Post-Quantum Security of Authenticated Key Establishment Protocols

by

Jason LeGrow

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Combinatorics and Optimization

Waterloo, Ontario, Canada, 2016

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

We present a security model for authenticated key establishment that allows for quantum interactions between the adversary and quantum oracles that emulate classical parties, resulting in a truly post-quantum security definition. We then give a generic construction for a secure protocol in the quantum random oracle model by combining a signature scheme which is existentially unforgeable under adaptive quantum chosen message attack in the quantum random oracle model (EUF-qCMA-QRO secure) with an unauthenticated key establishment protocol which is secure against a passive adversary. This construction allows us to give an explicit example of a secure protocol whose security is based on a variant of the Diffie-Hellman problem for isogenies of supersingular elliptic curves; in particular, generic security-strengthening transformations allow us to take a signature scheme which is EUF-CMA-RO secure against a quantum adversary and transform it into an EUF-qCMA-QRO signature scheme, which we combine with a standard secure unauthenticated key establishment protocol to achieve the desired result.

Acknowledgements

I wish to thank my supervisor, David Jao, for his encouragement, guidance, and patience as I prepared this thesis. I would also like to thank my other readers, Alfred Menezes and Michele Mosca for their helpful and constructive comments.

I must also express my gratitude to the Department of Combinatorics and Optimization, the University of Waterloo, the Ontario Ministry of Training, Colleges and Universities, and the Natural Sciences and Engineering Research Council of Canada for their generous financial support.

Table of Contents

List of Figures

"Au milieu de l'hiver, j'apprenais enfin qu'il y avait en moi un été invincible."

– Albert Camus, *Retour à Tipasa*

"In the midst of winter, I finally learned that within me there lay an invincible summer."

– Albert Camus, *Return to Tipasa*

Chapter 1

Introduction

One of the primary objectives of cryptography is to enable secure communications over public channels. Frequently this goal is achieved by establishing a shared secret among communicating parties to be used for an encryption scheme; the process of establishing such a shared secret is called key establishment. Of course, key establishment is not useful if the communicating parties cannot be convinced that they are establishing keys with the intended peers (rather than with a malicious third party who is hijacking the communications), and so for that reason so-called *authenticated* key establishment protocols have been developed.

In order to talk about the efficacy of a protocol we need a suitable notion of "security;" this is a delicate subject, as the specifications of party and attacker capabilities that lead to the security definition (called the *security model*) must accurately reflect the security properties to be modelled. In light of the development of quantum computers, cryptographers are becoming more and more interested in modelling *post-quantum* cryptography—classical cryptographic protocols which are secure even against a quantum attacker[1]. Though post-quantum security models have been proposed for encryption [12, 34] and signature schemes [5], no adequate model has yet been proposed for authenticated key establishment. In this work we propose such a security model.

[1] What exactly is meant by a "quantum attacker" depends on the specific security model. At the least, a quantum attacker will have access to a quantum computer.

The motivating idea of the security model presented here is that a post-quantum security model must allow for *quantum interactions* between legitimate communicating parties and an attacker; this leads naturally to the security model we present in Chapter 4. Once the model is defined we give a general method for constructing a secure authenticated key establishment protocol and give a specific example of a secure protocol.

1.1 Thesis Outline

The material in this thesis is covered in the following order. In the remainder of Chapter 1 we cover the mathematical background of algebraic geometry and elliptic curves in particular, the fundamentals of quantum information processing, some notions regarding functions, and examples of computational assumptions used in cryptography. In Chapter 2 we discuss some fundamental parts of public-key cryptography that will enable us to construct secure authenticated key establishment protocols. Chapter 3 is a discussion of what a security model for authenticated key establishment should entail, and a survey of some important security models. In Chapter 4 we detail our new security model for post-quantum authenticated key establishment, and give a generic construction for secure protocols in this model; in Chapter 5 we apply this construction to obtain a concrete example of a secure authenticated key establishment protocol. Finally, we conclude and discuss future work in Chapter 6.

1.2 Mathematical Background – Algebraic Geometry

To understand the construction we will use for a secure authenticated key establishment protocol, one must first understand the fundamentals of algebraic geometry. We very briefly cover the required definitions here, extracting the relevant parts of [33, Chapter I]; for a more thorough reference see, for instance, [14].

1.2.1 Affine Varieties

Definition 1.1 (Affine n-Space)**.** For a fixed field $\mathbb{K}$, *affine n-space over* $\mathbb{K}$ is the set $\mathbb{A}^n = \{(x_1, \ldots, x_n) : x_1, \ldots, x_n \in \overline{\mathbb{K}}\}$, where $\overline{\mathbb{K}}$ is the algebraic closure of $\mathbb{K}$.

For any field $\mathbb{L} \subseteq \overline{\mathbb{K}}$ the set of $\mathbb{L}$-*rational points* of $\mathbb{A}^n$ is the set

$$\mathbb{A}^n(\mathbb{L}) = \{(x_1, \ldots, x_n) : x_1, \ldots, x_n \in \mathbb{L}\} = \mathbb{A}^n \cap \mathbb{L}^n.$$

We will be concerned with special subsets of $\mathbb{A}^n$ called algebraic sets.

Definition 1.2 (Affine Algebraic Set)**.** For any ideal $J \subseteq \overline{\mathbb{K}}[x_1, x_2, \ldots, x_n]$ we associate the set $V(J) \subseteq \mathbb{A}^n$ defined by

$$V(J) = \{P \in \mathbb{A}^n \colon f(P) = 0 \text{ for all } f \in J\}.$$

Any set $U \subseteq \mathbb{A}^n$ of the form $U = V(J)$ for some ideal $J \subseteq \overline{\mathbb{K}}[x_1, x_2, \ldots, x_n]$ is called an *affine algebraic set*.

Just as we can construct affine algebraic sets in $\mathbb{A}^n$ from ideals in $\overline{\mathbb{K}}[x_1, x_2, \ldots, x_n]$, we can construct ideals from affine algebraic sets.

Definition 1.3 (Ideal of an Affine Algebraic Set)**.** For any affine algebraic set $U \subseteq \mathbb{A}^n$ we associate the ideal $I(U)$ defined by

$$I(U) = \{f \in \overline{\mathbb{K}}[x_1, x_2, \ldots, x_n] : f(P) = 0 \text{ for all } P \in U\}.$$

We call an affine algebraic set U an *affine variety* if its ideal $I(U)$ is prime.

For any field $\mathbb{L} \subseteq \overline{\mathbb{K}}$, we say that an affine algebraic set U is *defined over* $\mathbb{L}$ if $I(U)$ is generated by a set of polynomials in $\mathbb{L}[x_1, x_2, \ldots, x_n]$. In this case the set of $\mathbb{L}$-*rational points* of U is the set $U(\mathbb{L}) = \{(x_1, \ldots, x_n) \in U : x_1, \ldots, x_n \in \mathbb{L}\} = U \cap \mathbb{A}^n(\mathbb{L})$.

For any variety $U \subseteq \mathbb{A}^n$, define its *coordinate ring* as $\overline{\mathbb{K}}[U] = \overline{\mathbb{K}}[x_1, x_2, \ldots, x_n]/I(U)$; moreover, define its *function field* $\overline{\mathbb{K}}(U)$ to be the field of fractions of $\overline{\mathbb{K}}[U]$; *i.e.,*

$$\overline{\mathbb{K}}(U) \cong (\overline{\mathbb{K}}[U] \times \overline{\mathbb{K}}[U] \backslash \{0\}) / \sim$$

where $\sim$ is an equivalence relation defined by $(f_1, g_1) \sim (f_2, g_2)$ if and only if $f_1 g_2 - f_2 g_1 \in I(U)$. It is easy to see that $\overline{\mathbb{K}}(U)$ is a finite-dimensional $\overline{\mathbb{K}}$-vector space; we define the *dimension* of U—written $\dim U$—to be the transcendence degree of $\overline{\mathbb{K}}(U)$ over $\overline{\mathbb{K}}$; that is, the size of the largest algebraically independent subset (over $\overline{\mathbb{K}}$) of $\overline{\mathbb{K}}(U)$.

For the purposes of elliptic curve cryptography we require a type of variety which is especially well-behaved.

Definition 1.4 (Non-Singular Point; Non-Singular Variety)**.** Let U be a variety. By Hilbert's Basis Theorem [17, Section VIII.4, Theorem 4.9], $I(U)$ is finitely-generated; say $I(U) = (f_1, f_2, \ldots, f_t)$ for some $f_1, f_2, \ldots, f_t \in \overline{\mathbb{K}}[x_1, x_2, \ldots, x_n]$. We say that $P \in U$ is a *non-singular* point of U if the Jacobian matrix

$$\mathbb{J}(f_1, \ldots, f_t) = \left(\frac{\partial f_i}{\partial x_j}(P) \right)_{\substack{1 \le i \le t \\ 1 \le j \le n}}$$

has rank $n - \dim U$, where $\frac{\partial f_i}{\partial x_j}(P)$ is the formal partial derivative of f_i with respect to x_j evaluated at P. We say that U is non-singular if each point of U is non-singular.

1.2.2 Projective Varieties

We now define a new space and type of variety derived from affine space and affine varieties.

Definition 1.5 (Projective n-space)**.** Given a field $\mathbb{K}$, *projective n-space* over $\mathbb{K}$ is defined as

$$\mathbb{P}^n = (\mathbb{A}^{n+1} \backslash \{\vec{0}\}) / \sim$$

where $\sim$ is the equivalence relation defined by $(x_0, x_1, \ldots, x_n) \sim (y_0, y_1, \ldots, y_n)$ if and only if there is $\lambda \in \overline{\mathbb{K}} \backslash \{0\}$ such that $x_i = \lambda y_i$ for all $0 \le i \le n$.

We denote the elements of $\mathbb{P}^n$ as $[x_0, x_1, \ldots, x_n]$, where $(x_0, x_1, \ldots, x_n)$ is any coset representative. The values $x_0, x_1, \ldots, x_n$ are called the *homogeneous coordinates* of the point.

As in the case of affine n-space, we can consider the set of $\mathbb{L}$-rational points in projective n-space, defined as $\mathbb{P}^n(\mathbb{L}) = \{[x_0, x_1, \ldots, x_n] : x_i \in \mathbb{L} \text{ for all } 0 \le i \le n\}$. As well, we will consider projective algebraic sets; we first require the notion of a homogeneous polynomial.

Definition 1.6 (Homogeneous Polynomial)**.** A polynomial $f \in \overline{\mathbb{K}}[x_0, x_1, \ldots, x_n]$ is said to be *homogeneous* if there exists $d \in \mathbb{N}$ so that

$$f(\lambda P) = \lambda^d f(P)$$

for all $\lambda \in \overline{\mathbb{K}}$ and for all $P \in \mathbb{P}^n$. The number d is the *degree* of homogeneity.

An ideal is said to be homogeneous if it is generated by homogeneous polynomials. Projective algebraic sets are defined in terms of homogeneous ideals.

Definition 1.7 (Projective Algebraic Set)**.** Let $J \subseteq \overline{\mathbb{K}}[x_0, x_1, \ldots, x_n]$ be a homogeneous ideal. To J we associate a subset $V(J)$ of $\mathbb{P}^n$ defined by

$$V(J) = \{[x_0, x_1, \ldots, x_n] \in \mathbb{P}^n \,:\, f(x_0, x_1, \ldots, x_n) = 0 \text{ for all homogeneous } f \in J\}$$

A subset U of $\mathbb{P}^n$ is called a *projective algebraic set* if $U = V(J)$ for some homogeneous ideal J of $\overline{\mathbb{K}}[x_0, x_1, \ldots, x_n]$.

Analogous to the case of affine algebraic sets, we associate to every projective algebraic set U a homogeneous ideal $I(U)$ defined by

$$I(U) = \left(\left\{f \in \overline{\mathbb{K}}[x_0, x_1, \ldots, x_n] \,:\, f \text{ is homogeneous and } f(P) = 0 \text{ for all } P \in U\right\}\right)$$

and we say that a projective algebraic set is defined over a field $\mathbb{L} \subseteq \overline{\mathbb{K}}$ if its ideal is generated homogeneous by polynomials in $\mathbb{L}[x_0, x_1, \ldots, x_n]$. If U is defined over $\mathbb{L}$, we define the set of $\mathbb{L}$-rational points of U as $U(\mathbb{L}) = U \cap \mathbb{P}^n(\mathbb{L})$.

Definition 1.8 (Projective Variety)**.** An algebraic set U is called a *projective variety* if $I(U)$ is a homogeneous prime ideal.

We would like a notion of dimension and non-singularity for projective varieties; we rely on the corresponding definitions for affine varieties.

Definition 1.9 (Dimension)**.** Let $U \subseteq \mathbb{P}^n$ be a projective variety, and let S be a copy of $\mathbb{A}^n$ contained in $\mathbb{P}^n$ such that $U \cap S \neq \varnothing$. The *dimension* of U is $\dim U \cap S$, interpreted as a variety in S.

Definition 1.10 (Non-Singular Point; Non-Singular Projective Variety)**.** Let P be a point in a projective variety U, and let S be a copy of $\mathbb{A}^n$ contained in $\mathbb{P}^n$ and containing P. We say that P is a non-singular point of U if it is a nonsingular point of $U \cap S$, interpreted as an affine variety of S. Moreover, U is said to be non-singular if each of its points is non-singular.

Finally, we define the function field of a projective variety, and regular maps.

Definition 1.11 (Projective Coordinate Ring; Function Field)**.** Let U be a projective variety. The *projective coordinate ring* of U is the set $\overline{\mathbb{K}}[U] = \overline{\mathbb{K}}[x_0, x_1, \ldots, x_n]/I(U)$. The *function field* of U is the set

$$\overline{\mathbb{K}}(U) = \left\{ F(x_0, x_1, \ldots, x_n) = \frac{f(x_0, x_1, \ldots, x_n)}{g(x_0, x_1, \ldots, x_n)} \text{ and } g \notin I(U) \right\} / \sim$$

where $\sim$ is the equivalence relation defined by $\frac{f_1}{g_1} \sim \frac{f_2}{g_2}$ if and only if $f_1 g_2 - f_2 g_1 \in I(U)$.

Definition 1.12 (Regular Point; Regular Map)**.** An element $F \in \overline{\mathbb{K}}(U)$ is called *regular* at a point P if it can be written as $F = \frac{f}{g}$ where $g(P) \neq 0$. A map F is called regular if it is regular at every point in its domain.

Remark 1.1. At different points P, it may be necessary to take different functions f and g.

1.2.3 Rational Maps, Morphisms, and Isomorphisms

Here we consider important classes of functions between projective varieties, of which isogenies (Section 1.2.5) are a special case. As the name suggests, isogeny-based cryptography uses isogenies of certain algebraic varieties to achieve key establishment and other cryptographic goals.

Definition 1.13 (Rational Map)**.** Let U_1 and U_2 be projective varieties. A map $\phi \colon U_1 \to U_2$ is called a *rational map* if there exist coordinate functions $f_0, f_1, \ldots, f_m \in \overline{\mathbb{K}}(U_1)$ such that $\phi(P) = [f_i(P)]_{0 \leq i \leq m}$ for all $P \in U_1$ at which $f_0, f_1, \ldots, f_m$ are all defined.

If there is $\lambda \in \overline{\mathbb{K}}\backslash\{0\}$ such that $\lambda f_0, \lambda f_1, \ldots, \lambda f_m \in \mathbb{L}[x_0, x_1, \ldots, x_n]$ for some $\mathbb{L} \subseteq \overline{\mathbb{K}}$ we say that ϕ is *defined over* $\mathbb{L}$. We are concerned with especially well-behaved rational maps, called *morphisms*.

Definition 1.14 (Regular Point)**.** Let $\phi\colon U_1 \to U_2$ be a rational map, and let $P \in U_1$. We say that ϕ is *regular* at P if there exists $g \in \overline{\mathbb{K}}(U_1)$ so that, if the coordinate functions for ϕ are $f_0, f_1, \ldots, f_m$, then $g \cdot f_i$ is regular at P for each $0 \leq i \leq n$, and, moreover, there is $0 \leq i^* \leq m$ such that $(g \cdot f_{i^*})(P) \neq 0$. When such g exists, we define $\phi(P) = [(g \cdot f_i)(P)]_{0 \leq i \leq m}$.

A rational map which is regular at every point of its domain is called a *morphism*. This naturally leads to the definition of isomorphism of projective varieties.

Definition 1.15 (Isomorphism; Isomorphic Varieties)**.** Let U_1 and U_2 be projective varieties. We say that $\phi\colon U_1 \to U_2$ is an *isomorphism* if there exists a morphism ψ so that $\phi \circ \psi = \iota_{U_2}$, the identity on U_2, and $\psi \circ \phi = \iota_{U_1}$, the identity on U_1. When such an isomorphism exists, we say that U_1 and U_2 are *isomorphic*.

If an isomorphism $\phi\colon U_1 \to U_2$ is defined over a field $\mathbb{L}$, we sometimes say that U_1 and U_2 are $\mathbb{L}$-isomorphic for the sake of clarity. From the perspective of algebraic geometry, isomorphic varieties are indistinguishable, and so for geometric problems it suffices to consider any fixed representative in an isomorphism class. We will use this fact later.

1.2.4 Elliptic Curves

Definition 1.16 (Elliptic Curve [33, Section III.3])**.** An *elliptic curve* E is a nonsingular curve (*i.e.*, a projective variety of dimension one) of genus one, with a distinguished point $\mathcal{O}$, called the *point at infinity*.

We say that an elliptic curve E is *defined over* a field $\mathbb{K}$ if it is defined over $\mathbb{K}$ as an algebraic set.

The following proposition more concretely characterizes elliptic curves.

Proposition 1.1 (Characterization of Elliptic Curves [33, Section III.3, Proposition 3.1])**.** Let E be an elliptic curve defined over $\mathbb{K}$. Then:

i. There exist functions $X, Y \in \mathbb{K}(E)$ such that the map

$$\begin{aligned}\phi\colon E &\to \mathbb{P}^2(\overline{\mathbb{K}})\\ \phi\colon [x_0,\dots,x_n] &\mapsto [X(x_0,\dots,x_n), Y(x_0,\dots,x_n), 1]\end{aligned}$$

is an isomorphism of E onto a curve given by the Weierstrass Equation

$$C\colon Y^2Z + a_1XYZ + a_3YZ^2 = X^3 + a_2X^2Z + a_4XZ^2 + a_6Z^3 \tag{1.1}$$

for some $a_1,\dots,a_6 \in \mathbb{K}$ satisfying $\phi(\mathcal{O}) = [0,1,0]$. The functions X and Y are called *Weierstrass coordinates* for E.

ii. Any two Weierstrass equations for E are related by a linear change of variables of the form

$$\begin{aligned}X &= u^2X' + r\\ Y &= u^3Y' + su^2X' + t\end{aligned}$$

for some $u \in \mathbb{K}\backslash\{0\}$ and $r, s, t \in \mathbb{K}$.

iii. Every non-singular cubic curve C given by a Weierstrass equation of the form (1.1) is an elliptic curve defined over $\mathbb{K}$ with distinguished point $\mathcal{O} = [0,1,0]$.

In light of Proposition 1.1, we will exclusively speak of elliptic curves described by a Weierstrass equation of the form (1.1); moreover, for brevity of notation, we will often consider the *finite* points of an elliptic curve $E/\mathbb{K}$ as being pairs $(x,y) \in \mathbb{A}^2(\overline{\mathbb{K}})$, with the point at infinity being denoted only by $\mathcal{O}$. Whenever we define a function in terms of these coordinates, we will specify the image of $\mathcal{O}$ separately. In this formalism, we can write

$$E = \{(x,y) \in \mathbb{A}^2(\overline{\mathbb{K}}) \,:\, y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6\} \sqcup \{\mathcal{O}\}. \tag{1.2}$$

Remark 1.2 (Simplified Weierstrass Equation for Elliptic Curves)**.** It can be shown (*q.v.* [33, Section III.1]) that if $\operatorname{char}\mathbb{K} \neq 2, 3$ then the Weierstrass equation (1.1) can be further

simplified to

$$C\colon Y^2Z = X^3 + aXZ^2 + bZ^3$$

so that Equation (1.2) becomes

$$E = \{(x, y) \in \mathbb{A}^2(\overline{\mathbb{K}}) \,:\, y^2 = x^3 + ax + b\} \sqcup \{\mathcal{O}\}$$

for some $a, b \in \mathbb{K}$, by a linear change of variables.

1.2.5 Isogenies

We can introduce group structure on an elliptic curve E in a natural way [33, Section III.2, Proposition 2.2], and this group structure allows us to define a class of functions, called *isogenies*, which preserve some of this algebraic structure, as well as the geometric structure of the curves.

Definition 1.17 (Isogeny [33, Section III.4])**.** Let E and E' be elliptic curves defined over a field $\mathbb{K}$. An *isogeny* from E to E' is an algebraic morphism $\phi\colon E \to E'$ such that $\phi(\mathcal{O}_E) = \mathcal{O}_{E'}$.

We say that an isogeny $\phi\colon E \to E'$ is *defined over* a field $\mathbb{K}$ if it is defined over $\mathbb{K}$ as a rational map. The *degree* of an isogeny is its degree when considered as a rational map. If ϕ is a separable isogeny, then $\deg \phi = |\ker \phi|$ [9, Section 2].

For elliptic curves E and E', we say that E' is *isogenous* to E over $\mathbb{K}$ if and only if there is an isogeny ϕ from E to E' defined over $\mathbb{K}$ such that $\phi(E) \neq \{\mathcal{O}_{E'}\}$. It can be shown (*q.v.* [33, Section III.6, Theorem 6.1]) that E' is isogenous to E over $\mathbb{K}$ if and only if E is isogenous to E' over $\mathbb{K}$; that is, the property of "being isogenous" is an equivalence relation, and we defined the *isogeny class* of a curve E defined over $\mathbb{K}$ to be the set of all curves E' which are isogenous to E, up to $\overline{\mathbb{K}}$-isomorphism as algebraic sets. Since any algebraic morphism of curves is either constant or surjective [14, Chapter II, Section 6, Proposition 6.8], if $\phi\colon E \to E'$ is a nontrivial isogeny, then $\phi(E) = E'$.

