Stream ciphers

The One Time Pad

Symmetric Ciphers: definition

<u>Def</u>: a **cipher** defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$

is a pair of "efficient" algs (***E***, ***D***) where

$E: \mathcal{K} \times \mathcal{M} \to \mathcal{C}$, $D: \mathcal{K} \times \mathcal{C} \to \mathcal{M}$

s.t. $\forall m \in \mathcal{M}, k \in \mathcal{K}:$ $D(k, E(k,m)) = m$

- ***E*** is often randomized. ***D*** is always deterministic.

The One Time Pad (Vernam 1917)

First example of a "secure" cipher

$\mathcal{M} = \mathcal{C} = \{0,1\}^n$, $\mathcal{K} = \{0,1\}^n$

key = (random bit string as long the message)

The One Time Pad (Vernam 1917)

$c := E(k,m) = k \oplus m$

$D(k,c) = k \oplus c$

msg:	0	1	1	0	1	1	1	$\oplus$
key:	1	0	1	1	0	1	0	
CT:								

Indeed:

$D(k, E(k,m)) = D(k, k \oplus m) = k \oplus (k \oplus m) = (k \oplus k) \oplus m = 0 \oplus m = m$

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

Can you compute the OTP key from m and c ?

No, I cannot compute the key.

Yes, the key is $k = m \oplus c$.

I can only compute half the bits of the key.

Yes, the key is $k = m \oplus m$.

The One Time Pad (Vernam 1917)

Very fast enc/dec !!

… but long keys (as long as plaintext)

Is the OTP secure? What is a secure cipher?

What is a secure cipher?

Attacker's abilities: **CT only attack** (for now)

Possible security requirements:

attempt #1: **attacker cannot recover secret key**

$E(k,m)=m$ would be secure

attempt #2: **attacker cannot recover all of plaintext**

$E(k, m_0 \| m_1) = m_0 \| k \oplus m_1$ would be secure

Shannon's idea:

CT should reveal no "info" about PT

Information Theoretic Security
(Shannon 1949)

Def: A cipher $(\boldsymbol{E}, \boldsymbol{D})$ over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ has **perfect secrecy** if

$\forall m_0, m_1 \in \mathcal{M}$ $\quad (\text{len}(m_0) = \text{len}(m_1))$ and $\forall c \in \mathcal{C}$

$$\boxed{\Pr[E(k, m_0) = c] = \Pr[E(k, m_1) = c]}$$

where k is uniform in $\mathcal{K}$ $\quad (k \xleftarrow{R} \mathcal{K})$

Information Theoretic Security

<u>Def</u>: A cipher ***(E,D)*** over (K,M,C) has **perfect secrecy** if

$$\forall m_0, m_1 \in M \quad (|m_0| = |m_1|) \quad \text{and} \quad \forall c \in C$$

$$Pr[\, E(k,m_0)=c \,] \;=\; Pr[\, E(k,m_1)=c \,] \qquad \text{where } k \xleftarrow{R} K$$

⟹ Given CT can't tell if msg is m_0 or m_1 (for all m_0, m_1)

⟹ most powerful adv. learns nothing about PT from CT

⟹ no CT only attack!! (but other attacks possible)

Lemma: OTP has perfect secrecy.

Proof:

$$\forall m, c: \quad \Pr_k\left[E(k,m)=c\right] = \frac{\#\text{keys } k \in \mathcal{K} \text{ s.t. } E(k,m)=c}{|\mathcal{K}|}$$

So: if $\forall m, c: \quad \#\{k \in \mathcal{K} : E(k,m)=c\} = \text{const.}$

$\Longrightarrow$ cipher has perfect secrecy

Let $m \in \mathcal{M}$ and $c \in \mathcal{C}$.

How many OTP keys map $\boldsymbol{m}$ to $\boldsymbol{c}$?

None

1

2

Depends on $\boldsymbol{m}$

<u>Lemma</u>: OTP has perfect secrecy.

