How to Sign Quantum Messages

Mohammed Barhoush and Louis Salvail

Université de Montréal (DIRO), Montréal, Canada
mohammed.barhoush@umontreal.ca salvail@iro.umontreal.ca

Abstract. Signing quantum messages has long been considered impossible even under computational assumptions. In this work, we challenge this notion and provide three innovative approaches to sign quantum messages that are the first to ensure authenticity with public verifiability. Our contributions can be summarized as follows:

1. *Time-Dependent Signatures:* We introduce the concept of time-dependent (TD) signatures, where the signature of a quantum message depends on the time of signing and the verification process depends on the time of the signature reception. We construct this primitive assuming the existence of post-quantum secure one-way functions (pq-OWFs) and time-lock puzzles (TLPs).
2. *Dynamic Verification Keys:* By utilizing verification keys that evolve over time, we eliminate the need for TLPs in our construction. This leads to TD signatures from pq-OWFs with dynamic verification keys.
3. *Signatures in the BQSM:* We then consider the bounded quantum storage model, where adversaries are limited with respect to their quantum memories. We show that quantum messages can be signed with information-theoretic security in this model.

Moreover, we leverage TD signatures to achieve the following objectives, relying solely on pq-OWFs:

- *Authenticated Quantum Public Keys:* We design a public-key encryption scheme featuring authenticated quantum public-keys that resist adversarial tampering.
- *Public-Key Quantum Money:* We present a novel TD public-key quantum money scheme.

1 Introduction

Given the consistent advancements in quantum computing, it is expected that future communications will feature quantum computers transmitting over quantum channels. A fundamental question naturally arises: how can quantum data be securely transmitted within the emerging quantum internet? One option is for users to share secret keys through secure channels. Yet, this option quickly becomes unwieldy as the number of users grows or when secure channels are not readily available. Another option is to rely on quantum key distribution [14], but this too is inefficient for large-scale applications as it requires several rounds of interaction with each user. In contrast, classical information can be encrypted and authenticated non-interactively via public channels. Can a similar feat be achieved quantumly?

Towards this goal, several works have shown how to encrypt quantum information from standard assumptions [3, 4]. Yet, somewhat surprisingly, signing quantum information has been shown to be impossible [12, 4]. On the other hand, classical digital signature schemes have been crucial cryptographic primitives - realizing a range of applications including email certification, online transactions, and software distribution.

As a result of the impossibility, researchers have focused on an alternative approach to quantum authentication called *signcryption*. In this setting, the sender uses the recipient's public encryption key to encrypt a message before signing it. Only the recipient can verify that the signature is authentic by using their secret decryption key which means that signcryption does not allow for public verifiability – only a single receipient can verify. Such schemes clearly rely on assumptions for public-key encryption such as trapdoor functions. Until this point, it was widely believed that signcryption is the only way to evade the impossibility result. In fact, Alagic, Gagliardoni, and Majenz [4] carried out an in-depth analysis on the possibility of signing quantum information and concluded that *"signcryption provides the only way to achieve integrity and authenticity without a pre-shared secret"*. In this work, we address and revisit the questions:

Are there alternative methods to authenticate quantum information without a pre-shared secret?

Interestingly, this question has important implications in quantum public key encryption (QPKE). Traditionally, classical public-key encryption (PKE) can not be built from one-way functions [31] and requires stronger assumptions such as trapdoor functions. However, the works [29, 22] show that PKE can be achieved from post-quantum secure classical one-way functions (pq-OWF) if the public-keys are quantum! Yet, these constructions face a serious problem: authenticating a quantum state is difficult. This issue is not addressed in these works; as a result, these constructions need to assume secure quantum channels for key distribution which is quite a strong physical setup assumption given the goal of encryption is to establish secure communication over insecure channels.

On the other hand, there are well-established methods to distribute classical keys, for instance through the use of classical digital signatures. Such procedures are referred to as *public-key infrastructure*.

Ideally, we aim to establish encryption based on public-key infrastructure and one-way functions. More simply, we want to authenticate the quantum keys in QPKE schemes using classical certification keys. Prior to this work, the only way to do this would be through the use of signcryption. But, then we are back to relying on assumptions of classical PKE; not to mention the inconveniences of having to encrypt each public-key specific to the recipient. In particular, the following is another critical question addressed in this work:

Is QPKE with publicly-verifiable quantum public-keys possible from pq-OWFs?

Another relevant application of quantum signatures pertains to quantum money. Quantum money was first introduced by Weisner [51] in 1969 as a method to prevent counterfeiting by utilizing the unclonability of quantum states. There are two main categories of quantum money: private-key and public-key. Public-key systems rely on physical banknotes that are hard to counterfeit and are verifiable with a public-key. Meanwhile, private-key systems rely on a third party that oversees transactions and approves or declines payments. The main advantage of public-key systems is that transactions do not require interaction with a third party. As a result, such systems are more feasible for large-scale applications and provide more privacy to users.

Unfortunately, public-key systems are more difficult to realize. On one hand, several works have constructed private-key quantum money [51, 34, 43] from standard assumptions. On the other hand, public-key quantum money has only been constructed assuming indistinguishability obfuscation [2, 53, 48] or from new complex mathematical techniques and assumptions [25, 36, 35] which we are only beginning to understand and, some of which, have been shown to be susceptible to cryptoanalytic attacks [15]. Sadly, all existing constructions for indistinguishability obfuscation are built on assumptions that are post-quantum insecure [9, 33, 32] or on new assumptions [27, 50] that have been shown to be susceptible to cryptoanalytic attacks [30].

As a result, public-key quantum money remains an elusive goal. Prompted by this predicament, Ananth, Hu, and Yuen [8] recently conducted a comprehensive investigation into public-key quantum money, culminating in a strong negative result. In particular, they showed that public-key quantum money is impossible to build using any collusion-resistant hash function in a black-box manner under certain conditions. This discussion raises the following important question addressed in this work:

Is public-key quantum money possible from standard computational assumptions?

1.1 Summary of Results

The impossibility of signing quantum information was first discussed by Barnum, Crepeau, Gottesman, Smith, and Tapp [12] and later established more rigorously

by Alagic, Gagliardoni, and Majenz [4]. Informally, their rationale was that any verification algorithm, denoted as V, which can deduce the quantum message (or some information thereof) from a signature, can be effectively inverted $V^\dagger$ to sign a different message.

The central innovation of this work lies in recognizing that, in specific scenarios, this inversion attack consumes a prohibitive amount of resources. This study explores two variations of this concept, with the resource factor taking the form of either time or quantum space, as we clarify in the ensuing discussion.

The first approach to signing quantum information is to vary the signing and verification procedures with respect to time. We term this approach *time-dependent (TD) signatures* and present two constructions realizing this primitive. The first construction relies on the assumption of *time-lock puzzles* (TLP) [45, 16] and pq-OWFs. Here, the TLPs ensure that the verification procedure demands prohibitive time to inverse. We note that TLPs can be built from pq-OWFs [16, 21] in the quantum random oracle model (QROM) [17], where all parties are given quantum oracle access to a truly random function.

The second construction also enforces a verification process that is time-consuming to invert. However, in this case, this enforcement is achieved more directly by delaying the announcement of the verification key. Specifically, we authenticate and encrypt messages in a symmetric-key fashion and announce the key later. By the time the key is revealed, allowing users to validate old signatures, it is too late to exploit the key for forging new signatures. As a result, the verification key must be continually updated to allow for new signature generation. We denote schemes with dynamic verification keys as *TD-DVK signatures* for brevity. An attractive aspect of TD-DVK signatures is that they can be based solely on pq-OWFs, yielding surprisingly powerful applications from fundamental assumptions.

We demonstrate how to utilize TD-DVK signatures to build more secure QPKE schemes where the quantum public-keys are signed with TD-DVK signatures. This approach allows basing QPKE on pq-OWFs and public-key infrastructure, i.e. enabling encryption with tamper-resilient quantum public-keys. Furthermore, we employ TD-DVK signatures to construct the first public-key quantum money scheme based on a standard computational assumption namely pq-OWFs. The verification key in this setting is dynamic, preventing a completely offline money scheme. Fortunately, we are able to mitigate this issue and obtain a completely offline public-key quantum money scheme by utilizing our TD signature scheme, albeit by relying on TLPs.

Our second strategy for signing quantum information involves a verification process that necessitates an impractical amount of quantum memory to invert. To achieve this, we need to assume the adversary's quantum memory (qmemory) size is limited leading us to the *bounded quantum storage model* (BQSM) [23]. We construct a signature scheme for quantum messages in this model that requires users to have ℓ^2 qubits of quantum memory, where ℓ is the size of the quantum message to be signed, and is information-theoretically secure against adversaries with s quantum memory where s can be set to any fixed value that is polynomial

with respect to the security parameter. Note that $\mathtt{s}$ is not related to ℓ and only has an effect on the length of the quantum transmissions involved.

1.2 Future Work

Time acts a form of synchronization between players in a protocol. This work demonstrates the power of utilizing time in cryptography. Specifically, it shows how incorporating time-dependence can significantly aid in the construction of fundamental cryptographic primitives such as QPKE, quantum signatures, and public-key quantum money. This is particularly encouraging given the simplicity and security of implementing time-dependence in practice. Therefore, an interesting avenue for future work is to consider what other cryptographic primitives can benefit from this approach.

Another possible topic is the relation between the "time-dependent model" and alternative models in cryptography. For instance, in the well-studied *common reference string model*, a trusted setup provides all parties with the same string. Time is somewhat related to a form of dynamic common reference string that is completely non-random. A more rigorous abstraction of the time-dependent model might help us understand alternative methods to sign quantum messages.

Finally, we believe that the BQSM has not received sufficient attention. Given the practical challenges of storing and operating on quantum states, this model is very well motivated. However, there have been few studies on cryptographic primitives within this model. We hope that this work will encourage further research in this area.

1.3 Related Work

To the best of our knowledge, this is the first work to sign quantum messages without using signcryption. As previously discussed, while signcryption is effective in its own right, it lacks the feature of public verifiability and entails a form of interaction, requiring the distribution of public encryption keys to the signer. Consequently, this is the first approach ensuring the authenticity of quantum messages while maintaining public verifiability.

Meanwhile, the topic of public-key quantum money has been extensively studied. Nevertheless, to our knowledge, this is the first work to achieve this highly elusive goal from standard assumptions, albeit with a time dependence factor. However, it is relevant to acknowledge the work of Roberts and Zhandry in the creation of *franchised quantum money* [46]. Their franchised approach provides an inventive way to avoid some of the challenges of public-key quantum money by providing users with distinct verification keys. Notably, their scheme is based on pq-OWFs, although, it faces some serious drawbacks. Specifically, it requires secure private channels to each and every user at the start in order to distribute the secret verification keys. Moreover, the scheme only supports a limited number of participants and becomes vulnerable if a sufficiently large group of users were to collude.

Finally, no prior works presented methods to authenticate quantum public-keys in QPKE. However, concurrently and independently of our work, Kitagawa, Morimae, Nishimaki, and Yamakawa [37] explored the concept of QPKE with tamper-resilient quantum public-keys, achieving a similar goal. They introduced a security notion akin to IND-CKPA and build a scheme satisfying this notion through a different approach without relying on time. Specifically, they rely on recently discovered properties of quantum states [1] to create tamper-resilient quantum public keys. While their scheme is CKPA-secure under the assumption of pq-OWFs, it does not allow users to verify or detect modified keys, potentially leading to invalid encryptions that exhibit decryption errors. Their work concludes with two open questions:

1. Is it possible to design a CKPA-secure QPKE scheme without decryption errors?
2. Can a CKPA-secure QPKE scheme be built based on potentially weaker assumptions than pq-OWFs?

Both questions are answered affirmatively in our work by incorporating time-dependence, as detailed in Sec. 8. In Constructions 7 & 8, keys are publicly verifiable and thus ciphertexts exhibit no decryption errors, addressing the first question. Furthermore, Construction 8 is CKPA-secure and is based on PRFSs, addressing the second question. Our approach also has the advantage of being more generally applicable to any quantum public-key scheme and achieving a stronger level of security, demonstrating the effectiveness of time as a cryptographic tool.

Another related notion is signatures with certified deletion [41]. In this protocol, the receiver of a signature can perform an operation to delete the signature, producing a certificate that verifies the deletion. In our case, the signature is invalidated after a certain period regardless of the receiver's actions. Note that the techniques of certified deletion cannot be used to sign a quantum message, as otherwise, it would imply a standard signature scheme of quantum messages.

2 Technical Overview

We now describe each contribution in more detail.

Time-Dependent Signatures. A TD signature scheme is the same as a standard signature scheme except that the signing and verification algorithms depend on an additional variable: *time*. Henceforth, we let $\mathtt{t}$ denote the current time and it is assumed that all parties are aware of the current time up to some small error. This may implicitly assume a trusted third party for clock synchronization. More formally, TD signatures are defined as follows:

Definition 1 (Time-Dependent Signatures). *A* TD signature scheme Π *over quantum message space* $\mathcal{Q}$ *consists of the following algorithms:*

- $\mathsf{KeyGen}(1^\lambda)$: *Outputs* $(\mathsf{sk}, \mathsf{vk})$*, where* sk *is a secret key and* vk *is a verification key.*
- $\mathsf{Sign}(\mathsf{sk}, \mathsf{t}, \phi)$: *Outputs a signature* σ *for* $\phi \in \mathcal{Q}$ *based on the time* t *using* sk*.*
- $\mathsf{Verify}(\mathsf{vk}, \mathsf{t}, \sigma')$: *Verifies the signature* σ' *using* vk *and time* t*. Correspondingly outputs a message* ϕ' *or* $\perp$*.*

We now describe the construction for a TD signature in based on TLPs and pq-OWFs.

TLPs is a cryptographic primitive that enables hiding a message for a time t, but allows for decryption in a similar time $t' \approx t$.

Definition 2 (Time-Lock Puzzles (Informal)). *A TLP consists of two algorithms:*

1. $\mathsf{Puzzle.Gen}(1^\lambda, t, s)$: *Generates a classical puzzle* Z *encrypting the solution* $s \in \{0,1\}^\lambda$ *with difficulty parameter* t*.*
2. $\mathsf{Puzzle.Sol}(Z)$: *Outputs a solution* s *for the puzzle* Z*.*

Security informally requires that no QPT adversary can decipher the solution s *of a puzzle* Z *in time* t*.*

Bitansky, Goldwasser, Jain, Paneth, Vaikuntanathan, and Waters [16] showed that TLP can be constructed, assuming the existence of pq-OWFs, from any non-parallelizing language. This result holds true even in the quantum realm. It is worth noting that non-parallelizing languages have been derived from a random oracle [21].

We provide a brief description of our construction for TD signatures. We denote the algorithms for a one-time symmetric authenticated encryption scheme on quantum messages as $(\mathsf{1QGen}, \mathsf{1QEnc}, \mathsf{1QDec})$, which exists unconditionally [12]. Such a protocol allows for the secure authenticated encryption of a single quantum message. To sign a quantum message ϕ, we sample a key $\mathsf{k} \leftarrow \mathsf{1QGen}(1^\lambda)$ and authenticate the message as $\tau \leftarrow \mathsf{1QEnc}(\mathsf{k}, \phi)$. Following this, we generate a TLP Z requiring 1 hour to solve and whose solution is the message $(\mathsf{k}, T, \mathsf{sig})$, where T corresponds to the time at the moment of signing and sig is a signature of (k, T) under a classical signature scheme. Note that signatures on classical messages exist assuming pq-OWFs [47][1] Consequently, the signature of ϕ is (τ, Z).

Assume that a receiver gets the signature at time T'. To verify, the receiver solves the puzzle Z to learn $(\mathsf{k}, T, \mathsf{sig})$. If the signature sig is valid and the time of reception is close to T, i.e. T' is within half an hour from T, then the verifier outputs $\mathsf{1QDec}(\mathsf{k}, \tau)$ to retrieve ϕ. However, it is crucial to understand that the verifier can no longer use the pair (k, T) to sign a new message because by the time the puzzle is unlocked, it has already become obsolete! Specifically, the time elapsed is at least $\mathsf{t} \geq T + 1$, leading future verifiers to reject a message signed under time T.

Note that an alternative approach to using TLPs would be to simply send the encoded message $(\mathsf{k}, T, \mathsf{sig})$ after some delay. Hence, a signature would then

[1] This reduction is classical but holds in the quantum setting as well.

consist of two parts. We do not consider this solution further as it follows in the same way, albeit we need to use delays in the signature transmission. Later, we discuss a solution that does not rely on TLPs or on delays.

Notice how the use of TLP (or delays) gives us the ability to add intricacy to the verification process, and this is precisely what is needed to circumvent the impossibility result.

The Issue with TLPs. Our construction of TD signatures is seemingly hindered by its reliance on TLP. Formally, a TLP does not actually guarantee that no algorithm can decrypt without sufficient time; rather, it ensures that no algorithm can decrypt without sufficient algorithmic depth (CPU-time). We would like to translate this CPU-time into real-time. However, we have limited control over this conversion due to the variations in computer speeds. For instance, silicon gates are slower than gallium arsenide gates.

On a positive note, exact equivalence between CPU-time and real-time is not necessary. As long as there exists a bound on CPU-time based on real-time, then using this bound we can ensure security of our TD signatures. Explicitly, we modify the verification process to only pass if the time of reception is close to the time of generation, but the TLP requires much longer to solve depending on this bound. This prevents even a fast computer from unlocking the puzzle before it must send it, while still allowing slower computers to validate signatures, albeit with some delay.

However, such practical investigations are out of the scope of this work but can be explored more rigorously in future work. In this work, we only focus on the theoretical aspect and assume that CPU-time and real-time are equivalent.

Time-Dependent Signatures with Dynamic Verification Keys. We now show how we can discard TLPs, giving TD signatures in the plain model assuming only pq-OWFs. In this construction, we introduce a novel element: dynamic verification keys that evolve with time. First, note that post-quantum pseudorandom functions (pq-PRF) and symmetric key quantum authenticated encryption scheme (QAE) on quantum messages have both been derived from pq-OWFs [52, 5]. Let $(\mathsf{QEnc}, \mathsf{QDec})$ denote the algorithms for such an QAE scheme and let F be a pq-PRF.

Here is how our construction unfolds: we choose a time-independent secret key sk and generate temporary signing keys $\mathsf{k}_i := F(\mathsf{sk}, i)$ to be used within the time interval $[t_i, t_{i+1})$, where $t_i := i \cdot \delta$ for some fixed time $\delta > 0$. To sign a message ϕ within this time frame, we simply output $\tau_i \leftarrow \mathsf{QEnc}(\mathsf{k}_i, \phi)$. At time t_{i+1}, we announce the verification key $\mathsf{vk}_i := (t_i, t_{i+1}, \mathsf{k}_i)$. This allows any recipient that received τ_i to decrypt and verify the message. Users are required to take note of the time they receive a signature and store it until the verification key is unveiled.

Importantly, an adversary cannot predict a signing key k_i by the security of pq-PRFs. Consequently, they cannot forge a ciphertext in the time frame $[t_i, t_{i+1})$ by the security of the QAE scheme. Later, at time t_{i+1}, any attempt to forge a signature using k_i becomes invalid since users will then verify using newer keys.

On the downside, TD signatures come with an expiration period, meaning they must be utilized within a relatively short time frame from their generation. Note that the same issue exists in our first construction as well. This limitation may restrict their utility in certain applications compared to traditional signatures. However, in some scenarios, this limitation can also be seen as an advantage. In particular, our signatures satisfy a form of "disappearing" security which was heavily studied in the bounded memory (classical and quantum) models. Specifically, in our schemes, an adversary is incapable of forging a signature of a message ϕ even if it received a signature of ϕ earlier!

Signatures in the Bounded Quantum Storage Model. Next, we introduce an alternative approach to sign quantum information within the BQSM without relying on time. In this model, an adversary $\mathcal{A}_{\mathsf{s}}$ possesses unlimited resources at all times except at specific points, where we say the *qmemory bound applies*, reducing $\mathcal{A}_{\mathsf{s}}$'s quantum memory to a restricted size of s qubits[2]. The adversary is again unrestricted after the memory bound applies and is never restricted with respect to its computational power.

Our construction is inspired by the BQS signature scheme for classical messages presented by Barhoush and Salvail [11]. We first briefly describe their construction. In their scheme, the public-key is a large quantum state and each user can only store a subset of the quantum key. Consequently, different users end up with distinct subkeys. This idea is realized through a primitive called *program broadcast* which was achieved unconditionally in the BQSM [11] and is defined in Sec. 3.9[3]. The secret key is an appropriate large function P_{pk} and the public-key consists of a quantum program broadcast of P_{pk}. Each user acquires a distinct evaluation of P_{pk} which serves as a *verification subkey*. The broadcast is terminated after all users have received key copies and before the signing process.

To facilitate signing, which must be verifiable by all users, one-time programs are utilized – this powerful primitive can be built unconditionally in the BQSM [11][4]. We provide a simplified description as the actual construction is somewhat technical.

Split the output of $P_{\mathsf{pk}}(x) = (P_1(x), P_2(x))$ in half. The signature of a message m is a one-time program that on an input of the form $(x, P_1(x))$, for some x, outputs $(P_2(x), m)$. Consider an adversary $\mathcal{A}$ that wants to forge a signature to deceive a user, lets call them Bob, who possesses $(x_B, P_{\mathsf{pk}}(x_B))$ from the broadcast. Bob would evaluate the forged program on $(x_B, P_1(x_B))$ but there

[2] Note that in the BQSM, schemes rely on two parameters: adversary's memory s and the security parameter λ. The two parameters are not necessarily dependent, but in our schemes they are because the length of transmissions depends on s and correctness requires this to be polynomial in λ.

[3] In essence, a k-time program broadcast of a function P permits a polynomial number of users to each obtain a single evaluation of P on their chosen inputs while simultaneously preventing any single user from learning more than k evaluations of P.