A theorem of Tate [36, Section 3, Theorem 1] says that if E and E' are defined over a *finite* field $\mathbb{K} = GF(q)$, then E and E' are isogenous over $\mathbb{K}$ if and only if $|E(\mathbb{K}')| = |E'(\mathbb{K}')|$ for every finite extension $\mathbb{K}'$ of $\mathbb{K}$.

Let E be defined over a field of characteristic $p > 0$, and for each $\ell \in \mathbb{Z}$, let $E[\ell]$ denote the set of ℓ-torsion points of E. If $p \nmid \ell$, the map

$$[\ell] : E \to E$$
$$[\ell] : P \to \ell P$$

is separable and has degree ℓ^2; hence, since $E[\ell] = \ker[\ell]$ is a finite abelian group, it must be that $E[\ell] \cong \mathbb{Z}/\ell\mathbb{Z} \oplus \mathbb{Z}/\ell\mathbb{Z}$, since any other rank 2 Abelian group of order ℓ^2 has elements of order strictly greater than ℓ, by the Fundamental Theorem of Finitely Generated Abelian Groups [31, Chapter 10, Theorem 10.20]. Additionally, either $E[p^r] = \{\mathcal{O}\}$ for all $r \in \mathbb{Z}$, or $E[p^r] \cong \mathbb{Z}/p^r\mathbb{Z}$ for all $r \in \mathbb{Z}$ [33, Chapter V, Section 3, Theorem 3.1]; in the first case we say that E is supersingular, while in the second case we say that E is ordinary. Any two isogenous elliptic curves are either both ordinary or both supersingular. We will be concerned primarily with supersingular elliptic curves for our cryptographic applications.

Any supersingular elliptic curve E is defined over $GF(p^2)$ for some prime p, and for each prime $\ell \neq p$ there are $\ell + 1$ isogenies of degree ℓ with domain E (though not all of them are defined over $GF(p^2)$, in general) [9]. These isogenies of degree ℓ are in one-to-one correspondence with the subgroups Φ of E of order ℓ; moreover, each such subgroup is the kernel of a unique isogeny ϕ, and we write $\phi(E) = E/\Phi$ [33, Chapter III, Section 4, Proposition 4.12]. Hence to specify an isogeny it suffices to specify its kernel, and conversely given a subgroup Φ of E we can construct the isogeny ϕ whose kernel is Φ, using Vélu's formulae [38]. In particular, if Φ is generated by a point $R \in E(GF(p^2))$, then we have a compact representation of ϕ, and we can compute ϕ efficiently knowing only R [9]. We will use such isogenies in an authenticated key establishment protocol in Chapter 5.

1.2.6 The j-Invariant

Associated to every elliptic curve E defined over $\mathbb{K}$ is a number $j(E) \in \mathbb{K}$, called the *j-invariant* of the curve. As the name suggests, the j-invariant is invariant under $\overline{\mathbb{K}}$-

isomorphisms of algebraic sets, and so a j-invariant uniquely identifies a $\overline{\mathbb{K}}$-isomorphism class of elliptic curves over $\mathbb{K}$. Given an elliptic curve E, its j-invariant can be found in polynomial-time; moreover, given a j-invariant $j^* \in \mathbb{K}$, one can find in polynomial time the curve E with $j(E) = j^*$. Knowing this, we have a compact description of an elliptic curve for the purposes of communication during a key establishment protocol.

1.3 Fundamentals of Quantum Information

Quantum computers operate in a fundamentally different way from classical computers; we briefly cover the fundamentals here, along with some further results that will be necessary in later sections. For a more complete introduction see, for instance, [20] or [28].

In classical computing, the fundamental unit of information is the bit, which takes on values in $\{0, 1\}$. In quantum computing, the fundamental unit of information is the qubit, whose state can be *any unit length complex linear combination* of the standard basis qubits $|0\rangle$ and $|1\rangle$[2]. That is, the state of a qubit can be written as

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$

for some $\alpha, \beta \in \mathbb{C}$ with $|\alpha|^2 + |\beta^2| = 1$.

Just as we can consider classical bit strings of length n whose states are in $\{0, 1\}^n$, for quantum computing we consider systems of many qubits. If we have some qubits in states $|\psi_i\rangle = \alpha_i^{(0)}\,|0\rangle + \alpha_i^{(1)}\,|1\rangle$ for $1 \leq i \leq n$, then their joint state is the tensor product

$$|\psi_1\rangle \otimes \cdots \otimes |\psi_n\rangle = \sum_{\sigma \in \{0,1\}^n} \prod_{i=1}^{n} \alpha_i^{(\sigma_i)}\,|\sigma_1\rangle \otimes \cdots \otimes |\sigma_n\rangle\,.$$

For brevity, for $\sigma \in \{0, 1\}^n$ we use the notation $|\sigma\rangle = |\sigma_1\rangle \otimes \cdots \otimes |\sigma_n\rangle$. More generally, the state of an n-qubit system is a unit length complex linear combination

$$|\Psi\rangle = \sum_{\sigma \in \{0,1\}^n} \alpha_\sigma\,|\sigma\rangle\,.$$

[2]For concreteness, we can choose an orthonormal basis $\{\vec{v}_0, \vec{v}_1\}$ for $\mathbb{C}^2$, and define $|0\rangle = \vec{v}_0$ and $|1\rangle = \vec{v}_1$. The state of a qubit is just a unit length vector in this space.

Moreover, any set $\mathcal{M}$ of size at most 2^n can be embedded in $\{0,1\}^n$ and so we can consider states of the form

$$|\Psi\rangle = \sum_{m\in\mathcal{M}} \alpha_m |m\rangle \in \mathbb{C}^{\mathcal{M}}$$

by identifying $|m\rangle$ with $|\sigma\rangle$, where σ is the embedding of m in $\{0,1\}^n$.

1.3.1 Operations on Qubits

In contrast with classical computing, where, in principle, any bitwise operation can be implemented, on a quantum computer all operations must be reversible and must preserve the inner product on $\bigotimes_{j=1}^{n} \mathbb{C}^2$ [28, Chapter 2, Section 2.2.2, Postulate 2]. For any fixed basis of $\bigotimes_{j=1}^{n} \mathbb{C}^2$, this means that the admissible quantum operations are precisely the unitary matrices of dimension 2^n.

If n qubits are in some global state $|\Psi\rangle$, it is possible to apply a unitary operator to some subset of them. If we apply the operator U on the first m qubits[3] of the register, the resultant state is $U \otimes I_{2^{n-m}} |\Psi\rangle$; that is, the result is the same as if we had applied the (also unitary) operator $U \otimes I_{2^{n-m}}$ to the whole register.

1.3.2 Quantum Function Queries

Given a function $f\colon \mathcal{M} \to \mathcal{T}$ where $(\mathcal{T}, +)$ is a group, we define the quantum gate U_f by

$$\begin{aligned} U_f\colon \mathbb{C}^{\mathcal{M}} \otimes \mathbb{C}^{\mathcal{T}} &\to \mathbb{C}^{\mathcal{M}} \otimes \mathbb{C}^{\mathcal{T}} \\ |m\rangle\,|y\rangle &\mapsto |m\rangle\,|y + f(m)\rangle \text{ for all } m \in \mathcal{M}, y \in \mathcal{T}, \end{aligned}$$

extended linearly. Thus querying a function f on a superposition $|\psi\rangle$ of inputs is the same as applying U_f to $|\psi\rangle\,|0\rangle$.

[3] We can of course consider applying U to *any* m qubits of the register; we write only this result explicitly for the sake of simplicity of notation. The result is similar.

1.3.3 Measurements

Given a set of qubits in an unknown state, it is impossible to simply determine the state of the qubits; rather, a measurement must be performed to extract some classical information. Measurements are inherently probabilistic; in particular, if a set of qubits is in the state $|\Psi\rangle = \sum_{\sigma\in\{0,1\}^n} \alpha_\sigma |\sigma\rangle$ then for each $\sigma \in \{0,1\}^n$, the classical result of the measurement is σ with probability $|\alpha_\sigma|^2$.[4]

In principle, measurements can appear at any point of a quantum algorithm, and it is possible that not all qubits will be measured at the end of an algorithm. However, the following two principles say that it suffices to consider only those algorithms all of whose measurements are at the end, and which measure all available qubits.

Principle of Deferred Measurement [28, Section 4.4]: Measurements can always be moved from an intermediate stage of a quantum circuit to the end of the circuit; if the measurement results are used at any stage of the circuit then the classically-controlled operations can be replaced by conditional quantum operations.

Principle of Implicit Measurement [28, Section 4.4]: Without loss of generality, any qubits which are not measured at the end of a quantum circuit may be assumed to be measured.

These principles will simplify our analysis of key establishment protocols in Chapter 4.

1.3.4 Distinguishing Quantum States

A fundamental problem in quantum information processing is the following: given a register which is known to be either in state $|\psi\rangle$ or state $|\phi\rangle$, determine which is the case. The following result is a corollary of the Holevo-Helstrom theorem.

Theorem 1.2 (Corollary of the Holevo-Helstrom Theorem)**.** *Suppose a register is prepared in either state $|\psi_0\rangle$ or $|\psi_1\rangle$, each with probability $\frac{1}{2}$. Any quantum algorithm which correctly*

[4] In principle we can consider measurements in other bases; if $\{|\psi_i\rangle\}_{i=1}^{2^n}$ is an orthonormal basis for $\bigotimes_{i=1}^{n} \mathbb{C}^2$, then we can measure $|\Psi\rangle$ in this basis—for $1 \le i \le 2^n$, the result is $|\psi_i\rangle$ with probability $|\langle\Psi|\psi_i\rangle|^2$. Such measurements can be implemented as a unitary operator followed by a standard measurement, and so no generality is lost by considering only the standard measurement.

identifies b such that the actual state prepared is $|\psi_b\rangle$ succeeds with probability at most

$$\frac{1}{2} + \frac{1}{2}\sqrt{1 - |\langle\psi_0|\,\psi_1\rangle|^2}. \tag{1.3}$$

where $\langle\psi_0|\,\psi_1\rangle$ is the standard inner product of $|\psi_0\rangle$ and $|\psi_1\rangle$. Moreover, for each such pair of states, there is a measurement which succeeds with exactly the probability given in Equation (1.3).

This theorem will be useful for analyzing the security of protocols in our security model because the security definition in Chapter 4 relies on the computational indistinguishability of certain quantum states. We will use Theorem 1.2 to show that certain quantum states are indistinguishable provided that certain probability distributions are themselves computationally indistinguishable.

1.4 Functions

We will require some useful notions about functions.

Definition 1.18 (Negligible Function)**.** A function $\epsilon\colon \mathbb{N} \to \mathbb{R}$ is said to be *negligible* if, for every polynomial function $p(\lambda)$ which is positive on $\mathbb{N}$, there is a constant λ_p so that $\epsilon(\lambda) < \frac{1}{p(\lambda)}$ for all $\lambda > \lambda_p$. A function which is not negligible is said to be *non-negligible.*

Negligible functions frequently arise in cryptographic security definitions, and, intuitively, they are used in security definitions to account for the possibility of an adversary accomplishing some task "by chance." The following lemma gives a number of useful properties of negligible functions that we will use in later chapters.

Lemma 1.3. Let $\delta(\lambda)$ and $\epsilon(\lambda)$ be negligible functions, and let $q(\lambda)$ be a polynomial function which is positive on $\mathbb{N}$. Then

i. $\delta(\lambda) + \epsilon(\lambda)$ is negligible.

ii. $\delta(\lambda) \cdot \epsilon(\lambda)$ is negligible.

iii. $q(\lambda)\epsilon(\lambda)$ is negligible.

Proof. i. For any polynomial $p(\lambda)$, let $\lambda_{2p}^{(\delta)}$ and $\lambda_{2p}^{(\epsilon)}$ be such that

$$\delta(\lambda) < \frac{1}{2p(\lambda)} \text{ for all } \lambda > \lambda_{2p}^{(\delta)} \text{ and}$$
$$\epsilon(\lambda) < \frac{1}{2p(\lambda)} \text{ for all } \lambda > \lambda_{2p}^{(\epsilon)}.$$

Then for all $\lambda > \max\{\lambda_{2p}^{(\delta)}, \lambda_{2p}^{(\epsilon)}\}$ we see that

$$\delta(\lambda) + \epsilon(\lambda) < \frac{1}{2p(\lambda)} + \frac{1}{2p(\lambda)} = \frac{1}{p(\lambda)}$$

so that $\delta(\lambda) + \epsilon(\lambda)$ is negligible, as required.

ii. For any polynomial $p(\lambda)$, let $\lambda_1^{(\delta)}$ and $\lambda_p^{(\epsilon)}$ be such that

$$\delta(\lambda) < 1 \text{ for all } \lambda > \lambda_1^{(\delta)} \text{ and}$$
$$\epsilon(\lambda) < \frac{1}{p(\lambda)} \text{ for all } \lambda > \lambda_p^{(\epsilon)}.$$

Then for all $\lambda > \max\{\lambda_1^{(\delta)}, \lambda_p^{(\epsilon)}\}$ we see that

$$\delta(\lambda) \cdot \epsilon(\lambda) < 1 \cdot \frac{1}{p(\lambda)} = \frac{1}{p(\lambda)}$$

so that $\delta(\lambda) \cdot \epsilon(\lambda)$ is negligible, as required.

iii. For any polynomial $p(\lambda)$, let $\lambda_{pq}^{(\epsilon)}$ be such that

$$\epsilon(\lambda) < \frac{1}{p(\lambda)q(\lambda)} \text{ for all } \lambda > \lambda_{pq}^{(\epsilon)}.$$

Clearly this is equivalent to

$$q(\lambda)\epsilon(\lambda) < \frac{1}{p(\lambda)} \text{ for all } \lambda > \lambda_{pq}^{(\epsilon)},$$

so that $q(\lambda)\epsilon(\lambda)$ is negligible.

□

Remark 1.3. Lemma 1.3.i. generalizes to the sum of *polynomially-many* negligible functions.

1.4.1 Hash Functions

For many cryptographic purposes we require a function which is easily computable, and maps input of arbitrary length to a fixed output length. This prompts the following definition.

Definition 1.19 (Hash Function, adapted from [16, Section 8.1])**.** A *hash function* is a function h which takes as input a binary string[5] of any length, and returns a binary string of some fixed (and in particular, finite) length, which can be computed in polynomial time in the length of input.

Often when we want to use hash functions for cryptographic purposes, we require them to have specific security properties. Typical cryptographic security properties are stated as the inability for some polynomial-time adversary to accomplish some task, except with negligible probability. Of course, in order for the notion of "polynomial-time" to be meaningful, what we really need is a *family* of hash functions $\{h_k\}$ with a corresponding security parameter. For instance, we might define the family of hash functions $\mathcal{H} = \{h_{p,g}\}_{\substack{p \text{ prime} \\ g \in \mathbb{Z}_p}}$ by

$$\begin{aligned} h_{p,g} \colon \mathbb{Z} &\to \mathbb{Z}_p \\ x &\mapsto g^x \end{aligned}$$

and then define the security parameter on this family as $\lambda = \lfloor \log_2 p \rfloor$. Then a security property of this family would be stated as "there is no polynomial-time algorithm which,

[5] As discussed in Section 1.3, we can interpret any input m as a binary string, and so we will use hash functions with domains other than $\{0,1\}^*$, keeping in mind that we simply embed the domain in $\{0,1\}^*$ in some canonical way.

given h chosen uniformly at random from $\mathcal{H}$, accomplishes some task in time which is asymptotically polynomial in λ". We state some standard security properties of hash functions here.

Definition 1.20 (Preimage Resistance)**.** A hash function family $\mathcal{H} = \{h_t\}_{t\in\mathcal{T}}$ is said to be *preimage resistant* if there is no polynomial-time algorithm which, given h chosen uniformly at random from $\mathcal{H}$, and $y \in \operatorname{im} h$, returns $x \in \operatorname{dom} h$ such that $y = h(x)$ with non-negligible probability.

Definition 1.21 (Second Preimage Resistance)**.** A hash function family $\mathcal{H} = \{h_t\}_{t\in\mathcal{T}}$ is said to be *second preimage resistant* if there is no polynomial-time algorithm which, given h chosen uniformly at random from $\mathcal{H}$, and $x \in \operatorname{dom} h$, returns $x' \in \operatorname{dom} h \backslash \{x\}$ such that $h(x) = h(x')$ with non-negligible probability.

Definition 1.22 (Collision Resistance)**.** A hash function family $\mathcal{H} = \{h_t\}_{t\in\mathcal{T}}$ is said to be *collision resistant* if there is no polynomial-time algorithm which, given h chosen uniformly at random from $\mathcal{H}$, finds $x, x' \in \operatorname{dom} h$ such that $x \neq x'$ and $h(x) = h(x')$ with non-negligible probability.

1.5 Computational Assumptions

Often in order to prove that a cryptographic protocol is secure we need that some underlying computational problem is hard in some sense—for typical security definitions such as those given in Chapter 2, we need that the problem cannot be solved in polynomial time on a classical or quantum computer. The problems that underlie most cryptographic algorithms are in NP; that is, there exist polynomial-size certificates for "yes" instances of problems that can be verified in polynomial time. A proof that such a problem has no polynomial-time solution would prove that P $\neq$ NP. Given that no such proof currently exists, cryptographers make security *assumptions*; that is, they assume that there is no polynomial-time algorithm for a given problem. Then a typical security proof follows the argument that, if an efficient adversary can break the security of a protocol, then the underlying problem can be solved in polynomial time, breaking the hardness assumption. In this section we give some classical examples of security assumptions. and then describe the

security properties required for supersingular isogeny-based cryptographic schemes, which appear in later chapters.

1.5.1 Polynomial-Time Reducibility of Computational Problems

In order to quantify security we would like to be able to determine the "relative hardness" of computational problems—in particular, we typically want to say that compromising the security of a cryptosystem is harder than solving some standard problem. To this end we introduce the notion of *polynomial-time reducibility*.

Definition 1.23 (Polynomial-Time Reducibility)**.** Let C and D be computational problems. We say that C is *polynomial-time reducible* to D, denoted $D \geq_P C$, if there is an algorithm $\mathcal{A}$ which, using a polynomial number of queries to an oracle which solves D and polynomial time otherwise, solves instances of C.

The notation $D \geq_P C$ is meant to suggest that we are ordering problems by difficulty; if C is polynomial-time reducible to D, then being able to solve D efficiently allows one to solve C efficiently, and in that sense D is "harder" than C.

1.5.2 Examples of Classical Computational Assumptions

Factoring and the RSA Problem

Perhaps the best known public-key encryption scheme is RSA, which is based on the believed hardness of factoring and related problems. Because of its historical and instructive value, we discuss the computational problems associated with RSA here, and refer to them in Chapter 2 when we give examples of signature schemes.

Definition 1.24 (Factoring Problem for RSA Moduli)**.** Given an RSA modulus $n = pq$ where p and q are distinct odd primes, the *factoring problem* is to determine p and q.

The corresponding computational assumption, called the *factoring assumption* for RSA moduli is that there is no polynomial-time algorithm which solves the factoring problem

for RSA moduli on arbitrary inputs with non-negligible probability, where the security parameter is $\lambda = \min\{\lfloor \log_2 p \rfloor, \lfloor \log_2 q \rfloor\}$.

The security of the RSA cryptosystem and signature scheme is based on the hardness of factoring RSA moduli in sense that if factoring is easy, then these schemes are certainly insecure. However, there is no known proof of the contrapositive statement; that is, if factoring is difficult, then the schemes are secure. Unfortunately, this is not sufficient for a security proof; to rectify the issue, the following computational problem was devised.

Definition 1.25 (RSA Problem [30])**.** Given an RSA modulus $n = pq$, an integer $e \in (\mathbb{Z}/\varphi(n)\mathbb{Z})^*$ (where φ is Euler's totient function), and a number $c \in \mathbb{Z}/n\mathbb{Z}$, the *RSA problem* is to find $c^d \pmod n$, where d is the unique integer in $\mathbb{Z}/\varphi(n)\mathbb{Z}$ which satisfies $ed \equiv 1 \pmod{\varphi(n)}$.

The *RSA assumption* is that there is no polynomial-time algorithm which solves arbitrary instances of the RSA problem with non-negligible probability. This assumption is standard in classical cryptography. It is clear that

$$\text{Factoring} \ \geq_P \ \text{RSA}.$$

Discrete Logarithms and the Diffie-Hellman Problem

Let $G = \langle g \rangle$ be a group of order n. For each $h \in G$ there exists a unique $x \in \{0, 1, \ldots, n-1\}$ such that $h = g^x$. We call this number x the *discrete logarithm of h with respect base g*, denoted $\log_g h$.

Definition 1.26 (Discrete Logarithm Problem)**.** Given a group $G = \langle g \rangle$ and an element $h \in G$, the *Discrete Logarithm problem (DLP)* is to find $\log_g h$.

For a given group G, the *Discrete Logarithm Assumption* for G is that there is no polynomial-time (in size of input) algorithm which solves arbitrary instances of DLP in G with non-negligible probability.

For a group $G = \langle g \rangle$ of order n, we call a triple $(a, b, c) \in G^3$ a *Diffie-Hellman triple* if there are $x, y \in \{0, 1, \ldots, n\}$ such that $a = g^x, b = g^y$ and $c = g^{xy}$. There are two common security properties related to Diffie-Hellman triples, which we state here

Definition 1.27 (Computational Diffie-Hellman Problem [10])**.** Given a group $G = \langle g \rangle$ and two elements a and b with $a = g^x$ and $b = g^y$, where x and y are *not* given, the *Computational Diffie-Hellman problem (CDH)* is to find g^{xy}.

Definition 1.28 (Decisional Diffie-Hellman Problem [10])**.** Given a group $G = \langle g \rangle$ and a triple $(a, b, c) \in G^3$, the *Decisional Diffie-Hellman problem (DDH)* is to determine whether (a, b, c) is a Diffie-Hellman triple.

