Pre- and post-quantum Diffie–Hellman from groups, actions, and isogenies

Benjamin Smith

CARAMBA Seminar // LORIA, Nancy // May 14, 2019

Inria + Laboratoire d'Informatique de l'École polytechnique (LIX)

Let's talk about cryptographic **key exchange**.

The **problem**: two parties, "Alice" and "Bob", want to establish a **shared secret** over a **public channel**.

Solution: **Diffie–Hellman key exchange** (1976).

- Originally set in $\mathbb{G}_m(\mathbb{F}_q)$, but works in any cyclic group.
- Current state of the art: **elliptic curves**.
- Elliptic-curve DH security depends on problems that are classically hard but quantumly easy.

How can we **replace Diffie–Hellman** for a **post-quantum world**?

Classical Diffie–Hellman

The group setting for Diffie–Hellman

Consider a **finite cyclic group**

$$\mathcal{G} = \langle P \rangle \cong \mathbb{Z}/N\mathbb{Z}\,.$$

The most important operation is **scalar multiplication**:

$$[m]P := P + P + \cdots + P \quad (m \text{ copies of } P)\,,$$

for $P \in \mathcal{G}$ and m in $\mathbb{Z}$, with $[-m]P := [m](-P)$.

Inverting it is the **Discrete Logarithm Problem (DLP)** in $\mathcal{G}$:

$$\text{given } P \text{ and } Q = [x]P,\ \text{ compute } x\,.$$

Classic Diffie–Hellman key exchange

Phase 1	**Alice**	samples a secret $a \in \mathbb{Z}/N\mathbb{Z}$;
		Computes $A := [a]P$ and publishes A
	Bob	samples a secret $b \in \mathbb{Z}/N\mathbb{Z}$;
		computes $B := [b]P$ and publishes B

Breaking keypairs (e.g. recovering a from A) is the DLP.

Phase 2	**Alice**	computes $S = [a]B$.
	Bob	computes $S = [b]A$.

The protocol correctly computes a **shared secret** because

$$A = [a]P \qquad B = [b]P \qquad S = [ab]P$$

Recovering the secret S given only the public data P, A, B is the **Computational Diffie–Hellman Problem** (CDHP).

Static and ephemeral DH

Ephemeral: Alice & Bob use keypairs unique to this session. *Ephemeral DH is essentially **interactive***.

Static: Alice and/or Bob use long-term keypairs, which may be re-used across sessions. *Static DH can be **non-interactive***.

Static DH security requires public **key validation**:
i.e. checking public keys are **legitimate `KeyPair()` outputs**.
So far, this just means checking the key is in $\mathcal{G}$, which is easy.

Complex protocols may **mix ephemeral & static**.
*Example: **X3DH** initializes conversations in Signal & WhatsApp using **four** `DH()` calls, mixing ephemeral and longer-term keys.*

Conventional CDHP and DLP Hardness

Currently, our best algorithm for solving CDHP is to solve DLP.

Generic algorithms solve DLP instances in $O(\sqrt{\#\mathcal{G}})$:
— Shanks' Baby-step giant-step, Pollard ρ, etc...

Pohlig–Hellman–Silver: when the structure of $\mathcal{G}$ is known, solve DLP instances in $O(\sqrt{\#(\text{largest prime subgroup of } \mathcal{G})})$.

Faster DLP algorithms exist for many **concrete groups**:

- $\mathcal{G} \subset \mathbb{F}_p^\times$: subexponential DLP. Number Field Sieve: $L_p(1/3)$.
- $\mathcal{G} \subset \mathbb{F}_{p^n}^\times$ with p very small: quasipolynomial DLP.

Today's **hardest** DLP instances come from **elliptic curves**.

Elliptic curves

Elliptic curves are a convenient source of groups that can **replace multiplicative groups** in asymmetric crypto.

Classic **"short" Weierstrass model**:

$$\mathcal{E}/\mathbb{F}_p : y^2 = x^3 + ax + b \quad \text{with} \quad a, b \in \mathbb{F}_p, 4a^3 + 27b^2 \neq 0\,.$$

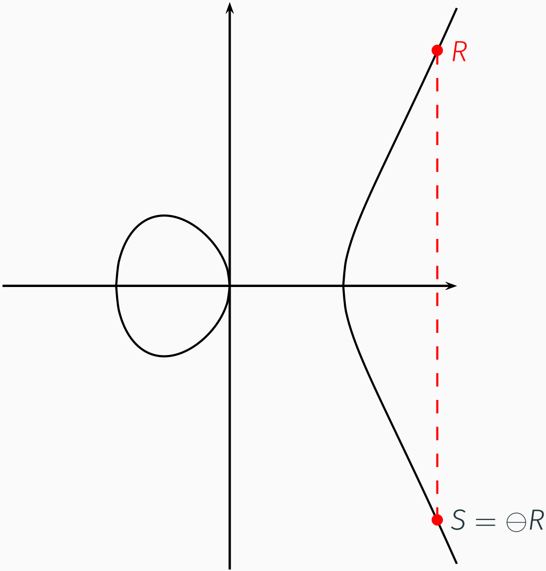
The **points** on $\mathcal{E}$ are

$$\mathcal{E}(\mathbb{F}_p) = \left\{(\alpha, \beta) \in \mathbb{F}_p^2 : \beta^2 = \alpha^3 + a \cdot \alpha + b\right\} \cup \{\mathcal{O}_\mathcal{E}\}$$

