N| \leq \pi/8$. Using the fact that $\cos(x) \geq 1 - x^2/2$, we therefore have that $\Pr[b_u = 0] \geq 1 - \pi^2/256 = 0.9614\ldots = 7/8 + \Omega(1)$. Then via standard concentration inequalities, after λ trials, except with probability $2^{-\Omega(\sqrt{\lambda})}$, at least 7/8 of the b_u will be 0. Therefore, Ver accepts with probability $1 - 2^{-\Omega(\sqrt{\lambda})}$.

On the other hand, if $g \neq g'$, then $\mathbf{x}^T\mathbf{A}^T$ is statistically close to uniform in $\mathbb{G}_\lambda$, and so $\mathbf{x}^T\cdot\mathbf{A}^T\cdot\phi(h'-h)$ is statistically close to uniform in a non-trivial subgroup $\mathbb{G}'$ of $\mathbb{Z}_N$. By Lemma 6.1 and our assumption that N is even, at least half of the elements of $\mathbb{Z}_N$ are at least $N/4$ in absolute value. In particular, this means $\Pr[|\mathbf{x}^T\cdot\mathbf{A}^T\cdot\phi(h'-h)| \geq N/4] \geq 1/2 - \mathsf{negl}$. On the other hand, $|\mathbf{x}^T\cdot\mathbf{e}| \leq N/16$ always. This means $\|\mathbf{x}^T\cdot\mathbf{A}^T\cdot\phi(h'-h) + \mathbf{x}^T\cdot\mathbf{e}\| \geq N/4 - N/16$ with probability at least $1/2 - \mathsf{negl}$.

In this case, we can use that $\cos(\pi/2+x) \leq |x|$ to bound $\cos\left[2\pi(\mathbf{x}^T \cdot \mathbf{A}^T \cdot \phi(h'-h) + \mathbf{x}^T \cdot \mathbf{e})/N\right] \leq 2\pi/16 = \pi/8$, meaning $\Pr[b_u = 0] \leq 1/2 + \pi/16$. Averaging over all u, we therefore have that: $\Pr[b_u = 0] \leq \frac{3}{4} + \pi/32 + \mathsf{negl} = 0.8481\ldots = 7/8 - \Omega(1)$. Then via standard concentration inequalities, after λ trials, except with probability $2^{-\Omega(\sqrt{\lambda})}$, fewer than $7/8$ of the b_u will be 0. therefore, Ver accepts with probability $2^{-\Omega(\sqrt{\lambda})}$. □

6.4 Security

Here, we state the security of Construction 6.6.

Assumptions. We first need to define slight variants of our assumptions, in order to be consistent with the more limited structure of a REGA. For example, in the ordinary Discrete Log assumption (Assumption 2.4), the challenger computes $y = g * x$ for a random g, and adversary produces g. But the adversary cannot even tell if it succeeded since it cannot compute the action of g in general. Instead, the adversary is required not to compute g, but instead to compute any short $\mathbf{x}$ such that $g = \sum_i x_i g_i$. The adversary can then check that it has a solution by computing the action of g using its knowledge of $\mathbf{x}$. We analogously update each of our assumptions to work with the limited ability to compute the group action on REGAs.

As above, let $\mathbb{G}_\lambda, \mathcal{X}_\lambda, *$ be a REGA, and $\mathcal{T} = (g_1, \ldots, g_m)$ a set such that $*$ can be efficiently computed for g_i and g_i^{-1}. Let $\mathcal{D}^*, B$ be as in Assumption 6.5.

Assumption 6.8. The *REGA quantum modified knowledge of group element assumption* (REGA-Q-KGEA) holds on a group action $(\mathbb{G}, \mathcal{X}, *)$ if the following is true. For any quantum polynomial time (QPT) adversary $\mathcal{A}$ which performs no measurements except for its final output, there exists a polynomial C, a QPT extractor $\mathcal{E}$ with outputs in $[-C, C]^m$, and negligible ϵ such that

$$\Pr\left[y \in \mathcal{X} \wedge y \neq g * x_\lambda : \substack{(y,|\psi\rangle)\leftarrow\mathcal{A}(1^\lambda) \\ \mathbf{x}\leftarrow\mathcal{E}(y,|\psi\rangle)} g \leftarrow \sum_i x_i g_i\right] \leq \epsilon(\lambda) \enspace .$$

