A comparison of pairing-friendly curves at the 192-bit security level

Aurore Guillevic

Inria Nancy, Caramba team

17/04/2019
WRACH workshop, Roscoff
Joint work with Shashank Singh, IISER Bhopal, India
Plan

Introduction: Discrete logarithm and NFS

Key sizes for DL-based crypto

Pairings

Key-sizes for pairing-based crypto

Future work
Asymmetric cryptography

Factorization (RSA cryptosystem)

Discrete logarithm problem (use in Diffie-Hellman, etc)

Given a finite cyclic group \((G, \cdot)\), a generator \(g\) and \(h \in G\), compute \(x\) s.t. \(h = g^x\).

\(\rightarrow\) can you invert the exponentiation function \((g, x) \mapsto g^x\)?

Common choice of \(G\):

- prime finite field \(\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z}\) (1976)
- characteristic 2 field \(\mathbb{F}_{2^n}\) (\(\approx\) 1979)
- elliptic curve \(E(\mathbb{F}_p)\) (1985)
Discrete log problem

How fast can you invert the exponentiation function \((g, x) \mapsto g^x\)?

- \(g \in G\) generator, \(\exists\) always a preimage \(x \in \{1, \ldots, \#G\}\)
- naive search, try them all: \(\#G\) tests
- random walk in \(G\), cycle path finding algorithm in a connected graph Floyd \(\rightarrow\) Pollard, baby-step-giant-step, \(O(\sqrt{\#G})\) (the cycle path encodes the answer)
- parallel search in each distinct subgroup (Pohlig-Hellman)
- algorithmic refinements
Discrete log problem

How fast can you invert the exponentiation function \((g, x) \mapsto g^x\)?

- \(g \in G\) generator, \(\exists\) always a preimage \(x \in \{1, \ldots, \#G\}\)
- naive search, try them all: \(\#G\) tests
- random walk in \(G\), cycle path finding algorithm in a connected graph Floyd \(\rightarrow\) Pollard, baby-step-giant-step, \(O(\sqrt{\#G})\) (the cycle path encodes the answer)
- parallel search in each distinct subgroup (Pohlig-Hellman)
- algorithmic refinements

\[\rightarrow\] Choose \(G\) of large prime order (no subgroup)
\[\rightarrow\] complexity of inverting exponentiation in \(O(\sqrt{\#G})\)
\[\rightarrow\] security level 128 bits means \(\sqrt{\#G} \geq 2^{128}\)
- analogy with symmetric crypto, keylength 128 bits (16 bytes)
Discrete log problem

How fast can you invert the exponentiation function \((g, x) \mapsto g^x\)?

\(G\) cyclic group of prime order, complexity \(O(\sqrt{\#G})\).
Discrete log problem

How fast can you invert the exponentiation function \((g, x) \mapsto g^x\)?

**G** cyclic group of prime order, complexity \(O(\sqrt{\#G})\).

better way?
How fast can you invert the exponentiation function \((g, x) \mapsto g^x\)?

**G** cyclic group of prime order, complexity \(O(\sqrt{\#G})\).

better way?
→ Use additional structure of **G**.
Discrete log problem when $G = (\mathbb{Z}/p\mathbb{Z})^*$

Index calculus algorithm [Western–Miller 68, Adleman 79], prequel of the Number Field Sieve algorithm (NFS)

- $p$ prime, $(p - 1)/2$ prime, $G = (\mathbb{Z}/p\mathbb{Z})^*$, gen. $g$, target $h$
- get many multiplicative relations in $G$
  \[ g^t = g_1^{e_1} g_2^{e_2} \cdots g_i^{e_i} \pmod{p}, \quad g, g_1, g_2, \ldots, g_i \in G \]
- find a relation $h = g_1^{e'_1} g_2^{e'_2} \cdots g_i^{e'_i} \pmod{p}$
- take logarithm: linear relations
  \[ t = e_1 \log g g_1 + e_2 \log g g_2 + \ldots + e_i \log g g_i \pmod{p - 1} \]
  \[ \vdots \]
  \[ \log g h = e'_1 \log g g_1 + e'_2 \log g g_2 + \ldots + e'_i \log g g_i \pmod{p - 1} \]
- solve a linear system
- get $x = \log g h$
Index calculus in \((\mathbb{Z}/p\mathbb{Z})^*\): example

\[ p = 1109, \ r = (p - 1)/4 = 277 \text{ prime} \]
Smoothness bound \( B = 13 \)
\( \mathcal{F}_{13} = \{2, 3, 5, 7, 11, 13\} \) small primes up to \( B \)
\( B\)-smooth integer: \( n = \prod_{p_i \leq B} p_i^{e_i}, \ p_i \text{ prime} \)

is \( g^i \) smooth? \( 1 \leq i \leq 72 \) is enough

\[

g^1 = 2 = 2 \\
g^{13} = 429 = 3 \cdot 11 \cdot 13 \\
g^{16} = 105 = 3 \cdot 5 \cdot 7 \\
g^{21} = 33 = 3 \cdot 11 \\
g^{44} = 1029 = 3 \cdot 7^3 \\
g^{72} = 325 = 5^2 \cdot 13
\]

