Chapter 4: The Intractable Mathematical Problems
Public-key cryptography is based on **trapdoor one-way functions**. It should be easy to encrypt a message or verify a signature, but inverting the transform (decryption or signature generation) should be difficult, unless some secret information (the trapdoor) is known.

Some difficult computational problems
- Factoring composite integers
- Computing square roots modulo a composite integer
- Computing discrete logarithms in certain groups (finite fields, elliptic hyperelliptic curves, class groups of number fields, and so on)
- Finding shortest/closest vectors in a lattice
- Solving the subset sum problem
- Finding roots of non-linear multivariate polynomials
- Solving the braid conjugacy problem
Many sophisticated algorithms are proposed to break the trapdoor functions. Most of these are fully exponential. **Subexponential algorithms** are sometimes known.

For suitably chosen domain parameters, these algorithms take infeasible time.

No non-trivial lower bounds on the complexity of these computational problems are known. Even existence of polynomial-time algorithms cannot be often ruled out.

Certain special cases have been discovered to be cryptographically weak. For practical designs, it is essential to avoid these special cases.

Polynomial-time quantum algorithms are known for factoring integers and computing discrete logarithms in finite fields.
Discrete Logarithms

Let $\mathbb{F}_q$ be a finite field, $g$ a generator of $\mathbb{F}_q^*$, and $a \in \mathbb{F}_q^*$. There exists a unique integer $x \in \{0, 1, 2, \ldots, q - 1\}$ such that $a = g^x$. We call $x$ the index or discrete logarithm of $a$ to the base $g$. We denote this by $x = \text{ind}_g a$.

Indices follow arithmetic modulo $q - 1$.

\[
\text{ind}_g(ab) \equiv \text{ind}_g a + \text{ind}_g b \pmod{q - 1},
\]
\[
\text{ind}_g(a^e) \equiv e \text{ind}_g a \pmod{q - 1}.
\]

The concept of discrete logarithms can be extended to other finite groups (including the elliptic curve group).
Discrete Logarithm: Example

- Take $p = 17$ and $g = 3$.

\[
\begin{array}{cccccccccccccccc}
\text{a} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 \\
\text{ind}_3(a) & 0 & 14 & 1 & 12 & 5 & 15 & 11 & 10 & 2 & 3 & 7 & 13 & 4 & 9 & 6 & 8 \\
\end{array}
\]

- $\text{ind}_3 6 = 15$ and $\text{ind}_3 11 = 7$. Since $6 \times 11 = 15 \pmod{17}$, we have $\text{ind}_3 15 \equiv \text{ind}_3 6 + \text{ind}_3 11 \equiv 15 + 7 \equiv 6 \pmod{16}$. 
The Most Common Intractable Problems

**Integer factorization problem (IFP):** Given $n \in \mathbb{N}$, compute the complete prime factorization of $n$. Suppose there is an algorithm $A$ that computes a non-trivial factor of $n$. We can use $A$ repeatedly in order to compute the complete factorization of $n$. If $n = pq$ (with $p, q \in \mathbb{P}$), then computing $p$ or $q$ suffices.

**Example**

**Input:** $n = 85067$.
**Output:** $85067 = 257 \times 331$.

**Discrete logarithm problem (DLP):** Let $g$ be a generator of $\mathbb{F}_q^*$. Given $a \in \mathbb{F}_q^*$, compute $\text{ind}_g a$.

**Example**

**Input:** $p = 17$, $g = 3$, $a = 11$.
**Output:** $\text{ind}_g a = 7$. 
The Most Common Intractable Problems (contd)

- IFP and DLP are believed to be computationally difficult.
- The best known algorithms for IFP and DLP are subexponential.
- IFP is the inverse of the integer multiplication problem.
- DLP is the inverse of the modular exponentiation problem.
- Integer multiplication and modular exponentiation are easy computational problems. They are believed to be one-way functions.
- There is, however, no proof that IFP and DLP must be difficult.
- Efficient quantum algorithms exist for solving IFP and DLP.
- IFP and DLP are believed to be computationally equivalent.
Intractable Problems: Variants

- **Diffie-Hellman problem (DHP):** Let \( g \) be a generator of \( \mathbb{F}_q^* \). Given the elements \( g^x \) and \( g^y \) of \( \mathbb{F}_q^* \), compute \( g^{xy} \).

**Example**
- **Input:** \( p = 17, g = 3, g^x \equiv 11 \pmod{p} \) and \( g^y \equiv 13 \pmod{p} \).
- **Output:** \( g^{xy} \equiv 4 \pmod{p} \).

\( (x = 7, y = 4, \text{ that is, } xy \equiv 28 \equiv 12 \pmod{p-1}, \text{ that is, } g^{xy} \equiv 3^{12} \equiv 4 \pmod{p}.) \)

- DHP is another believably difficult computational problem.
- If DLP can be solved, then DHP can be solved \( (g^{xy} = (g^x)^y) \).
- The converse is only believed to be true.
Intractable Problems: Variants

- **Elliptic Curve Discrete Logarithm Problem (ECDLP)**
  Let $E$ be an elliptic curve defined over the finite field $\mathbb{F}_q$. Given points $P$ and $xP$ in $E(\mathbb{F}_q)$, compute $x$.

