Advances in Mathematics of Communications
Volume 4, No. 2, 2010, 215–235
doi:10.3934/amc.2010.4.215

CONSTRUCTING PUBLIC-KEY CRYPTOGRAPHIC SCHEMES BASED ON CLASS GROUP ACTION ON A SET OF ISOGENOUS ELLIPTIC CURVES

Anton Stolbunov

Department of Telematics
Norwegian University of Science and Technology
O.S. Bragstads plass 2a, N-7491 Trondheim, Norway

(Communicated by Henk van Tilborg)

Abstract. We propose a public-key encryption scheme and key agreement protocols based on a group action on a set. We construct an implementation of these schemes for the action of the class group $\mathcal{CL}(\mathcal{O}_K)$ of an imaginary quadratic field K on the set $\mathcal{ELL}_{p,n}(\mathcal{O}_K)$ of isomorphism classes of elliptic curves over $\mathbb{F}_p$ with n points and the endomorphism ring $\mathcal{O}_K$. This introduces a novel way of using elliptic curves for constructing asymmetric cryptography.

1. Introduction

The two most practical mathematical problems constituting the basis for security of modern asymmetric cryptographic schemes are integer factoring and computing discrete logarithms. Security of the former problem has decreased fast as new factoring methods and computer technology are developed. The latter problem remains exponential-time for some groups, e.g. elliptic curves. However, the Shor's algorithm can solve factoring and discrete logarithm problems in polynomial time when sufficiently large quantum computer registers become available [38]. These facts put a need for the development of asymmetric schemes that are based on *new* hard computational problems.

A potential mathematical object for this purpose is a low degree isogeny graph of ordinary elliptic curves over a finite field. Vertices in this graph are elliptic curves and edges are morphisms between them. Among the popular applications of low degree isogenies are the elliptic curve point counting, e.g. the Schoof-Elkies-Atkin (SEA) algorithm [37], reduction of the elliptic curve discrete logarithm problem (ECDLP) between different elliptic curves [19, 27], computation of the endomorphism ring of an elliptic curve [28] and computation of modular polynomials [6, 11]. Galbraith [18] and Galbraith, Hess and Smart [19] have proposed algorithms for constructing an isogeny between two given elliptic curves, i.e. searching for a route on an isogeny graph. Elliptic curve isogeny graphs have also been proposed for building cryptographic primitives. Rostovtsev et al. [33] have described an ordered digital signature scheme that implements the sequence number functionality for digitally signed documents using a small degree isogeny sequence. Teske [44] has constructed a key escrow system where a curve isogenous to the public curve is stored at a trusted authority and can be used to feasibly solve the ECDLP on

2000 *Mathematics Subject Classification:* 94A60, 14G50.
Key words and phrases: Public-key cryptography, group action, elliptic curve, isogeny, ideal class group, finite field.

the public curve, if needed. Charles, Goren and Lauter [12] have designed a hash function based on an isogeny graph of supersingular elliptic curves.

In this paper we use elliptic curve isogeny graphs for constructing new asymmetric cryptographic schemes. First we generalize some existing cryptographic schemes to the context of a group action on a set, and discuss their security. We then apply results of the complex multiplication theory to implement the proposed cryptographic schemes. Namely we use the action of the ideal class group $\mathcal{CL}(\mathcal{O}_K)$ of an imaginary quadratic field K on the set $\mathcal{ELL}_{p,n}(\mathcal{O}_K)$ of isomorphism classes of elliptic curves over $\mathbb{F}_p$ with n points and the endomorphism ring $\mathcal{O}_K$. The involved implementation-related solutions are then explained in more detail. We take advantage of available computational algorithms on elliptic curves and ideal class groups to implement the necessary operations. Finally we present our experimental results. Besides being interesting from the theoretical point of view, the proposed cryptographic schemes might also have an advantage against quantum computer attacks. However this question requires a further research, as we only point at the inapplicability of some currently known quantum algorithms.

This paper develops ideas of Rostovtsev and Stolbunov [34] originally appeared in their draft article. Independently of this work, research on a similar topic has been reported by Couveignes [14].

2. Public-key cryptography based on group action

2.1. Notation. We start with the basic notation. Let G be a finite abelian group, and X a set. A (left) action of G on X is a map

$$\begin{aligned} G \times X &\to X \\ (g, x) &\mapsto g * x, \end{aligned}$$

which satisfies the associativity property $(gh) * x = g * (h * x)$ for all $g, h \in G$, $x \in X$, and the property $e * x = x$ for the identity element $e \in G$ and all $x \in X$. The orbit of a set element $x \in X$ is the subset $G * x = \{g * x \colon g \in G\}$. The orbits of the elements of X are equivalence classes.

By $a \leftarrow b$ we denote the assignment of value b to a variable a. By $a \xleftarrow{R} G$ we mean that a is sampled from the uniform distribution on the set of elements of G. We write $\#S$ for the number of elements in S. By $\log n$ we denote the binary logarithm of n.

As a general rule, we assume that all algorithms take descriptions of G and X, and an element $x \in X$, as part of implied *system parameters*. By descriptions of G and X we mean the information needed, besides the input, to implement the operations involved in the algorithm, such as the random sampling from G, the action of G on X, the group operation etc.

2.2. Key agreement protocols based on group action. We present three key agreement protocols based on the action of G on X. A key agreement protocol $\mathcal{KA}1$ pictured on Fig. 1 is a generalization of the Diffie-Hellman key agreement. The $\mathcal{KA}1$ protocol has been proposed by Monico [31]. By $\mathcal{A}$ and $\mathcal{B}$ we denote the algorithms run by the participants Alice and Bob, respectively.

Due to the commutativity of G and the associativity of the action, the following holds:

$$k_A = a * \beta = a * (b * x) = (ab) * x = (ba) * x = b * (a * x) = b * \alpha = k_B,$$

$\mathcal{A}$		$\mathcal{B}$
Input: $-$		**Input:** $-$
$a \xleftarrow{R} G$		$b \xleftarrow{R} G$
$m_A \leftarrow a * x$		$m_B \leftarrow b * x$
	$m_A \longrightarrow$	
	$\longleftarrow m_B$	
$k_A \leftarrow a * m_B$		$k_B \leftarrow b * m_A$
Output: k_A		**Output:** k_B

FIGURE 1. Key agreement protocol $\mathcal{KA}1$.

$\mathcal{A}$		$\mathcal{B}$
Input: $\mathrm{sk}_A, \mathrm{pk}_B$		**Input:** $\mathrm{sk}_B, \mathrm{pk}_A$
$a \xleftarrow{R} G$		$b \xleftarrow{R} G$
$m_A \leftarrow (a\,\mathrm{sk}_A) * \mathrm{pk}_B$		$m_B \leftarrow (b\,\mathrm{sk}_B) * \mathrm{pk}_A$
	$m_A \longrightarrow$	
	$\longleftarrow m_B$	
$k_A \leftarrow a * m_B$		$k_B \leftarrow b * m_A$
Output: k_A		**Output:** k_B

FIGURE 2. Key agreement protocol $\mathcal{KA}2$.

so $\mathcal{A}$ and $\mathcal{B}$ output the same session key k. The protocol $\mathcal{KA}1$ provides secrecy of k from passive adversaries.

The next two key agreement protocols make use of long-term public key pairs and thus require a one-time setup. Alice randomly chooses her *secret key* $\mathrm{sk}_A \in G$ and then computes the corresponding *public key* $\mathrm{pk}_A = \mathrm{sk}_A * x$. Alice then provides her public key to Bob in an authentic manner. Bob does the same setup with his key pair $\mathrm{sk}_B, \mathrm{pk}_B$. Key Agreement 2 and 3 protocols are shown in Fig. 2 and 3, respectively.

$\mathcal{A}$ and $\mathcal{B}$ output the same session key k in the protocol $\mathcal{KA}2$ since

$$k_A = a * m_B = a * ((b\,\mathrm{sk}_B) * (\mathrm{sk}_A * x)) = (a\,b\,\mathrm{sk}_B\,\mathrm{sk}_A) * x = \\ = b * ((a\,\mathrm{sk}_A) * (\mathrm{sk}_B * x)) = b * m_A = k_B.$$

In the $\mathcal{KA}3$ protocol, $\mathcal{A}$ and $\mathcal{B}$ output the same session key k since

$$k_A = (a\,\mathrm{sk}_A^{-1}) * m_B = (a\,\mathrm{sk}_A^{-1}) * (b * (\mathrm{sk}_A * x)) = (ab) * x = \\ = (b\,\mathrm{sk}_B^{-1}) * (a * (\mathrm{sk}_B * x)) = (b\,\mathrm{sk}_B^{-1}) * m_A = k_B.$$

Whereas the $\mathcal{KA}1$ protocol does not provide authenticity, the protocols $\mathcal{KA}2$ and $\mathcal{KA}3$ are designed to provide mutual key authentication, i.e. Alice is assured that no other party aside from the one in possession of sk_B may gain access to k, and vice-versa. The protocols $\mathcal{KA}2$ and $\mathcal{KA}3$ are generalizations of the MTI/C1 and MTI/C0 protocols proposed by Matsumoto, Takashima and Imai [30, §12.6].

$\mathcal{A}$		$\mathcal{B}$
Input: $\mathrm{sk}_A, \mathrm{pk}_B$		**Input:** $\mathrm{sk}_B, \mathrm{pk}_A$
$a \xleftarrow{R} G$		$b \xleftarrow{R} G$
$m_A \leftarrow a * \mathrm{pk}_B$		$m_B \leftarrow b * \mathrm{pk}_A$
	$m_A \longrightarrow$	
	$\longleftarrow m_B$	
$k_A \leftarrow (a\,\mathrm{sk}_A^{-1}) * m_B$		$k_B \leftarrow (b\,\mathrm{sk}_B^{-1}) * m_A$
Output: k_A		**Output:** k_B

FIGURE 3. Key agreement protocol $\mathcal{KA}3$.