\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 1 \\
0 & 1 & 1 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 3 & 0 & 0 \\
0 & 0 & 2 & 0 & 0 & 1 \\
\end{bmatrix}
\cdot
\begin{bmatrix}
1 \\
13 \\
16 \\
21 \\
44 \\
72 \\
\end{bmatrix}
= 
\begin{bmatrix}
1 \\
219 \\
40 \\
34 \\
79 \\
269 \\
\end{bmatrix}	ext{ mod 277}
\]

\[
\rightarrow \log_g 7 = 34 \text{ mod 277}, \text{ that is, } (g^{34})^4 = 7^4 \\
g^{34} = 7u \text{ and } u^4 = 1
\]
Index calculus in \((\mathbb{Z}/p\mathbb{Z})^*\): example

\(x = [1, 219, 40, 34, 79, 269] \mod 277\)

subgroup of order 4: \(g_4 = g^{(p-1)/4}\)

\(\{1, g_4, g_4^2, g_4^3\} = \{1, 354, 1108, 755\}\)

\[
\begin{align*}
3/g^{219} &= 1 \Rightarrow \log_g 3 = 219 \\
5/g^{40} &= -1 \Rightarrow \log_g 5 = 40 + (p-1)/2 = 594 \\
7/g^{34} &= g_4 \Rightarrow \log_g 7 = 34 + (p-1)/4 = 311 \\
11/g^{79} &= g_4^3 \Rightarrow \log_g 11 = 79 + 3(p-1)/4 = 910 \\
13/g^{269} &= g_4^3 \Rightarrow \log_g 13 = 269 + 3(p-1)/4 = 1100
\end{align*}
\]

\(v = [1, 219, 594, 311, 910, 1100] \mod p - 1\)

Target \(h = 777\)

\(g^{10} \cdot 777 = 495 = 3^2 \cdot 5 \cdot 11 \mod p\)

\(\log_2 777 = -10 + 2 \log_g 3 + \log_g 5 + \log_g 11 = 824 \mod p - 1\)

\(g^{824} = 777\)
Index calculus in \((\mathbb{Z}/p\mathbb{Z})^\ast\): example

**Trick**

Multiplicative relations over the **integers**

\(g_1, g_2, \ldots, g_i \longleftrightarrow\) small prime integers

Smooth integers \(n = \prod_{p_i \leq B} p_i^{e_i}\) are quite common \(\rightarrow\) it works
Index calculus in \((\mathbb{Z}/p\mathbb{Z})^*\): example

**Trick**

Multiplicative relations over the **integers**

\[ g_1, g_2, \ldots, g_i \leftrightarrow \text{small prime integers} \]

Smooth integers \( n = \prod_{p_i \leq B} p_i^{e_i} \) are quite common \( \rightarrow \) it works

**Improvements in the 80’s, 90’s:**

- Sieve (faster relation collection)
- Multiplicative relations in **number fields**
  Smaller integers and norms to factor
- Better **sparse linear algebra**
- Independent target \( h \)
Number Field: Toy example with $\mathbb{Z}[i]$

(1986 technology, Coppersmith–Odlyzko–Schroeppel) reduce further the size of the integers to factor

If $p = 1 \mod 4$, $\exists U, V$ s.t. $p = U^2 + V^2$ and $|U|, |V| < \sqrt{p}$

$U/V \equiv m \mod p$ and $m^2 + 1 = 0 \mod p$

Define a map from $\mathbb{Z}[i]$ to $\mathbb{Z}/p\mathbb{Z}$

$\phi: \mathbb{Z}[i] \rightarrow \mathbb{Z}/p\mathbb{Z}$

\[
i \mapsto m \mod p \text{ where } m = U/V, \ m^2 + 1 = 0 \mod p
\]

ring homomorphism $\phi(a + bi) = a + bm$

\[
\phi(a + bi) = a + bm = (a + b \frac{U}{V}) = (aV + bU)V^{-1} \equiv m \mod p
\]
Example in $\mathbb{Z}[i]$

- $p = 1109 = 1 \mod 4$, $r = (p - 1)/4 = 277$ prime
- $p = 22^2 + 25^2$
- $\max(|a|, |b|) = A = 20$, $B = 13$ smoothness bound

Rational side

- $\mathcal{F}_{\text{rat}} = \{2, 3, 5, 7, 11, 13\}$ primes up to $B$

Algebraic side: think about the complex number in $\mathbb{C}$

- $(1 + i)(1 - i) = 2$, $(2 + i)(2 - i) = 5$, $(2 + 3i)(2 - 3i) = 13$
- All primes $p = 1 \mod 4$
  - can be written as a sum of two squares $p = a^2 + b^2$
  - factor into two conjugate Gaussian integers $(a + ib)(a - ib)$

Units: $i^2 = -1$

- $\mathcal{F}_{\text{alg}} = \{1 + i, 1 - i, 2 + i, 2 - i, 2 + 3i, 2 - 3i\}$
- “primes” of norm up to $B$
- $\mathcal{U}_{\text{alg}} = \{-1, i\}$ Units
Example in $\mathbb{Z}[i]$

