Logistics

Carsten Baum

Building 322, room 210

Exercises as usual

Next Monday: Homework (sheet is online), i.e. no lecture

Homework is due on 20.03

Public Key Cryptography

And the RSA cryptosystem

Schedule for today

Math recap

1. Modular arithmetic
2. Gcd and the (extended) Euclidean algorithm
3. Coprimality and Euler’s totient function
4. Multiplicative inverses and how to compute them
5. Lagrange’s Theorem

Actual cryptography

1. What is Public Key Encryption?
2. Defining Security of PKE
3. The RSA cryptosystem
4. Why RSA decryption works

Modular arithmetic

Let N be a positive integer, called **modulus**

If we divide a by N over the integers, then $a = b + kN$ where $0 \leq b < N$ is unique

We call b the remainder of the division

Two integers a, b are called **congruent** if $N|(b - a)$ and we write $a = b\ (mod\ N)$

Since $0 = N\ (mod\ N), 1 = N + 1\ (mod\ N)$ every integer is equal to $0, \dots, N - 1$ modulo N

We write the remainders as $Z_N = \{0, \dots, N - 1\}$

Examples

Modular arithmetic $mod\ 11$

$24 = 2 + 2 \cdot 11$, so $24 = 2\ mod\ 11$ -> $11|(24 - 2)$

Any integer when divided by 11 must be a unique number between 0 and 10

$$Z_{11} = \{0,1,2,3,4,5,6,7,8,9,10\}$$

Examples continued

Modular arithmetic $mod\ 11$

$24 = 2\ mod\ 11$

$24 + 3 = 27 = 5\ mod\ 11, 2 + 3 = 5\ mod\ 11$
$24 \cdot 3 = 72 = 6\ mod\ 11, 2 \cdot 3 = 6\ mod\ 11$

If we add (or multiply) $mod\ 11$ it does not matter if we start from 24 or 2.

Rules of modular arithmetic

1. If $a = x \, mod \, N$ and $b = y \mod N$ then
    1. $a + b = x + y \, mod \, N$
    2. $a \cdot b = x \cdot y \, mod \, N$
2. Associativity:
   $(a + b) + c = a + (b + c) \, mod \, N$ and $(a \cdot b) \cdot c = a \cdot (b \cdot c)$
3. Commutativity:
   $a + b = b + a \, mod \, N$ and $a \cdot b = b \cdot a \, mod \, N$

Rules of modular arithmetic

4. Identity elements:
   $a + 0 = a \; mod \; N$ and $a \cdot 1 = a \; mod \; N$
5. Distributivity:
   $(a + b) \cdot c = a \cdot c + b \cdot c \; mod \; N$
6. $a + (N - a) = 0 \; mod \; N$

Computing the Greatest Common Divisor

$\gcd(x, y)$: largest positive integer d such that $d|x$ and $d|y$

Example: $\gcd(12,8) = 4$, $\gcd(9,3) = 3$, $\gcd(11,12) = 1$

If $\gcd(x, y) = 1$ then we say x, y are coprime

An algorithm to compute the gcd: the Euclidean algorithm

How the Euclidean algorithm works

To compute $\gcd(x, y)$:

1. Define $r_0 = x, r_1 = y$
2. Iteratively in round $i \in \{1, \ldots\}$
   1. Divide r_{i-1} by r_i, obtaining remainder r_{i+1} (i.e. $r_{i+1} = r_{i-1} \, mod \, r_i$)
   2. If $r_{i+1} = 0$ output r_i

The algorithm always terminates, because $r_{i+1} < r_i$ but division remainder never < 0

The output is correct, because $\gcd(r_0, r_1) = \gcd(r_1, r_2) = \cdots = \gcd(r_{i-1}, r_i)$

Computing the gcd - example

gcd(12,8):

1. $4 = 12 - 8 \cdot 1$
2. $0 = 8 - 4 \cdot 2$

-> $\gcd(12,8) = 4$

gcd(12,7) :

1. $5 = 12 - 7 \cdot 1$
2. $2 = 7 - 5 \cdot 1$
3. $1 = 5 - 2 \cdot 2$
4. $0 = 2 - 1 \cdot 2$