- **Elliptic Curve Diffie-Hellman Problem (ECDHP)**
  Let $E$ be an elliptic curve defined over the finite field $\mathbb{F}_q$. Given points $xP$ and $yP$ in $E(\mathbb{F}_q)$, compute the point $xyP$.

- **Example**
  - Consider the curve $E : y^2 = x^3 + x + 3$ defined over $\mathbb{F}_7$.
  - $E(\mathbb{F}_7)$ is cyclic of order 6.
  - $P = (4, 1)$ is a generator of $E(\mathbb{F}_7)$.
  - The index of $Q = (5, 0)$ to the base $P$ is 3, that is, $Q = 3P$.
  - Let $Q_1 = 3P = (5, 0)$ and $Q_2 = 4P = (6, 1)$. Then, $(3 \times 4)P = 12P = 0P = \mathcal{O}$. 
Intractable Problems: Variants

- All finite (Abelian) groups are not cyclic.
- Although all $\mathbb{F}_q^*$ are cyclic, all elliptic curve groups $E(\mathbb{F}_q)$ are not cyclic.
- Even if $G$ is a cyclic group, a generator of $G$ may be unknown.
- Computing a generator of $\mathbb{F}_q^*$ requires the complete factorization of $q - 1$.

**Generalized Discrete Logarithm Problem (GDLP)**

Let $G$ be a (multiplicative) Abelian group of size $n$ and let $g$ be an element of $G$ of order $m$ (we have $m | n$). Let $H$ be the subgroup of $G$ generated by $g$. Given $a \in G$, determine whether $a \in H$, and if so, determine the unique integer $x \in \{0, 1, 2, \ldots, m - 1\}$ such that $a = g^x$. 

Public-key Cryptography: Theory and Practice

Abhijit Das
GDLP: Example

Example
- Take $p = 661$ and $g = 29$. We have $\text{ord}_p(g) = 66$.
- We have $g^{15} \equiv 49 \pmod{p}$, that is, $\text{ind}_g(49) = 15$.
- $\text{ind}_g(94)$ does not exist.

If $G$ is cyclic, then $a \in H$ if and only if $a^m = e$.

Example
- $49^{66} \equiv 1 \pmod{661}$.
- $94^{66} \equiv -1 \pmod{661}$. 

Public-key Cryptography: Theory and Practice
Abhijit Das
The Integer Factorization Problem (IFP)

Let \( n \) be the integer to be factored.

**Older algorithms**

- Trial division (efficient if all prime divisors of \( n \) are small)
- Pollard’s rho method
- Pollard’s \( p - 1 \) method (efficient if \( p - 1 \) has only small prime factors for some prime divisor \( p \) of \( n \))
- Williams’ \( p + 1 \) method (efficient if \( p + 1 \) has only small prime factors for some prime divisor \( p \) of \( n \))

In the worst case, these algorithms take exponential (in \( \log n \)) running time.
Modern Factoring Algorithms

Subexponential running time:
\[ L(n, \omega, c) = \exp \left[ (c + o(1))(\ln n)^\omega (\ln \ln n)^{1-\omega} \right] \]

\( \omega = 0 \) : \( L(n, \omega, c) \) is polynomial in \( \ln n \).
\( \omega = 1 \) : \( L(n, \omega, c) \) is exponential in \( \ln n \).
\( 0 < \omega < 1 \) : \( L(n, \omega, c) \) is between polynomial and exponential.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Inventor(s)</th>
<th>Running time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continued fraction method (CFRAC)</td>
<td>Morrison &amp; Brillhart (1975)</td>
<td>( L(n, 1/2, c) )</td>
</tr>
<tr>
<td>Quadratic sieve method (QSM)</td>
<td>Pomerance (1984)</td>
<td>( L(n, 1/2, 1) )</td>
</tr>
<tr>
<td>Cubic sieve method (CSM)</td>
<td>Reyneri</td>
<td>( L(n, 1/2, 0.816) )</td>
</tr>
<tr>
<td>Elliptic curve method (ECM)</td>
<td>H. W. Lenstra (1987)</td>
<td>( L(n, 1/2, c) )</td>
</tr>
<tr>
<td>Number field sieve method (NFSM)</td>
<td>A. K. Lenstra, H. W. Lenstra, Manasse &amp; Pollard (1990)</td>
<td>( L(n, 1/3, 1.923) )</td>
</tr>
</tbody>
</table>
Trial Division

- Divide \( n \) by 2, 3, 4, 5, \ldots, \( \lfloor \sqrt{n} \rfloor \).
- It suffices to divide only by primes in the above range.
- A list of primes may be unavailable.
- Checking trial divisors for primality is time-consuming.
- Divide \( n \) by \( d \geq 30 \) if and only if \( d \equiv 1, 7, 11, 13, 17, 19, 23, 29 \pmod{30} \).

**Example**
- Take \( n = 1716617 \).
- Make trial divisions by primes \( \leq 30 \). Factor found: \( 7^2 \).
- Make trial divisions by 31, 37, 41, 43, 47, 49. No factor found.
- Trial division by 53 yields a factor.
- The remaining part 661 is a prime.
- Thus, we have \( n = 7^2 \times 53 \times 661 \).

Trial division is efficient if \( n \) has only small prime factors (except possibly one).
Pollard’s Rho Method

Let \( p \leq \sqrt{n} \) be an unknown prime divisor of \( n \).
Let \( f: \mathbb{Z}_n \to \mathbb{Z}_n \) be a pseudorandom function.