$p = 1109$

$$(a, b) = (-4, 7),$$
$\text{Norm}(-4 + 7i) = (-4)^2 + 7^2 = 65 = 5 \cdot 13$

In $\mathbb{Z}[i]$,

$\begin{align*}
\text{► } 5 &= (2 + i)(2 - i) \\
\text{► } 13 &= (2 + 3i)(2 - 3i)
\end{align*}$

Then,

$\begin{align*}
\rightarrow (2 \pm i)(2 \pm 3i) \text{ has norm 65} \\
\rightarrow \pm((i))(2 \pm i)(2 \pm 3i) &= (-4 + 7i)
\end{align*}$

We obtain $i(2 - i)(2 + 3i) = -4 + 7i$
### Example in $\mathbb{Z}[i]$  

<table>
<thead>
<tr>
<th>$a + bi$</th>
<th>$aV + bU = \text{factor in } \mathbb{Z}$</th>
<th>$a^2 + b^2$</th>
<th>$\text{factor in } \mathbb{Z}[i]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$-17 + 19i$</td>
<td>$-7 = -7$</td>
<td>$650 = 2 \cdot 5^2 \cdot 13$</td>
<td>$-(1 - i)(2 + i)^2(2 - 3i)$</td>
</tr>
<tr>
<td>$-11 + 2i$</td>
<td>$-231 = -3 \cdot 7 \cdot 11$</td>
<td>$125 = 5^3$</td>
<td>$i(2 + i)^3$</td>
</tr>
<tr>
<td>$-6 + 17i$</td>
<td>$224 = 2^5 \cdot 7$</td>
<td>$325 = 5^2 \cdot 13$</td>
<td>$(2 + i)^2(2 + 3i)$</td>
</tr>
<tr>
<td>$-4 + 7i$</td>
<td>$54 = 2 \cdot 3^3$</td>
<td>$65 = 5 \cdot 13$</td>
<td>$i(2 - i)(2 + 3i)$</td>
</tr>
<tr>
<td>$-3 + 4i$</td>
<td>$13 = 13$</td>
<td>$25 = 5^2$</td>
<td>$-(2 - i)^2$</td>
</tr>
<tr>
<td>$-2 + i$</td>
<td>$-28 = -2^2 \cdot 7$</td>
<td>$5 = 5$</td>
<td>$-(2 - i)$</td>
</tr>
<tr>
<td>$-2 + 3i$</td>
<td>$16 = 2^4$</td>
<td>$13 = 13$</td>
<td>$-(2 - 3i)$</td>
</tr>
<tr>
<td>$-2 + 11i$</td>
<td>$192 = 2^6 \cdot 3$</td>
<td>$125 = 5^3$</td>
<td>$-(2 - i)^3$</td>
</tr>
<tr>
<td>$-1 + i$</td>
<td>$-3 = -3$</td>
<td>$2 = 2$</td>
<td>$-(1 - i)$</td>
</tr>
<tr>
<td>$i$</td>
<td>$22 = 2 \cdot 11$</td>
<td>$1 = 1$</td>
<td>$i$</td>
</tr>
<tr>
<td>$1 + 3i$</td>
<td>$91 = 7 \cdot 13$</td>
<td>$10 = 2 \cdot 5$</td>
<td>$(1 + i)(2 + i)$</td>
</tr>
<tr>
<td>$1 + 5i$</td>
<td>$135 = 3^3 \cdot 5$</td>
<td>$26 = 2 \cdot 13$</td>
<td>$-(1 - i)(2 - 3i)$</td>
</tr>
<tr>
<td>$2 + i$</td>
<td>$72 = 2^3 \cdot 3^2$</td>
<td>$5 = 5$</td>
<td>$(2 + i)$</td>
</tr>
<tr>
<td>$5 + i$</td>
<td>$147 = 3 \cdot 7^2$</td>
<td>$26 = 2 \cdot 13$</td>
<td>$-i(1 + i)(2 + 3i)$</td>
</tr>
</tbody>
</table>
Example in $\mathbb{Z}[i]$: Matrix

Build the matrix of relations:
- one row per $(a, b)$ pair s.t. both norms are smooth
- one column per prime of $\mathcal{F}_{\text{rat}}$
- one column for $1/V$
- one column per prime ideal of $\mathcal{F}_{\text{alg}}$
- one column per unit $(-1, i)$
- store the exponents
Example in $\mathbb{Z}[i]$

$$M = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 2 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 3 & 0 & 0 & 0 \\
5 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 2 & 0 & 1 & 0 \\
1 & 3 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\
2 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
4 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
6 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 3 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 3 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
3 & 2 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 2 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0
\end{bmatrix}$$
Example in $\mathbb{Z}[i]$

$$M = \begin{bmatrix}
2 & 3 & 5 & 7 & 11 & 13 & \frac{1}{\sqrt{5}} & -1 & i \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 1 \\
5 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 3 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
4 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
6 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
3 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
3 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\end{bmatrix}$$
Example in $\mathbb{Z}[i]$