The *Computational Diffie-Hellman assumption* for a given group G is that there is no polynomial-time (in $\lambda = \lfloor \log_2 n \rfloor$) algorithm which solves arbitrary instances of the CDH problem in G with non-negligible probability; similarly, the *Decisional Diffie-Hellman assumption* in a group G is that there is no polynomial-time algorithm which solves arbitrary instances of the DDH problem in G with non-negligible advantage over $\frac{1}{2}$. For certain groups, these assumptions are standard. Moreover, it is clear that for any group

$$\text{DLP} \ \geq_P \ \text{CDH} \ \geq_P \ \text{DDH}.$$

1.5.3 Post-Quantum Computational Assumptions: Supersingular Elliptic Curve Isogenies

Unfortunately, many classical security assumptions—including, in particular, all the examples in Section 1.5.2—are known to be incorrect when we consider quantum algorithms. In particular, Shor's algorithm [32] solves both the factoring problem and the discrete logarithm problem in polynomial-time, completely invalidating the security of RSA and discrete logarithm-based cryptosystems. To that end, in order to ensure the security of classical communications once quantum computers are developed we develop classical protocols which are resistant to attacks by quantum computers. The security of the vast majority of current post-quantum cryptographic schemes is based upon the hardness of problems in four broad areas: lattices (*e.g.,* NTRU [15] and variants [19, 8]), multivariate polynomials over finite fields (*e.g.,* Unbalanced Oil and Vinegar [21]; Hidden Field Equations [29]), hash functions (*e.g.,* Merkle signatures [25]), and algebraic codes (*e.g.*, McEliece [24]; Niederreiter [27]). Recently, however, cryptographic schemes whose security is based on the quantum hardness of computing isogenies between supersingular elliptic

curves have been proposed [9, 35, 18]. These schemes have extremely promising security and efficiency results, and the underlying security assumptions are, in some sense, very similar to the Diffie-Hellman problems. We will use these security assumptions to build a secure authenticated key establishment protocol in Chapter 5, and so we present them here.

In the following definitions, let $p = \ell_A^{e_A}\ell_B^{e_B} f \pm 1$ be a prime, where ℓ_A and ℓ_B are distinct small primes, and f is a small cofactor used to ensure that p is prime—we make no effort to quantify what is meant by "small." Moreover, let E be a supersingular elliptic curve defined over $\mathbb{K} = GF(p^2)$ with $E(GF(p^2)) \cong \mathbb{Z}_{p\mp1} \oplus \mathbb{Z}_{p\mp1}$, and let $\{P_A, Q_B\}$ and $\{P_B, Q_B\}$ be bases for $E[\ell_A^{e_A}]$ and $E[\ell_B^{e_B}]$, respectively.

Definition 1.29 (Supersingular Isogeny Problem)**.** Let $\phi_A \colon E \to E_A$ be an isogeny with kernel $\langle m_A P_A + n_A Q_A \rangle$ where m_A, n_A are chosen uniformly at random from $\mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$, not both divisible by ℓ_A. The *supersingular isogeny problem (SSI)* is, given E, E_A, $\phi_A(P_B)$, and $\phi_A(Q_B)$, to find a generator of $\ker \phi_A$.

Definition 1.30 (Supersingular Computational Diffie-Hellman Problem)**.** Let $\phi_A \colon E \to E_A$ be an isogeny with kernel $\langle m_A P_A + n_A Q_A \rangle$ where m_A, n_A are chosen uniformly at random from $\mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$, not both divisible by ℓ_A. Similarly, let $\phi_B \colon E \to E_B$ be an isogeny with kernel $\langle m_B P_B + n_B Q_B \rangle$ where m_B, n_B are chosen uniformly at random from $\mathbb{Z}/\ell_B^{e_B}\mathbb{Z}$, not both divisible by ℓ_B. The *supersingular computational Diffie-Hellman problem (SSCDH)* is to find the j-invariant of

$$E_{AB} = E/\langle m_A P_A + n_A Q_A, m_B P_B + n_B Q_B \rangle$$

given E, E_A, E_B, $\phi_A(P_B)$, $\phi_A(Q_B)$, $\phi_B(P_A)$, and $\phi_B(Q_A)$.

Definition 1.31 (Supersingular Decisional Diffie-Hellman Problem)**.** Let $\phi_A \colon E \to E_A$ be an isogeny with kernel $\langle m_A P_A + n_A Q_A \rangle$ where m_A, n_A are chosen uniformly at random from $\mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$, not both divisible by ℓ_A. Similarly, let $\phi_B \colon E \to E_B$ be an isogeny with kernel $\langle m_B P_B + n_B Q_B \rangle$ where m_B, n_B are chosen uniformly at random from $\mathbb{Z}/\ell_B^{e_B}\mathbb{Z}$, not both divisible by ℓ_B. Given a tuple

$$(E, E_A, E_B, \phi_A(P_B), \phi_A(Q_B), \phi_B(P_A), \phi_B(Q_A), E_C)$$

where either $E_C = E_{AB} = E/\langle m_A P_A + n_A Q_A, m_B P_B + n_B Q_B \rangle$ or E_C is sampled uniformly at random from the set of all curves of the form

$$E/\langle x_A P_A + y_A Q_A, x_B P_B + y_B Q_B \rangle$$

where x_A, y_A and x_B, y_B are chosen with the same conditions as m_A, n_A and m_B, n_B, each with probability $\frac{1}{2}$, the *supersingular decisional Diffie-Hellman problem (SSDDH)* is to determine which is the case.

As in previous sections, the corresponding security assumptions is that arbitrary instances of the above problems cannot be solved in polynomial-time with non-negligible probability (non-negligible advantage in the case of SSDDH). At the time of writing, the best known quantum algorithm for these problems runs in fully-exponential time $\mathcal{O}(\sqrt[6]{p})$ [4]; thus these problems seem well-suited to being cryptographic primitives. Finally, observe that

$$\text{SSI} \geq_P \text{SSCDH} \geq_P \text{SSDDH}.$$

Chapter 2

Public-Key Cryptography

In this chapter we discuss the fundamental concepts in public-key cryptography which allow us to achieve authenticated key establishment. In particular, we discuss unauthenticated key establishment protocols, and consider signature schemes as an authentication method. In order to give specific examples of quantum-safe signature schemes we will introduce the quantum random oracle model and given generic constructions that amplify security properties of signature schemes in this model. We also introduce chameleon hash functions as a tool for these constructions.

2.1 The Quantum Random Oracle Model

2.1.1 Random Oracles and the Random Oracle Model

For the purposes of provable security, often we model hash functions with desirable security properties as an idealized, completely random function called a *random oracle*. In this paradigm, known as the *random oracle model*, an adversary in a security game can only obtain hash values by requesting them from the challenger. The challenger can maintain a table of hash values and, whenever an oracle call is made check the table for a matching entry and returns it or, if none is found, generate a uniformly random value and records it as the hash value for that input. The main benefit of this choice is that it prevents

the adversary from using the structure of the hash function to obtain forgeries. Use of the random oracle model is contentious among cryptographers since true random oracles cannot be feasibly realized, and because it is known that there exist protocols which are secure in the random oracle model but insecure when the random oracle is replaced by *any* concrete hash function [6]. Nevertheless, because of their usefulness for provable security, we use the random oracle model when necessary or when it greatly simplifies proofs.

2.1.2 Random Oracles in the Quantum Setting

When we move to the quantum setting, in recognition of the fact that any actual implementation will use a concrete hash function the most natural thing to do is to allow quantum calls to the random oracle; this is known as the *quantum* random oracle model. It is easy to see how this might cause trouble; it is not clear that the challenger can generate new random values for queries to new inputs while at the same time maintaining consistency with previously-returned hash values. For this reason, standard proof techniques in the classical random oracle model do not necessarily translate well to the quantum setting. Fortunately some progress has been made in developing proof techniques and constructions that work in the quantum random oracle model (see, for instance, [37, 39]); in particular, in Section 2.4.4 we detail a construction due to Eaton and Song [11] which can be used to construct a signature scheme which is secure in the quantum random oracle model from a signature scheme which is merely secure in the classical random oracle model against an adversary who can perform quantum computations. This construction will allow us to construct a secure authenticated key establishment protocol in the model we present in Chapter 4.

2.2 Chameleon Hash Functions

Intuitively, chameleon hash functions, introduced in 1997 by Krawczyk and Rabin [22], are a special type of hash function which are collision resistant for anybody who does not know an associated piece of secret information, but for which collisions can easily be found for any input given that piece of secret information. Since their introduction they have been used to establish signature schemes with many desirable properties; particularly non-repudiation, non-transferability, and recipient-specificity, which will not be discussed

here. For our purposes, we will use these to construct signature schemes which are secure in the quantum random oracle model. A precise definition of chameleon hash function is given in Definition 2.1.

Definition 2.1 (Chameleon Hash Function (Adapted from [5, Definition 3.9]))**.**

A *chameleon hash function* $\mathcal{H}$ is a tuple $(\text{KeyGen}, H, \text{Invert}, \text{Sample})$ of algorithms such that:

1. $\text{KeyGen}(\lambda)$ generates a private key/public key pair (sk, pk) with security parameter λ;
2. $H_{\text{pk}}(m, r)$ maps messages m to some target space $\mathcal{Y}$;
3. $\text{Sample}(\lambda)$ samples r in such a way that $H_{\text{pk}}(m, r)$ is distributed computationally indistinguishably from uniform over the image of $H_{\text{pk}}(m, \cdot)$ for every pair (pk, m);
4. $\text{Invert}_{\text{sk}}(h, m)$ produces r such that $H_{\text{pk}}(m, r) = h$ (where (sk, pk) is generated by $\text{KeyGen}(\lambda)$), with distribution computationally indistinguishable from that of $\text{Sample}(\lambda)$ conditioned on $H_{\text{pk}}(m, r) = h$; and,
5. For any pk, $H_{\text{pk}}(\cdot, \cdot)$ is collision resistant.

We say that a chameleon hash function is *quantum-safe* if the collision-resistance property holds against a quantum adversary who can query the function in superposition.

2.3 Unauthenticated Key Establishment

Before discussing security models for authenticated key establishment we briefly discuss the high-level idea of unauthenticated key establishment and provide some examples. Informally, an unauthenticated key establishment protocol is a way for a number of individuals to exchange messages in public (*i.e.,* potentially with eavesdroppers) and obtain a shared secret key (*i.e.,* some string that eavesdroppers cannot determine). Secrecy of the key is typically derived from the assumed intractability of some computational problem. In the two protocols we present, parties who wish to establish keys choose ephemeral secret keys

from which they derive ephemeral public values, which they exchange. Each party then, using the other's ephemeral public value and their own ephemeral secret keys constructs a session key; if the construction is suitably chosen they both get the same key. The computational assumption then is that it is difficult to "combine" the two public values without knowing at least one of the underlying secret values. The prototypical example of an unauthenticated key establishment protocol is the Diffie-Hellman protocol.

Diffie-Hellman Key Establishment

Definition 2.2 (Diffie-Hellman Key Establishment [10])**.** Suppose two parties A and B wish to establish a key.

Global Parameters: Fix a cyclic group G of order n, and choose a generator g of G.

Ephemeral Key Generation: A chooses $a \in \mathbb{Z}_n$ uniformly at random. Her ephemeral secret key is a; her ephemeral public value is $\alpha = g^a$. Similarly, B chooses $b \in \mathbb{Z}_{n-1}$ uniformly at random. His ephemeral secret key is b; his ephemeral public value is $\beta = g^b$.

Key Construction: A computes $K_a = \beta^a = (g^b)^a = g^{ab}$. B computes $K_b = \alpha^b = (g^b)^a = g^{ab}$. A and B share the same key.

Diffie-Hellman key establishment is illustrated in Figure 2.3

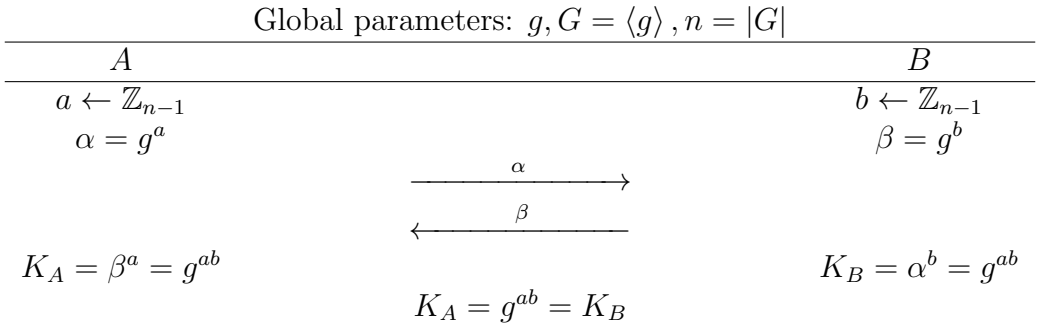

Figure 2.1: Diffie-Hellman Key Establishment

Notice that, by definition, if the Computational Diffie-Hellman assumption holds in G, then no eavesdropper can compute the secret key for an instance of the Diffie-Hellman protocol. Unfortunately, since the discrete logarithm problem can be solved efficiently by quantum computers, Diffie-Hellman key establishment cannot be quantum-safe.

Supersingular Elliptic Curve Isogeny Key Establishment

Definition 2.3 (Supersingular Elliptic Curve Isogeny Key Establishment (SSIKE) ([9]))**.** Suppose two parties A and B wish to establish a key.

Global Parameters: Fix a prime $p = \ell_A^{e_A}\ell_B^{e_B} f \pm 1$ where ℓ_A and ℓ_B are small primes; a supersingular elliptic curve E defined over $GF(p^2)$ such that $E[\ell_A^{e_A}]$ and $E[\ell_B^{e_B}]$ are defined over $GF(p^2)$, and; bases $\{P_A, Q_A\}$ and $\{P_B, Q_B\}$ for $E[\ell_A^{e_A}]$ and $E[\ell_B^{e_B}]$, respectively.

Ephemeral Key Generation: A chooses $m_A, n_A \in \mathbb{Z}_{\ell_A^{e_A}}$, not both divisible by ℓ_A, uniformly at random. Letting ϕ_A be the isogeny with domain E and kernel $\langle m_A P_A + n_A Q_A \rangle$, her ephemeral secret key is (m_A, n_A) and her ephemeral public value is

$$\alpha = (E_A = E/\langle m_A P_A + n_A Q_A \rangle, \phi_A(P_B), \phi_A(Q_B)).$$

Analogously, B chooses $m_B, n_B \in \mathbb{Z}_{\ell_B^{e_B}}$, not both divisible by ℓ_B, uniformly at random. Letting ϕ_B be the isogeny with domain E and kernel $\langle m_B P_B + n_B Q_B \rangle$, his ephemeral secret key is (m_B, n_B) and his ephemeral public value is

$$\beta = (E_B = E/\langle m_B P_B + n_B Q_B \rangle, \phi_B(P_A), \phi_B(Q_A)).$$

Key Construction: A computes

$$K_a = j\left(E_B/\langle m_A \phi_B(P_A) + n_A \phi_B(Q_A)\rangle\right) = \left(E/\langle m_A P_A + n_A Q_A, m_B P_B + n_B Q_B\rangle\right).$$

B computes

$$K_b = \left(E_A/\langle m_B \phi_A(P_B) + n_B \phi_A(Q_B)\rangle\right) = \left(E/\langle m_B P_B + n_B Q_B, m_A P_A + n_A Q_A\rangle\right).$$

We have $K_a = K_b$.

It is clear that if the supersingular isogeny computational Diffie-Hellman problem is intractable, then an eavesdropper cannot compute the derived key using the public information exchanged in an instance of this protocol. SSCDH is conjectured to be infeasible

for a quantum computer, and so SSIKE is a candidate for post-quantum unauthenticated key establishment.

2.4 Signature Schemes

Definition 2.4 (Signature Scheme)**.** A *signature scheme* is a triple $(\text{KeyGen}, \text{Sign}, \text{Verify})$, where:

1. KeyGen is the key generation algorithm, which takes in a natural number λ and outputs a key pair (sk, pk);
2. Sign is the (possibly randomized) signing algorithm, which takes in a message and outputs a signature; and,
3. Verify is the verification algorithm, which takes in a message and signature and outputs 1 if the signature is valid, and 0 otherwise.

When Sign is not a randomized algorithm we say that $(\text{KeyGen}, \text{Sign}, \text{Verify})$ is *deterministic*, and otherwise it is non-deterministic; when a signature scheme is non-deterministic, if we wish to specify the signature on a message m signed with key sk and randomness r, we write $\text{Sign}_{\text{sk}}(m; r)$.

A signature scheme is *correct* if, whenever (sk, pk) is a valid private key/public key pair, $\text{Verify}_{\text{pk}}(m, \text{Sign}_{\text{sk}}(m)) = 1$ for all messages M, and if σ is *not* a valid signature for m under private key sk, then $\text{Verify}_{\text{pk}}(m, \sigma) = 0$.

2.4.1 Security of Signature Schemes

As with any type of cryptosystem, we must define what it means for a signature scheme to be secure; naturally, there are many possible security definitions. We present the most common classical and quantum definitions here; these will be sufficient for results in later chapters.

Definition 2.5 (Strong EUF-RMA Security)**.** A signature scheme $(\text{KeyGen}, \text{Sign}, \text{Verify})$ is *strongly existentially unforgeable against a random message attack (strongly EUF-RMA secure)* if the advantage that any polynomial-time adversary has at winning the following game is negligible:

1. The challenger $\mathcal{C}$ runs the key generation algorithm on input 1^λ to obtain the key pair (sk, pk), and publishes pk.
2. The adversary $\mathcal{A}$ sends an integer t to $\mathcal{C}$.
3. $\mathcal{C}$ selects messages $m_1, \ldots, m_t$ uniformly at random, and computes $\sigma_i = \text{Sign}_{\text{sk}}(m_i)$ for $1 \leq i \leq t$. $\mathcal{C}$ returns $\{(m_i, \sigma_i)\}_{i=1}^t$.
4. $\mathcal{A}$ produces (m^*, σ^*). $\mathcal{A}$ wins the game if $(m^*, \sigma^*) \neq (m_i, \sigma_i)$ for $1 \leq i \leq t$ and $\text{Verify}_{\text{pk}}(m^*, \sigma^*) = 1$.

We say that a signature scheme is weakly EUF-RMA secure if it satisfies the requirements of Definition 2.5 with the additional requirement that, in step 4. of the game, $m^* \neq m_i$ for $1 \leq i \leq t$.

Definition 2.6 (Strong EUF-CMA Security)**.** A signature scheme $(\text{KeyGen}, \text{Sign}, \text{Verify})$ is *strongly existentially unforgeable against an adaptive chosen message attack (strongly EUF-CMA secure)* if the advantage that any polynomial-time adversary has at winning the following game is negligible:

1. The challenger $\mathcal{C}$ runs the key generation algorithm on input 1^λ to obtain the key pair (sk, pk), and publishes pk.
2. For $i = 1, 2, \ldots, t$:
   a) The adversary $\mathcal{A}$ sends a message m_i to $\mathcal{C}$.
   b) $\mathcal{C}$ returns $\sigma_i = \text{Sign}_{\text{sk}}(m_i)$.
3. $\mathcal{A}$ produces (m^*, σ^*). $\mathcal{A}$ wins the game if $(m^*, \sigma^*) \neq (m_i, \sigma_i)$ for $1 \leq i \leq t$ and $\text{Verify}_{\text{pk}}(m^*, \sigma^*) = 1$.

We say that a signature scheme is weakly EUF-CMA secure if it satisfies the requirements of Definition 2.6 with the additional requirement that, in step 3. of the game, $m^* \neq m_i$ for $1 \leq i \leq t$.

In Definitions 2.5 and 2.6, part of the win condition for the adversary in the game is that the adversary's output message/signature pair must not have been provided by the challenger. These definitions are reasonable for classical security definitions, but they break down when we consider an adversary who is allowed to obtain signatures on quantum superpositions of messages; this is because there is not a reasonable notion of which messages the adversary has asked $\mathcal{C}$ to sign. This necessitates the following *quantum-safe* security definition.

Definition 2.7 (Strong EUF-qCMA Security)**.** A signature scheme $(\text{KeyGen}, \text{Sign}, \text{Verify})$ is *strongly existentially unforgeable against a quantum adaptive chosen message attack (strongly EUF-qCMA secure)* if the advantage that any polynomial-time adversary has at winning the following game is negligible:

1. The challenger $\mathcal{C}$ runs the key generation algorithm on input 1^λ to obtain the key pair (sk, pk), and publishes pk.

2. For $i = 1, 2, \ldots, t$:

   a) The adversary $\mathcal{A}$ sends a register in superposition of messages and target qubits $|\psi_i\rangle = \sum_{m,y} \alpha_{m,y} |m\rangle |y\rangle$.

   b) $\mathcal{C}$ returns $U_{\text{Sign}_{\text{sk}}} |\psi_i\rangle = \sum_{m,y} \alpha_{m,y} |m\rangle |y + \text{Sign}_{\text{sk}}(m)\rangle$

3. $\mathcal{A}$ produces $(m_1^*, \sigma_1^*), (m_2^*, \sigma_2^*), \ldots, (m_{t+1}^*, \sigma_{t+1}^*)$. $\mathcal{A}$ wins the game if the (m_i^*, σ_i^*) are distinct and $\text{Verify}_{\text{pk}}(m_i^*, \sigma_i^*) = 1$ for $1 \leq i \leq t+1$.

We say that a signature scheme is weakly EUF-qCMA secure if it satisfies the requirements of Definition 2.7 with the additional requirement that, in step 3. of the game, the messages m_i are pairwise distinct.

The requirement that to win $\mathcal{A}$ must produce one more valid message/signature pair than signing queries he makes is intuitively justified by the fact that the output of each such query, when measured, will yield a valid message/signature pair, and so it is trivial for $\mathcal{A}$ to produce t such pairs. The $(t+1)^{\text{th}}$ such pair is the forgery.

2.4.2 Security of Signatures in Random Oracle Models

The security definitions presented in the previous section intentionally make no mention of random oracles. For the purposes of this work we will always assume that a quantum adversary has quantum access to any hash functions used in a protocol unless otherwise specified. In the event that we do wish to specify the model in which the security property holds, we modify the security property name appropriately; for quantumly-accessible random oracles we append -QRO, and for only classically-accessible random oracles we append -RO.

