UNIVERSITY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Raul-Martin Rebane

Post-Quantum Secure Time-Stamping

Master's Thesis (30 ECTS)

Supervisor: Dr Dominique Peer Ghislain Unruh

Tartu 2018

Post-Quantum Secure Time-Stamping

Abstract:
Cryptographic timestamps are used as proof that a certain document existed before another. Post-quantum secure time-stamping examines whether these proofs can be forged using a quantum computer. The field is very unexplored as the primitives used in keyless time-stamping have not shown any serious weakness towards quantum computers. Until now no effort had been made towards formally defining post-quantum secure time-stamping. In this work, we define the notion of post-quantum time-stamping and examine how contemporary classical results change in this new framework. A key difference in the post-quantum setting is that we cannot retrieve multiple separate executions of an arbitrary quantum adversary. Currently known rewinding techniques allow an adversary to be ran again only under very specific conditions. We examine the possibility of combining existing rewinding techniques to prove a theorem for which there is currently no proof in the standard post-quantum model. We conjecture a rewinding construction which could possibly prove the theorem and establish a minimal open problem for formally proving the theorem.

Keywords: Time-stamping, quantum computing, quantum cryptography

CERCS: P170 - Computer science, numerical analysis, systems, control

Post-kvant turvalised ajatempliprotokollid

Lühikokkuvõte:
Krüptograafilisi ajatempliprotokolle kasutatakse tõestusena, et üks dokument eksisteeris enne teist. Postkvantkrüptograafiliselt turvalised ajatempliprotokollid uurivad kas neid tõestusi on võimalik võltsida kasutades kvantarvuteid. Tegu on suuresti uurimata alaga, kuna võtmeta ajatempliprotokollides kasutatavates primitiivides pole seni leitud kvantarvutite kontekstis tõsiseid nõrkusi. Selles töös me defineerime mis on post-kvant turvalised ajatempliprotokollid ning uurime kuidas klassikalised tulemused muutuvad uues raamistikus. Suur erinevus kvantvastaste puhul on see, et meil ei ole võimalik saada suvalise kvantalgoritmi mitut erinevat käivitust. Tänapäeval teadaolevad tagasipööramise võtted võimaldavad kvantalgoritmi tagasi pöörata ainult väga kindlatel tingimustel. Me uurime nende võtete kombineerimise võimalikkust ühe teoreemi tõestamiseks. Sellele teoreemile ei ole hetkel post-kvant standardmudelis ühtegi tõestust. Me pakume tõestuseta ühe tagasipööramise konstruktsiooni, mille abil võib osutuda teoreemi tõestamine võimalikuks. Me lisaks pakume välja ka minimaalse lahendamata probleemi, mis on esimene samm teoreemi formaalse tõestamiseni.

Võtmesõnad: Ajatempel, kvantarvutus, kvantkrüptograafia

CERCS: P170 - Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine (automaatjuhtimisteooria)

Contents

1 Introduction

Cryptographically secure time-stamping mostly deals in the security of hash-based time-stamping protocols. As quantum computers have not yet achieved very strong attacks against hash functions, there is seemingly little concern for the security of hash-based time-stamping protocols in the quantum setting. Most recently, Buldas *et al.* noted that with some small differences, classical results can be carried over to the quantum setting [BLT17]. However, the notion of a secure time-stamping scheme in the post-quantum setting is currently undefined. As it has been shown in the case of commitments, there can be subtleties when dealing with quantum states and adversaries which can cause classical definitions to no longer capture their intended meaning. In our work, we lift the appropriate classical definitions into the quantum setting and explore the outcomes.

We begin with an overview of current classical time-stamping schemes and their cryptographical results. In Section 5 we define post-quantum security for time-stamping schemes and show that the definition of an unpredictable quantum adversary needs to differ significantly from the classical setting. In Section 6 we give a brief overview of contemporary quantum rewinding techniques. We do this with the aim to prove that collapsing hash functions are sufficient for polynomially bounded keyless time-stamping schemes. This is a theorem which holds in the classical setting, but for which there is no proof in the post-quantum standard model.

2 Cryptographic Background

We expect the reader to have some knowledge of cryptography and of quantum computing. This section is intended to be a reference guide if one comes across a term they do not recognize or remember. Additionally, all definitions, theorems and lemmata can be found in the Index section of the Appendix. All quoted definitions, theorems and lemmata appear in a shaded box with the proper accompanying citation, with the exception of Lemma 1 where we cite both the original article as well as the source for our notation.

As we base several of our definitions on the article by Buldas and Laur, we use the following from their Notation and Definitions section: [BL06].

Definition 1 (Probabilistic functions (FP))**.** *Let* FP *be the class of all probabilistic functions* $f : \{0,1\}^\star \leftarrow \{0,1\}^\star$ *computable by a polynomial-time Turing machine.*

Definition 2 (Collision (C))**.** *Let* C *denote that* (x, x') *is a collision for* h *if*

$$C(x, x') = [x \neq x' \wedge h(x) = h(x')]$$

Definition 3 (Collision-resistance)**.** *A function* $h \xleftarrow{\$} \mathfrak{F}$ *is Collision-resistant if* $\forall A \in \mathsf{FP}$:

$$Pr[(x, x') \leftarrow A(1^k, h) : C(x, x')] = k^{-\omega(1)}$$

By $\mathcal{U}_n$ we denote the uniform distribution on $\{0,1\}^n$. A distribution family $\{\mathcal{D}_\|\}_{k\in\mathbb{N}}$ is *poly-sampleable* if there exists a $\mathcal{D} \in \mathcal{FP}$ with output distribution $D(1^k)$ equal to $\mathcal{D}_k$.

Definition 4 (2nd Preimage Resistance)**.** *A function* $h \xleftarrow{\$} \mathfrak{F}$ *is 2nd Preimage Resistant if* $\forall A \in \mathsf{FP}$:

$$\Pr[x \leftarrow \mathcal{U}_\ell : x' \leftarrow A(x) : C(x, x')] = k^{-\omega(1)}$$

Definition 5 (One-Way hash function)**.** *A function* $h \xleftarrow{\$} \mathfrak{F}$ *is One-Way if* $\forall A \in \mathsf{FP}$:

$$\Pr[x \leftarrow \mathcal{U}_\ell : x' \leftarrow A(h(x)) : h(x) = h(x')] = k^{-\omega(1)}$$

The following notation is adopted from Dominique Unruh's Quantum Cryptology class notes [Unr17].

Definition 6 (Quantum states)**.** *An n-dimensional* quantum state *is represented by a vector* $|\Psi\rangle \in \mathbb{C}^n$ *with* $|||\Psi\rangle|| = 1$ *(here* $\mathbb{C}^n$ *is a Hilbert space).*

We abbreviate $x \otimes \ldots \otimes x$ (n components) as $x^{\otimes n}$ where $\otimes$ is the tensor product (or Kronecker product).

Definition 7 (Unitary matrices)**.** *A matrix $M \in \mathbb{C}^{n \times n}$ is unitary if $M^\dagger M = M M^\dagger = I$ where I is the identity matrix.*

Definition 8 (Hadamard)**.** *The* Hadamard *gate (usually denoted H) is defined by*

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

or equivalently

$$H|0\rangle = \frac{1}{\sqrt{2}}(|1\rangle + |0\rangle)$$
$$H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

Definition 9 (Unitary transformation)**.** *A unitary transformation on a quantum state $|\Psi\rangle \in \mathbb{C}^n$ is represented by a unitary matrix $U \in \mathbb{C}^{n \times n}$. The state after the transformation is $U|\Psi\rangle$.*

Definition 10 (Measurement)**.** *A* (projective) measurement *on a Hilbert space $\mathcal{H}$ is specified by a family $\{P_i\}_{i \in I}$ of orthogonal projections on $\mathcal{H}$ labelled with the possible measurement outcomes $i \in I$. The projections have to be pairwise orthogonal, i.e. $P_i P_j = 0$ for $i \neq j$. And the projections sum to 1, i.e. $\sum_i P_i = 1_{\mathcal{H}}$ where $1_{\mathcal{H}}$ is the identity on $\mathcal{H}$.*

When measuring a state $|\Psi\rangle \in \mathcal{H}$, the outcome i occurs with probability

$$||P_i|\Psi\rangle||^2$$

If the outcome i occurs, the state after the measurement (post-measurement state) is

$$\frac{P_i|\Psi\rangle}{||P_i|\Psi\rangle||}$$

3 Time-Stamping Protocols

3.1 Introduction

A timestamp is used to note that a piece of data existed at a specific point in time. This has many real life applications - consider a simplified scenario where someone has a timestamped piece of media, where the timestamp predates the public release of this piece of media. That person could then claim it as proof that they are the original author of the media. However, in order for this proof to be convincing, it must be difficult to forge. While most operating systems keep track of file creation time, those timestamps would not be much help in a copyright dispute, as they can be easily forged by changing the local clock.

