On the complexity of computing discrete logarithms in the field \( \mathbb{F}_{36509} \)

Francisco Rodríguez-Henríquez
CINVESTAV-IPN

Joint work with:
Gora Adj CINVESTAV-IPN
Alfred Menezes University of Waterloo
Thomaz Oliveira CINVESTAV-IPN

Worcester Polytechnic Institute - September 17, 2013
Hard computational problems

1. **Integer factorization problem**: Given an integer $N = p \cdot q$ find its prime factors $p$ and $q$. [$2013 = 3 \cdot 11 \cdot 61$]
Hard computational problems

1. **Integer factorization problem**: Given an integer \( N = p \cdot q \) find its prime factors \( p \) and \( q \). \([2013 = 3 \cdot 11 \cdot 61]\)

2. **Discrete logarithm problem**: Given a prime \( p \) and \( g, h \in [1, p - 1] \), find an integer \( x \) (if one exists) such that, \( g^x \equiv h \mod p \).
   
   \([\text{find } x \text{ such that } 2^x \equiv 304 \mod 419]\)
Hard computational problems

1. **Integer factorization problem:** Given an integer $N = p \cdot q$ find its prime factors $p$ and $q$. [$2013 = 3 \cdot 11 \cdot 61$]

2. **Discrete logarithm problem:** Given a prime $p$ and $g, h \in [1, p - 1]$, find an integer $x$ (if one exists) such that, $g^x \equiv h \mod p$.  
   [find $x$ such that $2^x \equiv 304 \mod 419$] 
   answer: $2^{343} \equiv 304 \mod 419$. 

Hard computational problems

1. Integer factorization problem: Given an integer $N = p \cdot q$ find its prime factors $p$ and $q$. \([2013 = 3 \cdot 11 \cdot 61]\)

2. Discrete logarithm problem: Given a prime $p$ and $g, h \in [1, p - 1]$, find an integer $x$ (if one exists) such that, $g^x \equiv h \mod p$.

\[\text{[find } x \text{ such that } 2^x \equiv 304 \mod 419]\]

\[\text{answer: } 2^{343} \equiv 304 \mod 419.\]

More generally: Given $g, h \in \mathbb{F}_q^*$, find an integer $x$ (if one exists) such that, $g^x \equiv h$, where $q = p^l$ is the power of a prime
Hard computational problems

1. Integer factorization problem: Given an integer \( N = p \cdot q \) find its prime factors \( p \) and \( q \). \([2013 = 3 \cdot 11 \cdot 61]\)

2. Discrete logarithm problem: Given a prime \( p \) and \( g, h \in [1, p - 1] \), find an integer \( x \) (if one exists) such that, \( g^x \equiv h \mod p \).
   [find \( x \) such that \( 2^x \equiv 304 \mod 419 \)]
   answer: \( 2^{343} \equiv 304 \mod 419 \).

More generally: Given \( g, h \in \mathbb{F}_q^* \), find an integer \( x \) (if one exists) such that, \( g^x \equiv h \), where \( q = p^l \) is the power of a prime

3. Elliptic curve discrete logarithm problem: Given an elliptic curve \( E/\mathbb{F}_q \) and \( P, Q \in E(\mathbb{F}_{q^k}) \), find an integer \( x \) (if one exists) such that, \( xP = Q \)
Elliptic curves

borrowed from Quino.
Elliptic curves

- $E$ defined by a Weierstraß equation of the form over a prime field with characteristic different than 2,3:

$$y^2 = x^3 + Ax + B$$
Elliptic curves

- $E$ defined by a Weierstraß equation of the form over a prime field with characteristic different than 2, 3:
  \[ y^2 = x^3 + Ax + B \]

- $E(K)$ set of rational points over a field $K$
Elliptic curves

- $E$ defined by a Weierstraß equation of the form over a prime field with characteristic different than 2,3:

$$y^2 = x^3 + Ax + B$$

- $E(K)$ set of rational points over a field $K$
- Additive group law over $E(K)$
Elliptic curves

- $E$ defined by a Weierstraß equation of the form over a prime field with characteristic different than 2,3:
  \[y^2 = x^3 + Ax + B\]
- $E(K)$ set of rational points over a field $K$
- Additive group law over $E(K)$
- Many applications in cryptography since 1985
  - EC-based Diffie-Hellman key exchange
  - EC-based Digital Signature Algorithm
  - ...

Interest: smaller keys than usual cryptosystems (RSA, ElGamal, ...)

Francisco Rodríguez-Henríquez
Elliptic curves

- $E$ defined by a Weierstraß equation of the form over a prime field with characteristic different than 2,3:

$$y^2 = x^3 + Ax + B$$

- $E(K)$ set of rational points over a field $K$
- Additive group law over $E(K)$
- Many applications in cryptography since 1985
  - EC-based Diffie-Hellman key exchange
  - EC-based Digital Signature Algorithm
  - ...
- Interest: smaller keys than usual cryptosystems (RSA, ElGamal, ...)
Discrete logarithm cryptography

- \((G_1, +)\), an additively-written cyclic group of prime order \(#G_1 = \ell\)
Discrete logarithm cryptography

- $(\mathbb{G}_1, +)$, an additively-written cyclic group of prime order $\# \mathbb{G}_1 = \ell$
- $P$, a generator of the group: $\mathbb{G}_1 = \langle P \rangle$
Discrete logarithm cryptography

- \((G_1, +)\), an additively-written cyclic group of prime order \(\#G_1 = \ell\)
- \(P\), a generator of the group: \(G_1 = \langle P \rangle\)
- Scalar multiplication: for any integer \(k\), we have \(kP = P + P + \cdots + P\) \(k\) times

Scalar multiplication: for any integer \(k\), we have \(kP = P + P + \cdots + P\) \(k\) times
Discrete logarithm cryptography

- $(G_1, +)$, an additively-written cyclic group of prime order $\#G_1 = \ell$
- $P$, a generator of the group: $G_1 = \langle P \rangle$
- Scalar multiplication: for any integer $k$, we have $kP = P + P + \cdots + P$ ($k$ times)

Discrete logarithm: given $Q \in G_1$, compute $k$ such that $Q = kP$
Discrete logarithm cryptography

- $(G_1, +)$, an additively-written cyclic group of prime order $\#G_1 = \ell$
- $P$, a generator of the group: $G_1 = \langle P \rangle$
- **Scalar multiplication**: for any integer $k$, we have
  \[ kP = P + P + \cdots + P \]
  \(k\) times

Scalar multiplication on a curve key: $kP$
Discrete logarithm cryptography

- $(G_1, +)$, an additively-written cyclic group of prime order $\#G_1 = \ell$
- $P$, a generator of the group: $G_1 = \langle P \rangle$
- Scalar multiplication: for any integer $k$, we have $kP = P + P + \cdots + P$
- Discrete logarithm: given $Q \in G_1$, compute $k$ such that $Q = kP$
Discrete logarithm cryptography

- $(G_1, +)$, an additively-written cyclic group of prime order $\#G_1 = \ell$
- $P$, a generator of the group: $G_1 = \langle P \rangle$
- Scalar multiplication: for any integer $k$, we have $kP = P + P + \cdots + P$, $k$ times
- Discrete logarithm: given $Q \in G_1$, compute $k$ such that $Q = kP$
Discrete logarithm cryptography

