Public-Key Encryption

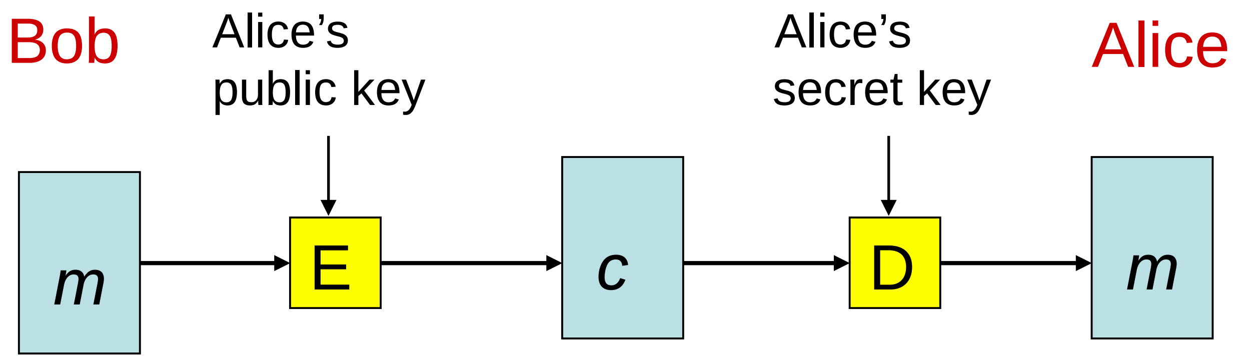
Reading: K&L Chapter 11

Public-Key Encryption

- Also known as asymmetric-key encryption.
- The receiver has a pair of keys:

  a public key *pk* and a private key *sk*.
- The public key, known to the public, is used for encryption.
- The private key, known only to its owner, is used for decryption.

Public-key Encryption



Why Public-Key Cryptography?

- Developed to address two main issues:
  - key distribution
  - digital signatures
- Invented by Diffie & Hellman in 1976.

Symmetric-key encryption scheme (for comparison)

- A tuple of polynomial-time algorithms: $\Pi = (Gen, Enc, Dec)$
- Key generation algorithm Gen: On input 1^n, outputs a key $k \in \{0,1\}^n$. We write $k \leftarrow Gen(1^n)$. (n: security parameter.)
- Encryption algorithm Enc: On input a key k and a message $m \in \{0,1\}^*$, outputs a ciphertext c. We write $c \leftarrow Enc_k(m)$.
- Decryption algorithm Dec: On input a key k and a ciphertext c, Dec outputs a message m or an error symbol $\perp$.
  We write $m := Dec_k(c)$.
- Correctness requirement: for every $k \leftarrow Gen(1^n)$ and $m \in \{0,1\}^*$,
  $$Dec_k\left(Enc_k(m)\right) = m.$$
- Gen, Enc are probabilistic. Dec, deterministic.

Public-key encryption scheme

- Gen: on input 1^n, outputs a pair of keys, (pk, sk), each of length at least n. We write $(pk, sk) \leftarrow Gen(1^n)$.
- Enc: on input a public key pk and a message $m \in M_{pk}$, outputs a ciphertext c. We write $c \leftarrow Enc_{pk}(m)$. (The message space M_{pk} may depend on pk.)
- Dec: On input a secret key sk and a ciphertext c, outputs a message m or an error symbol $\perp$. We write $m := Dec_{sk}(c)$.
- It is required that $\Pr\left[Dec_{sk}\left(Enc_{pk}(m)\right) = m :\ m \leftarrow M_{pk}\right] = 1$ except possibly with negligible probability over key pairs (pk, sk) output by $Gen(1^n)$.

Different notions of security

- EAV-security (against eavedroppers, ciphertext-only-attacks)
  - one encryption
  - multiple encryptions
- CPA-security (against chosen-plaintext attacks)
  - one encryption
  - multiple encryptions
- CCA-security (against chosen-ciphertext attacks)
  - one encryption
  - multiple encryptions

Ciphertext Indistinguishability

- Adversary A: a polynomial-time eavesdropper.
- $\Pi = (Gen, Enc, Dec)$: a public-key encryption scheme.
- Experiment $\mathrm{PubK}^{\mathrm{eav}}_{A,\Pi}(n)$:
  - $Gen(1^n)$ is run to obtain a pair of keys (pk, sk).
  - The adversary is given pk, and outputs a pair of messages $m_0, m_1 \in M_{pk}$ of the same length.
  - A random bit $b \leftarrow \{0,1\}$ is chosen; and a ciphertext $c \leftarrow E_{pk}(m_b)$ is computed and given to the adversary.
  - The adversary outputs a bit b'.
  - $\mathrm{PubK}^{\mathrm{eav}}_{A,\Pi}(n) = 1$ if and only if $b = b'$.

- Definition: A publick-key encryption scheme is EAV-secure if for every polynomial-time adversary A there exists a negligible function *negl* such that

$$\Pr\left[\mathrm{PubK}_{A,\Pi}^{\mathrm{eav}}(n)=1\right] \le \frac{1}{2}+\mathrm{negl}(n)$$

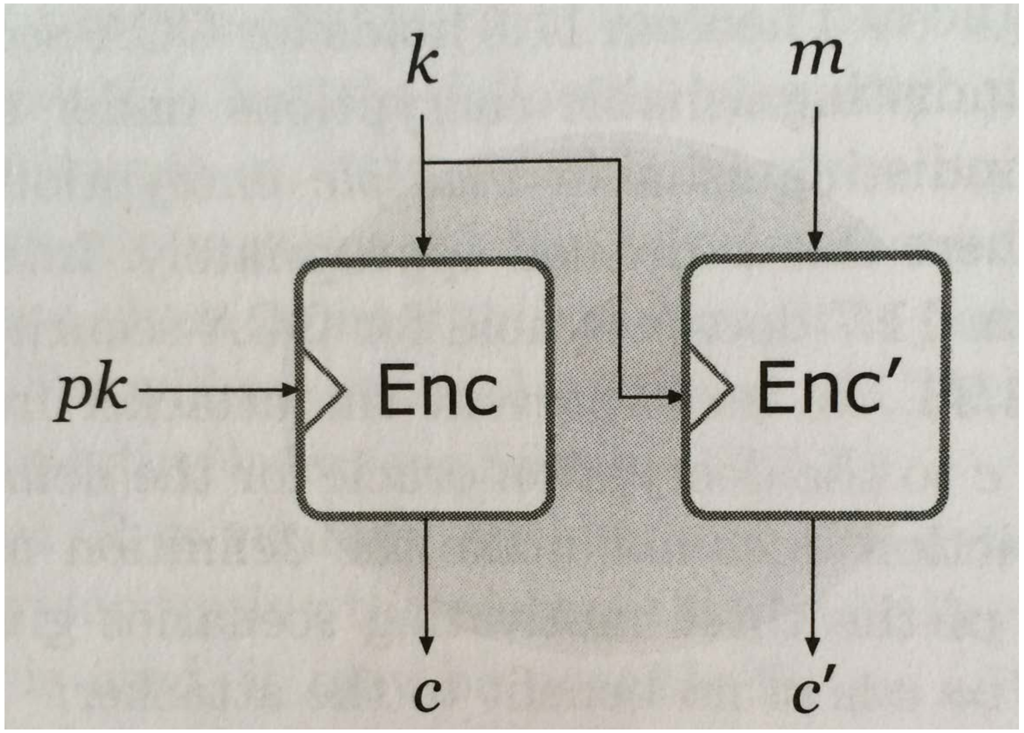
- We may similarly define CPA-security and CCA-security.

Remarks

- Since the adversary knows the publick key pk, it can encrypt any polynomial number of messages of its choice.
- That is, eavesdroppers are automatically capable of CPA's.
- Thus, if a public-key encryption scheme is EAV-secure, then it is also CPA-secure.
- A deterministic public-key encryption scheme is
  - not CPA-secure, and hence
  - not EAV-secure.
- If a public-key encryption scheme is CPA-secure, then it is CPA-secure for multiple encryptions.