[4] Informally, a one-time program can be used to learn a single evaluation of a program without revealing any additional information about the program.

is a very low probability the output will contain $P_2(x_B)$ as $\mathcal{A}$ is unlikely to have learned $P_{\mathsf{pk}}(x_B)$ from the broadcast.

The problem is that $\mathcal{A}$ can take a valid signature, edit it, and forward it to Bob. Even though $\mathcal{A}$ is limited with respect to its qmemory and cannot store signatures, this attack can be performed on the fly. One-time programs are potent tools due to their capacity to conceal information, but this attribute also renders it challenging for Bob to detect if a received signature has been tampered with. Exploiting this vulnerability, an adversary can modify a signature of m to sign a new message m'. Unfortunately, Barhoush and Salvail's work [11] did not resolve this issue, which consequently restricts their security to scenarios where the adversary is not allowed to query the signature oracle when it attempts to forge a signature and can only query it prior. Practically speaking, there is no way to enforce such a requirement.

Another issue with their construction is that the public-keys are quantum and yet are not certified. Their work assumes the keys can be distributed authentically to each and every user. Note that this issue cannot be solved by applying TD signatures to sign the keys because TD signatures can only be verified after some delay and users in the BQSM cannot store the quantum signatures. To sum up, there are 3 issues with their construction which we address in this work:

1. Can only be used to sign classical messages.
2. Security is only ensured against adversaries with restricted access to the signing oracle.
3. Assumes the secure distribution of quantum public-keys.

We developed a solution that resolves these issues and achieves quantum unforgeability with certified quantum public-keys against adversaries with unrestricted access to the signing oracle. Our construction is quite technical so we refer the reader to Sec. 6 for more details.

Application: Public-Key Quantum Money. We present two public-key quantum money schemes which are derived as applications of TD and TD-DVK signature. In particular, the banknote generation and verification algorithms depend on time so we term our schemes *TD public-key quantum money*. The first scheme we describe, derived from TD-DVK signatures, accomplishes semi-online public-key quantum money assuming pq-OWFs. Meanwhile, the second scheme we describe, derived from TD signatures, achieves offline public-key quantum money assuming pq-OWFs and TLPs.

We now provide a simplified overview of our first scheme. We let $(\mathsf{1QEnc}, \mathsf{1QDec})$ denote a one-time QAE scheme and let F denote a pq-PRF. For any $i \in \mathbb{N}$, the bank generates the key $\mathsf{k}_i \coloneqq F(\mathsf{sk}, i)$ using a secret key sk and the pq-PRF. To create a quantum banknote at time in $[t_j, t_{j+1})$, the bank samples a random string y, and generates a set of n keys $y_i \coloneqq F(\mathsf{k}_i, y)$ for $i \in [j, j+n]$, where n is a parameter that can be set by the bank depending on an efficiency/lifespan tradeoff. Subsequently, the bank prepares the quantum state $|0\rangle$ and, for each key $(y_i)_{i\in[j:j+n]}$, encrypts the state successively using the one-time encryption scheme. The resulting banknote is of the form $|\$\rangle \coloneqq (y, \mathsf{1QEnc}_{y_j}(\mathsf{1QEnc}_{y_{j+1}}(...\mathsf{1QEnc}_{y_{j+n}}(|0\rangle)...)$.

Assume Alice receives the banknote at a time in $[t_j, t_{j+1})$. At time t_{j+1}, the bank publicly announces k_j. Alice computes $y_j = F(\mathsf{k}_j, y)$ and validates the banknote by applying $\mathsf{1QDec}_{y_j}$. More generally, any user can validate the banknote as long as it is received before time t_{j+n} by performing verification with the appropriate key. However, after time t_{j+n}, all the keys used in generating $|\$\rangle$ would have been revealed rendering the banknote unverifiable. Hence, it is imperative to return the banknote to the bank before this expiration.

We term our scheme "semi-online" as users do not need to interact with a bank as in online schemes but they must receive key announcements to validate transactions which prevents complete isolation as in the offline setting. Importantly, our cash system is designed to be scalable and applicable on a macro-level, as it does not rely on a third party to validate each and every transaction as in private-key and online schemes. Nevertheless, a scheme allowing users to operate in a completely offline manner provides further advantages.

Fortunately, our second approach achieves this goal. However, security is only guaranteed assuming TLPs and pq-OWFs. The construction is very similar to the one just described, except the keys $y_j, y_{j+1}, ..., y_{j+n}$ are encoded in a TLP, allowing users to deduce y_i only at time t_i. Here, the TLP effectively performs the same function as the dynamic verification keys. The puzzle is signed classically ensuring authenticity with a time-independent verification key. Thus, this approach provides a completely offline public-key scheme.

Our money schemes have two drawbacks: the banknotes expire and the transactions take up to $\delta \coloneqq t_{i+1} - t_i$ time to validate. The first issue can be mitigated by increasing the number of authentication layers in a banknote (choosing a larger value for n). However, this would require users to allocate more quantum memory. The second issue can be addressed by announcing keys more frequently (decreasing δ) but this would shorten the lifespan of the banknotes. Ideally, the bank should find a compromise between these two issues. Note that traditional money also has an expiration date and large transactions, even with cryptocurrencies, need to be validated with proper time-consuming procedures. Therefore, these issues are inherent in current financial systems.

To sum up, we believe the drawbacks of our construction are a reasonable compromise to achieve a publicly-verifiable currency that resists counterfeiting from standard assumptions. To our knowledge, this is the first work to achieve this challenging goal.

Application: Encryption with Authenticated Public Keys. Our final contribution is to present a QPKE scheme that incorporates publicly verifiable quantum public-keys. Traditionally, QPKE schemes require that the quantum keys are distributed securely to all user [11, 29, 22], making them susceptible to attacks that interfere with the key distribution process. We introduce a security notion that we term *indistinguishability under adaptive chosen quantum key and ciphertext attack* (IND-qCKCA2). In this setting, an adversary can tamper with the quantum public-keys, but it is assumed that classical certification keys can be distributed authentically. More explicitly, the adversary can *choose* the quantum key used by the experiment for encryption but cannot tamper with the classical

certification keys, reflecting a realistic threat scenario where classical information can be reliably announced through public-key infrastructure, but quantum information cannot. Additionally, the adversary has quantum access to the encryption and decryption oracles (except on the challenge ciphertext) as in the standard IND-qCCA2 experiment [18, 6].

To achieve qCKCA2-security, we essentially sign the public-keys using TD signatures. Surprisingly, the adversary may still be able to mount quantum attacks by storing a state entangled with a user's public-key copy. This stored state can be exploited to extract information on the user's encryptions. To deal with this attack, we encode quantum keys in BB84 states that satisfy an unclonability property known as the monogamy of entanglement [49], preventing such splitting attacks.

We apply TD-DVK signatures to the public-keys in two ways to give two constructions. The first construction relies on many-time TD-DVK signatures based on pq-OWFs. Since the verification key evolves with time, the resulting signed keys also change and are thus mixed-state. Note that other works also utilize mixed-state quantum keys [11, 37]. We apply this approach to a QPKE scheme that is inspired by the construction given in [29] to get the first qCKCA2-secure QPKE scheme from pq-OWFs.

Next, by constraining the distribution of the quantum keys to some limited time frame, we show how to upgrade a QPKE scheme to generate publicly verifiable pure-state quantum keys without requiring any additional computational assumptions. The idea is to encrypt all pure-state keys with a one-time authenticated symmetric key encryption [12]. Then, after all users have received their encrypted key copies, the secret encryption key is revealed, enabling users to validate their keys. After the key is revealed, the distribution terminates, thus characterizing it as a limited-time key distribution. As a direct application to the scheme in [29], we achieve QPKE satisfying indistinguishability under chosen key and plaintext attack (IND-CKPA) with limited-time pure-state public-keys from pseudorandom function states (PRFS) with super-logarithmic input size. This is noteworthy because PRFS is potentially a weaker assumption than pq-OWF. More specifically, pq-OWFs imply PRFSs with super-logarithmic input-size [10] and there exists an oracle separation between the two assumptions [39]. See Sec. 8 for more details.

3 Preliminaries

3.1 Notation

We denote Hilbert spaces by calligraphic letters, such as $\mathcal{H}$, and the set of positive semi-definite operators in $\mathcal{H}$ as $\mathrm{Pos}(\mathcal{H})$. If a Hilbert space $\mathcal{X}$ holds a classical value, then we denote this variable as X.

The density matrix of a quantum state in a register E is denoted as ρ_E and, generally, lowercase Greek letters such as σ, ϕ, τ denote density matrices of quantum states. The trace norm is denoted as $\|\rho\| \coloneqq \mathrm{tr}(\sqrt{\rho\rho^\dagger})$, the trace

distance as $\delta(\rho,\sigma) \coloneqq \frac{1}{2}\|\rho-\sigma\|$, and the fidelity by δ_F. We say two states in registers A and B, respectively, are ϵ-*close* denoted as $A \approx_\epsilon B$ if $\delta(\rho_A,\rho_B) \leq \epsilon$ and are ϵ-*independent* if $\rho_{AB} \approx_\epsilon \rho_A \otimes \rho_B$. Note that two states are negligibly close if and only if their fidelity is negligibly close to 1 [26]. We let $\langle \phi, \rho \rangle$ denote a sequential transmission where ϕ is first sent, and then ρ is sent after.

For the complex space $\mathbb{C}^2$, the computational basis $\{|0\rangle, |1\rangle\}$ is denoted as $+$, while the diagonal basis $\{|0\rangle_\times, |1\rangle_\times\}$ is denoted as $\times$ where $|0\rangle_\times \coloneqq \frac{|0\rangle+|1\rangle}{\sqrt{2}}$ and $|1\rangle_\times \coloneqq \frac{|0\rangle-|1\rangle}{\sqrt{2}}$. The pair of basis $\{+,\times\}$ is often called the BB84 basis [14].

We say $x \leftarrow X$ if x is chosen from the values in X according to the distribution X. If X is a set, then $x \leftarrow X$ denotes an element chosen uniformly at random from the set. We say A is QPT if it is a quantum polynomial-time algorithm. We let A^{qP} denote an algorithm with access to q queries to an oracle of P, A^P denote access to a polynomial number of oracle queries, and $A^{|P\rangle}$ denote access to a polynomial number of quantum oracle queries. For any probabilistic algorithm $G(x)$ which makes p coin tosses, we let $G(x;y)$ be the deterministic version of G where the coin toss outcomes are determined by $y \in \{0,1\}^p$.

Also, we let $[n] \coloneqq [0,1,...,n-1]$ and negl denote any function that is asymptotically smaller than the inverse of any polynomial. For simplicity, we let $\{0,1\}^{m,n}$ denote the set $\{0,1\}^m \times \{0,1\}^n$. By abuse of notation, given a string x and matrix M, then, for simplicity, we let $M \cdot x$ denote the string representation of the vector $M \cdot \boldsymbol{x}$.

3.2 Algebra

The following algebraic result from [11] states that it is difficult to predict the output of a large matrix given a few evaluations.

Lemma 1 ([11]). *Let M be an arbitrary $\ell \times n$ binary matrix. Let A be an algorithm that is given as input: $(a_1,b_1),...,(a_m,b_m)$ and $(\hat{a}_1,\hat{b}_1),...,(\hat{a}_p,\hat{b}_p)$ where $a_i,\hat{a}_i \in \{0,1\}^n$, $b_i,\hat{b}_i \in \{0,1\}^\ell$, $m<n$, p is a polynomial in ℓ, $b_i = M \cdot a_i$ and $\hat{b}_i \neq M \cdot \hat{a}_i$. Then the following statements hold:*

1. *For any vector $a \in \{0,1\}^n$ not in the span of $\{a_1,...,a_m\}$, if A outputs a guess b' of $b \coloneqq M \cdot a$, then $\Pr[b'=b] \leq O(2^{-\ell})$.*
2. *Let $x_0,x_1 \in \{0,1\}^n$ be any two distinct vectors not in the span of $\{a_1,...,a_m\}$. Choose $r \leftarrow \{0,1\}$. If A is additionally given x_0,x_1,y_r where $y_r \coloneqq M \cdot x_r$ and outputs a guess r', then $\Pr[r'=r] \leq \frac{1}{2} + O(2^{-\ell})$.*

3.3 Min-Entropy

We recall the notion of min-entropy of a classical variable conditioned on a quantum state. The reader is referred to [38] for more detailed definitions.

Definition 3 (Conditional Min-Entropy). *Let $\rho_{XB} \in \mathrm{Pos}(\mathcal{H}_X \otimes \mathcal{H}_B)$ be classical on $\mathcal{H}_X$. The min-entropy of ρ_{XB} given $\mathcal{H}_B$ is given by*

$$H_\infty(X|B)_\rho \coloneqq -\lg p_{guess}(X|B)_\rho$$

where $p_{guess}(X|B)_\rho$ is the maximal probability to decode X from B with a POVM on $\mathcal{H}_B$.

Definition 4 (Smooth Min-Entropy). *Let $\epsilon \geq 0$ and $\rho_{XB} \in \mathrm{Pos}(\mathcal{H}_X \otimes \mathcal{H}_B)$ be classical on $\mathcal{H}_X$. The ϵ-smooth min-entropy of ρ_{XB} given $\mathcal{H}_B$ is given as*

$$H_\infty^\epsilon(X|B)_\rho := \sup_{\overline{\rho}} H_\infty(X|B)_{\overline{\rho}}$$

where the supremum is taken over all density operators $\overline{\rho}_{XB}$ acting on $\mathcal{H}_X \otimes \mathcal{H}_B$ such that $\delta(\overline{\rho}_{XB}, \rho_{XB}) \leq \epsilon$.

The min-entropy satisfies the following chain rule.

Lemma 2 (Chain Rule for Min-Entropy [44]). *Let $\epsilon \geq 0$ and $\rho_{XUW} \in \mathrm{Pos}(\mathcal{H}_X \otimes \mathcal{H}_U \otimes \mathcal{H}_W)$ where register W has size n. Then,*

$$H_\infty^\epsilon(X|UW)_\rho \geq H_\infty^\epsilon(XE|W)_\rho - n.$$

3.4 Privacy Amplification

We state the standard approach to amplify privacy through the use of hash functions.

Definition 5. *A class of hash functions $\mathsf{H}_{m,\ell}$ from $\{0,1\}^m$ to $\{0,1\}^\ell$ is* two-universal *if for $F \leftarrow \mathsf{H}_{m,\ell}$,*

$$Pr[F(x) = F(x')] \leq \frac{1}{2^\ell}.$$

Theorem 1 (Privacy Amplification [44]). *Let $\epsilon \geq 0$ and $\rho_{XB} \in \mathrm{Pos}(\mathcal{H}_X \otimes \mathcal{H}_B)$ be a cq-state where X is classical and takes values in $\{0,1\}^m$. Let $\mathsf{H}_{m,\ell}$ be a class of two-universal hash functions and let $F \leftarrow \mathsf{H}_{m,\ell}$. Then,*

$$\delta(\rho_{F(X)FB}, \mathbb{1} \otimes \rho_{FB}) \leq \frac{1}{2} 2^{-\frac{1}{2}(H_\infty^\epsilon(X|B)_\rho - \ell)} + \epsilon$$

where $\mathbb{1}$ is uniformly distributed.

3.5 Monogamy of Entanglement

We recall the monogamy of entanglement game which provides a way to test the unclonability of quantum states. This will be useful in proving security of our QPKE scheme. There are different variants of this game but we recall a version similar to the one presented in [49].

Experiment $\mathsf{G}_{\mathsf{BB84}}(n)$:

1. Alice chooses $x \leftarrow \{0,1\}^n$ and $\theta \leftarrow \{+,\times\}^n$.
2. Alice sends the state $|x\rangle_\theta := |x_0\rangle_{\theta_0}|x_1\rangle_{\theta_1}...|x_{n-1}\rangle_{\theta_{n-1}}$ to Bob and Charlie.
3. Bob and Charlie split the state between them. From this point on they are not allowed to communicate.
4. Alice sends θ to Bob and Charlie.
5. Bob and Charlie output x_B and x_C, respectively.
6. The output of the experiment is 1 if $x = x_B = x_C$ and 0 otherwise.

Theorem 2 (Monogamy of Entanglement, Theorem 3 in [49]). *For any quantum algorithms Bob and Charlie,*

$$\Pr\left[\mathsf{G}_{\mathsf{BB84}}(n) = 1\right] \leq 2^{-n/5}.$$

3.6 Pseudorandom Notions

In this section, we recall the definitions of post-quantum pseudorandom functions (pq-PRFs) and quantum-accessible pseudorandom function state generators (PRFSs).

Definition 6 (Post-Quantum Secure Pseudorandom Function). *Let $PRF = \{PRF_n\}_{n\in\mathbb{N}}$ where $PRF_n : \mathcal{K}_n \times \mathcal{X}_n \to \mathcal{Y}_n$ is an efficiently computable function. Then, PRF is a post-quantum secure pseudorandom function if for any QPT algorithm $\mathcal{A}$ there exists negligible function negl such that for any $n \in \mathbb{N}$:*

$$|\Pr_{k\leftarrow\mathcal{K}_n}[\mathcal{A}^{PRF_n(k,\cdot)}(1^n) = 1] - \Pr_{O\leftarrow\mathcal{Y}_n^{\mathcal{X}_n}}[\mathcal{A}^{O}(1^n) = 1]| \leq \mathsf{negl}(n).$$

where $\mathcal{Y}_n^{\mathcal{X}_n}$ is a function assigning each value in $\mathcal{X}_n$ a random value in $\mathcal{Y}_n$.

Next, PRFS were first introduced in [10] and the follow-up work [7] provided a stronger definition that allowed the adversary quantum access to the oracle. We now recall their definition here:

Definition 7 (Quantum-Accessible PRFS Generator). *A QPT algorithm G is a quantum-accessible secure pseudorandom function state generator if for all QPT (non-uniform) distinguishers $\mathcal{A}$ if there exists negligible function negl such that for any $n \in \mathbb{N}$:*

$$|\Pr_{k\leftarrow\{0,1\}^\lambda}[\mathcal{A}^{|O_{PRF}(k,\cdot)\rangle}(1^n) = 1] - \Pr[\mathcal{A}^{|O_{Haar}(\cdot)\rangle}(1^n) = 1]| \leq \mathsf{negl}(n).$$

where,

1. *$O_{PRF}(\mathsf{k},\cdot)$, on input a d-qubit register X does as follows: applies a channel that controlled on register X containing x, it creates and stores $G_n(\mathsf{k},x)$ on a new register Y and outputs the state on register X and Y.*
2. *$O_{Haar}(\cdot)$, modeled as a channel, on input a d-qubit register X, does as follows: applies a channel that, controlled on the register X containing x, stores $|\theta_x\rangle\langle\theta_x|$ in a new register Y where $|\theta\rangle_x$ is sampled from the Haar distribution and outputs the state on register X and Y.*

3.7 Signatures on Classical Messages

We recall the definition of a digital signature scheme on classical messages.

Definition 8 (Digital Signatures). *A* digital signature *scheme over classical message space $\mathcal{M}$ consists of the following QPT algorithms:*

- $\mathit{Gen}(1^\lambda)$*: Outputs a secret key* sk *and a verification key* vk*.*
- $\mathit{Sign}(\mathsf{sk}, \mu)$ *: Outputs a signature* σ *for* $\mu \in \mathcal{M}$ *using* sk*.*
- $\mathit{Verify}(\mathsf{vk}, \mu', \sigma')$*: Verifies whether* σ' *is a valid signature for* $\mu' \in \mathcal{M}$ *using* vk *and correspondingly outputs* $\top / \bot$*.*

Definition 9 (Correctness). *A digital signature scheme is* correct *if for any message* $\mu \in \mathcal{M}$*:*

$$\Pr\left[\mathit{Verify}(\mathsf{vk}, \mu, \sigma) = \top \;\middle|\; \begin{matrix}(\mathsf{sk}, \mathsf{vk}) & \leftarrow \mathit{Gen}(1^\lambda) \\ \sigma & \leftarrow \mathit{Sign}(\mathsf{sk}, \mu)\end{matrix}\right] \geq 1 - \mathsf{negl}(\lambda).$$

We recall the security experiment testing *existential unforgeability* (UF) of a digital signature scheme.

Experiment $\mathsf{Sign}^{\mathrm{UF}}_{\Pi,\mathcal{A}}(\lambda)$:

1. Sample $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{Gen}(1^\lambda)$.
2. $\mathcal{A}$ is given vk and classical access to the signing oracle $\mathsf{Sign}(\mathsf{sk}, \cdot)$. Let $\mathcal{Q}$ denote the set of messages queried.
3. $\mathcal{A}$ outputs (m^*, σ^*).
4. The output of the experiment is 1 if

$$m^* \notin \mathcal{Q} \text{ and } \mathsf{Verify}(\mathsf{vk}, m^*, \sigma^*) = 1.$$

5. Otherwise, the output is 0.

Definition 10. *A digital signature scheme Π is* existentially unforgeable (UF)*, if for any QPT $\mathcal{A}$,*

$$\Pr[\mathit{Sign}^{UF}_{\Pi,\mathcal{A}}(\lambda) = 1] \leq \mathsf{negl}(\lambda).$$

3.8 Cryptography from OWF

In this section, we state relevant results obtained from pq-OWFs. Firstly, Zhandry [52] showed how to construct pq-PRF from pq-OWF. Next, [19] used pq-PRFs to construct an IND-qCPA-secure symmetric encryption and this was later upgraded to IND-qCCA2 [20]. Informally, these security notions are the same as standard IND-CPA and IND-CCA2, except the adversary in the security experiments is given *quantum* access to the oracles (see [19, 20] for definitions).

Lemma 3 (Symmetric Encryption [19, 20]). *Assuming pq-PRFs, there exists an IND-qCCA2-secure symmetric encryption scheme on classical messages with classical ciphertexts.*

We will also need a way to authenticate and encrypt *quantum* messages in a symmetric-key setting. To this end, Alagic, Gagliardoni, and Majenz [5] provided an approach to test *quantum unforgeability* (QUF) and *quantum indistinguishability under adaptive chosen ciphertext attack* (QIND-CCA2) for an authenticated symmetric encryption scheme on quantum messages. Furthermore, they provided a scheme satisfying these notions under pq-PRFs. These experiments are quite technical and involved so we refer the reader to [5] for a detailed description. In Sections 5.1 & 6.1, we present unforgeability experiments adapted to our purposes.

