SIDH and SIKE

Presentation · December 2019

DOI: 10.13140/RG.2.2.13624.90885

1 author:

Cansu Yener
Middle East Technical University
3 PUBLICATIONS **0** CITATIONS

SEE PROFILE

SIDH and SIKE

Cansu YENER

ODTÜ/Kriptografi

11.12.2019

Outline

- Background
- SIDH
- Security and Efficiency of SIDH
- SIKE
- Advantages, Disadvantages of SIKE
- Comparison

Mathematical Background

- The j-invariant of an elliptic curve given by the Weierstrass equation $y^2 = x^3 + ax + b$ is:

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

- Isomorphic curves have the same j-invariant; over an algebraically closed field, two curves with the same j-invariant are isomorphic.

- An isogeny $\phi : E \to E'$ between elliptic curves E and E' , is a rational map which is also a group homomorphism.

- If separable, ϕ is determined by its kernel up to an isomorphism of E'.

Random walk on supersingular graph

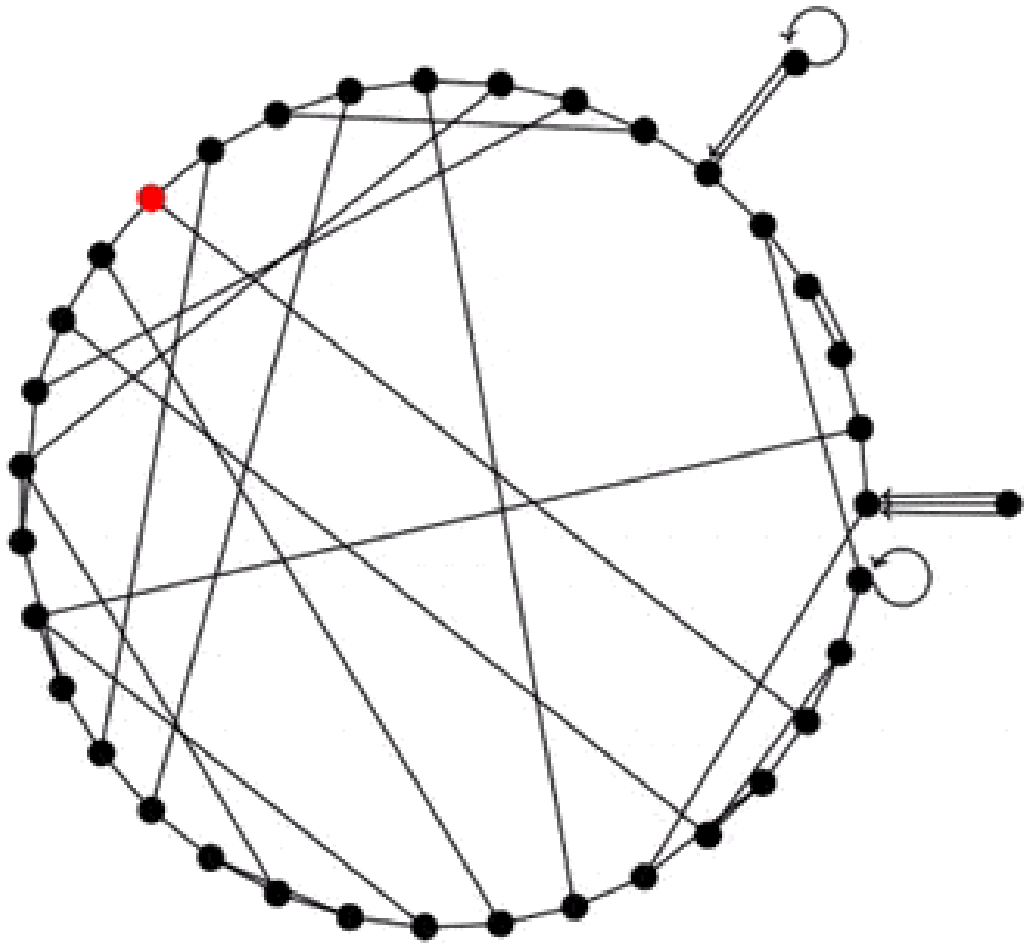

Figure: Supersingular isogeny graph [5]

Random walk on supersingular graph

- Each vertex of the graph represents a different j-invariant of a set of supersingular curves.
- The edges between vertices represent isogenies converting one elliptic curve to another.
- The graph is strongly connected, meaning every vertex can be reached from every other vertex.
- *Random Walk:* It is possible to walk a whole graph by starting from any vertex, randomly choosing an edge, following it to the next vertex and then start the process again on a new vertex.
- The security level of a system depends on value n - the number of steps taken during the walk.

