ECE 646 – Lecture 8

RSA:
Genesis, operation &
security.

Factorization in
Software & Hardware.

Public Key (Asymmetric) Cryptosystems

Public key of Bob - \( K_B \)

Private key of Bob - \( k_B \)

Alice

Encryption

Network

Decryption

Bob

Trap-door one-way function

Whitfield Diffie and Martin Hellman
"New directions in cryptography," 1976

PUBLIC KEY

\[ X \quad f(X) \quad Y \]

\[ f^{-1}(Y) \]

PRIVATE KEY
Professional (NSA) vs. amateur (academic) approach to designing ciphers

1. Know how to break Russian ciphers
2. Use only well-established proven methods
3. Hire 50,000 mathematicians
4. Cooperate with an industry giant
5. Keep as much as possible secret

1. Know nothing about cryptology
2. Think of revolutionary ideas
3. Go for skiing
4. Publish in “Scientific American”
5. Offer a $100 award for breaking the cipher
Challenge published in Scientific American

Ciphertext: 1977

9686 9613 7546 2206 1477 1409 2225 4355
8829 0575 9991 1245 7431 9874 6951 2093
0816 2982 2514 5708 3569 3147 6622 8839
8962 8013 3919 9055 1829 9451 5781 5145

Public key:

N = 114381625757 88886766923577997614
661201021829672124236256256184293
570693524573389783059712356395870
505989075147599290026879543541

e = 9007

(129 decimal digits)

Award: $100

RSA as a trap-door one-way function

\[ M = f^{-1}(C) = C^d \mod N \]

\[ N = P \cdot Q \quad P, Q - \text{large prime numbers} \]

\[ e \cdot d \equiv 1 \mod ((P-1)(Q-1)) \]
RSA keys

PUBLIC KEY
\{ e, N \}  

PRIVATE KEY
\{ d, P, Q \}  

\[ N = P \cdot Q \quad P, Q - \text{large prime numbers} \]
\[ e \cdot d = 1 \mod ((P-1)(Q-1)) \]

Why does RSA work? (1)

\[ M' = C^d \mod N = (M^e \mod N)^d \mod N = M \]
\text{decrypted} \quad \text{original} \quad \text{message} \quad \text{message} \n
\[ e \cdot d = 1 \mod ((P-1)(Q-1)) \quad \text{Euler's totient function} \]

Euler’s totient (phi) function (1)

\( \varphi(N) \) - number of integers in the range from 1 to N-1 that are relatively prime with N

Special cases:
1. P is prime
   \( \varphi(P) = P-1 \)
   Relatively prime with P: 1, 2, 3, ..., P-1
2. \( N = P \cdot Q \)  P, Q are prime
   \( \varphi(N) = (P-1) \cdot (Q-1) \)
   Relatively prime with N: \{1, 2, 3, ..., P-1 \} \setminus \{P, 2P, 3P, ..., (P-1)P\} \setminus \{Q, 2Q, 3Q, ..., (P-1)Q\}
Euler’s totient (phi) function (2)

Special cases:

3. \( N = P^2 \)  \( P \) is prime

\[ \varphi(N) = P \cdot (P-1) \]

Relatively prime with \( N \): \( \{1, 2, 3, \ldots, P^2-1\} - \{P, 2P, 3P, \ldots, (P-1)P\} \)

In general

If \( N = P_1^{e_1} \cdot P_2^{e_2} \cdot P_3^{e_3} \cdot \ldots \cdot P_t^{e_t} \)

\[ \varphi(N) = \prod_{i=1}^{t} P_i^{e_i-1} \cdot (P_i-1) \]

Euler’s Theorem

Leonard Euler, 1707-1783

\[ \forall \quad a^{\varphi(N)} \equiv 1 \pmod{N} \]

\( a: \gcd(a, N) = 1 \)

Euler’s Theorem - Justification (1)

For \( N=10 \)

\( R = \{1, 3, 7, 9\} \)

Let \( a=3 \)

\( S = \{3 \cdot 1 \mod 10, 3 \cdot 3 \mod 10, 3 \cdot 7 \mod 10, 3 \cdot 9 \mod 10\} = \{3, 9, 1, 7\} \)

For arbitrary \( N \)

\( R = \{x_1, x_2, \ldots, x_{\varphi(N)}\} \)

Let us choose arbitrary \( a \), such that \( \gcd(a, N) = 1 \)

\( S = \{a \cdot x_1 \mod N, a \cdot x_2 \mod N, \ldots, a \cdot x_{\varphi(N)} \mod N\} \)

