ROM-RSA, Rabin, Diffie-Hellman, El Gamal, and Discrete Logarithms

Douglas Wikström
KTH Stockholm
dog@csc.kth.se

February 18

- ROM-RSA
- Rabin
- Diffie-Hellman
- El Gamal
- Discrete Logarithms

The RSA Assumption

Definition. The RSA assumption states that if:

1. $N = pq$ factors into two randomly chosen primes p and q of the same bit-size,
2. e is in $\mathbb{Z}^*_{\phi(N)}$,
3. m is randomly chosen in $\mathbb{Z}^*_N$,

then for every polynomial time algorithm A

$$\Pr[A(N, e, m^e \bmod N) = m]$$

is negligible.

Semantically Secure ROM-RSA (1/2)

Suppose that $f : \{0,1\}^n \rightarrow \{0,1\}^n$ is a randomly chosen function (a random oracle).

- **Key Generation.** Choose a random RSA key pair $((N,e),(p,q,d))$, with $\log_2 N = n$.
- **Encryption.** Encrypt a plaintext $m \in \{0,1\}^n$ by choosing $r \in \mathbb{Z}_N^*$ randomly and computing

$$(u,v) = (r^e \bmod N, f(r) \oplus m) \ .$$

- **Decryption.** Decrypt a ciphertext (u,v) by

$$m = v \oplus f(u^d) \ .$$

Semantically Secure RSA in the ROM (2/2)

- We increase the ciphertext size by a factor of two.
- Our analysis is in the random oracle model, **which is unsound!**

Semantically Secure RSA in the ROM (2/2)

- We increase the ciphertext size by a factor of two.
- Our analysis is in the random oracle model, **which is unsound!**

Solutions.

- Using a "optimal" padding the first problem can be reduced. See standard OAEP+.

Semantically Secure RSA in the ROM (2/2)

- We increase the ciphertext size by a factor of two.
- Our analysis is in the random oracle model, **which is unsound!**

Solutions.

- Using a "optimal" padding the first problem can be reduced. See standard OAEP+.
- Using a scheme with much lower rate, the second problem can be removed.

Semantically Secure RSA in the ROM (2/2)

- We increase the ciphertext size by a factor of two.
- Our analysis is in the random oracle model, **which is unsound!**

Solutions.

- Using a "optimal" padding the first problem can be reduced. See standard OAEP+.
- Using a scheme with much lower rate, the second problem can be removed.
- If the key of an ideal cipher is encrypted, then we can avoid the ROM and still have "optimal padding"

Rabin's Cryptosystem (1/3)

Key Generation.

- Choose n-bit primes p and q such that $p, q = 3 \bmod 4$ randomly and define $N = pq$.
- Output the key pair $(N, (p, q))$, where (N, e) is the public key and (p, q) is the secret key.

Rabin's Cryptosystem (2/3)

Encryption. Encrypt a plaintext m by computing

$$c = m^2 \bmod N \ .$$

Decryption. Decrypt a ciphertext c by computing

$$m = \sqrt{c} \bmod N \ .$$

Rabin's Cryptosystem (2/3)

Encryption. Encrypt a plaintext m by computing

$$c = m^2 \bmod N \enspace .$$

Decryption. Decrypt a ciphertext c by computing

$$m = \sqrt{c} \bmod N \enspace .$$

There are **four** roots, so which one should be used!

Rabin's Cryptosystem (3/3)

Suppose y is a quadratic residue modulo p.

$$\begin{aligned}\left(\pm y^{(p+1)/4}\right)^2 &= y^{(p+1)/2} \bmod p \\ &= y^{(p-1)/2} y \bmod p \\ &= \left(\frac{y}{p}\right) y \\ &= y \bmod p\end{aligned}$$

In Rabin's cryptosystem:

- Find roots for $y_p = y \bmod p$ and $y_q = y \bmod q$.
- Combine roots to get the four roots modulo N. Choose the "right" root and output the plaintext.

Security of Rabin's Cryptosystem

Theorem. Breaking Rabin's cryptosystem is equivalent to factoring (provided we do not derandomize decryption!).

Idea.

1. Choose random element r.
2. Hand $r^2 \bmod N$ to adversary.
3. Consider outputs r' from the adversary such that $(r')^2 = r^2 \bmod N$. Then $r' \neq \pm r \bmod N$, with probability $1/2$, in which case $\gcd(r' - r, N)$ gives a factor of N.

