# Isogenies as a Cryptographic Primitive

David Jao
(joint work with R. Venkatesan)

University of Waterloo

Workshop on Cryptography: Underlying Mathematics, Provability and Foundations

November 28, 2006

# Outline

1. Elliptic Curves
   - Elliptic Curve Cryptosystems
   - Pairing Based Cryptosystems

# Outline

1. Elliptic Curves
   - Elliptic Curve Cryptosystems
   - Pairing Based Cryptosystems

2. Isogenies
   - Construction
   - Applications

# Outline

1. Elliptic Curves
   - Elliptic Curve Cryptosystems
   - Pairing Based Cryptosystems
2. Isogenies
   - Construction
   - Applications
3. Security issues
   - Reduction proofs
   - Attacks

# Outline

1. Elliptic Curves
   - Elliptic Curve Cryptosystems
   - Pairing Based Cryptosystems
2. Isogenies
   - Construction
   - Applications
3. Security issues
   - Reduction proofs
   - Attacks

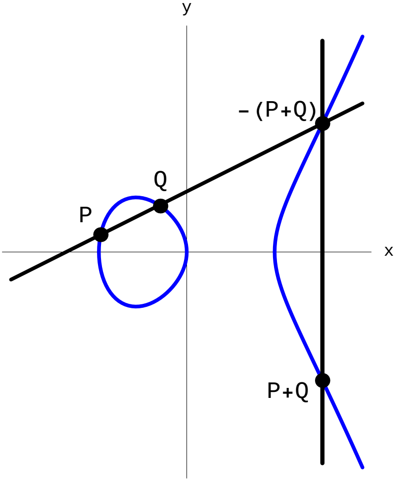
# The Discrete Logarithm Problem

## Definition

- Let $G$ be a cyclic group of order $n$, generated by $g \in G$.
- The *discrete logarithm* of an element $h \in G$, denoted $\mathrm{DLOG}_g(h)$, is the residue class $\alpha \in \mathbb{Z}/n\mathbb{Z}$ satisfying

$$g^\alpha = h.$$

- For additive groups, it's $\alpha P = Q$ instead of $g^\alpha = h$.
- Many cryptographic constructions require a group for which computing DLOG is hard.



# DLOG in various groups

Any group of order $n$:

- $O(\sqrt{p})$ where $p$ is the largest prime divisor of $n$ [Pollard]

Multiplicative group of a finite field $\mathbb{F}_q$:

- $O(L_q(\frac{1}{3}, c))$ where $L_q(\sigma, c) \stackrel{\text{def}}{=} \exp(c(\log q)^\sigma (\log\log q)^{1-\sigma})$

Ideal class group of an imaginary quadratic field:

- $L_n(\frac{1}{2}, c)$ [Hafner, McCurley; Düllmann]

Elliptic curves (with some exceptions):

- $O(\sqrt{p})$ where $p$ is the largest prime divisor of $n$.

Jacobians of hyperelliptic curves of genus $g$ over a finite field $\mathbb{F}_q$:

- $g = 2$: $O(n^{1/2})$
- $g = 3$: $O(n^{4/9})$ [Gaudry, Thomé, Thériault, Diem]
- $g = 4$: $O(n^{3/8})$ [ " ]
- $g \geq \log q$: $O(L_n(\frac{1}{2}, c))$ [Adelman, DeMarrais, Huang; Enge, Gaudry]

# Cryptographic protocols using DLOG & related problems

ElGamal encryption:

- Public key: $g, g^\alpha$. Private key: $\alpha$.
- Encrypt: Choose random $r$. Compute $c = m \cdot (g^\alpha)^r$. Send $(g^r, c)$.
- Decrypt: Compute $m = \dfrac{c}{(g^r)^\alpha}$.

ECDSA:

- Public key: $g, g^\alpha$. Private key: $\alpha$.
- Sign: Choose random $r$. Compute $k = x(rP)$, $s = (\mathrm{Hash}(m) + \alpha k)/r$. Send $(k, s)$.
- Verify: $x\left(\frac{\mathrm{Hash}(m)}{s} + \frac{k}{s}\alpha P\right) \stackrel{?}{=} k$.

Schnorr signatures:

- Public key: $g, g^{-\alpha}$. Private key: $\alpha$.
- Sign: Choose random $r$. Compute $k = \mathrm{Hash}(m||g^r)$, $s = r + \alpha k \pmod{n}$. Send $(k, s)$.
- Verify: $k \stackrel{?}{=} \mathrm{Hash}(m||g^s(g^{-\alpha})^k)$.

# Communications complexity

- Transmitting two group elements takes $2\log n$ bits.
- Computing discrete logarithms takes
  - $O(\sqrt{n})$ time and $O(1)$ space, for $G = E$,
  - $O(L_q(\frac{1}{3}, c))$ time and space, for $G = \mathbb{F}_q^*$.
- Elliptic curves achieve *fully exponential* computational security and *linear* communications complexity as far as we know . . .
- Finite fields can achieve exponential computational security and linear communications complexity, if you "cheat."
  - The trick is to use $G$ = subgroup of $\mathbb{F}_q^*$.
  - Efficiency rapidly degrades as $n$ increases.

NIST
Digital Signature Algorithm:

| Subgroup of size $n$ | Field of size $q$ |
|---|---|
| 160 bits | 1024 bits |
| 224 bits | 2048 bits |
| 256 bits | 3072 bits |
| 384 bits | 7680 bits |
| 512 bits | 15360 bits |

# Outline

1. Elliptic Curves
   - Elliptic Curve Cryptosystems
   - Pairing Based Cryptosystems
2. Isogenies
   - Construction
   - Applications
3. Security issues
   - Reduction proofs
   - Attacks

# Pairings

## Definition

Let $G_1, G_2, G_T$ be cyclic groups of prime order $n$.
A *pairing* is a function $e\colon G_1 \times G_2 \to G_T$ satisfying:

- $e(aP, bQ) = e(P, Q)^{ab}$ (bilinearity)
- $e(P, Q) \neq 1$ for $P, Q \neq 0$ (non-degeneracy)

- Note that $G_1$ and $G_2$ are additive groups, while $G_T$ is multiplicative.