Extended Euclidean Algorithm

In addition to Euclidean Algorithm, keep track of linear combinations

$\gcd(12,7):$

1. $5 = 1 \cdot 12 - 7 \cdot 1$
2. $2 = 7 - 5 \cdot 1 = 7 - (1 \cdot 12 - 7 \cdot 1) \cdot 1 = -1 \cdot 12 + 7 \cdot 2$
3. $1 = 5 - 2 \cdot 2$

   $= (1 \cdot 12 - 7 \cdot 1) - (-1 \cdot 12 + 7 \cdot 2) \cdot 2$

   $= 3 \cdot 12 - 7 \cdot 5$
4. $0 = 2 - 1 \cdot 2$

Formal Extended Euclidean Algorithm

$egcd(a, b)$:

1. $s \leftarrow 0, s' \leftarrow 1, t \leftarrow 1, t' \leftarrow 0, r \leftarrow b, r' \leftarrow a$
2. While $r \neq 0$
   1. $q = \lfloor \frac{r'}{r} \rfloor$
   2. $(r', r) \leftarrow (r, r' - q \cdot r)$
   3. $(s', s) \leftarrow (s, s' - q \cdot s)$
   4. $(t', t) \leftarrow (t, t' - q \cdot t)$
3. $d \leftarrow r', x \leftarrow t, y \leftarrow s$
4. Output d, x, y such that $d = \gcd(a, b) = x \cdot a + y \cdot b$

Same as in
Euclidean Algorithm

Coprimality and Euler's Totient function

a is coprime to N iff $\gcd(a, N) = 1$

<u>Euler's totient function $\varphi(N)$</u>

How many numbers $1 \leq a < N$ fulfill $\gcd(a, N) = 1$

Given $N = p_1^{e_1} \dots p_n^{e_n}$ prime factorization it is known that
$\varphi(N) = p_1^{e_1 - 1}(p_1 - 1) \cdots p_n^{e_n - 1}(p_n - 1)$

If $N = p \cdot q$ then $\varphi(N) = (p - 1) \cdot (q - 1)$

Coprimality and Euler's totient function

Let $N = 6$ then which numbers are coprime?

$\gcd(1,6) = 1$

$\gcd(2,6) = 2$

$\gcd(3,6) = 3$

$\gcd(4,6) = 2$

$\gcd(5,6) = 1$

But we need to know factorization of N

Faster: $N = 6 = 2 \cdot 3$ so $\varphi(N) = (2-1) \cdot (3-1) = 2$

How does coprimality help us?

Consider we want to solve $a \cdot x = 1\ mod\ N$ by finding x

Does a solution exist?

If $a \cdot x = 1\ mod\ N$ then $a \cdot x + k \cdot N = 1$:
If $gcd(a, N) = 1$ then $egcd(a, N)$ can compute x

Also: if $\gcd(a, N) = 1$ then x is unique

Example

We know that $\gcd(12,7) = 1$ so we can solve $12 \cdot x = 1 \; mod \; 7$

$\gcd(12,7):$

1. $5 = 12 - 7 \cdot 1$
2. $2 = 7 - 5 \cdot 1 = 7 - (12 - 7 \cdot 1) \cdot 1 = -12 + 7 \cdot 2$
3. $1 = 5 - 2 \cdot 2 = (12 - 7 \cdot 1) - (-12 + 7 \cdot 2) \cdot 2 = 3 \cdot 12 - 7 \cdot 5$
4. $0 = 2 - 1 \cdot 2$

Correct! $3 \cdot 12 = 36 = 1 + 5 \cdot 7 = 1 \; mod \; 7$

Lagrange’s Theorem

For positive integer N define $Z_N^* = \{x \in Z_N \mid gcd(x, N) = 1\}$
Euler Totient function: $|Z_N^*| = \varphi(N)$

Then for all $x \in Z_N^*$: $x^{\varphi(N)} = 1 \, mod \, N$

Meaning: $x^{\varphi(N)-1}$ is the inverse for x mod N for every coprime number!

