Integer Factorization Problem in Cryptography

Giordano Santilli

Università degli Studi di Trento

28 May 2021

CrypTO Conference 2021

Outline

1. The problem of Factorization
2. Public Key Encryption schemes based on IFP
3. Factorization Algorithms
4. A pattern in successive remainders

The problem of Factorization

Integer Factorization Problem (IFP)

Theorem (Fundamental Theorem of Arithmetic)

Every positive integer N greater than 1 can be represented in a unique way as a product of prime powers:

$$N = p_1^{e_1} \cdots p_k^{e_k},$$

where $k \in \mathbb{N}^+$, $p_1, \ldots, p_k$ prime numbers and $e_1, \ldots, e_k \in \mathbb{N}$.

Integer Factorization Problem (IFP)

Theorem (Fundamental Theorem of Arithmetic)

Every positive integer N greater than 1 can be represented in a unique way as a product of prime powers:

$$N = p_1^{e_1} \cdots p_k^{e_k},$$

where $k \in \mathbb{N}^+$, $p_1, \ldots, p_k$ prime numbers and $e_1, \ldots, e_k \in \mathbb{N}$.

One-way problem:

$$p_1^{e_1} \cdots p_k^{e_k} \xrightarrow{\text{easy}} N$$

$$N \xrightarrow{\text{hard}} p_1^{e_1} \cdots p_k^{e_k}$$

Integer Factorization Problem (IFP)

Integer Factorization Problem (IFP)

Given a semiprime $N \in \mathbb{Z}$, find its prime factors p and q.

Remark

We call p the smaller factor and q the bigger one.

Public Key Encryption schemes based on IFP

PKE based on IFP

- RSA (1976)
- Rabin Cryptosystem (1979)
- Goldwasser-Micali Cryptosystem (1982)
- Paillier Cryptosystem (1999)

RSA

Generation of the key

1. Generate two random prime numbers p and q and compute $N = pq$;
2. Generate a random invertible $e \in \mathbb{Z}_{\varphi(N)}$ and compute d such that $ed \equiv 1 \mod \varphi(N)$;
3. (N, e) is the public key, while (p, q, d) is the private key.

Encryption

1. Consider a message $m \in \mathbb{Z}_N$;
2. Compute and transmit $c \equiv m^e \mod N$.

Decryption

1. Compute $c^d \equiv m^{ed} \equiv m \mod N$.

Security of RSA

1. Given (N, e) and c is infeasible to recover m as $\sqrt[e]{c} \bmod N$.

RSA

Security of RSA

1. Given (N, e) and c is infeasible to recover m as $\sqrt[e]{c} \bmod N$.
2. Given (N, e) is infeasible to recover d.

RSA

Security of RSA

1. Given (N, e) and c is infeasible to recover m as $\sqrt[e]{c} \bmod N$.
2. Given (N, e) is infeasible to recover d.
3. Given N is infeasible to recover $\varphi(N)$.

RSA

Security of RSA

1. Given (N, e) and c is infeasible to recover m as $\sqrt[e]{c} \bmod N$.
2. Given (N, e) is infeasible to recover d.
3. Given N is infeasible to recover $\varphi(N)$.
4. Given N is infeasible to recover p and q.

Security of RSA

1. Given (N, e) and c is infeasible to recover m as $\sqrt[e]{c} \bmod N$.
2. Given (N, e) is infeasible to recover d.
3. Given N is infeasible to recover $\varphi(N)$.
4. Given N is infeasible to recover p and q.

$$3 \iff 4$$

$$4 \implies 1$$

$$4 \implies 2$$

$$2 \overset{\text{GRH}}{\implies} 4$$

$$1 \overset{?}{\implies} 4$$

Rabin Cryptosystem

Generation of the key

1. Generate two random prime numbers p and q such that $p \equiv q \equiv 3 \bmod 4$ and compute $N = pq$;
2. N is the public key, while (p, q) is the private key.