Lemma 4 (Symmetric Authenticated Encryption (Theorem 5 in [5])). *Assuming pq-PRFs, there exists a symmetric key encryption scheme on quantum messages that is QUF and QIND-CCA2.*

Also, it is well-known that pq-OWFs imply digital signatures on classical messages.

Lemma 5 (Digital Signatures [47]). *Digital signatures on classical messages exist assuming pq-OWFs.*

Note also that an information-theoretically secure *one-time* QAE exists unconditionally.

Lemma 6 (One-Time Symmetric Authenticated Encryption [12]). *There exists an information-theoretic unforgeable one-time QAE scheme on quantum messages. In the scheme, if λ denotes the security parameter, then a n-qubit message is encrypted with a $(n + 2\lambda)$-bit key and the resulting ciphertext is a $(n + \lambda)$-qubit state.*

3.9 Cryptography in the BQSM

In this section, we review major results in the BQSM based on [11]. These results will be useful in constructing the BQS signature scheme.

We first recall the definition of a BQS one-time program.

Definition 11 (BQS One-Time Program [11]). *An algorithm O is a (q, s)-*BQS one-time compiler *for the class of classical circuits $\mathcal{F}$ if it is QPT and satisfies the following:*

1. *(functionality) For any circuit $C \in \mathcal{F}$, the circuit described by $O(C)$ can be used to compute C on a single input x chosen by the evaluator.*
2. *For any circuit $C \in \mathcal{F}$, the receiver requires q qmemory to evaluate $O(C)$.*
3. *(security) For any computationally unbounded adversary $\mathcal{A}_s$ there exists a computationally unbounded simulator $\mathcal{S}_s$ such that for any circuit $C \in \mathcal{F}$:*

$$|\Pr[\mathcal{A}_s(O(C)) = 1] - \Pr[\mathcal{S}_s^{1C}(|0\rangle^{\otimes|C|}) = 1]| \leq \mathsf{negl}(|C|).$$

A one-time compiler, denoted as $\mathcal{O}_{\mathsf{s}}$, was constructed unconditionally in the BQSM [11]. The details of the construction are not necessary for this work so only the result is given.

Theorem 3 ([11]). *$\mathcal{O}_s$ is a disappearing and unclonable information-theoretically secure $(0, s)$ BQS one-time compiler for the class of polynomial classical circuits.*

Here, disappearing means the one-time program can no longer be evaluated after the transmission ends. Unclonability means the one-time program cannot be split into two pieces that can both be used to learn an evaluation. The reader is referred to [11] for more detailed definitions.

A similar, more powerful, notion known as BQS program broadcast was also introduced in [11].

Definition 12 (BQS Program Broadcast). *A (q, s, k)-BQS program broadcast for the class of circuits $\mathcal{C}$ consists of the following QPT algorithms:*

1. $\mathsf{KeyGen}(1^\lambda, t_{\text{end}})$: *Outputs a classical key ek.*
2. $\mathsf{Br}(1^s, \mathsf{ek}, C)$: *Outputs a quantum transmission O_C for the circuit $C \in \mathcal{C}$ during broadcast time (before t_{end}). Outputs ek after broadcast time.*
3. $\mathsf{Eval}(\langle O_C, \mathsf{ek}\rangle, x)$: *Outputs an evaluation y on input x from the transmission $\langle O_C, \mathsf{ek}\rangle$ using q qmemory.*

Correctness requires that for any circuit $C \in \mathcal{C}$ and input x,

$$\Pr\left[\mathsf{Eval}(\langle O_C, \mathsf{ek}\rangle, x) = C(x) \;\middle|\; \begin{matrix}\mathsf{ek} \leftarrow \mathsf{KeyGen}(1^\lambda, t_{\text{end}}) \\ O_C \leftarrow \mathsf{Br}(1^s, \mathsf{ek}, C)\end{matrix}\right] \geq 1 - \mathsf{negl}(\lambda).$$

Security requires that for any (computationally unbounded) adversary $\mathcal{A}_s$, there exists a (computationally unbounded) simulator $\mathcal{S}_s$ such that for any circuit $C \in \mathcal{C}$, and $\mathsf{ek} \leftarrow \mathsf{KeyGen}(1^\lambda, t_{\text{end}})$,

$$|Pr[\mathcal{A}_s^{\mathsf{Br}(1^s,\mathsf{ek},C)}(|0\rangle) = 1] - Pr[\mathcal{S}_s^{kC}(|0\rangle^{\otimes|C|}) = 1]| \leq \mathsf{negl}(\lambda).$$

A program broadcaster, denoted Π_{Br}, was also constructed unconditionally in the BQSM [11]. Let $\mathcal{C}_m$ be the class of polynomial classical circuits with m-bit outputs.

Theorem 4 ([11]). *Π_{Br} is an information-theoretically secure $(12m, \mathsf{s}, \frac{s}{2m})$-BQS program broadcast for the class $\mathcal{C}_m$ as long as 2^{-m} is negligible with respect to the security parameter.*

3.10 QROM and TLP

In the *random oracle model* [13], a hash function H is assumed to return truly random values, and all parties are given access to an oracle of H. This model aims to achieve security using the randomness of H. In practice, H is replaced with a suitable hash function. However, a quantum adversary can evaluate a hash

function on a superposition of inputs. This motivated the introduction of the *quantum random oracle model* (QROM) [17], where users can query the oracle on a superposition of inputs. The reader is referred to [17] for more details.

The important point for our purposes is that the QROM allows for the creation of time-lock puzzles which were first introduced in [45]. We briefly define this notion based on [16] and the reader is referred to this work for a detailed definition.

Definition 13 (Time-Lock Puzzle [16]). *A TLP with gap $\epsilon < 1$ consists of the following pair of algorithms:*

1. $\mathsf{Puzzle.Gen}(1^\lambda, t, s)$: *Generates a classical puzzle Z encrypting the solution $s \in \{0,1\}^\lambda$ with difficulty parameter t.*
2. $\mathsf{Puzzle.Sol}(Z)$: *Outputs a solution $s \in \{0,1\}^\lambda$ for the puzzle Z.*

Correctness requires that for any solution s and puzzle $Z \leftarrow \mathsf{Puzzle.Gen}(1^\lambda, t, s)$, $\mathsf{Puzzle.Sol}(Z)$ outputs s with probability $1 - \mathsf{negl}(\lambda)$.

Security requires that there exists a polynomial $\tilde{t}$ such that for every polynomial $t \geq \tilde{t}$ and every QPT $\mathcal{A}$ of depth $\mathsf{dep}(\mathcal{A}) \leq t^\epsilon(\lambda)$, there exists negligible function μ such that for every $\lambda \in \mathbb{N}^+$ and pair of distinct solutions $s_0, s_1 \in \{0,1\}^\lambda$:

$$\Pr\left[\mathcal{A}(Z_b) = b \;\middle|\; \begin{matrix} b \leftarrow \{0,1\} \\ Z_b \leftarrow \mathsf{Puzzle.Gen}(1^\lambda, t, s_b) \end{matrix}\right] \leq \frac{1}{2} + \mu(\lambda).$$

The security condition essentially states that if the puzzle is constructed with difficulty parameter t, then the puzzle cannot be deciphered in depth t^ϵ.

TLPs were constructed classically from non-parallelizing languages and OWFs [16]. The same construction can be adapted to the post-quantum setting, as noted in [40], assuming post-quantum non-parallelizing languages and pq-OWFs. Post-quantum non-parallelizing languages were constructed in the QROM in [21] giving the following result.

Lemma 7 ([16, 21]). *Let $\epsilon < 1$. TLPs with gap ϵ exist assuming pq-OWFs in the QROM.*

For the rest of the work, we use TLPs with polynomial $\tilde{t} = 0$ in order to simplify notation. Also, we informally say "a puzzle Z requires t time to solve" if $Z \leftarrow \mathsf{Puzzle.Gen}(1^\lambda, t, s)$ is constructed with difficulty parameter t.

4 Time-Dependent Signatures

In this section, we introduce the notion of TD signatures for quantum messages and present a construction that is secure in the QROM assuming the existence of pq-OWFs.

4.1 Definitions

We introduce the notion of TD signatures with verification keys that do not change with time. In this setting, the current time is denoted by $\mathtt{t}$ and it is assumed that all parties have access to the current time up to a small error.

Definition 14 (Time-Dependent Signatures). *A* TD signature scheme Π *over quantum message space* $\mathcal{Q}$ *consists of the following algorithms:*

- $\mathsf{KeyGen}(1^\lambda)$*: Outputs* $(\mathsf{sk}, \mathsf{vk})$*, where* sk *is a secret key and* vk *is a verification key.*
- $\mathsf{Sign}(\mathsf{sk}, \mathtt{t}, \phi)$ *: Outputs a signature* σ *for* $\phi \in \mathcal{Q}$ *based on the time* $\mathtt{t}$ *using* sk*.*
- $\mathsf{Verify}(\mathsf{vk}, \mathtt{t}, \sigma')$*: Verifies the signature* σ' *using* vk *and time* $\mathtt{t}$*. Correspondingly outputs a message* ϕ' *or* $\perp$*.*

Definition 15 (Correctness). *A TD signature scheme is* correct *if: for any times* T *and* $T' \leq T + \delta/2$ *and* $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$*:*

$$\| \mathsf{Verify}(\mathsf{vk}, T', \mathsf{Sign}(\mathsf{sk}, T, \cdot)) - \mathit{id}(\cdot) \| \leq \mathsf{negl}(\lambda).$$

We provide a security notion for quantum unforgeability (QUF) that is similar to the security definition for a one-time quantum message authentication introduced in [12]. Essentially, we want the forgery submitted by the adversary to be either invalid or the message extracted from the forgery to be the "same as" a state submitted to the signature oracle. By "same as" we require that the fidelity between the two states is high (close to 1).

Experiment $\mathsf{TD\text{-}Sign}^{\mathrm{QUF}}_{\Pi,\mathcal{A}}(\lambda)$:

1. Experiment $\mathcal{E}$ samples $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$.
2. Adversary outputs $\tilde{\sigma} \leftarrow \mathcal{A}^{|\mathsf{Sign}(\mathsf{sk},\mathtt{t},\cdot)\rangle}(\mathsf{vk})$ at some time T. Let $\phi_1, ..., \phi_p$ denote the density matrices of the quantum messages $\mathcal{A}$ submitted to the signature oracle (may be entangled).
3. $\mathcal{E}$ computes $\mathsf{Verify}(\mathsf{vk}, T, \tilde{\sigma})$ and obtains either $\perp$ or a state with density matrix $\tilde{\phi}$.
4. If the outcome is $\perp$, then the experiment is terminated and the output is 0.
5. Otherwise, let $F \coloneqq \max_{i \in [p]} \delta_F(\phi_i, \tilde{\phi})$ be the maximum fidelity.
6. $\mathcal{E}$ flips a biased coin that "produces" head with probability F and tails with probability $1 - F$.
7. The output of the experiment is 1 if the coin toss outcome is tails. Otherwise, the output is 0.

Security requires that there is a negligible chance that the coin toss outcome is tails. This translates to the requirement that there is a negligible chance that the state $\tilde{\phi}$ is not very close to one of the submitted states $\phi_1, ..., \phi_p$ with respect to the fidelity measure.

Definition 16. *A TD quantum signature scheme Π is* quantum unforgeable (QUF)*, if for any QPT $\mathcal{A}$,*

$$\Pr[\mathit{TD\text{-}Sign}^{QUF}_{\Pi,\mathcal{A}}(\lambda) = 1] \leq \mathsf{negl}(\lambda).$$

Our TD signatures also satisfy a "disappearing" or "expiration" property. Advantageously, this property implies that an adversary cannot produce a signature for a message ϕ even if it has received a signature of ϕ earlier! We present the following experiment to capture this intuition.

Experiment $\mathsf{TD\text{-}Sign}^{\mathsf{Dis}}_{\Pi,\mathcal{A}}(\lambda)$:

1. Experiment $\mathcal{E}$ samples $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$.
2. Adversary gets vk and quantum oracle access to $\mathsf{Sign}(\mathsf{sk}, \mathsf{t}, \cdot)$.
3. $\mathcal{A}$ outputs 1 at some time T and its query access is revoked.
4. $\mathcal{A}$ outputs $\tilde{\sigma}$.
5. $\mathcal{E}$ computes $\mathsf{Verify}(\mathsf{k}, T+1, \tilde{\sigma})$.
6. The output of the experiment is 0 if the result is $\perp$ and 1 otherwise.

Definition 17. *A TD quantum signature scheme Π is* disappearing *if for any QPT $\mathcal{A}$,*

$$\Pr[\mathit{TD\text{-}Sign}^{\mathit{Dis}}_{\Pi,\mathcal{A}}(\lambda) = 1] \leq \mathsf{negl}(\lambda).$$

4.2 Construction

We present a scheme for TD signatures that satisfies disappearing and QUF security assuming TLPs and digital signatures on classical messages. Recall, that both these assumptions can be based on pq-OWFs in the QROM (Lemmas 5 & 7).

Construction 1. *Let $\Pi_{1\mathrm{QAE}} := (\mathit{1QGen}, \mathit{1QEnc}, \mathit{1QDec})$ be the algorithms of a one-time QAE scheme on n-qubit messages. Let $\Pi_{\mathrm{CS}} := (\mathit{CS.Gen}, \mathit{CS.Sign}, \mathit{CS.Verify})$ be the algorithms for a digital signature scheme on classical messages. Let $(\mathit{Puzzle.Gen}, \mathit{Puzzle.Sol})$ be the algorithms of a TLP scheme. The construction for a TD signature scheme on n-qubit messages is as follows:*

- $\mathit{KeyGen}(1^\lambda)$: *Sample* $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathit{CS.Gen}(1^\lambda)$.
- $\mathit{Sign}(\mathsf{sk}, t, \phi)$:
  1. *Sample* $k \leftarrow \mathit{1QGen}(1^\lambda)$.
  2. *Compute* $\rho \leftarrow \mathit{1QEnc}_k(\phi)$.
  3. *Classically sign* (t, k) *i.e.* $\mathit{sig} \leftarrow \mathit{CS.Sign}(\mathsf{sk}, (t, k))$.
  4. *Generate TLP puzzle* $Z \leftarrow \mathit{Puzzle.Gen}(1^\lambda, 1, (t, k, \mathit{sig}))$.

  Output $\sigma := (\rho, Z)$.
- $\mathit{Verify}(\mathsf{vk}, t, \sigma')$:
  1. *Interpret* σ' *as* (ρ', Z') *and take note of the current time as* T.
  2. *Compute* $(T', k', \mathit{sig}') \leftarrow \mathit{Puzzle.Sol}(Z')$.
  3. *If* $T' + \delta/2 \leq T$ *or* $\mathit{CS.Verify}(\mathsf{vk}, (T', k'), \mathit{sig}') = 0$*, then output* $\perp$.

4. Otherwise, output $\mathsf{1QDec}_{k'}(\rho')$.

Theorem 5. *Construction 1 is a QUF and disappearing* TD *signature scheme assuming the existence of secure* TLPs *and EUF digital signatures on classical messages.*

Proof. We first prove QUF security through a sequence of hybrids. We let hybrid H_0 be the standard experiment, but adapted to our construction for an easier discussion.

Hybrid H_0 (standard experiment):

1. $\mathcal{E}$ samples $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$.
2. Adversary outputs $\tilde{\sigma} \leftarrow \mathcal{A}^{|\mathsf{Sign}(\mathsf{sk},\mathsf{t},\cdot)\rangle}(\mathsf{vk})$ at time T.
3. Let $\phi_1, ..., \phi_p$ denote the density matrices of the quantum messages $\mathcal{A}$ submitted to the signature oracle and let $(\rho_i, Z_i)_{i\in[p]}$ denote the responses.
4. $\mathcal{E}$ verifies the forgery as follows:
   (a) Interpret $\tilde{\sigma}$ as $(\tilde{\rho}, \tilde{Z})$.
   (b) Compute $(\tilde{T}, \tilde{\mathsf{k}}, \tilde{\mathsf{sig}}) \leftarrow \mathsf{Puzzle.Sol}(\tilde{Z})$.
   (c) If $\tilde{T} + \delta/2 \leq T$ or $\mathsf{CS.Verify}(\mathsf{vk}, (\tilde{T}, \tilde{\mathsf{k}}), \tilde{\mathsf{sig}}) = \perp$, then output $\perp$.
   (d) Otherwise, compute $\tilde{\phi} \leftarrow \mathsf{1QDec}_{\tilde{\mathsf{k}}}(\tilde{\rho})$.
5. Let $F := \max_{i\in[p]} \delta_F(\phi_i, \tilde{\phi})$ be the maximum fidelity.
6. $\mathcal{E}$ flips a biased coin that "produces" head with probability F and tails with probability $1 - F$.
7. The output of the experiment is 1 if the coin toss outcome is tails. Otherwise, the output is 0.

In the next hybrid, the experiment rejects any forgery which includes a signature of a new message under the classical signature scheme.

Hybrid H_1:

1. Steps 1-3 same as H_0.
4. $\mathcal{E}$ verifies the forgery as follows:
   (a) Interpret $\tilde{\sigma}$ as $(\tilde{\rho}, \tilde{Z})$.
   (b) Compute $(\tilde{T}, \tilde{\mathsf{k}}, \tilde{\mathsf{sig}}) \leftarrow \mathsf{Puzzle.Sol}(\tilde{Z})$.
   (c) Compute $(T_i, \mathsf{k}_i, \mathsf{sig}_i) \leftarrow \mathsf{Puzzle.Sol}(Z_i)$ for all $i \in [p]$.
   (d) If $\tilde{T} + \delta/2 \leq T$ or $(\tilde{T}, \tilde{\mathsf{k}}) \notin (T_i, \mathsf{k}_i)_{i\in[p]}$, then output $\perp$.
   (e) Otherwise, compute $\tilde{\phi} \leftarrow \mathsf{1QDec}_{\tilde{\mathsf{k}}}(\tilde{\rho})$.
5. Steps 5-7 same as H_0.

Hybrid H_2 is the same as hybrid H_1 except we discard the classical signatures. The new approach to sign is as follows:

$\mathsf{Sign}_{\mathsf{H}_2}(\mathsf{sk}, \mathsf{t}, \phi)$:

1. Sample $\mathsf{k} \leftarrow \mathsf{1QGen}(1^\lambda)$.
2. Compute $\rho \leftarrow \mathsf{1QEnc}_{\mathsf{k}}(\phi)$.
3. Generate TLP puzzle $Z \leftarrow \mathsf{Puzzle.Gen}(1^\lambda, 1, (\mathsf{t}, \mathsf{k}))$.

4. Output $\sigma := (\rho, Z)$.

Next, hybrid H_3 is the same as H_2, except we change the signing and verification algorithms again but only for messages close to time T. Specifically, the algorithms are as follows:

- $\mathsf{KeyGen}_{\mathsf{H}_3}(1^\lambda)$: Sample $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{CS.Gen}(1^\lambda)$.
- $\mathsf{Sign}_{\mathsf{H}_3}(\mathsf{sk}, \mathtt{t}, \phi_i)$: If $\mathtt{t} < T - \delta/2$, then output $\mathsf{Sign}_{\mathsf{H}_2}(\mathsf{sk}, \mathtt{t}, \phi_i)$. Otherwise:
  1. Sample $\mathsf{k}_i \leftarrow \mathsf{1QGen}(1^\lambda)$.
  2. Store k_i.
  3. Compute $\rho_i \leftarrow \mathsf{1QEnc}_{\mathsf{k}_i}(\phi_i)$.
  4. Generate TLP puzzle $Z_i \leftarrow \mathsf{Puzzle.Gen}(1^\lambda, 1, (i, 0^{|\mathsf{k}_i|}))$.
  5. Output (ρ_i, Z_i).
- $\mathsf{Verify}_{\mathsf{H}_3}(\mathsf{vk}, \mathtt{t}, \sigma')$:
  1. If $\mathtt{t} < T$, output $\mathsf{Verify}_{\mathsf{H}_2}(\mathsf{vk}, \mathtt{t}, \sigma')$.
  2. Otherwise, interpret $\tilde{\sigma}$ as $(\tilde{\rho}, \tilde{Z})$.
  3. Compute $\mathsf{Puzzle.Sol}(\tilde{Z})$.
  4. If the solution is not of the form $(j, 0^*)$ for some $j \in [p]$, then output $\perp$.
  5. Otherwise, obtain k_j from storage and output $\tilde{\phi} \leftarrow \mathsf{1QDec}_{\mathsf{k}_j}(\tilde{\rho})$.

Claim. $\Pr[\mathsf{H}_3 = 1] \leq \mathsf{negl}(\lambda)$.

Proof. In this hybrid, the experiment verifies the forgery $\tilde{\sigma}$ as follows. It first interprets $\tilde{\sigma}$ as $(\tilde{\rho}, \tilde{Z})$. Then, it computes $\mathsf{Puzzle.Sol}(\tilde{Z})$ and if the solution is not of the form $(j, 0^*)$ for some $j \in [p]$, then it outputs $\perp$. In the case this condition is satisfied, then the experiment runs $\mathsf{1QDec}_{\mathsf{k}_j}(\tilde{\rho})$. Note that the only information available regarding k_j is the authenticated state ρ_j from the j^{th} query to the signature oracle. Hence, if an adversary manages to produce a fresh forgery, then $\mathcal{A}$ can easily be converted into an a QPT adversary that breaks QUF of Π_{1QAE}. ■

Claim. No QPT adversary can distinguish hybrids H_3 and H_2 with non-negligible advantage.

