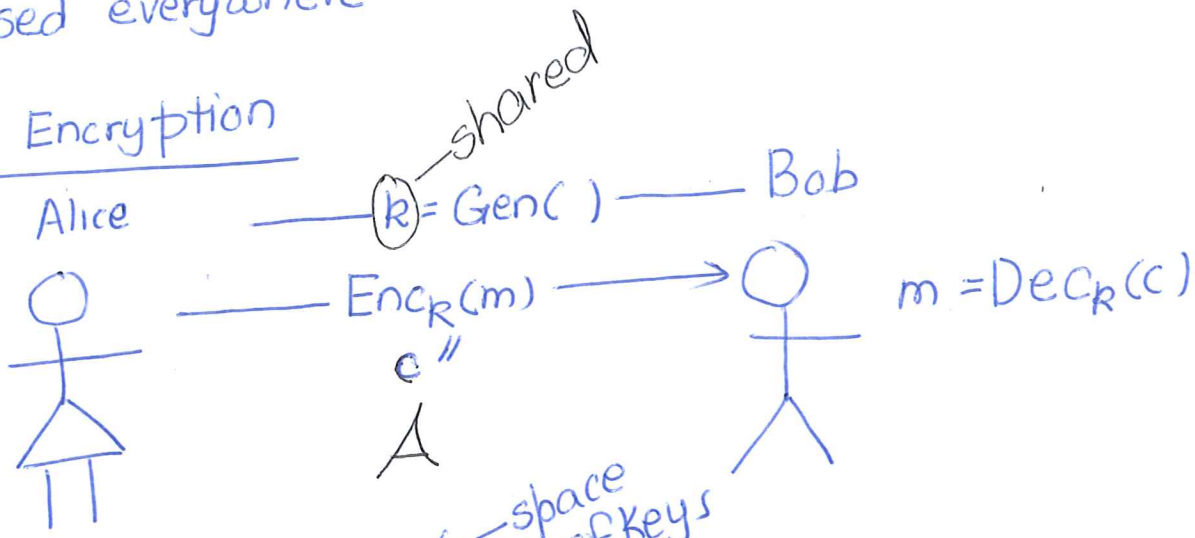
Lecture Let 1/2

Sep 5, 2020

- Classic Cryptography
  - art of writing/solving codes

- Modern Cryptography
  - not art (rigorous science)
  - used everywhere

Private-Key Encryption



space of keys

Gen: generate a key $k \in K$ (key generation)

K = 10 bit strings

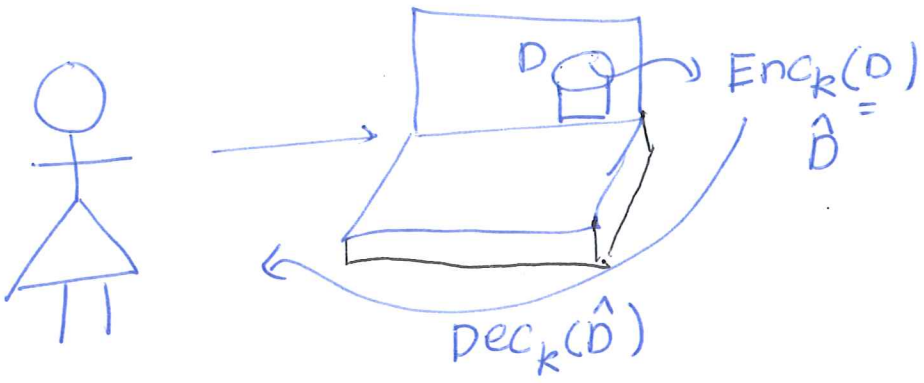
Enc: (encryption)

$c = Enc_k(m)$ $\in M$ ← all sentences in English (space of msgs)

ciphertext, key, msg

Dec (decryption)

$m = Dec_k(c)$ $\in \mathcal{C}$ (space of ciphertexts)



Sanity check

$$Dec_k(Enc_k(m)) = m$$

m — 0xxy...

I love 435

World I

Assume adversary does not know

k, (Gen, Enc, Dec) ← algs.

↑ key

Security by Obscurity ✗

Kerckhoff's principle

← encryption scheme

The cipher method must not be required to be secret, and it must be able to fall in the hands of the enemy/adversary.

Adversary knows: (Gen, Enc, Dec) ← A knows

But not k (key)

Historical ciphers

Why study them?