= rearranged set \( R \)
### Euler’s Theorem - Justification (2)

**For N=10**

- \( R = S \)
- \( x_1 \cdot x_2 \cdot x_3 \cdot x_4 \equiv (a \cdot x_1) \cdot (a \cdot x_2) \cdot (a \cdot x_3) \cdot (a \cdot x_4) \mod N \)
- \( x_1 \cdot x_2 \cdot x_3 \cdot x_4 \equiv a^4 \cdot x_1 \cdot x_2 \cdot x_3 \cdot x_4 \mod N \)
- \( a^4 \equiv 1 \pmod{N} \)

**For arbitrary N**

- \( R = S \)
- \( \prod_{i=1}^{\varphi(N)} x_i = \prod_{i=1}^{\varphi(N)} a \cdot x_i \pmod{N} \)
- \( \prod_{i=1}^{\varphi(N)} x_i = a^{\varphi(N)} \cdot \prod_{i=1}^{\varphi(N)} x_i \pmod{N} \)
- \( a^{\varphi(N)} \equiv 1 \pmod{N} \)

### Why does RSA work? (2)

\[
M' = C^d \mod N = (M^e \mod N)^d \mod N = \\
= M^{ed} \mod N = \left\lfloor \frac{e \cdot d \cdot M \mod \varphi(N)}{e \cdot d \cdot M \mod \varphi(N) + k \cdot \varphi(N)} \right\rfloor = \\
= M^{1+k \cdot \varphi(N)} \mod N = M^{(M^{\varphi(N)})^k} \mod N = \\
= M \cdot (M^{\varphi(N)})^k \mod N = \\
= M \cdot 1^k \mod N = M
\]

### Rivest estimation - 1977

The best known algorithm for factoring a 129-digit number requires:

- **40 000 trillion years**
- **= 40 \cdot 10^{15} years**

assuming the use of a supercomputer
being able to perform
1 multiplication of 129 decimal digit numbers in 1 ns

*Rivest’s assumption translates to the delay of a single logic gate = 10 ps*

Estimated age of the universe: 100 bln years = 10^{19} years
### Early records in factoring large numbers

<table>
<thead>
<tr>
<th>Years</th>
<th>Number of decimal digits</th>
<th>Number of bits</th>
<th>Required computational power (in MIPS-years)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1974</td>
<td>45</td>
<td>149</td>
<td>0.001</td>
</tr>
<tr>
<td>1984</td>
<td>71</td>
<td>235</td>
<td>0.1</td>
</tr>
<tr>
<td>1991</td>
<td>100</td>
<td>332</td>
<td>7</td>
</tr>
<tr>
<td>1992</td>
<td>110</td>
<td>365</td>
<td>75</td>
</tr>
<tr>
<td>1993</td>
<td>120</td>
<td>398</td>
<td>830</td>
</tr>
</tbody>
</table>

### Number of bits vs. number of decimal digits

\[10^{\#digits} = 2^{\#bits}\]

\[\#digits = (\log_{10} 2) \cdot \#bits \approx 0.30 \cdot \#bits\]

- 256 bits = 77 D
- 384 bits = 116 D
- 512 bits = 154 D
- 768 bits = 231 D
- 1024 bits = 308 D
- 2048 bits = 616 D

### How to factor for free?

* A. Lenstra & M. Manasse, 1989

- Using the spare time of computers, (otherwise unused)

- Program and results sent by e-mail (later using WWW)
### Practical implementations of attacks

#### Factorization, RSA

<table>
<thead>
<tr>
<th>Year</th>
<th>Number of bits of N</th>
<th>Number of decimal digits of N</th>
<th>Method</th>
<th>Estimated amount of computations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1994</td>
<td>430</td>
<td>129</td>
<td>QS</td>
<td>5000 MIPS-years</td>
</tr>
<tr>
<td>1996</td>
<td>433</td>
<td>130</td>
<td>GNFS</td>
<td>750 MIPS-years</td>
</tr>
<tr>
<td>1998</td>
<td>467</td>
<td>140</td>
<td>GNFS</td>
<td>2000 MIPS-years</td>
</tr>
<tr>
<td>1999</td>
<td>467</td>
<td>140</td>
<td>GNFS</td>
<td>8000 MIPS-years</td>
</tr>
</tbody>
</table>