$$M = \begin{bmatrix}
2 & 3 & 5 & 7 & 11 & 13 & \frac{1}{\sqrt{2}} & -1 & i \\
\begin{array}{cccc}
-1 & -2 \\
1 & 1 & 1 & -1-1 & -1-2 & -1 \\
1 & 1 & 1 & 1-1-1 & -3 \\
5 & 1 & 1 & -1 & -2 & -1 \\
1 & 3 & 1 & -1 & -1 & -1 & -1 \\
& 1 & 1-1 & -2 \\
2 & 1 & 1 & -1 \\
& 4 & 1-1 & -1 \\
6 & 1 & 1-1 & -3 \\
& 1 & 1 & -1 \\
1 & 1 & 1 & -1 \\
& 1 & 1 & 1-1 & -1 & -1 \\
3 & 1 & 1-1 & -1 & -1 \\
& 3 & 1 & 1-1-1-1 & -1 & -1 \\
3 & 2 & 1 & -1 \\
1 & 2 & 1-1-1-1 & -1 & -1 \\
\end{array}
\end{bmatrix}$$
Example in $\mathbb{Z}[i]$

Right kernel $M \cdot x = 0 \mod (p - 1)/4 = 277$:
\[ x = (\underbrace{1, 219, 40, 34, 79, 269}_\text{rational side}, \underbrace{197, 0, 0}_\text{1/V}, \underbrace{139, 139, 84, 233, 68, 201}_\text{algebraic side}) \]

Logarithms (in some basis)

Rational side: logarithms of $\{2, 3, 5, 7, 11, 13\}$
\[ \rightarrow \log x_i / \log 2 \]
\[ x = [1, 219, 40, 34, 79, 269] \mod 277 \]
\[ \rightarrow \text{order 4 subgroup} \]
\[ v = [1, 219, 594, 311, 910, 1100] \mod p - 1 \]

Target 314, generator $g = 2$
\[ g^2 \cdot 314 = 147 = 3 \cdot 7^2 \]
\[ \log_g 314 = \log_g 3 + 2 \log_g 7 - 2 = 219 + 2 \cdot 311 - 2 = 839 \mod p - 1 \]
\[ 2^{839} = 314 \mod p, \log_g 314 = 839 \]
Number Field Sieve today

- NFS: Gordon 93, improvements Schirokauer 93
- polynomial selection Joux–Lercier 03
- Franke–Kleinjung 08 sieve, ECM factorization H. Lenstra 87
- block Lanczos, Wiedemann 86 sparse linear algebra
- Joux–Lercier 03 descent, early-abort strategy Pomerance 82
Latest DL record computation: 768-bit $\mathbb{F}_p$

Kleinjung, Diem, A. Lenstra, Priplata, Stahlke, Eurocrypt’2017. $p = \lfloor 2^{766} \times \pi \rfloor + 62762$ prime, 768 bits, 232 decimal digits, $p = 1219344858334286932696341909195796109526657386154251328029$

2736561757668709803065055845773891258608267152015472257940
7293588325886803643328721799472154219914818284150580043314
8410869683590659346847659519108393837414567892730579162319
$(p - 1)/2$ prime

$f(x) = 140x^4 + 34x^3 + 86x^2 + 5x - 55$
$g(x) = 370863403886416141150505523919527677231932618184100095924x^3$
$- 1937981312833038778565617469829395544065255938015920309679x^2$
$- 217583293626947899787577441128333027617541095004734736415x$
$+ 277260730400349522890422618473498148528706115003337935150$

Enumerate ($\sim 10^{12}$) all $f(x)$ s.t. $|f_i| \leq 165$
By construction, $|g_i| \approx p^{1/4}$
Latest DL record computation: 768-bit $\mathbb{F}_p$

\[
gcd(f, g) = 1 \text{ in } \mathbb{Q}[x] \\
\exists \text{ root } m \text{ s.t. } f(m) = g(m) = 0 \pmod{p}, m = \\
4290295629231970357488936064013995423387122927373167219112 \\
8794979019508571426956110520280493413148710512618823586632 \\
14844974131883926532462067740277566464444183240629650904112 \\
110269916261074281303302883725258878464313312196475775222 \\

Multiplicative relations: for all $|a_i| \leq A \approx 2^{32}$, $\gcd(a_0, a_1) = 1$

- factors $\text{Norm}_f = \text{Resultant}(f, a_0 + a_1x) \approx 130$ bits, 39 dd
- factors $\text{Norm}_g = \text{Resultant}(g, a_0 + a_1x) \approx 290$ bits, 87 dd

Linear algebra: square sparse matrix of $23.5 \cdot 10^6$ rows
Total time: 5300 core-years on Intel Xeon E5-2660 2.2GHz
Plan

Introduction: Discrete logarithm and NFS

Key sizes for DL-based crypto

Pairings

Key-sizes for pairing-based crypto

Future work
Complexity and key-sizes for cryptography

[Lenstra-Verheul’01] gives RSA key-sizes

Security estimates use

- asymptotic complexity of the best known algorithm (here NFS)
- latest record computation (now 768-bit)
- extrapolation
Complexity

Subexponential asymptotic complexity:

\[ L_{p^n}(\alpha, c) = e^{(c+o(1))(\log p^n)^\alpha(\log \log p^n)^{1-\alpha}} \]