- $(G_1, +)$, an additively-written cyclic group of prime order $\#G_1 = \ell$
- $P$, a generator of the group: $G_1 = \langle P \rangle$
- Scalar multiplication: for any integer $k$, we have $kP = P + P + \cdots + P$ ($k$ times)
- Discrete logarithm: given $Q \in G_1$, compute $k$ such that $Q = kP$
Discrete logarithm cryptography

- \((G_1, +)\), an additively-written cyclic group of prime order \(\#G_1 = \ell\)
- \(P\), a generator of the group: \(G_1 = \langle P \rangle\)
- Scalar multiplication: for any integer \(k\), we have
  \[ kP = P + P + \cdots + P \]
  \(k\) times

- Discrete logarithm: given \(Q \in G_1\), compute \(k\) such that \(Q = kP\)
Discrete logarithm cryptography

- \((G_1, +)\), an additively-written cyclic group of prime order \(\#G_1 = \ell\)
- \(P\), a generator of the group: \(G_1 = \langle P \rangle\)
- Scalar multiplication: for any integer \(k\), we have \(kP = P + P + \cdots + P\) \(k\) times

- Discrete logarithm: given \(Q \in G_1\), compute \(k\) such that \(Q = kP\)

- We assume that the discrete logarithm problem (DLP) in \(G_1\) is hard
Pairing-based cryptography: Main properties

- $(G_2, \times)$, a multiplicatively-written cyclic group of order $\#G_2 = \#G_1 = \ell$
Pairing-based cryptography: Main properties

- $(\mathbb{G}_2, \times)$, a multiplicatively-written cyclic group of order $\#\mathbb{G}_2 = \#\mathbb{G}_1 = \ell$
- A bilinear pairing on $(\mathbb{G}_1, \mathbb{G}_2)$ is a map

$$\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$$

that satisfies the following conditions:

- **Non-degeneracy**: $\hat{e}(P, P) \neq 1_{\mathbb{G}_2}$ (equivalently $\hat{e}(P, P)$ generates $\mathbb{G}_2$)
- **Bilinearity**: $\hat{e}(Q_1 + Q_2, R) = \hat{e}(Q_1, R) \cdot \hat{e}(Q_2, R)$
- $\hat{e} (Q, R_1 + R_2) = \hat{e}(Q, R_1) \cdot \hat{e}(Q, R_2)$
- **Computability**: $\hat{e}$ can be efficiently computed

Immediate property: for any two integers $k_1$ and $k_2$

$$\hat{e}(k_1 Q, k_2 R) = \hat{e}(Q, R)^{k_1 k_2}$$
Pairing-based cryptography: Main properties

- $(G_2, \times)$, a multiplicatively-written cyclic group of order $\#G_2 = \#G_1 = \ell$
- A bilinear pairing on $(G_1, G_2)$ is a map

\[ \hat{e} : G_1 \times G_1 \rightarrow G_2 \]

that satisfies the following conditions:

- non-degeneracy: $\hat{e}(P, P) \neq 1_{G_2}$ (equivalently $\hat{e}(P, P)$ generates $G_2$)
Pairing-based cryptography: Main properties

- \((G_2, \times)\), a multiplicatively-written cyclic group of order 
  \(#G_2 = #G_1 = \ell\)
- A bilinear pairing on \((G_1, G_2)\) is a map

\[
\hat{e} : G_1 \times G_1 \rightarrow G_2
\]

that satisfies the following conditions:

- non-degeneracy: \(\hat{e}(P, P) \neq 1_{G_2}\) (equivalently \(\hat{e}(P, P)\) generates \(G_2\))
- bilinearity:

\[
\hat{e}(Q_1 + Q_2, R) = \hat{e}(Q_1, R) \cdot \hat{e}(Q_2, R) \quad \hat{e}(Q, R_1 + R_2) = \hat{e}(Q, R_1) \cdot \hat{e}(Q, R_2)
\]
Pairing-based cryptography: Main properties

- \((\mathbb{G}_2, \times)\), a multiplicatively-written cyclic group of order \(\#\mathbb{G}_2 = \#\mathbb{G}_1 = \ell\)
- A bilinear pairing on \((\mathbb{G}_1, \mathbb{G}_2)\) is a map

\[
\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2
\]

that satisfies the following conditions:

- non-degeneracy: \(\hat{e}(P, P) \neq 1_{\mathbb{G}_2}\) (equivalently \(\hat{e}(P, P)\) generates \(\mathbb{G}_2\))
- bilinearity:
  \[
  \hat{e}(Q_1 + Q_2, R) = \hat{e}(Q_1, R) \cdot \hat{e}(Q_2, R) = \hat{e}(Q, R_1 + R_2) = \hat{e}(Q, R_1) \cdot \hat{e}(Q, R_2)
  \]
- computability: \(\hat{e}\) can be efficiently computed
Pairing-based cryptography: Main properties

- $(G_2, \times)$, a multiplicatively-written cyclic group of order $\#G_2 = \#G_1 = \ell$
- A bilinear pairing on $(G_1, G_2)$ is a map

\[ \hat{e} : G_1 \times G_1 \to G_2 \]

that satisfies the following conditions:

- **non-degeneracy:** $\hat{e}(P, P) \neq 1_{G_2}$ (equivalently $\hat{e}(P, P)$ generates $G_2$)
- **bilinearity:**
  \[ \hat{e}(Q_1 + Q_2, R) = \hat{e}(Q_1, R) \cdot \hat{e}(Q_2, R) \]
  \[ \hat{e}(Q, R_1 + R_2) = \hat{e}(Q, R_1) \cdot \hat{e}(Q, R_2) \]
- **computability:** $\hat{e}$ can be efficiently computed

- **Immediate property:** for any two integers $k_1$ and $k_2$

\[ \hat{e}(k_1 Q, k_2 R) = \hat{e}(Q, R)^{k_1 k_2} \]
Pairing-based cryptography: The MOV attack

- At first, used to attack supersingular elliptic curves
    $$\text{DLP}_{G_1} \triangleleft_{P} \text{DLP}_{G_2}$$
    $$dP \rightarrow \hat{e}(dP, P) = \hat{e}(P, P)^d$$
  - for cryptographic applications, we will also require the DLP in $G_2$ to be hard
Pairing-based cryptography: The MOV attack

- At first, used to attack supersingular elliptic curves
    
    \[
    \text{DLP}_{\mathbb{G}_1} \lessdot_P \text{DLP}_{\mathbb{G}_2} \\
    dP \quad \rightarrow \quad \hat{e}(dP, P) = \hat{e}(P, P)^d
    \]

  - for cryptographic applications, we will also require the DLP in $\mathbb{G}_2$ to be hard

- Pairing-based cryptography Sakai-Oghishi-Kasahara, 2000

- One-round three-party key agreement (Joux, 2000)
Pairing-based cryptography: The MOV attack

- At first, used to attack supersingular elliptic curves
    \[
    \begin{align*}
    \text{DLP}_{G_1} & <_P \text{ DLP}_{G_2} \\
    dP & \rightarrow \hat{e}(dP, P) = \hat{e}(P, P)^d
    \end{align*}
    \]
  - for cryptographic applications, we will also require the DLP in \( G_2 \) to be hard
- Pairing-based cryptography Sakai-Oghishi-Kasahara, 2000
- One-round three-party key agreement (Joux, 2000)
- Identity-based encryption
  - Boneh–Franklin, 2001
  - Sakai–Kasahara, 2001
Pairing-based cryptography: The MOV attack