### Breaking RSA-129

**When:** August 1993 - April 1, 1994, 8 months

**Who:** D. Atkins, M. Graff, A. K. Lenstra, P. Leyland + 600 volunteers from the entire world

**How:** 1600 computers
- from Cray C90, through 16 MHz PC,
- to fax machines

*Only 0.03% computational power of the Internet*

**Results of cryptanalysis:**

"The magic words are squeamish ossifrage"

An award of $100 donated to Free Software Foundation

### Elements affecting the progress in factoring large numbers

- computational power
  
  1977-1993 increase of about 1500 times

- computer networks
  
  Internet

- better algorithms
Factoring methods

General purpose
Time of factoring depends only on the size of N

Special purpose
Time of factoring is much shorter if N or factors of N are of the special form

QS - Quadratic Sieve
GNFS - General Number Field Sieve
QS - Quadratic Sieve
Continued Fraction Method (historical)

ECM - Elliptic Curve Method
Pollard’s p-1 method
Cyclotomic polynomial method
SNFS - Special Number Field Sieve

Running time of factoring algorithms

\[ L_{\alpha, c}(N) = \exp((c+o(1))(\ln N)^{\alpha}(\ln \ln N)^{1-\alpha}) \]

For \( \alpha=0 \)

Algorithm polynomial as a function of the number of bits of \( q \)

\[ L_{0}[0, c] = (\ln N)^{c+o(1)} \]

For \( \alpha=1 \)

Algorithm exponential as a function of the number of bits of \( q \)

\[ L_{0}[1, c] = \exp((c+o(1))(\ln N)) \]

For \( 0 < \alpha < 1 \)

Algorithm subexponential as a function of the number of bits of \( q \)

\[ f(N) = o(1) \text{ if for any positive constant } c>0 \text{ there exist a constant } N_0>0, \text{ such that } 0 \leq f(N) < c, \text{ for all } N \geq N_0 \]

General purpose factoring methods

Expected running time

QS
NFS

\[ L_{\alpha}[1/2, 1] = \exp((1+o(1))(\ln N)^{1/2}(\ln \ln N)^{1/2}) \]

\[ L_{\alpha}[1/3, 1.92] = \exp((1.92+o(1))(\ln N)^{1/3}(\ln \ln N)^{2/3}) \]

<table>
<thead>
<tr>
<th>QS more efficient</th>
<th>NFS more efficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>100D</td>
<td>110D</td>
</tr>
<tr>
<td>120D</td>
<td>130D</td>
</tr>
</tbody>
</table>

size of the factored number \( N \) in decimal digits (D)
First RSA Challenge

Largest number factored to date

**RSA-200** May 2005

Second RSA Challenge

<table>
<thead>
<tr>
<th>Length of N in bits</th>
<th>Length of N in decimal digits</th>
<th>Award for factorization</th>
</tr>
</thead>
<tbody>
<tr>
<td>576</td>
<td>174</td>
<td>$10,000</td>
</tr>
<tr>
<td>640</td>
<td>193</td>
<td>$20,000</td>
</tr>
<tr>
<td>704</td>
<td>212</td>
<td>$30,000</td>
</tr>
<tr>
<td>768</td>
<td>232</td>
<td>$50,000</td>
</tr>
<tr>
<td>896</td>
<td>270</td>
<td>$75,000</td>
</tr>
<tr>
<td>1024</td>
<td>309</td>
<td>$100,000</td>
</tr>
<tr>
<td>1536</td>
<td>463</td>
<td>$150,000</td>
</tr>
<tr>
<td>2048</td>
<td>617</td>
<td>$200,000</td>
</tr>
</tbody>
</table>

Factoring 512-bit number

512 bits = 155 decimal digits
old standard for key sizes in RSA

17 March - 22 August 1999

Group of Herman te Riele
Centre for Mathematics and Computer Science (CWI), Amsterdam

**First stage** 2 months
168 workstations SGI and Sun, 175-400 MHz
120 Pentium PC, 300-450 MHz, 64 MB RAM
4 stations Digital/Compaq, 500 MHz

**Second stage**
Cray C916 - 10 days, 2.3 GB RAM
**Practical progress in factorization**

*March 2002, Financial Cryptography Conference*

Nicko van Someren, CTO nCipher Inc. announced that his company developed software capable of breaking 512-bit RSA key within 6 weeks using computers available in a single office.

---

**RSA vs. DES: Resistance to attack**

Number of operations in the best known attack

![Diagram showing comparison between DES (56-bit key) and 512-bit RSA with 1/50 NDES ratio.]