2.4.3 Examples of Signature Schemes

RSA Signatures

For the sake of demonstrating the concept, here we present arguably the simplest signature scheme: RSA signatures [30, Section IV]. As might be expected from the name, the computational problem whose hardness underlines RSA signature is the RSA problem (Definition 1.25).

Definition 2.8 (RSA Signatures (Adapted from [30, Sections IV-VI]))**.**

Global Parameters: A random oracle $\mathcal{O}$.

Key Generation: On input 1^λ choose two distinct primes p, q with $\lfloor \log_2 p \rfloor = \lfloor \log_2 q \rfloor = \lambda$, and define $n = pq$. Choose $e \in (\mathbb{Z}\backslash\varphi(n)\mathbb{Z})^*$, and let $d \equiv e^{-1} \pmod{\varphi(n)}$. The private key is d. The public key is (n, e).

Signing: To sign a message $m \in \mathbb{Z}$, compute $\sigma = \mathcal{O}(m)^d \pmod{n}$. The signature is σ.

Verification: To verify a signature σ on a message m, compute $\sigma^e \pmod{n}$. If the result is $\mathcal{O}(m)$, the signature is valid; otherwise it is invalid.

RSA signatures are EUF-CMA-RO secure under the RSA assumption, but of course they are not quantum-safe, since the RSA assumption does not hold against quantum adversaries.

Signatures from Supersingular Elliptic Curve Isogenies

Here we present a signature scheme due to Sun *et al.* [35] which is secure even when the adversary can perform polynomially-bounded quantum computations, under the SSCDH assumption. This signature scheme is what is known as a *strong designated verifier (SDV)* signature scheme; it differs from a standard signature scheme in that, in addition to the signer, the verifier must possess a private key/public key pair used for verification. When the signer wants to sign a message, he chooses an intended recipient and uses their public key (along with his own secret key) to sign the message; then only the intended recipient can verify the signature. For the purposes of authentication in key establishment, this in some sense limited form of signature is sufficient, and we will use it to construct a secure protocol in Chapter 5.

Definition 2.9 (Sun SDV Signatures [35, Section III])**.**

Public parameter setup: On input 1^λ, choose two distinct small primes ℓ_S and ℓ_V, integers e_S and e_V, and a small cofactor f such that $p = \ell_S^{e_S}\ell_V^{e_V} f \pm 1$ is prime, $e_S \log_2 \ell_S \approx e_V \log_2 \ell_V$, and $\lfloor \log_2 p \rfloor = \lambda$. Choose an elliptic curve E which is supersingular and defined over $\mathbb{K} = GF(p^2)$, and bases $\{P_S, Q_S\}$ and $\{P_V, Q_V\}$ for $E[\ell_S^{e_S}]$ and $E[\ell_V^{e_V}]$, respectively. These are the global parameters.

Signing Key Generation: On input 1^λ, choose two integers $m_S, n_S \in \mathbb{Z}/\ell_S^{e_S}\mathbb{Z}$, not both divisible by ℓ_S, uniformly at random. The private key is (m_S, n_S).

Set $R_S = m_S P_S + n_S Q_S$, and define ϕ_S to be the isogeny with kernel $\langle R_S \rangle$. Set $E_S = E/\langle R_S \rangle$. The public key is $(E_S, \phi_S(P_V), \phi_S(Q_V))$.

Verification Key Generation: On input 1^λ, choose two integers $m_V, n_V \in \mathbb{Z}/\ell_V^{e_V}\mathbb{Z}$, not both divisible by ℓ_V, uniformly at random. The private key is (m_V, n_V).

Set $R_V = m_V P_V + n_V Q_V$, and define ϕ_V to be the isogeny with kernel $\langle R_V \rangle$. Set $E_V = E/\langle R_V \rangle$. The public key is $(E_V, \phi_V(P_S), \phi_V(Q_S))$.

Signing: To sign a message m with private signing key (m_S, n_S) and public verification key $(E_V, \phi_V(P_S), \phi_V(Q_S))$, compute

$$E_{SV} = E_V/\langle m_S\phi_V(P_S) + n_S\phi_V(Q_S)\rangle\,;$$

the signature is then $\sigma = H(m||j(E_{SV}))$.

Verification: To verify a signature σ on m with public signing key $(E_S, \phi_S(P_V), \phi_S(Q_V))$ and private verification key m_V, n_V, compute

$$E_{VS} = E_S / \langle m_V \phi_S(P_V) + n_V \phi_S(Q_V) \rangle ;$$

the signature is valid if $\sigma = H(m||j(E_{VS}))$.

Theorem 2.1 (Correctness of Sun SDV Signatures)**.** *The scheme described in Definition 2.9 is correct.*

Proof. Observe that

$$\begin{aligned} E_{SV} &= E_V / \langle m_S \phi_V(P_S) + n_S \phi_V(Q_S) \rangle \\ &= (E / \langle m_V P_V + n_V Q_V \rangle) / \langle \phi_V(m_S P_S + n_S Q_S) \rangle \\ &= E / \langle m_V P_V + n_V Q_V, m_S P_S + n_S Q_S \rangle \\ &= (E / \langle m_S P_S + n_S Q_S \rangle) / \langle \phi_S(m_V P_V + n_V Q_V) \rangle \\ &= E_S / \langle m_V \phi_S(P_V) + n_V \phi_S(Q_V) \rangle = E_{VS}; \end{aligned}$$

the result follows. □

Theorem 2.2 (Security of Sun SDV Signatures; Derived from [35, Proposition IV.1])**.** *Under the SSCDH assumption, the signature scheme described in Definition 2.9 is strongly EUF-CMA-RO secure against an adversary who can perform polynomially-bounded quantum computations.*

2.4.4 Secure Signatures in the Quantum Random Oracle Model

We would like to use a signature scheme for authentication in an authenticated key establishment protocol which is secure in a post-quantum model where the adversary can make quantum queries to oracles which emulate classical parties, and to any random oracles. For this purpose, it is clear that we require a signature scheme which is, at the very least, EUF-qCMA secure in the quantum random oracle model (EUF-qCMA-QRO

secure). Unfortunately, the Sun strong designated verifier signature scheme has only been proven to be strongly EUF-CMA-RO secure. In this section we describe a construction which yields a strongly EUF-CMA-QRO secure signature scheme from a strongly or weakly EUF-CMA-RO signature scheme, which solves part of this problem.

Theorem 2.3 (Construction of Strongly EUF-CMA-QRO Signatures [11, Theorem 4])**.** *Let* $(\text{KeyGen}, \text{Sign}, \text{Verify})$ *be a signature scheme which is EUF-CMA-RO against a quantum adversary, and let* $\mathcal{H} = (\text{KeyGen}^{(H)}, H, \text{Sample}, \text{Invert})$ *be a quantum-safe chameleon hash function. Define a new signature scheme* $(\text{KeyGen}^{(Q)}, \text{Sign}^{(Q)}, \text{Verify}^{(Q)})$ *in the following way:*

1. $\text{KeyGen}^{(Q)}(\lambda)$*:*
   (a) *Set* $(\text{sk}, \text{pk}) = \text{KeyGen}(\lambda)$
   (b) *Set* $(\text{sk}^{(H)}, \text{pk}^{(H)}) = \text{KeyGen}^{(H)}(\lambda)$
   (c) *Return* $(\text{sk}^{(Q)} = (\text{sk}, \text{sk}^{(H)}), \text{pk}^{(Q)} = (\text{pk}, \text{pk}^{(H)}))$
2. $\text{Sign}^{(Q)}_{(\text{sk},\text{sk}^{(H)})}(m)$*:*
   (a) *Set* $r^{(H)} = \text{Sample}(\lambda)$
   (b) *Set* $\sigma = \text{Sign}_{\text{sk}}(r^{(H)})$
   (c) *Set* $M = \mathcal{O}(m||\sigma)$*, where* $\mathcal{O}$ *is a random oracle*
   (d) *Set* $r_I = \text{Invert}(\text{sk}^{(H)}, r^{(H)}, M)$
   (e) *Return* $\sigma^{(Q)} = (\sigma, r_I)$
3. $\text{Verify}^{(Q)}_{(\text{pk},\text{pk}^{(H)})}(m, (\sigma, r_I))$*:*
   (a) *Set* $M = \mathcal{O}(m||\sigma)$
   (b) *Set* $r^{(H)} = H^{(H)}_{\text{pk}}(M, r_I)$
   (c) *Output* $\text{Verify}_{\text{pk}}(r^{(H)}, \sigma)$

The signature scheme $(\text{KeyGen}^{(Q)}, \text{Sign}^{(Q)}, \text{Verify}^{(Q)})$ *is correct; moreover, it is also EUF-CMA-QRO secure.*

2.4.5 Generic Construction of EUF-qCMA Secure Signatures

In Section 4.5 we give a generic construction for a secure authenticated key establishment protocol which uses an EUF-qCMA signature scheme as a subroutine; in order for this to be useful, we of course need to have such a signature scheme. In this section we present a construction due to Boneh and Zhandry [5] which yields an EUF-qCMA signature scheme from a signature scheme which is EUF-CMA against an adversary who can perform quantum computations, and quantumly-accessible random oracles.

Theorem 2.4 (Construction of EUF-qCMA Signatures [5, Construction 3.12])**.** *Let* $\mathcal{S} = (\mathrm{KeyGen}, \mathrm{Sign}, \mathrm{Verify})$ *be a signature scheme which is EUF-CMA against an adversary who can perform quantum computations. Let* $\mathcal{Q}$ *be a set of pairwise independent functions, and let* H *be a random oracle. Define* $\mathcal{S}' = (\mathrm{KeyGen}', \mathrm{Sign}', \mathrm{Verify}')$ *in the following way:*

1. $\mathrm{KeyGen}'(1^\lambda) = \mathrm{KeyGen}(1^\lambda)$.
2. *To sign a message* m *with private key* sk*, choose* $Q \in \mathcal{Q}$ *and* $r \in \{0,1\}^k$ *at random. The signature is*
$$\mathrm{Sign}'_{\mathrm{sk}}(m) = (r, \mathrm{Sign}_{\mathrm{sk}}(H(m,r); Q(m)))$$
3. *To verify a signature* (r, σ) *by a party with key pair* $(\mathrm{sk}, \mathrm{pk})$ *on a message* m*, we compute*
$$\mathrm{Verify}'_{\mathrm{pk}}(m, (r, \sigma)) = \mathrm{Verify}_{\mathrm{pk}}(H(m,r), \sigma).$$

The signature scheme $\mathcal{S}'$ *is correct; moreover, if* $\mathcal{S}$ *is EUF-CMA-RO secure against an adversary who can perform quantum computations, then* $\mathcal{S}'$ *is EUF-qCMA-RO secure, and if* $\mathcal{S}$ *is EUF-CMA-QRO secure, then* $\mathcal{S}'$ *is EUF-qCMA-QRO secure.*

Notice that the family $\mathcal{Q}$ of independent functions is only used to specify randomness for input to the signing function, and so if the signature scheme is deterministic we can simplify this construction by omitting $\mathcal{Q}$. This observation is stated explicitly in the following corollary.

Corollary 2.5. Let $\mathcal{S} = (\mathrm{KeyGen}, \mathrm{Sign}, \mathrm{Verify})$ be a deterministic signature scheme. Let H be a random oracle. Define $\mathcal{S}' = (\mathrm{KeyGen}', \mathrm{Sign}', \mathrm{Verify}')$ in the following way:

1. $\mathrm{KeyGen}'(1^\lambda) = \mathrm{KeyGen}(1^\lambda)$.

2. To sign a message m with private key sk, choose $r \in \{0,1\}^k$ at random. The signature is
$$\mathrm{Sign}'_{\mathrm{sk}}(m) = (r, \mathrm{Sign}_{\mathrm{sk}}(H(m,r)))$$

3. To verify a signature (r,σ) by a party with key pair $(\mathrm{sk},\mathrm{pk})$ on a message m, we compute
$$\mathrm{Verify}'_{\mathrm{pk}}(m,(r,\sigma)) = \mathrm{Verify}_{\mathrm{pk}}(H(m,r),\sigma).$$

The signature scheme $\mathcal{S}'$ is correct; moreover, if $\mathcal{S}$ is EUF-CMA-RO secure against an adversary who can perform quantum computations, then $\mathcal{S}'$ is EUF-qCMA-RO secure, and if $\mathcal{S}$ is EUF-CMA-QRO secure, then $\mathcal{S}'$ is EUF-qCMA-QRO secure.

Chapter 3

Security of Authenticated Key Establishment

In this chapter we present general notions of what a security model for authenticated key establishment must do, and given examples of prominent security models. This will serve to motivate the model and constructions in Chapter 4, and highlight the novelty of our new security model.

3.1 Basic Format of a Security Model

We first fix some general terminology which is expounded upon in a security model. A *protocol* is a set of procedures used to establish a secret key among two or more *parties*. A *session* is a specific instance of an execution of a protocol. If a party $\mathcal{P}$ is establishing a key in session Ψ, then the parties with whom it is (or at least believes it is) establishing a key are *peers* to Ψ.

All communication among parties is routed through the *adversary*; that is, if parties wish to communicate, they send their messages to the adversary and indicate the intended recipient. What exactly the adversary is allowed to do with these messages depends on the goals of the model. For the purpose of modelling authenticated key establishment the adversary should be able to modify messages arbitrarily, or even drop messages entirely;

however, it is often useful to consider adversaries which are restricted to only delivering messages which have been sent, and only to their intended recipient. These restricted adversaries are *eavesdroppers* (rather than *active* adversaries) and so such a security model models unauthenticated key establishment—as mentioned in Chapter 2, we can use unauthenticated key establishment protocols as building blocks for authenticated protocols.

A model must also specify what it means for a protocol to be *secure*. In the case of the models presented here, security is defined in terms of a *security game* which it is infeasible for an adversary to win with high probability (typically with non-negligible probability or with non-negligible advantage). The game involves the adversary interacting with a challenger in some way, and then being issued a computational challenge. If the adversary successfully completes the challenge we say that the adversary wins the game.

Then a security model must define:

1. The capabilities of honest parties;
2. The capabilities of the adversary;
3. How the adversary can interact with parties; and,
4. The computational task used to define security;

these are the aspects of the security models that we focus on in this chapter.

3.2 The Bellare-Rogaway Model

Bellare and Rogaway were the first to formalize the notion of a security model for authenticated key establishment [2], following in the footsteps of Goldwasser and Micali [13] who formalized the notion of provable security of encryption schemes. In principle Bellare and Rogaway's model can be used to model protocols that achieve goals other than key establishment, such as mutual authentication. We present the model in full generality, and mention specifically how authenticated key establishment fits into the framework as appropriate.

3.2.1 Protocols

We begin with the formal definition of a general protocol, and then explain how an authenticated key establishment protocol can be modelled in this way.

Definition 3.1 (Protocol)**.** A *protocol* Π is a function which takes as input

1. 1^λ: the security parameter;
2. $i \in I \subseteq \{0,1\}^*$: the identity of the sender;
3. $j \in I \subseteq \{0,1\}^*$: the identity of the intended partner;
4. $a \in \{0,1\}^*$: the secret information of the sender;
5. $\kappa \in \{0,1\}^*$: a record of the previous messages in this invocation of the protocol; and
6. $r \in \{0,1\}^\infty$: the random input for the sender

and which outputs $(m, \delta, \alpha) = \Pi(1^\lambda, i, j, a, \kappa, r)$, where

1. $m \in \{0,1\}^*$ is the message to be sent;
2. $\delta \in \{\varepsilon, 0, 1\}$ is the decision made; and,
3. $\alpha \in \{0,1\}^*$ is the private output.

The intuition for Definition 3.1 is the following. Each party has an associated identification string $i \in I$; the adversary $\mathcal{A}$ is *not* a party and has no such identifier. Parties can have associated private information a—the "long-lived key"—which can be used during key establishment; generally these keys will be established by a key generation algorithm associated to a protocol, before the protocol is to be run. Once the security parameter, identities, and secret keys are fixed, the output of the protocol depends only on the previous messages received in a given session and potentially some random coin tosses.

Given these parameters, the output of a protocol includes an outgoing message (potentially empty) to the party identified by j, a decision as to whether to accept the last received messages, and potentially some private output. For key establishment, the private output is the computed key, and the decisions indicates whether all messages received at this point in the protocol have been accepted and hence whether a key should be constructed.

3.2.2 Adversarial Model

The adversary $\mathcal{A}$ is a classical Turing machine with access to a stream of random bits and oracles $\Pi^s_{i,j}$ which model parties attempting to communicate; in particular, $\Pi^s_{i,j}$ models party $\mathcal{P}_i$ attempting to communicate with party $\mathcal{P}_j$ in session s, for any admissible i, j, s. The adversary activates $\Pi^s_{i,j}(x)$ to obtain the message (m, δ) that $\mathcal{P}_i$ would send to P_j in session s upon receiving message x. We let $\kappa^s_{i,j}$ denote the concatenation of all messages received so far by $\Pi^s_{i,j}$; setting

$$\Pi(1^\lambda, i, j, a, \kappa^s_{i,j}||x, r^s_{i,j}) = (m^s_{i,j}(x), \delta^s_{i,j}(x), \alpha^s_{i,j}(x)),$$

when $\mathcal{A}$ queries $\Pi^s_{i,j}$ on input x, the oracle returns

$$\Pi^s_{i,j}(x) = (m^s_{i,j}(x), \delta^s_{i,j}(x))$$

and we update $\kappa^s_{i,j} = \kappa^s_{i,j}||x$.

For the purposes of authenticated key establishment we give the adversary the ability to reveal a session secret $\alpha^s_{i,j}$ for an oracle $\Pi^s_{i,j}$. If the adversary has issued a reveal query against an oracle that oracle is said to be *open* and is otherwise *unopened.* An oracle is called *fresh* if it is unopened, has accepted, and has not engaged in a matching conversation (defined in section 3.2.3) with an opened oracle.

We say that an adversary is *benign* if its action is restricted to choosing pairs $(\Pi^s_{i,j}, \Pi^t_{j,i})$ of oracles and then faithfully relaying messages between them, in order. The notion of a benign adversary will appear in the security definition for protocols.

3.2.3 Security Definition

The security definition for the Bellare-Rogaway model relies primarily on the notion of a *matching conversation.* Informally, the conversation of an oracle is the ordered concatenation of all messages sent and received by the oracle in the run of a protocol; oracles have matching conversations if the incoming messages of one's conversation are the outgoing messages of the other (in order), and vice versa. Given this definition of matching conversation, we can define security in terms of a security game.

To begin, the adversary is provided with the necessary oracles for the game. The adversary is allowed to interact with the oracles as usual and perform (classical, polynomially-bounded) computations. The adversary must eventually choose a fresh oracle $\Pi^s_{i,j}$ and issue a test query. Upon receiving a test query, $b \in \{0,1\}$ is chosen uniformly at random and, if $b = 0$ the correct secret key $\alpha^s_{i,j}$ is returned while otherwise a random string is chosen according to S_λ, where $\{S_\lambda\}_{\lambda\in\mathbb{N}}$ is the ensemble of distributions of session keys determined by the protocol Π, indexed by the security parameter. After this query the adversary can no longer interact with the oracles and must output a guess b' to b; the adversary wins if $b' = b$. This security game is one component of the following security definition.

Definition 3.2 (Secure Key Exchange Protocol)**.** A key exchange protocol Π with corresponding oracles $\{\Pi^s_{i,j}\}_{\substack{i,j\in I \\ s\in\{0,1\}^*}}$ is *secure* in the Bellare-Rogaway model if the following hold.

1. If $\mathcal{A}$ is a benign adversary who completes a session between $\Pi^s_{i,j}$ and $\Pi^t_{j,i}$, then $\Pi^s_{i,j}$ and $\Pi^t_{j,i}$ both accept; moreover $\alpha^s_{i,j} = \alpha^t_{j,i}$ and this value is distributed over the space of possible keys according to S_λ,

2. If two oracles have matching conversations, then they both accept and output the same secret information α.

3. The probability of an oracle accepting when there is no adversary with a matching conversation is negligibly (in the security parameter) greater than $\frac{1}{2}$.

4. The probability of the adversary correctly guessing the bit in a test query against a fresh oracle is negligible in the security parameter.

Notice that this is an *indistinguishability* definition; that is, the adversary doesn't need to be able to construct a session key, but simply determine whether a revealed string is the correct key for a session or not. This was in line with the philosophy of provable security at the time when the model was proposed, and has perpetuated until the present; each of the security models presented in this chapter will adopt this philosophy, as will the new model we present in Chapter 4.

3.3 The Canetti-Krawczyk Model

The Canetti-Krawczyk model [7] was designed with the philosophy that the most fundamental application of authenticated key establishment is to enable the construction of secret keys for symmetric key encryption schemes for implementing secure communications channels. To that end, the security model and definition used is such that if two parties use a secure protocol to establish a session key and then use that key for a sufficiently secure symmetric key encryption scheme, then the parties can be assured of the authenticity and secrecy of the messages sent and received using that key, in contrast with the Bellare-Rogaway model and related models. Moreover, while [2] focuses on the *shared-key* scenario (when all parties share the same LL-key), in [7] the authors instead primarily consider the public-key setting, where each party has their own private key/public key pair to use for authentication.

3.3.1 Parties, Protocols, and Sessions

A party $\mathcal{P}$ is a polynomial-time machine with an associated secret key/public key pair $(\mathrm{sk}, \mathrm{pk})$; for the purposes of the security experiment there are a polynomial number (chosen by the adversary) of parties that can participate in key establishment. A protocol is a specification of a set of subroutines used to establish keys; in particular these subroutines specify procedures for responding to messages and other requests.

Parties can run arbitrarily many instances of the protocol; each such instance is called a session. Each session is identified by a string s; session identifiers may be repeated, but not twice at the same party. Sessions are initiated when the adversary issues a command $(\mathcal{P}, \mathcal{P}', s, \texttt{role})$ to party $\mathcal{P}$; in this case the peer is $\mathcal{P}'$ and session identifier is s. Parties store session-specific information, such as the party's role (initiator or responder), peer (*i.e.*, the party with whom a key is being established) ephemeral information and messages sent and received in the session. Pairs of sessions initiated by commands $(\mathcal{P}_i, \mathcal{P}_j, s, \texttt{role})$ and $(\mathcal{P}_j, \mathcal{P}_i, s, \texttt{role}')$ are called *matching sessions*.