- \( \alpha = 1 \): exponential
- \( \alpha = 0 \): polynomial
- \( 0 < \alpha < 1 \): sub-exponential (including NFS)

1. polynomial selection (precomp., 5% to 10% of total time)
2. relation collection \( L_{p^n}(1/3, c) \)
3. linear algebra \( L_{p^n}(1/3, c) \)
4. individual discrete log computation \( L_{p^n}(1/3, c' < c) \)
\[ L_N^0(1/3, 1.923)/2^{8.2} \ (DL-768 \leftrightarrow 2^{68.32}) \]
\[ L_N^0(1/3, 1.923)/2^{14} \ (RSA-768 \leftrightarrow 2^{67}) \]
Key length

► keylength.com
► France: ANSSI RGS B

RSA modulus and prime fields for DL: 3072 to 3200 bits
sub-exponential complexity to invert DL in $\mathbb{F}_p$

Elliptic curves: over prime field of 256 bits (much smaller)
exponential cpx. to invert DL in $E(\mathbb{F}_p)$
Key length

► keylength.com
► France: ANSSI RGS B

RSA modulus and prime fields for DL: 3072 to 3200 bits
sub-exponential complexity to invert DL in $\mathbb{F}_p$

Elliptic curves: over prime field of 256 bits (much smaller)
exponential cpx. to invert DL in $E(\mathbb{F}_p)$

Why finite fields in 2019?

because old crypto in $\mathbb{F}_p$ is still in use
cpx $= L_p(1/3, 1.923)$ since 1993: very-well known
because of pairings: $\mathbb{F}_p^n$ since 2000
Plan

Introduction: Discrete logarithm and NFS

Key sizes for DL-based crypto

Pairings

Key-sizes for pairing-based crypto

Future work
Cryptographic pairing: black-box properties

\((G_1, +), (G_2, +), (G_T, \cdot)\) three cyclic groups of large prime order \(r\)

Bilinear Pairing: map \(e : G_1 \times G_2 \rightarrow G_T\)

1. bilinear: 
   \[ e(P_1 + P_2, Q) = e(P_1, Q) \cdot e(P_2, Q), \]
   \[ e(P, Q_1 + Q_2) = e(P, Q_1) \cdot e(P, Q_2) \]

2. non-degenerate: 
   \[ e(g_1, g_2) \neq 1 \text{ for } \langle g_1 \rangle = G_1, \langle g_2 \rangle = G_2 \]

3. efficiently computable.

Mostly used in practice:

\[ e([a]P, [b]Q) = e([b]P, [a]Q) = e(P, Q)^{ab}. \]

\(\sim\) Many applications in asymmetric cryptography.
Examples of application

- 1984: idea of identity-based encryption (IBE) by Shamir
- 1999: first practical identity-based cryptosystem of Sakai-Ohgishi-Kasahara
- 2000: constructive pairings, Joux’s tri-partite key-exchange
- 2001: IBE of Boneh-Franklin, short signatures Boneh-Lynn-Shacham

... 

- Broadcast encryption, re-keying
- aggregate signatures
- zero-knowledge (ZK) proofs
  - non-interactive ZK proofs (NIZK)
  - ZK-SNARK (Z-cash)
Bilinear Pairings

Rely on

- **Discrete Log Problem (DLP):**
  given \( g, h \in G \), compute \( x \) s.t. \( g^x = h \)

- **Diffie-Hellman Problem (DHP):**
  given \( g, g^a, g^b \in G \), compute \( g^{ab} \)

- bilinear DLP and DHP

- pairing inversion problem
Pairing-based cryptography

Weil or Tate pairing on an elliptic curve
Discrete logarithm problem with one more dimension.

\[ e : E(\mathbb{F}_{p^n})[r] \times E(\mathbb{F}_{p^n})[r] \rightarrow \mathbb{F}_{p^n}^*, \quad e([a]P, [b]Q) = e(P, Q)^{ab} \]
Pairing-based cryptography

Weil or Tate pairing on an elliptic curve
Discrete logarithm problem with one more dimension.

\[ e : E(\mathbb{F}_{p^n})[r] \times E(\mathbb{F}_{p^n})[r] \rightarrow \mathbb{F}_{p^n}^*, \quad e([a]P, [b]Q) = e(P, Q)^{ab} \]

Attacks
Pairing-based cryptography

Weil or Tate pairing on an elliptic curve
Discrete logarithm problem with one more dimension.

\[ e : E(\mathbb{F}_{p^n})[r] \times E(\mathbb{F}_{p^n})[r] \longrightarrow \mathbb{F}_p^*, \quad e([a]P, [b]Q) = e(P, Q)^{ab} \]

Attacks

- inversion of \( e \): hard problem (exponential)
Pairing-based cryptography

Weil or Tate pairing on an elliptic curve
Discrete logarithm problem with one more dimension.

\[ e : E(\mathbb{F}_{p^n})[r] \times E(\mathbb{F}_{p^n})[r] \rightarrow \mathbb{F}_{p^n}^*, \quad e([a]P, [b]Q) = e(P, Q)^{ab} \]