---

**Factoring RSA-576**

576 bits = 174 decimal digits

*When?*

Announced: December 3, 2003

*Who?*

J. Franke and T. Kleinjung
Bonn University
Max Planck Institute for Mathematics in Bonn
Experimental Mathematics Institute in Essen

P. Montgomery and H. te Riele - CWI
F. Bahr, D. Leclaire, P. Leyland and R. Wackerbarth

German Federal Agency for Information Technology Security (BIS)
### Factoring RSA-200

**When?**
- Dec 2003 - May 2005

**Who?**
- CWI (Netherlands), Bonn University,
- Max Planck Institute for Mathematics in Bonn
- Experimental Mathematics Institute in Essen
- German Federal Agency for Information Technology Security (BIS)

**Effort?**
- **First stage**
  - About 1 year on various machines, equivalent to 55 years on Opteron 2.2 GHz CPU
- **Second stage**
  - 3 months on a cluster of 80 2.2 GHz Opterons connected via a Gigabit network

### Factoring RSA-640

**When?**
- June 2005 - Nov 2005

**Who?**
- CWI (Netherlands), Bonn University,
- Max Planck Institute for Mathematics in Bonn
- Experimental Mathematics Institute in Essen
- German Federal Agency for Information Technology Security (BIS)

**Effort?**
- **First stage**
  - 3 months on 80 Opteron 2.2 GHz CPUs
- **Second stage**
  - 1.5 months on a cluster of 80 2.2 GHz Opterons connected via a Gigabit network

### Factoring RSA-768

**When?**

**Who?**
- Multiple researchers from EPFL, NTT, Bonn University, INRIA, MS Research, CWI

**Effort?**
- **Sieving time**
  - 3,300 Opteron 1 GHz CPU years
- **Total time**
  - 4,400 Opteron 1 GHz CPU years
Factorization records

<table>
<thead>
<tr>
<th>number</th>
<th>decimal digits</th>
<th>date</th>
<th>time (phase 1)</th>
<th>algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>C116</td>
<td>116</td>
<td>1990</td>
<td>275 MIPS years</td>
<td>mpqs</td>
</tr>
<tr>
<td>RSA-120</td>
<td>120</td>
<td>VI. 1993</td>
<td>830 MIPS years</td>
<td>mpqs</td>
</tr>
<tr>
<td>RSA-129</td>
<td>129</td>
<td>IV. 1994</td>
<td>5000 MIPS years</td>
<td>mpqs</td>
</tr>
<tr>
<td>RSA-130</td>
<td>130</td>
<td>IV. 1996</td>
<td>1000 MIPS years</td>
<td>gnfs</td>
</tr>
<tr>
<td>RSA-140</td>
<td>140</td>
<td>II. 1999</td>
<td>2000 MIPS years</td>
<td>gnfs</td>
</tr>
<tr>
<td>RSA-155</td>
<td>155</td>
<td>VIII. 1999</td>
<td>8000 MIPS years</td>
<td>gnfs</td>
</tr>
<tr>
<td>C158</td>
<td>158</td>
<td>I. 2002</td>
<td>3.4 Pentium 1GHz CPU years</td>
<td>gnfs</td>
</tr>
<tr>
<td>RSA-160</td>
<td>160</td>
<td>III. 2003</td>
<td>2.7 Pentium 1GHz CPU years</td>
<td>gnfs</td>
</tr>
<tr>
<td>RSA-576</td>
<td>174</td>
<td>XII. 2003</td>
<td>13.2 Pentium 1GHz CPU years</td>
<td>gnfs</td>
</tr>
<tr>
<td>C176</td>
<td>176</td>
<td>V. 2005</td>
<td>48.6 Pentium 1GHz CPU years</td>
<td>gnfs</td>
</tr>
<tr>
<td>RSA-200</td>
<td>200</td>
<td>V. 2005</td>
<td>121 Pentium 1GHz CPU years</td>
<td>gnfs</td>
</tr>
<tr>
<td>RSA-768</td>
<td>232</td>
<td>XII. 2009</td>
<td>4,400 Opteron 1 GHz CPU years</td>
<td>gnfs</td>
</tr>
</tbody>
</table>

He who has absolute confidence in linear regression will expect a 1024-bit RSA number to be factored on December 17, 2028.