Encryption

1. Consider a message $m \in \mathbb{Z}_N$;
2. Compute and transmit $c \equiv m^2 \bmod N$.

Decryption

1. Solve the system
$$\begin{cases} m \equiv \pm\sqrt{c} \equiv \pm c^{\frac{p+1}{4}} \mod p \\ m \equiv \pm\sqrt{c} \equiv \pm c^{\frac{q+1}{4}} \mod q; \end{cases}$$
2. The original message m is one of the four solutions found.

Rabin Cryptosystem

Generation of the key

1. Generate two random prime numbers p and q such that $p \equiv q \equiv 3 \bmod 4$ and compute $N = pq$;
2. N is the public key, while (p, q) is the private key.

Encryption

1. Consider a message $m \in \mathbb{Z}_N$;
2. Compute and transmit $c \equiv m^2 \bmod N$.

Decryption

1. Solve the system

$$\begin{cases} m \equiv \pm\sqrt{c} \equiv \pm c^{\frac{p+1}{4}} \mod p \\ m \equiv \pm\sqrt{c} \equiv \pm c^{\frac{q+1}{4}} \mod q; \end{cases}$$

2. The original message m is one of the four solutions found.

Rabin Cryptosystem

Security of Rabin cryptosystem

Recovering the plaintext m from the ciphertext c in the Rabin cryptosystem is as hard as finding a factorization for N.

Goldwasser-Micali Cryptosystem

Generation of the key

1. Generate two random prime numbers p and q and compute $N = pq$;
2. Generate $x \in \mathbb{Z}_N$ such that $\left(\frac{x}{p}\right) = \left(\frac{x}{q}\right) = -1$;
3. (N, x) is the public key, while (p, q) is the private key.

Encryption

1. Consider a message $\mathbf{m} = (m_1, \dots, m_k) \in (\mathbb{Z}_2)^k$;
2. Generate random $y_i \in \mathbb{Z}_N^*$ for $1 \leq i \leq k$;
3. Compute $c_i \equiv y_i^2 x^{m_i} \bmod N$ and transmit $\mathbf{c} = (c_1, \dots, c_k) \in (\mathbb{Z}_N)^k$.

Decryption

1. If c_i is a quadratic residue then $m_i = 0$, otherwise $m_i = 1$.

Goldwasser-Micali Cryptosystem

Generation of the key

1. Generate two random prime numbers p and q and compute $N = pq$;
2. Generate $x \in \mathbb{Z}_N$ such that $\left(\frac{x}{p}\right) = \left(\frac{x}{q}\right) = -1$;
3. (N, x) is the public key, while (p, q) is the private key.

Encryption

1. Consider a message $\mathbf{m} = (m_1, \dots, m_k) \in (\mathbb{Z}_2)^k$;
2. Generate $\boxed{\text{random}}$ $y_i \in \mathbb{Z}_N^*$ for $1 \leq i \leq k$;
3. Compute $c_i \equiv y_i^2 x^{m_i} \mod N$ and transmit $\mathbf{c} = (c_1, \dots, c_k) \in (\mathbb{Z}_N)^k$.

Decryption

1. If c_i is a quadratic residue then $m_i = 0$, otherwise $m_i = 1$.

Security of Goldwasser-Micali Cryptosystem

This algorithm is based on the quadratic residuosity problem (QRP): given (N, x) is computationally infeasible to decide whether x is a quadratic residue or not.

Goldwasser-Micali Cryptosystem

Security of Goldwasser-Micali Cryptosystem

This algorithm is based on the quadratic residuosity problem (QRP): given (N, x) is computationally infeasible to decide whether x is a quadratic residue or not.

$$\text{IFP} \Longrightarrow \text{QRP}$$

$$\text{QRP} \stackrel{?}{\Longrightarrow} \text{IFP}$$

Paillier Cryptosystem

Generation of the key

1. Generate two random prime numbers p and q and compute $N = pq$ and $\lambda = \mathrm{lcm}(p-1, q-1)$;
2. Choose a random $g \in \mathbb{Z}^*_{N^2}$ and compute
$$\mu \equiv \left(\frac{\left(g^\lambda \bmod N^2\right) - 1}{N} \right)^{-1} \bmod N;$$
3. (N, g) is the public key, while (p, q, λ, μ) is the private key.

