A post-quantum digital signature scheme based on supersingular isogenies

Youngho Yoo

Department of Combinatorics & Optimization

UNIVERSITY OF WATERLOO

April 3, 2017

CryptoWorks21 evolution INFOSEC GLOBAL

Post-quantum cryptography

Post-quantum cryptography is the study of cryptosystems that are secure against attacks by quantum computers.

Standardization efforts currently under way:

- NIST
- NSA Suite B
- ETSI

Major families of post-quantum cryptosystems:

- Lattice-based cryptography
- Code-based cryptography
- Multivariate polynomials
- Hash-based cryptography
- Isogeny-based cryptography

Isogeny-based cryptography

Jao and De Feo (2011):

- Supersingular Isogeny Diffie-Hellman key exchange (SIDH)
- Public-key encryption

Jao, De Feo, and Plût (2014):

- Zero-knowledge proof of identity

Costello, Longa, and Naehrig (2016):

- Efficient library for SIDH

Azarderakhsh et al. (2016) and Costello et al. (2016):

- Key-compression (pk: 330 bytes, sk: 48 bytes)

Isogeny-based cryptography

Authentication schemes:

- Zero-knowledge proof of identity (Jao, De Feo, and Plût, '14)
- Strong designated verifier signatures (Sun, Tian, and Wang, '12)
- Undeniable signature (Jao and Soukharev, '14)
- Undeniable blind signature (Seshadri and Chandrasekaran, '16)

Our work:

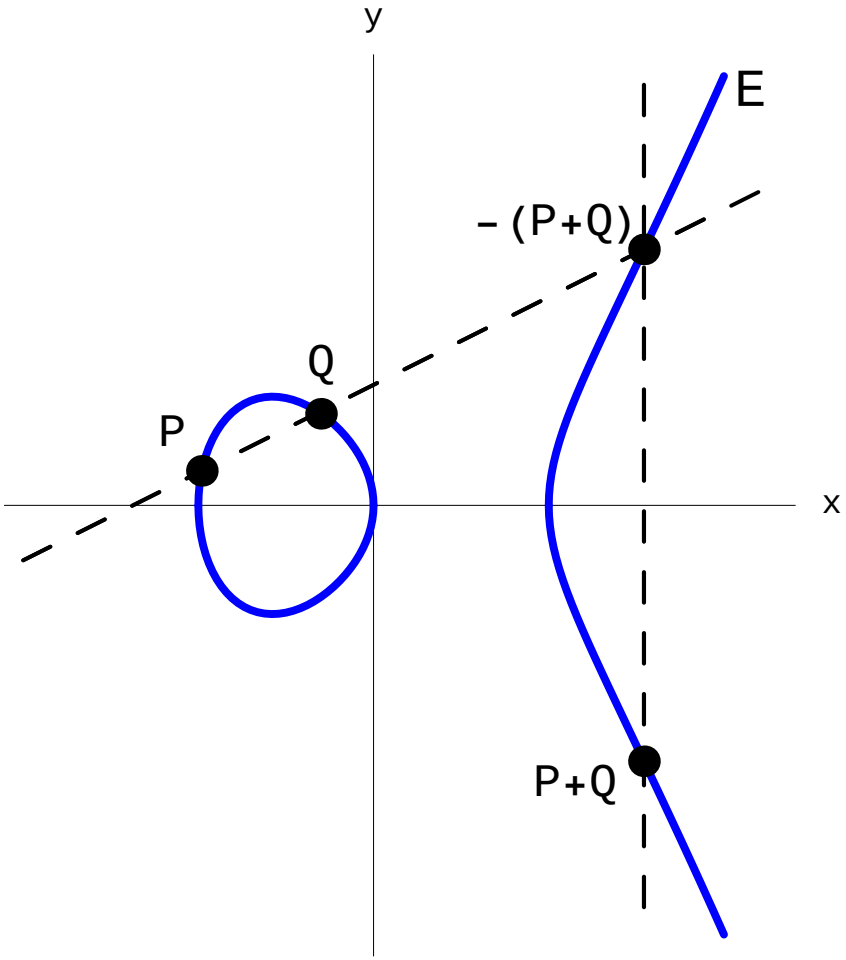
- General-purpose digital signature

Elliptic curves

Definition

An elliptic curve over a field F is a nonsingular plane curve E of the form $y^2 = x^3 + ax + b$, for fixed $a, b \in F$.