After enough messages have been sent and received, a session terminates according to the specifications of the protocol. At the time of session termination, either a session key is computed or special output is produced indicating that the session is invalid; in either

case the session's owner records this result and deletes all other memory associated with that session. We also allow for session keys to expire; when the adversary issues such a command to a party, that party deletes the session key associated to the indicated session.

3.3.2 Adversarial Model

As in the Bellare-Rogaway model, the adversary $\mathcal{A}$ is a polynomial-time machine through which all messages between parties are routed. The adversary interacts with parties by delivering messages to them and issuing action requests (such as to begin a session). To model secret information leakage the adversary can also issue `SessionStateReveal`, `SessionKeyReveal` and `Corrupt` queries. As the names suggest, `SessionStateReveal`(s) reveals all state-specific information a session's owner has saved for a given session s, a `SessionKeyReveal`(s) query reveals the session key (if the party has that key in memory) for session s and `Corrupt`($\mathcal{P}$) causes the party $\mathcal{P}$ to become adversarially-controlled and all that party's memory is revealed to the adversary.

3.3.3 Security Definition

A session owned by party $\mathcal{P}_i$ is said to be locally exposed if its state or key have been revealed by the adversary before it is expired, or if $\mathcal{P}_i$ is corrupted before it is expired. A session is exposed if it or its matching session is locally exposed and is otherwise fresh. For the security game, the adversary is allowed a single `Test` query; when this query is issued on a fresh, complete, unexpired session s, the challenger will choose a bit b uniformly at random. If $b = 0$ the true session key will be revealed, while if $b = 1$ and random key will be revealed. At this point, the adversary continues to interact as usual with the parties, except that he *cannot expose the test session.* Eventually the adversary gives a guess b', and wins the security game if $b' = b$. This prompts the security definition:

Definition 3.3 (Secure Key Exchange Protocol)**.** A protocol Π is secure in the Canetti-Krawczyk model if the following hold:

1. If two parties complete matching sessions they output the same session key; and,

2. The advantage any adversary has in guessing the bit used in a fresh, complete, unexpired test session is a negligible function of the security parameter.

3.4 A Model for Quantum Cryptography

Once quantum computers become a reality, it will be possible (in principle, at least) to use them to establish secure session keys. In particular, parties may use quantum computers to compute values they otherwise couldn't in polynomial time; or, more interestingly, parties may exchange qubits as part of a protocol and use these qubits in key establishment. In particular, in 1984, Bennett and Brassard [3] demonstrated a way to use quantum key distribution to establish keys in an information-theoretically secure way in the presence of an eavesdropper. In this section we present a security model due to Mosca, Stebila, and Ustaoğlu [26] which was designed to formally analyze the security properties of that protocol, known as BB84, in an active adversarial model.

The model presented here is an extension of the extended Canetti-Krawczyk model [23], which extends the Canetti-Kracwczyk model by giving the adversary new information-reveal queries which do not necessarily immediately expose an associated session—this means that the adversary is more powerful and can launch new types of attacks. We do not cover these new queries here, because they are not the main technical contribution of the security model in [26].

3.4.1 Parties, Protocols, and Sessions

In this model, a party $\mathcal{P}$ is a *pair* of polynomial-time machines $(C_{\mathcal{P}}, Q_{\mathcal{P}})$, where $C_{\mathcal{P}}$ is classical, with source of random bits, and $Q_{\mathcal{P}}$ is quantum. The classical machine can activate the quantum machine using a special activation request and can likewise receive the results of quantum measurements. Both machines can have messages delivered to them by the adversary. We assume that the link between the machines is noiseless and perfectly secure; that is, the adversary cannot tamper with measurement values being passed from $Q_{\mathcal{P}}$ to $C_{\mathcal{P}}$, and cannot interfere with activation requests sent from $C_{\mathcal{P}}$ to $Q_{\mathcal{P}}$. For authentication parties may have an associated secret key/public key pair.

As usual, a protocol is a collection of subroutines which output a shared secret key among parties. In this model, some subroutines of the protocol may be quantum, and the key is eventually returned by the classical machine. As well, a session is an execution of a protocol at a party. Parties store session-specific information called the session state, including a unique session identifier while the session is active, and when the session completes the and the classical machine outputs either $\perp$ (a value indicating failure) or a tuple $(\kappa, \mathtt{id}, \vec{v}, \vec{u})$ where:

1. κ is the session key;
2. `id` is the party identifier of the peer;
3. $\vec{v} = (\vec{v}_1, \ldots, \vec{v}_t)$ is a vector of vectors of public values which bind the key to the public information used in the session; and,
4. $\vec{u} = (\vec{u}_1, \ldots, \vec{u}_t)$ is a vector of vectors of public values used to authenticate the peer identified by `id` in the session.

Intuitively, we think of $\vec{v}_i$ as the session-specific public values provided by the j^{th} party in the session (the public ephemeral value). The session key is, of course, kept secret. Naturally, a protocol is said to be correct if whenever it is executed according to its specifications and when all messages are relayed faithfully, the parties participating in the protocol output the same key κ.

Aside from session keys, parties may store in memory value pairs of the form (x, X), where x is a private value associated to a public value X; for instance, if parties are participating in Diffie-Hellman key establishment, such a value pair might be (a, g^a). This value pair formalism is used to streamline the definition of special information-reveal queries the adversary can perform.

3.4.2 Adversarial Model and the Security Definition

Like a party, the adversary $\mathcal{A}$ is a pair of machines $(C_{\mathcal{A}}, Q_{\mathcal{A}})$ where $C_{\mathcal{A}}$ is classical and $Q_{\mathcal{A}}$ is quantum. The time and memory of these machines may be bounded; typically we consider polynomially-bounded computation time and storage. The adversary performs

computations and can interact with all the parties; in particular all communication (classical and quantum) is routed through the adversary. The adversary can also establish dishonest parties and corrupt honest parties; parties cannot, *a priori*, distinguish between honest and dishonest parties.

The information-reveal queries of the Bellare-Rogaway and Canetti-Krawczyk models have been streamlined into a single $\texttt{Partner}(X)$ query; when this query is issued to a party, if that party has a value pair (x, X) in memory then the corresponding private value x is returned; in the special case of $\texttt{Partner}(\Psi)$ for a session identifier Ψ, the session key (if it exists) is revealed. If the adversary has issued the query $\texttt{Partner}(X)$ to a party that has in its memory a value pair (x, X) then the adversary is said to be *partner* to X. The adversary is never partner to X until it issues $\texttt{Partner}(X)$.

This leads to the definition of a fresh session.

Definition 3.4 (Fresh Session)**.** A session Ψ owned by an honest party $\mathcal{P}$ is said to be fresh if:

1. The adversary has not revealed the session key for any session with the same public output vector as Ψ (including Ψ itself);
2. For each $\vec{v}_i$ there is some entry to which $\mathcal{A}$ is not partner; and,
3. For each $\vec{u}_i$ there was, at the time of session completion, some entry to which $\mathcal{A}$ was not partner.

Intuitively, this says that for a fresh session, the adversary must not have revealed all secret authentication information for parties participating in the protocol before the session was complete, and must *never* reveal all session-specific information for a party participating in the session.

As in the Canetti-Krawczyk model, the adversary must eventually issue a `Test` query, specifying a session. If the session does not yet have a session key, the returned value is $\perp$ and the adversary loses. Otherwise the challenger chooses $b \in \{0, 1\}$ uniformly at random and, if $b = 0$, reveals the session key and if $b = 1$ reveals a randomly-chosen key. The adversary can continue to interact with the parties in any way *provided that the test*

session remains fresh, and eventually returns a bit b'. The adversary wins the game if $b' = b$. A protocol is secure if no adversary satisfying specified constraints on computation time and memory can win this game with probability non-negligibly greater than $\frac{1}{2}$.

Chapter 4

A Security Model for Post-Quantum Authenticated Key Establishment

In this chapter we discuss our new security model for authenticated key establishment. The primary difference between this model and the ones presented in Chapter 3 is that the security definition for this model allows quantum interactions between the adversary and quantum emulators for strictly classical parties, in much the same way that the security game for EUF-qCMA security of signature schemes allows for quantum queries to a signing oracle in order to create a stronger security definition. We then present a generic construction of a secure protocol using a signature scheme and a key establishment protocol which is secure in a restricted model with a passive adversary.

4.1 Motivation

Whenever a security model is proposed, it is important to motivate the specific choices made by the model; in particular, we would like to justify choices by considering their implications in the context of real-world communications security and in the broader context of theoretical cryptography. Most aspects of this model—parties, protocols, and the concept of sessions, for instance—are standard and need no further justification; the most fundamental change is allowing quantum queries to ordinarily classical procedures. In this

section we make a case for this decision.

Perhaps the most natural justification of this decision is that protocols which are secure in this model when run on classical machines remain secure when run on a quantum computer instead. The use of this is clear: a quantum cryptographic routine could, in principle, be simplified by having a subroutine that requires secure classical communications; for instance, in order to establish which bits are agreed-upon in the BB84 protocol [3], parties must communicate classically. If a classical authenticated key establishment protocol is known to be secure even when the adversary is allowed to pass quantum superpositions of messages between parties, then parties to the larger cryptographic routine can simply simulate classical computers running the protocol and still be assured of the desired security properties.

In the context of theoretical cryptography, it is sometimes useful to have security definitions which are stronger than might be strictly necessary, so that protocols can be used as parts of more complicated protocols with different security definitions, while still allowing the security proofs to go smoothly. In this case, if a cryptographic protocol requires an authenticated key establishment protocol as a subroutine, and its security definition requires that the protocol must remain secure even when, for instance, there is an active quantum adversary, then a key establishment protocol which is only known to be secure in the Canetti-Krawczyk model with a quantum adversary may not be sufficient for the purposes of the security proof. For this reason our new security model may be a useful theoretical tool for constructing generic subroutines for more complicated cryptographic protocols.

4.2 Definitions

4.2.1 Parties, Protocols, and Sessions

Definition 4.1 (Party)**.** A party $\mathcal{P}$ is an interactive classical Turing machine with access to a source of a random bits.

Associated to each party $\mathcal{P}$ is a (possibly empty) private key/public key pair $(\mathrm{sk}, \mathrm{pk})$. For the purposes of the model, it is assumed that each party has a genuine copy of each

other party's public key—this is what will allow authentication. Moreover, to each party is associated a unique identifier $\mathtt{id}$—it is assumed that each party has a genuine copy of each other party's identifier.

Definition 4.2 (Protocol)**.** A protocol Π is a specification of a set of subroutines, to be run by some number of parties, for the purpose of establishing a session key.

A protocol Π is said to be *correct* if, when Π is executed according to its specifications and when all messages are relayed faithfully (*i.e.*, without changes to their content or ordering), all parties involved compute the same key.

As in the security models presented in Chapter 3, protocols are *message-driven*; that is, upon receiving a message, a party computes the response message and sends it to the intended recipient. The party does no further computations and sends no more messages until it is activated again by an incoming message.

Definition 4.3 (Session)**.** A session Ψ is an instantiation of a protocol at a given party.

Associated to each session is a unique[6] session identifier Ψ, chosen by the session's owner. If a party $\mathcal{P}$ owns a session Ψ, the parties with whom $\mathcal{P}$ believes they are attempting to establish a key are called *peers* to $\mathcal{P}$ in session Ψ, and the peers' associated sessions (if they exist) are called *matching sessions.*

If a party $\mathcal{P}$ with identifier $\mathtt{id}$ who owns a given session Ψ, with matching session Ψ' and peer $\mathcal{P}'$ with identifier $\mathtt{id}'$ has received messages $m_1, \ldots, m_{k-1}$ in this session, then we denote by

$$\mathcal{P}(\Psi, \Psi', \mathtt{id}, \mathtt{id}'; \mathrm{pk}, \mathrm{pk}', \mathrm{sk}; m_1, \ldots, m_k; r_\Psi)$$

the message that $\mathcal{P}$ sends given that the next message it receives in this session is m_k, and it uses randomness r_Ψ for this session. For brevity, we abbreviate this expression as $\mathcal{P}(m_k)$ if the other inputs are clear from context.

[6]Since parties need not consult one another, there is, in principle, the possibility that two parties may choose the same session identifier for some sessions. This possibility can be eliminated by, for instance, requiring that the session identifier contain the party's identity.

4.2.2 Invalid Messages

In a classical key establishment security model, it is typical to have a mechanism by which a party can prematurely end a session in the event that it receives an "invalid" message; what exactly constitutes an invalid message is defined by a given protocol, but typically an invalid message is one which either does not make sense in the context of the protocol or which fails to validate under the public key of the party believed to have sent the message. If we are to allow the adversary to deliver quantum superpositions of messages, however, it does not make sense to consider such termination for two reasons:

1. The party who receives the superposition of messages cannot simply read off the delivered messages, and thus cannot easily tell whether or not to terminate the session. Measuring the state would collapse it and defeat the purpose of considering quantum queries entirely.

2. Even if the party could read off the messages in the superposition, it is possible that some messages in the superposition are valid while some are invalid. It is not clear what should be done in this scenario.[7]

For this reason we introduce a special failure character $\perp$ to be used whenever a session would be terminated. We define $\mathcal{P}(\Psi, \Psi', \mathtt{id}, \mathtt{id}'; \mathrm{pk}, \mathrm{pk}', \mathrm{sk}; m_1, \ldots, m_k; r_\Psi) = \perp$ whenever m_k is an invalid message, and we further define all further messages in a session after a response has been $\perp$ to be $\perp$. This formalism essentially allows a session to be in a superposition of terminated and active.

4.3 Party and Adversarial Capabilities

Aside from classical computations, parties can issue a special $\mathtt{Send}(\mathtt{id}, m)$ command; this requests that the adversary deliver message m to the party identified by the identifier $\mathtt{id}$. Parties may store private/public value pairs (k, K) in memory, associated to sessions. In

[7] One might argue that a superposition of messages should be declared invalid if there is any invalid message in the superposition with non-zero amplitude. This idea may appear natural, but it is not easily implemented (see point 1.) and results in a weaker security definition than we propose.

particular, for a key establishment session Ψ a party may draw an ephemeral secret key sk_Ψ and derive a corresponding ephemeral public value pk_Ψ.

In order to make protocols meaningfully quantum-resistant, for the purposes of the security experiment the challenger will provide a *quantum messaging oracle* $O_\mathcal{P}$ for each party $\mathcal{P}$, defined inductively as follows. Before $O_\mathcal{P}$ receives any messages in session Ψ with matching session Ψ' and peer $\mathcal{P}'$ with identifier $\mathtt{id}'$, we define

$$O_\mathcal{P}(\Psi)\left|m_1\right\rangle\left|y\right\rangle = \left|m_1\right\rangle\left|y \oplus \mathcal{P}(\Psi, \Psi', \mathtt{id}, \mathtt{id}'; \mathrm{pk}, \mathrm{pk}', \mathrm{sk}; m_1; r_\Psi)\right\rangle$$

for all $m_1 \in \mathcal{M}$; notice that the session Ψ considered by $O_\mathcal{P}$ must be given as *classical* input. $O_\mathcal{P}$ then *holds onto* the first register, and returns the second. After receiving $k-1$ messages, $O_\mathcal{P}$ will be holding onto $k-1$ registers; when it is queried again, we can consider its input as the first $k+1$ registers of some global state

$$\left|\Gamma\right\rangle = \sum \alpha_{m_1,\ldots,m_{k-1},m_k,y} \underbrace{\left|m_1\right\rangle \cdots \left|m_{k-1}\right\rangle}_{\text{Held by Challenger}} \overbrace{\left|m_k\right\rangle\left|y\right\rangle}^{\text{Provided by } \mathcal{A}} \underbrace{\left|\mu_{m_1,\ldots,m_{k-1},m_k,y}\right\rangle}_{\text{Remaining registers}}.$$

Then its action is defined by

$$O_\mathcal{P}(\Psi)\left|m_1\right\rangle \cdots \left|m_{k-1}\right\rangle\left|m_k\right\rangle\left|y\right\rangle = \left|m_1\right\rangle \cdots \left|m_{k-1}\right\rangle\left|m_k\right\rangle\left|y \oplus \mathcal{P}(m_k)\right\rangle.$$

If $\mathcal{P}$ does not own a given session $\hat{\Psi}$, we simply define $O_\mathcal{P}(\hat{\Psi})\left|m\right\rangle\left|y\right\rangle = \left|m\right\rangle\left|y \oplus \bot\right\rangle$. For simplicity, for the purposes of the security experiment, the adversary interacts only with these quantum messaging oracles.

In addition to standard quantum computations, the adversary interacts with the quantum messaging oracles by delivering (quantum superpositions of) messages to them. The adversary may also issue the following:

1. $\mathtt{RevealEphemeralKey}(\mathtt{id}, \Psi)$: If the party identified by $\mathtt{id}$ owns a session Ψ, the challenger reveals any ephemeral secret key[8] associated to the session and party.

[8] In this context, "ephemeral secret key" refers to session-specific information derived from the random input; in particular, it does *not* depend on incoming messages or any other quantities. In this way, the result of this query is strictly classical. This can be done without loss of generality.

2. `RevealPrivateKey(id)`: Requests the party identified by `id` to provide their private key.

3. `Corrupt(id)`: When this command is issued, the party identified by the identifier `id` becomes adversarially-controlled; the adversary learns all classical information known by the party, is given all quantum memory associated to the party, and chooses all future actions it performs.

As well, the adversary may issue the following quantum query: `RevealSessionKey(id, Ψ)`, defined in the following way. If $K(\Psi, \Psi', \mathtt{id}, \mathtt{id}'; \mathrm{pk}, \mathrm{pk}', \mathrm{sk}; m_1, \ldots, m_k; r_\Psi)$ is the key that the party identified by `id` would compute in session Ψ with peer identifier $\mathtt{id}'$ and matching session Ψ', and it has so far received messages $m_1, \ldots, m_k$, then if the global state is

$$|\Gamma\rangle = \sum \alpha_{m_1,\ldots,m_k,y} \underbrace{|m_1\rangle \cdots |m_k\rangle}_{\text{Held by Challenger}} \overbrace{|y\rangle}^{\text{Provided by } \mathcal{A}} \underbrace{|\mu_{m_1,\ldots,m_k,y}\rangle}_{\text{Remaining registers}} ,$$

the result of this query is defined by

$$|m_1\rangle \cdots |m_k\rangle |y\rangle \mapsto |m_1\rangle \cdots |m_k\rangle |y \oplus K(\Psi, \Psi', \mathtt{id}, \mathtt{id}'; \mathrm{pk}, \mathrm{pk}', \mathrm{sk}; m_1, \ldots, m_k; r_\Psi)\rangle .$$

As a result of this query $O_{\mathcal{P}}$ returns the first $k+1$ registers of the global state (that is, it returns the target register provided by the adversary, and the received message registers it was holding).

4.4 The Security Experiment

A session Ψ owned by party $\mathcal{P}$ with peer $\mathcal{P}'$ and partner session Ψ' is said to be "clean" if all of the following are true:

1. At the time of session completion, neither $\mathcal{P}$ nor $\mathcal{P}'$ was adversarially-controlled.

2. At the time of session completion, $\mathcal{A}$ had not issued `RequestPrivateKey(id)` nor `RequestPrivateKey(id′)`.

3. $\mathcal{A}$ has not revealed the ephemeral secret key for Ψ or Ψ'.

4. $\mathcal{A}$ has not issued $\texttt{RevealSessionKey}(\texttt{id}, \Psi)$ nor $\texttt{RevealSessionKey}(\texttt{id}', \Psi')$.

The security experiment is the following: the adversary issues a $\texttt{Test}(\texttt{id}, \Psi)$ query on a clean session Ψ owned by the party with identifier $\texttt{id}$, defined in the following way. The adversary provides a target register $|y\rangle$, and the challenger selects $b \in \{0, 1\}$ uniformly at random. If $b = 1$, $\texttt{Test}(\texttt{id}, \Psi)$ acts like a $\texttt{RevealSessionKey}$ query; if $b = 0$, the result is defined by

$$|m_1\rangle \cdots |m_k\rangle |y\rangle \mapsto |m_1\rangle \cdots |m_k\rangle |y \oplus R(m_1, \ldots m_k)\rangle$$

where for each tuple of messages, $R(m_1, m_2, \ldots, m_k)$ is a random string in $\mathcal{K}$ with the stipulation that if $K(\Psi, \Psi', \texttt{id}, \texttt{id}'; \mathrm{pk}, \mathrm{pk}', \mathrm{sk}; m_1, \ldots, m_k; r_\Psi) = \perp$, then $R(m_1, \ldots, m_k) = \perp$. In any case, all $k + 1$ of these registers is returned. We say that the key establishment protocol Π is *secure* if for any polynomial time adversary $\mathcal{A}$, the probability that $\mathcal{A}$ can correctly guess the value of b is at most negligibly greater than $\frac{1}{2}$.

4.5 Generic Constructions for Secure Protocols using Signature Schemes

In this section we discuss how to use a signature scheme to provide authentication to an unauthenticated key exchange protocol. The idea is simple: if each party simply signs every message that they send along with their identifier, the identifier of the intended recipient, the session identifier and peer session identifier (if known), then any protocol which is secure when messages are delivered faithfully becomes a protocol which is secure even with an active adversary. More precisely, we prove the following theorem.

Theorem 4.1. *Let* $\mathcal{S} = (\mathrm{KeyGen}, \mathrm{Sign}, \mathrm{Verify})$ *be a strongly EUF-qCMA secure signature scheme, and let* Π *be a two-round key establishment protocol which is secure when the adversary* $\mathcal{A}$ *in the security game is required to deliver all messages faithfully. Consider the protocol* Π' *with the following properties:*

1. *Each party* $\mathcal{P}'_k$ *has a key pair* $(\mathrm{sk}_k, \mathrm{pk}_k)$ *for* $\mathcal{S}$ *and, moreover, each party knows each other party's public key.*

2. *Whenever an initiating party $\mathcal{P}'_I$ would send a message m, it instead sends the triple $(m, \Psi^{(I)}, \sigma)$, where*

$$\sigma = \mathrm{Sign}_{\mathrm{sk}_I}(m, \mathtt{id}_I, \mathtt{id}_R, \Psi^{(I)})$$

where $\mathtt{id}_R$ is the identifier of the intended recipient, and $\Psi^{(I)}$ is the session in which the message is being sent.