- Choose a random \( x_0 \in \mathbb{Z}_n \).
- Generate a sequence \( x_0, x_1, x_2, \ldots \) as \( x_i = f(x_{i-1}) \).
- Let \( y_i \equiv x_i \mod p \).
- The sequence \( y_0, y_1, y_2, \ldots \) plays from behind the curtain.
- The sequence \( y_0, y_1, y_2, \ldots \) must be (eventually) periodic.
- Suppose \( y_i \equiv y_j \mod p \) for \( i < j \).
- If \( x_i \not\equiv x_j \mod n \), then \( \gcd(x_i - x_j, n) \) is a non-trivial factor of \( n \).
- By the **Birthday Paradox**, the expected running time of Pollard’s rho method is \( O^\sim(\sqrt[4]{n}) \).
Pollard’s Rho Method: Example

Let \( n = 83947 \).

Take \( f(x) = x^2 - 1 \) (mod \( n \)).

- Start with \( x_0 = 123 \).
- The first 20 terms in the \( x \) sequence are:
  
  123, 15128, 16861, 48778, 67409, 6117, 61273, 18847, 29651, 4869, 34106, 49603, 54985, 82966, 38943, 54693, 40797, 61986, 10005, 35200.

- The corresponding terms in the \( y \) sequence are:
  
  123, 15, 97, 10, 99, 21, 59, 51, 60, 43, 70, 73, 121, 35, 81, 83, 30, 10, 99, 21.

- The periodic part in the \( y \) sequence is
  
  10, 99, 21, 59, 51, 60, 43, 70, 73, 121, 35, 81, 83, 30.

- The \( x \) sequence does not show the same periodicity.

- \( \gcd(61986 - 48778, n) = 127 \).
Fermat’s Factoring Method

Examples

- Take $n = 899$.
  
  
  
  
  $n = 900 - 1 = 30^2 - 1^2 = (30 - 1) \times (30 + 1) = 29 \times 31$.

- Take $n = 833$.
  
  
  
  
  $3 \times 833 = 2500 - 1 = 50^2 - 1^2 = (50 - 1) \times (50 + 1) = 49 \times 51$.
  
  
  
  
  $\gcd(50 - 1, 833) = 49$ is a non-trivial factor of 833.

Objective

To find integers $x, y \in \mathbb{Z}_n$ such that $x^2 \equiv y^2 \pmod{n}$. Unless $x \equiv \pm y \pmod{n}$, $\gcd(x - y, n)$ is a non-trivial divisor of $n$.

If $n$ is composite (but not a prime power), then for a randomly chosen pair $(x, y)$ with $x^2 \equiv y^2 \pmod{n}$, the probability that $x \not\equiv \pm y \pmod{n}$ is at least 1/2.
Let $n$ be an odd integer with no small prime factors. 

$H = \lceil \sqrt{n} \rceil$ and $J = H^2 - n$.

$(H + c)^2 \equiv J + 2Hc + c^2 \pmod{n}$ for small integers $c$.

Call $T(c) = J + 2Hc + c^2$.

Suppose $T(c)$ factors over small primes $p_1, p_2, \ldots, p_t$:

$$(H + c)^2 \equiv p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_t^{\alpha_t} \pmod{n}.$$ 

This is called a **relation**.

The left side is already a square.

The right side is also a square if each $\alpha_j$ is even.

But this is very rare.
QSM (contd)

Collect many relations:

Relation 1: \((H + c_1)^2 = p_1^{\alpha_{11}} p_2^{\alpha_{12}} \cdots p_t^{\alpha_{1t}}\)
Relation 2: \((H + c_2)^2 = p_1^{\alpha_{21}} p_2^{\alpha_{22}} \cdots p_t^{\alpha_{2t}}\)
\(\cdots\)
Relation \(r\): \((H + c_r)^2 = p_1^{\alpha_{r1}} p_2^{\alpha_{r2}} \cdots p_t^{\alpha_{rt}}\)

\[
\begin{align*}
\left[ (H + c_1)^{\beta_1} (H + c_2)^{\beta_2} \cdots (H + c_r)^{\beta_r} \right]^2 & \equiv p_1^{\gamma_1} p_2^{\gamma_2} \cdots p_t^{\gamma_t} \pmod{n}.
\end{align*}
\]

The left side is already a square.
Tune \(\beta_1, \beta_2, \ldots, \beta_r\) to make each \(\gamma_i\) even.
The Integer Factorization Problem
The Discrete Logarithm Problem
Solving Large Sparse Linear Systems

QSM (contd)

\[
\begin{align*}
\alpha_{11} \beta_1 + \alpha_{21} \beta_2 + \cdots + \alpha_{r1} \beta_r & = \gamma_1, \\
\alpha_{12} \beta_1 + \alpha_{22} \beta_2 + \cdots + \alpha_{r2} \beta_r & = \gamma_2, \\
& \quad \cdots \\
\alpha_{1t} \beta_1 + \alpha_{2t} \beta_2 + \cdots + \alpha_{rt} \beta_r & = \gamma_t.
\end{align*}
\]