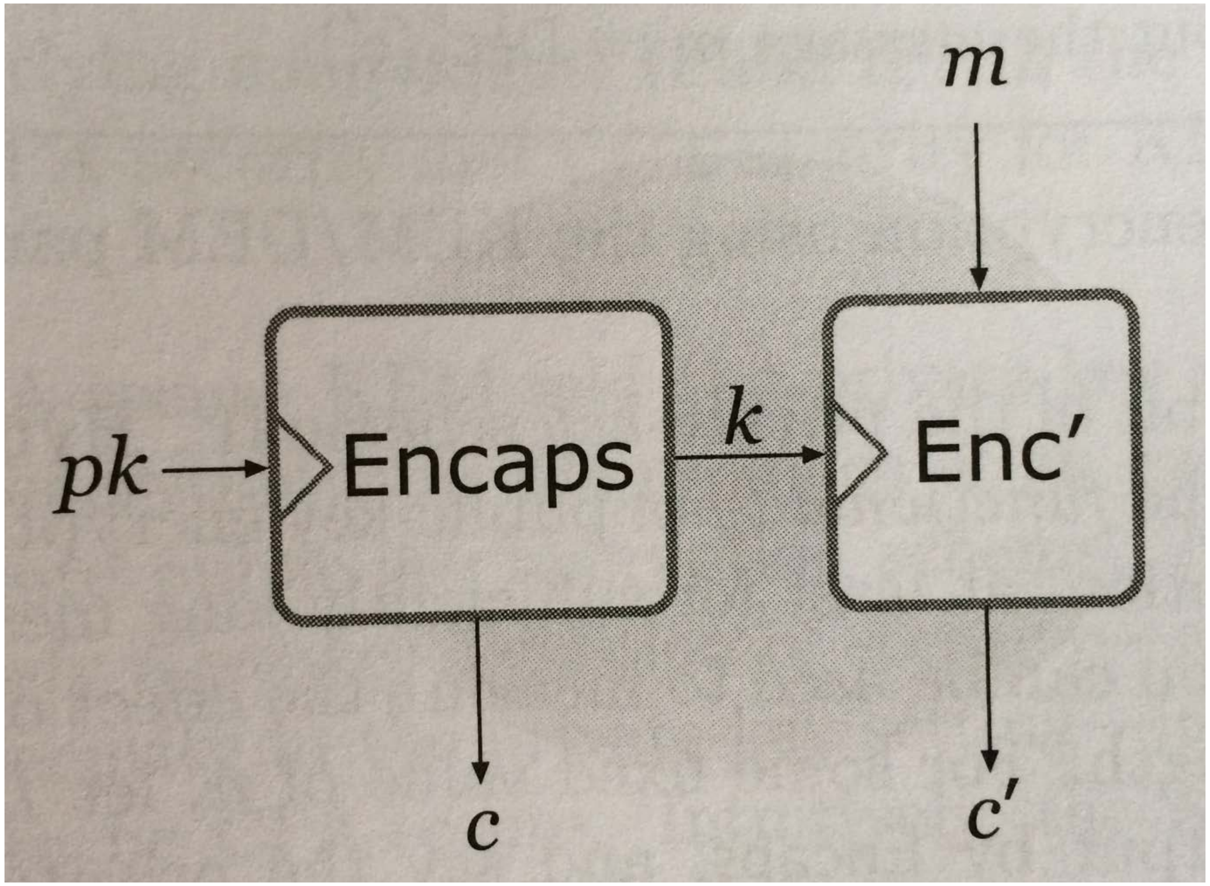
Hybrid Encryption

- Compared with private-key encryption, public-key encryption
  - is slower
  - has longer ciphertexts
- Hybrid encryption
  - Use public-key encryption to obtain a shared key k
  - Use private-key encryption to encrypt the message under key k



k
m
pk
Enc
Enc′
c
c′

The KEM/DEM Paradigm

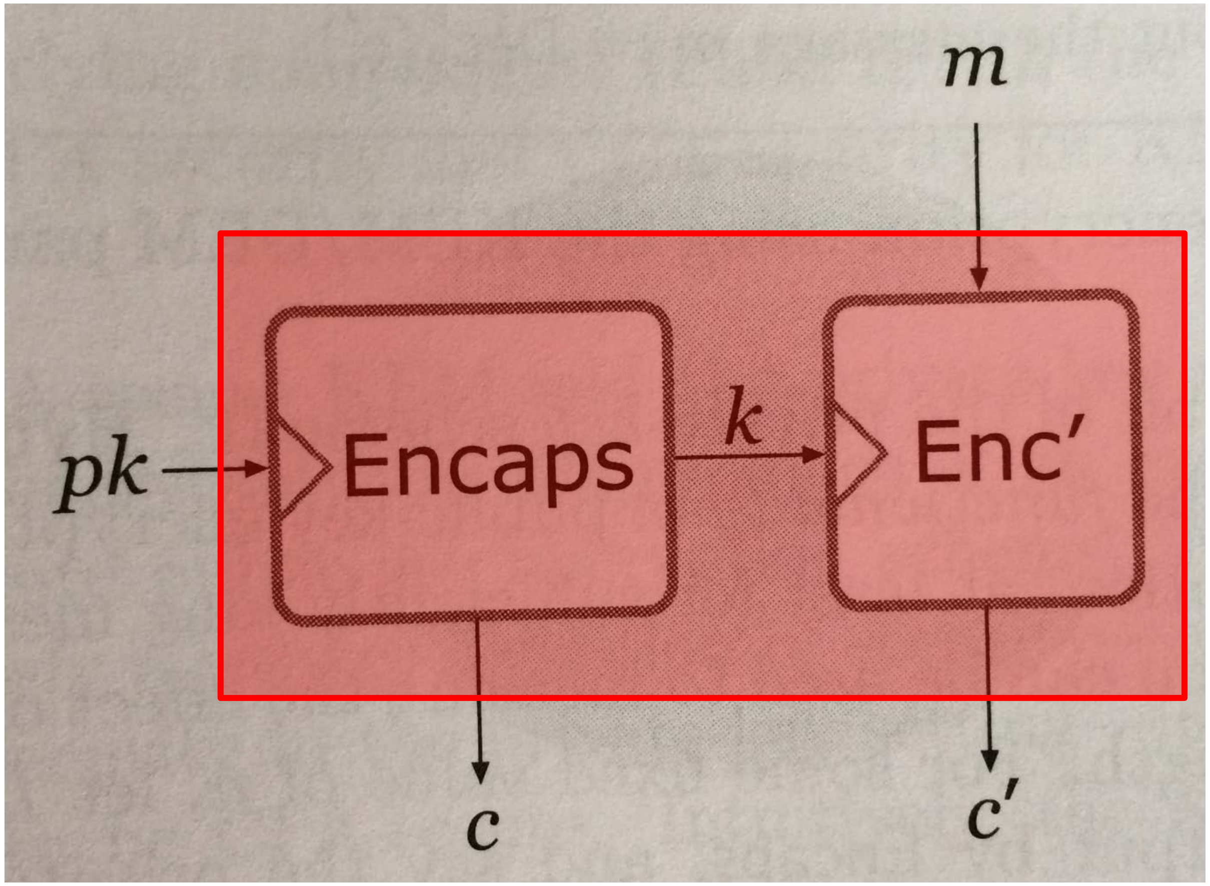


Key-encapsulation mechanism (KEM)

- *Gen*: on input 1^n, outputs a pair of keys, (pk, sk), each of length at least *n*. $(pk, sk) \leftarrow Gen(1^n)$.
- $Encaps:$ on input 1^n and a public key $pk,$ outputs a ciphertext c and a key $k \in \{0,1\}^{\ell(n)}$. We write $(c, k) \leftarrow Encaps_{pk}(1^n)$.
- $Decaps:$ on input a secret key sk and a ciphertext c, outputs a key k or an error symbol $\perp$. We write $k := Decaps_{sk}(c)$.
- It is required that with all but negligible probability over (pk, sk) output by $Gen(1^n)$, it holds:

$$(c, k) \leftarrow Encaps_{pk}(1^n) \quad \Rightarrow \quad Decaps_{sk}(c) = k$$

Hybrid encryption using KEM



Hybrid encryption using KEM

- Construct an encryption scheme $\Pi^{\text{hy}} = (Gen^{\text{hy}}, Enc^{\text{hy}}, Dec^{\text{hy}})$ from a KEM $\Pi = (Gen, Encaps, Decaps)$ and a private-key encryption scheme $\Pi' = (Gen', Enc', Dec')$.
- Gen^{hy} : on input 1^n, run $(pk, sk) \leftarrow Gen(1^n)$.
- Enc^{hy} : on input a public key pk and message $m \in \{0,1\}^*$,
  - $(c, k) \leftarrow Encaps_{pk}(1^n)$
  - $c' \leftarrow Enc'_k(m)$
  - output the ciphertext (c, c').
- Dec^{hy} : on input a secret key sk and a ciphertext (c, c'),
  - $k := Decaps_{sk}(c)$
  - $m := Dec'_k(c')$

CPA-security of KEMs

- $\Pi = (Gen, Encaps, Decaps)$: a KEM.
- Experiment $\mathrm{KEM}^{\mathrm{cpa}}_{A,\Pi}(n)$:
  - $Gen(1^n)$ is run to obtain a pair of keys (pk, sk).
  - $Encaps_{pk}(1^n)$ is run to generate (c, k) with $k \in \{0,1\}^{\ell(n)}$.
  - A random bit $b \leftarrow \{0,1\}$ is chosen; and

$$\hat{k} := \begin{cases} k & \text{if } b = 0 \\ \text{a random string in}\{0,1\}^{\ell(n)} & \text{else} \end{cases}$$

  - The adversary, given $(pk, c, \hat{k})$, outputs a bit b'.
  - $\mathrm{KEM}^{\mathrm{cpa}}_{A,\Pi}(n) = 1$ if and only if $b = b'$.
- Π is CPA-secure iff $\forall A, \Pr\left[\mathrm{KEM}^{\mathrm{cpa}}_{A,\Pi}(n) = 1\right] \leq \frac{1}{2} + \mathrm{negl}(n)$.

Security of hybrid encryption

- Let $\Pi^{\text{hy}} = (Gen^{\text{hy}}, Enc^{\text{hy}}, Dec^{\text{hy}})$ be constructed from

  $\Pi = (Gen, Encaps, Decaps)$ and

  $\Pi' = (Gen', Enc', Dec')$

  as above.

- Theorem: If Π is CPA-secure and Π' is EAV-secure, then Π^{hy} is CPA-secure.

- Theorem: If Π is CCA-secure and Π' is CCA-secure, then Π^{hy} is CCA-secure.

One-way function with trapdoor (informal)

Easy: $x \xrightarrow{f} y$

Hard: $x \xleftarrow{f^{-1}} y$

Easy: $x \xleftarrow[\text{trapdoor}]{f^{-1}} y$

Use trapdoor as the private key.

- Most public-key encryption schemes are based on assumed one-way functions.
- Most one-way functions come from number theory.

Modular Arithmetic

Reading: K&L Section 8.1

Integers

- $a \mid b$: a divides b, a is a divisor of b.
- $\gcd(a,b)$: greatest common divisor of a and b.
- Coprime or relatively prime: $\gcd(a,b)=1$.
- Euclid's algorithm: compute $\gcd(a,b)$.
- Extented Euclid's algorithm: compute integers x and y such that $ax+by=\gcd(a,b)$.