# Pairings

## Definition

Let $G_1, G_2, G_T$ be cyclic groups of prime order $n$.
A *pairing* is a function $e \colon G_1 \times G_2 \longrightarrow G_T$ satisfying:

- $e(aP, bQ) = e(P, Q)^{ab}$ (bilinearity)
- $e(P, Q) \neq 1$ for $P, Q \neq 0$ (non-degeneracy)

- Note that $G_1$ and $G_2$ are additive groups, while $G_T$ is multiplicative.
- Construction of pairings:
  - $G_1, G_2 \subset E$, of prime order $n$, where $E$ is an elliptic curve over $\mathbb{F}_q$.
  - $G_T \subset \mathbb{F}_{q^k}^*$. This implies $n$ divides $q^k - 1$.
  - $e$ equals the Weil pairing or Tate pairing.

# Pairings

## Definition

Let $G_1, G_2, G_T$ be cyclic groups of prime order $n$.
A *pairing* is a function $e\colon G_1 \times G_2 \to G_T$ satisfying:

- $e(aP, bQ) = e(P, Q)^{ab}$ (bilinearity)
- $e(P, Q) \neq 1$ for $P, Q \neq 0$ (non-degeneracy)

- Note that $G_1$ and $G_2$ are additive groups, while $G_T$ is multiplicative.
- Construction of pairings:
  - $G_1, G_2 \subset E$, of prime order $n$, where $E$ is an elliptic curve over $\mathbb{F}_q$.
  - $G_T \subset \mathbb{F}_{q^k}^*$. This implies $n$ divides $q^k - 1$.
  - $e$ equals the Weil pairing or Tate pairing.
- Define $\rho = \frac{\log q}{\log n}$.
- For best communications complexity, we want $\rho$ to be small. Ideally $\rho = 1$.

# Pairing based cryptography

Short signatures [Boneh-Lynn-Shacham]:

- Public key: $P, \alpha P$.
- Private key: $\alpha$.
- Sign: Compute $s = \alpha \cdot \mathrm{Hash}(m)$. Send $s$.
- Verify: $e(\alpha P, \mathrm{Hash}(m)) \stackrel{?}{=} e(P, s)$.

Secure if the *Diffie-Hellman problem* is hard.

**Diffie-Hellman problem**

Given $P, \alpha P, Q \in G$, compute $\alpha Q$.

Note that only one group element is transmitted, as compared to two group elements for DLOG based signatures.

- However, this one element is of length $\rho \log n$.
- If $\rho < 2$, you save bandwidth.
- If $\rho = 2$, bandwidth is the same as before.

# Pairing based cryptography (cont'd)

Identity based encryption [Boneh-Franklin]:

- Master key: $P, \alpha P$
- Private key: $\alpha Q$ where $Q = \mathrm{Hash}(\mathrm{ID})$.
- Encrypt: Choose random $r$, compute $c = e(\alpha P, rQ) \oplus m$, send $(rP, c)$.
- Decrypt: $m = c \oplus e(rP, \alpha Q)$.

Secure if the *bilinear Diffie-Hellman problem* is hard.

**Bilinear Diffie-Hellman problem**

Given $P, aP, bP, Q \in G_i$, compute $e(P, Q)^{ab}$.

- Many other constructions possible ...
  - Broadcast encryption and traitor tracing
  - Blind signatures
  - Aggregate signatures
  - etc.

# Pairing-Friendly Elliptic Curves

For a random elliptic curve $E$, the smallest integer $k$ satisfying $q^k \equiv 1 \bmod n$ is of size $O(n)$.

- $\mathbb{F}_{q^k}$, for $k = O(n)$, cannot be efficiently implemented. Hence, random curves cannot be used.

# Pairing-Friendly Elliptic Curves

For a random elliptic curve $E$, the smallest integer $k$ satisfying $q^k \equiv 1 \bmod n$ is of size $O(n)$.

- $\mathbb{F}_{q^k}$, for $k = O(n)$, cannot be efficiently implemented. Hence, random curves cannot be used.

1. Supersingular elliptic curves:
   - Curves are defined over $\mathbb{F}_p$ or $\mathbb{F}_{p^2}$
   - $k \leq 6$, $\rho = 1$.
   - Many computational optimizations possible.

# Pairing-Friendly Elliptic Curves

For a random elliptic curve $E$, the smallest integer $k$ satisfying $q^k \equiv 1 \bmod n$ is of size $O(n)$.

- $\mathbb{F}_{q^k}$, for $k = O(n)$, cannot be efficiently implemented. Hence, random curves cannot be used.

1. Supersingular elliptic curves:
   - Curves are defined over $\mathbb{F}_p$ or $\mathbb{F}_{p^2}$
   - $k \leq 6$, $\rho = 1$.
   - Many computational optimizations possible.
2. Complex Multiplication curves of low discriminant:
   - $k \leq 12$, $\rho = 1$ [Miyaji-Nakabayashi-Takano, Barreto-Naehrig]
   - $k$ arbitrary, $1 < \rho \leq 2$ [Cocks-Pinch, Barreto-Lynn-Scott, Brezing-Weng]
   - Not as computationally efficient as supersingular curves, especially with $k$ large.

# Solving DLOG via pairings

## Proposition

Let $e\colon G_1 \times G_2 \to G_T$ be a pairing. If you can solve DLOG on $G_T$, then you can solve DLOG on $G_1$ and $G_2$.

Proof: Let $P, \alpha P \in G_1$. Choose $Q \in G_2$, $Q \neq 0$. Compute

- $g = e(P, Q)$,
- $h = e(\alpha P, Q)$.