Linear system with \( t \) equations and \( r \) variables \( \beta_1, \beta_2, \ldots, \beta_r \):

\[
\begin{pmatrix}
\alpha_{11} & \alpha_{21} & \cdots & \alpha_{r1} \\
\alpha_{12} & \alpha_{22} & \cdots & \alpha_{r2} \\
\vdots & \vdots & \ddots & \vdots \\
\alpha_{1t} & \alpha_{2t} & \cdots & \alpha_{rt}
\end{pmatrix}
\begin{pmatrix}
\beta_1 \\
\beta_2 \\
\vdots \\
\beta_t
\end{pmatrix}
\equiv
\begin{pmatrix}
0 \\
0 \\
\vdots \\
0
\end{pmatrix}
\pmod{2}.
\]
QSM (contd)

For \( r \geq t \), there are non-zero solutions for \( \beta_1, \beta_2, \ldots, \beta_r \). Take

\[
x \equiv (H + c_1)^{\beta_1} (H + c_2)^{\beta_2} \cdots (H + c_r)^{\beta_r} \pmod{n},
\]

\[
y \equiv \rho_1^{\gamma_1/2} \rho_2^{\gamma_2/2} \cdots \rho_t^{\gamma_t/2} \pmod{n}.
\]

If \( x \not\equiv \pm y \pmod{n} \), then \( \gcd(x - y, n) \) is a non-trivial factor of \( n \).

Let \( p = p_i \) be a small prime.

\( p \mid T(c) \) implies \( (H + c)^2 \equiv n \pmod{p} \).

If \( n \) is not a quadratic residue modulo \( p \), then \( p \nmid T(c) \) for any \( c \).

Consider only the small primes \( p \) modulo which \( n \) is a quadratic residue.
Example of QSM: Parameters

\[ n = 7116491. \]

\[ H = \lceil \sqrt{n} \rceil = 2668. \]

Take all primes < 100 modulo which \( n \) is a square:

\[ B = \{2, 5, 7, 17, 29, 31, 41, 59, 61, 67, 71, 79, 97\}. \]

\[ t = 13. \]

Take \( r = 13. \) (In practice, one takes \( r \approx 2t. \))
Example of QSM: Relations

<table>
<thead>
<tr>
<th>Relation</th>
<th>Equation</th>
<th>Equivalent Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relation 1:</td>
<td>$(H + 3)^2$</td>
<td>$2 \times 5^3 \times 71$</td>
</tr>
<tr>
<td>Relation 2:</td>
<td>$(H + 8)^2$</td>
<td>$5 \times 7 \times 31 \times 41$</td>
</tr>
<tr>
<td>Relation 3:</td>
<td>$(H + 49)^2$</td>
<td>$2 \times 41^2 \times 79$</td>
</tr>
<tr>
<td>Relation 4:</td>
<td>$(H + 64)^2$</td>
<td>$7 \times 29^2 \times 59$</td>
</tr>
<tr>
<td>Relation 5:</td>
<td>$(H + 81)^2$</td>
<td>$2 \times 5 \times 7^2 \times 29 \times 31$</td>
</tr>
<tr>
<td>Relation 6:</td>
<td>$(H + 109)^2$</td>
<td>$2 \times 7 \times 17 \times 41 \times 61$</td>
</tr>
<tr>
<td>Relation 7:</td>
<td>$(H + 128)^2$</td>
<td>$5^3 \times 71 \times 79$</td>
</tr>
<tr>
<td>Relation 8:</td>
<td>$(H + 145)^2$</td>
<td>$2 \times 71^2 \times 79$</td>
</tr>
<tr>
<td>Relation 9:</td>
<td>$(H + 182)^2$</td>
<td>$17^2 \times 59^2$</td>
</tr>
<tr>
<td>Relation 10:</td>
<td>$(H + 228)^2$</td>
<td>$5^2 \times 7^2 \times 17 \times 61$</td>
</tr>
<tr>
<td>Relation 11:</td>
<td>$(H + 267)^2$</td>
<td>$2 \times 7^2 \times 17 \times 29 \times 31$</td>
</tr>
<tr>
<td>Relation 12:</td>
<td>$(H + 382)^2$</td>
<td>$7 \times 59 \times 67 \times 79$</td>
</tr>
<tr>
<td>Relation 13:</td>
<td>$(H + 411)^2$</td>
<td>$2 \times 5^4 \times 31 \times 61$</td>
</tr>
</tbody>
</table>

(mod n).
Example of QSM: Linear System

\[
\begin{pmatrix}
1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\
3 & 1 & 0 & 0 & 1 & 0 & 3 & 0 & 0 & 2 & 0 & 0 & 4 \\
0 & 1 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 2 & 2 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 2 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 1 & 2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\begin{pmatrix}
\beta_1 \\
\beta_2 \\
\beta_3 \\
\beta_4 \\
\beta_5 \\
\beta_6 \\
\beta_7 \\
\beta_8 \\
\beta_9 \\
\beta_{10} \\
\beta_{11} \\
\beta_{12} \\
\beta_{13}
\end{pmatrix}
\equiv
\begin{pmatrix}
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0
\end{pmatrix}
\pmod{2}.
\]
Example of QSM: Solution of Relations