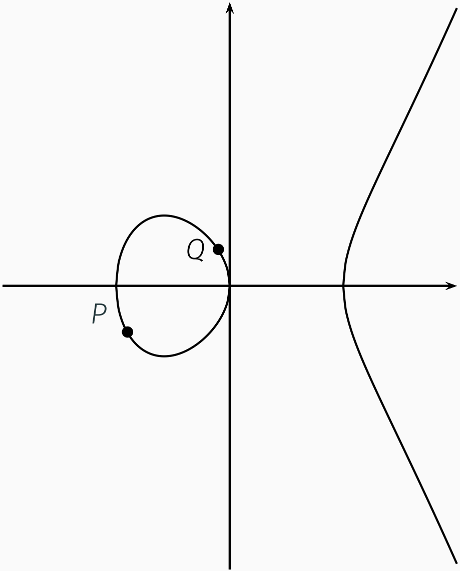
where $\mathcal{O}_\mathcal{E}$ is the unique **"point at infinity"**.

$\mathcal{E}(\mathbb{F}_p)$ is an algebraic group, with $\mathcal{O}_\mathcal{E}$ the identity element.

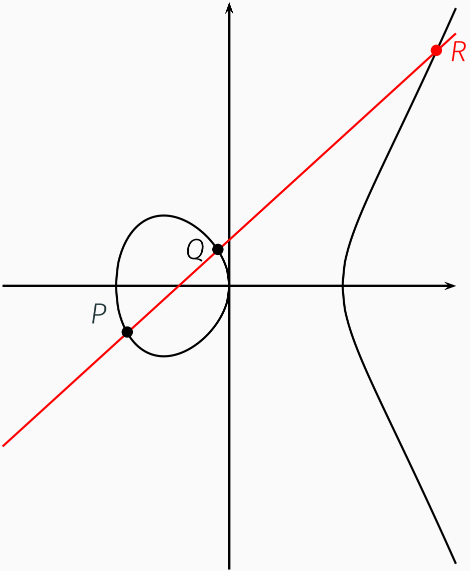
Elliptic curve negation: $\ominus R = S$



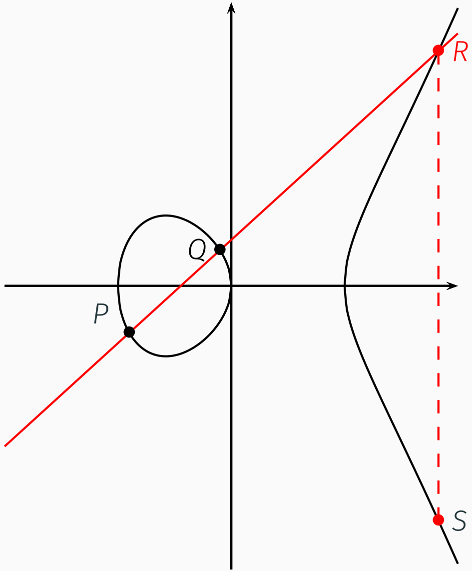
Elliptic curve addition: $P \oplus Q = ?$



Elliptic curve addition: $P \oplus Q \oplus R = 0$



Elliptic curve addition: $P \oplus Q = \ominus R = S$



Elliptic curve group operations

If $P = Q$, the **chord** through P and Q degenerates to a **tangent**.

The important thing is that elliptic curve group operations, being geometric, have **algebraic expressions**.

$\implies$ They can be computed as a series of $\mathbb{F}_p$-operations, which can in turn be reduced to a series of machine instructions.

In particular, **negation:** $\ominus(x, y) = (x, -y)$ and $\ominus\mathcal{O}_\mathcal{E} = \mathcal{O}_\mathcal{E}$. *Up to "sign", group elements are encoded by x-coordinates.*

The Elliptic Curve Discrete Logarithm Problem (ECDLP)

Amazing fact: for subgroups $\mathcal{G}$ of **general**[1] **elliptic curves**, we still do not know how to solve discrete logs significantly faster than by using **generic black-box group algorithms**.

