Rem:nd

Bob —m→ Alice; $m \xrightarrow{e_k} c \xrightarrow{d_k} m$; Eve sees c w/o k

Impt
* Eve cannot know or guess key k
* Eve cannot recover m from c w/o k.

Warmup Symmetric Ciphers

Idea
$\mathcal{K}$ = Set of keys
$\mathcal{M}$ = Set of messages
$\mathcal{C}$ = Set of ciphertexts

Encryption
$e: \mathcal{K} \times \mathcal{M} \longrightarrow \mathcal{C}$
$e(k,m) = c$
$e_k(m) = c$

Decryption
$d: \mathcal{K} \times \mathcal{C} \longrightarrow \mathcal{M}$
$d(k,c) = m$
$d_k(c) = m$

Compatibility
$d(k, e(k,m)) = m$
$d_k(e_k(m)) = m$

Rmk $\Rightarrow e_k: \mathcal{M} \longrightarrow \mathcal{C}$ is one-to-one.
PS/ $m, m' \in \mathcal{M}$
$e_k(m) = e_k(m')$
$d_k(e_k(m)) = d_k(e_k(m'))$
$m = m'$ ⊗

Kerchoffs Principle
Assume the attacker knows your encryption scheme.
$(\mathcal{K}, \mathcal{M}, \mathcal{C}, d, e)$

security should depend on secrecy of key (an elt $k \in \mathcal{K}$)

Def^n A cipher is the data $(\mathcal{K}, \mathcal{M}, \mathcal{C}, e, d)$

Principle
1) e should be easy to compute. (& key k, $m \in \mathcal{M}$, easy to find $e(k,m)$)
2) d should be easy to compute.
3) Longtime Listener Attack
Given $c_1, \dots, c_n \in \mathcal{C}$ hard to compute $d_k(c_1), \dots, d_k(c_n)$ w/o knowing k. & hard to find k
4) Known Plaintext Attack
Know pairs $(m_1, c_1), \dots, (m_n, c_n)$ w/ $e_k(m_i) = c_i$
Hard to decrypt general ciphertext w/o k.
Hard to find k.

5) Chosen Plaintext Attack
Choose $m_1, \dots, m_n$ & get $c_i = e_k(m_i)$.
still hard to decrypt gen. c & hard to find k.

Eg Substitution fails
3 - probability attack
4 - Get most of table
5 - e_k (unless $\dots z$) get whole table

Easy ≈ All computer in seconds
Hard ≈ Takes supercomputer hundreds of years.

Encoding Schemes/Blocks

Example
$\{a, b, c, \dots, z\} \leftrightarrow \mathbb{Z}/26\mathbb{Z}$
$a \leftrightarrow 0$
$b \leftrightarrow 1$
$\vdots$
$z \leftrightarrow 25$
$\mathcal{M} = \mathbb{Z}/26\mathbb{Z} = \mathcal{C}$

$\mathcal{K} = \mathbb{Z}/26\mathbb{Z}$
$e: \mathcal{K} \times \mathcal{M} \longrightarrow \mathcal{C}$
$e(k,m) = k+m \mod 26$
$d: \mathcal{K} \times \mathcal{C} \longrightarrow \mathcal{M}$
$d(k,c) = c-k \mod 26$
$d_k(e_k(m)) = m+k-k \equiv m \mod 26$

Idea
Message Blocks $\longleftrightarrow$ Set of #s $(\mathbb{Z}/N\mathbb{Z}) = \mathcal{M}$ $\xrightarrow{e_k}$ $\mathcal{C}$

Standard Way (Take)
Encode a character as a sequence of 8 bits
Can 0 or 1.
$2^8 = 256$ choices
each choice $\leftrightarrow 0 \le n \le 255$

Ex a = 00000000 → 0
b = 00000001 → 1
c = 00000010 → 2
$\vdots$
z = 00011001 → 25

write bcd =
00000001 00000010 00000011
= 00000001 00000000 00000000 00000000 ← $b \times 2^{16}$, 1×2^{16}
+ 00000000 00000010 00000000 ← $c \times 2^8$, 2×2^8
+ 00000000 00000011 ← 1×2^0, 3×2^0
bcd = $1 \times 2^{16} + 2 \times 2^8 + 3 \times 2^0$

To 3 letter sequence
$\Rightarrow$ n w/ 24 bits
i.e. $0 \le n \le 2^{24} - 1 = 16777216$

So e.g.
$\mathcal{M} = \mathbb{Z}/17000000\mathbb{Z}$
↑ encode all 3 character sequences in here.

