** Introduction

⇒ Symmetric Encryption.

** ① $E(K, m) = C$ * Encryption.

** ② $D(K, C) = m$ * Decryption.

→ Encryption algorithm (publicly known)

⇒ Never use a proprietary ~~[illegible]~~ cipher.

** عشان يكون محمي (لازم نخفي ال Key . لا نخفي كله public.

** 2 - cases.

1. Single use key.   -email.-
2. Multi use key   -encrypted files.
   ↳ need more machinery ~~[illegible]~~ عكس ال Single use Key.

** لازم ناخد بعين الاعتبار كل خطوة لنتأكد من أمان جدول ال cases **

** Cryptography: fantastic for protecting inf.
↳ but it has limitations.

● 1 ليس هو الحل لكل المشاكل الأمنية
↳ not the solution for all security problems.
↳ 1. for example (software bugs).
2. Social engineering attacks (لما ال attacker يحاول يخدع المستخدم)

✓ ** ∴ ال Cryptography مش حل الكل security problems

- [2] Crypt. becomes useless if it's implemented incorrectly
- [3] Crypt. is not something you should try to invent and design. *هو مش اشي يجب علينا اختراعه !!

** ∴ في ال Crypt. ⇐ Standards.

** Crypto core ∴ بنغطي التشفير.

1. Secret key establishment ‏1. انشاء المفتاح السري.
2. Secure communication. ‏2. اتصال آمن

~~these~~

- Encryption schemes provides both confidentiality & integrity. ع كل اشي بس بتوفر السرية والنزاهة.

→ Crypto can do more.

⇒ ① digital signiture. التوقيعات الرقمية.

توقيعي بالوضع العادي (كخربشة على كاغد) كلشي اوكي، لكن بقدرش أعمله من خلال الحاسوب نفس التوقيع (لأنه ممكن ال attacker يعمله cut / copy ويعمله paste !!

** فبنقول ال dig. sign. هو عبارة function of the content being signed.

فلما يجي ال attacker بده ينسخ التوقيع من ملف لملف آخر ما بيقدر / لأنه ال function (الدالة) بالملف الأول غير من الثاني

⇒ (2.) Anonymous communication. (Bidelectional)

يعني زي ما بين اصحاب server (بدون ما ال server يعرف مين انا) اتصال بشكل مجهول.

• الطرف الآخر ما يعرف مين بحكي معه ⇒ exp: Anonymous digital cash

ولكن يستمر تواصله (يرسل ويستقبل). ** ولكن ممكن الطرف الاول (Alice)

يعمل duplicate ال conin (يغش) كيف ممكن نمنعه!!

• لازم نتأكد انه Alice ← spind the coin once. (see later).

•• Protocols.

*- Elections. (الانتخابات)

نستخدم في الانتخابات (election center في) نرسل الصوت عليها (encryption) معمول له. وهاد ال election center يقدر يحسب ويطلع الفائز. لبدون ما ناخذ الأصوات (ال inputs)

*- Private auctions. (المزادات)

** Secure multi-party computation. تأمين حساب متعدد الأطراف

زي ال election center ← هو زي function ما نعرف شو

ال inputs (as individual)، ولكن نعرف النتيجة ✓. يعني ما نعرفش مثلاً صوتك

كل هذا ممكن بس ال (trusted authority) ممكن واحد منهم ما يكون محل ثقة!!

** Crypto magic.

(1) • privately outsourcing computation الاستعانة بجهات خارجية خفية كـ

•• مثلاً Alice بده يستخدم query محرك جوجل ويبحث بشكل encryption

E(query) ⇐ وجوجل يقدر يتعامل معه عادي بدون ما يعرف ال plaintext (يعني ال plaintext الأصلي)، ويعمل E(result) ويرجعها لـ Alice و [illegible]

وبشوف ال result

② • Zero Knowledge (proof of knowledge).

• Alice بعرف رقم N $N = \boxed{p \cdot q}$ ، بعرف ال factorization اله.

• دي Bob نفسو بعرف N مبقدر يعمل prove لـ Alice بالرغم إنه بجربش

غير N وما بعرف ال factors ← (p 4 q).

** بالاستيان أكثر مادة **

• Crypto is a rigorous science (علم صارم).

** 3 steps:-

تقسيم

- ① Precisely specify exp. (what the attacker do to attack a digital sig. تحديد التهديد بدقة
  threat model وشو هدفه!
- ② Propose a construction. اقتراح بناء.
- ③ Prove that breaking ↘ under threat model
  will solve an underlying hard problem سيحل مشكلة صعبة

** History.

• Symmetric ciphers

Alice , Bob بدهم يتواصلوا :. (shared secret key بينهم)

ال attacker ما بعرف هاد ال secret key. // Alice, Bob (نفس ال key)

↓

Symmetric.

- Exp.s of ciphers.

① Substitution cipher. (mapped letters).

↳ Caesar Cipher (not really cipher ⇒ because doesn't have a Key). Key is fixed!!

→ (letter shift by 3) العدد 3

** What is the size of Key space in the subs. cipher assuming 26 letters ?!

** |K| = 26! **

- but it still insecure.?! (subs. cipher).

⇒ ● ⇒ the most common letter in English : * E *

فهاد الاشي بعد لاقترامه العالية لل cipher :. بنقدر نكتشف ال plaintext.

[1] Use the freq. of English letters. الخطوات :-

نعرف مثلاً إنه ال e باله english نسبته ⇐ 12.7%

⇐ منجمع من ال Cipher، ونحسب لكل letter ال freq بالنص (كم مرة تكرر)

وأكثر واحد تكرر بكون ال enc. تبعت هو e . :. هيك قدرت أسترجع حرف ✓ ثم

الحرف الثاني الأكثر باللغة هو t ...... الخ. بحيث بجمع ال freq تبعات بهلك فنبرزها

[2] Use freq. of pairs of letters. ب الخطوة التالية (diagrams).

"the", "an" ...

⇒ Cipher Text only attack !!

