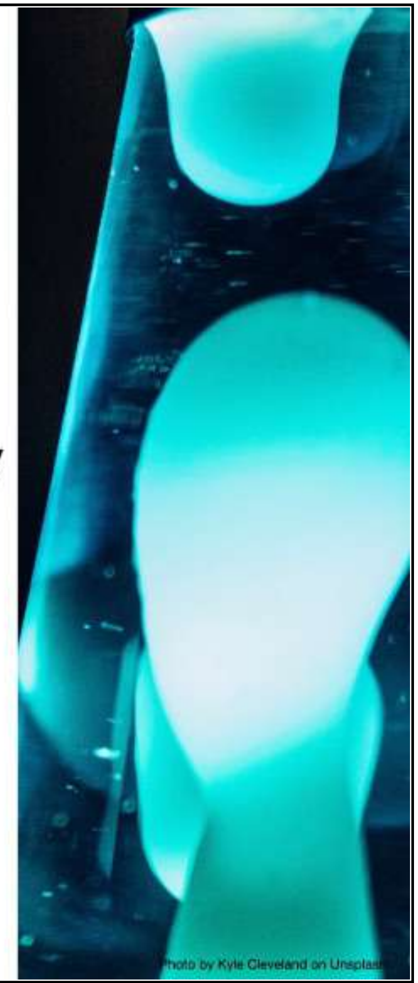
Randomness in Cryptography

Vanesa Daza



Fundamentals Principles of Modern Cryptography

1. Security depends on the resources of the attacker.
2. Kerckhoffs Principle
3. Security is impossible without randomness

Photo by Solen Feyissa on Unsplash

The One-Time Pad

Quick Review

(KeyGen, Enc, Dec)

$$\frac{\text{KeyGen:}}{\begin{array}{c} k \leftarrow \{0,1\}^\lambda \\ \text{return } k \end{array}} \qquad \frac{\text{Enc}_k(m)\text{:}}{\text{return } k \oplus m} \qquad \frac{\text{Dec}_k(c)\text{:}}{\text{return } k \oplus c}$$

- Perfectly Secure (unconditional security)
- Computationally Efficient
- Key too long and only used one time
- Malleable

crucial point: keys are uniformly selected at random, and true randomness is expensive.

How to make the OTP practical?

PRNG

basic idea : randomness → pseudo-randomness

Definition 2.2 *A* pseudorandom number generator (PRNG) *is a function*

$$G : \{0,1\}^{\ell} \to \{0,1\}^{h}$$

such that no efficient adversary can distinguish the output distribution of G from the uniform distribution on $\{0,1\}^h$. → for practical purposes, the output is considered uniformly random in $\{0,1\}^h$.

$s_0 = \text{seed}$

$s_i = G(s_{i-1})$, $i = 0,1,2,\dots$ or in general : $s_{i+1} = G(s_i, s_{i-1}, \dots, s_{i-t})$ for some t

Photo by Eduardo Soares on Unsplash

PRNG Example

Linear Congruential Generator

$$s_0 = \text{seed}$$
$$s_{i+1} \equiv a s_i + b \bmod m, \quad i = 0, 1, \ldots$$

integer constants (secret)

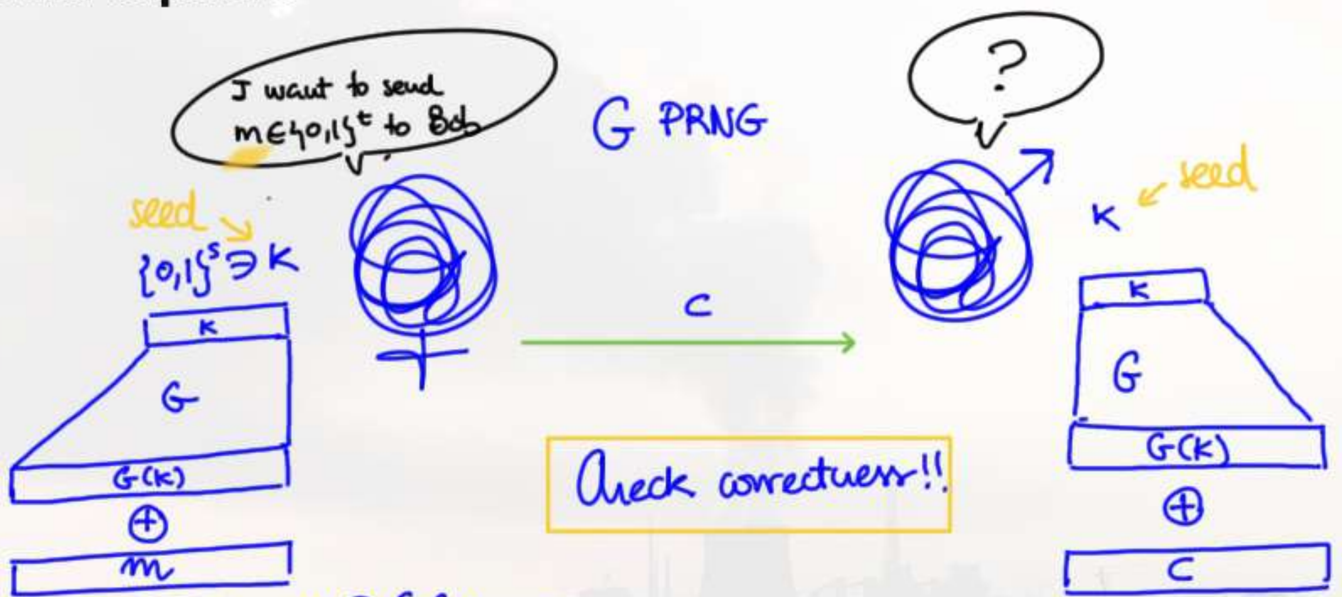
- produce a sequence of random looking integers between 0 and m-1
- choice of a, b and m is very important to guarantee good statistical properties

