3/12

Table of Contents

- OTP: One-Time Pad(Symmetric Cipher)(Vernam 1917)("secure" cipher)
- RNG: Random Number Generator(Pseudorandom Generators)
- $\Delta \leq \varepsilon$: Distinguish-er?

Symmetric Ciphers

Def:

- a cipher defined over is a pair of "efficient" $\mathbf{algs}(E, D)$ where
  - E is often randomized. $\rightarrow$ RNG
  - D is always deterministic.

$$\left.\begin{matrix}(\text{randomized})E(k,m)=c\\(\text{deterministic})D(k,c)=m\end{matrix}\right\}D(k,E(k,m))=m$$

For example, xor($\oplus$) is a function that satisfy the need above($f^{-1}=f$).

$$\begin{aligned}D(k,E(k,m)) &= D(k,k\oplus m)\\ &= k\oplus(k\oplus m)\\ &= (k\oplus k)\oplus m\\ &= 0\oplus m = m\end{aligned}$$

Given a message(m) and its OTP encryption(c).

$\rightarrow \text{the key is } k = m \oplus c.$

Fast enc/dec...but long keys...

Secure Cipher--Information Theoretic Security

Attacker's abilities: CT only attack.

Shannon: CT should reveal no "info" about PT

Def: $\text{A cipher } (E, D) \text{ over } (K, M, C) \text{ has } \textbf{perfect secrecy} \text{ if}$
$\forall m_0, m_1 \in M \ (|m_0| = |m_1|) \text{ and } \forall c \in C$
$Pr[E(k, m_0) = c] = Pr[E(k, m_1) = c] \text{ where } k \leftarrow K.$

so that OTP has perfect secrecy. -- Lemma

Eg: Let $m \in M$ and $c \in C$. How many OTP keys map m to c? -- One.

$$\begin{cases} |m_i| = |c_i| = |k_i| (\text{Important}) \\ m_0 \neq m_1 \\ E(m_0, k_0) = c_0 \end{cases} \Rightarrow E(m_1, k_0) \neq c_0$$

$$\begin{aligned} &\text{For OTP:} \\ &\forall m, c, \text{if } E(k, m) = c \\ &\Rightarrow k \oplus m = c \Rightarrow k = m \oplus c \\ &\Rightarrow \#\{k \in K : E(k, m) = c\} = 1 \\ &\Rightarrow \text{OTP has perfect secrecy.} \end{aligned}$$

However, in order to achieve **perfect secrecy**, the length of key has to be greater than or equal to the length of message, which means that:

$$|\mathcal{K}| \geq |\mathcal{M}|$$

has to be satisfied and this is hard to be used in practice.

Stream Ciphers: making OTP practical using "pseudorandom" key rather than "random" key

课题：可被计算机产生且不可预测的真随机源。

PRG is a function: $G : \{0,1\}^{\text{Seed space}} \rightarrow \{0,1\}^n \quad (n >> s)$. It is efficient for it's computable by a **deterministic** algorithm.

$$\begin{aligned} c &= E(k, m) = m \oplus G(k) \\ m &= D(k, c) = c \oplus G(k) \end{aligned}$$

A stream cipher can not have perfet secrecy since the key is shorter than the message...

So that we need a different definition of security, which will depend on specific PRG. This means that **PRG must be unpredictable.**

(Suppose PRG is predictable?)

$$\exists i:\ G(k)\Big|_{1,\dots,i} \overset{alg}{\to} G(k)\Big|_{i+1,\dots,n}$$

$$\Rightarrow c\Big|_{1,n} \oplus m\Big|_{1,\dots,i} = G(k)\Big|_{1,\dots,i} \overset{alg}{\to} G(k)\Big|_{i+1,\dots,n}$$

$$\Rightarrow G(k)\Big|_{i+1,\dots,n} \oplus c\Big|_{1,\dots,n} = m\Big|_{i+1,\dots,n}$$

Even $G(k)\Big|_{1,\dots,i} \overset{alg}{\to} G(k)\Big|_{i+1,\dots,n}$ is a problem!!!

$G: K \to \{0,1\}^n$ is **predictable** if:

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

$$\underset{k \leftarrow K}{Pr}\left[A(G(k))\Big|_{1,\dots,i} = G(k)\Big|_{i+1}\right] > \frac{1}{2} + \varepsilon$$

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

ε 由计算资源决定……

Def: PRG is **unpredictable** if it is not predictable
$\Rightarrow \forall i$: no "eff" adv. can predict bit$(i+1)$ for "non-neg" ε

Weak PRG:

e.g.:

```
glibc random():
    r[i] <- (r[i - 3] + r[i - 31]) % pow(2, 32)
    output r[i] >> 1
```

Never use `random()` for crypto!!! (e.g. Kereros V4)

Negligible VS Non-negligible

- **In practice:** ε is a scalar and

- ε non-neg: $\geq 2^{-30}$ (likely to happen over 1 GB of data)
- ε negligible: $\leq 2^{-80}$ (won't happen over life of key)

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

("poly" means polynomial?)

