Introduction to Cryptology

10.3 - Number Theory Review

Federico Pintore

Mathematical Institute, University of Oxford (UK)

Michaelmas term 2020

Integers modulo N

Euclidean division: given two integers a, b, with $b \neq 0$, there exist unique $q, r \in \mathbb{Z}$ such that $a = bq + r$, with $0 \leq r < |b|$.

Integers modulo N

Euclidean division: given two integers a, b, with $b \neq 0$, there exist unique $q, r \in \mathbb{Z}$ such that $a = bq + r$, with $0 \leq r < |b|$.

Given a positive integer N, and $a, b \in \mathbb{Z}$:

- $a \pmod N$ denotes the reminder of a when divided by N;
- $[a]_N$ is the set of all integers having the same reminder of a when divided by N;
- we write $a = b \pmod N$ if $[a]_N = [b]_N$.

Integers modulo N

Euclidean division: given two integers a, b, with $b \neq 0$, there exist unique $q, r \in \mathbb{Z}$ such that $a = bq + r$, with $0 \leq r < |b|$.

Given a positive integer N, and $a, b \in \mathbb{Z}$:

- $a \pmod N$ denotes the reminder of a when divided by N;
- $[a]_N$ is the set of all integers having the same reminder of a when divided by N;
- we write $a = b \pmod N$ if $[a]_N = [b]_N$.

$\mathbb{Z}_N = \{[i]_N \mid i = 0, 1, \ldots, N-1\}$ is the set of integers modulo N.

Two binary operations can be defined on $\mathbb{Z}_N$:

$$[a]_N + [b]_N := [a+b]_N, \quad [a]_N \cdot [b]_N := [ab]_N.$$

Integers modulo N

- $(\mathbb{Z}_N, +, \cdot)$ is an abelian ring ($[0]_N$ is the zero element, $[1]_N$ is the identity).
- $[a]_N$ is invertible if there exists $[b]_N \in \mathbb{Z}_N$ such that $[a]_N \cdot [b]_N = [1]_N$.

Integers modulo N

- $(\mathbb{Z}_N, +, \cdot)$ is an abelian ring ($[0]_N$ is the zero element, $[1]_N$ is the identity).
- $[a]_N$ is invertible if there exists $[b]_N \in \mathbb{Z}_N$ such that $[a]_N \cdot [b]_N = [1]_N$.

Which are the invertible elements in $\mathbb{Z}_N \setminus \{[0]_N\}$?

- Given $a, b \in \mathbb{Z}$, a is divided by b if $a = bc$ for some $c \in \mathbb{Z}$.
- The greatest common divisor $\gcd(a, b)$ of $a, b \in \mathbb{Z}$ is the biggest integer dividing both a and b.
- For $a, b \in \mathbb{Z}$, $\gcd(a, b)$ is the smallest positive integer of the form $aX + bY$, with $X, Y \in \mathbb{Z}$.

Integers modulo N

Proposition

Given two integers $b \geq 1$ and $N > 1$, $[b]_N$ is invertible if and only if $\gcd(b, N) = 1$ (i.e. b and N are relatively prime).

Integers modulo N

Proposition

Given two integers $b \geq 1$ and $N > 1$, $[b]_N$ is invertible if and only if $\gcd(b, N) = 1$ (i.e. b and N are relatively prime).

- The set $\mathbb{Z}_N^* = \{[b]_N \in \mathbb{Z}_N \mid \gcd(b, N) = 1\}$ contains all the invertible elements of $\mathbb{Z}_N \setminus \{[0]_N\}$.
- $(\mathbb{Z}_N^*, \cdot)$ is a group.

Integers modulo N

Proposition

Given two integers $b \geq 1$ and $N > 1$, $[b]_N$ is invertible if and only if $\gcd(b, N) = 1$ (i.e. b and N are relatively prime).

- The set $\mathbb{Z}_N^* = \{[b]_N \in \mathbb{Z}_N \mid \gcd(b, N) = 1\}$ contains all the invertible elements of $\mathbb{Z}_N \setminus \{[0]_N\}$.
- $(\mathbb{Z}_N^*, \cdot)$ is a group.
- Define $\phi(N)$ as the cardinality of $\mathbb{Z}_N^*$ ($\phi : \mathbb{N} \to \mathbb{N}$ is called the Euler phi function).
- If N is a prime, then $\phi(N) = N - 1$. If $N = pq$ is a semi-prime (product of two primes), $\phi(N) = (p - 1)(q - 1)$.