Shift cipher

m: crypto

$a=0, b=1, \ldots, Z=25$

Gen: $k \in \{0, \ldots, 25\}$ ← choose uniformly

Enc:

$m = m_1 \ldots m_\ell$ ← letters in m

$Enc_k(m_1 \ldots m_\ell) = c_1 \ldots c_\ell$

$c_i = [(m_i + k) \bmod 26]$ ← 0..25

$k = 3$

check: $Enc_k(m) =$ fubwr ← crypto

Dec

$Dec_k(c_1 \ldots c_\ell) = m_1 \ldots m_\ell$

$m_i = [(c_i - k) \bmod 26]$ ← opposite

sanity check: $Dec_k(Enc_k(m)) = m$ (check)

Dec_k(fubwr) = crypto

$k = 3$

↓ check. [Be careful: $(-3 \bmod 25) = 22$]

a	0
b	1
c	2
d	3
e	4
f	5
g	6
h	7
i	8
j	9
k	10
l	11
m	12
n	13
o	14
p	15
q	16
r	17
s	18
t	19
u	20
v	21
w	22
x	23
y	24
z	25

0..25

Is it secure? shift cipher

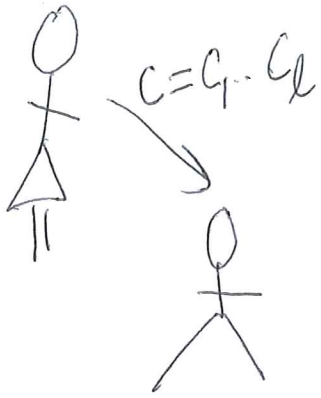
$\mathcal{A}$: adversary

$\mathcal{A}$

knows: (Gen, Enc, Dec) ← Kerckoff's principle
(not key k of course)

Given: $c = c_1 \cdots c_\ell$ (ciphertext)

Goal: m s.t $Enc_k(m) = c$
$\parallel$
$Dec_k(c) = m$



for k in 0 .... 25 do

$m = Dec_k(c)$

Does m look like English?

manually → If IsEnglish(m) = 1 then

then done

exhaustive attack

end for

$\mathcal{A}$ Key space should be large enough to make exhaustive attack infeasible

Next time we will make the key space bigger.

Fri 3-5 pm

Sept 9, 2020

Lecture Let 3

Last time
- private-key
- shift cipher

$|perm(S)| = |S|!$ $S = \{3,4,10\}$ $3!$

$perm(S)$ = all permutations of set S

$|perm(\{0..25\})| = 26!$ large

Monoalphabetic cipher

Gen: $k \leftarrow perm(\{0 \dots 25\})$

k

Enc: $c = Enc_k(m)$ ← plaintext; $m = m_1 \dots m_\ell$

ciphertext → $c = c_1 \dots c_\ell$

$c_i = k(m_i)$

m = crypto

c = UHPSGF

Dec: $m = Dec_k(c)$; $c = c_1 \dots c_\ell$; $m = m_1 \dots m_\ell$

$m_i = k^{-1}(c_i)$ — opposite.

c = UHPSGH

m = crypto

Sanity check: $Dec_k(Enc_k(m)) = m$

obvious ✓

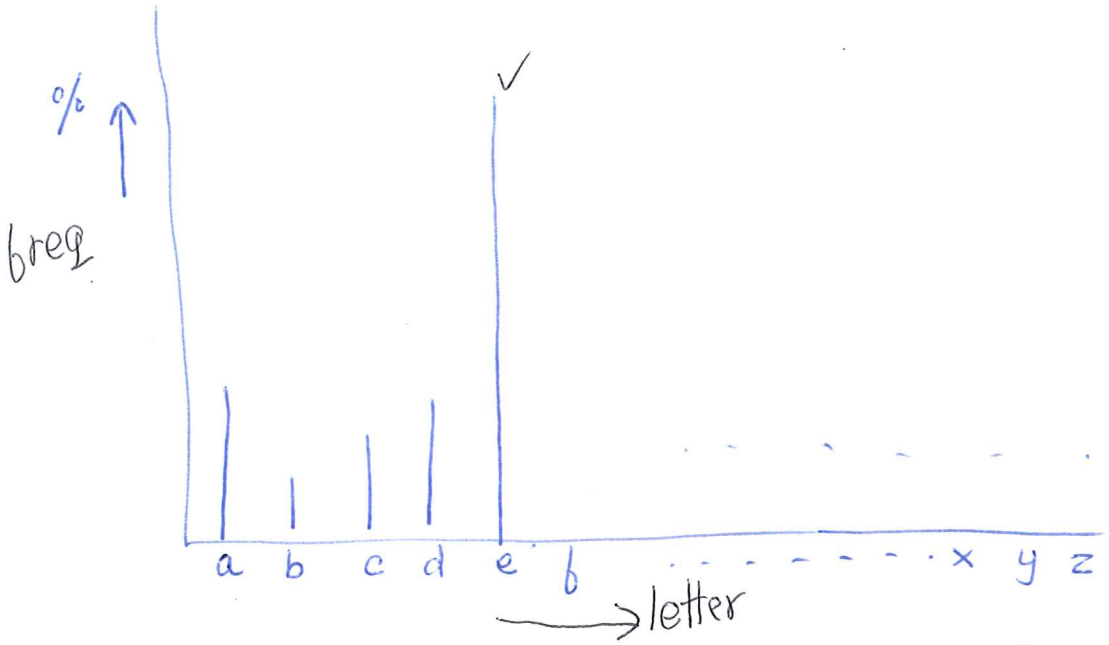
	key
a	X
b	E
c	U
d	A
e	D
f	N
g	B
h	K
i	V
j	M
k	R
l	O
m	C
n	Q
o	F
p	S
q	Y
r	H
s	W
t	G
u	L
v	Z
w	I
x	J
y	P
z	T