Proof. The only difference between these hybrids is the information contained in the TLPs produced after time $T - \delta/2$. The TLPs require an hour to solve and $\mathcal{A}$ submits its output at time T. Hence, by the security of TLPs, no adversary can distinguish between these hybrids. ■

Claim. The output of H_2 and H_1 are indistinguishable.

Proof. There is no difference between the output of these hybrids as the classical signature scheme used in H_1 is redundant. ■

Claim. No QPT adversary can distinguish hybrids H_1 and H_0 with non-negligible advantage.

Proof. The only difference between these hybrids is that in H_0, the output of the experiment is 0 if $\mathsf{CS.Verify}(\mathsf{vk}, (\tilde{T}, \tilde{\mathsf{k}}), \tilde{\mathsf{sig}}) = \perp$, whereas in H_1, this condition is replaced with $(\tilde{T}, \tilde{\mathsf{k}}) \notin (T_i, \mathsf{k}_i)_{i\in[p]}$. Hence, if $\mathcal{A}$ can distinguish between these hybrids, then with non-negligible probability $\mathcal{A}$'s forgery satisfies $(\tilde{T}, \tilde{\mathsf{k}}) \notin (T_i, \mathsf{k}_i)_{i\in[p]}$ and $\mathsf{CS.Verify}(\mathsf{vk}, (\tilde{T}, \tilde{\mathsf{k}}), \tilde{\mathsf{sig}}) = 1$. Such an algorithm can easily be converted into a QPT algorithm that breaks the unforgeability of Π_{CS}. ■

The past three claims imply that $\Pr[\mathsf{H}_0 = 1] \leq \mathsf{negl}(\lambda)$.

Disappearing security is similar but easier to show. In the disappearing experiment $\mathsf{TD\text{-}Sign}^{\mathsf{Dis}}_{\Pi,\mathcal{A}}(\lambda)$, $\mathcal{A}$ submits the forgery after a time delay of 1 from the submission of any signature query i.e. $T \geq T_i + 1$ for any $i \in [p]$. However, we showed that to pass verification, (whp) there exists $j \in [p]$ such that $\tilde{T} = T_j$ which means $T \geq \tilde{T} + 1$. So (whp) the forgery does not pass verification and the output of the experiment is 0. ■

5 Time-Dependent Signatures with Dynamic Verification Keys

In this section, we first define TD-DVK signatures which are TD signatures but where the verification key varies with time. Then, we provide a construction for this primitive assuming pq-OWFs.

5.1 Definitions

Definition 18 (Time-Dependent Signatures with Dynamic Verification Keys). *A* TD-DVK signature scheme *over quantum message space* $\mathcal{Q}$ *starting at time* t_0 *with interval* δ *consists of the following algorithms:*

- $\mathsf{KeyGen}(1^\lambda)$*: Outputs a time-independent secret key* sk.
- $\mathsf{vkAnnounce}(\mathsf{sk}, \mathsf{t})$ *: If time* $\mathsf{t} = (i+1)\delta + t_0$ *for some* $i \in \mathbb{N}$*, outputs a verification key* vk_i *using the secret key* sk.
- $\mathsf{Sign}(\mathsf{sk}, \mathsf{t}, \phi)$ *: Outputs a signature* σ *for* $\phi \in \mathcal{Q}$ *based on the time* t *using* sk.
- $\mathsf{Verify}(\mathsf{vk}_i, \sigma')$*: Verifies the signature* σ' *using* vk_i *and correspondingly outputs a message* ϕ' *or* $\perp$.

Henceforth, we let $t_i := i\delta + t_0$ and we generally set $\delta = 1$ and $t_0 = 0$ without loss of generality.

Definition 19 (Correctness). *A TD-DVK signature scheme is* correct *if: for any* $i \in \mathbb{N}$*, time* $\mathsf{t} \in [t_i, t_{i+1})$*,* $\mathsf{sk} \leftarrow \mathsf{KeyGen}(1^\lambda)$ *and* $\mathsf{vk}_i \leftarrow \mathsf{vkAnnounce}(\mathsf{sk}, t_{i+1})$ *we have:*

$$\|\mathsf{Verify}(\mathsf{vk}_i, \mathsf{Sign}(\mathsf{sk}, \mathsf{t}, \cdot)) - \mathsf{id}(\cdot)\| \leq \mathsf{negl}(\lambda).$$

Security requires that any forgery submitted is close to a signature produced by the signing oracle in the same time frame as the forgery. Note that we assume that the verification keys vk_i can be distributed authentically. This can be ensured through the use of any digital signature scheme on classical messages under a time-independent verification key.

Experiment $\mathsf{TDDVK\text{-}Sign}^{\mathrm{QUF}}_{\Pi,\mathcal{A}}(\lambda, i)$:

1. Experiment $\mathcal{E}$ samples $\mathsf{sk} \leftarrow \mathsf{KeyGen}(1^\lambda)$.
2. For all $j < i$, $\mathcal{E}$ generates $\mathsf{vk}_j \leftarrow \mathsf{vkAnnounce}(\mathsf{sk}, t_{j+1})$.
3. Adversary outputs $\tilde{\sigma} \leftarrow \mathcal{A}^{|\mathsf{Sign}(\mathsf{sk},\cdot,\cdot)\rangle}((\mathsf{vk}_j)_{j<i})$. Let $(T_1, \phi_1), ..., (T_p, \phi_p)$ denote the subset of messages submitted to the signing oracle which satisfy $t_i \leq T_k < t_{i+1}$ for $k \in [p]$.
4. $\mathcal{E}$ generates $\mathsf{vk}_i \leftarrow \mathsf{vkAnnounce}(\mathsf{sk}, t_{i+1})$.
5. $\mathcal{E}$ computes $\tilde{\phi} \leftarrow \mathsf{Verify}(\mathsf{vk}_i, \tilde{\sigma})$.
6. If the outcome is $\perp$, then the experiment is terminated and the outcome is 0.
7. Otherwise, let $F \coloneqq \max_{i\in[p]} \delta_F(\phi_i, \tilde{\phi})$ be the maximum fidelity.
8. $\mathcal{E}$ flips a biased coin that "produces" head with probability F and tails with probability $1 - F$.
9. The output of the experiment is 1 if the coin toss outcome is tails. Otherwise, the output is 0.

Definition 20. *A TD-DVK quantum signature scheme Π is* quantum unforgeable (QUF)*, if for any QPT $\mathcal{A}$ and $i \in \mathbb{N}$,*

$$\Pr[\mathit{TDDVK\text{-}Sign}^{QUF}_{\Pi,\mathcal{A}}(\lambda, i) = 1] \leq \mathsf{negl}(\lambda).$$

5.2 One-Time TD-DVK Signature Scheme

We first present a very simple information-theoretic TD-DVK one-time signature scheme. The basic idea is to let the verification key be the same as the signing key but with a *delay*. In the next section, we extend it to allow for many signatures using pq-OWFs.

In this scheme, a single message can be signed until a certain time, which we denote t_{end}, chosen by the signer. After this time, the verification key is announced and no new signatures can be securely generated. While such a construction seems trivial, it has practical and useful applications as shown in Sec. 8.3. In particular, it allows for the authenticated distribution of pure-state quantum keys in QPKE schemes answering one of the open questions posed in an earlier work [37].

Construction 2. *Let $\Pi_{\mathrm{QAE}} \coloneqq (\mathit{1QGen}, \mathit{1QEnc}, \mathit{1QDec})$ be the algorithms for a symmetric one-time QAE scheme on n-qubit messages. The construction for a one-time TD-DVK signature scheme on n-qubit messages with end time t_{end} is as follows:*

- $\mathit{KeyGen}(1^\lambda)$: *Sample* $k \leftarrow \mathit{1QGen}(1^\lambda)$.
- $\mathit{vkAnnounce}(k, t)$: *If* $t = t_{\text{end}}$*, then announce* $\mathsf{vk} := (t_{\text{end}}, k)$.
- $\mathit{Sign}(k, \phi)$: *Output state* $\sigma \leftarrow \mathit{1QEnc}_k(\phi)$.
- $\mathit{Verify}(\mathsf{vk}, \sigma')$: *Interpret* vk *as* (t_{end}, k)*. If* σ' *is received after time* t_{end}*, then output* $\bot$*. Otherwise, output* $\mathit{1QDec}_k(\sigma')$.

5.3 Many-Time TD-DVK Signature Scheme

In this section, we build a (many-time) TD-DVK signature scheme from pq-OWFs. In a time frame $[t_i, t_{i+1})$, messages are authenticated with a symmetric QAE scheme using a key k_i which is generated from the secret key and a pseudo-random function. Then, k_i is revealed at time t_{i+1} allowing receivers to verify transmissions received in the specified time frame. Following this, new messages are signed with k_{i+1}, and the process is repeated.

Recall, the existence of pq-OWFs implies the existence of pq-PRFs and a QUF QAE scheme (Lemma 4).

Construction 3. *Let* $\Pi_{\text{QAE}} := (\mathit{QGen}, \mathit{QEnc}, \mathit{QDec})$ *be the algorithms of a QUF QAE scheme on* n*-qubit messages where the key generation algorithm makes* p *coin tosses. Let* $F : \{0,1\}^\lambda \times \{0,1\}^n \to \{0,1\}^p$ *be a pq-PRF. The construction for TD-DVK signatures on* n*-qubit messages is as follows:*

- $\mathit{KeyGen}(1^\lambda)$: *Sample* $\mathsf{sk} \leftarrow \{0,1\}^\lambda$.
- $\mathit{vkAnnounce}(\mathsf{sk}, t)$: *If* $t = t_{i+1}$ *for some* $i \in \mathbb{N}$*, then compute* $y_i := F(\mathsf{sk}, i)$ *and announce* $\mathsf{vk}_i := (t_i, t_{i+1}, y_i)$.
- $\mathit{Sign}(\mathsf{sk}, t, \phi)$: *Let* $j \in \mathbb{N}$ *be such that* $t \in [t_j, t_{j+1})$.
  1. *Compute* $y_j := F(\mathsf{sk}, j)$.
  2. *Generate* $\mathsf{k}_j \leftarrow \mathit{QGen}(1^\lambda; y_j)$.
  3. *Output* $\sigma \leftarrow \mathit{QEnc}_{\mathsf{k}_j}(\phi)$.
- $\mathit{Verify}(\mathsf{vk}_i, \sigma')$:
  1. *Interpret* vk_i *as* (t_i, t_{i+1}, y_i).
  2. *Generate* $\mathsf{k}_i \leftarrow \mathit{QGen}(1^\lambda; y_i)$.
  3. *Output* $\mathit{QDec}_{\mathsf{k}_i}(\sigma')$.

Theorem 6. *Construction 3 is a quantum unforgeable TD-DVK signature scheme assuming the existence of a QUF QAE and pq-PRFs.*

Proof. Fix $i \in \mathbb{N}$ and consider the unforgeability experiment $\mathsf{TDVK\text{-}Sign}^{\text{QUF-Forge}}_{\Pi,\mathcal{A}}(\lambda, i)$. $\mathcal{A}$ is given oracle access to $\mathsf{Sign}(\mathsf{sk}, \cdot, \cdot)$ and tries to output a signature that is verified with key vk_i, specifically by using $\mathsf{k}_i = F(\mathsf{sk}, i)$.

By the security of pq-PRFs, we can switch to a hybrid where the keys $(\mathsf{k}_j)_{j \le i}$ are chosen uniformly at random. In particular, $\mathcal{A}$ cannot distinguish k_i from random and must forge a signature that is verified with key k_i. $\mathcal{A}$ has oracle access to $\mathsf{QEnc}_{\mathsf{k}_i}(\cdot)$ and $\mathsf{QDec}_{\mathsf{k}_i}(\cdot)$ but this is the only information $\mathcal{A}$ can receive related to k_i. If the output of the experiment is 1 with non-negligible probability, then $\mathcal{A}$ can easily be converted into an adversary that brakes the QUF-security of Π_{QAE}. ■

It seems that Construction 3 is insecure after time t_{2^n} since the keys start repeating. However, this issue can easily be resolved using the pseudorandom function. For instance, for any integer p, by letting $\mathsf{k}_{i+2^{pn}} := F^p(\mathsf{sk}, i)$, where F^p denotes p applications of F, the keys are unpredictable at any time. In other words, we can keep producing seemingly random keys indefinitely.

6 Signatures in the Bounded Quantum Storage Model

In this section, we present a signature scheme for quantum messages that is information-theoretically secure against adversaries with bounded quantum memory. We note that signing and verification algorithms in this setting do not depend on time. The public-keys are mixed-states and are certified with classical certification keys.

6.1 Definitions

We define the notion of BQS signatures on quantum messages. Such constructions are built against adversaries $\mathcal{A}_{\mathsf{s}}$ with qmemory bound s. The public keys are distributed for a limited time only. In fact, this is unavoidable for information-theoretic security as noted in [11]. We note that the signature scheme on classical messages in the bounded *classical* storage model suffers from this limitation as well [24].

Definition 21 (BQS Signature Scheme). *A* BQS signature scheme Π *over quantum message space* $\mathcal{Q}$ *against adversaries with memory bound* s*, with key distribution end time* t_{end}*, consists of the following algorithms:*

- $\mathsf{KeyGen}(1^\lambda, 1^s)$*: Outputs a secret key* sk*.*
- $\mathsf{ckAnnounce}(\mathsf{sk}, t)$*: Outputs a classical certification key* ck *at time* t_{end}*.*
- $\mathsf{pkSend}(1^s, \mathsf{sk})$: *Outputs a quantum public-key* ρ*.*
- $\mathsf{vkReceive}(\rho, \mathsf{ck})$: *Processes* ρ *using* ck *and outputs a classical verification key* vk*.*
- $\mathsf{Sign}(\mathsf{sk}, \phi)$: *Outputs a signature* σ *for* $\phi \in \mathcal{Q}$ *using* sk*.*
- $\mathsf{Verify}(\mathsf{vk}, \sigma')$*: Outputs a message* ϕ' *or* $\perp$ *using* vk*.*

Definition 22 (Correctness). *A BQS signature scheme* Π *is* correct *if for any* $\mathsf{sk} \leftarrow \mathsf{KeyGen}(1^\lambda, 1^s)$*,* $\rho \leftarrow \mathsf{pkSend}(1^s, \mathsf{sk})$*,* $\mathsf{ck} \leftarrow \mathsf{ckAnnounce}(\mathsf{sk}, t_{\text{end}})$*, and* $\mathsf{vk} \leftarrow \mathsf{vkReceive}(\rho, \mathsf{ck})$*:*

$$\| \mathsf{Verify}(\mathsf{vk}, \mathsf{Sign}(\mathsf{sk}, \cdot)) - \mathsf{id}(\cdot) \| \leq \mathsf{negl}(\lambda).$$

We introduce a security notion for BQS signature schemes which allows the adversary unrestricted access to the signing oracle and the verification oracle under the verification key that the experiment extracts. We also give the adversary the ability to tamper with the quantum public-keys, meaning the adversary provides the quantum key which the experiment uses to extract its

verification key. We call our notion *quantum unforgeability under chosen key attack (BQS-QUCK).*

Experiment $\mathsf{BQS\text{-}Sign}^{\mathrm{QUCK}}_{\Pi,\mathcal{A}}(\lambda, \mathsf{s})$:

1. Experiment $\mathcal{E}$ samples $\mathsf{sk} \leftarrow \mathsf{KeyGen}(1^\lambda, 1^\mathsf{s})$.
2. $\tilde{\rho} \leftarrow \mathcal{A}^{\mathsf{pkSend}(1^\mathsf{s},\mathsf{sk})}$.
3. $\mathcal{E}$ announces certification key $\mathsf{ck} \leftarrow \mathsf{ckAnnounce}(\mathsf{sk}, t_{\mathrm{end}})$.
4. $\mathcal{E}$ extracts key $\mathsf{k} \leftarrow \mathsf{vkReceive}(\tilde{\rho}, \mathsf{ck})$.
5. $\tilde{\sigma} \leftarrow \mathcal{A}^{|\mathsf{Sign}(\mathsf{sk},\cdot)\rangle,|\mathsf{Verify}(\mathsf{k},\cdot)\rangle}$. Let $\phi_1, ..., \phi_p$ denote the density matrices of the quantum messages $\mathcal{A}$ submitted to the signature oracle.
6. $\mathcal{E}$ computes $\tilde{\phi} \leftarrow \mathsf{Verify}(\mathsf{k}, \tilde{\sigma})$.
7. If $\tilde{\phi} = \perp$, then the experiment is terminated and the outcome is 0.
8. Otherwise, let $F \coloneqq \max_{i\in[p]} \delta_F(\phi_i, \tilde{\phi})$ be the maximum fidelity.
9. $\mathcal{E}$ flips a biased coin that "produces" head with probability F and tails with probability $1 - F$.
10. The output of the experiment is 1 if the coin toss outcome is tails. Otherwise, the output is 0.

Definition 23. *A BQS signature scheme Π satisfies* quantum unforgeability under chosen key attack (BQS-QUCK) *if for $\mathsf{s} \in \mathbb{N}$ and any (computationally-unbounded) adversary $\mathcal{A}_\mathsf{s}$,*

$$\Pr[\mathit{BQS\text{-}Sign}^{QUCK}_{\Pi,\mathcal{A}_s}(\lambda, s) = 1] \leq \mathsf{negl}(\lambda).$$

6.2 Construction

In this section, we show how to sign quantum messages in the BQSM. The quantum public-keys consist of a program broadcast of an appropriate function. A subset of the function image is revealed after the broadcast to allow users to certify their evaluations. The signature of a message consists of two one-time programs followed by an QAE of the message state. The one-time programs reveal the necessary information to verify the state only when evaluated on inputs corresponding to correct evaluations of the broadcasted program. The scheme requires the honest user to have $O(l^2)$ qmemory where l is the size of the quantum messages to be signed.

Construction 4. *Let $\Pi_{1\mathrm{QAE}} = (\mathit{1QGen}, \mathit{1QEnc}, \mathit{1QDec})$ be the algorithms of a one-time QAE scheme over l-qubit messages where the key generation makes ℓ coin tosses. Let $\Pi_{Br} = (\mathit{KeyGen}, \mathit{Br}, \mathit{Eval})$ be the algorithms for a program broadcast. Let t_{end} denote the time when key distribution phase ends. The construction for BQS signature scheme over l-qubit messages is as follows:*

- *KeyGen*$(1^\lambda, 1^s)$*:*
  1. *Let $m = 100\ell^2$ and $n = \max(\lceil \frac{s}{2m} \rceil + 1, m)$.*
  2. *Choose uniformly at random a $m \times n$ binary matrix M and define $P(x) \coloneqq M \cdot x$.*

3. *Sample an evaluation key* $\mathsf{ek} \leftarrow \mathsf{KeyGen}(1^\lambda, t_{\text{end}})$.
4. *Output* $\mathsf{sk} = (P, \mathsf{ek})$

– $\mathsf{pkSend}(1^s, \mathsf{sk})$*: Broadcast the program* P*, i.e. output* $\rho \leftarrow \mathsf{Br}(1^s, \mathsf{ek}, P)$.

– $\mathsf{ckAnnounce}(\mathsf{sk}, t)$*:*
1. *If* $t \neq t_{\text{end}}$*, then output* $\perp$.
2. *Otherwise, choose a random subset* S *of* $[m]$ *of size* $\frac{m}{2}$ *and hash function* $H \leftarrow \mathbf{H}_{m/2,m/8}$.
3. *Let* P_S *be the matrix consisting of rows in* P *whose index is in* S.
4. *Output* $\mathsf{ck} := (t_{\text{end}}, S, P_S, H)$.

– $\mathsf{vkReceive}(\mathsf{ck}, \rho, t)$*: Process the public and certification keys as follows:*
1. *Interpret* ck *as* $(t_{\text{end}}, S, P_S, H)$.
2. *If* ρ *is received after* t_{end}*, then output* $\perp$.
3. *Sample* $v \leftarrow \{0,1\}^n$.
4. *Compute* $\mathsf{Eval}(\rho, v)$ *and let* $P'(v)$ *denote the output.*
5. *If the entries in* $P'(v)$ *whose index is in* S *do not agree with* $P_S(v)$*, then output* $\perp$.
6. *Otherwise, let* $\overline{S}$ *be the complement of* S *and let* $P_v := H(P'_{\overline{S}}(v))$.
7. *Output* $\mathsf{k}_v := (v, P_v)$.

– $\mathsf{Sign}(\mathsf{sk}, \phi)$*:*
1. *Sample* $\mathsf{k} \leftarrow \mathsf{1QGen}(1^\lambda)$ *and* $a, b, a', b', a'', b'' \leftarrow \{0,1\}^\ell$.
2. *Define* $f(x) := ax + b \mod 2^\ell$ *and similarly define* f' *and* f''.
3. *Let* $F(x) := (f(x), f'(x), f''(x))$ *and let* $P_x := H(P_{\overline{S}}(x))$.
4. *Construct a program* $S_{P,F,\mathsf{k}}$ *which takes as input* $(x, y) \in \{0,1\}^{n,2\ell}$ *and does as follows:*
   (a) *Compute* P_x *and interpret the resulting string as an encoding for* $M_x \in \{0,1\}^{2\ell \times 2\ell}$, $M'_x \in \{0,1\}^{3\ell \times 3\ell}$, $z_x \in \{0,1\}^{2\ell}$ *and* $z'_x \in \{0,1\}^{3\ell}$.
   (b) *If* $y = M_x \cdot (t \| f(t)) + z_x$ *for some* $t \in \{0,1\}^\ell$*, then output* $(M'_x \cdot (f(t) \| f'(t) \| f''(t)) + z'_x, f'(t)\mathsf{k} + f''(t), \mathsf{k})$.
   (c) *Output* $\perp$ *otherwise.*
5. *Output the signature state* $\sigma \leftarrow \langle \mathcal{O}_s(F), \mathcal{O}_s(S_{P,F,\mathsf{k}}), \mathsf{1QEnc}_\mathsf{k}(\phi) \rangle$ *sent in sequence.*

– $\mathsf{Verify}(\mathsf{k}_v, \sigma)$*:*
1. *Interpret* k_v *as* (v, Y_v).
2. *Check the signature is of the correct form and output* $\perp$ *if it is not.*
3. *Evaluate the first one-time program on a random input* $t \leftarrow \{0,1\}^\ell$ *and let* $(f(t), f'(t), f''(t))$ *denote the result.*
4. *Use* Y_v *to deduce* (M_v, M'_v, z_v, z'_v) *and evaluate the second program on* $(v, M_v \cdot (t \| f(t)) + z_v)$.
5. *If the output is* $(M'_v \cdot (f(t) \| f'(t) \| f''(t)) + z'_v, f'(t)\mathsf{k}' + f''(t), \mathsf{k}')$ *for some* $\mathsf{k}' \in \{0,1\}^\ell$*, then output* $\mathsf{1QDec}_{\mathsf{k}'}(|\overline{c}\rangle)$ *where* $|\overline{c}\rangle$ *is the final state in* σ *and otherwise output* $\perp$.

