Postquantum Cryptography: what, why, and how?

SIMBA

Enric Florit Zacarías

November 27, 2019

Public-key cryptography

Imagine Alice and Bob want to communicate through a channel, but they've never met before.

Public-key cryptography

Imagine Alice and Bob want to communicate through a channel, but they've never met before.
How can they agree on a secret key to encrypt their communications, using e.g. AES?

Diffie and Hellman (1976)

Use the group $(\mathbb{Z}/p\mathbb{Z})^{\times} = \langle \alpha \rangle$.

Diffie and Hellman (1976)

Use the group $(\mathbb{Z}/p\mathbb{Z})^\times = \langle \alpha \rangle$.
Alice chooses a private key $1 < a < p$, and publishes $A = \alpha^a \bmod p$.

Diffie and Hellman (1976)

Use the group $(\mathbb{Z}/p\mathbb{Z})^\times = \langle \alpha \rangle$.
Alice chooses a private key $1 < a < p$, and publishes $A = \alpha^a \mod p$.
Bob chooses a private key $1 < b < p$, and publishes $B = \alpha^b \mod p$.

Diffie and Hellman (1976)

Use the group $(\mathbb{Z}/p\mathbb{Z})^\times = \langle \alpha \rangle$.
Alice chooses a private key $1 < a < p$, and publishes $A = \alpha^a \mod p$.
Bob chooses a private key $1 < b < p$, and publishes $B = \alpha^b \mod p$.
They may use the **shared secret** $A^b \equiv B^a \equiv \alpha^{ab} \mod p$.

Computational problems

Problem (Discrete Logarithm - DLP)

Given a cyclic group $G = \langle \alpha \rangle$ *and an element* $\beta \in G$*, find* $x \in \mathbb{Z}$ *such that* $\beta = \alpha^x$*.*

Computational problems

Problem (Discrete Logarithm - DLP)

Given a cyclic group $G = \langle \alpha \rangle$ *and an element* $\beta \in G$*, find* $x \in \mathbb{Z}$ *such that* $\beta = \alpha^x$*.*

Problem (Diffie-Hellman - DHP)

Given a cyclic group $G = \langle \alpha \rangle$ *and elements* $\alpha^a, \alpha^b \in G$*, find* α^{ab}*.*

Introduction: Diffie-Hellman

Why? Solving the DLP

What? Postquantum Cryptography

How? Isogenies and SIDH

Why? Solving the DLP

Let's see some algorithms to solve for discrete logarithms!

Problem (Discrete Logarithm - DLP)

Given a cyclic group $G = \langle \alpha \rangle$ and an element $\beta \in G$, find $x \in \mathbb{Z}$ such that $\beta = \alpha^x$.

Baby step – giant step

Let $m > \sqrt{N}$ be an integer. Then for every $x \leq N$, $x = am + b$, with $0 \leq a, b < m$.

Baby step – giant step

Let $m > \sqrt{N}$ be an integer. Then for every $x \leq N$, $x = am + b$, with $0 \leq a, b < m$.

1. Compute and store α^b, for $0 \leq b < m$.

Baby step – giant step

Let $m > \sqrt{N}$ be an integer. Then for every $x \leq N$, $x = am + b$, with $0 \leq a, b < m$.

1. Compute and store α^b, for $0 \leq b < m$.
2. Compute $\beta\alpha^{-am}$, for $0 \leq a < m$, and check for a match $\beta\alpha^{-am} = \alpha^b$.

Baby step – giant step

Let $m > \sqrt{N}$ be an integer. Then for every $x \leq N$, $x = am + b$, with $0 \leq a, b < m$.

1. Compute and store α^b, for $0 \leq b < m$.
2. Compute $\beta\alpha^{-am}$, for $0 \leq a < m$, and check for a match $\beta\alpha^{-am} = \alpha^b$.
3. If so, $\beta = \alpha^{am+b}$ and $x = am + b$.

Pohlig-Hellman

Idea: factor $N = \prod_{i=1}^{r} p_i^{e_i}$, and obtain $x \mod p_i^{e_i}$ for each i. Then use the Chinese Remainder Theorem to combine the information.

Pohlig-Hellman

Idea: factor $N = \prod_{i=1}^{r} p_i^{e_i}$, and obtain $x \mod p_i^{e_i}$ for each i. Then use the Chinese Remainder Theorem to combine the information.