Proof:

For OTP: $\forall m, c$: if $E(k,m) = c$

$\Longrightarrow k \oplus m = c \Longrightarrow k = m \oplus c$

$\Longrightarrow \boxed{\#\{k \in \mathcal{K} : E(k,m) = c\} = 1}$

$\Longrightarrow$ OTP has perfect secrecy $\blacksquare$

The bad news …

Thm: perfect secrecy $\Rightarrow$ $|\mathcal{K}| \geq |\mathcal{M}|$

i.e. perfect secrecy $\Longrightarrow$ key-len $\geq$ msg-len

$\Longrightarrow$ hard to use in practice !!

End of Segment

Stream ciphers

Pseudorandom Generators

Review

Cipher over (K,M,C): a pair of "efficient" algs (***E***, ***D***) s.t.

$$\forall\ m \in M,\ k \in K: \quad \boldsymbol{D}(k, \boldsymbol{E}(k, m)) = m$$

Weak ciphers: subs. cipher, Vigener, …

A good cipher: **OTP** $M=C=K=\{0,1\}^n$

$$E(k, m) = k \oplus m \quad , \quad D(k, c) = k \oplus c$$

<u>Lemma</u>: OTP has perfect secrecy (i.e. no CT only attacks)

Bad news: perfect-secrecy $\Rightarrow$ key-len $\geq$ msg-len

Stream Ciphers: making OTP practical

idea: replace "random" key by "pseudorandom" key

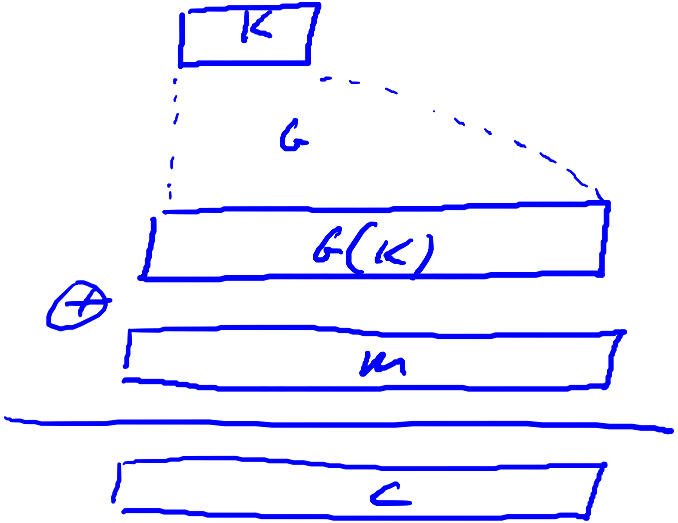
PRG is a function $G: \underbrace{\{0,1\}^s}_{\text{seed space}} \to \{0,1\}^n$ $n \gg s$

(eff. computable by a deterministic algorithm)

Stream Ciphers: making OTP practical

$c := E(k, m) = m \oplus G(k)$

$D(k, c) = c \oplus G(k)$



Can a stream cipher have perfect secrecy?

- Yes, if the PRG is really “secure”
- No, there are no ciphers with perfect secrecy
- Yes, every cipher has perfect secrecy
- No, since the key is shorter than the message

Stream Ciphers: making OTP practical

Stream ciphers cannot have perfect secrecy !!

- Need a different definition of security

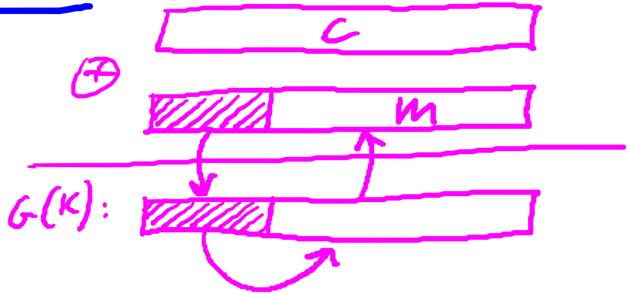
- Security will depend on specific PRG

PRG must be unpredictable

Suppose PRG is predictable :

$$\exists i: \quad G(k)\big|_{1,\dots,i} \xrightarrow{\text{alg}} G(k)\big|_{i+1,\dots,n}$$

Then:



even $G(k)\big|_{1,\dots,i} \longrightarrow G(k)\big|_{i+1}$

is a problem!