<table>
<thead>
<tr>
<th>$(\beta_1, \beta_2, \beta_3, \ldots, \beta_{13})$</th>
<th>$x$</th>
<th>$y$</th>
<th>$\gcd(x - y, n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$</td>
<td>1</td>
<td>1</td>
<td>7116491</td>
</tr>
<tr>
<td>$(1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0)$</td>
<td>1755331</td>
<td>560322</td>
<td>1847</td>
</tr>
<tr>
<td>$(0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0)$</td>
<td>526430</td>
<td>459938</td>
<td>1847</td>
</tr>
<tr>
<td>$(1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0)$</td>
<td>7045367</td>
<td>7045367</td>
<td>7116491</td>
</tr>
<tr>
<td>$(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0)$</td>
<td>2850</td>
<td>1003</td>
<td>1847</td>
</tr>
<tr>
<td>$(1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0)$</td>
<td>6916668</td>
<td>6916668</td>
<td>7116491</td>
</tr>
<tr>
<td>$(0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0)$</td>
<td>5862390</td>
<td>5862390</td>
<td>7116491</td>
</tr>
<tr>
<td>$(1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0)$</td>
<td>3674839</td>
<td>6944029</td>
<td>1847</td>
</tr>
<tr>
<td>$(0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1)$</td>
<td>1079130</td>
<td>3965027</td>
<td>3853</td>
</tr>
<tr>
<td>$(1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1)$</td>
<td>5466596</td>
<td>1649895</td>
<td>1</td>
</tr>
<tr>
<td>$(0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1)$</td>
<td>5395334</td>
<td>1721157</td>
<td>1</td>
</tr>
<tr>
<td>$(1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1)$</td>
<td>6429806</td>
<td>3725000</td>
<td>3853</td>
</tr>
<tr>
<td>$(0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1)$</td>
<td>1196388</td>
<td>5920103</td>
<td>1</td>
</tr>
<tr>
<td>$(1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1)$</td>
<td>1799801</td>
<td>3818773</td>
<td>3853</td>
</tr>
<tr>
<td>$(0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1)$</td>
<td>5081340</td>
<td>4129649</td>
<td>3853</td>
</tr>
<tr>
<td>$(1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1)$</td>
<td>7099266</td>
<td>17225</td>
<td>1</td>
</tr>
</tbody>
</table>
Sieving

- To identify which values of \( T(c) = J + 2Hc + c^2 \) are smooth with respect to the factor base \( B \).
- Trial division is too expensive.
- Sieving replaces trial division by single-precision subtractions.
- Use an array \( A \) indexed by \( c \) in the range \(-M \leq c \leq M\).
- Initialize: \( A_c = \log |T(c)|. \)
- Let \( q \in B \), and \( h \) a small positive integer.
- Solve \( T(c) \equiv 0 \pmod{q^h} \).
- For each solution \( \chi \), subtract \( \log q \) from \( A_c \) for all \( c \equiv \chi \pmod{q^h} \).
- \( T(c) \) is \( B \)-smooth if and only if the remaining \( A_c \approx 0 \).
- Each smooth \( T(c) \) is factored by trial division.
The Discrete Logarithm Problem

To compute the discrete logarithm of \( a \) in \( \mathbb{F}_q^* \) to the primitive base \( g \).

**Older algorithms**

- Brute-force search
- Shanks’ Baby-step-giant-step method
- Pollard’s rho method
- Pollard’s lambda method
- Pohlig-Hellman method (Efficient if \( p - 1 \) has only small prime divisors)

**Worst-case complexity:** Exponential in \( \log q \)
Modern algorithms

Based on the index calculus method (ICM)

Subexponential running time:
\[ L(q, \omega, c) = \exp \left[ (c + o(1)) (\ln q)^{\omega} (\ln \ln q)^{1-\omega} \right]. \]

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Inventor(s)</th>
<th>Running time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic ICM</td>
<td>Western &amp; Miller (1968)</td>
<td>( L(q, 1/2, c) )</td>
</tr>
<tr>
<td>Linear sieve method (LSM)</td>
<td>Coppersmith, Odlyzko</td>
<td>( L(q, 1/2, 1) )</td>
</tr>
<tr>
<td>Residue list sieve method</td>
<td>&amp; Schroeppel (1986)</td>
<td></td>
</tr>
<tr>
<td>Gaussian integer method</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cubic sieve method (CSM)</td>
<td>Reyneri</td>
<td>( L(q, 1/2, 0.816) )</td>
</tr>
<tr>
<td>Number field sieve method (NFSM) [for ( \mathbb{F}_p ) only]</td>
<td>Gordon (1993)</td>
<td>( L(q, 1/3, 1.923) )</td>
</tr>
<tr>
<td>Coppersmith’s method [for ( \mathbb{F}_{2^n} ) only]</td>
<td>Coppersmith</td>
<td>( L(q, 1/3, 1.526) )</td>
</tr>
</tbody>
</table>
The Baby-Step-Giant-Step Method

Let $G$ be a cyclic multiplicative group of size $n$. Let $g$ be a generator of $G$. We plan to compute $\text{ind}_a(g)$ for some $a \in G$.