If $p^e \mid N$, then α^{N/p^e} has order p^e, and $\beta^{N/p^e} = \left(\alpha^{N/p^e}\right)^x$. We can compute $x \mod p^e$!

Pohlig-Hellman

Idea: factor $N = \prod_{i=1}^{r} p_i^{e_i}$, and obtain $x \mod p_i^{e_i}$ for each i.
Then use the Chinese Remainder Theorem to combine the information.
If $p^e \mid N$, then α^{N/p^e} has order p^e, and $\beta^{N/p^e} = \left(\alpha^{N/p^e}\right)^x$. We can compute $x \mod p^e$!

*Only useful if N is smooth (all prime factors are small).

Index calculus

It applies to finite fields: $\mathbb{Z}/p\mathbb{Z}$ and $\mathbb{F}_{p^r}$.

Index calculus

It applies to finite fields: $\mathbb{Z}/p\mathbb{Z}$ and $\mathbb{F}_{p^r}$.

1. Choose a **factor base** $\mathcal{S}$. For each $g_i \in \mathcal{S}$ we will compute the integer y_i for which $g_i = \alpha^{y_i}$.

Index calculus

It applies to finite fields: $\mathbb{Z}/p\mathbb{Z}$ and $\mathbb{F}_{p^r}$.

1. Choose a **factor base** $\mathcal{S}$. For each $g_i \in \mathcal{S}$ we will compute the integer y_i for which $g_i = \alpha^{y_i}$.
2. Find a relation of the form $\alpha^k \beta = \prod_{i=1}^{t} g_i^{e_i}$.

Index calculus

It applies to finite fields: $\mathbb{Z}/p\mathbb{Z}$ and $\mathbb{F}_{p^r}$.

1. Choose a **factor base** $\mathcal{S}$. For each $g_i \in \mathcal{S}$ we will compute the integer y_i for which $g_i = \alpha^{y_i}$.
2. Find a relation of the form $\alpha^k \beta = \prod_{i=1}^{t} g_i^{e_i}$.
3. The discrete logarithm will be

$$x = \log_\alpha(\beta) = \sum_{i=1}^{t} e_i \log_\alpha(g_i) - k = \sum_{i=1}^{t} e_i y_i - k.$$

Index calculus

This algorithm has the best complexity: it is subexponential!

$$L_n[t, \gamma] = e^{(\gamma + o(1))(\log n)^t (\log \log n)^{1-t}}$$

Index calculus

This algorithm has the best complexity: it is subexponential!

$$L_n[t, \gamma] = e^{(\gamma + o(1))(\log n)^t (\log \log n)^{1-t}}$$

If $t = 0$, then $L_n[0, \gamma] = (\log n)^{\gamma + o(1)}$ is polynomial in $\log n$.

Index calculus

This algorithm has the best complexity: it is subexponential!

$$L_n[t,\gamma] = e^{(\gamma+o(1))(\log n)^t(\log\log n)^{1-t}}$$

If $t = 0$, then $L_n[0,\gamma] = (\log n)^{\gamma+o(1)}$ is polynomial in $\log n$.
If $t = 1$, then $L_n[1,\gamma] = n^{\gamma+o(1)}$ is exponential in $\log n$.

Summary of complexities

Algorithm	Complexity
Exhaustive search	$O(N)$
Baby step – giant step	Time $O(\sqrt{N})$, memory $O(\sqrt{N})$
Pohlig-Hellman	$O(\sum_{i=1}^{r} e_i(\log N + \sqrt{p_i}))$
Index calculus in $\mathbb{F}_{p^n}$	$L_{p^n}[1/2, \sqrt{2}]$
NFS-DLP in $\mathbb{F}_{\mathbf{p^n}}$	$\mathbf{L_{p^n}[1/3, c]}$

Table: Algorithms solving DLP in a group of order $N = \prod_{i=1}^{r} p_i^{e_i}$.

Shor's algorithm

In 1994, Peter Shor [Sho94] published a quantum algorithm that would *factor integers* and solve *discrete logarithms* in **polynomial time**...

Shor's algorithm

In 1994, Peter Shor [Sho94] published a quantum algorithm that would *factor integers* and solve *discrete logarithms* in **polynomial time**...

... but don't worry, because quantum computers are just theoretical.

Shor's algorithm

In 1994, Peter Shor [Sho94] published a quantum algorithm that would *factor integers* and solve *discrete logarithms* in **polynomial time**...

