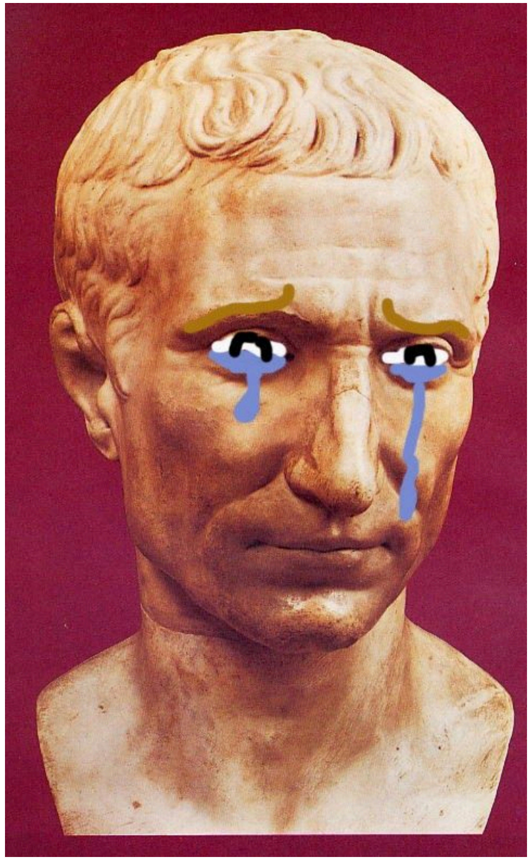
Cryptography
CS 555

Topic 3: Perfect Secrecy

Recap

- Caesar Cipher, Shift Cipher, Substitution Cipher, Vigenere Cipher

- All historical ciphers have fallen



Perfect Secrecy Intuition

- Regardless of information an attacker *already* has, a ciphertext should leak no *additional information* about the underlying plaintext.

- We will formalize this intuition
  - And show how to achieve it

Private Key Encryption Syntax

- Message Space: $\mathcal{M}$
- Key Space: $\mathcal{K}$
- Three Algorithms
  - $\mathrm{Gen}(R)$ (Key-generation algorithm)
    - Input: Random Bits R
    - Output: Secret key $\mathrm{k} \in \mathcal{K}$
  - $\mathrm{Enc_k}(m)$ (Encryption algorithm)
    - Input: Secret key $\mathrm{k} \in \mathcal{K}$ and message $\mathrm{m} \in \mathcal{M}$
    - Output: ciphertext c
  - $\mathrm{Dec_k}(c)$ (Decryption algorithm)
    - Input: Secret key $\mathrm{k} \in \mathcal{K}$ and a ciphertex c
    - Output: a plaintext message $\mathrm{m} \in \mathcal{M}$

- Invariant: $\mathrm{Dec_k}(\mathrm{Enc_k}(m))=m$

Typically picks $\mathrm{k} \in \mathcal{K}$ uniformly at random

Trusted Parties (e.g., Alice and Bob) must run Gen in advance to obtain secret k.

Assumption: Adversary does not get to see output of Gen

An Example

- Enemy knows that Caesar likes to fight in the rain and it is raining today

$$\Pr[m = wait] = 0.3$$
$$\Pr[m = attack] = 0.7$$

- Suppose that Caesar sends c=Enc_K(m) to generals and that the attacker calculates

$$\Pr[m = wait \,|\text{c=EncK(m)}] = 0.2$$
$$\Pr[m = attack \,|\text{c=EncK(m)}] = 0.8$$

- Did the attacker learn anything useful?

Perfect Secrecy

Definition 1: An encryption scheme (Gen, Enc, Dec) with message space $\mathcal{M}$ is perfectly secret if for *every* probability distribution over $\mathcal{M}$, every message $\mathrm{m} \in \mathcal{M}$ and every ciphertext $\mathrm{c} \in \mathcal{C}$ for which $\Pr[C = c] > 0$:

$$\Pr[M = m | C = c] = \Pr[M = m].$$

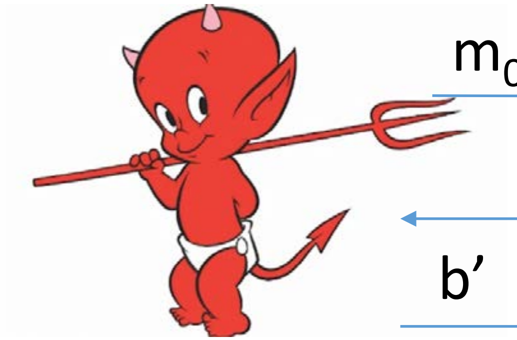
Definition 2: For every $\mathrm{m}, \mathrm{m}' \in \mathcal{M}$ and $\mathrm{c} \in \mathcal{C}$

$$\Pr[\mathrm{Enc}_{\mathrm{K}}(m) = c] = \Pr[\mathrm{Enc}_{\mathrm{K}}(m') = c].$$

(where the probabilities are taken over the randomness of Gen and Enc)

Lemma 2.4: The above definitions are equivalent.