- Let $m = \lceil \sqrt{n} \rceil$.
- **Baby steps:** For $i \in \{0, 1, 2, \ldots, m - 1\}$, compute $g^i$, and store $(i, g^i)$ sorted with respect to the second element.
- **Giant steps:** For $j = 0, 1, 2, \ldots, m - 1$, compute $ag^{-jm}$, and try to locate $ag^{-jm}$ in the table of baby steps.
- If a search is successful, we have $ag^{-jm} = g^i$ for some $i, j$, that is, $a = g^{jm+i}$, that is, $\text{ind}_g(a) = jm + i$. 
The Integer Factorization Problem
The Discrete Logarithm Problem
Solving Large Sparse Linear Systems

The BSGS Method: Example

Take $G = \mathbb{F}_{43}^*$ with size $n = 42$, $m = \lceil \sqrt{42} \rceil = 7$, and $g = 19$.

Table of Baby Steps

<table>
<thead>
<tr>
<th>$i$</th>
<th>0</th>
<th>6</th>
<th>2</th>
<th>1</th>
<th>3</th>
<th>5</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$g^i$</td>
<td>1</td>
<td>11</td>
<td>17</td>
<td>19</td>
<td>22</td>
<td>30</td>
<td>31</td>
</tr>
</tbody>
</table>

Giant Steps: Take $a = 3$.
- $j = 0$: $ag^{-0m} \equiv 3 \pmod{43}$ is not in the table.
- $j = 1$: $ag^{-m} \equiv 21 \pmod{43}$ is not in the table.
- $j = 2$: $ag^{-2m} \equiv 18 \pmod{43}$ is not in the table.
- $j = 3$: $ag^{-3m} \equiv 40 \pmod{43}$ is not in the table.
- $j = 4$: $ag^{-4m} \equiv 22 \equiv g^3 \pmod{43}$, so $\text{ind}_g(a) = 4 \times 7 + 3 = 31$. 

Public-key Cryptography: Theory and Practice
Abhijit Das
Goal: To compute $\text{ind}_g(a)$ in $\mathbb{F}_p^*$ to a primitive root $g$ modulo $p$.

Factor base: First $t$ primes $B = \{p_1, p_2, \ldots, p_t\}$

To compute $d_i = \text{ind}_g p_i$ for $i = 1, 2, \ldots, t$

For random $j \in \{1, 2, \ldots, p - 2\}$, try to factor $g^j \pmod{p}$ over $B$.

Relation: $g^j \equiv p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_t^{\alpha_t} \pmod{p}$

Linear equation in $t$ variables $d_1, d_2, \ldots, d_t$:

$$j \equiv \alpha_1 d_1 + \alpha_2 d_2 + \cdots + \alpha_t d_t \pmod{p - 1}$$
Generate $r \geq t$ relations for different values of $j$:

Relation 1: $j_1 \equiv \alpha_{11}d_1 + \alpha_{12}d_2 + \cdots + \alpha_{1t}d_t$ 
Relation 2: $j_2 \equiv \alpha_{21}d_1 + \alpha_{22}d_2 + \cdots + \alpha_{2t}d_t$ 
\[ \vdots \] 
Relation $r$: $j_r \equiv \alpha_{r1}d_1 + \alpha_{r2}d_2 + \cdots + \alpha_{rt}d_t$ 

\[
\begin{aligned}
\{&j_1, j_2, \ldots, j_r\} \pmod{p-1}.
\end{aligned}
\]

Solve the system modulo $p - 1$ to determine $d_1, d_2, \ldots, d_t$. 

Public-key Cryptography: Theory and Practice

Abhijit Das
Choose random $j \in \{1, 2, \ldots, p - 2\}$. 
Try to factor $a g^j \pmod{p}$ over $B$.

A successful factorization gives:

$$a g^j \equiv p_1^{\beta_1} p_2^{\beta_2} \cdots p_t^{\beta_t} \pmod{p}.$$ 

Take discrete log:

$$\text{ind}_g a \equiv -j + \beta_1 d_1 + \beta_2 d_2 + \cdots + \beta_t d_t \pmod{p - 1}.$$ 

Substitute the values of $d_1, d_2, \ldots, d_t$ to get $\text{ind}_g a$. 
The Basic ICM: Example (Precomputation)

Parameters: \( p = 839, \ g = 31, \ B = \{2, 3, 5, 7, 11\}, \ t = 5, \ r = 10. \)

Relations

- Relation 1: \( g^{118} \equiv 2^3 \times 5^2 \) (mod \( p \)).
- Relation 2: \( g^{574} \equiv 2^7 \times 5 \) (mod \( p \)).
- Relation 3: \( g^{318} \equiv 2^2 \times 3^3 \) (mod \( p \)).
- Relation 4: \( g^{46} \equiv 2^7 \) (mod \( p \)).
- Relation 5: \( g^{786} \equiv 2^2 \times 3^3 \times 7 \) (mod \( p \)).
- Relation 6: \( g^{323} \equiv 2 \times 3 \times 11 \) (mod \( p \)).
- Relation 7: \( g^{606} \equiv 3^4 \) (mod \( p \)).
- Relation 8: \( g^{252} \equiv 2^3 \times 3^2 \times 7 \) (mod \( p \)).
- Relation 9: \( g^{160} \equiv 3 \times 5^2 \) (mod \( p \)).
- Relation 10: \( g^{600} \equiv 2 \times 3^3 \times 5 \) (mod \( p \)).
The Basic ICM: Example (Precomputation)