The set of projective points on an elliptic curve forms a group.



Isogenies

- An **isogeny** is a rational map between two elliptic curves that preserves the point at infinity. Isogenies are also group homomorphisms.
- A **supersingular elliptic curve** is a special type of curve with extra endomorphisms. Otherwise, they are **ordinary** curves.
- A **supersingular isogeny** is an isogeny between supersingular elliptic curves.

Why supersingular isogenies?

- Ordinary isogenies can be computed in subexponential time (Childs, Jao, Soukharev 2014).
- Supersingular isogenies: best known quantum algorithm takes $O(p^{1/6})$ time.

Basic key exchange

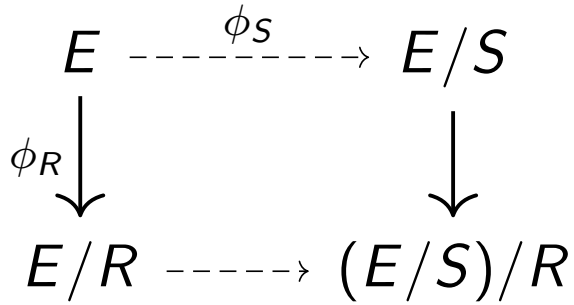
1. Public parameters: An elliptic curve E defined over a finite field F of characteristic p.
2. Alice chooses a kernel A and sends E/A to Bob.
3. Bob chooses a kernel B and sends E/B to Alice.
4. The shared secret is $(E/A)/B = (E/B)/A$.

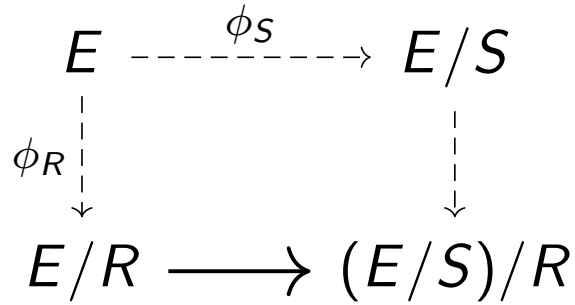
$$\begin{array}{ccc} E & \xrightarrow{\phi_A} & E/A \\ {\scriptstyle \phi_B}\downarrow & & \downarrow \\ E/B & \longrightarrow & (E/A)/B \end{array}$$

To achieve ℓ-bit (quantum) security level against current attacks, use 6ℓ-bit primes.

Zero-knowledge proof of identity

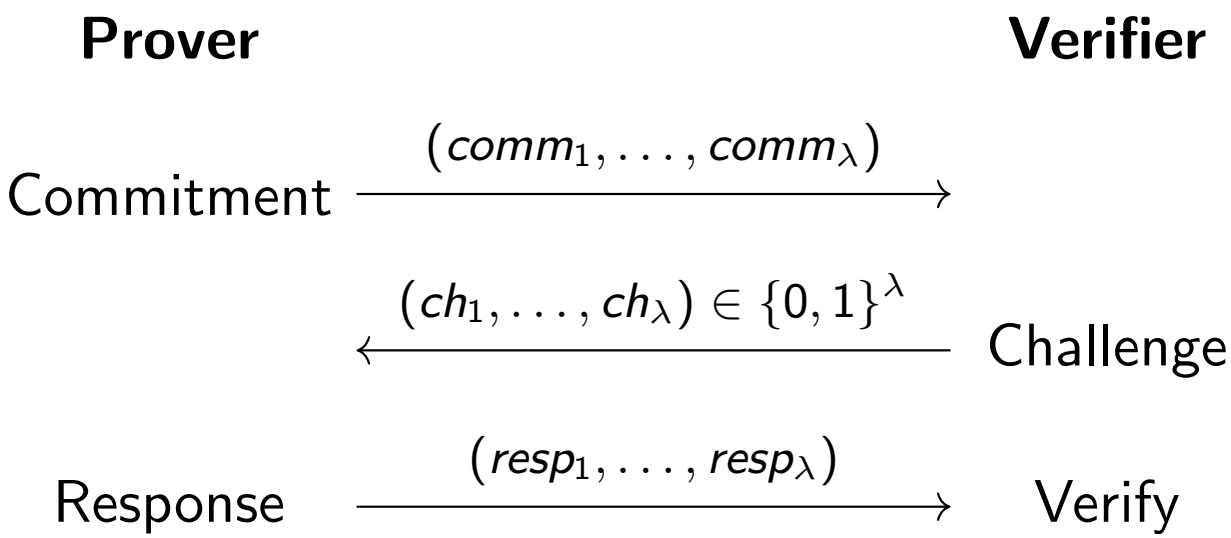
1. Peggy chooses a kernel R and sends E/R and $(E/R)/S$.
2. Victor sends a challenge bit $ch \in \{0, 1\}$.
3. Peggy responds with:
   - $(B, \phi_A(B))$ if $ch = 0$
   - $(\phi_B(A))$ if $ch = 1$
4. Victor verifies that the response generate kernels for the correct isogenies.
5. Repeat λ times.