Remark 1. In the Verify algorithm, the signature is checked to be of the correct form. More explicitly, this means the signature is checked to be of the form $\langle \mathcal{O}_\mathsf{s}(F), \mathcal{O}_\mathsf{s}(S), \overline{c} \rangle$ for $\mathbf{NC}^1$ functions $F : \{0,1\}^\ell \to \{0,1\}^{\ell,\ell,\ell}$ and $S : \{0,1\}^{n,2\ell} \to \{0,1\}^{3\ell,\ell,\ell}$. If a query submitted to the verification oracle is not of this form, then it is rejected. One-time programs satisfy this form of receiver security as shown in [11].

We briefly give an intuition behind the security of the scheme before presenting the formal proof. Intuitively, the signing algorithm consists of two one-time programs and an encryption of the message. The first one-time program is independent of the secret key and consists of random linear functions, and the second only returns non-$\perp$ outputs on inputs that involve the secret key. Therefore, it seems that the adversary, not knowing the secret key, cannot learn anything from these one-time programs. The problem is that the adversary may take a valid signature and edit in some way so that it signs another message. Intuitively, this attack is thwarted using this dual one-time program structure. Notice that F consists of linear functions so the evaluation on any input is indistinguishable from random when given a single evaluation of another input. In the scheme, F is sampled fresh at random for every signature and only a single one-time program of F is provided. In other words, an adversary can learn a single evaluation and all other evaluations of F are indistinguishable from random. In the second one-time program, for valid inputs, we essentially perform an authentication of a randomly sampled key k along with an output from the first one-time program using a part of the secret key. The key k is used to perform an authenticated encryption of the quantum message. It is difficult for the adversary to meaningfully edit the first one-time program (without failing verification) since its output is authenticated in the second one-time program. At the same time, it is difficult to edit the output of the second one-time program since the adversary only has knowledge of a single evaluation of F. In essence, the two one-time programs are interdependent in a way that makes it difficult to meaningfully modify either of them.

Theorem 7. *Construction 4 is a quantum unforgeable BQS signature scheme.*

Proof. In the $\mathsf{BQS\text{-}Sign}^{\mathrm{QUCK}}_{\Pi,\mathcal{A}_\mathsf{s}}(\lambda)$ experiment, the adversary $\mathcal{A}_\mathsf{s}$ outputs a key $\tilde{\rho} \leftarrow \mathcal{A}_\mathsf{s}^{\mathsf{pkSend}(1^\mathsf{s},\mathsf{sk})}(\mathsf{ck})$ and, then the experiment extracts a key $\mathsf{k}_v \leftarrow \mathsf{vkReceive}(\tilde{\rho},\mathsf{ck})$ for some random v. $\mathcal{A}_\mathsf{s}$ must send $\tilde{\rho}$ before t_{end} or else $\mathsf{k}_v = \perp$. In the case $\tilde{\rho}$ is received before t_{end}, $\mathcal{E}$ obtains an evaluation from the public-key copy which we denote by $\tilde{P}(v)$. Note that with high probability either $\mathsf{k}_v = \perp$ or $\|\tilde{P}(v) - P(v)\| \leq n/8$. This is because if $\tilde{P}(v)$ differs from $P(v)$ on more than $n/8$ positions, then there is less than $2^{-n/8}$ chance that none of these positions are a subset of S. If one of the positions is in S, then $\mathsf{k}_v = \perp$ by definition of $\mathsf{vkReceive}$. Hence, we can assume that $\tilde{\rho}$ is received before t_{end} and that $\|\tilde{P}(v) - P(v)\| \leq n/8$.

By the security of program broadcast (Theorem 4), $\mathcal{A}_\mathsf{s}$'s access to the public-key copies can be simulated with $\frac{\mathsf{s}}{2m}$ queries to an oracle of P. By Lemma 1, since M is a matrix of dimension $m \times n$ and $n > \frac{\mathsf{s}}{2m}$, $\mathcal{A}_\mathsf{s}$ cannot guess the output of $P(x) = M \cdot x$ on a random input except with negligible probability. In other words, if K_v represents the random variable for $P(v)$ from the perspective of $\mathcal{A}_\mathsf{s}$, then there exists negligible ϵ such that $H^\epsilon_\infty(K_v) = O(m)$. This implies that if $\tilde{K}_v$ represents the random variable for $\tilde{P}(v)$ from the perspective of $\mathcal{A}_\mathsf{s}$, then there exists negligible ϵ' such that $H^{\epsilon'}_\infty(\tilde{K}_v) \geq O(3m/4)$. Privacy Amplification Theorem 1, then implies that (whp) $\mathcal{A}$ cannot distinguish $\tilde{P}_v \coloneqq H(\tilde{P}_{\overline{S}}(v))$ from random.

$\mathcal{A}_{\mathsf{s}}$ gets a polynomial number of signature and verification oracle queries where verification is performed using $\mathsf{k}_v \coloneqq (v, \tilde{P}_v)$. Denote the signatures produced by the oracle as $\langle \mathcal{O}_{\mathsf{s}}(F_i), \mathcal{O}_{\mathsf{s}}(S_{P,F_i,\mathsf{k}_i}), |\overline{c}\rangle_i \rangle_{i \in Q_S}$ and denote the verification queries submitted to the oracle as $\langle \mathcal{O}_{\mathsf{s}}(\hat{F}_j), \mathcal{O}_{\mathsf{s}}(\hat{S}_j), \hat{c}_j \rangle_{j \in Q_V}$ for $\mathbf{NC}^1$ functions $\hat{F}_j : \{0,1\}^{\ell} \to \{0,1\}^{\ell,\ell,\ell}$ and $\hat{S}_j : \{0,1\}^{n,2\ell} \to \{0,1\}^{3\ell,\ell,\ell}$. If a query submitted to the verification oracle is not of this form, then it is rejected.

$\mathcal{A}_{\mathsf{s}}$ cannot distinguish the key $\tilde{P}_v$ from random with the signature queries alone since such an attack can be simulated with a single oracle query to each program in the set $\{\mathcal{O}_{\mathsf{s}}(S_{P,F_i,\mathsf{k}_i})\}_{i \in Q_S}$ by the security of one-time programs (Theorem 3). Lemma 1 shows that $\tilde{P}_v$ remains indistinguishable from random given polynomial guesses.

We now consider what $\mathcal{A}_{\mathsf{s}}$ can learn from the first verification query $\langle \mathcal{O}_{\mathsf{s}}(\hat{F}_0), \mathcal{O}_{\mathsf{s}}(\hat{S}_0), \hat{c}_0 \rangle$. Suppose $\mathcal{A}_{\mathsf{s}}$ received $\mathcal{Q}_0 \coloneqq \langle \mathcal{O}_{\mathsf{s}}(F_i), \mathcal{O}_{\mathsf{s}}(S_{P,F_i,\mathsf{k}_i}), |\overline{c}\rangle_i \rangle_{i \in [Q_0]}$ from the signature oracle prior to submitting the verification query. $\mathcal{A}_{\mathsf{s}}$ may utilize these signatures in the construction of the verification query.

Note that, the memory bound for $(\mathcal{O}_{\mathsf{s}}(F_i))_{i \in [Q_0]}$ must have already applied before the verification query is answered meaning there exists $(\hat{t}_i)_{i \in [Q_0]}$ such that $\mathcal{A}_{\mathsf{s}}$ can only distinguish $F_i(x)$ from random on $x = \hat{t}_i$ for all $i \in [Q_0]$ by the security of one-time programs.

Now let A_0 be the sub-algorithm of $\mathcal{A}_{\mathsf{s}}$ which produces the verification query using the signatures $\mathcal{Q}_0$. When the verification oracle receives the query, it evaluates the first one-time program on a random input say t_0 to get $\hat{F}_0(t_0) = (\hat{f}_0(t_0), \hat{f}'_0(t_0), \hat{f}''_0(t_0))$. Next, it evaluates the second one-time program on $\hat{S}_0(v, M_v \cdot (t_0 \| \hat{f}_0(t_0)) + z_v)$ and gets either an output of the form $(M'_v \cdot (\hat{f}_0(t_0) \| \hat{f}'_0(t_0) \| \hat{f}''_0(t_0)) + z'_v, \hat{f}'(t_0)\hat{\mathsf{k}}_0 + \hat{f}''(t_0), \hat{\mathsf{k}}_0)$ for some string $\hat{\mathsf{k}}_0$ or it rejects and outputs $\perp$.

The goal is to show that (whp) either the output of the verification query is $\perp$ or that $\mathcal{A}_{\mathsf{s}}$ cannot distinguish $\hat{F}_0(t_0)$ from random. We split the verification procedure performed by the oracle into three parts. Firstly, let Or_0 be the sub-algorithm of the oracle which evaluates $\mathcal{O}_{\mathsf{s}}(\hat{F}_0)$ on t_0 without access to $\tilde{P}_v$. Let Or'_0 be the sub-algorithm which evaluates $\mathcal{O}_{\mathsf{s}}(\hat{S}_0)$ on $(v, M_v \cdot (t_0 \| \hat{f}_0(t_0)) + z_v)$ with access only to this input (without access to $\tilde{P}_v$) and outputs the result of the evaluation denoted by y_0. Finally, let Or''_0 be the sub-algorithm of the oracle with full access to $\tilde{P}_v$ and that checks whether $y_0 = (M'_v \cdot (\hat{f}_0(t_0) \| \hat{f}'_0(t_0) \| \hat{f}''_0(t_0)) + z'_v, \hat{f}'_0(t_0)\hat{\mathsf{k}}_0 + \hat{f}''_0(t_0), \hat{\mathsf{k}}_0)$ for some key $\hat{\mathsf{k}}_0$ and correspondingly outputs $\mathsf{1QDec}_{\hat{\mathsf{k}}_0}(|\overline{c}\rangle)$ or $\perp$. We denote this output by m_0. Then we can write:

$$\begin{aligned} \hat{F}_0(t_0) &\leftarrow \mathsf{Or}_0 \circ \mathsf{A}_0(\mathcal{Q}_0) \\ y_0 &\leftarrow \mathsf{Or}'_0 \circ \mathsf{A}_0(\mathcal{Q}_0). \\ m_0 &\leftarrow \mathsf{Or}''_0(y_0) \end{aligned}$$

Critically, the algorithm A_0 is oblivious of M_v, z_v and Or'_0 has access to only a single evaluation of the function $Q_v(x) \coloneqq M_v \cdot (x) + z_v$, namely $Q_v(t_0 \| \hat{f}_0(t_0))$. By the security of one-time programs, the second procedure can be simulated by a simulator $\mathcal{S}_0$ with access to only a single query to each program $(S_{P,F_i,\mathsf{k}_i})_{i \in [Q_0]}$

and to the value $Q_v(t_0\|\hat{f}_0(t_0))$. The query to S_{P,F_i,k_i} will yield $\perp$ except with negligible probability unless $f_i(t_0) = \hat{f}_0(t_0)$ since $\mathcal{S}_0$ cannot guess $Q_v(x)$ on $x \neq t_0\|\hat{f}_0(t_0)$. If this condition is not satisfied for any $i \in [Q_0]$, then $\mathcal{S}_0$ will (whp) receive $\perp$ from all its oracle queries. This would mean $\mathcal{S}_0$ cannot guess $Q'_v(x) \coloneqq M'_v \cdot x + z'_v$ on any value and thus the output of Or''_0 will be $m_0 = \perp$. Note that there is at most a single value i that satisfies this condition (whp) since the parameters of the functions $(F_i)_{i\in[Q_0]}$ are chosen independently at random.

Assume this condition is satisfied only by the function F_n where $n \in [Q_0]$. There is negligible probability that $t_0 = \hat{t}_n$ and so $\mathcal{A}_\mathsf{s}$ cannot distinguish $F_n(t_0)$ from random. To sum up, (whp) the query to S_{P,F_i,k_i} will yield $\perp$ except if $i = n$, in which case $\mathcal{S}_0$ can learn $(Q'_v(f_n(t_0)\|f'_n(t_0)\|f''_n(t_0)), f'_n(t_0)\mathsf{k}_n + f''_n(t_0), \mathsf{k}_n)$. Thus, the simulator only has a single evaluation of Q'_v and (whp) cannot guess $Q'_v(x)$ on $x \neq f_n(t_0)\|f'_n(t_0)\|f''_n(t_0)$. Hence, we must have either $m_0 = \perp$ or $\hat{F}_0(t_0) = F_n(t_0)$. In the second case, $\mathcal{A}_\mathsf{s}$ cannot distinguish $\hat{F}_0(t_0)$ from random which is what we wanted to show.

Next, we must have (whp) $\hat{\mathsf{k}}_0 = \mathsf{k}_n$ since $\mathcal{S}_0$ cannot guess $f'_n(t_0)x + f''_n(t_0)$ on $x \neq \mathsf{k}_n$. By the security and unclonability of one-time programs, since the oracle is able to learn information regarding the evaluation of S_{P,F_n,k_n} on $(v, Q_v(t_0\|\hat{f}_0(t_0)))$, the adversary cannot learn any information from the one-time program $\mathcal{O}_\mathsf{s}(S_{P,F_n,\mathsf{k}_n})$ and thus cannot authenticate a new message using $\mathsf{1QEnc}$ with the key k_n by the security of one-time QAE $\Pi_{1\mathrm{QAE}}$.

Overall, the values involved in the verification query are v, $Q_v(t_0\|\hat{f}_0(t_0))$, $Q'_v(\hat{f}_0(t_0)\|\hat{f}'_0(t_0)\|\hat{f}''_0(t_0))$, $\hat{f}'(t_0)\hat{\mathsf{k}}_0 + \hat{f}''(t_0)$ and $\hat{\mathsf{k}}_0$. Even if $\mathcal{A}_\mathsf{s}$ learns all these values, it does not help in determining the M_v, M'_v, z_v, z'_v since $\hat{F}_0(t_0)$ is indistinguishable from random.

Notice that the only requirement to apply this argument on a verification query is that $\mathcal{A}$ cannot distinguish the key k_v from random when it submits the query. Hence, this argument can be applied to all verification queries and it can be deduced inductively that $\mathcal{A}_\mathsf{s}$ cannot distinguish the key from random when it submits the forgery.

Let $\langle \mathcal{O}_\mathsf{s}(\hat{F}), \mathcal{O}_\mathsf{s}(\hat{S}), \hat{c}\rangle$ be the forged signature. By the same arguments used above, if $\hat{\mathsf{k}}$ is the key output of program $\mathcal{O}_\mathsf{s}(\hat{S})$ when evaluated by the experiment, then there exists n' such that $\hat{\mathsf{k}} = \mathsf{k}_{n'}$ or the signature is invalid. In the former case, we showed that $\mathcal{A}_\mathsf{s}$ cannot authenticate a new message with $\hat{\mathsf{k}}$ so either the signature is invalid or $\hat{c} \approx_{\mathsf{negl}(\lambda)} \overline{c}$ which is equivalent to the fidelity being close to 1. In either case, the outcome of the experiment is 0 except with negligible probability. ■

7 Time-Dependent Quantum Money

In this section, we present two publicly verifiable quantum money schemes where the money generation and verification algorithms depend on time. The first construction is based on $\mathsf{TD\text{-}DVK}$ signatures with time-dependent verification keys, and the second is based on TD signatures with time-independent verification key.

7.1 Definitions

We first define TD public-key quantum money. Such a scheme consists of the following algorithms for the bank: KeyGen for generating a secret key, and CoinGen for generating a banknote. For the public user, we define the algorithm CoinVerify which describes how to validate a banknote allowing the public to carry out transactions and payments.

Definition 24 (TD Public-Key Quantum Money). *A* TD public-key quantum money *scheme consists of the following algorithms for the bank:*

- $\mathsf{KeyGen}(1^\lambda)$: *Outputs a secret key* sk *and verification key* vk.
- $\mathsf{CoinGen}(\mathsf{sk}, n, \mathtt{t}, v)$: *Generates a quantum banknote* $\$_v$ *worth* v *dollars with lifespan* n *based on the time* $\mathtt{t}$.

and the following algorithm for the public user:

- $\mathsf{CoinVerify}(\mathsf{vk}, \mathtt{t}, v, \$)$: *Checks if* $\$$ *is a valid banknote worth* v *dollars using* vk *based on the time* $\mathtt{t}$. *Outputs a banknote* $\$'$ *or* $\bot$.

We can also allow the verification key to vary with time. This is required for our second construction but not the first. In this case, we use an algorithm $\mathsf{vkAnnounce}(\mathsf{sk}, \mathtt{t})$ which describes how to vary the verification key with respect to time.

Definition 25 (Correctness). *A TD public-key quantum money scheme* Π *is* correct *if for any* $i, n, v \in \mathbb{N}$, *time* $T \in [t_i, t_{i+1})$, $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$, *and banknote* $\$ \leftarrow \mathsf{CoinGen}(\mathsf{sk}, n, T, v)$, *there is negligible probability (in* λ*) that* $\$$ *gets rejected in* n *transactions consisting of consecutive applications of* CoinVerify. *More mathematically, correctness requires that:*

$$\|\mathsf{CoinVerify}(\mathsf{vk}, T+n, v, \mathsf{CoinVerify}(\mathsf{vk}, T+(n-1), v, \ldots \mathsf{CoinVerify}(\mathsf{vk}, T+1, v, \$)\ldots)) - \bot \| \geq 1 - \mathsf{negl}(\lambda).$$

We now define an experiment to test the security of a TD quantum money scheme. Informally, this experiment requires that an adversary with access to a polynomial number of banknotes cannot generate a set of banknotes of greater value.

Experiment $\mathsf{TD\text{-}PKQM}_{\Pi,\mathcal{A}}(\lambda, T)$:

1. Sample $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$.
2. $\mathcal{A}$ is given vk and access to an oracle for $\mathsf{CoinGen}(\mathsf{sk}, \cdot, \cdot, \cdot)$ which rejects queries for time after T.
3. $\mathcal{A}$ receives $(\$_{v_i})_{i\in[p]}$ from the oracle, where v_i signifies the value of the banknote.
4. $\mathcal{A}$ outputs $(c_i, \tilde{\$}_{c_i})_{i\in[p']}$.
5. The output of the experiment is 1 if:

$$\sum_{z:\mathsf{CoinVerify}(\mathsf{vk},T,c_z,\tilde{\$}_{c_z})\neq\perp} c_z > \sum_{j\in[p]} v_j.$$

6. Otherwise, the output is 0.

Definition 26 (Security). *A TD-DVK public-key quantum money scheme Π is* secure *if for any QPT adversary $\mathcal{A}$ and $T \in \mathbb{R}$,*

$$\Pr\left[\mathit{TD\text{-}PKQM}_{\Pi,\mathcal{A}}(\lambda, T) = 1\right] \leq \mathsf{negl}(\lambda).$$

7.2 Construction for TD-DVK-PKQM

In this section, we present a scheme for TD-DVK public-key quantum money assuming pq-PRFs. Banknotes are quantum states secured with multiple layers of one-time QAE. The bank reveals a classical secret encryption key every second (or some fixed interval δ) allowing the public to validate banknotes by verifying one authentication layer. Eventually, after all the keys used in a banknote's generation have been revealed, the banknote expires. Hence, users must return the banknote to the bank before this occurs for replacement. The scheme is built for money values v such that $v \in \{0,1\}^\lambda$. However, it can easily be generalized to allow for larger values.

Construction 5. *Let $n, \lambda \in \mathbb{N}$ be security parameters and $p \in \mathsf{poly}(\lambda)$. Let $\Pi_{\mathrm{1QAE}} = (\mathit{1QGen}, \mathit{1QEnc}, \mathit{1QDec})$ be the algorithms for a one-time QAE scheme on $(n\lambda + 1)$-qubit messages such that the encryption of a m-qubit message is $(m + \lambda)$-qubits and where the key generation algorithm makes p coin tosses. Let $F : \{0,1\}^\lambda \times \{0,1\}^n \to \{0,1\}^\lambda$ and $F' : \{0,1\}^\lambda \to \{0,1\}^{3\lambda+p}$ be pq-PRFs. The construction for a TD-DVK quantum public-key money scheme is as follows:*

- $\mathit{KeyGen}(1^\lambda)$: *(Bank) Sample* $\mathsf{sk} \leftarrow \{0,1\}^\lambda$. *Output* sk.
- $\mathit{vkAnnounce}(\mathsf{sk}, \mathsf{t})$: *(Bank) If* $\mathsf{t} = t_{i+1}$ *for some* $i \in \mathbb{N}$, *then compute* $\mathsf{k}_i := F(\mathsf{sk}, i)$ *and announce* $\mathsf{vk}_i := (t_i, t_{i+1}, \mathsf{k}_i)$. *Otherwise, output* $\perp$.
- $\mathit{CoinGen}(\mathsf{sk}, n, \mathsf{t}, v)$: *(Bank) Let j be such that* $\mathsf{t} \in [t_j, t_{j+1})$.
  1. *Sample* $y \leftarrow \{0,1\}^p$ *and set* $y_j = y$.
  2. *For each* $i \in [j : j + n]$, *compute:*
     (a) $\mathsf{ek}_i = \mathit{1QGen}(1^\lambda; y_i)$.

