Basic public-key constructions with elliptic curves

Benjamin Smith

Team **GRACE**

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

Summer school on real-world crypto and privacy

Sibenik, Croatia, June 6 2016

0: Public-Key Cryptography

Public-key cryptography

Universal introduction: **Alice** and **Bob**. *Ingredients:*

1. Picture of Alice in Wonderland (or Alice Cooper)
2. Picture of Bob Dylan, or Spongebob Squarepants.
3. "Alice wants to send a message to Bob"
4. "Alice uses Bob's public key to encrypt the message, Bob uses his private key to decrypt it"
5. *Public keys state instances of hard computational problems, private keys give the solutions.*
6. Hard problems: Factoring, RSA Problem, Subset Sum, Discrete Logarithm Problem, Closest Vector Problem, Decoding Random Codes, Learning With Errors, ...

Stop!

PKC is a *huge* field of research,
overflowing with problems, protocols, and primitives.
It's way too huge to tour in 90 minutes.

We'll talk about two constructions that really matter in the "real world": key exchange and signatures.

We'll restrict ourselves to *one* computational hard problem: the Discrete Logarithm Problem (DLP).

(This is still more than enough trouble for 90 minutes.)

1: Discrete Logarithms

Let $\mathcal{G} = \langle P \rangle$ be a (fixed) cyclic group of order N, with group law $\oplus$, identity 0, inverse $\ominus$.

Exponentiation ("scalar multiplication") is

$$[m]P : P \longmapsto \underbrace{P \oplus \cdots \oplus P}_{m \text{ times}} \quad \text{for } m \in \mathbb{Z} .$$

i.e.: $\mathcal{G} = \langle P \rangle = \{0, P, [2]P, [3]P, \ldots, [N-1]P\}$.

For the moment, $\mathcal{G}$ is a *black-box group*:
— *Elements* are identified with $(\log_2 N)$-bit labels
— *Group law* $\oplus$ is an oracle that takes the labels of two elements and returns the label of their sum.

Polynomial time means polynomial in $\log_2 N$.

Exponentiation is easy

We can compute *any* scalar multiple in $O(\log N)$ $\mathcal{G}$-ops.

Algorithm 1 Classic double-and-add scalar multiplication

function NaiveMultiplication($m = \sum_{i=0}^{\beta-1} m_i 2^i, P$)
    $R \leftarrow \mathcal{O}_{\mathcal{E}}$
    for $i := \beta - 1$ down to 0 **do** ▷ invariant: $R = [\lfloor m/2^i \rfloor]P$
        $R \leftarrow [2]R$
        if $m_i = 1$ **then** ▷ Danger! *Branching leaks m_i to side channels*
            $R \leftarrow R \oplus P$
        end if
    end for
    return R ▷ $R = [m]P$
end function

The Discrete Logarithm Problem

Given P and $[x]P$, find x.

In any $\mathcal{G}$, we can *always* solve the DLP in time $O(\sqrt{N})$.

- Time-memory tradeoff:
  Shanks' *Baby-step giant-step*
- Low-memory pseudo-random walks:
  Pollard's ρ and *Kangaroo* (λ)...

Shoup: *if* $\mathcal{G}$ *is a black box group* and N is prime, then the DLP is in $\Omega(\sqrt{N})$.

