COL759:
CRYPTOGRAPHY AND COMPUTER SECURITY

2022-23 (SEMESTER 1)

LECTURE 30: THE RSA PKE SCHEME

LAST TWO LECTURES

- The Discrete Log (DLOG) problem, the Decisional Diffie-Hellman (DDH) problem
- CRHF based on DLOG
- Public Key Encryption based on DDH

PLAN FOR TODAY'S LECTURE

- The RSA problem
  - The 'root finding' problem
  - Generalise (mod p) arithmetic to (mod N) arithmetic
- Public Key Encryption based on RSA

ROOT-FINDING IN $\mathbb{Z}_p$

$$p = 47$$

Goal: Finding cube roots modulo 47

Example: $x^3 \mod 47 = 2$

Deg. 3 polynomial, so at most 3 solutions.
Does there exist even one? Will we have multiple solutions?

Brute force: $O(p)$ computations

Obs. 1: If $x^3 \mod 47 = 2$, then for all d, $x^{3d} \mod 47 = 2^d \mod 47$

Obs. 2: $x^{46} \mod 47 = 1$, therefore for all d, $x^{3d} \mod 47 = x^{3d \mod 46} \mod 47$

Obs. 3: $\gcd(3,46) = 1$. Using Euclid's algorithm, $\exists d$ s.t. $3d = 1 \mod 46$.

$$2^d \mod 47 = x^{3d} \mod 47 = x^{3d \mod 46} \mod 47 = x^1 \mod 47$$

ROOT-FINDING IN $\mathbb{Z}_p$

Goal: Finding e^{th} roots modulo p

Solve for x : $x^e \mod p = t$

Given p, e, t s.t. $\gcd(e, p-1) = 1$

- Find d s.t. $e \cdot d = 1 \mod (p-1)$ Using Euclid's alg. d can be found efficiently
- $t^d \mod p = x^{e \cdot d} \mod p = x^{e \cdot d \mod (p-1)} \mod p = x \mod p$ Using repeated squaring $t^{e \cdot d} \mod p$ can be computed efficiently

THEOREM 30.01

Let $\gcd(e, p-1) = 1$. For every $t \in \mathbb{Z}_p$, there exists unique x such that $x^e \mod p = t$

If $\gcd(e, p-1) \neq 1$, then some t will have multiple solutions, others will have no solutions

ROOT-FINDING IN $\mathbb{Z}_N$

$$N = 48$$

Goal: Finding cube roots modulo 48

Example: $x^3 \mod 48 = 2$

Obs. 1: If $x^3 \mod 48 = 2$, then for all d, $x^{3d} \mod 48 = 2^d \mod 48$

Obs. 2: $x^{\cdots} \mod 48 = 1$, therefore for all d, $x^{3d} \mod 48 = x^{3d \mod \cdots} \mod 48$

Obs. 3: $\gcd(3, \ldots) = 1$. Using Euclid's algorithm, $\exists d$ s.t. $3d = 1 \mod \ldots$.

$$2^d \mod 48 = x^{3d} \mod 48 = x^{3d \mod \cdots} \mod 48 = x^1 \mod 48$$

What is this …?

LET'S GENERALISE SOME OF THE PROPERTIES OF $\mathbb{Z}_p^*$

Obs: $\mathbb{Z}_p^*$ is a group under operation $\times_p$

Definition of Group:
set with multiplicative operation

- Contains 1
- Closed under multiplication
- Every element has an inverse

We also need associativity, I'm skipping it here.

$$\mathbb{Z}_p^*$$

- For all $a \in \mathbb{Z}_p^*$, $a^{p-1} = 1$
- For all $a \in \mathbb{Z}_p^*$, $|\langle a \rangle|$ divides $p - 1$

$\mathbb{G}$

Strictly speaking, our proof of FLT, works only for Abelian groups. Luckily, we can restrict ourselves to Abelian groups for this course

- For all $a \in \mathbb{G}$, $a^{|\mathbb{G}|} = 1$
- For all $a \in \mathbb{G}$, $|\langle a \rangle|$ divides $|\mathbb{G}|$

LET'S GENERALISE SOME OF THE PROPERTIES OF $\mathbb{Z}_p^*$

Definition of Group:
set with multiplicative operation

- Contains 1
- Closed under multiplication
- Every element has an inverse

Examples of groups/non-groups

- $\{0,1,\ldots,9\}$ under mult. mod 10 — Not group, 0 has no inv.
- $\{1,\ldots,9\}$ under mult. mod 10 — Not group, 2 has no inv.
- $\{1,3,7,9\}$ under mult. mod 10 — This is a group.

