Similarities and Differences between Diffie-Hellman and Isogeny Crypto, 26 August 2020

Steven Galbraith

University of Auckland, New Zealand

Plan

- Diffie-Hellman key exchange.
- Authenticated key exchange (AKE), Signatures, Oblivious Transfer (OT), Pedersen commitments.
- Supersingular isogeny key exchange (SIDH).
- Authenticated key exchange from SIDH.
  - S. D. Galbraith, Authenticated key exchange for SIDH, eprint 2018/266.
- CSIDH
- Signatures
  - S. D. Galbraith, C. Petit and J. Silva, Identification Protocols and Signature Schemes Based on Supersingular Isogeny Problems', ASIACRYPT 2017.
  - L. de Feo and S. D. Galbraith, SeaSign: Compact isogeny signatures from class group actions, EUROCRYPT 2019.
- Oblivious Transfer from CSIDH
  - Y.-F. Lai, S. D. Galbraith and C. Delpech de Saint Guilhem, Compact, Efficient and UC-Secure Isogeny-Based Oblivious Transfer, eprint 2020/1012.

Please interrupt and ask questions any time.

Discrete Logarithm Problem (DLP)

Let G be a group of order ℓ.
The DLP is: Given $g \in G$ and $h = g^a$, to compute a.

Diffie-Hellman key exchange:

- Alice chooses a and sends $t_A = g^a$ to Bob.
- Bob chooses b and sends $t_B = g^b$ to Alice.
- Alice computes $t_B^a = g^{ab}$.
- Bob computes $t_A^b = g^{ab}$.
- Usually hash g^{ab} to get a short-ish binary string as output (key derivation function).

Person in middle attack on Diffie-Hellman

Alice		Eve		Bob
$t_A = g^a$	$\xrightarrow{t_A}$			
			$\xleftarrow{t_B}$	$t_B = g^b$
$t_B^a = g^{ab}$				$t_A^b = g^{ab}$

Person in middle attack on Diffie-Hellman

Alice		Eve		Bob
$t_A = g^a$	$\xrightarrow{t_A}$			
		$t'_A = g^e$	$\xrightarrow{t'_A}$	

Person in middle attack on Diffie-Hellman

$$
\begin{array}{ccccc}
\text{Alice} & & \text{Eve} & & \text{Bob} \\
t_A = g^a & \xrightarrow{t_A} & & & \\
 & & t'_A = g^e & \xrightarrow{t'_A} & \\
 & & & \xleftarrow{t_B} & t_B = g^b \\
 & \xleftarrow{t'_B} & t'_B = g^f & &
\end{array}
$$

Person in middle attack on Diffie-Hellman

$$\begin{array}{ccccc}
\text{Alice} & & \text{Eve} & & \text{Bob} \\
t_A = g^a & \xrightarrow{t_A} & & & \\
 & & t'_A = g^e & \xrightarrow{t'_A} & \\
 & & & \xleftarrow{t_B} & t_B = g^b \\
 & \xleftarrow{t'_B} & t'_B = g^f & & \\
(t'_B)^a = g^{af} & & t_A^f = g^{af} & &
\end{array}$$

Person in middle attack on Diffie-Hellman

$$
\begin{array}{ccccc}
\text{Alice} & & \text{Eve} & & \text{Bob} \\
t_A = g^a & \xrightarrow{t_A} & & & \\
 & & t'_A = g^e & \xrightarrow{t'_A} & \\
 & & & \xleftarrow{t_B} & t_B = g^b \\
 & \xleftarrow{t'_B} & t'_B = g^f & & \\
 & & & & \\
(t'_B)^a = g^{af} & & t_A^f = g^{af} \; ; \; t_B^e = g^{be} & & (t'_A)^b = g^{eb}
\end{array}
$$

Authenticated key agreement (AKE)