$\mathcal{K}$: Key generation	$\mathcal{E}$: Encryption	$\mathcal{D}$: Decryption
Input: -	**Input:** pk, m	**Input:** sk, pk, ct
$\mathrm{sk} \xleftarrow{R} G$	$a \xleftarrow{R} G$	$u \leftarrow \mathrm{sk} * z$
$y \leftarrow \mathrm{sk} * x$	$u \leftarrow a * y$	$h \leftarrow H_k(u)$
$k \xleftarrow{R} K$	$h \leftarrow H_k(u)$	$m \leftarrow h \oplus c$
$\mathrm{pk} \leftarrow (y, k)$	$z \leftarrow a * x$	**Output:** m
Output: sk, pk	$c \leftarrow h \oplus m$	
	$\mathrm{ct} \leftarrow (c, z)$	
	Output: ct	

FIGURE 4. Public-key encryption scheme $\mathcal{PE}$.

2.3. PUBLIC-KEY ENCRYPTION BASED ON GROUP ACTION. We now generalize the ElGamal public-key encryption scheme to the context of group action. An approach proposed by Monico [31] requires the set X to be a group in order to mask a message $m \in X$. In contrast with this, we use a "hashed" version of the ElGamal encryption scheme, which eliminates these restrictions on X and m through the use of a hash function family $\mathcal{H}$, which, however, introduces a need for a security assumption about $\mathcal{H}$ (see Theorem 2).

For a fixed message length w, the message space is the set of bit strings $\{0,1\}^w$, and thus we can write $m \in \{0,1\}^w$. We use a hash function family $\mathcal{H} = \{H_k : k \in K\}$ indexed by a finite set K, such that each H_k is a function

$$H_k : G * x \to \{0,1\}^w.$$

The public-key encryption scheme $\mathcal{PE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is pictured on Fig. 4. Some of the notation we use are: m is a message, $\mathrm{sk} \in G$ is a secret key, $\mathrm{pk} \in X \times K$ is a public key and $\mathrm{ct} \in \{0,1\}^w \times X$ is a ciphertext. We have that $x, y, z, u \in X$. The algorithm $\mathcal{K}$ also takes the description of K as part of implied system parameters.

The encryption scheme $\mathcal{PE}$ is sound, that is to say, for all pairs (sk, pk) which can be output by $\mathcal{K}()$ and for all $m \in \{0,1\}^w$ we have that $\mathcal{D}(\mathrm{sk}, \mathrm{pk}, \mathcal{E}(\mathrm{pk}, m)) = m$. Indeed, we can write $H_k(a * r) = H_k(\mathrm{sk} * y)$ since

$$a * r = a * (\mathrm{sk} * x) = (a\,\mathrm{sk}) * x = \mathrm{sk} * (a * x) = \mathrm{sk} * y.$$

2.4. SECURITY OF CRYPTOGRAPHIC SCHEMES BASED ON GROUP ACTION. In this section we provide reductionist security arguments for the $\mathcal{KA}1$ protocol and the $\mathcal{PE}$ scheme, thus showing that breaching the security of these schemes is not easier than solving particular computational problems. A detailed discussion, together with proofs of Theorems 1 and 2, can be found in our earlier work [42].

For a finite abelian group G acting on a set X and a fixed element $x \in X$ we define the following computational problems:

Problem 1 (Group Action Inverse Problem (GAIP))**.** *Given a randomly chosen element $y \in G * x$, find a group element $g \in G$ such that $g * x = y$.*

A problem similar to GAIP was defined for semigroups by Maze et al. [29] as the semigroup action problem. However we prefer the name GAIP as the problem asks to invert the function $f(a) = a * x$.

Problem 2 (Decisional Diffie-Hellman Group Action Problem (DDHAP))**.** *Given a triple $(y, z, u) \in X^3$ sampled with probability $\frac{1}{2}$ from one of the two following probability distributions:*

- *$(a * x,\ b * x,\ (ab) * x)$, where a and b are randomly chosen from G,*
- *$(a * x,\ b * x,\ c * x)$, where a, b and c are randomly chosen from G,*

decide which distribution the triple is sampled from.

Using a GAIP solver it is straightforward to construct a solver for the DDHAP, thus the DDHAP is not harder than the GAIP.

For a DDHAP distinguisher $\mathcal{S}$, its probability of returning the correct solution will be denoted by $\Pr_{\mathcal{S}}^{\mathrm{DDH}}$. $\Pr_{\mathcal{S}}^{\mathrm{DDH}}$ is a function of a security parameter $s = \log \# G * x$. Since the distinguisher $\mathcal{S}$ can gain a success probability of $\frac{1}{2}$ by returning a random solution, the advantage of $\mathcal{S}$ is defined to be

$$\mathrm{Adv}_{\mathcal{S}}^{\mathrm{DDH}} = \left| \Pr_{\mathcal{S}}^{\mathrm{DDH}} - \frac{1}{2} \right|.$$

We can now define the following assumption about the computational complexity of the DDHAP:

Assumption 1 (DDHAP)**.** *For any polynomial-time DDHAP distinguisher $\mathcal{S}$, the advantage $\mathrm{Adv}_{\mathcal{S}}^{\mathrm{DDH}}$ is a negligible[1] function of s.*

To model the security of the $\mathcal{KA}1$ protocol we will use a notion of *session-key (SK) security in the authenticated-links adversarial model (AM)* proposed by Canetti and Krawczyk [9, 10]. In outline, this security notion asserts that any polynomial-time adversary $\mathcal{I}$ that cannot change the information transmitted between parties, does not learn anything about the value of the session key established between uncorrupted parties. This is formalized via the infeasibility for $\mathcal{I}$ to distinguish between the real value of the session key and an independent random value in a specially designed experiment. We refer to the papers of Canetti and Krawczyk for a formal definition of the SK security in the AM. After a few implementation-specific modifications to the protocol $\mathcal{KA}1$, namely introducing party identifiers, session identifiers and requiring to erase variables a and b before returning the output, the following theorem can be proved:

[1] A function $\mu(x)$ is negligible, if for every positive integer c there exists an $N_c > 0$ such that for all $x > N_c$, the following holds: $|\mu(x)| < 1/x^c$.

Theorem 1. *If the DDHAP assumption holds for the finite abelian group G acting on the set X, then the $\mathcal{KA}1$ protocol is SK-secure in the AM.*

The classical goal of encryption is to preserve the privacy of messages: an adversary should not be able to learn from a ciphertext information about its plaintext beyond the length of that plaintext. This idea is captured via the notion of semantic security of an encryption scheme, proposed by Goldwasser and Micali [21], which asserts that any polynomial-time adversary cannot effectively distinguish between the encryption of two messages of his choosing. We will use an equivalent notion, indistinguishability of encryptions in a chosen-plaintext attack (IND-CPA) [3]. The equivalence of these two security notions has been shown by Goldreich [20].

In our security argument we will also use a property of a hash function family $\mathcal{H}$ to be entropy smoothing (ES). The smooth entropy denotes the number of almost uniform random bits in a random variable [7, 8]. The ES hash function should be able to produce almost uniformly distributed outputs by decreasing the output size, as compared to the size of the input. This is formalized via the requirement that any polynomial-time adversary cannot effectively distinguish between the values $(k, H_k(u))$ and (k, h), where $k \in K$, $u \in U$ and $h \in \{0,1\}^w$ are chosen at random, and U is the domain of the hash function. When applied to the $\mathcal{PE}$ scheme, U is the set of bit strings that represent the elements of $G * x$.

Assumption 2 (ES)**.** *The hash function family $\mathcal{H}$ is entropy smoothing.*

Theorem 2. *If the DDHAP assumption holds for the finite abelian group G acting on the set X, and the hash function family $\mathcal{H}$ is ES, then the public-key encryption scheme $\mathcal{PE}$ is secure in the sense of IND-CPA.*

We have shown that the security of the encryption scheme $\mathcal{PE}$ and the protocol $\mathcal{KA}1$ is based on the hardness of the DDHAP and the GAIP. The $\mathcal{PE}$ security is also subject to the ES assumption about the used hash function family.

3. Isogenous elliptic curves over prime fields

We show that elliptic curves provide an option for implementing the cryptographic schemes presented in Sect. 2. The aim of this section is to define mathematical structures that will be used as a set X and a finite abelian group G acting on X.

An elliptic curve E over a field F, $\mathrm{char}(F) \notin \{2,3\}$, is an algebraic curve defined by an equation $y^2 = x^3 + Ax + B$, where $A, B \in F$ and $4A^3 + 27B^2 \neq 0$. The set of rational points on E over F, together with the "point at infinity" O, is an additive group with the zero element O. The elliptic curve E over F is denoted E/F, and its group of rational points is denoted $E(F)$. For elliptic curves E_1/F and E_2/F, an isogeny between E_1 and E_2 is a morphism $\phi\colon E_1 \to E_2$ of varieties that satisfies $\phi(O) = O$. The elliptic curves E_1 and E_2 are called isogenous if there is a non-constant isogeny between them. Every isogeny $E_1 \to E_2$ induces a homomorphism of the groups $E_1(F)$ and $E_2(F)$. For an elliptic curve E/F, the set of all isogenies $E \to E$ defined over $\bar{F}$ is called the endomorphism ring of E and denoted $\mathrm{End}(E)$.

We review basic facts about elliptic curves over $\mathbb{C}$ to establish notation [40, Ch. II §1]. For a pair of complex numbers $\omega_1, \omega_2 \in \mathbb{C}$, such that $\omega_2/\omega_1 \notin \mathbb{R}$, the additive group $\Lambda = \omega_1\mathbb{Z} + \omega_2\mathbb{Z}$ is called a complex lattice. For a complex lattice Λ, the Weierstrass $\wp$-function gives rise to an elliptic curve

$$E_\Lambda/\mathbb{C}\colon\ y^2 = 4x^3 - g_2(\Lambda)x - g_3(\Lambda) \tag{1}$$

and a group isomorphism $\mathbb{C}/\Lambda \cong E_\Lambda(\mathbb{C})$. For any nonzero $\alpha \in \mathbb{C}$ the multiplication-by-α map induces the isomorphism $E_\Lambda(\mathbb{C}) \cong E_{\alpha\Lambda}(\mathbb{C})$, and for the endomorphism ring of E_Λ we have $\mathrm{End}(E_\Lambda) \cong \{\alpha \in \mathbb{C}\colon \alpha\Lambda \subset \Lambda\}$. When $\mathrm{End}(E_\Lambda)$ is larger than $\mathbb{Z}$, E_Λ is said to have complex multiplication, and $\mathrm{End}(E_\Lambda)$ is then isomorphic to an imaginary quadratic order.