$\mathbb{G}$

- For all $a \in \mathbb{G}$, $a^{|\mathbb{G}|} = 1$
- For all $a \in \mathbb{G}$, $|\langle a \rangle|$ divides $|\mathbb{G}|$

LET'S GENERALISE SOME OF THE PROPERTIES OF $\mathbb{Z}_p^*$

THEOREM 30.02

$S = \{a \leq N : \gcd(a, N) = 1\}$. This set is a group under multiplication modulo N.

Definition of Group:
set with multiplicative operation

- Contains 1
- Closed under multiplication
- Every element has an inverse

Proof sketch:

The set contains 1.
Using extended Euclid's algorithm
every element in this set has an inverse that is also in this set.
Finally, it is closed under multiplication, because if $a, b \in S,$ then $\gcd(a \cdot b \mod N, N) = 1$

$\mathbb{G}$

- For all $a \in \mathbb{G}$, $a^{|\mathbb{G}|} = 1$
- For all $a \in \mathbb{G}$, $|\langle a \rangle|$ divides $|\mathbb{G}|$

LET'S GENERALISE SOME OF THE PROPERTIES OF $\mathbb{Z}_p^*$

Definition

$$\mathbb{Z}_N^* = \{a \leq N : \gcd(a, N) = 1\}$$

$$\phi(N) = |\mathbb{Z}_N^*|$$

$\mathbb{Z}_N^*$

- For all $a \in \mathbb{Z}_N^*$, $a^{\phi(N)} = 1$
  (Euler's Totient Theorem)

Definition of Group:
set with multiplicative operation

- Contains 1
- Closed under multiplication
- Every element has an inverse

$\mathbb{G}$

- For all $a \in \mathbb{G}$, $a^{|\mathbb{G}|} = 1$
- For all $a \in \mathbb{G}$, $|\langle a \rangle|$ divides $|\mathbb{G}|$

HOW TO COMPUTE $\phi(N)$

Special case: $N = p \cdot q,$ where p and q are primes

$$\mathbb{Z}_N^* = \{1, \ldots, N\} \setminus \{\text{multiples of } p \text{ or } q\}$$

$$|\mathbb{Z}_N^*| = N - q - p + 1 = (p-1) \cdot (q-1)$$

ROOT-FINDING IN $\mathbb{Z}_N$

$$N = p \cdot q, \quad \phi(N) = (p-1) \cdot (q-1)$$

Goal: Finding e^{th} roots modulo N

Solve for x : $x^e \mod N = t$

Given $N = pq$, e, t s.t. $\gcd(e, \phi(N)) = 1$

- Find d s.t. $e \cdot d = 1 \mod \phi(N)$

  Using Euclid's alg.
  d can be found efficiently
  if $\phi(N)$ is known

- $t^d \mod N = x^{e \cdot d} \mod N = x^{e \cdot d \mod \phi(N)} \mod N = x \mod N$

  Using repeated squaring
  $t^{e \cdot d} \mod N$ can be
  computed efficiently

ROOT-FINDING IN $\mathbb{Z}_N$ WHEN $\phi(N)$ IS NOT KNOWN : RSA PROBLEM

RSA PROBLEM

Sample large primes $p, q,$ set $N = pq$.