In particular: currently, for prime-order $\mathcal{G} \subseteq \mathcal{E}(\mathbb{F}_p)$, we can do no better than $O(\sqrt{\#\mathcal{G}})$.

Apart from improvements in distributed computing, and a constant-factor speedup of about $\sqrt{2}$, there has been **absolutely no progress** on general ECDLP algorithms. *Ever.*

Current world record for prime-order ECDLP: in a 112-bit group, which is a *long* way away from the 256-bit groups we use today!

[1] That is, for all but a very small and easily identifiable subset of curves.

The quantum menace

Shor's quantum algorithm solves DLPs in **polynomial time**.

Global effort: replacing group-based public-key cryptosystems with **post-quantum** alternatives.

NIST has started a standardization process ("non-competition") for postquantum public-key cryptosystems.

The process has **many** candidate **Key Encapsulation Mechanisms**, but **no direct Diffie–Hellman replacements** because most major postquantum settings (lattices, codes, multivariate, hashes) don't have *exact* DH equivalents.

Modern Diffie–Hellman

Modern Elliptic Curve Diffie–Hellman (ECDH)

Classic ECDH is just classic DH with $\mathcal{E}(\mathbb{F}_q)$ in place of $\mathbb{G}_m(\mathbb{F}_q)$:

$$A = [a]P \qquad B = [b]P \qquad S = [ab]P$$

Miller (1985) suggested ECDH using **only x-coordinates:**

$$\begin{aligned} A &= x([a]P) & B &= x([b]P) & S &= x([ab]P) \\ &= \pm[a]P & &= \pm[b]P & &= \pm[ab]P \end{aligned}$$

We compute $x(Q) \mapsto x([m]Q)$ with **differential addition chains** such as the **Montgomery ladder**.

We have **replaced** $\mathcal{G} \subset \mathcal{E}(\mathbb{F}_q)$ with a **quotient set** $\mathcal{G}/\langle\pm 1\rangle \subset \mathbb{F}_q$.

Example: `Curve25519` (Bernstein 2006), the benchmark for conventional DH (and now standard in TLS 1.3).

Modern ECDH: where is the group?

Modern x-only ECDH is interesting: it highlights the fact that **Diffie–Hellman does not explicitly require a group operation**.

$$A = [a]P \qquad B = [b]P \qquad S = [ab]P$$

Formally, we have **an action of $\mathbb{Z}$ on a set $\mathcal{X}$** (here, $\mathcal{X} = \mathcal{G}/\langle\pm 1\rangle$).

In fact, the quotient structure $\mathcal{G}/\langle\pm 1\rangle$ is important: it facilitates

- ***security proofs** by relating CDHPs in $\mathcal{X}$ and $\mathcal{G}$*
- ***efficient evaluation** of the $\mathbb{Z}$-action on $\mathcal{X}$: $\oplus$ on $\mathcal{G}$ induces an operation $(\pm P, \pm Q, \pm(P-Q)) \mapsto \pm(P+Q)$ on $\mathcal{X}$, which we can use to compute $(m, x(P)) \mapsto x([m]P)$ using differential addition chains.*

Towards postquantum Diffie–Hellman: Hard Homogeneous Spaces

Towards postquantum Diffie–Hellman

Starting point for postquantum DH: an obscure framework proposed by Couveignes in 1997, ***Hard Homogeneous Spaces***.

Old DH	$\mathbb{Z}$ acts on a group $\mathcal{G}$
Modern DH	$\mathbb{Z}$ acts on a set $\mathcal{X}$ (via a group $\mathcal{G}$)
HHS-DH	a group $\mathfrak{G}$ acts on a set $\mathcal{X}$.

(We use the symbol $\mathfrak{G}$ for groups written multiplicatively, and $\mathcal{G}$ for groups written additively.)

Homogeneous Spaces

Let $\mathfrak{G}$ be a finite commutative group acting on a set $\mathcal{X}$.

This means: for each $\mathfrak{g} \in \mathfrak{G}$ and $P \in \mathcal{X}$, there is a $\mathfrak{g} \cdot P \in \mathcal{X}$, and

$$\mathfrak{a} \cdot (\mathfrak{b} \cdot P) = \mathfrak{a}\mathfrak{b} \cdot P \qquad \forall \mathfrak{a}, \mathfrak{b} \in \mathfrak{G}, \quad \forall P \in \mathcal{X} .$$

$\mathcal{X}$ is a **principal homogeneous space** (PHS) under $\mathfrak{G}$ if

$$P, Q \in \mathcal{X} \implies \exists!\ \mathfrak{g} \in \mathfrak{G} \text{ such that } Q = \mathfrak{g} \cdot P .$$

So: $\varphi_P : \mathfrak{g} \mapsto \mathfrak{g} \cdot P$ is a bijection $\mathfrak{G} \to \mathcal{X}$ for each $P \in \mathcal{X}$.