- Want Alice and Bob to interactively generate a random key.
- Want to prevent "person-in-the-middle" attacks, by providing authentication.
- One solution is to MAC the protocol messages, but this assumes a shared MAC key between Alice and Bob.
- Another solution is for Alice and Bob to have public keys (authenticated by a PKI) for a digital signature scheme and to sign all protocol messages.
- Generic Constructions of AKE from IND-CCA KEMs
  - Colin Boyd, Yvonne Cliff, Juan Manuel González Nieto, Kenneth G. Paterson. One-round key exchange in the standard model. IJACT 1(3): 181-199 (2009)
  - Atsushi Fujioka, Koutarou Suzuki, Keita Xagawa, Kazuki Yoneyama. Strongly secure authenticated key exchange from factoring, codes, and lattices. Des. Codes Cryptography 76(3): 469-504 (2015)
  - Cyprien de Saint Guilhem, Nigel P. Smart, Bogdan Warinschi. Generic Forward-Secure Key Agreement Without Signatures. ISC 2017: 114-133

MQV and related schemes

- Alice has public key g^a and Bob has public key g^b.
- Alice sends g^x. Bob sends g^y.
- The shared key is $H(g^{(x+aU)(y+bV)})$ for some values U and V that are derived from the protocol messages and are “short” to provide efficiency.
- This exploits the algebra

$$\left((g^x)(g^a)^U\right)^{y+bV} = g^{(x+aU)(y+bV)}.$$

Public Key Signatures from DLP

- (Security relies on DLP, not Diffie-Hellman.)
- Apply Fiat-Shamir to an interactive identification protocol.
- Alice has public key g^a, where g has order ℓ.
- Alice sends g^k for some random k.
- Bob responds with challenge $c \in \mathbb{Z}_\ell$.
- Alice sends $z = k + ac \pmod{\ell}$.
- Bob checks whether

$$g^z = g^k (g^a)^c.$$

- Many variants: Schnorr, DSA, ECDSA, EdDSA, etc.

Oblivious Transfer (OT)

- Sender (Alice) has two messages. Receiver Bob has a bit i.
- Bob wants to learn only the message m_i and Alice should learn nothing.
- Chou and Orlandi protocol: Alice sends $A = g^s$ to Bob.
- If $i = 0$ Bob sets $B = g^r$ and if $i = 1$ set $B = Ag^r$. Send B to Alice.
- Alice computes $k_0 = H(B^s)$ and $k_1 = H((B/A)^s)$.
- Note: If $i = 0$ then $k_0 = H(g^{rs})$. If $i = 1$ then $k_1 = H(g^{rs})$.
- Alice computes $c_j = Enc_{k_j}(m_j)$ and sends (c_0, c_1) to Bob.
- Bob computes $k_i = H(A^r)$ and computes $m_i = Dec_{k_i}(c_i)$.

Pedersen commitment

- Let g, h be elements of group G of order ℓ such that no-one knows DLP of h with respect to g.
- To commit to an integer $0 \leq x < \ell$ choose random $0 \leq r < \ell$ and compute the Pedersen commitment

$$C(x, r) = g^x h^r.$$

- To open the commitment publish (x, r).

Important point about DLP crypto

We gave four different and important schemes:

- MQV scheme for authenticated key exchange
- DLP signatures
- Chou and Orlandi OT protocol
- Pedersen commitments

All them exploit the fact that we can exponentiate and multiply group elements, and that we have the exponent rules

$$(g^a)^b = g^{ab} \quad \text{and} \quad g^a g^b = g^{a+b}.$$

Spoiler for the rest of the talk

Which schemes can be done with isogenies?

- We have practical authenticated key exchange from isogenies
- Practical public key signatures are a huge open problem
- We have efficient isogeny-based OT protocols
- An isogeny-based analogue of Pedersen commitments is a huge open problem

Generalised Discrete Logarithm Problem

Let G be a group of order ℓ.
Given $g \in G$ and $h = \phi(g)$, to compute the group homomorphism ϕ.

Generalised Diffie-Hellman key exchange (with commuting homomorphisms):

- Alice chooses ϕ_A and sends $t_A = \phi_A(g)$ to Bob.
- Bob chooses ϕ_B and sends $t_B = \phi_B(g)$ to Alice.
- Alice computes $\phi_A(t_B)$.
- Bob computes $\phi_B(t_A)$.

Generalised Discrete Logarithm Problem

Let G be a group of order ℓ.
Given $g \in G$ and $a \in \mathbb{Z}_\ell^*$ write $a * g$ for g^a.

