Quantum Walks and Application to Quantum Money

Jake Doliskani* and Seyed Ali Mousavi†

Department of Computing and Software, McMaster University

*jake.doliskani@mcmaster.ca

†mousas26@mcmaster.ca

Abstract

This thesis explores the foundations of quantum computation, focusing on quantum walks and their application to quantum money. Quantum walks, particularly continuous-time quantum walks based on group actions, serve as a powerful computational tool with applications in search algorithms and cryptographic protocols. We examine their mathematical structure and their advantages over classical random walks, emphasizing their efficiency in state evolution and probability distribution spreading. As a part of this work, we examine efficient implementations of transforms such as the Quantum Fourier Transform (QFT) and the Quantum Hartley Transform (QHT), analyzing their role in encoding quantum states for secure cryptographic applications. In particular, we discuss a novel instantiation of a quantum money scheme based on QHT, leveraging its unique properties for improved security and efficiency. To ensure the robustness of this quantum money scheme, we develop a verification mechanism utilizing quantum walks. Unlike previous approaches, which rely on standard quantum state measurements, our method employs continuous-time quantum walks to authenticate quantum money, preventing counterfeiting while maintaining computational feasibility. Additionally, we present a detailed discussion on the efficient implementation of this scheme, including optimized circuit designs and error mitigation strategies.

1 Introduction to Quantum Computation

Introduction

Quantum money, first introduced in a seminal paper by Wiesner [29], is a form of currency represented by quantum states. Unlike classical money, an ideal quantum bill cannot be counterfeited due to the no-cloning theorem of quantum mechanics, which prohibits making an identical copy of an unknown quantum state. Wiesner's original scheme (now termed a *private-key quantum money* scheme) had significant practical drawbacks: it required the issuing bank's secret key to verify each banknote, meaning the bank had to be involved in every transaction. This reliance on the bank for verification severely limits the usability of private-key quantum money schemes.

In 2009, Aaronson [1] proposed the first *public-key quantum money* scheme, in which anyone can verify a quantum banknote while only the bank can mint (issue) new valid banknotes. Public-key quantum money removes the transactional bottleneck of involving the bank every time. Aaronson's specific scheme was later broken by Lutomirski et al. [24]. In the years since, several alternative public-key quantum money constructions have been explored [2, 17, 30, 19, 20, 23, 31]. However, each of these proposals has either been broken by further cryptanalysis [14, 26, 7, 23] or relies on unconventional cryptographic assumptions, making their security or practicality questionable.

Quantum Money from Group Actions and the Fourier Transform. A promising candidate for secure public-key quantum money based on more standard assumptions was recently proposed by Zhandry [31]. In Zhandry's scheme, the core idea is to use an abelian group action as the foundation for the quantum state. Each money state (quantum banknote) is prepared as a *group-action Fourier state*, and the corresponding serial number is an element of the underlying group (we will elaborate on these terms in later chapters). The verification algorithm in this scheme uses a group-action phase kickback routine to extract the serial number from the quantum banknote and check its validity. Doliskani [15] later proved that Zhandry's scheme is secure in the generic group action model, providing evidence for its security under well-defined assumptions.

This Work – Motivation and Contributions. In this thesis, we adapt Zhandry's group-action quantum money scheme by replacing its use of the Quantum Fourier Transform with the *Quantum Hartley Transform* (QHT) over finite abelian groups. The motivation behind this substitution is multifold. First, using the Hartley transform causes the banknotes to have *real* amplitudes instead of complex amplitudes. We believe this shift from complex to real quantum states may offer both computational and theoretical advantages. For example, certain mathematical identities hold for any two real orthonormal bases that do not generally hold for complex bases; such differences can affect the properties of quantum states and were a barrier in prior analyses (indeed, an attempted "quantum lightning" construction in [31] failed due to properties of complex phases that do not arise with real amplitudes). Beyond this specific context, our work is a first step toward employing real-valued quantum transforms in quantum cryptography and quantum algorithms. To the best of our knowledge, this is the first significant quantum cryptographic construction that makes essential use of the quantum Hartley transform. We hope that exploring real-amplitude quantum states will inspire further research into new quantum algorithms and optimizations for real-valued transforms.