Example

$N = 12 = 3 \cdot 2^2$

$\varphi(N) = 2 \cdot 2 = 4$

Coprime numbers: $\mathrm{Z}_N^* = \{1,5,7,11\}$

$$1^4 = 1 \cdot 1^3 = 1 \; mod \; 12$$
$$5^4 = 5 \cdot 125 = 5 \cdot (10 \cdot 12 + 5) = 25 = 2 \cdot 12 + 1 = 1 \; mod \; 12$$

$$7^4 = 7 \cdot 343 = 7 \cdot (28 \cdot 12 + 7) = 49 = 4 \cdot 12 + 1 \; mod \; 12$$

Public Key Encryption

Disadvantages of Symmetric Cryptography

- The **chicken-and-egg** problem
  - You need a shared key k to establish a secure channel
  - You need a secure channel to share the key

- **Scalability** problems
  - A network of n users needs $n(n-1)/2$ exchanged keys
    - $O(n^2)$ for n nodes
  - Collaborative networks (e.g. sensor networks) may use a single network-wide key
    - If one node gets compromised, whole network get compromised

Public key (asymmetric) encryption

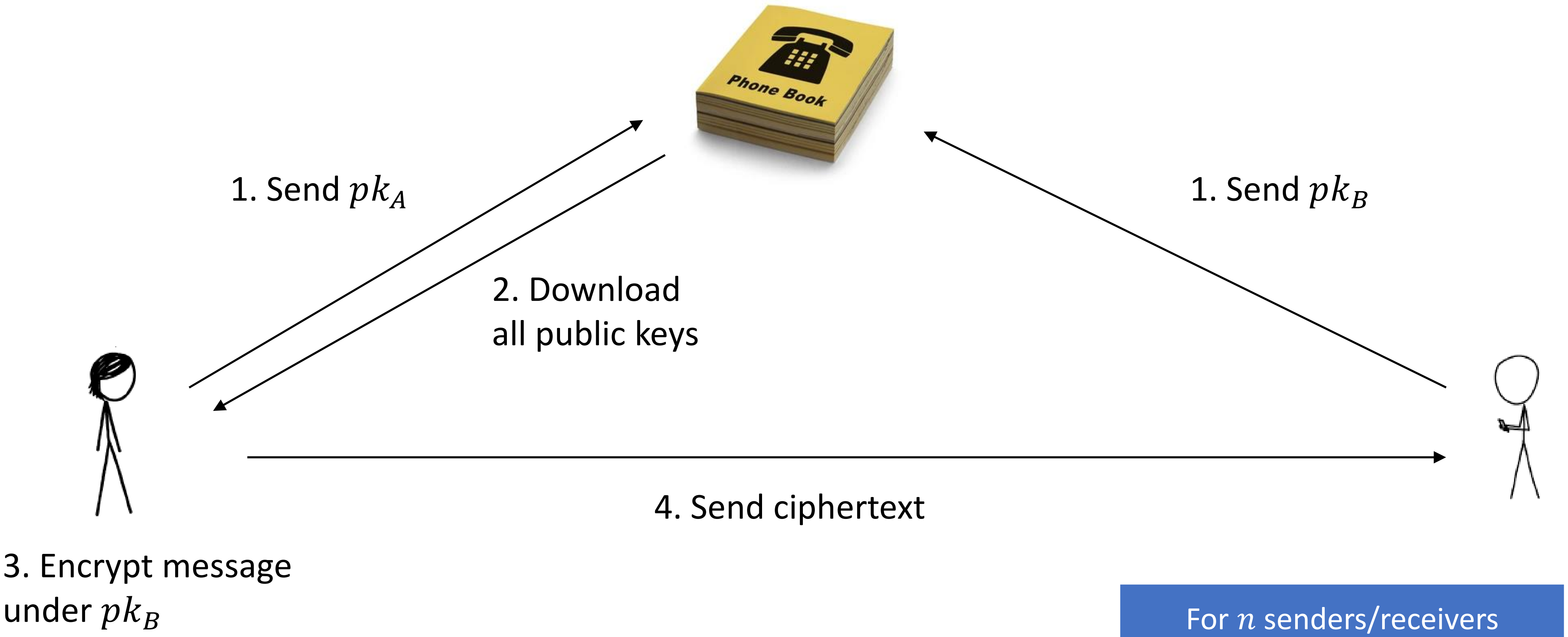
Involves two separate but mathematically related keys **per user**