(b) $\mathsf{k}_i \coloneqq F(\mathsf{sk}, i)$.
(c) $y_{i+1} \coloneqq F'(\mathsf{k}_i, i\|v\|t_{j+n}\|y_i)$.

3. *Generate*

$$\sigma \coloneqq \mathit{1QEnc}_{\mathsf{ek}_j}(\mathit{1QEnc}_{\mathsf{ek}_{j+1}}(...\mathit{1QEnc}_{\mathsf{ek}_{j+n}}(|0\rangle)...)).$$

4. *Output* $\$_{v,j,y} \coloneqq (t_j, t_{j+n}, v, y_j, \sigma)$.

– $\mathit{CoinVerify}(\mathsf{vk}_i, v, \$_{v,i,y})$:

1. *Interpret* $\$_{v,i,y}$ *as* $(t, t', v', y_i, \sigma_i)$ *and* vk_i *as* $(t_i, t_{i+1}, \mathsf{k}_i)$.
2. *If* $t \neq t_i$, $t' < t_{i+1}$, *or* $v' \neq v$, *then output* $\perp$.
3. *Compute* $y_{i+1} \coloneqq F'(\mathsf{k}_i, i\|v'\|t'\|y_i)$.
4. *Compute* $\mathsf{ek}_{i+1} = \mathit{1QGen}(1^\lambda; y_{i+1})$.
5. *Compute* $\sigma_{i+1} \coloneqq \mathit{1QDec}_{\mathsf{ek}_{i+1}}(\sigma_i)$.
6. *If* $\sigma_{i+1} = \perp$, *then output* $\perp$.
7. *Otherwise, accept banknote and output* $\$_{v,i+1,y} \coloneqq (t_{i+1}, t', v, y_{i+1}, \sigma_{i+1})$.

Theorem 8. *Construction 5 is a secure TD-DVK public-key quantum money scheme assuming the existence of pq-PRFs.*

Proof. We prove security through a sequence of hybrids. The first hybrid is the standard security experiment with $T \in [t_n, t_{n+1})$.

Hybrid H_0 (standard experiment):

1. Sample $\mathsf{sk} \leftarrow \mathsf{KeyGen}(1^\lambda)$.
2. For each $j \leq n$, generate $\mathsf{vk}_j \leftarrow \mathsf{vkAnnounce}(\mathsf{sk}, t_{j+1})$.
3. $\mathcal{A}$ is given $(\mathsf{vk}_j)_{j<n}$ and access to an oracle for $\mathsf{CoinGen}(\mathsf{sk}, \cdot, \cdot, \cdot)$ which rejects queries for time after T.
4. $\mathcal{A}$ receives $(\$_{v_i})_{i\in[p]}$ from the oracle, where v_i signifies the value of the banknote.
5. $\mathcal{A}$ outputs $(c_i, \tilde{\$}_{c_i})_{i\in[p']}$.
6. The output of the experiment is 1 if:

$$\sum_{z:\mathsf{CoinVerify}(\mathsf{vk}_n, c_z, \tilde{\$}_{c_z})\neq\perp} c_z > \sum_{j\in[p]} v_j.$$

7. Otherwise, the output is 0.

Hybrid H_1 is the same as H_0 except the pq-PRF $F(\mathsf{sk}, \cdot)$ is replaced with a completely random function $R(\cdot)$ on the same domain and co-domain. This only changes the CoinGen procedure, where now the keys $\mathsf{k}_i = R(i)$ are generated using R.

Hybrid H_2 is the same as H_1, except that the pq-PRF $F'(\mathsf{k}_n, \cdot)$ is replaced with a random function R'.

It is easy to see that hybrids H_0 and H_1 are indistinguishable by the security of pq-PRFs. Similarly, H_1 and H_2 are indistinguishable given that k_n is

indistinguishable from random as it is generated with a random function R, allowing us to replace $F'(\mathsf{k}_n, \cdot)$ with a random function. As a result, it is sufficient to prove security of H_2 which is what we do now.

Assume there exists a QPT adversary $\mathcal{A}$ such that $\Pr[\mathsf{H}_2 = 1]$ is non-negligible. Assume, without loss of generality, that $1 \in \mathcal{C}$ where $\mathcal{C} := \{z : \mathsf{CoinVerify}(\mathsf{vk}_n, c_z, \tilde{\$}_{c_z}) \neq \perp\}$. Interpret $\tilde{\$}_{c_1}$ as $(t_n, \tilde{t}', c_1, \tilde{y}_n, \tilde{\sigma})$.

When the experiment runs $\mathsf{CoinVerify}(\mathsf{vk}_n, c_1, \tilde{\$}_{c_1})$, it computes $\tilde{y}_{n+1} := R'(c_1 \| \tilde{t}' \| \tilde{y}_n)$ and then $\mathsf{ek}_{n+1} = \mathsf{1QGen}(1^\lambda; \tilde{y}_{n+1})$. If the bank did not perform the evaluation $R'(c_1 \| \tilde{t}' \| \tilde{y}_n)$ during the generation of banknotes, then it is clear that there is negligible probability that $\tilde{\$}_{c_1}$ passes verification. It is also clear that (whp) this evaluation only occurs at most in a single banknote. Without loss of generality, assume $\$_{v_1}$ involves this evaluation.

Interpret $\$_{v_1}$ as (t, t', v_1, y, σ). We have $R'(n \| c_1 \| \tilde{t}' \| \tilde{y}_n) = R'(n \| v_1 \| t' \| y_n)$ which implies (whp) that $c_1 = v_1$, $\tilde{y}_n = y_\ell$ and $t' = \tilde{t}'$.

As ek_{n+1} is indistinguishable from random with respect to $\mathcal{A}$, by the security of $\Pi_{1\mathrm{QAE}}$, $\mathcal{A}$ cannot produce two states that pass verification with key ek_{n+1} except with negligible probability. In other words, applying this argument to all banknotes, there is an injective mapping from $(c_z)_{z \in \mathcal{C}}$ to $(v_i)_{i \in [p]}$. This means that $\sum_{z \in [\mathcal{C}]} c_z \leq \sum_{j \in [p]} v_j$ giving a contradiction. ■

Remark 2. We make a short note regarding the efficiency of the scheme. First of all, the one-time QAE scheme $\Pi_{1\mathrm{QAE}}$ utilized in the scheme which encrypts m-qubit messages into $(m + \lambda)$-qubit messages exists unconditionally (Lemma 6). Now a quantum banknote consists of n applications of the one-time QAE on a single qubit $|0\rangle$. Hence, the banknote is a $(n\lambda + 1)$-qubit state, and so grows linearly with respect to its lifespan.

7.3 Construction for TD-PKQM

In this section, we present a public-key quantum money scheme where the generation and verification algorithms depend on time but, importantly, the verification key is time-independent allowing users to be completely offline. We prove security assuming the existence of TLPs and $\mathsf{pq\text{-}OWF}$s.

Construction 6. *Let $n, \lambda \in \mathbb{N}$ be security parameters and $p \in \mathsf{poly}(\lambda)$. Let $\Pi_{1\mathrm{QAE}} = (\mathit{1QGen}, \mathit{1QEnc}, \mathit{1QDec})$ be the algorithms of a one-time QAE scheme on $(n\lambda + 1)$-qubit messages where the key generation algorithm makes p coin tosses. Let $\Pi_{1\mathrm{E}} = (\mathit{1Gen}, \mathit{1Enc}, \mathit{1Dec})$ be the algorithms of a one-time encryption scheme on classical messages. Let $\Pi_{\mathrm{CS}} := (\mathit{CS.Gen}, \mathit{CS.Sign}, \mathit{CS.Verify})$ be the algorithms for a signature scheme on classical messages. Let $(\mathit{Puzzle.Gen}, \mathit{Puzzle.Sol})$ be the algorithms of a TLP scheme. The construction for a TD quantum public-key money scheme is as follows:*

- $\mathit{KeyGen}(1^\lambda)$: *(Bank) Sample* $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathit{CS.Gen}(1^\lambda)$. *Output* $(\mathsf{sk}, \mathsf{vk})$.
- $\mathit{CoinGen}(\mathsf{sk}, n, \mathsf{t}, v)$: *(Bank) Let j be such that $\mathsf{t} \in [t_j, t_{j+1})$.*
  1. *Sample tag* $r \leftarrow \{0,1\}^\lambda$ *and key* $\mathsf{k}_{j-1} \leftarrow \mathit{1Gen}(1^\lambda)$.

2. *For each* $i \in [j : j+n]$*:*
   (a) *Sample* $\mathsf{ek}_i = \mathit{1QGen}(1^\lambda)$.
   (b) *Sample* $\mathsf{k}_i = \mathit{1Gen}(1^\lambda)$.
   (c) *Generate* $P_i \leftarrow \mathit{Puzzle.Gen}(1^\lambda, 1, (\mathsf{ek}_i, \mathsf{k}_i))$.
   (d) *Encrypt* $Z_i = \mathit{1Enc}_{\mathsf{k}_{i-1}}(P_i)$.
   (e) *Sign* $s_i \leftarrow \mathit{CS.Sign}(\mathsf{sk}, (Z_i, v, r, t_{j+n}, t_i))$.
3. *Let* $Z \coloneqq (Z_j, \ldots, Z_{j+n})$ *and* $S \coloneqq (s_j, \ldots, s_{j+n})$.
4. *Generate*

$$\sigma \coloneqq \mathit{1QEnc}_{\mathsf{ek}_j}(\mathit{1QEnc}_{\mathsf{ek}_{j+1}}(\ldots \mathit{1QEnc}_{\mathsf{ek}_{j+n}}(|0\rangle)\ldots)).$$

5. *Output* $\$_{v,r,j} \coloneqq (t_{j+n}, r, v, \mathsf{k}_{j-1}, Z, S, \sigma)$.

– $\mathit{CoinVerify}(\mathsf{vk}, \mathsf{t}, v, \$)$: *Let* $i \in \mathbb{N}$ *be such that* $\mathsf{t} \in [t_i, t_{i+1})$.
1. *Interpret* $\$$ *as* $(t_n, r', v', \mathsf{k}', Z', S', \sigma')$.
2. *If* $v' \neq v$ *or* $i+1 > n$*, then output* $\bot$.
3. *For each* $k \in [n]$*, if* $\mathit{CS.Verify}(\mathsf{vk}, (Z'_k, v, r', t_n, t_k), S'_k) = \bot$*, then output* $\bot$.
4. *Compute* $P_i \coloneqq \mathit{1Dec}_{\mathsf{k}'}(Z'_i)$.
5. *Compute* $\mathit{Puzzle.Sol}(P_i)$ *and interpret the result as* $(\mathsf{ek}_i, \mathsf{k}_i)$.
6. *Compute* $\sigma_i \coloneqq \mathit{1QDec}_{\mathsf{ek}_i}(\sigma')$.
7. *If* $\sigma_{i+1} = \bot$*, then output* $\bot$.
8. *Otherwise, accept the banknote and output* $\$_{v,r',i+1} \coloneqq (t_n, r', v, \mathsf{k}_i, Z', S', \sigma_i)$.

Theorem 9. *Construction 6 is a secure TD-QM scheme assuming the existence of a EUF classical digital signature scheme and secure TLPs.*

Proof. Fix $T \in \mathbb{R}$ and let $m \in \mathbb{N}$ be such that $T \in [t_m, t_{m+1})$. We prove the scheme is secure through a sequence of hybrids. Hybrid H_0 is the standard security experiment which we restate here.

Hybrid H_0 (standard experiment):

1. Sample $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$.
2. $\mathcal{A}$ is given vk and access to an oracle for $\mathsf{CoinGen}(\mathsf{sk}, \cdot, \cdot, \cdot)$ which rejects queries for time after T.
3. $\mathcal{A}$ receives $(\$_{v_i})_{i \in [p]}$ from the oracle, where v_i signifies the value of the banknote.
4. $\mathcal{A}$ outputs $(c_i, \tilde{\$}_{c_i})_{i \in [p']}$.
5. The output of the experiment is 1 if:

$$\sum_{z : \mathsf{CoinVerify}(\mathsf{vk}, T, c_z, \tilde{\$}_{c_z}) \neq \bot} c_z > \sum_{j \in [p]} v_j.$$

6. Otherwise, the output is 0.

Next, hybrid H_1 is the same as H_0 except that the $\mathsf{CoinVerify}$ procedure rejects any coin where the classical part of the banknote was not generated by the bank. To do this, the bank keeps a storage M on information regarding previous banknotes generated. The new algorithms for the bank are as follows:

- $\mathsf{KeyGen}_{\mathsf{H}_1}(1^\lambda)$: Sample $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{CS.Gen}(1^\lambda)$. Output $(\mathsf{sk}, \mathsf{vk})$.
- $\mathsf{CoinGen}_{\mathsf{H}_1}(\mathsf{sk}, n, \mathtt{t}, v)$: Let $j \in \mathbb{N}$ be such that $\mathtt{t} \in [t_j, t_{j+1})$.
  1. Run $(t_{j+n}, r, v, \mathsf{k}_{j-1}, Z, S, \sigma) \leftarrow \mathsf{CoinGen}(\mathsf{sk}, n, \mathtt{t}, v)$.
  2. Store (Z, v, r, t_{j+n}) in storage $\mathbf{M}$.
  3. Output $(t_{j+n}, r, v, \mathsf{k}_{j-1}, Z, S, \sigma)$.
- $\mathsf{CoinVerify}_{\mathsf{H}_1}(\mathsf{vk}, \mathtt{t}, v, \$)$: Let $i \in \mathbb{N}$ be such that $\mathtt{t} \in [t_i, t_{i+1})$.
  1. Interpret $\$$ as $(t_{j+n}, r', v', \mathsf{k}', Z', S', \sigma')$.
  2. If $(Z', v', r', t_{j+n}) \notin \mathbf{M}$, then output $\perp$.
  3. Otherwise, output $\mathsf{CoinVerify}(\mathsf{vk}, \mathtt{t}, v, \$)$.

Next, hybrid H_2 is the same as hybrid H_1, except we modify the TLPs P_i for $i \geq m$ to instead encrypt $\perp$. The steps written in **bold** are the difference from H_1. The resulting algorithms for the bank are as follows:

- $\mathsf{KeyGen}_{\mathsf{H}_2}(1^\lambda)$: Sample $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{CS.Gen}(1^\lambda)$. Output $(\mathsf{sk}, \mathsf{vk})$.
- $\mathsf{CoinGen}_{\mathsf{H}_2}(\mathsf{sk}, n, \mathtt{t}, v)$:
  1. Let j be such that $\mathtt{t} \in [t_j, t_{j+1})$.
  2. Sample tag $r \leftarrow \{0,1\}^\lambda$ and key $\mathsf{k}_{j-1} \leftarrow \mathsf{1Gen}(1^\lambda)$.
  3. For each $i \in [j : j+n]$:
     (a) Sample $\mathsf{ek}_i = \mathsf{1QGen}(1^\lambda)$.
     (b) Sample $\mathsf{k}_i = \mathsf{1Gen}(1^\lambda)$.
     (c) **If $i \geq m$, generate $P_i \leftarrow \mathsf{Puzzle.Gen}(1^\lambda, 1, 0^{|\mathsf{k}_i|+|\mathsf{ek}_i|})$.**
     (d) Otherwise, $P_i \leftarrow \mathsf{Puzzle.Gen}(1^\lambda, 1, (\mathsf{ek}_i, \mathsf{k}_i))$.
     (e) $Z_i = \mathsf{1Enc}_{\mathsf{k}_{i-1}}(P_i)$.
     (f) Sign $s_i \leftarrow \mathsf{CS.Sign}(\mathsf{sk}, (Z_i, v, r, t_{j+n}, t_i))$.
  4. Let $Z \coloneqq (Z_j, \ldots, Z_{j+n})$ and $S \coloneqq (s_j, \ldots, s_{j+n})$.
  5. **Store (Z, v, r, t_{j+n}) and $(\mathsf{ek}_i, \mathsf{k}_i)_{i \geq m}$ in storage $\mathbf{M}$.**
  6. Generate
  $$\sigma \coloneqq \mathsf{1QEnc}_{\mathsf{ek}_j}(\mathsf{1QEnc}_{\mathsf{ek}_{j+1}}(\ldots \mathsf{1QEnc}_{\mathsf{ek}_{j+n}}(|0\rangle)\ldots)).$$
  7. Output $\$_{v,r,j} \coloneqq (t_{j+n}, r, v, \mathsf{k}_{j-1}, Z, S, \sigma)$.
- $\mathsf{CoinVerify}_{\mathsf{H}_2}(\mathsf{vk}, \mathtt{t}, v, \$)$: Let $i \in \mathbb{N}$ be such that $\mathtt{t} \in [t_i, t_{i+1})$.
  1. Interpret $\$$ as $(t_{j+n}, r', v', \mathsf{k}', Z', S', \sigma')$.
  2. If $v' \neq v$ or $i+1 > n$, then output $\perp$.
  3. For each $k \in [n]$, check if $\mathsf{CS.Verify}(\mathsf{vk}, (Z'_k, v, r', t_{j+n}, t_k), S'_k) = \perp$, then output $\perp$.
  4. Compute $P_i \coloneqq \mathsf{1Dec}_{\mathsf{k}'}(Z'_i)$.
  5. Otherwise, compute $\mathsf{Puzzle.Sol}(P_i)$ and interpret the result as $(\mathsf{ek}_i, \mathsf{k}_i)$.
  6. Compute $\sigma_{i+1} \coloneqq \mathsf{1QDec}_{\mathsf{ek}_i}(\sigma')$.
  7. If $\sigma_{i+1} = \perp$, then output $\perp$.
  8. Otherwise, accept the banknote and output $\$_{v,r',i+1} \coloneqq (t_{j+n}, r', v, \mathsf{k}_i, Z', S', \sigma_{i+1})$.
  9. Interpret $\$$ as $(t_n, r', v', \mathsf{k}', Z', S', \sigma')$.
  10. If $(Z', v', r', t_{j+n}) \notin \mathbf{M}$, then output $\perp$.
  11. If $v' \neq v$ or $i+1 > n$, then output $\perp$.
  12. For each $k \in [n]$, if $\mathsf{CS.Verify}(\mathsf{vk}, (Z'_k, v, r', t_n, t_k), S'_k) = \perp$, then output $\perp$.

13. Compute $P_i \coloneqq \mathsf{1Dec}_{\mathsf{k}'}(Z'_i)$.
14. **If $i \geq m$, then determine $(\mathsf{ek}_i, \mathsf{k}_i)$ from $\mathbf{M}$.**
15. Otherwise, compute $\mathsf{Puzzle.Sol}(P_i)$ and interpret the result as $(\mathsf{ek}_i, \mathsf{k}_i)$.
16. Compute $\sigma_i \coloneqq \mathsf{1QDec}_{\mathsf{ek}_i}(\sigma')$.
17. If $\sigma_{i+1} = \perp$, then output $\perp$.
18. Otherwise, accept the banknote and output $\$_{v,r',i+1} \coloneqq (t_n, r', v, \mathsf{k}_i, Z', S', \sigma_i)$.

Claim. $\Pr[\mathsf{H}_2 = 1] \leq \mathsf{negl}(\lambda)$.

Proof. Assume that there exists a QPT adversary $\mathcal{A}$ such that $\Pr[\mathsf{H}_2 = 1]$ is non-negligible. Interpret the banknote $\$_{v_1}$ generated by the bank in H_2 as $(t, r, v_1, \mathsf{k}, Z, S, \sigma)$. Let $\mathcal{C} \coloneqq \{z : \mathsf{CoinVerify}(\mathsf{vk}, T, c_z, \tilde{\$}_{c_z}) \neq \perp\}$ and assume without loss of generality that $1 \in \mathcal{C}$. Consider the corresponding banknote $\tilde{\$}_{c_1}$ submitted by $\mathcal{A}$. Interpret $\tilde{\$}_{c_1}$ as $(\tilde{t}, \tilde{r}, c_1, \tilde{\mathsf{k}}, \tilde{Z}, \tilde{S}, \tilde{\sigma})$.

By our assumption $(\tilde{Z}, c_1, \tilde{r}, \tilde{t}) \in \mathbf{M}$. Hence, without loss of generality, assume $(\tilde{Z}, c_1, \tilde{r}, \tilde{t}) = (Z, v_1, r, t)$. Since verification is performed at time $T \in [t_m, t_{m+1})$, the experiment retrieves $(\mathsf{ek}_{1,m}, \mathsf{k}_{1,m})$ from storage $\mathbf{M}$ where $(\mathsf{ek}_{1,m}, \mathsf{k}_{1,m})$ correspond to the keys sampled in the generation of $\$_{v_1}$. The experiment performs verification by running $\mathsf{1QDec}_{\mathsf{ek}_{1,m}}(\tilde{\sigma})$.

If each forgery with index in $\mathcal{C}$ has a unique tag, then this would imply that there is an injective mapping from the values $(c_i)_{i \in \mathcal{C}}$ to $(v_i)_{i \in [p]}$, giving $\sum_{z \in \mathcal{C}} c_z \leq \sum_{j \in [p]} v_j$ and an experiment outcome of 0.

Hence, assume without loss of generality that $\mathcal{A}$ submits another forgery in $\mathcal{C}$ with tag r. Then $\mathsf{1QDec}_{\mathsf{ek}_{1,m}}$ is performed with the same key $\mathsf{ek}_{1,m}$. If both forgeries pass verification, then this would mean that $\mathcal{A}$ is given one state, namely σ, that is authenticated with $\mathsf{ek}_{1,m}$, and produces two states that pass verification with $\mathsf{ek}_{1,m}$. Notice that $\mathcal{A}$ is not given any other information regarding $\mathsf{ek}_{1,m}$. Hence, $\mathcal{A}$ can easily be converted into a QPT algorithm that breaks QUF of Π_{1QAE} giving a contradiction. ■

Claim. No QPT adversary can distinguish between hybrids H_2 and H_1 with non-negligible advantage.