\[
\begin{pmatrix}
3 & 0 & 2 & 0 & 0 \\
7 & 0 & 1 & 0 & 0 \\
2 & 3 & 0 & 0 & 0 \\
7 & 0 & 0 & 0 & 0 \\
2 & 3 & 0 & 1 & 0 \\
1 & 1 & 0 & 0 & 1 \\
0 & 4 & 0 & 0 & 0 \\
3 & 2 & 0 & 1 & 0 \\
0 & 1 & 2 & 0 & 0 \\
1 & 3 & 1 & 0 & 0
\end{pmatrix}
\begin{pmatrix}
d_1 \\
d_2 \\
d_3 \\
d_4 \\
d_5
\end{pmatrix}
\equiv
\begin{pmatrix}
118 \\
574 \\
318 \\
46 \\
786 \\
323 \\
606 \\
252 \\
160 \\
600
\end{pmatrix}
\pmod{p - 1}.
\]
The coefficient matrix has full column rank (5) modulo $p−1 = 838$.

The solution is unique.

\[
\begin{align*}
    d_1 &\equiv \text{ind}_{31} 2 \equiv 246 \\
    d_2 &\equiv \text{ind}_{31} 3 \equiv 780 \\
    d_3 &\equiv \text{ind}_{31} 5 \equiv 528 \\
    d_4 &\equiv \text{ind}_{31} 7 \equiv 468 \\
    d_5 &\equiv \text{ind}_{31} 11 \equiv 135
\end{align*}
\] (mod $p − 1$).
The Basic ICM: Example (Second Stage)

Take $a = 561$.

\[ ag^{312} \equiv 600 \equiv 2^3 \times 3 \times 5^2 \pmod{p}, \quad \text{that is,} \]
\[ \text{ind}_{31} 561 \equiv -312 + 3 \times 246 + 780 + 2 \times 528 \equiv 586 \pmod{p - 1}. \]

Take $a = 89$.

\[ ag^{342} \equiv 99 \equiv 3^2 \times 11 \pmod{p}, \quad \text{that is,} \]
\[ \text{ind}_{31} 89 \equiv -342 + 2 \times 780 + 135 \equiv 515 \pmod{p - 1}. \]

Take $a = 625$.

\[ ag^{806} \equiv 70 \equiv 2 \times 5 \times 7 \pmod{p}, \quad \text{that is,} \]
\[ \text{ind}_{31} 625 \equiv -806 + 246 + 528 + 468 \equiv 436 \pmod{p - 1}. \]
The Basic ICM for $\mathbb{F}_{2^n}$

Represent $\mathbb{F}_{2^n} = \mathbb{F}_2(\alpha)$, where $f(\alpha) = 0$. Let $g(\alpha)$ be a generator of $\mathbb{F}^*_2$. We plan to compute $\text{ind}_{g(\alpha)} t(\alpha)$.

- **Factor base**: $B = \{u(\alpha) \mid \deg g \leq m\}$.
- **Relation**: Choose $j \in \{0, 1, 2, \ldots, 2^n - 2\}$ randomly, and try to arrive at factorizations of the form:
  \[ g(\alpha)^j = \prod_{u(\alpha) \in B} u(\alpha)^{\gamma_{u(\alpha)}}. \]
- **Linear algebra**: Solve the resulting system of congruences modulo $2^n - 1$, and obtain the indices $\text{ind}_{g(\alpha)} u(\alpha)$ for all $u(\alpha) \in B$.
- **Second stage**: Generate a single relation of the form:
  \[ t(\alpha) g(\alpha)^j = \prod_{u(\alpha) \in B} u(\alpha)^{\delta_{u(\alpha)}}. \]
The Basic ICM: Example

- Represent $F_{128} = F_2(\alpha)$ with $\alpha^7 + \alpha + 1 = 0$.
- $|F_{128}^*| = 127$ is prime. Take $g(\alpha) = \alpha^5 + \alpha^2 + 1$.
- Take $m = 2$, that is, $B = \{\alpha, \alpha + 1, \alpha^2 + \alpha + 1\}$.
- Relations in the first stage
  
  $$g(\alpha)^7 = \alpha^6 + \alpha^2 = \alpha^2(\alpha + 1)^4,$$
  $$g(\alpha)^{101} = \alpha^4 + \alpha^3 + \alpha + 1 = (\alpha + 1)^2(\alpha^2 + \alpha + 1),$$
  $$g(\alpha)^{121} = \alpha^5 + \alpha^2 = \alpha^2(\alpha + 1)(\alpha^2 + \alpha + 1).$$

- Linear system of congruences
  
  $$\begin{pmatrix}
  2 & 4 & 0 \\
  0 & 2 & 1 \\
  2 & 1 & 1 
  \end{pmatrix}
  \begin{pmatrix}
  d_\alpha \\
  d_{\alpha+1} \\
  d_{\alpha^2+\alpha+1} 
  \end{pmatrix}
  \equiv
  \begin{pmatrix}
  7 \\
  101 \\
  121 
  \end{pmatrix}
  \pmod{127},$$

  where $d_\beta = \text{ind}_{g(\alpha)}(\beta)$. 