Trusted time stamps are used to establish relative temporal authenticity, as absolute time stamps are incredibly complicated if not impossible [BLLV98]. Relative temporal authenticity allows the verifier to correctly identify which of the two time stamped documents was signed earlier. In this section we outline different types of timestamping schemes and their properties. This section is optional for those with previous knowledge of *hash-and-publish* time-stamping.

3.2 PKI-Based Time-Stamping

A PKI-based time-stamping scheme relies on a completely trusted third party (a TSS - Time stamping service), which distributes timestamps that are signed with the trusted party's digital signature. The benefits of this approach are its simplicity, fast verification of timestamps, and security links to already existing primitives. However, it requires unconditional trust in the third-party TSS and the continuous secrecy of private keys.

In a PKI-based time-stamping scheme, a client who wants a timestamp s of a document X provides the TSS with $x = H(X)$ where H is a hash function. The server then generates the current timestamp t using a trusted time source and creates the combined hash $c \leftarrow h(x||t)$. The TSS then signs c using the digital signing function $\sigma = Sign(sk, c)$ and then returns $s = (\sigma, t)$ to the client.

$Client(X)$
- $s \leftarrow TSS(H(x))$
- **return** s

$TSS(x)$
- $t \leftarrow$ Device
- $c \leftarrow h(x||t)$
- $\sigma \leftarrow$ Sign(sk, c)
- **return** (σ, t)

For verification, a similar process is carried out locally - the file is hashed using

the hash function used on the client side. The timestamp generated by the device is extracted from the TSS-issued timestamp and concatenated to the hash. This is hashed together using the TSS-side hash function as the server did earlier, resulting in c. Then the digital signature verification procedure is called to ensure that this was indeed the signed document.

$$
\begin{aligned}
&\text{Verify}(X, s, pk)\\
&\left[\begin{array}{l}
x \leftarrow H(X)\\
(\sigma, t) \leftarrow s\\
c \leftarrow h(x||t)\\
\textbf{return } \text{Verify}(pk, c, \sigma)
\end{array}\right.
\end{aligned}
$$

Many commercial timestamping services use PKI-based time-stamping, confirming to the RFC-5816 standard [SxSPT10]. The security of PKI-based timestamping schemes in the post-quantum setting has been studied by Clupek *et al* [CMZ15]. As a result, this paper focuses on other timestamping schemes.

3.3 Hash-Chain Time-Stamping

In hash-chain time-stamping schemes, sometimes referred to as *hash-and-publish* schemes, data is hashed and linked together using public cryptographic hash functions in some sort of data structure. The root of this data structure is then published in a widely available piece of public media, such as a newspaper, and a certificate of the timestamp are the necessary pieces of data required to reach a published root node.

In contrast to PKI-based time-stamping, hash-chain timestamping only requires the existence of hash functions with certain properties, and does not rely on the secrecy of any private key. However, as the certificate for a piece of data has to contain everything needed to reach the published root node, the size of the certificate is dependent on the number of issued certificates. Different data structures can be used to achieve better certificate sizes with subtle differences in security.

3.3.1 Linked List Schemes

The first hash-chain time-stamping scheme proposed by Haber and Stornetta was based on the linked list data structure [HS91]. The scheme relies on the fact that if a certificate includes the hash of another issued timestamp, it must have been issued after it, as the hash of the timestamp could not be known in advance. Thus by always including the hash of the previous timestamp in a new certificate, relative temporal authenticity can be achieved.

Formally, the TSS receives the time-stamping request for a string y for a client with a unique identifier ID. For the n-th query, a TSS does the following:

$TSS(y, ID_n)$

$t_n \leftarrow Device$

$C_n \leftarrow (n, t_n, ID_n, y_n; L_n)$

$s \leftarrow \sigma(C_n)$

Send ID_n to client ID_{n-1}

return s

Where t_n is the actual measured time, σ is the TSS's signing function, and $L_n = (t_{n-1}, ID_{n-1}, y_{n-1}, H(L_{n-1}))$ comes from the internal memory of the TSS and is based on the previously issued time-stamp. This linking information contains the hash of the previous timestamp's information, and thus establishes temporal order between two timestamps.

Once the client that requested the n-th timestamp has s and ID_{n+1}, they check whether s is a valid signature and whether t_n is correct. Any challenger to their timestamp can then check the legitimacy of their certificate, and to make sure that the client is not colluding with the TSS by producing a one-off certificate that is not actually part of the "legitimate" hash chain, they can contact person ID_{n+1} and check if their timestamp is legitimate and their linking information contains the data from s. They can then do this for ID_{n+2} and so forth, if they believe that ID_{n+2} is also colluding. Alternatively they have access to ID_{n-1} from L_n and can go backwards in the chain as well. Without using publishing, the only possible way to fool a challenger would be to create a fake chain of certificates long enough to exhaust any challenger.

If publishing is involved, the most recent linking information can be periodically published in a trusted publicly-accessible medium. Then, one can verify that the time t_n is at least within a time frame specified by two instances of the periodic publishing L_{m-1} and L_m if there is a valid chain $(L_{m-1}, \ldots L_n, \ldots L_m)$. Additionally, if it is constantly verified that each published L_m is linked to L_{m-1}, the published linking values create legitimate "anchor points" for certificate verification.

There are numerous drawbacks to the linked list scheme, and such it is only used for pedagogic purposes. For instance, checking the relative temporal authenticity between two timestamps is linear to the number of issued certificates. Additionally this requires all certificates to be stored indefinitely for future verification of other certificates. While Haber and Stornetta proposed a change to the linked list scheme, other data structures have proven to be more efficient and in fact close to optimal.

3.3.2 Tree-Like Schemes

In tree-like schemes, the time-stamping procedure constructs tree at regular intervals. Between these intervals, clients can submit their documents to be timestamped. A tree

is constructed from these documents by hashing two requests together into an inner node, and then hashing inner nodes together into a single root node, as in the hash trees proposed by Merkle [Mer80]. The certificate is then the information required to reach the published root hash from a given document. This causes the certificate size to be logarithmic with respect to the number of issued timestamps, rather than linear.

The Haber-Stornetta scheme is a tree-based scheme, where after the round tree is generated, the published string is the the root of the round tree hashed with the previously published string [BHS93]. This is done in order to strengthen temporal order between the rounds. An alternative approach to achieve this was offered by Benaloh and de Mare, where each leaf node has to be hashed together with the root of the previous round tree [BdM91].

This paper deals with linked list schemes and tree-like schemes as defined in Definition 11.

Let σ denote the empty string. If $x = (x_1, x_2) \in \{0,1\}^{2k}$ and $x_1, x_2 \in \{0,1\}^k$ then by $y \in x$ we mean $y \in \{x_1, x_2\}$.

Definition 11 (Hash-chain [BL06])**.** *Let $h : \{0,1\}^{2k} \to \{0,1\}^k$ be a hash function. By an h-chain from $x \in \{0,1\}^k$ to $r \in \{0,1\}^k$ we mean a (possibly empty) sequence $c = (c_1, \ldots, c_l)$ of pairs $c_i \in \{0,1\}^{2k}$, such that the following two conditions hold.*

1. if $c = \sigma$ then $x = r$
2. if $c \neq \sigma$ then $x \in c_1$, $r = h(c_l)$, and $h(c_i) \in c_{i+1}$ for every $i \in \{1, \ldots, l-1\}$.

We denote by $F_h(x; c) = r$ the proposition that c is an h-chain from x to r. Note that $F_h(x; \sigma) = x$ for every $x \in \{0,1\}^k$.

3.3.3 Optimal Binary Linking Scheme

Buldas *et al.* have proposed a binary linking scheme which they have proven to be asymptotically optimal [BLLV98]. This scheme is out of scope for the purposes of this paper.

4 Classical Security

In this section we outline the different security definitions and properties related to time-stamping. This section is optional for those already familiar with secure time-stamping. While this paper concerns itself with time-stamping definitions which use unpredictable adversaries such as the ones outlined in Section 4.2, we also briefly mention the list commitment approach to time-stamping.