Note that $h = g^\alpha$ in $G_T$. Compute $\mathrm{DLOG}_g(h) = \alpha$ to find $\alpha$. $\square$

# Solving DLOG via pairings

## Proposition

Let $e\colon G_1 \times G_2 \to G_T$ be a pairing. If you can solve DLOG on $G_T$, then you can solve DLOG on $G_1$ and $G_2$.

Proof: Let $P, \alpha P \in G_1$. Choose $Q \in G_2$, $Q \neq 0$. Compute

- $g = e(P, Q)$,
- $h = e(\alpha P, Q)$.

Note that $h = g^\alpha$ in $G_T$. Compute $\mathrm{DLOG}_g(h) = \alpha$ to find $\alpha$. $\square$

1. For supersingular elliptic curves, DLOG on $G_T = \mathbb{F}_{q^k}$ is easier than on $G_1 = E$ [Menezes-Okamoto-Vanstone].
   - DLOG on $\mathbb{F}_{q^k}$ has $O(L_{q^k}(\frac{1}{3}, c))$ security, and $k \leq 6$.
   - DLOG on $E$ has $O(\sqrt{q})$ security.

# Solving DLOG via pairings

## Proposition

Let $e\colon G_1 \times G_2 \to G_T$ be a pairing. If you can solve DLOG on $G_T$, then you can solve DLOG on $G_1$ and $G_2$.

Proof: Let $P, \alpha P \in G_1$. Choose $Q \in G_2$, $Q \neq 0$. Compute

- $g = e(P, Q)$,
- $h = e(\alpha P, Q)$.

Note that $h = g^\alpha$ in $G_T$. Compute $\mathrm{DLOG}_g(h) = \alpha$ to find $\alpha$. $\square$

1. For supersingular elliptic curves, DLOG on $G_T = \mathbb{F}_{q^k}$ is easier than on $G_1 = E$ [Menezes-Okamoto-Vanstone].
   - DLOG on $\mathbb{F}_{q^k}$ has $O(L_{q^k}(\frac{1}{3}, c))$ security, and $k \leq 6$.
   - DLOG on $E$ has $O(\sqrt{q})$ security.
2. For CM curves, $k$ can grow as needed.
   - $k = O((\log q)^2)$ is needed to achieve overall $O(\sqrt{q})$ security.
   - $G_T$ has size 1024, 2048, 3072, etc. bits for $\log n = 160, 224, 256, \ldots$.

# Comparison of cryptographic primitives

1. Elliptic curve cryptography without pairings:
   - + Can achieve fully exponential computational security
   - − Bandwidth is twice as much as with pairings
   - − Cannot use optimized arithmetic of supersingular curves

# Comparison of cryptographic primitives

1. Elliptic curve cryptography without pairings:
   - \+ Can achieve fully exponential computational security
   - – Bandwidth is twice as much as with pairings
   - – Cannot use optimized arithmetic of supersingular curves
2. Pairing based cryptography with CM curves:
   - \+ Intermediate bandwidth ($1 \leq \rho \leq 2$)
   - \+ For $\rho > 1$, can achieve fully exponential security, by increasing $k$
     - – However, $\rho = 1$ is presently limited to $k \leq 12$.
   - – Implementation cost increases rapidly for fully exponential security

# Comparison of cryptographic primitives

1. Elliptic curve cryptography without pairings:
   - \+ Can achieve fully exponential computational security
   - − Bandwidth is twice as much as with pairings
   - − Cannot use optimized arithmetic of supersingular curves
2. Pairing based cryptography with CM curves:
   - \+ Intermediate bandwidth ($1 \leq \rho \leq 2$)
   - \+ For $\rho > 1$, can achieve fully exponential security, by increasing $k$
     - − However, $\rho = 1$ is presently limited to $k \leq 12$.
   - − Implementation cost increases rapidly for fully exponential security
3. Pairing based cryptography with supersingular curves:
   - \+ Bandwidth is half of that without pairings
   - \+ Can use optimized arithmetic of supersingular curves
   - − Cannot achieve fully exponential security because $k \leq 6$ [MOV]

# Outline

1. Elliptic Curves
   - Elliptic Curve Cryptosystems
   - Pairing Based Cryptosystems
2. Isogenies
   - Construction
   - Applications
3. Security issues
   - Reduction proofs
   - Attacks

# The Diffie-Hellman Problem

## Discrete Logarithm Problem

Given $g, g^\alpha \in G$, compute $\alpha$.

## Diffie-Hellman Problem

Given $g, g^\alpha, h \in G$, compute $h^\alpha$.

- Think of $\alpha$ as a function mapping $g$ to $g^\alpha$.
- Discrete Logarithm Problem: Find the function.
- Diffie-Hellman Problem: Find the value of the function at $h$.
- Note that $\alpha$ as a function is a *group homomorphism*.

# Isogenies

**Definition**

An *isogeny* is a group homomorphism $\phi\colon E_1 \to E_2$ between elliptic curves.

A scalar $\alpha$, when viewed as a homomorphism, is an isogeny: $\alpha\colon E \to E$ sending $P$ to $\alpha P$.

# Isogenies

## Definition

An *isogeny* is a group homomorphism $\phi\colon E_1 \longrightarrow E_2$ between elliptic curves.

A scalar $\alpha$, when viewed as a homomorphism, is an isogeny: $\alpha\colon E \longrightarrow E$ sending $P$ to $\alpha P$.

## Main idea

Replace the scalar isogeny $\alpha\colon E \longrightarrow E$ with some non-scalar isogeny $\phi\colon E_1 \longrightarrow E_2$.

# Isogenies

## Definition

An *isogeny* is a group homomorphism $\phi \colon E_1 \longrightarrow E_2$ between elliptic curves.