For the most recent records see

Factorization Announcements & Records at

http://www.crypto-world.com/FactorAnnouncements.html

Estimation of RSA Security Inc. regarding the number and memory of PCs necessary to break RSA-1024

Attack time: 1 year
Single machine: PC, 500 MHz, 170 GB RAM
Number of machines: 342,000,000

Best Algorithm to Factor Large Numbers

NUMBER FIELD SIEVE

Complexity: Sub-exponential time and memory
\( N = \text{Number to factor}, \quad k = \text{Number of bits of } N \)

Polynomial function, \( a \cdot k^m \)
Exponential function, \( e^k \)
Sub-exponential function, \( e^{10^{0.7353k^{1/3}}(\ln k)^{2/3}} \)

Execution time

Factoring 1024-bit RSA keys using Number Field Sieve (NFS)

Polynomial Selection

Relation Collection

Sieving 200 bit & 350 bit smooth numbers
Minifactoring (Cofactoring, Norm Factoring) ECM, p-1 method, rho method
Linear Algebra
Square Root
TWINKLE
“The Weizmann Institute Key Locating Engine”

Adi Shamir, Eurocrypt, May 1999
CHES, August 1999

Electrooptical device capable to speed-up
the first phase of factorization from 100 to 1000 times

If ever built it would increase the size of the key
that can be broken from 100 to 200 bits

Cost of the device (assuming that the prototype was
earlier built) - $5000

Bernstein’s Machine (1)

Fall 2001

Daniel Bernstein, professor of mathematics
at University of Illinois in Chicago
submits a grant application to NSF
and publishes fragments of this application
as an article on the web

D. Bernstein, Circuits for Integer Factorization: A Proposal
http://cr.yp.to/papers.html#nfscircuit

Bernstein’s Machine (2)

March 2002

• Bernstein’s article “discovered” during
Financial Cryptography Conference

• Informal panel devoted to analysis of consequences
of the Bernstein’s discovery

• Nicko Van Someren (nCipher) estimates that machine
costing $1 billion is able to break 1024-bit RSA
within several minutes
March 2002

- **alarming voices** on e-mailing discussion lists
calling for revocation of all currently used
1024-bit keys

- **sensational articles** in newspapers about
Bernstein’s discovery

April 2002

**Response of the RSA Security Inc.:**

Error in the estimation presented at the conference;
according to formulas from the Bernstein’s article
machine costing

\$ 1 billion \text{ is able to break}
1024-bit RSA within

\( 10 \text{ billion} \times \text{ several minutes} = \text{ tens of years} \)

According to estimations of Lenstra i Verheul, machine
breaking 1024-bit RSA within \textit{one day}
would cost \$ 160 billion in 2002

- **Bernstein’s Machine (3)**

- **Bernstein’s Machine (4)**

- **Bernstein’s Machine (5)**

  **Carl Pomerance, Bell Labs:**

  “….fresh and fascinating idea...”

  **Arjen Lenstra, Citibank & U. Eindhoven:**

  “….I have no idea what is this all fuss about...”

  **Bruce Schneier, Counterpane:**

  “... enormous improvements claimed are more a result
  of redefining efficiency than anything else...”
Bernstein’s Machine (6)

RSA keylength that can be broken using Bernstein’s machine

RSA key lengths that can be broken using classical computers

Computational cost = time [days] * memory [$]

TWIRL

February 2003

Adi Shamir & Eran Tromer, Weizmann Institute of Science

Hardware implementation of the sieving phase of Number Field Sieve (NFS)

Assumed technology:
- CMOS, 0.13 µm
- clock 1 GHz
- 30 cm semiconductor wafers at the cost of $5,000 each

TWIRL

A. Shamir, E. Tromer
Crypto 2003

Tentative estimations
(no experimental data):

512-bit RSA:
- < 10 minutes
- $10k

1024-bit RSA:
- < 1 year
- $10 million
Theoretical Designs for Sieving (1)

1999-2000
TWINKLE (Shamir, CHES 1999; Shamir & Lenstra, Eurocrypt 2000)
- based on optoelectronic devices (fast LEDs)
- not even a small prototype built in practice
- not suitable for 1024 bit numbers

2003
TWIRL (Shamir & Tromer, Crypto 2003)
- semiconductor wafer design
- requires fast communication between chips located on the same 30 cm diameter wafer
- difficult to realize using current fabrication technology

Theoretical Designs for Sieving (2)

2003-2004
Mesh Based Sieving / YASD
(Geiselmann & Steinwandt, PKC 2003
Geiselmann & Steinwandt, CT-RSA 2004)
- not suitable for 1024 bit numbers

2005
SHARK (Franke et al., SHARCS & CHES 2005)
- relies on an elaborate butterfly switch connecting large number of chips
- difficult to realize using current technology