Encryption

1. Consider a message $m \in \mathbb{Z}_N$;
2. Generate a random $r \in \mathbb{Z}^*_N$ and compute $c \equiv g^m \cdot r^N \bmod N^2$.

Decryption

1. Compute $m \equiv \left(\frac{\left(c^\lambda \bmod N^2\right) - 1}{N} \right) \cdot \mu \bmod N$.

Paillier Cryptosystem

Homomorphic Properties

Paillier encryption is homomorphic:

$$\mathsf{Decrypt}(\mathsf{Encrypt}(m_1) \cdot \mathsf{Encrypt}(m_2)) \equiv m_1 + m_2 \bmod N.$$

Paillier Cryptosystem

Homomorphic Properties

Paillier encryption is homomorphic:

$$\mathsf{Decrypt}(\mathsf{Encrypt}(m_1) \cdot \mathsf{Encrypt}(m_2)) \equiv m_1 + m_2 \bmod N.$$

Security of Paillier Cryptosystem

Paillier Cryptosystem is based on the composite residuosity problem (CRP): given (N, x), it is computationally infeasible to decide whether there exists $y \in \mathbb{Z}_{N^2}$ such that $x \equiv y^N \bmod N^2$.

Paillier Cryptosystem

Homomorphic Properties

Paillier encryption is homomorphic:

$$\mathsf{Decrypt}(\mathsf{Encrypt}(m_1) \cdot \mathsf{Encrypt}(m_2)) \equiv m_1 + m_2 \bmod N.$$

Security of Paillier Cryptosystem

Paillier Cryptosystem is based on the composite residuosity problem (CRP): given (N, x), it is computationally infeasible to decide whether there exists $y \in \mathbb{Z}_{N^2}$ such that $x \equiv y^N \bmod N^2$.

$$\text{IFP} \implies \text{CRP}$$

$$\text{RSA} \implies \text{CRP}$$

$$\text{CRP} \overset{?}{\implies} \text{IFP}$$

Factorization Algorithms

A naive algorithm

Suppose we want to recover p and q from N.

Brute Force Algorithm

1. For any prime $s \in \mathbb{P}$ starting from 2 check if $N \equiv 0 \mod s$;
2. Stop when p is found, then $q = \frac{N}{p}$.

A naive algorithm

Suppose we want to recover p and q from N.

Brute Force Algorithm

1. For any prime $s \in \mathbb{P}$ starting from 2 check if $N \equiv 0 \bmod s$;
2. Stop when p is found, then $q = \frac{N}{p}$.

Since $p < q$ then $p \leq \lfloor \sqrt{N} \rfloor$, meaning that we have to check, in the worst case, $\pi(\sqrt{N}) \sim \frac{\sqrt{N}}{\log \sqrt{N}} \sim O\left(\sqrt{N}\right)$ values.

A naive algorithm

Suppose we want to recover p and q from N.

Brute Force Algorithm

1. For any prime $s \in \mathbb{P}$ starting from 2 check if $N \equiv 0 \bmod s$;
2. Stop when p is found, then $q = \frac{N}{p}$.

Since $p < q$ then $p \leq \lfloor \sqrt{N} \rfloor$, meaning that we have to check, in the worst case, $\pi(\sqrt{N}) \sim \frac{\sqrt{N}}{\log \sqrt{N}} \sim O\left(\sqrt{N}\right)$ values.

Effectiveness

This method is called Trial Division. It works best when p is small.