- At first, used to attack supersingular elliptic curves

\[
\text{DLP}_{G_1} \xleftrightarrow[P] \text{DLP}_{G_2}
\]

\[
dP \rightarrow \hat{e}(dP, P) = \hat{e}(P, P)^d
\]

- for cryptographic applications, we will also require the DLP in \(G_2\) to be hard

- Pairing-based cryptography Sakai-Oghishi-Kasahara, 2000
- One-round three-party key agreement (Joux, 2000)
- Identity-based encryption
  - Boneh–Franklin, 2001
  - Sakai–Kasahara, 2001
- Short digital signatures
  - Boneh–Lynn–Shacham, 2001
  - Zang–Safavi-Naini–Susilo, 2004
- ...
Pairing-based cryptography: How to define pairings using elliptic curves

Let us define

- \( \mathbb{F}_q \), a finite field, with \( q = 2^m, 3^m \) or \( p \)
- \( E \), an elliptic curve defined over \( \mathbb{F}_q \)
- \( \ell \), a large prime factor of \( \#E(\mathbb{F}_q) \)
Let us define

- $\mathbb{F}_q$, a finite field, with $q = 2^m$, $3^m$ or $p$
- $E$, an elliptic curve defined over $\mathbb{F}_q$
- $\ell$, a large prime factor of $\#E(\mathbb{F}_q)$

$k$ is the embedding degree, the smallest integer such that $\ell | q^k - 1$

- usually large for ordinary elliptic curves
- bounded in the case of supersingular elliptic curves
  (4 in characteristic 2; 6 in characteristic 3; and 2 in characteristic $> 3$)
Time complexity

I have nothing to do, so I'm trying to calculate the prime factors of the time each minute before it changes.

It was easy when I started at 1:00, but with each hour the number gets bigger.

I wonder how long I can keep up.

253 is 11×23

What?

I'm factoring the time.

borrowed from the xkcd site.
Running time complexity

- The **efficiency** of an algorithm is measured in terms of its input size.
Running time complexity

- The **efficiency** of an algorithm is measured in terms of its input size.
  - For the discrete logarithm problem in $\mathbb{F}_q$, the input size is $O(\log q)$ bits.
Running time complexity

- The **efficiency** of an algorithm is measured in terms of its input size.
  - For the discrete logarithm problem in $\mathbb{F}_q$, the input size is $O(\log q)$ bits.
- A **polynomial-time algorithm** is one whose running time is bounded by a polynomial in the input size: $(\log q)^c$, where $c$ is a constant.
Running time complexity

- The efficiency of an algorithm is measured in terms of its input size.
  - For the discrete logarithm problem in $\mathbb{F}_q$, the input size is $O(\log q)$ bits.
- A polynomial-time algorithm is one whose running time is bounded by a polynomial in the input size: $(\log q)^c$, where $c$ is a constant.
- A fully exponential-time algorithm is one whose running time is of the form $q^c$, where $c$ is a constant.

$\alpha = 0$: polynomial
$\alpha = 1$: fully exponential
$\alpha = 0.5$: subexponential
Running time complexity

- The efficiency of an algorithm is measured in terms of its input size.
  - For the discrete logarithm problem in \( \mathbb{F}_q \), the input size is \( O(\log q) \) bits.
- A polynomial-time algorithm is one whose running time is bounded by a polynomial in the input size: \( (\log q)^c \), where \( c \) is a constant.
- A fully exponential-time algorithm is one whose running time is of the form \( q^c \), where \( c \) is a constant.
- A subexponential-time algorithm as one whose running time is of the form,
  \[
  L_q[\alpha, c] = e^{c(\log q)^\alpha(\log \log q)^{1-\alpha}},
  \]
  where \( 0 < \alpha < 1 \), and \( c \) is a constant.
  \( \alpha = 0 \): polynomial \hspace{1cm} \( \alpha = 1 \): fully exponential
Historic major developments

- **Integer factorization** ($N$)
  - Number field sieve (1990): $L_N[\frac{1}{3}, 1.923]$. 

- **Discrete logarithm over** ($\mathbb{F}_p$)

- **Discrete logarithm over** ($\mathbb{F}_{2^m}$)

- **Elliptic curve discrete logarithm** over ($\mathbb{F}_q$)
  - Pollard (1978): $q^{1/2}$.
Historic major developments

- **Integer factorization** ($N$)
  - Number field sieve (1990): $L_N[\frac{1}{3}, 1.923]$.

- **Discrete logarithm over** ($\mathbb{F}_p$)

---

Francisco Rodríguez-Henríquez

Computing discrete logarithms in the field $\mathbb{F}_{36\cdot509}$ (11 / 37)
Historic major developments

- **Integer factorization (N)**
  - Number field sieve (1990): $L_N[\frac{1}{3}, 1.923]$.

- **Discrete logarithm over ($\mathbb{F}_p$)**

- **Discrete logarithm over ($\mathbb{F}_{2^m}$)**
Historic major developments

- **Integer factorization** ($N$)
  - Number field sieve (1990): $L_N[\frac{1}{3}, 1.923]$.

- **Discrete logarithm over** ($\mathbb{F}_p$)

- **Discrete logarithm over** ($\mathbb{F}_{2^m}$)

- **Elliptic curve discrete logarithm over** ($\mathbb{F}_q$)
  - Pollard (1978): $q^{1/2}$. 
## Recommended key sizes

<table>
<thead>
<tr>
<th>Security in bits</th>
<th>RSA $|N|_2$</th>
<th>DL: $\mathbb{F}_p$ $|p|_2$</th>
<th>DL: $\mathbb{F}_{2^m}$ $m$</th>
<th>ECC $|q|_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>1024</td>
<td>1024</td>
<td>1500</td>
<td>160</td>
</tr>
<tr>
<td>112</td>
<td>2048</td>
<td>2048</td>
<td>3500</td>
<td>224</td>
</tr>
<tr>
<td>128</td>
<td>3072</td>
<td>3072</td>
<td>4800</td>
<td>256</td>
</tr>
<tr>
<td>192</td>
<td>7680</td>
<td>7680</td>
<td>12500</td>
<td>384</td>
</tr>
<tr>
<td>256</td>
<td>15360</td>
<td>15360</td>
<td>25000</td>
<td>512</td>
</tr>
</tbody>
</table>
Pairing-based cryptography: Believed security circa 2012 for supersingular curves

\[ \hat{e} : E(F_q)[\ell] \times E(F_q)[\ell] \rightarrow \mu_\ell \subseteq F_{q^k} \]

- The embedding degree \( k \) depends on the field characteristic \( q \)
Pairing-based cryptography: Believed security circa 2012 for supersingular curves

\[ \hat{e} : E(\mathbb{F}_q)[\ell] \times E(\mathbb{F}_q)[\ell] \rightarrow \mu_\ell \subseteq \mathbb{F}_{q^k} \]

- The embedding degree \( k \) depends on the field characteristic \( q \)

