Algorithm Theory
5 Randomized Algorithms: Public Key Cryptosystems

Christian Schindelhauer

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Wintersemester 2007/08
Randomized algorithms

- Classes of randomized algorithms
- Randomized Quicksort
- Randomized primality test
- Cryptography
Classes of randomized algorithms

- **Las Vegas algorithms**
  - **always correct**; expected running time ("probably fast")
  - Examples:
    - randomized Quicksort,
    - randomized algorithm for closest pair

- **Monte Carlo algorithms (mostly correct):**
  - **probably correct**; guaranteed running time
  - Example: randomized primality test
Application: cryptosystems

Traditional encryption of messages with secret keys

Disadvantages:

1. The key $k$ has to be exchanged between A and B before the transmission of the message.
2. For messages between $n$ parties $n(n-1)/2$ keys are required.

Advantage:

Encryption and decryption can be computed very efficiently.
Duties of security providers

Guarantee…

• confidential transmission
• integrity of data
• authenticity of the sender
• reliable transmission
Public-key cryptosystems

Diffie and Hellman (1976)

Idea: Each participant A has two keys:

1. a **public** key $P_A$ accessible to every other participant

2. a **private** (or: **secret**) key $S_A$ only known to $A$. 
Public-key cryptosystems

\[ D = \text{set of all legal messages,} \]
\[ \text{e.g. the set of all bit strings of finite length} \]

\[ P_A(\ ), S_A(\ ): D \rightarrow D \]

**Three conditions:**

1. \( P_A \) and \( S_A \) can be computed efficiently

2. \( S_A(P_A(M)) = M \) and \( P_A(S_A(M)) = M \)
   \( (P_A, S_A \text{ are inverse functions}) \)

3. \( S_A \text{ cannot be computed from } P_A \) (with reasonable effort)
Encryption in a public-key system

A sends a message $M$ to $B$.

Dear Bob,
I just checked the new ...

$\#^k- + ;}{?, @-) #$<9 {o7::-&$3 (-##!]?8 ...

Dear Bob,
I just checked the new ...

Encryption in a public-key system

1. A accesses B’s public key $P_B$ (from a public directory or directly from B).
2. A computes the encrypted message $C = P_B(M)$ and sends $C$ to B.
3. After B has received message C, B decrypts the message with his own private key $S_B$: $M = S_B(C)$
Generating a digital signature

A sends a digitally signed message $M'$ to B:

1. A computes the digital signature $\sigma$ for $M'$ with her own private key:
   \[ \sigma = S_A(M') \]

2. A sends the pair $(M', \sigma)$ to B.

3. After receiving $(M', \sigma)$, B verifies the digital signature:
   \[ P_A(\sigma) = M' \]

$\sigma$ can be verified by anybody via the public $P_A$. 
RSA cryptosystems

R. Rivest, A. Shamir, L. Adleman

Generating the public and private keys:

1. Randomly select two primes $p$ and $q$ of similar size, each with $l+1$ bits ($l \geq 500$).

2. Let $n = p \cdot q$

3. Let $e$ be an integer that does not divide $(p - 1)(q - 1)$.

4. Calculate $d = e^{-1} \mod (p - 1)(q - 1)$

   i.e.: $d \cdot e \equiv 1 \mod (p - 1)(q - 1)$
RSA cryptosystems

5. Publish $P = (e, n)$ as public key

6. Keep $S = (d, n)$ as private key

Divide message (represented in binary) in blocks of size $2^{-l}$.
Interpret each block $M$ as a binary number: $0 \leq M < 2^{2^{-l}}$

$$P(M) = M^e \mod n \quad S(C) = C^d \mod n$$
Multiplicative Inverse

- **Theorem** (GCD recursion theorem)
  - For any numbers $a$ and $b$ with $b > 0$
    $\text{GCD}(a, b) = \text{GCD}(b, a \mod b)$

- **Algorithm Euclid**
  **Input**: Two integers $a$ and $b$ with $b \geq 0$
  **Output**: $\text{GCD}(a, b)$
  
  ```plaintext
  if $b = 0$
    then return $a$
  else return $\text{Euclid}(b, a \mod b)$
  ```
Multiplicative Inverse

- **Algorithm** Extended-Euclid
  - **Input:** Two integers $a$ and $b$ with $b \geq 0$
  - **Output:** $\text{GCD}(a,b)$ and two integers $x$ and $y$ with $xa+yb=\text{GCD}(a,b)$
    - if $b=0$ then return $(a,1,0)$
    - else $(d,x',y') := \text{Extended-Euclid}(b,a \mod b)$
    - $x := y'$; $y = x' - \lfloor a/b \rfloor y'$
    - return $(d,x,y)$

- **Application:** $a=(p-1)(q-1)$, $b = u$
  The algorithm returns numbers $x$ and $y$ with
  $$x(p-1)(q-1) + ye = \text{GCD}((p-1)(q-1),e) = 1$$
Algorithm Theory
5 Randomized Algorithms: Public Key Cryptosystems

Christian Schindelhauer
Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Wintersemester 2007/08