... but don't worry, because quantum computers are just theoretical.

Right?

Shor's algorithm

In 1994, Peter Shor [Sho94] published a quantum algorithm that would *factor integers* and solve *discrete logarithms* in **polynomial time**...
... but don't worry, because quantum computers are just theoretical.

Right?

Right?

Why Post-Quantum Cryptography, then?

PQCRYPTO EU-Project
"The EU and governments around the world are investing heavily in building quantum computers; society needs to be prepared for the consequences, including cryptanalytic attacks accelerated by these computers." [Lan15]

Why Post-Quantum Cryptography, then?

NIST's Report on Post-Quantum Cryptography
"Some experts even predict that within the next 20 or so years, sufficiently large quantum computers will be built to break essentially all public key schemes currently in use." [Moo+16]

Introduction: Diffie-Hellman

Why? Solving the DLP

What? Postquantum Cryptography

How? Isogenies and SIDH

What is Postquantum Cryptography?

A postquantum cryptosystem must meet two requirements:

What is Postquantum Cryptography?

A postquantum cryptosystem must meet two requirements:

1. It must be efficient to use with existing hardware.

What is Postquantum Cryptography?

A postquantum cryptosystem must meet two requirements:

1. It must be efficient to use with existing hardware.
2. It must be resistent both to classical and quantum adversaries.

What do we need to develop?

We can't use ciphers based on **discrete logarithms** (Diffie-Hellman) or **integer factorization** (RSA). That is, we need to look for new kinds of asymmetric encryption.

What do we need to develop?

We can't use ciphers based on **discrete logarithms** (Diffie-Hellman) or **integer factorization** (RSA). That is, we need to look for new kinds of asymmetric encryption.

However, "symmetric algorithms [...] should be usable in a quantum era", because breaking them usually involves brute-force search in the key space, and "doubling the key size will be sufficient to preserve security" [Moo+16].

What techniques are involved in PQ Cryptography?

- Lattice-based cryptography
- Code-based cryptography
- **Isogeny-based cryptography**

Elliptic curves

Let K be a field of characteristic different from $2, 3$, and $A, B \in K \subseteq L$ with $4A^3 + 27B^2 \neq 0$. An **elliptic curve** E is the set of points (x, y) that satisfy the equation

$$E\colon y^2 = x^3 + Ax + B.$$

Elliptic curves

Let K be a field of characteristic different from $2, 3$, and $A, B \in K \subseteq L$ with $4A^3 + 27B^2 \neq 0$. An **elliptic curve** E is the set of points (x, y) that satisfy the equation

$$E\colon y^2 = x^3 + Ax + B.$$

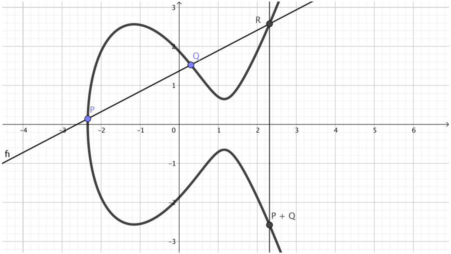
More precisely, we define the set of L-rational points,

$$E(L) := \{(x, y) \in L \times L \mid y^2 = x^3 + Ax + B\} \cup \{\mathcal{O}\}.$$

In homogeneous coordinates, the equation is $y^2 z = x^3 + Axz^2 + Bz^3$, and $\mathcal{O} = (0 : 1 : 0)$ is the only **point at infinity**.

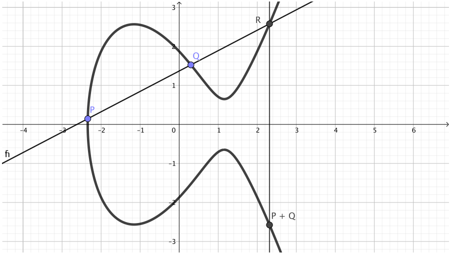
Elliptic curves are groups

Given two points $P, Q \in E(K)$, we define an operation on the points:



Elliptic curves are groups

Given two points $P, Q \in E(K)$, we define an operation on the points:



Theorem

The set $E(K)$ with the operation $+$ is an abelian group.