3. *Whenever a responding party $\mathcal{P}'_R$ would respond to a message $(m, \Psi^{(I)}, \sigma)$ from an initiating party $\mathcal{P}'_I$ with a message m', it computes $b = \mathrm{Verify}_{\mathrm{pk}_I}(m, \mathtt{id}_I, \mathtt{id}_R, \Psi^{(I)}, \sigma)$. If $b = 0$, P'_R responds with $(\perp, \perp, \perp)$; otherwise, $\mathcal{P}'_R$ computes*

$$\sigma' = \mathrm{Sign}_{\mathrm{sk}_R}(m', \mathtt{id}_I, \mathtt{id}_R, \Psi^{(I)}, \Psi^{(R)})$$

and responds with $(m', \Psi^{(R)}, \sigma')$.

4. *Whenever a party $\mathcal{P}_k$ would compute a session key for a session $\Psi^{(k)}$ with partner $\mathcal{P}_\ell$ and partner session $\Psi^{(\ell)}$, it determines whether the signature in the message it received was valid; if not it outputs session key $\perp$. If the signature is valid, it outputs the session key as usual.*

   The protocol Π' is secure.

To prove Theorem 4.1, we will first show that no adversary can construct messages for which the probability amplitude of correctly-signed messages that were not sent in a clean session is non-negligible by using such an adversary as a signature forger in the strongly EUF-qCMA game. Then, given that the probability amplitude of a valid unsent message is negligible, we show that any adversary that distinguishes a superposition of session keys from a superposition of random strings breaks the security of the underlying unauthenticated key establishment protocol, contradicting the assumptions of the theorem, and thus establishing the security of this new protocol.

For the first part of the argument, we must first show how, given an instance (pk) of the strongly EUF-qCMA game, we can emulate a quantum messaging oracle $O_{\mathcal{P}_k}$ for a party $\mathcal{P}_k$ with public key pk for $\mathcal{S}$.

For an unauthenticated key exchange protocol Π, let the parties be denoted by $\mathcal{P}_k$ for some values of k, and for each such party let $\mathcal{P}'_k$ denote the corresponding party for protocol

Π' defined as in Theorem 4.1. Notice that

$$\mathcal{P}'_k(m,\sigma) = \begin{cases} (\mathcal{P}_k(m), \mathrm{Sign}_{\mathrm{sk}_{\mathcal{P}_k}}(\mathcal{P}_k(m)) \text{ if } \mathrm{Verify}_{\mathrm{pk}_{\mathcal{P}'}}(m,\sigma) = 1 \\ (\bot,\bot) \text{ otherwise} \end{cases} .$$

Then, to emulate the quantum messaging oracle, first write $\mathcal{P}(m)$ to an auxiliary register to obtain

$$\sum_{m,\sigma,y} \alpha_{m,\sigma} |m\rangle |\sigma\rangle |\mathcal{P}(m)\rangle |y\rangle .$$

Use the strongly EUF-qCMA signing oracle on the third register to obtain

$$\sum_{m,\sigma,y} \alpha_{m,\sigma} |m\rangle |\sigma\rangle |\mathcal{P}(m)\rangle |\mathrm{Sign}_{\mathrm{sk}}(\mathcal{P}(m))\rangle |y\rangle .$$

Then apply U_{Verify}, defined by

$$U_{\mathrm{Verify}} \colon |m\rangle |\sigma\rangle |m'\rangle |\sigma'\rangle |y\rangle |z\rangle \mapsto \begin{cases} |m\rangle |\sigma\rangle |m'\rangle |\sigma'\rangle |y \oplus m'\rangle |z \oplus \sigma'\rangle & \text{if } \mathrm{Verify}(m,\sigma) = 1 \\ |m\rangle |\sigma\rangle |m'\rangle |\sigma'\rangle |y \oplus \bot\rangle |z \oplus \bot\rangle & \text{otherwise} \end{cases}$$

to obtain

$$\sum_{m,\sigma,y} \alpha_{m,\sigma} |m\rangle |\sigma\rangle |\mathcal{P}(m)\rangle |\mathrm{Sign}_{\mathrm{sk}}(\mathcal{P}(m))\rangle |y \oplus \mathcal{P}(m,\sigma)\rangle ;$$

the last register is the one we give to $\mathcal{A}$. Note in particular that we are holding onto the registers that contain valid message/signature pairs; in fact, we hold one such pair of registers for each query we make to the signing oracle. It follows that if we can persuade $\mathcal{A}$ to send us a pair of registers which, when measured, yield a valid message-signature pair different from those that we will obtain by measuring the registers we already hold, then with non-negligible probability we can win the strongly EUF-qCMA game.

Knowing that we can use the quantum signing oracle for the strongly EUF-qCMA game to emulate a quantum messaging oracle for a party, we show that the security of the signature scheme restricts the class of messages that an adversary can construct. The results are presented in the following technical lemmas.

We first demonstrate two restrictions that we can place on the behaviour of $\mathcal{A}$ without loss of generality.

Lemma 4.2. Suppose there is an adversary $\mathcal{A}$ who wins the security game for Π' with advantage Adv who delivers two or more messages to a single party in a given session. Then there is an adversary $\mathcal{A}'$ who wins the security game for Π' with the same advantage who never delivers two or more messages to a single party in a given session.

Proof. The response to any message delivered to a party in a given session beyond the first is $(\perp, \perp)$; this is because Π is a two-round key establishment protocol and so any message delivered beyond the first is invalid. Let $\mathcal{A}'$ be defined as $\mathcal{A}$ is, except that whenever $\mathcal{A}'$ would deliver the second message to a party in a given session, it instead simply writes $(\perp, \perp)$ to its target register. It is clear that $\mathcal{A}'$ wins the security game with the same probability as $\mathcal{A}$. □

Lemma 4.3. Suppose there is an adversary $\mathcal{A}$ who wins the security game for Π' with advantage Adv, and who at some point sends the last register of the global state

$$|\Gamma\rangle = \sum_{\vec{\mu},m,\sigma} \alpha_{\vec{\mu},m,\sigma} |\vec{\mu}\rangle |m,\sigma\rangle$$

to a responding party $\mathcal{P}_R$ in session Ψ', who believes they are participating in a session Ψ with initiating party $\mathcal{P}_I$, such that

$$\sum_{\mathrm{Verify}_{\mathrm{sk}_I}(m,\sigma)=1} \sum_{\vec{\mu}} |\alpha_{\vec{\mu},m,\sigma}|^2$$

is negligible. Then there is an adversary $\mathcal{A}'$ who wins the security game with advantage Adv' which differs only negligibly from Adv, such that $\mathcal{A}'$ never sends a register in such a state.

Proof. Let $\mathcal{A}$ be such an adversary. First we show that Ψ' cannot possibly be the session on which $\mathcal{A}$ will choose to be tested. Suppose to the contrary that Ψ' is the test session. The global state after the `Test` query will be

$$|\Gamma\rangle = \sum_{\mathrm{Verify}_{\mathrm{sk}_I}(m,\sigma)=0} \sum_{\vec{\mu}} \alpha_{\vec{\mu},m,\sigma} |\vec{\mu}\rangle |m,\sigma\rangle |\perp\rangle + \sum_{\mathrm{Verify}_{\mathrm{sk}_I}(m,\sigma)=1} \sum_{\vec{\mu}} \alpha_{\vec{\mu},m,\sigma} |\vec{\mu}\rangle |m,\sigma\rangle |\kappa_b(m)\rangle$$

where $\kappa_b(m)$ is either a correct key for session Ψ' on incoming message m, or a random string; in particular, in either case is it not $\perp$.

Consider the state

$$|\Gamma'\rangle = \sum_{\mathrm{Verify}_{\mathrm{sk}_I}(m,\sigma)=0} \sum_{\vec{\mu}} \alpha_{\vec{\mu},m,\sigma} |\vec{\mu}\rangle |m,\sigma\rangle |\perp\rangle + \sum_{\mathrm{Verify}_{\mathrm{sk}_I}(m,\sigma)=1} \sum_{\vec{\mu}} \alpha_{\vec{\mu},m,\sigma} |\vec{\mu}\rangle |m,\sigma\rangle |r(m)\rangle$$

for randomly chosen strings $r(m)$. In particular, observe that the ensembles $\mathcal{D} = \{D_{\lambda,r}\}$ and $\mathcal{D}' = \{D_{\lambda,r}\}$ of measurement outcomes of $|\Gamma\rangle$ and $|\Gamma'\rangle$ (parameterized by the security parameter and random input) are computationally indistinguishable by the previous lemma, since if they were not, we could distinguish $|\Gamma\rangle$ from $|\Gamma'\rangle$ with non-negligible advantage. In particular, in this case this means that if $\mathcal{A}$ were instead given $|\Gamma'\rangle$, and then performed his measurement in order to guess the value b, the result would, except with negligible probability, be indistinguishable from the result of measuring $|\Gamma\rangle$, *regardless of the value of* b. Since $|\Gamma'\rangle$ carries no information about b, $\mathcal{A}$ can't possibly guess b by measuring $|\Gamma'\rangle$ with probability different from $\frac{1}{2}$. Thus when measuring $|\Gamma\rangle$ and guessing, $\mathcal{A}$ guesses correctly with probability at most negligibly greater than $\frac{1}{2}$, contradicting our assumption. Hence Ψ' cannot be the test session.

By a similar argument, $\mathcal{A}$ can construct a register that is indistinguishable from the response $\mathcal{P}_R$ would give on this input register. Hence $\mathcal{A}'$ proceeds exactly as $\mathcal{A}$ would, except that whenever he would send a register as described in the statement of the lemma, he instead constructs an indistinguishable register.

Since $\mathcal{A}$ deals with at most polynomially-many registers, we can make as many substitutions of this kind as required and the probability that the resultant state is distinguishable from the correct state is negligible; hence, $\mathcal{A}'$, defined in this way, wins the security game with advantage at most negligibly different from Adv, as required.

□

Hence we can assume without loss of generality that our adversary $\mathcal{A}$ never delivers more than one message in a session and never delivers a superposition of messages for which the total probability amplitude of the valid content is negligible. This will allow us to use an adversary $\mathcal{A}$ who delivers a superposition of messages in a session for which

the amplitude of a valid, but unsent, message is non-negligible as a forger for a signature scheme; this tells us then that the probability of the adversary delivering such a message is negligible.

Lemma 4.4. Let $\mathcal{A}$ be an adversary who wins the security game with non-negligible advantage. Let

$$|\Gamma\rangle = \sum_{\vec{\mu},m_I,\sigma_I} \alpha_{\vec{\mu},m_I,\sigma_I} |\vec{\mu}\rangle |m_I,\sigma_I\rangle$$

be the global state after the last register is delivered by $\mathcal{A}$ to $\mathcal{P}_R$ in a clean session. Further, let (m^*,σ^*) be the message and signature that $\mathcal{P}_I$ would send in this session. Let

$$\mathscr{F} = \{(m_I,\sigma_I)\colon \mathrm{Verify}_{\mathrm{pk}_I}(m_I,\sigma_I) = 1 \text{ and } (m_I,\sigma_I) \neq (m_I^*,\sigma_I^*)\}$$

be the set of potential "forgeries." If $(\mathrm{KeyGen},\mathrm{Sign},\mathrm{Verify})$ is strongly EUF-qCMA, then except with negligible probability, the quantity Φ, defined by

$$\Phi \equiv \sum_{\vec{\mu},(m_I,\sigma_I)\in\mathscr{F}} |\alpha_{\vec{\mu},m_I,\sigma_I}|^2$$

is negligible.

Proof. We show how to forge a signature against $(\mathrm{KeyGen},\mathrm{Sign},\mathrm{Verify})$ in the strongly EUF-qCMA game if Φ is non-negligible.

Suppose we are given an instance (pk) of the EUF-qCMA game. We will run $\mathcal{A}$ essentially as normal, by establishing public key/private key pairs for as many parties as $\mathcal{A}$ requires; for one party $\mathcal{P}_{i^*}$ chosen at random, however, we will set their private key as pk (and the underlying secret key will remain unknown to us). With probability at least $\frac{1}{p(\lambda)}$, where $p(\lambda)$ is a bound on the number of parties $\mathcal{A}$ requires (and which is at most polynomial in λ), we have selected the initiator of this clean session. In particular this means that, at least until the session is over, $\mathcal{A}$ will not issue `RequestPrivateKey`($\mathtt{id}_{i^*}$), and so we will not have to produce it. Whenever a party needs to sign a message we use their private key, unless that party is $\mathcal{P}_{i^*}$, in which case we simply use the signing oracle from the strongly EUF-qCMA game, as described above. Notice that by our assumption that $\mathcal{A}$ never delivers two or more messages to a party in the same session, each time we

query the signing oracle we are querying it for a different session; since the session identifier is included in the signed message, this means that, in particular, if we measure the results of our queries to the signing oracle we will never obtain the same message/signature pairs. Moreover, because of the construction we use to model the party from the signing oracle, each use of the signing oracle results in a register which will, with probability 1, yield a valid message/signature pair upon measurement. In particular, this means that if $\mathcal{A}$ ever sends us a superposition of messages for which the probability amplitude of a forged message is non-negligible, then by measuring that register and the registers we hold, we will obtain $q+1$ distinct valid message/signature pairs, where q is the number of calls we have made to the signing oracle.

$\mathcal{A}$ will perform some unitary operations on the qubits he holds; thus the global state becomes

$$|\Gamma\rangle = \sum_{\vec{\mu},m_I,\sigma_I} \alpha_{\vec{\mu},m_I,\sigma_I} |\vec{\mu}\rangle |m^*,\sigma^*\rangle |m_I,\sigma_I\rangle$$

and $\mathcal{A}$ sends the last register to $\mathcal{P}_R$ (*i.e.,* to us). If we now measure the qubits we hold, then with probability Φ, we will obtain $q+1$ valid message/signature pairs. If Φ is non-negligible, we can win the strongly EUF-qCMA game for our signature scheme; since the signature scheme is strongly EUF-qCMA, this forgery can occur with at most negligible probability, and so the probability that $\frac{\Phi}{p(\lambda)}$ is non-negligible (and hence that Φ is non-negligible) is negligible, as required. □

Lemma 4.5. Let

$$|\Gamma\rangle = \sum_{\vec{\mu},m_I,\sigma_I,m_R,\sigma_R} \alpha_{\vec{\mu},m_I,\sigma_I,m_R,\sigma_R} |\vec{\mu}\rangle |m_I,\sigma_I\rangle |m_R,\sigma_R\rangle$$

be the global state after the completion of a clean session Ψ owned by $\mathcal{P}_R$, the responding party, where the second register is the message register sent by $\mathcal{A}$ to $\mathcal{P}_R$, and the third is the message register sent by $\mathcal{A}$ to $\mathcal{P}_I$, the initiating party, if it exists. Let (m_I^*,σ_I^*) be the message and signature that would actually be sent by $\mathcal{P}_I$ in step 3e of the protocol, and let (m_R^*,σ_R^*) be the message and signature that $\mathcal{P}_R$ would respond with if the messages were relayed faithfully. Let

$$\mathscr{F} = \{(m_I,\sigma_I,m_R,\sigma_R)\colon \mathrm{Verify}_{\mathrm{pk}_\iota}(m_\iota,\sigma_\iota) = 1 \text{ and } (m_\iota,\sigma_\iota) \neq (m_\iota^*,\sigma_\iota^*) \text{ for some } \iota \in \{I,R\}\}$$

be the set of potential tuples containing a "forged" signature. If $(\mathrm{KeyGen}, \mathrm{Sign}, \mathrm{Verify})$ is strongly EUF-qCMA, then, except with negligible probability, the quantity Φ, defined by

$$\Phi \equiv \sum_{(m_I,\sigma_I,m_R,\sigma_R)\in\mathscr{F}} \sum_{\vec{\mu}} |\alpha_{\vec{\mu},m_I,\sigma_I,m_R,\sigma_R}|^2$$

is negligible.

Proof. Define

$$\begin{aligned}\mathscr{F}_I &= \{(m_I,\sigma_I,m_R,\sigma_R)\in\mathscr{F} \colon \mathrm{Verify}_{\mathrm{pk}_I}(m_I,\sigma_I)=1 \text{ and } (m_I,\sigma_I)\neq(m_I^*,\sigma_I^*)\}, \text{ and}\\ \mathscr{F}_{\neg I} &= \{(m_I,\sigma_I,m_R,\sigma_R)\in\mathscr{F} \colon \mathrm{Verify}_{\mathrm{pk}_I}(m_I,\sigma_I)=0 \text{ or } (m_I,\sigma_I)=(m_I^*,\sigma_I^*)\}\end{aligned}$$

and observe that $\mathscr{F} = \mathscr{F}_I \sqcup \mathscr{F}_{\neg I}$. By Lemma 4.4, we know that

$$\Phi_I = \sum_{(m_I,\sigma_I,m_R,\sigma_R)\in\mathscr{F}_I} \sum_{\vec{\mu}} |\alpha_{\vec{\mu},m_I,\sigma_I,m_R,\sigma_R}|^2$$

is negligible. Then we need only prove that

$$\Phi_{\neg I} = \sum_{(m_I,\sigma_I m_R,\sigma_R)\in\mathscr{F}_{\neg I}} \sum_{\vec{\mu}} |\alpha_{\vec{\mu},m_R,\sigma_R,m_I,\sigma_I}|^2$$

is negligible, since $\Phi = \Phi_I + \Phi_{\neg I}$. As in the proof of lemma 4.4, we can try to win the EUF-qCMA game by choosing a random party $\mathcal{P}$ and hoping that $\mathcal{A}$ "forges" a signature against them; then we measure the registers we hold to obtain a forgery. This succeeds with probability at least $\Phi_{\neg I}$, and so this quantity must be negligible except with negligible probability, as required. □

Lemma 4.6. Let

$$\begin{aligned}|\Gamma\rangle &= \sum_{m\in M} \alpha_m |m\rangle + \sum_{c\in C} \alpha_c |c\rangle \text{ and}\\ |\Gamma'\rangle &= \sum_{m\in M} \alpha_m |m\rangle + \sum_{c\in C} \alpha_c |r(c)\rangle\end{aligned}$$

be normalized quantum states, where M and C are disjoint, nonempty, finite sets, $r(c)\notin M$

for all $c \in C$, and $\sum_{c \in C} |\alpha_c|^2$ is negligible. Then the advantage that any adversary has in distinguishing $|\Gamma\rangle$ from $|\Gamma'\rangle$ is negligible.

Proof. Note that $1 - |\langle\Gamma|\,\Gamma'\rangle|^2 \leq 2\sum_{c \in C} |\alpha_c|^2$. The result then follows from the Holevo-Helstrom theorem. □

Lemma 4.7. Consider the following state distinguishing game for some fixed quantum states states $|\Gamma_0\rangle$, $|\Gamma_1\rangle$, $|\Gamma'_0\rangle$ and $|\Gamma'_1\rangle$:

i. $\mathcal{C}$ selects $b \in \{0, 1\}$ uniformly at random, and sends $|\Gamma_b\rangle$ to $\mathcal{A}$.

ii. $\mathcal{A}$ performs some computations and outputs a guess b'.

iii. $\mathcal{A}$ wins if $b' = b$.

Let $\mathcal{A}$ be an adversary for this game, and now consider the following game:

i. $\mathcal{C}$ selects $b \in \{0, 1\}$ uniformly at random, and sends $|\psi_b\rangle$ to $\mathcal{A}$.

ii. $\mathcal{A}$ performs some computations and outputs a guess b'.

iii. $\mathcal{A}$ wins if $b' = b$.

The probability that $\mathcal{A}$ wins the second game differs from the probability that $\mathcal{A}$ wins the first game by at most

$$\frac{1}{2}\left(\sqrt{1 - |\langle\Gamma_0|\,\Gamma'_0\rangle|^2} + \sqrt{1 - |\langle\Gamma_1|\,\Gamma'_1\rangle|^2}\right)$$

Proof. Consider the problem of distinguishing $|\Gamma_0\rangle$ from $|\Gamma'_0\rangle$. By Theorem 1.2 the advantage that any procedure has in distinguishing these two states is at most $\frac{1}{2}\sqrt{1 - |\langle\Gamma_0|\,\Gamma'_0\rangle|^2}$.