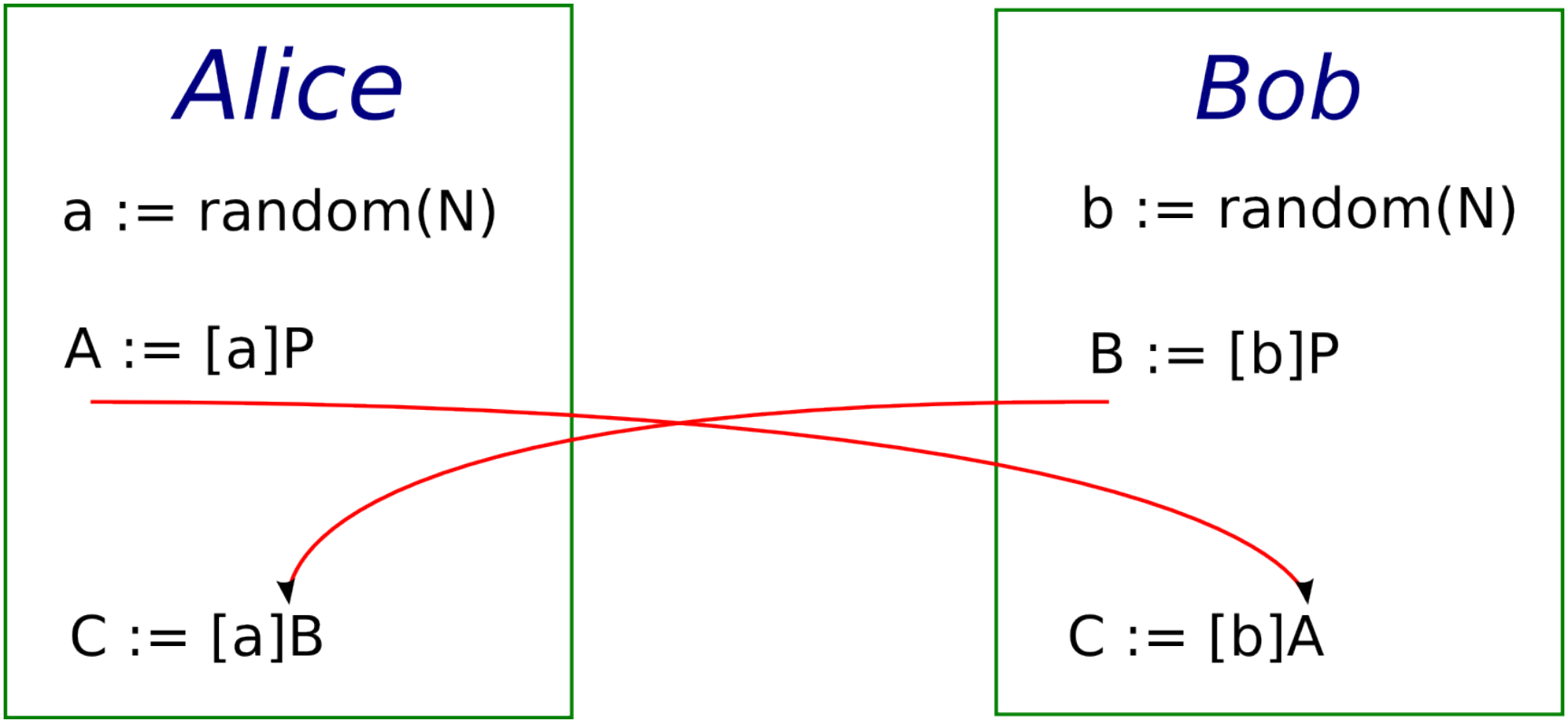
The Silver–Pohlig–Hellman reduction

If we know $N = \prod_{i=1}^{n} p_i^{e_i}$ for primes p_i and exponents e_i
then we can solve any DLP in $\mathcal{G}$
using $O(\sum_{i=1}^{n} e_i(\log N + \sqrt{p_i}))$ $\mathcal{G}$-operations.

Key point: The DLP in $\mathcal{G}$ is dominated by
the DLP in the largest prime-order subgroup of $\mathcal{G}$.

For t-bit security in a DLP-based cryptosystem,
we need a generic $\mathcal{G}$ with prime order $N \sim 2^{2t}$.

Diffie–Hellman Key Exchange



Now **Alice** and **Bob** have a shared secret $C = [ab]P$.

There are so many things wrong with that picture...

From top to bottom:

1. What's this "random"? (Ask the NSA.)
2. $A \leftarrow [a]P$, $B \leftarrow [b]P$: Assumes efficient *side-channel-safe* scalar multiplication. Is this reasonable? (Yes: see later talks.)
3. Sending A, B: Trivial man in the middle.
   We're going to need some kind of authentication.
4. Security of shared secret $[ab]P$ is based on the *wrong problem.*
   Diffie–Hellman problem (given $P, [a]P, [b]P$, compute $[ab]P$) instead of DLP (given $P, [x]P$, compute x).
   Reductions: DLP $\Longrightarrow$ CDHP obvious, CDHP $\Longrightarrow$ DHP tricky.
5. But first: even if we solve these theoretical problems,
   we don't have black-box groups in practice...
   What about algorithms and security for concrete groups?

2: Abstract ⟶ Concrete

In an ideal world...

In practice we compute with concrete groups,
not abstract black-box groups.

To maximise cryptographic efficiency
(security level / key length ratio)
we need concrete groups that act like black box groups:

- Prime (or almost-prime) order N
- Elements stored in $\sim \log_2 N$ bits each
- Operations computed in $\widetilde{O}(\log_2^c N)$ bit-ops, c small
- Best known DLP solutions in $O(\sqrt{N})$ $\mathcal{G}$-ops

Concrete groups to model black box groups

- Prime (or almost-prime) order N
- Elements stored in $\sim \log_2 N$ bits each
- Operations computed in $\widetilde{O}(\log_2^c N)$ bit-ops, c small
- Best known DLP solutions in $O(\sqrt{N})$ $\mathcal{G}$-ops

Concretely: want ≥ 128 bits of security,
i.e. attackers need $\geq 2^{128}$ bit operations.

$\Longrightarrow$ prime order $N \sim 2^{256}$; ideally, elements in 256 bits.

Algebraic groups

Natural candidates: *algebraic groups*.