A scalar $\alpha$, when viewed as a homomorphism, is an isogeny: $\alpha \colon E \longrightarrow E$ sending $P$ to $\alpha P$.

## Main idea

Replace the scalar isogeny $\alpha \colon E \longrightarrow E$ with some non-scalar isogeny $\phi \colon E_1 \longrightarrow E_2$.

- Introduced by Couveignes in 1997 (eprint 2006/291)

# Isogenies

## Definition

An *isogeny* is a group homomorphism $\phi\colon E_1 \to E_2$ between elliptic curves.

A scalar $\alpha$, when viewed as a homomorphism, is an isogeny: $\alpha\colon E \to E$ sending $P$ to $\alpha P$.

## Main idea

Replace the scalar isogeny $\alpha\colon E \to E$ with some non-scalar isogeny $\phi\colon E_1 \to E_2$.

- Introduced by Couveignes in 1997 (eprint 2006/291)
- Questions raised in that work:

# Isogenies

## Definition

An *isogeny* is a group homomorphism $\phi\colon E_1 \longrightarrow E_2$ between elliptic curves.

A scalar $\alpha$, when viewed as a homomorphism, is an isogeny: $\alpha\colon E \longrightarrow E$ sending $P$ to $\alpha P$.

## Main idea

Replace the scalar isogeny $\alpha\colon E \longrightarrow E$ with some non-scalar isogeny $\phi\colon E_1 \longrightarrow E_2$.

- Introduced by Couveignes in 1997 (eprint 2006/291)
- Questions raised in that work:
  1. Can we evaluate an isogeny on an input point efficiently?

# Isogenies

## Definition

An *isogeny* is a group homomorphism $\phi\colon E_1 \to E_2$ between elliptic curves.

A scalar $\alpha$, when viewed as a homomorphism, is an isogeny: $\alpha\colon E \to E$ sending $P$ to $\alpha P$.

## Main idea

Replace the scalar isogeny $\alpha\colon E \to E$ with some non-scalar isogeny $\phi\colon E_1 \to E_2$.

- Introduced by Couveignes in 1997 (eprint 2006/291)
- Questions raised in that work:
    1. Can we evaluate an isogeny on an input point efficiently?
    2. Can we efficiently select a random isogeny with uniform probability?

An isogeny is a rational morphism. Each coordinate is a quotient of polynomials.

**Definition**

The *degree* of an isogeny is the degree of the polynomials appearing in the coordinate functions.

The only known examples of isogenies that can be efficiently evaluated are:

# 1. Evaluating isogenies

An isogeny is a rational morphism. Each coordinate is a quotient of polynomials.

**Definition**

The *degree* of an isogeny is the degree of the polynomials appearing in the coordinate functions.

The only known examples of isogenies that can be efficiently evaluated are:

1. Isogenies of low degree

# 1. Evaluating isogenies

An isogeny is a rational morphism. Each coordinate is a quotient of polynomials.

**Definition**

The *degree* of an isogeny is the degree of the polynomials appearing in the coordinate functions.

The only known examples of isogenies that can be efficiently evaluated are:

1. Isogenies of low degree
2. Isogenies from a curve to itself (e.g. scalars)

# 1. Evaluating isogenies

An isogeny is a rational morphism. Each coordinate is a quotient of polynomials.

**Definition**

The *degree* of an isogeny is the degree of the polynomials appearing in the coordinate functions.

The only known examples of isogenies that can be efficiently evaluated are:

1. Isogenies of low degree
2. Isogenies from a curve to itself (e.g. scalars)
3. Short compositions of isogenies of the above type

# Example of an isogeny

- $p = 7925599076663155737601$
- $E_1 : y^2 = x^3 + 1204616268305869473 4 * x + 7901506751297038348133$ in $\mathrm{GF}(p)$
- $E_2 : y^2 = x^3 + (3021319262486407622796 * u + 4101162511412606196442) * x + (7040333493178698383420 * u + 1745772756766632103431)$ in $\mathrm{GF}(p^2)$
- $\phi : E_1 \longrightarrow E_2$ given by $\phi(x, y) = ((x^7 + (2646061772402770501474 * u + 2877560530788931592 65) * x^6 + (1329353072286150565 38 * u + 3530390499615039152484) * x^5 + (4637494718376492302 73 * u + 1073811655050424931224) * x^4 + (2474785317056152334847 * u + 1839199255709390890698) * x^3 + (4285381276738035289332 * u + 2268033696082534919907) * x^2 + (1160928171089162069604 * u + 4478674184021543260793) * x + (3220829138361157238167 * u + 4664892256879213165649))/(x^6 + (2646061772402770501474 * u + 2877560530788931592 65) * x^5 + (1945985508507744496834 * u + 648093055215868995 31) * x^4 + (4591727489633569666202 * u + 1570102870983786495532) * x^3 + (1500460390828721967700 * u + 6921704443614513097635) * x^2 + (1297386801518789580736 * u + 2850698740908333936400) * x + (3945372319876153578002 * u + 3619742011015309009 68)), (x^9 * y + (3969092658604155752211 * u + 4394433617949917607698) * x^8 * y + (6535035589862015193348 * u + 7790532914920049821109) * x^7 * y + (1421987375027510985091 * u + 4768123726723570863 6) * x^6 * y + (2303968995096096349661 * u + 3345680927799022267788) * x^5 * y + (2433277735802437441789 * u + 3351794627925587500553) * x^4 * y + (1516026795707698480046 * u + 8182604557381627324 67) * x^3 * y + (1027058177737636125614 * u + 3693613550368489401398) * x^2 * y + (4508645841065025978909 * u + 4918593070183032256585) * x * y + (833381860377767758 0 * u + 6166744817175250513803) * y)/(x^9 + (3969092658604155752211 * u + 4394433617949917607698) * x^8 + (4721985388582885753052 * u + 3330515032350346336461) * x^7 + (3559772126678288264097 * u + 6153422006988745781765) * x^6 + (1902940951990305913452 * u + 8321454977725295839 98) * x^5 + (2553891553651967378833 * u + 5494296243979572742 32) * x^4 + (5821041363528144243281 * u + 4895514527158720628918) * x^3 + (7465572282966743894034 * u + 1236456037884661923 32) * x^2 + (4752216567890970620978 * u + 4978298713068198015 22) * x + (6192295778031003334018 * u + 4253951270570522230194)))$

# 2. Selecting random isogenies

What does it mean to select a random isogeny with uniform probability?