Theoretical Designs for Sieving (3)

2007
Non-Wafer-Scale Sieving Hardware
(Geiselmann & Steinwandt, Eurocrypt 2007)
- based on moderate size chips (2.2 x 2.2 cm)
- communication among chips seems to be realistic
- 2 to 3.5 times slower than TWIRL
- supports only linear sieving, and not more optimal lattice sieving
Estimated recurring costs with current technology (US$\times$year)

by Eran Tromer, May 2005

<table>
<thead>
<tr>
<th></th>
<th>768-bit</th>
<th>1024-bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traditional PC-based</td>
<td>$1.3\times10^7$</td>
<td>$10^{12}$</td>
</tr>
<tr>
<td>TWINKLE</td>
<td>$8\times10^6$</td>
<td></td>
</tr>
<tr>
<td>TWIRL</td>
<td>$5\times10^3$</td>
<td>$10^6$</td>
</tr>
<tr>
<td>Mesh-based</td>
<td>$3\times10^4$</td>
<td></td>
</tr>
<tr>
<td>SHARK</td>
<td></td>
<td>$230\times10^6$</td>
</tr>
</tbody>
</table>

But: non-recurring costs, chip size, chip transport networks...

However...

None of the theoretical designs ever built.

Just analytical estimations, no real implementations, no concrete numbers

First Practical Implementation of the Relation Collection Step in Hardware

2007

Japan  Tetsuya Izu and Jun Kogure
and Takeshi Shimoyama (Fujitsu)

CHES 2007 - CAIRN 2 machine, September 2007
SHARCS 2007 – CAIRN 3 machine, September 2007
First large number factored using FPGA support

Factored number:

\[ N = P \cdot Q \]

- 423-bits
- 205 bits
- 218 bits

Time of computations:

One month of computations using a PC supported by CAIRN 2 for a 423-bit number

CAIRN 3 about 40 times faster than CAIRN 2

Time of sieving with CAIRN 3 for a 768-bit key estimated at 270 years

Problems:

- Speed up vs. one PC (AMD Opteron): only about 4 times
- Limited scalability

Workshop Series

SHARCS - Special-purpose Hardware for Attacking Cryptographic Systems

1\textsuperscript{st} edition: Paris, Feb. 24-25, 2005
2\textsuperscript{nd} edition: Cologne, Apr. 3-4, 2006
3\textsuperscript{rd} edition: Vienna, Sep. 9-10, 2007
4\textsuperscript{th} edition: Lausanne, Sep. 9-10, 2009


Keylengths in public key cryptosystems that provide the same level of security as AES and other secret-key ciphers

Arjen K. Lenstra, Eric R. Verheul

*Selecting Cryptographic Key Sizes*

*Journal of Cryptology, 2001*

Arjen K. Lenstra

*Unbelievable Security: Matching AES Security Using Public Key Systems*

ASIACRYPT’ 2001
Keylengths in RSA providing the same level of security as selected secret-key cryptosystems

The recommended key sizes for RSA
RSA Laboratories, 1996

Old standard:
Individual users
- 512 bits (155 decimal digits)

New standard:
Individual users
- 768 bits (231 decimal digits)

Organizations (short term)
- 1024 bits (308 decimal digits)

Organizations (long term)
- 2048 bits (616 decimal digits)
Recommendations of RSA Security Inc.
May 6, 2003

<table>
<thead>
<tr>
<th>Validity period</th>
<th>Minimal RSA key length (bits)</th>
<th>Equivalent symmetric key length (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2003-2010</td>
<td>1024</td>
<td>80</td>
</tr>
<tr>
<td>2010-2030</td>
<td>2048</td>
<td>112</td>
</tr>
<tr>
<td>2030-</td>
<td>3072</td>
<td>128</td>
</tr>
</tbody>
</table>

Five security levels allowed by American government
NIST SP 800-56

<table>
<thead>
<tr>
<th>Level</th>
<th>RSA / DH</th>
<th>ECC</th>
<th>Symmetric ciphers</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>1024</td>
<td>160</td>
<td>80</td>
</tr>
<tr>
<td>II</td>
<td>2048</td>
<td>224</td>
<td>112</td>
</tr>
<tr>
<td>III</td>
<td>3072</td>
<td>256</td>
<td>128</td>
</tr>
<tr>
<td>IV</td>
<td>8192</td>
<td>384</td>
<td>192</td>
</tr>
<tr>
<td>V</td>
<td>15360</td>
<td>512</td>
<td>256</td>
</tr>
</tbody>
</table>