Elements = tuples of (finite) field elements (coordinates);

Operations = tuples of polynomials in the coordinates.

We work over $\mathbb{F}_q$, where q is a power of p

Normally, $p \neq 2, 3$.

...in practice: $q = p, p^2$, or 2^n with n prime.

The main unit of measure is $\log q$.

Additive groups of finite fields

Naïve attempt at a concrete cryptographic $\mathcal{G}$:
The additive group $\mathbb{G}_a(\mathbb{F}_q) = (\mathbb{F}_q, +)$.

How do subgroups of $\mathbb{G}_a(\mathbb{F}_q)$ measure up against black-box groups?

Prime order subgroups have order p, where $q = p^n$. *simple!*

Storage $\log_2 p$ bits *ideal!*

Group operations addition in $\mathbb{F}_q$: $O(\log_2 q)$ bit-ops *great!*

What about the DLP ? *Division in $\mathbb{F}_q$.*
Euclidean algorithm $\Longrightarrow$ fast polynomial-time solution.

Multiplicative groups of finite fields

Second attempt at a concrete cryptographic $\mathcal{G}$:
prime-order subgroups of $\mathbb{G}_m(\mathbb{F}_q)$.

Historical choice of group
for Diffie–Hellman (1970s) and signatures (1980s).

How do subgroups of $\mathbb{G}_m(\mathbb{F}_q)$ measure up against black-box groups?

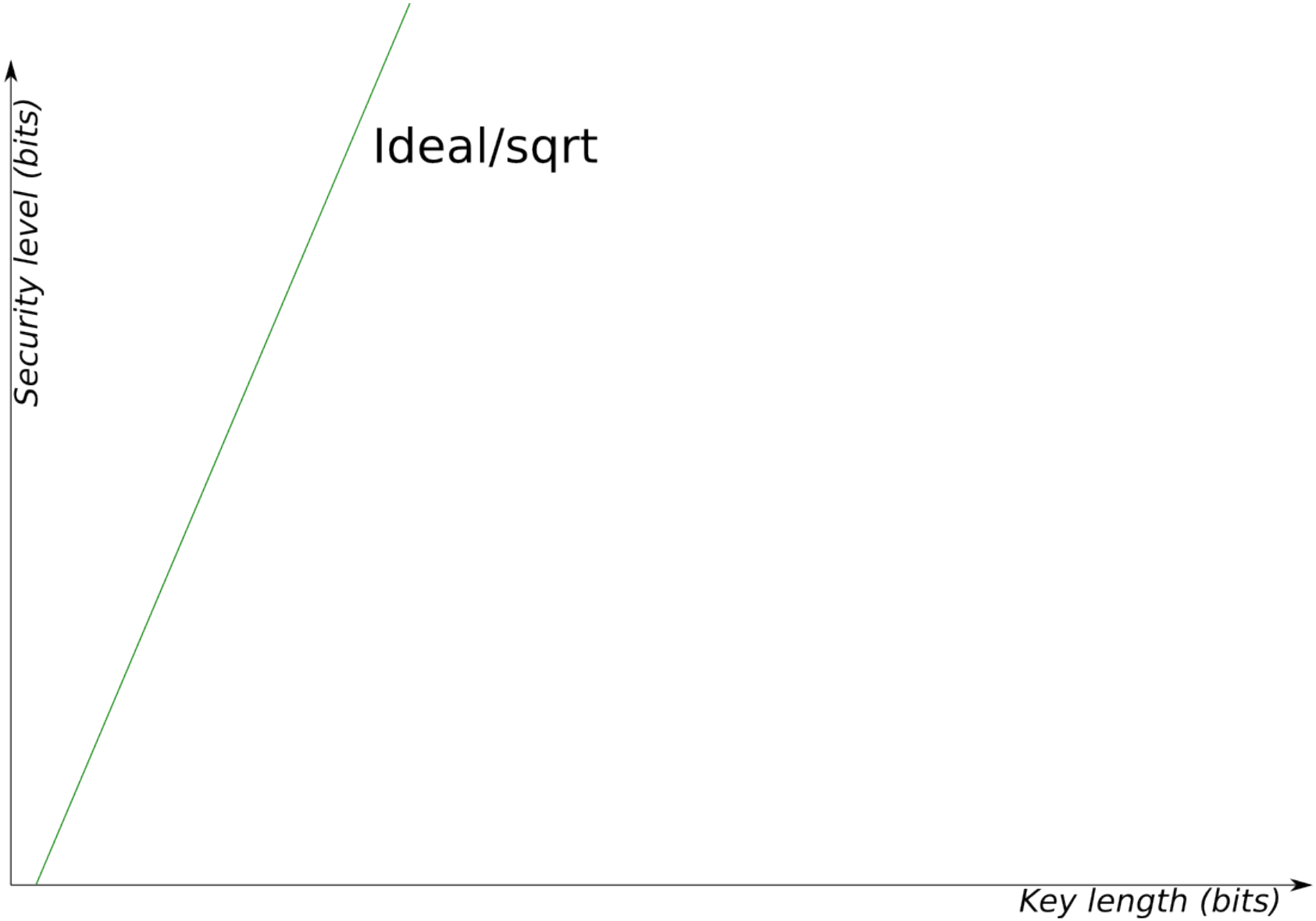
Prime order $N \mid (q-1)$: need to choose q carefully