As with the non-REGA Q-KGEA assumption, we expect the REGA-Q-KGEA assumption is likely false. Certainly it is false on group actions with oblivious sampling. However, we note that it is unclear if our attack from Theorem 5.3 can be adapted to REGAs. Nevertheless, to mitigate any risks associated with the plain REGA-Q-KGEA assumption, we can likewise define a *modified* REGA KGEA assumption (REGA-Q-mKGEA), in the same spirit as Assumption 5.4.

We next define our REGA analog of Assumption 5.6.

Assumption 6.9. We say that the *REGA Discrete Log with a single minimal CDH query* assumption (REGA-DLog/1-minCDH) assumption holds if the following is true. For any QPT adversary $\mathcal{A}$ playing the following game, parameterized by λ, there is a negligible ϵ such that $\mathcal{A}$ wins with probability at most $\epsilon(\lambda)$:

- The challenger, on input λ, chooses a random $g \in \mathbb{G}_\lambda$. It sends λ to $\mathcal{A}$
- $\mathcal{A}$ submits a superposition query $\sum_{y \in \mathcal{X}, z \in \{0,1\}^*} \alpha_{y,z} |y, z\rangle$. Here, y is a set element that forms the query, and z is the internal state of the adversary when making the query. The challenger responds with $\sum_{y \in \mathcal{X}, z \in \{0,1\}^*} \alpha_{y,z} |(-g) * y, z\rangle$.
- The challenger sends $g * x$ to $\mathcal{A}$.

- $\mathcal{A}$ outputs a $\mathbf{x} \in \mathbb{Z}^m$, encoded in unary. It wins if $g = \sum_i x_i g_i$.

Note that the challenger in Assumption 6.9 is inefficient on a REGA. However, under Assumption 6.5, the challenger can be made efficient by first sampling $\mathbf{y} \leftarrow \mathcal{D}^*$ and then computing $g = \sum_i y_i g_i$.

Theorem 6.10. *Assuming REGA-DLog/1-minCDH (Assumption 6.9) and REGA-Q-KGEA (Assumption 6.8) (or more generally, REGA-Q-mKGEA) both hold on a group action $(\mathbb{G}, \mathcal{X}, *)$, then Construction 6.6 is a quantum lightning scheme. Alternatively, if D2X/min (Assumption 4.9) holds on a group action with $\mathcal{X} \subseteq \{0,1\}^m$, then Construction 6.6 is a quantum lightning scheme in the generic group action model $\mathsf{GGAM}_{\mathbb{G},m'}$ with label length m'.*

We only sketch the proof. Like in the proof of Theorems 4.12 and 5.9, we can assume the adversary wins the quantum lightning experiment with probability $1 - \mathsf{negl}(\lambda)$. In order for a supposed note \$ to be accepted relative to serial number $\mathbf{t}$ with overwhelming probability, $\mathbf{t}$ must have the form $\mathbf{t} = \mathbf{A}^T \cdot \phi(h) + \mathbf{e}$ for "short" $\mathbf{e}$, and \$ must be negligibly close to $|\mathbb{G}_\lambda^h * x_\lambda\rangle$. Therefore, a quantum lightning adversary outputs two copies of $|\mathbb{G}_\lambda^h * x_\lambda\rangle$ for some h. The security reduction of Theorem 4.12 did not rely on knowing h, just that the adversary outputted two copies of $|\mathbb{G}_\lambda^h * x_\lambda\rangle$. Hence, a near-identical proof holds for Construction 6.6. The only difference is that when the extractor $\mathcal{E}$ outputs a group element, it instead outputs a small linear combination of the g_i giving that group element, and then the DLog/1-minCDH adversary uses this small representation to compute the action by that group element.

7 Further Discussion

7.1 Quantum Group Actions

Here, we consider a generalization of group actions where set elements are replaced with quantum states.