* 2. Vigener Cipher $(\overset{m}{P}+K) \bmod 26. = C$

** Key is a word **

- exp K = CRYPTO.

m = WHAT ANICEDAYTODAY

K = CRYPTOCRYPTOCRYPT

C =

- بصير أكرر ال Key تحت الجملة؟ (لحد ما يغطي ال message) بعدها بنجمعهم ع بعض
ا 26) وبنطلع الحرف.

- break Vigener Ciper is easy

① assume that we know the length of the Key. ① **

** ② نقسم أحرف ال Cipher لـ groups بعدد ال Key)
يعني مثلاً Key بتكون من 6 حروف. / نقسم ال Cipher كل 6 أحرف
سوى

** ③ بعدها بنطلع ع أول حرف من كل group (كل عدد لأحرف م encrypted)
لأول حرف بال Key. ⇐ وبنصير نشوف أكثر حرف متكرر بال Cipher بنسبة
لل (E) بالـلغة وبنقارن.

°° وإذا ما كنت أعرف عدد أحرف ال Key بصير أجرب 6 مرة باخذ 1، مرة 2 ... إلخ
وبالآخر رح أعطيني ال plaintext

* 3. Rotor Machines (Rot - ___)

!! CT only attack !!

* 4. Data Enc. standard. 8 chars at time.

\# Keys = 2^{56} , bolcksize = ~~8~~ 64. bits

- because ~~es~~ es use small key space => these days can be broken by a bute-force ∴ Insecure.

* هنلاقي AES بيستخدموا 128 bit keys

Exp (Vigener Cipher)

→ m = "Hello There" , k = "ITEAM"

Sol. * m H e l l o T h e r e

k I T E A M I T E A M

$\}$ + → مرتبين ازاي بال alphabit

0 1 2 3 - - -

a b c d

- Finite Set $U = \{0,1\}^n$

Exp $\{0,1\}^2 \Rightarrow \{00, 01, 10, 11\}$.

$1/4 \quad 1/4 \quad 1/4 \quad 1/4 = 1$

** $P : U \to [0,1]$

prob → $\sum P(X) = 1$ $\quad \frac{1}{2} \quad \frac{1}{8} \quad \frac{1}{4} \quad \frac{1}{8} = 1$

- Uniform dis $P(X) = 1/|U|$
- Point dist. at X_0 $P(X_0) = 1$ $\forall x \neq x_0 \quad P(X)=0$.

** Events. $\underset{\text{is event.}}{\boxed{A}} \subseteq U$ $\quad Pr[A] = \sum_{x \in A.} P(x) \in [0,1]$

- $U = \{0,1\}^8$ $\quad \therefore |U| = 2^8 = 256.$
- $A = \{ \text{lsb}_2(x) = 11 \}$

$0101101\underbrace{10}_{\text{not } A}$ X

$0101101\underbrace{11}_{A}.$

$\therefore Pr[A] = \frac{1}{4} = 64 * \frac{1}{256}.$

** Union Bound A_1 + A_2 (events).

$Pr[A_1 \cup A_2] \leq Pr[A_1] + Pr[A_2].$

$A_1 \cap A_2 = \emptyset$ لما المساواة بتحصل

- if A & B are indep $\therefore Pr[A \& B] = Pr[A] \cdot Pr[B].$

REDMI NOTE 9
AI QUAD CAMERA

** Stream Cipher (chapter -2-).

• $\underline{E} : K * m \rightarrow C$, • $\underline{D} : K \times C \rightarrow m$

$$\therefore D(K, \underbrace{E(K,m)}_{c}) = m.$$

→ E is often randomized / D is always Deterministic.

** The one Time Pad **

• $m = C = \{0,1\}^n$ Key space = message space $\{0,1\}^n$

• * ⇒ Key = (random bit string as long as the message).

$$\Leftrightarrow \begin{cases} C = E(K,m) = K \oplus m. \\ m = D(K,c) = K \oplus C. \end{cases}$$

Exp → $D(K, E(K,m)) = D(K, K \oplus m) = \underbrace{K \oplus K} \oplus m$

$$= 0 \oplus m = \boxed{m}$$

** it is security ?!

* You are given a message (m) and its OTP enc. (c)

Can You compute the OTP Key from m & c ?!

••• Yes, the Key is $K = m \oplus C$.

** OTP $\overset{\text{very}}{\text{Fast}}$ enc/dec

↳ but Key long as plaintext.

- Is the OTP a good cipher?!

** Security Cipher.

حسب ال Shannon ال Security تعني ⇐ إذا كان عند ال ciphertext
مش لازم يكسب عندك أي معلومة عن ال plaintext.

* A cipher (E,D) over (K,M,C) has perfect secrecy if:

$$len(m_0) = len(m_1)$$

$$\forall m_0, m_1 \in M \quad \& \quad c \in C$$

~~Pr[E(K,m0)]~~

$$\boxed{Pr[E(K,m_0)=C] = Pr[E(K,m_1)=C]}$$

- where K is uniform. in K. $(K \xleftarrow{R} K)$.

⇐ شو فائدتها ؟!

لما ال attacker يكون عنده C فال prob إنه يكون ال enc من m0 هو
نفس ال prob إنه يكون من m1 !ما :. ما في أي طريقة يعرف مين هو الصح !!

** In OTP (Most powerful adversary learns nothing about PT from CT.

⟹ No CT only attack !! (but other attacks are possible).

* OTP has perfect secrecy !!

$\forall m, c$: $\Pr[E(k,m)=c] =$ # of keys $k \leftarrow K$

s.t $E(k,m)=c$ divided by $|K|$.

$\therefore \frac{\#\text{Keys}}{|K|} = \Pr[E(k,m)=c]$

- So :- $\forall m, c$: $\#\{k \in K : E(k,m)=c\} =$ constant ?!

$\Rightarrow$ ∴ Cipher has perfect secrecy.

Q let $m \in \mathcal{M}$ & $c \in C$, How many OTP Keys map m to c ?! ①

- Proof (perfect secrecy): For OTP, if $E(k,m)=c$

$\Rightarrow k \oplus m = c \Rightarrow \underline{\underline{k = m \oplus c}}$

$\Rightarrow \#$ of $k \in K : E(k,m)=c\} = \underline{1} \quad \forall m, c$

∴ OTP has perfect secrecy.