Factorization Methods

First-Category Algorithms

- These methods return the smaller prime divisor p of N.
- They are effective if $p \approx 7 - 40$ digits.

Factorization Methods

First Category Algorithms	
Factorization Method	Execution Time
Trial Division	$O\left(N^{\frac{1}{2}}\right)$
Pollard's $p-1$ Algorithm	$O\left(N^{\frac{1}{2}}\right)$
Pollard's ρ	$O\left(N^{\frac{1}{4}}\right)$
Shanks' Class Group Method	$O\left(N^{\frac{1}{4}}\right)$
Lenstra's Elliptic Curves Method (ECM)	$O\left(e^{\sqrt{2\log N \log\log N}}\right)$

Table: Recap of some famous first category factorization methods for $N = p \cdot q$.

Fermat's method

Fermat's approach

IFP can be solved finding $x, y \in \mathbb{Z}_N$ such that

$$x^2 \equiv y^2 \bmod N,$$

meaning that

$$N = pq | (x^2 - y^2) = (x-y)(x+y) \Longrightarrow p | (x-y)(x+y) \text{ and } q | (x+y)(x-y).$$

But since p and q are primes:

$$\begin{cases} p | (x-y) \vee p | (x+y) \\ q | (x-y) \vee q | (x+y) \end{cases}$$

Fermat's method

The possible cases are the following:

$p \mid (x-y)$	$p \mid (x+y)$	$q \mid (x-y)$	$q \mid (x+y)$	$\gcd(x-y, N)$	$\gcd(x+y, N)$	Factorization
✓	✓	✓	✓	N	N	✗
✓	✓	✓	✗	N	p	✓
✓	✓	✗	✓	p	N	✓
✓	✗	✓	✓	N	q	✓
✓	✗	✓	✗	N	1	✗
✓	✗	✗	✓	p	q	✓
✗	✓	✓	✗	q	p	✓
✗	✓	✗	✓	1	N	✗
✗	✓	✓	✓	q	N	✓

Table: Output for $x^2 \equiv y^2 \bmod N$.

It is possible to recover a successful factorization in 6 cases over $9 \approx 66\%$.

Fermat's method

The possible cases are the following:

$p \mid (x-y)$	$p \mid (x+y)$	$q \mid (x-y)$	$q \mid (x+y)$	$\gcd(x-y,N)$	$\gcd(x+y,N)$	Factorization
✓	✓	✓	✓	N	N	✗
✓	✓	✓	✗	N	p	✓
✓	✓	✗	✓	p	N	✓
✓	✗	✓	✓	N	q	✓
✓	✗	✓	✗	N	1	✗
✓	✗	✗	✓	p	q	✓
✗	✓	✓	✗	q	p	✓
✗	✓	✗	✓	1	N	✗
✗	✓	✓	✓	q	N	✓

Table: Output for $x^2 \equiv y^2 \mod N$.

It is possible to recover a successful factorization in 6 cases over $9 \approx 66\%$. Adding the condition $x \not\equiv \pm y \mod N$ it is **always** possible to recover a non-trivial factor of N.

Factorization methods

Second-Category Algorithms

- Do not take into account the distance between p and q and the complexity only depends on the size of N.
- Are effective if N has more than ≈ 100 digits and no small factors.
- They are based on Fermat's idea.

Factorization methods

Second Category Algorithms	
Factorization Method	Execution Time
Lehman's method	$O\left(N^{\frac{1}{3}}\right)$
Shanks' Square Forms Factorization (SQUFOF)	$O\left(N^{\frac{1}{4}}\right)$
Dixon's Factorization Method	$O\left(e^{2\sqrt{2\log N \log\log N}}\right)$
Continued Fractions Method (CFRAC)	$O\left(e^{\sqrt{2\log N \log\log N}}\right)$
Multiple Polynomial Quadratic Sieve (MPQS)	$O\left(e^{\sqrt{\log N \log\log N}}\right)$
General Number Field Sieve (GNFS)	$O\left(e^{\sqrt[3]{\frac{64}{9}\log N(\log\log N)^2}}\right)$

Table: Recap of some second category factorization methods for $N = p \cdot q$.