4.1 Real-Life Attack Scenario

The security of time-stamping protocols concerns itself with protection against *back-dating* attacks. That is, consider a scenario where a client produces an original document (e.g. a proof or invention) and time-stamps it. Then once it is released to the public, an adversary (potentially the time-stamping server itself) tries to claim ownership of the document by providing a timestamp for it which predates the client's. To do so, they must manufacture a fake certificate which is a hash-chain to a root value that was published before the document was known to the time-stamping server.

4.2 Unpredictability Based Definition

An interesting problem when dealing with time-stamping is how to define the notion of an original document. The approach used in this section is one where any future documents are considered unpredictable. In "On Provably Secure Time-Stamping Schemes"[BS04], Buldas and Saarepera define new documents as ones that are sampled from an unpredictable distribution $\mathcal{D}$, such that

$$\Pr[y \leftarrow D, x \leftarrow D : x = y] = k^{-\omega(1)}$$

In later years, Buldas and Laur define unpredictability as a property of an adversary, rather than a distribution of documents [BL06]. This models real life more closely as the adversary can have some influence over the document (and more importantly its hash) by altering things such as whitespace symbols. The article by Buldas and Laur also establish many other important security definitions which we will later use as a basis for our post-quantum definitions.

Definition 12 (FPU [BL06])**.** *Let* FPU *be the class of all two-staged probabilistic poly-time adversaries (A_1,A_2), such that the first output component of A_2 is unpredictable, even if the output of A_1 is known to the predictor, i.e. for every poly-time predictor Π:*

$$Pr[(r, a) \leftarrow A_1(1^k), x' \leftarrow \Pi(r, a), (x, c) \leftarrow A_2(a) : x' = x] = k^{-\omega(1)}$$

Note that the predictor has access to the advice string a, allowing it to run A_2 within itself. This requires that the probability of each individual output from the output distribution of A_2 be negligible.

Buldas and Laur define a secure time-stamping scheme as one where a two-staged adversary first outputs r (the value to be published) and a (an advice string for A_2) given access to the server-side hash function h. Note that r need not be created using h. A_2 then needs to produce an unpredictable value X (the document) and a certificate c such that the client-side hash of the document verifies with respect to r.

Definition 13 (Secure (H, h)-time-stamping [BL06])**.** *A (H, h)-time-stamping scheme is secure if for every $(A_1, A_2) \in \mathsf{FPU}$ the next probability is negligible:*

$$\Pr[H \leftarrow \mathfrak{F}_k^c, h \leftarrow \mathfrak{F}_k^s, (r, a) \leftarrow A_1(1^k, H, h), (X, c) \leftarrow A_2(a) : F_h(H(X; c) = r]$$

Additionally they explore the necessary and sufficient properties for secure time-stamping. As they show that h does not even need to be one-way, they define new security conditions for both the client and server side hash functions.

Theorem 1 (Sufficient requirement for secure time-stamping [BL06])**.** *For secure (H,h)-time-stamping in terms of Definition 13 it is sufficient that h is* sChain*, H is* uPre *and the distribution $H \leftarrow \mathfrak{F}_k^c$ is poly-samplable.*

The sufficient property for the server side hash function is effectively the definition for secure time-stamping without client-side hash functions.

Definition 14 (Strong chain-resistance - sChain [BL06])**.** *A function distribution family $\{\mathfrak{F}\}$ is strongly chain-resistant, if for every $(A_1, A_2) \in \mathsf{FPU}$:*

$$\varepsilon(k) = Pr[h \leftarrow \mathfrak{F}_k, (r, a) \leftarrow A_1(1^k, h), (x, c) \leftarrow A_2(a) : F_h(x; c) = r] = k^{-\omega(1)}$$

For client-side hash functions, it is a necessary and sufficient requirement that the hash function preserves the unpredictability of its inputs.

Definition 15 (Unpredictability preservation - uPre [BL06]). *A function distribution family* $\{\mathfrak{F}_k\}$ *is* unpredictability preserving *if for every unpredictable poly-samplable distribution family* $\{\mathcal{D}_k\}$ *and for every predictor* $\Pi \in FP$*:*

$$\Pr[H \leftarrow \mathfrak{F}_k, y \leftarrow \Pi(1^k, H), x \leftarrow \mathcal{D}_{\|} : y = H(x)] = k^{-\omega(1)}$$

4.3 Results Within the Unpredictability Definition

In addition to defining the sufficient security definitions for secure time-stamping, Buldas and Laur also explore the relationships between the new definitions and existing properties of hash functions [BL06]. Most notably, they show that server-side hash functions do not need to be one-way. They show full separation between uPre and the previously known *2nd Preimage Resistance* property. They show that uPre is instead equivalent to the following property:

Definition 16 (weSec [BL06]). *We say that a fixed family* $H = \{H_k\}$ *is weak everywhere 2nd preimage resistant if for every poly-samplable un-predictable distribution family* $\mathcal{A}_k$ *on* $\{0,1\}^{\ell(k)}$*:*

$$\max_{X \in \{0,1\}^{\ell(k)}} \Pr[X' \leftarrow \mathcal{A}_k : X' \neq X, H(X') = H(X)] = k^{-\omega(1)}$$

During earlier work in the field of time-stamping, Buldas *et al.* had previously shown that a collision-resistant function is also unpredictability preserving [BLSW05]. Buldas and Jürgenson have shown that there are no black-box reductions from collision-resistant hash functions to time-stamping schemes using an oracle separation technique [BJ07]. However, it has been shown that collision-resistance does imply secure time-stamping for hash chains and Merkle trees with a polynomially bounded amount of allowed tree "shapes" [BN10]. Buldas and Laaneoja have also shown that this polynomial bound is necessary, as without it they are able to turn any pre-image aware function into one which is insecure for time-stamping [BL13]. They also show that under the Random Oracle and Preimage Awareness assumptions, the following holds:

Theorem 2 (Security under Random Oracle and Preimage Awareness Assumptions [BL13])**.** *If $h : \{0,1\}^{2n} \to \{0,1\}^n$ is a random oracle, then the corresponding (bounded or unbounded) hash-tree time-stamping schemes are $2^{\frac{n-1}{2}}$ secure.*

Note that this is independent from the number of issued timestamps.

With regards to MD-hash functions specifically, Buldas and Laur have shown that collision-finding attacks with regards to a random initialization vector are sufficient to violate uPre [BL06]. They also have shown that this is likely to be the case for all iterative hash functions as they show that when assuming *computational uniformity* from the compression function, the average and worst case complexities do not differ greatly.

4.4 Black-Box Definition

An alternate approach to secure time-stamping is to view one round in a tree-based scheme as a list commitment. Buldas and Laur show that every binding N-bounded list commitment scheme doubles as a secure time-stamping scheme [BL07].

Definition 17 (Black-box security [BL07])**.** *A time-stamping scheme is (t, τ, ε)-secure if there exists a τ-time black-box extractor machine $\mathcal{K}$ such that for every t-time A:*

$$Adv^{\mathsf{ts}}(\mathsf{A}) = \Pr\left[\begin{array}{c} \omega_1 \leftarrow \Omega, \mathsf{pk} \leftarrow \mathsf{Gen}, \hat{\mathcal{X}} \leftarrow \mathcal{K}^{\mathsf{A}(pk;\omega_1;\cdot)}(\mathsf{pk}) \\ (c, n, \phi) \leftarrow \mathsf{A}_1(\mathsf{pk}, \omega_1), (x, s) \leftarrow \mathsf{A}_2(\phi) : \\ (\mathsf{Ver}_{\mathsf{pk}}(c, n, x, s) = \mathsf{true} \wedge x \notin \hat{\mathcal{X}}) \vee |\hat{\mathcal{X}}| > n \end{array}\right]$$

where ω_1 denotes random coins of A_1 and $\mathcal{K}$ gets a black-box access to $\mathsf{A}_1(\mathsf{pk}; \omega_1)$ and $\mathsf{A}_2(\phi; \cdot)$. The working time of $\mathcal{K}^{\mathsf{A}(\mathsf{pk};\omega_1,\cdot)}$ includes the time needed to execute all oracle calls. For list commitments, we treat $\hat{\mathcal{X}}$ as a list and write $x \in \hat{\mathcal{X}}$ iff $x = \hat{\mathcal{X}}[\![\mathsf{loc}(s)]\!]$.

This definition states that a list commitment c can't open to an unknown document x, as there exists an algorithm $\mathcal{K}$ which is able to efficiently extract the list of all valid documents using $(\mathsf{A}_1, \mathsf{A}_2)$. This definition is currently unsuitable in the post-quantum setting as rewinding of arbitrary quantum algorithms currently can't be done (see Section 6).