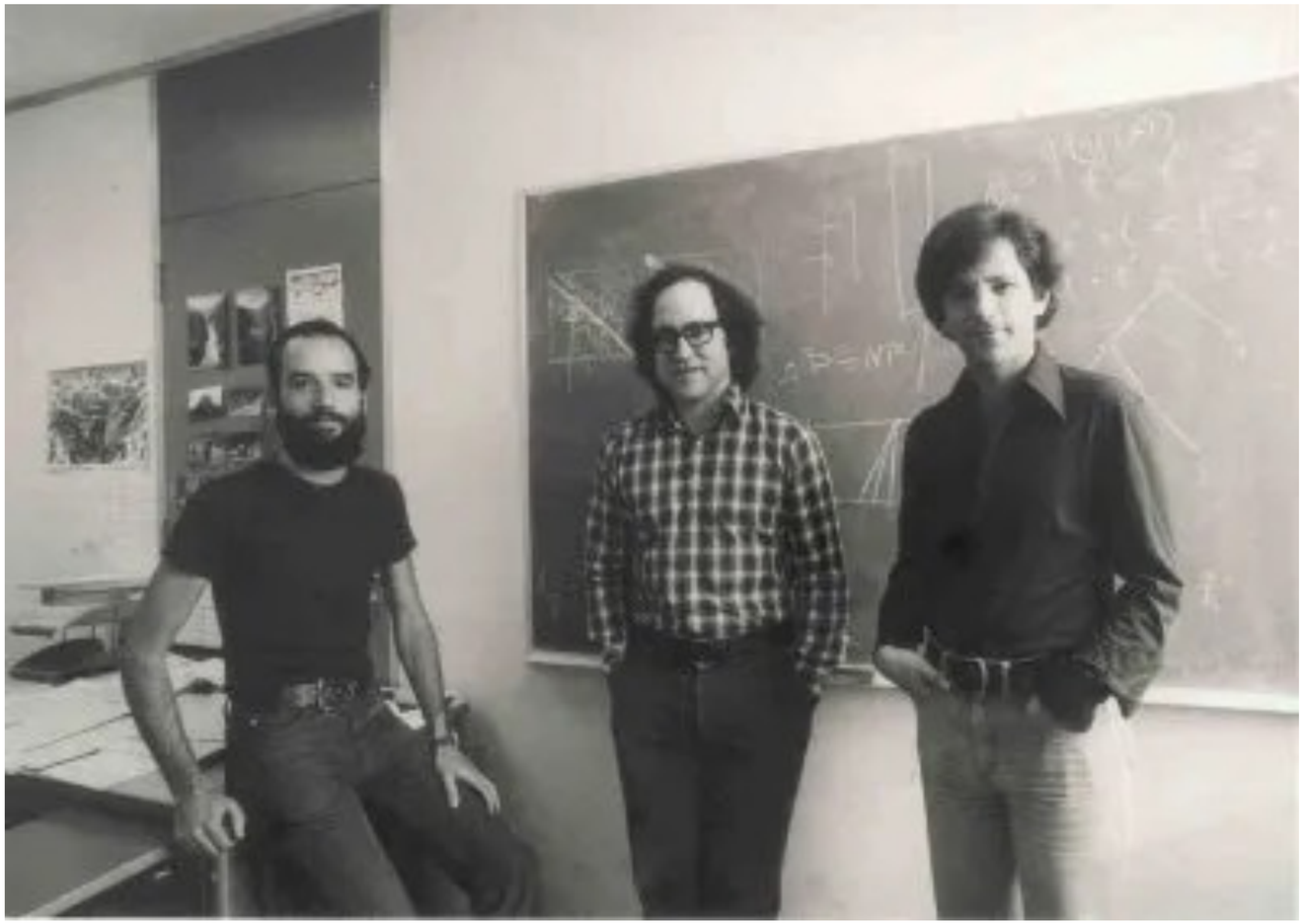
Sample e s.t. $\gcd(e, \phi(N)) = 1$.