A Goldwasser-Micali Variant of Rabin

Theorem [CG98]. If factoring is hard and r is a random quadratic residue modulo N, then for every polynomial time algorithm A

$$\Pr[A(N, r^2 \bmod N) = \mathrm{lsb}(r)]$$

is negligible.

- **Encryption.** Encrypt a plaintext $m \in \{0, 1\}$ by choosing a random quadratic residue r modulo N and computing

$$(u, v) = (r^2 \bmod N, \mathrm{lsb}(r) \oplus m) \ .$$

- **Decryption.** Decrypt a ciphertext (u, v) by

$$m = v \oplus \mathrm{lsb}(\sqrt{u}) \quad \text{where } \sqrt{u} \text{ is a quadratic residue} \ .$$

Diffie-Hellman Key Exchange (1/3)

Diffie and Hellman asked themselves:

How can two parties efficiently agree on a secret key using only **public** communication?

Diffie-Hellman Key Exchange (2/3)

Construction.
Let G be a cyclic group of order q with generator g.

1. - Alice picks $a \in \mathbb{Z}_q$ randomly, computes $y_a = g^a$ and hands y_a to Bob.
   - Bob picks $b \in \mathbb{Z}_q$ randomly, computes $y_b = g^b$ and hands y_b to Alice.
2. - Alice computes $k = y_b^a$.
   - Bob computes $k = y_a^b$.
3. The joint secret key is k.

Diffie-Hellman Key Exchange (3/3)

Problems.

- Susceptible to man-in-the-middle attack without authentication.
- How do we map a random element $k \in G$ to a random symmetric key in $\{0, 1\}^n$?

The El Gamal Cryptosystem (1/2)

Definition. Let G be a cyclic group of order q with generator g.

- The **key generation** algorithm chooses a random element $x \in \mathbb{Z}_q$ as the private key and defines the public key as
$$y = g^x \enspace .$$
- The **encryption** algorithm takes a message $m \in G$ and the public key y, chooses $r \in \mathbb{Z}_q$, and outputs the pair
$$(u, v) = \mathrm{E}_y(m, r) = (g^r, y^r m) \enspace .$$
- The **decryption** algorithm takes a ciphertext (u, v) and the secret key and outputs
$$m = \mathrm{D}_x(u, v) = vu^{-x} \enspace .$$

The El Gamal Cryptosystem (2/2)

- El Gamal is essentially Diffie-Hellman + OTP.
- Homomorphic property (with public key y)

$$\mathsf{E}_y(m_0, r_0)\mathsf{E}_y(m_1, r_1) = \mathsf{E}_y(m_0 m_1, r_0 + r_1) \ .$$

This property is very important in the construction of cryptographic protocols!

Discrete Logarithm (1/2)

Definition. Let G be a cyclic group of order q and let g be a generator G. The **discrete logarithm** of $y \in G$ in the basis g (written $\log_g y$) is defined as the unique $x \in \{0, 1, \ldots, q-1\}$ such that

$$y = g^x \quad .$$

Compare with a "normal" logarithm! ($\ln y = x$ iff $y = e^x$)

Discrete Logarithm (2/2)

Example. 7 is a generator of $\mathbb{Z}_{12}$ additively, since $\gcd(7, 12) = 1$.

What is $\log_7 3$?

Discrete Logarithm (2/2)

Example. 7 is a generator of $\mathbb{Z}_{12}$ additively, since $\gcd(7, 12) = 1$.

What is $\log_7 3$? $(9 \cdot 7 = 63 = 3 \bmod 12$, so $\log_7 3 = 9)$

Discrete Logarithm (2/2)

Example. 7 is a generator of $\mathbb{Z}_{12}$ additively, since $\gcd(7, 12) = 1$.

What is $\log_7 3$? $(9 \cdot 7 = 63 = 3 \bmod 12$, so $\log_7 3 = 9)$

Example. 7 is a generator of $\mathbb{Z}_{13}^*$.

What is $\log_7 9$?

Discrete Logarithm (2/2)

Example. 7 is a generator of $\mathbb{Z}_{12}$ additively, since $\gcd(7, 12) = 1$.

What is $\log_7 3$? ($9 \cdot 7 = 63 = 3 \bmod 12$, so $\log_7 3 = 9$)

Example. 7 is a generator of $\mathbb{Z}_{13}^*$.