Integers modulo N

- Let $N \geq 2$ be an integer.
- Definition: a is congruent to b modulo N, written

  $$a \equiv b \mod N \qquad (\text{or } a = b \mod N \text{ as in the book})$$

  if $N \mid (a-b)$, i.e., a and b have the same remainder when divided by N.
- Define $[a]_N = \{x \in \mathbb{Z} : x \equiv a \mod N\}$.
- $[a]_N$ is called a residue class modulo N, and a is a representative of that class.

- There are exactly N residue classes modulo N:

  $[0]_N,\ [1]_N,\ [2]_N,\ \ldots,\ [N-1]_N.$

- $\mathbb{Z} = [0]_N \cup [1]_N \cup \cdots \cup [N-1]_N.$

- If $x \in [a]_N$, $y \in [b]_N$, then

  $$x + y \in [a+b]_N \text{ and } x \cdot y \in [a \cdot b]_N.$$

- Define addition and multiplication for residue classes:

  $$[a]_N + [b]_N = [a+b]_N$$

  $$[a]_N \cdot [b]_N = [a \cdot b]_N.$$

Group

- A group, denoted by $(G, \circ)$, is a set G along with a binary operation $\circ$ such that:

  1. (Closure) For all $x, y \in G$, $x \circ y \in G$
  2. (Associativity) $x \circ (y \circ z) = (x \circ y) \circ z$
  3. (Existence of an identity) There exists an identity $e \in G$ s.t. $\forall x \in G$, $e \circ x = x \circ e = x$
  4. (Existence of inverses) For all $x \in G$, there exists an element $y \in G$ s.t. $x \circ y = y \circ x = e$. Such a y is called an inverse of x.

- A group $(G, \circ)$ is abelian (or commutative) if for all $x, y \in G$, $x \circ y = y \circ x$.

- A group $(G, \circ)$ is finite if $|G|$ is finite.
- The identity of a group is unique.
- The inverse of an element is unique.
- If $(G, \circ)$ is a group and $H \subseteq G$ itself is a group under the same operation $\circ$, then H is a subgroup of G.
- Examples: $(\mathbb{Z}, +)$, $(\mathbb{Q}, +)$, $(\mathbb{Q} \setminus \{0\}, \times)$, $(\mathbb{R}, +)$, $(\mathbb{R} \setminus \{0\}, \times)$.

- Define $\mathbb{Z}_N = \{[0]_N,\ [1]_N,\ \ldots,\ [N-1]_N\}$.
- Or, more conveniently, $\mathbb{Z}_N = \{0,\ 1,\ \ldots,\ N-1\}$.
- $(\mathbb{Z}_N, +)$ forms an abelian (additive) group.
- For $a,\ b \in \mathbb{Z}_N$,
  - $a + b = \big((a+b) \bmod N\big)$.

    (That is, $[a]_N + [b]_N = [a+b]_N = [a+b \bmod N]_N$.)
  - 0 is the identity element.
  - The inverse of a, denoted by $-a$, is $N-a$.
- When doing addition/substraction in $\mathbb{Z}_N$, just do the regular addition/substraction and reduce the result modulo N.
  - In $\mathbb{Z}_{10}$, $5+5+9+4+6+2+8+3=?$

- $(\mathbb{Z}_N, *)$ is not a group, because 0^{-1} does not exist.
- Even if we exclude 0 and consider only $\mathbb{Z}_N^+ = \mathbb{Z}_N \setminus \{0\}$,

  $(\mathbb{Z}_N^+, *)$ is not necessarily a group; some a^{-1} may not exist.
- For $a \in \mathbb{Z}_N$, a^{-1} exists if and only if $\gcd(a, N) = 1$.

- Let $\mathbb{Z}_N^* = \{a \in \mathbb{Z}_N : \ \gcd(a, N) = 1\}$.
- $(\mathbb{Z}_N^*, *)$ is an abelian (multiplicative) group.
  - $a * b = (ab \bmod N)$.
  - 1 is the identity element.
  - The inverse of a, written a^{-1}, can be computed by the Extended Euclidean Algorithm.
- For example, $Z_{12}^* = \{1, 5, 7, 11\}$. $5 * 7 = 35 \bmod 12 = 11$.
- Q: How many elements are there in $\mathbb{Z}_N^*$?

- Euler's totient function:

$$\varphi(N) = \left|\mathbb{Z}_N^*\right|$$

$$= \left|\left\{a : \ 1 \le a \le N \ \text{ and } \ \gcd(a, N) = 1\right\}\right|$$

- Theorem:

1. $\varphi(p^e) = (p-1)\,p^{e-1}$ for prime p
2. $\varphi(ab) = \varphi(a)\,\varphi(b)$ if $\gcd(a,b) = 1$

- Let G be a (multiplicative) finite group.
- The order of G is defined as $\operatorname{ord}(G) = |G|$.
- The order of $a \in G$, written $\operatorname{ord}(a)$, is the smallest positive integer k such that $a^k = e$. (e, identity element.)
- Lagrange's theorem: For any element $a \in G$, $\operatorname{ord}(a) \mid \operatorname{ord}(G)$.
- Corollary: For any element $a \in G$, $a^{\operatorname{ord}(G)} = a^{|G|} = e$.
- Corollary: For any element $a \in G$, $a^m = a^{m \bmod |G|}$.

- Fermat's little theorem:

  If $a \in \mathbb{Z}_p^*$ (p a prime), then $a^{\varphi(p)} = a^{p-1} = 1$ in $\mathbb{Z}_p^*$.

- Euler's theorem:

  If $a \in \mathbb{Z}_N^*$ (for any $N > 1$), then $a^{\varphi(N)} = 1$ in $\mathbb{Z}_N^*$.

- Corollary:

  If $a \in \mathbb{Z}_N^*$ (for any $N > 1$), then $a^m = a^{m \bmod \varphi(N)}$ in $\mathbb{Z}_N^*$.

Example: $N = 15$

- $\mathbb{Z}_{15}^* = \{1, 2, 4, 7, 8, 11, 13, 14\}$
- $\left|\mathbb{Z}_{15}^*\right| = \varphi(15) = \varphi(3) \times \varphi(5) = 2 \times 4 = 8$
-

$a \in \mathbb{Z}_{15}^*$:	1	2	4	7	8	11	13	14
$\mathrm{ord}(a)$:	1	4	2	4	4	2	4	2

- $a^{\varphi(15)} = a^8 = 1$
- $13^{816243240481} = ?$

The Chinese Remainder Problem

- A problem described in an ancient Chinese arithmetic book by Sun Tze (around 300AD, the author of The Art of War).
- Problem: We have a number of objects, but we do not know exactly how many. If we count them by threes we have two left over. If we count them by fives we have three left over. If we count them by sevens we have two left over. How many objects are there?

- Mathematically, if $x \equiv 2 \bmod 3$, $x \equiv 3 \bmod 5$, $x \equiv 2 \bmod 7$, what is x?

Chinese remainder theorem

If integers $n_1, \ldots, n_k$ are pairwise coprime,

then the system of congruences

$$\begin{cases} x \equiv a_1 \bmod n_1 & // \; a_1 \in \mathbb{Z}_{n_1} \; // \\ x \equiv a_2 \bmod n_2 & // \; a_2 \in \mathbb{Z}_{n_2} \; // \\ \quad \vdots & \\ x \equiv a_k \bmod n_k & // \; a_k \in \mathbb{Z}_{n_k} \; // \end{cases}$$

has a unique solution modulo $N = n_1 n_2 \ldots n_k$:

$$x \equiv \sum_{i=1}^{k} a_i N_i y_i \bmod N$$

where $N_i = N/n_i$ and $y_i = N_i^{-1} \bmod n_i$ (A formula by Gauss)

Example: Chinese remainder theorem

Suppose

$$x \equiv 1 \mod 3$$
$$x \equiv 6 \mod 7$$
$$x \equiv 8 \mod 10$$

By the Chinese remainer theorem, the solution is:

$$\begin{aligned}
x &\equiv \left(1\times 70\times(70^{-1} \bmod 3)+6\times 30\times(30^{-1} \bmod 7)+8\times 21\times(21^{-1} \bmod 10)\right) \bmod 210 \\
&\equiv \left(1\times 70\times(1^{-1} \bmod 3)+6\times 30\times(2^{-1} \bmod 7)+8\times 21\times(1^{-1} \bmod 10)\right) \bmod 210 \\
&\equiv \left(1\times 70\times 1+6\times 30\times 4+8\times 21\times 1\right) \bmod 210 \\
&\equiv 958 \bmod 210 \\
&\equiv 118 \bmod 210
\end{aligned}$$

Chinese remainder theorem

Let $N = n_1 n_2 \ldots n_k$, where $n_1, \ldots, n_k$ are pairwise coprime.

There is a one-to-one correspondence

$$\mathbb{Z}_N \longleftrightarrow \mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2} \times \cdots \times \mathbb{Z}_{n_k}$$

$$x \longleftrightarrow (x \bmod n_1,\ x \bmod n_2,\ \ldots,\ x \bmod n_k)$$

Then,

- Denote the mapping $\psi : \mathbb{Z}_N \to \mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2} \times \cdots \times \mathbb{Z}_{n_k}$.
- $\psi(x \cdot y) = \psi(x) \cdot \psi(y)$.
- $\psi(x + y) = \psi(x) + \psi(y)$.