Dan Boneh

PRG must be unpredictable

We say that G: K ⟶ $\{0,1\}^n$ is **predictable** if:

$\exists$ "eff" alg. A and $\exists\ 0 \le i \le n-1$ s.t.

$$\Pr_{k \xleftarrow{R} \mathcal{K}}\left[A(G(k))\Big|_{1,\dots,i} = G(k)\Big|_{i+1} \right] > \frac{1}{2} + \varepsilon$$

For non-negligible ε (e.g. $\varepsilon = 1/2^{30}$)

Def: PRG is **unpredictable** if it is not predictable

⇒ ∀i: no "eff" adv. can predict bit (i+1) for "non-neg" ε

Suppose $G:K \longrightarrow \{0,1\}^n$ is such that for all k: **$XOR(G(k)) = 1$**

Is G predictable ??

Yes, given the first bit I can predict the second

No, G is unpredictable

Yes, given the first (n-1) bits I can predict the n'th bit

It depends

End of Segment

Stream ciphers

Negligible vs. non-negligible

Negligible and non-negligible

- In practice: ε is a scalar and
  - ε non-neg: $\varepsilon \geq 1/2^{30}$ (likely to happen over 1GB of data)
  - ε negligible: $\varepsilon \leq 1/2^{80}$ (won't happen over life of key)

- In theory: ε is a function $\varepsilon: Z^{\geq 0} \longrightarrow R^{\geq 0}$ and
  - ε non-neg: $\exists d$: $\varepsilon(\lambda) \geq 1/\lambda^{d}$ inf. often ($\varepsilon \geq 1/\text{poly}$, for many λ)
  - ε negligible: $\forall d, \lambda \geq \lambda_d$: $\varepsilon(\lambda) \leq 1/\lambda^{d}$ ($\varepsilon \leq 1/\text{poly}$, for large λ)

Few Examples

$\varepsilon(\lambda) = 1/2^{\lambda}$: negligible

$\varepsilon(\lambda) = 1/\lambda^{1000}$: non-negligible

$$\varepsilon(\lambda) = \begin{cases} 1/2^{\lambda} & \text{for odd } \lambda \\ 1/\lambda^{1000} & \text{for even } \lambda \end{cases}$$

Negligible

Non-negligible

PRGs: the rigorous theory view

PRGs are "parameterized" by a security parameter λ

- **PRG** becomes "more secure" as λ increases

Seed lengths and output lengths grow with λ

For every $\lambda=1,2,3,\ldots$ there is a different PRG G_λ:

$$G_\lambda : K_\lambda \longrightarrow \{0,1\}^{n(\lambda)}$$

(in the lectures we will always ignore λ)

An example asymptotic definition

We say that $\mathbf{G_\lambda : K_\lambda \longrightarrow \{0,1\}^{n(\lambda)}}$ is **predictable** at position i if:

there exists a polynomial time (in $\boldsymbol{\lambda}$) algorithm A s.t.

$$\Pr_{k \leftarrow K_\lambda}\Big[\, A\big(\boldsymbol{\lambda}, \mathbf{G_\lambda}(k)\big|_{1,\dots,i}\big) = \mathbf{G_\lambda}(k)\big|_{i+1} \,\Big] \;>\; 1/2 + \varepsilon(\boldsymbol{\lambda})$$

for some non-negligible function $\varepsilon(\boldsymbol{\lambda})$

End of Segment