Storage $\geq \log_2 N + 1$ bits (best case $q = 2N+1$, N prime)

Group operations $\sim \log_2^c N$ bit-ops $(1 < c \leq 2)$

What about the DLP ? *Good news for people who like bad news...*

Discrete Logarithm hardness in finite fields



Discrete Logarithms in finite fields

This improvement isn't just asymptotic/theoretical:

Finite Field Discrete Logarithm records have been repeatedly and spectacularly broken since 2013.

The large characteristic case is still in $L(1/3)$, comparable with RSA, but
Finite Field Discrete Logs are on the way out
for cryptographic primitives.

(This is a big problem for pairing-based cryptography.)

3: Elliptic Curves

Elliptic curves

"Short Weierstrass" models: nonsingular plane cubics

$$\mathcal{E} : y^2 = x^3 + ax + b$$

where a and b are parameters in $\mathbb{F}_q$

satisfying $4a^3 + 27b^2 \neq 0$ (nonsingularity)

Natural *involution* $\ominus : (x, y) \mapsto (x, -y)$ (negation)

We write $\mathcal{E}(\mathbb{F}_q)$ for the set of *points* on $\mathcal{E}$:

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

where $\mathcal{O}_\mathcal{E}$ is a unique *point at infinity* (zero element)

Store each (α, β) as (α, "sign" of β) using $\log_2 q + 1$ bits

Projective space

Consider the projective plane $\mathbb{P}^2$.

Two-dimensional, with three coordinates:

$$\mathbb{P}^2(\mathbb{F}_q) = \left\{(\alpha : \beta : \gamma) \in \mathbb{F}_q^3 \setminus \{(0,0,0)\}\right\} / \sim$$

where $\sim$ is the equivalence relation defined by

$$(\alpha : \beta : \gamma) \sim (\lambda\alpha : \lambda\beta : \lambda\gamma) \text{ for all } \lambda \neq 0 \in \mathbb{F}_q \ .$$

Projective elliptic curves

Putting $(x, y) = (X/Z, Y/Z)$ gives a projective model

$$\mathcal{E} : Y^2Z = X^3 + aXZ^2 + bZ^3 \subseteq \mathbb{P}^2 .$$

Affine points (α, β) become projective points $(\alpha : \beta : 1)$

The point at infinity $\mathcal{O}_\mathcal{E}$ is $(0 : 1 : 0)$
(it is the unique point with $Z = 0$)

Every other projective point $(X : Y : Z)$ on $\mathcal{E}$
corresponds to a unique affine point $(x, y) = (X/Z, Y/Z)$

This is not the only projective closure/model of $\mathcal{E}$.

The group law

Every line intersects $\mathcal{E}$ in exactly three (multiple?) points.

If two of the points are in $\mathcal{E}(\mathbb{F}_q)$, then so is the third.

The group law on $\mathcal{E}$ is then:

$$P, Q, R \text{ collinear} \iff P \oplus Q \oplus R = 0$$

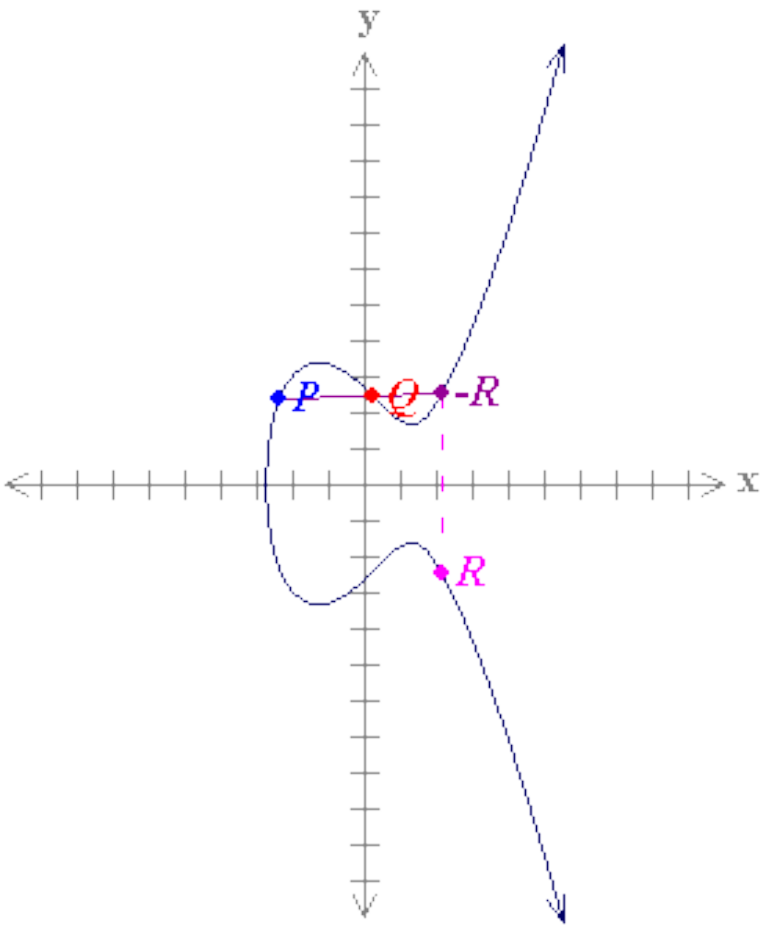
Identity element: $0 = \mathcal{O}_{\mathcal{E}} = (0 : 1 : 0)$