Example: $\mathfrak{G} =$ a vector space, $\mathcal{X} =$ the underlying affine space.

Examples of Homogeneous Spaces

A PHS is like a copy of $\mathfrak{G}$ with the identity $1_{\mathfrak{G}}$ forgotten.

Each map $\varphi_P : \mathfrak{g} \mapsto \mathfrak{g} \cdot P$ endows $\mathcal{X}$ with the structure of $\mathfrak{G}$, with P as the identity element, via

$$(\mathfrak{a} \cdot P)(\mathfrak{b} \cdot P) = \varphi_P(\mathfrak{a})\varphi_P(\mathfrak{b}) := \varphi_P(\mathfrak{a}\mathfrak{b}) = (\mathfrak{a}\mathfrak{b}) \cdot P\,.$$

Each choice of P yields a different group structure on $\mathcal{X}$.

DH in a group again

Expressing DH in a group as functions `KeyPair` and `DH`:

Algorithm 1: Key generation for a group $\mathcal{G} = \langle P \rangle$

```
function KeyPair( )
    x ← Random(ℤ/Nℤ)
    Q ← [x]P                    // Scalar multiplication
    return (Q, x)               // (Public, private)
```

Algorithm 2: Compute a Diffie–Hellman shared secret

```
function DH(m ∈ ℤ, Q ∈ 𝒢)
    S ← [m]Q                    // Scalar multiplication
    return S                    // Shared secret
```

DH in a PHS

We define analogous functions `KeyPair` and `DH` for a PHS:

Algorithm 3: Key generation for a PHS $(\mathfrak{G}, \mathcal{X})$

function `KeyPair()`
    $\mathfrak{x} \leftarrow$ `Random(`$\mathfrak{G}$`)`
    $Q \leftarrow \mathfrak{x} \cdot P$ `// Group action`
    return $(Q, \mathfrak{x})$ `// (Public, private)`

Algorithm 4: Compute a Diffie–Hellman shared secret

function DH($\mathfrak{m} \in \mathfrak{G}, Q \in \mathcal{X}$)
    $S \leftarrow \mathfrak{m} \cdot Q$ `// Group action`
    return S `// Shared secret`

A Diffie–Hellman analogue

We have an **obvious analogy** between Group-DH and HHS-DH:

$$A = [a]P \qquad B = [b]P \qquad S = [ab]P$$

$$A = \mathfrak{a} \cdot P \qquad B = \mathfrak{b} \cdot P \qquad S = \mathfrak{a}\mathfrak{b} \cdot P$$

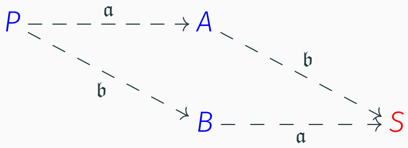
Security: need PHS analogues of DLP and CDHP to be hard.

Hard Homogeneous Spaces

Vectorization (Vec: breaking public keys):
Given P and Q in $\mathcal{X}$, compute the (unique) $\mathfrak{g} \in \mathfrak{G}$ s.t. $Q = \mathfrak{g} \cdot P$.

$$P \overset{\mathfrak{g}}{\dashrightarrow} Q$$

Parallelization (Par: recovering shared secrets):
Given P, A, B in $\mathcal{X}$ with $A = \mathfrak{a} \cdot P$, $B = \mathfrak{b} \cdot P$, compute $S = (\mathfrak{a}\mathfrak{b}) \cdot P$.



Hard homogeneous spaces

A **Hard Homogeneous Space (HHS)** is a PHS where VEC and PAR are computationally infeasible.

We will give an example of a conjectural HHS later.

We have a lot **intuition** and folklore about DLP and CDHP.

- Decades of algorithmic study
- Conditional polynomial-time equivalences

What carries over to VEC and PAR?

...It's only an analogy

Warning: HHS-DH is **not a true generalization** of Group-DH.

For group-DH in a group $\mathcal{G}$ of order N:

- Group-DH scalars are elements of $\mathbb{Z}/N\mathbb{Z}$
- The group operation in $\mathbb{Z}/N\mathbb{Z}$ is $+$, not the $\times$ of Group-DH.
- Scalars do *not* form a group under $\times$.

Homogeneous spaces from cyclic groups

However, there is a hack relating important **special cases**.

Given a cyclic $\mathcal{G}$ of order N, we have a PHS

$$Exp(\mathcal{G}) = (\mathfrak{G}, \mathcal{X}) := \left((\mathbb{Z}/N\mathbb{Z})^\times, \{P \in \mathcal{G} : \mathcal{G} = \langle P \rangle\}\right)$$

Action: $(\mathfrak{a}, P) \mapsto [\mathfrak{a}]P$.