Consider the following distinguishing procedure: given a state $|\Gamma''_0\rangle$ which is either in the state $|\Gamma_0\rangle$ or $|\Gamma'_0\rangle$, each with probability $\frac{1}{2}$, give the state to $\mathcal{A}$. If $\mathcal{A}$ produces the guess

$b' = 0$, guess that the state is $|\Gamma_0\rangle$, and otherwise guess that the state is $|\Gamma'_0\rangle$. Then

$$\begin{aligned} P[\text{This Procedure is Correct}] &= P[b' = 0 \wedge |\Gamma''_0\rangle = |\Gamma_0\rangle] + P[b' = 1 \wedge |\Gamma''_0\rangle = |\Gamma'_0\rangle] \\ &= P[b' = 0 | \, |\Gamma''_0\rangle = |\Gamma_0\rangle] \cdot P[|\Gamma''_0\rangle = |\Gamma_0\rangle] \\ &\quad + P[b' = 1 | \, |\Gamma''_0\rangle = |\Gamma'_0\rangle] \cdot P[|\Gamma''_0\rangle = |\Gamma'_0\rangle] \\ &= \frac{1}{2} P[b' = 0 | \, |\Gamma''_0\rangle = |\Gamma_0\rangle] \\ &\quad + \frac{1}{2} \left(1 - P[b' = 0 | \, |\Gamma''_0\rangle = |\Gamma'_0\rangle]\right) \end{aligned}$$

so that

$$P[\text{This Procedure is Correct}] = \frac{1}{2} + \frac{1}{2} \left(P[b' = 0 | \, |\Gamma''_0\rangle = |\Gamma_0\rangle] - P[b' = 0 | \, |\Gamma''_0\rangle = |\Gamma'_0\rangle]\right)$$

If this quantity is not at least $\frac{1}{2}$, we obtain a strictly better procedure by switching our guesses; in any case, there is a procedure that can be used to distinguish $|\Gamma_0\rangle$ from $|\Gamma'_0\rangle$ with advantage $\frac{1}{2} \left(P[b' = 0 | \, |\Gamma''_0\rangle = |\Gamma_0\rangle] - P[b' = 0 | \, |\Gamma''_0\rangle = |\Gamma'_0\rangle]\right)$, and so

$$|P[b' = 0 | \, |\Gamma''_0\rangle = |\Gamma_0\rangle] - P[b' = 0 | \, |\Gamma''_0\rangle = |\Gamma'_0\rangle]| \leq \sqrt{1 - |\langle \Gamma_0 | \, \Gamma'_0 \rangle|^2}.$$

A similar argument gives that

$$|P[b' = 1 | \, |\Gamma''_1\rangle = |\Gamma_1\rangle] - P[b' = 1 | \, |\Gamma''_1\rangle = |\Gamma'_1\rangle]| \leq \sqrt{1 - |\langle \Gamma_1 | \, \Gamma'_1 \rangle|^2}.$$

Let $|\Gamma\rangle$ be the state given to $\mathcal{A}$ in the first game, and $|\Gamma'\rangle$ be the state given to $\mathcal{A}$ in

the second game. Then

$$\begin{aligned}
&|P[\mathcal{A} \text{ wins the first game}] - P[\mathcal{A} \text{ wins the second game}]| \\
=\ &|P[b' = 0 \wedge |\Gamma\rangle = |\Gamma_0\rangle] + P[b' = 1 \wedge |\Gamma\rangle = |\Gamma_1\rangle] \\
&\quad - P[b' = 0 \wedge |\Gamma'\rangle = |\Gamma'_0\rangle] - P[b' = 1 \wedge |\Gamma'\rangle = |\Gamma'_1\rangle]| \\
\leq\ &|P[b' = 0 \,|\, |\Gamma\rangle = |\Gamma_0\rangle]P[|\Gamma\rangle = |\Gamma_0\rangle] - P[b' = 0 \,|\, |\Gamma''_0\rangle = |\Gamma'_0\rangle]P[|\Gamma'\rangle = |\Gamma_0\rangle]| \\
&\quad + |P[b' = 1 \,|\, |\Gamma\rangle = |\Gamma_1\rangle]P[|\Gamma\rangle = |\Gamma_1\rangle] - P[b' = 1 \,|\, |\Gamma''_1\rangle = |\Gamma'_1\rangle]P[|\Gamma'\rangle = |\Gamma_1\rangle]| \\
\leq\ &\frac{1}{2}\left(\sqrt{1 - |\langle\Gamma_0|\,\Gamma'_0\rangle|^2} + \sqrt{1 - |\langle\Gamma_1|\,\Gamma'_1\rangle|^2}\right)
\end{aligned}$$

as required. □

Corollary 4.8. Suppose the global state in an instance of the security experiment for the protocol just before the `Test` query is issued by $\mathcal{A}$ be

$$\begin{aligned}
|\Gamma\rangle = &\sum_{\vec{\mu}} \alpha_{\vec{\mu},m_I^*,\sigma_I^*,m_R^*,\sigma_R^*} |m_I^*, \sigma_I^*\rangle\, |m_R^*, \sigma_R^*\rangle \\
&+ \sum_{\substack{\vec{\mu} \\ \text{Verify}_{\text{pk}_I}(m_I,\sigma_I)=0 \\ \text{or } \text{Verify}_{\text{pk}_R}(m_R,\sigma_R)=0}} \alpha_{\vec{\mu},m_I,\sigma_I,m_R,\sigma_R} |m_I^*, \sigma_I^*\rangle\, |m_R^*, \sigma_R^*\rangle \\
&+ \sum_{\substack{\vec{\mu} \\ (m_I,\sigma_I,m_R,\sigma_R)\in\mathscr{F}}} \alpha_{\vec{\mu},m_I,\sigma_I,m_R,\sigma_R} |m_I^*, \sigma_I^*\rangle\, |m_R^*, \sigma_R^*\rangle
\end{aligned}$$

where (m_I^*, σ_I^*) is the message/signature pair that would actually have been sent by the initiating party in the test session, and (m_R^*, σ_R^*) is the corresponding response. After the `Test` query is issued, the challenger selects b uniformly at random from $\{0, 1\}$, and should

return the last three registers of the global state

$$\begin{aligned}|\Gamma_b\rangle = &\sum_{\vec{\mu}} \alpha_{\vec{\mu},m_I^*,\sigma_I^*,m_R^*,\sigma_R^*} |m_I^*,\sigma_I^*\rangle |m_R^*,\sigma_R^*\rangle |\kappa_b(m_I^*,m_R^*)\rangle \\ &+ \sum_{\substack{\vec{\mu} \\ \text{Verify}_{\text{pk}_I}(m_I,\sigma_I)=0 \\ \text{or } \text{Verify}_{\text{pk}_R}(m_R,\sigma_R)=0}} \alpha_{\vec{\mu},m_I,\sigma_I,m_R,\sigma_R} |m_I,\sigma_I\rangle |m_R,\sigma_R\rangle |\perp\rangle \\ &+ \sum_{\substack{\vec{\mu} \\ (m_I,\sigma_I,m_R,\sigma_R)\in\mathscr{F}}} \alpha_{\vec{\mu},m_I,\sigma_I,m_R,\sigma_R} |m_I,\sigma_I\rangle |m_R,\sigma_R\rangle |\kappa_b(m_I,m_R)\rangle\end{aligned}$$

to $\mathcal{A}$, where as before $\kappa_0(m_I, m_R)$ is the session key corresponding to messages m_I, m_R and $\kappa_1(m_I, m_R)$ is simply a random function. If instead $\mathcal{C}$ returns the last three registers of the state

$$\begin{aligned}|\Gamma'_b\rangle = &\sum_{\vec{\mu}} \alpha_{\vec{\mu},m_I^*,\sigma_I^*,m_R^*,\sigma_R^*} |m_I^*,\sigma_I^*\rangle |m_R^*,\sigma_R^*\rangle |\kappa_b(m_I^*,m_R^*)\rangle \\ &+ \sum_{\substack{\vec{\mu} \\ \text{Verify}_{\text{pk}_I}(m_I,\sigma_I)=0 \\ \text{or } \text{Verify}_{\text{pk}_R}(m_R,\sigma_R)=0}} \alpha_{\vec{\mu},m_I,\sigma_I,m_R,\sigma_R} |m_I,\sigma_I\rangle |m_R,\sigma_R\rangle |\perp\rangle \\ &+ \sum_{\substack{\vec{\mu} \\ (m_I,\sigma_I,m_R,\sigma_R)\in\mathscr{F}}} \alpha_{\vec{\mu},m_I,\sigma_I,m_R,\sigma_R} |m_I,\sigma_I\rangle |m_R,\sigma_R\rangle |\kappa_1(m_I,m_R)\rangle\end{aligned}$$

then except with negligible probability, the probability that $\mathcal{A}$ guesses the value of b correctly given this state differs at most negligibly from the probability that $\mathcal{A}$ guesses the value of b correctly given $|\Gamma_b\rangle$, regardless of the value of b.

Corollary 4.9. Let $|\Gamma\rangle$ be drawn from one of the following distributions, each with probability $\frac{1}{2}$:

$$\Delta : \sum_{\vec{\mu}} \alpha^*_{\vec{\mu}} |\vec{\mu}\rangle |m^*\rangle + \sum_{\vec{\mu};\, m\in M} \alpha_{\vec{\mu},m} |\vec{\mu}\rangle |m\rangle + \sum_{\vec{\mu};\, c\in C} \alpha_{\vec{\mu},c} |\vec{\mu}\rangle |c\rangle \text{ for } m^* \leftarrow \mathcal{D} \text{ and}$$

$$\hat{\Delta} : \sum_{\vec{\mu}} \alpha^*_{\vec{\mu}} |\vec{\mu}\rangle |\hat{m}^*\rangle + \sum_{\vec{\mu};\, m\in M} \alpha_{\vec{\mu},m} |\vec{\mu}\rangle |m\rangle + \sum_{\vec{\mu};\, c\in C} \alpha_{\vec{\mu},c} |\vec{\mu}\rangle |r(c)\rangle \text{ for } \hat{m}^* \leftarrow \hat{\mathcal{D}}$$

where $\mathcal{D}$ and $\mathcal{D}'$ are probability distributions on some set, M, C, and the supports of $\mathcal{D}$ and $\hat{\mathcal{D}}$ are disjoint, finite, nonempty sets, $r(c) \notin M$ for all $c \in C$, and $\sum_{\vec{\mu};\, c \in C} |\alpha_c|^2$ is negligible. Then if there is an efficient quantum adversary $\mathcal{A}$ which determines from which distribution $|\Gamma\rangle$ is drawn with non-negligible advantage Adv, then there is an efficient procedure, using $\mathcal{A}$ as a subroutine, which distinguishes $\mathcal{D}$ from $\hat{\mathcal{D}}$ with non-negligible advantage.

Proof. Let $\mathcal{A}$ be as described. Suppose you are given $\tilde{m}$ and wish to know from which distribution it is drawn. Construct the state

$$|\Gamma'\rangle = \sum_{\vec{\mu}} \alpha^*_{\vec{\mu}} |\vec{\mu}\rangle |\tilde{m}\rangle + \sum_{\vec{\mu};\, m \in M} \alpha_{\vec{\mu},m} |\vec{\mu}\rangle |m\rangle + \sum_{\vec{\mu};\, c \in C} \alpha_{\vec{\mu},c} |\vec{\mu}\rangle |c\rangle .$$

Notice that if $\tilde{m}$ is drawn from $\mathcal{D}$ then $|\Gamma'\rangle$ is drawn from Δ, while if $\tilde{m}$ is drawn from $\hat{\mathcal{D}}$, then $\sqrt{1 - |\langle\Gamma|\,\Gamma'\rangle|^2}$ is negligible; hence the probability that $\mathcal{A}$ wins the game given $|\Gamma'\rangle$ differs only negligibly from the probability that $\mathcal{A}$ wins the game given a true sample from Δ or $\hat{\Delta}$ by Lemma 4.7.

We will guess that $\tilde{m}$ is drawn from $\mathcal{D}$ if $\mathcal{A}$ guesses that $|\Gamma'\rangle$ is drawn from Δ, and we guess that $\tilde{m}$ is drawn from $\hat{\mathcal{D}}$ if $\mathcal{A}$ guesses that $|\Gamma'\rangle$ is drawn from $\hat{\Delta}$. The probability that we guess correctly is then

$$\begin{aligned}
&P[\text{We guess correctly}] \\
&\qquad = P[\mathcal{A} \text{ guesses } \Delta \mid \tilde{m} \leftarrow \mathcal{D}] + P[\mathcal{A} \text{ guesses } \hat{\Delta} \mid \tilde{m} \leftarrow \hat{\mathcal{D}}] \\
&\qquad \geq P[\mathcal{A} \text{ guesses } \Delta \mid\ |\Gamma\rangle \leftarrow \Delta] + P[\mathcal{A} \text{ guesses } \hat{\Delta} \mid\ |\Gamma'\rangle \leftarrow \hat{\Delta}] \\
&\qquad\quad - |P[\mathcal{A} \text{ is correct} \mid\ |\Gamma\rangle \leftarrow \Delta \text{ or } \hat{\Delta}] - P[\mathcal{A} \text{ is correct} \mid\ |\Gamma\rangle = |\Gamma'\rangle]| \\
&\qquad \geq \frac{1}{2} + \text{Adv} - \epsilon
\end{aligned}$$

for a negligible function ϵ. Indeed, this procedure works with non-negligible advantage Adv $- \epsilon$, as required. □

Finally we are able to prove the security of the constructed protocol Π'.

Proof (Theorem 4.1). Suppose we are faced with an instance of the security game for Π; that is, given the messages sent by the initiator $\mathcal{P}_I$ in session Ψ with responder $\mathcal{P}_R$ in

session Ψ' are m_I^* and m_R^*, respectively, we wish to determine whether a given string κ_b is the true session key for session Ψ if $b = 0$, or a random string if $b = 1$. Suppose to the contrary that there is an adversary $\mathcal{A}$ who wins the security game against Π' with non-negligible advantage Adv. We will use $\mathcal{A}$ as a distinguisher for our instance of the security game against Π.

Before starting an instance of $\mathcal{A}$, select two indices i^*, j^* which are less than the (polynomially-bounded) number of parties p that $\mathcal{A}$ will require. Further, choose a number s^* less than the (polynomially-bounded) number of pairs of session ψ that $\mathcal{A}$ will use. Run $\mathcal{A}$ essentially as usual, but with the following modifications.

Set the private key/public key information for $\mathcal{P}_{i^*}$ and $\mathcal{P}_{j^*}$ to that of $\mathcal{P}_I$ and $\mathcal{P}_R$, respectively. If $\mathcal{A}$ initiates fewer than s^* sessions, if its $s^{*\text{th}}$ session is not initiated by $\mathcal{P}_{i^*}$ with responder $\mathcal{P}_{j^*}$, or if its $s^{*\text{th}}$ session is not the test session, abort and select new i^*, j^* and s^*.

Given that $\mathcal{A}$ initiates the $s^{*\text{th}}$ with initiator $\mathcal{P}_{i^*}$ and responder P_{j^*}, set the session identifier as Ψ and the peer session identifier as Ψ'. Set the initiator's outgoing message as $(m_I^*, \sigma_I^* = \text{Sign}_{\text{sk}_I}(\texttt{id}_I, \texttt{id}_R; \Psi; m_I^*; r_\Psi))$ and the responder's message as (m_R^*, σ_R^*), where $\sigma_R^* = \text{Sign}_{\text{sk}_R}(\texttt{id}_I, \texttt{id}_R; \Psi, \Psi'; m_R^*; r_{\Psi'})$. If this is not eventually the test session, abort and choose i^*, j^* and s^* again; in particular, if $\mathcal{A}$ ever issues a command that would make either session no longer clean, abort.

By Lemma 4.5, we know that the adversary cannot construct a state for which the amplitude of a valid responding message is non-negligible; hence by Lemma 4.3 we know that $\mathcal{A}$ must pass some registers to the responding party $\mathcal{P}_{j^*}$, since otherwise the probability amplitude of states for which the session key obtained from the registers sent to the initiating party will be valid for Ψ is negligible, and there will be no matching session Ψ' to test. Moreover, the adversary must either deliver some response registers to the initiating party or the test session must be Ψ', since otherwise the adversary cannot win the game

with non-negligible advantage. In either case, the global state before the test session is

$$\begin{aligned}|\Gamma\rangle = &\sum_{\vec{\mu}} \alpha_{\vec{\mu},m_I^*,\sigma_I^*,m_R^*,\sigma_R^*} |m_I^*,\sigma_I^*\rangle |m_R^*,\sigma_R^*\rangle \\ &+ \sum_{\substack{\vec{\mu} \\ \mathrm{Verify}_{\mathrm{pk}_I}(m_I,\sigma_I)=0 \\ \text{or } \mathrm{Verify}_{\mathrm{pk}_R}(m_R,\sigma_R)=0}} \alpha_{\vec{\mu},m_I,\sigma_I,m_R,\sigma_R} |m_I,\sigma_I\rangle |m_R,\sigma_R\rangle \\ &+ \sum_{\substack{\vec{\mu} \\ (m_I,\sigma_I,m_R,\sigma_R)\in\mathscr{F}}} \alpha_{\vec{\mu},m_I,\sigma_I,m_R,\sigma_R} |m_I,\sigma_I\rangle |m_R,\sigma_R\rangle\end{aligned}$$

where the second register is the one delivered to $\mathcal{P}_{j^*}$ and the third is the one obtained from $\mathcal{P}_{j^*}$ in session Ψ', possibly after applying some unitary operator, and, except with negligible probability,

$$\Phi = \sum_{\substack{\vec{\mu} \\ (m_I,\sigma_I,m_R,\sigma_R)\in\mathscr{F}}} |\alpha_{\vec{\mu},m_I,\sigma_I,m_R,\sigma_R}|$$

is negligible, again by Lemma 4.5.

When the `Test` query is issued, if the test session is Ψ apply the map:

$$|m_R,\sigma_R\rangle |y\rangle \mapsto \begin{cases} |m_R,\sigma_R\rangle |y \oplus \kappa_b\rangle & \text{if } m_R = m_R^* \text{ and } \sigma_R = \sigma_R^* \\ |m_R,\sigma_R\rangle |y \oplus \perp\rangle & \text{if } \mathrm{Verify}_{\mathrm{sk}_R}(m_R,\sigma_R) = 0 \\ |m_R,\sigma_R\rangle |y \oplus \rho(m_R)\rangle & \text{otherwise} \end{cases}$$

to the register received by $\mathcal{P}_{i^*}$ in session Ψ and the target register provided by $\mathcal{A}$, where ρ maps pairs of messages to random strings. If instead the test session is Ψ', apply the map

$$|m_I,\sigma_I\rangle |y\rangle \mapsto \begin{cases} |m_I,\sigma_I\rangle |y \oplus \kappa_b\rangle & \text{if } m_I = m_I^* \text{ and } \sigma_I = \sigma_I^* \\ |m_I,\sigma_I\rangle |y \oplus \perp\rangle & \text{if } \mathrm{Verify}_{\mathrm{sk}_I}(m_I,\sigma_I) = 0 \\ |m_I,\sigma_I\rangle |y \oplus \rho(m_I)\rangle & \text{otherwise} \end{cases}$$

In either case, the global state after the test query is

$$\begin{aligned}|\Gamma\rangle = &\sum_{\vec{\mu},y} \alpha_{\vec{\mu},m_I^*,\sigma_I^*,m_R^*,\sigma_R^*} |m_I^*,\sigma_I^*\rangle |m_R^*,\sigma_R^*\rangle |y \oplus \kappa_b\rangle \\ &+ \sum_{\substack{\vec{\mu},y \\ \text{Verify}_{\text{pk}_I}(m_I,\sigma_I)=0 \\ \text{or } \text{Verify}_{\text{pk}_R}(m_R,\sigma_R)=0}} \alpha_{\vec{\mu},m_I,\sigma_I,m_R,\sigma_R} |m_I,\sigma_I\rangle |m_R,\sigma_R\rangle |y \oplus \perp\rangle \\ &+ \sum_{\substack{\vec{\mu},y \\ (m_I,\sigma_I,m_R,\sigma_R)\in\mathscr{F}}} \alpha_{\vec{\mu},m_I,\sigma_I,m_R,\sigma_R} |m_I,\sigma_I\rangle |m_R,\sigma_R\rangle |y \oplus \rho'(m_I,m_R)\rangle\,.\end{aligned}$$

Notice that this is simply $|\Gamma'_b\rangle$ from Corollary 4.8, and so $\mathcal{A}$ will guess b correctly (in the context of its security game) with advantage $\text{Adv} - \epsilon$ for some negligible function ϵ. Then, by Corollary 4.9, this correct guess will be the correct guess for our security game with advantage $\text{Adv} - \epsilon - \epsilon'$ where ϵ' is negligible; in particular, our advantage is non-negligible provided that we have chosen the correct i^*, j^*, and s^*. Since we choose these correctly with probability at least $\frac{1}{p^2\psi}$, a polynomial fraction, we see that our probability of winning the game using $\mathcal{A}$ as a subroutine is at least $\frac{1}{2} + \frac{\text{Adv}-\epsilon-\epsilon'}{p^2\psi}$ which is non-negligibly greater than a half; that is, protocol Π is insecure. This is a contradiction, and so no such adversary $\mathcal{A}$ must exist; that is, the protocol Π' is secure. □

Chapter 5

A Secure Protocol from Supersingular Elliptic Curve Isogenies

In this chapter we apply the generic construction from Section 4.5 to construct a secure authenticated key establishment protocol whose underlying computational problem is the Supersingular Isogeny Decision Diffie-Hellman Problem (SSDDH). The underling key establishment protocol is De Feo, Jao, and Plût's scheme [9], with authentication provided by a signature scheme constructed by applying Eaton and Song's [11] and Boneh and Zhandry's [5] transformations to Sun *et. al's* [35] strong designated verifier signature scheme, similar to the method applied in Soukharev, Jao, and Seshadri's work [34].

To begin we define the required global parameters. For authentication we require

1. $p_A = \ell_S^{e_S} \ell_V^{e_V} f_A \pm 1$, a prime, where ℓ_S and ℓ_V are prime, and f_A is a small cofactor used so that p_A is prime;
2. E_A, a supersingular elliptic curve defined over $\mathbb{K}_A = GF(p_A^2)$;
3. $\{P_S, Q_S\}$ and $\{P_V, Q_V\}$, bases for $E_A[\ell_S^{e_S}]$ and $E_A[\ell_V^{e_V}]$, respectively;
4. $\mathcal{H} = (\text{KeyGen}^{(H_c)}, H_c, \text{Invert}, \text{Sample})$, a quantum-safe chameleon hash function; and,

5. $\mathcal{O}_1, \mathcal{O}_2, \mathcal{O}_3$, random oracles.

For key establishment we require

1. $p_K = \ell_I^{e_I} \ell_R^{e_R} f_K \pm 1$, a prime, where ℓ_I and ℓ_R are prime, and f_K is a small cofactor used so that p_K is prime;
2. E_K, a supersingular elliptic curve defined over $\mathbb{K}_K = GF(p_K^2)$; and,
3. $\{P_I, Q_I\}$ and $\{P_R, Q_R\}$, bases for $E_K[\ell_I^{e_I}]$ and $E_K[\ell_R^{e_R}]$, respectively.