Attacks
- inversion of \( e \): hard problem (exponential)
- discrete logarithm computation in \( E(\mathbb{F}_p) \): hard problem (exponential, in \( O(\sqrt{r}) \))
Pairing-based cryptography

Weil or Tate pairing on an elliptic curve
Discrete logarithm problem with one more dimension.

\[ e : E(\mathbb{F}_{p^n})[r] \times E(\mathbb{F}_{p^n})[r] \to \mathbb{F}^*_{p^n}, \quad e([a]P, [b]Q) = e(P, Q)^{ab} \]

Attacks

- inversion of \( e \): hard problem (exponential)
- discrete logarithm computation in \( E(\mathbb{F}_p) \): hard problem (exponential, in \( O(\sqrt{r}) \))
- discrete logarithm computation in \( \mathbb{F}^*_{p^n} \): easier, subexponential \( \to \) take a large enough field
Pairing-friendly curves are special

\[ r \mid p^n - 1, \ G_T \subset \mathbb{F}_{p^n}, \ n \text{ is minimal: embedding degree} \]

Tate Pairing: \( e : G_1 \times G_2 \rightarrow G_T \)

When \( n \) is small, the curve is pairing-friendly. This is very rare: usually \( \log n \sim \log r \) ([Balasubramanian Koblitz]).

<table>
<thead>
<tr>
<th>Curve</th>
<th>G_T \subset p^n</th>
<th>p^2, p^6</th>
<th>p^3, p^4, p^6</th>
<th>p^{12}</th>
<th>p^{16}</th>
<th>p^{18}</th>
<th>p^{24}</th>
</tr>
</thead>
<tbody>
<tr>
<td>super-singular</td>
<td>MNT</td>
<td>BN</td>
<td>BLS12</td>
<td>KSS16</td>
<td>KSS18</td>
<td>BLS24</td>
<td></td>
</tr>
</tbody>
</table>

MNT, \( n = 6 \):
\[ p(x) = 4x^2 + 1, \ #E(\mathbb{F}_p) = r(x) = x^2 \mp 2x + 1 \]

BN, \( n = 12 \):
\[ p(x) = 36x^4 + 36x^3 + 24x^2 + 6x + 1 \]
\[ r(x) = 36x^4 + 36x^3 + 18x^2 + 6x + 1 \]
Plan

Introduction: Discrete logarithm and NFS

Key sizes for DL-based crypto

Pairings

Key-sizes for pairing-based crypto

Future work
Discrete Log in $\mathbb{F}_p^n$

$\mathbb{F}_p^n$ much less investigated than $\mathbb{F}_p$ or integer factorization. Much better results in pairing-related fields.
Discrete Log in $\mathbb{F}_{p^n}$

$\mathbb{F}_{p^n}$ much less investigated than $\mathbb{F}_p$ or integer factorization. Much better results in pairing-related fields

- Special NFS in $\mathbb{F}_{p^n}$: Joux–Pierrot 2013
- Tower NFS (TNFS): Barbulescu Gaudry Kleinjung 2015
- Tower of number fields

Use more structure: subfields
\( F_{p^6} \), subfield \( F_{p^2} \) defined by \( y^2 + 1 \)

\( g = (g_{00} + g_{01}i) + (g_{10} + g_{11}i)x + (g_{20} + g_{21}i)x^2 \in F_{p^6} \)

Idea: \( a_0 + a_1x \rightarrow a = (a_{00} + a_{01}i) + (a_{10} + a_{11}i)x \)

Integers to factor are **much smaller**

- factors integer \( \text{Norm}_f = \text{Res}(\text{Res}(a, f_y(x)), y^2 + 1) \)
- factors integer \( \text{Norm}_g = \text{Res}(\text{Res}(a, g_y(x)), y^2 + 1) \)

\( \text{Res} = \) resultant of polynomials
### Complexities

**large characteristic** $p = L_p^n(\alpha), \ \alpha > 2/3$:  

<table>
<thead>
<tr>
<th>Expression</th>
<th>Approximation</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(64/9)^{1/3}$</td>
<td>$\approx 1.923$</td>
<td>NFS</td>
</tr>
</tbody>
</table>

**special $p$:**  

<table>
<thead>
<tr>
<th>Expression</th>
<th>Approximation</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(32/9)^{1/3}$</td>
<td>$\approx 1.526$</td>
<td>SNFS</td>
</tr>
</tbody>
</table>

**medium characteristic** $p = L_p^n(\alpha), \ 1/3 < \alpha < 2/3$:  

<table>
<thead>
<tr>
<th>Expression</th>
<th>Approximation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(96/9)^{1/3}$</td>
<td>$\approx 2.201$</td>
<td>prime $n$ NFS-HD (Conjugation)</td>
</tr>
<tr>
<td>$(48/9)^{1/3}$</td>
<td>$\approx 1.747$</td>
<td>composite $n$, best case of TNFS: when parameters fit perfectly</td>
</tr>
</tbody>
</table>

**special $p$:**  

<table>
<thead>
<tr>
<th>Expression</th>
<th>Approximation</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(64/9)^{1/3}$</td>
<td>$\approx 1.923$</td>
<td>NFS-HD+Joux–Pierrot’13</td>
</tr>
<tr>
<td>$(32/9)^{1/3}$</td>
<td>$\approx 1.526$</td>
<td>composite $n$, best case of STNFS</td>
</tr>
</tbody>
</table>
Latest variants of TNFS (Kim–Barbulescu, Kim–Jeong) seem most promising for $\mathbb{F}_{p^n}$ where $n$ is composite.