Integers modulo N

Proposition

If $(\mathbb{G}, \cdot)$ is a finite abelian group of cardinality m, then $g^m = 1$ for every $g \in \mathbb{G}$.

Integers modulo N

Proposition

If $(\mathbb{G}, \cdot)$ is a finite abelian group of cardinality m, then $g^m = 1$ for every $g \in \mathbb{G}$.

- For $[a]_N \in \mathbb{Z}_N^*$, we have $([a]_N)^{\phi(N)} = [1]_N$.
- Let $e \in \mathbb{Z}$ be relatively prime with N. Then the map:

$$f_e([x]_N) = ([x]_N)^e$$

is a permutation of $\mathbb{Z}_N^*$. Indeed, its inverse is the map f_d, where d is such that $[d]_{\phi(N)}[e]_{\phi(N)} = [1]_{\phi(N)}$
($de = \ell\phi(N) + 1$, $([x]_N)^{\ell\phi(N)} = [1]_N$ and $([x]_N)^{\ell\phi(N)+1} = [x]_N$)

The factoring problem

Let GenModulus be a PPT algorithm that, on input n, returns (N, p, q), where $N = pq$ and p, q are n-bit primes.

The factoring problem

Let GenModulus be a PPT algorithm that, on input n, returns (N, p, q), where $N = pq$ and p, q are n-bit primes.

In the experiment $\text{Factor}_{\mathcal{A},\text{GenModulus}}(n)$, the adversary $\mathcal{A}$ is given the composite number N output by GenModulus on input n, and has to determine the divisors p, q.

Factoring is hard relative to GenModulus if, for every $\mathcal{A}$, their success probability in the above experiment is negligible in n.

The factoring problem

Let GenModulus be a PPT algorithm that, on input n, returns (N, p, q), where $N = pq$ and p, q are n-bit primes.

In the experiment $\text{Factor}_{\mathcal{A},\text{GenModulus}}(n)$, the adversary $\mathcal{A}$ is given the composite number N output by GenModulus on input n, and has to determine the divisors p, q.

Factoring is hard relative to GenModulus if, for every $\mathcal{A}$, their success probability in the above experiment is negligible in n.

Factoring assumption: there exists a GenModulus relative to which factoring is hard.

The RSA problem

Let GenRSA be a PPT algorithm that, on input n, outputs (N, p, q, e, d), where p and q are n-bit primes, $N = pq$, and $[e]_{\varphi(N)}[d]_{\varphi(N)} = [1]_{\varphi(N)}$.

The RSA problem

Let GenRSA be a PPT algorithm that, on input n, outputs (N, p, q, e, d), where p and q are n-bit primes, $N = pq$, and $[e]_{\varphi(N)}[d]_{\varphi(N)} = [1]_{\varphi(N)}$.

In the experiment $\mathrm{RSA}-\mathrm{inv}_{\mathcal{A},\mathrm{GenRSA}}(n)$:

- GenRSA is run on input n;
- the adversary $\mathcal{A}$ is given N, e and a uniform element $[y]_N \in \mathbb{Z}_N^*$;
- $\mathcal{A}$ has to determine $[x]_N \in \mathbb{Z}_N^*$ such that $([x]_N)^e = [y]_N$.

The RSA problem is hard relative to GenRSA if, for every $\mathcal{A}$, their success probability in the above experiment is negligible in n.

The RSA problem

Let GenRSA be a PPT algorithm that, on input n, outputs (N, p, q, e, d), where p and q are n-bit primes, $N = pq$, and $[e]_{\varphi(N)}[d]_{\varphi(N)} = [1]_{\varphi(N)}$.

In the experiment $\mathrm{RSA} - \mathrm{inv}_{\mathcal{A},\mathrm{GenRSA}}(n)$:

- GenRSA is run on input n;
- the adversary $\mathcal{A}$ is given N, e and a uniform element $[y]_N \in \mathbb{Z}_N^*$;
- $\mathcal{A}$ has to determine $[x]_N \in \mathbb{Z}_N^*$ such that $([x]_N)^e = [y]_N$.

The RSA problem is hard relative to GenRSA if, for every $\mathcal{A}$, their success probability in the above experiment is negligible in n.

RSA assumption: there exists a GenRSA relative to which the RSA problem is hard.