Let $\mathcal{O}_K$ be the ring of integers of an imaginary quadratic field K. By $\mathcal{ELL}(\mathcal{O}_K)$ we denote the set of isomorphism classes of elliptic curves over $\mathbb{C}$ having the endomorphism ring $\mathcal{O}_K$:

$$\mathcal{ELL}(\mathcal{O}_K) = \frac{\{E/\mathbb{C}\colon \mathrm{End}(E) \cong \mathcal{O}_K\}}{\text{isomorphism over } \mathbb{C}}.$$

For convenience we will write $E \in \mathcal{ELL}(\mathcal{O}_K)$ meaning that E belongs to the corresponding isomorphism class $[E]$ in $\mathcal{ELL}(\mathcal{O}_K)$.

There is a well-defined action of the ideal class group $\mathcal{CL}(\mathcal{O}_K)$ on $\mathcal{ELL}(\mathcal{O}_K)$ given by

$$[\mathfrak{a}] * E_\Lambda = E_{\mathfrak{a}^{-1}\Lambda},$$

where $\mathfrak{a}$ is a fractional ideal, $[\mathfrak{a}]$ is the corresponding ideal class and Λ is a lattice with $E_\Lambda \in \mathcal{ELL}(\mathcal{O}_K)$.

For every ideal class $[\mathfrak{a}]$ there exists an integral ideal $\mathfrak{b} \subset \mathcal{O}_K$ for which $[\mathfrak{b}] = [\mathfrak{a}]$. Then $\Lambda \subset \mathfrak{b}^{-1}\Lambda$, and the homomorphism

$$\begin{array}{ccc} \mathbb{C}/\Lambda & \to & \mathbb{C}/\mathfrak{b}^{-1}\Lambda \\ z & \mapsto & z, \end{array}$$

induces a natural isogeny $\psi\colon E_\Lambda \to E_{\mathfrak{b}^{-1}\Lambda}$. The kernel of ψ is isomorphic to $\mathfrak{b}^{-1}\Lambda/\Lambda \cong \mathcal{O}_K/\mathfrak{b}$, and the degree of ψ equals the norm $N_{\mathbb{Q}}^K(\mathfrak{b})$.

We now reduce elliptic curves over $\mathbb{C}$ to the ones over a finite field[2] [16, §14.C]. Let H be the Hilbert class field of K, and $\mathcal{O}_H$ its ring of integers. Then the elliptic curves in $\mathcal{ELL}(\mathcal{O}_K)$ are defined over H. Let $p > 3$ be a prime in $\mathbb{Z}$ which splits completely in H, and fix a prime $\mathfrak{P}$ of H lying above p, so that $\mathcal{O}_H/\mathfrak{P} \cong \mathbb{F}_p$. Finally, let $E \in \mathcal{ELL}(\mathcal{O}_K)$ be an elliptic curve which has good reduction at $\mathfrak{P}$. Hence g_2 and g_3 of (1) can be written in the form α/β where $\alpha, \beta \in \mathcal{O}_H$, $\beta \notin \mathfrak{P}$, so that $[g_2]$ and $[g_3]$ can be defined in $\mathcal{O}_H/\mathfrak{P}$. The elliptic curve $\bar{E}\colon y^2 = 4x^3 - [g_2]x - [g_3]$ over the finite field $\mathcal{O}_H/\mathfrak{P}$ is called the reduction of E modulo $\mathfrak{P}$. The reduction preserves the endomorphism ring, namely $\mathrm{End}_{\bar{\mathbb{F}}_p}(\bar{E}) \cong \mathcal{O}_K$, and every elliptic curve over $\mathbb{F}_p$ with the endomorphism ring $\mathcal{O}_K$ arises in this way. For two elliptic curves $E_1, E_2 \in \mathcal{ELL}(\mathcal{O}_K)$ that have good reduction at $\mathfrak{P}$, the natural reduction map $\mathrm{Hom}(E_1, E_2) \to \mathrm{Hom}(\bar{E}_1, \bar{E}_2)$ is injective and preserves degrees [40, Proposition II.4.4].

Let now $E_1/\mathbb{F}_p$ be an ordinary elliptic curve such that $\mathrm{End}(E_1) \cong \mathcal{O}_K$ for some imaginary quadratic field K. According to a theorem of Tate, E_1 is $\mathbb{F}_p$-isogenous to an elliptic curve $E_2/\mathbb{F}_p$ if and only if $\#E_1(\mathbb{F}_p) = \#E_2(\mathbb{F}_p)$ [43, §3 Theorem 1]. We denote $n = \#E_1(\mathbb{F}_p)$ and restrict ourselves to the isogeny class of elliptic curves with n points. We define a set of isomorphism classes of elliptic curves with the endomorphism ring $\mathcal{O}_K$ to be

$$\mathcal{ELL}_{p,n}(\mathcal{O}_K) = \frac{\{E/\mathbb{F}_p\colon \#E(\mathbb{F}_p) = n \text{ and } \mathrm{End}(E) \cong \mathcal{O}_K\}}{\text{isomorphism over } \bar{\mathbb{F}}_p}.$$

[2]We use ordinary elliptic curves. With supersingular curves the endomorphism ring is noncommutative and so does not lead to an action by an abelian group. This is why our cryptographic schemes do not have the same efficiency as the hash function of Charles, Goren and Lauter [12].

Elements of $\mathcal{CL}(\mathcal{O}_{-152})$	Permutations on $\mathcal{ELL}_{47,42}(\mathcal{O}_{-152})$
$g = [(3, 2, \cdot)]$	(27 12 15 24 41 19)
$g^2 = [(6, 4, \cdot)]$	(27 15 41)(19 12 24)
$g^3 = [(2, 0, \cdot)]$	(27 24)(19 15)(41 12)
$g^4 = [(6, -4, \cdot)]$	(27 41 15)(19 24 12)
$g^5 = [(3, -2, \cdot)]$	(27 19 41 24 15 12)
$g^6 = [(1, 0, \cdot)]$	(27)(19)(41)(24)(15)(12)

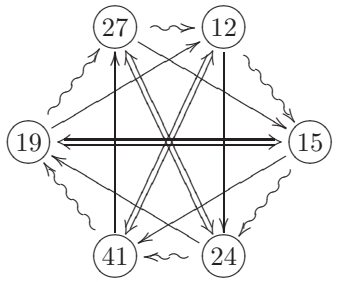

FIGURE 5. $\mathcal{CL}(\mathcal{O}_{-152})$ action on $\mathcal{ELL}_{47,42}(\mathcal{O}_{-152})$.

In terms of horizontal and vertical isogenies introduced by Kohel [28] this set corresponds to the surface, i.e. the topmost level, of the isogeny graph.

The reduction modulo $\mathfrak{P}$ maps $\mathcal{ELL}(\mathcal{O}_K)$ to $\mathcal{ELL}_{p,n}(\mathcal{O}_K)$ and induces the action of $\mathcal{CL}(\mathcal{O}_K)$ on $\mathcal{ELL}_{p,n}(\mathcal{O}_K)$. Since the reduction preserves isogeny degrees, for an action $[\mathfrak{a}] * E_1/\mathbb{F}_p = E_2/\mathbb{F}_p$ by an integral ideal $\mathfrak{a}$ there exists a separable $\mathbb{F}_p$-isogeny $\psi\colon E_1 \to E_2$ of degree $N_{\mathbb{Q}}^K(\mathfrak{a})$.

In order to shorten the notation, we will often write $\mathcal{ELL}_{p,n}$ or even $\mathcal{ELL}$ instead of $\mathcal{ELL}_{p,n}(\mathcal{O}_K)$, and $\mathcal{CL}$ instead of $\mathcal{CL}(\mathcal{O}_K)$, where we do not need to explicitly refer to $\mathcal{O}_K$, p and n.

Remarkably, the set $\mathcal{ELL}_{p,n}$ is a principal homogeneous space over the group $\mathcal{CL}$ [45, Theorem 4.5]. In particular, $\mathcal{CL}$ acts freely and transitively on $\mathcal{ELL}_{p,n}$, i.e. for any $x, y \in \mathcal{ELL}_{p,n}$ there exists precisely one $g \in \mathcal{CL}$ such that $g * x = y$. It follows that $\#\mathcal{CL} = \#\mathcal{ELL}_{p,n}$.

For the sake of a small example, consider the elliptic curve $E\colon y^2 = x^3 + x + 5$ over the field $\mathbb{F}_{47}$. E has 42 points, the Frobenius discriminant $\Delta_\pi = -152$ is fundamental and $j_E = 27$. Figure 5 shows the action of $\mathcal{CL}(\mathcal{O}_{-152})$ on $\mathcal{ELL}_{47,42}(\mathcal{O}_{-152})$. Nodes of the graph are the j-invariants of the elliptic curves in $\mathcal{ELL}_{47,42}(\mathcal{O}_{-152})$. Ideal classes are represented by the reduced binary quadratic forms. Permutations are written in the cyclic notation, that is (27 15 41) means that the ideal class $[(6, 4, \cdot)]$ maps the curves with j-invariant 27 to curves with j-invariant 15, 15 to 41 and 41 to 27. Since the ideal $7\mathbb{Z} + \frac{-4+\sqrt{-152}}{2}\mathbb{Z}$ belongs to the ideal class $[(6, 4, \cdot)]$ and has norm 7, isogenies of degree 7 form cycles (27 15 41) and (19 12 24).

Thus for an abelian group G and a set X defined by

$$G = \mathcal{CL}(\mathcal{O}_K),$$
$$X = \mathcal{ELL}_{p,n}(\mathcal{O}_K)$$

it is possible to implement the cryptographic schemes proposed in Sect. 2. The next three sections describe the implementation in greater detail.

4. Elements of $\mathcal{ELL}_{p,n}$ and $\mathcal{CL}$

We show how one can store elements of X and G when implementing the cryptographic schemes of Sect. 2 on a set of isomorphism classes of elliptic curves.