SIDH

Supersingular **I**sogeny **D**iffie **H**ellman Key Exchange

- Created in 2011 by De Feo, Jao, and Plut
- Based on walks in a supersingular isogeny graph
- Uses 2688-bit public keys at a 128-bit quantum security level
- Distinguishes itself from similar systems such as NTRU and Ring-LWE by supporting perfect forward secrecy

SIDH

	ECDH	**SIDH**
Elements	point on a curve group	j-invariant isogeny class
Computation	$k, P \to [k]P$	$\phi, E \to \phi(E)$
Secret	scalars k	isogenies ϕ
Hard problem	given $P, [k]P$ find k	given $E, \phi(E)$ find ϕ

Figure: Comparison between ECDH and SIDH [5]

Set-up for SIDH

- different small primes l_A, l_B
- large exponents e_A, e_B
- small cofactor f
- a prime $p = l_A^{e_A}.l_B^{e_B}.f \pm 1$
- supersingular elliptic curve E defined over $\mathbb{F}_{p^2}$
- E has two large torsion subgroups $E[l_A^{e_A}]$ and $E[l_B^{e_B}]$ which are assigned to Alice and Bob
- fixed elliptic points P_A, Q_A, P_B, Q_B on E from torsion subgroups
- two random integers $m_A, n_A < (l_A)^{e_A}$
- two random integers $m_B, n_B < (l_B)^{e_B}$

SIDH Key Exchange

Alice		**Bob**
$\mathtt{sk}_A : m_A, n_A \in_R /\ell_A^{e_A}$		$\mathtt{sk}_B : m_B, n_B \in_R /\ell_B^{e_B}$
$\phi_A := E/\langle [m_A]P_A + [n_A]Q_A \rangle$		$\phi_B := E/\langle [m_B]P_B + [n_B]Q_B \rangle$
$\mathtt{pk}_A : \phi_A(P_B), \phi_A(Q_B), E_A$		$\mathtt{pk}_B : \phi_B(P_A), \phi_B(Q_A), E_B$
	$\xrightarrow{\mathtt{pk}_A}$	
	$\xleftarrow{\mathtt{pk}_B}$	
$E_{AB} :=$		$E_{BA} :=$
$E_B/\langle [m_A]\phi_B(P_A) + [n_A]\phi_B(Q_A) \rangle$		$E_A/\langle [m_B]\phi_A(P_B) + [n_B]\phi_A(Q_B) \rangle$
Shared secret: $j(E_{AB})$		**Shared secret:** $j(E_{BA})$

Figure: SIDH key exchange protocol [1]

Security of SIDH

- Related to the problem of finding the isogeny mapping between two supersingular elliptic curves with the same number of points
- De Feo, Jao and Plut suggest that the security of SIDH will be $O(p^{1/4})$ for classical computers and $O(p^{1/6})$ for quantum computers
- A 2014 study of the isogeny mapping problem by Delfs and Galbraith confirmed the $O(p^{1/4})$ security analysis for classical computers

Efficiency of SIDH

- Each elliptic curve coefficient requires $log_2 p^2$ bits.
- Each elliptic curve point can be transmitted in $log_2 p^2+1$ bits, hence the transmission is $8log_2 p^2+4$ bits.
- This is 6144 bits for a 768-bit modulus p (128-bit security).
- This can be reduced by over half to 2640 bits (330 bytes) using key-compression techniques.
- With these compression techniques, SIDH has a similar bandwidth requirement to traditional 3072-bit RSA signatures or Diffie-Hellman key exchanges.

Efficiency of SIDH

- In 2014,for a 768-bit modulus researchers were able to complete the key exchange computations in 200 milliseconds.
- In 2016, researchers from Microsoft posted software for the SIDH which runs in constant time.
- In 2016, researchers developed efficient ARM implementations of SIDH.
- In 2017, researchers developed the first FPGA implementations of SIDH.

SIKE

Supersingular **I**sogeny **K**ey **E**ncapsulation

- IND-CCA2 KEM
- Based on SIDH
- SIKEp434, SIKEp503, SIKEp610, SIKEp751

The SIKE protocol specifies:

- Parameter sets
- Key/ciphertext formats
- Encapsulation/decapsulation mechanisms
- Choice of symmetric primitives (hash functions, etc.)