5 Defining Post-Quantum Security

Since there is currently no highly efficient quantum attack against hash functions, they are sometimes regarded as seemingly "quantum immune" [BLT14]. As such, there has not been much study in the field of *hash-and-publish* time-stamping schemes in the post-quantum setting. More recently in 2017, Buldas *et al.* also note that some differences between the classical and post-quantum setting, stating among other things that as there are no restrictions placed on the advice string in the unpredictability model, it can be quantum data [BLT17]. However the definition of FPU involves state copying, which is not possible in the case of quantum states. More interestingly, a direct lifting of FPU leads to a less generalizable definition of unpredictability, which we will show in Section 6.3.5. This shows the need for clear definitions that are designed for the post-quantum setting.

In this section we define unpredictable quantum adversaries, which lays most of the groundwork for defining the unpredictability-based model for post-quantum time-stamping. In order to make the definition realizable, we remove the runtime requirement of the predictor and show that this restriction was unnecessary, even in the post-quantum setting. We then define all the previous definitions in the post-quantum setting and perform analysis on several proofs to verify whether or not they hold with respect to the new definitions.

5.1 Defining Unpredictable Quantum Adversaries

When defining post-quantum security, it is important to look at the previous definitions with a critical eye. Classical definitions can lose the real-life meaning they try to convey if one just hastily replaces normal bitstrings with quantum states and Turing machines with quantum Turing machines. In the case of commitment schemes, a definition that perfectly satisfies the real-life scenario for computationally binding commitments becomes extremely weak when directly translated into the quantum setting [ARU14]. In this section we define post-quantum security of time-stamping schemes and outline some key obstacles in doing so.

When lifting the FPU definition into the quantum setting, one must consider which variables and algorithms become quantum, and which not. In terms of adversaries, there seems to be no reason to require either part of the adversary to be classical. As for data, r is modeled to be the published root, and so it does not make sense for it to be a quantum state. However, a is the advice string which is supposed to, in part, represent the internal state of A_1. As such it should be given the possibility to be a quantum state.

This proposes an obstacle, as an unknown arbitrary quantum stated cannot be copied according to the No-Cloning Theorem. [KWHZ82] In the definition of FPU there is explicit copying of the advice string a to hand out to both Π and A_2, which must therefore be eliminated. An intuitive solution might seem to require Π to not significantly disturb

the state a during its computation. However, as we demonstrate in Section 6.3.5 this would result in a definition that does not generalize to a scenario where r can be a quantum state. The definition of a two-stage unpredictable quantum adversary could be useful in other subfields, and so we should prefer a definition which could also allow r to be quantum. Instead, we require that invoking Π does not significantly disturb the output distribution of A_2.

Definition 18 (Undisturbing quantum predictor)**.** *A quantum predictor Π is undisturbing if for every quantum detector D the value $|\Pr[b = 1|\mathsf{Game}_1] - \Pr[b = 1|\mathsf{Game}_2]|$ is negligible, where:*

$$\mathsf{Game}_1 : (r, a) \leftarrow A_1(1^k), (x', a') \leftarrow \Pi(r, a), \qquad (x, c) \leftarrow A_2(a'), b \leftarrow D(x, c)$$
$$\mathsf{Game}_2 : (r, a) \leftarrow A_1(1^k), \qquad (x, c) \leftarrow A_2(a), b \leftarrow D(x, c)$$

Definition 19 (qFPU)**.** *Let* qFPU *be the class of all two-staged quantum poly-time adversaries (A_1,A_2), such that the first output component of A_2 is unpredictable, even if the output of A_1 is known to an undisturbing quantum predictor, i.e. for every undisturbing quantum predictor Π:*

$$\Pr[(r, a) \leftarrow A_1(1^k), (x', a') \leftarrow \Pi(r, a), (x, c) \leftarrow A_2(a') : x' = x] = k^{-\omega(1)}$$

Where a and a' are quantum states and all other inputs and outputs are classical bitstrings.

A natural question that one might have is if this new definition is even satisfiable, as measurements performed on quantum states collapses them. Somewhat surprisingly, a computationally unlimited predictor that has full knowledge related to state a (in our case, it also requires access to r) can perform spectral decomposition and measure the state without disturbing it.

Theorem 3 (Spectral decomposition [NC11])**.** *Any normal operator M on a vector space V is diagonal with respect to some orthonormal basis for V. Conversely, any diagonaliazable operator is normal.*

In terms of the outer product representation, this means that M can be written as $M = \sum_i \lambda_i |i\rangle\langle i|$ where λ_i are the eigenvalues of M, $|i\rangle$ is an orthonormal basis for V, and each $|i\rangle$ an eigenvector of M with eigenvalue λ_i. In terms of projectors, $M = \sum_i \lambda_i P_i$, where λ_i are again the eigenvalues of M, and P_i is the projector onto the λ_i eigenspace of M. These projectors satisfy the completeness relation $\sum_i P_i = I$, and the orthonormality relation $P_i P_j = \delta_{ij} Pi$.

However, it is not known how to achieve spectral composition in polynomial time, which is why Π in qFPU is left unbounded. There might also be additional techniques that would allow Π to have access to the state a.

This raises a new question - is the definition with an unlimited predictor too strict for modelling real-life scenarios? The answer is no - first consider this in the classical setting.

Definition 20 (CP-FPU, CU-FPU)**.** *Let* CP-FPU *refer to the version of FPU where* Π *is a classical probabilistic poly-time Turing machine. Let* CU-FPU *refer to the version of FPU where* Π *is a classical computationally unlimited probabilistic Turing machine.*

Theorem 4 (CU-FPU = CP-FPU)**.** *For every two-staged adversary* (A_1, A_2)*:*

$$(A_1, A_2) \in \text{CP-FPU} \Leftrightarrow (A_1, A_2) \in \text{CU-FPU}$$

Proof. CP-FPU $\Rightarrow$ CU-FPU is trivial, as CU-FPU only gives Π more computational power. For CU-FPU $\Rightarrow$ CP-FPU, let U be the computationally unlimited predictor Π in the definition of CU-FPU. U has non-negligible advantage ε. We construct the following poly-time predictor P for use in CP-FPU and show that it succeeds with non-negligible probability.

$$\begin{array}{l} P(r, a) \\ \left[\begin{array}{l} (x, c) \leftarrow A_2(a) \\ \textbf{return } x \end{array}\right. \end{array}$$

Consider the following games $\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3$. Note that $\mathcal{G}_1$ is the definition of CU-FPU with an extra function call to P which is of no consequence. Thus $\mathcal{G}_1$ returns $true$ with probability ε. As all P does is call A_2, the same applies for $\mathcal{G}_\in$. Since the calls to P and A_2 are independent (one can reorder them with no consequence), $\mathcal{G}_3$ returns $true$ with probability ε^2.

$$\begin{array}{l} \mathcal{G}_1 \\ \left[\begin{array}{l} (r, a) \leftarrow A_1(1^k) \\ y \leftarrow U(r, a) \\ x' \leftarrow P(r, a) \\ (x, c) \leftarrow A_2(a) \\ \textbf{return } x = y \end{array}\right. \end{array} \qquad \begin{array}{l} \mathcal{G}_2 \\ \left[\begin{array}{l} (r, a) \leftarrow A_1(1^k) \\ y \leftarrow U(r, a) \\ x' \leftarrow P(r, a) \\ (x, c) \leftarrow A_2(a) \\ \textbf{return } x' = y \end{array}\right. \end{array} \qquad \begin{array}{l} \mathcal{G}_3 \\ \left[\begin{array}{l} (r, a) \leftarrow A_1(1^k) \\ y \leftarrow U(r, a) \\ x' \leftarrow P(r, a) \\ (x, c) \leftarrow A_2(a) \\ \textbf{return } x = y \wedge x' = y \end{array}\right. \end{array}$$

Due to transitivity, we can then claim that the probability of $x = x'$ is ε^2. Removing the unnecessary call to U then gives us our final game

$$\begin{array}{l}\mathcal{G}_4 \\ \left[\begin{array}{l}(r, a) \leftarrow A_1(1^k) \\ x' \leftarrow P(r, a) \\ (x, c) \leftarrow A_2(a) \\ \textbf{return } x = x'\end{array}\right.\end{array}$$

which is the definition of CP-FPU with the poly-time predictor P in place of Π, and this has non-negligible advantage ε^2. □

This is an extremely intuitive result when considering that as the statistical distance between the predictor's output distribution and the distribution of the predicted adversary decreases, the more accurate the predictor will be. A simulating predictor will have statistical distance of 0, making it the ideal predictor.