- Computations in $\mathbb{Z}_N$ can be done by performing corresponding computations in $\mathbb{Z}_{n_1}$, $\mathbb{Z}_{n_2}$, …, $\mathbb{Z}_{n_k}$, and then solve the CRP.
- If $\begin{cases} a & \leftrightarrow & (a_1, \ \dots, a_k) \\ b & \leftrightarrow & (b_1, \ \dots, b_k) \end{cases}$

  then

$$\begin{array}{lcll} a \pm b & \leftrightarrow & (a_1 \pm b_1, \ \dots, a_k \pm b_k) & \\ a \times b & \leftrightarrow & (a_1 \times b_1, \ \dots, a_k \times b_k) & \\ a \div b & \leftrightarrow & (a_1 \div b_1, \ \dots, a_k \div b_k) & \text{if } b \in Z_N^* \\ \uparrow & & \quad \uparrow \qquad\qquad\qquad \uparrow & \\ \text{mod } N & & \text{mod } n_1 \qquad \text{mod } n_k & \end{array}$$

Example: Chinese remainder theorem

- $\mathbb{Z}_{15} \leftrightarrow \mathbb{Z}_3 \times \mathbb{Z}_5 \quad \left(\mathbb{Z}_{15}^* \leftrightarrow \mathbb{Z}_3^* \times \mathbb{Z}_5^*\right)$

  $8 \leftrightarrow \left(8 \bmod 3,\ 8 \bmod 5\right) = (2,3)$

  $11 \leftrightarrow \left(11 \bmod 3,\ 11 \bmod 5\right) = (2,1)$

- Suppose we want to compute $8 \times 11 \bmod 15$.
- $8 \times 11 \bmod 15 \leftrightarrow (2 \times 2 \bmod 3,\ 3 \times 1 \bmod 5) = (1,3)$.
- $x \leftrightarrow (1,3)$ (which number $x \in \mathbb{Z}_{15}$ corresponds to (1,3)?)
- Solve $\begin{cases} x \equiv 1 \bmod 3 \\ x \equiv 3 \bmod 5 \end{cases} \Rightarrow x = 13$

Algorithms

- $\gcd(a,b) \qquad //1 \le a,b \le N//$
- $a^{-1} \bmod N$
- $a^{k} \bmod N$
- Running time: $O\left(\log^{3} N\right) = O\left(\|N\|^{3}\right)$

Euclidean Algorithm

Comment: compute $\gcd(a,b)$, where $a > b > 1$.

```
$r_0 := a$
$r_1 := b$
for $i := 1,\ 2,\ \ldots$ until $r_{n+1} = 0$
    $r_{i+1} := r_{i-1} \bmod r_i$
return $(r_n)$
```

Running time:

- $O(\log a)$ iterations; $O(\log^2 a)$ time for each mod.
- Overall running time: $O(\log^3 a)$

Extended Euclidean Algorithm

Given $a > b > 0$, compute x, y such that $\gcd(a,b) = ax + by$.

Example: $\gcd(299, 221) = ?$

$$299 = 1 \times 221 + 78$$
$$221 = 2 \times 78 + 65$$
$$78 = 1 \times 65 + 13$$
$$65 = 5 \cdot 13 + 0$$

$$\begin{aligned} \gcd(229, 221) &= 13 = 78 - 65 \\ &= 78 - (221 - 2 \times 78) = 3 \cdot 78 - 221 \\ &= 3 \times (299 - 1 \cdot 221) - 221 \\ &= 3 \times 299 - 4 \times 221 \end{aligned}$$

How to compute $a^{-1} \bmod N$

- That is, for $a \in \mathbb{Z}_N^*$, compute a^{-1} in $\mathbb{Z}_N^*$.
- a^{-1} exists if and only if $\gcd(a, N) = 1.$
- Use the extended Euclidean algorithm to find $x,\ y$ such that $ax + Ny = 1.$
- Then, in $\mathbb{Z}_N^*$, we have

$$[a]\cdot[x]+[N]\cdot[y]=[1]$$

$$\Rightarrow\ [a]\cdot[x]=[1]$$

$$\Rightarrow\ [a]^{-1}=[x]$$

Example

- Compute $15^{-1} \mod 47$.

$$47 = 15 \times 3 + 2 \quad \text{(divide 47 by 15; remainder} = 2)$$

$$15 = 2 \times 7 + 1 \quad \text{(divide 15 by 2; remainder} = 1)$$

$$\begin{aligned} 1 &= 15 - 2 \times 7 && (\mathrm{mod}\ 47) \\ &= 15 - (47 - 15 \times 3) \times 7 && (\mathrm{mod}\ 47) \\ &= 15 \times 22 - 47 \times 7 && (\mathrm{mod}\ 47) \\ &= 15 \times 22 && (\mathrm{mod}\ 47) \end{aligned}$$

$15^{-1} \mod 47 = 22$

That is, $15^{-1} = 22$ in $\mathbb{Z}_{47}^{*}$

Algorithm: Square-and-Multiply(x, c, N)

Comment: compute $x^c \bmod N$, where $c = c_k c_{k-1} \ldots c_0$ in binary.

```
$z \leftarrow 1$
for $i \leftarrow k$ downto 0 do
        $z \leftarrow z^2 \bmod N$
         if $c_i = 1$
         then $z \leftarrow (z \times x) \bmod N$     } i.e., $z \leftarrow (z \times x^{c_i}) \bmod N$
return ($z$)
```

Note: At the end of iteration i, $z = x^{c_k \ldots c_i}$.

Example: $11^{23} \bmod 187$

$$23 = 10111_b$$

$$z \leftarrow 1$$

$z \leftarrow z^2 \cdot 11 \bmod 187 = 11$ (square and multiply)

$z \leftarrow z^2 \bmod 187 = 121$ (square)

$z \leftarrow z^2 \cdot 11 \bmod 187 = 44$ (square and multiply)

$z \leftarrow z^2 \cdot 11 \bmod 187 = 165$ (square and multiply)

$z \leftarrow z^2 \cdot 11 \bmod 187 = 88$ (square and multiply)

RSA Encryption

The RSA Cryptosystem

- By Rivest, Shamir & Adleman of MIT in 1977.
- Best known and most widely used public-key scheme.
- Based on the assumed one-way property of modular powering:

$$f: x \rightarrow x^e \mod N \qquad \text{(easy)}$$

$$f^{-1}: x^e \rightarrow x \mod N \qquad \text{(hard)}$$

- Related to the hardness of integer factorization.

Idea behind RSA

It works in group $\mathbb{Z}_N^*$.

Encryption (easy): $x \xrightarrow{\text{RSA}} x^e$

Decryption (hard): $x \xleftarrow{\text{RSA}^{-1}} x^e$

Looking for a trapdoor: $(x^e)^d = x.$

If d is a number such that $ed \equiv 1 \bmod \varphi(N),$ then

$ed = k\varphi(N) + 1$ for some $k,$ and thus in the group $\mathbb{Z}_N^*$,

$$(x^e)^d = x^{ed} = x^{\varphi(N)k+1} = \left(x^{\varphi(N)}\right)^k \cdot x = 1 \cdot x = x.$$

RSA encryption scheme

- Key generation:

  (a) Choose two large primes p and q, and let $N := pq$.

  ($\|p\|$, $\|q\|$ determined by the security parameter.)

  (b) Choose e, $1 < e < \varphi(N)$, coprime to $\varphi(N)$, and

  compute $d := e^{-1} \bmod \varphi(N)$. ($ed \equiv 1 \bmod \varphi(N)$.)

  (c) Public key: $pk = (N, e)$. Secret key: $sk = (N, d)$.

- Encryption: $Enc_{pk}(x) := x^e \bmod N$, where $x \in \mathbb{Z}_N^*$.
- Decryption: $Dec_{sk}(y) := y^d \bmod N$, where $y \in \mathbb{Z}_N^*$.

Why RSA Works?

- The setting of RSA is the group $\left(\mathbb{Z}_N^*, \bullet\right)$:
  - In group $\left(\mathbb{Z}_N^*, \bullet\right)$, for any $x \in \mathbb{Z}_N^*$, we have $x^{\varphi(N)} = 1$ and, thus, $x^m = x^{m \bmod \varphi(N)}$.
  - We have chosen $e,\ d$ such that $ed \equiv 1 \mod \varphi(N)$, so, $ed \bmod \varphi(N) = 1$.
  - For $x \in \mathbb{Z}_N^*$, $\left(x^e\right)^d = x^{ed} = x^{ed \bmod \varphi(N)} = x^1 = x$.

What if $x \in \mathbb{Z}_N \setminus \mathbb{Z}_N^*$?

- RSA still works, but not secure.
- $x \notin \mathbb{Z}_N^* \;\Rightarrow\; \gcd(x, N) \neq 1 \;\Rightarrow\; p \mid x$ or $q \mid x$.
- Say $p \mid x$ and $x \neq 0$ (it is trivial if $x = 0$). Then,

$$\begin{cases} x \equiv 0 \mod p \\ x \equiv x \mod q \end{cases} \quad \text{and} \quad \begin{cases} x^{ed} \equiv 0 \mod p \\ x^{ed} \equiv x \mod q \end{cases}$$

(The last "$\equiv$" holds $\because ed \equiv 1 \bmod \varphi(N) \Rightarrow ed \equiv 1 \bmod \varphi(q)$.)

- Both x and x^{ed} are a solution of the system, so by CRT

$$x^{ed} \equiv x \mod N \Rightarrow x^{ed} \bmod N = x \Rightarrow Dec\big(Enc(x)\big) = x.$$

RSA Example: Key Setup

- Select two primes: $p = 17,\ q = 11.$
- Compute the modulus $N = pq = 187.$
- Compute $\varphi(N) = (p-1)(q-1) = 160.$
- Select e between 0 and 160 such that $\gcd(e, 160) = 1.$

  Say $e = 7.$
- Compute $d = e^{-1} \bmod \varphi(N) = 7^{-1} \bmod 160 = 23$

  (using extended Euclid's algorithm).
- Public key: $pk = (e,\ N) = (7,\ 187).$
- Secret key: $sk = (d,\ N) = (23,\ 187).$

RSA Example: Encryption & Decryption

- Suppose $m = 88$.
- Encryption: $c = m^e \bmod N = 88^7 \bmod 187 = 11$.
- Decryption: $m = c^d \bmod N = 11^{23} \bmod 187 = 88$.
- When computing $11^{23} \bmod 187$, we do not first compute 11^{23} and then reduce it modulo 187.
- Rather, use square-and-multiply, and reduce intermediate results modulo 187 whenever they get bigger than 187.