Now if N is prime (or almost), then

- $\textsc{Vec}(\mathfrak{G}, \mathcal{X}) \iff \text{DLP}(\mathcal{G})$
- $\textsc{Par}(\mathfrak{G}, \mathcal{X}) \iff \text{CDHP}(\mathcal{G})$

How hard are hard homogeneous spaces?

Obviously, if we can solve VECs

$$(P, Q = \mathfrak{x} \cdot P) \longmapsto \mathfrak{x}\,,$$

then we can solve PARs

$$(P, A = \mathfrak{a} \cdot P, B = \mathfrak{b} \cdot P) \longmapsto S = \mathfrak{a}\mathfrak{b} \cdot P\,.$$

Let's focus on VEC for a moment.

We can solve any DLP classically in time $O(\sqrt{N})$
using Pollard's ρ or Shanks' Baby-step giant-step.

We can solve VEC in time $O(\sqrt{N})$ using the same algorithms!

Generic DLP: Shanks’ BSGS in $\mathfrak{G}$

Algorithm 5: Baby-step giant-step in $\mathfrak{G}$

Input: $\mathfrak{g}$ and $\mathfrak{h}$ in $\mathfrak{G}$

Output: x such that $\mathfrak{h} = \mathfrak{g}^x$

```
β ← ⌈√#𝔊⌉
(𝔰_i) ← (𝔤^i : 1 ≤ i ≤ β)
Sort/hash ((𝔰_i, i))_{i=1}^{β}
𝔱 ← 𝔥
for j in (1, ..., β) do
    if 𝔱 = 𝔰_i for some i then
        return i − jβ
    𝔱 ← 𝔤^β 𝔱
return ⊥                                  // Only if 𝔥 ∉ ⟨𝔤⟩
```

Generic vectorization: Shanks' BSGS in $(\mathfrak{G}, \mathcal{X})$

Algorithm 6: Baby-step giant-step in $(\mathfrak{G}, \mathcal{X})$

Input: P and Q in $\mathcal{X}$, and a generator $\mathfrak{g}$ for $\mathfrak{G}$

Output: x such that $Q = \mathfrak{g}^x \cdot P$

$\beta \leftarrow \lceil \sqrt{\#\mathfrak{G}} \rceil$

$(P_i) \leftarrow (\mathfrak{g}^i \cdot P : 1 \leq i \leq \beta)$

Sort/hash $((P_i, i))_{i=1}^{\beta}$

$T \leftarrow Q$

for j *in* $(1, \ldots, \beta)$ **do**

    if $T = P_i$ *for some* i **then**

        return $i - j\beta$

    $T \leftarrow \mathfrak{g}^{\beta} \cdot T$

return $\perp$ `// Only if` $Q \notin \langle \mathfrak{e} \rangle \cdot P$

Why is this post-quantum?

Shor's algorithm solves DLP in polynomial time, but **not** VEC.

VEC is an instance of the abelian **hidden shift problem**.
Solve using (variants of) Kuperberg's algorithm in quantum **subexponential** time $L_N(1/2)$.

$\implies$ **upper bound for quantum VEC hardness** is $L_N(1/2)$.

$\implies$ **upper bound for quantum PAR hardness** is $L_N(1/2)$.

In a sense, BSGS and Pollard ρ are actually **PHS algorithms** (with $\mathfrak{G}$ acting on itself), not group algorithms!

Quantum equivalence of Vec and Par

Galbraith–Panny–S.–Vercauteren (2019): Unconditional **quantum** polynomial equivalence Par $\iff$ Vec.

Vec $\implies$ Par: obvious. Par $\implies$ Vec: quantum Par circuit $(P, \mathfrak{a} \cdot P, \mathfrak{b} \cdot P) \mapsto \mathfrak{ab} \cdot P$ gives $\mathcal{X}$ an implicit group structure.

1. We can compute a basis $\{\mathfrak{g}_1, \ldots, \mathfrak{g}_r\}$ for $\mathfrak{G}$ using Kitaev/Shor (if not already known)
2. The map $\mu : (x_1, \ldots, x_r, y) \mapsto \left(\prod_i \mathfrak{g}_i^{x_i}\right) \cdot \mathfrak{a}^y \cdot P$ is a homomorphism $(\mathbb{Z}^r \times \mathbb{Z}) \to \mathcal{X}$ (implicit group).
3. Evaluate $(y, \mathfrak{a} \cdot P) \mapsto \mathfrak{a}^y \cdot P$, hence μ, using $\Theta(\log n)$ Pars
4. Computing $\ker \mu = \{(x_1, \ldots, x_r, y) : \mathfrak{g}_1^{x_1} \cdots \mathfrak{g}_r^{x_r} \mathfrak{a}^y = 1_{\mathfrak{G}}\}$ is a hidden subgroup problem (Shor again);
5. Any $(a_1, \ldots, a_r, 1)$ in $\ker \mu$ gives a representation $\mathfrak{a} = \prod_i \mathfrak{g}_i^{a_i}$.