Relationship between the two Assumptions

If the factorisation of N is known, it is possible to compute $\phi(N)$ and hence $[d]_{\phi(N)} = ([e]_{\phi(N)})^{-1}$.

The other implication is still open! The best we can say is:

Theorem

Given a composite integer N and integers e, d such that $[e]_{\phi(N)}[d]_{\phi(N)} = [1]_{\phi(N)}$, there is a PPT algorithm that can output a factor of N except with negligible probability (in $\|N\|$).

Prime numbers

If a positive integer a divides $b \in \mathbb{Z}$, we call a a divisor of b. If $a \notin \{1, b\}$, a is said a non trivial divisor of b.

A positive integer p is prime if it has only trivial divisors.

Prime numbers

If a positive integer a divides $b \in \mathbb{Z}$, we call a a divisor of b. If $a \notin \{1, b\}$, a is said a non trivial divisor of b.

A positive integer p is prime if it has only trivial divisors.

- There are infinitely many primes.
- Fundamental Theorem of Arithmetic: any $n \in \mathbb{Z}$ can be decomposed uniquely as a product of prime numbers.
- Bertrand's postulate: for any $n \in \mathbb{N} \setminus \{0\}$, the fraction of n-bit integers that are prime is at least $1/3n$.

Prime numbers

If a positive integer a divides $b \in \mathbb{Z}$, we call a a divisor of b. If $a \notin \{1, b\}$, a is said a non trivial divisor of b.

A positive integer p is prime if it has only trivial divisors.

- There are infinitely many primes.
- Fundamental Theorem of Arithmetic: any $n \in \mathbb{Z}$ can be decomposed uniquely as a product of prime numbers.
- Bertrand's postulate: for any $n \in \mathbb{N} \setminus \{0\}$, the fraction of n-bit integers that are prime is at least $1/3n$.

How to efficiently generate random n-bit primes?

Generating Random Primes

Naive approach: pick random n-bit integers and check if they are prime.

Input : length n, parameter t

for $i = 1, \ldots, t$ do

  $p' \leftarrow \{0,1\}^{n-1}$

  $p := 1||p'$

  if $Primality\_test(p) = 1$

    return p

return $\perp$

Generating Random Primes

Set $t = 3n^2$. Then the probability that the previous algorithm does not output a prime in t iterations is at most

$$\left(1 - \frac{1}{3n}\right)^t = \left(\left(1 - \frac{1}{3n}\right)^{3n}\right)^n \leq (e^{-1})^n = e^{-n}$$

since $(1 - 1/x)^x \leq e^{-1}$ for all $x \geq 1$.

This probability is negligible in n.

Generating Random Primes

Set $t = 3n^2$. Then the probability that the previous algorithm does not output a prime in t iterations is at most

$$\left(1 - \frac{1}{3n}\right)^t = \left(\left(1 - \frac{1}{3n}\right)^{3n}\right)^n \leq (e^{-1})^n = e^{-n}$$

since $(1 - 1/x)^x \leq e^{-1}$ for all $x \geq 1$.

This probability is negligible in n.

We still need to study algorithms that test primality.

Primality testing algorithms

On input a $n \in \mathbb{N}$, they decide whether n is prime or not.

There exist deterministic algorithms (see the AKS test, proposed in 2002).

In practice, probabilistic algorithms are used, since they are much faster.

Probabilistic algorithms have a small probability to return "prime" for composite numbers.

Fermat test

Fermat's little theorem: if n is prime, then $([a]_n)^{n-1} = [1]_n$ for all $[a]_n \in \mathbb{Z}_n^*$.

Fermat test

Fermat's little theorem: if n is prime, then $([a]_n)^{n-1} = [1]_n$ for all $[a]_n \in \mathbb{Z}_n^*$.

Idea: choose a uniform $a \in \{1, 2, \ldots, n-1\}$ and check whether $([a]_n)^{n-1} = [1]_n$. If not, then n is composite.