We need record computations if we want to extrapolate from asymptotic complexities.

The asymptotic complexities do not correspond to a fixed $n$, but to a ratio between $n$ and $p$. 

Estimating key sizes for DL in $\mathbb{F}_{p^n}$
Simulation of STNFS: why?

- upper bound on the norms
- (heuristic) upper bound on the running-time of STNFS
- bound is not tight: running-time could be much faster
- security is over-estimated

Possible solution:

- remove combinatorial factor from the bound
- smaller norms, faster STNFS, lower security
- much larger key-sizes
- bad for practical applications: larger keys are required

Example BN curves, targeted 128-bit security level:

- $p$ was 256 bits before STNFS
- Now $p$ from 384 to 512 bits

But we don’t want to use too large $p$ for nothing.
Largest record computations in $\mathbb{F}_{p^n}$ with NFS\textsuperscript{1}

<table>
<thead>
<tr>
<th>Finite field $\mathbb{F}_{p^n}$</th>
<th>Size of $p^n$</th>
<th>Cost: CPU days</th>
<th>Authors</th>
<th>sieving dim</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathbb{F}_{p^{12}}$</td>
<td>203</td>
<td>11</td>
<td>[HAKT13]</td>
<td>7</td>
</tr>
<tr>
<td>$\mathbb{F}_{p^{6}}$</td>
<td>422</td>
<td>9,520</td>
<td>[GGMT17]</td>
<td>3</td>
</tr>
<tr>
<td>$\mathbb{F}_{p^{5}}$</td>
<td>324</td>
<td>386</td>
<td>[GGM17]</td>
<td>3</td>
</tr>
<tr>
<td>$\mathbb{F}_{p^{4}}$</td>
<td>392</td>
<td>510</td>
<td>[BGGM15b]</td>
<td>2</td>
</tr>
<tr>
<td>$\mathbb{F}_{p^{3}}$</td>
<td>593</td>
<td>8,400</td>
<td>[GGM16]</td>
<td>2</td>
</tr>
<tr>
<td>$\mathbb{F}_{p^{2}}$</td>
<td>595</td>
<td>175</td>
<td>[BGGM15a]</td>
<td>2</td>
</tr>
<tr>
<td>$\mathbb{F}_{p}$</td>
<td>768</td>
<td>1,935,825</td>
<td>[KDLPS17]</td>
<td>2</td>
</tr>
</tbody>
</table>

None used TNFS, only NFS and NFS-HD were implemented.

\textsuperscript{1}Data extracted from DiscreteLogDB by L.Grémy
Simulation without sieving

Implementation of Barbulescu–Duquesne technique

Space: $S = \{ \sum a_{0i}y^i + (\sum a_{1i}y^i)x, \; |a_{ji}| < A \}$

Variants:

- compute $\alpha(f), \alpha(g)$ (w.r.t. subfield) **bias in smoothness**
- select polys $f, g$ with negative bias $\alpha(f), \alpha(g)$
- Monte-Carlo simulation with $10^6$ points in $S$ taken at random. For each point:
  1. compute its algebraic norm $N_f, N_g$ in each number field
  2. smoothness probability with Dickman-$\rho$

- Average smoothness probability over the subset of points
  $\rightarrow$ estimation of the total number of possible relations in $S$

- dichotomy to approach the best balanced parameters: smoothness bound $B$, coefficient bound $A$. 
Simulation without sieving

Python/SageMath experimental implementation

Nice “bug”:

\[ A = 8 \]
\[ h = y^{2+1} \]
\[ a_0 = [\text{randint}(-A,A+1) \text{ for } ai \text{ in } \text{range}(h.\text{degree}())] \]
\[ a_1 = [\text{randint}(-A,A+1) \text{ for } ai \text{ in } \text{range}(h.\text{degree}())] \]

\[ A = 8 \]
\[ h = y^{2+1} \]
\[ a_0 = [\text{randrange}(-A,A+1) \text{ for } ai \text{ in } \text{range}(h.\text{degree}())] \]
\[ a_1 = [\text{randrange}(-A,A+1) \text{ for } ai \text{ in } \text{range}(h.\text{degree}())] \]
## Key size for pairings

