

T.R.
GEBZE TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

POST-QUANTUM
CRYPTOGRAPHIC PROTOCOLS
BASED ON ISOGENIES OF ELLIPTIC CURVES

MEHMET EMİN GÖNEN
A THESIS SUBMITTED FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
DEPARTMENT OF MATHEMATICS

GEBZE

2022

T.R.
GEBZE TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

POST-QUANTUM
CRYPTOGRAPHIC PROTOCOLS
BASED ON ISOGENIES OF ELLIPTIC CURVES

MEHMET EMİN GÖNEN
A THESIS SUBMITTED FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
DEPARTMENT OF MATHEMATICS

THESIS SUPERVISOR
ASSOC. PROF. DR. SEHER TUTDERE KAVUT
II. THESIS SUPERVISOR
PROF. DR. OSMANBEY UZUNKOL

GEBZE
2022

T.C.
GEBZE TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ELİPTİK EĞRİLERİN İZOJENİLERİ TABANLI
KUANTUM SONRASI
KRİPTOGRAFİK PROTOKOLLER

MEHMET EMİN GÖNEN
DOKTORA TEZİ
MATEMATİK ANABİLİMDALI

DANIŞMANI
DOÇ. DR. SEHER TUTDERE KAVUT
II. DANIŞMANI
PROF. DR. OSMANBEY UZUNKOL

GEBZE

2022

GTÜ Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 07/02/2022 tarih ve 2022/08 sayılı kararıyla oluşturulan jüri tarafından 28/02/2022 tarihinde tez savunma sınavı yapılan Mehmet Emin GÖNEN'in tez çalışması Matematik Anabilim Dalında DOKTORA tezi olarak kabul edilmiştir.

JÜRİ

ÜYE

(TEZ DANIŞMANI) : Doç. Dr. Seher TUTDERE KAVUT

ÜYE : Prof. Dr. Mustafa AKKURT

ÜYE : Doç. Dr. Kağan KURŞUNGÖZ

ÜYE : Doç Dr. Oğuz YAYLA

ÜYE : Dr. Öğr. Üyesi Roghayeh HAFAZİEH

ONAY

Gebze Teknik Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun
...../...../..... tarih ve/..... sayılı kararı.

SUMMARY

Elliptic curves defined over $\mathbb{F}_q$ are essential algebraic objects used in the field of cryptography. There are many cryptographic algorithms and protocols whose security is based on elliptic curve discrete logarithm problem, namely, asymmetric encryption, key agreement protocols and digital signature. In this thesis, cryptographic protocols based on elliptic curves in the literature have been reviewed. A new key agreement protocol is defined and compared with existing protocols. Moreover, within the scope of this thesis, the mathematical ideas behind isogeny based cryptography has been comprehensively studied on. Isogeny based key agreement protocols have been analysed and new attacks have been applied to some of the protocols in the literature. In addition, a new supersingular isogeny based key agreement protocol is designed and the security proof of this protocol is given.

Keywords: Elliptic Curves, Asymmetric Encryption, Key Agreement Protocols, Post-Quantum Cryptography, Isogenies.

ÖZET

Sonlu $\mathbb{F}_q$ cismi üzerinde tanımlı eliptik eğriler, kriptografi alanında kullanılan önemli cebirsel nesnelerdir. Ayrık logaritma probleminin zorluğuna dayalı; asimetrik şifreleme, anahtar anlaşıma ve sayısal imzalama gibi eliptik eğri tabanlı birçok kriptografik algoritma ve protokol bulunmaktadır. Bu tezde, literatürde bulunan eliptik eğri tabanlı kriptografik protokoller incelenmiştir. Yeni bir anahtar anlaşıma protokolü tanımlanmış ve mevcut protokollerle karşılaştırılmıştır. Ayrıca, bu tez kapsamında kuantum hesaplama dayanaklı olarak tasarlanan, eliptik eğrilerde izojeni bulma probleminin zorluğuna dayalı algoritma ve protokoller de kapsamlı bir şekilde incelenmiştir. Mevcut protokoller analiz edilip, kriptografik ataklar uygulanmıştır. Bununla birlikte, süpersingüler eliptik eğriler kullanılarak yeni bir anahtar anlaşıma protokolü tasarlanmış ve güvenlik ispatı yapılmıştır.

Anahtar Kelimeler: Eliptik Eğriler, Asimetrik Şifreleme, Anahtar Anlaşıma Protokolleri, Kuantum Sonrası Kriptografi, İzojeniler.

ACKNOWLEDGEMENTS

First of all, I would like to send my special thanks to my thesis supervisor Assoc. Prof. Seher Tutdere Kavut, for her great supervision, guidance and encouragement. I am also grateful to her for her trust in me.

I wish to express my sincere thanks to Prof. Osmanbey Uzunkol from Flensburg University of Applied Sciences, the thesis co-advisor. He has always supported me in bad and good times, never let me give up and helped me so much to keep my motivation.

Besides my advisors, I would like to thank the rest of my thesis committee: Assoc. Prof. Kağan Kurşungöz, Asst. Prof. Altan Erdoğan, Asst. Prof. Roghayeh Hafezieh, Assoc. Prof. Oğuz Yayla and Prof. Mustafa Akkurt for devoting their time to evaluate my thesis and their insightful comments.

My very specific thanks go to my wife Gamze, who was always there for me, making the distance shorter. And for many many other things that cannot be listed here... Her encouragement and constant support, together with her endless patience, I was able to overcome many difficulties during this study and also in general. I would like to thank my children Yusuf Kerem and Feride Bilge, without them nothing would be possible.

I also thank my parents and my brothers for their everlasting love and support.

And last but not least, the huge thanks go to my friends for tea time talk. Especially İbrahim Kendirli, Fatih Demirbaş, Burak Taha Akbaş, Tuğba Öztaban and Nafiz Başaran. Moreover, I would like to thank Kamil Ota, Muhammed Said Gündoğan and Atakan Arslan for their friendship and valuable comments.

Finally, I am grateful to TUBITAK BİLGEM for the support throughout my doctoral studies.

TABLE of CONTENTS

	<u>Page</u>
SUMMARY	v
ÖZET	vi
ACKNOWLEDGMENTS	vii
TABLE of CONTENTS	viii
LIST of ABBREVIATIONS and ACRONYMS	x
LIST of FIGURES	xi
LIST of TABLES	xii
1. INTRODUCTION	1
2. PRELIMINARIES	3
2.1. Elliptic Curves	3
2.2. Public Key Cryptography	12
2.3. Cryptographic Protocols	13
2.4. Security Definitions	16
3. PRE-QUANTUM CRYPTOGRAPHIC PROTOCOLS	19
3.1. Full Unified Diffie-Hellman	20
3.2. MTI-C0 and MTI-A0 Key Agreement Protocols	21
3.3. MQV and HMQV Key Agreement Protocols	23
3.4. The New Key Agreement Protocol	25
4. POST-QUANTUM CRYPTOGRAPHIC PROTOCOLS	27
4.1. Supersingular Isogeny Diffie-Hellman (SIDH)	28
4.2. Supersingular Isogeny Key Encapsulation (SIKE)	31
4.3. Commutative Supersingular Isogeny Diffie- Hellman(CSIDH)	32
4.4. Other Protocols	34
5. CRYPTANALYSIS OF SOME RECENT ISOGENY BASED CRYPTOGRAPHIC PROTOCOLS	36

5.1. Daghigh <i>et al.</i> Key Exchange Method	36
5.2. A Key Manipulation Attack on Daghigh <i>et al.</i> Protocols	37
5.3. Recovery of Isogeny Using Key Manipulation Attack	39
5.3.1. Countermeasures of Key Manipulation Attack	43
6. A NEW POST-QUANTUM AUTHENTICATED KEY AGREEMENT PROTOCOL (SI-MQV)	45
6.1. Security Models: CK Security and CK^+ Security	46
7. NEW CRYPTANALYSIS OF SOME PROTOCOLS	59
7.1. K-CI Attack on SIDH variant of the Jeong-Katz-Lee protocol	59
7.2. UK-S Attack on SIDH variant of MTI-A0	62
8. CONCLUSION AND REMARKS	65
REFERENCES	66
BIOGRAPHY	70
APPENDICES	71

LIST of ABBREVIATIONS and ACRONYMS

Abbreviations Explanations

and Acronyms

E, E', E_1, E_2	:	Elliptic Curves over a Finite Field $\mathbb{F}_q$
$\mathcal{O}_E$	:	The Identity Element of E
$j(E)$	:	The j -invariant of E
$\mathbb{Q}$	:	The Set of Rational Numbers
$\mathbb{C}$	:	The Set of Complex Numbers
$\overline{\mathbb{F}}_q$	:	The Algebraic Closure of $\mathbb{F}_q$
$\text{Hom}(E_1, E_2)$	:	The Group of Isogenies from E_1 to E_2
$E[\ell]$	:	The ℓ -torsion Group of E
$\langle X \rangle$	:	The Group Generated by the Point X
ϕ	:	The Isogeny Map
$\hat{\phi}$	:	The Dual Isogeny of ϕ
n_A	:	Static Private Key of Alice
n_B	:	Static Private Key of Bob
n_X	:	Ephemeral Private Key of Alice
n_Y	:	Ephemeral Private Key of Bob
P_A, Q_A	:	Public Points of Alice
P_B, Q_B	:	Public Points of Bob
E_A, E_X	:	Public Curves of Alice
E_B, E_Y	:	Public Curves of Bob
R_A, R_X	:	Private Values of Alice
R_B, R_Y	:	Private Values of Bob

LIST of FIGURES

<u>Figure No:</u>	<u>Page</u>
2.1: The geometric representation of point addition and point doubling.	5
3.1: Full Unified Diffie-Hellman.	21
3.2: MTI-C0 Protocol.	22
3.3: MTI-A0 Protocol.	22
3.4: Protocol-1.	23
3.5: MQV Protocol.	24
3.6: H-MQV Protocol.	25
3.7: The New Key Agreement Protocol.	26
4.1: Jao-De Feo Method.	29
4.2: SIKE Algorithm.	33
4.3: CSIDH Key Agreement Protocol.	34
5.1: Daghigh <i>et al.</i> Generalized Method 2.	37
5.2: A Key Manipulation Attack on Generalized Method 2 of [10].	38
5.3: Modified version of Daghigh <i>et al.</i> Generalized Method 2.	44
6.1: SI-MQV Protocol.	47
7.1: SIDH variant of the Jeong-Katz-Lee protocol.	60
7.2: K-CI attack scenario of Galbraith's protocol.	62
7.3: SIDH Variant of MTI-A0.	63
7.4: UK-S attack scenario.	64

LIST of TABLES

<u>Table No:</u>	<u>Page</u>
2.1: Security levels of algorithms (bits).	13
3.1: Security services offered by authenticated key agreement protocols.	20
4.1: SIKE parameter sets for each NIST security level.	32
7.1: Security services offered by post quantum key agreement protocols.	63

1. INTRODUCTION

Elliptic curves were first proposed as a basis for public key cryptography in the mid 1980's. They provide asymmetric cryptographic services based on the difficulty of the elliptic curve discrete logarithm problem, which is so called because of its similarity to the discrete logarithm problem over the integer modulo a large prime. One benefit of elliptic curve cryptosystems is that they can use a much shorter key length than other public key cryptosystems to provide an equivalent level of security.

Key agreement over insecure channels is one of the most important cryptographic services. In a key agreement protocol, two parties exchange some messages, and then, they can share a key. Authenticated key agreement protocols allow parties communicating over an insecure but authenticated channel to establish a secure communication channel. The first modern key agreement protocol is the Diffie-Hellman protocol given by Diffie and Hellman [15] in 1976. Its security is based on the difficulty of solving discrete logarithm problems. As the first practical key agreement protocol without authentication, it is not secure to the man-in-the-middle attack. After that, lots of protocols have been published and some of them use certificates to prevent attacks such as the man-in-the-middle attack. In addition to the need for sound security, the many practical applications of key exchange have driven designers to improve on the performance cost associated with authentication mechanisms. Of these protocols, the most famous, most efficient and most standardized is the MQV protocol of Law, Menezes, Qu, Solinas and Vanstone [33]. After that Hugo Krawczyk designed HMQV [32] which is more secure than MQV in some security claims. In this thesis, we design a new key agreement protocol which is as secure as HMQV.

However, all key agreement protocols and public key algorithms which are based on discrete logarithm problems have been insecure after quantum computer will arrive. Therefore, in 2016, the National Institute of Standards and Technology (NIST) has been started a post-quantum cryptography standardization process taking into account the recent advances in quantum computing and the potential threat of Shor's algorithm [40] on public-key cryptography. The current international PQC competition promoted by NIST and the recent search for new "post-quantum" cryptographic primitives

has motivated a new era of research in the mathematics of cryptography. Ideas for cryptographic primitives based on hard mathematical problems are being actively proposed and examined. This thesis focuses on isogeny-based cryptography, and in particular its possible applications in cryptography and on the hardness of computing endomorphism rings of supersingular elliptic curves.

Recently, there are many isogeny based key agreement protocols which are used ordinary or supersingular elliptic curves have been designed [5], [7], [11], [13], [20]. In this thesis, we analyze some of these protocols, and we find a new weakness for some of them. Also, we define a new and first MQV type isogeny based key agreement protocol and we give the security proof of this protocol. Moreover, we discuss the advantages or disadvantages of our protocol to other isogeny based protocols.

The rest of this thesis is structured as follows: Chapter 2 briefly summarizes the mathematical structures of elliptic curves and security notions in cryptographic protocols. Chapter 3 describes the details of the pre-quantum protocols and includes the details of the protocol we design. In Chapter 4, we give details of the post-quantum isogeny based cryptographic protocols. We then describe the key manipulation attack on a protocol of [10] in Chapter 5. Subsequently, the new isogeny based key agreement protocols are presented in Chapter 6. Finally, in Section 7, we define some new attacks to some isogeny based protocols and in 8 we conclude the thesis.

2. PRELIMINARIES

The main purposes of this chapter are to give some fundamental definitions about elliptic curves, especially isogenies, and to establish some notational conventions which are used throughout this thesis. We introduce the concept of the public key cryptography and key agreement protocols. Furthermore, we explain basic cryptographic definitions to understand how to design and analyze the cryptographic protocols. We refer to the books [41] and [46] for the basic definitions and well-known results.

2.1. Elliptic Curves

To understand main definitions for elliptic curves, it is useful that basic concepts about groups, rings and fields have been reviewed.

Elliptic curves are projective curves of genus one having a specified base point. Projective space initially appeared through the process of adding points at infinity, as a method to understand the geometry of projections. One can define a projective space as the collection of all lines in affine space passing through the origin.

Definition 2.1: The projective space of dimension n , denoted by $\mathbb{P}^n$ or $\mathbb{P}^n(\overline{K})$ is the set of $(n + 1)$ tuples,

$$(x_0, \dots, x_n) \in \overline{K}^{n+1} \quad (2.1)$$

such that $(x_0, \dots, x_n) \neq (0, \dots, 0)$ taken modulo the equivalence relation,

$$(x_0, \dots, x_n) \sim (y_0, \dots, y_n) \quad (2.2)$$

if and only if there exists $\lambda \in \overline{K}$ such that $x_i = \lambda y_i$ for all i .

The equivalence class of a projective point $(x_0, \dots, x_n)$ is customarily denoted by $(x_0 : \dots : x_n)$. The set of the k -rational points, denoted by $\mathbb{P}^n(K)$, is defined as

$$\mathbb{P}^n(K) = \{(x_0 : \dots : x_n) \in \mathbb{P}^n \mid x_i \in K \text{ for all } i\} \quad (2.3)$$

By fixing arbitrarily the coordinate $x_n = 0$, we define a projective space of dimension $n-1$, which we call the space at infinity; its points are called points at infinity.

Definition 2.2: An elliptic curve E over a field K is given by a non-singular projective curve of the form;

$$Y^2Z + a_1XYZ + a_3YZ^2 = X^3 + a_2X^2Z + a_4XZ^2 + a_6Z^3 \quad (2.4)$$

for $a_1, a_2, a_3, a_4, a_6 \in K$ along with a base point $\mathcal{O}_E = [0 : 1 : 0]$ which is referred to as the point at infinity.

Equation 2.4 has the form of *Generalized Weierstrass* equation. This more general form is useful when working with fields of characteristic 2 and 3. In this thesis, we assume that the field K has characteristic different from 2 and 3. Without loss of generality, the elliptic curve equation can be simplified and given in the short Weierstrass form if the characteristic of K is neither 2 or 3.

Definition 2.3: (Weierstrass equation) An elliptic curve defined over a field K is the locus in $\mathbb{P}^2(\overline{K})$ of an equation

$$Y^2Z = X^3 + aXZ^2 + bZ^3 \quad (2.5)$$

with $a, b \in K$ and $4a^3 + 27b^2 \neq 0$ (non-singularity condition).

By defining the coordinates $x = X/Z$ and $y = Y/Z$, we equivalently define the elliptic curve as the locus of the equation with affine coordinates

$$y^2 = x^3 + ax + b \quad (2.6)$$

plus the point at infinity $\mathcal{O} = (0 : 1 : 0)$. To be more precise, for $a, b \in K$, an elliptic curve E defined over a field K is a set such that,

$$E(K) = \{\mathcal{O}\} \cup \{(x, y) \in K \times K \mid y^2 = x^3 + ax + b\} \quad (2.7)$$

Since any elliptic curve is defined by a cubic equation, Bezout's Theorem shows that any line in $\mathbb{P}^2$ intersects the curve in definitely three points, taken with multiplicity. Using this property, addition operation for the points on an elliptic curve can be defined properly.

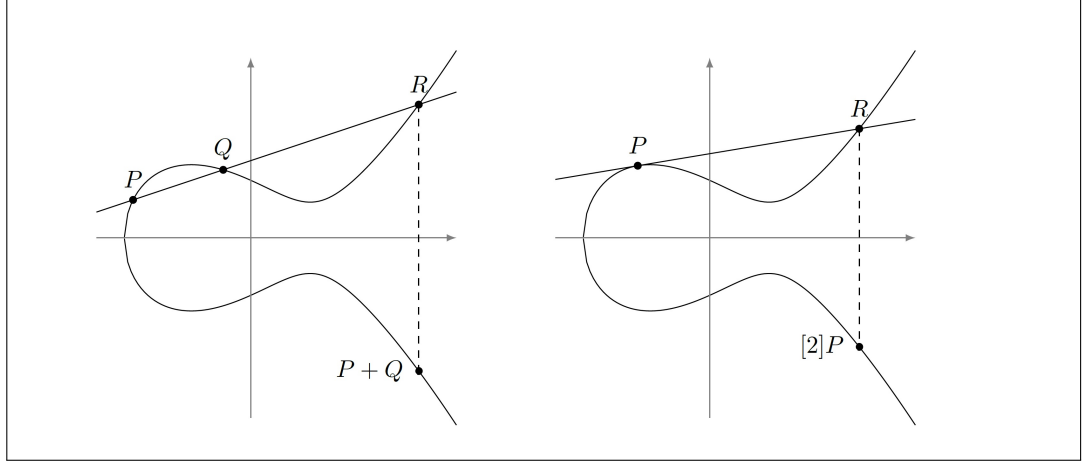


Figure 2.1: The geometric representation of point addition and point doubling.

The addition law by requiring that three co-linear points sum to zero can be defined as follows:

Definition 2.4: (Addition Operation on Elliptic Curves) Let E be an elliptic curve defined by $y^2 = x^3 + ax + b$ and $P = (x_1, y_1)$, $Q = (x_2, y_2)$ be points on E with $P, Q \neq \mathcal{O}$. Define $P + Q = R = (x_3, y_3)$ as follows:

i) If $x_1 \neq x_2$, then

$$x_3 = m^2 - x_1 - x_2 \quad y_3 = m(x_1 - x_3) - y_1 \quad (2.8)$$

where $m = \frac{y_2 - y_1}{x_2 - x_1}$.

ii) If $x_1 = x_2$ but $y_1 \neq y_2$, then $P + Q = \mathcal{O}$.

iii) If $P = Q$ and $y_1 \neq 0$, then

$$x_3 = m^2 - 2x_1 \quad y_3 = m(x_1 - x_3) - y_1 \quad (2.9)$$

where $m = \frac{3x_1^2 + a}{2y_1}$.

iv) If $P = Q$ and $y_1 = 0$, then $P + Q = \mathcal{O}$.

Moreover,

$$P + \mathcal{O} = P \quad (2.10)$$

for all points P on E .

Theorem 2.1: (Group Law) The addition of points on an elliptic curve E satisfies the following properties:

- (closure property) For all points P_1, P_2 on E , $P_1 + P_2$ is also on the curve E .
- (commutativity) $P_1 + P_2 = P_2 + P_1$ for all P_1, P_2 on E .
- (existence of identity) $P + \mathcal{O} = P$ for all points P on E .
- (existence of inverses) Given P on E , there exists P' on E with $P + P' = \mathcal{O}$. The point P' is the inverse of P and it is denoted by $-P$.
- (associativity) $(P_1 + P_2) + P_3 = P_1 + (P_2 + P_3)$ for all P_1, P_2, P_3 on E .

In other words, the points on E form an additive abelian group with $\mathcal{O}$ as the identity element.

Definition 2.5: (Order) The order of an elliptic curve point P is the smallest positive integer k such that $kP = \mathcal{O}$. Also, for an integer n , we have $nP = \mathcal{O}$ if and only if the order of P divides n .

An important result from group theory (a corollary of Lagrange's Theorem) is that the order of a point always divides the order of the group $E(\mathbb{F}_q)$.

Definition 2.6: Let E be an elliptic curve over a field K , and let $P \in E(\overline{K})$. For any $m \in \mathbb{Z}$, define the map,

$$[m] : E(\overline{K}) \rightarrow E(\overline{K}) \quad (2.11)$$

to be the multiplication-by- m map, where

$$[m]P = \underbrace{P + \cdots + P}_{m \text{ times}}. \quad (2.12)$$

The multiplication map can be defined in a similar way for negative integers: $[-m]P = [m](-P)$.

Definition 2.7: (Torsion subgroup) For any $m \in \mathbb{Z}$ and an elliptic curve E over a field K , the subgroup

$$E[m] = \{P \in E(\overline{K}) \mid mP = \mathcal{O}\} \quad (2.13)$$

is called the m -torsion subgroup of $E(\overline{K})$, and an element $P \in E[m]$ is called an m -torsion point.