Any $a \in \{1, 2, \ldots, n-1\}$ s.t. $([a]_n)^{n-1} \neq [1]_n$ is a witness that n is composite

Fermat test

Input : integer n, parameter t

for $i = 1, \ldots, t$ do

  $a \leftarrow \{1, 2 \cdots, n-1\}$

  if $([a]_n)^{n-1} \neq [1]_n$

    return “composite”

return “prime”

Fermat test

Input : integer n, parameter t

for $i = 1, \ldots, t$ do

  $a \leftarrow \{1, 2 \cdots, n-1\}$

  if $([a]_n)^{n-1} \neq [1]_n$

    return "composite"

return "prime"

Theorem

If the set $\{witnesses\}_n$ of witnesses that n is composite is not empty, then

$$|\{witnesses\}_n| \geq |\mathbb{Z}_n^*|/2.$$

Fermat test

Input : integer n, parameter t

for $i = 1, \ldots, t$ do

  $a \leftarrow \{1, 2 \cdots, n-1\}$

  if $([a]_n)^{n-1} \neq [1]_n$

    return "composite"

return "prime"

Theorem

If the set $\{witnesses\}_n$ of witnesses that n is composite is not empty, then

$$|\{witnesses\}_n| \geq |\mathbb{Z}_n^*|/2.$$

Having a witness is not necessary for being composite.

Testing Primality

Carmichael numbers: composite numbers that do not have any witnesses.

Fermat test needs to be refined.

Testing Primality

Carmichael numbers: composite numbers that do not have any witnesses.

Fermat test needs to be refined.

- Let $n - 1 = 2^k u$, where u is odd and $k \geq 1$ (n is odd).

Testing Primality

Carmichael numbers: composite numbers that do not have any witnesses.

Fermat test needs to be refined.

- Let $n - 1 = 2^k u$, where u is odd and $k \geq 1$ (n is odd).
- Fermat test for n checks if $([a]_n)^{n-1} = ([a]_n)^{2^k u} = [1]_n$.

Testing Primality

Carmichael numbers: composite numbers that do not have any witnesses.

Fermat test needs to be refined.

- Let $n - 1 = 2^k u$, where u is odd and $k \geq 1$ (n is odd).
- Fermat test for n checks if $([a]_n)^{n-1} = ([a]_n)^{2^k u} = [1]_n$.
- What about $([a]_n)^u, ([a]_n)^{2u}, \cdots, ([a]_n)^{2^{k-1}u}$?

Testing Primality

Carmichael numbers: composite numbers that do not have any witnesses.

Fermat test needs to be refined.

- Let $n-1 = 2^k u$, where u is odd and $k \geq 1$ (n is odd).
- Fermat test for n checks if $([a]_n)^{n-1} = ([a]_n)^{2^k u} = [1]_n$.
- What about $([a]_n)^u, ([a]_n)^{2u}, \cdots, ([a]_n)^{2^{k-1}u}$?
- A strong witness that n is composite is an element $a \in \{1, 2 \ldots, n-1\}$ such that
  - $([a]_n)^u \neq \pm[1]_n$
  - $([a]_n)^{2^i u} \neq [-1]_n$ for all $i \in \{1, \cdots, k-1\}$

Testing Primality

Theorem

Let n be an odd positive integer that is not a prime power. Then we have that at least half of the elements of $\mathbb{Z}_n^*$ are strong witnesses that n is composite.

Testing whether n is a perfect power (power of an integer, not necessarily prime) can be done in polynomial time.

Miller-Rabin test

Input : integer $n > 2$, parameter t

if n is even

  return "composite"

if n is a perfect power

  return "composite"

determine u, k s.t. $n - 1 := 2^k u$, where u is odd and $k \geq 1$

for $j = 1, \ldots, t$ do

  $a \leftarrow \{1, \cdots, n-1\}$

  if $([a]_n)^u \neq \pm[1]_n$ and $([a]_n)^{2^i u} \neq -[1]_n$ for $i \in \{1, \cdots, k-1\}$

    return "composite"

return "prime"

Miller-Rabin test

Theorem

If n is prime, then the Miller-Rabin test always outputs "prime". If n is composite, the algorithm outputs "composite" except with probability at most 2^{-t}.

Quadratic Residues

Definition

For any positive integer m, we define the set of quadratic residues modulo m as

$$QR(m) := \{a \in \mathbb{Z}_m | \quad \exists b \in \mathbb{Z}_m \text{ such that } b^2 = a\}.$$

Quadratic Residues

Definition

For any positive integer m, we define the set of quadratic residues modulo m as

$$QR(m) := \{a \in \mathbb{Z}_m | \quad \exists b \in \mathbb{Z}_m \text{ such that } b^2 = a\}.$$

Theorem

Given a prime $p > 2$, for each $a \in QR(p) \cap \mathbb{Z}_p^*$ there exist two elements $b, b' \in \mathbb{Z}_p^*$ s.t. $b^2 = (b')^2 = a$.