An quantum (abelian) group action consists of a family of (abelian) groups $\mathbb{G} = (\mathbb{G}_\lambda)_\lambda$ (written additively), a family $\mathcal{X} = (\mathcal{X}_\lambda)_\lambda$ of sets $\mathcal{X}_\lambda$ of quantum states over a system $\mathcal{M}_\lambda$, and an operation $*$. We will require that the states in $\mathcal{X}_\lambda$ are orthogonal. $*$ is a quantum algorithm that takes as input a group element $g \in \mathbb{G}_\lambda$ and a quantum state $|\psi\rangle$ over $\mathcal{M}_\lambda$, and outputs another state over $\mathcal{M}_\lambda$. $*$ satisfies the following properties:

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

We can also relax the above properties to only hold to within negligible error, and/or relax the orthogonality requirement to being near-orthogonal. We will additionally require the following properties:

- **Efficiently computable:** There is a pseudodeterministic QPT procedure $\mathsf{Construct}$ which, on input 1^λ, outputs a description of $\mathbb{G}_\lambda$ and an specific element $|\psi_\lambda\rangle \in \mathcal{X}_\lambda$. The operation $*$ is also computable by a QPT algorithm.
- **Efficiently Recognizable:** There is a QPT procedure Recog which recognizes elements in $\mathcal{X}_\lambda$. That is, $\mathsf{Recog}(1^\lambda, \cdot)$ projects onto the span of the states in $\mathcal{X}_\lambda$.

- **Regular:** For every $|\phi\rangle \in \mathcal{X}_\lambda$, there is exactly one $g \in \mathbb{G}_\lambda$ such that $|\phi\rangle = g * |\psi_\lambda\rangle$.

Again, we can also relax the above properties to only hold to within negligible error.

Cryptographic group actions. At a minimum, a cryptographically useful quantum group action will satisfy the following discrete log assumption:

Assumption 7.1. The *discrete log assumption* (DLog) holds on a quantum group action $(\mathbb{G}, \mathcal{X}, *)$ if, for all QPT adversaries $\mathcal{A}$, there exists a negligible λ such that

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

Note that if we do not insist on orthogonality of the states in $\mathcal{X}_\lambda$, then it is trivial to construct a quantum group action in which DLog holds: simply have all $|\psi\rangle \in \mathcal{X}_\lambda$ be identical, or negligibly close. Then it will be information-theoretically impossible to determine g. Orthogonality essentially says that the group action is classical, except that the basis for the set elements is potentially different than the computational basis.

7.2 Quantum Group Actions From Lattices

Here, we describe a simple quantum group action from lattices.

The group $\mathbb{G}_{\mathsf{LWE,N,n,m},\sigma}$ will be set to $\mathbb{Z}_N^n$ for some integers N, n. We will fix a short wide matrix $\mathbf{A} \in \mathbb{Z}_N^{n\times m}$; we can think of $\mathbf{A}$ as being sampled randomly and included in a common reference string. Note that $\mathbb{G}$ is independent of σ, but we include it for notational consistency.

The set $\mathcal{X}_{\mathsf{LWE,N,n,m},\sigma}$ will be the set of states $|\psi_{\mathbf{s}}\rangle = \sum_{\mathbf{e}\in\mathbb{Z}_N^n} \sqrt{\mathcal{D}_{\sigma,N/2}(\mathbf{e})}|\mathbf{A}^T \cdot \mathbf{s} + \mathbf{e}\rangle$. In other words, we take the discrete Gaussian vector superposition of some width, and add the vector $\mathbf{A}^T \cdot \mathbf{s}$.

$\mathbb{G}_{\mathsf{LWE,N,n,m},\sigma}$ acts on $\mathcal{X}_{\mathsf{LWE,N,n,m},\sigma}$ in the following obvious way: $\mathbf{r} * |\psi_{\mathbf{s}}\rangle = |\psi_{\mathbf{r}+\mathbf{s}}\rangle$, which can be computed by simply adding $\mathbf{A}^T \cdot \mathbf{r}$ in superposition.