Proposition 2.1: (Structure of elliptic curve subgroups) Let E be an elliptic curve defined over a field K , and let $m \neq 0$ be an integer. $E[m]$, has the following structure:

- $E[m] \cong \mathbb{Z}/m\mathbb{Z} \oplus \mathbb{Z}/m\mathbb{Z}$, if the characteristic of K does not divide m .
- If p is the characteristic of K , then

$$E[p^i] \cong \begin{cases} \mathbb{Z}/p^i\mathbb{Z} & \text{for any } i \geq 0, \text{ or} \\ \{\mathcal{O}\} & \text{for any } i \geq 0. \end{cases} \quad (2.14)$$

For the elliptic curves, the case $E[p] \cong \mathbb{Z}/p\mathbb{Z}$ is called ordinary elliptic curves, while the case $E[p] \cong \mathcal{O}$ is called supersingular elliptic curves.

All elliptic curves in cryptographic applications have been defined over finite fields. Hence, from now on, we assume that E is an elliptic curve defined over a finite field $\mathbb{F}_q$ with q elements (where $q = p^n$, p is a prime number and n is a positive integer).

Theorem 2.2: (Hasse Theorem) Let E be an elliptic curve over the finite field $\mathbb{F}_q$. Then the order of $E(\mathbb{F}_q)$ satisfies

$$|q + 1 - \#E(\mathbb{F}_q)| \leq 2\sqrt{q} \quad (2.15)$$

Definition 2.8: (Weil Pairing). Let E be an elliptic curve defined over a field K . Let m be a positive integer such that $\text{char}(K)$ does not divide m . Let μ_m be the group of m -th roots of unity in $\overline{K}$ such that

$$\{\mu_m = x \in \overline{K} \mid x^m = 1\} \quad (2.16)$$

Then, there is a pairing

$$e_m : E[m] \times E[m] \rightarrow \mu_m \quad (2.17)$$

called the Weil pairing, such that it satisfies the following properties:

- For all $P, Q, S \in E[m]$, e_m is bilinear in each variable.

$$e_m(P + Q, S) = e_m(P, S) + e_m(Q, S) \quad (2.18)$$

$$e_m(P, Q + S) = e_m(P, Q) + e_m(P, S)$$

- If $e_m(P, Q) = 1$, for all $Q \in E[m]$, then $P = \mathcal{O}_E$ and also if $e_m(P, Q) = 1$, for all $P \in E[m]$, then $Q = \mathcal{O}_E$. Therefore e_m is non-degenerate in each variable.
- $e_m(P, P) = 1$ for all $P \in E[m]$.
- $e_m(P, Q) = e_m(Q, P)^{-1}$, for all $P, Q \in E[m]$.
- $e_m(\sigma P, \sigma Q) = \sigma(e_m(P, Q))$ for all automorphisms σ of $\overline{K}$ such that σ is the identity map on the coefficients of E .
- $e_m(\alpha(P), \alpha(Q)) = e_m(P, Q)^{\deg(\alpha)}$ for all separable endomorphisms α of E .

The elliptic curve discrete logarithm problem is an essential problem in cryptography. In general it is a very complex problem; the best known solving algorithms all have exponential running time. However, for supersingular elliptic curves there exists a sub-exponential solving algorithm called the MOV attack [36]. The MOV attack reduces an elliptic curve discrete logarithm to a logarithm over a finite field using the Weil Pairing. The discrete logarithm problem in a finite field can be solved efficiently using Index Calculus. Therefore, ordinary elliptic curves are more secure than supersingular curves for the discrete logarithm problem.

Definition 2.9: (Morphism). A rational map that is defined everywhere is called a morphism. For projective curves, a morphism $\phi : E \rightarrow E'$ is defined by a polynomial mapping such that;

$$\phi : (X: Y: Z) \rightarrow (\phi_0(X: Y: Z) : \phi_1(X: Y: Z) : \phi_2(X: Y: Z)) \quad (2.19)$$

where the ϕ_i 's are homogeneous polynomials of equal degree satisfying the defining equation of E' .

Definition 2.10: (Homomorphism). A homomorphism $\phi : E \rightarrow E'$ over a field K is a morphism of elliptic curves, such that

$$\phi(P + Q) = \phi(P) + \phi(Q) \quad (2.20)$$

for all $P, Q \in E(K)$. That is, ϕ respects the group law and structure of the curve.

Definition 2.11: (Isomorphism). An isomorphism of elliptic curves $\phi : E \rightarrow E'$ is an isomorphism over $\overline{K}$ of algebraic varieties such that $\phi(\mathcal{O}_E) = \mathcal{O}_{E'}$. If there is an isomorphism from E to E' , then we write $E \cong E'$.

Isomorphism classes are labeled by an invariant called the *j-invariant*, whose origins can be traced back to complex analysis.

Definition 2.12: (j-invariant). Let $E : y^2 = x^3 + ax + b$ be an elliptic curve defined over a field K , define

$$j(E) = 1728 \frac{4a^3}{4a^3 + 27b^2} \in K \quad (2.21)$$

to be the *j-invariant* of E .

Proposition 2.2: ([41], Chapter 3.1). Two curves are isomorphic over the algebraic closure $\overline{K}$ if and only if they have the same *j-invariant*.

Definition 2.13: (Isogeny) Let E and E' be elliptic curves defined over a finite field $\mathbb{F}_q$. An isogeny $\phi : E \rightarrow E'$ defined over $\mathbb{F}_q$ is a non-constant rational map defined over $\mathbb{F}_q$ which is also a group homomorphism from E to E' [41].

Definition 2.14: (Kernel) The kernel of an isogeny ϕ is defined as,

$$\ker(\phi) = \{P | P \in E(\overline{K}) \text{ and } P \in \phi^{-1}(\mathcal{O}_{E'})\} \quad (2.22)$$

The degree of an isogeny is its degree as a rational map. For separable isogenies, to have degree l means to have kernel of size l . Every isogeny of degree greater than one can be factored into a composition of isogenies of prime degree over $\overline{\mathbb{F}}_q$ [9].

Definition 2.15: (Dual Isogeny) Every isogeny $\phi : E \rightarrow E'$ with degree d has a unique dual isogeny $\widehat{\phi} : E' \rightarrow E$ of the same degree such that $\phi \circ \widehat{\phi} : E \rightarrow E$ is the multiplication map $P \rightarrow [d]P$.

Definition 2.16: (Endomorphism Ring) An endomorphism of an elliptic curve E defined over $\mathbb{F}_q$ is an isogeny $E \rightarrow E$ defined over $\mathbb{F}_{q^m}$ for some m . The set of endomorphisms of E together with the zero map forms a ring under the operations of pointwise addition and composition; this ring is called the endomorphism ring of E and denoted by $\text{End}(E)$.

Definition 2.17: (Order). Let K be a finitely generated $\mathbb{Q}$ -algebra. An order $\mathcal{O} \subset K$ is a sub-ring of K that is a finitely generated $\mathbb{Z}$ -module of maximal dimension. The prototypical example of order is the ring of integers $\mathcal{O}_K$ of a number field K , i.e., the ring of all elements of K such that their monic minimal polynomial has coefficients in $\mathbb{Z}$. It turns out that $\mathcal{O}_K$ is the maximal order of K , i.e., it contains any other order of K .

Definition 2.18: (Quaternion algebra). A quaternion algebra is an algebra of the form

$$K = \mathbb{Q} + \alpha\mathbb{Q} + \beta\mathbb{Q} + \alpha\beta\mathbb{Q} \quad (2.23)$$

where the generators satisfy the relations

$$\alpha^2, \beta^2 \in \mathbb{Q}, \quad \alpha^2 < 0, \quad \beta^2 < 0, \quad \beta\alpha = -\alpha\beta \quad (2.24)$$

Theorem 2.3: (Deuring). Let E be an elliptic curve defined over a field K of characteristic p . The $\text{End}(E)$ is isomorphic to one of the following:

- $\mathbb{Z}$, only if $p = 0$.

- An order $\mathcal{O}$ in a quadratic imaginary field (a number field of the form $\mathcal{O}[\sqrt{-D}]$ for some $D > 0$); in this case we say that E is ordinary with complex multiplication by $\mathcal{O}$.
- Only if $p > 0$, a maximal order in the quaternion algebra ramified at p and ∞ ; in this case we say that E is supersingular.

The $\text{End}(E)$ is isomorphic either to an order in a quaternion algebra or to an order in an imaginary quadratic field [41]. In the first case E is called *supersingular* and in the second case E is called *ordinary*.

Two elliptic curves E_1 and E_2 defined over $\mathbb{F}_q$ are said to be isogenous over $\mathbb{F}_q$ if there exists an isogeny $\phi : E_1 \rightarrow E_2$ defined over $\mathbb{F}_q$. A theorem of Tate states that two curves E_1 and E_2 are isogenous over $\mathbb{F}_q$ if and only if $\#E_1(\mathbb{F}_q) = \#E_2(\mathbb{F}_q)$ [43]. Since every isogeny has a dual isogeny, the property of being isogenous over $\mathbb{F}_q$ is an equivalence relation on the finite set of $\overline{\mathbb{F}}_q$ -isomorphism classes of elliptic curves defined over $\mathbb{F}_q$. Accordingly, an isogeny class is defined to be an equivalence class of elliptic curves, taken up to $\overline{\mathbb{F}}_q$ -isomorphism, under this equivalence relation.

Definition 2.19: (Velu's Formula). Let R be a finite subgroup of odd order ℓ of an elliptic curve E given in short Weierstrass form. Vélu [45] showed how to explicitly find the rational function form of a normalized separable isogeny $\phi : E \rightarrow E/R$ with kernel R . For $P = (x_P, y_P) \notin R$, let

$$\phi(P) = \left(x_P + \sum_{Q \in R \setminus \{\mathcal{O}\}} (x_{P+Q} - x_Q), y_P + \sum_{Q \in R \setminus \{\mathcal{O}\}} (y_{P+Q} - y_Q) \right) \quad (2.25)$$

and for any point $P \in R$, set $\phi(P) \in \mathcal{O}_{E/R}$. Moreover, if $E : y^2 = x^3 + ax + b = f(x)$ then E/R is,

$$y^2 = x^3 + (a - 5t)x + b - 7w \quad (2.26)$$

where,

$$t = \sum_{Q \in R \setminus \{\mathcal{O}\}} f'(Q) \quad \text{and} \quad w = \sum_{Q \in R \setminus \{\mathcal{O}\}} x_Q f'(Q) \quad (2.27)$$

2.2. Public Key Cryptography

Public key cryptography is a modern form of cryptography that allows different parties to exchange information securely over an insecure network, without having first to agree upon some secret key. The main use of public key cryptography is to provide information security in computer science, for example to transfer securely email, credit card details or other secret information between sender and recipient via the internet.

In 1976, Diffie and Hellman proposed the use of certain one-way functions called trap-door functions, to make it almost impossible to decipher encrypted data without a key [15]. The idea is that for each key k , M is the set of messages and C is the set of ciphertext, one can choose an encryption function $f_k : M \rightarrow C$ such that it is almost impossible to compute f_k^{-1} without knowing k , in polynomial time. They used exponentiation as a trap-door function in order to exchange secrets and agree on a key. The Diffie-Hellman key agreement protocol was the first practical method for establishing a shared secret over an insecure communication channel. The point is to agree on a key that two parties can use for a symmetric encryption, in such a way that an eavesdropper cannot obtain the key.

Assume G is a cyclic group of prime order p and g is a generator of G . Given that $g' = g^x$ where $x \in [1, p - 1]$, the *discrete logarithm problem* grounds the difficulty of finding x . Diffie-Hellman key agreement protocol based on discrete logarithm problem is defined as follows:

Definition 2.20: [2] Suppose that A and B want to share a secret. A has a secret $x \in_R [1, p - 1]$ and B has a secret $y \in_R [1, p - 1]$ ($x \in_R S$ means that x is chosen uniformly at random from the set S).

- i) A computes $X = g^x$ and sends to B
- ii) B computes $Y = g^y$ and sends to A
- iii) A computes $Z = Y^x = g^{xy}$
- iv) B computes $Z = X^y = g^{xy}$

Hence, A and B agreed on a secret Z .

Since the introduction of public-key cryptography by Diffie and Hellman [15] in 1976, the cryptographic importance of the apparent intractability of the discrete

Table 2.1: Security levels of algorithms (bits).

Symmetric Algorithms	Diffie-Hellman	Elliptic Curve Diffie-Hellman
80	1024	160-223
112	2048	224-255
128	3072	256-383
192	7680	384-511
256	15360	512

logarithm problem has been recognized. ElGamal [24] first described how this problem may be utilized in public-key encryption and digital signature schemes. ElGamal's methods have been refined and incorporated into various protocols to meet a variety of applications, and one of its extensions forms the basis for the U.S. government digital signature algorithm (DSA) [29].

In 1985, Koblitz [30] and Miller [37] independently proposed using the group of points on an elliptic curve defined over a finite field in asymmetric cryptosystems. The primary advantage that elliptic curve systems have over the systems based on the multiplicative group of a finite field (and also over systems based on the intractability of integer factorization) is the absence of a sub-exponential-time algorithm (such as those of "index-calculus" type) that could find discrete logs in these groups. Consequently, one can use an elliptic curve group that is smaller in size while maintaining the same level of security (Table 1). The result is smaller key sizes, bandwidth savings, and faster implementations, features which are especially attractive for security applications where computational power and integrated circuit space is limited, such as smart cards, PC (personal computer) cards, and wireless devices.

2.3. Cryptographic Protocols

Cryptographic protocols are communication protocols which are designed to provide security assurances of various kinds, using cryptographic mechanisms. A protocol consists of a set of rules (conventions) which determine the exchange of messages between two or more participants. Cryptographic protocols, also called security protocols, use cryptographic primitives in communication protocols

to provide information security, such as confidentiality, authentication, integrity or non-repudiation, in an insecure network. Encryption schemes, digital signatures, hash functions, and random number generations are among the cryptographic primitives which may be utilized to build cryptographic protocols.

A cryptographic protocol is a distributed algorithm defined by a sequence of steps precisely specifying the actions required of two or more entities to achieve a specific security objective. Cryptographic protocols involving message exchanges require precise definition of both the messages to be exchanged and the actions to be taken by each party. Cryptographic protocols may be typically divided into four main categories, depending on the security objectives of the cryptographic protocol:

- *Authentication Protocol*: It is a protocol to provide one party some degree of assurance regarding the identity of another with whom it is purportedly communicating.
- *Key Establishment Protocol*: It is a protocol to establish shared secrets, which are typically called or used to derive session keys.
- *Electronic Commerce Protocol*: It is a protocol to provide secure electronic trades over network for two (or more) parties.
- *Secure Multi-party Protocol*: It is a protocol to assure secure collaborated run of computation for any parties of the protocol run in a distributed system.

Authenticated key establishment protocols are designed to provide two or more specified entities communicating over an open network with a shared secret key which may subsequently be used to achieve some cryptographic goal such as confidentiality or data integrity. Secure authenticated key establishment protocols are important as effective replacements for traditional key establishment achieved using expensive and inefficient messengers.

Fundamental security goals: Let A and B be two honest entities, i.e., legitimate entities who execute the steps of a protocol correctly.

- (IKA) *Implicit Key Authentication*: A key agreement protocol is said to provide implicit key authentication of B to A if party A is assured that no other party aside from a specifically identified second party B can possibly learn the value of a particular secret key. Note that the property of implicit key authentication does not necessarily mean that A is assured of B actually possessing the key.

- (EKA) *Explicit Key Authentication*: A key agreement protocol is said to provide explicit key confirmation of B to A if party A is assured that the second party B has actually computed the agreed key.

A key agreement protocol which provides implicit key authentication to both participating parties is called an *authenticated key agreement* (AK) protocol, while one providing explicit key authentication to both participating parties is called an *authenticated key agreement with key confirmation* (AKC) protocol.

A number of other desirable security attributes have also been identified. Typically the importance of supplying these attributes will depend on the application. In the following, A and B are two honest parties.

- (K-KS) *Known-Key Security*: Each run of a key agreement protocol between A and B should produce a unique secret key; such keys are called session keys. Session keys are desirable in order to limit the amount of data available for cryptanalytic attack and to limit exposure in the event of session key compromise. A protocol should still achieve its goal in the face of an adversary who has learned some other session keys.
- (FS) *Forward Secrecy*: If long-term private keys of one or more entities are compromised, the secrecy of previous session keys established by honest parties is not affected.
- (K-CI) *Key-Compromise Impersonation*: Suppose A 's long-term private key is disclosed. Clearly an adversary that knows this value can now impersonate A , since it is precisely this value that identifies A . However, it is desirable in some circumstances that this loss does not enable the adversary to impersonate other parties to A .
- (UK-S) *Unknown Key-Share*: Entity B cannot be coerced into sharing a key with entity A without B 's knowledge, i.e., when B believes the key is shared with some entity $C \neq A$, and A (correctly) believes the key is shared with B . A hypothetical scenario where an unknown key-share attack can have damaging consequences, this scenario was first described in [16].

For the performance of protocols, the intended properties are as follows;

- Minimal number of passes (the number of messages exchanged).

- Low communication overhead (total number of bits transmitted).
- Low computation overhead (total number of arithmetical operations required).
- Possibility of pre-computation (to minimize on-line computational overhead).

In the literature, many key agreement protocols have been designed by now. For these protocols there are two types of key, namely, static and ephemeral keys. In this thesis, pre-quantum key agreement protocols have been analyzed. Analyses of protocols were conducted under two main attack scenarios. These are passive attacks, where messages only are gathered but not altered, and active attacks, where there are some modification of the messages. There are quite a lot of attack methods under these two categories. The attack scenarios that can be emphasized in the thesis as follows:

- *Eavesdropping*: an adversary captures information that is sent in the protocol.
- *Modification*: the information sent in the protocol is altered by an adversary.
- *Replay*: a transmission is recorded, and then later retransmitted.
- *Reflection*: sending the same message back to the sender in a new protocol run.
- *Denial of service attacks*: the attacker overwhelms the server so that legitimate users will not get a connection with the server.
- *Typing attack*: replacing a message field of one type with a message field of another type.
- *Cryptanalysis*: the study of methods for obtaining the meaning of encrypted information.
- *Certificate manipulation*: modification of the certificate information.
- *Protocol interaction*: using a new protocol to interact with a known protocol.

2.4. Security Definitions

Canetti-Krawczyk (CK) Model [3] is one of the most famous security models in the literature to prove the security of authenticated key exchange protocols. This security model is strong enough to get a wide range of security properties, including basic session key security, and yet is flexible enough to cover additional requirements such as perfect forward secrecy (PFS) and key-compromise impersonation (KCI) resilience. The key exchange constructions studied in this thesis use the CK model in security proofs.

We focus on the main approaches to authenticating a key exchange: using implicitly authenticated schemes (such as in (H)MQV [32], [33]). The security proofs follow the original CK model (referred to as CK model in the pre-specified peer setting). Protocols proven secure under this model are called session key (SK) secure in the unauthenticated links model [3]. This model is appropriately extended in several follow-up works to capture forward secrecy and KCI resilience (e.g., see [32]). Also, Canetti and Krawczyk in [4] relaxed the security definition in the model above to support the “postspecified peers” setting. Protocols proven secure under this modified model are called session-key (SK) secure in the post-specified peer model [4].

Definition 2.21: (Supersingular Isogeny Computational Diffie-Hellman Problem (SI-CDH))

Let $\phi_A : E \rightarrow E_A$ be an isogeny whose kernel is equal to $\langle P_A + [n_A]Q_A \rangle$, and let $\phi_B : E \rightarrow E_B$ be an isogeny whose kernel is equal to $\langle P_B + [n_B]Q_B \rangle$. Given the curves E_A, E_B and the points $\phi_A(P_B), \phi_A(Q_B), \phi_B(P_A), \phi_B(Q_A)$, find the j -invariant of $E/\langle P_A + [n_A]Q_A, P_B + [n_B]Q_B \rangle$.

Definition 2.22: (Supersingular Isogeny Decision Diffie-Hellman Problem (SI-DDH))

Given a tuple sampled with probability $1/2$ from one of the following two distributions:

i) $(E_A, E_B, \phi_A(P_B), \phi_A(Q_B), \phi_B(P_A), \phi_B(Q_A), E_{AB}),$

where $E_A, E_B, \phi_A(P_B), \phi_A(Q_B), \phi_B(P_A)$ and $\phi_B(Q_A)$ are as in the SI-CDH problem and

$$E_{AB} \cong E/\langle P_A + [n_A]Q_A, P_B + [n_B]Q_B \rangle \quad (2.28)$$

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

where $E_A, E_B, \phi_A(P_B), \phi_A(Q_B), \phi_B(P_A)$ and $\phi_B(Q_A)$ are as in the SI-CDH problem and

$$E_C \cong E/\langle P_A + [n'_A]Q_A, P_B + [n'_B]Q_B \rangle \quad (2.29)$$

where n'_A, n'_B are chosen at random from $\mathbb{Z}/l_A^e \mathbb{Z}, \mathbb{Z}/l_B^e \mathbb{Z}$ and not both divisible by l_A, l_B respectively,

determine from which distribution the tuple is sampled.

Definition 2.23: (Supersingular Isogeny Gap Diffie-Hellman Problem (SI-GDH))

Supersingular Isogeny Gap DH Problem is a problem that there is a polynomial time algorithm to solve the SI-DDH problem and there is no polynomial time algorithm to solve the SI-CDH problem.

Definition 2.24: (Degree Insensitive Supersingular Isogeny Gap DH Problem (di-SI-GDH))

This problem is also a type of SI-GDH problem such that the degrees of ϕ'_A and ϕ'_B are the powers of the degrees of ϕ_A and ϕ_B , respectively.

Definition 2.25: A key exchange protocol is called SK-secure with PFS if the following properties hold for any attacker:

- i) Two uncorrupted parties complete matching sessions and derive the same session key,*
- ii) The probability that an attacker succeeds in its test-session distinguishing attack is not more than $1/2$ plus some negligible function (in the security parameter).*

3. PRE-QUANTUM CRYPTOGRAPHIC PROTOCOLS

In this section, we give details of pre-quantum cryptographic protocols, especially key establishment protocols. We focus on asymmetric authenticated key agreement protocols whose security is based on the difficulty of the Diffie-Hellman problem. We discuss the practical and provable security aspects of some protocols which are being standardized by accredited standards organizations. As a result of the analyzes, important protocols can be listed as follows:

- Ephemeral Diffie-Hellman [15]
- Static Diffie-Hellman [15]
- MTI-C0 [33]
- MTI-A0 [33]
- Full Unified Model [1]
- MQV [33]
- HMQV [32]

As we have seen in the analysis, six of the seven analyzed protocols fail to meet some of the security features described in this section. Only one protocol, namely HMQV, provides the desired security level. Ephemeral Diffie-Hellman, MTI-C0 and MTI-A0 provide neither implicit key authentication nor explicit key authentication. Unified model and MQV ensure not explicit key authentication but implicit key authentication.