We now summarize our main contributions:

- We propose a new **public-key quantum money scheme** based on abelian group actions and the quantum Hartley transform. This scheme is an adaptation of Zhandry's group-action money scheme, modified to output real-amplitude quantum states.
- We identify a breakdown in the original verification procedure when the Fourier transform is

naively replaced by the Hartley transform: the Hartley-based verification fails to distinguish certain illegitimate banknotes. To overcome this, we design a **new verification algorithm** that relies on applying additional group action "twists" and utilizes quantum walks. This new verification algorithm successfully differentiates valid banknotes from forgeries despite the use of real amplitudes.

- We show that the serial number associated with a money state (which is a hidden group element) can be efficiently computed from the quantum state. In particular, we develop a **continuous-time quantum walk algorithm** to extract the serial number of a given banknote state. This algorithm leverages the structure of the group action and the spectral properties of an associated Cayley graph. Our method demonstrates that one can recover the hidden group element (serial) with high success probability, which not only underpins the new verification technique but is also of independent interest.

- In the course of our construction, we introduce a new algorithm for efficiently implementing the **quantum Hartley transform** and related real transforms. We present a recursive technique that exploits the structure of the Hartley transform, yielding a lower gate complexity than prior approaches in the literature. We also illustrate how other real-valued quantum transforms (e.g., the quantum sine transform) can be implemented using the Hartley transform as a subroutine. These results contribute to the broader goal of optimizing real quantum transforms for practical use.

In summary, our work combines ideas from quantum cryptography (public-key quantum money), quantum algorithms (quantum walks), and quantum signal processing (Hartley and related transforms) to create a novel and potentially practical quantum money scheme.

Thesis Organization. The remainder of this thesis is organized as follows. In **Chapter 2**, we provide background on the basics of quantum computation and the theory of group actions, including the concept of cryptographic group actions which underlie our money scheme. **Chapter 3** covers quantum walks, both continuous-time and discrete-time, outlining their differences and importance in quantum algorithms. In **Chapter 4**, we apply the quantum walk framework to group actions: we describe how a continuous-time quantum walk on a group action can be simulated efficiently, a crucial step for our serial number extraction algorithm. **Chapter 5** details the quantum money scheme itself and the role of the Hartley transform in its construction. We explain how the scheme is formulated and discuss the advantages and challenges introduced by using the Hartley transform. Finally, **Chapter 6** presents the verification algorithm for the Hartley-based quantum money. This chapter shows how we use quantum walks (as developed in Chapter 4) to reliably verify banknotes and compute their serial numbers, thereby addressing the verification issues that arose from the use of real amplitudes.

Verification Algorithm Using Quantum Walks The Hartley-based quantum money scheme introduced in the previous chapter requires a new verification procedure to address the issues that arise from using real amplitude states. In this chapter, we describe and analyze the verification algorithm, which leverages continuous-time quantum walks on the group action graph (and their efficient simulation from Chapter 4) to extract the information needed to authenticate a banknote.

2 Challenges with Naïve Verification

Before detailing the new algorithm, let us briefly recap why the straightforward approach fails. In the Fourier-based scheme, verification was done by a single "kickback" operation using the claimed

serial g, and measuring an auxiliary register. In the Hartley-based scheme, if we attempted the analogous one-step verification, we would perform the controlled-g action and measure the auxiliary system. A genuine state $|\$_g\rangle$ (the Hartley money state for serial g) would cause some interference pattern in the auxiliary register, but unlike the Fourier case, it does not return the auxiliary to $|x_0\rangle$ deterministically. In fact, there is an ambiguity: certain superpositions of eigenstates corresponding to g and $-g$ (or other group-related variants) can produce the same measurement statistics in that one-step test. This means an adversary might prepare a counterfeit state that is not a legitimate $|\$_g\rangle$ but still passes the one-step verification with non-zero probability. Essentially, the Hartley transform being real means we lost some phase information, and a single measurement cannot distinguish some mirrored states.

To overcome this, our strategy is to perform a more complete measurement of the state's "phase spectrum." Instead of just one operation and measurement, we will use the quantum walk (with Hamiltonian $A = A(X, S)$ as defined earlier) to perform a form of phase estimation.