Another Equivalent Definition (Game)



m_0, m_1 →

← **c**

b' →

Random bit b

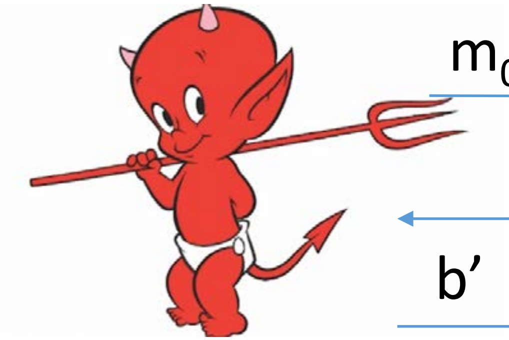
K = Gen(.)

$c = Enc_K(m_b)$

$\forall$ 

$\Pr\Big[$  $\mathit{Guesses}\ b' = b\Big] = \dfrac{1}{2}$

Another Equivalent Definition (Game)



m_0, m_1 →

← **c**

b' →

Random bit b
K = Gen(.)
$c = Enc_K(m_b)$

Suppose we have m,m',c' s.t. $Pr[Enc_K(m)= c'] > Pr[Enc_K(m')=c']$ then the adversary can win the game w.p > ½. How?

What else do we need to establish to prove that the definitions are equivalent?

One Time Pad [Vernam 1917]

$$\mathrm{Enc}_K(m) = K \oplus m \qquad\qquad \mathrm{Dec}_K(c) = K \oplus c$$

$$\mathbf{Example} = \mathbf{1011 \oplus 0011 = ???}$$

Theorem: The one-time pad encryption scheme is perfectly secret

The following calculation holds for any c, m

$$\Pr[\mathrm{Enc}_K(m)=c] = \Pr[K \oplus m = c] = \Pr[K = c \oplus m] = {}^{1}/_{|\mathcal{K}|}.$$

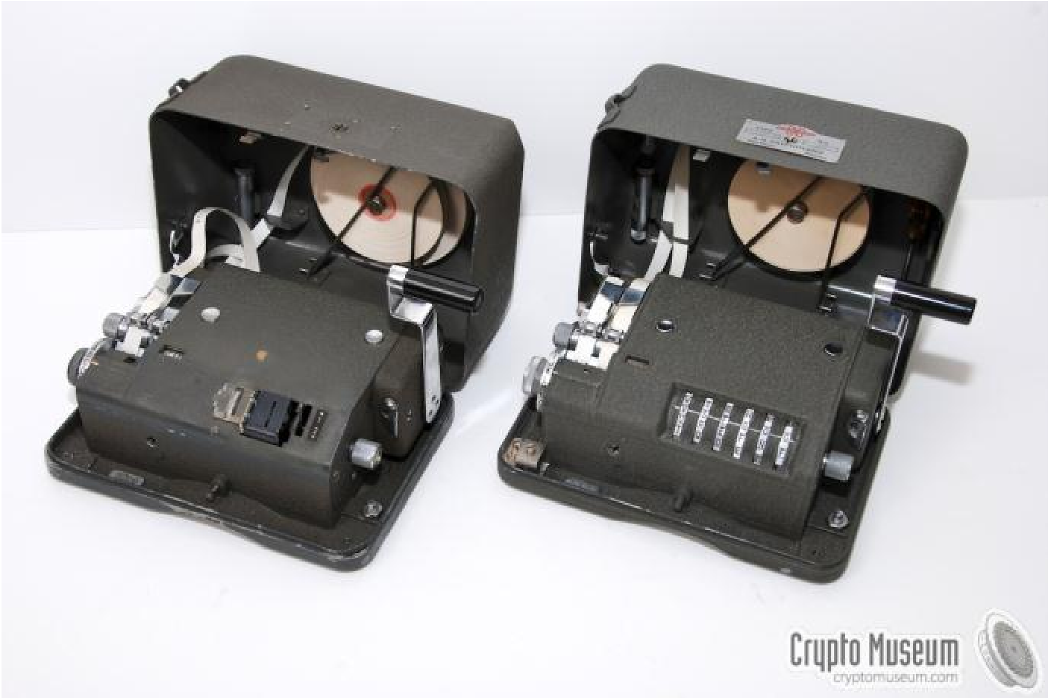
Thus, for any m, m’, c we have

$$\Pr[\mathrm{Enc}_K(m)=c] = {}^{1}/_{|\mathcal{K}|} = \Pr[\mathrm{Enc}_K(m')=c].$$