Encryption Key *e*

- To speed up encryption, small values are usually used for *e*.
- Popular choices include $3,\ 17 = 2^4 + 1,\ 65537 = 2^{16} + 1.$ These values have only two 1's in their binary representation.
- There is an interesting attack on small *e*.

Decryption Key *d*

- One may be tempted to use a small d to speed up decryption.
- Unfortunately, that is risky.
- Wiener's attack: If $d < \frac{N^{1/4}}{3}$ and $p < q < 2p$, then the decryption exponent d can be computed from (N, e).
- CRT can be used to speed up decryption.

Speeding up Decryption by CRT

- Decryption: $c^d \bmod N$ (i.e., compute c^d in $\mathbb{Z}_N^*$)
- Time: $O(\|N\|^3)$. $\|N\| = \lfloor \log_2 N \rfloor + 1$
- Instead of computing $c^d \bmod N$ directly, we compute
  - $c_1 := c \bmod p$, and $c_2 := c \bmod q$
  - $m_1 := c_1^{d \bmod \varphi(p)} \bmod p$, and $m_2 := c_2^{d \bmod \varphi(q)} \bmod q$
  - recover the plaintext by solving $\begin{cases} x \equiv m_1 \bmod p \\ x \equiv m_2 \bmod q \end{cases}$
- Time: about $1/4$ of the direct computation.
- If $N = p_1 p_2 \ldots p_t$, this strategy will speed up even more.

Attacks on RSA

Attacks on RSA

- Five categories of attacks on RSA:
  - brute-force key search

    (infeasible given the large key space)
  - mathematical attacks
  - miscellaneous attacks
  - timing attacks
  - chosen ciphertext attacks

Mathematical Attacks

- Factor N into *pq*. Then $\varphi(N)=(p-1)(q-1)$ and $d=e^{-1} \bmod \varphi(N)$ can be calculated easily.
- Determine $\varphi(N)$ directly. Equivalent to factoring N. Knowing $\varphi(N)$ will enable us to factor N by solving

$$\begin{cases} N = pq \\ \varphi(N)=(p-1)(q-1) \end{cases}$$

- Determine d directly. If d is known, N can be factored with high probability.

Integer Factorization

- A difficult problem.
- More and more efficient algorithms have been developed.
- In 1977, RSA challenged researchers to decode a ciphertext encrypted with a key (N) of 129 digits (428 bits). Prize: \$100. RSA thought it would take quadrillion years to break the code using fastest algorithms and computers of that time. Solved in 1994.
- In 1991, RSA put forward more challenges, with prizes, to encourage research on factorization.

RSA Numbers

- Each RSA number is a semiprime. (A number is semiprime if it is the product of two primes.)
- There are two labeling schemes.
  - by the number of decimal digits:
    RSA-100, ..., RSA-500, RSA-617.
  - by the number of bits:
    RSA-576, 640, 704, 768, 896, 1024, 1536, 2048.

RSA Numbers which have been factored

- RSA-100 (332 bits), 1991, 7 MIPS-year, Quadratic Sieve.
- RSA-110 (365 bits), 1992, 75 MIPS-year, QS.
- RSA-120 (398 bits), 1993, 830 MIPS-year, QS.
- RSA-129 (428 bits), 1994, 5000 MIPS-year, QS.
- RSA-130 (431 bits), 1996, 1000 MIPS-year, GNFS.
- RSA-140 (465 bits), 1999, 2000 MIPS-year, GNFS.
- RSA-155 (512 bits), 1999, 8000 MIPS-year, GNFS.
- RSA-160 (530 bits), 2003, Lattice Sieve.
- RSA-576 (174 digits), 2003, Lattice Sieve.
- RSA-640 (193 digits), 2005, Lattice Sieve.
- RSA-200 (663 bits), 2005, Lattice Sieve.

RSA-200 =

27,997,833,911,221,327,870,829,467,638,

722,601,621,070,446,786,955,428,537,560,

009,929,326,128,400,107,609,345,671,052,

955,360,856,061,822,351,910,951,365,788,

637,105,954,482,006,576,775,098,580,557,

613,579,098,734,950,144,178,863,178,946,

295,187,237,869,221,823,983.

Remarks

- In light of current factorization technologies, RSA recommends $\|N\| = 1024$ to 2048 bits.
- If a message $m \in \mathbb{Z}_N \setminus \mathbb{Z}_N^*$,
  - RSA works, but
  - Since $\gcd(m, N) > 1$, the sender can factor N.
  - Since $\gcd(m^e, N) > 1$, the adversary can factor N, too.
- Question: how likely is $m \in \mathbb{Z}_N \setminus \mathbb{Z}_N^*$?

Miscellaneous attacks against RSA

- Common modulus:
  - If two users use the same modulus N and their encryption exponents e_1 and e_2 are coprime, then a message m sent to them, encrypted as $c_1 := m^{e_1} \bmod N$ and $c_2 := m^{e_2} \bmod N$, is not protected by RSA.
  - For e_1, e_2 coprime

$$\Rightarrow \; re_1 + se_2 = 1 \text{ for some } r,\ s$$

$$\Rightarrow \; m = m^{re_1+se_2} = m^{re_1+se_2} \bmod N = c_1{}^r c_2{}^s \bmod N.$$

- Another problem with common modulus:
  - Owners of keys (N, e, d) usually do not know $N = pq$.
  - But, actually, given (N, e, d), one can factor N with high probability of success.
  - Thus, if two RSA users share the same N, they can figure out each other's secret key (d value).
  - So, do not use a common N.
  - Also, if your d is compromised, do not just change e and d. You should also change N.

If d is known, we can factor N: (may skip)

- $x^2 \equiv 1 \bmod N$ has four solutions:

  $\pm 1,\ \pm a$ for some $a \neq \pm 1$.

- If $a^2 \equiv 1 \bmod N$ and $a \neq \pm 1$

  $\Rightarrow\ a^2 - 1 \equiv 0 \bmod N$

  $\Rightarrow\ N \mid (a+1)(a-1)$

  $\Rightarrow\ \gcd(n, a \pm 1)$ yield the factors of N.

- Factor N by looking for a nontrivial square root of 1 mod N (i.e., an $a \neq \pm 1$ such that $a^2 \equiv 1 \bmod N$).

- For all $w \in \mathbb{Z}_N^*$, $w^{ed-1} \equiv 1 \mod N$.
- Write $ed-1 = r2^s$, where r is odd. (So, $w^{r2^s} \equiv 1 \mod N$)
- Pick any $w \in \mathbb{Z}_N^*$. (What if $w \in \mathbb{Z}_N \setminus \mathbb{Z}_N^*$?)
- Compute $w^r, w^{r2}, w^{r2^2}, w^{r2^3}, \ldots, w^{r2^{t-1}}, w^{r2^t}, \ldots, w^{r2^s}$

  until we find the first $w^{r2^t} \equiv 1 \mod N$ for some t.
- If $t \neq 0$, let $a = w^{r2^{t-1}} \mod N$. Then $a^2 \equiv 1 \mod N$, and $a \neq 1$.
- If $a \neq N-1$, then a is a nontrivial square root of $1 \mod N$.
- Otherwise (i.e., $t = 0$ or $a = N-1$), try another w.

- Low encryption exponent attack
  - A message m sent to e users who employ the same encryption exponent e is not protected by RSA.
  - Say, $e = 3,$ and Bob sends a message m to three recipients encrypted as:

    $$c_1 = m^3 \bmod n_1, \;\; c_2 = m^3 \bmod n_2, \;\; c_3 = m^3 \bmod n_3.$$

  - Eve intercepts the three ciphertexts, and recovers m:
    - $m^3 \equiv c_1 \bmod n_1, \;\; m^3 \equiv c_2 \bmod n_2, \;\; m^3 \equiv c_3 \bmod n_3.$
    - By CRT, $m^3 \equiv c \; \bmod n_1 n_2 n_3$ for some $c < n_1 n_2 n_3.$
    - Also, $m^3 < n_1 n_2 n_3.$ So, $m^3 = c,$ and $m = \sqrt[3]{c}.$

- Wiener's low decryption exponent attack:

Recall RSA decryption: $m := c^d \bmod N$.
One may be tempted to use a small d to speed up decryption. Unfortunately, that may be risky.

The decryption exponent d may be computed from (N, e) if $d < N^{1/4}/3$ and $p < q < 2p$.

(Before Wiener's attack, the condition $p < q < 2p$ usually held because p, q were usually chosen to have the same number of bits.)

- Continued fraction :

$$q_1 + \cfrac{1}{q_2 + \cfrac{1}{q_3 + \cdots + \cfrac{1}{q_m}}} = [q_1, q_2, ..., q_m]$$

- Any (positive) rational number a/b can be expressed as a continued fraction, called its continued fraction expansion.
- Convergents of $[q_1, q_2, ..., q_m]$: $[q_1]$, $[q_1, q_2]$, $[q_1, q_2, q_3]$, $[q_1, q_2, ..., q_m]$. (This sequence converges to $[q_1, q_2, ..., q_m]$.)