** OTP : no CT only attack !! (but other attack possible).

intro. زي ما قلنا بال

** Problem in OTP ⇒ The secret Key is long **

هل يوجد ciphers

have perfect secrecy

but shorter Keys ?!

** if the cipher has perfect secrecy, the num. of keys in the cipher must be at least the num of m. that cipher can handle.

**** Perfect Secrecy $\Rightarrow |K| \geq |M|$

● لازم يتحقق الشرط ← لتحقيق عكسه

** OTP hard to use, because long key. in practice

- ① OTP has perfect secrecy (No CT only attack) but hard to use.
- ② perfect-secrecy $\Rightarrow |K| \overset{\text{len}}{} \geq \overset{\text{len}}{} |M|$

⇒ * Stream cipher. (making OTP practical)

⇐ الـ key "pseudorandom"

- PRG (PseudoRandom Generator). is a func. $\underbrace{G:\{0,1\}^{\overset{\text{seed}}{(s)}}}_{\text{seed space}}$

$$G\ \underbrace{\{0,1\}^s}\ \xrightarrow[\text{map}]{}\ \{0,1\}^n \qquad n \gg s$$

الـ Generated بـ seed ثم لـ map

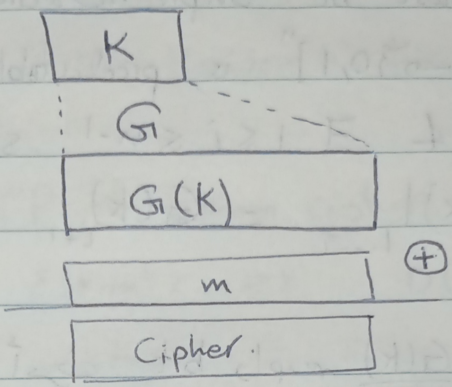
أكبر بكتير

G must "efficient" computable:

** seed : random

** بنستخدم الـ seed ⇐ كأنها الـ Key. وبنستخدم الـ Generator (G) عشان

أعمل expand للـ seed عشان أطلع much larger seq.



$$\therefore\ C = E(k,m) = m \oplus G(K)$$

$$D(K,C) = C \oplus G(K).$$

⇒ ✱ Why is secure ?!

●● Can a stream cipher have perfect secrecy ?!

∴ ⇒ No, since the key is shorter than the message.

?!

** PRG must be Unpredictible !!

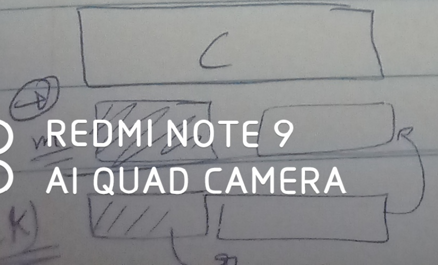
↳ Suppose PRG is predictible !

$$\exists i :\ G(k)\Big|_{1,\dots,i} \xrightarrow{algo.} G(k)\Big|_{i+1,\dots,n}$$

إذا عرفت أول bit ✱ هقدر أحسب الـ bits الباقية داوجد.

∴ هيبقى بجير not secure فلازم

الـ PRG تكون Unpredictible



لو كان الـ attacker يعرف بداية الـ PT

∴ هقدر يعرف الـ prefix للـ G(K) وبما predictible

∴ يعرف باقي الـ G(K) و هقدر هلاقي يطلع الـ m

⊕ :. إذا كان PRG (predictable) يكون no secure ✓. حتى لو بس عرفت أول $\underline{\underline{\text{bit}}}$ ⇐ رح أقدر أعرف الباقي ✓

$\therefore \Rightarrow$ PRG must be Unpredictable.

- We say that $G: K \to \{0,1\}^n$ is predictable if.

  $\exists$ "eff." alg. A & $\exists\ 1 \le i \le n-1$ s.t.

  $$\Pr[A(G(K))|_{1\ldots i} = G(K)_{i+1}] \geq \left(\frac{1}{2} + \epsilon\right)$$

  $$K \xrightarrow{R} K$$

* يعني إذا كنت أعرف bit واحدة من $\underline{\underline{G(K)}}$ أنا قادر أعرف ال bit اللي بعده بنسبة أكبر من أو تساوي $(\frac{1}{2} + \epsilon)$.

$\epsilon \geq \frac{1}{2^{30}}$ (not negligible).

* $\therefore$ PRG is unpredictable if it's no "eff" can predict bit $(i+1)$ for non-neg. ϵ.

$\underline{\underline{Q}}$ Suppose $G: K \to \{0,1\}^n$ is s.t for all K $\boxed{XOR(G(K)) = 1}$

Is G predictable ?!

●● Yes, given the first $n-1$ bits I can predict the n'th bit.

↳ Weak $\underline{\underline{\text{PRGs}}}$ ⇒ a, b, p parameters.

$r[0] \equiv$ seed

```
r[i] ← a.r[i-1] + b mod p
output few bits of r[i]
i++
```

≫ Easy to predict !!!

∴ glibc random() : (Easy to predict) !!

$r[i] \leftarrow (r[i-3] + r[i-31]) \% 2^{32}$

output $r[i] >1$

⇒★ Never use Random() for crypto!!

★★ ATTacks on OTP 4 Stream ciphers.

ملخص

- → OTP $E(k,m) = m \oplus k$ $D(K,C) = C \oplus K$.
- → mak OTP practical use PRG $G: K \rightarrow \{0,1\}^n$.
- → stream cipher. $E(K,m) = m \oplus G(K)$, $D(K,m) = C + G(K)$
- → For Security ● PRG must be unpredictable.

★ [I] Attack #1 :-

- never use stream cipher Key more than once !!

Exp

$C_1 \leftarrow m_1 \oplus PRG(K)$

$C_2 \leftarrow m_2 \oplus PRG(K)$ ⇒ same pad!