Each "vertical" line $x = \alpha$ intersects $\mathcal{E} : y^2 = x^3 + ax + b$ in $\{(\alpha : \beta : 1), (\alpha : -\beta : 1), \mathcal{O}_{\mathcal{E}}\}$ where $\beta^2 = \alpha^3 + a\alpha + b$

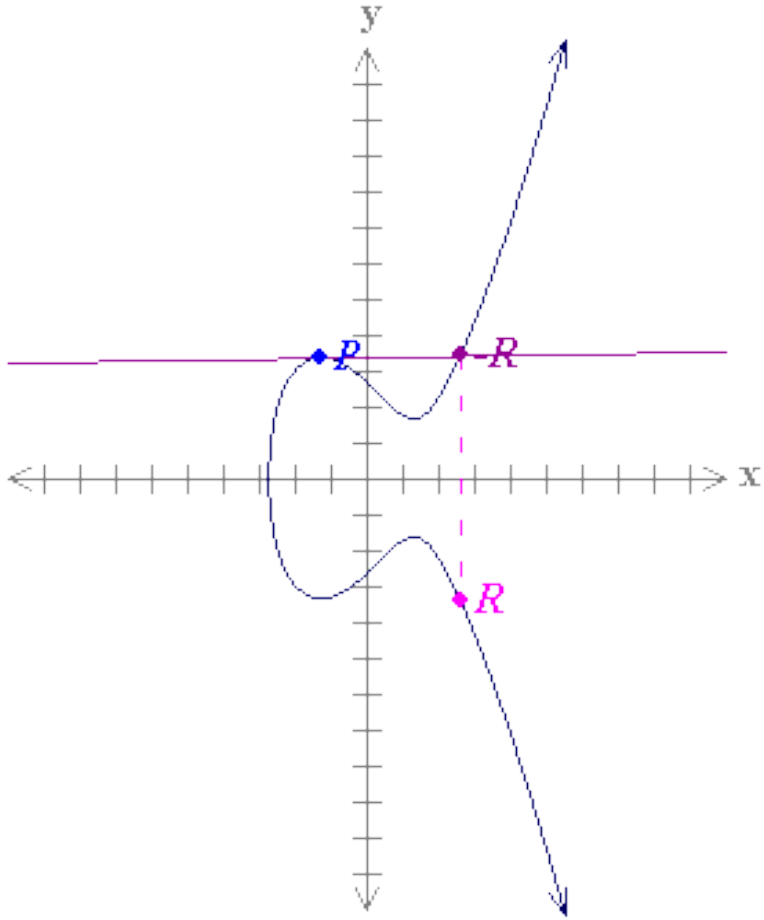
$\implies \ominus : (x : y : 1) \mapsto (x : -y : 1)$ is the negation map

The group law

Adding: $R = P \oplus Q$



...and doubling: $R = [2]P$



If you apply this law to *singular* cubics, you get $\mathbb{G}_m(\mathbb{F}_q)$ and $\mathbb{G}_a(\mathbb{F}_q)$.