- Example: $\dfrac{34}{99} = 0 + \cfrac{1}{2 + \cfrac{1}{1 + \cfrac{1}{10 + \cfrac{1}{3}}}} = [0, 2, 1, 10, 3]$

- Obtained from Euclidean algorithm:

  $34 = 0 \times 99 + 34, \quad 99 = 2 \times 34 + 31, \quad 34 = 1 \times 31 + 3,$

  $31 = 10 \times 3 + 1, \quad 3 = 3 \times 1$

- Convergents of $[0, 2, 1, 10, 3]$:

  $[0],\ [0, 2],\ [0, 2, 1],\ [0, 2, 1, 10],\ [0, 2, 1, 10, 3]$

- Theorem. If $\left|\frac{a}{b}-\frac{c}{d}\right|<\frac{1}{2d^2}$, where $\gcd(c,d)=1$, then c/d equals one of the convergents of the continued fraction expansion of a/b.
- For RSA, $ed=t\varphi(N)+1$ for some t. So, $\frac{e}{N}\approx\frac{e}{\varphi(N)}\approx\frac{t}{d}$.
- If $d<N^{1/4}/3$ and $p<q<2p$, then $\left|\frac{e}{N}-\frac{t}{d}\right|<\frac{1}{2d^2}$.
- So, t/d equals one of the convergents of e/N. Check the convergents one by one to find the right one.

- Small message space attack:
  - If the message space is small. The adversary can encrypt all messages and compare them with the intercepted ciphertext.
  - This attack is not specific to RSA.

Timing Attacks

- Paul Kocher in mid-1990's demonstrated that a snooper can determine a private key by keeping track of how long a computer takes to decrypt messages.
- RSA decryption: $c^d \bmod N$.
- Countermeasures:
  - Use constant decryption time
  - Add a random delay to decryption time
  - Blinding: modify the ciphertext c to c' and compute $(c')^d \bmod N$.

Blinding in Some of RSA Products

- RSA encryption has a homomorphism property:

$$\mathrm{RSA}(m \cdot r) = \mathrm{RSA}(m) \cdot \mathrm{RSA}(r).$$

- To decrypt a ciphertext $c_m = \mathrm{RSA}(m)$:
  - Generate a random message r.
  - Encrypt r as $c_r = \mathrm{RSA}(r)$.
  - Multiply the two ciphertexts: $c = c_m c_r = \mathrm{RSA}(mr)$.
  - Decrypting c yields a value equal to mr.
  - Multiplying that value by r^{-1} yields m.
- Note: all calculations are done in $\mathbb{Z}_N^*$ (i.e., modulo N).

A chosen-ciphertext attack

- Based on RSA's homomorphism property:

$$\mathrm{RSA}(m \cdot r) = \mathrm{RSA}(m) \cdot \mathrm{RSA}(r)$$

- Assume Eve has acess to a decryption oracle.
- The attack:
  - Given $c_m := \mathrm{RSA}(m),$ Eve wants to know $m = ?$
  - She computes $c_r := \mathrm{RSA}(r)$ for an arbitrary $r \in \mathbb{Z}_N^*$.
  - Now, presenting $c_{m \cdot r} = \mathrm{RSA}(m \cdot r) = c_m c_r$

    to the Oracle, Eve obtains $m \cdot r,$ from which she

    can compute $m = (m \cdot r) \cdot r^{-1}.$

Padded RSA

Security of RSA

- We have seen many attacks on RSA.
- Also, RSA is deterministic and, therefore, not CPA-secure.
- We wish to make RSA secure against CPA and aforementioned attacks.
- The RSA we have described so far is called:
  - RSA primitive, plain RSA, or textbook RSA

Padded RSA

- Encryption: $E_{pk}(m) = \text{RSA}(r \,||\, m) = (r \,||\, m)^e \bmod N$, where r is a random string.
- Thus, Padded-RSA$(m) = \text{RSA}(r \,||\, m)$ for some random r.
- Secure against many of aforementioned attacks.
- Theorem (informal): Under some assumption, Padded RSA is CPA-secure if $|m| = O(\log n)$, where $n = \|N\|$.
- Padded RSA was adopted in PKCS #1 v.1.5.

Padded RSA as in PKCS #1 v.1.5

- PKCS: Public Key Cryptography Standard.
- Let (N, e, d) give a pair of RSA keys.
- Let $k = \|N\|$ be the length of N in bytes (e.g., $k = 216$).
- To encrypt a message m:
  - pad m so that $m' = 00 \,||\, 02 \,||\, r \,||\, 00 \,||\, m$ (k bytes)
  - where $r = 8$ or more random bytes $\neq 00$.
  - original message m must be $\leq k - 11$ bytes.
  - the ciphertext is $c := \mathrm{RSA}(m') = (m')^e \bmod N$.
- In 1998, Bleichenbacher published a chosen-ciphertext attack on this padded RSA.

Bleichenbacher's chosen-ciphertext attack

- A (padded) message is called PKCS conforming if it has the specified format:

  00 || 02 || padding string || 00 || original message.
- PKCS #1 implementations usually send you (sender) an error message if $\mathrm{RSA}^{-1}(c)$ is not PKCS conforming.
- It is just like you have an Oracle which, given c, answers whether or not $\mathrm{RSA}^{-1}(c)$ is PKCS conforming.
- Bleichenbacher's attack takes advange of such an Oracle.

- Given $c = \text{RSA}(m)$, Eve tries to find m.

  (Assume m is PKCS conforming.)
- How can the Oracle help?
  - Recall that RSA is homomorphic:

    $$\text{RSA}(m \cdot s) = \text{RSA}(m) \cdot \text{RSA}(s) \quad (\text{computed in } \mathbb{Z}_N^*)$$
  - Given RSA(m), Eve can compute RSA($m \cdot s$) for many $s \in \mathbb{Z}_N^*$.
  - She then asks the Oracle,

    $$\text{Is } ms \in Z_N^* \text{ PKCS conforming?}$$

    (That is, is $ms \bmod N$ PKCS conforming?)
- Why is this information useful?

- Recall PKCS format (k bytes):

  $$00 \,||\, 02 \,||\, \text{padding string} \,||\, 00 \,||\, \text{original message.}$$

- Let $B = 00 \,||\, 01 \,||\, (00)^{k-2}$ (hexadecimal) $= 2^{8(k-2)}$ (binary)

- Then $2B = 00 \,||\, 02 \,||\, (00)^{k-2}$ and $3B = 00 \,||\, 03 \,||\, (00)^{k-2}$

- If m is PKCS conforming $\Rightarrow$ $2B < m < 3B.$

- If $ms \bmod N$ is also PKCS conforming

  $\Rightarrow$ $2B < ms \bmod N < 3B$

  $\Rightarrow$ $2B + tN < ms < 3B + tN$ for some t

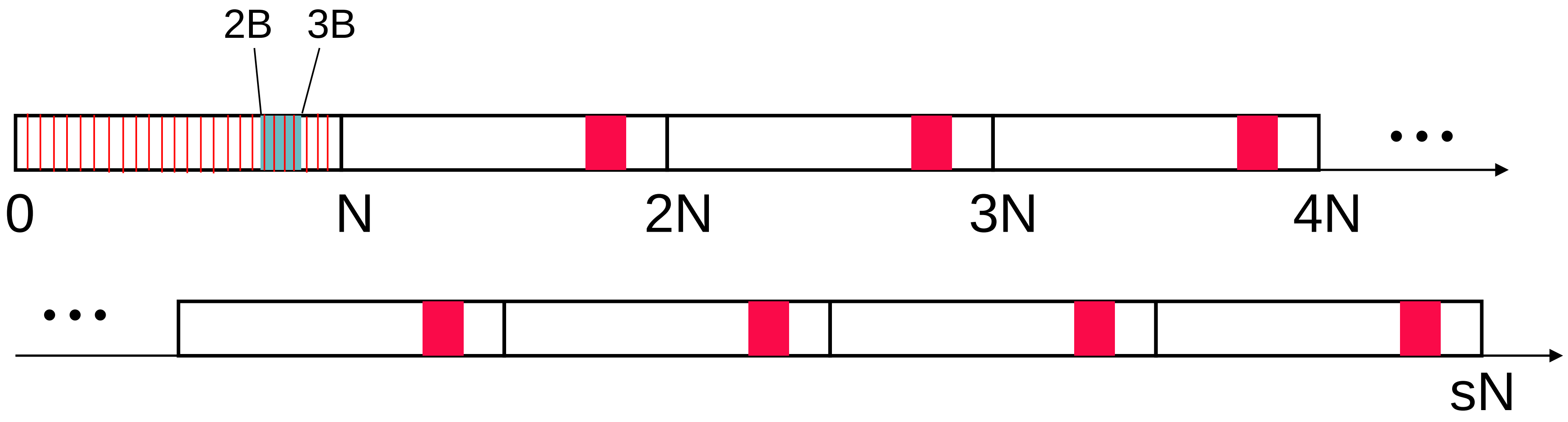
  $\Rightarrow$ $(2B + tN)/s < m < (3B + tN)/s$

- If m is PKCS conforming $\Rightarrow$ m is in the blue area.
- If $ms \bmod N$ is also PKCS conforming

  $\Rightarrow$ $ms \bmod N$ is in the blue area

  $\Rightarrow$ ms is in the red areas

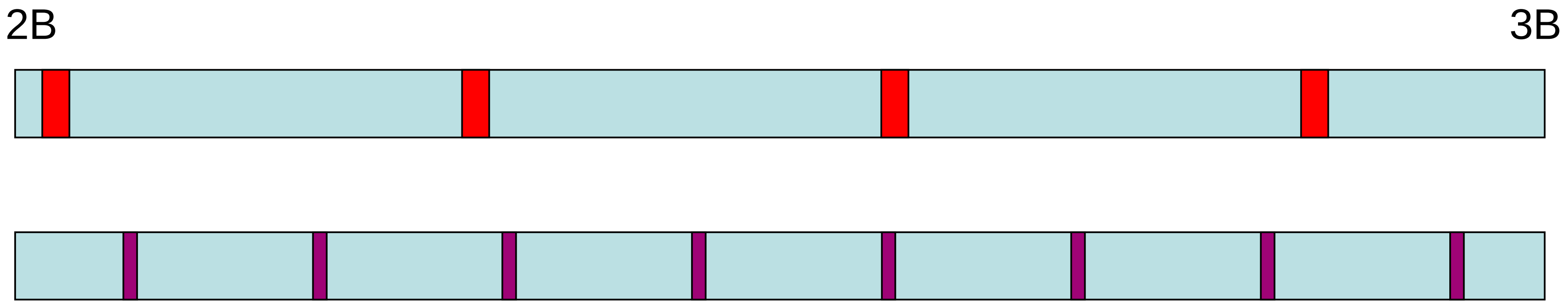
  $\Rightarrow$ m is in the red lines.
- Thus, m is in the red lines of the blue area.