Generalised Diffie-Hellman key exchange (with commuting homomorphisms):

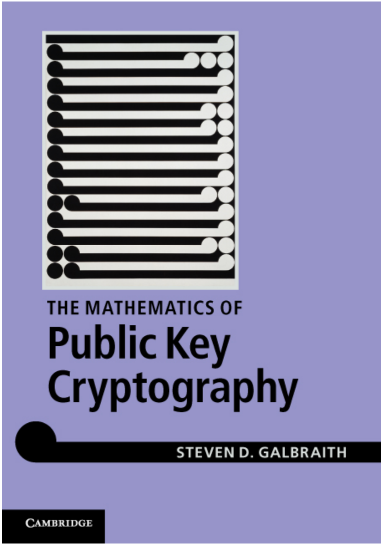
- Alice chooses a and sends $t_A = a * g$ to Bob.
- Bob chooses b and sends $t_B = b * g$ to Alice.
- Alice computes $a * t_B$.
- Bob computes $b * t_A$.

Isogenies

- An **elliptic curve** over a field $\mathbb{k}$ is a non-singular projective cubic curve. The set of $\mathbb{k}$-rational points is a group.
- An **isogeny** $\phi : E_0 \to E_1$ of elliptic curves is a morphism and a group homomorphism.
- Examples of isogenies include: $[n] : E \to E$ and Frobenius map when $\mathbb{k}$ is a finite field.
- An isogeny has finite kernel $G \subseteq E_0(\overline{\mathbb{k}})$.
- If the isogeny is separable then $\#G = \deg(\phi)$.
- Given a finite subgroup $G \subseteq E_0(\overline{\mathbb{k}})$ there exists an elliptic curve E_1 and a (separable) isogeny $\phi : E_0 \to E_1$ with $\ker(\phi) = G$.
- The pair (E_1, ϕ) can be computed using Vélu's formulae.

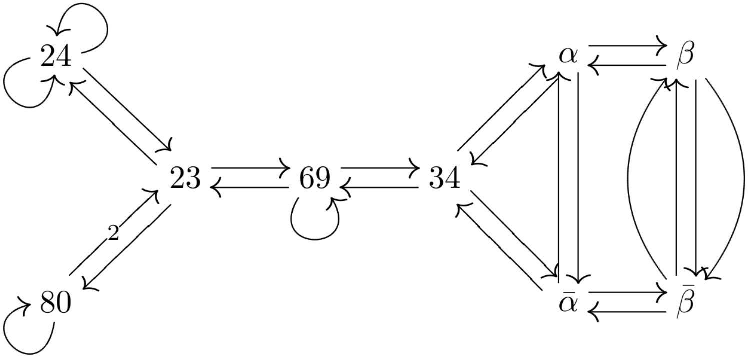
If you want to learn more about isogenies . . .

If you want to learn more about isogenies . . .



Isogeny Graphs

Figure 9. Supersingular Isogeny Graph $X(\bar{\mathbb{F}}_{103}, 2)$



Main Computational Problem Regarding Isogenies

Given E_1, E_2 elliptic curves over $\mathbb{F}_q$ with $\#E_1(\mathbb{F}_q) = \#E_2(\mathbb{F}_q)$, find an isogeny $\phi : E_1 \to E_2$.

Issues:

- Unique?
- Representation?
- Algorithms?
- Classical or Quantum algorithms?

For a survey of algorithmic questions see: Galbraith and Vercauteren, eprint 2017/774.

Jao and De Feo key exchange (SIDH)

$$\begin{tikzcd}
 & E/G_A \arrow[dr] & \\
E \arrow[ur, "\phi_A"] \arrow[dr, "\phi_B"'] & & E/\langle G_A, G_B \rangle \\
 & E/G_B \arrow[ur] &
\end{tikzcd}$$

Jao and De Feo key exchange (SIDH)

- Let $p = \ell_1^{e_1} \ell_2^{e_2} f \pm 1$ be prime.
- Let E over $\mathbb{F}_{p^2}$ be a supersingular elliptic curve.
- Then group structure of $E(\mathbb{F}_{p^2})$ is a product of two cyclic groups of order $\ell_1^{e_1} \ell_2^{e_2} f$.
- Fix points $R_1, S_1 \in E[\ell_1^{e_1}]$ such that $\langle R_1, S_1 \rangle = E[\ell_1^{e_1}]$.
- Fix R_2, S_2 such that $\langle R_2, S_2 \rangle = E[\ell_2^{e_2}]$.
- The **system parameters** are (E, R_1, S_1, R_2, S_2).