Proof. The only difference between H_2 and H_1 is that for $i \geq m$, the TLP P_i encrypting $(\mathsf{ek}_i, \mathsf{k}_i)$ is replaced with TLP encrypting $0^{|\mathsf{ek}_i|+|\mathsf{k}_i|}$.

If $\mathcal{A}$ receives a banknote at time t_j, then $\mathcal{A}$ requires 1 hour to unravel P_j and deduce $(\mathsf{ek}_j, \mathsf{k}_j)$ by the security of TLPs. Only after this step can $\mathcal{A}$ decrypt Z_{j+1} to retrieve P_{j+1}. Then, $\mathcal{A}$ again requires 1 hour to decrypt P_{j+1}. As $\mathcal{A}$ submits its forgery at time $T \in [t_m, t_{m+1})$, it cannot distinguish a TLP P_i hiding $(\mathsf{ek}_i, \mathsf{k}_i)$ from a TLP hiding $0^{|\mathsf{ek}_i|+|\mathsf{k}_i|}$ for any $i \geq m$.

Specifically, if an adversary can distinguish the two hybrids with non-negligible advantage, then this enables the construction of an algorithm that can distinguish whether P_i is hiding $(\mathsf{ek}_i, \mathsf{k}_i)$ or $0^{|\mathsf{ek}_i|+|\mathsf{k}_i|}$ for $i \geq m$ in time less than t_{m+1} contradicting the security of TLPs. ■

Claim. No QPT adversary can distinguish between hybrids H_1 and H_0 with non-negligible advantage.

Proof. The only difference between H_1 and H_0 is that the experiment rejects any forgeries where the classical part has not been generated by the bank. In particular, if a forgery is interpreted as $(t_{j+n}, r', v', \mathsf{k}', Z', S', \sigma')$, then in H_1, we must have $(Z', v', r', t_{j+n}) \in \mathbf{M}$.

However, note that (Z', v', r', t_{j+n}) is signed under the scheme Π_{CS}. Hence, if $\mathcal{A}$ can distinguish between these hybrids with non-negligible advantage, then, with non-negligible probability, $\mathcal{A}$ outputs a banknote where $(Z', v', r', t_{j+n}) \notin \mathbf{M}$, along with a valid signature of (Z', v', r', t_{j+n}). Such an adversary can be converted into a QPT algorithm that breaks the unforgeability of Π_{CS} using standard reductions. ■

To sum up, we have deduced that hybrid $\Pr[\mathsf{H}_2 = 1] \leq \mathsf{negl}(\lambda)$, no adversary can distinguish H_2 from H_1 with non-negligible advantage, and no adversary can distinguish H_1 from H_0 with non-negligible advantage. Hence, $\Pr[\mathsf{H}_0 = 1] \leq \mathsf{negl}(\lambda)$. ■

8 Public Key Encryption with Authenticated Quantum Public-Keys

In this section, we first define security for quantum public key encryption to take into account attacks that tamper with the quantum public-keys. Following this, we present two constructions that satisfy this notion of security by applying our many-time and one-time TD-DVK signature schemes to authenticate the public keys.

8.1 Definitions

We first recall the algorithms for a QPKE scheme [28, 42, 11] where the public-key is a quantum state. Due to the unclonability of quantum states, multiple copies of the key must be created and distributed to each user individually – this is the general approach taken in quantum public-key works [28, 42, 11]. Thus, we include an algorithm pkSend which generates a public-key copy, and an algorithm ekReceive which describes how to process a public-key copy and extract a classical encryption key. The public-key copies are certified with a certification key that may vary with time – hence, we include an algorithm ckAnnounce to describe how to announce the certification keys.

In this work, we only deal with the encryption of classical messages. This is because any public-key encryption scheme on classical messages can be adapted to quantum messages with a simple modification as shown in [4].

Definition 27 (Quantum Public Key Encryption). *A quantum public encryption scheme Π over classical message space $\mathcal{M}$ consists of the following algorithms:*

- $\mathsf{KeyGen}(1^\lambda)$: *Outputs a classical secret key* sk.

- $\mathsf{pkSend}(\mathsf{sk}, \mathsf{t})$: *Output a quantum public-key* ρ *using the secret key and depending on the time* t.
- $\mathsf{ckAnnounce}(\mathsf{sk}, \mathsf{t})$: *If* $\mathsf{t} = t_{i+1}$ *for some* $i \in \mathbb{N}$, *then outputs a classical certification key* ck_i *using the secret key* sk.
- $\mathsf{ekReceive}(\rho, \mathsf{ck}_i)$: *Extract a classical encryption key* k *from* ρ *and* ck_i.
- $\mathsf{Enc}(\mathsf{k}, \mu)$: *Outputs a classical ciphertext* c *for* $\mu \in \mathcal{M}$ *using* k.
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{c})$: *Outputs a message* μ' *by decrypting* c *using the secret key* sk.

Definition 28 (Correctness). *A QPKE scheme* Π *is* correct *if for any message* $\mu \in \mathcal{M}$, $i \in \mathbb{N}$, *and time* $\mathsf{t} \in [t_i, t_{i+1})$,

$$\Pr\left[\mathsf{Dec}(\mathsf{sk}, \mathsf{c}) = \mu \;\middle|\; \begin{array}{c} \mathsf{sk} \leftarrow \mathsf{KeyGen}(1^\lambda) \\ \rho \leftarrow \mathsf{pkSend}(\mathsf{sk}, \mathsf{t}) \\ \mathsf{ck}_i \leftarrow \mathsf{ckAnnounce}(\mathsf{sk}, t_{i+1}) \\ \mathsf{k} \leftarrow \mathsf{ekReceive}(\rho, \mathsf{ck}_i) \\ \mathsf{c} \leftarrow \mathsf{Enc}(\mathsf{k}, \mu) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda).$$

We present an experiment to test a notion we call *indistinguishability against adaptive chosen key and ciphertext attack* (IND-qCKCA2) which strengthens the well-known indistinguishability against adaptive chosen ciphertext attack (IND-qCCA2) [18, 6]. In our experiment, the adversary can receive a polynomial number of public-key copies and has quantum access to the encryption and decryption oracles (except on the challenge ciphertext) as in the standard experiment. However, in our case, the adversary is additionally given full control over the distribution of the quantum public-keys and can choose the quantum key used for generating the challenge ciphertext. The certification key is assumed to be securely announced and can be used for validating the quantum public-key.

Experiment $\mathsf{QPKE}^{\text{IND-qCKCA2}}_{\Pi,\mathcal{A}}(\lambda, j)$:

1. Experiment samples classical secret $\mathsf{sk} \leftarrow \mathsf{KeyGen}(1^\lambda)$, a bit $b \leftarrow \{0,1\}$, and creates an empty list $\mathcal{C}$.
2. For each $i \leq j$, experiment generates $\mathsf{ck}_i \leftarrow \mathsf{ckAnnounce}(\mathsf{sk}, t_{i+1})$.
3. $\tilde{\rho} \leftarrow \mathcal{A}^{\mathsf{pkSend}(\mathsf{sk})}((\mathsf{ck}_i)_{i<j})$.
4. Experiment extracts encryption key $\mathsf{k} \leftarrow \mathsf{ekReceive}(\tilde{\rho}, \mathsf{ck}_j)$.
5. $\mathcal{A}$ is allowed to make 3 types of queries:
   (a) **Challenge queries:** $\mathcal{A}$ sends two messages m_0, m_1 and experiment responds with $\mathsf{c} \leftarrow \mathsf{Enc}(\mathsf{k}, m_b)$. Experiment adds c to $\mathcal{C}$.
   (b) **Encryption queries:** Experiment chooses randomness r and encrypt messages in a superposition using r as randomness:
   $$\sum_{s,m,c} \alpha_{s,m,c}|s\rangle|m,c\rangle \rightarrow \sum_{s,m,c} \alpha_{s,m,c}|s\rangle|m, c \oplus \mathsf{Enc}(\mathsf{k}, m; r)\rangle$$
   (c) **Decryption queries:** Experiment decrypts all ciphertexts in superposition except the challenge ciphertexts:
   $$\sum_{s,m,c} \alpha_{s,m,c}|s\rangle|c,m\rangle \rightarrow \sum_{s,m,c} \alpha_{s,c,m}|s\rangle|c, m \oplus f(c)\rangle$$
   where
   $$f(c) = \begin{cases} \mathsf{Dec}(\mathsf{sk}, c) & c \notin \mathcal{C} \\ \perp & c \in \mathcal{C}. \end{cases}$$
6. $\mathcal{A}$ outputs a guess b'.
7. The output of the experiment is 1 if $b = b'$ and 0 otherwise.

Definition 29 (Security). *A QPKE scheme Π is IND-qCKCA2 if for any QPT adversary $\mathcal{A}$,*

$$\Pr\left[\mathit{QPKE}^{\mathit{IND\text{-}qCKCA2}}_{\Pi,\mathcal{A}}(\lambda) = 1\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

We can also define weaker notions such as IND-qCKPA and IND-qCKCA1 security in a natural way by restricting access to the decryption oracle. We can also weaken the security definition by allowing only classical oracle access to the encryption and decryption functionalities. We denote the resulting notions by IND-CKCA2, IND-CKCA1, and IND-CKPA.

8.2 Construction with Mixed-State Keys

We now construct a qCKCA2-secure QPKE scheme with an unrestricted distribution of mixed-state keys. The quantum public-key is a state encoding an

evaluation of a pq-PRF and is signed using a TD-DVK signature scheme. Each user can learn a random evaluation of the pq-PRF which can be used as a secret encryption key. Messages are then encrypted under any qCCA2-secure symmetric encryption scheme. Recall that the existence of pq-OWFs implies the existence of a pq-PRFs and a qCCA2-secure symmetric encryption scheme (Lemma 3).

Construction 7. *Let $\Pi_{\mathrm{SE}} = (\mathsf{SE.Gen}, \mathsf{SE.Enc}, \mathsf{SE.Dec})$ be the algorithms for an IND-qCCA2-secure symmetric encryption scheme on n-bit messages such that key generation makes $p < \lambda/2$ coin tosses. Let $\boldsymbol{H}_{\lambda,p}$ be a two-universal class of hash functions and let m denote its size. Let $F : \{0,1\}^\lambda \times \{0,1\}^n \to \{0,1\}^\lambda$, $F' : \{0,1\}^\lambda \times \{0,1\}^\lambda \to \{0,1\}^\lambda$, $F'' : \{0,1\}^\lambda \times \{0,1\}^n \to \{+,\times\}^{2\lambda}$ and $F''' : \{0,1\}^\lambda \times \{0,1\}^n \to \{0,1\}^{\lg m}$ be pq-PRFs. The construction for QPKE on n-bit messages is as follows:*

- $\mathsf{KeyGen}(1^\lambda)$: *Sample $\mathsf{k}, \mathsf{k}', \mathsf{k}'', \mathsf{k}''' \leftarrow \{0,1\}^\lambda$ and output $\mathsf{sk} := (\mathsf{k}, \mathsf{k}', \mathsf{k}'', \mathsf{k}''')$.*
- $\mathsf{pkSend}(\mathsf{sk}, \mathsf{t})$: *Let $i \in \mathbb{N}$ be such that $\mathsf{t} \in [t_i, t_{i+1})$.*
  1. *Set $\mathsf{k}'_i := F(\mathsf{k}', i)$, $\mathsf{k}''_i := F(\mathsf{k}'', i)$ and choose hash function H_i from the set $\boldsymbol{H}_{\lambda,p}$ based on $\mathsf{k}'''_i := F'''(\mathsf{k}''', i)$. (Note for the final step, k'''_i is a string of length $\lg(m)$ and thus can be interpreted to encode a choice of hash function from the set $\boldsymbol{H}_{\lambda,p}$.)*
  2. *Choose $x \leftarrow \{0,1\}^n$ and let $y_x := F(\mathsf{k}, x)$, $s_{x,i} := F'(\mathsf{k}'_i, y_x)$, and $\theta_{x,i} := F''(\mathsf{k}''_i, x)$.*
  3. *Output $|\mathsf{pk}_{i,x}\rangle := |x\rangle \otimes |y_x \| s_{x,i}\rangle_{\theta_{x,i}}$.*
- $\mathsf{ckAnnounce}(\mathsf{sk}, \mathsf{t})$: *If $\mathsf{t} = t_{i+1}$ for some $i \in \mathbb{N}$, announce $\mathsf{ck}_i := (t_i, t_{i+1}, \mathsf{k}'_i, \mathsf{k}''_i, \mathsf{k}'''_i)$.*
- $\mathsf{ekReceive}(\rho, \mathsf{ck}_j)$:
  1. *Interpret ck_j as $(t_j, t_{j+1}, \mathsf{k}'_j, \mathsf{k}''_j, \mathsf{k}'''_j)$.*
  2. *Measure the first register of ρ in computational basis to obtain say v.*
  3. *Measure the second register in basis $F'''(\mathsf{k}''_j, v)$ to obtain say (y, s).*
  4. *Determine H_j from k'''_j.*
  5. *Compute $\mathsf{ek}_v := \mathsf{SE.Gen}(1^\lambda; H_j(y_v))$.*
  6. *If $s = F'(\mathsf{k}'_j, y)$, then output $\mathsf{k}_v := (v, j, \mathsf{ek}_v)$.*
  7. *Otherwise, output $\perp$.*
- $\mathsf{Enc}(\mathsf{k}_v, m)$:
  1. *If $\mathsf{k}_v = \perp$, then output $\perp$.*
  2. *Interpret k_v as (v, j, ek_v).*
  3. *Output $(v, j, \mathsf{SE.Enc}(\mathsf{ek}_v, m))$.*
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$:
  1. *Interpret ct as (v, j, c).*
  2. *If $c = \perp$, then output $\perp$.*
  3. *Otherwise, compute $F(\mathsf{k}, v) = y_v$ and $F'''(\mathsf{k}''', j)$ to determine H_j.*
  4. *Compute $\mathsf{ek}_v = \mathsf{SE.Gen}(1^\lambda; H_j(y_v))$.*
  5. *Output $\mathsf{SE.Dec}(\mathsf{ek}_v, c)$.*

Theorem 10. *Construction 7 is a qCKCA2-secure QPKE scheme assuming the existence of a qCCA2-secure symmetric encryption and pq-PRFs.*

Proof. We prove security through a sequence of hybrids. Hybrid H_0 is the standard experiment adapted to our construction:

Hybrid H_0 (standard experiment):

1. Experiment samples classical secret and certification key $\mathsf{sk} \leftarrow \mathsf{KeyGen}(1^\lambda)$, a bit $b \leftarrow \{0,1\}$, and creates an empty list $\mathcal{C}$.
2. For each $i \leq j$, experiment generates $\mathsf{ck}_i \leftarrow \mathsf{ckAnnounce}(\mathsf{sk}, t_{i+1})$.
3. $\tilde{\rho} \leftarrow \mathcal{A}^{\mathsf{pkSend}(\mathsf{sk})}((\mathsf{ck}_i)_{i<j})$. Let $(|\mathsf{pk}_{n_i,x_i}\rangle)_{i\in p}$ denote the public-key copies received.
4. Experiment extracts encryption key $\mathsf{k} \leftarrow \mathsf{ekReceive}(\tilde{\rho}, \mathsf{ck}_j)$.
5. $\mathcal{A}$ is allowed to make 3 types of queries:
   (a) **Challenge queries:** $\mathcal{A}$ sends two messages m_0, m_1 and experiment responds with $\mathsf{c} \leftarrow \mathsf{Enc}(\mathsf{k}, m_b)$. Experiment adds c to $\mathcal{C}$.
   (b) **Encryption queries:** Experiment chooses randomness r and encrypt messages in a superposition using r as randomness:

$$\sum_{s,m,c} \alpha_{s,m,c}|s\rangle|m,c\rangle \to \sum_{s,m,c} \alpha_{s,m,c}|s\rangle|m, c \oplus \mathsf{Enc}(\mathsf{k}, m; r)\rangle$$

   (c) **Decryption queries:** Experiment decrypts all ciphertexts in superposition except the challenge ciphertexts:

$$\sum_{s,m,c} \alpha_{s,m,c}|s\rangle|c,m\rangle \to \sum_{s,m,c} \alpha_{s,c,m}|s\rangle|c, m \oplus f(c)\rangle$$

   where

$$f(c) = \begin{cases} \mathsf{Dec}(\mathsf{sk}, c) & \mathsf{c} \notin \mathcal{C} \\ \bot & \mathsf{c} \in \mathcal{C}. \end{cases}$$

6. $\mathcal{A}$ outputs a guess b'.
7. The output of the experiment is 1 if $b = b'$ and 0 otherwise.

Hybrid H_1: This is the same as H_0 except $F(\mathsf{k}, \cdot)$, $F(\mathsf{k}', \cdot)$, $F(\mathsf{k}'', \cdot)$ and $F'''(\mathsf{k}''', \cdot)$ are replaced with completely random functions with the same domain and co-domain.

Hybrid H_2: This hybrid is the same as H_1 except the functions $F'(\mathsf{k}'_j, \cdot)$ and $F''(\mathsf{k}''_j, \cdot)$ are replaced with completely random functions R' and R'' on the same domain and co-domain.

Claim. For any QPT adversary,

$$\Pr[\mathsf{H}_2 = 1] \leq \mathsf{negl}(\lambda).$$

Proof. Assume there exists a QPT adversary $\mathcal{A}$ such that $\Pr[\mathsf{H}_2 = 1]$ is non-negligible. In H_2, the adversary gets public-key copies $(|\mathsf{pk}_{n_i,x_i}\rangle)_{i\in p}$ and outputs $\tilde{\rho}$. The experiment then extracts a subkey $\mathsf{k} \leftarrow \mathsf{ekReceive}(\tilde{\rho}, \mathsf{ck}_j)$ as follows. It measures the first register of $\tilde{\rho}$ in the computational basis to obtain an outcome, which we denote as $\tilde{x}$. Then, it measures the second register in basis $R''(\tilde{x})$ to obtain an outcome, which we denote $(\tilde{y}, \tilde{s})$.

The only way that $\mathsf{k} \neq \perp$ is if $\tilde{s} = R'(\tilde{y})$. Since R' is a random function, (whp) this can only occur if one of the key copies $(|\mathsf{pk}_{n_i,x_i}\rangle)_{i\in p}$ contains the value $R'(\tilde{y})$. Note also that this must be a key generated after time t_j as R' is only used in this case.

Assume, without loss of generality, that the first public-key $|\mathsf{pk}_{n_0,x_0}\rangle = |x_0\rangle \otimes |y_0\|R'(y_0)\rangle_{R''(x_0)}$ satisfies $y_0 = \tilde{y}$ and $n_0 = j$. In this hybrid, y_0 is generated uniformly at random. So (whp) $|\mathsf{pk}_{j,x_0}\rangle$ is the only key that contains any information regarding y_0.

We are now in the same setting as the monogamy of entanglement game (see Sec. 3.5). Recall, in the monogamy of entanglement game, Alice chooses a random binary string and sends it to Bob and Charlie in a random BB84 basis. In this case, Alice is the part of the experiment that produces the public-keys and sends $|y_0\rangle_{R''(x_0)}$, Bob is the adversary and Charlie is the part of the experiment that runs $\mathsf{ekReceive}(\tilde{\rho}, \mathsf{ck}_j)$. Since the second experiment (Charlie) learns y_0, monogamy of entanglement implies that $\mathcal{A}$ must be uncertain about y_0.

Mathematically, if Y_0 is the random variable which represents the distribution of y_0 from the perspective of $\mathcal{A}$, then Theorem 2 implies that $H_\infty(Y_0) \geq \frac{\lambda}{5}$. Then, Privacy Amplification Theorem 1 says that $H_j(Y_0)$ is indistinguishable from random, given that H_j is sampled using a random function. Therefore, $\mathcal{A}$ cannot distinguish the key used for encryption from random but can distinguish the bit b from random. As a result, $\mathcal{A}$ can, with standard reductions, be converted into a QPT adversary that breaks the IND-qCCA2 security of Π_{QAE}, giving a contradiction. ■

Claim. No QPT adversary can distinguish between hybrids H_2 and H_1 with non-negligible advantage.

Proof. The only difference between these hybrids is that the functions $F'(\mathsf{k}'_j, \cdot)$ and $F''(\mathsf{k}''_j, \cdot)$ in H_1 are replaced with completely random functions R' and R'' in H_2. Note that in H_1, the keys k'_j and k''_j are generated using a random function and hence are indistinguishable from random with respect to any adversary. Therefore, by the security of $\mathsf{pq\text{-}PRFs}$, no adversary can distinguish between these hybrids. ■

Claim. No QPT adversary can distinguish between hybrids H_1 and H_0 with non-negligible advantage.

Proof. The only difference between these hybrids is that $F(\mathsf{k}, \cdot)$, $F(\mathsf{k}', \cdot)$, $F(\mathsf{k}'', \cdot)$ and $F'''(\mathsf{k}''', \cdot)$ in H_0 are replaced with completely random functions in H_1. Given that the keys $\mathsf{k}, \mathsf{k}', \mathsf{k}'', \mathsf{k}'''$ are sampled uniformly at random, no adversary can distinguish between these two hybrids by the security of $\mathsf{pq\text{-}PRFs}$. ■

To sum up, for any QPT adversary, the probability that H_2 outputs 1 is negligible, no adversary can distinguish between H_2 and H_1 with non-negligible advantage, and no adversary can distinguish between H_1 and H_0 with non-negligible advantage. Therefore, $\Pr[\mathsf{H}_0 = 1] \leq \mathsf{negl}(\lambda)$. ■

Note that the public-keys in Construction 7 are mixed-state. Other QPKE constructions also rely on mixed-state keys [37, 11], however, it might be desirable to have pure-state keys. Unfortunately, it is not possible to achieve publicly verifiable pure-state public-keys with unrestricted distribution. This is because the impossibility result [4] on standard quantum signatures implies that we cannot sign the quantum public-keys with a time-independent verification key in the standard model. Hence, the verification key must change which implies the signed states must also change. However, this issue can be solved by distributing keys only at the beginning of the protocol as shown in the next section.