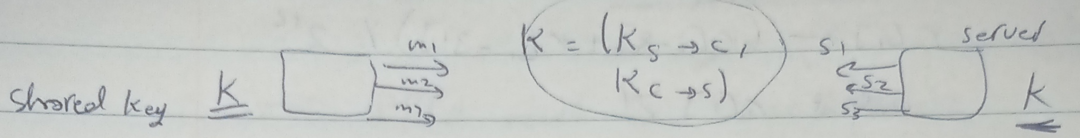
لنفرض إنه عندنا C_1 & C_2

- → $C_1 \oplus C_2 \rightarrow m_1 \oplus m_2$ $(PRG(K) \oplus PRG(K) = 0$
- → $m_1 \oplus m_2 \rightarrow \underbrace{m_1 , m_2}$!!

★★ ∴ إذا استخدمت نفس ال pad حتى أعمل encrypt ل different mes.

رح يكون not secure.

- MS-PPTP Same Key.



[m1 || m2 || m3 ...] ⊕ PRG(K) [s1 || s2 || s3 ...] ⊕ PRG(K)

* need different keys for c→s & s→c.

- 802.11b WEP.
  - length of IV ∵ 24 bits.
  - ∴ Repeated IV after 2^{24} ≈ 16 M frames.

∴ نقدر نحل encrypt لـ 2 diff frames ∴ الـ attacker نقدر نعرف الـ PT

- حل آخر إنه كل frame نخليه encrypt باستخدام diff. key || with IV.
  - key frame 1 (1 || k) 24 bits
  - key " 2 (2 || k)

  these keys very much Related to others.

  كلهم إلهم نفس الـ suffix 2^{24}

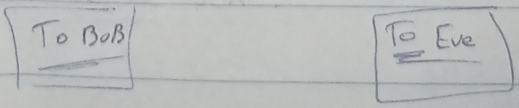
- 10^6 Frame ⇒ You can recover the secret key.

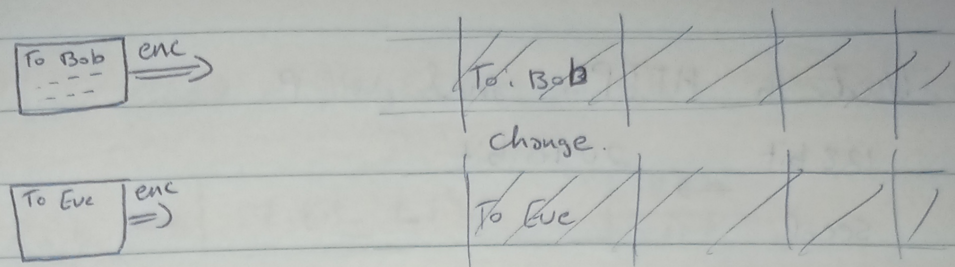
  = 40,000 Frames suffix.

* الحل الأفضل كل frame أعطيه key مختلف (ما إلهم شي). (indep.)

  → each frame has a pseudorandom key.

- disk Eve.





الـ attacker بلاحظ إنه التكرار بسبب بيؤدي للثغرة !!

!* (2-time pad attack) *! not secure.

because Used the same pad.

** 2-Time pad.

- never use stream cipher more than once.
- Network traffic (نعمل Key جديد لكل session.)
- Disk encryption (لما نستخدم نفس الـ stream cipher)

* Attack 2.

No Integrity. cipher

$m \longrightarrow$ (enc $m+k$) $m \oplus k$

$\underline{P}$ $\Leftarrow$ attacker عدل modify

!؟ $\underline{m \oplus P} \Leftarrow$ dec($\oplus k$) $\Longleftarrow$ $m \oplus k \oplus P$.

∴ modification the ciphertext are undetected & have predictable impact on plaintext.

** RC4 Used in HTTPS and WEP.

128 bit → 2048 bit - - -

seed

* Weakness : ① Bias initial in output.

$Pr[2^{nd} \text{ byte} = 0] = 2/256$.

② Prob of (0,0) is $\frac{1}{256^2} + \frac{1}{256^3}$

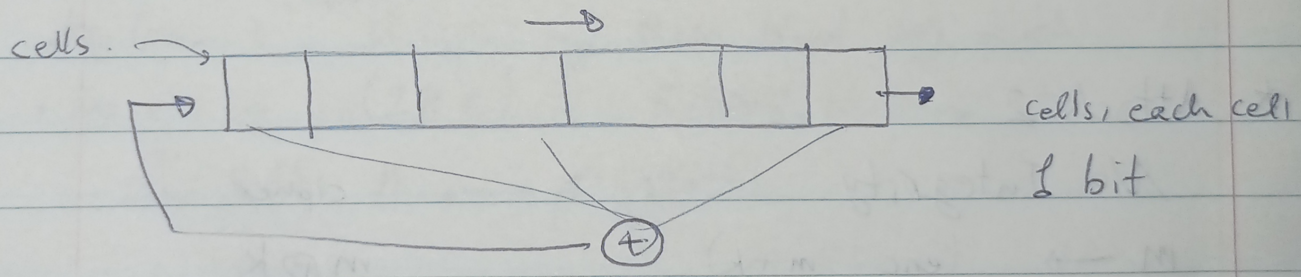
③ Related Key attacks.

⇓

Exp (Related Keys to others).

** CSS (Badly Broken).

↳ Linear Feedback shift Reg. (LFSR).



بعدكل clock بنعمل shift وآخر bit بنرجع

ونتيجة ال Xor بنعمل أول bit

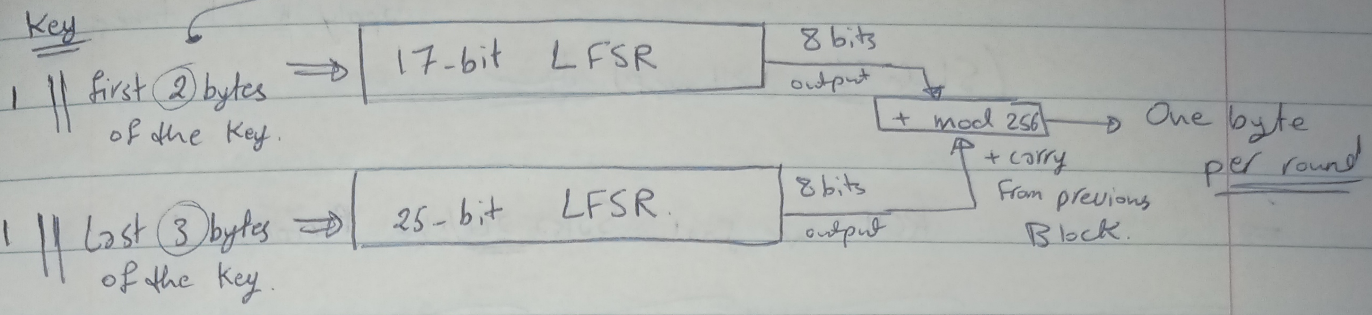
• seed = initial state of LFSR.