Computing $P \oplus Q$ on $\mathcal{E} : y^2 = x^3 + ax + b$

- $P = \mathcal{O}_\mathcal{E}$ or $Q = \mathcal{O}_\mathcal{E}$? Nothing to be done.
- If $P = \ominus Q$, then $P \oplus Q = \mathcal{O}_\mathcal{E}$

Otherwise: *compute $P \oplus Q$ using low-degree polynomial expressions*

$$x(P \oplus Q) = \lambda^2 - x(P) - x(Q),$$
$$y(P \oplus Q) = -\lambda x(P \oplus Q) - \nu,$$

where

$$\lambda := \begin{cases} (y(P) - y(Q))/(x(P) - x(Q)) & \text{if } x(P) \neq x(Q), \\ (3x(P)^2 + a)/(2y(P)) & \text{if } P = Q \end{cases}$$

$$\nu := \begin{cases} (x(P)y(Q) - x(Q)y(P))/(x(P) - x(Q)) & \text{if } x(P) \neq x(Q), \\ -y(P)/2 + (2ax(P) + 3b)/(2y(P)) & \text{if } P = Q. \end{cases}$$

Complete group laws for odd-order curves

Given a short Weierstrass model $\mathcal{E}/\mathbb{F}_p : Y^2Z = X^3 + aXZ^2 + bZ^3$ with $2 \nmid \#\mathcal{E}(\mathbb{F}_p)$, the following group law works *for all points* in $\mathcal{E}(\mathbb{F}_p)$:

$$(X_3 : Y_3 : Z_3) = (X_1 : Y_1 : Z_1) \oplus (X_2 : Y_2 : Z_2) \text{ where}$$

$$\begin{aligned}
X_3 &= (X_1Y_2 + X_2Y_1)(Y_1Y_2 - a(X_1Z_2 + X_2Z_1) - 3bZ_1Z_2) \\
&\quad - (Y_1Z_2 + Y_2Z_1)(aX_1X_2 + 3b(X_1Z_2 + X_2Z_1) - a^2Z_1Z_2) \\
Y_3 &= (3X_1X_2 + aZ_1Z_2)(aX_1X_2 + 3b(X_1Z_2 + X_2Z_1) - a^2Z_1Z_2) \\
&\quad + (Y_1Y_2 + a(X_1Z_2 + X_2Z_1) + 3bZ_1Z_2)(Y_1Y_2 - a(X_1Z_2 + X_2Z_1) - 3bZ_1Z_2) \\
Z_3 &= (Y_1Z_2 + Y_2Z_1)(Y_1Y_2 + a(X_1Z_2 + X_2Z_1) + 3bZ_1Z_2) \\
&\quad + (X_1Y_2 + X_2Y_1)(3X_1X_2 + aZ_1Z_2)
\end{aligned}$$

Renes–Costello–Batina, Eurocrypt 2016:
This can be computed in $12\mathbf{M} + 3\mathbf{m_a} + 2\mathbf{m_{3b}} + 23\mathbf{a}$.

Edwards models

Go much faster using a *twisted Edwards model* for $\mathcal{E}$:

$$\mathcal{E}/\mathbb{F}_p : au^2 + v^2 = 1 + du^2v^2.$$

The group law on $\mathcal{E}(\mathbb{F}_p)$ is completely described by

$$(u_1, v_1) \oplus (u_2, v_2) = \left(\frac{u_1v_2 + v_1u_2}{1 + du_1u_2v_1v_2}, \frac{v_1v_2 - au_1u_2}{1 - du_1u_2v_1v_2} \right)$$

with $\ominus(u, v) = (-u, v)$ and $(0, 1)$ as the identity element.

In suitable projective coordinates
we get much faster, uniform group operations.
(see Bernstein–Birkner–Lange–Peters, Hisil–Wong–Carter–Dawson, Kohel, . . .)

Restriction: twisted Edwards models require $4 \mid \#\mathcal{E}(\mathbb{F}_p)$.

Group orders

We have $\#\mathbb{G}_a(\mathbb{F}_q) = q$ and $\#\mathbb{G}_m(\mathbb{F}_q) = q - 1$.

What about $\#\mathcal{E}(\mathbb{F}_q)$?

Hasse's theorem:

If $\mathcal{E} : y^2 = x^3 + ax + b$ is an elliptic curve over $\mathbb{F}_q$, then

$$\#\mathcal{E}(\mathbb{F}_q) = q + 1 - t \quad \text{where} \quad |t| \leq 2\sqrt{q} \ .$$

Deuring's theorem: every t in this interval occurs (except for some t divisible by p, when $q = p^n$ with $n > 1$)

Cryptographic sized q: $\#\mathcal{E}(\mathbb{F}_q) \sim q$.

Possible group structures

We have $\mathbb{G}_a(\mathbb{F}_q) \cong (\mathbb{Z}/p\mathbb{Z})^n$ for $q = p^n$ and $\mathbb{G}_m(\mathbb{F}_q) \cong \mathbb{Z}/(q-1)\mathbb{Z}$.

What is the group structure of $\mathcal{E}(\mathbb{F}_q)$?

The possible group structures for elliptic curves over finite fields are extremely limited.

Theorem: If $\mathcal{E}$ is defined over $\mathbb{F}_q$, then

$$\mathcal{E}(\mathbb{F}_q) \cong \mathbb{Z}/d_1\mathbb{Z} \times \mathbb{Z}/d_2\mathbb{Z}$$

where $d_2 \mid d_1$ and $d_2 \mid (q-1)$.