We have the following theorem:

Theorem 7.2. *Let σ, σ_0 be non-negative real numbers such that σ/σ_0 is super-logarithmic. Assuming the Learning with Errors problem is hard for noise distribution $\mathcal{D}_{\sigma_0}$, discrete logarithms are hard in the group action $(\mathbb{G}_{\mathsf{LWE,N,n,m},\sigma}, \mathcal{X}_{\mathsf{LWE,N,n,m},\sigma}, *)$.*

Proof. The learning with errors assumption states that it is hard to compute $\mathbf{s}$ given $\mathbf{A}^T \cdot \mathbf{s} + \mathbf{e}$ with $\mathbf{e}$ sampled from $\mathcal{D}_{\sigma_0}$. We need to show that it is hard to compute $\mathbf{s}$ given the analogous superposition over $\mathbf{A}^T \cdot \mathbf{s} + \mathbf{e}$, where here $\mathbf{e}$ comes from the Gaussian superposition $|\mathcal{D}_\sigma\rangle$. The idea is a simple application of noise flooding: given $\mathbf{u} = \mathbf{A}^T \cdot \mathbf{s} + \mathbf{e}$, compute the state $|\psi'_{\mathbf{s}}\rangle := \sum_{\mathbf{e}'\in\mathbb{Z}_N^n} \sqrt{\mathcal{D}_{\sigma,N/2}(\mathbf{e}')}|\mathbf{A}^T \cdot \mathbf{s} + \mathbf{e} + \mathbf{e}'\rangle$. Since σ/σ_0 is super-polynomial, $\mathbf{e} + \mathbf{e}'$ where $\mathbf{e}' \leftarrow \mathcal{D}_{\sigma,N/2}$ is negligibly close to a Gaussian centered at 0. Therefore, $|\psi'_{\mathbf{s}}\rangle$ is negligibly close to $|\psi_{\mathbf{s}}\rangle$. Plugging into a supposed DLog adversary then gives $\mathbf{s}$, breaking LWE. □

Unfortunately, this LWE-based group action is missing a crucial feature: it is not possible to recognize states in $\mathcal{X}$. In particular, the states in $\mathcal{X}$ are indistinguishable from states of the form $\sum_{\mathbf{e}\in\mathbb{Z}_N^n} \sqrt{\mathcal{D}_{\sigma,N/2}(\mathbf{e})}|\mathbf{v} + \mathbf{e}\rangle$, where $\mathbf{v}$ is an arbitrary vector in $\mathbb{Z}_N^m$. As we will see in the next sub-section, the inability to recognize $\mathcal{X}$ will prevent us from using this group action to instantiate our quantum money scheme.

7.3 Relation to Quantum Money Approaches based on Lattices

Here, we see that our quantum money scheme is conceptually related to a folklore approach to building quantum money from lattices. This approach has not been able to work; in our language, the reason is exactly due to the inability to recognize $\mathcal{X}_{\mathsf{LWE,N,n,m,\sigma}}$.

The approach is the following. Let $\mathbf{A} \in \mathbb{Z}_N^{n \times m}$ be a random short wide matrix over $\mathbb{Z}_n$. To mint a banknote, construct the discrete Gaussian superposition $|\mathcal{D}_\sigma\rangle^{\otimes m}$ in register $\mathcal{M}$. Then compute and measure $\mathbf{A} \cdot \mathbf{x}$ applied to $\mathcal{M}$. The result is a vector $\mathbf{h} \in \mathbb{Z}_N^n$, which will be the serial number, and $\mathcal{M}$ collapses to a superposition $|\$_{\mathbf{h}}\rangle \propto \sum_{\mathbf{x}:\mathbf{A}\cdot\mathbf{x}=\mathbf{h}} \sqrt{\mathcal{D}_\sigma(\mathbf{x})}|\mathbf{x}\rangle$ of short vectors $\mathbf{x}$ such that $\mathbf{A} \cdot \mathbf{x} = \mathbf{h}$. This is the banknote. A simple argument shows that it is impossible to construct two copies of $|\$_{\mathbf{h}}\rangle$ for the same $\mathbf{h}$: given such a pair, measure each to get $\mathbf{x}, \mathbf{x}'$ such that $\mathbf{A} \cdot \mathbf{x} = \mathbf{A} \cdot \mathbf{x}' = \mathbf{h}$. Then subtract to get a short vector $\mathbf{x} - \mathbf{x}'$ such that $\mathbf{A} \cdot (\mathbf{x} - \mathbf{x}') = 0^n$. We can conclude $\mathbf{x} - \mathbf{x}'$ is non-zero with overwhelming probability, since the measurement of $|\$_{\mathbf{h}}\rangle$ has high entropy. Such a non-zero short kernel vector would solve the Short Integer Solution (SIS) problem, which is widely believed to be hard and is the foundation of lattice-based cryptography.