Given $N, e,$ random $y \leftarrow \mathbb{Z}_N^*,$ find x s.t. $x^e = y$

Equivalent to choosing
random $x \leftarrow \mathbb{Z}_N^*$ and
setting $y = x^e \mod N$

Key points to remember:
1. RSA problem is about root-finding
2. Prime modulus doesn't work, so use composite modulus
3. $\gcd(e, \phi(N)) = 1$ needed for uniqueness of solution

RIVEST-SHAMIR-ADLEMAN



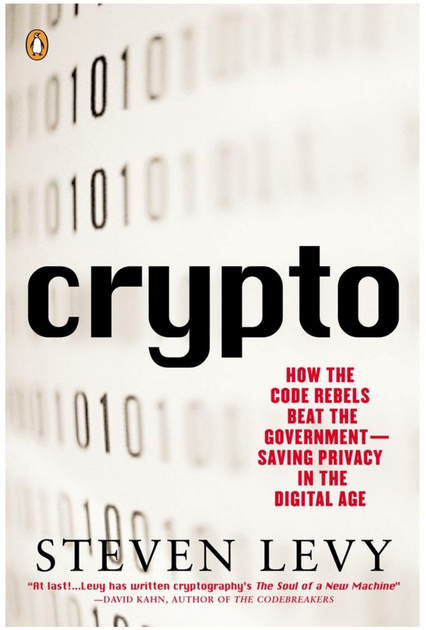
Adi Shamir Ron Rivest Leonard Adleman

Turing Award in 2002

Check out their Turing Award lectures (15-20 mins each)

Leonard Adleman: Pre-RSA Days
Ron Rivest: Early Days of RSA
Adi Shamir- Cryptography: State of the Science

Read about ‘crypto wars’



RSA PROBLEM (with security parameter n)

Sample large primes p, q in interval $[2^{n-1}, 2^n]$, set $N = pq$.

Sample e s.t. $\gcd(e, \phi(N)) = 1$.

Given N, e, random $y \leftarrow \mathbb{Z}_N^*$, find x s.t. $x^e = y$

FACTORING N $\iff$ COMPUTING $\phi(N)$

Clearly, given the factors of N, one can compute $\phi(N)$

Given $\phi(N)$ and N, one can compute $p + q$

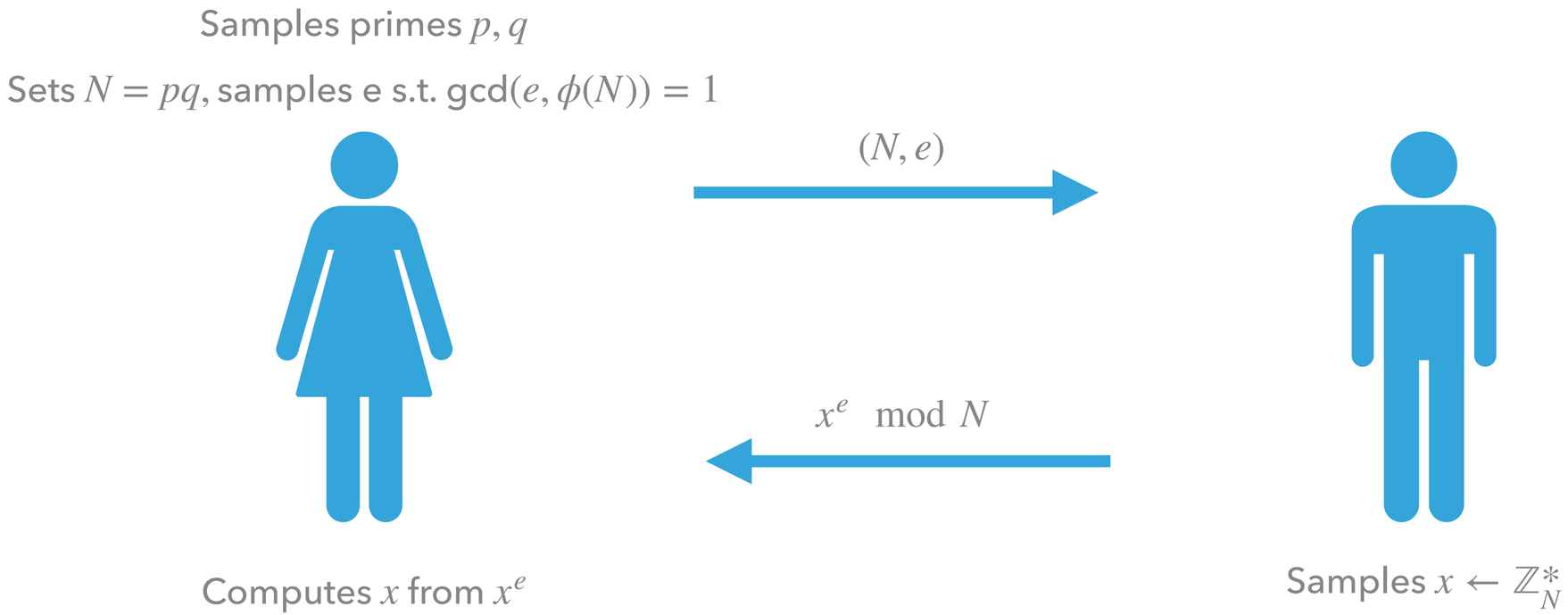
Given $p \cdot q$ and $p + q$, one can set up a quadratic equation whose roots are p, q