Public-key Cryptography: Theory and Practice  Abhijit Das
The Basic ICM: Example (contd)

- **Indices of factor base elements**
  
  \[ d_\alpha = 123, \quad d_{\alpha+1} = 99, \quad \text{and} \quad d_{\alpha^2+\alpha+1} = 30. \]

- **Second stage**

  - \[ t(\alpha) = \alpha^3 + 1. \]
    
    \[ t(\alpha)g(\alpha)^{57} = \alpha^5 + \alpha^3 = \alpha^3(\alpha + 1)^2. \]
    
    \[ \text{ind}_{g(\alpha)} t(\alpha) \equiv -57 + 3d_\alpha + 2d_{\alpha+1} \equiv 2 \pmod{127}. \]

  - \[ t(\alpha) = \alpha^6 + \alpha^5 + \alpha^4 + \alpha^3 + \alpha^2 + \alpha + 1. \]
    
    \[ t(\alpha)g(\alpha)^{73} = \alpha^5 + \alpha^4\alpha^2 + \alpha = \alpha(\alpha + 1)^2(\alpha^2 + \alpha + 1). \]
    
    \[ \text{ind}_{g(\alpha)} t(\alpha) \equiv -73 + d_\alpha + 2d_{\alpha+1} + d_{\alpha^2+\alpha+1} \equiv 24 \pmod{127}. \]

  - \[ t(\alpha) = \alpha^5 + 1. \]
    
    \[ t(\alpha)g(\alpha)^{18} = \alpha^5 + \alpha^3 + \alpha = \alpha(\alpha^2 + \alpha + 1)^2. \]
    
    \[ \text{ind}_{g(\alpha)} t(\alpha) \equiv -18 + d_\alpha + 2d_{\alpha^2+\alpha+1} \equiv 38 \pmod{127}. \]
For a general curve, only the exponential square-root methods apply.

Index calculus methods for elliptic curves are neither known nor likely to exist.

The subexponential **MOV attack** applies to supersingular curves.

The linear-time **anomalous attack** (also called the **SmartASS attack**) applies to anomalous curves.

Supersingular and anomalous curves are not used in cryptography.

The **Xedni calculus method** applies to general curves, but is found to be impractical.
To solve $A\mathbf{x} \equiv \mathbf{b} \pmod{M}$, where $A$ a sparse $m \times n$ matrix.

If $M$ is prime, we work in the finite field $\mathbb{F}_M$.

If $M$ is composite with known factorization, we solve the system modulo prime power divisors of $M$.

If $M$ is composite with unknown factorization, we pretend $M$ as prime. If inversion modulo $M$ fails, we discover non-trivial factors of $M$, and solve the system modulo each factor thus discovered.

We have $m = \Theta(n)$.

If $A$ is dense, the system solving phase runs in $O^\sim(n^3)$ time.

If $A$ is sparse, there are $O^\sim(n^2)$-time algorithms.
Structured Gaussian Elimination

- Used to reduce the size of $A$.
- The size reduction often becomes substantial.
- The reduced matrix becomes much denser.

Some steps of structured Gaussian elimination:
- Delete zero columns.
- Delete columns with single non-zero entries and the corresponding rows.
- Delete rows with single non-zero entries.
- Throw excess rows with large numbers of non-zero entries.
Lanczos Method

Let $A$ be a symmetric positive-definite matrix with real entries. We plan to solve $Ax = b$.

We generate a set of pairwise orthogonal directions $d_0, d_1, d_2, \ldots$ until we run out of new orthogonal directions.

- **Initialization**
  
  $d_0 = b$, $v_1 = Ad_0$, $d_1 = v_1 - d_0(v_1^t Ad_0)/(d_0^t Ad_0)$,
  
  $a_0 = (d_0^t d_0)/(d_0^t Ad_0)$, and $x_0 = a_0 d_0$.

- **Iteration:** For $i = 1, 2, 3, \ldots$, repeat:
  
  $v_{i+1} = Ad_i$.
  
  $d_{i+1} = v_{i+1} - d_i(v_{i+1}^t Ad_i)/(d_i^t Ad_i) - d_{i-1}(v_{i+1}^t Ad_{i-1})/(d_{i-1}^t Ad_{i-1})$.
  
  $a_i = (d_i^t b)/(d_i^t Ad_i)$.
  
  $x_i = x_{i-1} + a_i d_i$. 
In general, $A$ is not a square matrix.

**Remedy**

- $(A^t A)x = A^t b$ is a square $(n \times n)$ system.
- Moreover, $A^t A$ is symmetric.
- Instead of computing $A^t A$, multiply separately by $A$ and $A^t$.

Positive-definiteness makes no sense in $\mathbb{Z}_M$. Problem arises when we encounter a non-zero vector $d_i$ with $d_i^t A d_i = 0$. The problem is likely to occur unless $M$ is large.

**Remedy**

- Work in extension fields ($\mathbb{F}_{M^s}$ in place of $\mathbb{F}_M$).
- Solve $D(A^t A)x = DA^t b$ for random non-singular diagonal matrices $D$. 
Other Sparse System Solvers

- **Conjugate gradient method**: An iterative method similar to the Lanczos method.

- **Wiedemann method**: Computes the minimal polynomial of $A$ in $\mathbb{F}_M[x]$.

- **Block Lanczos method**:
  - Meant for systems modulo 2.
  - Bits are packed into words.
  - Multiple direction vectors are computed per iteration.
  - The problem of self-orthogonality of non-zero vectors is less acute.

- **Block Wiedemann method**: The block implementation of the Wiedemann method for systems modulo 2.