Jao and De Feo key exchange (SIDH)

- Alice chooses a secret subgroup G_A of $E[\ell_1^{e_1}]$ by choosing an integer $0 \leq a < \ell_1^{e_1}$ and setting $T_A = R_1 + [a]S_1$, and $G_A = \langle T_A \rangle$.
- Alice computes an isogeny $\phi_A : E \to E_A$ with kernel G_A generated by T_A and sends $(E_A, \phi_A(R_2), \phi_A(S_2))$ to Bob.
- Bob chooses $0 \leq b < \ell_2^{e_2}$, computes $\phi_B : E \to E_B$ with kernel G_B generated by $T_B = R_2 + [b]S_2$ and sends $(E_B, \phi_B(R_1), \phi_B(S_1))$ to Alice.
- Alice computes

$$T_A' = \phi_B(R_1) + [a]\phi_B(S_1) = \phi_B(R_1 + [a]S_1) = \phi_B(T_A)$$

and then computes an isogeny $\phi_A' : E_B \to E_{AB}$ with kernel generated by T_A'.

Jao and De Feo key exchange (SIDH)

- Bob computes an isogeny $\phi'_B : E_A \to E'_{AB}$ with kernel $\langle \phi_A(R_2) + [b]\phi_A(S_2) \rangle$.
- The composition $\phi'_A \circ \phi_B : E \to E_{AB}$ has kernel $\langle T_1, T_2 \rangle$.
- The actual elliptic curve equations E_{AB} and E'_{AB} computed by Alice and Bob are not likely to be the same, but the curves are isomorphic.
- Hence, the shared key for Alice and Bob is $j(E_{AB}) = j(E'_{AB})$.

SIKE submission to NIST

- Submission to the NIST standardization process on post-quantum cryptography.
- Authors: Jao, Azarderakhsh, Campagna, Costello, De Feo, Hess, Jalali, Koziel, LaMacchia, Longa, Naehrig, Renes, Soukharev and Urbanik.
- Submission contains specification of an IND-CCA KEM.
- `http://sike.org/`

Differences between Diffie-Hellman and Isogenies

- With groups we have $(g^a)^b = g^{ab}$ and $g^a g^b = g^{a+b}$.
- With SIDH we just have $(E/G_A)/G_B = E/\langle G_A, G_B \rangle$.
- So lots of protocols from DLP-land do not have an analogue in isogeny-land.

Authenticated Key Exchange (AKE) from Isogenies

- Natural problem to develop authenticated key exchange based on the SIDH protocol.
- Want to prevent "person-in-the-middle" attacks, by providing authentication.
- There is a general solution using signatures.
  Patrick Longa (eprint 2018/267) has presented an SIDH scheme based on this idea, and argues it can be appropriate to use a non-post-quantum signature (ECDSA).
- For a full SIDH solution we want to avoid using public key signatures.
- We consider classical attackers in the random oracle model, not quantum attackers.

Generic Constructions of AKE from IND-CCA KEMs

- Recall there are generic constructions of AKE from IND-CCA KEMs.
- Can get IND-CCA KEM from isogenies and the Fujisaki-Okamoto transform (see SIKE submission).
- Note that there can be adaptive attacks on SIDH keys as predicted by Kirkwood et al and analysed by Galbraith, Petit, Shani and Ti.

  Hence a scheme that has shared key $H(g^{ay}, g^{bx})$ would be secure in the DLP setting (assuming group membership tests are done) but insecure using SIDH.
- Many other schemes: KEA, MQV, HMQV, NAXOS, etc. Most of them rely on some kind of "public key validation". Some of them need "gap assumptions" in the security proof, which we do not believe are safe in the isogeny setting.
- Can get secure schemes based on Boyd et al, Fujioka et al, etc.