<table>
<thead>
<tr>
<th>Base field ((\mathbb{F}_q))</th>
<th>( \mathbb{F}_{2^m} )</th>
<th>( \mathbb{F}_{2^m} )</th>
<th>( \mathbb{F}_p )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Embedding degree ((k))</td>
<td>4</td>
<td>6</td>
<td>2</td>
</tr>
</tbody>
</table>
Pairing-based cryptography: Believed security circa 2012 for supersingular curves

\[ \hat{e} : E(\mathbb{F}_q)[\ell] \times E(\mathbb{F}_q)[\ell] \rightarrow \mu_\ell \subseteq \mathbb{F}_{q^k}^\times \]

- The embedding degree \( k \) depends on the field characteristic \( q \)

<table>
<thead>
<tr>
<th>Base field ((\mathbb{F}_q))</th>
<th>( \mathbb{F}_{2^m} )</th>
<th>( \mathbb{F}_{2^m} )</th>
<th>( \mathbb{F}_p )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Embedding degree ((k))</td>
<td>4</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>Lower security ((\sim 2^{64}))</td>
<td>( m = 239 )</td>
<td>( m = 97 )</td>
<td>(</td>
</tr>
<tr>
<td>Medium security ((\sim 2^{80}))</td>
<td>( m = 373 )</td>
<td>( m = 163 )</td>
<td>(</td>
</tr>
<tr>
<td>Higher security ((\sim 2^{128}))</td>
<td>( m = 1103 )</td>
<td>( m = 503 )</td>
<td>(</td>
</tr>
</tbody>
</table>
Pairing-based cryptography: Believed security circa 2012 for supersingular curves

\[ \hat{e} : E(\mathbb{F}_q)[\ell] \times E(\mathbb{F}_q)[\ell] \rightarrow \mu_\ell \subseteq \mathbb{F}_{q^k}^{\times} \]

- The embedding degree \( k \) depends on the field characteristic \( q \)

<table>
<thead>
<tr>
<th>Base field ( (\mathbb{F}_q) )</th>
<th>( \mathbb{F}_{2^m} )</th>
<th>( \mathbb{F}_{2^m} )</th>
<th>( \mathbb{F}_p )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Embedding degree ( (k) )</td>
<td>4</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>Lower security (( \sim 2^{64} ))</td>
<td>( m = 239 )</td>
<td>( m = 97 )</td>
<td>(</td>
</tr>
<tr>
<td>Medium security (( \sim 2^{80} ))</td>
<td>( m = 373 )</td>
<td>( m = 163 )</td>
<td>(</td>
</tr>
<tr>
<td>Higher security (( \sim 2^{128} ))</td>
<td>( m = 1103 )</td>
<td>( m = 503 )</td>
<td>(</td>
</tr>
</tbody>
</table>

- \( \mathbb{F}_{2^m} \): simpler finite field arithmetic
Pairing-based cryptography: Believed security circa 2012 for supersingular curves

\[ \hat{e} : E(\mathbb{F}_q)[\ell] \times E(\mathbb{F}_q)[\ell] \to \mu_\ell \subseteq \mathbb{F}_{q^k}^\times \]

- The embedding degree \( k \) depends on the field characteristic \( q \)

<table>
<thead>
<tr>
<th>Base field ((\mathbb{F}_q))</th>
<th>( \mathbb{F}_{2^m} )</th>
<th>( \mathbb{F}_{3^m} )</th>
<th>( \mathbb{F}_p )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Embedding degree ((k))</td>
<td>4</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>Lower security ((\sim 2^{64}))</td>
<td>( m = 239 )</td>
<td>( m = 97 )</td>
<td>(</td>
</tr>
<tr>
<td>Medium security ((\sim 2^{80}))</td>
<td>( m = 373 )</td>
<td>( m = 163 )</td>
<td>(</td>
</tr>
<tr>
<td>Higher security ((\sim 2^{128}))</td>
<td>( m = 1103 )</td>
<td>( m = 503 )</td>
<td>(</td>
</tr>
</tbody>
</table>

- \( \mathbb{F}_{2^m} \): simpler finite field arithmetic
- \( \mathbb{F}_{3^m} \): smaller field extension
Pairing-based cryptography: Believed security circa 2012 for supersingular curves

\[ \hat{e} : E(F_q)[\ell] \times E(F_q)[\ell] \rightarrow \mu_\ell \subseteq F_{q^k} \]

- The embedding degree \( k \) depends on the field characteristic \( q \)

<table>
<thead>
<tr>
<th>Base field (( F_q ))</th>
<th>( F_{2^m} )</th>
<th>( F_{2^m} )</th>
<th>( F_p )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Embedding degree (( k ))</td>
<td>4</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>Lower security (( \sim 2^{64} ))</td>
<td>( m = 239 )</td>
<td>( m = 97 )</td>
<td>(</td>
</tr>
<tr>
<td>Medium security (( \sim 2^{80} ))</td>
<td>( m = 373 )</td>
<td>( m = 163 )</td>
<td>(</td>
</tr>
<tr>
<td>Higher security (( \sim 2^{128} ))</td>
<td>( m = 1103 )</td>
<td>( m = 503 )</td>
<td>(</td>
</tr>
</tbody>
</table>

- \( F_{2^m} \): simpler finite field arithmetic
- \( F_{3^m} \): smaller field extension
- \( F_p \): prohibitive field sizes\[really?\]
Pairing-based cryptography: Believed security circa 2012
for supersingular curves

\[ \hat{e} : E(\mathbb{F}_q)[\ell] \times E(\mathbb{F}_q)[\ell] \to \mu_\ell \subseteq \mathbb{F}_{q^k}^\times \]

- The embedding degree \( k \) depends on the field characteristic \( q \)

<table>
<thead>
<tr>
<th>Base field ( (\mathbb{F}_q) )</th>
<th>( \mathbb{F}_{2^m} )</th>
<th>( \mathbb{F}_{2^m} )</th>
<th>( \mathbb{F}_p )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Embedding degree ( (k) )</td>
<td>4</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>Lower security ( (\sim 2^{64}) )</td>
<td>( m = 239 )</td>
<td>( m = 97 )</td>
<td>(</td>
</tr>
<tr>
<td>Medium security ( (\sim 2^{80}) )</td>
<td>( m = 373 )</td>
<td>( m = 163 )</td>
<td>(</td>
</tr>
<tr>
<td>Higher security ( (\sim 2^{128}) )</td>
<td>( m = 1103 )</td>
<td>( m = 503 )</td>
<td>(</td>
</tr>
</tbody>
</table>

- \( \mathbb{F}_{2^m} \): simpler finite field arithmetic
- \( \mathbb{F}_{3^m} \): smaller field extension
- \( \mathbb{F}_p \): prohibitive field sizes [really?]
Index-Calculus Algorithms for DLP in $\mathbb{F}_{q^n}$

The elements of $\mathbb{F}_{q^n}$ can be viewed as the polynomials of degree at most $n - 1$ in the ring $\mathbb{F}_q[X]$.

Field arithmetic is performed by means of a degree $n$ polynomial whose coefficients are in $\mathbb{F}_q$, irreducible over the base field $\mathbb{F}_q$.

Index-Calculus Algorithms for DLP in $\mathbb{F}_{q^n}$ comprises four main phases:
Index-Calculus Algorithms for DLP in $\mathbb{F}_{q^n}$

The elements of $\mathbb{F}_{q^n}$ can be viewed as the polynomials of degree at most $n - 1$ in the ring $\mathbb{F}_q[X]$. Field arithmetic is performed by means of a degree $n$ polynomial whose coefficients are in $\mathbb{F}_q$, irreducible over the base field $\mathbb{F}_q$.