# 2. Selecting random isogenies

What does it mean to select a random isogeny with uniform probability?

0. Assume we are only interested in pairing friendly curves.

# 2. Selecting random isogenies

What does it mean to select a random isogeny with uniform probability?

0. Assume we are only interested in pairing friendly curves.
1. Supersingular curves

# 2. Selecting random isogenies

What does it mean to select a random isogeny with uniform probability?

0. Assume we are only interested in pairing friendly curves.
1. Supersingular curves
   - Supersingular curves only admit isogenies to other supersingular curves.

# 2. Selecting random isogenies

What does it mean to select a random isogeny with uniform probability?

0. Assume we are only interested in pairing friendly curves.
1. Supersingular curves
   - Supersingular curves only admit isogenies to other supersingular curves.
   - The number of supersingular elliptic curves over $\mathbb{F}_q$ is finite.

## 2. Selecting random isogenies

What does it mean to select a random isogeny with uniform probability?

0. Assume we are only interested in pairing friendly curves.
1. Supersingular curves
   - Supersingular curves only admit isogenies to other supersingular curves.
   - The number of supersingular elliptic curves over $\mathbb{F}_q$ is finite.
   - For each pair of curves, the set of functions between that pair is finite.

# 2. Selecting random isogenies

What does it mean to select a random isogeny with uniform probability?

0. Assume we are only interested in pairing friendly curves.
1. Supersingular curves
   - Supersingular curves only admit isogenies to other supersingular curves.
   - The number of supersingular elliptic curves over $\mathbb{F}_q$ is finite.
   - For each pair of curves, the set of functions between that pair is finite.
2. CM curves of low discriminant

# 2. Selecting random isogenies

What does it mean to select a random isogeny with uniform probability?

0. Assume we are only interested in pairing friendly curves.
1. Supersingular curves
   - Supersingular curves only admit isogenies to other supersingular curves.
   - The number of supersingular elliptic curves over $\mathbb{F}_q$ is finite.
   - For each pair of curves, the set of functions between that pair is finite.
2. CM curves of low discriminant
   - CM curves only admit isogenies to other curves of the same field discriminant.

# 2. Selecting random isogenies

What does it mean to select a random isogeny with uniform probability?

0. Assume we are only interested in pairing friendly curves.
1. Supersingular curves
   - Supersingular curves only admit isogenies to other supersingular curves.
   - The number of supersingular elliptic curves over $\mathbb{F}_q$ is finite.
   - For each pair of curves, the set of functions between that pair is finite.
2. CM curves of low discriminant
   - CM curves only admit isogenies to other curves of the same field discriminant.
   - The number of CM curves of a given discriminant is finite.

# 2. Selecting random isogenies

What does it mean to select a random isogeny with uniform probability?

0. Assume we are only interested in pairing friendly curves.
1. Supersingular curves
   - Supersingular curves only admit isogenies to other supersingular curves.
   - The number of supersingular elliptic curves over $\mathbb{F}_q$ is finite.
   - For each pair of curves, the set of functions between that pair is finite.
2. CM curves of low discriminant
   - CM curves only admit isogenies to other curves of the same field discriminant.
   - The number of CM curves of a given discriminant is finite.
   - For each pair of curves, the set of functions between that pair is finite.

# 2. Selecting random isogenies

What does it mean to select a random isogeny with uniform probability?

0. Assume we are only interested in pairing friendly curves.
1. Supersingular curves
   - Supersingular curves only admit isogenies to other supersingular curves.
   - The number of supersingular elliptic curves over $\mathbb{F}_q$ is finite.
   - For each pair of curves, the set of functions between that pair is finite.
2. CM curves of low discriminant
   - CM curves only admit isogenies to other curves of the same field discriminant.
   - The number of CM curves of a given discriminant is finite.
   - For each pair of curves, the set of functions between that pair is finite.

A random isogeny means: pick a random pair of curves, and select a random isogeny within that pair.

# 2. Selecting random isogenies (cont'd)

**Theorem (Jao, Miller, Venkatesan)**

Assuming the generalized Riemann hypothesis, a random composition of polynomially many isogenies of polynomially bounded degree produces a near-uniform distribution of isogenies among CM curves of a given discriminant.

# 2. Selecting random isogenies (cont'd)

**Theorem (Jao, Miller, Venkatesan)**

Assuming the generalized Riemann hypothesis, a random composition of polynomially many isogenies of polynomially bounded degree produces a near-uniform distribution of isogenies among CM curves of a given discriminant.

**Theorem (Mestre, Pizer)**

A random composition of polynomially many isogenies of degree $\leq 3$ produces a near-uniform distribution of isogenies among supersingular curves of a given characteristic.

# 2. Selecting random isogenies (cont'd)

**Theorem (Jao, Miller, Venkatesan)**

Assuming the generalized Riemann hypothesis, a random composition of polynomially many isogenies of polynomially bounded degree produces a near-uniform distribution of isogenies among CM curves of a given discriminant.

**Theorem (Mestre, Pizer)**

A random composition of polynomially many isogenies of degree $\leq 3$ produces a near-uniform distribution of isogenies among supersingular curves of a given characteristic.

**Theorem (Enge)**

An isogeny of degree $d$ can be obtained in quasi-linear time.