What is $\log_7 9$? ($7^4 = 9 \bmod 13$, so $\log_7 9 = 4$)

Discrete Logarithm Assumption

Let G_{q_n} be a cyclic group of prime order q_n such that $\lfloor \log_2 q_n \rfloor = n$ for $n = 2, 3, 4, \ldots$, and denote the family $\{G_{q_n}\}_{n \in \mathbb{N}}$ by G.

Definition. The **Discrete Logarithm Assumption (DLA)** in G states that if generators g_n and y_n of G_{q_n} are randomly chosen, then for every polynomial time algorithm A

$$\Pr\left[A(g_n, y_n) = \log_{g_n} y_n\right]$$

is negligible.

Discrete Logarithm Assumption

Let G_{q_n} be a cyclic group of prime order q_n such that $\lfloor \log_2 q_n \rfloor = n$ for $n = 2, 3, 4, \ldots$, and denote the family $\{G_{q_n}\}_{n \in \mathbb{N}}$ by G.

Definition. The **Discrete Logarithm Assumption (DLA)** in G states that if generators g and y of G are randomly chosen, then for every polynomial time algorithm A

$$\Pr\left[A(g, y) = \log_g y\right]$$

is negligible.

We usually remove the indices from our notation!

Diffie-Hellman Assumption

Definition. Let g be a generator of G. The **Diffie-Hellman Assumption (DHA)** in G states that if $a, b \in \mathbb{Z}_q$ are randomly chosen, then for every polynomial time algorithm A

$$\Pr\left[A(g^a, g^b) = g^{ab}\right]$$

is negligible.

Decision Diffie-Hellman Assumption

Definition. Let g be a generator of G. The **Decision Diffie-Hellman Assumption (DDHA)** in G states that if $a, b, c \in \mathbb{Z}_q$ are randomly chosen, then for every polynomial time algorithm A

$$\left|\Pr\left[A(g^a, g^b, g^{ab}) = 1\right] - \Pr\left[A(g^a, g^b, g^c) = 1\right]\right|$$

is negligible.

Relating DL Assumptions

- Computing discrete logarithms is at least as hard as computing a Diffie-Hellman element g^{ab} from g^a and g^b.
- Computing a Diffie-Hellman element g^{ab} from g^a and g^b is at least as hard as distinguishing a Diffie-Hellman triple (g^a, g^b, g^{ab}) from a random triple (g^a, g^b, g^c).
- In most groups where the DLA is conjectured, DHA and DDHA are conjectured as well.
- There exists special elliptic curves where DDHA is easy, but DHA is conjectured!

Security of El Gamal

- Finding the secret key is equivalent to DLA.
- Finding the plaintext from the ciphertext and the public key and is equivalent to DHA.
- The semantic security of El Gamal is equivalent to DDHA.

Brute Force and Shank's

Let G be a cyclic group of order q and g a generator. We wish to compute $\log_g y$.

- **Brute Force.** $O(q)$
- **Shanks.** Time and **Space**$O\left(\sqrt{q}\right)$.
  1. Set $z = g^m$.
  2. Compute z^i for $0 \leq i \leq q/m$.
  3. Find $0 \leq j \leq m$ and $0 \leq i \leq q/m$ such that $yg^j = z^i$ and output $x = mi - j$.

Pollard-ρ (1/2)

Partition G into S_1, S_2, and S_3 "randomly".

- Generate "random" sequence $\alpha_0, \alpha_1, \alpha_2 \ldots$

$$\alpha_0 = g$$

$$\alpha_i = \begin{cases} \alpha_{i-1} g & \text{if } \alpha_{i-1} \in S_1 \\ \alpha_{i-1}^2 & \text{if } \alpha_{i-1} \in S_2 \\ \alpha_{i-1} y & \text{if } \alpha_{i-1} \in S_3 \end{cases}$$

Pollard-ρ (1/2)

Partition G into S_1, S_2, and S_3 “randomly”.

- Generate “random” sequence $\alpha_0, \alpha_1, \alpha_2 \ldots$

$$\alpha_0 = g$$
$$\alpha_i = \begin{cases} \alpha_{i-1}g & \text{if } \alpha_{i-1} \in S_1 \\ \alpha_{i-1}^2 & \text{if } \alpha_{i-1} \in S_2 \\ \alpha_{i-1}y & \text{if } \alpha_{i-1} \in S_3 \end{cases}$$

- Each $\alpha_i = g^{a_i} y^{b_i}$, where $a_i, b_i \in \mathbb{Z}_q$ are known!

Pollard-ρ (1/2)

Partition G into S_1, S_2, and S_3 "randomly".