One Time Pad [Vernam 1917]

$$\mathrm{Enc}_K(m) = K \oplus m \qquad \mathrm{Dec}_K(c) = K \oplus c$$

$$\mathbf{Example} = \mathbf{1011 \oplus 0011 = ???}$$



One Time Pad

Perfect Secrecy Limitations

Theorem: If (Gen,Enc,Dec) is a perfectly secret encryption scheme then

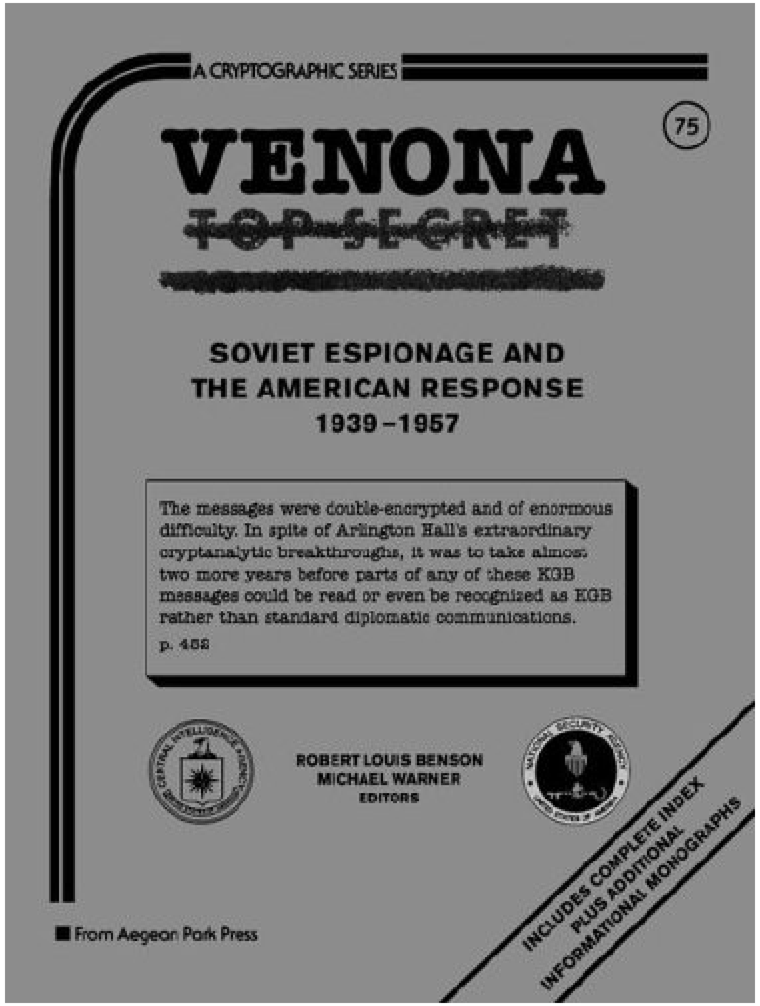
$$|\mathcal{K}| \geq |\mathcal{M}|$$

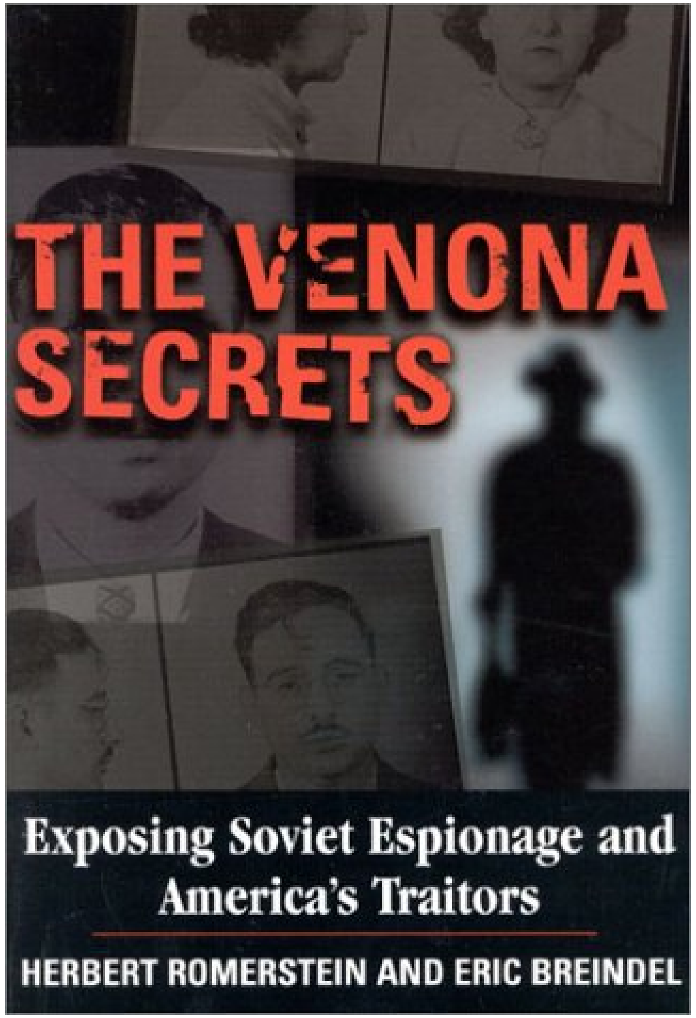
One Time Pad Limitations

- The key is as long as the message
  - How to exchange long messages?
  - Need to exchange/secure lots of one-time pads!
- OTPs can only be used once
  - As the name suggests
- VENONA project (US + UK)
  - Decrypt ciphertexts sent by Soviet Union which were mistakenly encrypted with portions of the same one-time pad over several decades

$$c \oplus c' = (m \oplus k) \oplus (m' \oplus k) = m \oplus m'$$