rand() function used in ANSI C

$$s_0 = 12345$$
$$s_{i+1} \equiv 1103515245 s_i + 12345 \bmod 2^{31}, \quad i = 0, 1, \ldots$$

How to make the OTP practical

Stream Ciphers



$Enc_K(m) = m \oplus G(K) = c$

$Dec_K(c) = c \oplus G(K)$

Photo by Hassan Afridhi on Unsplash

<u>Example</u>: Linear feedback shift registers (LFSR)

$S_0 = K = k_1 \dots k_\ell$

$k_i = p_1 k_{i-1} \oplus \dots \oplus p_\ell \, k_{i-\ell}$

$p_i \in \{0,1\}$

$\ell = 3, \; p_1 = 1, \; p_2 = 0, \; p_3 = 1$

$K = 011$

$k_4 = 1 \cdot 1 + 0 \cdot 1 + 1 \cdot 0 = 1$

$k_5 = 1 \cdot 1 + 0 \cdot 1 + 1 \cdot 1 = 0$

$k_6 = 1 \cdot 0 + 0 \cdot 1 + 1 \cdot 1 = 1$

$\vdots$

But, the output of an LSFR of length ℓ repeats periodically, with a period of at most $2^\ell - 1$ $\longrightarrow$ not suitable for cryptography.

Still, a clever combination of some LFSR remains secure !! -

Stream Ciphers

Security

Length key <<< Length Plaintext → no perfect secrecy!!

We need to define what security means, and it will depend on PRNG

⊚ unpredictability

Photo by Erol Ahmed on Unsplash

Predictability of PRNG

Informally, a PRNG is *predictable* if there exists an efficient way to predict a bit from previous computed bits.

$\exists i \quad G(k)|_{1,\dots,i} \xrightarrow{A} G(k)|_{i+1,\dots,n}$

Observe that if PRNG G is predictable, then the corresponding stream cipher is insecure. (known plaintext attack)

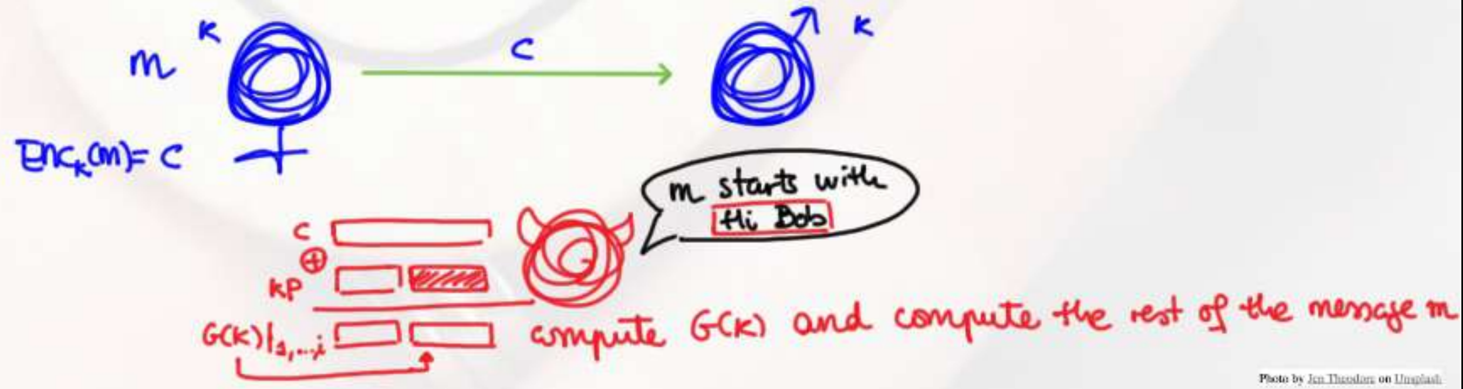


Photo by Jen Theodore on Unsplash

Can we apply the 2-time attack to Stream Ciphers?

$$\left.\begin{array}{l} c_1 = m_1 \oplus G(k) \\ c_2 = m_2 \oplus G(k) \end{array}\right\} \longrightarrow m_1, m_2 ?$$

Observe that $c_1 \oplus c_2 = m_1 \oplus c_2$ and as it happened with the OTP, this could leak information on m_1 and m_2.