The j-invariant

Given a curve E: $y^2 = x^3 + Ax + B$, its j-invariant is

$$j(E) = 1728 \frac{4A^3}{4A^3 + 27B^2}.$$

The j-invariant

Given a curve E: $y^2 = x^3 + Ax + B$, its j-invariant is

$$j(E) = 1728 \frac{4A^3}{4A^3 + 27B^2}.$$

Two curves are isomorphic over $\bar{K}$ if and only if they have the same j-invariant.

The j-invariant

Given a curve E: $y^2 = x^3 + Ax + B$, its j-invariant is

$$j(E) = 1728 \frac{4A^3}{4A^3 + 27B^2}.$$

Two curves are isomorphic over $\bar{K}$ if and only if they have the same j-invariant.
For each $j_0 \in \bar{K}$, there exists a curve E with $j(E) = j_0$.

Isogenies

Given two elliptic curves E_1, E_2 over K, an **isogeny** between them is a non-constant map

$$\phi\colon E_1(\bar{K}) \to E_2(\bar{K})$$

that is both a morphism of algebraic curves and a group homomorphism.

Isogenies

Given two elliptic curves E_1, E_2 over K, an **isogeny** between them is a non-constant map

$$\phi\colon E_1(\bar{K}) \to E_2(\bar{K})$$

that is both a morphism of algebraic curves and a group homomorphism.
Isogenies can be put in a **standard form**:

$$\phi(x,y) = \left(\frac{p(x)}{q(x)}, y\frac{s(x)}{t(x)}\right)$$

Multiplication by n

The *multiplication-by-n* map $[n]\colon E \to E$ is an isogeny for all non-zero $n \in \mathbb{Z}$. Its kernel is written as $E[n]$, the **group of n-torsion points**.

Multiplication by n

The *multiplication-by-n* map $[n]\colon E \to E$ is an isogeny for all non-zero $n \in \mathbb{Z}$. Its kernel is written as $E[n]$, the **group of n-torsion points**.

Let $p = \operatorname{char} K$. For any prime $\ell \neq p$, we have $E[\ell^n] \cong \mathbb{Z}/\ell^n\mathbb{Z} \times \mathbb{Z}/\ell^n\mathbb{Z}$. This group has $\ell^{n-1}(\ell + 1)$ cyclic subgroups of order ℓ^n.

Quotient curve

Every isogeny $\phi \colon E_1 \to E_2$ has finite kernel, a subgroup $G \subset E_1(\bar{K})$.

Quotient curve

Every isogeny $\phi\colon E_1 \to E_2$ has finite kernel, a subgroup $G \subset E_1(\bar{K})$.

Theorem

Let E_1 be an elliptic curve over K, and let G be a finite subgroup of $E_1(\bar{K})$. There exist a curve E_2 and an isogeny $\phi\colon E_1 \to E_2$, such that $\ker\phi = G$. Moreover, ϕ and E_2 are unique up to isomorphism.

We will write $E_2 = E_1/G$.

Hasse's theorem

Theorem

Let E be an elliptic curve defined over a finite field $\mathbb{F}_q$, $q = p^r$. The number of $\mathbb{F}_q$-rational points of E is

$$\#E(\mathbb{F}_q) = q + 1 - t,$$

with $|t| \leq 2\sqrt{q}$.

Supersingular curves

Theorem

Let E be a curve over a finite field $\mathbb{F}_q$, $q = p^r$. TFAE:

- *E is supersingular.*
- $E[p] = \{\mathcal{O}\}$.
- *$[p]$ is purely inseparable.*
- $\#E(\mathbb{F}_q) = q + 1 - t$, *with* $t \equiv 0 \mod p$.
- *$End(E) \otimes_{\mathbb{Z}} \mathbb{Q}$ is a quaternion algebra.*

Given a prime p, there are about $p/12$ supersingular elliptic curve isomorphism classes defined over $\bar{\mathbb{F}}_p$.

Supersingular Isogeny Diffie Hellman - Setting

Let $p = 2^{e_A}3^{e_B} - 1$ be a prime with $2^{e_A} \approx 3^{e_B}$, set $K = \mathbb{F}_{p^2}$.

Supersingular Isogeny Diffie Hellman - Setting

Let $p = 2^{e_A}3^{e_B} - 1$ be a prime with $2^{e_A} \approx 3^{e_B}$, set $K = \mathbb{F}_{p^2}$.