Unfortunately, the above approach is broken. The problem is that there is no way to actually verify banknotes. One can verify that a banknote has support on short vectors with $\mathbf{A} \cdot \mathbf{x} = \mathbf{h}$, but it is impossible to verify that the banknote is in superposition. If one could solve the Learning with Errors (LWE) problem, it would be possible to verify banknotes as follows: first perform the QFT to the banknote state. If an honest banknote, the QFT will give a state negligibly close to

$$|\$'_{\mathbf{h}}\rangle := \frac{1}{N^{n/2}} \sum_{\mathbf{s},\mathbf{e} \in \mathbb{Z}_N^n} \sqrt{\mathcal{D}_{N/\sigma}(\mathbf{e})} e^{i2\pi \mathbf{h}\cdot\mathbf{s}/N} |\mathbf{A}^T \cdot \mathbf{s} + \mathbf{e}\rangle \ . \tag{7.1}$$

The second step is to simply apply the supposed LWE solver to this state in superposition, ensuring that the state has support on vectors of the form $\mathbf{A}^T \cdot \mathbf{s} + \mathbf{e}$ for small $\mathbf{e}$.

Unfortunately, LWE is likely hard. In fact, it is quantumly equivalent to SIS [Reg05], meaning if one could verify banknotes using an LWE solver, then SIS is easy. Not only does this mean we are reducing from an easy problem, but it would be possible to turn such a SIS algorithm into an attack.

Without the ability to verify that banknotes are in supeprosition, the attacker can simply measure a banknote to get $\mathbf{x}$, and then pass off $|\mathbf{x}\rangle$ as a fake banknote that will pass verification. Since $\mathbf{x}$ is trivially copied, this would break security. Interestingly, [LZ19] prove that, no matter what efficient verification procedure is used, even if the verification diverged from the LWE-based approach above, this attack works. [LMZ23] extend this to a variety of potential schemes based on similar ideas, including a recent proposed instantiation of this approach by [KLS22].

We now see how the above approach is essentially equivalent to our construction of quantum money from group actions, instantiated over our LWE-based quantum group action. The inability to recognize $\mathcal{X}$ is the reason this instantiation is insecure, despite natural hardness assumptions presumably holding on the group action.

We consider the quantum group action $(\mathbb{G}_{\mathsf{LWE,N,n,m,N/\sigma}}, \mathcal{X}_{\mathsf{LWE,N,n,m,N/\sigma}}, *)$, where σ is from the folklore construction above. When applied to $(\mathbb{G}_{\mathsf{LWE,N,n,m,N/\sigma}}, \mathcal{X}_{\mathsf{LWE,N,n,m,N/\sigma}}, *)$, a banknote in our scheme, up to negligibly error from truncating discrete Gaussians, is the state $|\$'_{\mathbf{h}}\rangle$ from Equation 7.1 above, where the serial number is $\mathbf{h}$. Thus, we see that our quantum money scheme is simply the folklore construction but moved to the Fourier domain. The attack on the folklore construction can therefore easily be mapped to an attack on our scheme: if the adversary is given $|\$'_{\mathbf{h}}\rangle$, it measures in the Fourier domain (which is the primal domain for the folklore construction) to get a short vector

$\mathbf{x}$ such that $\mathbf{A} \cdot \mathbf{s} = \mathbf{h}$. Then it switched back to the primal domain, giving the state

$$\frac{1}{N^{m/2}} \sum_{\mathbf{u}} e^{i2\pi \mathbf{e} \cdot \mathbf{x}} |\mathbf{x}\rangle$$

This is a state that lies outside the span of $\mathcal{X}$. However, no efficient verification procedure can distinguish it from an honest banknote state.