Quadratic Residues

Definition

Given a prime $p > 2$ and an integer x s.t. $[x]_p \in \mathbb{Z}_p^*$, we define the *Legendre symbol of x modulo p* as follows:

$$\mathcal{L}_p(x) = \begin{cases} +1 & \text{if } [x]_p \in QR(p) \\ -1 & \text{if } [x]_p \notin QR(p). \end{cases}$$

Quadratic Residues

Definition

Given a prime $p > 2$ and an integer x s.t. $[x]_p \in \mathbb{Z}_p^*$, we define the *Legendre symbol of x modulo p* as follows:

$$\mathcal{L}_p(x) = \begin{cases} +1 & \text{if } [x]_p \in QR(p) \\ -1 & \text{if } [x]_p \notin QR(p). \end{cases}$$

Theorem

Given a prime $p > 2$ and an integer x s.t. $[x]_p \in \mathbb{Z}_p^*$, we have

$$[\mathcal{L}_p(x)]_p = ([x]_p)^{\frac{p-1}{2}}.$$

Quadratic Residues

Theorem

Let $N = pq$ - where p and q are distinct primes - and let y be an integer such that $[y]_N \in \mathbb{Z}_N^*$. Then $[y]_N \in QR(N)$ ***if and only if*** *$[y]_p \in QR(p)$ and $[y]_q \in QR(q)$.*

Quadratic Residues

Theorem

Let $N = pq$ - where p and q are distinct primes - and let y be an integer such that $[y]_N \in \mathbb{Z}_N^*$. Then $[y]_N \in QR(N)$ ***if and only if*** *$[y]_p \in QR(p)$ and $[y]_q \in QR(q)$.*

Theorem

Let $N = pq$, where p and q are two distinct odd primes. Given $x, \tilde{x}$ s.t. $[x]_N^2 = [\tilde{x}]_N^2$ but $[x]_N \neq \pm[\tilde{x}]_N$, it is possible to factor N in time polynomial in $\|N\|$.

Quadratic Residues

Theorem

Let $N = pq$ - where p and q are distinct primes - and let y be an integer such that $[y]_N \in \mathbb{Z}_N^*$. Then $[y]_N \in QR(N)$ ***if and only if*** *$[y]_p \in QR(p)$ and $[y]_q \in QR(q)$.*

Theorem

Let $N = pq$, where p and q are two distinct odd primes. Given $x, \tilde{x}$ s.t. $[x]_N^2 = [\tilde{x}]_N^2$ but $[x]_N \neq \pm[\tilde{x}]_N$, it is possible to factor N in time polynomial in $\|N\|$.

Theorem

Let $N = pq$, where p and q are two distinct odd primes such that $[p]_4 = [q]_4 = [3]_4$. Then every quadratic residue modulo N has exactly one square root that belongs to $QR(N)$.

Further Reading I

Mihir Bellare, Alexandra Boldyreva, and Silvio Micali.
Public-Key Encryption in a Multi-user Setting: Security Proofs and Improvements.
In Bart Preneel, editor, Advances in Cryptology — EUROCRYPT 2000, volume 1807 of Lecture Notes in Computer Science, pages 259–274. Springer Berlin Heidelberg, 2000.

Dan Boneh.
Simplified OAEP for the RSA and Rabin Functions.
In Joe Kilian, editor, Advances in Cryptology — CRYPTO 2001, volume 2139 of Lecture Notes in Computer Science, pages 275–291. Springer Berlin Heidelberg, 2001.

Further Reading II


Ronald Cramer and Victor Shoup.
Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack.
SIAM Journal on Computing, 33(1):167–226, 2003.


Whitfield Diffie and Martin E Hellman.
New directions in cryptography.
Information Theory, IEEE Transactions on, 22(6):644–654, 1976.


Itai Dinur, Orr Dunkelman, Nathan Keller, and Adi Shamir.
New attacks on Feistel Structures with Improved Memory Complexities.
In Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I, pages 433–454, 2015.

Further Reading III

Naofumi Homma, Atsushi Miyamoto, Takafumi Aoki, Akashi Satoh, and Adi Shamir.
Collision-Based Power Analysis of Modular Exponentiation Using Chosen-Message Pairs.
In Cryptographic Hardware and Embedded Systems - CHES 2008, 10th International Workshop, Washington, D.C., USA, August 10-13, 2008. Proceedings, pages 15–29, 2008.