The curve $E_0 \colon y^2 = x^3 + x$ is supersingular, and

$$\#E_0(\mathbb{F}_{p^2}) = (p+1)^2 = (2^{e_A}3^{e_B})^2.$$

Supersingular Isogeny Diffie Hellman - Setting

Let $p = 2^{e_A}3^{e_B} - 1$ be a prime with $2^{e_A} \approx 3^{e_B}$, set $K = \mathbb{F}_{p^2}$.

The curve $E_0 \colon y^2 = x^3 + x$ is supersingular, and

$$\#E_0(\mathbb{F}_{p^2}) = (p+1)^2 = (2^{e_A}3^{e_B})^2.$$

We have $E_0[2^{e_A}] = \langle P_A, Q_A \rangle,\ E_0[3^{e_B}] = \langle P_B, Q_B \rangle \subset E_0(\mathbb{F}_{p^2})$.

SIDH - Private keys

Alice chooses a pair $(m_A, n_A) \in \mathbb{Z}/2^{e_A}\mathbb{Z} \times \mathbb{Z}/2^{e_A}\mathbb{Z}$ (not both divisible by 2). This is her private key.

SIDH - Private keys

Alice chooses a pair $(m_A, n_A) \in \mathbb{Z}/2^{e_A}\mathbb{Z} \times \mathbb{Z}/2^{e_A}\mathbb{Z}$ (not both divisible by 2). This is her private key.

Bob chooses a pair $(m_B, n_B) \in \mathbb{Z}/3^{e_B}\mathbb{Z} \times \mathbb{Z}/3^{e_B}\mathbb{Z}$ (not both divisible by 3). This is his private key.

SIDH - Key exchange

$$
\begin{tikzcd}
 & E_0/\langle [m_A]P_A + [n_A]Q_A \rangle \\
E_0 \arrow[ur, "\phi_A"] \arrow[dr, "\phi_B"'] & \\
 & E_0/\langle [m_B]P_B + [n_B]Q_B \rangle
\end{tikzcd}
$$

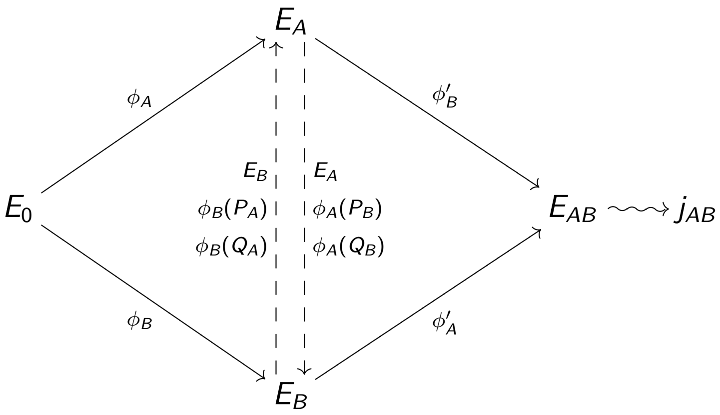
SIDH - Key exchange

$$
\begin{tikzcd}
 & E_A \\
E_0 \arrow[ur, "\phi_A"] \arrow[dr, "\phi_B"'] & \\
 & E_B
\end{tikzcd}
$$

SIDH - Key exchange

$$
\begin{tikzcd}
 & E_A \arrow[dd, dashed, shift left=2, "\begin{matrix} E_A \\ \phi_A(P_B) \\ \phi_A(Q_B) \end{matrix}"] \\
E_0 \arrow[ur, "\phi_A"] \arrow[dr, "\phi_B"'] & \\
 & E_B \arrow[uu, dashed, shift left=2, "\begin{matrix} E_B \\ \phi_B(P_A) \\ \phi_B(Q_A) \end{matrix}"]
\end{tikzcd}
$$

SIDH - Key exchange



SIDH - Key exchange

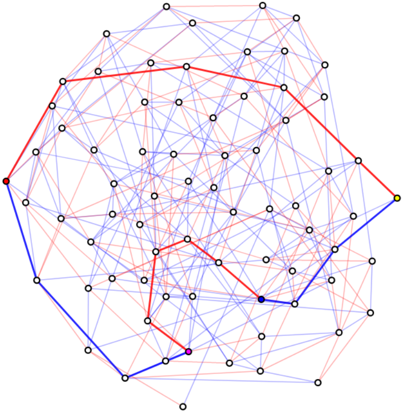


Figure: SIDH graph with $p = 2^5 3^3 - 1 = 863$.

