A Knapsack Cryptosystem
Secure Against Attacks Using
Basis Reduction and Integer Programming

Bala Krishnamoorthy
William Webb
Nathan Moyer
Washington State University

ISMP 2006
August 2, 2006
Public Key Cryptosystems

- Diffie and Hellman (1976)

- Bob wants to receive a message from Alice
  - a 0–1 $n$-vector

- Bob keeps a public key

- Alice encrypts message and transmits

- Bob decrypts message using his private key

- hard to decrypt without private key
Public Key Cryptosystems

- Eve (eavesdropper) has to solve a “hard” problem to intercept

- one-way functions
  - given $x$, easy to find $y = f(x)$
  - hard to find $x = f^{-1}(y)$, given $y$

- RSA (Rivest-Shamir-Adleman)
  - given $N$, need to find two large primes $p, q$ s.t. $N = pq$
Knapsack Cryptosystems

- Merkle-Hellman scheme (1978)
- Bob creates a superincreasing knapsack $S = \{s_1, \ldots, s_n\}$
  - $s_i > \sum_{j=1}^{i-1} s_j$ for $i = 2, \ldots, n$
- chooses $p, q$ with $\gcd(p, q) = 1$
- private key $- (S, p, q)$
- can find $p^{-1} \mod q$
Knapsack Cryptosystems: Encryption

- Computes $a_i \equiv ps_i \mod q$
- Public key $- A = \{a_1, \ldots, a_n\}$
- To send message $x = (x_1, \ldots, x_n)$, $x_i = 0$ or $1$,
- Alice computes $M = \sum_{i=1}^{n} a_i x_i$ (encryption)
- Transmits $M$ to Bob
Knapsack Cryptosystems: Decryption

- Bob solves

\[ \sum_{i=1}^{n} a_i x_i \equiv M \mod q \]

\[ \Rightarrow p^{-1} \sum_{i=1}^{n} a_i x_i \equiv p^{-1} M \mod q \]

\[ \Rightarrow \sum_{i=1}^{n} s_i x_i \equiv M' \mod q \]

- easy to solve (as \( S \) is superincreasing)

\[ \text{while } M' > 0 \text{ do} \]
\[ \quad \text{find largest } s_i \in S \text{ s.t. } s_i \leq M'; \]
\[ \quad \text{set } x_i = 1; \]
\[ \quad \text{set } M' \leftarrow M' - s_i; \]
\[ \text{end while} \]
An Example

- $n = 5; \quad S = \{1, 3, 7, 13, 26\};$
  - $p = 467, \quad q = 523; \quad p^{-1} \mod q \equiv 28;\quad A = \{467, 355, 131, 318, 113\} = a$

- for message $x = (0, 1, 1, 0, 1)$, Alice transmits
  - $M = ax = 599$

- Bob calculates $M' \equiv p^{-1}M \mod q = 36$
  - $36 - 26 = 10 \Rightarrow x_5 = 1$
  - $10 - 7 = 3 \Rightarrow x_3 = 1$
  - $3 - 3 = 0 \Rightarrow x_2 = 1$
Knapsack Cryptosystems: Security

• to intercept, Eve has to solve

\[ \sum_{i=1}^{n} a_i x_i = M \]

\[ x_i \in \{0, 1\}, \quad i = 1, \ldots, n \]

• 0–1 knapsack problem is NP-complete

• difficulty to solve knapsack implies security

• Merkle offered a $100 prize for breaking the code!!
Attacks using Diophantine Approximation

- Shamir (1982) used the superincreasing property of $S$

- find $p', q'$ such that $\{p' A \mod q'\}$ is superincreasing

- with $A = \{467, 355, 131, 318, 113\}$,
  
  - $\frac{p'}{q'}$ must lie in one of 466 intervals $\left[ \frac{k}{467}, \frac{k}{467} + \frac{1}{2^{4}467} \right]$ for $k = 1, \ldots, 466$
  
  - $\frac{p'}{q'}$ must lie in one of 354 intervals $\left[ \frac{k}{355}, \frac{k}{355} + \frac{1}{2^{3}355} \right]$ for $k = 1, \ldots, 354$
  
  - intersect intervals, try many choices.