- One **private** and one **public**
- Given public key, it is hard to compute private key

<u>Confidentiality</u>

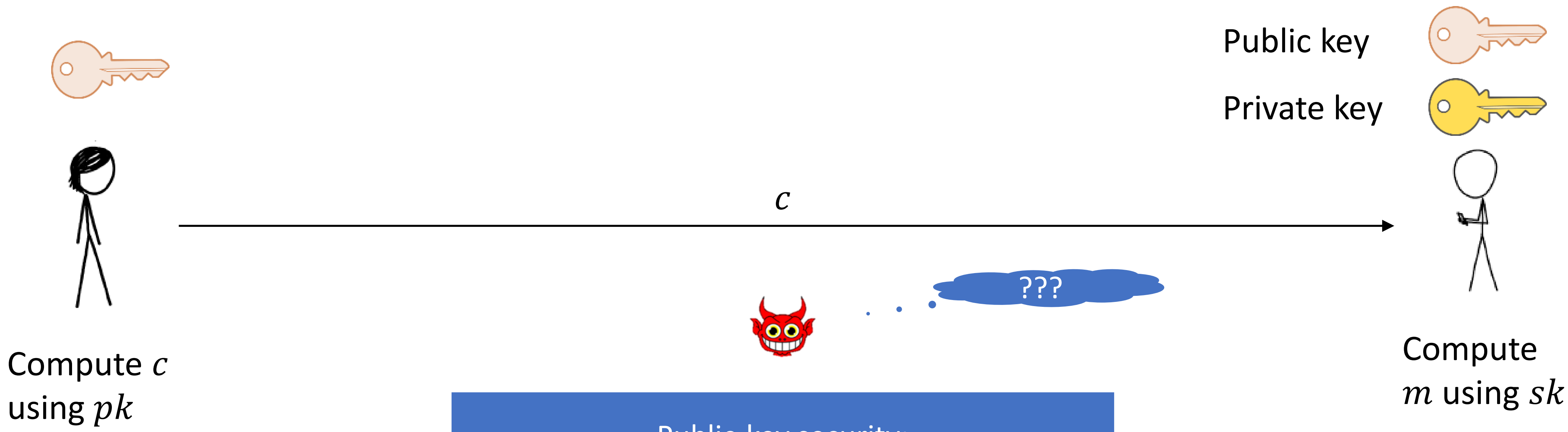
- The sender encrypts the message with the **public key** of the receiver
- Only receiver can decrypt it using private key