wired equivalent privacy

Attacking the WEP protocol (simplified version)

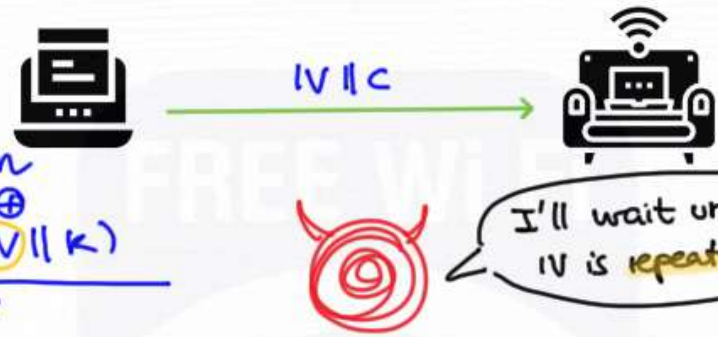
802.11b.WEP

G PRNG function called RC4

k $\qquad$ k

in $\{0,1\}^{24}$

$$\frac{m \oplus G(IV \| k)}{c}$$



after 2^{24} frames reset to 0

$$\left.\begin{array}{l} IV \| c_1 \\ IV \| c_2 \end{array}\right\} \rightarrow c_1 \oplus c_2 = m_1 \oplus G(IV \| k) \oplus m_2 \oplus G(IV \| k) =$$
$$= m_1 \oplus m_2 \text{ as in a 2-time attack!!}$$

More efficient attacks were found because of the closeness of the keys: $k_1 = IV_1 \| k$, $k_2 = IV_2 \| k$ =!!

Are the Stream Ciphers malleable?

Similarly as it happens in the OTP, stream ciphers are malleable. Indeed just noticed that

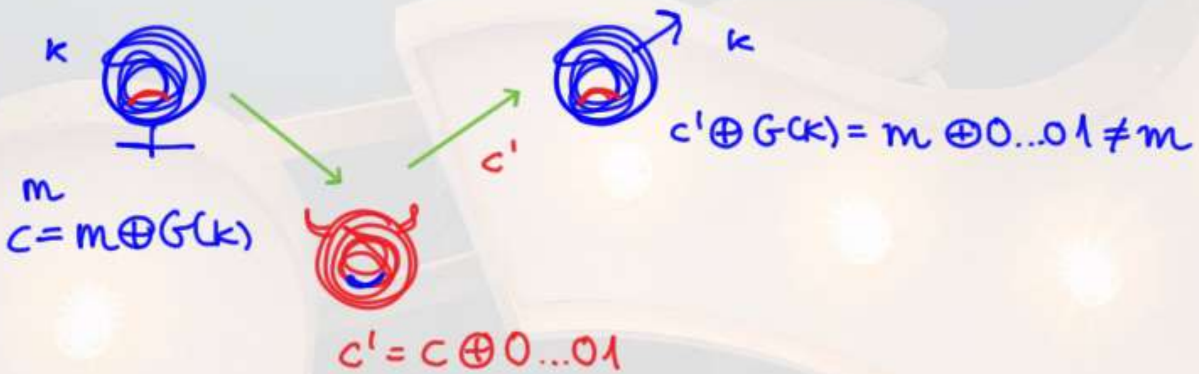
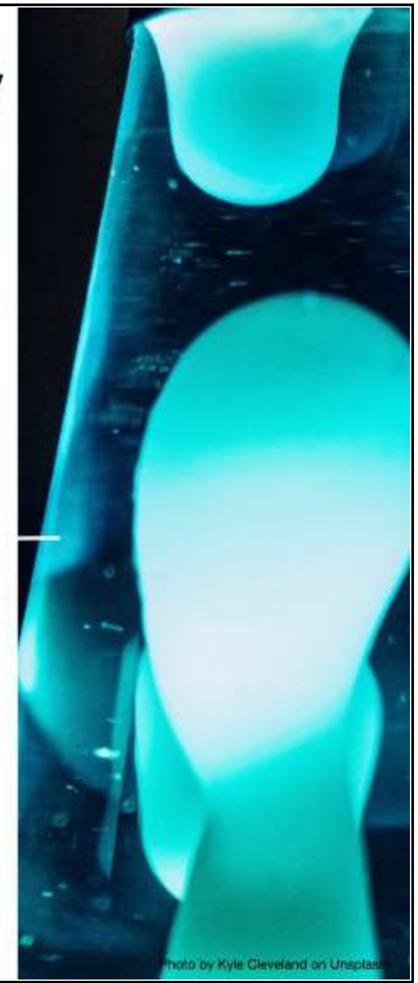


Photo by Jon Tyson on Unsplash


TOUR OF ACCOUNTING
OVER HERE WE HAVE OUR RANDOM NUMBER GENERATOR.
NINE NINE NINE NINE NINE NINE
10/25/01 © 2001 United Feature Syndicate, Inc.
ARE YOU SURE THAT'S RANDOM?
THAT'S THE PROBLEM WITH RANDOMNESS: YOU CAN NEVER BE SURE.

Randomness in Cryptography



Vanesa Daza