Two of the protocols fail to meet the forward secrecy goal, and three of the protocols fail to meet the key compromise impersonation resilience goal. Problems in protocols are caused by disclosure of the long-term private key. If we take the necessary steps in order to keep this key secret, the encountered problems will not have any impact on the security. Moreover, four of the protocols were found to be vulnerable to unknown key share attack. All results are summarized in Table 3.1. The result of the research showed that the compatibility between efficiency and security level cannot be ignored when designing the key agreement protocol.

Table 3.1: Security services offered by authenticated key agreement protocols.

Scheme	IKA	EKA	K-KS	FS	K-CI	UK-S
Ephemeral Diffie Hellman	x	x	✓	n/a	n/a	x
MTI - C0	x	x	✓	✓	x	x
MTI - A0	x	x	✓	x	✓	x
Static Diffie Hellman	✓	x	x	x	x	✓
Unified Model	✓	x	✓	✓	x	✓
MQV	✓	x	✓	✓	✓	x
HMQV	✓	✓	✓	✓	✓	✓
The New Key Agreement Protocol	✓	✓	✓	✓	✓	✓

We give some extra notation for the group of points on elliptic curve over a finite field to understand protocols details precisely.

- G is a base point of E which has an order n ,
- a, b are static private keys of *Alice* and *Bob*; $a, b \in_R [1, n - 1]$,
- A, B are static public keys of *Alice* and *Bob*; $A = aG, B = bG$,
- x, y are ephemeral private keys of *Alice* and *Bob*; $x, y \in_R [1, n - 1]$,
- X, Y are ephemeral public keys of *Alice* and *Bob*; $X = xG, Y = yG$,
- $\overline{Q}$ denotes most significant k bits of x coordinate of point $Q = (x_q, y_q)$ where $Q \in E$ and x_q is a $2k$ bits number,
- H is a cryptographic hash function.

3.1. Full Unified Diffie-Hellman

The Unified Model [1], proposed by Ankney, Johnson and Matyas, is an authenticated key agreement protocol. One of its advantages is that it is conceptually simple and consequently easier to analyze. $Cert(Alice)$ and $Cert(Bob)$ represent the digital certificate for Alice and Bob, respectively. Their static public keys are in the certificate. x and y are the ephemeral secret keys while X and Y represents their ephemeral public keys.

- Alice* sends $Cert_A$ and X to *Bob*,
- Bob* sends $Cert_B$ and Y to *Alice*,
- Alice* verifies that $Y \in E, Y \neq \mathcal{O}$ and $nY = \mathcal{O}$,

- iv) *Bob* verifies that $X \in E$, $X \neq \mathcal{O}$ and $nX = \mathcal{O}$,
- v) If any check fails, then *Alice* terminates the protocol run with failure. Otherwise, *Alice* computes the session key $K = H(a(B)||x(Y)) = H(abG||xyG)$,
- vi) If any check fails, then *Bob* terminates the protocol run with failure. Otherwise, *Bob* computes the session key $K = H(b(A)||y(X)) = H(abG||xyG)$.

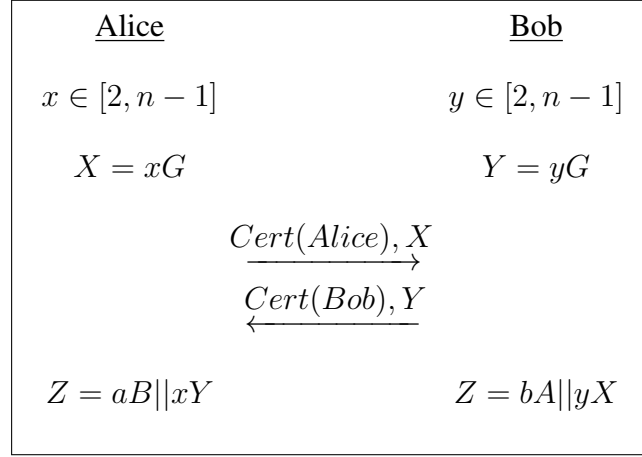


Figure 3.1: Full Unified Diffie-Hellman.

3.2. MTI-C0 and MTI-A0 Key Agreement Protocols

The MTI-C0 and MTI-A0 key agreement protocols are special cases of the three infinite families of key agreement protocols proposed by Matsumoto, Takashima and Imai [35] in 1986. They were designed to provide implicit key authentication, and do not provide key confirmation. Closely related protocols are KEA [26]. MTI-C0 and MTI-A0 families of protocols are vulnerable to several active attacks if not implemented correctly.

The drawbacks of the ephemeral and static Diffie-Hellman protocols can be alleviated by using both static and ephemeral keying material in the formation of shared secrets. An example of an early protocol designed in this manner is the MTI-C0 protocol.

- i) *Alice* selects $x \in_R [1, n - 1]$ and sends $T_A = xB$ to *Bob*,
- ii) *Bob* selects $y \in_R [1, n - 1]$ and sends $T_B = yA$ to *Alice*,
- iii) *Alice* computes $Z = a^{-1}xT_B = xyG$.
- iv) *Bob* computes $Z = b^{-1}yT_A = xyG$.

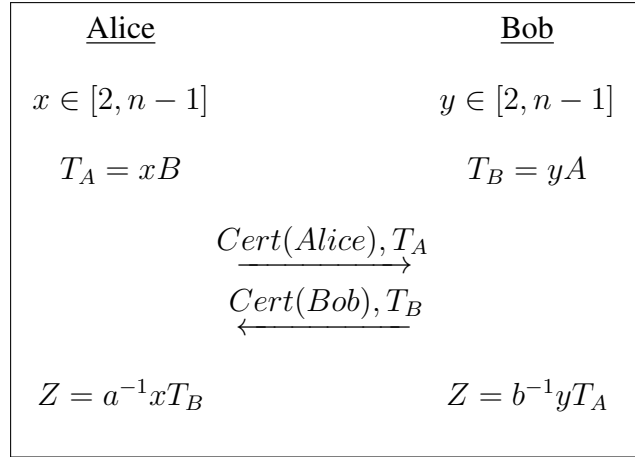


Figure 3.2: MTI-C0 Protocol.

At first glance, this protocol looks like a secure one. Unfortunately, some subtle problems have raised because of combining static and ephemeral Diffie-Hellman protocols. As an example, consider the following instance of the small subgroup attack [33] on the MTI-C0 protocol. An adversary *Eve* replaces T_A and T_B with the identity element. Both *Alice* and *Bob* now form Z , which is also known to *Eve*. This attack demonstrates that the MTI-C0 protocol does not offer implicit key authentication.

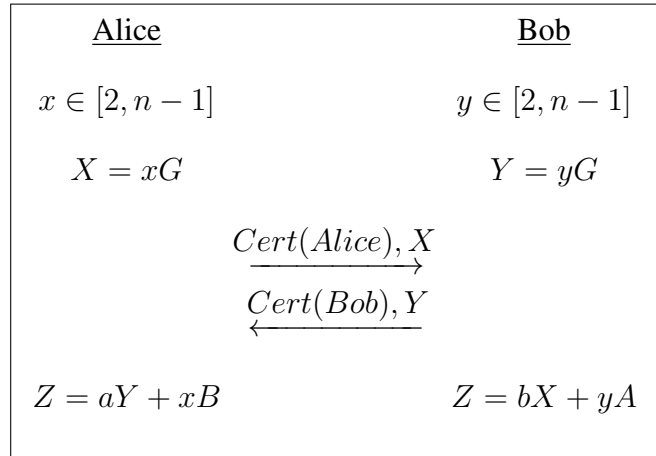


Figure 3.3: MTI-A0 Protocol.

The small subgroup attack cannot be problem for MTI-A0 protocol, but MTI-A0 protocol does not possess the unknown key-share resilience attribute in some circumstances. The active attack and the many variants of it, illustrate that it is prudent to require an entity A to prove possession of the private key corresponding to its static public key to a CA before the CA certifies the public key as belonging to A . This proof of possession can be accomplished by zero-knowledge techniques.

- i) *Alice* sends $Cert_A$ and X to *Bob*,
- ii) *Bob* sends $Cert_B$ and Y to *Alice*,
- iii) *Alice* computes $Z = aY + xB = (ay + xb)G$,
- iv) *Bob* computes $Z = bX + yA = (ay + xb)G$.

3.3. MQV and HMQV Key Agreement Protocols

The MQV protocol of Law, Menezes, Qu, Solinas and Vanstone is possibly the most efficient of all known authenticated Diffie-Hellman protocols that use public-key authentication. In addition to great performance, the protocol has been designed to achieve a remarkable list of security properties. In order to provide an answer to whether the protocol can be proven secure in a rigorous model of key-exchange security, Krawczyk [32] analyze the MQV protocol in the Canetti-Krawczyk model of key exchange. He then showed that MQV fails to a variety of attacks in this model that invalidate its basic security as well as many of its stated security goals. On the basis of these findings, Krawczyk presented HMQV, a carefully designed variant of MQV, that provides the same superb performance and functionality of the original protocol but for which all the MQV's security goals can be formally proved to hold in the random oracle model under the computational Diffie-Hellman assumption. Basic model for MQV and HMQV is Protocol-1, but there are some changes in that protocol.

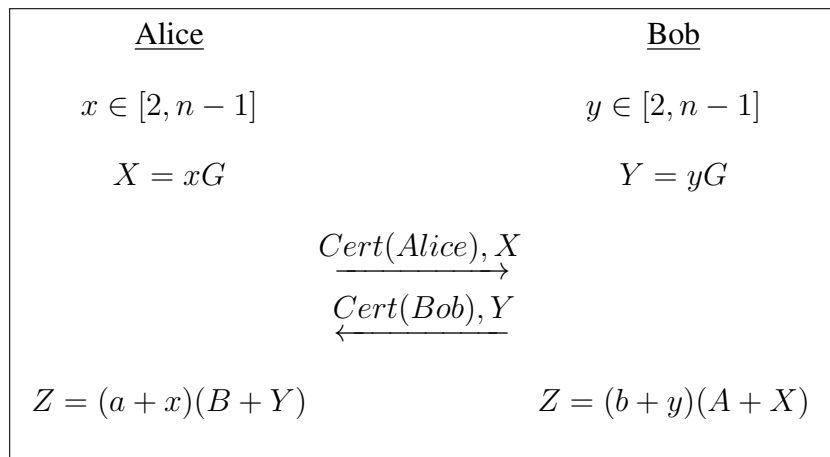


Figure 3.4: Protocol-1.

We can summarize *MQV* protocol as follows:

- i) *Alice* sends $Cert_A$ and X to *Bob*,
- ii) *Bob* sends $Cert_B$ and Y to *Alice*,
- iii) *Alice* verifies that $Y \in E$, $Y \neq \mathcal{O}$ and $nY = \mathcal{O}$,
- iv) *Bob* verifies that $X \in E$, $X \neq \mathcal{O}$ and $nX = \mathcal{O}$,
- v) If any check fails, then *Alice* terminates the protocol run with failure. Otherwise, *Alice* computes $s_A = (x + a\bar{X}) \bmod n$ and the shared secret $Z = s_A(Y + \bar{Y}B) = (x + a\bar{X})(y + b\bar{B})G$,
- vi) If any check fails, then *Bob* terminates the protocol run with failure. Otherwise, *Bob* computes $s_B = (y + b\bar{Y}) \bmod n$ and the shared secret $Z = s_B(X + \bar{X}A) = (x + a\bar{X})(y + b\bar{B})G$,
- vii) If $Z = \mathcal{O}$ then the protocol is terminated. Otherwise, the session key is $K = H(Z)$.

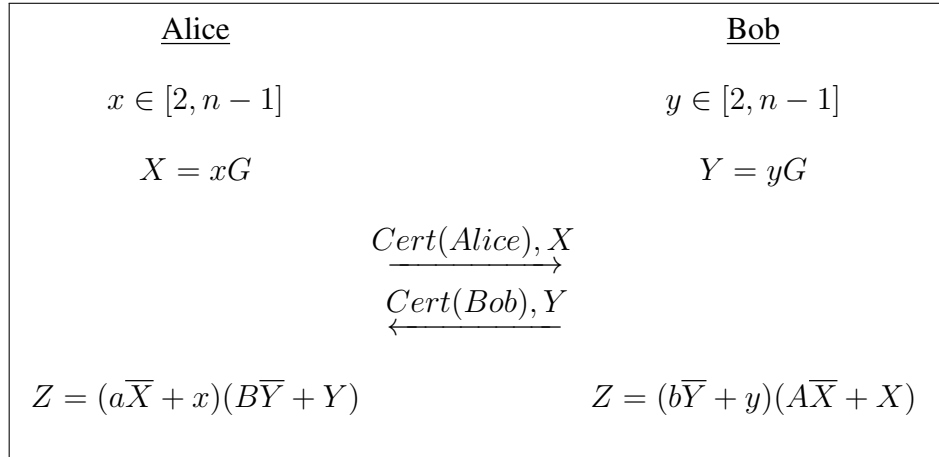


Figure 3.5: MQV Protocol.

The only difference from *MQV* protocol is the calculation of s_A and s_B . Moreover, $s_A = H(X, B)$ and $s_B = H(Y, A)$ in the protocol.

- i) *Alice* sends $Cert_A$ and X to *Bob*,
- ii) *Bob* sends $Cert_B$ and Y to *Alice*,
- iii) *Alice* verifies that $Y \in E$, $Y \neq \mathcal{O}$ and $nY = \mathcal{O}$,
- iv) *Bob* verifies that $X \in E$, $X \neq \mathcal{O}$ and $nX = \mathcal{O}$,
- v) If any check fails, then *Alice* terminates the protocol run with failure. Otherwise, *Alice* computes the shared secret $Z = (as_A + x)(Bs_B + Y) = (x + as_A)(y + bs_B)G$,

- vi) If any check fails, then *Bob* terminates the protocol run with failure. Otherwise, *Bob* computes the shared secret $Z = (bs_B + y)(As_A + X) = (x + as_A)(y + bs_B)G$,
- vii) If $Z = \mathcal{O}$ then the protocol is terminated. Otherwise, the session key is $K = H(Z)$.

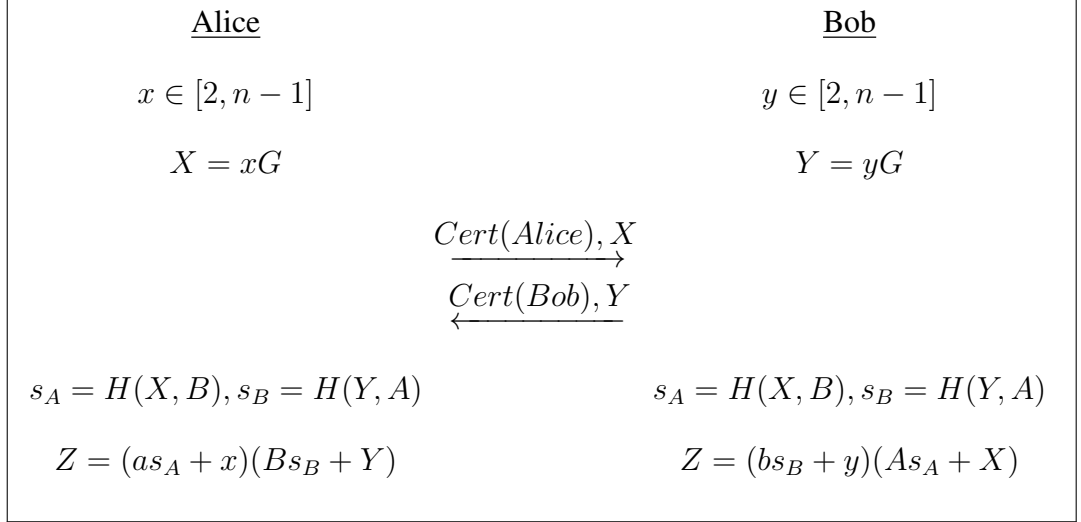


Figure 3.6: H-MQV Protocol.

3.4. The New Key Agreement Protocol

In this thesis, we define a new key agreement protocol which has the same security level with H-MQV. Therefore, it is more secure than the MQV protocol. However, in our protocol, there are two elliptic curve point multiplication for one party while in H-MQV and MQV protocol, there are 1.5 elliptic curve point multiplications. The details of our protocol are in Figure 3.7.

- i) *Alice* sends Cert_A and X to *Bob*,
- ii) *Bob* sends Cert_B and Y to *Alice*,
- iii) *Alice* verifies that $Y \in E, Y \neq \mathcal{O}$ and $nY = \mathcal{O}$,
- iv) *Bob* verifies that $X \in E, X \neq \mathcal{O}$ and $nX = \mathcal{O}$,
- v) If any check fails, then *Alice* terminates the protocol run with failure. Otherwise, *Alice* computes the shared secret $Z = (a + x)(B + Y) || (a - x)(B + Y)$,
- vi) If any check fails, then *Bob* terminates the protocol run with failure. Otherwise, *Bob* computes the shared secret $Z = (b + y)(A + X) || (b + y)(A - X)$,

- vii) If $Z = \mathcal{O}$ then the protocol is terminated. Otherwise, the session key is $K = H(Z)$.

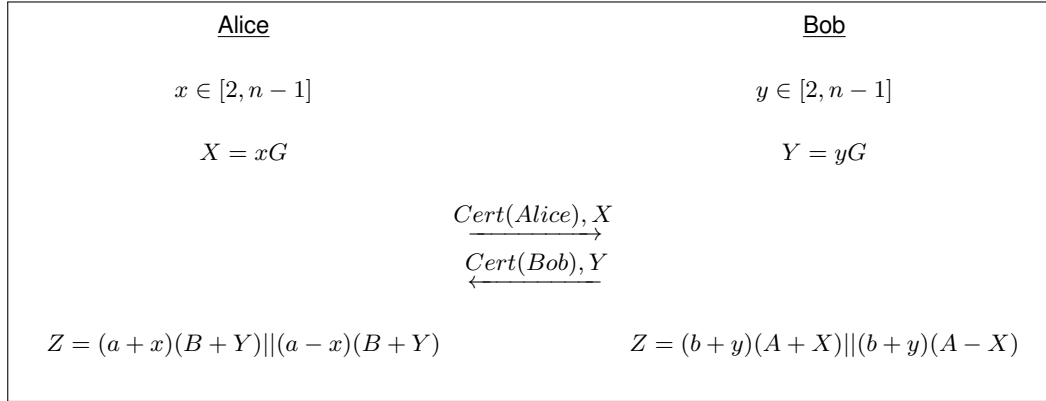


Figure 3.7: The New Key Agreement Protocol.

4. POST-QUANTUM CRYPTOGRAPHIC PROTOCOLS

Research on post-quantum cryptographic algorithms and protocols has gained recently an increasing attention. The main reason is the prediction that Shor's algorithm [40] can be efficiently implemented on large scale quantum computers which would break the security guarantees of the existing public-key cryptosystems, which are typically based on the intractability of the factorization and the discrete logarithm problems on suitable abelian groups. There are some alternative techniques based on various known mathematical problems which is believed to be secure even against a quantum adversary. One of these problems is to find the isogeny between elliptic curves, also known as isogeny-based cryptography. In other words, isogeny-based cryptography refers to cryptographic algorithms and protocols based on elliptic curves, whose intractability relies basically on the problem of finding an explicit isogeny between two given isogenous elliptic curves over a finite field $\mathbb{F}_q$. The first proposal of an isogeny-based cryptosystem was made by Couveignes in 1997 [9]. It described a non-interactive key exchange protocol based on the isogenies of ordinary elliptic curves over $\mathbb{F}_q$. His work was not published as an article and solely posted on ePrint in 2006. This method was also independently rediscovered by Rostovtsev and Stolbunov earlier in 2004 [39] and published in 2006. In 2010, Childs, Jao and Soukharev [6] showed that the Couveignes–Rostovtsev–Stolbunov(CRS) scheme can be broken by a quantum adversary in sub-exponential time using the commutativity of the endomorphism rings of the underlying elliptic curves. This led Jao and De Feo [17] to consider the use of supersingular elliptic curves. The full ring of endomorphism of a supersingular elliptic curve is a maximal order in a quaternion algebra; hence the attack cannot be used to break the supersingular version of the non-interactive key exchange protocol based on the isogenies due to non-commutativity of the endomorphism ring. Hence, the protocol of Jao and De Feo [17] provides security against quantum adversaries, which is known as the Supersingular Isogeny Diffie-Hellman (SIDH) key exchange protocol. Since then, research in the area of isogenies has become highly popular and some new schemes based on SIDH have recently been presented [20], [21],

[34]. Note also that, SIKE (Supersingular Isogeny Key Encapsulation) [27], a variant of SIDH, has been submitted to the NIST competition on post-quantum cryptography. The Round 3 candidates of this competition were recently announced on July 22, 2020, and SIKE advanced to the third round as an alternate candidate.

4.1. Supersingular Isogeny Diffie-Hellman (SIDH)

The SIDH key exchange protocol by Jao and De Feo [17] is based on the difficulty of computing exponentially large-degree isogenies between supersingular elliptic curves. This key exchange protocol has a specific method. First choose different small primes ℓ_A, ℓ_B and $e_A, e_B \in \mathbb{N}$. Then choose a small integer $f \in \mathbb{N}$ as a cofactor such that $p = \ell_A^{e_A} \ell_B^{e_B} f \pm 1$ is a prime number. Choose E to be a supersingular elliptic curve over $\mathbb{F}_{p^2}$ such that $E(\mathbb{F}_{p^2})$ has group structure isomorphic to a product of two cyclic groups of order $(\ell_A^{e_A} \ell_B^{e_B} f)^2$. Fix points $P_A, Q_A \in E[\ell_A^{e_A}]$ in such a way that the group generated by P_A and Q_A is the whole group $E[\ell_A^{e_A}]$. Similarly, choose P_B, Q_B such that $\langle P_B, Q_B \rangle = E[\ell_B^{e_B}]$.