In general going to have blocks $\mathcal{B}$ bits long
$\mathcal{M} = \{m_{n-1} m_{n-2} \cdots m_1 m_0 \mid m_i = 0,1\}$
$\{0, 1, \dots, 2^B - 1\}$

Message $\mathcal{B}$ byte (character 8 bits)
$m = m_{B-1} m_{B-2} \cdots m_1 m_0 \quad 0 \le m \le 256$
$\Updownarrow$
$m_0 + m_1 \cdot 2^8 + m_2 \cdot 2^{16} + \dots + m_{B-1} \cdot 2^{(B-1)\cdot 8}$
$\in [0, 2^{8B})$

HW Implement
words $\longleftrightarrow$ Message of $\mathcal{B}$ bytes

ASCII
Chars $\xrightarrow{ord}$ Byte $\xrightarrow{chr}$ Chars
ord('A') = 01000001 = 17
ord('A') = 01000001 = 65
chr(17) = 'A'.

w/ this figured out this becomes a sending secret numbers problem.
$\mathcal{M} = \{n \in \mathbb{Z} \mid 0 \le m < 2^{8m}\}$
$\mathcal{C} = \{c \in \mathbb{Z} \mid 0 \le c < 2^{8c}\}$
$\mathcal{K} = \{k \in \mathbb{Z} \mid 0 \le k < 2^{8k}\}$

Question: Size of $\mathcal{B}$?
* $\mathcal{B}_k$ too small Eve can try each k. (knows d)
$c \stackrel{d_k(c)}{\longrightarrow}$ each k until decrypted).
* Big enough to be safe
So exhaustive search was 2^{80} in 2012.
2^{128}? 2020
2^{128} in meet in middle attack.

Examples
① Find $\mathbb{Z}/p\mathbb{Z})^* = \mathcal{C} = \mathcal{K}$
$= \{1, 2, \dots, p-1\}$
↑ message size 160 bits s.t.
$e_k(m) = k \cdot m \mod p$
$d_k(c) = k^{-1} \cdot c \mod p$.
Bob & Alice share k.
Alice computes k^{-1}....
Extended Euc. $< 2\log_2 p + 2$ steps

1) e easy.
2) d easy.
3) Eve try each k one @ a time. Hard.
4) Eve knows (m, c)
$\Rightarrow c \equiv k \cdot m \mod p$
in $\log_2 p$ steps she computes $m^{-1} \mod p$.
She finds $k \equiv m^{-1} c \mod p$
Done! Decrypt everything

Example $N = 2^{320}$
$\mathcal{M} = \mathcal{C} = \mathcal{K} = \mathbb{Z}/N\mathbb{Z}$
$e_k(m) = m+k \mod N$
$d_k(c) = c-k \mod N$
"Mixing going on that avoids statistical attack"
1) ✓ 2) ✓ 3) ✓
4) Attack succeeds
(m, c) know $k \equiv c-m \mod N$!

Ex $\mathcal{M}, \mathcal{C}, \mathcal{K} = \mathbb{Z}$.
$e_k(m) = k \cdot m$
$d_k(c) = c/k$.
To decrypt Eve has to factor large #s.
Eve sees $c_1, c_1 \cdot c_2, \dots, c_{n-1} \cdot c_n$
$\gcd(c_1, \dots, c_n) = \gcd(km_1, \dots, km_n)$
$= k \gcd(m_1, \dots, m_n)$
$\stackrel{\text{high}}{=} 1$
In log time she can find k.
Fails 3)

Asymmetric Ciphers
Can Alice & Bob send encrypted messages v/o ever not being monitored by Eve.

Public Key Crypto

Visual Example
Alice's safe; slot in safe; Bob
Knowing how to encode (drop in slot) doesn't help you decode (open safe).

$\mathcal{K}$ = space of keys
$\mathcal{M}$ = messages
$\mathcal{C}$ = ciphers.
k_{pq} is a pair (k_{pub}, k_{priv})
$e(k_{pub}, m) = c$
$e_{k_{pub}}(m) = c$
$d(k_{priv}, c) = m$
$d_{k_{priv}}(c) = m$
$d_{k_{priv}}(e_{k_{pub}}(m)) = m$

e - put in slot
k_{pub} - which safe ← public info
d - open safe
k_{priv} - Code for safe. ↑ private

Goal
Develop trap door function
Easy to compute
Hard to undo without an extra piece of info