Classical limits of the analogy

Curiously, in the **classical** setting we *don't* have $\textsc{Par} \implies \textsc{Vec}$.

Compare with classical CDHP $\implies$ DLP, where we have a standard **black-box field** approach:

1. Reduce to prime order case (Pohlig–Hellman algorithm);
2. View $\mathfrak{G}$ as a representation of $\mathbb{F}_p$ via $\mathfrak{G} \ni \mathfrak{g}^a \leftrightarrow a \in \mathbb{F}_p$;
   - for $+$, use group operation $(\mathfrak{g}^a, \mathfrak{g}^b) \mapsto \mathfrak{g}^a \mathfrak{g}^b = \mathfrak{g}^{a+b}$
   - for $\times$, use $\mathfrak{G}$-DH oracle $(\mathfrak{g}, \mathfrak{g}^a, \mathfrak{g}^b) \mapsto \mathfrak{g}^{ab}$
3. den Boer, Maurer, Wolf: conditional polynomial reduction.

Does not work for $\textsc{Par} \implies \textsc{Vec}$ because $(P, \mathfrak{a} \cdot P, \mathfrak{b} \cdot P) \mapsto \mathfrak{ab} \cdot P$ oracle yields a group structure on $\mathcal{X}$, not a field structure.

Classical limits: Pohlig–Hellman

The **Pohlig–Hellman** algorithm exploits subgroups of $\mathfrak{G}$ to solve DLP instances in time $\widetilde{O}(\sqrt{\text{largest prime factor of } \#\mathfrak{G}})$.

Simplest case: $\#\mathfrak{G} = \prod_i \ell_i$, with the ℓ_i prime.
To find x such that $\mathfrak{h} = \mathfrak{g}^x$, for each i we

1. compute $\mathfrak{h}_i \leftarrow \mathfrak{h}^{m_i}$ and $\mathfrak{g}_i \leftarrow \mathfrak{g}^{m_i}$, where $m_i = \#\mathfrak{G}/\ell_i$;
2. compute x_i such that $\mathfrak{h}_i = \mathfrak{g}_i^{x_i}$ (DLP in order-ℓ_i subgroup)

We then recover x from the (x_i, ℓ_i) using the CRT.

Problem: the **HHS analogue of Step 1 is supposedly hard**!
(Computing $Q_i = \mathfrak{g}^i \cdot P$ where $Q = \mathfrak{g} \cdot P$ is an instance of PAR.)

No Pohlig–Hellman

Funny: We don't know how to use the structure of $\mathfrak{G}$ to accelerate algorithms for VEC *or* PAR *in* $(\mathfrak{G}, \mathcal{X})$.

Surprise: classical acceleration **shouldn't exist** in general.

Why?

- Choose p from a family of primes such that the largest prime factor of $p-1$ is in $o(p)$.
- Now take a black-box group $\mathcal{G}$ of order p.
- **Shoup's theorem**: DLP$(\mathcal{G})$ is in $\Theta(\sqrt{p})$.
- The Group-DH$\to$HHS-DH "hack" above yields a HHS $(\mathfrak{G}, \mathcal{X}) = \mathrm{Exp}(\mathcal{G}) = ((\mathbb{Z}/p\mathbb{Z})^\times, \mathcal{G} \setminus \{0\})$.
- Now $\#\mathfrak{G} = p - 1$, whose prime factors are in $o(p)$, so classical subgroup DLPs and VECs are in $o(\sqrt{p})$; a HHS Pohlig–Hellman analogue would **contradict Shoup**.

Isogeny-based key exchange:
A concrete HHS

Couveignes' isogeny HHS

Couveignes suggested a **concrete example** of an HHS, based on isogeny classes of elliptic curves.

Comparison with DLP-based elliptic curve crypto:

	Pre-quantum *Conventional ECC*	**Post-quantum** *Isogeny HHS*
Universe	One elliptic curve $\mathcal{E}$	One isogeny class $\mathcal{X}$
Elements	Points P and Q in $\mathcal{E}$	Curves $\mathcal{E}$ and $\mathcal{F}$ in $\mathcal{X}$
Relations	DLP: $Q = [x]P$	Isogeny: $\phi : \mathcal{E} \to \mathcal{F}$

Endomorphism rings of elliptic curves

An **isogeny** is just a nonzero homomorphism of elliptic curves.
Geometrically, isogenies = nonconstant algebraic mappings.

Existence of isogenies between curves is an **equivalence relation**, so we can talk about **isogeny classes** of curves.

An **endomorphism** is a homomorphism from a curve to itself.

The endomorphisms of a given curve form a **ring**.

Isogeny classes decompose into subclasses of curves with isomorphic endomorphism rings.