- for $p' = 53, q' = 990, p' A \mod q' = \{1, 5, 13, 24, 49\}$
  and $p' M \mod q' = 67$ solves the problem!
Attacks using Diophantine Approximation

• Adleman (1983) and Brickell, Lagarias, Odlyzko (1983)

• find integers $k_1, \ldots, k_\ell$ such that $\frac{k_i}{a_i}$ approximates $\frac{p}{q}$

• need only a few $k_i$’s

• solve an IP to find $k_i$’s (Lenstra’s algorithm)

• IP created using superincreasing property of $S'$
Attacks using Basis Reduction

- Lagarias and Odlyzko (LO) (1985)

- to solve $ax = M$, consider the lattice $\mathbb{L}(B)$ generated by

  $$B = \begin{bmatrix} I & 0 \\ Na & -NM \end{bmatrix}, \quad N \text{ large}$$

- $\hat{x} = \begin{bmatrix} x \\ 0 \end{bmatrix}$ is a shortest vector in $\mathbb{L}(B)$

- apply LLL basis reduction to $B$

- check for solution in the reduced basis
Attacks using Basis Reduction

- density = \( \frac{n}{\max_{i=1,\ldots,n} \log_2 a_i} \)

- \( \text{Prob}(\exists \tilde{x} = \begin{bmatrix} x' \\ y \end{bmatrix} \in \mathbb{Z}^{n+1} | ax' = My, \| x' \| \leq \| x \| ) \rightarrow 0 \) as \( n \rightarrow \infty \) if density < 0.647

- LO method solves almost all knapsack problems of density < 0.647 in poly time if there is a poly time oracle for solving the shortest vector problem

- claim: LLL acts like such an oracle in practice??

- Coster et al. (1991): density < 0.941
Attacks using Basis Reduction

- LaMacchia (1993) – empirical testing

- used Seysen-Lovász reduction

- instances had density $< 1$ (needed for unique decoding?)

- $n/2$ of the $x_i$’s are set to 1

- for $n = 106$, with density $= 0.393$, ($a_i$’s had $\approx 80$ digits), on an average
  - solved 50% instances
  - in 34147 seconds
Attacks using Integer Programming

• try to solve the IP directly:

\[
\sum_{i=1}^{n} a_i x_i = M
\]

\[
x_i \in \{0, 1\}, \quad i = 1, \ldots, n
\]

• solvers cannot handle the huge numbers involved

• try exact solvers? (Espinoza et al.)

• apply **Column Basis Reduction**
  
  – obtain reformulation with smaller numbers using BR
  
  – run CPLEX on reformulation
Reasons for Insecurity

- structure – superincreasing sequence
- low density
- existence of lattices in which the correct solution is a shortest vector
- success of BR in finding the shortest vectors in such lattices (in practice)
Why Knapsack??

- simple in construction
- fast encryption and decryption
  - addition/subtraction instead of multiplication/exponentiation
- claim that LLL is *highly likely* to find the shortest vector in *almost all* instances:
  - still too early to throw in the towel!!
A New Knapsack Cryptosystem

- construction and structure
- an example
- security against BR-based attacks
- security against IP attacks
- further work
New Cryptosystem: Construction

- pick $r$ primes $p_1, \ldots, p_r$ (private)

- $\forall i$ pick $m_i \leq p_i$ numbers distinct $\mod p_i$, put them in set $S_i$ (Note: these numbers can be bigger than $p_i$)

- $\forall i$ find $A_i = S_i \times \frac{p_1 \cdots p_r}{p_i}$

- $n = \sum_{i=1}^{r} m_i$; knapsack coefficients are $A = \{A_1, \ldots, A_r\} = a$ (public)

- can receive messages ($0$–$1$ $n$-vectors) which pick one element from $A_i$ for each $i$

- can further disguise $a_{ij}$’s using modular multiplication
New Cryptosystem: Construction