Two features that distinguish isogeny-based group actions from the LWE-based action above. The first is the ability to recognize elements in $\mathcal{X}$. Suppose it were possible to recognize elements of $\mathcal{X}$ in the LWE-based action, and we had the verifier check to see if the banknote belonged to the span of the elements in $\mathcal{X}$. In the language of quantum group actions, this check would prevent the attacker from sending $\frac{1}{N^{m/2}} \sum_{\mathbf{u}} e^{i2\pi \mathbf{e} \cdot \mathbf{x}} |\mathbf{x}\rangle$, which lies outside the span of $\mathcal{X}$. In the language of the folklore construction, this check would correctly distinguish between an honest banknote and the easily clonable state $|\mathbf{x}\rangle$ in the attack. If such a check were possible, the proof sketched above would work to base the security of the scheme on SIS. Unfortunately, such a check is computationally intractable under the decision LWE problem, which is equivalent to SIS and most likely hard.

The issue of recognizing set elements is also crucial in our security arguments. Indeed, the first step in our proof was to characterize the states accepted by the verifier, showing that only honest banknote states are accepted. This step in the proof fails in the LWE-based scheme, which would prevent the proof from going through. Thus, even though the scheme based on LWE is broken, it does not contradict our DLog/1-minCDH and Q-KGEA assumptions holding on the LWE-based group action.

The second difference, is that, with the LWE-based group action, taking the QFT of money states gives elements with meaningful structure: short vectors $\mathbf{x}$ such that $\mathbf{A} \cdot \mathbf{x} = \mathbf{h}$. This structure and it's relation to the original money state are what enables the attack. In contrast, taking the QFT of money states over $\mathcal{X}$ coming from isognies will give terms with no discernible structure.

We believe the above perspective adds to the confidence in our proposal. Indeed, in the LWE-based scheme, the key missing piece is recognizing set elements; if not for this missing piece the scheme *could* be proven secure. By switching to group actions based on isogenies, we add the missing piece. The hope is that even though the source of hardness is now from hard problems on isogenies over elliptic curves instead of lattices, by adding the missing piece we can finally obtain a secure scheme.

References

[Aar09] 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.

[AC12] 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.

[ADMP20] 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.

[AMR22] 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.

[BBBV97] Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and weaknesses of quantum computing. *SIAM J. Comput.*, 26(5):1510–1523, Oct 1997.

[BCM$^+$18] 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.

[BDGM20] 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. https://eprint.iacr.org/2020/1024.

[BDS16] Shalev Ben-David and Or Sattath. Quantum tokens for digital signatures, 2016. https://arxiv.org/abs/1609.09047.

[BGMZ18] 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.

[BGZ23] 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.

[BKV19] 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.

[BS20a] Amit Behera and Or Sattath. Almost public quantum coins. Cryptology ePrint Archive, Report 2020/452, 2020. https://eprint.iacr.org/2020/452.

[BS20b] 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.

[CD23] 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.

[CJS14] Andrew Childs, David Jao, and Vladimir Soukharev. Constructing elliptic curve isogenies in quantum subexponential time. *Journal of Mathematical Cryptology*, 8(1):1–29, 2014.

[CK20] Leonardo Colò and David Kohel. Orienting supersingular isogeny graphs. *Journal of Mathematical Cryptology*, 14:414–437, 10 2020.

[CLM+18] 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.

[Cou06] Jean-Marc Couveignes. Hard homogeneous spaces. Cryptology ePrint Archive, Report 2006/291, 2006. https://eprint.iacr.org/2006/291.

[CPDDF+19] 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.

[Dam92] Ivan Damgård. Towards practical public key systems secure against chosen ciphertext attacks. In Joan Feigenbaum, editor, *CRYPTO'91*, volume 576 of *LNCS*, pages 445–456. Springer, Heidelberg, August 1992.

[DFK+23] 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.

[DG19] Luca De Feo and Steven D. Galbraith. SeaSign: Compact isogeny signatures from class group actions. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part III*, volume 11478 of *LNCS*, pages 759–789. Springer, Heidelberg, May 2019.

[DHK+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.

[DJP14] 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.

[DM20] 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.