The ECDLP is believed to be hard

The best known Discrete Log solvers
for generic elliptic curves over $\mathbb{F}_p$ and $\mathbb{F}_{p^2}$
are *all* algorithms operating on "black box groups"

Apparent exponential "square-root" difficulty
in prime-order subgroups:

- currently, the ECDLP is *as hard as you can get*
- beats *subexponential* finite field DLP, RSA/factoring
- $\Longrightarrow$ better scaling, far more security per bit

Important: *This hardness is unproven,*
and nobody knows why it should/should not be true

Bad elliptic curves

What do we mean when we say that
the DLP in a "generic" prime-order elliptic curve is hard?

Some prime-order curves are weak:

- Curves over $\mathbb{F}_{p^n}$ where n has a moderate-sized factor
  (vulnerable to Weil descent attacks)
- Anomalous elliptic curves: where $\#\mathcal{E}(\mathbb{F}_q) = q$
  (can map DLP into $\mathbb{G}_a(\mathbb{F}_q)$)
- Pairing-friendly curves (including "supersingular" elliptic curves):
  where $N \mid q^k - 1$ for a small k *(can map DLP into $\mathbb{G}_m(\mathbb{F}_{q^k})$)*

These weak curves are easily identified,
and easily avoided.

Good elliptic curves

Conclusion:

Prime-order subgroups of elliptic curves are our best* concrete approximation of generic groups for PKC.

If you want t bits of security, use a (almost-) prime order $\mathcal{E}/\mathbb{F}_p$ with $\log_2 p \sim 2t$.

**...At least until we have large quantum computers*

Elliptic Curve vs $\mathbb{F}_p$/RSA parameters

Security level (bits)	Elliptic $\mathcal{E}(\mathbb{F}_p)$ ($\log_2 p$)	$\mathbb{G}_m(\mathbb{F}_p)$/RSA ($\log_2 p$)	keylength ratio
56	112	512	4.57
64	128	704	5.5
80	160	1024	6.4
96	192	1536	8.0
112	224	2048	9.14
128	256	3072	12.0
192	384	7680	20.0
256	512	15360	30.0

4: Towards Authenticity

An identification game

Suppose Alice has public-private key pair $(Q = [x]P, x)$.
"Alice" wants to prove her identity to Bob: ie, she possesses x.

Let's start with a 3-move game:

Commitment: Alice chooses a random r in $\mathbb{Z}/N\mathbb{Z}$,
computes the witness $R := [r]P$, and sends R to Bob.

Challenge: Bob chooses $e \in \{0, 1\}$ at random, and sends e to Alice.

Response: Alice sends $s := r - ex \pmod{N}$ to Bob.

Bob computes $[s]P \oplus [e]Q$; if this is R then he thinks Alice is genuine.

Only one bit of security: Alice can cheat if she guesses e in advance
(then she can send $R := [r]P \oplus [e]Q$ and $s := r$).

So Bob bets that Alice can't guess correctly t times in a row,
and they repeat the whole game t times over.

Schnorr identification

The Schnorr identification protocol saves space and time by running t of the previous games "in parallel".

Suppose Alice has public-private key pair $(Q = [x]P, x)$.
Alice wants to prove her identity (possession of x) to Bob.

Commitment: Alice chooses a random r in $\mathbb{Z}/N\mathbb{Z}$, computes the witness $R := [r]P$, and sends R to Bob.

Challenge: Bob chooses a random e from $[1..2^t)$, and sends e to Alice.

Response: Alice sends $s = r - ex \pmod N$ to Bob.

Verification: Bob accepts Alice's identity if $[s]P \oplus [e]Q = R$.

To cheat, Alice must guess e in advance (then she can send $R := [r]P \oplus [e]Q$ and $s := r$).

$1/2^t$ chance of guessing $\implies$ security level: t bits

From identification to signatures

The *Fiat–Shamir transform* converts this 3-move identification scheme into a signature scheme, by letting a hash function play the role of the "verifier"

(...sorry, Bob!)

Let $H : \{0,1\}^* \to [0..2^t)$
be a cryptographic hash function
(we won't need collision resistance, just preimage resistance)
and let $\mathcal{G}$ be a t-bit secure group:
ie, an (almost)-prime $\mathcal{E}(\mathbb{F}_p)$ where $\log_2 p \sim 2t$.

Schnorr Signatures: Key Generation

To generate public-private key pairs (Q, x):

Algorithm 2 Key generation for Schnorr signatures

1: **function** KeyGen
2: $x \leftarrow$ **random**$(\mathbb{Z}/N\mathbb{Z})$
3: $Q \leftarrow [x]P$ ▷ mult. public point by secret scalar
4: **return** $(Q, x) \in \mathcal{E}(\mathbb{F}_p) \times \mathbb{Z}/N\mathbb{Z}$.
5: **end function**

$Q \in \mathcal{G} \subseteq \mathcal{E}(\mathbb{F}_p)$ is public, $x \in \mathbb{Z}/N\mathbb{Z}$ is private.
Q and x each need $2t$ bits of storage.