Index-Calculus Algorithms for DLP in $\mathbb{F}_{q^n}$ comprises four main phases:

1. **Factor base**: Composed by all irreducible polynomials of degree $\leq t$
2. **Relation generation**: Find individual linear relations of the logarithms of factor base elements
3. **Linear system**: Obtain the logarithms of factor base elements by solving a linear system of equations that arises from collecting all the relations found in the previous phase
4. **Descent**: Compute the logarithm of the given element
Attacks on discrete log computation over small char $\mathbb{F}_{q^n}$: Main developments in the last 30+ years

Let $Q$ be defined as $Q = q^n$.

- **Hellman-Reyneri 1982**: Index-calculus $L_Q[\frac{1}{2}, 1.414]$  
- **Coppersmith 1984**: $L_Q[\frac{1}{3}, 1.526]$  
- **Joux-Lercier 2006**: $L_Q[\frac{1}{3}, 1.442]$ when $q$ and $n$ are “balanced”  
- **Hayashi et al. 2012**: Used an improved version of the Joux-Lercier method to compute discrete logs over the field $\mathbb{F}_{36^{36} \cdot 97}$  
- **Joux 2012**: $L_Q[\frac{1}{3}, 0.961]$ when $q$ and $n$ are “balanced”  
- **Joux 2013**: $L_Q[\frac{1}{4} + o(1), c]$ when $Q = q^{2m}$ and $q \approx m$  
- **Göloğlu et al. 2013**: similar to Joux 2013, BPA @ Crypto'2013
Attacks on discrete log computation over small char $\mathbb{F}_{q^n}$: security level consequences

Let us assume that one wants to compute discrete logarithms in the field $\mathbb{F}_{q^n}$, with $q = 3^6$, $n = 509$ Notice that the multiplicative group size of that field is,

$$\#\mathbb{F}_{3^6 \cdot 509} = \lceil \log_2(3) \cdot 6 \cdot 509 \rceil = 4841 \text{ bits}.$$  

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Time complexity</th>
<th>Equivalent bit security level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hellman-Reyneri 1982</td>
<td>$L_Q[\tfrac{1}{2}, 1.414]$</td>
<td>337</td>
</tr>
<tr>
<td>Coppersmith 1984</td>
<td>$L_Q[\tfrac{1}{3}, 1.526]$</td>
<td>134</td>
</tr>
<tr>
<td>Joux-Lercier 2006</td>
<td>$L_Q[\tfrac{1}{3}, 1.442]$</td>
<td>126</td>
</tr>
</tbody>
</table>
2010: The year we make contact
[2010] 2013: The year we make contact
Feb 11 2013 Joux: $\mathbb{F}_{2^{1778}} = \mathbb{F}_{(2^7)^{2.127}}$.
  ▶ 215 CPU hours

Feb 19 2013 Göloğlu et al.: $\mathbb{F}_{2^{1971}} = \mathbb{F}_{(2^9)^{3.73}}$.
  ▶ 3,132 CPU hours

Mar 22 2013 Joux: $\mathbb{F}_{2^{4080}} = \mathbb{F}_{(2^8)^{2.255}}$.
  ▶ 14,100 CPU hours

April 6 2013, Barbulescu et al.: $\mathbb{F}_{2^{809}}$, 
  ▶ notice that 809 is a prime number.
  ▶ using conventional techniques based on the Coppersmith algorithm
  ▶ 30,000+ CPU hours

Apr 11 2013 Göloğlu et al.: $\mathbb{F}_{2^{6120}} = \mathbb{F}_{(2^8)^{3.255}}$.
  ▶ 750 CPU hours

May 21 2013 Joux: $\mathbb{F}_{2^{6168}} = \mathbb{F}_{(2^8)^{3.257}}$.
  ▶ 550 CPU hours
Let $q$ be a prime power, and let $n \leq q + 2$. The DLP in $\mathbb{F}_{q^{2 \cdot n}}$ can be solved in time $q^{O(\log n)}$.

In the case where $n \approx q$, the DLP in $\mathbb{F}_{q^{2 \cdot n}} = \mathbb{F}_Q$ can be solved in time, \[
\log Q^{O(\log \log Q)}
\]

This is smaller than $L_Q[\alpha, c]$ for any $\alpha > 0$ and $c > 0$. 

(Barbulescu-Gaudry-Joux-Thomé)
Cryptographic implications

PJCrypto: Post-Joux Cryptography

1. Discrete log cryptography
2. Pairing-based cryptography
3. Elliptic curve cryptography
Discrete log cryptography

Diffie-Hellman, ElGamal, DSA, ...

- DL cryptography over $\mathbb{F}_p$ is not affected.
- DL cryptography over $\mathbb{F}_{2^m}$, $m$ prime, might be affected.
- Note that $\mathbb{F}_{2^m}$ can be embedded in $\mathbb{F}_{2^{\ell m}}$ for any $\ell \geq 2$.
  - $\mathbb{F}_{2^{809}}$ can be embedded in $\mathbb{F}_{2^{10 \cdot 2 \cdot 809}}$. It is unlikely that the new algorithms will be faster in this larger field.
Pairing-based cryptography

Efficient discrete log algorithms in small char $\mathbb{F}_{q^n}$ fields have a direct negative impact on the security level that small characteristic symmetric pairings can offer:

1. Supersingular elliptic curves over $\mathbb{F}_{2^n}$ with embedding degree $k = 4$
2. Supersingular elliptic curves over $\mathbb{F}_{3^n}$ with embedding degree $k = 6$
3. Supersingular genus-two curves over $\mathbb{F}_{2^n}$ with embedding degree $k = 12$
4. Elliptic curves over $\mathbb{F}_p$ with embedding degree $k = 2$
5. BN curves: Elliptic curves over $\mathbb{F}_p$ with embedding degree $k = 12$

Curves 1, 2 and 3 are potentially vulnerable to the new attacks. Curves 4 and 5 are not affected by the new attacks.
Pairing-based cryptography

Efficient discrete log algorithms in small char $\mathbb{F}_{q^n}$ fields have a direct negative impact on the security level that small characteristic symmetric pairings can offer:

1. Supersingular elliptic curves over $\mathbb{F}_{2^n}$ with embedding degree $k = 4$
2. Supersingular elliptic curves over $\mathbb{F}_{3^n}$ with embedding degree $k = 6$
3. Supersingular genus-two curves over $\mathbb{F}_{2^n}$ with embedding degree $k = 12$
4. Elliptic curves over $\mathbb{F}_p$ with embedding degree $k = 2$
5. BN curves: Elliptic curves over $\mathbb{F}_p$ with embedding degree $k = 12$

Curves 1, 2 and 3 are potentially vulnerable to the new attacks. Curves 4 and 5 are not affected by the new attacks.
Example: Consider the supersingular elliptic curve, \( Y^2 = X^3 - X + 1 \), with \( \#E(\mathbb{F}_{3509}) = 7r \), and where, \( r = (3^{509} - 3^{255} + 1)/7 \) is an 804-bit prime.
Pairing-based cryptography

Example: Consider the supersingular elliptic curve, $Y^2 = X^3 - X + 1$, with $\#E(\mathbb{F}_{3^{509}}) = 7r$, and where, $r = (3^{509} - 3^{255} + 1)/7$ is an 804-bit prime.

