Add \((x, y)\)

\[ x \\
 y \]

2 integers \(\rightarrow\) decimal
\[ \downarrow \]
convert \(x, y\) to binary

What is the max value
\[ x \cdot y \leq N \]

\[ N = \mathcal{O}(2^n) \]
\[ n = \log N \]

\[ 2^x \]

Double \(\exp(n)\)

\[ \text{or } m \text{ bits} \]
\[ m = \log n \]

\[ \exp(x, y) : x^y \]
\[ x, y \text{ are values } \leq N \]
\[ x \text{ takes } m \text{ bits} \]
\[ y \text{ takes } n \text{ bits} \]
\[ m = \log x \]
\[ n = \log y \]

Analysis based on the "size" of the inputs vs

"value"
RSA: Rivest - Shamir - Adleman

Alice → Bob

↑

Eve

X is some message (X is a "huge" number)

1) Encoding (x)

Alice: \[ y = x^e \]

She sends \( y \) to Bob

2) Decoding (y)

\[ x = y^d = (x^e)^d \]

Eve: \[ y \]

\( e \) is a public key

\[ y^d = 1 \]

Bob:

- Generate a pair of numbers
- Public key \( e \)
- Private key \( d \)

RSA: \( y, e \) are public
RSA: \(y, e\) are public

It is hard to "crack" \(x\)

figure out \(d\) given \(y, d, e\)

RSA: details

Bob: select 2 large prime numbers \(p\) and \(q\)

\[N = p \cdot q\]

(\(N\) is very hard to factorize into the correct \(p \cdot q\))

\(e:\) public key

choose some integer \(e\)

(there is no known efficient also for factorization)

that is relatively prime to \((p-1)(q-1)\)

\(\gcd(e, (p-1)(q-1)) = 1\)

d: private key

\(d\) is \(e\) mod \((p-1)(q-1)\)

\[d \cdot e = 1 \text{ mod } (p-1)(q-1)\]

1. Bob publishes \(N\) and \(e\)
2. Alice wants to send \(x\)

\[\text{Encoding: } y = x^e \text{ mod } N\]

send \(y\) to Bob

3. Bob:

\[\text{decode } y^d \text{ mod } N\]

\[= (x^e)^d \text{ mod } N\]

Eve:

\(N, e, y\)

encrypted message

\(\) largest \(x\) to have been cracked

lecture7 Page 3
\[
\left( x^e \right)^d \equiv x \pmod{N}
\]

We know that \( d e \equiv 1 \pmod{(p-1)(q-1)} \).

Show that
\[
x^e \equiv x \pmod{N}, \quad x^e - x \equiv 0 \pmod{N}
\]

we know that
\[
de \equiv 1 \pmod{(p-1)(q-1)}
\]

\[
\Rightarrow \quad d = k(p-1)(q-1) + 1
\]

\[
\Rightarrow \quad k \cdot (p-1)(q-1) + 1
\]

in divisible by \( p \cdot q \)

\[
\Rightarrow \quad 1 \) show that \( x, x \pmod{N} \)
\]

\[
\Rightarrow \quad \text{in divisible by } p
\]
1) Show that \( k(p-1)(q-1) \) is divisible by \( p \).

2) Since \( p \) and \( q \) are prime,

\( \Rightarrow \) \( x \cdot k(p-1)(q-1) - x \) is divisible by \( p \).

Show:

\( x \cdot x^{k(p-1)(q-1)} - x \) is divisible by \( p \).

We know that \( x^{p-1} \equiv 1 \pmod{p} \leftarrow \text{Fermat's little theorem} \).

Whether

\( x \cdot x^{k(p-1)(q-1)} - x \equiv 0 \pmod{p} \)

\( x \cdot (x^{p-1})^{k(q-1)} - x \pmod{p} \)

\( x \cdot (1)^k - x \equiv 0 \pmod{p} \)

\( x - x \equiv 0 \pmod{p} \).

2) \( x^{q-1} \equiv 1 \pmod{q} \).

Second: knowing \( p, q, e \), we have to find the \( d \) such as \( pdq + 1 \).

RSPA Example
Bob: \[ p = 5, \ q = 11 \]
\[ N = p.q = 55 \]
\[ \phi = 3 \times 10 = 30 \]
\[ e = 3 \rightarrow \ d = 3^{-1} \mod 30 = 13 \mod 30 = 27 \]

Alice: \( x = 13 \)
\[ y = 13^3 \mod 55 = 169 \times 13 \mod 55 \]
\[ = y \times 13 \mod 55 \]
\[ = 52 \mod 55 \]
\[ y = 5^2 \]

Bob: \[ y^d \mod 55 \]
\[ (52)^{27} \mod 55 \]
\[ = 52 \times (52^2)^{13} \mod 55 \]
\[ = 52 \times (27^2)^{13} \]
\[ = 52 \times (9)^{13} \]
\[ = 13 \]

---

**Diffie-Hellman Key Exchange**

Alice ↔ Bob

1) Exchange same message
2) Both will compute the same "shared" key \( \Box \)
   (This key will never have been exchanged)
3) Use \( d \) for a one-time message encoding & decoding
3) Use $d$ for a one-time message encoding & decoding 
   (one time pad)

Details

1) Alice and Bob both agree on a prime $P$
   and some $2 < g < P - 2$
   
   $P$ & $g$ are public

2) Alice chooses a number $a$, $1 < a < P - 1$
   
   $g$ is secret
   
   Bob chooses a number $b$, $1 < b < P - 1$
   
   $b$ is secret

3) Alice computes
   
   $A = (g^a) \mod P$

   Bob computes
   
   $B = g^b \mod P$

   Alice sends $A$ to Bob
   
   Bob sends $B + A$

4) Alice: \[ (B^b) \mod P = (g^b)^a \mod P = (g^{ab} \mod P)^b = d \]

   Bob: \[ A^b \mod P = (g^a)^b \mod P = (g^{ab} \mod P)^b = d \]

   Alice & Bob now have their own shared secret $d$

5) $[\text{key}]$ is known only to Alice & Bob

X
Alice: Encoding: $y = x \oplus d$, compute the XOR of $x$ and $d$

Alice sends $y + b$

Bob: Decode: $y \oplus d = x$, XOR of $y$ and $d$

1) $p = 23$, $2 \leq g \leq p - 2$, $g = 14$

public

2) Alice: $a = 3$

Bob: $b = 4$

3) Alice $A = g^a \mod 23$

$= 14^3 \mod 23$

$= 7 \mod 23$

Bob $B = g^b \mod 23 = 14^b \mod 23 = 6 \mod 23$

4) Alice: $B^a = 6^3 \mod 23 = 216 \mod 23 = 9$

Bob: $A^b = 7^4 \mod 23 = 2401 \mod 23 = 9$

5) $d = 9$ is shared secret.

$d = 00001001 \leftarrow 9$ in 8 bits

Alice needs to send $x = 10101010$

Encoding: $x \oplus d = 10100011 = y$
Encoding: \[ x \oplus d = 10100011 = y \]

Alice sends \( y \) to Bob.

Bob: decode
\[
\begin{align*}
00001001 & \leftarrow d \\
10100011 & \leftarrow y \\
\hline
x & \rightarrow 10101010 \quad d \oplus y \quad \text{xor}
\end{align*}
\]

\[ A = g^x \mod p \]

we cannot figure \( x \) easily.

\[ \log_g A = x \mod p \]

Diffie-Hellman rests on hardness of discrete log problem.