3 Using Quantum Walks to Extract the Serial

The central observation is that the money state $|\$_g\rangle$ is an eigenstate of the walk's Hamiltonian A on the group action Cayley graph. To see this, note that $|\$_g\rangle$ (Fourier or Hartley) lies in the subspace spanned by $\{|x\rangle : x \in X\}$ (we no longer explicitly keep the group element register, since after minting it's entangled or measured away). Consider the adjacency operator A acting on a basis state $|x\rangle$. By definition,

$$A|x\rangle = \sum_{s\in S} |s * x\rangle,$$

summing over all neighbors under the generators. Now, $|\$_g\rangle$ is (up to normalization) $\sum_{h\in G} f(h)\,|h * x_0\rangle$ for some coefficient function $f(h)$. If one applies A to this state:

$$A|\$_g\rangle = \sum_{s\in S}\sum_{h\in G} f(h)\,|s * (h * x_0)\rangle = \sum_{h\in G} f(h) \sum_{s\in S} |(sh) * x_0\rangle.$$

Now change variable $h' = sh$ in the inner sum. Since s runs over all of S, h' runs over $Sh = \{sh : s \in S\}$ which is exactly the set of neighbors of h in the Cayley graph of G. For an abelian group, one can show (and it is known from spectral graph theory) that $\sum_{s\in S} |sh * x_0\rangle$ corresponds to the same state as $\lambda_g|h * x_0\rangle$ where $\lambda_g = \sum_{s\in S} \chi_g(s)$ for Fourier states, or $\lambda'_g = \sum_{s\in S} \mathrm{cas}(2\pi\langle g, s\rangle/N)$ for Hartley states. In short, $|\$_g\rangle$ is an eigenvector of A with eigenvalue λ'_g (a real number that depends on g and S). This fact was formalized earlier: Lemma 8.2 showed that the money state is an eigenstate of A.

For example, if $G = \mathbb{Z}_N$ and $S = \{1, -1\}$, a Fourier-based $|\$_g\rangle$ is essentially the character χ_g, and indeed χ_g is an eigenfunction of the adjacency (which is like a cosine operator) with eigenvalue $2\cos(2\pi g/N)$. The Hartley-based state corresponds to a combination of χ_g and χ_{-g} (the real and imaginary parts combined), and ends up with the same eigenvalue $2\cos(2\pi g/N)$ as well. So either way, the "frequency" g maps to a specific eigenvalue λ.

The verification algorithm will do the following: it will run phase estimation on the unitary $U = e^{iA\tau}$ for some carefully chosen time τ, using the state $|\psi\rangle$ as input. Phase estimation is a standard quantum algorithm that, given an eigenstate of U, yields an estimate of the eigenphase $2\pi\phi$ (where $e^{i2\pi\phi}$ is the eigenvalue of U) to some specified precision. In our case, since A has eigenvalue λ on $|\psi\rangle = |\$_g\rangle$, the unitary $e^{iA\tau}$ has eigenvalue $e^{i\lambda\tau}$. Thus phase estimation will give us an estimate of $\lambda\tau \pmod{2\pi}$. By choosing τ appropriately, we can ensure $\lambda\tau$ uniquely corresponds to λ (within the precision we work at).

We cannot run $e^{iA\tau}$ directly, but Chapter 4 showed how to simulate it efficiently with a discrete-time algorithm. So when we say "phase estimation on $U = e^{iA\tau}$," we imply using the simulation routine as a subroutine in the phase estimation circuit. This will incur some error ϵ, but we can make it negligible with enough resources.

Now, because λ (the eigenvalue of A) is related to g, obtaining λ in effect gives us information about g. However, one eigenvalue might correspond to two possible g values (e.g., $\cos(2\pi g/N) = \cos(2\pi(N-g)/N)$). So a single run of phase estimation may not fully determine g. This is where performing the procedure for multiple different times τ (or perhaps using a slightly different Hamiltonian) helps. By getting λ at different scales, we can solve for g using the known functional form of λ in terms of g.

In [31], it was shown that by using two or three carefully chosen multiples of time (or equivalently using the fact that λ as a function of g has a known form like a cosine series), one can uniquely recover g. In our scheme, since S is known and fixed, and presumably S is such that the map $g \mapsto \lambda_g = \sum_{s \in S} \chi_g(s)$ is injective up to trivial symmetries, a few phase estimates suffice. For instance, if $S = \{1, -1\}$, knowing $\cos(2\pi g/N)$ to high precision lets you determine g up to the symmetry $g \leftrightarrow N - g$. But if we also had a way to break the symmetry (maybe by using an asymmetric choice of τ or an additional generator in S that is not symmetric), we can get the sign too. Alternatively, we run phase estimation on a slightly modified Hamiltonian A' (maybe corresponding to a different generating set that breaks symmetry) if available.

For simplicity, let's suppose two runs with different τ values are enough. The output of the verification algorithm is then determined by: - If the estimated g^* from the phase(s) matches the provided serial g, output `accept`. - Otherwise, output `reject`.