•• DVD (CSS) 2 LFSR

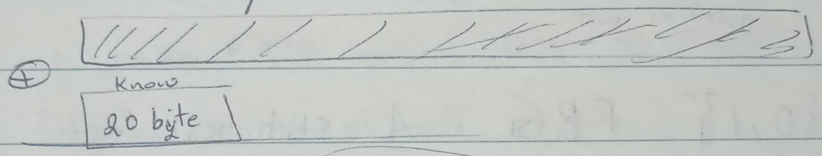
•• GSM (A5) 3 LFSR ⇒ all broken.

•• Bluetooth (E0) 4 LFSR

- CSS : $\underbrace{\text{Seed} = 5 \text{ bytes}}$ = 40 bits. 2 LFSR's



★ Easy to Break ≈ 2^{17} time.



✓ $\boxed{\text{CSS}}$ ← ونجرب ال 17 bit دي 20 byte لي بال LFSR

بنفضل نجرب ولحد ما عرفناه. نقدر نجيب ال 25 bit LFSR

∴ وبعدين نقدر نعرف باقي ال out تاع ال CSS ✓

★ $\underbrace{\text{estream}}$:- 2 inputs $\underbrace{\{0,1\}^s}_{\text{seed}} \times \underline{\underline{R}}_{\text{nonce}} \rightarrow \{0,1\}^n$ n>>s

↳ non-~~repeat~~ repeating value for given Key.

∴ $E(K, m; r) = m \oplus PRG(K; r)$

★ nonce ↳ The pair (k, r) is never used more than once

→ Salsa 20 :- $\underbrace{\{0,1\}^{128 \text{ or } 256}}_{\text{seed}} \times \underbrace{\{0,1\}^{64}}_{\text{nonce}} = \{0,1\}^n$ max $n = 2^{73}$ bit

- $Salsa20(K; r) = H(K, (r, 0)) \,||\, H(K, (r, 1)) \,||$.

≪ long pseudorandom seq.)

(K, r, counter for steps) ← 3 inputs باخد function هو H

apply H 10 times

الشكل لقريب الحل (Slide 46

⇒ ال Performance كان Salsa 20 أفضل من RC4 . . .

** **Adv** حتعرف هل ال statistical good or not.

→ let G: K {0,1}^{n} PRG   A statistical test on $\{0,1\}^n$

- $\text{Adv}_{PRG}[A, G] = \left| \Pr_{K \to K}[A(G(K)) = 1] - \Pr_{r \xleftarrow{R} \{0,1\}^n}[A(r) = 1] \right|$

- adv colse to 1 ? ⇒ behave differently ⇐ ① يعني ال
  ∴ A can disti G From rand

- adv ≈ ≈ 0 ? ⇒ A Could not dist.

Q $A(X) = 0$ $\text{Adv}_{PRG}[A, G] = 0$

Q if [msb(X) = 1] output '1' else 0.

$\text{Adv} = 2/3 - \frac{1}{2} = 1/6$

★ We say that $G : K \to \{0,1\}^n$ is a secure PRG if :-

$\forall$ "eff" stat. tests A.

$Adv_{PRG}[A, G]$ is "neg"

★ Are there provably secure PRGs?! Unknown!!

★★ ⇒ A secure PRG is unpredictable.

∘∘∘ PRG predictable ⇒ PRG is insecure.

Solution Suppose A is an eff algorith.

$$Pr[A(G(k)|_{1,\dots,i}) = G(k)|_{i+1}] = \frac{1}{2} + \varepsilon$$

⇐ يعني إذا كان معروضين لي أول bit تقدر أتوقع ال bit اللي بعده ابقى هو

For non-neg $\varepsilon = 1/1000$.

★ Define Statistical test B

$$B(X) = \begin{cases} \text{if } A(X/_{1,\dots,i}) = X_{i+1} & \text{output } 1 \\ \text{else output } 0 \end{cases}$$

- $r \xleftarrow{R} \{0,1\}^n$ $Pr[B(r) = 1] = \frac{1}{2}$.

- $k \leftarrow K$ $Pr[B(G(k)) = 1] > \frac{1}{2} + \varepsilon$

$$Adv[B, G] > \varepsilon$$

∴ A is a good predictor & B is a good statisticaltest that breaks the Generator.

⇒ Good Generator : no good stat. test & no predictors.

يعني مانقدر أعرف X_{i+1} من X_i !!.

★ Unpredictable PRG is Secure.

إذا كان $\underline{G}$ هو Unpredictable لـ i لكل i = من $0 \to n-1$

∴ يعتبر G is secure PRG.

★ إذا ما في predictor يقدر يميز بين G & random ★

نهائي Statisticall نقدر !!.

- We say that p_1 & p_2 are computationally indistinguishable if

لا يمكن التمييز بينهم

- if ∀ "eff" stat. tests A

$$\left| \Pr_{X \to P_1}[A(X)=1] - \Pr_{X \to P_2} A(X)=1] \right| < \text{neg.}$$

∴ PRG is secure if $\{ K \xleftarrow{R} K : G(K) \}$ $\approx$ Uniform $(\{0,1\}^n)$

★ Semantic cipher.

① Attacker cannot recover secret key.

$E(K,m) = m.$ <

② // // // all plain text

$E(K, m_0 \| m_1) = m_0 \| E(K, m_1)$

insecure $\underline{m_0}$ available for attacker.

~~[illegible] hard~~

~~to attack~~

③ Shannon : CT should reveal no "info" about PT.

Cipher

** Shannon perfect secrecy :-

(E, D) has J if $\forall m_0, m_1 \in M$ $(|m_0| = |m_1|)$

$\underline{\{E(k, m_0)\}} \; \textcircled{=} \; \underline{\{E(k, m_1)\}}$ where $k \leftarrow K$

$\approx_p$

computationally indistinguishable.