Each party $\mathcal{P}_k$ must establish authentication keys; associated to each will be a private key and public key for signing and a private key and public key for verification. In particular, $\mathcal{P}_k$ selects $m_S^{(k)}, n_S^{(k)} \in \mathbb{Z}/\ell_S^{e_S}\mathbb{Z}$ not both divisible by ℓ_S uniformly at random, and sets $E_S^{(k)} = E_A / \left\langle m_S^{(k)} P_S + n_S^{(k)} Q_S \right\rangle$. Further define $\phi_S^{(k)}$ to be the isogeny with domain E_A and image $E_S^{(k)}$. Similarly $\mathcal{P}_k$ selects $m_V^{(k)}, n_V^{(k)} \in \mathbb{Z}/\ell_V^{e_V}\mathbb{Z}$ not both divisible by ℓ_V uniformly at random, sets $E_V^{(k)} = E_A / \left\langle m_V^{(k)} P_V + n_V^{(k)} Q_V \right\rangle$, and further sets $\phi_V^{(k)}$ to be the isogeny with domain E_A and image $E_V^{(k)}$. The party must also select a private key/public key pair $(\mathrm{sk}_H^{(k)}, \mathrm{pk}_H^{(k)})$ for the chameleon hash function. Then $\mathcal{P}_k$'s authentication key pair is

$$\begin{aligned}(\mathrm{sk}^{(k)}, \mathrm{pk}^{(k)}) &= \left((\mathrm{sk}_S^{(k)}, \mathrm{sk}_V^{(k)}, \mathrm{sk}_H^{(k)}), (\mathrm{pk}_S^{(k)}, \mathrm{pk}_V^{(k)}, \mathrm{pk}_H^{(k)})\right) \\ &= \Big(\left((m_S^{(k)}, n_S^{(k)}), (m_V^{(k)}, n_V^{(k)}), \mathrm{sk}_H^{(k)} \right), \\ &\qquad \left((E_S^{(k)}, \phi_S^{(k)}(P_V), \phi_S^{(k)}(Q_V)), (E_V^{(k)}, \phi_V^{(k)}(P_S), \phi_V^{(k)}(Q_S)), \mathrm{pk}_H^{(k)} \right) \Big).\end{aligned}$$

Then associated to each ordered pair $(\mathcal{P}_k, \mathcal{P}_\ell)$ of parties is a curve $E_{SV}^{(k,\ell)}$ defined by

$$E_{SV}^{(k,\ell)} = E_V^{(\ell)} / \left\langle m_S^{(k)} \phi_V^{(\ell)}(P_S) + n_S^{(k)} \phi_V^{(\ell)}(Q_S) \right\rangle = E_S^{(k)} / \left\langle m_V^{(\ell)} \phi_S^{(k)}(P_V) + n_V^{(\ell)} \phi_S^{(k)}(Q_V) \right\rangle$$

which both $\mathcal{P}_k$ and $\mathcal{P}_\ell$ can compute using their secret keys and the other's public keys. This curve will be used for $\mathcal{P}_k$ to sign a message to $\mathcal{P}_\ell$.

The protocol is as follows (here $\mathcal{P}_k$ is the initiator and $\mathcal{P}_\ell$ is the responder).

1. Upon being instructed to start a session with $\mathcal{P}_\ell$, $\mathcal{P}_k$:

a) Selects a session identifier Ψ;

b) Selects $x_I^{(\Psi)}, y_I^{(\Psi)} \in \mathbb{Z}/\ell_I^{e_I}\mathbb{Z}$, not both divisible by ℓ_I, uniformly at random;

c) Constructs $R^{(\Psi)} = x_I^{(\Psi)} P_I + y_I^{(\Psi)} Q_I$, defines $\phi^{(\Psi)}$ to be the isogeny with kernel $\left\langle R^{(\Psi)} \right\rangle$, and sets

$$m^{(\Psi)} = (E^{(\Psi)} = E_K / \left\langle R^{(\Psi)} \right\rangle, \phi^{(\Psi)}(P_R), \phi^{(\Psi)}(Q_R), \mathtt{id}_k, \mathtt{id}_\ell, \Psi);$$

d) Selects $r^{(\Psi)} \in \{0,1\}^*$ at random;

e) Sets

$$\begin{aligned} \sigma^{(\Psi)} &= (\sigma_1, \sigma_2, \sigma_3) \\ &= (\text{Sample}(\lambda), \mathcal{O}_1(r^{(\Psi)} || j(E_{SV}^{(k,\ell)})), \text{Invert}_{\text{sk}_H^{(k)}}(r^{(\Psi)}, \mathcal{O}_2(\mathcal{O}_3(m^{(\Psi)}, \sigma_1) || \sigma_2))), \end{aligned}$$

and;

f) Activates Send$(\mathtt{id}_\ell, m^{(\Psi)}, \sigma^{(\Psi)})$

2. Upon receiving (m, σ), $\mathcal{P}_\ell$:

a) Computes

$$b^{(\Psi')} = \begin{cases} 1 \text{ if } \sigma_2 = \mathcal{O}_1(H_{\text{pk}_H^{(k)}}(\mathcal{O}_2(\mathcal{O}_3(m, \sigma_1) || \sigma_2), \sigma_3) || j(E_{SV}^{(k,\ell)})) \\ 0 \text{ otherwise} \end{cases};$$

If $b^{(\Psi')} = 0$ the delivered message is invalid and is hence rejected; then $\mathcal{P}_\ell$ activates Send$(\mathtt{id}_k; \perp, \perp, \perp)$. Otherwise, $\mathcal{P}_\ell$:

b) Selects a session identifier Ψ';

c) Selects $x_R^{(\Psi')}, y_R^{(\Psi')} \in \mathbb{Z}/\ell_R^{e_R}\mathbb{Z}$, not both divisible by ℓ_R, uniformly at random;

d) Constructs $R^{(\Psi')} = x_R^{(\Psi')} P_R + y_R^{(\Psi')} Q_R$, defines $\phi^{(\Psi')}$ to be the isogeny with kernel $\left\langle R^{(\Psi')} \right\rangle$, and sets

$$m^{(\Psi')} = (E^{(\Psi')} = E_K / \left\langle R^{(\Psi')} \right\rangle, \phi^{(\Psi')}(P_I), \phi^{(\Psi')}(Q_I), \mathtt{id}_k, \mathtt{id}_\ell, \Psi, \Psi');$$

e) Selects $r^{(\Psi')} \in \{0,1\}^*$ at random;

f) Sets

$$\begin{aligned}\sigma^{(\Psi')} &= (\sigma_1, \sigma_2, \sigma_3) \\ &= (\text{Sample}(\lambda), \mathcal{O}_1(r^{(\Psi')}||j(E_{SV}^{(\ell,k)})), \text{Invert}_{\text{sk}_H^{(\ell)}}(r^{(\Psi')}, \mathcal{O}_2(\mathcal{O}_3(m^{(\Psi')}, \sigma_1)||\sigma_2))),\end{aligned}$$

and;

g) Activates $\text{Send}(\texttt{id}_\ell, m^{(\Psi')}, \sigma^{(\Psi')})$

After receiving the message, if $\mathcal{P}_\ell$ needs to compute the session key, it computes

$$K^{(\Psi')} = \begin{cases} \perp \text{ if } b^{(\Psi')} = 0 \\ m_1 / \left\langle x_R^{(\Psi')} m_2 + y_R^{(\Psi')} m_3 \right\rangle \text{ otherwise} \end{cases}.$$

3. Upon receiving (m, σ), $\mathcal{P}_k$ computes

$$b^{(\Psi)} = \begin{cases} 1 \text{ if } \sigma_2 = \mathcal{O}_1(H_{\text{pk}_H^{(\ell)}}(\mathcal{O}_2(\mathcal{O}_3(m, \sigma_1)||\sigma_2), \sigma_3)||j(E_{SV}^{(\ell,k)})) \\ 0 \text{ otherwise} \end{cases};$$

After receiving the message, if $\mathcal{P}_k$ needs to compute the session key, it computes

$$K^{(\Psi)} = \begin{cases} \perp \text{ if } b^{(\Psi)} = 0 \\ m_1 / \left\langle x_I^{(\Psi)} m_2 + y_I^{(\Psi)} m_3 \right\rangle \text{ otherwise} \end{cases}.$$

Theorem 5.1 (Correctness of the Scheme)**.** *The scheme described above is correct.*

Proof. Suppose that all messages are relayed faithfully. It is clear that $b^{(\Psi')} = 1$ and

$b^{(\Psi)} = 1$ in this case, and so

$$\begin{aligned}
K^{(\Psi)} &= j\left(E^{(\Psi')} / \left\langle x_I^{(\Psi)} \phi^{(\Psi')}(P_I) + y_I^{(\Psi)} \phi^{(\Psi')}(Q_I) \right\rangle\right) \\
&= j\left(\left(E_K / \left\langle x_R^{(\Psi')} P_R + y_R^{(\Psi')} Q_R \right\rangle\right) / \left\langle \phi^{(\Psi')}\left(x_I^{(\Psi)} P_I + y_I^{(\Psi)} Q_I\right) \right\rangle\right) \\
&= j\left(E_K / \left\langle x_R^{(\Psi')} P_R + y_R^{(\Psi')} Q_R, x_I^{(\Psi)} P_I + y_I^{(\Psi)} Q_I \right\rangle\right) \\
&= j\left(\left(E_K / \left\langle x_I^{(\Psi)} P_I + y_I^{(\Psi')} Q_I \right\rangle\right) / \left\langle \phi_\Psi\left(x_R^{(\Psi')} P_R + y_R^{(\Psi')} Q_R\right) \right\rangle\right) \\
&= j\left(E_\Psi / \left\langle x_R^{(\Psi')} \phi_\Psi(P_R) + y_R^{(\Psi')} \phi_\Psi(Q_R) \right\rangle\right) = K^{(\Psi')};
\end{aligned}$$

that is, the session keys are equal, as required. □

Theorem 5.2 (Security of the Scheme)**.** *Under the Supersingular Isogeny Decisional Diffie-Hellman assumption, the scheme described above is secure in the security model described in Chapter 4 in the quantum random oracle model.*

Proof. By Theorem 4.1 it suffices to show that the underlying signature scheme is EUF-qCMA, and that the underlying key establishment protocol is secure if the adversary is restricted to delivering messages faithfully.

The signature scheme is constructed by applying the constructions from Theorem 2.3 and Corollary 2.5 to Sun *et al.*'s SDV signature scheme, which is EUF-CMA against an adversary who can perform polynomially-bounded quantum computations in the random oracle model by Theorem 2.2. Thus the signature scheme in use is EUF-qCMA with quantumly-accessible random oracles, as required.

The underlying key establishment protocol is simply the protocol from [9, Section 3.1], with some additional information included in each message. We show that this is secure when all messages are relayed faithfully. Suppose there is an adversary $\mathcal{A}$ who breaks the security of the protocol with advantage ϵ while relaying all messages faithfully. Suppose we are faced with an instance

$$(E, E_A, E_B, \phi_A(P_B), \phi_A(Q_B), \phi_B(P_A), \phi_B(Q_A), E_C)$$

of the Supersingular Isogeny Decision Diffie-Hellman problem. To solve the problem, choose an integer i^* between 1 and ψ, a polynomial upper bound on the number of ses-

sion pairs that $\mathcal{A}$ will require. Run $\mathcal{A}$ as usual, except that if the $i^{*\text{th}}$ session pair is reached, set the initiator's message as $E_A, \phi_A(P_B), \phi_A(Q_B))$ and the responder's message as $(E_B, \phi_B(P_A), \phi_B(Q_A))$. If $\mathcal{A}$ requests a session key reveal on either session in this session pair, if this session pair is not reached, or if neither session in the pair is the test session, abort. Otherwise, one of the sessions is the test session, and in that session $\mathcal{A}$ must deliver the appropriate message to the appropriate party. When the `Test` query is issued, write E_C to the appropriate target registers. If $E_C = E_{AB}$, then it is the proper session key for the session, while if $E_C \neq E_{AB}$, it is simply a random element of the keyspace. Thus by guessing that $E_C = E_{AB}$ if and only if $\mathcal{A}$ guesses that the key is the true session key, we solve the Supersingular Isogeny Decisional Diffie-Hellman problem with advantage ϵ. The probability that we select the correct session pair is at least $\frac{1}{\psi}$, and so our overall advantage is $\frac{\epsilon}{\psi}$; thus under the Supersingular Isogeny Decisional Diffie-Hellman assumption $\frac{\epsilon}{\psi}$ is negligible, and hence ϵ is negligible; that is, the scheme is secure against an adversary who is restricted to delivering messages faithfully.

Hence the protocol is indeed secure, as required. □

Chapter 6

Conclusions and Future Work

We have presented a security model for authenticated key establishment in which the adversary can deliver quantum superpositions of messages to parties who would ordinarily be participating in a classical protocol, analogous to allowing quantum signing queries in EUF-qCMA security of signature schemes or quantum encryption/decryption queries in standard post-quantum security definitions of encryption [5, 12]. We demonstrate that the corresponding new security definition is achievable by constructing a specific example of a secure key establishment protocol assuming the quantum hardness of a Diffie-Hellman-type problem for isogenies of supersingular elliptic curves, and give a generic construction for secure protocols using sufficiently secure signature schemes and unauthenticated key establishment protocols.

Although I would argue that the security model and definition presented in Chapter 4 is a natural one for post-quantum authenticated key establishment, before it is to be adopted it remains to establish a separation between this security definition and, for instance, the more standard Canetti-Krawczyk model with a quantum adversary—that is, we must show that there are protocols which are secure in the Canetti-Krawczyk model when the adversary has access to a quantum computer which are insecure when the adversary can deliver quantum superpositions of messages. We would like to establish this separation to demonstrate the *necessity* of our post-quantum security model and to convince people of its utility. Once this separation is established, we can work toward answering the following questions:

- Are current "post-quantum" authenticated key establishment protocols secure in this model?
- Are their other simple generic constructions for secure protocols? Does the encrypt-and-MAC paradigm from [1, Section 3.2] carry over the same way the signature-based method [1, Section 3.1] does?
- Are the protocols that arise from these generic constructions efficient? If not, how can we do better?
- How can we modify the model to include more sophisticated security properties such as key compromise impersonation resilience and resilience against malicious insiders [23]?

References

[1] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. A modular approach to the design and analysis of authentication and key exchange protocols. In *Proceedings of STOC '98*, pages 419–428. ACM Press, 1998.

[2] Mihir Bellare and Phillip Rogaway. Entity Authentication and Key Distribution. In Stinson, Douglas R., editor, *Advances in Cryptology — CRYPTO' 93: 13th Annual International Cryptology Conference Santa Barbara, California, USA August 22–26, 1993 Proceedings*, pages 232–249, Berlin, Heidelberg, 1994. Springer Berlin Heidelberg.

[3] Charles H. Bennett and Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing . *Theoretical Computer Science* , 560, Part 1():7 – 11, 2014.

[4] Jean-François Biasse, David Jao, and Anirudh Sankar. A Quantum Algorithm for Computing Isogenies between Supersingular Elliptic Curves. In Meier, Willi and Mukhopadhyay, Debdeep, editor, *Progress in Cryptology – INDOCRYPT 2014: 15th International Conference on Cryptology in India, New Delhi, India, December 14-17, 2014, Proceedings*, pages 428–442, Cham, 2014. Springer International Publishing.

[5] Dan Boneh and Mark Zhandry. Secure Signatures and Chosen Ciphertext Security in a Quantum Computing World. In Canetti, Ran and Garay, Juan A., editor, *Advances in Cryptology – CRYPTO 2013: 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II*, pages 361–379, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[6] Ran Canetti, Oded Goldreich, and Shai Halevi. The Random Oracle Methodology, Revisited. *J. ACM*, 51(4):557–594, July 2004.

[7] Ran Canetti and Hugo Krawczyk. Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels. In Pfitzmann, Birgit, editor, *Advances in Cryptology — EUROCRYPT 2001: International Conference on the Theory and Application of Cryptographic Techniques Innsbruck, Austria, May 6–10, 2001 Proceedings*, pages 453–474, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[8] Michael Coglianese and Bok-Min Goi. MaTRU: A New NTRU-Based Cryptosystem. In Maitra, Subhamoy and Veni Madhavan, C.E. and Venkatesan, Ramarathnam, editor, *Progress in Cryptology - INDOCRYPT 2005*, volume 3797 of *Lecture Notes in Computer Science*, pages 232–243. Springer Berlin Heidelberg, 2005.

[9] Luca De Feo, David Jao, and Jérôme Plût. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. *J. Math. Cryptol.*, 8:209–247, 2014.

[10] Whitfield Diffie and Martin Hellman. New Directions in Cryptography. *IEEE Trans. Inf. Theor.*, 22(6):644–654, September 2006.

[11] Edward Eaton and Fang Song. Making Existential-Unforgeable Signatures Strongly Unforgeable in the Quantum Random-Oracle Model. In *Proceedings of TQC*, pages 1–16, Germany, 2015. Dagstuhl Publishing.

[12] Tommaso Gagliardoni, Andreas H ulsing, and Christian Schaffner. Semantic Security and Indistinguishability in the Quantum World. *CoRR*, abs/1504.05255, 2015.

[13] Shafi Goldwasser and Silvio Micali. Probabilistic Encryption & How to Play Mental Poker Keeping Secret All Partial Information. In *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing*, STOC '82, pages 365–377, New York, NY, USA, 1982. ACM.

[14] Robin Hartshorne. *Algebraic Geometry*. Number 52 in Graduate Texts in Mathematics. Springer-Verlag, New York, 1977.

[15] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A Ring-Based Public Key Cryptosystem. In Buhler, Joe P., editor, *Algorithmic Number Theory*, volume 1423 of *Lecture Notes in Computer Science*, pages 267–288. Springer Berlin Heidelberg, 1998.

[16] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. *An Introduction to Mathematical Cryptography*. Springer, New York, 2008.

[17] Thomas W. Hungerford. *Algebra*. Number 73 in Graduate Texts in Mathematics. Springer-Verlag, New York, 1974.

[18] David Jao and Vladimir Soukharev. Isogeny-Based Quantum-Resistant Undeniable Signatures. In Mosca, Michele, editor, *Post-Quantum Cryptography: 6th International Workshop, PQCrypto 2014, Waterloo, ON, Canada, October 1-3, 2014. Proceedings*, pages 160–179, Cham, 2014. Springer International Publishing.

[19] Katherine Jarvis and Monica Nevins. ETRU: NTRU over the Eisenstein integers. *Designs, Codes and Cryptography*, pages 1–24, 2013.

[20] Phillip Kaye, Raymond Laflamme, and Michele Mosca. *An Introduction to Quantum Computing*. Oxford University Press, Oxford, United Kingdom, 2007.

[21] Aviad Kipnis, Jacques Patarin, and Louis Goubin. Unbalanced Oil and Vinegar Signature Schemes. In Stern, Jacques, editor, *Advances in Cryptology EUROCRYPT 99*, volume 1592 of *Lecture Notes in Computer Science*, pages 206–222. Springer Berlin Heidelberg, 1999.

[22] Hugo Krawczyk and Tal Rabin. Chameleon Hashing and Signatures, 1997.

[23] Brian LaMacchia, Kristin Lauter, and Anton Mityagin. Stronger Security of Authenticated Key Exchange. In Susilo, Willy and Liu, Joseph K. and Mu, Yi, editor, *Provable Security: First International Conference, ProvSec 2007, Wollongong, Australia, November 1-2, 2007. Proceedings*, pages 1–16, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[24] Robert J. McEliece. A Public-Key Cryptosystem Based on Algebraic Coding Theory. *DSN Progress Report*, 1978.

[25] Ralph Charles Merkle. *Secrecy, Authentication, and Public Key Systems*. PhD thesis, Stanford University, Stanford, CA, USA, 1979.

[26] Michele Mosca, Douglas Stebila, and Berkant Ustaoğlu. Quantum Key Distribution in the Classical Authenticated Key Exchange Framework. In Gaborit, Philippe, editor, *Post-Quantum Cryptography: 5th International Workshop, PQCrypto 2013, Limoges, France, June 4-7, 2013. Proceedings*, pages 136–154, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[27] Harald Neiderreiter. Knapsack-Type Cryptosystems and Algebraic Coding Theory. *Problems of Control and Information Theory*, 15:159–166, 1986.

[28] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, United Kingdom, 10th Anniversary edition, 2010.

[29] Jacques Patarin. Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): Two New Families of Asymmetric Algorithms. In Maurer, Ueli, editor, *Advances in Cryptology EUROCRYPT 96*, volume 1070 of *Lecture Notes in Computer Science*, pages 33–48. Springer Berlin Heidelberg, 1996.

[30] Ron L. Rivest, Adi Shamir, and Leonard Adleman. A Method for Obtaining Digital Signatures and Public-key Cryptosystems. *Commun. ACM*, 21(2):120–126, February 1978.

[31] Joseph J. Rotman. *An Introduction to the Theory of Groups*. Number 148 in Graduate Texts in Mathematics. Springer, New York, 1995.

[32] Peter W. Shor. Algorithms for Quantum Computation: Discrete Logarithms and Factoring. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, SFCS '94, pages 124–134, Washington, DC, USA, 1994. IEEE Computer Society.

[33] Joseph H. Silverman. *The Arithmetic of Elliptic Curves*. Number 106 in Graduate Texts in Mathematics. Springer, New York, 1986.

[34] Vladimir Soukharev, David Jao, and Srinath Seshadri. Post-Quantum Security Models for Authenticated Encryption. In Takagi, Tsuyoshi, editor, *Post-Quantum Cryptography: 7th International Workshop, PQCrypto 2016, Fukuoka, Japan, February 24-26, 2016, Proceedings*, pages 64–78, Cham, 2016. Springer International Publishing.

[35] Xi Sun, Haibo Tian, and Yumin Wang. Toward Quantum-resistant Strong Designated Verifier Signature. *Int. J. Grid Util. Comput.*, 5(2):80–86, 2014.

[36] John Tate. Endomorphisms of Abelian Varieties Over Finite Fields. *Invent. Math.*, 2:134 – 144, 1966.

[37] Dominique Unruh. Quantum Position Verification in the Random Oracle Model. In Garay, Juan A. and Gennaro, Rosario, editor, *Advances in Cryptology – CRYPTO 2014: 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part II*, pages 1–18, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[38] Jacques Vélu. Isogénies Entre Courbes Elliptiques. *C. R. Acad. Sci. Paris Sér. A-B*, 273:A238 – A241, 1971.

[39] Mark Zhandry. Secure Identity-Based Encryption in the Quantum Random Oracle Model. In *Proceedings of CRYPTO*, 2012.