Is it secure?

exhaustive attack doesn't work

$26!$

statistical
Frequency analysis



A

knows:

(Gen, Enc, Dec) ← Kerckhoff's principle

$c = c_1 \cdots c_\ell$ ← ciphertext (Encryption of eng. msg)

→ most frequent letter in c say e → D

→ next frequent letter in c say t → L

→ most frequent trigram (3 letters) the → GDKD

GKD

extract the key k

Key idea: freq histogram of c is shifted/permuted histogram for English.

Let us revisit attack on shift cipher

A knows

$c = c_1 \cdots c_\ell$ ← ciphertext

$c_i = ((m_i + k) \bmod 26)$ ← i^{th} letter of plaintext

↑ Secret key (shift)

for k in $0..25$

$m = Dec_k(c)$

↑

g_s English

* (adhoc)

Freq. histogram of eng.

↓

$\langle p_0 \cdots p_{25} \rangle$ ← freq. of z

↑ freq of a

$\sum_{i=0}^{25} p_i^2 \approx 0.065$ (*)

Let us say $m_1 \cdots m_\ell$ ← is English

Let $\langle q_0, \cdots, q_{25} \rangle$ ← freq. vector for c ← ciphertext

↑ # of a's in c

# of z's in c

$\langle q_0, \cdots, q_{25} \rangle$ shifted by $\langle p_0, \cdots, p_{25} \rangle$

(k)

$I_j = \sum_{i=0}^{25} p_i \cdot q_{i+j}$ ← j shift

↑ don't forget mod 26

$I_0 \neq 0.065$

$I_1 \neq 0.065$

⋮

$I_k = 0.065$ *

⋮

$I_{25} \neq 0.065$

Key idea: $\langle q_0, \cdots, q_{25} \rangle$ shifted by k is $\langle p_0, \cdots, p_{25} \rangle$

What was wrong with Monoalphabetic cipher?

a letter got transformed the same way?

a → X
s → W

fixes this

Vigenére cipher

tellhimaboutme ← Plaintext
cafecafecafeca ← key (cafe)
VEQP.... ← ciphertext

Gen: $k = k_0 \cdots k_{t-1}$ [t letters] $t=4$ $k=$ cafe.

$t=4$ cafe
$k_0 k_1 k_2 k_3$

Enc:

$C = Enc_k(m)$ ← plaintext

$m_1 \cdots m_\ell$

$c_1 \cdots c_\ell$ ← ciphertext

what letter of the key (?)

$c_i = [(m_i + k_{(i-1) \bmod t})] \bmod 25$

$1 \le i \le \ell$

$c_1 = [(m_1 + k_0)] \bmod 25$

$c_{t+1} = [(m_{t+1} + k_0)] \bmod 25$

opposite

Dec

$m = Dec_k(c)$

$m_1 \cdots m_\ell$ ← $c_1 \cdots c_\ell$

$m_i = [(c_i - k_{(i-1) \bmod t})] \bmod 25$

$1 \le i \le \ell$

Sanity check: $Dec_k(Enc_k(m)) = m$

Last time

- Monoalphabetic → attack
- Vignére cipher

Lecture Let 4

Sep 10, 2024

Attacking the Vigner'e cipher

A knows the length of the Key / t

$t = 11$

cafe(4)

$k = k_1 \cdots k_t$

ciphertext → $c = c_1 \cdots c_t \; c_{t+1} \cdots c_{2t} \; c_{2t+1} \cdots c_{3t} \; c_{3t+1} \cdots$

key is repeated → $k_1 \cdots k_t \; k_1 \cdots k_t \; k_1 \cdots k_t \; k_1 \cdots$

message/plaintext → $m = m_1 \cdots m_t \; m_{t+1} \cdots m_{2t} \; m_{2t+1} \cdots m_{3t} \; m_{3t+1} \cdots$