** Semantic Security.

∴ E is semantically secure if for all "eff" A $Adv_{SS}[A, E]$ is neg.

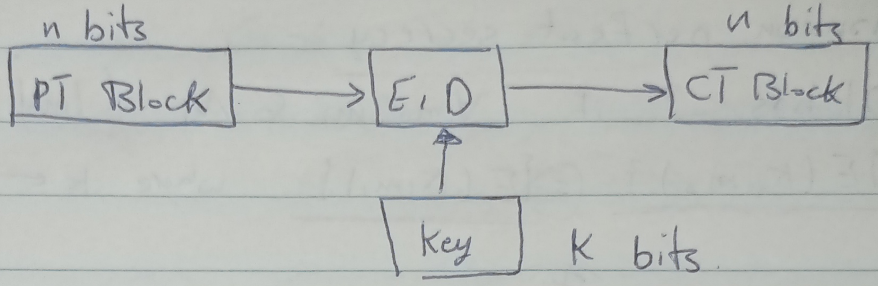
→ For explicit $m_0, m_1 \in M$

$\{E(k, m_0)\} \approx_p \{E(k, m_1)\}$.

** OTP is semantically secure.

* التفصيل بالسلايد ٧٧ !

★ Block Cipher :- ★



** Exp
① 3DES n = 64 bits K = 168 bits
② AES n = 128 bits K = 128, 192, 256 bits

** ال Key بنحلها expansion ل $K_1, K_2 ... K_n$.
↳ Rounding => by iteratively encr. the message again & again & again. using round function

** Round Function : $R(K, m)$
inputs

(num of rounds) for 3DES (n = 48) (for AES - 128) (n = 10)

بناخد ال message مع K_1 ثم الناتج مع K_2 ، وهكذا ..... لحد ما
يخلص كل ال rounds (والناتج بكون هو ال Cipher

ال Stream Cipher عكس ال Block Cipher.

- pseudo random Function in a pseudo random permutation.

① PRFs PRPs

↳ defined over a Key space, input space, output space

K X Y

✓✓ * $\{F : K \times X \to Y\}$

- s.t exists "efficient" algo. to evaluate $F(K,X)$

② PRPs

↳ defined over (K, X).

$\{\mathcal{E} : K \times X \to X\}$ s.t.

↳ 1. Exists "efficient" deterministic algo. to evaluate $\mathcal{E}(K,x)$.

2. The Function $\mathcal{E}(K,\cdot)$ is one to one.

3. Exists "eff." inversion algo. $D(K,y)$

•• Exp.s on PRPs

→ AES : $K \times X \to X$ where $K = X = \{0,1\}^{128}$

→ 3DES : $K \times X \to X$ // $X = \{0,1\}^{64}$, $K = \{0,1\}^{168}$.

⊛ Functionally, any PRP is also a PRF

∴ A PRP is a PRF where $X = Y$ & is efficiently invertible.

((PRP is a special case of PRF)) ⇐ *

** Secure PRFs.

- Let $F : K \times X \to Y$ be a PRF
  - Funs [X, Y]
    - all functions from a set X to the set Y
    - $\Rightarrow \therefore$ size $|Y|^{|X|} = 2^{128 \cdot (2^{128})}$
- $S_F = \{F(k, \cdot) \text{ s.t } k \in K\} \subseteq \text{Funs}[X, Y]$

Smaller set of functions (set s sub F).

$\Rightarrow$ In AES   128-bit Keys إذا استخدمنا

** ال size ح يكون ال S_F

$$|S_{AES}| = \underbrace{2^{128}}_{\text{much smaller than Funs}}$$

Funs: all possible Func. From X to Y.

** PRF is Secure if :-

- random Function in Funs[X, Y] is indistiguishable from a random function in S_F.

** يعني بنقدرش نميز ال random Function في Funs[X, Y] عن ال S_F

- $f \leftarrow$ Funs[X, Y]   Truly Random Function.
- $k \leftarrow K$ pseudo " "

** هنا لما أجيب نقطة $\boxed{X}$ $(X_1, X_2 \ldots)$ ** واعوضه في

- $f(X)$ — Truly ②
- $F(k, x)$ — pseudo ①

$\Rightarrow$ الخصم (ال attacker) ما برع
قدر يعرف هل جاية م
① ولا م ②

:PRF بيكون secure إذا الـ attacker ما قدر يحدد
الـ difference ، يعني بين الـ pseudo & Truly

* PRP ⇐ تقريبًا the same as PRF وبس بدل الـ $f \leftarrow Funs[X,Y]$
بدل Random func.

Per [X] ⇒ Random permutation on the set X.
- random one to one function on X.

Q Let $F : K \times X \to Y \to \{0,1\}^{128}$ be secure PRF

- Is the following secure PRF?

$$G(K,x) = \begin{cases} 0^{128} & \text{if } X = 0 \\ F(K,x) & \text{o.w.} \end{cases}$$

⇒ * No, it is easy to distinguish G from a Rand. Func.
← عشان الـ attacker يقدر ينفذ الـ func عند X=0
:الـ prob إن الـ result بتكون 0 ⇐ $\frac{1}{2^{128}}$

* PRF ⇒ PRG

Let $F : K \times \{0,1\}^n \to \{0,1\}^n$ be a secure PRF.
∴ ⇒ Then the following $G : K \to \{0,1\}^{nt}$ is secure PRG
generator
t blocks of n bits each.

$$** \quad G(K) = F(K,\underline{0}) \,||\, F(K,\underline{1}) \,||\, \ldots \,||\, F(K,\underline{t})$$

PRF

:يعني ناخد الـ Key الـ PRF ونعمل Expand لـ n*T

* Key property : parallelizable يعني اذا كنت تشتغل

2 processors ⇐ عندك core (process) يشتغل ال even entries

وآخر يشتغل ال odd entries

$$G(K) = \underbrace{F(K,0)}_{\text{even}} \,||\, \underbrace{F(K,1)}_{\text{Odd}} \,||\, - - - \quad . - - -$$

** ال stream ciphers لي قبل كلام بشتغلوا Sequential فقط، زي

RC4 : أما هاد ال stream cipher ⇐ بيقدر يشتغل in parallel (لـ t)

⇒ PRF (Security) F(K, ·) indist from rand. fun. f(·)

لا يمكن تمييزه

يعني ما بنقدر نميز ال Truly من pseudo . . . ---

Truly Generator $G(k) = F(0) \,||\, F(1) \,||\, - - \; F(t)$

∴ لا يمكن تمييزه من pseudo Rand. Func. Generator

perfectly Secure generator

output (Truly Random Output

~~** إذا ال pseudo Rand Func. Generator~~

⇒ Data Encry. Standard.