- Eve needs to solve

\[
\sum_{i=1}^{r} \sum_{j=1}^{m_i} a_{ij} x_{ij} = M
\]

\[
\sum_{j=1}^{m_i} x_{ij} = 1 \quad i = 1, \ldots, r
\]

\[
x_{ij} \in \{0, 1\}, \quad i = 1, \ldots, r, \quad j = 1, \ldots, m_i
\]

- can set

\[
\sum_{j=1}^{m_i} x_{ij} = v_i \quad i = 1, \ldots, r, \text{ where } v_i \geq 1, \quad \text{or}
\]

- can set

\[
\sum_{j=1}^{m_i} w_{ij} x_{ij} = v_i \quad i = 1, \ldots, r, \text{ for some}
\]

\[
w_{ij} \in \mathbb{Z}, \quad v_i \geq 1
\]
New Cryptosystem: Example

- $r = 2$, $p_1 = 5$, $p_2 = 7$;
  $S_1 = \{21, 17, 13, 34, 25\}$, $S_2 = \{22, 25, 31, 33\}$;

- $A_1 = S_1 \times 7 = \{147, 119, 91, 238, 175\}$,
  $A_2 = S_2 \times 5 = \{110, 125, 155, 165\}$;

- All sums $a_{1i} + a_{2j}$ are distinct $\mod 35$

- Alice sends $119 + 165 = 284 = M$

- $M = 284 \equiv 4 \mod 5$, hence Bob needs $7s_1 \equiv 4 \mod 5$
  i.e., $s_1 \equiv 12 \mod 5 \equiv 2 \mod 5$
  looks up $S_1$ to find $17 \equiv 2 \mod 5$
  hence, first part of message is $(0, 1, 0, 0, 0)$
New Cryptosystem: Density

- Message space has \( \prod_{i=1}^{r} m_i \) messages

- restrictive, but helps to increase density!

- with \( m_i = m \ \forall i \),

\[
\text{density} = \frac{rm}{\max_{i=1,\ldots,n} \log_2 a_i} \geq \frac{rm}{((r - 1) \log_2 p_{\max} + \log_2 R)}
\]

where \( p_{\max} = \max_i p_i \) and \( R = \max_{i,j} S_{ij} \)

- can choose \( m, p_i \)'s and \( R \) such that density is much bigger than 1!

- only \( m \) of the \( x_{ij} \)'s is 1
Security against Basis Reduction

- a (simpler) test problem for Lagarias-Odlyzko method

- \( m_i = m \) \( \forall i \); \( I = [9R, 10R], J = [10R, 11R] \) for large \( R \)
- choose \( s_{i1} \in I \) randomly \( \forall i \), let \( \sum_{i=1}^{r} s_{i1} = M \)
- for \( j = 2, \ldots, m \), choose \( s_{ij} \in J \) randomly for \( i = 1, \ldots, r - 1 \)
- set \( s_{rj} = \sum_{i=1}^{r-1} s_{ij} + 2s_{r1} - M \)

try LO method on

\[
\sum_{i=1}^{r} \sum_{j=1}^{m} s_{ij} x_{ij} = M
\]

\[
\sum_{j=1}^{m} x_{ij} = 1 \quad i = 1, \ldots, r
\]

\( x_{ij} \in \{0, 1\}, \quad i = 1, \ldots, r, \quad j = 1, \ldots, m \)
Test for Lagarias-Odlyzko Method

- apply LLL reduction to $B = \begin{bmatrix} I & 0 \\ ND & -Nb \end{bmatrix}$, where

$$D = \begin{bmatrix} S_1 & \ldots & S_r \\ 1 \ldots 1 & \ldots & 1 \ldots 1 \end{bmatrix}, \text{ and } b = \begin{bmatrix} M \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

- solution is shortest vector in $\mathbb{L}(B)$: $x_{i1} = 1 \forall i$, $\|\hat{x}\| = \sqrt{r}$

- there are $m - 1$ “short” vectors in $\mathbb{L}(B)$ with length $\sqrt{r + 4}$