- Generate "random" sequence $\alpha_0, \alpha_1, \alpha_2 \ldots$

$$\alpha_0 = g$$
$$\alpha_i = \begin{cases} \alpha_{i-1} g & \text{if } \alpha_{i-1} \in S_1 \\ \alpha_{i-1}^2 & \text{if } \alpha_{i-1} \in S_2 \\ \alpha_{i-1} y & \text{if } \alpha_{i-1} \in S_3 \end{cases}$$

- Each $\alpha_i = g^{a_i} y^{b_i}$, where $a_i, b_i \in \mathbb{Z}_q$ are known!
- If $\alpha_i = \alpha_j$ and $(a_i, b_i) \neq (a_j, b_j)$ then $y = g^{(a_i - a_j)(b_j - b_i)^{-1}}$.

Pollard-ρ (2/2)

- If $\alpha_i = \alpha_j$, then $\alpha_{i+1} = \alpha_{j+1}$.
- The sequence $\alpha_0, \alpha_1, \alpha_2, \ldots$ is "essentially random".
- The Birthday bound implies that the expected running time is $O(\sqrt{q})$.

Index Calculus

- Let $\mathcal{B} = \{p_1, \ldots, p_B\}$ be a set of small prime **integers**.

Index Calculus

- Let $\mathcal{B} = \{p_1, \ldots, p_B\}$ be a set of small prime **integers**.
- Compute $a_i = \log_g p_i$ for all $p_i \in \mathcal{B}$.

Index Calculus

- Let $\mathcal{B} = \{p_1, \ldots, p_B\}$ be a set of small prime **integers**.
- Compute $a_i = \log_g p_i$ for all $p_i \in \mathcal{B}$.
  1. Choose $s_j \in \mathbb{Z}_q$ randomly and attempt to factor $g^{s_j} = \prod_i p_i^{e_{j,i}}$ as an **integer**.

Index Calculus

- Let $\mathcal{B} = \{p_1, \ldots, p_B\}$ be a set of small prime **integers**.
- Compute $a_i = \log_g p_i$ for all $p_i \in \mathcal{B}$.
  1. Choose $s_j \in \mathbb{Z}_q$ randomly and attempt to factor $g^{s_j} = \prod_i p_i^{e_{j,i}}$ as an **integer**.
  2. If g^{s_j} factored in $\mathcal{B}$ and $e_j = (e_{j,1}, \ldots, e_{j,B})$ is linearly independent of $e_1, \ldots, e_{j-1}$, then $j \leftarrow j + 1$.

Index Calculus

- Let $\mathcal{B} = \{p_1, \ldots, p_B\}$ be a set of small prime **integers**.
- Compute $a_i = \log_g p_i$ for all $p_i \in \mathcal{B}$.
  1. Choose $s_j \in \mathbb{Z}_q$ randomly and attempt to factor $g^{s_j} = \prod_i p_i^{e_{j,i}}$ as an **integer**.
  2. If g^{s_j} factored in $\mathcal{B}$ and $e_j = (e_{j,1}, \ldots, e_{j,B})$ is linearly independent of $e_1, \ldots, e_{j-1}$, then $j \leftarrow j + 1$.
  3. If $j < B$, then go to (1)

Index Calculus

- Let $\mathcal{B} = \{p_1, \ldots, p_B\}$ be a set of small prime **integers**.
- Compute $a_i = \log_g p_i$ for all $p_i \in \mathcal{B}$.

Index Calculus

- Let $\mathcal{B} = \{p_1, \ldots, p_B\}$ be a set of small prime **integers**.
- Compute $a_i = \log_g p_i$ for all $p_i \in \mathcal{B}$.
- Repeat:
  1. Choose $s \in \mathbb{Z}_q$ randomly.
  2. Attempt to factor $yg^s = \prod_i p_i^{e_i}$ as an **integer**.
  3. If a factorization is found, then output $\left(\sum_i a_i e_i - s\right) \bmod q$.

Example Groups

- $\mathbb{Z}_n$ additively?

Example Groups

- $\mathbb{Z}_n$ additively? ($(\log_g y)g = y \bmod n$, so $\log_g y = yg^{-1} \bmod n$) **Bad for crypto!**
- Large prime order subgroup of $\mathbb{Z}_p^*$ with p prime. In particular $p = 2q + 1$ with q prime.
- Large prime order subgroup of $\mathrm{GF}_{p^k}^*$.
- "Carefully chosen" elliptic curve group.