RSA Factoring Challenge (1991)

RSA-Number	Binary Digits	Date of Factorization	Method used
RSA-100	330	1 April 1991	MPQS
RSA-110	364	14 April 1992	MPQS
RSA-120	397	9 July 1993	MPQS
RSA-129	426	26 April 1994	MPQS
RSA-130	430	10 April 1996	GNFS
RSA-140	463	2 February 1999	GNFS
RSA-150	496	16 April 2004	GNFS
RSA-155	512	22 August 1999	GNFS
RSA-160	530	1 April 2003	GNFS
RSA-170	563	29 December 2009	GNFS
RSA-576	576	3 December 2003	GNFS
RSA-180	596	8 May 2010	GNFS
RSA-190	629	8 November 2010	GNFS
RSA-640	640	2 November 2005	GNFS
RSA-200	663	9 May 2005	GNFS
RSA-210	696	26 September 2013	GNFS
RSA-704	704	2 July 2012	GNFS
RSA-220	729	13 May 2016	GNFS
RSA-230	762	15 August 2018	GNFS
RSA-232	768	17 February 2020	GNFS
RSA-768	768	12 December 2009	GNFS
RSA-240	795	2 December 2019	GNFS
RSA-250	829	28 February 2020	GNFS

Table: Known factorizations of RSA moduli.

A pattern in successive remainders

Let m be $\left\lfloor\sqrt{\frac{N}{2}}\right\rfloor \leq m \leq \left\lfloor\sqrt{N}\right\rfloor$ and let

$$\begin{cases} N \equiv a_0 \bmod m \\ N \equiv a_1 \bmod (m+1) \\ N \equiv a_2 \bmod (m+2), \end{cases}$$

where a_0, a_1, a_2 are $a_0 \leq a_1 \leq a_2$ or $a_0 \geq a_1 \geq a_2$.
We define $k := a_1 - a_0$ and

$$w := \begin{cases} a_2 - 2a_1 + a_0 & \text{if } a_2 - 2a_1 + a_0 \geq 0, \\ a_2 - 2a_1 + a_0 + m + 2 & \text{if } a_2 - 2a_1 + a_0 < 0. \end{cases}$$

Successive moduli

Proposition

Let N *be such that* $N \geq 50$ *and let* $m \in \mathbb{N}^+$ *with* $\left\lfloor \sqrt{\frac{N}{2}} \right\rfloor \leq m \leq \left\lfloor \sqrt{N} \right\rfloor$*, then*

$$w = \begin{cases} 2, \\ 4, \\ 6. \end{cases}$$

Corollary

If there exists a value for m *such that* $\left\lfloor \sqrt{\frac{N}{2}} \right\rfloor + 1 \leq m \leq \left\lfloor \sqrt{N} \right\rfloor - 1$*, then* $w = 4$.

Successive moduli

Example

$N = 925363$ and $m = 680$:

$$
\begin{aligned}
N &\equiv a_0 = 563 && \mod m \\
N &\equiv a_1 = 565 && \mod (m+1) \\
N &\equiv a_2 = 571 && \mod (m+2) \\
N &\equiv 581 && \mod (m+3) \\
N &\equiv 595 && \mod (m+4) \\
N &\equiv 613 && \mod (m+5) \\
N &\equiv 635 && \mod (m+6) \\
N &\equiv 661 && \mod (m+7) \\
N &\equiv 3 && \mod (m+8)
\end{aligned}
$$