In the case of Theorem 4, A_2 is being ran multiple times, once as part of P and then by itself. Lifting this proof into the quantum setting would then require efficient poly-time rewinding of A_2. But could it be possible that there is a problem in proof technique, and we could prove this theorem in a radically different way? Interestingly no, as if the theorem held, there would exist a P which we could plug into the proof. So either a polynomial time predictor is not meaningful, or it is equivalent to an unlimited runtime predictor. This gives us a negative answer to the question raised earlier - qFPU is not too strict of a definition.

5.2 Changes to the Unpredictability Based Approach

As qFPU fixed which adversaries and variables are quantum, the other definitions from the unpredictability-based model outlined in Section 4.2 can be lifted fairly easily into the post-quantum setting. For formality we will present them here in order to display how each definition changes (if at all).

Definition 21 (Post-quantum secure (H, h)-time-stamping)**.** *A (H, h)-time-stamping scheme is secure if for every $(A_1, A_2) \in$ qFPU the following probability is negligible:*

$$\Pr[H \leftarrow \mathfrak{F}_k^c, h \leftarrow \mathfrak{F}_k^s, (r, a) \leftarrow A_1(1^k, H, h), (X, c) \leftarrow A_2 : F_h(H(X); c) = r]$$

Definition 22 (quPre)**.** *A function distribution family $\{\mathfrak{F}_k\}$ is unpredictability preserving if for every unpredictable quantum poly-sampleable distribution family $\{D_k\}$ and for every poly-time quantum predictor Π :*

$$\Pr[H \leftarrow \mathfrak{F}_k, y \leftarrow \Pi(1^k, H), x \leftarrow D_k : y = H(x)] = k^{-\omega(1)}$$

Definition 23 (q-weSec)**.** *A fixed function family $H = \{H_k\}$ satisfies* q-weSec *if for every quantum poly-samplable unpredictable distribution family $\mathcal{A}_k$ on $\{0,1\}^{\ell(k)}$:*

$$\max_{X \in \{0,1\}^{\ell(k)}} \Pr[X' \leftarrow \mathcal{A}_k : X' \neq X, H(X') = H(X)] = k^{-\omega(1)}$$

Definition 24 (qsChain)**.** *A function distribution family $\{\mathfrak{F}_k\}$ strongly quantum chain resistant if for every $(A_1, A_2) \in$ qFPU :*

$$\varepsilon(k) = \Pr[h \leftarrow \mathfrak{F}_k, (r, a) \leftarrow A_1(1^k, h), (x, c) \leftarrow A_2(a) : F_h(x; c) = r] = k^{-\omega(1)}$$

We have also performed preliminary review for the proofs for the properties outlined in Section 4.3 to see if the argumentation holds in the post-quantum setting. We claim the following results:

- The *Post-quantum secure (H, h)-time-stamping* $\Rightarrow$ quPre proof holds in the post-quantum setting.
- The proof for Theorem 1 holds in the post-quantum setting.
- The proof for uPre $\not\Rightarrow$ *2nd Preimage Resistance* holds in the post-quantum setting.
- The proof for *2nd Preimage Resistance* $\not\Rightarrow$ uPre requires additional review from someone versed in quantum information theory. The reason for this is that it should be analyzed whether being able to evaluate H'_k in superposition has any impact on the proof.
- The proof for *Post-quantum secure (H, h)-time-stamping* $\not\Rightarrow$ *One-Way* holds in the post-quantum setting.
- The rewinding proof for CR $\Rightarrow$ sChain does not hold in the quantum setting, see Section 6.

The oracle separation technique used for showing CR $\Rightarrow$ qsChain and the theorems for security under preimage awareness assumptions could not be reviewed due to time constraints.

6 Quantum Rewinding

Rewinding is a common proof technique in cryptography which refers to the technique of saving the inner state of an adversary in order to run it multiple times. While two separate techniques for rewinding have been developed by Watrous [Wat09] and Unruh [Unr12], they are only applicable in certain scenarios. It has been shown that, with respect to a certain oracle, some schemes which were proven to be classically secure via rewinding are insecure in the quantum setting. [ARU14]

Rewinding plays a crucial role in certain proofs and definitions related to time-stamping. Most notably, all proofs that show $\mathsf{CR} \Rightarrow \mathsf{sChain}$ (for chains or limited amount of tree shapes without the Random Oracle assumption) rely on rewinding [BN10]. Additionally, the black-box definition hinges on turning the adversary into an efficient extractor. As such we will not be defining a quantum version of that security definition, since it would not be meaningful.

This section gives an overview on existing quantum rewinding techniques, states different rewinding problems related to time-stamping, and our attempts to solve them.

6.1 Watrous' Rewinding Technique

The Watrous rewinding lemma allows a simulator to repeatedly run a unitary operation Q with success probability p a number of times, making it succeed with overwhelming probability. However, the Watrous technique hinges on not keeping information about past executions, meaning that it is not possible to use this technique to gain two different executions of an algorithm.

Lemma 1 (Quantum rewinding lemma [Wat09, Unr17])**.** *Let Q be a unitary operation from $\mathcal{H}_{in} \otimes \mathcal{H}_{anc}$ to $\mathcal{H}_{out} \otimes \mathcal{H}_{succ}$ with $\mathcal{H}_{succ} = \mathbb{C}^2$. (This implies that $\dim \mathcal{H}_{in} \otimes \mathcal{H}_{anc} = \dim \mathcal{H}_{out} \otimes \mathcal{H}_{succ}$ since a unitary operation is a square matrix.)*

Assume that there is a value $p \leq \frac{1}{2}$ such that for any $|\Psi\rangle \in \mathcal{H}_{in}$, we have that applying Q to $|\Psi\rangle \otimes |0\rangle$ and then measuring $\mathcal{H}_{succ}$ in the computational basis gives outcome 1 (success) with probability p (not $\geq p$). Let $|\phi_{succ}\rangle$ denote the post measurement state in $\mathcal{H}_out$ in that case.

Consider the following algorithm R (depending on a parameter q):

1. *Let $|\Psi\rangle$ denote the input of the algorithm (in $\mathcal{H}_in$).*
2. *Initialize $\mathcal{H}_{anc}$ with $|0\rangle$.*
3. *Apply Q.*
4. *Measure $\mathcal{H}_{succ}$ in the computational basis.*

5. *If the outcome is* 1*, exit (successfully).*
6. *Apply* $Q^\dagger$.
7. *Apply* FLIP *to* H_{anc} *where* $\mathrm{FLIP}|0\rangle := |0\rangle$ *and* $\mathrm{FLIP}|x\rangle := -|x\rangle$ *for* $x \neq 0$.
8. *Go to 3. (But at most* q *times.)*

Then for a suitable $q \in poly(1/p)$*, we have that*

- *The probability that* R *exits successfully is overwhelming.*
- *The post measurement state in* $\mathcal{H}_{out}$ *in that case is* $|\phi_{succ}\rangle$.

The difficulties in applying the Watrous lemma are often related to the fact that it requires the success probability to be precisely p rather than $\geq p$ and for it to be independent of the auxiliary input. For instance, while it may seem intuitive to apply the Watrous lemma to the first run in the Unruh Rewinding Technique, but Unruh has stated in that this condition is not always fulfilled [Unr12].

Another subtlety of the Watrous technique is the FLIP operation. This is required in order to slightly disrupt the state, as otherwise one would be applying Q right after applying $Q^\dagger$. Since Q is unitary, by definition $QQ^\dagger = Q^\dagger Q = I$, which would cancel out the second run.

6.2 Unruh Rewinding Technique

In contrast to the Watrous lemma, the Unruh rewind allows one to rewind in order to gain two successful executions of an algorithm. We view the Unruh lemma in the context of a sigma protocol of a commitment scheme $(\mathsf{Com}, \mathsf{Ver})$ where we want two different executions (c, m, u) and (c, m', u').

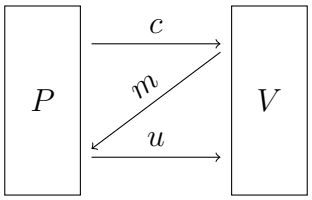

Figure 1. Example sigma protocol for a commitment scheme

If the probability of the sigma protocol succeeding is ε then the Unruh rewind provides us with the different executions with probability ε^3. However, for the rewind to work, it requires *strict binding* or *strict soundness* - that $\forall c\, \forall m : \exists_{\leq 1} u : \mathsf{Ver}(c, m, u) = 1$. That is, the first two messages uniquely fix the third. The Unruh rewind splits the prover

P into two unitary operations, (A_1, A_2) where there are multiple A_2's that are indexed by the message m sent to the prover.