4.1. Elements of $\mathcal{ELL}_{p,n}$. Elements of a set $\mathcal{ELL}_{p,n}$ are isomorphism classes of elliptic curves. Since two elliptic curves are isomorphic over $\bar{\mathbb{F}}_p$ if and only if they have the same j-invariant [39, Proposition III.1.4.b], we represent the elements of

$\mathcal{ELL}_{p,n}$ by j-invariants. As j-invariants lie in $\mathbb{F}_p$ they can be stored as non-negative integers less than p.

However, for calculations described in Sect. 5 one will need an explicit equation of a curve $E \in \mathcal{ELL}_{p,n}$ with a given j-invariant. Here we see two ways for implementation: either to transmit the explicit curve equation, or to transmit the j-invariants and compute the equations when it is needed. The former approach saves computational resources while the latter one saves bandwidth. In the latter case, the elliptic curve equation can be obtained in the following way. At first one sets $c = j/(j - 1728)$ and considers the curve $E\colon y^2 = x^3 - 3cx + 2c$. E is either a desired elliptic curve or a twist, i.e. $\#E(\mathbb{F}_p)$ correspondingly equals n or $2p+2-n$. When these numbers are relatively prime, $\#E$ can be tested by taking a random point on E and multiplying it by n. If in the twisted case, one takes a quadratic non-residue $v \in \mathbb{F}_p^*$ and sets the desired curve to be $E'\colon y^2 = x^3 - 3cv^2x + 2cv^3$.

4.2. ELEMENTS OF $\mathcal{CL}$. Elements of an ideal class group $\mathcal{CL}(\mathcal{O}_K)$ are classes of fractional ideals modulo principal fractional ideals of the imaginary quadratic order $\mathcal{O}_K$. A traditional approach is to use reduced ideals or, equivalently, reduced binary quadratic forms, as representatives of the ideal classes. However for storing elements of $\mathcal{CL}$ we will use a different approach, which is more suitable in the context of the action on $\mathcal{ELL}_{p,n}$.

We first recall an important result by Kohel [28]. For an ordinary elliptic curve $E/\mathbb{F}_p$ with endomorphism ring $\mathcal{O}_K$ of discriminant Δ, and a prime l, there are $\left(\frac{\Delta}{l}\right)+1$ isogenies of degree l to curves with endomorphism ring isomorphic to $\mathcal{O}_K$. We immediately observe that inert primes, namely those satisfying $\left(\frac{\Delta}{l}\right) = -1$, do not appear as degrees of isogenies inside $\mathcal{ELL}_{p,n}$, and therefore will not be further considered. Ramified primes, that is when $l \mid \Delta$, yield only one horizontal isogeny of degree l. Due to the existence of dual isogenies, isogenies of a ramified degree form loops of length 2. In other words, since $l\,\mathcal{O}_K = \mathfrak{l}^2$, the ideal $\mathfrak{l}$ has order 2 in $\mathcal{CL}$. As compared to split primes, a ramified isogeny degree l does not introduce much diversity when moving on the isogeny graph, because the number of hops by l-isogenies has to be considered modulo 2. When l is split, there are two horizontal isogenies of degree l, and l-isogeny cycles can be much longer. For this reason it is beneficial to select an elliptic curve with an endomorphism ring where most of the small primes are split, when choosing the system parameters. Our further attention will be concentrated on split primes.

For a fixed positive integer $l_{\max}$ let L be an indexed set

$$L = \left\{ \text{primes } l_i\colon \left(\frac{\Delta}{l_i}\right) = 1 \text{ and } l_i \le l_{\max} \right\}. \tag{2}$$

Let also $d = \#L$. Then for each i, $1 \le i \le d$, we define a prime ideal $\mathfrak{l}_i$ of $\mathcal{O}_K$ to be

$$\mathfrak{l}_i = l_i\mathbb{Z} + \frac{b_i + \sqrt{\Delta}}{2}\mathbb{Z}, \tag{3}$$

where $b_i = \min\{b \in \mathbb{N}\colon b^2 \equiv \Delta \pmod{4l_i}\}$. Now a homomorphism

$$\begin{aligned} \phi\colon \quad \mathbb{Z}^d &\to \mathcal{CL} \\ (v_1, \ldots, v_d) &\mapsto \prod_{i=1}^{d} [\mathfrak{l}_i]^{v_i} \end{aligned} \tag{4}$$

lets us use vectors in $\mathbb{Z}^d$ to store elements of $\mathcal{CL}$. Obviously, the addition of vectors corresponds to the multiplication in $\mathcal{CL}$, and the additive vector inverse is to be used as the ideal class inverse (used in the $\mathcal{KA}3$ protocol).

Remark 1. Whether the map ϕ is onto depends on the choice of L. It is proved that, if the generalized Riemann hypothesis (GRH) is true, then there exists a constant c_0 such that for $l_{\max} = c_0 \log^2 |\Delta|$ in the setting above, the ideal classes $[\mathfrak{l}_i]$, $1 \leq i \leq d$, generate the group $\mathcal{CL}$ [22, Theorem 2]. In practice for a majority of ideal class groups it suffices to take $c_0 = 0.5$. Moreover, it is believed that the average minimum value of $l_{\max}$ needed to generate $\mathcal{CL}$ is $O(\log^{1+\epsilon} |\Delta|)$ for any $\epsilon > 0$ [2]. In our practical experiments we chose $l_{\max} \approx \log |\Delta|$. Note that when the structure of $\mathcal{CL}$ is pre-computed during the system parameters selection, the fact that the prime ideals $\mathfrak{l}_i$ of norms in L generate $\mathcal{CL}$ can be tested and nonconforming class groups can be discarded. Otherwise, when the structure of $\mathcal{CL}$ is not pre-computed, the fact that any element of $\mathcal{CL}$ can be represented as a product in (4) depends on the GRH and the choice of $l_{\max}$.

4.3. GENERATING THE SYSTEM PARAMETERS. In order to choose the set $\mathcal{ELL}_{p,n}$ one first picks a prime p of sufficient size (see Sect. 7.1). One then tries arbitrary elliptic curves over $\mathbb{F}_p$ until an appropriate curve is found. For every curve E one computes $\#E(\mathbb{F}_p)$ using the SEA algorithm. The trace t of the Frobenius endomorphism $\pi\colon (x, y) \mapsto (x^p, y^p)$ is then obtained as $t = p + 1 - \#E(\mathbb{F}_p)$, and to test that E is ordinary one verifies that $t \neq 0$. The Frobenius discriminant Δ_π is then obtained by the formula

$$\Delta_\pi = t^2 - 4p. \tag{5}$$

The discriminant Δ of $\mathrm{End}(E)$ satisfies

$$\Delta = \Delta_\pi / g^2 \tag{6}$$

for some integer g. A straightforward way to ensure that $\mathrm{End}(E)$ is a maximal imaginary quadratic order is to check whether Δ_π is a fundamental discriminant. Besides, when the conductor of Δ_π is not divisible by a large prime, one may use an algorithm of Kohel [28] to move from E to a curve with the maximal endomorphism ring. In this case one should also check that $l_i \nmid g$ for all $l_i \in L$, so that there are no l_i-isogenies down.

Once Δ is known, one chooses $l_{\max}$ (see Remark 1) and examines how many of the primes less or equal to $l_{\max}$ are split. The more small primes are split the better performance the $\mathcal{CL}$ action will have. One can even do an "early abort" of the SEA algorithm when t is computed modulo small primes and too few of them satisfy $(\frac{t^2-4p}{l_i}) = 1$. When the curve is chosen, the structure of $\mathcal{CL}(\mathcal{O}_\Delta)$ should be computed, as we discuss later in Sect. 6.

The obtained elliptic curve $E \in \mathcal{ELL}_{p,n}$ is to be used as the set element x in the proposed cryptographic schemes.

5. IMPLEMENTATION OF $\mathcal{CL}$ ACTION ON $\mathcal{ELL}_{p,n}$

We show how to implement the group action $*$ used in the cryptographic schemes of Sect. 2. Let an elliptic curve $E \in \mathcal{ELL}_{p,n}$ be defined by $y^2 = x^3 + Ax + B$, and an ideal class be given by a vector $\vec{v} \in \mathbb{Z}^d$ according to the notation of Sect. 4.2. Our aim is to compute $\phi(\vec{v}) * E \in \mathcal{ELL}_{p,n}$.

From (4) we have that

$$\phi(\vec{v}) * E = (\prod_{i=1}^{d} [\mathfrak{l}_i]^{v_i}) * E. \tag{7}$$

The associativity property of the group action implies that we can compute (7) by gradually acting by the factors $[\mathfrak{l}_i]$ or $[\mathfrak{l}_i]^{-1}$ in (7), depending on the sign of v_i. We call the operation $[\mathfrak{l}_i] * E = E_1$ a hop, this corresponds to an isogeny $E \to E_1$ of degree l_i. The computation of (7) consists of $\sum_{i=1}^{d} |v_i|$ hops between elliptic curves. We further explain the implementation of a single hop.

Throughout this section we use a notation

$$(l, b, \cdot) = l\mathbb{Z} + \frac{b + \sqrt{\Delta}}{2}\mathbb{Z}$$

for the prime ideals $\mathfrak{l}_i$ defined by (3), in order to explicitly refer to b. Since $\left(\frac{\Delta}{l}\right) = 1$, the prime l is split and we can write $l\mathcal{O}_K = (l, b, \cdot)(l, -b, \cdot)$. It follows that $[(l, b, \cdot)]^{-1} = [(l, -b, \cdot)]$, and for the negative coordinates in $\vec{v}$ we should use the action by $[(l, -b, \cdot)]$.