Ex: tell him about me
cafe cafe cafe c

shifted by same amount

attack

shift k_1 : $c_1 \; c_{t+1} \; c_{2t+1} \cdots$

shift k_2 : $c_2 \; c_{t+2} \; c_{2t+2} \cdots$

$\vdots$

shift k_t : $c_t \; c_{t+t} \; c_{2t+t} \cdots$

each them behaves like a shift cipher

$t = 4$

Use previous attack to find $k_1, \ldots, k_t$ ✓

A know max of key length $T = 100$

repeat the above attack for $t \in \{1, \ldots, T\}$

$t \in \{1, \ldots, 100\}$

Kasiski's attack

let $t=4$ $k=$ cafe

8 = 2×4

....the.....the.....	← plaintext
...cafe...cafe	← key
..tmi....tmi...	← ciphertext

repeated in ciphertext

- find repeated bigrams/trigrams in ciphertext (3 letters / 2 letters)
- distance between the repeatitions

$d_1, \dots, d_m$ — multiple of key len

$\gcd(d_1, \dots, d_m) = T$ (greatest common divisor)

8, 4 → 4

$t \mid T$ (why?)

Now use previous attack.

Index of coincidence

key len t (but (A) doesn't know) $k = k_1 \cdots k_t$

cipher text ↓

$c_1, c_{1+t}, c_{1+2t} \cdots$ (corresponding to shift by k_1)

k_1 k_1 k_1

freq. hist → $\langle q_0, \dots, q_{25} \rangle$ → #a, #z's

$t=4$ cafe

shifted hist. of english

if t was key size then

$\sum_{i=0}^{25} q_i^2 \simeq 0.065$ ✓

else

$\sum_{i=0}^{25} q_i^2 = \sum_{i=0}^{25} (1/26)^2 \simeq 0.038$ (why?) ✗

end if

$t = 1, 2, 3, \ldots$

stop when $\sum_{i=0}^{25} q_i^2 \approx 0.065$

Kahn

Note that it can be completed automated.

Modern Cryptography

I). Formal definitions
- security goal.
- threat model. (capabilities of A)

II). Precise assumptions

III). Formal proofs.

secure encryption.

gt should be impossible for an attacker to ...

I) - gt should be impossible to recover the key ✗

$Enc_k(m) = m$

- recover the entire plaintext from the ciphertext ✗

SSN#: C reveals 10% of the plaintext m

- to recover any character of the plaintext from ciphertext ✗

$> 100,000$

- ciphertext should leak no additional information about the underlying plaintext ✓

threat model

ciphertext only

attacker

$\mathcal{A}$: ← knows $c_1, \dots, c_k$ ← ciphertexts

goal: $m_1, \dots, m_k$

$m_i = Dec_k(c_i)$

powerful

known plaintext

$\mathcal{A}$: knows (m_1, c_1) ← plaintexts, ← ciphertexts

(m_2, c_2)

(m_k, c_k)

goal: $c \neq c_j$ $(1 \le j \le k)$

find $m = Dec_k(c)$

chosen plaintext

$\mathcal{A}$ can chose these

chosen-ciphertext

$\mathcal{A}$ chose

(m_1', c_1')

(m_j', c_j')

goal: $c \neq c_m'$ $(1 \le m \le j)$

$m = Dec_k(c)$

$\mathcal{A}$'s power gets stronger

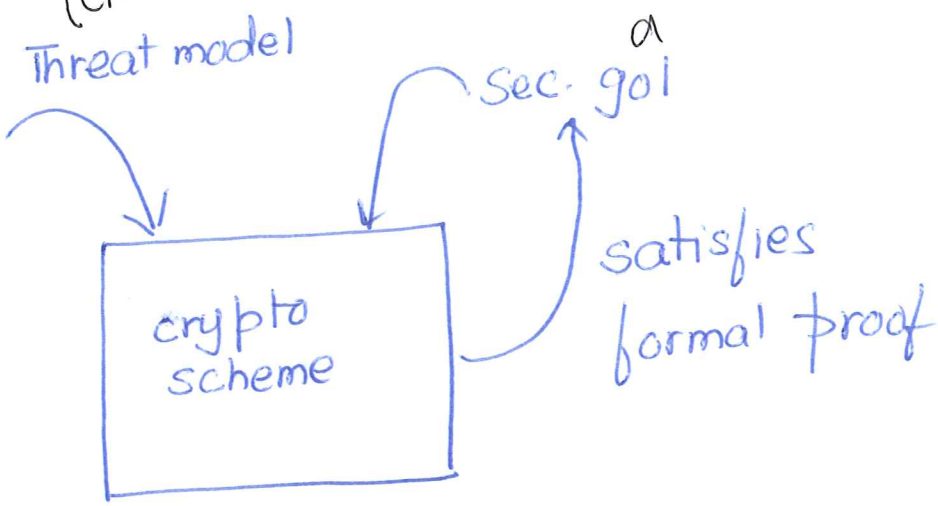
the attack model gets stronger.

Precise assumptions

- my scheme is secure if factoring is hard.

$N = pq$

Rigorous proof



Testing is not enough

3-5pm.