In [17], the sender randomly chooses two secret integers $m_A, n_A \in (0, \ell_A^{e_A})$. Similarly, the receiver randomly chooses also two chosen secret integers $m_B, n_B \in (0, \ell_B^{e_B})$. However, in [8], Costello *et al.* showed that choosing one secret number for each of the parties provides the same security level as in SIDH. In Figure 4.1 we describe SIDH using one secret for each parties Alice (the sender) and Bob (the receiver).

Parameters: Prime number $p = \ell_A^{e_A} \ell_B^{e_B} f \pm 1$, an elliptic curve E over $\mathbb{F}_{p^2}$, $P_A, Q_A \in E[\ell_A^{e_A}]$ and $P_B, Q_B \in E[\ell_B^{e_B}]$.

The common key is $j(E_{AB}) = j(E_{BA})$, since two elliptic curves E_{AB} and E_{BA} are both isomorphic to the curve E over $\mathbb{F}_p^2$ where

$$E = \langle P_A + [n_A]Q_A, P_B + [n_B]Q_B \rangle \quad (4.1)$$

The security of SIDH informally relies on the (supersingular) isogeny walk problem: given two elliptic curves E, E' in the same isogeny class, find a path made of isogenies of small degree between E and E' .

The isogeny walk problem has been considered in the literature even before

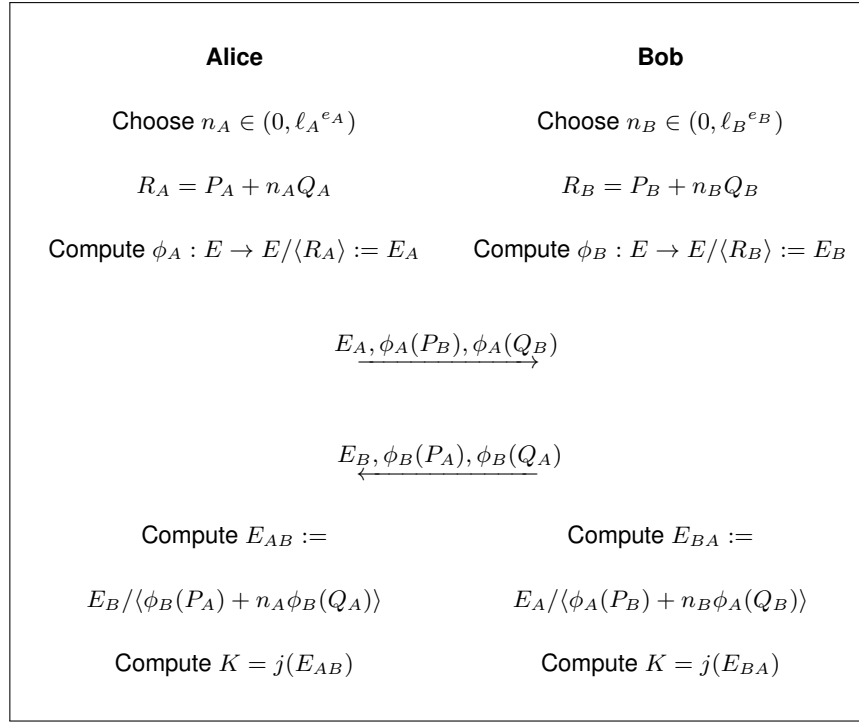


Figure 4.1: Jao-De Feo Method.

the introduction of isogeny based cryptography. The best generic algorithm currently known is due to Galbraith [22]. It is a meet in-the-middle strategy that, on average, requires a number of elementary steps proportional to the square root of the size of the isogeny class of E and E' . In the supersingular case, an improvement due to Delfs and Galbraith has roughly the same computational complexity, but only uses a constant amount of memory.

Over $\mathbb{F}_{p^2}$, there is a unique isogeny class of supersingular elliptic curves (up to twist), and it has size roughly $p/12$. Thus, the algorithm of Delfs and Galbraith would find an isogeny between the starting curve E_0 and a public curve E' in $O(\sqrt{p})$ time. Nevertheless, these generic algorithms do not improve upon exhaustive search. Indeed, if $p = 2^n 3^m - 1$, the key spaces have sizes roughly 2^n and 3^m ; thus, if these are chosen to balance out, then the size of the key spaces is roughly $\sqrt{p}$.

However, the idea of Galbraith's meet-in-the-middle approach can be easily adapted to attack SIDH in only $O(\sqrt[4]{p})$ operations. To find the secret isogeny of degree l^e between E' and E_0 , an attacker builds a tree of all curves isogenous to E_0 via isogenies of degree $l^{e/2}$, and a similar tree of all curves isogenous to E' of degree $l^{e/2}$. Since we suppose that an isogeny of degree l^e exists between E' and E_0 , and since the length of this walk is much shorter than the size of the graph, with high probability

the two trees will have exactly one curve E'' in common, so the secret isogeny will be recovered by composing the paths $E_0 \rightarrow E''$ and $E'' \rightarrow E'$. This procedure only requires $O(\sqrt{l^e})$ steps or $O(\sqrt[4]{p})$.

Definition 4.1: (Claw Problem). Given two functions $f : A \rightarrow C$ and $g : B \rightarrow C$ with domains of equal size, find a pair (a, b) such that $f(a) = g(b)$.

The claw problem can be solved in time $O(|A| + |B|)$ and space $O(|A|)$ on a classical computer by building a hash table for $f(a)$ for $a \in A$ and looking for hits with $g(b)$, where $b \in B$. This gives a $O(\sqrt[4]{p})$ classical attack on SIDH key exchange. Tani [42] showed that on a quantum computer this problem can be solved in quantum complexity $O((|A| \cdot |B|)^{1/3})$, thus giving a $O(\sqrt[6]{p})$ quantum attack (where $|A| = |B| = O(p^{1/4})$).

For the security of protocols, there is no need to use two different secret keys. Hence, we choose one secret key to compute isogenies in this section. In addition, we fix supersingular elliptic curve as

$$E : y^2 = x^3 + x \quad (4.2)$$

where $p = 2^n 3^m - 1$.

In [23], there is an observation for validation check. We summarize this validation in Remark 4.1.

Remark 4.1: The Weil pairing can be used to check a lot more than just independence. A standard fact is that if $\phi : E \rightarrow E'$ is an isogeny and if $P, Q \in E[n]$ then

$$e_N(\phi(P), \phi(Q)) = e_N(P, Q)^{\deg(\phi)} \quad (4.3)$$

where the first Weil pairing is computed on E and the second one is on E' .

This allows to validate not only that the points are independent but also that they are consistent with being the image of the correct points under an isogeny of the correct degree. Hence, a natural validation step for Alice to run in the protocols is to check

$$e_{2^n}(\phi_B(P_A), \phi_B(Q_A)) = e_{2^n}(P_A, Q_A)^{3^m} \quad (4.4)$$

This will give her some assurance that the points $\phi_B(P_A), \phi_B(Q_A)$ provided by Bob are consistent with being the images of the correct points under an isogeny of the correct degree.

Lemma 4.1: Let $R, S \in E[2^n]$ be linearly independent points of order 2^n and let $a \in \mathbb{Z}$. Then,

$$\langle R + [a](S + [2^{n-1}]R) \rangle = \langle R + [a]S \rangle \quad (4.5)$$

if and only if a is even.

Bob learns the parity of a based on whether or not Alice computes the same curve. This attack can be extended adaptively to learn each bit of efficiently.

Moreover, Weil pairing is

$$e_{2^n}(R, (S + [2^{n-1}]R)) = e_{2^n}(R, S) = e_{2^n}(P, Q)^{3^m} \quad (4.6)$$

and so the attack is not detectable using pairings. However, we solve this problem in our new protocol which is defined in Section 6.

4.2. Supersingular Isogeny Key Encapsulation (SIKE)

SIKE is a key encapsulation mechanism that is an actively-secure variant of the SIDH protocol, i.e., it offers resistance against indistinguishability under adaptive chosen ciphertext (IND-CCA2) attacks. In practice, this means that SIDH keys are ephemeral while SIKE's do not need to be. Algorithm of SIKE can be summarized as follows:

- *Public Parameters:* Alice and Bob start from supersingular elliptic curve $E_0/\mathbb{F}_{p^2}$ with prime number $p = 2^{e_A}3^{e_B} - 1$ along with basis points $\{P_A, Q_A\}$ and $\{P_B, Q_B\}$ that generate $E_0 = [2^{e_A}]$ and $E_0 = [3^{e_B}]$, respectively.
- *Key Generation:* Alice computes a random integer $sk_A \in_R \mathbb{Z}/2^{e_A}\mathbb{Z}$ and uses it to compute a secret isogeny ϕ_A , where she uses the image curve E_A and the image points $\phi_A(P_B), \phi_A(Q_B)$ to form her public key pk_A .

Table 4.1: SIKE parameter sets for each NIST security level.

SIKE Scheme	Security Level	As Strong as	Prime Form	Secret Key	Public Key	Cipher Text	Shared Secret
SIKEp434	NIST level 1	AES128	$p_{434} = 2^{216}3^{137} - 1$	374	330	346	16
SIKEp503	NIST level 2	SHA256	$p_{503} = 2^{250}3^{159} - 1$	434	378	402	24
SIKEp610	NIST level 3	AES192	$p_{610} = 2^{305}3^{192} - 1$	524	462	486	24
SIKEp751	NIST level 5	AES256	$p_{751} = 2^{372}3^{239} - 1$	644	564	596	32

- *Encapsulation:* Bob computes his secret key r based on Alice's public key and a random message m , which he uses to compute the kernel of his secret isogeny and to form his public key as $pk_B = \{E_B, \phi_B(P_A), \phi_B(Q_A)\}$. He computes a second isogeny, which leads him to the final image curve E_{AB} . He generates a ciphertext appending the message m masked by $j(E_{AB})$ of the curve to his public key.
- *Decapsulation:* Alice uses her secret key and Bob's public key projection points to reach curve E_{BA} , which since it is isomorphic to E_{AB} features the same j -invariant. She uses the unmasked value m to obtain Bob's secret key and uses it to re-compute his public key, where by comparing the result with c_0 , she computes the shared secret or uses random value, preventing further communication.

4.3. Commutative Supersingular Isogeny Diffie-Hellman(CSIDH)

Castnyck et al. [5] proposed an efficient commutative group action suitable for non-interactive key exchange in a post-quantum setting. Their construction follows the layout of the Couveignes–Rostovtsev–Stolbunov cryptosystem, but they applied it to supersingular elliptic curves defined over a large prime field $\mathbb{F}_p$, rather than to ordinary elliptic curves. The Diffie–Hellman scheme resulting from the group action allows for publickey validation at very little cost. To understand this protocol indeed, we now give the definition of the class group action.

Public Parameters: $p = 2^{e_A}3^{e_B} - 1$, $E_0/\mathbb{F}_{p^2}$, $\{P_A, Q_A\}$ and $\{P_B, Q_B\}$	
Alice	Bob
Input: -	
Output: s, sk_A, pk_A	
1. $sk_A \in_R \mathbb{Z}/2^{e_A}\mathbb{Z}$	
2. $\phi_A : E_0 \rightarrow E_A$ with $ker(\phi_A) = \langle P_A + [sk_A]Q_A \rangle$	
3. $pk_A = (E_A, \phi_A(P_B), \phi_A(Q_B))$	
4. $s \in_R \{0, 1\}^t$	
Compress public key	
Input: s, sk_B, pk_B, c	Input: pk_A
Output: ss	Output: c, ss
Decompress ciphertext	Decompress public key
1. $\phi'_A : E_B \rightarrow E_{BA}$ with $ker(\phi'_A) = \langle \phi_B(P_A) + [sk_A]\phi_B(Q_A) \rangle$	1. $m \in_R \{0, 1\}^t$
2. $m' = c_1 \oplus K(j(E_{BA}))$	2. $r = H(m pk_A) \bmod 3^{e_B}$
3. $r' = H(m' pk_A) \bmod 3^{e_B}$	3. $\phi_B : E_0 \rightarrow E_B$ with $ker(\phi_B) = \langle P_B + [r]Q_B \rangle$
4. $\phi''_A : E_0 \rightarrow E_{B'}$ with $ker(\phi''_A) = \langle P_B + [r']Q_B \rangle$	4. $pk_B = \{E_B, \phi_B(P_A), \phi_B(Q_A)\}$
5. $pk'_B = \{E_{B'}, \phi''_A(P_A), \phi''_A(Q_A)\}$	5. $\phi'_B : E_A \rightarrow E_{AB}$ with $ker(\phi'_B) = \langle \phi_A(P_B) + [r]\phi_A(Q_B) \rangle$
6. IF $pk'_B = pk_B$	6. $c = (c_0, c_1) = (pk_B, K(j(E_{AB})) \oplus m)$
$ss = (J(m' c))$	7. $ss = (J(m c))$
ELSE $ss = (J(s c))$	Compress ciphertext

Figure 4.2: SIKE Algorithm.

Definition 4.2: (The class-group action) Fix a prime $p \geq 5$ and an ordinary or supersingular elliptic curve E defined over $\mathbb{F}_p$. The Frobenius endomorphism π of E satisfies a characteristic equation,

$$\pi^2 - t\pi + p = 0 \quad (4.7)$$

in $End_p(E)$, where t is the trace of Frobenius. The curve E is supersingular if and only if $t = 0$. The $\mathbb{F}_p$ -rational endomorphism ring $End_p(E)$ is an order $\mathcal{O}$ in the imaginary quadratic field K . We note that $\mathcal{O}$ always contains the Frobenius endomorphism π , and hence the order $\mathbb{Z}[\pi]$.

Any invertible ideal $\mathfrak{a}$ of $\mathcal{O}$ splits into a product of $\mathcal{O}$ -ideals as $(\pi\mathcal{O})^r \mathfrak{a}_s$, where $\mathfrak{a}_s \not\subseteq \pi\mathcal{O}$. This defines an elliptic curve $E/\mathfrak{a}$ and an isogeny

$$\varphi_{\mathfrak{a}} : E \rightarrow E/\mathfrak{a} \quad (4.8)$$

of degree $N(\mathfrak{a})$.

Setup. Global parameters of the scheme are a large prime $p = 4 \cdot \ell_1 \cdots \ell_n - 1$, where the ℓ_i are small distinct odd primes, and the supersingular elliptic curve $E_0: y^2 = x^3 + x$ over $\mathbb{F}_p$ with endomorphism ring $\mathcal{O} = \mathbb{Z}[\pi]$.

Key generation. The private key is an n -tuple $(e_1, \dots, e_n)$ of integers, each sampled randomly from a range $\{-m, \dots, m\}$. These integers represent the ideal class $[\mathfrak{a}] = [\mathfrak{l}_1^{e_1} \cdots \mathfrak{l}_n^{e_n}] \in \text{cl}(\mathcal{O})$, where $\mathfrak{l}_i = (\ell_i, \pi - 1)$. The public key is the Montgomery coefficient $A \in \mathbb{F}_p$ of the elliptic curve $[\mathfrak{a}]E_0: y^2 = x^3 + Ax^2 + x$ obtained by applying the action of $[\mathfrak{a}]$ to the curve E_0 .

Key exchange. Suppose Alice and Bob have key pairs $([\mathfrak{a}], A)$ and $([\mathfrak{b}], B)$. Upon receiving Bob's public key $B \in \mathbb{F}_p \setminus \{\pm 2\}$, Alice verifies that the elliptic curve $E_B: y^2 = x^3 + Bx^2 + x$ is indeed in $\mathcal{Ell}_p(\mathcal{O}, \pi)$. She then applies the action of her secret key $[\mathfrak{a}]$ to E_B to compute the curve $[\mathfrak{a}]E_B = [\mathfrak{a}][\mathfrak{b}]E_0$. Bob proceeds analogously with his own secret $[\mathfrak{b}]$ and Alice's public key A to compute the curve $[\mathfrak{b}]E_A = [\mathfrak{b}][\mathfrak{a}]E_0$. The shared secret is the Montgomery coefficient S of the common secret curve $[\mathfrak{a}][\mathfrak{b}]E_0 = [\mathfrak{b}][\mathfrak{a}]E_0$ written in the form $y^2 = x^3 + Sx^2 + x$, which is the same for Alice and Bob due to the commutativity of $\text{cl}(\mathcal{O})$.

Figure 4.3: CSIDH Key Agreement Protocol.

Let $\mathcal{O}$ be an order in an imaginary quadratic field and $\pi \in \mathcal{O}$ such that $\mathcal{Ell}_p(\mathcal{O}, \pi)$ is non-empty. Then the ideal-class group $\text{cl}(\mathcal{O})$ acts freely and transitively on the set $\mathcal{Ell}_p(\mathcal{O}, \pi)$ via the map

$$\text{cl}(\mathcal{O}) \times \mathcal{Ell}_p(\mathcal{O}, \pi) \rightarrow \mathcal{Ell}_p(\mathcal{O}, \pi) \text{ such that } ([\mathfrak{a}], E) \rightarrow E/\mathfrak{a} \quad (4.9)$$

in which $\mathfrak{a}$ is chosen as an integral representative.

To emphasize the fact that we are dealing with a group action, we will from now on write $[\mathfrak{a}] * E$ or simply $[\mathfrak{a}]E$ for the curve $E/\mathfrak{a}$ defined above. In Figure 4.3, we explain CSIDH in details.

4.4. Other Protocols

B-SIDH [7]: This paper explored a new way of instantiating isogeny-based cryptography in which parties can work in both the $(p + 1)$ -torsion of a set of supersingular curves and in the $(p - 1)$ -torsion corresponding to the set of their quadratic twists. Although the isomorphism between a given supersingular curve

and its quadratic twist is not defined over $\mathbb{F}_{p^2}$ in general, restricting operations to the x -lines of both sets of twists allows all arithmetic to be carried out over $\mathbb{F}_{p^2}$ as usual. Furthermore, since supersingular twists always have the same $\mathbb{F}_{p^2}$ rational j -invariant, the SIDH protocol remains unchanged when Alice and Bob are free to work in both sets of twists.

SiGamal [38]: In this paper, Moriya et al. proposed two new supersingular isogeny-based public key encryptions: SiGamal and C-SiGamal. These public key encryptions are developed by giving an additional point of the order 2^r to CSIDH. SiGamal seems similar to ElGamal encryption, while C-SiGamal is a compressed version of SiGamal.

SETA [13]: Guilhem et al. provided new PKE schemes and KEMs based on isogeny problems. Key recovery security for their schemes only relies on the original isogeny problem for supersingular curves, and the standard OW-CPA and INDCCA security rely on different problems than those used in SIDH and SIKE. They argued that, depending on future cryptanalysis progress, their schemes can provide an interesting alternative to the SIKE family.

SeaSign [11]: De Feo and Galbraith defined a new signature scheme for isogenies that connects the class group actions of CSIDH with the notion of Fiat-Shamir with aborts. Their techniques allow to have signatures of size less than one kilobyte at the 128-bit security level, even with tight security reduction (to a non-standard problem) in the quantum random oracle model. Hence, their signatures are potentially shorter than lattice signatures, but signing and verification are currently very expensive.

SQISign [18]: Feo et al. introduced a new signature scheme from isogeny graphs of supersingular elliptic curves. The signature scheme is derived from a new one-round, high soundness, interactive identification protocol. Targeting the post-quantum NIST-1 level of security, their implementation results in signatures of 204 bytes, secret keys of 16 bytes and public keys of 64 bytes. In particular, the signature and public key sizes combined are an order of magnitude smaller than all other post-quantum signature schemes.

5. CRYPTANALYSIS OF SOME RECENT ISOGENY BASED CRYPTOGRAPHIC PROTOCOLS

Portions of this chapter were published in Bull. Iran. Math. Soc. 47 (2021). The contents of this chapter represent studies in that work.

Recently, Daghigh *et al.* proposed some isogeny based key agreement protocols in [10]. In that paper, Daghigh *et al.* argued that their key exchange protocols based on supersingular isogenies provide the same security level as the Jao *et al.*'s protocol SIDH [17]. In [25], we firstly study on a weakness of these protocols to a key manipulation attack. Therefore, unlike the security claims of Daghigh *et al.*, these protocols unfortunately do not provide the same security when compared with SIDH. Secondly, we then prove that in particular the secret dual isogenies of the protocols can easily be retrieved by applying our key manipulation attack. Moreover, we discuss the countermeasures to this attack.

5.1. Daghigh *et al.* Key Exchange Method

In [10], Daghigh *et al.* suggested three Diffie-Hellman type key exchange protocols using isogenies of elliptic curves. One of these methods is just suitable for ordinary elliptic curves while the others, Method 2 and Generalized Method 2, are suitable for both ordinary and supersingular elliptic curves. Method 2 uses the endomorphism ring of an elliptic curve E . By using isogenies between two elliptic curves, they extend this method and named as Generalized Method 2. They claimed that the main advantage of their protocols is that the transmitted information is only $\phi_A(P)$ and $\phi_B(P)$. Thus, the protocols are more efficient in terms of communication overhead than SIDH. The exchanged information in [17] includes the image of the generators of $E[\ell_A^{e_A}]$ and $E[\ell_B^{e_B}]$ under the secret isogenies ϕ_B and ϕ_A , respectively. We now give the description of the Generalized Method 2 on which we apply our attack in the next section.

Generalized Method 2: Assume that E and E' are two isogenous elliptic curves. Let n be the order of E and E' and $P \in E(\mathbb{F}_q)[n]$. Suppose that Alice and Bob agree on

E, E' and point P . Alice chooses an isogeny $\phi_A \in \text{Hom}(E, E')$ and sends $\phi_A(P)$ to Bob. Similarly, Bob chooses an isogeny $\phi_B \in \text{Hom}(E, E')$ and sends $\phi_B(P)$ to Alice. $\widehat{\phi_A}$ and $\widehat{\phi_B}$ denote the dual isogenies of ϕ_A and ϕ_B , respectively. Alice and Bob can share a common key as described in Figure 5.1.