How to use Public key encryption



For n senders/receivers
we need n keys

Public key encryption

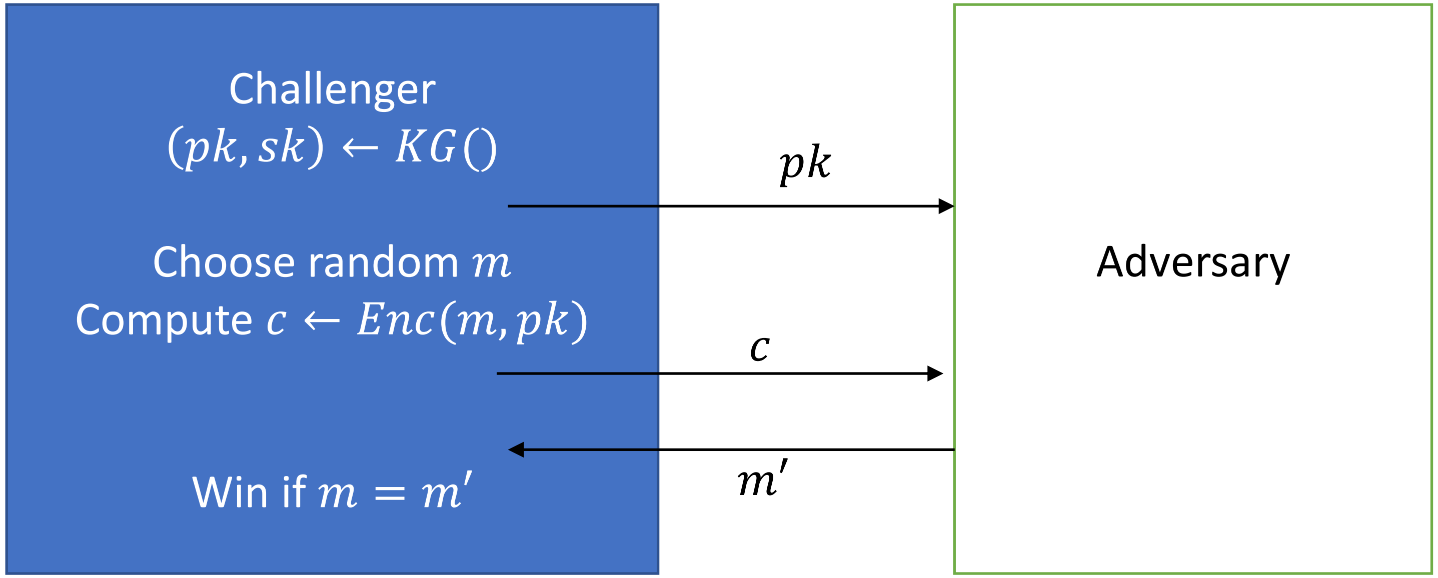


Public-key security:

No adversary with pk should be able to have any information about message in c

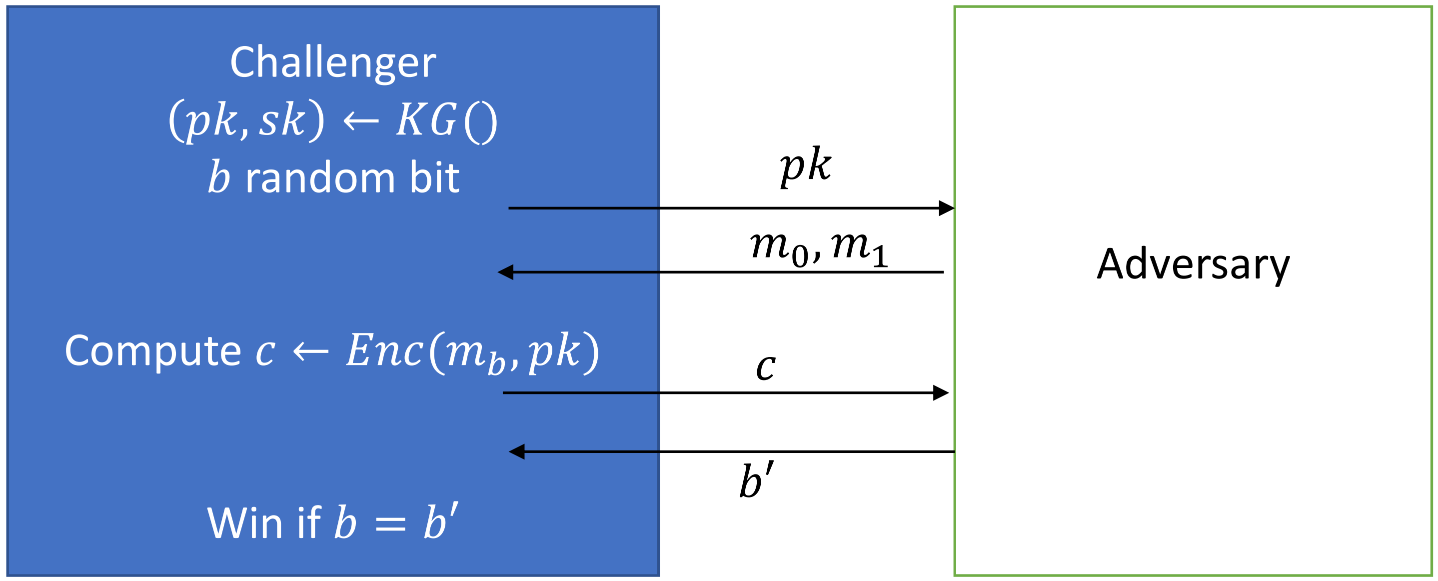
Crucial: Adversary can encrypt using pk!