JKL Scheme

- Adapt an authenticated key exchange protocol by Jeong, Katz and Lee (2004).
- Jeong-Katz-Lee is in the DLP setting: Alice has public key g^a and Bob has public key g^b.
- Alice sends g^x. Bob sends g^y.
- The shared key is $H(A, B, g^x, g^y, g^{xy}, g^{ab})$.
- Security claim: In random oracle model, the JKL scheme is a secure authenticated key exchange protocol in the CK model.
- The new scheme replaces Diffie-Hellman by supersingular isogeny Diffie-Hellman (SIDH).

JKL Scheme (1)

- The shared key in the JKL scheme is $H(A, B, g^x, g^y, g^{xy}, g^{ab})$.
- Natural isogeny variant is for Alice to send E_X and auxiliary points, and Bob to send E_Y and auxiliary points, just as in SIDH.
- Shared key $H(A, B, j(E_X), j(E_Y), j(E_{XY}), j(E_{AB}))$.

JKL Scheme (2)

- System parameters (E, P_1, Q_1, P_2, Q_2).
- Alice has private keys $0 \leq a_1 < \ell_1^{e_1}$ and $0 \leq a_2 < \ell_2^{e_2}$, and public key $(E_{A,1}, P_{A,1}, Q_{A,1}, E_{A,2}, P_{A,2}, Q_{A,2})$ where $E_{A,1} = E/\langle P_1 + [a_1]Q_1 \rangle$, $E_{A,2} = E/\langle P_2 + [a_2]Q_2 \rangle$, $P_{A,1} = \phi_{A,1}(P_2)$, $Q_{A,1} = \phi_{A,1}(Q_2)$, etc.
- **Alice:** (Initiator)
  - Choose $0 \leq x < \ell_1^{e_1}$ and set $\phi_X : E \to E_X = E/\langle P_1 + [x]Q_1 \rangle$.
  - Set sid $= (E_X, P_2' = \phi_X(P_2), Q_2' = \phi_X(Q_2))$.
  - Send (Alice, Bob, sid) to Bob; Save $x, j(E_X)$.
- **Bob:** (Receiver)
  - Check E_X supersingular, and P_2', Q_2' indep pts on E_X order $\ell_2^{e_2}$.
  - Choose $0 \leq y < \ell_2^{e_2}$ and set $\phi_Y : E \to E_Y = E/\langle P_2 + [y]Q_2 \rangle$.
  - Send (Alice, Bob, sid, $E_Y, P_1' = \phi_Y(P_1), Q_1' = \phi_Y(Q_1)$) to Alice.

JKL Scheme (3)

- **Bob:** (Completion)
  - Compute $E_{XY} = E_X / \langle P_2' + [y] Q_2' \rangle$.
  - Look up Alice's public key, verify certificate, look up long-term private key b_2, compute $E_{AB} = E_{A,1} / \langle P_{A,1} + [b_2] Q_{A,1} \rangle$.
  - Compute session key

    $$k = H(\text{Alice}, \text{Bob}, j(E_X), P_2', Q_2', j(E_Y), P_1', Q_1', j(E_{XY}), j(E_{AB})).$$

  - Flush working storage.
- **Alice:** (Completion)
  - Check E_Y supersingular, P_1', Q_1' are indep pts on E_Y order $\ell_1^{e_1}$.
  - Retrieve $x, j(E_X)$, compute $E_{XY} = E_Y / \langle P_1' + [x] P_2' \rangle$.
  - Get Bob's public key $(E_{B,2}, P_{B,2}, Q_{B,2})$, verify cert, get long-term priv key a_1, compute $E_{AB} = E_{B,2} / \langle P_{B,2} + [a_1] Q_{B,2} \rangle$.
  - Compute session key

    $$k = H(\text{Alice}, \text{Bob}, j(E_X), P_2', Q_2', j(E_Y), P_1', Q_1', j(E_{XY}), j(E_{AB})).$$

  - Flush working storage.