* DES : amazing successful cipher used in banking industry

وبعدها استبدلوه ب AES

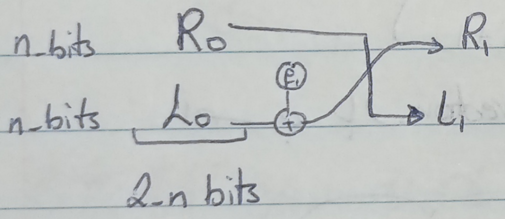
● * **DES** has a rich history of attacks.

* DES - Core Idea - Feistel Network.

- Given functions $f_1, \dots f_d : \{0,1\}^n \rightarrow \{0,1\}^n$ ⇒ n-bits
- build a new function. (invertible func.)

$F : \{0,1\}^{2n} \rightarrow \{0,1\}^{2n}$ ⇒ 2n-bits

● Feistel Network : (From to bottom)



● [1] ننسخ R_0 لـ L_1 بدون أي تغيير.

● [2] نعمل $R_1 = L_0 \oplus F_1$

1-Round of a Feistel Network وهاي بسموها

ونعمل ال Rounds مع تغيير ال f_1 لـ f_2, f_3 ... حسب ال round

كل ما نوصل آخر Round ح f_d ⇐ أ/ح بنتج هنا

ال Output

هو R_d + L_d.

** Formula :
$$\begin{cases} R_i = F_i(R_{i-1}) \oplus L_i \\ L_i = R_{i-1} \end{cases} \quad i = 1, --- d.$$

⟺ ∴ This Feistel Network mapping a 2n bit input to 2n bit output

** ال function مش شرط يكون أي function (عادي ما يهمني).
$(f_1 --- f_2)$

دايمًا ال Feistel Network's result ⇐ is invertible.

→ Prove that !! ⇒ inverse

عندي $\{R_{i+1}, L_{i+1}\}$ بده أحسب R_i L_i ∴ بدي

أحسب بال reverse direction

$$\boxed{\begin{aligned} &① \text{ For } R_i = L_{i+1} \quad ✓ \\ &② \text{ ״ } L_i = f_{i+1}(L_{i+1}) \oplus R_{i+1} \end{aligned}}$$ Inverse

الرسمة بال Slides

لما نخط الرسمة دي بجهتين بقدر أحسبها ال inverse

بلش م Round num D ____ D-1 ____ First Round ← وبعدها نحصل على

R_0

L_0

∴ inversion the same as the encryption (the only difference (functions applied in reverse order)
$f_1, --- f_d$.

- Encryption (start with f_1 + end with f_d)
- inversion (start with f_d + end with f_1)

∴ Decrypt. Circuit ≫ 1) General method for build invertible func.
From arbitory Func. (F_1 -- F_d)
2) Used in many block cipher but not AES

* ⇒ Rockoff

** إذا استخدمنا Secure PRF
وعملنا عليه

3-round Feistel

$F : K^3 \times \{0,1\}^{2n} \rightarrow \{0,1\}^{2n}$ ∃ secure PRP

∴ ينتج

invertible function that's indisting.
From ∃ ∃ truly random invertible

** IF you start with secure PRP
You will end with secure block cipher.

- لو استخدمنا PRF في كل round في ال Feistel Network

$\underbrace{F(K_0, R_0)}_{\text{Round } 0}$ $\underbrace{F(K_1, R_1)}_{\text{Round } 1}$ -- $\underbrace{F(K_3, R_3)}_{\text{Round } 3}$

Used 3 indep. Keys ⇒ لع نضمن بالآخر ونحصل
على secure PRP

✓✓

* 16 round Feistel network

$f_1, \ldots f_{16} : \{0,1\}^{32} \rightarrow \{0,1\}^{32}$ $f_i(x) = F(K_i, x)$