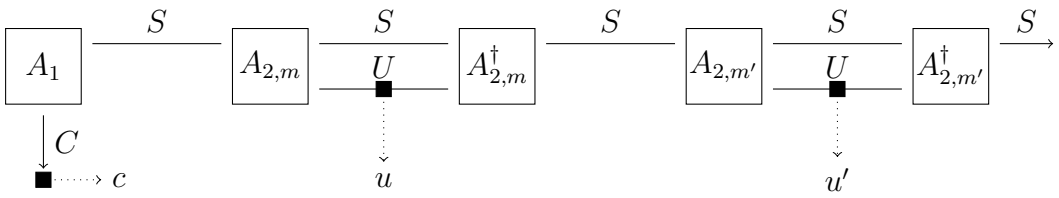

Figure 2. The Unruh rewind technique informally

Figure 2 presents a simplified circuit of the Unruh rewind where black squares indicate measurement. In essence, it allows one to rewind $A_{2,m}$ by running $A^\dagger_{2,m}$ since c and m uniquely determine what u can be, meaning the measurement does not disrupt the state S. In addition to collapsing hash functions, which by definition disrupt the state S negligibly. Note that every collapsing hash function is collision resistant.

Definition 25 (Collapse [Unr16])**.** *For a function H and algorithms A, B, consider the following games:*

$$\begin{aligned} \mathsf{Game}_1: &\quad (S, M, c) \leftarrow A(1^\eta), m \leftarrow M_{comp}(M), && b \leftarrow B(1^\eta, S, M) \\ \mathsf{Game}_2: &\quad (S, M, c) \leftarrow A(1^\eta), && b \leftarrow B(1^\eta, S, M) \end{aligned}$$

Here S, M are quantum registers. $M_{comp}(M)$ is a measurement of M in the computational basis.

We call an adversary (A, B) valid if $\Pr[H(m) = c] = 1$ when we run $(S, M, c) \leftarrow A(1^\eta)$ and measure M in the computational basis as m.

A function H is collapsing iff for any quantum-polynomial-time valid adversary (A, B), the difference $adv := |\Pr[b = 1 : \mathsf{Game}_1] - \Pr[b = 1 : \mathsf{Game}_2]|$ is negligible. (We call adv the advantage.)

6.3 Rewinding in Collapse $\Rightarrow$ qsChain

We focus on an important property which does not carry over from classical results, namely that CR $\Rightarrow$ qsChain. As every classical proof for this property uses rewinding, whether requiring the hash function to be collapsing would allow us to prove Collapse $\rightarrow$ qsChain. For simplicity we focus on hash chains and attempt to perform the rewinding proof in the quantum setting.

6.3.1 Classical Rewinding Proof

We begin by proving CR $\Rightarrow$ sChain for hash-chains for a fixed hash function, in order to analyze and lift the proof to the quantum setting. First as a matter of formality we define hash-chains for linked lists rather than Merkle trees.

Definition 26 (Linear hash chains)**.** *By an h-chain from $x \in \{0,1\}^k$ to $r \in \{0,1\}^k$ we mean a (possibly empty) sequence $c = (c_1, \ldots, c_l)$ of values $c_i \in \{0,1\}^k$, such that the following two conditions hold:*

1. *if c is empty then $x = r$*
2. *if c is not empty then $x = c_1, r = h(c_l)$ and $h(c_i) = c_{i+1}$ for every $i \in 1, \ldots, l-1$.*

Definition 27 (sChain for fixed functions)**.** *A function h is strongly chain-resistant, if for every $(A_1, A_2) \in$ FPU:*

$$\varepsilon(k) = Pr[(r,a) \leftarrow A_1(1^k, h), (x,c) \leftarrow A_2(a) : F_h(x;c) = r] = k^{-\omega(1)} \qquad (1)$$

The general proof sketch is as follows: We run A_2 twice to get two valid certificates $r = F_h(x;c) = F_h(x',c')$. From FPU it follows that $\Pr[x = x']$ is negligible. Then we can easily find a collision in (c,c'). However, this does not always hold - for instance, if $x' = h(x)$ and c' is the chain from x's parent to r. We can show that due to A_2 being unpredictable, the probability of this is negligible.

Lemma 2 (Witnesses don't predict future statements)**.** *Given an adversary $(A_1, A_2) \in$ FPU against* sChain*, for two independent runs of A_2 with the same input the probability that the witness of one run contains the statement of the other is negligible. That is*

$$Pr[(r,a) \leftarrow A_1, (x,c) \leftarrow A_2(a), (x',c') \leftarrow A_2(a) : \exists i : c_i = x'] = k^{-\omega(1)} \qquad (2)$$

Proof. Let ε be the probability defined in (2). Let ε be non-negligible. Consider a

predictor Π such that

$$\begin{aligned}&\Pi(a)\\&\left\lceil\begin{array}{l}(x,c) \leftarrow A_2(a)\\ M = \{m_i | \exists j : m_i \in c_j \vee m_i = c_j\}\\ r \xleftarrow{\$} M\\ \textbf{return } r\end{array}\right.\end{aligned}$$

Consider this Π as the adversary in (12). The probability that Π succeeds in predicting is $\frac{\varepsilon}{|M|}$ which is non-negligible based on our assumption that ε is non-negligible and the fact that $|M|$ must be polynomial due to the runtime constraints in FPU. This contradicts the fact that $(A_1, A_2) \in \mathsf{FPU}$. Therefore ε must be negligible. □

Theorem 5 ($\mathsf{CR} \Rightarrow \mathsf{sChain}$)**.**

Proof. Let h be a collision-resistant function distribution family. Assume that h is not strongly chain-resistant, meaning $\varepsilon(k)$ from (1) is non-negligible.

Consider an adversary B such that

$$\begin{aligned}&B(h)\\&\left\lceil\begin{array}{l}(r,a) \leftarrow A_1(1^k, h)\\ (x,c) \leftarrow A_2(a)\\ (\bar{x},\bar{c}) \leftarrow A_2(a)\\ \textbf{return } g(c,\bar{c})\end{array}\right.\end{aligned}$$

Where g is a recursive collision-finding function defined in pattern matching syntax as

$$\begin{aligned}&g(\sigma, \_) := \bot\\&g(\_, \sigma) := \bot\\&g(c_i : c', \bar{c}_i : \bar{c'}) := \text{if } c_i = \bar{c}_i \text{ then } g(c', \bar{c'}) \text{ else } (c_l, \bar{c}_i)\end{aligned}$$

From the runtime of A_2, the length of $(c, \bar{c})$ is polynomial, so B is also poly-time. If both runs of A_2 succeed then $h(c_l) = h(\bar{c}_l) = r$. If $c_l \neq \bar{c}_l$ then we have a collision. Otherwise we apply this search recursively as eventually either a collision is found or one of $(c, \bar{c})$ ends. If c ends before we find a collision then $c_1 \in \bar{c}$. From Lemma 2 we know that the probability of this happening is negligible. Therefore we can quantify the success of B as

$$Adv(B) = \Pr[F_h(x,c) = F_h(\bar{x},\bar{c}) = r \wedge \neg(x \in \bar{c} \vee \bar{x} \in c)] = \varepsilon(k)^2 - Pr[x \in \bar{c} \vee \bar{x} \in c]$$

which is non-negligible. This makes B a polynomial time collision finder for h. Therefore $\varepsilon(k)$ must be negligible and h satisfies sChain. □

6.3.2 Difficulties With Direct Lifting

Theorem 5 provides some interesting challenges when lifting into the quantum setting. For one, notice that due to its wording, Lemma 2 does hold in the quantum setting, but will not be of much use in the proof. Namely it does not hold when, for instance, performing the Unruh Rewinding Technique as the two runs of A_2 will no longer be independent - the state S will be changed after performing the measurement of u. We cannot claim that this preserves the second run's unpredictability, so it is possible that the second run of A_2 will output an x' that is contained in c.

Additionally, if one was to apply the Unruh Rewinding Technique directly with $c = r, m = "hello", u = (x, c)$ then this will not lead to success. This is because there is no m which will uniquely determine (x, c). In fact, due to the fact that A_2 is unpredictable, there must be exponentially many possible values for x, which means that any measurement to find the value of x will disrupt the state catastrophically.