Security theorem and proof

- Assume classical attacker in random oracle model.
- SIDH-CDH problem: Given $(E, P_1, Q_1, P_2, Q_2, E_A, \phi_A(P_2), \phi_A(Q_2), E_B, \phi_B(P_1), \phi_B(Q_1))$ to compute $j(E_{AB})$.
- **Theorem:** Suppose SIDH-CDH is hard. Then, in the random oracle model, the new protocol is a secure authenticated key exchange protocol in the classical Canetti-Krawczyk model.
- The scheme has weak forward secrecy but not KCI security.
- Public key validation: Can it be avoided? How to make it more efficient?

Current state of the art

Xiu Xu, Haiyang Xue, Kunpeng Wang, Man Ho Au, Song Tian. Strongly Secure Authenticated Key Exchange from Supersingular Isogenies. ASIACRYPT 2019.

Signatures from isogenies

- Suppose Alice has public key E_0 and $E_1 = E/G$. Her private key is G and the isogeny $\phi : E_0 \to E_1$.
- Alice can prove to Bob that she knows ϕ:
- Alice choose a subgroup H (order of H coprime to order of G) and sends to Bob (E_2, E_3) where $E_2 = E_0/H$ and $E_3 = E_1/\phi(H)$.
- Bob sends a bit b.
- If $b = 0$ Alice responds with H and $\phi(H)$. Bob checks E_2 and E_3 correctly formed.
- If $b = 1$ Alice responds with $G' = \psi(G)$ where $\psi : E_0 \to E_2$ is the isogeny with kernel H.
  Bob checks that $E_2/G' = E_3$.
- Repeat.

Signatures from isogenies

$$\begin{tikzcd}
E_0 \arrow[r, "\phi"] \arrow[d, "\psi"'] & E_1 \arrow[d, "\psi'"] \\
E_2 \arrow[r, "\phi'"] & E_3
\end{tikzcd}$$

- The zero-knowledge proof can be turned into a signature scheme using the Fiat-Shamir transform.
- This signature scheme was proposed independently in 2017 by: Yoo, Azarderakhsh, Jalali, Jao and Soukharev; Galbraith, Petit and Silva.

Isogeny Signatures - Second scheme

- For details see: S. D. Galbraith, C. Petit and J. Silva, "Identification Protocols and Signature Schemes Based on Supersingular Isogeny Problems", ASIACRYPT 2017, Springer LNCS 10624 (2017) 3–33.
- Full version: `https://eprint.iacr.org/2016/1154`
- Zero-knowledge proof of an isogeny $\phi : E_1 \to E_2$.
- Similar to graph isomorphism.
- The techniques include a set of algorithms for working with orders in quaternion algebras and finding elements in quaternion modules with "power-smooth" norms.
- $1/2$ cheating probability means protocol must be repeated (in parallel) many times.
- This scheme is not practical.

Generalised Discrete Logarithm Problem

(Brassard and Yung 1990, Couveignes 1997)

Let G be a group.
Given $g \in G$ and $a \in \mathbb{Z}$ write $a * g$ for g^a.

Generalised Diffie-Hellman key exchange (with commuting homomorphisms):

- Alice chooses a and sends $t_A = a * g$ to Bob.
- Bob chooses b and sends $t_B = b * g$ to Alice.
- Alice computes $a * t_B$.
- Bob computes $b * t_A$.

Class Group Action on Elliptic Curves

(Couveignes 1997, Rostovtsev and Stolbunov 2006)

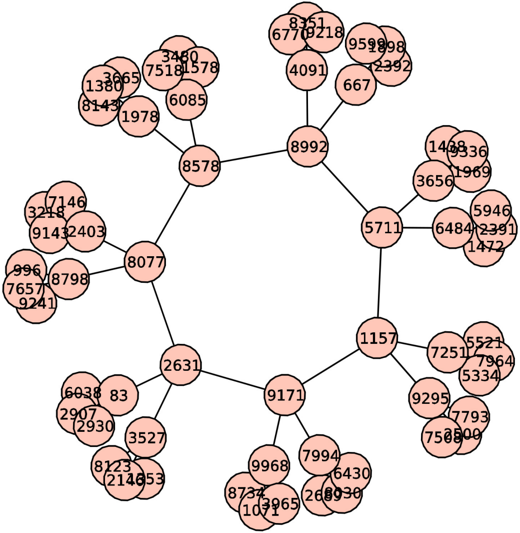
- Let E be an ordinary elliptic curve over $\mathbb{F}_q$ with $\mathrm{End}(E) \cong \mathcal{O}$ an order in an imaginary quadratic field.
- Let $\mathfrak{a}$ be an invertible $\mathcal{O}$-ideal.
- Can define the subgroup