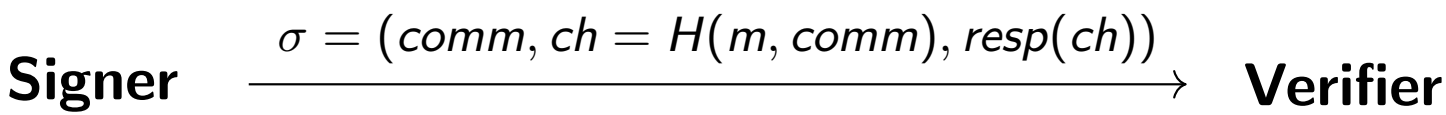


Signatures from zero-knowledge proofs

Interactive zero-knowledge proof:



Fiat-Shamir:



- not quantum-safe

Unruh's construction

Idea: Include the hashes of all possible responses, and use them to obtain the random challenge.

1. For each $comm_i$, and for each possible challenge $ch_{i,j}$, compute the hash $h_{i,j} = G(resp_{i,j})$.
2. Set $J = (J_i)_i = H(m, (comm_i)_i, (ch_{i,j})_{i,j}, (h_{i,j})_{i,j})$
3. Send $(ch_{i,j})_{i,j}$ and $(h_{i,j})_{i,j}$ (so the verifier can compute J)
4. Send the responses $resp_i = resp_{i,J_i}$

$$\textbf{Signer} \xrightarrow{\sigma = ((comm_i)_i, (ch_{i,j})_{i,j}, (h_{i,j})_{i,j}, (resp_{i,J_i})_i)} \textbf{Verifier}$$

Signature scheme

- **Public parameters**: A supersingular curve E.
- **Key generation**: Choose a kernel S and publish E/S.
- **Sign**(m):

  Repeat (in parallel) for $i = 1, \ldots, 2\lambda$:
  - Choose a kernel R_i and set $comm_i = (E/R_i, (E/R_i)/S)$.
  - For each $ch_{i,j} \in \{0, 1\}$
    - Let $resp_{i,j}$ be the response to $ch_{i,j}$.
    - Set $h_{i,j} = G(resp_{i,j})$

  Set $J = H(m, (comm_i)_i, (ch_{i,j})_{i,j}, (h_{i,j})_{i,j})$.

  Send $\sigma = ((comm_i)_i, (ch_{i,j})_{i,j}, (h_{i,j})_{i,j}, (resp_{i,J_i})_i)$.
- **Verify**(σ):

  Compute $J = H(m, (comm_i)_i, (ch_{i,j})_{i,j}, (h_{i,j})_{i,j})$.

  Verify the responses for each ch_{i,J_i}.

Parameter Sizes

Sizes in bytes:

Scheme	Public-key	Private-key	Signature
Isogeny-based	768	48	141,312
Compressed	336	48	122,880
Hash-based[1]	1,056	1,088	41,000
Code-based[2]	192,192	1,400,288	370
Lattice-based[3]	7,168	2,048	5,120
Ring-LWE[4]	7,168	4,608	3,488
Multivariate[5]	99,100	74,000	424

[1] SPHINCS-256 (Bernstein et al. '15)
[2] Niederreiter variant of McEliece (Bernstein et al. '08)
[3] BLISS (Ducas et al. '13)
[4] TESLA# (Barreto et al. '16)
[5] Rainbow (Ding and Schmidt '05)

Performance

Performance (in 10^6 clock cycles) on Intel Xeon E5-2637 v3 3.5 GHz and ARM Cortex-A57:

Platform	Threads	Keygen	Sign	Verify
PC	1	63	28,776	19,679
	2	-	14,474	10,042
	4	-	7,449	5,536
ARM (C)	-	1,656	767,928	493,797
ARM (ASM)	-	123	57,092	36,757