- $E$ has embedding degree $k = 6$
- The elliptic curve group $E(\mathbb{F}_{3^{509}})$ can be efficiently embedded in $\mathbb{F}_{3^{6\cdot509}}$
- **Question**: Can logarithms in $\mathbb{F}_{3^{6\cdot509}}$ be efficiently computed using the new algorithms? Or, at least significantly faster than the previously-known algorithms?
Example: Consider the supersingular elliptic curve, \( Y^2 = X^3 - X + 1 \), with \( \#E(\mathbb{F}_{3^{509}}) = 7r \), and where, \( r = (3^{509} - 3^{255} + 1)/7 \) is an 804-bit prime.

- E has embedding degree \( k = 6 \)
- The elliptic curve group \( E(\mathbb{F}_{3^{509}}) \) can be efficiently embedded in \( \mathbb{F}_{3^{6} \cdot 509} \)
- Question: Can logarithms in \( \mathbb{F}_{3^{6} \cdot 509} \) be efficiently computed using the new algorithms? Or, at least significantly faster than the previously-known algorithms?
- Note: \( \mathbb{F}_{3^{6} \cdot 509} \) can be embedded in \( \mathbb{F}_{3^{6} \cdot 2 \cdot 509} \)
Elliptic curve cryptography

- The recent advances do not affect the security of (ordinary) elliptic curve cryptosystems.
- **Example:** NIST elliptic curve K-163: 
  \[ E : Y^2 + XY = X^3 + X^2 + 1 \text{ over } \mathbb{F}_{2^{163}} \]
  \[ E(\mathbb{F}_{2^{163}}) \text{ can be embedded in } \mathbb{F}_{2^{163} \cdot 2^{17932535427373041941149514581590332356837787037}} \]
  Elements in this large field are 
  \[ 5846006549323611672814741753598448348329118574062 \approx 2^{163} \] bits in length.
Elliptic curve cryptography

- The recent advances do not affect the security of (ordinary) elliptic curve cryptosystems.
- **Example:** NIST elliptic curve K-163:
  \[ E : Y^2 + XY = X^3 + X^2 + 1 \] over \( \mathbb{F}_{2^{163}} \). \( E(\mathbb{F}_{2^{163}}) \) can be embedded in \( \mathbb{F}_{2^{163} \cdot 2 \cdot 17932535427373041941149514581590332356837787037} \).
  Elements in this large field are
  \[ 5846006549323611672814741753598448348329118574062 \approx 2^{163} \] bits in length.

- The **Eddington number**, \( N_{Edd} \), is the “provable” number of protons in the observable universe estimated as, \( N_{Edd} = 136 \cdot 2^{256} \).
A mainstream belief in the crypto community

- Several records broken in rapid succession by Joux, Göloğlu et al. and the Caramel team, the last of the series as of today: a discrete log computation over $\mathbb{F}_{2^{6128}} = \mathbb{F}_{(2^8)^{3.257}}$ Joux (May 21, 2013)

As a consequence of these astonishing results, a mainstream belief in the crypto community is that small characteristic symmetric pairings are broken, both in theory and in practice. More than that, some distinguished researchers have expressed in blogs/chats the opinion that all these new developments may sooner or later bring fatal consequences for integer factorization, which eventually would lead to the death of RSA.

Nevertheless, none of the records mentioned above have attacked finite field extensions that have been previously proposed for performing pairing-based cryptography in small char.
A mainstream belief in the crypto community

- Several records broken in rapid succession by Joux, Göloğlu et al. and the Caramel team, the last of the series as of today: a discrete log computation over $\mathbb{F}_{2^{6128}} = \mathbb{F}_{(2^8)^{3.257}}$ Joux (May 21, 2013)
- As a consequence of these astonishing results, a mainstream belief in the crypto community is that small characteristic symmetric pairings are broken, both in theory and in practice
A mainstream belief in the crypto community

- Several records broken in rapid succession by Joux, Göloğlu et al. and the Caramel team, the last of the series as of today: a discrete log computation over $\mathbb{F}_{2^{6128}} = \mathbb{F}_{(2^8)^{3 \cdot 257}}$ Joux (May 21, 2013)

- As a consequence of these astonishing results, a mainstream belief in the crypto community is that small characteristic symmetric pairings are broken, both in theory and in practice.

- More than that, some distinguished researchers have expressed in blogs/chats the opinion that all these new developments may sooner or later bring fatal consequences for integer factorization, which eventually would lead to the death of RSA.
A mainstream belief in the crypto community

- Several records broken in rapid succession by Joux, Göloğlu et al. and the Caramel team, the last of the series as of today: a discrete log computation over $F_{2^{6128}} = F_{(2^8)^3 \cdot 2^{257}}$ Joux (May 21, 2013)

- As a consequence of these astonishing results, a mainstream belief in the crypto community is that small characteristic symmetric pairings are broken, both in theory and in practice

- More than that, some distinguished researchers have expressed in blogs/chats the opinion that all these new developments may sooner or later bring fatal consequences for integer factorization, which eventually would lead to the death of RSA

- Nevertheless, none of the records mentioned above have attacked finite field extensions that have been previously proposed for performing pairing-based cryptography in small char
Our question: can the new attacks or a combination of them be effectively applied to compute discrete logs in finite field extensions of interest in pairing-based cryptography?
Discrete log descent

discrete log descent
Computing discrete logarithms in $\mathbb{F}_{3^{6} \cdot 509}$

- We present a **concrete analysis** of the DLP algorithm for computing discrete logarithms in $\mathbb{F}_{3^{6} \cdot 509}$.
We present a **concrete analysis** of the DLP algorithm for computing discrete logarithms in $\mathbb{F}_{3^6 \cdot 509}$.

In fact, this field is **embedded** in the quadratic extension field $\mathbb{F}_{3^{12} \cdot 509}$, and it is in this latter field where the DLP algorithm is executed.

Thus, we have $q = 3^6 = 729$, $n = 509$, and the size of the group is $N = 3^{12 \cdot 509} - 1$. Note that $3^{12 \cdot 509} \approx 2^{9681}$.

We wish to find $\log_g h$, where $g$ is a generator of $\mathbb{F}_{3^{12} \cdot 509}^*$ and $h \in \mathbb{F}_{3^{12} \cdot 509}^*$. 
We present a concrete analysis of the DLP algorithm for computing discrete logarithms in $\mathbb{F}_{36 \cdot 509}$.

In fact, this field is embedded in the quadratic extension field $\mathbb{F}_{312 \cdot 509}$, and it is in this latter field where the DLP algorithm is executed.

Thus, we have $q = 3^6 = 729$, $n = 509$, and the size of the group is $N = 3^{12 \cdot 509} - 1$. Note that $3^{12 \cdot 509} \approx 2^{9681}$.

We wish to find $\log_g h$, where $g$ is a generator of $\mathbb{F}_{312 \cdot 509}^*$ and $h \in \mathbb{F}_{312 \cdot 509}^*$.