Couveignes' HHS: Class groups acting on isogeny classes

A Well-understood PHS from **complex multiplication** theory.

The group: $\mathfrak{G} = \mathrm{Cl}(O_K)$, the group of ideal classes of a quadratic imaginary field K

The space: $\mathcal{X}$ = the set of ($\mathbb{F}_q$-isomorphism classes of) elliptic curves $\mathcal{E}/\mathbb{F}_q$ with $\mathrm{End}(\mathcal{E}) \cong O_K$.

The action: Ideals in O_K correspond to **isogenies**, which take us from one curve to another.

We have $\#\mathfrak{G} = \#\mathcal{X} \sim \sqrt{|\Delta|}$, where $\Delta = \mathrm{disc}(O_K) \sim q$.

Why is this a HHS? When $\#\mathfrak{G} \sim \sqrt{q}$,

- The best known classical solution to Vec is in $O(q^{1/4})$.
- The best known quantum solution to Vec is in $L_q(1/2)$.

HHS-DH: the action

The action of an ideal (class) $\mathfrak{a} \subset O_K$ on a curve (class) $\mathcal{E} \in \mathcal{X}$:

Suppose $\mathfrak{a}$ is an integral ideal.

1. We can identify $\mathrm{End}(\mathcal{E})$ with O_K, so $\mathfrak{a} \subset \mathrm{End}(\mathcal{E})$.
2. Then $\mathcal{E}$ has a subgroup $\mathcal{E}[\mathfrak{a}] = \{P \in \mathcal{E} : \psi(P) = 0 \quad \forall \psi \in \mathfrak{a}\}$
3. We can compute a *quotient isogeny* $\phi : \mathcal{E} \to \mathcal{E}/\mathcal{E}[\mathfrak{a}]$. We let $\mathfrak{a} \cdot E$ be the quotient curve $\mathcal{E}/\mathcal{E}[\mathfrak{a}]$;

This is all well-defined up to isomorphism.

$\mathfrak{a} = (\phi)$ principal $\implies \phi \in \mathrm{End}(\mathcal{E})$, so $\mathfrak{a} \cdot \mathcal{E} = \mathcal{E}$.
So: action extends to fractional ideals, factors through $\mathrm{Cl}(O_K)$.

HHS-DH: Computing the action

We need to be able to compute this action efficiently for random-looking $\mathfrak{a}$ in $\mathrm{Cl}(O_K)$.

Bad news: Computing the isogenous $\mathfrak{a} \cdot E$ **directly**, by computing the quotient isogeny, is **exponential** in $N(\mathfrak{a})$.

Couveignes suggested using LLL to compute an equivalent $\prod_i \mathfrak{l}_i^{e_i} \sim \mathfrak{a}$ with each $N(\mathfrak{l}_i)$ small, then act with the $\mathfrak{l}_i$ in serial.

Each small ideal $\mathfrak{l}_i$ acts as an isogeny of degree $\ell_i = \mathrm{Norm}(\mathfrak{l}_i)$, called an ℓ_i-isogeny.

What happened?

1997: Couveignes submitted to Crypto; rejected.

Later published in French, in an obscure special SMF issue.

QUELQUES MATHÉMATIQUES DE LA CRYPTOLOGIE À CLÉS PUBLIQUES

par

Jean-Marc Couveignes

Résumé. — Cette note présente quelques développements mathématiques plus ou moins récents de la cryptologie à clés publiques.

Abstract **(A few mathematical tools for public key cryptology)**
I present examples of mathematical objects that are of interest for public key cryptography.

HHS-DH: Rostovtsev–Stolbunov

1997: Couveignes submitted to Crypto; rejected.
Later published in French, in an obscure special SMF issue.
$\cong$ Unknown / Forgotten.

2006: Rostovtsev and Stolbunov independently rediscover isogeny-based key exchange.

The (minor) ***essential difference***:

Couveignes samples a secret $\mathfrak{a}$ in $\mathrm{Cl}(\mathcal{O}_K)$ and smooths to $\prod_i \mathfrak{l}_i^{e_i}$;

Rostovtsev–Stolbunov sample a smooth product $\prod_i \mathfrak{l}_i^{e_i}$ directly, and hope this distribution is very close to uniform on $\mathrm{Cl}(\mathcal{O}_K)$.

Moving to isogeny graphs

Rostovtsev and Stolbunov sample exponent vectors $(e_1, \ldots, e_r)$ as secret keys, corresponding to ideal products $\prod_i \mathfrak{l}_i^{e_i}$.