[EH00] Mark Ettinger and Peter Høyer. On quantum algorithms for noncommutative hidden subgroups. *Advances in Applied Mathematics*, 25(3):239–251, 2000.

[FGH+12] 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.

[FKL18] 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.

[GGH15] 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.

[HYY23] Minki Hhan, Takashi Yamakawa, and Aaram Yun. Quantum complexity for discrete logarithms and related problems. Cryptology ePrint Archive, Paper 2023/1054, 2023. https://eprint.iacr.org/2023/1054.

[JLS21] 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.

[JS08] Tibor Jager and Jörg Schwenk. On the equivalence of generic group models. In Joonsang Baek, Feng Bao, Kefei Chen, and Xuejia Lai, editors, *ProvSec 2008*, volume 5324 of *LNCS*, pages 200–209. Springer, Heidelberg, October / November 2008.

[Kan18] Daniel M. Kane. Quantum money from modular forms, 2018. https://arxiv.org/abs/1809.05925.

[KLS22] Andrey Boris Khesin, Jonathan Z Lu, and Peter W Shor. Publicly verifiable quantum money from random lattices, 2022. https://arxiv.org/abs/2207.13135v2.

[KSS21] Daniel M. Kane, Shahed Sharif, and Alice Silverberg. Quantum money from quaternion algebras. Cryptology ePrint Archive, Report 2021/1294, 2021. https://eprint.iacr.org/2021/1294.

[LAF$^+$10] 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.

[LMZ23] 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.

[Lut10] Andrew Lutomirski. An online attack against wiesner's quantum money, 2010. https://arxiv.org/abs/1010.0256.

[LZ19] 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.

[Mau05] 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.

[MM22] Luciano Maino and Chloe Martindale. An attack on SIDH with arbitrary starting curve. Cryptology ePrint Archive, Report 2022/1026, 2022. https://eprint.iacr.org/2022/1026.

[MZ22] 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.

[NOTT20] Kohei Nakagawa, Hiroshi Onuki, Atsushi Takayasu, and Tsuyoshi Takagi. L_1-norm ball for CSIDH: Optimal strategy for choosing the secret key space. Cryptology ePrint Archive, Report 2020/181, 2020. https://eprint.iacr.org/2020/181.

[OZ23] Emmanuela Orsini and Riccardo Zanotto. Simple two-round OT in the explicit isogeny model. Cryptology ePrint Archive, Report 2023/269, 2023. https://eprint.iacr.org/2023/269.

[Pan23] Lorenz Panny. Csi-fish really isn't polynomial-time, 2023. https://yx7.cc/blah/2023-04-14.html.

[Pei20] 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.

[Reg05] 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.

[Rob21] 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.

[Rob23] 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.

[Rog06] Phillip Rogaway. Formalizing human ignorance. In Phong Q. Nguyen, editor, *Progress in Cryptology - VIETCRYPT 06*, volume 4341 of *LNCS*, pages 211–228. Springer, Heidelberg, September 2006.

[RS06] Alexander Rostovtsev and Anton Stolbunov. Public-Key Cryptosystem Based On Isogenies. Cryptology ePrint Archive, Report 2006/145, 2006. https://eprint.iacr.org/2006/145.

[RZ21] 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.

[Sho94] Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *35th FOCS*, pages 124–134. IEEE Computer Society Press, November 1994.

[Sho97] 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.

[Shp08] 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.

[Sti05] 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.

[Wie83] Stephen Wiesner. Conjugate coding. *SIGACT News*, 15(1):78–88, January 1983.

[Win99] A. Winter. Coding theorem and strong converse for quantum channels. *IEEE Trans. Inf. Theor.*, 45(7):2481–2485, November 1999.

[WW21] 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.

[YZ22] Takashi Yamakawa and Mark Zhandry. Verifiable quantum advantage without structure. In *63rd FOCS*, pages 69–74. IEEE Computer Society Press, October / November 2022.

[Zha19a] Mark Zhandry. How to record quantum queries, and applications to quantum indifferentiability. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 239–268. Springer, Heidelberg, August 2019.

[Zha19b] 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.

[Zha21] 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.

[Zha22] 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.