To compute the elliptic curve $E_1 = [(l, b, \cdot)] * E$ we apply ideas used in the SEA algorithm [15, 37]. For the action $[(l, b, \cdot)] * E = E_1$ there exists a separable $\mathbb{F}_p$-isogeny $\psi\colon E \to E_1$ of degree $N_{\mathbb{Q}}^{K}(l, b, \cdot) = l$. The same holds for $[(l, -b, \cdot)] * E = E_2$. The j-invariants of E_1 and E_2 are computed as roots of the equation

$$\Phi_l(x, j(E)) = 0 \pmod{p}, \tag{8}$$

where Φ_l is the modular polynomial of level l. When l does not divide the conductor g of Δ_π, the equation (8) has exactly 2 roots. We should determine which root is the j-invariant of the curve E_1. For that we take one of the roots $\hat{\jmath}$ and apply the algorithm of Elkies to compute the equation of an isogenous elliptic curve $\hat{E}$, $j(\hat{E}) = \hat{\jmath}$, and the polynomial $\hat{h}(x)$ that vanishes on the l-isogeny $E \to \hat{E}$ kernel.

The kernel of an l-degree isogeny is a subgroup of the l-torsion group, and the Frobenius endomorphism π on the kernel points satisfies the characteristic equation

$$\pi^2 - t\pi + p \equiv 0 \pmod{l}. \tag{9}$$

For split l the equation (9) has two different roots $\pi_1, \pi_2 \in \mathbb{Z}_l$ called Frobenius eigenvalues. These are related to the ideals $(l, b, \cdot)$ and $(l, -b, \cdot)$ by the following formula:

$$\pi_{1,2} \equiv \frac{t \pm gb}{2} \pmod{l},$$

where t is the trace of the Frobenius endomorphism. Indeed, we have that $b^2 \equiv \Delta \pmod{4l}$, thus $g^2 b^2 \equiv \Delta_\pi \pmod{4l}$ and $(t + gb)/2$ satisfies (9):

$$\left(\frac{t + gb}{2}\right)^2 - t\frac{t + gb}{2} + p = \frac{g^2 b^2 - \Delta_\pi}{4} \equiv 0 \pmod{l}.$$

The same holds for $(t - gb)/2$. The eigenvalue $\pi_1 \equiv (t - gb)/2 \pmod{l}$ corresponds to the action of π on the kernel of the isogeny associated with $(l, b, \cdot)$ and $\pi_2 \equiv (t + gb)/2 \pmod{l}$ corresponds to the isogeny associated with $(l, -b, \cdot)$.

We then check that the eigenvalue π_1 satisfies the relation

$$(x^p, y^p) \equiv [\pi_1](x, y) \pmod{y^2 - x^3 - Ax - B,\ \hat{h}(x)}, \tag{10}$$

where $[\pi_1](x, y)$ stands for the point multiplication by π_1. If (10) holds, we set the resulting elliptic curve E_1 to be $\hat{E}$. Otherwise E_1 is obtained from the second root of (8).

The computational complexity of a single hop between l-isogenous elliptic curves is dominated by solving the equation (8). The degree of the polynomial $f(x) = \Phi_l(x, j(E))$ is $l + 1$. The roots are found by computing the $\gcd(x^p - x, f(x))$. The left-right binary exponentiation $x^p \pmod{f(x)}$ takes $\log p$ polynomial squarings, and multiplications by x are given "for free". Each polynomial multiplication through the number-theoretic transform (NTT) requires $O(l \log l)$ field multiplications. Also the division with remainder of the $2l$-degree product polynomial by $f(x)$ can be implemented in $O(l \log l)$ multiplications in $\mathbb{F}_p$ [5]. In practice for l less than approximately 70 these operations are faster implemented with the "schoolbook" and related algorithms, but for the asymptotic analysis we use the $O(l \log l)$ estimation anyway. This results in a total of $O(l \log l \log p)$ field multiplications needed to compute $x^p \pmod{f(x)}$. The GCD of l-degree polynomials is computed with $O(l \log^2 l)$ field multiplications [5], and since $\log l < \log p$, the resulting complexity of solving $f(x) = 0$ is $O(l \log l \log p)$.

The second most demanding operation in an l-isogenous hop is the verification of (10). A substitution $y^{p-1} = (x^3 + Ax + B)^{(p-1)/2}$ allows the exponentiation to be performed in the univariate polynomial ring $\mathbb{F}_p[x]/h_1(x)$. Since the degree of $h_1(x)$ is $(l - 1)/2$, the binary exponentiation requires $O(l \log l \log p)$ field multiplications. When several consecutive hops are done along the same isogeny degree l, the verification of (10) is needed only on the first hop. This is because when moving along a cycle, at the second and the subsequent hops we know where we came from.

Thus the running time of one hop between l-isogenous elliptic curves is

$$O(l \log l \log p) \tag{11}$$

multiplications in $\mathbb{F}_p$.

To estimate the average running time of the action (7), we use the following approximations: $h \approx 0.46(-\Delta)^{1/2}$ for the class number $\#\mathcal{CL}(\mathcal{O}_\Delta)$ [13]; $l_{\max} \approx \log|\Delta|$ for the biggest prime in L (see Remark 1); $\Delta = cp$, where c is a constant (follows from (5), the Hasse's bound $|t| \le 2\sqrt{p}$, (6) and the fact that the conductor g is chosen to be small during the system parameter generation); $l_i \approx 2i \ln 2i$ for the value of the i-th prime in L; and $d \approx \frac{1}{2} l_{\max} / \ln l_{\max}$ for the number of primes in L. In the last two approximations we use the prime number theorem and the fact that almost half of the primes are split. Let each v_i take values from an interval $[-\hat{v}, \hat{v}]$. The number of allowed values for each v_i approximates as $\log^{1/\log e} p$, since being raised to power d it gives $p^{1/2}$ possible vectors. The average running time of the group action (7) is then $O(\sum_{i=1}^{d} \log^{0.7} p \; l_i \log l_i \log p)$. We use the fact that $\sum_{i=1}^{d} i \ln^2 i$ is $O(d^2 \ln^2 d)$ and the above approximation for d. This gives

$$O\left(\log^{3.7} p\right)$$

multiplications in $\mathbb{F}_p$ needed for an average action of $\mathcal{CL}$ on $\mathcal{ECL}_{p,n}$.

6. Implementation of sampling from $\mathcal{CL}$

We show how to implement the sampling operation $\cdot \xleftarrow{R} G$ used in the cryptographic schemes of Sect. 2.

6.1. RANDOM SAMPLING FROM $\mathcal{CL}$. We use the notation for L, $\mathfrak{l}_i$, d, $\vec{v}$ and $\phi(\cdot)$ introduced in Sect. 4.2. A vector $\vec{v} \in \mathbb{Z}^d$ is said to be a relation if $\phi(\vec{v}) = [\mathcal{O}_K]$, i.e. $\vec{v}$ maps to the identity element of $\mathcal{CL}(\mathcal{O}_K)$. Let the set L be chosen such that the set of ideal classes $[\mathfrak{l}_i]$, $1 \le i \le d$, generates the group $\mathcal{CL}(\mathcal{O}_K)$. Following a class group computation algorithm of Jacobson [25], we compute a lattice $\Lambda = \ker \phi \subset \mathbb{Z}^d$ of relations among the ideal classes $[\mathfrak{l}_i]$, $1 \le i \le d$, so that $\mathcal{CL}(\mathcal{O}_K) \cong \mathbb{Z}^d/\Lambda$. Then we choose a minimal generating set of ideals $\mathfrak{l}_i$, $i \in J$, for a subset of indices $J \subset \{1, ..., d\}$, and find the orders $m_i = \mathrm{ord}[\mathfrak{l}_i]$, $i \in J$.

Now, using the Lenstra-Lenstra-Lovasz (LLL) lattice basis reduction algorithm, we compute a short basis of Λ and store it in a matrix B_Λ of column vectors. All the above described steps are needed only once and therefore can be done during the parameter choice or the pre-computation phase. This allows to reject elliptic curves that require $l_{\max}$ larger than approximately $\log|\Delta|$ (see Remark 1) or yield long vectors in B_Λ. Practical experiments for $\lceil \log p \rceil = 224$ and $l_{\max} \approx \log|\Delta|$ show that the coordinates of vectors in B_Λ are generally less than 50.

In order to implement the random sampling from $\mathcal{CL}$, we construct a vector $\vec{u} \in \mathbb{Z}^d$ by choosing the coordinates

$$u_i \leftarrow \begin{cases} \xleftarrow{R} \{0, 1, \ldots, m_i - 1\}, & i \in J; \\ \leftarrow 0, & i \notin J. \end{cases}$$

Now $\phi(\vec{u})$ is a random element of $\mathcal{CL}$, as the uniform distributions on the cyclic subgroups $\langle[\mathfrak{l}_i]\rangle$, $i \in J$, give the uniform distribution on $\mathcal{CL}$.

Some of the coordinates of $\vec{u}$ will be large. For instance, if the class group is generated by an ideal $\mathfrak{l}_i$, then u_i is a random number between 0 and $\#\mathcal{CL} - 1$. The following optimisation steps are aimed at computing an equivalent vector $\vec{v} \equiv \vec{u} \pmod{\Lambda}$ which is faster than $\vec{u}$ in terms of its action on $\mathcal{ELL}$.

We first find a lattice vector $\vec{b} \in \Lambda$ close to $\vec{u}$ and set $\vec{w} = \vec{u} - \vec{b}$. The vector $\vec{b}$ can be found by Babai rounding [1] as $\vec{b} = B_\Lambda \lfloor B_\Lambda^{-1} \vec{u} \rceil$, where $\lfloor \cdot \rceil$ is the coordinate-wise floor function, and $\vec{u}$ is a column vector, or by any other algorithm for the closest lattice vector problem.

To further optimize $\vec{w}$, we will need a d-dimensional row vector $\vec{t} = (t_1, \ldots, t_d)$, where each coordinate t_i is the average time used to compute the action $[\mathfrak{l}_i] * E$. The row vector $\vec{t}$ can be obtained experimentally during the parameter choice phase. Now for a column vector $\vec{v} \in \mathbb{Z}^d$, the approximate time needed to compute the action $\phi(\vec{v}) * E$ is $\vec{t}\,|\vec{v}|$, where $|\cdot|$ is the coordinate-wise absolute value function. We thus need to solve the following (mixed) integer linear program (ILP):

$$\begin{aligned} &\text{minimize} \quad \vec{t}\,|\vec{v}| \\ &\text{subject to} \quad \vec{v} = \vec{w} + B_\Lambda \vec{k}, \quad \text{int } \vec{k}. \end{aligned} \tag{12}$$

The problem (12) can be solved by various ILP algorithms, e.g. the branch and bound algorithm, the simplex algorithm or a primitive search. Note that we do not require finding the optimal solution, as it can take a long time. Even a quick run of an ILP algorithm allows to significantly improve the value of the objective function in (12). Moreover, in some applications, for instance when $\vec{v}$ is the private key in the encryption scheme $\mathcal{PE}$ (Fig. 4), it is beneficial to spend more time on the optimization of $\vec{v}$ during the pre-computation phase.