- Results:
  - LO method always finds $\hat{x}$ for $r \leq 6, m \leq 6, R \leq 10^8$
  - 100 trials with $r = m = 20, R = 10^{20}$, LO found $\hat{x}$ in none of the instances!
Security against Basis Reduction

- if cardinality constraints are ignored for the LO method
  - can prove that there exists exponentially many vectors in $\mathbb{L}(B)$ equal or shorter in length to the solution vector

- performance of LLL similar on actual instances
  (with $p_1, \ldots, p_r$)

- even block Korkine-Zolotarev (BKZ) reduction produced similar results
  - arguably, the strongest BR algorithm implemented
Security against IP Reformulations

- original IP is \( \{ x \in \{0, 1\}^n \mid Dx = b \} \) where

\[
D = \begin{bmatrix}
A_1 & \ldots & A_r \\
1 & \ldots & 1 \\
\vdots & \ddots & \vdots \\
1 & \ldots & 1
\end{bmatrix}, \quad \text{and} \quad b = \begin{bmatrix}
M \\
1 \\
\vdots \\
1
\end{bmatrix}
\]

- rewrite IP as \( \{ x \in \mathbb{Z}^n \mid Bx(\leq) f \} \) where

\[
B = \begin{bmatrix}
D \\
-I \\
I
\end{bmatrix}, \quad \text{and} \quad f = \begin{bmatrix}
b \\
0 \\
e
\end{bmatrix}
\]

- using BKZ reduction, find reduced \( \tilde{B}, \tilde{f} \); try to solve \( \{ y \in \mathbb{Z}^n \mid \tilde{B}y(\leq) \tilde{f} \} \)
**IP Reformulation Tests**

** – unsolved instances
† – CPLEX encountered numerical instability

<table>
<thead>
<tr>
<th>$r$</th>
<th>$m$</th>
<th>$n = rm$</th>
<th>$R$</th>
<th>BRED</th>
<th>CPLEX</th>
<th>#BB</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>13</td>
<td>130</td>
<td>100</td>
<td>1</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>40</td>
<td>13</td>
<td>520</td>
<td>100</td>
<td>74</td>
<td>4</td>
<td>59</td>
</tr>
<tr>
<td>80</td>
<td>13</td>
<td>1040</td>
<td>100</td>
<td>1097</td>
<td>13</td>
<td>1110</td>
</tr>
<tr>
<td>150</td>
<td>13</td>
<td>1950</td>
<td>100</td>
<td>&gt; 4 hrs</td>
<td>**</td>
<td>**</td>
</tr>
<tr>
<td>3</td>
<td>73</td>
<td>219</td>
<td>100</td>
<td>2</td>
<td>1</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>179</td>
<td>537</td>
<td>100</td>
<td>35</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>541</td>
<td>1623</td>
<td>100</td>
<td>2764</td>
<td>9</td>
<td>705</td>
</tr>
<tr>
<td>3</td>
<td>1229</td>
<td>3687</td>
<td>100</td>
<td>&gt; 4 hrs</td>
<td>**</td>
<td>**</td>
</tr>
<tr>
<td>10</td>
<td>23</td>
<td>230</td>
<td>$10^{10}$</td>
<td>1</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>10</td>
<td>23</td>
<td>230</td>
<td>$10^{20}$</td>
<td>7</td>
<td>4</td>
<td>190</td>
</tr>
<tr>
<td>10</td>
<td>23</td>
<td>230</td>
<td>$10^{30}$</td>
<td>9</td>
<td>18</td>
<td>312</td>
</tr>
<tr>
<td>10</td>
<td>23</td>
<td>230</td>
<td>$10^{32}$</td>
<td>9</td>
<td>†</td>
<td>†</td>
</tr>
</tbody>
</table>
Further Work

- a promising knapsack cryptosystem
- have clever ideas to defeat diophantine approximation
  - even under modular multiplication
- formalize hardness results for BR and IP reformulations
- need to consider other possible attacks
  - *better* BR algorithms
  - more numerically stable CPLEX
  - exact MIP solver (Espinoza et al.) and improvements