$$E[\mathfrak{a}] = \{P \in E(\overline{\mathbb{F}}_q) : \phi(P) = 0 \ \forall \phi \in \mathfrak{a}\}.$$

(Waterhouse 1969)

- There is an isogeny $E \to E'$ with kernel $E[\mathfrak{a}]$. Define $\mathfrak{a} * E$ to be $E' = E/E[\mathfrak{a}]$.
- $\mathfrak{a} * E$ depends only on the ideal class of $\mathfrak{a}$.
- This gives an action of the ideal class group $\mathrm{Cl}(\mathcal{O})$ on the set of E with $\mathrm{End}(E) \cong \mathcal{O}$.

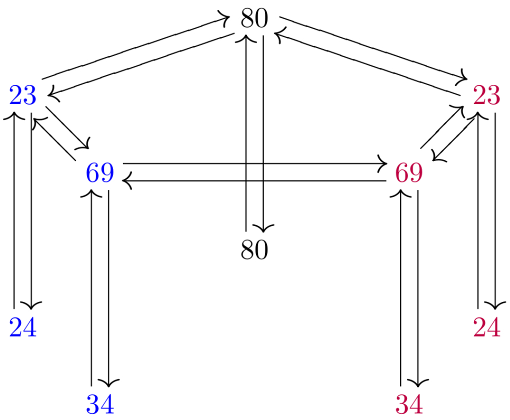
Ordinary Isogeny Graph ($\ell = 3$)



Credit: Dustin Moody

Supersingular Isogeny Graph

FIGURE 10. Rational Supersingular Isogeny Graph $X(\mathbb{F}_{103}, 2)$



See: C. Delfs and S. Galbraith, Computing isogenies between

CSIDH (Castryck, Lange, Martindale, Panny, Renes 2018)

- Let $p = 4\ell_1 \cdots \ell_k - 1$.
- Let X be the set of isomorphism classes of supersingular elliptic curves E with j-invariant in $\mathbb{F}_p$.
- All $E \in X$ have $\mathrm{End}_{\mathbb{F}_p}(E)$ an order in $\mathbb{Q}(\sqrt{-p})$.
  Here $\mathrm{End}_{\mathbb{F}_p}(E) = \{\phi : E \to E \text{ defined over } \mathbb{F}_p\}$.
- CSIDH is an instantiation of group action crypto using supersingular curves, which gives **massive** performance improvements.
- Features:
  - No public key validation needed, so can do non-interactive key exchange.
  - Better bandwidth.
  - Only sub-exponentially quantum secure.

CSIDH

- Choose exponents $|e_i| \leq B$ and define

$$\mathfrak{a} = \prod_i \mathfrak{l}_i^{e_i}$$

  where $\mathfrak{l}_i$ are ideals of small prime norm ℓ_i.
- Efficient to compute $E_A = \mathfrak{a} * E$ using Vélu.
  (Well, not very efficient. Actually quite slow.)
- Straightforward to get a Diffie-Hellman style key exchange.

Public Key Signatures

- L. De Feo and S. Galbraith "SeaSign: Compact isogeny signatures from class group actions", EUROCRYPT 2019.
- Public key: E and $E_A = \mathfrak{a} * E$ where

  $$\mathfrak{a} = \prod_i \mathfrak{l}_i^{e_i}$$

  and $\mathfrak{l}_i$ ideals of small prime norm, $|e_i| \le B$.
- Basic idea:
  - Send $E_B = \mathfrak{b} * E$ to verifier.
  - Verifier sends a challenge bit c.
  - If $c = 0$ the prover replies with $\mathfrak{b}$ and if $c = 1$ the prover replies with $\mathfrak{b}\mathfrak{a}^{-1}$.
  - Verifier checks $E_B = \mathfrak{b} * E$ respectively $E_B = (\mathfrak{b}\mathfrak{a}^{-1}) * E_A$.