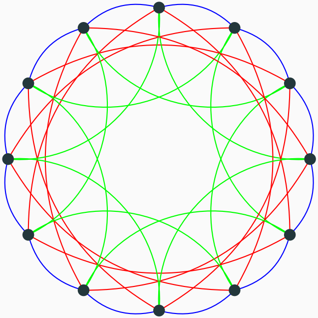
- Act e_1 times by $\mathfrak{l}_1$, then
- act e_2 times by $\mathfrak{l}_2$, then
- …

Actions expressed as **random walks** in **isogeny graphs**.

For each prime ℓ, restrict to ℓ-**isogeny graphs**:

- vertices = $\mathcal{X}$,
- edges = isogenies of degree ℓ
  (corresponding to actions of ideals $\mathfrak{l}$ of norm ℓ).

Isogeny graphs



1. A walk of length e_1 in the ℓ_1-isogeny graph, then
2. A walk of length e_2 in the ℓ_2-isogeny graph, then
3. A walk of length e_3 in the ℓ_3-isogeny graph,
4. More walks ...

From Rostovtsev–Stolbunov to SIDH and back

Plain Rostovtsev–Stolbunov: **totally impractical** key exchange.

This prompted Jao & De Feo's **SIDH** (Supersingular Isogeny DH)

- Uses only tiny-degree isogenies (fast)
- between curves with quaternionic endomorphism rings
- forming isogeny graphs that are expanders

SIDH is cool, but it has some **disadvantages**:

1. **Static** key exchange (long term keys) is **unsafe**
2. The API doesn't match Diffie–Hellman
   (e.g. Alice and Bob's public keys don't have the same type).

Our idea: go back and **improve Rostovtsev–Stolbunov**.

Towards practical isogeny key exchange

De Feo–Kieffer–S. (Asiacrypt 2018):
algorithmic improvements and **security proofs**.

- Use **ordinary** curves, following Couveignes and Stolbunov.
- Faster isogeny steps when $\mathcal{E}[\mathfrak{l}_i]$ has rational points.
- **Problem**: no efficient algorithm to construct ordinary $\mathcal{E}$ with a point of degree ℓ for hundreds of very small ℓ.

Towards practical isogeny key exchange

Castryck et al. (Asiacrypt 2018): CSIDH.

- Solves the parametrization problem by using **supersingular** curves over $\mathbb{F}_p$.
- Supersingular curves are easy to construct.
  Order $p+1$, so choose p s.t. $\ell \mid (p+1)$ for lots of small ℓ.

$\implies$ **Practical isogeny-based Diffie–Hellman**.

Keysize $= \log_2 p$	Classical queries	Quantum queries*
512	128	62
1024	256	94
1792	448	129

**Claimed by CSIDH authors. Precise quantum query counts and costs are the subject of current research and debate.*

Conclusions

- In CSIDH, isogeny-based crypto now has a **practical postquantum drop-in replacement** for Diffie–Hellman.
  Can also be used for OT; no practical signatures though.
- Couveignes' **Hard Homogeneous Spaces** framework helps to model postquantum DH protocols on an abstract level, without understanding the mechanics of isogenies
- Pre- and post-quantum DH have the same "API",
  but **HHS-DH does not respect Group-DH intuitition**.

The Maurer reduction: how does it work?

We want to **solve a DLP** instance $\mathfrak{h} = \mathfrak{g}^x$ in $\mathfrak{G}$ of prime order p, **given a DH oracle** for $\mathfrak{G}$ (so we can compute $\mathfrak{g}^{F(x)}$, $\forall$ poly F):

1. Find an $\mathcal{E}/\mathbb{F}_p$ s.t. $\mathcal{E}(\mathbb{F}_p)$ has **polynomially smooth order**[2] and compute a generator (x_0, y_0) for $\mathcal{E}(\mathbb{F}_p)$.
   Pohlig–Hellman: solve DLPs in $\mathcal{E}(\mathbb{F}_p)$ in polynomial time.
2. Use Tonelli–Shanks to compute a $\mathfrak{g}^y$ s.t. $\mathfrak{g}^{y^2} = \mathfrak{g}^{x^3+ax+b}$.
   If this fails: replace $\mathfrak{h} = \mathfrak{g}^x$ with $\mathfrak{h}\mathfrak{g}^\delta = \mathfrak{g}^{x+\delta}$ and try again...
   Now $(\mathfrak{g}^x, \mathfrak{g}^y)$ is a point in $\mathcal{E}(\mathfrak{G})$; we still don't know x or y.
3. Solve the DLP instance $(\mathfrak{g}^x, \mathfrak{g}^y) = [e](\mathfrak{g}^{x_0}, \mathfrak{g}^{y_0})$ in $\mathcal{E}(\mathfrak{G})$ for e.
4. Compute $(x, y) = [e](x_0, y_0)$ in $\mathcal{E}(\mathbb{F}_p)$ and return x.

[2] This is the tricky part! *Seems to work in practice for cryptographically useful p, even in not in theory for arbitrary p.*