6.3.3 Adversary With Fixed Chain Length

We use Lemma 2 in Theorem 5 to guarantee that one chain does not fully overlap the other. Fortunately by having A_1 fix the length of the chain we achieve the same result, but without requiring the runs to be independent. Now the adversary B that we need to lift would be

$$
\begin{array}{l}
B(h) \\
\left[\begin{array}{l}
(r, a, i) \leftarrow A_1(1^k, h) \\
(x, c) \leftarrow A_2(a) \\
(\bar{x}, \bar{c}) \leftarrow A_2(a) \\
\textbf{return } g(c, \bar{c})
\end{array}\right.
\end{array}
$$

Notice that requiring the length of the chain to be fixed keeps the success probability of (A_1, A_2) non-negligible. Indeed, as the length of the longest chain that A_2 would output would be polynomial, the success probability of (A_1, A_2) drops at worst by a polynomial degree.

6.3.4 Commitments With Unpredictable Openings and Messages

Consider the game in Figure 3 for a commitment scheme, where A_2 is unpredictable and A_1 sends c before A_2 starts.

In the classical setting, it can be easily shown that if the commitment scheme is binding then $\Pr[b = 1]$ is negligible. However, this is not known to hold in the quantum setting. When viewing the special case of $i = 1$ for fixed length hash chains, it is a hash commitment that conforms to this problem. Therefore we present formally proving

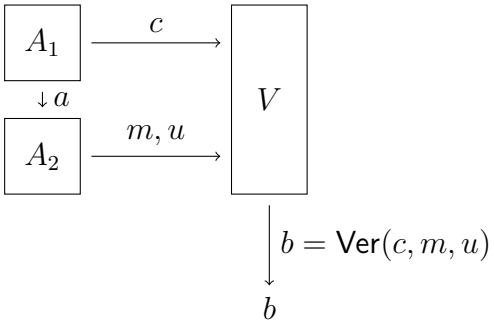

Figure 3. The unpredictable commitment subproblem

that $\Pr[b = 1]$ is negligible in the quantum case as a minimal open problem in showing $\mathsf{Collapse} \Rightarrow \mathsf{qsChain}$.

6.3.5 Subtleties in Defining No Disturbance

When defining qFPU, we noted that requiring the predictor to not disturb the state leads to a definition which is not as neatly generalizeable. Consider the following definition:

Definition 28 (Undisturbing quantum predictor - candidate)**.** *A quantum predictor* Π *is undisturbing if for every quantum detector* D *the value* $|\Pr[b = 1|\mathsf{Game}_1] - \Pr[b = 1|\mathsf{Game}_2]|$ *is negligible, where:*

$$\begin{aligned} &\mathsf{Game}_1: && (r, a) \leftarrow A_1(1^k), (x', a') \leftarrow \Pi(r, a), && b \leftarrow D(r, a') \\ &\mathsf{Game}_2: && (r, a) \leftarrow A_1(1^k), && b \leftarrow D(r, a) \end{aligned}$$

This effectively states that A_2 cannot tell the difference in whether or not Π was ran, therefore Π also has a negligible impact on A_2's output. Otherwise D could run A_2 and distinguish from its output, similar to the actual qFPU definition.

However, it may be beneficial to allow Π to collapse the state a. This is in the case where we also allow r to be quantum. If that were the case there exists a pair of adversaries (A_1, A_2) for the game pictured in Figure 3 where V does not measure c until he has received (m, u), for which $\Pr[b = 1] = 1$ and $(A_1, A_2) \in \mathsf{qFPU}$ but which clearly should not belong in qFPU. Consider the following (A_1, A_2):

Let the third wire of A_1 be the commitment r and the first and second wire together form the inner state a. A_1 initializes its wires with all zeroes, then runs $H^{\otimes n}$ on the first wire, after which its state is $\sum_i |i\rangle \otimes |0\rangle \otimes |0\rangle$. Then it runs U_h which is defined as

$$U_h|x, y, z\rangle = |x, y \oplus h(y), z \oplus h(y)\rangle$$

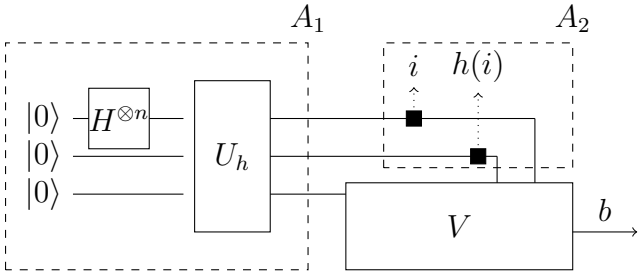

Figure 4. Adversary against Definition 28

After which its internal state is described at $\sum_i |i, f(i), f(i)\rangle$. A_2 receives $\sum_i |i, h(i)\rangle$ at which point it measures both wires, fixing its state for some uniformly chosen i as $|i, h(i)\rangle$. Thus V has received $(h(i), i, h(i))$, which verifies.

But is $(A_1, A_2) \in \mathsf{qFPU}$? If the predictor is defined as in definition Definition 28 then yes. That is because if Π who is allowed to read the full state a and create an identical quantum state a', the measurement that it performs on a' as a result of running $A_2(a')$ will output a uniform i which is completely independent of what A_2 will measure. Thus Π will not be able to predict A_2.

However, our definition for the predictor in Definition 18 rejects A_2 as an unpredictable adversary. A predictor Π which measures both wires similarly to A_2 will indeed disturb the state, but not in any way that would be noticeable from the output of A_2, as repeated identical measurements do not change the state.

6.4 Conjecture for Collapse $\Rightarrow$ qsChain

We present, without proof, a possible solution for showing Collapse $\Rightarrow$ qsChain. Recall that to show this via rewinding, we require two transcripts $(r, x, c), (r, x', c')$ such that

1. $x \neq x'$
2. $x \notin c' \wedge x' \notin c$
3. $F_h(x; c) = F_h(x'; c') = r$

In which case a collision (against Collision-resistance) is guaranteed. Recall also that we cannot apply the Unruh rewind directly as an analogue from commitment schemes as by definition the space of possible statements is exponential. We propose a rewinding scheme which only performs multiple negligibly disturbing measurements through the use of collapsing functions. This allows us to produce a collision, thus showing Collapse $\Rightarrow$ qsChain as Collapse $\Rightarrow$ CR.

Condition 1 follows from the adversary being in qFPU. Condition 2 was satisfied through unpredicatibility in the classical proof, but in the quantum setting we will achieve

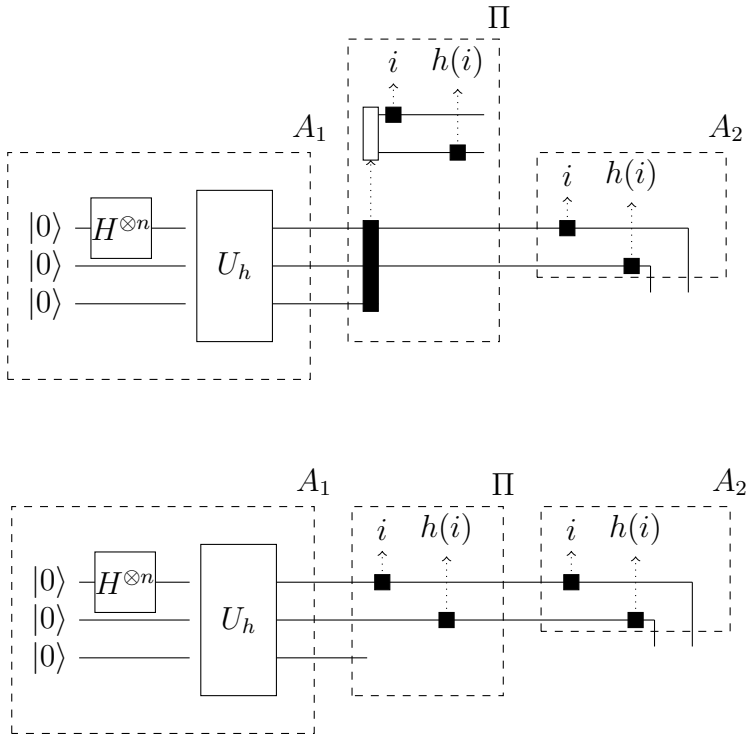

Figure 5. Predicting A_2 w.r.t Definition 28 (top) and w.r.t Definition 18.

this by requiring A_1 to fix the chain length (see Section 6.3.3). For the third condition, we could assume for simplicity that the adversary succeeds against $qsChain$ with probability 1. Since in rewinding we must only consider purified adversaries (ones which have been converted to purely unitary operations), we will also capture the auxiliary wires.