The random sampling from $\mathcal{CL}$ proposed in this section requires the pre-computation of an ideal class group structure and therefore cannot be used with large class

groups. However the group size threshold keeps increasing due to the development of computational resources and algorithms. The up-to-date class group computation record reported by Biasse [4] employs a 366-bit discriminant. This is enough for achieving the 112-bit security level, as discussed later in Sect. 7.

6.2. PSEUDO-RANDOM SAMPLING FROM LARGE $\mathcal{CL}$. We show how to implement the sampling operation without prior knowledge of the class group structure.

An algorithm for random sampling proposed by Srinivasan [41] outputs an ideal of a large norm. To further factor this ideal over the smooth factor base (2), techniques from index calculus algorithms for imaginary quadratic fields could be used as it is done by Galbraith et al. [19]. However this approach would have exponential running time and still yield an ideal representation which is slow in terms of its action on $\mathcal{ELL}_{p,n}$, as compared to optimized representations discussed in Sect. 6.1.

We propose to use a non-uniform probability distribution S on the set of elements of $\mathcal{CL}$ instead of the uniform distribution R. Note that in order to complete the proofs of Theorems 1 and 2 we have to additionally assume that R and S are computationally indistinguishable. Several authors, including Galbraith [18], Jao et al. [27] and Teske [44], construct samplings from $\mathcal{CL}$ that employ ideals with small split norms, in order to emulate the random sampling from $\mathcal{CL}$. Below we propose a candidate probability distribution S that is constructed to optimize the speed of the $\mathcal{CL}$ action on $\mathcal{ELL}$. How plausible it is that S is computationally indistinguishable from R is a difficult question that requires further analysis.

The probability distribution S is constructed as follows. For a fixed $l_{\max}$ we use the notations Δ, d, $\mathfrak{l}_i$ and ϕ from Sect. 4.2. Let also $h \approx 0.46(-\Delta)^{1/2}$ be an approximation for the class number $\#\mathcal{CL}(\mathcal{O}_\Delta)$. We then choose a set $V \subset \mathbb{Z}^d$ such that $\#V = ch$ for a small $c > 1$, and the random sampling from V is easy to implement. For instance, if V is the set of vectors inside a d-dimensional box

$$V = \{\vec{v}\colon\ -\hat{v}_i \le v_i \le \hat{v}_i,\ 1 \le i \le d\} \tag{13}$$

defined by non-negative integers $\hat{v}_i$, $1 \le i \le d$, then the random sampling from V can be achieved through d random samplings of the coordinates. We define S to be the probability distribution on $\mathcal{CL}$ induced by the uniform probability distribution on V and the map (4).

To construct the box V we start with a d-cube. Since for smaller primes l_i the action $[\mathfrak{l}_i] * E$ can be computed faster, we stretch the box V along the faster dimensions and squeeze it along the slower ones, so that the average time used for the computation along the i-th axis is the same for all the dimensions $1 \le i \le d$. The values $\hat{v}_i$ in (13) can be computed using the timing vector $\vec{t}$ (see Sect. 6.1). This approach has an advantage against timing side-channel attacks, as the running time of the group action is almost the same for different randomly chosen vectors from V.

7. SECURITY OF $\mathcal{ELL}$-BASED CRYPTOGRAPHIC SCHEMES

In this section we discuss the plausibility of the DDHAP and the ES assumptions with respect to the cryptographic schemes based on the $\mathcal{CL}$ action on $\mathcal{ELL}_{p,n}$.

7.1. PLAUSIBILITY OF THE $\mathcal{CL}$-DDHAP ASSUMPTION. The DDHAP formulated for the $\mathcal{CL}$ action on $\mathcal{ELL}_{p,n}$ ($\mathcal{CL}$-DDHAP) has not been considered in the literature. As far as we are concerned, the most efficient approach is to solve the corresponding $\mathcal{CL}$ group action inverse problem ($\mathcal{CL}$-GAIP).

Let us estimate the computational complexity of the $\mathcal{CL}$-GAIP. Galbraith et al. proposed an algorithm for constructing isogenies between elliptic curves [19]. Stages 2 and 3 of this algorithm particularly solve the $\mathcal{CL}$-GAIP, that is, find an ideal that maps a given elliptic curve to another given curve in the set $\mathcal{ELL}_{p,n}(\mathcal{O}_K)$. The algorithm is based on the Pollard's rho method [32] and requires approximately $(\pi h)^{1/2}$ hops between l_i-isogenous curves, $l_i \in L$, $h = \#\mathcal{CL}$. We have estimated the running time of one hop by (11), so the average running time of the algorithm equals $O(h^{1/2}\sum_{i=1}^{d}\frac{1}{d}l_i \log l_i \log p)$. After using the approximations for h, d and l_i as in Sect. 5 this becomes

$$O(p^{1/4}(\log^2 p)\log\log p) \tag{14}$$

multiplications in $\mathbb{F}_p$. We do not count the $O(p^{1/4+\epsilon})$ complexity of finding a short smooth representation of the resulting ideal.

In order to choose appropriate system parameters we use cryptographic security levels defined by the European Network of Excellence in Cryptology II (ECRYPT2) [47]. Computational complexities of 80, 96, 112 and 128 bits are assumed to be infeasible during the next 4, 10, 20 and 30 years, respectively. The size of p is chosen such that the number of bits in (14) equals the corresponding security level recommendation. The resulting values of $\log p$ are listed in Table 1.

7.2. SOLVING THE $\mathcal{CL}$-GAIP WITH A QUANTUM COMPUTER. Quantum computers allow to solve certain computational problems with a significantly greater efficiency than classical computers. An intriguing question is whether the $\mathcal{CL}$-GAIP can be efficiently solved on a quantum computer.

Since the problem involves the action of an ideal class group $\mathcal{CL}$ of an imaginary quadratic field, we will firstly review current advances in the computations in $\mathcal{CL}$. In the classical computation case, sub-exponential algorithms have been proposed for computing the structure of $\mathcal{CL}$ and solving the discrete logarithm problem (DLP) in a cyclic subgroup of $\mathcal{CL}$ (see, for example, Jacobson [25, 26]). Note however that these results do not imply sub-exponential complexity for the $\mathcal{CL}$-GAIP, which is still exponential-time (14).

In the quantum computation case, a polynomial-time algorithm for the structure of $\mathcal{CL}$ has been described by Hallgren [23]. The ability of quantum computer to solve the hidden subgroup problem is employed for computing the lattice of relations between generators of $\mathcal{CL}$. Schmidt [36] has carefully described an implementation of Shor's algorithm for solving the DLP in a cyclic subgroup of $\mathcal{CL}$ and estimated the necessary quantum register size.

These classical and quantum computation results for problems in the ideal class group do not apply to the $\mathcal{CL}$-GAIP. Indeed, the $\mathcal{CL}$-GAIP is defined for elements of the set $\mathcal{ELL}_{p,n}$ that the group $\mathcal{CL}$ acts on. One is given the j-invariants of two elliptic curves $E_x, E_y \in \mathcal{ELL}_{p,n}$, and the problem is to find an ideal $\mathfrak{r}$ such that $E_y = \mathfrak{r} * E_x$. In order to reduce the $\mathcal{CL}$-GAIP to a similar problem over $\mathbb{C}$ and look at the relationship between corresponding complex lattices, one has to lift these elliptic curves to $\mathbb{C}$, namely to solve the following problem:

Problem 3 (Coherent Lifting Problem)**.** *For given ordinary elliptic curves $E_x/\mathbb{F}_p$, $E_y/\mathbb{F}_p$ with $\mathrm{End}(E_x) \cong \mathrm{End}(E_y) \cong \mathcal{O}_K$, find complex lattices Λ_x, Λ_y such that there is a prime $\mathfrak{P} \mid p$ of the Hilbert class field H of K for which*

$$E_{\Lambda_x} \pmod{\mathfrak{P}} \cong E_x\,,$$
$$E_{\Lambda_y} \pmod{\mathfrak{P}} \cong E_y\,.$$

One can solve the coherent lifting problem by constructing H through the computation of the class polynomial $H_\Delta = \prod_{i=1}^{h}(X - j(\tau_i))$, where the complex numbers τ_i are obtained from the reduced representatives of the elements of $\mathcal{CL}$. One then finds a prime $\mathfrak{P}$ of H and reduces the values $j(\tau_i)$ modulo $\mathfrak{P}$. Since the degree of H_Δ is $h = \#\mathcal{CL}$, this approach requires time and space exponential in $\log h$. In fact, the best way to solve the coherent lifting problem seems to be to solve the $\mathcal{CL}$-GAIP first.

Now we will try to apply Shor's DLP algorithm [38] directly to the elements of $\mathcal{ELL}_{p,n}$. The original algorithm relies on the following idea. Let $y = x^r$ in a finite cyclic group, so that the DLP asks to compute r knowing x and y. A function $f(a,b) = x^a y^b$ has the value $(r,-1)$ as its period, since $f(a,b) = f(a+r, b-1)$. When $f(a,b)$ is implemented in quantum gates, one can efficiently find the period by means of the quantum Fourier transform, thus obtaining r. Quantum computers use reversible computation, and implementing a deterministic function on a quantum computer reversibly requires as much space as it does time. So it is essential for implementing the Shor's algorithm that the periodic function is polynomial-time. For solving the $\mathcal{CL}$-GAIP we try to construct a function $\hat{f}(\mathfrak{a},\mathfrak{b})$ similar to $f(a,b)$, that takes imaginary quadratic ideals $\mathfrak{a}$ and $\mathfrak{b}$. Even though it is possible to compute $\mathfrak{a} * E_x$ and $\mathfrak{b} * E_y$, we do not know of any polynomial-time composition operation for the two obtained elliptic curves that would be suitable for the purpose. We leave it as an open question to find a polynomial-time periodic function on $\mathcal{ELL}_{p,n}$ with the period dependent on $\mathfrak{r}$, such that $E_y = \mathfrak{r} * E_x$.