# 2. Selecting random isogenies (cont'd)

**Theorem (Jao, Miller, Venkatesan)**

Assuming the generalized Riemann hypothesis, a random composition of polynomially many isogenies of polynomially bounded degree produces a near-uniform distribution of isogenies among CM curves of a given discriminant.

**Theorem (Mestre, Pizer)**

A random composition of polynomially many isogenies of degree $\leq 3$ produces a near-uniform distribution of isogenies among supersingular curves of a given characteristic.

**Theorem (Enge)**

An isogeny of degree $d$ can be obtained in quasi-linear time.

**Corollary:** Random isogenies can be efficiently constructed and evaluated by composing random low degree isogenies together with random scalars.

# Outline

# ElGamal encryption with isogenies

## Isogeny Diffie-Hellman problem

Let $\phi\colon E_1 \to E_2$ be an isogeny. Given $P, Q \in E_1$ and $\phi(P) \in E_2$, compute $\phi(Q) \in E_2$.

ElGamal encryption:

- Public key: $P \in E_1$, $\phi(P) \in E_2$.
- Private key: $\phi$.
- Encryption: Choose random $r$. Compute $c = m + r\phi(P)$. Send $(rP, c)$.
- Decryption: Compute $m = c - \phi(rP)$.

Provably secure assuming that Isogeny Diffie-Hellman is hard.

# Short signatures

## Isogeny equivariance

Let $\phi \colon E_1 \to E_2$ be an isogeny. There is a unique *dual isogeny* $\hat{\phi} \colon E_2 \to E_1$ such that

$$e(\phi(P), Q) = e(P, \hat{\phi}(Q))$$

for $P \in E_1$ and $Q \in E_2$.

A short signature scheme using isogenies:

- Public key: $P \in E_1$, $\phi(P) \in E_2$.
- Private key: $\phi$.
- Sign: Compute $s = \hat{\phi}(\mathrm{Hash}(m))$. where $\mathrm{Hash}(m) \in E_2$. Send $s$.
- Verify: $e(\phi(P), \mathrm{Hash}(m)) \stackrel{?}{=} e(P, s)$.

Provably secure assuming that Dual Isogeny Diffie-Hellman is hard.

## Dual Isogeny Diffie-Hellman

Given $P \in E_1$ and $\phi(P), Q \in E_2$, compute $\hat{\phi}(Q) \in E_1$.

# Identity based encryption

Identity based encryption using isogenies:

- Master key: $P \in E_1$, $\phi(P) \in E_2$
- Private key: $\hat{\phi}(Q)$ where $Q = \text{Hash}(\text{ID}) \in E_2$.
- Encrypt: Choose random $r$, compute $c = e(\phi P, rQ) \oplus m$, send $(rP, c)$.
- Decrypt: $m = c \oplus e(rP, \hat{\phi}(Q))$.

Provably secure assuming that Isogeny Bilinear Diffie-Hellman is hard.

**Isogeny Bilinear Diffie-Hellman**

Given $P, rP, Q \in E_1$ and $\phi(P) \in E_2$, compute $e(\phi(P), rQ)$.

# Outline

# Relationship to discrete logarithms

**Theorem**

An algorithm $\mathcal{A}$ which solves Isogeny Diffie-Hellman with non-negligible probability can solve Diffie-Hellman with non-negligible probability.

# Relationship to discrete logarithms

**Theorem**

An algorithm $\mathcal{A}$ which solves Isogeny Diffie-Hellman with non-negligible probability can solve Diffie-Hellman with non-negligible probability.

Sketch of proof:

# Relationship to discrete logarithms

## Theorem

An algorithm $\mathcal{A}$ which solves Isogeny Diffie-Hellman with non-negligible probability can solve Diffie-Hellman with non-negligible probability.

Sketch of proof:

- Let $(P, \alpha P, Q)$ be a Diffie-Hellman triple.

# Relationship to discrete logarithms

**Theorem**

An algorithm $\mathcal{A}$ which solves Isogeny Diffie-Hellman with non-negligible probability can solve Diffie-Hellman with non-negligible probability.

Sketch of proof:

- Let $(P, \alpha P, Q)$ be a Diffie-Hellman triple.
- Construct a random, efficiently computable isogeny $\phi$.

# Relationship to discrete logarithms

**Theorem**

An algorithm $\mathcal{A}$ which solves Isogeny Diffie-Hellman with non-negligible probability can solve Diffie-Hellman with non-negligible probability.

Sketch of proof:

- Let $(P, \alpha P, Q)$ be a Diffie-Hellman triple.
- Construct a random, efficiently computable isogeny $\phi$.
- Evaluate $\mathcal{A}$ on $(P, \phi(\alpha P), Q)$.

# Relationship to discrete logarithms

**Theorem**

An algorithm $\mathcal{A}$ which solves Isogeny Diffie-Hellman with non-negligible probability can solve Diffie-Hellman with non-negligible probability.

Sketch of proof:

- Let $(P, \alpha P, Q)$ be a Diffie-Hellman triple.
- Construct a random, efficiently computable isogeny $\phi$.
- Evaluate $\mathcal{A}$ on $(P, \phi(\alpha P), Q)$.
- Eventually $\mathcal{A}$ will return $\phi(\alpha Q)$.

# Relationship to discrete logarithms

**Theorem**

An algorithm $\mathcal{A}$ which solves Isogeny Diffie-Hellman with non-negligible probability can solve Diffie-Hellman with non-negligible probability.

Sketch of proof:

- Let $(P, \alpha P, Q)$ be a Diffie-Hellman triple.
- Construct a random, efficiently computable isogeny $\phi$.
- Evaluate $\mathcal{A}$ on $(P, \phi(\alpha P), Q)$.
- Eventually $\mathcal{A}$ will return $\phi(\alpha Q)$.
- Compute $\phi^{-1}\phi(\alpha Q) = \alpha Q$.