Discovery of public key cryptography by British Intelligence
CESG - Communications-Electronics Security Group
British intelligence agency existing for over 80 yrs

Employees of CESG discovered the idea of public key cryptography, the RSA cryptosystem and the Diffie-Hellman key agreement scheme several years before their discovery in open research

Story disclosed only in December 1997
<table>
<thead>
<tr>
<th><strong>General concept of public key cryptography</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Secret research</strong></td>
<td><strong>Open research</strong></td>
</tr>
<tr>
<td><em>James H. Ellis</em></td>
<td><em>Whit Diffie, Martin Hellman</em></td>
</tr>
<tr>
<td>“The possibility of Secure Non-Secret Digital Encryption”</td>
<td>“New Directions in Cryptography”</td>
</tr>
<tr>
<td><strong>January 1970</strong></td>
<td><strong>November 1976</strong></td>
</tr>
<tr>
<td>• proof of a possibility of constructing non-secret-key cryptography</td>
<td>• example of a public-key agreement scheme</td>
</tr>
<tr>
<td></td>
<td>• concept of a digital signature</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>RSA cryptosystem</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Secret research</strong></td>
<td><strong>Open research</strong></td>
</tr>
<tr>
<td><em>Clifford Cocks</em></td>
<td><em>Ron Rivest, Adi Shamir, Martin Hellman</em></td>
</tr>
<tr>
<td><strong>November 1973</strong></td>
<td><strong>January 1978</strong></td>
</tr>
<tr>
<td>$C = M^N \mod N$</td>
<td>$C = M^e \mod N$</td>
</tr>
<tr>
<td>Decryption always based on the Chinese Remainder Theorem</td>
<td>Decryption based on the Chinese Remainder Theorem optional</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Diffie-Hellman Key Agreement Scheme</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Secret Research</strong></td>
<td><strong>Open Research</strong></td>
</tr>
<tr>
<td><em>Malcolm Williamson</em></td>
<td><em>Whit Diffie, Martin Hellman</em></td>
</tr>
<tr>
<td><strong>1974</strong></td>
<td><strong>June 1976</strong></td>
</tr>
</tbody>
</table>
Discovery of the public key cryptography by British Intelligence

- Discovery in the secret research had only historical significance
- Discovery in the open research initiated the revolution in cryptography
- British Intelligence never considered applying public key cryptography for digital signatures
- It is still unclear whether and if so when public key cryptography was discovered by NSA.

ACM A.M. Turing Award 2002

R. Rivest
A. Shamir
L. Adleman

“For Seminal Contributions to the Theory and Practical Applications of Public Key Cryptography”

Turing Award Lectures

Dr. Leonard M. Adleman
University of Southern California
Pre RSA Days

Dr. Ronald L. Rivest
Massachusetts Institute of Technology
Early RSA Days

Dr. Adi Shamir
The Weizmann Institute
Cryptology: A Status Report
Bases of the public cryptosystems security

<table>
<thead>
<tr>
<th>Factorization</th>
<th>Discrete Logarithm</th>
<th>Elliptic Curve Discrete Logarithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Given: $N = p \cdot q$</td>
<td>$y = g^x \mod p = g \cdot g \cdot g \cdot \ldots \cdot g$ \text{ x times}</td>
<td>$Q = x \cdot P = P + P + \ldots + P$ \text{ x times}$</td>
</tr>
<tr>
<td>Unknown: $p$, $q$</td>
<td>$x$</td>
<td>$x$</td>
</tr>
</tbody>
</table>

Best known attacks

<table>
<thead>
<tr>
<th>Basis of the cryptosystem security</th>
<th>Factorization</th>
<th>Discrete Logarithm</th>
<th>Elliptic Curve Discrete Logarithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best known attack</td>
<td>General Number Field Sieve</td>
<td>1. General Number Field Sieve</td>
<td>2. Parallel collision search</td>
</tr>
<tr>
<td>Complexity of the attack:</td>
<td>subexponential</td>
<td>1. subexponential</td>
<td>exponential</td>
</tr>
<tr>
<td>2. exponential</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Best known attacks