<u>32 bit</u> // The DES acts on <u>64</u> bits blocks

$2 * 32 = 64$ ✓

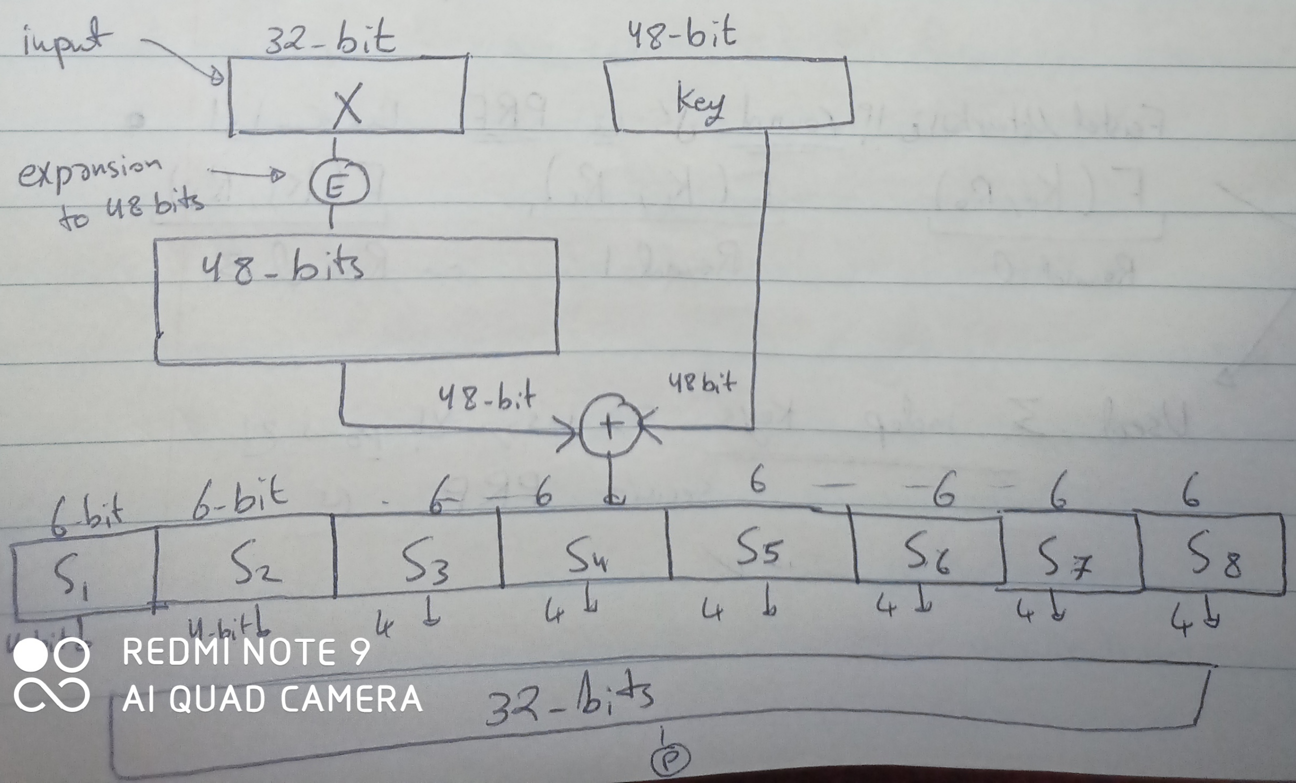
derive from <u>F(ki,x)</u> ⟸ Round بدل Functiond الـ **
$F_1 \cdot - \cdot F_{d}$

≪ Single Function
with diff. Keys ≫

** ∴ (16 diff Keys $\overset{give}{\Longrightarrow}$ 16 diff. round functions.)

- 56 bits Key is expanded into these round Keys
each Round Key = <u>48 bit</u>

** The Function F(Ki, x) (Work!)



طريقة عمل $\underline{\underline{F(K_i, X)}}$

[1] بعمل expand لل input.

[2] بعمل Xor بين ال expand input + key

[3] بعدها رح ينتج ناتج منه 48-bits ⇐ بنقسمه لـ 8 groups

وكل group ← 6-bit

S-boxes

[4] بعمل map كل 6-bit لـ 4 bit

[5] وبعدها بجمعهم بطلع عندي 32-bits

[6]. بعمل another permut. ← يعني بعمل maps the bits around

For example (bit 1 become bit 9 ---).

** وبعدها $\boxed{\text{32 bits}}$ هاد بيكون ال output لل Function.

** S-box → Function $\{0,1\}^6 \to \{0,1\}^4$ which is implemented as look-up table.

⟺ How these boxes chosen?!.

[1] Very Bad choice of S-boxes

تعتبرهم linear

** تعبير ال Sbox ⇐ $S_i(x) = A_i \cdot X$

matrix * Vector ⇒ Linear Function!

⊛ ∴ if $S_i(x)$ Linear Function, then DES is insecure ✓!

والسبب **

- matrix B $\underbrace{64}_{}^{m} + \overset{Keys}{16} * \overset{bits}{48} = \underset{\text{one long vector}}{\underline{\underline{832}}}$

ولما نضرب ال B * vector بتطلعلي different bits of CT → 64-bits

∴ ⇒ This mean that DES is linear.

* 3 outputs of DES.

$DES(K, m_1) \oplus DES(K, m_2) \oplus DES(K, m_3)$

>> لاحظ بال Slide $\boxed{B}$ ← عبارة matrix

$\Rightarrow DES(K, \underline{m_1 \oplus m_2 \oplus m_3})$

** : 832 بحتاجلي اعمل recover for the entire secret key.

** : Choosing the S-boxes and P-box at random ⇒ insecure (Linear !!) and Key recovery after ≈ 2^{24} outputs).

** No output bit should be close to Linear Function. of the input bits.

**

★ Exhaustive Search Attacks

- given a few input output pairs $(m_i, c_i = E(k, m_i))$ $i = 1, \dots 3$ Find key.

!! $(m_1\ \ m_2\ \ m_3) \xrightarrow{K} (c_1\ \ c_2\ \ c_3)$

Unique ⇐ Key كيف نعرف إنه ال ①

↳ Suppose DES is an Ideal cipher

2^{56} d random invertible func.

DES implements a random ال ← Key يعني لكل

invertible function.

★★ 2^{56} Keys in DES

∴ DES is a collection

$(\pi_1, \dots \pi_{256}\ \ \{0,1\}^{64} \to \{0,1\}^{64})$

just for Ideal !!

⇒★ $C = DES(K, m)$ (need one Key)

∴ key is Unique.

- Proof: $Pr[\exists K' \neq K \,;\, C = DES(K, m) = DES(K', m)]$

$$\leq \sum_{k' \in \{0,1\}^{56}} Pr[DES(k, m) = DE(k', m)] \leq 2^{56} \frac{1}{2^{64}}$$

for premutat

$$= \frac{1}{2^8} = \frac{1}{256}$$

① ∴ The prob that the Key is not Unique $= \frac{1}{256}$ ★

② ★ The prob that the key Unique.

$1 - \frac{1}{256} = 99.5\%$ ★

• فـ إذا كان فقط **1 Key** بين m 4 C ⇐ نقدر نوجد ال K !

** Now, Let see if we took 2-pairs
$(m_1, m_2) \Rightarrow (c_1, c_2)$

DES • Unicity prob $\approx \underbrace{1 - \frac{1}{2^{71}}}_{\text{close to } 2}$

• this seems that only **Key** will map this pair of message to this pair of cipher text

** For Aes -128 $\left(\bullet \text{ Unicity prob} \approx 1 - \frac{1}{2^{128}} \right)$

** ∴ 2 input/output pairs are enough for exhaustive Key search.

كافي لكشف الـ Keys !!

** ∴ إذا أعطانا أكثر من **pair** للـ m4C وسهلنا نوجد
الـ Key

** exhaustive Key Search بنلغي **