It has been recently proposed to use quantum computers for solving the hidden shift problem. The problem asks, for a given finite group G, a finite set X and maps $f, g\colon G \to X$ such that $g(a) = f(a+r)$ for all $a \in G$ and a fixed shift $r \in G$, to find r. When applied to the $\mathcal{CL}$-GAIP, these functions can be defined as $g(\mathfrak{a}) = \mathfrak{a} * E_y$ and $f(\mathfrak{a}) = \mathfrak{a} * E_x$. Then $\mathfrak{r}$ is a (multiplicative) shift because $f(\mathfrak{ar}) = \mathfrak{ar} * E_x = \mathfrak{a} * E_y = g(\mathfrak{a})$. However polynomial-time quantum algorithms for the hidden shift problem have been described only for some special types of functions, namely the Legendre symbol [17] and several classes of bent functions, a type of boolean functions [35].

7.3. PLAUSIBILITY OF THE ES ASSUMPTION. The ES assumption about a hash function family $\mathcal{H} = \{H_k\colon k \in K\}$ concerns the hash function's ability to extract entropy from a "partially random" source. In the $\mathcal{ELL}$-based $\mathcal{PE}$ scheme the hash function $H_k(u)$ is applied to the elliptic curve u represented by its j-invariant. Since there is $\log p$ bits in the representation of a j-invariant as an element of $\mathbb{F}_p$, but only $h \approx p^{1/2}$ elliptic curve isomorphism classes exist in the $\mathcal{ELL}_{p,n}$, the entropy of the random variable u is approximately $\frac{1}{2}\log p$ bits. Thus, on input u, the hash function H_k should output an almost uniformly distributed string of $\frac{1}{2}\log p$ bits. A similar problem appears, for example, when the Diffie-Hellman key agreement protocol is implemented in, say, a 160-bit multiplicative subgroup of a 1024-bit finite field. The shared secret output by the key agreement protocol needs to be transformed into a secret keying material. A function that implements this transformation is called a key derivation function. The National Institute of Standards and Technology (NIST) defines two approved key derivation functions [46]. The length of the output keying material in these functions is adjustable. The NIST key derivation functions are based on hash algorithms approved by NIST, namely the SHA-1 function and the SHA-2 family of hash functions. The applicability for key derivation is also a

Security (bits)	$\lceil \log p \rceil$ (bits)	Time (seconds)
75	224	19
80	244	21
96	304	56
112	364	90
128	428	229

TABLE 1. Average running time of one $\mathcal{CL}$ action on $\mathcal{ELL}_{p,n}$.

requirement for the upcoming SHA-3 family of hash functions. These facts suggest that it is indeed possible to construct an ES hash function family suitable for the $\mathcal{ELL}$-based $\mathcal{PE}$ scheme.

8. NUMERICAL EXPERIMENTS

In this section we report about a trial implementation of the arithmetic used in the proposed cryptographic schemes[3].

We have implemented the $\mathcal{CL}$ group action on $\mathcal{ELL}_{p,n}$ using the computer algebra system PARI/GP 2.4.3 created by Cohen, Belabas et al. The exponentiation in $\mathbb{F}_p[x]/f(x)$ for degrees of $f(x)$ higher than approximately 70 is more efficiently realized in the Number Theory Library (NTL) 5.5.2 by Shoup, so we make external calls to NTL when it is appropriate. PARI and NTL libraries were compiled with GNU Multiple Precision Arithmetic Library (GMP) 5.0.0. A database of Atkin modular polynomials of levels 3–499 was downloaded from the web site of the Elliptic Curves and Higher Dimensional Analogues (ECHIDNA) project by Kohel. For the class group structure computation we used the `quadratic_order` class implemented by Jacobson in the LiDIA 2.2.0 library.

The largest discriminant for which we could compute the class group structure using LiDIA was 226 bits long. This took approximately 3 hours and the process occupied 3 gigabytes of memory space. This size of discriminant corresponds to 75 bits of security and we have incuded it in our results. The class group structure was employed in the random sampling as described in Sect. 6.1. In the top row of Table 1, Time is the average time used for one random sampling of $\vec{v}$ followed by an optimization of $\vec{v}$ that lasts about 2 seconds and then the action $\phi(\vec{v}) * E$. Longer optimisation runs generally yield better group action times, which is relevant when the optimised vector can be precomputed off-line.

Table 1 shows the average time for one $\mathcal{CL}$ action on $\mathcal{ELL}_{p,n}$. For security levels 80–128, the Time column is the average time for a sampling of $\vec{v}$ from V, as described in Sect. 6.2, followed by the action $\phi(\vec{v}) * E$.

The system parameters which were used for time measurements in Table 1 are listed in Appendix A.

We stress that the provided time estimations are valid for one class group action. For the $\mathcal{PE}$ scheme, an encryption requires two group actions that can be computed in parallel. Since modern processors often have several processing cores the "wall clock" encryption time can be treated as one group action time. The decryption in the $\mathcal{PE}$ scheme takes one group action. In the three proposed $\mathcal{KA}$ protocols each

[3]The source code of the implementation is available at the author's personal web page, currently at http://www.item.ntnu.no/people/personalpages/phd/anton/software.

party has to perform two consecutive group actions. However the first action can be precomputed before the protocol starts.

Timings in Table 1 were obtained on a Ubuntu Linux 9.04 system with Intel Core i7 920 processor clocked at 3.6 GHz. The implementation is single-threaded, so only one of the four processor cores was used at a time. Note that whereas PARI, NTL and GMP are fast computation oriented libraries, a customized implementation of the arithmetic may result in better speeds. For instance the reduction modulo p can be implemented faster for $p = 2^n \pm a$ for small a, as compared to the generic long division algorithm used in GMP. Also the polynomial multiplication and division operations can be efficiently parallelized [5].

9. Concluding remarks

Let us compare the proposed cryptographic schemes with the ones based on the ECDLP, namely the elliptic curve Diffie-Hellman key agreement scheme, the elliptic curve digital signature algorithm and others. For an elliptic curve over a field $\mathbb{F}_q$, the ECDLP is usually considered in a cyclic subgroup of order approximately q. For the $\mathcal{CL}$-GAIP, the cardinality of the set $\mathcal{ELL}_{p,n}$ is the class number h, and the elliptic curves are defined over $\mathbb{F}_p$, where $p \approx h^2$. We shall consider the ECDLP and the $\mathcal{CL}$-GAIP of similar sizes q and h, respectively. First of all, we note that the system parameters for an ECDLP-based cryptographic scheme can be generated in polynomial in $\log q$ time, whereas a $\mathcal{CL}$-GAIP-based cryptographic scheme requires sub-exponential in $\log h$ time for computing the class group structure. The second aspect is the cryptosystem running time. The average running time of one scalar multiplication on an elliptic curve is $10 \log q$ [24], or $O(\log q)$ multiplications in $\mathbb{F}_q$. For the $\mathcal{CL}$ action on $\mathcal{ELL}$, the average running time is $O(\log^{3.7} h)$ multiplications in $\mathbb{F}_p$, which is much slower than for the ECDLP-based schemes. The third aspect we will consider is the computational complexity of the problems. The ECDLP complexity for a cryptographically strong elliptic curve is widely believed to be $O(q^{1/2} \log q)$ field operations. The computational complexity of the $\mathcal{CL}$-GAIP is $O(h^{1/2}(\log^2 h) \log\log h)$ multiplications in $\mathbb{F}_p$, i.e. the $\mathcal{CL}$-GAIP has higher complexity than the ECDLP.

It is not yet clear whether the $\mathcal{CL}$-GAIP can be efficiently solved on a quantum computer. Arguments against the applicability of some currently known quantum algorithms have been provided in Sect. 7.2. In case a quantum attack is discovered later, the proposed cryptographic schemes would seemingly become of theoretical interest only.

The encryption scheme and the key agreement protocols proposed in this paper use random sampling from the ideal class group $\mathcal{CL}$. Since the implementation of the $\mathcal{CL}$ action on $\mathcal{ELL}_{p,n}$ employs short smooth ideal representations, efficient random sampling is only possible for class groups with known structure. This provides security levels of up to 112 bits, as of 2009 class group computation records. Higher security levels are only achievable with a pseudo-random sampling which has to offer good randomness characteristics. It should be also noted that when the class group structure is not pre-computed, the fact that all elements of $\mathcal{CL}$ can be represented and used in the cryptographic scheme depends on the GRH.

Acknowledgements

The author would like to thank Prof. Alexander Rostovtsev, Prof. Alexei Rudakov and Prof. Stig F. Mjølsnes for their valuable support during the period

of work on this topic. Very useful comments were also received from the Advances in Mathematics of Communications journal's anonymous reviewers. The author is grateful to Dr. Steven Galbraith for his feedback on this paper and for suggestion of the coherent lifting problem.