<table>
<thead>
<tr>
<th>Basis of the cryptosystem security</th>
<th>Factorization</th>
<th>Discrete Logarithm</th>
<th>Elliptic Curve Discrete Logarithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cryptosystem</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RSA</td>
<td>DSA, DH</td>
<td></td>
<td>EC-DSA</td>
</tr>
<tr>
<td>DSA, N-R</td>
<td></td>
<td></td>
<td>EC-DH</td>
</tr>
<tr>
<td>Modulus N</td>
<td></td>
<td></td>
<td>Size q of the subgroup generated by P</td>
</tr>
<tr>
<td>1. Length of the modulus p</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Size q of the subgroup generated by g</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Typical lengths of the security parameter (in bits)</td>
<td>1024, 2048</td>
<td>1. 1024, 2048</td>
<td>160, 224</td>
</tr>
<tr>
<td>2. 160 (for DSA)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Theoretical computational security of the best known attacks

<table>
<thead>
<tr>
<th>Basis of the cryptosystem security</th>
<th>Complexity of the best known attack</th>
</tr>
</thead>
<tbody>
<tr>
<td>Factorization</td>
<td>subexponential</td>
</tr>
<tr>
<td></td>
<td>$L_{[1/3, 1.92]} = \exp((1.92 + o(1))(\ln N)^{1/3})(\ln \ln N)^{2/3}$</td>
</tr>
<tr>
<td>Discrete Logarithm</td>
<td>subexponential</td>
</tr>
<tr>
<td></td>
<td>$L_{[1/3, 1.92]} = \exp((1.92 + o(1))(\ln p)^{1/3})(\ln \ln p)^{2/3}$</td>
</tr>
<tr>
<td>Elliptic Curve Discrete Logarithm</td>
<td>exponential</td>
</tr>
<tr>
<td></td>
<td>$(\pi q / 2)^{1/2} / r$</td>
</tr>
<tr>
<td></td>
<td>r - number of processors working in parallel</td>
</tr>
</tbody>
</table>

### Most known public key cryptosystems

<table>
<thead>
<tr>
<th>Based on the difficulty of</th>
<th>Signature</th>
<th>Encryption</th>
<th>Key agreement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Factorization</td>
<td>RSA</td>
<td>RSA</td>
<td>RSA</td>
</tr>
<tr>
<td>Discrete logarithm</td>
<td>DSA, N-R</td>
<td>El-Gamal</td>
<td>Diffie-Hellman (DH)</td>
</tr>
<tr>
<td>Elliptic curve discrete logarithm</td>
<td>EC-DSA</td>
<td>EC-EI-Gamal</td>
<td>EC-DH</td>
</tr>
</tbody>
</table>
### IEEE P1363

<table>
<thead>
<tr>
<th></th>
<th>Factorization</th>
<th>Discrete logarithm</th>
<th>Elliptic curve discrete logarithm</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>encryption</strong></td>
<td>RSA with OAEP</td>
<td>new scheme</td>
<td>new scheme</td>
</tr>
<tr>
<td><strong>signature</strong></td>
<td>RSA &amp; R-W with ISO-14888 or ISO 9796</td>
<td>DSA, NR with ISO 9796</td>
<td>EC-DSA, EC-NR with ISO 9796</td>
</tr>
<tr>
<td><strong>key agreement</strong></td>
<td>new scheme</td>
<td>DH1, DH2 &amp; MQV</td>
<td>EC-DH1, EC-DH2 &amp; EC-MQV</td>
</tr>
</tbody>
</table>

### IEEE P1363a

<table>
<thead>
<tr>
<th></th>
<th>Factorization</th>
<th>Discrete logarithm</th>
<th>Elliptic curve discrete logarithm</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>encryption</strong></td>
<td>RSA with OAEP</td>
<td>new scheme</td>
<td>new scheme</td>
</tr>
<tr>
<td><strong>signature</strong></td>
<td>RSA &amp; R-W with ISO-14888 or ISO 9796</td>
<td>DSA, NR with ISO-9796</td>
<td>EC-DSA, EC-NR with ISO 9796</td>
</tr>
<tr>
<td><strong>key agreement</strong></td>
<td>new scheme</td>
<td>DH1, DH2 &amp; MQV</td>
<td>EC-DH1, EC-DH2 &amp; EC-MQV</td>
</tr>
</tbody>
</table>

### IEEE P1363

Working group of IEEE including representatives of major cryptographic companies and university centers from USA, Canada and other countries

Part of the Microprocessors Standards Committee

Modern, open style

Quarterly meetings + multiple teleconferences + discussion list + very informative web page with the draft versions of standards
IEEE P1363

Combined standard including the majority of modern public key cryptography
Several algorithms for implementation of the same function

Tool for constructing other, more specific standards
Specific applications or implementations may determine a profile (subset) of the standard