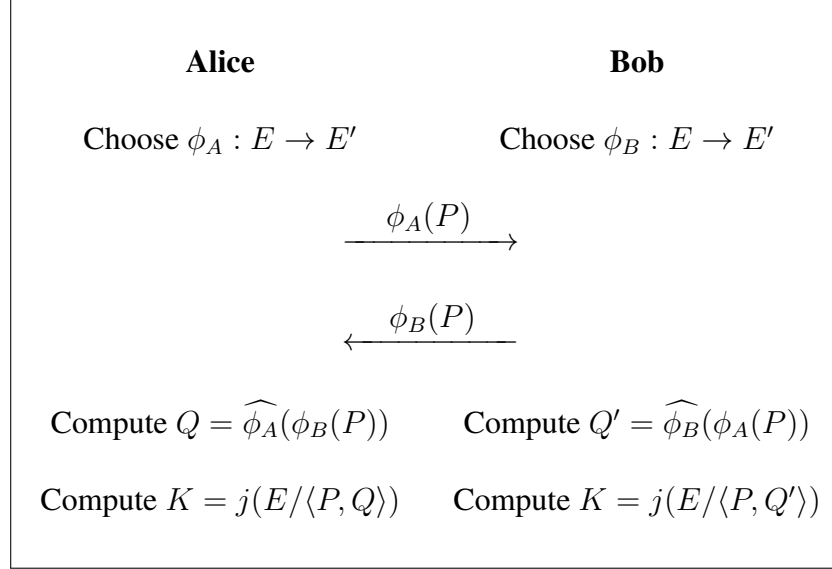


Figure 5.1: Daghigh *et al.* Generalized Method 2.

5.2. A Key Manipulation Attack on Daghigh *et al.* Protocols

Isogeny based key agreement protocols described in [10], are vulnerable to a key manipulation attack. In this part, we apply this attack to Generalized Method 2 protocol. Note that this attack can be easily applied to their other protocols of [10] in the same way.

Suppose Eve is an attacker and she can modify the messages in the protocol. Let e be an integer such that $e \in (0, n)$. Alice wants to send $\phi_A(P)$ to Bob. However, Eve blocks the message and sends $e\phi_A(P)$ to Bob. In the same way, Bob wants to send $\phi_B(P)$ to Alice. However, Eve blocks the message and sends $e\phi_B(P)$ to Alice.

Proposition 5.1: The values of K_1 and K_2 in Figure 5.2 are equal.

Proof Assume $\alpha \in \text{End}(E)$ and $\text{Tr}(\alpha) = \widehat{\alpha} + \alpha = k \in \mathbb{Z}$. For $e \in (0, n)$, we have

$$\langle P, e\widehat{\alpha}(P) \rangle = \langle P, ekP - e\alpha(P) \rangle = \langle P, e\alpha(P) \rangle \quad (5.1)$$

Alice	Eve	Bob
Choose $\phi_A \in \text{Hom}(E, E')$	Choose $e \in (0, n)$	Choose $\phi_B \in \text{Hom}(E, E')$
$\xrightarrow{\phi_A(P)}$	$\xrightarrow{e\phi_A(P)}$	
$\xleftarrow{e\phi_B(P)}$	$\xleftarrow{\phi_B(P)}$	
Compute $eQ = e\widehat{\phi_A}(\phi_B(P))$		Compute $eQ' = e\widehat{\phi_B}(\phi_A(P))$
Compute $K_1 = j(E/\langle P, eQ \rangle)$		Compute $K_2 = j(E/\langle P, eQ' \rangle)$

Figure 5.2: A Key Manipulation Attack on Generalized Method 2 of [10].

If $\phi_A, \phi_B \in \text{Hom}(E, E')$ then $\widehat{\phi_A}\phi_B, \widehat{\phi_B}\phi_A \in \text{End}(E)$. Hence,

$$\begin{aligned}
K_1 &= j(E/\langle P, eQ \rangle) = j(E/\langle P, e\widehat{\phi_A}\phi_B(P) \rangle) \\
&= j(E/\langle P, e\widehat{\phi_B}\phi_A(P) \rangle) = j(E/\langle P, eQ' \rangle) = K_2
\end{aligned} \tag{5.2}$$

■

At the end of the protocol, Alice and Bob agree necessarily on the same key. They presume that the key is equal to $j(E/\langle P, Q \rangle)$ but it is equal to $j(E/\langle P, eQ \rangle)$. Hence, Eve changed the agreed key by modifying the messages in the protocol. We will use this attack to find the secret isogenies of Alice and Bob. We note on the other hand that this attack cannot be used to SIDH of Jao *et al.* [17]. Assume Eve is an attacker and she can modify messages. Eve sends $E_A, e\phi_A(P_B), e\phi_A(Q_B)$ to Bob and $E_B, e\phi_B(P_A), e\phi_B(Q_A)$ to Alice.

Alice computes the key as follows:

$$E_{AB} := E_B / \langle e\phi_B(P_A) + en_A\phi_B(Q_A) \rangle = E_B / \langle e\phi_B(R_A) \rangle \tag{5.3}$$

Bob computes the key as follows:

$$E_{BA} := E_A / \langle e\phi_A(P_B) + en_B\phi_A(Q_B) \rangle = E_A / \langle e\phi_A(R_B) \rangle \tag{5.4}$$

However, E_{AB} and E_{BA} are not always isomorphic curves since,

$$E_B / \langle e\phi_B(R_A) \rangle \not\cong E_A / \langle e\phi_A(R_B) \rangle \quad (5.5)$$

where $e|\ell_A, \ell_B$. Hence, their j -invariants must be necessarily different.

5.3. Recovery of Isogeny Using Key Manipulation Attack

In [44], Bo Ti showed that changing the base point to a random point on the curve via a fault injection will give an information about the dual isogeny. We will give the lemma in [44] and then apply it to the Generalized Method 2 of [10].

Lemma 5.1: Let E_1 be a supersingular elliptic curve over $\mathbb{F}_{p^2}$, where $p = \ell_A^{e_A} \ell_B^{e_B} f \pm 1$. Suppose $\phi : E_1 \rightarrow E_2$ is an isogeny of degree $\ell_A^{e_A}$ with a cyclic kernel and let $\{P, Q\}$ be generators of $E_1[\ell_A^{e_A}]$. Then for any $X \in E_1[\ell_A^{e_A}]$, define $\psi : E_2 \rightarrow E'$ such that $\ker \psi = \langle \phi(X) \rangle$, then there exists some $\theta : E' \rightarrow E_1$ of degree ℓ_A^ϵ , $\epsilon \leq e_A$, such that

$$\widehat{\phi} = \theta \circ \psi \quad (5.6)$$

The lemma implies that given the image of a point in $E_1[\ell_A^{e_A}]$ under an $\ell_A^{e_A}$ -isogeny, ϕ , we are able to find an isogeny ψ which is close to the dual isogeny of ϕ . To obtain the dual isogeny, one has to first recover θ . If ϵ is chosen sufficiently small, one will be able to recover θ by brute force. In fact, Ti examined the size of ϵ in [44] and showed that ϵ is small in most cases.

We will show that this attack can always be applied to the protocols introduced in [10]. Assume ℓ denotes the degree of ψ defined in Lemma 5.1.

Theorem 5.1: ℓ -degree secret dual isogenies can be recovered in protocols of [10] by applying the key manipulation attack in $O(\ell \log \ell)$.

Proof In the protocols introduced in [10], Alice and Bob agree on two curves E, E' and a point $P \in E(\mathbb{F}_q)[n]$, where n is any positive integer. Assume $p = \ell_1^{e_1} \ell_2^{e_2} \cdots \ell_k^{e_k} \pm 1$ where $\ell_1, \ell_2, \dots, \ell_k$ are distinct small primes and $q = p^2$. Then with $(P_1, Q_1), (P_2, Q_2), \dots, (P_k, Q_k)$ being the generators of $E[\ell_1^{e_1}], E[\ell_2^{e_2}], \dots, E[\ell_k^{e_k}]$ respectively, the point P takes the form

$$P = [n_1]P_1 + [m_1]Q_1 + [n_2]P_2 + [m_2]Q_2 + \cdots + [n_k]P_k + [m_k]Q_k \quad (5.7)$$

where $n_i, m_i \in [0, \ell_i^{e_i}]$ for $i = 1, 2, \dots, k$.

Note that in Lemma 5.1, the degree of ϕ_A is equal to $\ell_1^{e_1}$. However, in this proof it can be chosen as ℓ_1^f where f is an integer such that $f \leq e_1$. Without loss of generality we can assume that the degree of the isogeny $\phi_A : E \rightarrow E'$ is an integer d with $d \mid \ell_1^{e_1}$. In the protocol, Alice calculates $\phi_A(P)$ and then sends it to Bob. Eve can now apply key manipulation attack. Eve chooses $e = \ell_2^{e_2} \cdots \ell_k^{e_k}$ and calculates:

$$\begin{aligned} e\phi_A(P) &= e\phi_A([n_1]P_1 + [m_1]Q_1 + [n_2]P_2 + [m_2]Q_2 + \\ &\quad \cdots + [n_k]P_k + [m_k]Q_k) = e[n_1]\phi_A(P_1) + e[m_1]\phi_A(Q_1) + \\ &\quad e[n_2]\phi_A(P_2) + e[m_2]\phi_A(Q_2) + \cdots + e[n_k]\phi_A(P_k) + e[m_k]\phi_A(Q_k) \\ &= [n'_1]\phi_A(P_1) + [m'_1]\phi_A(Q_1) \end{aligned} \quad (5.8)$$

Eve has the point $[n'_1]\phi_A(P_1) + [m'_1]\phi_A(Q_1)$, the subgroup generated by this point will help with the construction of the dual isogeny of ϕ_A by using Lemma 5.1. We apply Lemma 5.1, with the point $X = [n'_1]P_1 + [m'_1]Q_1$ where $X \in E[\ell_1^{e_1}]$ and the isogeny ϕ_A of degree d . We then have,

$$e\phi_A(P) = [n'_1]\phi_A(P_1) + [m'_1]\phi_A(Q_1) = \phi_A(X) \quad (5.9)$$

The adversary can obtain $\psi : E' \rightarrow E''$, where $\ker \psi = \langle \phi_A(X) \rangle$ and computes $\theta : E'' \rightarrow E$. Hence, $\widehat{\phi_A} = \theta \circ \psi$ can be efficiently computed. Note that the adversary can also compute $\widehat{\phi_B}$ using the same method. This implies that the secret key $K =$

$j(E/\langle P, Q \rangle)$ be efficiently computed in $O(\ell \log \ell)$ corresponding to computing ψ using the Velu's formula. ■

We implemented key manipulation attack in SAGE software [14]. Without our countermeasure, we can get the session key by choosing e specifically. In the following example, we apply our implementation using the same parameters presented in [10, Example 5.2]. We can easily change the shared key and obtain the new shared key efficiently. The complexity of this attack is $O(\log n)$ corresponding only to a computation of an order of a point on an elliptic curve.

Example 5.1: Let $\mathbb{F}_{503^2} = \mathbb{F}_{503}(w)$ where $w^2 + w + 1 = 0$. Let $E : y^2 = x^3 - 1$ and $G = (496, 320w + 160) \in E$. Alice and Bob choose a random point $P = (258w + 158, 210w + 90) \in E[504]$ where $504 = 2^3 3^2 7$.

Using Velu's formula, they compute

$$\psi : E \rightarrow E/\langle G \rangle = E' \quad (5.10)$$

where $E' : y^2 = x^3 + 159x + 155$. The order of G is equal to $63 = 3^2 7$. Hence the degree of ψ is also equal to 63. Eve can calculate the order of G since it is a public point. Then, Eve chooses $e = 504/63 = 8$ in order to apply the key manipulation attack on this protocol. We explained this choice in detail in the proof of Theorem 5.1

To agree on a secret key, Alice chooses ϕ_A and sends $\phi_A(P) = (163w + 197, 81w + 224)$ to Bob. Eve blocks the message and sends $8 * \phi_A(P) = (279, 466)$ to Bob. Similarly, Bob chooses ϕ_B and sends $\phi_B(P) = (359w + 317, 88w + 356)$ to Alice. Eve blocks the message and sends $8 * \phi_B(P) = (175, 488)$ to Alice.

Now Alice computes,

$$Q = \widehat{\phi_A}(8 * \phi_B(P)) = 8 * \widehat{\phi_A}(\phi_B(P)) = (0 : 1 : 0) = \mathcal{O} \quad (5.11)$$

and $E/\langle P, Q \rangle \equiv E/\langle P \rangle : y^2 = x^3 + (478w + 23)x + (73w + 399)$ and obtains

$$K = j(E/\langle P \rangle) = 179w + 490 \quad (5.12)$$

Similarly Bob computes,

$$Q' = \widehat{\phi}_B(8 * \phi_A(P)) = 8 * \widehat{\phi}_B(\phi_A(P)) = (0 : 1 : 0) = \mathcal{O} \quad (5.13)$$

and $E/\langle P, Q' \rangle \equiv E/\langle P \rangle : y^2 = x^3 + (478w + 23)x + (73w + 399)$ and obtains

$$K = j(E/\langle P \rangle) = 179w + 490 \quad (5.14)$$

We know that P is the public point. Hence, Eve can calculate $j(E/\langle P \rangle)$ and find the secret key through the key manipulation attack.

We give another example of key manipulation attack with different parameters.

Example 5.2: Let $p = 2^8 * 3^6 * 7 - 1$, $\mathbb{F}_{p^2} = \mathbb{F}_p(i)$ where $i^2 + 1 = 0$. Let $E : y^2 = x^3 - 1$ and $G = (683855 * i + 259415, 391779 * i + 812888) \in E$. Alice and Bob choose a random point $P = (101638 * i + 867340, 101670 * i + 1302290) \in E[n]$ where $n = 2^{16} * 3^{12} * 7^2$.

Using Velu's formula, they compute

$$\psi : E \rightarrow E/\langle G \rangle = E' \quad (5.15)$$

where E' : The order of G is equal to $\ell = 2^6 * 3^6 * 7$. Hence the degree of ψ is also equal to ℓ . Eve can calculate the order of G since it is a public point. Then, Eve chooses $e = n/\ell = 2^{10} * 3^6 * 7$ in order to apply the key manipulation attack on this protocol. We explained this choice in detail in the proof of Theorem 5.1

To agree on a secret key, Alice chooses ϕ_A and sends $\phi_A(P) = (48804 * i + 122598, 864631 * i + 1201450)$ to Bob. Eve blocks the message and sends $e * \phi_A(P)$ to Bob. Similarly, Bob chooses ϕ_B and sends $\phi_B(P) = (1217823 * i + 1017871, 993087 * i + 298277)$ to Alice. Eve blocks the message and sends $e * \phi_B(P)$ to Alice.

Now Alice computes,

$$Q = \widehat{\phi}_A(e * \phi_B(P)) = e * \widehat{\phi}_A(\phi_B(P)) = (0 : 1 : 0) = \mathcal{O} \quad (5.16)$$

and $E/\langle P, Q \rangle \equiv E/\langle P \rangle : y^2 = x^3 + (848628 * i + 304031) + (1087793 * i + 405868)$
and obtains

$$K = j(E/\langle P \rangle) = 150552 * i + 667176 \quad (5.17)$$

Similarly Bob computes,

$$Q' = \widehat{\phi_B}(e * \phi_A(P)) = e * \widehat{\phi_B}(\phi_A(P)) = (0 : 1 : 0) = \mathcal{O} \quad (5.18)$$

and $E/\langle P, Q' \rangle \equiv E/\langle P \rangle : y^2 = x^3 + (848628 * i + 304031) + (1087793 * i + 405868)$
and obtains

$$K = j(E/\langle P \rangle) = 150552 * i + 667176 \quad (5.19)$$

We know that P is the public point. Hence, Eve can calculate $j(E/\langle P \rangle)$ and find the secret key through the key manipulation attack.

5.3.1. Countermeasures of Key Manipulation Attack

In order to resist the key manipulation attack, we will give some countermeasures for Daghigh *et al.* protocols in this section. The first countermeasure is to choose the public point P with some *restrictions*. With these restrictions, the attacker cannot apply Lemma 5.1 to the protocol in order to recover the secret isogeny. The second countermeasure is to *check the points* Q and Q' . We define a modified version of the Daghigh *et al.* protocol as follows:

Modified Generalized Method 2: Alice and Bob agree on two isogenous elliptic curves E and E' with order n which is a positive integer. Alice chooses an isogeny $\phi_A \in \text{Hom}(E, E')$ and Bob chooses an isogeny $\phi_B \in \text{Hom}(E, E')$. Assume the generator elements of the kernel of ϕ_A and ϕ_B in $E(\mathbb{F}_q)[\ell]$ where $\ell|n$. Alice and Bob agree also on a point P with restrictions:

- i) $P \in E(\mathbb{F}_q)[t]$ such that $t|n$ and $\gcd(\ell, t) = 1$.

Alice sends $\phi_A(P)$ to Bob and Bob sends $\phi_B(P)$ to Alice. $\widehat{\phi_A}$ and $\widehat{\phi_B}$ denote the dual isogenies of ϕ_A and ϕ_B . Alice and Bob check the points Q and Q' , respectively:

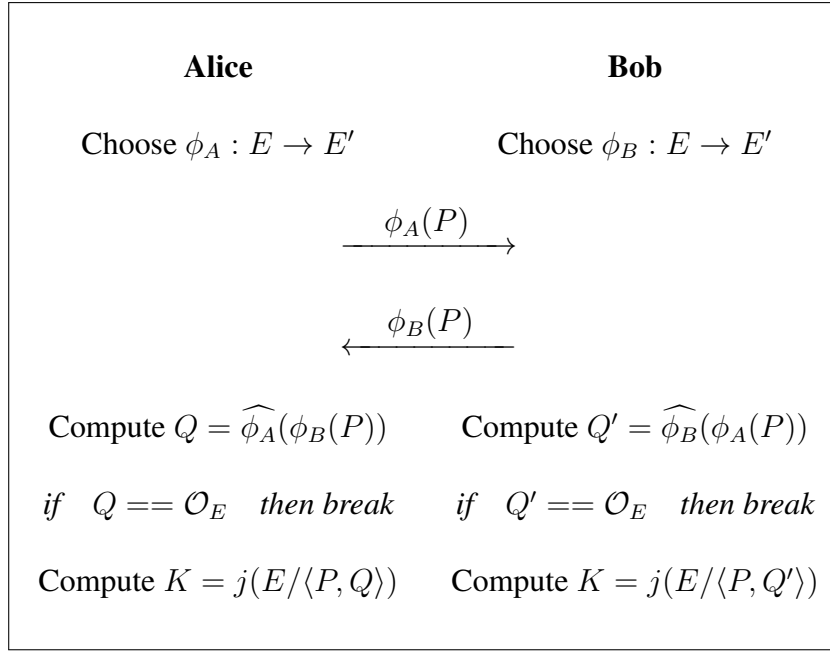


Figure 5.3: Modified version of Daghigh *et al.* Generalized Method 2.

ii) If Q or Q' is the identity element then the protocol must be cancelled.

Alice and Bob can share a common key as it is described in Figure 5.3.

6. A NEW POST-QUANTUM AUTHENTICATED KEY AGREEMENT PROTOCOL (SI-MQV)

We designed a new isogeny based key agreement protocol. In this section, we give the details of this protocol. The public parameters are the supersingular curve $E/\mathbb{F}_{p^2}$ whose group order is $(l_A^{e_A} l_B^{e_B} f)^2$, two independent points P_A and Q_A that generate $E[l_A^{e_A}]$ and two independent points P_B and Q_B that generate $E[l_B^{e_B}]$. For simplicity, we will take $l_A^{e_A} = 2^n, l_B^{e_B} = 3^m$.

- *Alice* calculates $R_A = P_A + [n_A]Q_A$ and $R_X = P_A + [n_X]Q_A$ which have order 2^n .
- *Bob* calculates $R_B = P_B + [n_B]Q_B$ and $R_Y = P_B + [n_Y]Q_B$ which has order 3^m .
- *Alice* computes the isogeny $\phi_A : E \rightarrow E_A$ whose kernel is R_A .
- *Alice* computes the isogeny $\phi_X : E \rightarrow E_X$ whose kernel is R_X .
- *Alice* computes the isogeny $\phi_{AX} : E \rightarrow E_{AX}$ whose kernel is $R_A + R_X$.
- *Bob* computes the isogeny $\phi_B : E \rightarrow E_B$ whose kernel is R_B .
- *Bob* computes the isogeny $\phi_Y : E \rightarrow E_Y$ whose kernel is R_Y .
- *Bob* computes the isogeny $\phi_{BY} : E \rightarrow E_{BY}$ whose kernel is $R_B + R_Y$.
- *Alice* sends $E_A, E_X, E_{AX}, \phi_A(P_B), \phi_A(Q_B), \phi_X(P_B), \phi_X(Q_B), \phi_{AX}(P_B), \phi_{AX}(Q_B)$ to *Bob*.
- *Bob* sends $E_B, E_Y, E_{BY}, \phi_B(P_A), \phi_B(Q_A), \phi_Y(P_A), \phi_Y(Q_A), \phi_{BY}(P_A), \phi_{BY}(Q_A)$ to *Alice*.
- *Alice* controls that E_A, E_X, E_{AX} are supersingular elliptic curves over $\mathbb{F}_{p^2}$, if it fails then terminates the protocol.
- *Bob* controls that E_B, E_Y, E_{BY} are supersingular elliptic curves over $\mathbb{F}_{p^2}$, if it fails then terminates the protocol.
- *Alice* controls whether $\phi_A(P_B), \phi_A(Q_B)$ are independent points on E_A and $\phi_X(P_B), \phi_X(Q_B)$ are independent points on E_X of order 2^n , if it fails then terminates the protocol.
- *Bob* controls whether $\phi_B(P_A), \phi_B(Q_A)$ are independent points on E_B and $\phi_Y(P_A), \phi_Y(Q_A)$ are independent points on E_Y of order 3^m , if it fails then terminates the protocol.

- *Alice* controls $e_{2^n}(\phi_{BY}(P_A), \phi_{BY}(Q_A)) = e_{2^n}(\phi_B(P_A), \phi_B(Q_A))^{3^m}$ and $e_{2^n}(\phi_{BY}(P_A), \phi_{BY}(Q_A)) = e_{2^n}(\phi_Y(P_A), \phi_Y(Q_A))^{3^m}$, if it fails then terminates the protocol.
- *Bob* controls $e_{3^n}(\phi_{AX}(P_B), \phi_{AX}(Q_B)) = e_{3^n}(\phi_A(P_B), \phi_A(Q_B))^{2^n}$ and $e_{3^n}(\phi_{AX}(P_B), \phi_{AX}(Q_B)) = e_{3^n}(\phi_X(P_B), \phi_X(Q_B))^{2^n}$, if it fails then terminates the protocol.
- *Alice* computes $\phi_{BYAX} : E_{BY} \rightarrow E_{BYAX}$ whose kernel is the point $\phi_{BY}(P_A) + [n_A]\phi_{BY}(Q_A) + \phi_{BY}(P_A) + [n_X]\phi_{BY}(Q_A) = \phi_{BY}(P_A + [n_A]Q_A + P_A + [n_X]Q_A) = \phi_{BY}(R_A + R_X)$.
- *Bob* computes $\phi_{AXBY} : E_{AX} \rightarrow E_{AXBY}$ whose kernel is the point $\phi_{AX}(P_B) + [n_B]\phi_{AX}(Q_B) + \phi_{AX}(P_B) + [n_Y]\phi_{AX}(Q_B) = \phi_{AX}(P_B + [n_B]Q_B + P_B + [n_X]Q_B) = \phi_{AX}(R_B + R_Y)$.
- *Alice* computes the shared secret $K = H(j(E_{BYAX}), Alice, Bob)$.
- *Bob* computes the shared secret $K = H(j(E_{AXBY}), Alice, Bob)$.