# Relationship to discrete logarithms

**Theorem**

An algorithm $\mathcal{A}$ which solves Isogeny Diffie-Hellman with non-negligible probability can solve Diffie-Hellman with non-negligible probability.

Sketch of proof:

- Let $(P, \alpha P, Q)$ be a Diffie-Hellman triple.
- Construct a random, efficiently computable isogeny $\phi$.
- Evaluate $\mathcal{A}$ on $(P, \phi(\alpha P), Q)$.
- Eventually $\mathcal{A}$ will return $\phi(\alpha Q)$.
- Compute $\phi^{-1}\phi(\alpha Q) = \alpha Q$.

# Relationship to discrete logarithms

**Theorem**

An algorithm $\mathcal{A}$ which solves Isogeny Diffie-Hellman with non-negligible probability can solve Diffie-Hellman with non-negligible probability.

Sketch of proof:

- Let $(P, \alpha P, Q)$ be a Diffie-Hellman triple.
- Construct a random, efficiently computable isogeny $\phi$.
- Evaluate $\mathcal{A}$ on $(P, \phi(\alpha P), Q)$.
- Eventually $\mathcal{A}$ will return $\phi(\alpha Q)$.
- Compute $\phi^{-1}\phi(\alpha Q) = \alpha Q$.

**Theorem**

An algorithm $\mathcal{A}$ which solves Dual Isogeny Diffie-Hellman with non-negligible probability can solve Diffie-Hellman with non-negligible probability.

# Outline

# Attacking the Isogeny Diffie-Hellman problem

Method #1: Find the isogeny.

# Attacking the Isogeny Diffie-Hellman problem

Method #1: Find the isogeny.

- Regular Diffie-Hellman on an elliptic curve takes $O(\sqrt{n})$ operations to solve (birthday paradox).

# Attacking the Isogeny Diffie-Hellman problem

Method #1: Find the isogeny.

- Regular Diffie-Hellman on an elliptic curve takes $O(\sqrt{n})$ operations to solve (birthday paradox).
- Let $\phi\colon E_1 \to E_2$ be an isogeny. Finding the isogeny involves two steps:

# Attacking the Isogeny Diffie-Hellman problem

Method #1: Find the isogeny.

- Regular Diffie-Hellman on an elliptic curve takes $O(\sqrt{n})$ operations to solve (birthday paradox).
- Let $\phi\colon E_1 \to E_2$ be an isogeny. Finding the isogeny involves two steps:
  1. **Isogeny stage:** Find **any** isogeny $\psi\colon E_1 \to E_2$.

# Attacking the Isogeny Diffie-Hellman problem

Method #1: Find the isogeny.

- Regular Diffie-Hellman on an elliptic curve takes $O(\sqrt{n})$ operations to solve (birthday paradox).
- Let $\phi\colon E_1 \to E_2$ be an isogeny. Finding the isogeny involves two steps:
  1. **Isogeny stage:** Find **any** isogeny $\psi\colon E_1 \to E_2$.
  2. **DLOG stage:** Find the scalar $\alpha$ such that $\alpha\psi = \phi$.

# Attacking the Isogeny Diffie-Hellman problem

Method #1: Find the isogeny.

- Regular Diffie-Hellman on an elliptic curve takes $O(\sqrt{n})$ operations to solve (birthday paradox).
- Let $\phi\colon E_1 \to E_2$ be an isogeny. Finding the isogeny involves two steps:
  1. **Isogeny stage:** Find **any** isogeny $\psi\colon E_1 \to E_2$.
  2. **DLOG stage:** Find the scalar $\alpha$ such that $\alpha\psi = \phi$.
- DLOG stage takes $O(\sqrt{n})$ operations.

## Attacking the Isogeny Diffie-Hellman problem

Method #1: Find the isogeny.

- Regular Diffie-Hellman on an elliptic curve takes $O(\sqrt{n})$ operations to solve (birthday paradox).
- Let $\phi\colon E_1 \to E_2$ be an isogeny. Finding the isogeny involves two steps:
  1. **Isogeny stage:** Find **any** isogeny $\psi\colon E_1 \to E_2$.
  2. **DLOG stage:** Find the scalar $\alpha$ such that $\alpha\psi = \phi$.
- DLOG stage takes $O(\sqrt{n})$ operations.
- Isogeny stage requires $O(\sqrt{N})$ operations, where $N$ is the number of possible curves [Galbraith-Hess-Smart].

# Attacking the Isogeny Diffie-Hellman problem

Method #1: Find the isogeny.

- Regular Diffie-Hellman on an elliptic curve takes $O(\sqrt{n})$ operations to solve (birthday paradox).
- Let $\phi\colon E_1 \to E_2$ be an isogeny. Finding the isogeny involves two steps:
  1. **Isogeny stage:** Find **any** isogeny $\psi\colon E_1 \to E_2$.
  2. **DLOG stage:** Find the scalar $\alpha$ such that $\alpha\psi = \phi$.
- DLOG stage takes $O(\sqrt{n})$ operations.
- Isogeny stage requires $O(\sqrt{N})$ operations, where $N$ is the number of possible curves [Galbraith-Hess-Smart].
  - For CM curves of low discriminant $D$, we have $N = O(\sqrt{D})$ and $O(\sqrt{N}) = O(D^{1/4})$.

# Attacking the Isogeny Diffie-Hellman problem

Method #1: Find the isogeny.

- Regular Diffie-Hellman on an elliptic curve takes $O(\sqrt{n})$ operations to solve (birthday paradox).
- Let $\phi\colon E_1 \to E_2$ be an isogeny. Finding the isogeny involves two steps:
  1. **Isogeny stage:** Find **any** isogeny $\psi\colon E_1 \to E_2$.
  2. **DLOG stage:** Find the scalar $\alpha$ such that $\alpha\psi = \phi$.
- DLOG stage takes $O(\sqrt{n})$ operations.
- Isogeny stage requires $O(\sqrt{N})$ operations, where $N$ is the number of possible curves [Galbraith-Hess-Smart].
  - For CM curves of low discriminant $D$, we have $N = O(\sqrt{D})$ and $O(\sqrt{N}) = O(D^{1/4})$.
  - For supersingular curves over $\mathbb{F}_q$, we have $N = O(\sqrt{q})$ and $O(\sqrt{N}) = O(q^{1/4})$.