- Let's focus on the blue area, (2B, 3B).
- If m is PKCS conforming $\Rightarrow$ m is in the blue area.
- If $ms \bmod N$ is also PKCS conforming

  $\Rightarrow m$ is in red areas/lines
- If $ms' \bmod N$ is also PKCS conforming

  $\Rightarrow m$ is in purple areas/lines
- So, $m \in \text{blue} \cap \text{red} \cap \text{purple}$



- So, starting with the fact that m is PKCS conforming, Eve finds a sequence of integers $s_1, s_2, s_3,$ ... such that

  $2s_{i-1} \leq s_i$ and

  $ms_i \bmod N$ is PKCS conforming.
- To find s_i, randomly choose an $s \geq 2s_{i-1}$, and ask the oracle whether $ms \bmod N$ is PKCS conforming. If not, then try a different s.
- This way, Eve can repeatedly narrow down the area containing m and eventually find m.
- For N having 1024 bits, it takes roughly 1 million accesses to the oracle in order to find $s_1, s_2, s_3,$ ...

CCA-Secure RSA in the Random Oracle Model

Protecting Every Bit

- There are CCAs that only require the oracle to reveal partial information about the plaintext such as:
  - whether the plaintext is PKCS conforming
  - whether the plaintext is even or odd
  - whether the plaintext $x \in \mathbb{Z}_N^*$ is in the first half or the second half of $\mathbb{Z}_N$ (i.e., $x < N/2$ or $x \geq N/2$?)
- It is desired to protect every bit (or any partial information) of the plaintext.

OAEP: basic idea

- Message padding: not simply $m \| r$ or $r \| m$, but $m \oplus r \| r$, where r is a random string.
- As such, however, there is a 50% overhead. So, we wish to use a shorter bit string r.
- Besides, r should be protected, too.
- This leads to a scheme called Optimal Asymmetric Encryption Padding (OAEP). It can be applied to RSA and other trapdoor functions.

OAEP

- Choose $k,\ l\ (k \ll l)$ s.t. $k + l = \|N\|$ $(N,$ RSA modulus).
- $G:\{0,1\}^k \to \{0,1\}^l$, a pseudorandom generator.
- $h:\{0,1\}^l \to \{0,1\}^k$, a hash function.

- Encryption. To encrypt a block m of l bits:

  1. choose a random bit string $r \in \{0,1\}^k$.
  2. encode m as $x := m \oplus G(r) \,||\, r \oplus h(m \oplus G(r))$

     (if $x \notin Z_N$, the message space of RSA, return to step 1).
  3. compute the ciphertext $y := Enc_{pk}(x)$.

- Decryption: $x := Dec_{sk}(y) = a \,||\, b.$ $\quad m = a \oplus G\big(b \oplus h(a)\big).$

Remarks on OAEP

- OAEP is adopted in current version of RSA PKCS #1 (v. 2.1).
- It is a padding/encoding scheme.
- Intuitively, with OAEP, the ciphertext would not reveal any information about the plaintext if RSA is one-way and h and G are truely random (random oracles).
- A slightly more complicated version of OAEP, in which

$$x = (m0^{k'} \oplus G(r) \,||\, r \oplus h(m0^{k'} \oplus G(r))),$$

has been proved CCA-secure in the random oracle model (i.e., if G, h are random oracles.)

- In practice, hash functions such as SHA-1 are used for G, h.

Generating large primes

To set up an RSA cryptosystem,

we need two large primes p and q.

How many prime numbers are there?

- Infinitely many.
- First proved by Euclid:
  - Assume only a finite number of primes $p_1,\ p_2,\ \ldots,\ p_n$.
  - Let $M = p_1 p_2 \ldots p_n + 1$.
  - M is not a prime, because $M \neq p_i,\ 1 \leq i \leq n$.
  - So, M is composite and has a prime factor p_i for some i

    $\Rightarrow\ p_i \mid M\ \Rightarrow\ p_i \mid 1\ \Rightarrow\Leftarrow$ contradiction.

Distribution of Prime Numbers

The Prime Number Theorem:

Let $\pi(x)$ denote the number of primes $\leq x$. Then

$$\pi(x) \approx \frac{x}{\ln x} \text{ for large } x.$$

Bertrand's Theorem: For any $n > 1$, the fraction of n-bit integers that are prime is at least $1/3n$.

How to generate a large prime number?

- Generate a random odd number N of desired length.
- Test if N is prime.
- If not, discard it and try a different number.
- Q: How many numbers are expected to be tested before a prime is found?

Primality test: Is N a prime?

- Can it be solved in polynomial time?
- A long standing open problem until 2002.
- AKS(Agrawal, Kayal, Saxena): $O\left(\left(\log N\right)^{12+\varepsilon}\right)$.
  - Later improved by others to $O\left(\left(\log N\right)^{10.5}\right)$, and then to $O\left(\left(\log N\right)^{6+\varepsilon}\right)$.
- In practice, Miller-Rabin's probabilistic algorithm is still the most popular --- much faster, $O\left(\left(\log N\right)^{3}\right)$.

Miller-Rabin primality test: Is N a prime?

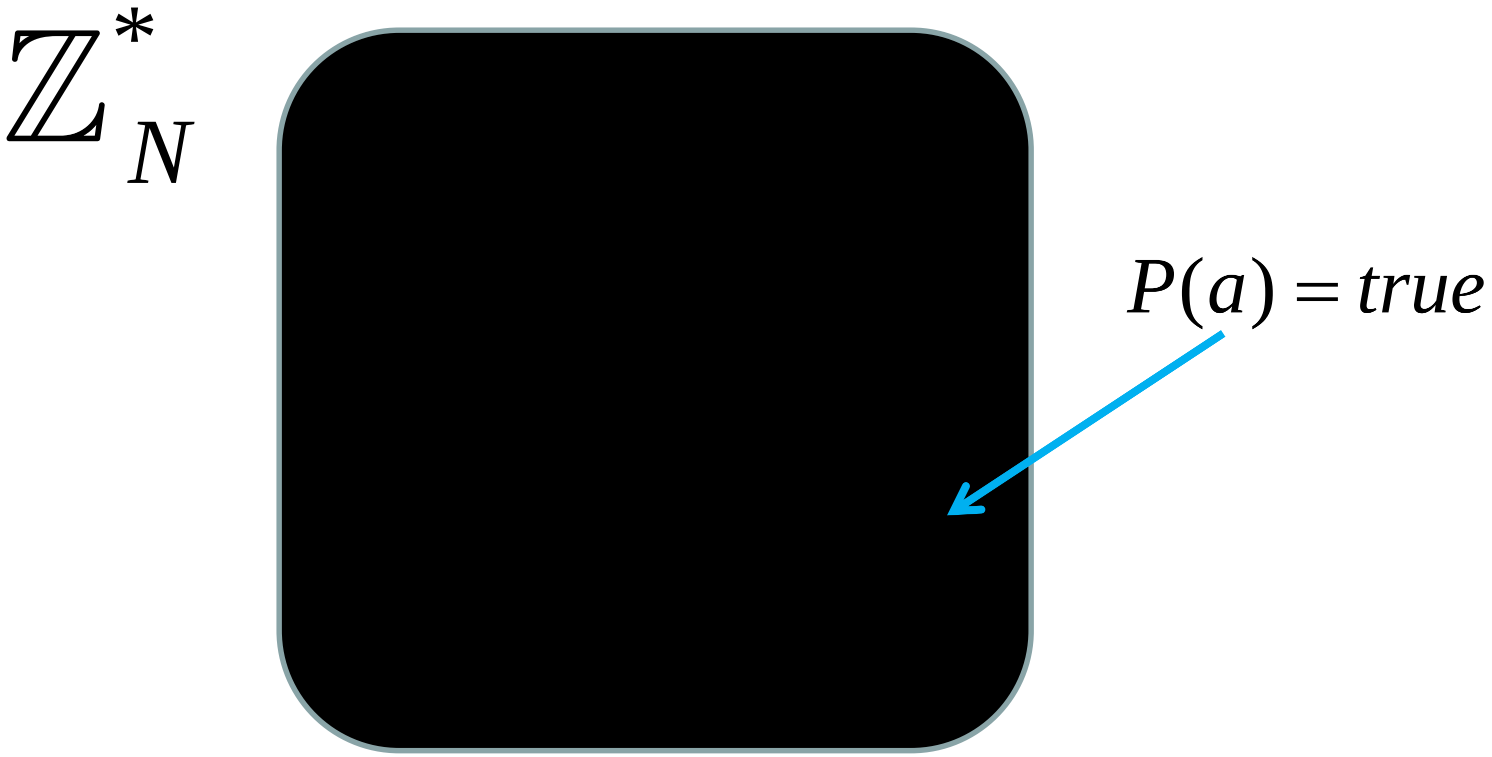
- Looking for a characteristic property of prime numbers:
  - N is prime $\Leftrightarrow$ what?
  - N is prime $\Leftrightarrow$ $\forall a \in \mathbb{Z}_N^*,\ P(a) = true$
  - N is prime $\Rightarrow$ $\forall a \in \mathbb{Z}_N^*,\ P(a) = true$

    N not prime $\Rightarrow$ $\exists \geq k$ elements $a \in \mathbb{Z}_N^*,\ P(a) = false$
- Algorithm: Check $P(a)$ for t random elements $a \in \mathbb{Z}_N^*$.