<table>
<thead>
<tr>
<th>$\mathbb{F}_{p^n}$, curve</th>
<th>cost DL $2^{128}$</th>
<th>cost DL $2^{192}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathbb{F}_p$</td>
<td>$\log_2 p$</td>
<td>$\log_2 p^n$</td>
</tr>
<tr>
<td></td>
<td>3072–3200</td>
<td>7400–8000</td>
</tr>
<tr>
<td>$\mathbb{F}_{p^6}$, MNT</td>
<td>640–672</td>
<td>3840–4032</td>
</tr>
<tr>
<td></td>
<td>1536</td>
<td>9216</td>
</tr>
<tr>
<td>$\mathbb{F}_{p^{12}}$, BN</td>
<td>416–448</td>
<td>4992–5376</td>
</tr>
<tr>
<td></td>
<td>1024</td>
<td>12288</td>
</tr>
<tr>
<td>$\mathbb{F}_{p^{12}}$, BLS</td>
<td>416–448</td>
<td>4992–5376</td>
</tr>
<tr>
<td></td>
<td>1120</td>
<td>13440</td>
</tr>
<tr>
<td>$\mathbb{F}_{p^{16}}$, KSS</td>
<td>330</td>
<td>5280</td>
</tr>
<tr>
<td></td>
<td>768</td>
<td>12288</td>
</tr>
<tr>
<td>$\mathbb{F}_{p^{18}}$, KSS</td>
<td>348</td>
<td>6264</td>
</tr>
<tr>
<td></td>
<td>640</td>
<td>11520</td>
</tr>
</tbody>
</table>
Plan

Introduction: Discrete logarithm and NFS

Key sizes for DL-based crypto

Pairings

Key-sizes for pairing-based crypto

Future work
Future work

- automatic tool (currently developed in Python/SageMath)
- $\mathbb{F}_p^{15}, \mathbb{F}_p^{21}, \mathbb{F}_p^{27}$
- Compare Special-TNFS and TNFS
- $a_0 + a_1 x \rightarrow$ consider $a_0 + a_1 x + a_2 x^2, a_i = a_{i0} + a_{i1} y + \ldots$
- Estimate the proportion of duplicate relations (2%, 20%, 60%?)
- How to sieve very efficiently in even dimension 4 to 24 to avoid costly factorization in the relation collection?
- Record computation in $\mathbb{F}_p^6$
L. Adleman.
A subexponential algorithm for the discrete logarithm problem with applications to cryptography.
https://doi.org/10.1109/SFCS.1979.2.

DL record computation in $\text{GF}(p^4)$ of 392 bits (120dd).
Announcement at the CATREL workshop, October 2nd 2015.

Improving NFS for the discrete logarithm problem in non-prime finite fields.

The tower number field sieve.
R. Barbulescu and A. Lachand.
Some mathematical remarks on the polynomial selection in NFS.
https://hal.inria.fr/hal-00954365,
https://doi.org/10.1090/mcom/3112.

E. R. Canfield, P. Erdős, and C. Pomerance.
On a problem of Oppenheim concerning “factorisatio numerorum”.

S. Chatterjee, A. Menezes, and F. Rodríguez-Henríquez.
On instantiating pairing-based protocols with elliptic curves of embedding degree one.

D. Coppersmith.
Fast evaluation of logarithms in fields of characteristic two.
https://doi.org/10.1109/TIT.1984.1056941.


D. M. Gordon.
Discrete logarithms in \( \text{GF}(p) \) using the number field sieve.

L. Grémy, A. Guillevic, and F. Morain.
Discrete logarithm record computation in \( \text{GF}(p^5) \) of 100 decimal digits using NFS with 3-dimensional sieving.
Number Theory list, item 004981, August 1st 2017.
https://listserv.nodak.edu/cgi-bin/wa.exe?A2=NMBRTHRY;68019370.1708.

Computing discrete logarithms in \( \mathbb{F}_{p^6} \).

A. Guillevic, F. Morain, and E. Thomé.
Solving discrete logarithms on a 170-bit MNT curve by pairing reduction.
An experiment of number field sieve for discrete logarithm problem over GF($p^{12}$).
In M. Fischlin and S. Katzenbeisser, editors, *Number Theory and Cryptography*,

A construction of 3-dimensional lattice sieve for number field sieve over $\mathbb{F}_{p^n}$.

A. Joux, R. Lercier, N. Smart, and F. Vercauteren.
The number field sieve in the medium prime case.

T. Kim and R. Barbulescu.
Extended tower number field sieve: A new complexity for the medium prime case.


D. V. Matyukhin.
Effective version of the number field sieve for discrete logarithms in the field $\text{GF}(p^k)$ (in Russian).

K. S. McCurley.
The discrete logarithm problem.

A. Menezes, P. Sarkar, and S. Singh.
Challenges with assessing the impact of NFS advances on the security of pairing-based cryptography.
C. Pomerance.  
Analysis and comparison of some integer factoring algorithms.  

C. Pomerance.  
Fast, rigorous factorization and discrete logarithm algorithms.  

P. Sarkar and S. Singh.  
A general polynomial selection method and new asymptotic complexities for the tower number field sieve algorithm.  
P. Sarkar and S. Singh.
New complexity trade-offs for the (multiple) number field sieve algorithm in non-prime fields.

O. Schirokauer.
Discrete logarithms and local units.
http://rsta.royalsocietypublishing.org/content/345/1676/409,

A. E. Western and J. C. P. Miller.
*Tables of Indices and Primitive Roots*, volume 9 of *Royal Society Mathematical Tables*.

D. H. Wiedemann.
Solving sparse linear equations over finite fields.