Now, we prove that our key agreement protocol is secure under some security assumption. Firstly, we give some definition to understand our main theorem and its proof.

6.1. Security Models: CK Security and CK^+ Security

This section outlines the CK and CK^+ security definitions for two-pass PKI-based authenticated key exchange protocols. We assume that all parties are modeled by probabilistic polynomial-time (ppt) Turing machines while the adversary is modeled by a polynomial time quantum machine in our post-quantum CK and CK^+ models. For further CK and CK^+ details and explanations, see [19], [32]. It is worth to note here that the proposed protocol is one-round and thus, it is enough to describe the security model as for two-pass $AKAP$ because a two-pass model includes a one-round one.

We denote a party's identity $\hat{A}, \hat{B}, \hat{C}, \dots$. A party honestly generates its own keys, static public and static secret ones, and the static public key is linked with the party's identity in some systems like PKI. The maximum numbers of parties and sessions are polynomially bound in the security parameter.

<i>Alice</i>	<i>Bob</i>
$n_A \in (0, 2^n), P_A, Q_A$	$n_B \in (0, 3^m), P_B, Q_B$
$R_A = P_A + n_A Q_A$	$R_B = P_B + n_B Q_B$
$n_X \in (0, 2^n)$	$n_Y \in (0, 3^m)$
$R_X = P_A + n_X Q_A$	$R_Y = P_B + n_Y Q_B$
$\phi_A : E \rightarrow E/\langle R_A \rangle := E_A$	$\phi_B : E \rightarrow E/\langle R_B \rangle := E_B$
$\phi_X : E \rightarrow E/\langle R_X \rangle := E_X$	$\phi_Y : E \rightarrow E/\langle R_Y \rangle := E_Y$
$\phi_{AX} : E \rightarrow E/\langle R_A + R_X \rangle := E_{AX}$	$\phi_{BY} : E \rightarrow E/\langle R_B + R_Y \rangle := E_{BY}$
$E_A, E_X, E_{AX}, \phi_A(P_B), \phi_A(Q_B), \phi_X(P_B), \phi_X(Q_B), \phi_{AX}(P_B), \phi_{AX}(Q_B)$	
$E_B, E_Y, E_{BY}, \phi_B(P_A), \phi_B(Q_A), \phi_Y(P_A), \phi_Y(Q_A), \phi_{BY}(P_A), \phi_{BY}(Q_A)$	
$E_{BY} \rightarrow \left\langle \frac{E_{BY}}{\phi_{BY}(R_A + R_X)} \right\rangle := E_{BYAX}$	$E_{AXBY} \rightarrow \left\langle \frac{E_{AX}}{\phi_{AX}(R_B + R_Y)} \right\rangle := E_{AXBY}$
$K = H(j(E_{BYAX}), \text{Alice}, \text{Bob})$	$K = H(j(E_{AXBY}), \text{Alice}, \text{Bob})$

Figure 6.1: SI-MQV Protocol.

Session: An invocation of a protocol is called a *session*. A session is activated via an incoming message of the forms $(\Pi, I, \hat{A}, \hat{B})$ or $(\Pi, R, \hat{A}, \hat{B}, Y)$, where Π is a protocol identifier. If $\hat{A}$ is activated with $(\Pi, I, \hat{A}, \hat{B})$, then $\hat{A}$ is the session initiator, otherwise it is the session responder. We say that $\hat{A}$ is the owner (resp. peer) of session *sid* if the third (resp. fourth) coordinate of *sid* is $\hat{A}$. After activation, session initiator $\hat{A}$ creates ephemeral public key X and a new session identified with $(\Pi, I, \hat{A}, \hat{B}, X, \perp)$, and sends $(\Pi, R, \hat{B}, \hat{A}, X)$ to the session responder $\hat{B}$, who then prepares ephemeral public key Y and a new session identified with $(\Pi, R, \hat{B}, \hat{A}, X, Y)$, computes the session key and sends $(\Pi, I, \hat{A}, \hat{B}, X, Y)$ to $\hat{A}$. Upon receiving $(\Pi, I, \hat{A}, \hat{B}, X, Y)$, $\hat{A}$ updates the

session identifier $(\Pi, I, \hat{A}, \hat{B}, X, \perp)$ with $(\Pi, I, \hat{A}, \hat{B}, X, Y)$ and computes a session key for that session. We say that a session is completed if its owner computes a session key.

If $\hat{A}$ is the initiator of a session, the session is identified via $sid = (\Pi, I, \hat{A}, \hat{B}, X, \perp)$ or $sid = (\Pi, I, \hat{A}, \hat{B}, X, Y)$. If $\hat{B}$ is the responder of a session, the session is identified via $sid = (\Pi, R, \hat{B}, \hat{A}, X, Y)$. The matching session of the session identified via $(\Pi, I, \hat{A}, \hat{B}, X, Y)$ is a session with identifier $(\Pi, R, \hat{B}, \hat{A}, X, Y)$ and vice versa.

Adversary: Adversary $\mathcal{M}$ is modeled as a probabilistic Turing machine that controls all communications including session activation. Activation is performed via a $Send(MESSAGE)$ query. The message has one of the following forms: $(\Pi, I, \hat{A}, \hat{B})$, $(\Pi, R, \hat{A}, \hat{B}, X)$, or $(\Pi, I, \hat{A}, \hat{B}, X, \perp)$. Each party submits its responses to adversary $\mathcal{M}$, who decides the global delivery order. The secret information of a party is not accessible to adversary $\mathcal{M}$; however, leakage of secret information is obtained via the following adversary queries:

- $SessionKeyReveal(sid)$: $\mathcal{M}$ obtains the session key for the session with session identifier sid , provided that the session is completed.
- $SessionStateReveal(sid)$: $\mathcal{M}$ obtains the session state of the owner of session sid if the session is not completed (the session key is not established yet). The session state includes all ephemeral secret keys and intermediate computation results except for immediately erased information but does not include the static secret key.
- $Corrupt(\hat{A})$: The query allows $\mathcal{M}$ to obtain all information of party $\hat{A}$. If a party, $\hat{A}$, is corrupted by a $Corrupt(\hat{A})$ query issued by $\mathcal{M}$, then we call the party, $\hat{A}$, dishonest. If not, we call the party honest.

Definition 6.1: (Freshness) Let sid^* be the session identifier of a completed session, owned by an honest party $\hat{A}$ with an honest peer $\hat{B}$. If the matching session exists, then let $\overline{sid^*}$ be the session identifier of the matching session of sid^* . Define sid^* to be fresh if none of the following conditions hold:

- $\mathcal{M}$ issues $SessionKeyReveal(sid^*)$, or $SessionKeyReveal(\overline{sid^*})$ if $\overline{sid^*}$ exists.

- $\overline{sid^*}$ exists and $\mathcal{M}$ makes either of the following queries
 - $SessionStateReveal(sid^*)$ or $SessionStateReveal(\overline{sid^*})$,
- $\overline{sid^*}$ does not exist and $\mathcal{M}$ makes the following query
 - $SessionStateReveal(sid^*)$.

Security Experiment: Initially, adversary $\mathcal{M}$ is given a set of honest parties, for whom $\mathcal{M}$ selects identifiers. Then the adversary makes any sequence of the queries described above. During the experiment, $\mathcal{M}$ makes a special query $Test(sid^*)$, where sid^* is the session identifier of a fresh session, and is given with equal probability either the session key held by sid^* or a random key; the query does not terminate the experiment. The experiment continues until $\mathcal{M}$ makes a guess whether the key is random or not. The adversary *wins* the game if the test session sid^* is still fresh and if the guess by $\mathcal{M}$ was correct. The advantage of adversary $\mathcal{M}$ in the $AKAP$ experiment with $AKAP$ protocol Π is defined as

$$Adv_{\Pi}^{AKAP} = Pr[\mathcal{M} \text{ wins}] - \frac{1}{2} \quad (6.1)$$

Definition 6.2: (Post-quantum CK^+ security) We say that an $AKAP$ protocol Π is post-quantum secure in the CK^+ model if the following conditions hold:

- i) If two honest parties complete matching sessions, then, except with negligible probability, they both compute the same session key.
- ii) For any polynomial-time adversary $\mathcal{M}$, Adv_{Π}^{AKAP} is negligible in security parameter λ for the test session sid^* ,
 - if $\overline{sid^*}$ does not exist, and the static secret key of the owner of sid^* is given to $\mathcal{M}$,
 - if $\overline{sid^*}$ does not exist, and the static ephemeral key of the owner of sid^* is given to $\mathcal{M}$,
 - if $\overline{sid^*}$ exists, and the static secret key of the owner of sid^* and the static secret key of the owner of $\overline{sid^*}$ are given to $\mathcal{M}$,
 - if $\overline{sid^*}$ exists, and the ephemeral secret key of the owner of sid^* and the ephemeral secret key of the owner of $\overline{sid^*}$ are given to $\mathcal{M}$,

- if $\overline{sid^*}$ exists, and the static secret key of the owner of sid^* and the ephemeral secret key of the owner of $\overline{sid^*}$ are given to $\mathcal{M}$,
- if $\overline{sid^*}$ exists, and the ephemeral secret key of the owner of sid^* and the static secret key of the owner of $\overline{sid^*}$ are given to $\mathcal{M}$.

The CK^+ model not only covers the security requirement in CK model, but also captures some advanced attacks such as key compromise impersonation (KCI), the breaking of weak perfect forward secrecy ($wPFS$) and maximal exposure attacks (MEX).

We focus on two-party key exchange protocols whose ultimate goal is to securely derive a common and secret session key, even under the prospect of an active adversary. The most popular application of this mechanism is the establishment of a “secure-channel” by which the two parties can exchange messages securely using symmetric encryption under the derived session key. It can be argued that a “secure key exchange protocol” features the following three core requirements [31]:

- Authentication, which refers to the ability of each party in a key exchange session to verify the identity of the peer.
- Secrecy, which refers to the inability of any third party to learn any information about a session key established between two other parties.
- Consistency, which refers to the ability of each party in a key exchange session to know with certainty the identity of the peer with which a session key is established.

We give the main theorem and the security proof of our key exchange protocol under these assumption.

Theorem 6.1: Suppose that H is modeled as a random oracle and that the di-SI-GDH assumption hold for SI-MQV. Then the SI-MQV protocol is a post-quantum CK^+ secure authenticated key exchange protocol in the random oracle model.

In particular, for any AKE adversary $\mathcal{M}$ against the SI-MQV protocol that runs in time at most t , involves at most n honest parties and activate at most s sessions, and makes at most h queries to the random oracle, there exists a di-SI-GDH adversary $\mathcal{S}$ such that

$$Adv^{di-SI-GDH}(\mathcal{S}) \geq \min\left\{\frac{1}{sn}, \frac{1}{n^2}, \frac{1}{s^2}\right\} Adv_{SI-MQV}^{AKE}(\mathcal{M}) \quad (6.2)$$

where $\mathcal{S}$ runs in time t plus time to perform $\mathcal{O}((n+s)\lambda)$ low-degree isogeny operations and make $\mathcal{O}(h+s)$ queries to SI-DDH oracle.

Before starting the proof, we now give two propositions and three lemmas which will be used for the proof of the Theorem 6.1.

Proposition 6.1: Assume $\mathcal{H}$ is modeled as a quantum random oracle, adversary $\mathcal{M}$ has only three ways to distinguish a session key of the test session from a random string,

- *Guessing attack:* $\mathcal{M}$ correctly guesses the session key.
- *Key replication attack:* $\mathcal{M}$ creates a session that is not matching to the test session, but has the same session key as the test session.
- *Forging attack:* $\mathcal{M}$ computes Z used in the test session identified with $(\Pi, I, \hat{A}, \hat{B}, X, Y)$, and queries $\mathcal{H}$ with a superposition including $(\Pi, Z, \hat{A}, \hat{B}, X, Y)$.

Proof Since $\mathcal{H}$ is a quantum random oracle, the probability of guessing the output of $\mathcal{H}$ is $\mathcal{O}(1/2^\lambda)$. Since non-matching sessions have different communicating parties or ephemeral public keys, key replication is equivalent to finding $\mathcal{H}$ -collision; therefore the probability of succeeding key replication is $\mathcal{O}(s^2/2^\lambda)$ where s is the number of active sessions for SI-MQV. However to detect collision the adversary has to query with both inputs the random oracle, in particular query with Z used in the test session as described in Forging attack above. Let $\mathbf{M}$ be the event that $\mathcal{M}$ wins the security experiment with SI-MQV, $\mathbf{H}$ be the event that $\mathcal{M}$ succeeds forging attack, and $\overline{\mathbf{H}}$ the complementary event of $\mathbf{H}$. Thus we have $Pr[\mathbf{M}|\overline{\mathbf{H}}] = \frac{1}{2}$, and therefore

$$Adv_{SI-MQV}^{AKAP}(\mathcal{M}) = Pr[\mathbf{M}] - \frac{1}{2} \leq Pr[\mathbf{M} \cap \mathbf{H}] \quad (6.3)$$

By the definition of freshness in the CK^+ -model, there are six cases that $\mathcal{M}$ chooses a test session:

- E_1 : $\mathcal{M}$ chooses a test session without a matching session, and does not reveal the ephemeral secret key of the owner of the test session.

- E_2 : $\mathcal{M}$ chooses a test session without a matching session, and does not reveal the static secret key of the owner of the test session.
- E_3 : $\mathcal{M}$ chooses a test session with a matching session, and does not reveal the ephemeral secret key of both the owner of the test session and the owner of its matching session.
- E_4 : $\mathcal{M}$ chooses a test session with a matching session, and does not reveal the static secret keys of both the owner of the test session and the owner of its matching session.
- E_5 : $\mathcal{M}$ chooses a test session with a matching session, and does not reveal the ephemeral secret key of the owner of the test session and the static secret key of the peer of its matching session.
- E_6 : $\mathcal{M}$ chooses a test session with a matching session, and does not reveal the static secret key of the owner of the test session and the ephemeral secret key of the peer of its matching session.

In each case, we can show how to construct a di-SI-GDH solver $\mathcal{S}$. Solver $\mathcal{S}$ is given a SI-CDH instance $\text{publicparameters}(pk, U, V)$. Let R be an event that $\mathcal{M}$ chooses a test session whose owner and peer are the same party, and let $\bar{R}$ be its complement. Whether a certain event took place or not is decided at the end of the experiment. In other words for each event analysis below it is assumed that the event conditions are satisfied upon the adversary termination. Before analyzing the events, we note that the session state of a session in the SI-MQV protocol is equivalent to the ephemeral secret key in the session as no other information (except the static secret key) is necessary to compute the shared secrets and the session key. ■

Lemma 6.1: *There exists a di-SI-GDH adversary $\mathcal{S}$ such that*

$$\text{Adv}^{\text{di-SI-GDH}}(\mathcal{S}) \geq \frac{1}{sn} \text{Adv}_{\text{SI-MQV}}^{\text{AKE}}(\mathcal{M}). \quad (6.4)$$

Proof (Case $E_1 \cap \bar{R}$). $\mathcal{S}$ prepares n honest parties, selects one party $\hat{B}$ to whom $\mathcal{S}$ assigns the static public key $B_2 = V$ and a random static public and secret key pair for B_1 . The remaining $n-1$ parties are assigned random static public and secret key pairs. $\mathcal{S}$ selects $i \in_R \{1, \dots, s\}$, and chooses i -th session sid^* among sessions, activated by $\mathcal{M}$ and owned by an honest party different from $\hat{B}$. When $\mathcal{M}$ activates sessions

between honest peers, $\mathcal{S}$ follows the protocol description. Since $\mathcal{S}$ knows static secret keys of at least one peer, it can respond all queries faithfully. The only exception is the session sid^* , for which $\mathcal{S}$ sets ephemeral public key of sid^* to U , and chooses a random $\zeta \in \{0, 1\}^\lambda$ as the session key of sid^* .

The simulator has difficulty in responding queries related to $\hat{B}$ because $\mathcal{S}$ does not know one of the static secret keys of $\hat{B}$. More precisely, for sessions owned by $\hat{B}$ with a peer $\hat{C}$ controlled by $\mathcal{M}$, $\mathcal{S}$ cannot compute the shared secret, Z , but may have to answer *SessionKeyReveal* queries. $\mathcal{M}$ could also derive session keys of these session by computing the shared secret, Z , and query H . If the value does not coincide, then $\mathcal{S}$ fails its simulation. To handle this situations, $\mathcal{S}$ prepares R^{List} with entries of the form $(pid, uid, uid', W, W', SK) \in \chi \subset \{0, 1\}^* \times \{0, 1\}^* \times \{0, 1\}^* \times SSEC_{p,1} \times SSEC_{p,2} \times \{0, 1\}^\lambda$. and H^{List} with entries of the form $(pid, Z, uid, uid', epk, epk') \in \chi \subset \{0, 1\}^* \times \mathbb{F}_{p^2} \times \{0, 1\}^* \times \{0, 1\}^* \times SSEC_{p,1} \times SSEC_{p,2}$, where pid is a string which gives a protocol identifier, and uid is a string which gives a user identifier, and $\mathcal{S}$ maintains two lists for consistent responses to H and *SessionKeyReveal* queries as follows. Below, Y is generated by $\mathcal{S}$ on behalf of $\hat{B}$.

- $Send(\Pi, I, \hat{A}, \hat{B})$: Solver $\mathcal{S}$ selects uniformly random ephemeral secret key, computes ephemeral public key honestly, records $(\Pi, I, \hat{A}, \hat{B}, X)$ in List R^{List} , and returns it.
- $Send(\Pi, R, \hat{A}, \hat{B}, X)$: $\mathcal{S}$ selects uniformly random ephemeral secret key, computes ephemeral public key honestly, records $(\Pi, R, \hat{A}, \hat{B}, X, Y)$ in List R^{List} as completed, and returns it.
- $(\Pi, I, \hat{A}, \hat{B}, X, Y)$: If session $Send(\Pi, R, \hat{A}, \hat{B}, X)$ is not recorded in List R^{List} , $\mathcal{S}$ records session $(\Pi, I, \hat{A}, \hat{B}, X, Y)$ in List R^{List} as not completed. Otherwise, $\mathcal{S}$ records the session in List R^{List} as completed.
- $H(\cdot)$: $\mathcal{S}$ simulates a random oracle in the usual way except for queries of the form $(\Pi, Z, \hat{B}, \hat{C}, Y, X)$ and $(\Pi, Z, \hat{C}, \hat{B}, X, Y)$. When $(\Pi, Z, \hat{B}, \hat{C}, Y, X)$ is queried, $\mathcal{S}$ responds to these queries in the following way: (when $(\Pi, Z, \hat{C}, \hat{B}, X, Y)$ is queried, $\mathcal{S}$ responds in a similar way)
 - i) if $(\Pi, Z, \hat{B}, \hat{C}, Y, X, SK) \in H^{List}$ for some SK , $\mathcal{S}$ returns SK to $\mathcal{M}$.
 - ii) else if the validity condition $SI-DDH(B_1, C_2, X, Y, Z) = 1$ holds, then if there exists $(\Pi, I, \hat{B}, \hat{C}, Y, X, SK) \in R^{List}$ returns SK otherwise, $\mathcal{S}$

- chooses $SK \in_R \{0, 1\}^\lambda$, returns SK and stores $(\Pi, I, \hat{B}, \hat{C}, Y, X, SK) \in R^{List}$. $\mathcal{S}$ also stores the new tuple $(\Pi, Z, \hat{B}, \hat{C}, Y, X, SK) \in H^{List}$.
- iii) else $\mathcal{S}$ chooses $SK \in_R \{0, 1\}^\lambda$, returns it to $\mathcal{M}$ and stores the new tuple $(\Pi, Z, \hat{B}, \hat{C}, Y, X, SK) \in H^{List}$.
- *SessionKeyReveal(.)*: $\mathcal{S}$ simulates these queries in the usual way except for queries of the form $(\Pi, I, \hat{B}, \hat{C}, Y, X)$ and $(\Pi, R, \hat{C}, \hat{B}, X, Y)$. When $(\Pi, I, \hat{B}, \hat{C}, Y, X)$ is queried, $\mathcal{S}$ does one of the following: (when $(\Pi, R, \hat{C}, \hat{B}, X, Y)$ is queried, $\mathcal{S}$ responds in a similar way)
 - if there is no session with identifier $(\Pi, I, \hat{B}, \hat{C}, Y, X)$, the query is aborted.
 - else if $(\Pi, I, \hat{B}, \hat{C}, Y, X, SK) \in R^{List}$ for some SK , $\mathcal{S}$ returns SK to $\mathcal{M}$.
 - else if $(\Pi, Z, \hat{B}, \hat{C}, Y, X, SK) \in H^{List}$ such that $SI-DDH(B_1, C_2, X, Y, Z) = 1$, $\mathcal{S}$ returns SK and stores the new tuple $(\Pi, I, \hat{B}, \hat{C}, Y, X, SK) \in R^{List}$.
 - *SessionStateReveal(sid)*: If the corresponding ephemeral public key is U , then solver $\mathcal{S}$ aborts with failure. Otherwise, solver $\mathcal{S}$ responds to the query faithfully.
 - *Corrupt($\hat{A}$)*: If $\hat{A}$ is queried before, solver $\mathcal{S}$ returns error. Otherwise, solver $\mathcal{S}$ responds to the query faithfully.
 - *Test(sid)*: If sid is not the t -th session of $\hat{A}$, then solver $\mathcal{S}$ aborts with failure. Otherwise, solver $\mathcal{S}$ responds to the query faithfully.
 - If adversary $\mathcal{M}$ outputs guess ephemeral secret key, solver $\mathcal{S}$ aborts with failure.