Once again, this field was selected to attack the elliptic curve discrete logarithm problem in $E(\mathbb{F}_{3509})$, where $E$ is the supersingular elliptic curve $Y^2 = X^3 - X + 1$ with $#E(\mathbb{F}_{3509}) = 7r$, and where $r = (3^{509} - 3^{255} + 1)/7$ is an 804-bit prime.
Computing discrete logarithms in $\mathbb{F}_{36.509}$: Main steps

Our attack was divided in three main steps

- Finding logarithms of linear polynomials
- Finding logarithms of irreducible quadratic polynomials
- Descent, divided into four different strategies:
  - Continued-fraction descent
  - Classical descent
  - QPA descent
  - Gröbner bases descent
Computing discrete logarithms in $\mathbb{F}_{36 \cdot 509}$: Main steps

Our attack was divided in three main steps:

- Finding logarithms of linear polynomials
- Finding logarithms of irreducible quadratic polynomials
- Descent, divided into four different strategies:
  1. Continued-fraction descent
  2. Classical descent
  3. QPA descent
  4. Gröbner bases descent
The factor base for linear polynomials $B_1$ has size $3^{12} \approx 2^{19}$.

- The cost of relation generation is approximately $2^{30} M_{q^2}$,
- The cost of the linear algebra is approximately $2^{48} M_r$,

where $M_{q^2}$ and $M_r$ stands for field multiplication in the field $\mathbb{F}_{q^2}$ and $\mathbb{F}_r$, respectively.
Finding logarithms of linear polynomials

- The factor base for linear polynomials $B_1$ has size $3^{12} \approx 2^{19}$.
  - The cost of relation generation is approximately $2^{30} M_{q^2}$,
  - The cost of the linear algebra is approximately $2^{48} M_r$,

where $M_{q^2}$ and $M_r$ stands for field multiplication in the field $\mathbb{F}_{q^2}$ and $\mathbb{F}_r$, respectively.

- Note that relation generation can be effectively parallelized, unlike the linear algebra where parallelization on conventional computers provides relatively small benefits.
Finding logarithms of irreducible quadratic polynomials

- Let $u \in \mathbb{F}_{q^2}$, and let $Q(X) = X^2 + uX + v \in \mathbb{F}_{q^2}[X]$ be an irreducible quadratic.
  - Define $B_{2,u}$ to be the set of all irreducible quadratics of the form $X^2 + uX + w$ in $\mathbb{F}_{q^2}[X]$. 

One expects that $|B_{2,u}| \approx \frac{(q^2 - 1)}{2}$.

The logarithms of all elements in $B_{2,u}$ are found simultaneously using one application of QPA descent.

For each $u \in \mathbb{F}_{36509}$, the expected cost of computing logarithms of all quadratics in $B_{2,u}$ is $2^{39} M_{q^2}$ for relation generation, and $2^{48} M_r$ for the linear algebra.

This step is somewhat parallelizable on conventional computers since each set $B_{2,u}$ can be handled by a different processor.
Finding logarithms of irreducible quadratic polynomials

- Let $u \in \mathbb{F}_{q^2}$, and let $Q(X) = X^2 + uX + v \in \mathbb{F}_{q^2}[X]$ be an irreducible quadratic.
  - Define $\mathcal{B}_{2,u}$ to be the set of all irreducible quadratics of the form $X^2 + uX + w$ in $\mathbb{F}_{q^2}[X]$
  - one expects that $\#\mathcal{B}_{2,u} \approx (q^2 - 1)/2$
Finding logarithms of irreducible quadratic polynomials

Let $u \in \mathbb{F}_{q^2}$, and let $Q(X) = X^2 + uX + v \in \mathbb{F}_{q^2}[X]$ be an irreducible quadratic.

- Define $\mathcal{B}_{2,u}$ to be the set of all irreducible quadratics of the form $X^2 + uX + w$ in $\mathbb{F}_{q^2}[X]$.
- One expects that $\#\mathcal{B}_{2,u} \approx (q^2 - 1)/2$.
- The logarithms of all elements in $\mathcal{B}_{2,u}$ are found simultaneously using one application of QPA descent.
Finding logarithms of irreducible quadratic polynomials

Let \( u \in \mathbb{F}_{q^2} \), and let \( Q(X) = X^2 + uX + v \in \mathbb{F}_{q^2}[X] \) be an irreducible quadratic.

- Define \( \mathcal{B}_{2,u} \) to be the set of all irreducible quadratics of the form \( X^2 + uX + w \) in \( \mathbb{F}_{q^2}[X] \)
- one expects that \( \#\mathcal{B}_{2,u} \approx (q^2 - 1)/2 \)
- The logarithms of all elements in \( \mathcal{B}_{2,u} \) are found simultaneously using one application of QPA descent

For each \( u \in \mathbb{F}_{312} \), the expected cost of computing logarithms of all quadratics in \( \mathcal{B}_{2,u} \) is \( 2^{39} M_{q^2} \) for relation generation, and \( 2^{48} M_r \) for the linear algebra.
Finding logarithms of irreducible quadratic polynomials

- Let $u \in \mathbb{F}_{q^2}$, and let $Q(X) = X^2 + uX + v \in \mathbb{F}_{q^2}[X]$ be an irreducible quadratic.
  - Define $\mathcal{B}_{2,u}$ to be the set of all irreducible quadratics of the form $X^2 + uX + w$ in $\mathbb{F}_{q^2}[X]$.
  - One expects that $\# \mathcal{B}_{2,u} \approx (q^2 - 1)/2$.
  - The logarithms of all elements in $\mathcal{B}_{2,u}$ are found simultaneously using one application of QPA descent.

- For each $u \in \mathbb{F}_{3^{12}}$, the expected cost of computing logarithms of all quadratics in $\mathcal{B}_{2,u}$ is $2^{39} M_{q^2}$ for relation generation, and $2^{48} M_r$ for the linear algebra.

- This step is somewhat parallelizable on conventional computers since each set $\mathcal{B}_{2,u}$ can be handled by a different processor.
Descent: General approach

- Recall that we wish to compute $\log_g h$, where $h \in \mathbb{F}_{q^n} = \mathbb{F}_{q^2}[X]/(lX)$. We assume that $\deg h = n - 1$. 

- The descent stage begins by multiplying $h$ by a random power of $g$, namely, $h' = h \cdot g^i$ for some $i \in \mathbb{F}_r$. 

- The descent algorithm gives $\log_g h'$ as a linear combination of logarithms of polynomials of degree at most two using the combination of four different strategies.
Recall that we wish to compute $\log_g h$, where $h \in \mathbb{F}_{q^n} = \mathbb{F}_{q^2}[X]/(l_X)$. We assume that $\deg h = n - 1$.

The descent stage begins by multiplying $h$ by a random power of $g$, namely, $h' = h \cdot g^i$ for some $i \in \mathbb{F}_r$. 

The descent algorithm gives $\log_g h'$ as a linear combination of logarithms of polynomials of degree at most two using the combination of four different strategies.
Recall that we wish to compute $\log_g h$, where $h \in \mathbb{F}_{q^{2n}} = \mathbb{F}_{q^2}[X]/(lX)$. We assume that $\deg h = n - 1$.

The descent stage begins by multiplying $h$ by a random power of $g$, namely, $h' = h \cdot g^i$ for some $i \in \mathbb{F}_r$.

The descent algorithm gives $\log_g h'$ as a linear combination of logarithms of polynomials of degree at most two using the combination of four different strategies.
A descent into four steps

1. **Continued-fraction descent**: Starting from a polynomial of degree $n = 508$ gives its discrete log as a linear combination of logarithms of polynomials of degree at most $m = 30$
A descent into four steps

1. **Continued-fraction descent**: Starting from a polynomial of degree $n = 508$ gives its discrete log as a linear combination of logarithms of polynomials of degree at most $m = 30$

2. **Classical descent**: given the degree-30 polynomials of the previous step, finds their discrete log as a linear combination of logarithms of polynomials of degree at most 11 (using two applications of this strategy)
A descent into four steps

1. **Continued-fraction descent**: Starting from a polynomial of degree \( n = 508 \) gives its discrete log as a linear combination of logarithms of polynomials of degree at most \( m = 30 \).