Consider an adversary (A_1, A_2) that against a collapsing hash function h achieves

$$\begin{aligned} adv := \Pr[z_0 \leftarrow |0\rangle, (r, a, i, z_1) \leftarrow A_1(1^k, h, z_0), \\ z_2 \leftarrow |0\rangle, (x, c, z_3) \leftarrow A_{2,i}(a, z_2) : \\ F_h(x; c) = r \wedge |c \models i] = 1 \end{aligned}$$

Note that since i is output by A_1, we can index the possible adversaries A_2 by i rather than provide it as an input. This simplifies the rewinding process considerably. Most notably, we can view the c output of $A_{2,i}$ as consisting of i wires. Recall that we could not measure x directly as it would disrupt the state considerably. However, notice that by definition of F_h, if $i > 0$ then $h(c_i) = r$. Since h is collapsing, we can measure c_i and only disturb the state negligibly. Additionally, now that c_i is fixed, we can measure c_{i-1} as $h(c_{i-1}) = c_i$. We repeat this process until we reach x. Once the measurement is complete, we use the FLIP function on the auxiliary input, similarly to the Watrous rewinding technique. This is needed since otherwise $A^\dagger_{2,i} A_{2,i}$ would cancel out. The

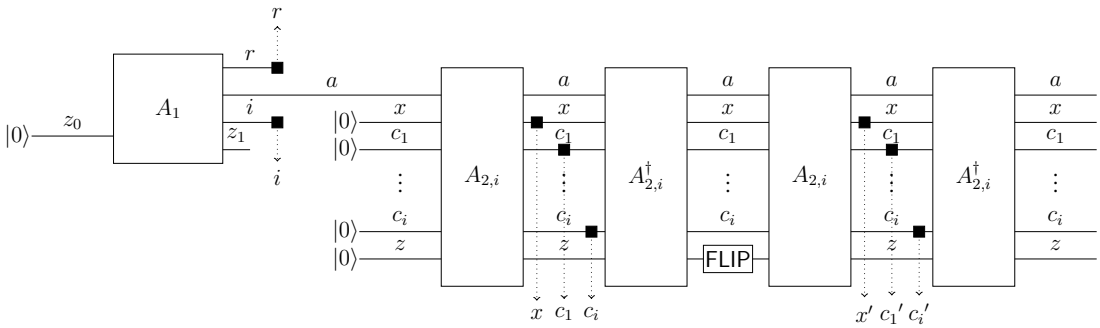

Figure 6. Proposed rewinding technique for (A_1, A_2).

Unruh rewind escapes this problem since it has different adversaries for messages m, m'. In principle, using FLIP would result in a different execution of $A_{2,i}$. This process is informally illustrated in Figure 6.

However we are unable to prove the efficacy of this construction. This is largely due to the complex nature of both the rewinding technique and the potential proof. There can be substantial difficulty in showing something that may seem intuitive at first glance, for example using the Watrous rewind inside the Unruh rewind. As such, we leave the analysis of this construction as an open problem.

7 Conclusion

We have formally outlined the notion of post-quantum timestamping and how it differs from the classical setting. Most notably in the case of FPU for which the direct quantum definition is impossible. We have also proven that out new definition qFPU is not too strict of a definition, as it is equivalent to a definition with a polynomial runtime, if such a definition can be meaningful at all. We have also outlined two different versions of qFPU and have shown that the one we decided to use generalizes better. We also examined which currently known classical results can be lifted easily into the quantum setting. Finally, we conjecture a rewinding scheme which could potentially prove a theorem which currently has no proof in the post-quantum standard model. We leave the formal analysis of this construction as an open problem. We leave another open problem in the form of a small subproblem which is the first step towards formally proving the theorem.

References

[ARU14] A. Ambainis, A. Rosmanis, and D. Unruh. Quantum attacks on classical proof systems: The hardness of quantum rewinding. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 474–483, Oct 2014.

[BdM91] Josh Benaloh and Michael de Mare. Efficient broadcast time-stamping. Technical report, 1991.

[BHS93] Dave Bayer, Stuart Haber, and W. Scott Stornetta. Improving the efficiency and reliability of digital time-stamping. In Renato Capocelli, Alfredo De Santis, and Ugo Vaccaro, editors, *Sequences II*, pages 329–334, New York, NY, 1993. Springer New York.

[BJ07] Ahto Buldas and Aivo Jürgenson. Does secure time-stamping imply collision-free hash functions? In Willy Susilo, Joseph K. Liu, and Yi Mu, editors, *Provable Security*, pages 138–150, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[BL06] Ahto Buldas and Sven Laur. Do broken hash functions affect the security of time-stamping schemes? In Jianying Zhou, Moti Yung, and Feng Bao, editors, *Applied Cryptography and Network Security*, pages 50–65, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[BL07] Ahto Buldas and Sven Laur. Knowledge-binding commitments with applications in time-stamping. In Tatsuaki Okamoto and Xiaoyun Wang, editors, *Public Key Cryptography – PKC 2007*, pages 150–165, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[BL13] Ahto Buldas and Risto Laanoja. Security proofs for hash tree time-stamping using hash functions with small output size. In Colin Boyd and Leonie Simpson, editors, *Information Security and Privacy*, pages 235–250, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[BLLV98] Ahto Buldas, Peeter Laud, Helger Lipmaa, and Jan Villemson. Time-stamping with binary linking schemes. In Hugo Krawczyk, editor, *Advances in Cryptology — CRYPTO '98*, pages 486–501, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.

[BLSW05] Ahto Buldas, Peeter Laud, Märt Saarepera, and Jan Willemson. Universally composable time-stamping schemes with audit. In Jianying Zhou, Javier Lopez, Robert H. Deng, and Feng Bao, editors, *Information Security*, pages 359–373, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[BLT14] Ahto Buldas, Risto Laanoja, and Ahto Truu. Efficient quantum-immune keyless signatures with identity. Cryptology ePrint Archive, Report 2014/321, 2014. https://eprint.iacr.org/2014/321.

[BLT17] Ahto Buldas, Risto Laanoja, and Ahto Truu. Keyless signature infrastructure and PKI: hash-tree signatures in pre- and post-quantum world. *IJSTM*, 23(1/2):117–130, 2017.

[BN10] Ahto Buldas and Margus Niitsoo. Optimally tight security proofs for hash-then-publish time-stamping. In Ron Steinfeld and Philip Hawkes, editors, *Information Security and Privacy*, pages 318–335, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[BS04] Ahto Buldas and Märt Saarepera. On provably secure time-stamping schemes. In Pil Joong Lee, editor, *Advances in Cryptology - ASIACRYPT 2004*, pages 500–514, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[CMZ15] V. Clupek, L. Malina, and V. Zeman. Secure digital archiving in post-quantum era. In *2015 38th International Conference on Telecommunications and Signal Processing (TSP)*, pages 622–626, July 2015.

[HS91] Stuart Haber and W. Scott Stornetta. How to time-stamp a digital document. *Journal of Cryptology*, 3(2):99–111, Jan 1991.

[KWHZ82] W K. Wootters and W H. Zurek. A single quantum cannot be cloned. 299:802, 10 1982.

[Mer80] R. C. Merkle. Protocols for public key cryptosystems. In *1980 IEEE Symposium on Security and Privacy*, pages 122–122, April 1980.

[NC11] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, New York, NY, USA, 10th edition, 2011.

[SxSPT10] S. Santesson, 3xA Security, N. Pope, and Thales. ESSCertIDv2 Update for RFC 3161. RFC 5816, RFC Editor, March 2010.

[Unr12] Dominique Unruh. Quantum proofs of knowledge. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, pages 135–152, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[Unr16] Dominique Unruh. Computationally binding quantum commitments. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology – EUROCRYPT 2016*, pages 497–527, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

[Unr17] Dominique Unruh. Quantum cryptography - short notes, fall 2013. November 2017.

[Wat09] John Watrous. Zero-knowledge against quantum attacks. *SIAM J. Comput.*, 39(1):25–58, May 2009.

Appendix

I. Index

List of definitions

List of theorems

List of lemmata

List of Figures

II. Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, **Raul-Martin Rebane**,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to:

   1.1 reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and

   1.2 make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright,

   of my thesis

   Post-quantum secure time-stamping

   supervised by Dominique Peer Ghislain Unruh

2. I am aware of the fact that the author retains these rights.

3. I certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu, 21.05.2018