Provided that $E_1 \cap \bar{R}$ occurs and $\mathcal{M}$ selects sid^* as the test session with peer $\hat{B}$, the simulation does not fail. In this case, the session identifier of sid^* is $(\Pi, I, \hat{A}, \hat{B}, U, Y)$, where Y is the incoming ephemeral public key of sid^* . If $\mathcal{M}$ wins the security game, it must have queried H with inputs $Z = SI-CDH(A_1, B_2, U, Y)$. To solve the $SI-CDH$ instance, $\mathcal{S}$ checks if there is an H query made by $\mathcal{M}$ of the form $(\Pi, Z, \hat{A}, \hat{B}, U, Y)$, such that $SI-DDH(A_1, B_2, U, Y) = 1$. If such an H query exists, $\mathcal{S}$ outputs Z . With probability at least $\frac{1}{sn}$ the test session is sid^* with peer $\hat{B}$. Thus the advantage of $\mathcal{S}$ is

$$Adv^{di-SI-GDH}(\mathcal{S}) \geq \frac{1}{sn} Pr[M \cap H \cap E_1 \cap \bar{R}] \quad (6.5)$$

Notice that in the above simulation $\mathcal{S}$ cannot respond to $\text{StaticKeyReveal}(\hat{B})$ query. However, given that event E_1 occurs, $\mathcal{S}$ correctly guesses the test session and the test session is fresh at the end of the experiment, then $\mathcal{M}$ have not queried for the static secret keys of the test session $\hat{B}$. Such static key reveal query would contradict the freshness of the test session and thus the simulation terminated without errors.

(Case $E_1 \cap R$). $\mathcal{S}$ prepares n honest parties, selects an honest party $\hat{A}$ and assigns its static public keys as $A_1 = U$ and $A_2 = V$. The remaining $n - 1$ parties are assigned random static and secret key pairs $\mathcal{S}$ simulates the environment of $\mathcal{M}$ by following the protocol description. The party $\hat{A}$ is simulated as $\hat{B}$ in $E_1 \cap \bar{R}$. If $\mathcal{M}$ selected a session whose owner and peer are the same party $\hat{A}$ as the test session, and $E_1 \cap R$ occurs, this simulation does not fail. Let $(\Pi, I, \hat{A}, \hat{A}, X, Y)$ be the session identifier of the test session. When $\mathcal{M}$ is successful, $\mathcal{S}$ checks if there is an H query made by $\mathcal{M}$ of the form $(\Pi, Z, \hat{A}, \hat{A}, X, Y)$, such that $\text{SI-DDH}(B_1, C_2, X, Y, Z) = 1$. If such an H query exists, $\mathcal{S}$ outputs Z . With probability at least $\frac{1}{n}$, $\mathcal{M}$ will select a test session whose owner and peer is the same party $\hat{A}$. Thus, the advantage of $\mathcal{S}$ is

$$\text{Adv}^{\text{di-SI-GDH}}(\mathcal{S}) \geq \frac{1}{n} \Pr[M \cap H \cap E_1 \cap R] \quad (6.6)$$

$\mathcal{S}$ cannot respond to $\text{StaticKeyReveal}(\hat{A})$ query during the simulation. As before if event E_1 occurs, $\mathcal{S}$ correctly guesses the test session and the test session is fresh at the end of the experiment, then $\mathcal{M}$ have not queried for the static secret keys of $\hat{A}$, and therefore the simulation does not terminate with error. ■

Lemma 6.2: *There exists a di-SI-GDH adversary $\mathcal{S}$ such that*

$$\text{Adv}^{\text{di-SI-GDH}}(\mathcal{S}) \geq \frac{1}{n^2} \text{Adv}_{\text{SI-MQV}}^{\text{AKE}}(\mathcal{M}) \quad (6.7)$$

Proof (Case $E_2 \cap \bar{R}$). $\mathcal{S}$ prepares n honest parties, selects two distinct honest parties $\hat{A}$ and $\hat{B}$, and assigns $\hat{A}$'s and $\hat{B}$'s static public keys as $A_1 = U$ and $B_2 = V$, respectively. $\mathcal{S}$ assigns random static public and secret key pairs for A_2, B_1 , and the remaining $n-2$ parties are assigned random static and secret key pairs. $\mathcal{S}$ follows the

protocol description when $\mathcal{M}$ activates session between honest peers, and simulate $\mathcal{S}$'s queries related to $\hat{A}$ or $\hat{B}$ as explained in E_1 .

If $\mathcal{M}$ selected a session whose participants are $\hat{A}$, $\hat{B}$ as the test session, and $E_2 \cap \bar{R}$ occurs, this simulation does not fail. Let $(\Pi, I, \hat{A}, \hat{B}, X, Y)$ be the session identifier of the test session. Note that $\mathcal{S}$ generated X and so knows x . When $\mathcal{M}$ is successful, $\mathcal{S}$ checks if there is an H query made by $\mathcal{M}$ of the form $(\Pi, Z, \hat{A}, \hat{B}, X, Y)$, such that $SI-DDH(A_1, B_2, X, Y) = 1$. If such an H query exists, $\mathcal{S}$ outputs as the $SI-CDH$ answer where $Z = SI-CDH(A_1, B_2) = SI-CDH(X, Y)$. With probability at least $\frac{1}{n^2}$, $\mathcal{M}$ will select a test session with owner $\hat{A}$ and peer $\hat{B}$, respectively. Thus, the advantage of $\mathcal{S}$ is

$$Adv^{di-SI-GDH}(\mathcal{S}) \geq \frac{1}{n^2} Pr[M \cap H \cap E_2 \cap \bar{R}] \quad (6.8)$$

$\mathcal{S}$ can respond to neither $StaticKeyReveal(\hat{A})$ nor $StaticKeyReveal(\hat{B})$ queries during the simulation. As before if event E_2 occurs, $\mathcal{S}$ correctly guesses the test session and the test session is fresh at the end of the experiment, then $\mathcal{S}$ have queried for the static secret key of neither $\hat{A}$ nor $\hat{B}$, and therefore the simulation does not terminate with error.

(Case $E_2 \cap R$). This case is essentially the same as $E_1 \cap R$. In this case, we can construct a $di-SI-GDH$ solver $\mathcal{S}$ with

$$Adv^{di-SI-GDH}(\mathcal{S}) \geq \frac{1}{n} Pr[M \cap H \cap E_2 \cap R] \quad (6.9)$$

■

Lemma 6.3: *There exists a $di-SI-GDH$ adversary $\mathcal{S}$ such that*

$$Adv^{di-SI-GDH}(\mathcal{S}) \geq \frac{1}{s^2} Adv_{SI-MQV}^{AKE}(\mathcal{M}) \quad (6.10)$$

Proof (Case E_3). $\mathcal{S}$ prepares n honest parties, and assigns random static public and secret key pairs for these parties. $\mathcal{S}$ selects $i, j \in_R \{1, \dots, s\}$, and chooses i -th session sid^* and j -th session sid^* among sessions activated by $\mathcal{S}$ and owned by honest

parties. When activated, $\mathcal{S}$ sets the ephemeral public key of sid^* to be U and of $\overline{\text{sid}^*}$ to be V . Since $\mathcal{S}$ knows the static secret keys of all honest parties, it can respond all queries, faithfully, except those that related to sid^* and $\overline{\text{sid}^*}$.

Provided that $\mathcal{M}$ selects sid^* as the test session, $\overline{\text{sid}^*}$ as its matching session, and E_3 occurs, the simulation does not fail. Let $(\Pi, I, \hat{A}, \hat{B}, U, Y)$ and $(\Pi, R, \hat{B}, \hat{A}, U, V)$ be the session identifiers of sid^* and $\overline{\text{sid}^*}$, respectively. When $\mathcal{S}$ wins the security game, $\mathcal{S}$ checks if there is an H query made by $\mathcal{M}$ of the form $(\Pi, Z, \hat{A}, \hat{B}, U, V)$, such that $\text{SI-DDH}(A_1, B_2, X, Y) = 1$. If such an H query exists, $\mathcal{S}$ outputs Z as the SI-CDH answer. With probability at least $\frac{1}{s^2}$, $\mathcal{M}$ selects sid^* as the test session and $\overline{\text{sid}^*}$ as its matching session. Thus the advantage of $\mathcal{S}$ is

$$\text{Adv}^{\text{di-SI-GDH}}(\mathcal{S}) \geq \frac{1}{s^2} \Pr[M \cap H \cap E_3] \quad (6.11)$$

$\mathcal{S}$ cannot respond to *SessionStateReveal* queries against the test session and its matching during the simulation. However, under event E_3 adversary does not issue such queries, and hence the simulation does not fail. ■

Proposition 6.2: The analysis of E_4 , E_5 , and E_6 is similar to E_2 , E_1 , and E_1 , respectively. In each case, we can construct a di-SI-GDH solver $\mathcal{S}$ as follows:

$$\text{Adv}^{\text{di-SI-GDH}}(\mathcal{S}) \geq \frac{1}{n^2} \Pr[M \cap H \cap E_4] \quad (6.12)$$

$$\text{Adv}^{\text{di-SI-GDH}}(\mathcal{S}) \geq \frac{1}{sn} \Pr[M \cap H \cap E_i \cap R] \quad (6.13)$$

$$\text{Adv}^{\text{di-SI-GDH}}(\mathcal{S}) \geq \frac{1}{n} \Pr[M \cap H \cap E_j \cap \bar{R}] \quad (6.14)$$

for $i = 5, 6$ and $j = 4, 5, 6$.

Proof (Proof of the Theorem 6.1) By using Proposition 6.1, 6.2 and Lemma 6.1, 6.2, 6.3, if we combine all inequalities we have,

$$Adv^{di-SI-GDH}(\mathcal{S}) \geq \min\{\frac{1}{sn}, \frac{1}{n^2}, \frac{1}{s^2}\} Adv_{SI-MQV}^{AKE}(\mathcal{M}) \quad (6.15)$$

*During the simulation, the solvers $\mathcal{S}$ and $\mathcal{M}$ perform $\mathcal{O}((n+s)\lambda)$ low-degree isogeny operations for assigning static and ephemeral keys, and make $\mathcal{O}(h+s)$ times SI-DDH oracle queries for simulating *SessionKeyReveal* and the random oracle H queries.*

■

7. NEW CRYPTANALYSIS OF SOME PROTOCOLS

7.1. K-CI Attack on SIDH variant of the Jeong-Katz-Lee protocol

In [21], Galbraith surveyed authenticated key exchange in the context of supersingular isogeny Diffie-Hellman key exchange (SIDH). One of the protocol in the paper (called by SIDH-AKE) is based on the Jeong, Katz and Lee [28]. We mentioned this protocol in the previous progress report, namely Protocol 1. Galbraith claimed that SIDH-AKE is a secure authenticated key exchange protocol. However, we proved that this protocol is vulnerable to KCI attack. We summarized the protocol and attack below.

- *Alice* calculates $R_A = P_A + [n_A]Q_A$ which has order 2^n
- *Bob* calculates $R_B = P_B + [n_B]Q_B$ which has order 3^m
- *Alice* computes the isogeny $\phi_A : E \rightarrow E_A$ whose kernel is R_A
- *Alice* chooses an integer $n_X \in (0, 2^n)$ and computes the isogeny $\phi_X : E \rightarrow E_X$ whose kernel is $R_X = P_A + n_X Q_A$
- *Bob* computes the isogeny $\phi_B : E \rightarrow E_B$ whose kernel is R_B
- *Bob* chooses an integer $n_Y \in (0, 3^m)$ and computes the isogeny $\phi_Y : E \rightarrow E_Y$ whose kernel is $R_Y = P_B + n_Y Q_B$
- *Alice* sends $E_A, E_X, \phi_A(P_B), \phi_A(Q_B), \phi_X(P_B), \phi_X(Q_B)$ to *Bob*
- *Bob* sends $E_B, E_Y, \phi_B(P_A), \phi_B(Q_A), \phi_Y(P_A), \phi_Y(Q_A)$ to *Alice*
- *Alice* controls that E_A and E_X are supersingular elliptic curves over $\mathbb{F}_{p^2}$
- *Bob* controls that E_B and E_Y are supersingular elliptic curves over $\mathbb{F}_{p^2}$
- *Alice* controls that are $\phi_A(P_B), \phi_A(Q_B)$ independent points on E_A and $\phi_X(P_B), \phi_X(Q_B)$ independent points on E_X of order 2^n
- *Bob* controls that are $\phi_B(P_A), \phi_B(Q_A)$ independent points on E_B and $\phi_Y(P_A), \phi_Y(Q_A)$ independent points on E_Y of order 3^m
- *Alice* computes $E_B \rightarrow E_{BA}$ whose kernel is the point $\phi_B(P_A) + [n_A]\phi_B(Q_A) = \phi_B(P_A + [n_A]Q_A) = \phi_B(R_A)$
- *Alice* computes $E_Y \rightarrow E_{YX}$ whose kernel is the point $\phi_Y(P_A) + [n_X]\phi_Y(Q_A) = \phi_Y(P_A + [n_X]Q_A) = \phi_Y(R_X)$

<i>Alice</i>	<i>Bob</i>
$n_A \in (0, 2^n), P_A, Q_A$	$n_B \in (0, 3^m), P_B, Q_B$
$R_A = P_A + n_A Q_A$	$R_B = P_B + n_B Q_B$
$n_X \in (0, 2^n)$	$n_Y \in (0, 3^m)$
$R_X = P_A + n_X Q_A$	$R_Y = P_B + n_Y Q_B$
$\phi_A : E \rightarrow E/\langle R_A \rangle := E_A$	$\phi_B : E \rightarrow E/\langle R_B \rangle := E_B$
$\phi_X : E \rightarrow E/\langle R_X \rangle := E_X$	$\phi_Y : E \rightarrow E/\langle R_Y \rangle := E_Y$
$E_A, E_X, \phi_A(P_B), \phi_A(Q_B), \phi_X(P_B), \phi_X(Q_B) \xrightarrow{\hspace{1.5cm}}$	
$E_B, E_Y, \phi_B(P_A), \phi_B(Q_A), \phi_Y(P_A), \phi_Y(Q_A) \xleftarrow{\hspace{1.5cm}}$	
$E_B \rightarrow \frac{E_B}{\langle \phi_B(R_A) \rangle} := E_{BA}$	$E_A \rightarrow \frac{E_A}{\langle \phi_A(R_B) \rangle} := E_{AB}$
$E_Y \rightarrow \frac{E_Y}{\langle \phi_Y(R_X) \rangle} := E_{YX}$	$E_X \rightarrow \frac{E_X}{\langle \phi_X(R_Y) \rangle} := E_{XY}$
$K = H(M, j(E_{BA}), j(E_{YX}))$	$K = H(M, j(E_{AB}), j(E_{XY}))$

Figure 7.1: SIDH variant of the Jeong-Katz-Lee protocol.

- *Bob* computes $E_A \rightarrow E_{AB}$ whose kernel is the point $\phi_A(P_B) + [n_B]\phi_A(Q_B) = \phi_A(P_B + [n_B]Q_B) = \phi_A(R_B)$
- *Bob* computes $E_X \rightarrow E_{XY}$ whose kernel is the point $\phi_X(P_B) + [n_Y]\phi_X(Q_B) = \phi_X(P_B + [n_Y]Q_B) = \phi_X(R_Y)$
- *Alice* computes the session key $K = H(M, j(E_{BA}), j(E_{YX}))$
- *Bob* computes the session key $K = H(M, j(E_{AB}), j(E_{XY}))$
- where $M = \text{Alice}, \text{Bob}, j(E_X), \phi_X(P_B), \phi_X(Q_B), j(E_Y), \phi_Y(P_A), \phi_Y(Q_A)$

Attack Scenario: Suppose Alice's long-term private key is disclosed. In this case, Eve can impersonate Bob to Alice.

- Alice calculates $R_A = P_A + [n_A]Q_A$ which has order 2^n
- Bob calculates $R_B = P_B + [n_B]Q_B$ which has order 3^m
- Alice computes the isogeny $\phi_A : E \rightarrow E_A$ whose kernel is R_A
- Alice chooses an integer $n_X \in (0, 2^n)$ and computes the isogeny $\phi_X : E \rightarrow E_X$ whose kernel is $R_X = P_A + n_X Q_A$
- Bob computes the isogeny $\phi_B : E \rightarrow E_B$ whose kernel is R_B
- Bob chooses an integer $n_Y \in (0, 3^m)$ and computes the isogeny $\phi_Y : E \rightarrow E_Y$ whose kernel is $R_Y = P_B + n_Y Q_B$
- Eve chooses an integer $n_{Y^*} \in (0, 3^m)$ and computes the isogeny $\phi_{Y^*} : E \rightarrow E_{Y^*}$ whose kernel is $R_{Y^*} = P_B + n_{Y^*} Q_B$
- Alice sends $E_A, E_X, \phi_A(P_B), \phi_A(Q_B), \phi_X(P_B), \phi_X(Q_B)$ to Bob, Eve interrupts the message
- Bob sends $E_B, E_Y, \phi_B(P_A), \phi_B(Q_A), \phi_Y(P_A), \phi_Y(Q_A)$ to Alice
- Eve interrupts the message and sends $E_B, E_{Y^*}, \phi_B(P_A), \phi_B(Q_A), \phi_{Y^*}(P_A), \phi_{Y^*}(Q_A)$ to Alice
- Alice computes $E_B \rightarrow E_{BA}$ whose kernel is the point $\phi_B(P_A) + [n_A]\phi_B(Q_A) = \phi_B(P_A + [n_A]Q_A) = \phi_B(R_A)$
- Alice computes $E_{Y^*} \rightarrow E_{Y^*X}$ whose kernel is the point $\phi_{Y^*}(P_A) + [n_X]\phi_{Y^*}(Q_A) = \phi_{Y^*}(P_A + [n_X]Q_A) = \phi_{Y^*}(R_X)$
- Eve has the static key of Alice, thus she can compute $E_A \rightarrow E_{AB}$ whose kernel is the point $\phi_A(R_B)$
- Eve computes $E_X \rightarrow E_{XY^*}$ whose kernel is the point $\phi_X(P_B) + [n_{Y^*}]\phi_X(Q_B) = \phi_X(P_B + [n_{Y^*}]Q_B) = \phi_X(R_{Y^*})$
- Alice and Eve compute the shared secret K

<i>Alice</i>	<i>Eve</i>	<i>Bob</i>
$n_A \in (0, 2^n), P_A, Q_A$		$n_B \in (0, 3^m), P_B, Q_B$
$R_A = P_A + n_A Q_A$		$R_B = P_B + n_B Q_B$
$n_X \in (0, 2^n)$	$n_{Y^*} \in (0, 3^m)$	$n_Y \in (0, 3^m)$
$R_X = P_A + n_X Q_A$	$R_{Y^*} = P_B + n_{Y^*} Q_B$	$R_Y = P_B + n_Y Q_B$
$\phi_A : E \rightarrow E / \langle R_A \rangle := E_A$		$\phi_B : E \rightarrow E / \langle R_B \rangle := E_B$
$\phi_X : E \rightarrow E / \langle R_X \rangle := E_X$	$\phi_{Y^*} : E \rightarrow E / \langle R_{Y^*} \rangle := E_{Y^*}$	$\phi_Y : E \rightarrow E / \langle R_Y \rangle := E_Y$
$E_A, E_X, \phi_A(P_B), \phi_A(Q_B), \phi_X(P_B), \phi_X(Q_B)$		
$E_B, E_Y, \phi_B(P_A), \phi_B(Q_A), \phi_Y(P_A), \phi_Y(Q_A)$		
$E_B, E_{Y^*}, \phi_B(P_A), \phi_B(Q_A), \phi_{Y^*}(P_A), \phi_{Y^*}(Q_A)$		
$E_B \rightarrow \frac{E_B}{\langle \phi_B(R_A) \rangle} := E_{BA}$	$E_B \rightarrow \frac{E_B}{\langle \phi_B(R_A) \rangle} := E_{BA}$	
$E_{Y^*} \rightarrow \frac{E_{Y^*}}{\langle \phi_{Y^*}(R_X) \rangle} := E_{Y^*X}$	$E_X \rightarrow \frac{E_X}{\langle \phi_X(R_{Y^*}) \rangle} := E_{XY^*}$	
$K = H(M', j(E_{BA}), j(E_{Y^*X}))$	$K = H(M', j(E_{BA}), j(E_{XY^*}))$	

Figure 7.2: K-CI attack scenario of Galbraith's protocol.

7.2. UK-S Attack on SIDH variant of MTI-A0

In this protocol, Alice and Bob shared secret using two isogeny computations. The protocol provide forward secrecy and authentication. However, it is not resistant to UK-S attack.

(UKS) *unknown key-share*: Bob cannot be coerced into sharing a key with Alice without Bob's knowledge, i.e., when Bob believes the key is shared with Eve, and Alice (correctly) believes the key is shared with Bob. The attack on the protocol is summarized in Figure 7.4.

Outputs of the research on post-quantum protocols are also summarized in Table 7.1.

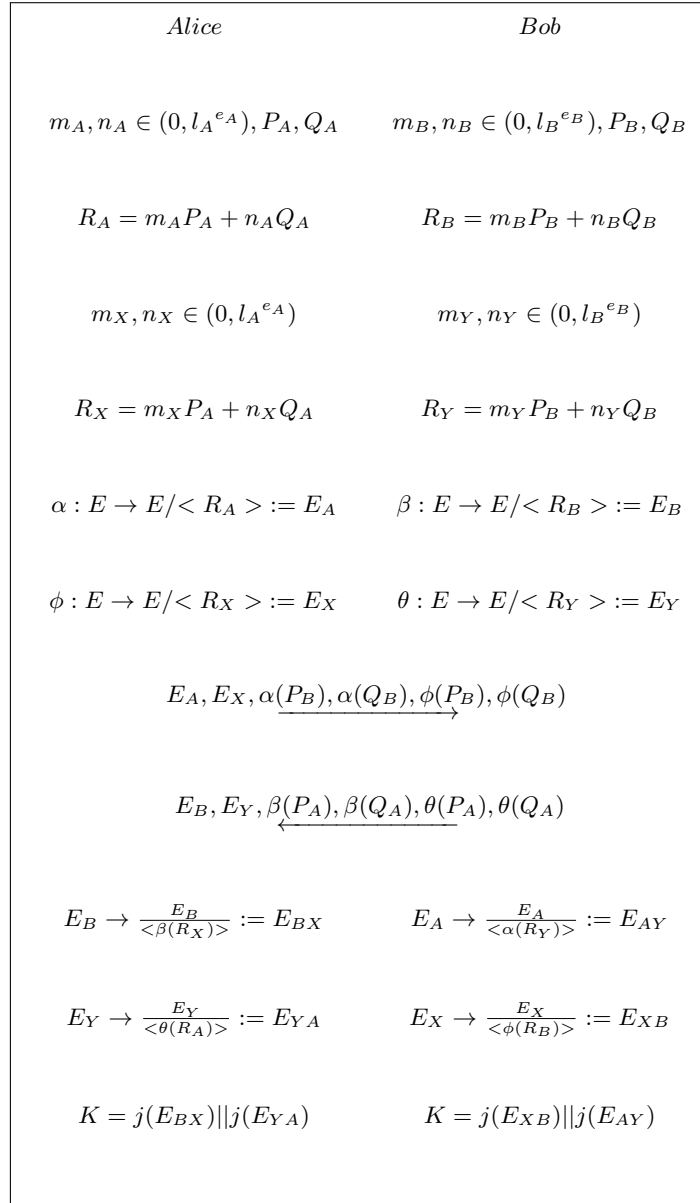


Figure 7.3: SIDH Variant of MTI-A0.