Successive moduli

Example

$N = 925363$ and $m = 680$:

$$
\begin{aligned}
N &\equiv a_0 = 563 && \mod m \\
N &\equiv a_1 = 565 = a_0 + k = 563 + 2 && \mod (m+1) \\
N &\equiv a_2 = 571 = a_1 + k + w = 565 + 2 + 4 && \mod (m+2) \\
N &\equiv 581 = 571 + 2 + 2 \cdot 4 && \mod (m+3) \\
N &\equiv 595 = 581 + 2 + 3 \cdot 4 && \mod (m+4) \\
N &\equiv 613 = 595 + 2 + 4 \cdot 4 && \mod (m+5) \\
N &\equiv 635 = 613 + 2 + 5 \cdot 4 && \mod (m+6) \\
N &\equiv 661 = 635 + 2 + 6 \cdot 4 && \mod (m+7) \\
N &\equiv 3 = 661 + 2 + 7 \cdot 4 = 691 && \mod (m+8)
\end{aligned}
$$

A formula for successive moduli

Proposition

Let $N \geq 50$ *and such that* $\left\lfloor \sqrt{\frac{N}{2}} \right\rfloor \leq m \leq \left\lfloor \sqrt{N} \right\rfloor$, *then for every* $i \in \mathbb{N}$,

$$N \equiv \left(a_0 + ik + w \cdot \frac{i(i-1)}{2} \right) \bmod (m+i).$$

Corollary

If $\left\lfloor \sqrt{\frac{N}{2}} \right\rfloor + 1 \leq m \leq \left\lfloor \sqrt{N} \right\rfloor - 1$, *then for every* $i \in \mathbb{N}$,

$$N \equiv \left(a_0 + ik + 2i^2 - 2i \right) \bmod (m+i).$$

Interpolating polynomial

Consider the polynomial $f \in \mathbb{Q}[x]$ of degree 2, such that

$$\begin{cases} f(0) = a_0, \\ f(1) = a_1, \\ f(2) = a_2. \end{cases}$$

Proposition

Let $\left\lfloor \sqrt{\frac{N}{2}} \right\rfloor + 1 \leq m \leq \left\lfloor \sqrt{N} \right\rfloor - 1$. *Then, the interpolating polynomial* $f \in \mathbb{Q}(x)$ *is such that, for every* $i \in \mathbb{Z}$,

$$N \equiv f(i) \bmod (m+i).$$

Successive moduli in factorization

In order to find a factor of N, we would like to solve the following equation for some $x \in \mathbb{Z}$:

$$a_0 + ik + 2i^2 - 2i = x(m + i).$$

Successive moduli in factorization

In order to find a factor of N, we would like to solve the following equation for some $x \in \mathbb{Z}$:

$$a_0 + ik + 2i^2 - 2i = x(m+i).$$

Proposition

Let N be a semiprime and m such that $\left\lfloor \sqrt{\frac{N}{2}} \right\rfloor + 1 \leq m \leq \left\lfloor \sqrt{N} \right\rfloor - 1.$ *Then producing the factorization of N is equivalent to finding an integer $i \in \mathbb{N}^+$ for which*

$$N \equiv \left(a_0 + ik + 2i^2 - 2i\right) \equiv 0 \bmod (m+i).$$

Successive moduli in factorization

If we consider the interpolating polynomial f, then if m is close to one of the factor of N, then the roots of f are exactly the $i \in \mathbb{Z}$ such that

$$f(i) \equiv 0 \bmod (m+i).$$

However to achieve this result, we need to choose the first remainder a_0 in the monotonic descending sequence that leads to 0.

Successive moduli in factorization

Example

$N = 925363$ and $m = 943$, then

$$\begin{cases} N \equiv 280 \bmod 943, \\ N \equiv 243 \bmod 944, \\ N \equiv 208 \bmod 945. \end{cases}$$

The interpolating polynomial is

$$f(i) = i^2 - 38i + 280,$$

which has two roots: $i_1 = 10$ and $i_2 = 28$. Therefore the two factors of N are:

$$m + i_1 = 953 \qquad m + i_2 = 971.$$

THANK YOU
FOR THE ATTENTION!

giordano.santilli@unitn.it