Recovering x from $Q \implies$ solving the DLP in $\mathcal{G}$.

Signing a message

To sign a message m with the key pair (Q, x):

Algorithm 3 Schnorr signature signing operation

1: **function** $\textsc{Sign}(m \in \{0,1\}^*, x \in \mathbb{Z}/N\mathbb{Z})$
2: $\quad r \leftarrow$ **random**$(\mathbb{Z}/N\mathbb{Z})$
3: $\quad R \leftarrow [r]P$ ▷ mult. public point by secret scalar
4: $\quad e \leftarrow H(m||R)$
5: $\quad s \leftarrow r - ex \pmod N$ ▷ (so $[s]P \oplus [e]Q = R$)
6: $\quad$ **return** $(s, e) \in (\mathbb{Z}/N\mathbb{Z}) \times [0..2^t)$
7: **end function**

Signatures (s, e) require $3t$ bits of storage.

Verifying a signature

To verify a claimed signature (m, e) on a message m against a public key Q,

Algorithm 4 Schnorr signature verification

1: **function** VERIFY((s, e), m, Q)
2: $R' \leftarrow [s]P \oplus [e]Q$ ▷ mult. public points, scalars
3: $e' \leftarrow H(m||R')$
4: **return** $e' = e$
5: **end function**

DLP hardness + hardness of hash preimages gives t bits of authenticity, integrity, and non-repudiability.

Alice and Bob can now safely exchange keys.

5: Diffie–Hellman, the "Wrong Problem"

Relating the DLP and DHP

Suppose $\mathcal{G} = \langle P \rangle$ is generic/black-box of prime order N.

DHP: Given P, $[a]P$, and $[b]P$, compute $[ab]P$.
DLP: Given P and $[x]P$, compute x.

Obvious reduction: DLP $\implies$ DHP.

What about the other way? Maurer reduction DHP $\implies$ DLP.

View $\mathcal{G}$ as a finite field, $\mathbb{F}_{\mathcal{G}} \cong \mathbb{F}_N$, via $[a]P \in \mathbb{F}_{\mathcal{G}} \longleftrightarrow a \in \mathbb{F}_N$

addition: $[a+b]P = [a]P \oplus [b]P$

multiplication: $[ab]P = DH([a]P, [b]P)$ (Diffie–Hellman oracle)

inverses: $[a^{-1}]P = [a^{N-2}]P$ ($\leq 2 \log p$ calls to DH)

The Maurer reduction

1. Construct an $\mathcal{E} : Y^2 = X^3 + AX + B$ over $\mathbb{F}_N$ such that
   - $\mathcal{E}(\mathbb{F}_N) = \langle(\alpha, \beta)\rangle$ is cyclic
   - all prime divisors of $\#\mathcal{E}(\mathbb{F}_N)$ are less than some bound $\mathcal{B}$

   Key: we have $(x, y) = [k](\alpha, \beta)$ in $\mathcal{E}(\mathbb{F}_N)$
   if and only if $([x]P, [y]P) = [k]([\alpha]P, [\beta]P)$ in $\mathcal{E}(\mathbb{F}_\mathcal{G})$.
2. Now, to solve a DLP $Q = [x]P$ in $\mathcal{G}$;
   1. Compute $[x^3 + Ax + B]P = DH(DH(Q, Q), Q) \oplus [A]Q \oplus [B]P$
   2. Compute $R := [y]P = [\sqrt{x^3 + Ax + B}]P$ (implicit Tonelli–Shanks)
   3. Solve the DLP $(Q, R) = [k]([\alpha]P, [\beta]P)$ in $\mathcal{E}(\mathbb{F}_\mathcal{G})$ (Pohlig–Hellman)
   4. Compute x from $(x, y) = [k](\alpha, \beta)$ in $\mathcal{E}(\mathbb{F}_N)$

Complexity: $O(\sqrt{\mathcal{B}} \cdot \log^3 N)$ $\mathbb{F}_N$-ops and calls to the DH oracle.

Questionable theory, acceptable practice

The Maurer reduction doesn't work in theory.

Tricky part: finding an $\mathcal{E}/\mathbb{F}_N$ with
all prime factors of $\#\mathcal{E}(\mathbb{F}_N)$ polynomial in $\log N$.

No guarantee that such a curve order exists
in the Hasse interval $[N+1-2\sqrt{N}, N+1+2\sqrt{N}]$!

...But in practice, things still work out
(cf. Muzereau–Smart–Vercauteren).