COMPUTING $\phi(N) \implies$ SOLUTION TO RSA PROBLEM

Given $\phi(N)$, one can solve the RSA problem as discussed in Slide 12

The other direction is not known.

WARMUP: RSA-BASED KEY AGREEMENT PROTOCOL



Adversary sees (N, e, x^e), cannot learn x (o/w it solves RSA problem)

However, adversary can learn some partial information about x. (For more info, read about the 'Jacobi symbol'. This is not part of syllabus)

‘TEXTBOOK’ RSA ENCRYPTION SCHEME

Key Generation: Sample large primes p, q. Set $N = p \cdot q$.

Sample e co-prime to $\phi(N)$, compute d s.t. $e \cdot d = 1 \mod \phi(N)$

Set $pk = (N, e)$, $sk = (p, q, d)$

$\text{Enc}(m, pk = (N, e))$: Message space is $\mathbb{Z}_N^*$

Output ct $= m^e \mod N$

$\text{Dec}(\text{ct}, sk = (p, q, d))$: Output $\text{ct}^d \mod N$

NOT SEMANTICALLY SECURE, AS ENC IS DETERMINISTIC.

HOW TO MAKE IT SECURE?

ATTEMPTS TO MAKE 'TEXTBOOK' RSA ENCRYPTION SCHEME SECURE

ATTEMPT 1

Key Generation: Sample large primes p, q. Set $N = p \cdot q$.

Sample e co-prime to $\phi(N)$, compute d s.t. $e \cdot d = 1 \mod \phi(N)$

Set $pk = (N, e)$, $sk = (p, q, d)$

$\text{Enc}(m, pk = (N, e))$: Message space is $\mathbb{Z}_N^*$

Sample $x \leftarrow \mathbb{Z}_N^*$

Output $\text{ct}_1 = x^e \mod N$, $\text{ct}_2 = (m \cdot x)^e \mod N$

NOT SEMANTICALLY SECURE

Given $x^e \mod N$ and $(m \cdot x)^e \mod N$,
one can compute $m^e \mod N$.
Can be used to distinguish
encryption of m_0 and m_1

ATTEMPTS TO MAKE 'TEXTBOOK' RSA ENCRYPTION SCHEME SECURE

ATTEMPT 2

Key Generation: Sample large primes p, q. Set $N = p \cdot q$.

Sample e co-prime to $\phi(N)$, compute d s.t. $e \cdot d = 1 \mod \phi(N)$

Set $pk = (N, e)$, $sk = (p, q, d)$

$\text{Enc}(m, pk = (N, e))$: Message space is $\mathbb{Z}_N^*$

Sample $x \leftarrow \mathbb{Z}_N^*$

Output $\text{ct}_1 = x^e \mod N$, $\text{ct}_2 = m \cdot x \mod N$

NOT SEMANTICALLY SECURE

Given $x^e \mod N$ and $m \cdot x \mod N$,
one can compute $(m \cdot x)^e \mod N$.
Then this is same as previous attempt.

ATTEMPTS TO MAKE 'TEXTBOOK' RSA ENCRYPTION SCHEME SECURE

ATTEMPT 3

Key Generation: Sample large primes p, q. Set $N = p \cdot q$.