Because a genuine $|\$_g\rangle$ is exactly an eigenstate with eigenvalue λ_g, phase estimation will return (with high probability) a number very close to $\lambda_g \tau$ modulo 2π. Given the precision, we decode it to the nearest valid $\lambda_{g'}$ value, which should be λ_g. Hence we recover $g' = g$ and accept. For a counterfeit state, two things can happen: 1. The state $|\psi\rangle$ is a superposition of eigenstates (not a single eigenstate). In that case, phase estimation will give some distribution of eigenvalues, effectively picking one at random (with probabilities proportional to the projection of $|\psi\rangle$ onto those eigencomponents). It is highly unlikely that this random outcome consistently points to the same g^* that the adversary wants (especially if they try to match a specific g that they announced as serial). In fact, unless the counterfeit state was itself aligned with one particular eigenvalue, the measurement will likely produce a result that does not correspond to the claimed serial, leading to rejection. 2. The state $|\psi\rangle$ happens to be an eigenstate, but with a wrong eigenvalue (i.e., corresponding to some $g' \neq g$). In that case, the phase estimation will reliably give $\lambda_{g'}$, and we will decode g' which won't match the claimed g, so we reject. This scenario would occur if the adversary somehow prepared a valid-looking state for a different serial g' but presented it with serial g—clearly that should be rejected.

Thus, the only way to pass is to present a state that is an eigenstate with eigenvalue matching the serial. But the only known way to produce such a state is essentially to follow the minting procedure (due to the one-wayness of the group action, they can't derive $|\$_g\rangle$ without knowing g or breaking the assumption).

One might wonder: phase estimation is a probabilistic algorithm; what if it sometimes yields the correct g^* and sometimes not, could an adversary repeat the verification or something? In practice, the banknote verification either passes or fails once; an adversary doesn't get to try multiple times on the same note without potentially decohering it or being caught. We design our parameters such that the completeness (accepting a good note) is very high and the soundness (accepting a forgery) is extremely low, say negligible in n. This might involve repeating the phase estimation a constant number of times to amplify confidence, or using a high precision so that the error probability is

negligible.

Finally, our algorithm implicitly uses the ability to perform the controlled-U operations for phase estimation. That is non-trivial but our quantum walk simulation provided a way to implement $e^{iA\tau}$, and controlled versions can be constructed by controlling each gate in the simulation (which increases complexity by at most a factor poly(n)). We have to ensure that the simulation error is accounted for in the phase estimation accuracy—this can be handled by choosing simulation parameters fine enough that the error is smaller than the inverse of our runtime or something along those lines.

4 Detailed Verification Procedure

To summarize, here is a more step-by-step description of the verification algorithm for a banknote $(g, |\psi\rangle)$:

1. **Preparation:** Determine a set of one or more time parameters $\{\tau_1, \tau_2, \ldots, \tau_m\}$ to be used for phase estimation. These might be hardcoded or depend on N and S. For example, τ_1 could be a basic time and τ_2 another that breaks degeneracy.

2. **Phase Estimation Rounds:** For each τ_j:

   (a) Use the quantum walk simulation circuit to implement controlled-$e^{iA\tau_j}$ on $|\psi\rangle$. (In practice, we append an ancilla register to accumulate the phase, as in standard phase estimation algorithms. We prepare a uniform superposition over some range of time steps, apply controlled-U gates, and perform inverse Fourier transform on the phase register to get an estimate. This is the standard routine.)

   (b) Measure the phase register to obtain an estimate $\tilde{\phi}_j$ of $\lambda_g \tau_j/(2\pi)$ (mod 1). Multiply by $2\pi/\tau_j$ to get an estimate $\tilde{\lambda}_j$ of λ_g.

3. **Compute Serial Candidate:** From the collection $\{\tilde{\lambda}_j\}$, solve for the group element g^* that would produce those eigenvalues. This might involve inverting the known relation $\lambda = \sum_{s\in S} \mathrm{cas}(2\pi\langle g, s\rangle/N)$ or whatever the formula is in our context. Because of estimation error, we choose the nearest matching g^* in G that would give eigenvalues close to the $\tilde{\lambda}_j$ observed.

4. **Accept/Reject:** If $g^* = g$ (and all rounds of phase estimation yielded results consistent with a single g^*, which should happen for a valid note), then output `accept`. Otherwise, output `reject`.

This algorithm may seem complex, but it can be optimized. In practice, we could combine multiple phase estimation steps into one larger quantum circuit that extracts g in one go, but conceptually, it's easier to think of them separately.