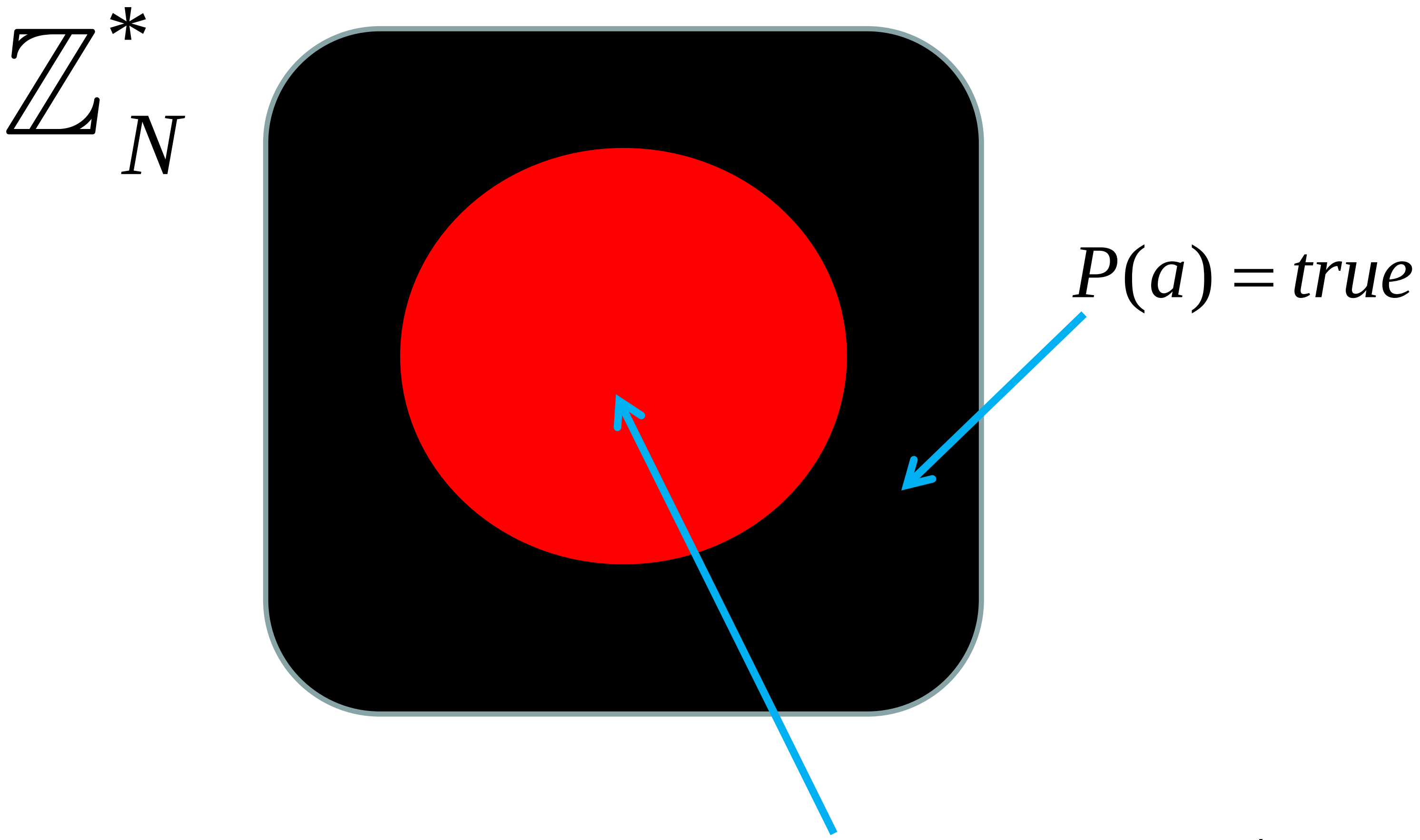
  If $P(a)$ is true for all of them then return prime

  else return composite.
- A "prime" answer may be incorrect with probability

  $\leq \left(1 - k/\varphi(N)\right)^t$



If N is prime, then for all $a \in \mathbb{Z}_N^*$, $P(a)$ is true.



If N is not prime, then there are elements $a \in \mathbb{Z}_N^*$, called strong witnesses, s.t. $P(a) = false$.

- Looking for $P(a)$:
  - How about $P(a) = \left[a^{N-1} \equiv 1 \bmod N \right]$?
  - Fermat's little theorem:

    If N is prime $\Rightarrow$ $\forall a \in \mathbb{Z}_N^*,\ a^{N-1} \equiv 1 \bmod N.$
  - If N is not prime $\Rightarrow$ possible that $\forall a \in \mathbb{Z}_N^*,\ a^{N-1} \equiv 1 \bmod N.$

    (Carmichael numbers: composite numbers N for which

    $a^{N-1} \equiv 1 \bmod N\ \forall a \in \mathbb{Z}_N^*$.)
  - Need to strengthen the condition $a^{N-1} \equiv 1 \bmod N$.

- Fact: if $N \neq 2$ is prime, then 1 has exactly two square roots in $\mathbb{Z}_N^*$, namely ± 1.
- Write $N - 1 = u2^k$, where u is odd.
- If N is prime

$$\Rightarrow \forall a \in \mathbb{Z}_N^*, \ a^{u2^k} \equiv 1 \bmod N \qquad \text{(Fermat's little theorem)}$$

$$\Rightarrow \forall a \in \mathbb{Z}_N^*, \ P(a) = true, \text{ where}$$

$$P(a) = \begin{cases} a^u \equiv 1 \bmod N \text{ or} \\ a^{u2^i} \equiv -1 \bmod N \text{ for some } i, \ 0 \leq i \leq k-1 \end{cases}$$

- Why? Consider the sequence

$$a^u, \ a^{u2}, \ a^{u2^2}, \ \ldots, \ a^{u2^{k-1}}, \ a^{u2^k} \equiv 1$$

Example: $N = 41$

- $N-1 = 40 = 2^3 \cdot 5 = 2^k u.\ \ (k = 3,\ u = 5)$
- For $a = 2,\ (a^5,\ a^{10},\ a^{20},\ a^{40}) \equiv (32,\ -1,\ 1,\ 1) \mod 41$

  $P(2) = true.$
- For every $a \in \mathbb{Z}_{41}^*,\ P(a)$ is true.

Example: $N = 25$

- $N-1 = 24 = 2^3 \cdot 3 = 2^k u.\ \ (k = 3,\ u = 3)$
- For $a = 2,\ a^u = 2^3 \equiv 8 \mod 25$

  $\left(a^3,\ a^6,\ a^{12}\ a^{24}\right) \equiv \left(8, 14, 21, 16\right) \mod 25$

  $P(2) = false.$

- If N not prime $\Rightarrow$ strong witnesses always exist ?
- Loosely speaking, yes:
  - If N is an odd composite and not a prime power, then at least one half of the elements $a \in \mathbb{Z}_N^*$ are strong witnesses.
  - For such an N, we may check $P(a)$ for t random elements $a \in \mathbb{Z}_N^*$. If $P(a)$ is true for all of them then return prime else return composite.
  - A "prime" answer may be incorrect with probability $\leq 2^{-t}$.

- A composite number N is a prime power if $N = p^e$ for some prime p and integer $e \geq 2$. (It is a perfect power if $N = k^e$ for some integer k and $e \geq 2$.)
- Theorem: If N is an odd composite and not a prime power, then at least one half of the elements $a \in \mathbb{Z}_N^*$ are strong witnesses that N is not prime.
- Idea of Proof: The set B of all *non*-strong witnesses forms a proper subgroup of $\mathbb{Z}_N^*$. So, $\mathrm{ord}(B) < \mathrm{ord}(\mathbb{Z}_N^*)$ and $\mathrm{ord}(B) \mid \mathrm{ord}(\mathbb{Z}_N^*)$. So, $\mathrm{ord}(B) \leq \frac{1}{2}\mathrm{ord}(\mathbb{Z}_N^*)$.

Algorithm: Miller-Rabin primality test

- Input: integer $N > 2$ and parameter t
- Output: a decision as to whether N is prime or composite

```
1.  if $N$ is even, return "composite"
2.  if $N$ is a perfect power, return "composite"
3.  for $i := 1$ to $t$ do
        choose a random integer $a \in \mathbb{Z}_N$
        if $\gcd(a, N) \neq 1$, return "composite"
        if $a$ is a strong witness,  return "composite"
4.  return ("prime")
```

Analysis: Miller-Rabin primality test

- If the algorithm answers "composite", it is always correct.
- If the algorithm answers "prime", it may be incorrect with probability at most 2^{-t}.
- Actually, at most 4^{-t}, by a more sophisticated analysis.

Monte Carlo algorithms

- A Monte Carlo algorithm is a probabilistic algorithm
  - which always gives an answer
  - but sometimes the answer may be incorrect.
- A Monte Carlo algorithm for a decision problem is yes-biased if its “yes” answer is always correct but a “no” answer may be incorrect with some error probability.
- A t-iteration Miller-Rabin is a “composite”-biased Monte Carlo algorithm with error probability at most $1/4^t$.

Las Vegas algorithms

- A Las Vegas algorithm is a probabilistic algorithm
  - which may sometimes fail to give an answer
  - but never gives an incorrect one
- A Las Vegas algorithm can be converted into a Monte Carlo algorithm.