Computational problems

Problem (Supersingular Isogeny problem (CSSI))

Let $\phi_A \colon E_0 \to E_A$ be an isogeny with kernel $\langle [m_A]P_A + [n_A]Q_A \rangle$, where m_A, n_A are chosen randomly in $\mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$ and not both divisible by ℓ_A.
Given the curves E_0, E_A and the values $\phi_A(P_B)$ and $\phi_A(Q_B)$, find a generator R_A of $\langle [m_A]P_A + [n_A]Q_A \rangle$.

Analog to DLP in the Diffie-Hellman setting.

Computational problems

Problem (Supersingular D.-H. problem (SSCDH))

Let

$$\begin{cases} \phi_A \colon E_0 \to E_A = E_0/\langle [m_A]P_A + [n_A]Q_A \rangle, \\ \phi_B \colon E_0 \to E_B = E_0/\langle [m_B]P_B + [n_B]Q_B \rangle \end{cases}$$

be isogenies defined as in the SIDH protocol.
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 the curve

$$E_0/\langle [m_A]P_A + [n_A]Q_A, [m_B]P_B + [n_B]Q_B \rangle.$$

Analog to DHP in the Diffie-Hellman setting.

SIDH security

- The same problems in the **ordinary** case (e.g., non-supersingular) can be solved with a quantum computer in subexponential time.
- The best strategy to break SIDH is almost brute-force, at $O(\sqrt[4]{p})$ and $O(\sqrt[6]{p})$ (exponential in $\log p \sim e_A, e_B$).
- It looks like the **auxiliary points** ($\phi_A(P_B)$ and so on) are revealing too much information, but so far nobody* has been able to exploit them.

SIDH/SIKE in production

KEM	Public Key size (bytes)	Ciphertext (bytes)	Secret size (bytes)	KeyGen (op/sec)	Encaps (op/sec)	Decaps (op/sec)	NIST level
HRSS-SXY	1138	1138	32	3952.3	76034.7	21905.8	1
SIKE/p434	330	346	16	367.1	228.0	209.3	1

Figure: Comparison between lattice-based HRSS-SXY and isogeny-based SIKE [Kwi19].

SIDH/SIKE in production

Figure: Ostrich vs turkey [KV19].

Conclusions

- Public-key cryptosystems based in RSA and Diffie-Hellman could be broken in a few years.
- Current efforts in finding and **testing** new postquantum standards.
- SIDH/SIKE is the most prominent isogeny-based cryptography proposal, *however* there are other constructions to explore (CGL, CSIDH, higher genus...).

References I

[Gor11] Dan Gordon. "Discrete Logarithm Problem". In: *Encyclopedia of Cryptography and Security*. Ed. by Henk C. A. van Tilborg and Sushil Jajodia. Boston, MA: Springer US, 2011, pp. 352–353. ISBN: 978-1-4419-5906-5. URL: https://doi.org/10.1007/978-1-4419-5906-5_445.

[KV19] Kris Kwiatkowski and Luke Valenta. *The TLS Post-Quantum Experiment*. Last accessed 24 November 2019. Oct. 2019. URL: https://blog.cloudflare.com/the-tls-post-quantum-experiment/.

References II

[Kwi19] Kris Kwiatkowski. *Towards Post-Quantum Cryptography in TLS*. Last accessed 24 November 2019. June 2019. URL: https://blog.cloudflare.com/towards-post-quantum-cryptography-in-tls/.

[Lan15] Tanja Lange. "Initial recommendations of long-term secure post-quantum systems". In: 2015.

[Moo+16] Dustin Moody et al. "NIST Report on Post-Quantum Cryptography". In: (Apr. 2016). DOI: 10.6028/NIST.IR.8105.

References III

[Ngu11] Kim Nguyen. “Index Calculus Method”. In: *Encyclopedia of Cryptography and Security*. Ed. by Henk C. A. van Tilborg and Sushil Jajodia. Boston, MA: Springer US, 2011, pp. 597–600. ISBN: 978-1-4419-5906-5. URL: https://doi.org/10.1007/978-1-4419-5906-5_454.

[Sho94] Peter W. Shor. “Algorithms for Quantum Computation: Discrete Logarithms and Factoring”. In: *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*. SFCS ’94. Washington, DC, USA: IEEE Computer Society, 1994, pp. 124–134. ISBN: 0-8186-6580-7.

Thank you!