5 Analysis of Security and Efficiency

The new verification algorithm is more involved than the original Fourier-based one, so we must check that it is still efficient and that it indeed thwarts counterfeiting.

Efficiency: Each phase estimation round uses polynomial-time quantum operations. The Hamiltonian simulation of $e^{iA\tau}$ for sparse A on N-dimensional space can be done in poly$(\log N)$ time (since queries to the group action oracle are $O(1)$ and $|S|$ is small). The phase estimation itself might need $O(\log N)$ qubits of precision to distinguish different g values, and $O(\log N)$ repetitions of U in

superposition. Overall, each round is poly(n) time. A constant number of rounds doesn't change the polynomial class. So yes, verification is QPT. In fact, in asymptotic terms, if minting took $\tilde{O}(n^c)$ time, verification might take $\tilde{O}(n^{c+1})$ or similar due to extra overhead, but still polynomial.

Completeness: A genuine banknote state $|\$_g\rangle$ will be accepted with high probability. The state is an eigenstate of A, so phase estimation yields exactly λ_g (with small error ϵ). Decoding that yields $g^* = g$. The only possible errors are from phase estimation not being perfect: there is a small probability it gives the wrong eigenvalue estimate (exponentially small in the number of ancilla qubits used). By choosing parameters, we ensure this error is negligible, say $< 2^{-n}$ or similar. So the chance a real note is mistakenly rejected is negligible.

Soundness: A counterfeiter's goal is to have $\mathsf{Verify}(g, |\psi\rangle) = \texttt{accept}$ for some $|\psi\rangle$ that they created without running Mint. Essentially, they either try to reuse a state they already saw or combine states to get two accepted notes, etc. If they attempt to create a new state for an unused serial g, they face the following: - If $|\psi\rangle$ is not an eigenstate of A, when we run phase estimation, the outcome is a random eigenvalue from its spectral decomposition. It would be a huge coincidence if that random eigenvalue corresponded to exactly the claimed g. And even if by some luck one phase estimation round gave a favorable result, with two rounds the chance to consistently impersonate the same g is even smaller. In fact, by the union bound, the probability that a wrong state passes all checks should be negligible. A rigorous argument would be that if $|\psi\rangle$ has overlap on eigenstates $\{|e_j\rangle\}$ with eigenvalues λ_j, then the verification accepts only if for all j with nonzero overlap, the decoded g_j equals the claimed g. If any overlap component has a different underlying group element, that component will cause failure in at least one of the phase estimation rounds with high probability. Unless the state was aligned to one particular g', in which case the best they can do is claim it is g'. - If $|\psi\rangle$ *is* an eigenstate but for $g' \neq g$, then our algorithm will find g' and compare to g and reject. So the adversary would have to guess the serial number's eigenstate perfectly, which means they basically guessed the bank's secret or solved the one-way problem.

The most cunning forgery could be to use a genuine note $(g, |\$_g\rangle)$ and try to produce a second independent state $|\$_g\rangle$ for the same g. But that is essentially copying a quantum state which is unknown to them (they only know g classically). Given g, can they produce $|\$_g\rangle$? That's the one-way group action problem: producing $|\$_g\rangle$ is as hard as computing $g * x_0$ for all group elements in superposition, which requires knowledge of g in a way that they don't have (they know g but to create the superposition exactly with correct phases is akin to solving for the hidden phase structure—likely hard without the trapdoor). In fact, if they could do that efficiently for any given g, they would break the scheme by minting their own money. So by assumption, that's infeasible.

Thus, the only resource the adversary has is perhaps other genuine notes they possess. Could they entangle two notes or something to produce two valid ones? If they have one note, trying to make two will likely destroy the original or produce two imperfect copies that fail verification (no-cloning strikes again). If they have two different notes $(g_1, |\$_{g_1}\rangle)$ and $(g_2, |\$_{g_2}\rangle)$, those states are independent, and forging a new one still reduces to above.

Therefore, we argue the scheme remains secure: forging in any meaningful sense would require solving a hard problem (inverting the group action or cloning a quantum state).

In conclusion, by using continuous-time quantum walks and phase estimation, our verification algorithm retrieves the hidden serial number encoded in the quantum money state and compares it to the claimed serial. This not only fixes the issue introduced by using the Hartley transform (real amplitudes), but it does so in polynomial time. We have thus achieved a public-key quantum money scheme based on standard assumptions (one-way group actions) with an extra novel feature of using real-valued quantum states and demonstrating the power of quantum walks in quantum cryptanalysis and verification.