# Attacking the Isogeny Diffie-Hellman problem (cont'd)

For CM curves of low discriminant $D$, the isogeny stage takes $O(D^{1/4})$ operations, and the DLOG stage takes $O(n^{1/2})$ operations.

- $O(D^{1/4}) \ll O(n^{1/2})$, since $D$ must be small.
- Open question: Construct a pairing friendly curve of large $D$.
- System is secure because DLOG stage is intractable.

For CM curves of low discriminant $D$, the isogeny stage takes $O(D^{1/4})$ operations, and the DLOG stage takes $O(n^{1/2})$ operations.

- $O(D^{1/4}) \ll O(n^{1/2})$, since $D$ must be small.
- Open question: Construct a pairing friendly curve of large $D$.
- System is secure because DLOG stage is intractable.

For supersingular curves, the isogeny stage takes $O(q^{1/4})$ operations, and the DLOG stage takes $O(L_n(\frac{1}{3}, c))$ operations.

- $O(q^{1/4}) \gg O(L_n(\frac{1}{3}, c))$.
- System is conjecturally more secure than DLOG alone.
- System is *not less secure* than Diffie-Hellman.

Method #2: Evaluate the isogeny on points without finding the isogeny.

- Recall that you are given $P, Q \in E_1$ and $\phi(P) \in E_2$.
- Suppose $Q = \alpha P$. Find $\alpha$ using a DLOG solver.
- Then $\phi(Q) = \phi(\alpha P) = \alpha\phi(P)$. Knowing $\alpha$, you can compute $\phi(Q)$.

$$\begin{tikzcd} P \arrow[r, "\phi"] \arrow[d, dotted, "\alpha"'] & \phi(P) \arrow[d, dotted, "\alpha"] \\ Q \arrow[r, "\phi"'] & \phi(Q) \end{tikzcd}$$

# Attacking the Isogeny Diffie-Hellman problem

Method #2: Evaluate the isogeny on points without finding the isogeny.

- Recall that you are given $P, Q \in E_1$ and $\phi(P) \in E_2$.
- Suppose $Q = \alpha P$. Find $\alpha$ using a DLOG solver.
- Then $\phi(Q) = \phi(\alpha P) = \alpha\phi(P)$. Knowing $\alpha$, you can compute $\phi(Q)$.

$$\begin{tikzcd} P \arrow[r, "\phi"] \arrow[d, dotted, "\alpha"'] & \phi(P) \arrow[d, dotted, "\alpha"] \\ Q \arrow[r, "\phi"'] & \phi(Q) \end{tikzcd}$$

- This does not contradict the proof that Isogeny Diffie-Hellman is at least as secure as Diffie-Hellman.
- Requires a new discrete logarithm computation (of subexponential complexity) each time you break a message.

# Attacking the Isogeny Diffie-Hellman problem

Method #2: Evaluate the isogeny on points without finding the isogeny.

- Recall that you are given $P, Q \in E_1$ and $\phi(P) \in E_2$.
- Suppose $Q = \alpha P$. Find $\alpha$ using a DLOG solver.
- Then $\phi(Q) = \phi(\alpha P) = \alpha\phi(P)$. Knowing $\alpha$, you can compute $\phi(Q)$.

$$\begin{tikzcd} P \arrow[r, "\phi"] \arrow[d, dotted, "\alpha"'] & \phi(P) \arrow[d, dotted, "\alpha"] \\ Q \arrow[r, "\phi"'] & \phi(Q) \end{tikzcd}$$

- This does not contradict the proof that Isogeny Diffie-Hellman is at least as secure as Diffie-Hellman.
- Requires a new discrete logarithm computation (of subexponential complexity) each time you break a message.
- Even index calculus algorithms require subexponential time and space per invocation.

# Security reductions

$$\begin{tikzcd}
 & \mathrm{DL}_{G_i} \arrow[dl] & & \mathrm{DH}_{G_i} \arrow[dl] & \\
\mathrm{DL}_{G_T} & \mathrm{PI} \arrow[l] & \mathrm{IDH} \arrow[l] \arrow[ul] & \mathrm{IBDH} \arrow[l] \arrow[dl] & \mathrm{BDH} \arrow[l] \arrow[ul] \\
 & & \mathrm{DH}_{G_T} \arrow[ul] & &
\end{tikzcd}$$

Legend:

- DL = Discrete Logarithm
- PI = Pairing Inversion
- DH = Diffie-Hellman
- IDH = [Dual] Isogeny Diffie-Hellman
- BDH = Bilinear Diffie-Hellman
- IBDH = Isogeny Bilinear Diffie-Hellman

# Conclusions and open questions

1. Isogenies on supersingular curves:
   - + Achieves fully exponential security, assuming that Isogeny Diffie-Hellman is of exponential difficulty.
   - + Can use optimized arithmetic of supersingular curves.
   - + Provably *not less secure* than regular Diffie-Hellman.
   - – Same bandwidth as without using pairings (because of $q^{1/4}$ security).
   - – Can break individual messages using DLOG.
2. Isogenies on CM curves of low discriminant:
   - + Provably *not less secure* than regular Diffie-Hellman.
   - – With low discriminants, does not appear to be any more secure.
   - – Can break individual messages using DLOG.
3. Open questions:
   - ? Need pairing friendly curves of high discriminant for added security.
   - ? Quantify the security relationship between Isogeny Diffie-Hellman and other DLOG based problems.