PRGs: The Rigorous Theory View(严格的理论角度？)

PRG的安全性为参数 λ 控制，λ 越大，安全性越高。

种子长度和输出密文长度与 λ 同增。

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

$$G_\lambda : K_\lambda \to \{0,1\}^{n(\lambda)} (\text{often ignore } \lambda)$$

Example of asymptotic definition:

We say that $G_\lambda \to \{0,1\}^{n(\lambda)}$ is **predictable** at postion i if:

there exists a **polynomial** time (in λ) algorithm A s.t.

$$\underset{k \leftarrow K_\lambda}{Pr}\left[A(\lambda, G_\lambda(k)\Big|_{1,\ldots,i}) = G_\lambda(k)\Big|_{i+1}\right] > \frac{1}{2} + \varepsilon(\lambda)$$

for some **non-negligibe** function $\varepsilon(\lambda)$.

Review

OTP(such as XOR) is "Perfect Secrety" under the circumstance of $|k| \geq |m|$, which means key-len must be no shorter than message-len. However in practical, this isn't possible to be satisfied, so we use PRG to generate a key.

Though key-len is shorter than message-len, we use PRG to expand the key to $G(k)$ that has a length that is equal to, or even longer than the length of message. But is this $G(k)$ unpredictable so that the stream cipher is security? This is important when we generate the same $G(k)$ and give it to a sender and a reciever in real world.

Therefore, based on probability's theory, we came to a conclusion that if there's a probability(Pr) that the $(i+1)$-th bit of $G(k)$ can be calculated using an Algorithm A, then this Pr must be no less than the sum of $\frac{1}{2}$ and a non-neg ε. Presented in formula:

$$\Delta \leq \frac{1}{2} + \varepsilon \text{ where } \Delta = 1 - \underset{k \leftarrow K_\lambda}{Pr}\left[A(\lambda, G_\lambda(k)\Big|_{1,\ldots,i}) = G_\lambda(k)\Big|_{i+1}\right]$$

Attack on OTP and Stream Ciphers

1. Two-Time Pad is Insecure

Never use stream cipher key more than once!!!

$$\left.\begin{array}{l} c_1 \leftarrow m_1 \oplus PRG(k) \\ c_2 \leftarrow m_2 \oplus PRG(k) \end{array}\right\} \Rightarrow c_1 \oplus c_2 \rightarrow m_1 \oplus m_2 \rightarrow$$
$$\text{redundancy and ASCII encoding} \rightarrow m_1, m_2$$

Real world examples:

- Project Venona
- MS-PPTP(windows NT) different keys for C $\rightarrow$ S and S $\rightarrow$ C
- 802.11b WEP: Avoid Related Keys(Use pseudorandom key for each frame, better, use stronger encryption method(as in WPA2))
- Disk Encryption: How do you edit a part of the message with only the cipher...

What's above mainly discussed about **Network Traffic** and **Disk Encryption**. For the former, one solution is to negotiate new key for every session(e.g. TLS). For the latter, typically don't use a stream cipher!

2. No Integrity(OTP is malleable易受影响的)

Modifications to ciphertext are undetected and have **predictable** impact on plaintext.

随机序列的安全性

长周期性、非线性、统计上的预期性、可伸缩性、不可预测性

长周期性：有线长度的数字序列一定可重复（有周期）。希望长度为 n 的二元序列重复周期能够等于 2^n。n 足够大时序列重复周期长，使得密钥空间大。
非线性：基本要求（线性问题更容易计算，更容易受到基于数学的攻击），并且具有足够的复杂度。
统计上的预期性：比如0和1频率接近相等，0、1游程数量接近相等。
可伸缩性：序列随机 $\Leftrightarrow$ 任何子序列随机。
不可预测性：人们不能人为有意识地重复产生该随机序列（真伪）；不能由已知的序列数据求出未知的序列数据（随机？）。
真随机性和伪随机性：真随机序列具有上述两方面的不可预测性

1. 单比特频率检测
2. 游程分布检测
3. 自相关检测

随机序列的产生

1. 数学算法（非线性）（不是真随机）
2. 基于物理学（如电子噪声）（真随机）（统计随机特性不够好）
3. 结合（真随机+伪随机）

线性移位寄存器序列密码

$$S' = HS \pmod 2$$

$$H = \begin{pmatrix} 0, & 1, & 0, & \dots, & 0 \\ 0, & 0, & 1, & \dots, & 0 \\ 0, & 0, & 0, & \dots, & 0 \\ \vdots, & \vdots, & \vdots, & \ddots, & \vdots \\ 0, & 0, & 0, & \dots, & 1 \\ g_0, & g_1, & g_2, & \dots, & g_{n-1} \end{pmatrix},$$

$$S' = \begin{pmatrix} s'_0 \\ s'_1 \\ \vdots \\ s'_{n-1} \end{pmatrix}, S = \begin{pmatrix} s_0 \\ s_1 \\ \vdots \\ s_{n-1} \end{pmatrix},$$

矩阵 H 称为连接多项式的伴侣矩阵。称为连接多项式的伴侣矩阵。

……

非线性序列密码

- 非线性移位寄存器序列。
- 对线性移位寄存器序列进行非线性组合对线性移位寄存器序列进行非线性组合。
- 利用非线性分组码产生非线性序列利用非线性分组码产生非线性序列。