Sample e co-prime to $\phi(N)$, compute d s.t. $e \cdot d = 1 \mod \phi(N)$

Set $pk = (N, e)$, $sk = (p, q, d)$

$\text{Enc}(m, pk = (N, e))$: Message space is $\mathbb{Z}_N^*$

Sample $x \leftarrow \mathbb{Z}_N^*$

Output $\text{ct}_1 = x^e \mod N$, $\text{ct}_2 = (m + x)^e \mod N$

Not sure if this scheme is secure or broken.
My guess is that there should be an attack,
but we couldn't find one during the lecture.
You all are encouraged to think about it.

Note that $m + x$ may not be in $\mathbb{Z}_N^*$.
This is not an issue for correctness as
$\alpha^{e \cdot d} = \alpha$ holds for all $\alpha \in \mathbb{Z}_N$

PROVABLY SECURE ENCRYPTION SCHEME BASED ON RSA

Key Generation: Sample large primes p, q. Set $N = p \cdot q$. Let ℓ denote the number of bits needed to represent N.

Sample e co-prime to $\phi(N)$, compute d s.t. $e \cdot d = 1 \mod \phi(N)$

Set $pk = (N, e)$, $sk = (p, q, d)$

Enc$(m, pk = (N, e))$: Message space is $\{0,1\}$

Choose $r \leftarrow \{0,1\}^{\ell-1}$, let x denote the unique number in $\mathbb{Z}_N^*$ whose binary representation is $r||m$

Output ct = $x^e \mod N$.

Dec(ct, $sk = (p, q, d)$): Compute $y = \text{ct}^d \mod N$

Output the last bit of y

This scheme is provably secure. The security proof relies on the following variant of the RSA assumption:

Given (N, e, x^e) for random $x \leftarrow \mathbb{Z}_N^*$, the least significant bit of x is indistinguishable from a uniformly random bit.

This variant is equivalent to the standard RSA assumption, although the proof is beyond the scope of our course. See this paper for more details.

HEURISTIC ENCRYPTION SCHEME BASED ON RSA

The scheme uses a hash function, say SHA, and a symmetric key enc. scheme

Key Generation: Sample large primes p, q. Set $N = p \cdot q$. Let ℓ denote the number of bits needed to represent N.

Sample e co-prime to $\phi(N)$, compute d s.t. $e \cdot d = 1 \mod \phi(N)$

Set $pk = (N, e)$, $sk = (p, q, d)$

$\mathsf{Enc}(m, pk = (N, e))$: Message space is $\{0,1\}^*$

Choose $x \leftarrow \mathbb{Z}_N^*$, let $\mathsf{ct}_1 = x^e \mod N$.

Compute $k = \mathsf{SHA}(x)$, $\mathsf{ct}_2 \leftarrow \mathsf{Enc}_{\mathsf{sym}}(m, k)$. Output $(\mathsf{ct}_1, \mathsf{ct}_2)$

$\mathsf{Dec}(\mathsf{ct}, sk = (p, q, d))$: Let $\mathsf{ct} = (\mathsf{ct}_1, \mathsf{ct}_2)$, $x = \mathsf{ct}_1^d \mod N$.

Compute $k = \mathsf{SHA}(x)$, output $m = \mathsf{Dec}_{\mathsf{sym}}(\mathsf{ct}_2, k)$

This is a heuristic scheme, however this is the version that is used in practice. Next class, we will discuss how to 'prove' security of such heuristic schemes. These proofs will not be in the standard model. Instead, we will introduce a new 'idealised setting' for discussing security of such schemes.

END OF LECTURE. THANK YOU!

EXERCISES

1. Prove that the following variant is as hard as the RSA problem.

RSA' PROBLEM (with security parameter n)

Sample large primes p, q in interval $[2^{n-1}, 2^n]$ set $N = pq$.

Sample e s.t. $\gcd(e, \phi(N)) = 1$.

Given (N, e, x^e) for random $x \leftarrow \mathbb{Z}_N$, find x.

2. Consider the following computational problem. Show that if there exists an efficient adversary that solves this problem, then there exists an efficient adversary that solves RSA problem.

Sample large primes p, q in interval $[2^{n-1}, 2^n]$ set $N = pq$.

Given N, output an element in $\mathbb{Z}_N \setminus \mathbb{Z}_N^*$