Public Key Signatures

- Public: E and $E_A = \mathfrak{a} * E$.
- Signer generates random ideals $\mathfrak{b}_k = \prod_{i=1}^{n} \mathfrak{l}_i^{f_{k,i}}$ for $1 \le k \le t$ and computes $\mathcal{E}_k = \mathfrak{b}_k * E$.
- Compute $H(j(\mathcal{E}_1), \dots, j(\mathcal{E}_t), \text{message})$ where H is a cryptographic hash function with t-bit output $b_1, \dots, b_t$.
- If $b_k = 0$ signature includes $\mathbf{f}_k = (f_{k,1}, \dots, f_{k,n})$ and if $b_k = 1$ it includes

$$\mathbf{f}_k - \mathbf{e} = (f_{k,1} - e_1, \dots, f_{k,n} - e_n).$$

- Use Lyubashevsky's "Fiat-Shamir with aborts".

Improvements

- Thomas Decru, Lorenz Panny, Frederik Vercauteren. Faster SeaSign signatures through improved rejection sampling. PQCrypto 2019.
- Ward Beullens, Thorsten Kleinjung, Frederik Vercauteren. CSI-FiSh: Efficient Isogeny Based Signatures Through Class Group Computations. ASIACRYPT 2019
- Ali El Kaafarani, Shuichi Katsumata, Federico Pintore. Lossy CSI-FiSh: Efficient Signature Scheme with Tight Reduction to Decisional CSIDH-512. PKC 2020.
- But still none of these gives a practical signature.

Isogeny OT

- The Chou-Orlandi scheme does not adapt to isogenies.
- We propose a simple scheme from CSIDH.
- Yi-Fu Lai, Steven D. Galbraith and Cyprien Delpech de Saint Guilhem. Compact, Efficient and UC-Secure Isogeny-Based Oblivious Transfer. eprint 2020/1012.
- Alice sends $A = s * E$ to Bob
- If $i = 0$ Bob computes $C = r * E$ else $C = r * A$ and sends to Alice
- Alice computes $k_0 = H(s * C)$ and $k_1 = H(s^{-1} * C)$.
- Alice computes $c_j = Enc_{k_j}(m_j)$ and sends (c_0, c_1) to Bob.
- if $i = 0$ Bob computes $k_i = H(r * A)$ else $k_i = H(r * E)$, and computes $m_i = Dec_{k_i}(c_i)$.

Isogeny OT

- For a curve E define E^t to be the quadratic twist.
- Let $E : y^2 = x^3 + Ax + B$, then $E^t : dy^2 = x^3 + Ax + B$ where d is a non-square.
- If E is supersingular and defined over $\mathbb{F}_p$ then E^t is supersingular and is also defined over $\mathbb{F}_p$ and is isogenous to E.
- Special gadget with CSIDH: $(\mathfrak{a} * E)^t = \mathfrak{a}^{-1} * E^t$

Isogeny OT

- We give a two rounds OT scheme.
- If $i = 0$ Bob computes $C = r * E$ else $C = (r * E)^t$ and sends C to Alice.
- Alice computes $k_0 = H(s * C)$ and $k_1 = H(s^{-1} * C)$.
- Alice computes $c_j = Enc_{k_j}(m_j)$ and sends $(s * E, c_0, c_1)$ to Bob.
- Let $A = s * E$. If $i = 0$ Bob computes $k_0 = H(r * A)$ and if $i = 1$ Bob computes $k_1 = H(r^{-1} * A^t)$.
- Thus Bob gets m_i.
- Paper contains full security analysis in the UC model. The full scheme has 3 rounds.
- There is no analogous 2-round OT protocol based on DLP, as far as we know.

Summary

- We have practical authenticated key exchange from isogenies
- Practical public key signatures are a huge open problem
- We have efficient isogeny-based OT protocols
- An isogeny-based analogue of Pedersen commitments is a huge open problem

Thank You