Table 7.1: Security services offered by post quantum key agreement protocols.

Protocols	Method	Sec. M.	Assump.	Signature	# of Pass	In. Isogeny	Res. Isogeny	(w)FS	MEX	K-CI	UK-S	M. Size
<i>SIDH</i> [17]	Key Agr.	-	DSIDH	x	2	2	2	x	x	x	✓	72λ
Jeong-Katz-Lee [21]	Key Agr.	CK	CSIDH	x	2	3	3	✓	x	x	✓	108λ
Galbraith [21]	Key Agr.	BR	CSIDH	✓	2	4	4	✓	x	✓	✓	108λ
$AK E_{SIDH-2}$ [47]	Encap.	CK^+	DSIDH	x	2	6	5	✓	✓	✓	?	114λ
$AK E_{SIDH-3}$ [47]	Encap.	CK^+	1-O SIDH	x	3	5	5	✓	✓	✓	?	80λ
Longa [34]	Encap.	CK	DSIDH	x	2	6	6	✓	x	✓	✓	148λ
SIDH UM [20]	Key Agr.	CK	DSIDH	x	2	4	4	✓	x	x	✓	108λ
SIDH Biclique [20]	Key Agr.	CK^+	GSIDH	x	2	6	6	✓	✓	✓	✓	108λ
<i>GSW</i> [12]	Encap.	CK	DSIDH	x	3	6	6	✓	x	x	✓	186λ
<i>STKE</i>	Encap.	CK^+	DSIDH	x	3	6	5	✓	✓	✓	✓	114λ
SI-MQV	Key Agr.	CK^+	DSIDH	x	2	4	4	✓	✓	✓	✓	144λ

<i>Alice</i>	<i>Eve</i>	<i>Bob</i>
$m_A, n_A \in (0, l_A^{e_A}), P_A, Q_A$	$P_C = m_C P_A, Q_C = n_C Q_A$	$m_B, n_B \in (0, l_B^{e_B}), P_B, Q_B$
$R_A = m_A P_A + n_A Q_A$	$R_C = P_C + Q_C$	$R_B = m_B P_B + n_B Q_B$
$m_X, n_X \in (0, l_A^{e_A})$		$m_Y, n_Y \in (0, l_B^{e_B})$
$R_X = m_X P_A + n_X Q_A$		$R_Y = m_Y P_B + n_Y Q_B$
$\alpha : E \rightarrow E / \langle R_A \rangle := E_A$		$\beta : E \rightarrow E / \langle R_B \rangle := E_B$
$\phi : E \rightarrow E / \langle R_X \rangle := E_X$		$\theta : E \rightarrow E / \langle R_Y \rangle := E_Y$
$E_A, E_X, \alpha(P_B), \underline{\alpha(Q_B)}, \phi(P_B), \phi(Q_B)$		
$E_B, E_Y, \beta(P_C), \underline{\beta(Q_C)}, \theta(P_C), \theta(Q_C)$		
$\gamma : E_Y \rightarrow E_Y / \langle \theta(R_C) \rangle := E_{YC}$		
$\kappa : E_A \rightarrow E_A / \langle \alpha(R_C) \rangle := E_{AC}$		
$E_B, E_{YC}, \beta(P_A), \underline{\beta(Q_A)}, \gamma(\theta(P_A)), \gamma(\theta(Q_A))$		
$E_{AC}, E_X, \kappa(\alpha(P_B)), \underline{\kappa(\alpha(Q_B))}, \phi(P_B), \phi(Q_B)$		
$E_B \rightarrow \frac{E_B}{\langle \beta(R_X) \rangle} := E_{BX}$		$E_X \rightarrow \frac{E_X}{\langle \phi(R_B) \rangle} := E_{XB}$
$E_{YC} \rightarrow \frac{E_{YC}}{\langle \gamma(\theta(R_A)) \rangle} := E_{YCA}$		$E_{AC} \rightarrow \frac{E_{AC}}{\langle \kappa(\alpha(R_Y)) \rangle} := E_{ACY}$
$K = j(E_{BX}) j(E_{YCA})$		$K = j(E_{XB}) j(E_{ACY})$

Figure 7.4: UK-S attack scenario.

8. CONCLUSION AND REMARKS

The first purpose of this thesis is proposing a new and efficient two pass authenticated key agreement protocol for pre-quantum cryptography. The protocol is based on Diffie-Hellman key agreement protocol and provide high performance and many desirable security properties. In the literature, there is a deficiency of key agreement protocol which has a high performance and high security level at the same time. Therefore, designing a key agreement protocol with useful properties contributes the efficiency and security level of the existing cryptosystems. A new one is given in this thesis.

The second purpose of the thesis is analyzing and improving the isogeny based cryptosystems. We have analyzed some recent isogeny based key agreement protocols in the context of key manipulation attack. As a result of our attack, we have recovered the secret dual isogenies in these protocols and derived the session key. Moreover, we have applied KCI attack and UKS attack to the protocols in [21] and SIDH variant of MTI-A0, respectively.

Especially, supersingular isogeny based key agreement protocol (SIDH) has been drawing close attention in recent times. There are some studies about security and efficiency of the SIDH protocol [8], [17], [23]. However, security level of the SIDH protocol is not completely analyzed. Therefore, proving security model of the SIDH protocol is essential and interesting topic for the literature. SIDH protocol does not provide all security properties. Proof of possession is a main problem for these type of protocols. Hence, designing a new post-quantum key agreement algorithm which resists all cryptographic attacks is an open problem in cryptography. We designed a new key agreement protocol, namely SI-MQV and we proved that this protocol is secure and it resists to all known cryptographic attacks.

REFERENCES

- [1] Barker E. B., Johnson D., Smid M. E., (2007), “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography (Revised) SP 800-56A”, Technical Report, Gaithersburg, MD, United States.
- [2] Blake-Wilson S., Menezes A., (1999), “Authenticated Diffie-Hellman Key Agreement Protocols”, In Proceedings of SAC, SAC 98, 339–361.
- [3] Canetti R., Krawczyk H., (2001), “Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels”, In Pfitzmann, B., editor, Advances in Cryptology - EUROCRYPT 2001, Austria, May 6-10, 2001, Proceeding, volume 2045 of Lecture Notes in Computer Science, 453–474.
- [4] Canetti R., Krawczyk H., (2002), “Security Analysis of IKE’s Signature-Based Key-Exchange Protocol”, In Yung, M., editor, Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings, volume 2442 of Lecture Notes in Computer Science, 143–161.
- [5] Castryck W., Lange T., Martindale C., Panny L., Renes J., (2018), “CSIDH: An Efficient Post-Quantum Commutative Group Action”, In Advances in Cryptology – ASIACRYPT 2018, volume 11274 of Lecture Notes in Computer Science, 395–427.
- [6] Childs A. M., Jao D., Soukharev V., (2014), “Constructing Elliptic Curve Isogenies in Quantum Subexponential Time”, J. Mathematical Cryptology, 8(1), 1–29.
- [7] Costello C., (2020), “B-SIDH: Supersingular Isogeny Diffie-Hellman Using Twisted Torsion”, In Moriai, S. and Wang, H., editors, Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part II, volume 12492 of Lecture Notes in Computer Science, 440–463.
- [8] Costello C., Longa P., Naehrig, M. (2016), “Efficient Algorithms for Supersingular Isogeny Diffie-Hellman”, In Advances in Cryptology - CRYPTO 2016 - USA, August 14-18, 2016, Proceedings, Part I, 572–601.
- [9] Couveignes J. M., (2006), “Hard Homogeneous Spaces”, Cryptology ePrint Archive, Report 2006/291.
- [10] Daghigh H., Khodakaramian Gilan R., Seifi Shahpar F., (2017), “Diffie-Hellman Type Key Exchange Protocols Based on Isogenies”, Bulletin of the Iranian Mathematical Society, 43(4), 77–88.
- [11] De Feo L., Galbraith S. D., (2019), “SeaSign: Compact Isogeny Signatures From Class Group Actions”, In Ishai, Y. and Rijmen, V., editors, Advances in Cryptology – EUROCRYPT 2019, 759–789.

- [12] de Saint Guilhem C., Smart N. P., Warinski B., (2017), “Generic Forward-Secure Key Agreement without Signatures”, In Information Security - 20th International Conference, ISC 2017, Ho Chi Minh City, Vietnam, November 22-24, 2017, Proceedings, 114–133.
- [13] Delpech de Saint Guilhem C., Kutas P., Petit C., Silva J., (2019), “SÉTA: Supersingular Encryption From Torsion Attacks”, IACR Cryptol. ePrint Arch., 2019, 1291.
- [14] Developers T. S., (2016), “SageMath, the Sage Mathematics Software System (Version 7.4.1)”.
- [15] Diffie W., Hellman M. E., (1976), “New Directions in Cryptography”, IEEE Trans. Information Theory, 22(6), 644–654.
- [16] Diffie W., van Oorschot P. C., Wiener M. J., (1992), “Authentication and Authenticated Key Exchanges”, Des. Codes Cryptography, 2(2), 107–125.
- [17] Feo L. D., Jao D., Plût J., (2014), “Towards Quantum-Resistant Cryptosystems From Supersingular Elliptic Curve Isogenies”, J. Mathematical Cryptology, 8(3), 209–247.
- [18] Feo L. D., Kohel D., Leroux A., Petit C., Wesolowski B., (2020), “SQISign: Compact Post-Quantum Signatures From Quaternions and Isogenies”, In Moriai, S. and Wang, H., editors, Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part I, volume 12491 of Lecture Notes in Computer Science, 64–93.
- [19] Fujioka A., Suzuki K., Xagawa K., Yoneyama K., (2013), “Practical and Post-Quantum Authenticated Key Exchange From One-Way Secure Key Encapsulation Mechanism”, In ASIA CCS 2013.
- [20] Fujioka A., Takashima K., Terada S., Yoneyama K., (2018), “Supersingular Isogeny Diffie-Hellman Authenticated Key Exchange”, Cryptology ePrint Archive, Report 2018/730.
- [21] Galbraith S. D., (2018), “Authenticated Key Exchange for SIDH”, Cryptology ePrint Archive, Report 2018/266.
- [22] Galbraith S. D., Hess F., Smart N. P., (2002), “Extending the GHS Weil Descent Attack”, In Knudsen, L. R., editor, Advances in Cryptology - EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 - May 2, 2002, Proceedings, volume 2332 of Lecture Notes in Computer Science, 29–44.
- [23] Galbraith S. D., Petit C., Shani B., Ti Y. B., (2016), “On the Security of Supersingular Isogeny Cryptosystems”, In Advances in Cryptology - ASIACRYPT 2016, Vietnam, December 4-8, 2016, Proceedings, Part I, 63–91.
- [24] Gamal T. E., (1984), “A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms”, In Advances in Cryptology, Proceedings of CRYPTO ’84, USA, August 19-22, 1984, Proceedings, 10–18.

- [25] Gönen M., Tutdere S., Uzunkol O., (2022), “A Key Manipulation Attack on Some Recent Isogeny-Based Key Agreement Protocols”, *Bull. Iran. Math. Soc.* 48, 613–625.
- [26] Housley R., Yee P. E., Nace W. (2000), “Encryption Using KEA and Skipjack”, *RFC*, 2773, 1–9.
- [27] Jao D., Azarderakhsh R., Campagna M., Costello C., Feo L. D., Hess B., Jalali A., Koziel B., LaMacchia B., Longa P., Naehrig M., Renes J., Soukharev V., Urbanik D., (2016), “SIKE (Supersingular Isogeny Key Encapsulation)”, Submission to NIST Post-Quantum Cryptography Standardization.
- [28] Jeong I. R., Katz J., Lee D. H., (2004), “One-Round Protocols for Two-Party Authenticated Key Exchange”, *Applied Cryptography and Network Security*, 220–232.
- [29] Kerry C. F., Secretary A., Director C. R., (2013), “FIPS Publication 186-4 Digital Signature Standard (DSS)”.
- [30] Koblitz N., (1987), “Elliptic Curve Cryptosystems”, *Mathematics of Computation*, 48(177), 203–209.
- [31] Krawczyk H., (2003), “SIGMA: The ‘Sign-and-Mac’ Approach to Authenticated Diffie-Hellman and Its Use in the IKE-Protocols”, In Boneh, D., editor, *Advances in Cryptology - CRYPTO 2003*, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17–21, 2003, Proceedings, volume 2729 of *Lecture Notes in Computer Science*, 400–425, Springer.
- [32] Krawczyk H., (2005), “HMQV: High-Performance Secure Diffie-Hellman Protocol”, In *Advances in Cryptology - CRYPTO 2005*: California, USA, August 14–18, 2005, Proceedings, 546–566.
- [33] Law L., Menezes A., Qu M., Solinas J. A., Vanstone S. A., (2003), “An Efficient Protocol for Authenticated Key Agreement”, *Design, Codes Cryptography*, 28(2), 119–134.
- [34] Longa P., (2018), “A Note on Post-Quantum Authenticated Key Exchange From Supersingular Isogenies”, *IACR Cryptology ePrint Archive*, 2018, 267.
- [35] Matsumoto T., Takashima Y., Imai H., (1986), “On Seeking Smart Public-Key-Distribution Systems”, *Transactions of the Institute of Electronics and Communication Engineers of Japan, Section E*, 69, 99–106.
- [36] Menezes A., Okamoto T., Vanstone S. A., (1993). “Reducing Elliptic Curve Logarithms to Logarithms in a Finite Field”, *IEEE Trans. Inf. Theory*, 39, 1639–1646.
- [37] Miller V. S., (1986), “Use of Elliptic Curves in Cryptography”, In *Lecture Notes in Computer Sciences; 218 on Advances in cryptology—CRYPTO 85*, 417–426, New York, NY, USA, Springer-Verlag New York.
- [38] Moriya T., Onuki H., Takagi T., (2020), “SIGamal: A Supersingular Isogeny-Based PKE and Its Application to a PRF”, In Moriai, S. and Wang, H., editors, *Advances in Cryptology, ASIACRYPT 2020*, 551–580.

- [39] Rostovtsev A., Stolbunov A., (2006), “Public-Key Cryptosystem Based on Isogenies”, IACR Cryptology ePrint Archive, 2006, 145.
- [40] Shor P.W., (1994), “Algorithms for Quantum Computation: Discrete Logarithms and Factoring”, In 35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, 20-22 November 1994, 124–134.
- [41] Silverman J., (2009), “The Arithmetic of Elliptic Curves. Graduate Texts in Mathematics”, 5th Edition, Springer New York.
- [42] Tani S., (2009), “Claw Finding Algorithms Using Quantum Walk”, Theor. Comput. Sci., 410, 5285–5297.
- [43] Tate J., (1966), “Endomorphisms of Abelian Varieties Over Finite Fields”, Invent. Math., 2, 134–144.
- [44] Ti Y. B., (2017), “Fault Attack on Supersingular Isogeny Cryptosystems”, Post-Quantum Cryptography - 8th International Workshop, PQCrypto 2017, Utrecht, The Netherlands, June 26-28, 2017, Proceedings, volume 10346 of Lecture Notes in Computer Science, 107–122.
- [45] Velu J., (1971), “Isogenies Entre Courbes Elliptiques”, C. R. Acad. Sci. Paris Ser., (A-B)273, A238–A241.
- [46] Washington L. C., (2003), 2nd Edition, Elliptic Curves: Number Theory and Cryptography.
- [47] Xu X., Xue H., Wang K., Liang B., Tian S., Yu W., (2018), “Strongly Secure Authenticated Key Exchange From Supersingular Isogeny”, Cryptology ePrint Archive, Report 2018/760.

BIOGRAPHY

Mehmet Emin Gönen graduated from the Mathematics Department of Boğaziçi University in 2011. He worked as a research assistant at Bahçeşehir University between 2011-2013, and he has got his master's degree from the Applied Mathematics Department of the same university. In 2014, he started his doctoral research in Gebze Technical University.

In 2013, he started working as a researcher in Cryptographical Architecture and Algorithms Department of TUBİTAK BİLGEM. He has worked on the cryptographic algorithm and protocol design in many projects in the field of data security and cryptography. He works as a Chief Researcher and Project Manager in BİLGEM.

APPENDICES

APPENDIX A: PUBLICATIONS

Gönen M., Tutdere S., Uzunkol O., (2022), “A Key Manipulation Attack on Some Recent Isogeny-Based Key Agreement Protocols”, Bull. Iran. Math. Soc. 48, 613–625.

APPENDIX B: SOURCE CODES

In this section, we give the source codes of the key manipulation attack and the details of the SI-MQV protocol in SAGE.

Table B1.1: SAGE Source Codes

```
-----Key Manipulation-----
Example 5.1
p = 503
F.<w>=GF(p^2,modulus=x^2+x+1)
E = EllipticCurve(F,[0,0,0,0,-1])
print E
G = E(496, 320*w + 160)
print G
P = E(258*w + 158, 210*w + 90)
print P
Eprime=E.isogeny_codomain(G)
phi=E.isogeny(G)
phiA_P=Eprime(163*w +197, 81*w + 224)
8*phiA_P
phiB_P=Eprime(359*w +317, 88*w + 356)
8*phiB_P
Efinal=E.isogeny_codomain(P)
K=Efinal.j_invariant()
print K

Elliptic Curve defined by y^2 = x^3 + 502
over Finite Field in w of size 503^2
G=(496 : 320*w + 160 : 1)
P=(258*w + 158 : 210*w + 90 : 1)
(279 : 466 : 1)
(175 : 488 : 1)
K=179*w + 490
```

Table B1.1: Devam

<pre> -----Example 5.2----- p=2^8*3^6*7-1 p p.is_prime() F.<w>=GF(p^2,modulus=x^2+1) E = EllipticCurve(F,[0,0,0,0,-1]) n=E.order() G=E.random_element() l=G.order() e=n/l P=E.random_element() P.order() Eprime=E.isogeny_codomain(G) Eprime Efinal=E.isogeny_codomain(P) K=Efinal.j_invariant() -----Key Generation----- def generate_public(E,order,l_e): G=E(0) while G.order()!=l_e: G=E.random_element() G=Integer(order/(l_e)^2)*G return G def generate_secret(E,order,l_e): P=E(0) Q=E(0) while P.weil_pairing(Q,l_e)==1: P=generate_public(E,order,l_e) Q=generate_public(E,order,l_e) m=0 n=0 while (m.mod(Integer(factor(l_e)[0][0]))==0 and n.mod(Integer(factor(l_e)[0][0]))==0:) m=Integer(randint(0,l_e)) n=Integer(randint(0,l_e)) R=m*P+n*Q return (P,Q,m,n,R) </pre>

Table B1.1: Devam

```
def control_point(phi,P,Q,lA,lB):
    if P.weil_pairing(Q, lA) ==

        phi(P).weil_pairing(phi(Q), lB)
        return 1
    return 0
```

-----SIDH-----

```
def SIDH(E,la_ea,lb_eb):
    order=E.order()
    temp=generate_secret(E,order,la_ea)
    R_A=temp[4]
    print R_A
    R_B=generate_secret(E,order,lb_eb)[4]
    print R_B
    phiA=E.isogeny(R_A)
    print phiA
    phiB=E.isogeny(R_B)
    print phiB
    R_AB=phiB(R_A)
    print R_AB
    R_BA=phiA(R_B)
    print R_BA
    E_A=phiA.codomain()
    E_B=phiB.codomain()
    E_BA=E_B.isogeny_codomain(R_AB)
    print E_BA
    E_AB=E_A.isogeny_codomain(R_BA)
    print E_AB
    print "key1"
    print E_BA.j_invariant()
    print "key2"
    print E_AB.j_invariant()

    return (E_BA.j_invariant(),temp[0],temp[1])
%time SIDH()
```

Table B1.1: Devam

<pre> -----SI-MQV----- def SI-MQV(E, la_ea, lb_eb) : order=E.order() temp1=generate_secret(E, order, la_ea) R_A=temp1[4] print "R_A" print R_A temp2=generate_secret(E, order, la_ea) R_X=temp2[4] print "R_X" print R_X temp3=generate_secret(E, order, lb_eb) R_B=temp3[4] print "R_B" print R_B temp4=generate_secret(E, order, lb_eb) R_Y=temp4[4] print "R_Y" print R_Y phiA=E.isogeny(R_A) print "phiA" print phiA phiB=E.isogeny(R_B) print "phiB" print phiB R_AX=phiA(R_X) print "R_AX" print R_AX E_A=phiA.codomain() print "E_A" print E_A E_AX=E_A.isogeny_codomain(R_AX) print "E_AX" print E_AX phiAX=E_A.isogeny(R_AX) R_BY=phiB(R_Y) print "R_BY" print R_BY E_B=phiB.codomain() print "E_B" print E_B E_BY=E_B.isogeny_codomain(R_BY) print "E_BY" print E_BY phiBY=E_B.isogeny(R_BY) </pre>
--

Table B1.1: Devam

```
E_BYAX=E_BY.isogeny_codomain

([phiBY(phiB(R_X)),phiBY(phiB(R_A))])
print "E_BYAX"
print E_BYAX
E_AXBY=E_AX.isogeny_codomain

([phiAX(phiA(R_Y)),phiAX(phiA(R_B))])
print "E_AXBY"
print E_AXBY
print "key1"
print E_BYAX.j_invariant()
print "key2"
print E_AXBY.j_invariant()
return (E_AXBY.j_invariant())

%time SI-MQV()
```