References

[1] L. Babai, *On Lovász' lattice reduction and the nearest lattice point problem*, Combinatorica, **6** (1986), 1–13.
[2] K. Belabas, F. Diaz y Diaz and E. Friedman, *Small generators of the ideal class group*, Math. Comp., **77** (2008), 1185–1197.
[3] M. Bellare, A. Desai, D. Pointcheval and P. Rogaway, *Relations among notions of security for public-key encryption schemes*, in "Advances in Cryptology—CRYPTO 1998" (ed. H. Krawczyk), Springer, (1998), 26–46.
[4] J.-F. Biasse, *Improvements in the computation of ideal class group of imaginary quadratic number fields*, Adv. Math. Commun., **4** (2010), 141–154.
[5] D. Bini and V. Y. Pan, "Polynomial and Matrix Computations I," Birkhäuser Boston Inc., Boston, MA, 1994.
[6] R. Broker, K. Lauter and A. V. Sutherland, *Modular polynomials via isogeny volcanoes*, preprint, arXiv:1001.0402.
[7] C. Cachin and U. Maurer, *Smoothing probability distributions and smooth entropy (extended abstract)*, Institute for Theoretical Computer Science, ETH Zürich, preprint, 1996.
[8] C. Cachin and U. Maurer, *Smoothing probability distributions and smooth entropy*, in "IEEE International Symposium on Information Theory (ISIT 1997)," (1997), 91.
[9] R. Canetti and H. Krawczyk, *Analysis of key-exchange protocols and their use for building secure channels*, in "Advances in Cryptology—EUROCRYPT 2001 (Innsbruck)," Springer, (2001), 453–474.
[10] R. Canetti and H. Krawczyk, *Analysis of key-exchange protocols and their use for building secure channels*, Cryptology ePrint Archive, available online at http://eprint.iacr.org/2001/040.
[11] D. Charles and K. Lauter, *Computing modular polynomials*, LMS J. Comput. Math., **8** (2005), 195–204.
[12] D. X. Charles, K. E. Lauter and E. Z. Goren, *Cryptographic hash functions from expander graphs*, J. Cryptology, **22** (2009), 93–113.
[13] H. Cohen, "A Course in Computational Algebraic Number Theory," Springer-Verlag, Berlin, 1993.
[14] J.-M. Couveignes, *Hard homogeneous spaces*, Cryptology ePrint Archive, available online at http://eprint.iacr.org/2006/291.
[15] J.-M. Couveignes, L. Dewaghe and F. Morain, "Isogeny Cycles and the Schoof-Elkies-Atkin Algorithm," research report LIX/RR/96/03, Laboratoire d'Informatique de l'Ecole Polytechnique, 1996.
[16] D. A. Cox, "Primes of the Form $x^2 + ny^2$," John Wiley & Sons Inc., New York, 1989.
[17] W. van Dam, S. Hallgren and L. Ip, *Quantum algorithms for some hidden shift problems*, SIAM J. Comput., **36** (2006), 763–778.
[18] S. D. Galbraith, *Constructing isogenies between elliptic curves over finite fields*, LMS J. Comput. Math., **2** (1999), 118–138.
[19] S. D. Galbraith, F. Hess and N. P. Smart, *Extending the GHS Weil descent attack*, in "Advances in Cryptology—EUROCRYPT 2002 (Amsterdam)" (ed. L.R. Knudsen), Springer, (2002), 29–44.
[20] O. Goldreich, "Foundations of Cryptography," Cambridge University Press, Cambridge, 2001.
[21] S. Goldwasser and S. Micali, *Probabilistic encryption*, J. Comput. System Sci., **28** (1984), 270–299.
[22] J. L. Hafner and K. S. McCurley, *A rigorous subexponential algorithm for computation of class groups*, J. Amer. Math. Soc., **2** (1989), 837–850.
[23] S. Hallgren, *Fast quantum algorithms for computing the unit group and class group of a number field*, in "STOC'05: Proceedings of the 37th Annual ACM Symposium on Theory of Computing," ACM, (2005), 468–474.

[24] H. Hisil, K. K.-H. Wong, G. Carter and E. Dawson, *Faster group operations on elliptic curves*, in "Australasian Information Security Conference (AISC 2009)" (eds. L. Brankovic and W. Susilo), ACS, (2009), 7–19.
[25] M. J. Jacobson, Jr., *Applying sieving to the computation of quadratic class groups*, Math. Comp., **68** (1999), 859–867.
[26] M. J. Jacobson, Jr., *Computing discrete logarithms in quadratic orders*, J. Cryptology, **13** (2000), 473–492.
[27] D. Jao, S. D. Miller and R. Venkatesan, *Do all elliptic curves of the same order have the same difficulty of discrete log?*, in "Advances in Cryptology—ASIACRYPT 2005," Springer, (2005), 21–40.
[28] D. Kohel, "Endomorphism Rings of Elliptic Curves over Finite Fields," Ph.D thesis, University of California at Berkeley, 1996.
[29] G. Maze, C. Monico and J. Rosenthal, *Public key cryptography based on semigroup actions*, Adv. Math. Commun., **1** (2007), 489–507.
[30] A. J. Menezes, P. C. van Oorschot and S. A. Vanstone, "Handbook of Applied Cryptography," CRC Press, Boca Raton, FL, 1997.
[31] C. J. Monico, "Semirings and Semigroup Actions in Public-Key Cryptography," Ph.D thesis, The Graduate School of the University of Notre Dame, Indiana, 2002.
[32] J. M. Pollard, *Monte Carlo methods for index computation* (mod p), Math. Comp., **32** (1978), 918–924.
[33] A. Rostovtsev, E. Makhovenko and O. Shemyakina, *Elliptic curve ordered digital signature*, Saint-Petersburg State Polytechnical University, preprint, available online at http://www.ssl.stu.neva.ru/ssl/archieve/ordered_digital_signature.pdf.
[34] A. Rostovtsev and A. Stolbunov, *Public-key cryptosystem based on isogenies*, Cryptology ePrint Archive, available online at http://eprint.iacr.org/2006/145.
[35] M. Rötteler, *Quantum algorithms for highly non-linear boolean functions*, preprint, arXiv:0811.3208.
[36] A. Schmidt, *Quantum algorithm for solving the discrete logarithm problem in the class group of an imaginary quadratic field and security comparison of current cryptosystems at the beginning of quantum computer age*, in "Emerging Trends in Information and Communication Security (ETRICS 2006)" (ed. G. Müller), Springer, (2006), 481–493.
[37] R. Schoof, *Counting points on elliptic curves over finite fields*, J. Théor. Nombres Bordeaux, **7** (1995), 219–254.
[38] P. W. Shor, *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer*, SIAM J. Comput., **26** (1997), 1484–1509.
[39] J. H. Silverman, "The Arithmetic of Elliptic Curves," Springer-Verlag, New York, 1986.
[40] J. H. Silverman, "Advanced Topics in the Arithmetic of Elliptic Curves," Springer-Verlag, New York, 1994.
[41] A. Srinivasan, *Computations of class numbers of real quadratic fields*, Math. Comput., **67** (1998), 1285–1308.
[42] A. Stolbunov, *Reductionist security arguments for public-key cryptographic schemes based on group action*, in "Norwegian Information Security Conference (NISK 2009)" (ed. S.F. Mjølsnes), Tapir Akademisk Forlag, (2009), 97–109.
[43] J. Tate, *Endomorphisms of abelian varieties over finite fields*, Invent. Math., **2** (1966), 134–144.
[44] E. Teske, *An elliptic curve trapdoor system*, J. Cryptology, **19** (2006), 115–133.
[45] W. C. Waterhouse, *Abelian varieties over finite fields*, Ann. Sci. École Norm. Sup., **2** (1969), 521–560.
[46] "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography (Revisited)," NIST special publication 800-56A, National Institute for Standards and Technology, 2007.
[47] "Yearly Report on Algorithms and Keysizes," ECRYPT2 report D.SPA.7, European Network of Excellence in Cryptology II, 2009.

Appendix A. System parameters used in time measurements

We use the notation param_s to refer to the s-bit security level specified in Table 1. The following parameters are listed in Table 2: p is the prime field characteristic;

Parameter	Value
p_{75}	$2^{224} - 63$
E_{75}	$y^2 = x^3 + x + 5217$
t_{75}	706283819635943803784155145600859
h_{75}	367259891656247020496958525412808 1
$l_{\max,75}$	163
p_{80}	$2^{243} + 59$
E_{80}	$y^2 = x^3 + x + 20321$
t_{80}	69756469485406525843258580792418758 1
$l_{\max,80}$	263
p_{96}	$2^{303} + 101$
E_{96}	$y^2 = x^3 + x + 3704$
t_{96}	57847001694414882341709570340537328669512200 27
$l_{\max,96}$	443
p_{112}	$2^{363} + 309$
E_{112}	$y^2 = x^3 + x + 1919$
t_{112}	−59344974584391105805850406935841437299180143301389290 37
$l_{\max,112}$	487
p_{128}	$2^{427} + 69$
E_{128}	$y^2 = x^3 + x + 1025$
t_{128}	14933846636205862089595788320503564108095537034421076949568186265
$l_{\max,128}$	499
$\{(l_i, \hat{v}_i)\}_{128}$	$(3, 2669)$, $(7, 1094)$, $(13, 475)$, $(17, 317)$, $(19, 269)$, $(29, 121)$, $(31, 107)$, $(37, 74)$, $(41, 64)$, $(53, 41)$, $(59, 37)$, $(61, 35)$, $(71, 25)$, $(83, 20)$, $(89, 19)$, $(97, 17)$, $(101, 17)$, $(103, 16)$, $(107, 14)$, $(127, 13)$, $(131, 9)$, $(137, 9)$, $(149, 8)$, $(167, 7)$, $(181, 6)$, $(197, 6)$, $(199, 6)$, $(223, 5)$, $(229, 5)$, $(239, 5)$, $(307, 2)$, $(311, 3)$, $(313, 2)$, $(317, 2)$, $(331, 2)$, $(337, 2)$, $(347, 2)$, $(367, 2)$, $(379, 2)$, $(383, 2)$, $(389, 1)$, $(409, 2)$, $(421, 1)$, $(449, 1)$, $(457, 1)$, $(461, 1)$, $(499, 1)$

TABLE 2. System parameters used in the numerical experiments.

E is an elliptic curve from the set $\mathcal{ELL}_{p,n}$, where $n = p + 1 - t$; t is the Frobenius trace for the curves in $\mathcal{ELL}_{p,n}$; h is the class number (calculated only for the 75-bit security); $l_{\max}$ is the maximal isogeny degree used in the implementation; $\{(l_i, \hat{v}_i)\}$ is the list of tuples consisting of an isogeny degree l_i and the corresponding hop limit $\hat{v}_i$, as in (13) (listed only for the 128-bit security). The elliptic curves were chosen to have fundamental Frobenius discriminants $\Delta_\pi = t^2 - 4p$ in order to ensure that their endomorphism rings are maximal imaginary quadratic orders.

Received June 2009; revised March 2010.

E-mail address: anton@item.ntnu.no