2. **Classical descent**: given the degree-30 polynomials of the previous step, finds their discrete log as a linear combination of logarithms of polynomials of degree at most 11 (using two applications of this strategy).

3. **QPA descent**: given the degree-11 polynomials of the previous step, finds their discrete log as a linear combination of logarithms of polynomials of degree at most 7.
A descent into four steps

1. **Continued-fraction descent:** Starting from a polynomial of degree \( n = 508 \) gives its discrete log as a linear combination of logarithms of polynomials of degree at most \( m = 30 \)

2. **Classical descent:** given the degree-30 polynomials of the previous step, finds their discrete log as a linear combination of logarithms of polynomials of degree at most 11 (using two applications of this strategy)

3. **QPA descent:** given the degree-11 polynomials of the previous step, finds their discrete log as a linear combination of logarithms of polynomials of degree at most 7

4. **Gröbner bases descent:** given the degree-7 polynomials of the previous step, finds their discrete log as a linear combination of logarithms of quadratic polynomials. This concludes the descent
A positive answer: Announcing the weak field $\mathbb{F}_{36 \cdot 509}$

<table>
<thead>
<tr>
<th>Finding logarithms of linear polynomials</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Relation generation</td>
<td>$2^{22} M_r$</td>
</tr>
<tr>
<td>Linear algebra</td>
<td>$2^{48} M_r$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Finding logarithms of irreducible quadratic polynomials</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Relation generation</td>
<td>$2^{50} M_r$</td>
</tr>
<tr>
<td>Linear algebra</td>
<td>$2^{67} M_r$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Descent</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Continued-fraction (254 to 30)</td>
<td>$2^{71} M_r$</td>
</tr>
<tr>
<td>Classical (30 to 15)</td>
<td>$2^{71} M_r$</td>
</tr>
<tr>
<td>Classical (15 to 11)</td>
<td>$2^{73} M_r$</td>
</tr>
<tr>
<td>QPA (11 to 7)</td>
<td>$2^{63} M_r$</td>
</tr>
<tr>
<td>Gröbner bases (7 to 4)</td>
<td>$2^{65} M_r$</td>
</tr>
<tr>
<td>Gröbner bases (4 to 3)</td>
<td>$2^{64} M_r$</td>
</tr>
<tr>
<td>Gröbner bases (3 to 2)</td>
<td>$2^{69} M_r$</td>
</tr>
</tbody>
</table>

**Table:** Estimated costs of the main steps of the new DLP algorithm for computing discrete logarithms in $\mathbb{F}_{(3^6)^2 \cdot 509}$. $M_r$ denotes the costs of a multiplication modulo the 804-bit prime $r = (3^{509} - 3^{255} + 1)/7$. We also assume that $2^{22}$ multiplications modulo $r$ can be performed in 1 second.
Descent path for a polynomial of degree \( \leq 508 \) over \( \mathbb{F}_{36^2} \)

- 254 (2)
  - Continued fraction descent
  - Time: \( 2^{71} M_r \)

- 30 (30)
  - Classical descent
  - Time: \( 2^{71} M_r \)

- 15 (870)
  - Classical descent
  - Time: \( 2^{73} M_r \)

- 11 (23,490)
  - QPA descent
  - Time: \( 2^{63} M_r \)

- 7 (\( 2^{37} \))
  - Gröbner bases descent
  - Time: \( 2^{65} M_r \)

- 4 (\( 2^{47} \))
  - Gröbner bases descent
  - Time: \( 2^{64} M_r \)

- 3 (\( 2^{55.5} \))
  - Gröbner bases descent
  - Time: \( 2^{69} M_r \)

- 2

The numbers in parentheses are the expected number of nodes at that level. 'Time' is the expected time to generate all nodes at a level.
Descent path for a polynomial of degree $\leq 508$ over $\mathbb{F}_{36^2}$

- 254 (2)
  - Continued fraction descent
  - Time: $2^{71} M_r$

- 30 (30)
  - Classical descent
  - Time: $2^{71} M_r$

- 15 (870)
  - Classical descent
  - Time: $2^{73} M_r$

- 11 (23,490)
  - QPA descent
  - Time: $2^{63} M_r$

- 7 ($2^{37}$)
  - Gröbner bases descent
  - Time: $2^{65} M_r$

- 4 ($2^{47}$)
  - Gröbner bases descent
  - Time: $2^{64} M_r$

- 3 ($2^{55.5}$)
  - Gröbner bases descent
  - Time: $2^{69} M_r$

- 2

The numbers in parentheses are the expected number of nodes at that level. 'Time' is the expected time to generate all nodes at a level.

All the technical details are discussed in the eprint report 2013/446.
Revisiting fields of pairing interest, the authors in the eprint report 2013/446, find that the running time of computing discrete logs has complexity,

$$L_Q \left( \frac{1}{3}, \left[ \frac{64}{9} \cdot \frac{(\lambda + 1)}{\lambda} \right]^{1/3} \right),$$

where $\lambda$ is the degree of the polynomial that defines the field characteristic $p$ (usually, $\lambda \leq 10$).

For fields of pairing interest where $p$ is 'large' the complexity of the attack drops to,

$$L_Q \left( \frac{1}{3}, \left[ \frac{32}{9} \cdot \frac{(\lambda + 1)}{\lambda} \right]^{1/3} \right),$$

and even to, $L_Q \left( \frac{1}{3}, \left[ \frac{32}{9} \right]^{1/3} \right)$, for some large 'low-weight' primes with low embedding degree $k$. 

---

Francisco Rodríguez-Henríquez
Revisiting fields of pairing interest, the authors in the eprint report 2013/446, find that the running time of computing discrete logs has complexity,

$$L_Q\left(\frac{1}{3}, \left[\frac{64}{9} \cdot \frac{\lambda + 1}{\lambda}\right]^{1/3}\right),$$

where $\lambda$ is the degree of the polynomial that defines the field characteristic $p$ (usually, $\lambda \leq 10$)

- For fields of pairing interest where $p$ is 'large' the complexity of the attack drops to,

$$L_Q\left(\frac{1}{3}, \left[\frac{32}{9} \cdot \frac{\lambda + 1}{\lambda}\right]^{1/3}\right),$$

and even to, $L_Q\left(\frac{1}{3}, \left[\frac{32}{9}\right]^{1/3}\right).$ for some large 'low-weight' primes with low embedding degree $k$.

The analysis is asymptotic. In particular, this attack does not affect the 128-bit security level parameters used for the curves of class 5 in slide 21.
In his ECC’2013 talk, Robert Granger announced a refined version of the attack described in this presentation.
In his ECC’2013 talk, Robert Granger announced a refined version of the attack described in this presentation.

This allows him to report several more weak fields in characteristic two, including, $\mathbb{F}_{24\cdot1223}$, a field that not long ago was assumed to offer a security level of 128 bits.
Merci-Thanks-Obrigado-Gracias
for your attention

borrowed from Quino.

Questions?