Authors of SIKE

- David Jao, University of Waterloo and evolutionQ, Inc. (principal submitter)
- Reza Azarderakhsh, Florida Atlantic University and PQSecure Technologies, LLC
- Matthew Campagna, Amazon
- Craig Costello, Microsoft Research
- Luca De Feo, Université de Versailles – Saint-Quentin
- Basil Hess, Infosec Global, Switzerland
- Amir Jalali, LinkedIn Corporation
- Brian Koziel, Texas Instruments
- Brian LaMacchia, Microsoft Research
- Patrick Longa, Microsoft Research
- Michael Naehrig, Microsoft Research
- Geovandro Pereira, University of Waterloo and evolutionQ, Inc
- Joost Renes, Radboud University
- Vladimir Soukharev, Infosec Global, Canada
- David Urbanik, University of Toronto

Set-up for SIKE

Public Parameters:

- Defined over a prime of the form $p = l_A^{e_A}.l_B^{e_B}.f \pm 1$
- For efficiency reasons, $l_A = 2$, $l_B = 3$, and $f = 1$ are fixed; $p = 2^{e_A}.3^{e_B}$-1
- Starting supersingular elliptic curve $E_0/\mathbb{F}_{p^2}: y^2 = x^3 + 6x^2 + x$
  - with cardinality $(2^{e_A}.3^{e_B})^2$
  - along with base points $< P_A, Q_A >= E_0[2^{e_A}]$ and $< P_B, Q_B >= E_0[2^{e_B}]$

- H_1, H_2: cSHAKE256 hash function

SIKE mechanism

Alice		**Bob**
Key generation:		
$\mathbf{pk}_A = [E_A, \phi_A(P_B), \phi_A(Q_B)]$		
$s \in_R \{0,1\}^t$		
		Encapsulation:
		$m \in_R \{0,1\}^t$
		$r = H_1(m \parallel \mathbf{pk}_A)$
		$\mathbf{pk}_B(r) = [E_B, \phi_B(P_A), \phi_B(Q_A)]$
		$j = j(E_{BA})$
		$c = (c_0, c_1) = (\mathbf{pk}_B(r), H_2(j) \oplus m)$
		$K = H_3(m \parallel c)$
	$\xleftarrow{(c_0,c_1)}$	
Decapsulation:		
$j = j(E_{AB})$		
$m' = c_1 \oplus H_2(j)$		
$r' = H_1(m' \parallel \mathbf{pk}_A)$		
If $(\mathbf{pk}_B(r') = c_0) \rightarrow K = H_3(m' \parallel c)$		
If $(\mathbf{pk}_B(r') \neq c_0) \rightarrow K = H_3(s \parallel c)$		

Figure: SIKE Mechanism [1]

Advantages of SIKE

- Very small key sizes
- No possibility for decryption error
- No complicated error distributions, rejection sampling, etc
- Simple, conservative security analysis when assuming only generic attacks

Disadvantages of SIKE

- Relatively slow
- Future analysis may uncover non-generic attacks against SIKE (though none are known so far)

Comparison to Other NIST Round 2 Candidates

PQC Submission	NIST Security Level	Public Key Size (Bytes)	Area			Total Time (ms)
			# FFs	# LUTs	# Slices	
BIKE[1] [42]	1	2,541	-	-	1,559	10.2
McEliece[2] [43]	5	1,044,992	111,299	66,615	-	1.4[3]
SIKEp751 [14]	5	**564**	51,914	44,822	16,756	33.4
SIKEp503 (this work)	2	**378**	26,971	25,893	9,514	13.6
SIKEp751 (this work)	5	**564**	50,390	45,893	17,530	26.9

Figure: Hardware comparison of Round 1 PQC submissions that have moved on to Round 2. BIKE measures total time with key generation and encapsulation. All others measure total time with key encapsulation and decapsulation. BIKE is on an Artix-7 FPGA and all other implementations are on a Virtex-7 FPGA [4]

References

[1] H. Seo, A. Jalali, R. Azarderakhsh, "Optimized SIKE Round 2 on 64-bit ARM" Cryptology ePrint Archive, Report 2019/721, 2019. https://eprint.iacr.org/2019/721.pdf

[2] L.D. Feo, "Mathematics of Isogeny Based Cryptography" 2017. https://arxiv.org/pdf/1711.04062.pdf

[3] D. Jao, "Supersingular Isogeny Key Encapsulation" 2019. https://sike.org/files/SIDH-spec.pdf

[4] B. Koziel, A. Ackie, R.E. Khatib, R. Azarderakhsh, M.M. Kermani, "SIKE'd Up: Fast and Secure Hardware Architectures for Supersingular Isogeny Key Encapsulation" Cryptology ePrint Archive, Report 2019/711, 2019. https://eprint.iacr.org/2019/711.pdf

[5] K. Kwiatkowski, "Towards Post-Quantum Cryptography in TLS" 2019. https://blog.cloudflare.com/towards-post-quantum-cryptography-in-tls/

[6] D. Jao, "Supersingular Isogeny Key Encapsulation" 2018. https://csrc.nist.gov/CSRC/media/Presentations/SIKE/images-media/SIKE-April2018.pdf