Defining security of PKE

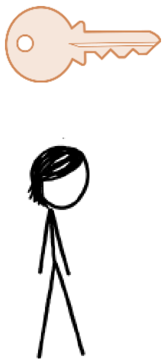


Problems with this definition?

Defining security of PKE properly – IND-CPA



Building Public key encryption



Compute c using pk

c

1. Given sk it can be easy to find pk, but not the reverse!

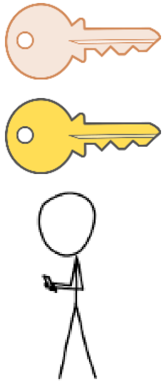
2. We need a mathematical problem that is simple using sk but hard using pk.

sk serves as a mathematical ``trapdoor''.

Example correspondence:
pk large number, sk factorization
What is difficult to compute without factorization: $\varphi(\cdot)$

Public key

Private key



Compute m using sk

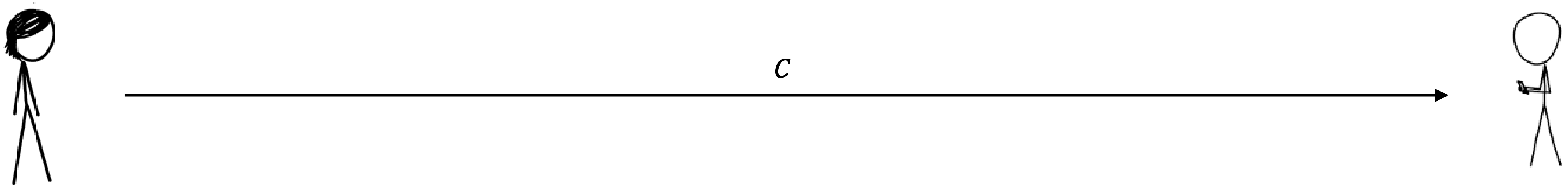
The RSA cryptosystem

Turing Award in 2002

Invented 1977 by Rivest, Shamir & Adleman

<u>Key Generation</u>

1. Find two large primes p, q and e with $\gcd\big(e, (p-1)\cdot(q-1)\big) = 1$
2. Compute $N = p \cdot q$
3. Find d such that $d \cdot e = 1\ mod\ (p-1)(q-1)$



Compute $c = m^e\ mod\ N$

Compute $m = c^d\ mod\ N$

RSA – an example

Key Generation

1. Find two large primes $p = 13, q = 17$ and $e = 5$. $\gcd(5{,}12 \cdot 16) = 1$
2. Compute $N = p \cdot q = 221$
3. Find d such that $d \cdot e = 1\ mod\ (p-1)(q-1)$
   Solve $5d = 1\ mod\ 192$ -> $d = 77$
4. $pk = (N, e) = (221{,}5), sk = (d) = (77)$

Encrypt:

1. Message $m \in Z_N^*$. Set $m = 17$.
2. Set $c \leftarrow m^e\ mod\ N = 153$

Decrypt:

1. Ciphertext $c \in Z_N^*$
2. Set $m' = c^d\ mod\ N = 153^{77} mod\ 221 = 17$

Why it works

- $N = p \cdot q, \varphi(N) = (p-1) \cdot (q-1)$
- We choose $\gcd\big(e, (p-1) \cdot (q-1)\big) = 1$
  so $d = e^{-1}\ mod\ \varphi(N)$ always exists

- Lagrange's Theorem: $\forall x \in Z_N^*: x^{\varphi(N)} = 1\ mod\ N$

- $Dec(Enc(x, pk), sk) = (x^e)^d\ mod\ N = x^{1+k\varphi(N)} = x\ mod\ N$

Summary

We looked at the mathematics necessary for RSA
Modular arithmetic, computing modular inverses, Lagrange's theorem

Public Key encryption
1. What is Public Key Encryption?
2. Defining Security of PKE

The RSA cryptosystem
1. The RSA cryptosystem
2. Why RSA decryption works