8.3 Construction with Pure-State Keys

In this section, we construct a CKCPA-secure QPKE scheme from PRFSs with pure-state public-keys distributed only at the start of the protocol. The idea is to apply the one-time TD-DVK signature scheme (Construction 2) discussed in Sec. 5.2 to sign the quantum public-keys. Once the certification key is revealed, allowing users to validate the public-keys, new public-keys can no longer be distributed securely so the distribution phase terminates. As the one-time signature scheme is information-theoretically secure, this approach can be applied to any QPKE scheme with no extra assumptions.

Our construction is inspired by the QPKE scheme presented in [29] which is built from PRFSs. Their scheme does not address attacks that tamper with public-keys which is where our contribution is made. Recall, PRFSs are potentially a weaker assumption than pq-OWFs [39].

Unlike the previous construction, users extract a quantum encryption key from the public-key and each encryption requires a new public-key copy. Furthermore, the scheme is defined only for 1-bit messages, however, it can easily be generalized to encrypt a string of bits by encrypting them one by one.

Construction 8. *Let $\{\psi_{k,x}\}_{k,x}$ be a PRFS family with super-logarithmic input-size and let $\Pi_{\mathrm{1QAE}} = (\mathit{1QGen}, \mathit{1QEnc}, \mathit{1QDec})$ be the algorithms from a one-time QAE scheme. Let t_{end} be the time that the key distribution ends. The construction for QPKE is as follows:*

- $\mathit{KeyGen}(1^\lambda)$: *Sample $k \leftarrow \{0,1\}^\lambda$ and $k' \leftarrow \mathit{1QGen}(1^\lambda)$. Output $\mathsf{sk} \coloneqq (k, k')$.*
- $\mathit{pkSend}(\mathsf{sk}, t)$: *If $t \leq t_{\mathrm{end}}$, then output $|\mathsf{pk}\rangle \coloneqq \mathit{1QEnc}_{k'}(\frac{1}{\sqrt{2^\lambda}} \sum_{x\in\{0,1\}^\lambda} |x\rangle|\psi_{k,x}\rangle)$. Otherwise, output $\bot$.*
- $\mathit{ckAnnounce}(\mathsf{sk}, t)$: *If $t = t_{\mathrm{end}}$, then output $\mathit{ck} \coloneqq (t_{\mathrm{end}}, k')$. Otherwise, output $\bot$.*
- $\mathit{ekReceive}(\rho, \mathit{ck}, t)$: *Interpret ck as (t_{end}, k'). If ρ is received after t_{end}, then output $\bot$. Otherwise, output $\gamma \coloneqq \mathit{1QDec}_{k'}(\rho)$.*
- $\mathit{Enc}(\gamma, m)$:
  1. *Measure the first register of γ in the computational basis and let x denote the output. Let ψ be the state in the second register.*
  2. *If $m = 0$, then let $\phi \coloneqq \psi$ and if $m = 1$, then let ϕ be the maximally mixed state.*
  3. *Output $\sigma \coloneqq (x, \phi)$.*
- $\mathit{Dec}(\mathsf{sk}, \sigma)$:
  1. *Interpret σ as (x, ϕ).*

2. If $\phi = |\psi_{k,x}\rangle$, then output 0. Otherwise, output 1.

Theorem 11. *Construction 8 is a CKPA-secure QPKE scheme assuming the existence of PRFS with super-logarithmic input size.*

Proof. We first introduce the following hybrid for $y \in \{0,1\}^\lambda$:

$\mathbf{H}_y$: This is the same as the standard experiment except the public-keys are generated differently. The experiment chooses a random Haar state $|\phi^{\mathrm{H}_y}\rangle$ and sends public-keys of the form $|\mathsf{pk}^{\mathrm{H}_y}\rangle \coloneqq \mathsf{1QEnc}_{k'}(\frac{1}{\sqrt{2^\lambda}}\sum_{x\neq y}(|x\rangle|\psi_{k,x}\rangle)+|y\rangle|\phi^{\mathrm{H}_y}\rangle)$ whenever requested.

For any $y \in \{0,1\}^\lambda$, the only difference between $|\mathsf{pk}^{\mathrm{H}_y}\rangle$ and $|\mathsf{pk}\rangle$ is that the state $|\psi_{k,y}\rangle$ in the public-key is replaced with $|\phi^{\mathrm{H}_y}\rangle$. Since $|\psi_{k,y}\rangle$ is indistinguishable from a random Haar state (by the security of PRFS), $\mathcal{A}$ cannot distinguish between the public-keys in $\mathbf{H}_y$ and in the standard experiment.

In the standard security experiment, $\mathcal{A}$ submits a forged key, denoted by $\tilde{\rho}$, which is used by the experiment to generate the challenge ciphertext. By the security one-time authentication scheme Π_{1QAE} (Lemma 6), (whp) either $\tilde{\rho} \approx_{\mathsf{negl}(\lambda)} |\mathsf{pk}\rangle$ or the key extracted by the experiment $\mathsf{ekReceive}(\tilde{\rho}, \mathsf{ck})$ is $\perp$. Only the first case needs to be studied since in the second case, the output of any encryption is $\perp$ so $\mathcal{A}$ cannot distinguish between two ciphertexts.

Upon receiving $\tilde{\rho}$, the experiment measures the first register to obtain outcome x' and the second register collapses to a state $\tilde{\psi}_{x'}$. Since $\tilde{\rho} \approx_{\mathsf{negl}(\lambda)} |\mathsf{pk}\rangle$, (whp) we have $\tilde{\psi}_{x'} \approx_{\mathsf{negl}(\lambda)} |\psi_{k,x'}\rangle$. Typically $\mathcal{A}$ cannot distinguish $|\psi_{k,x'}\rangle$ from a random Haar state; however, $\tilde{\psi}_{x'}$ may be entangled with the state stored by $\mathcal{A}$ which would allow $\mathcal{A}$ to distinguish it from random.

Let R be the state stored by $\mathcal{A}$ when it sends $\tilde{\rho}$ to the experiment and, similarly, let $R^{\mathrm{H}_{x'}}$ denote the state stored by $\mathcal{A}$ if it was in hybrid $\mathbf{H}_{x'}$. Note that R only depends on the adversary and the public-keys since it is produced prior to the encryption of any messages. But $|\mathsf{pk}^{\mathrm{H}_{x'}}\rangle \approx_{\mathsf{negl}(\lambda)} |\mathsf{pk}\rangle$, which implies that $R^{\mathrm{H}_{x'}} \approx_{\mathsf{negl}(\lambda)} R$. As a result, there exists negligible ϵ such that the state $|\phi^{\mathrm{H}_{x'}}\rangle$ is ϵ-independent of $R^{\mathrm{H}_{x'}}$ as $R^{\mathrm{H}_{x'}} \approx_{\mathsf{negl}(\lambda)} R$ and R is clearly independent of $|\phi^{\mathrm{H}_{x'}}\rangle$.

In hybrid $\mathbf{H}_{x'}$, $\mathcal{A}$ will receive a state close to $|\phi^{\mathrm{H}_{x'}}\rangle$ for the encryption of 0 and a random Haar state for the encryption of 1. Since $|\phi^{\mathrm{H}_{x'}}\rangle$ is ϵ-independent of $R^{\mathrm{H}_{x'}}$, $\mathcal{A}$ does not learn information regarding $|\phi^{\mathrm{H}_{x'}}\rangle$ from $R^{\mathrm{H}_{x'}}$. Hence, $\mathcal{A}$ cannot distinguish between $|\phi^{\mathrm{H}_{x'}}\rangle$ and a random Haar state. In other words, $\mathcal{A}$ cannot distinguish between the encryption of 0 and 1 in $\mathbf{H}_{x'}$. As $\mathcal{A}$ cannot distinguish between the hybrid and the standard experiment, it cannot distinguish the encryptions in the standard experiment as well. ■

References

[1] Scott Aaronson, Yosi Atia, and Leonard Susskind. "On the hardness of detecting macroscopic superpositions". In: *arXiv preprint arXiv:2009.07450* (2020). DOI: 10.48550/arXiv.2009.07450.

[2] Scott Aaronson and Paul Christiano. "Quantum money from hidden subspaces". In: *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*. 2012, pp. 41–60. DOI: 10.1145/2213977.2213983.

[3] Gorjan Alagic, Anne Broadbent, Bill Fefferman, Tommaso Gagliardoni, Christian Schaffner, and Michael St. Jules. "Computational security of quantum encryption". In: *Information Theoretic Security: 9th International Conference, ICITS 2016, Tacoma, WA, USA, August 9-12, 2016, Revised Selected Papers 9*. Springer. 2016, pp. 47–71. DOI: 10.1007/978-3-319-49175-2_3.

[4] Gorjan Alagic, Tommaso Gagliardoni, and Christian Majenz. "Can you sign a quantum state?" In: *Quantum* 5 (2021), p. 603. URL: https://eprint.iacr.org/2018/1164.

[5] Gorjan Alagic, Tommaso Gagliardoni, and Christian Majenz. "Unforgeable quantum encryption". In: *Advances in Cryptology–EUROCRYPT 2018: 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29-May 3, 2018 Proceedings, Part III*. Springer. 2018, pp. 489–519. DOI: 10.1007/978-3-319-78372-7_16.

[6] Gorjan Alagic, Stacey Jeffery, Maris Ozols, and Alexander Poremba. "On quantum chosen-ciphertext attacks and learning with errors". In: *Cryptography* 4.1 (2020), p. 10. DOI: 10.3390/cryptography4010010.

[7] Prabhanjan Ananth, Aditya Gulati, Luowen Qian, and Henry Yuen. "Pseudorandom (Function-Like) Quantum State Generators: New Definitions and Applications". In: *Theory of Cryptography: 20th International Conference, TCC 2022, Chicago, IL, USA, November 7–10, 2022, Proceedings, Part I*. Springer. 2022, pp. 237–265. DOI: 10.1007/978-3-031-22318-1_9.

[8] Prabhanjan Ananth, Zihan Hu, and Henry Yuen. "On the (Im) plausibility of Public-Key Quantum Money from Collision-Resistant Hash Functions". In: *arXiv preprint arXiv:2301.09236* (2023). DOI: 10.48550/arXiv.2301.09236.

[9] Prabhanjan Ananth, Aayush Jain, Huijia Lin, Christian Matt, and Amit Sahai. "Indistinguishability obfuscation without multilinear maps: new paradigms via low degree weak pseudorandomness and security amplification". In: *Advances in Cryptology–CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2019, Proceedings, Part III*. Springer. 2019, pp. 284–332. DOI: 10.1007/978-3-030-26954-8_10.

[10] Prabhanjan Ananth, Luowen Qian, and Henry Yuen. "Cryptography from pseudorandom quantum states". In: *Advances in Cryptology–CRYPTO 2022: 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15–18, 2022, Proceedings, Part I*. Springer. 2022, pp. 208–236. DOI: 10.1007/978-3-031-15802-5_8.

[11] Mohammed Barhoush and Louis Salvail. "Powerful Primitives in the Bounded Quantum Storage Model". In: *arXiv preprint arXiv:2302.05724* (2023). DOI: 10.48550/arXiv.2302.05724.

[12] Howard Barnum, Claude Crépeau, Daniel Gottesman, Adam Smith, and Alain Tapp. "Authentication of quantum messages". In: *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.* IEEE. 2002, pp. 449–458. DOI: 10.1109/SFCS.2002.1181969.

[13] Mihir Bellare and Phillip Rogaway. "Random oracles are practical: A paradigm for designing efficient protocols". In: *Proceedings of the 1st ACM Conference on Computer and Communications Security*. 1993, pp. 62–73. DOI: 10.1145/168588.168596.

[14] Charles H. Bennett and Gilles Brassard. "Quantum cryptography: Public key distribution and coin tossing". In: *Theoretical Computer Science* 560 (2014). Original work published 1984, pp. 7–11. DOI: 10.1016/j.tcs.2014.05.025.

[15] Andriyan Bilyk, Javad Doliskani, and Zhiyong Gong. "Cryptanalysis of three quantum money schemes". In: *Quantum Information Processing* 22.4 (2023), p. 177. DOI: 10.1007/s11128-023-03919-0.

[16] Nir Bitansky, Shafi Goldwasser, Abhishek Jain, Omer Paneth, Vinod Vaikuntanathan, and Brent Waters. "Time-lock puzzles from randomized encodings". In: *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*. 2016, pp. 345–356. DOI: 10.1145/2840728.2840745.

[17] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. "Random oracles in a quantum world". In: *Advances in Cryptology–ASIACRYPT 2011: 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings 17*. Springer. 2011, pp. 41–69. DOI: 10.1007/978-3-642-25385-0_3.

[18] Dan Boneh and Mark Zhandry. "Secure signatures and chosen ciphertext security in a quantum computing world". In: *Advances in Cryptology–CRYPTO 2013: 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II*. Springer. 2013, pp. 361–379. DOI: 10.1007/978-3-642-40084-1_21.

[19] Tore Vincent Carstens, Ehsan Ebrahimi, Gelo Noel Tabia, and Dominique Unruh. "Relationships between quantum IND-CPA notions". In: *Theory of Cryptography: 19th International Conference, TCC 2021, Raleigh, NC, USA, November 8–11, 2021, Proceedings, Part I*. Springer. 2021, pp. 240–272. DOI: 10.1007/978-3-030-90459-3_9.

[20] Céline Chevalier, Ehsan Ebrahimi, and Quoc-Huy Vu. "On security notions for encryption in a quantum world". In: *Progress in Cryptology–INDOCRYPT 2022: 23rd International Conference on Cryptology in India, Kolkata, India, December 11–14, 2022, Proceedings*. Springer. 2023, pp. 592–613. DOI: 10.1007/978-3-031-22912-1_26.

[21] Kai-Min Chung, Serge Fehr, Yu-Hsuan Huang, and Tai-Ning Liao. "On the compressed-oracle technique, and post-quantum security of proofs of sequential work". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2021, pp. 598–629. DOI: 10.1007/978-3-030-77886-6_21.

[22] Andrea Coladangelo. "Quantum trapdoor functions from classical one-way functions". In: *arXiv preprint arXiv:2302.12821* (2023). DOI: 10.48550/arXiv.2302.12821.

[23] Ivan Damgård, Serge Fehr, Louis Salvail, and Christian Schaffner. "Cryptography in the Bounded-Quantum-Storage Model". In: *SIAM Journal on Computing* 37.6 (2008), pp. 1865–1890. ISSN: 0097-5397. DOI: 10.1137/060651343. arXiv: quant-ph/0508222.

[24] Yevgeniy Dodis, Willy Quach, and Daniel Wichs. "Authentication in the Bounded Storage Model". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2022, pp. 737–766. DOI: 10.1007/978-3-031-07082-2_26.

[25] Edward Farhi, David Gosset, Avinatan Hassidim, Andrew Lutomirski, and Peter Shor. "Quantum money from knots". In: *Proceedings of the 3rd Innovations in*

Theoretical Computer Science Conference. 2012, pp. 276–289. DOI: 10.1145/2090236.2090260.

[26] Christopher A Fuchs and Jeroen Van De Graaf. "Cryptographic distinguishability measures for quantum-mechanical states". In: *IEEE Transactions on Information Theory* 45.4 (1999), pp. 1216–1227. DOI: 10.1109/18.761271.

[27] Romain Gay and Rafael Pass. "Indistinguishability obfuscation from circular security". In: *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*. 2021, pp. 736–749. DOI: 10.1145/3406325.3451070.

[28] Daniel Gottesman and Isaac Chuang. "Quantum digital signatures". In: *arXiv preprint quant-ph/0105032* (2001). DOI: 10.48550/arXiv.quant-ph/0105032.

[29] Alex B Grilo, Or Sattath, and Quoc-Huy Vu. "Encryption with Quantum Public Keys". In: *arXiv preprint arXiv:2303.05368* (2023). DOI: 10.48550/arXiv.2303.05368.

[30] Sam Hopkins, Aayush Jain, and Huijia Lin. "Counterexamples to new circular security assumptions underlying iO". In: *Advances in Cryptology–CRYPTO 2021: 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16–20, 2021, Proceedings, Part II 41*. Springer. 2021, pp. 673–700. DOI: 10.1007/978-3-030-84245-1_23.

[31] Russell Impagliazzo and Steven Rudich. "Limits on the provable consequences of one-way permutations". In: *Proceedings of the twenty-first annual ACM symposium on Theory of computing*. 1989, pp. 44–61. DOI: 10.1145/73007.73012.

[32] Aayush Jain, Huijia Lin, and Amit Sahai. "Indistinguishability obfuscation from LPN over F_p, DLIN, and PRGs in NC_0". In: *Advances in Cryptology–EUROCRYPT 2022: 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30–June 3, 2022, Proceedings, Part I*. Springer. 2022, pp. 670–699. DOI: 10.1007/978-3-031-06944-4_23.

[33] Aayush Jain, Huijia Lin, and Amit Sahai. "Indistinguishability obfuscation from well-founded assumptions". In: *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*. 2021, pp. 60–73. DOI: 10.1145/3406325.3451093.

[34] Zhengfeng Ji, Yi-Kai Liu, and Fang Song. "Pseudorandom Quantum States". In: *Advances in Cryptology–CRYPTO 2018: 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19–23, 2018, Proceedings, Part III 38*. Springer. 2018, pp. 126–152. DOI: 10.1007/978-3-319-96878-0_5.

[35] Daniel M Kane, Shahed Sharif, and Alice Silverberg. "Quantum money from quaternion algebras". In: *arXiv preprint arXiv:2109.12643* (2021). DOI: 10.48550/arXiv.2109.12643.

[36] Andrey Boris Khesin, Jonathan Z Lu, and Peter W Shor. "Publicly verifiable quantum money from random lattices". In: *arXiv preprint arXiv:2207.13135* (2022). DOI: 10.48550/arXiv.2207.13135.

[37] Fuyuki Kitagawa, Tomoyuki Morimae, Ryo Nishimaki, and Takashi Yamakawa. "Quantum Public-Key Encryption with Tamper-Resilient Public Keys from One-Way Functions". In: (2023). DOI: 10.48550/arXiv.2304.01800.

[38] Robert Konig, Renato Renner, and Christian Schaffner. "The operational meaning of min-and max-entropy". In: *IEEE Transactions on Information theory* 55.9 (2009), pp. 4337–4347. DOI: 10.1109/TIT.2009.2025545.

[39] William Kretschmer. "Quantum Pseudorandomness and Classical Complexity". In: *16th Conference on the Theory of Quantum Computation, Communication and Cryptography*. 2021. DOI: 10.4230/LIPIcs.TQC.2021.2.

[40] Qipeng Liu. "Depth-Bounded Quantum Cryptography with Applications to One-Time Memory and More". In: *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik. 2023. DOI: 10.4230/LIPIcs.ITCS.2023.82.

[41] Tomoyuki Morimae, Alexander Poremba, and Takashi Yamakawa. "Revocable quantum digital signatures". In: *arXiv preprint arXiv:2312.13561* (2023). DOI: 10.48550/arXiv.2312.13561.

[42] Tomoyuki Morimae and Takashi Yamakawa. "Quantum commitments and signatures without one-way functions". In: *Advances in Cryptology–CRYPTO 2022: 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15–18, 2022, Proceedings, Part I*. Springer. 2022, pp. 269–295. DOI: 10.1007/978-3-031-15802-5_10.

[43] Roy Radian and Or Sattath. "Semi-quantum money". In: *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*. 2019, pp. 132–146. DOI: 10.1145/3318041.3355462.

[44] Renato Renner and Robert König. "Universally Composable Privacy Amplification Against Quantum Adversaries". In: *Theory of Cryptography*. Ed. by Joe Kilian. Vol. 3378. Lecture Notes in Computer Science. 2005, pp. 407–425. ISBN: 978-3-540-24573-5. DOI: 10.1007/978-3-540-30576-7_22. arXiv: quant-ph/0403133.

[45] Ronald L Rivest, Adi Shamir, and David A Wagner. "Time-lock puzzles and timed-release crypto". In: (1996).

[46] Bhaskar Roberts and Mark Zhandry. "Franchised quantum money". In: *Advances in Cryptology–ASIACRYPT 2021: 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6–10, 2021, Proceedings, Part I 27*. Springer. 2021, pp. 549–574. DOI: 10.1007/978-3-030-92062-3_19.

[47] John Rompel. "One-way functions are necessary and sufficient for secure signatures". In: *Proceedings of the twenty-second annual ACM symposium on Theory of computing*. 1990, pp. 387–394. DOI: 10.1145/100216.100269.

[48] Omri Shmueli. "Public-Key Quantum Money With a Classical Bank". In: *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*. 2022, pp. 790–803. DOI: 10.1145/3519935.3519952.

[49] Marco Tomamichel, Serge Fehr, Jedrzej Kaniewski, and Stephanie Wehner. "A monogamy-of-entanglement game with applications to device-independent quantum cryptography". In: *New Journal of Physics* 15.10 (2013), p. 103002. DOI: 10.1088/1367-2630/15/10/103002.

[50] Hoeteck Wee and Daniel Wichs. "Candidate obfuscation via oblivious LWE sampling". In: *Advances in Cryptology–EUROCRYPT 2021: 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17–21, 2021, Proceedings, Part III*. Springer. 2021, pp. 127–156. DOI: 10.1007/978-3-030-77883-5_5.

[51] Stephen Wiesner. "Conjugate coding". In: *ACM Sigact News* 15.1 (1983), pp. 78–88. DOI: 10.1145/1008908.1008920.

[52] Mark Zhandry. "How to construct quantum random functions". In: *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*. IEEE. 2012, pp. 679–687. DOI: 10.1109/FOCS.2012.37.

[53] Mark Zhandry. "Quantum lightning never strikes the same state twice. or: quantum money from cryptographic assumptions". In: *Journal of Cryptology* 34 (2021), pp. 1–56. DOI: 10.1007/s00145-020-09372-x.