VENONA project





Shannon's Theorem

Theorem: Let (Gen,Enc,Dec) be an encryption scheme with $|\mathcal{K}| = |\mathcal{M}| = |\mathcal{C}|$. Then the scheme is perfectly secret if and only if:

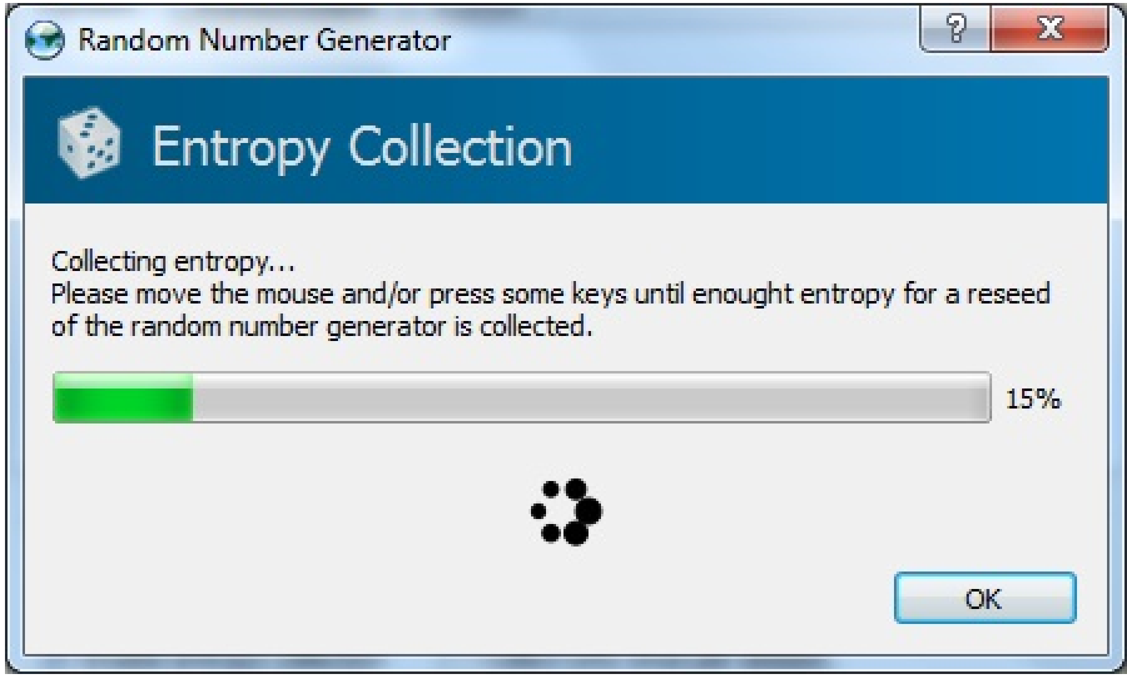
1. Every key k $\in \mathcal{K}$ is chosen with (equal) probability $^1/_{|\mathcal{K}|}$ by the algorithm Gen, and
2. For every $\mathrm{m} \in \mathcal{M}$ and every $\mathrm{c} \in \mathcal{C}$ there exists a unique key k $\in \mathcal{K}$ such that $\mathrm{Enc}_k(\mathrm{m})=\mathrm{c}$.

An Important Remark on Randomness

- In our analysis we have made (and will continue to make) a key assumption:
- We have access to true “randomness” to generate the one time pad K
- Independent Random Bits
  - Unbiased Coin flips
  - Radioactive decay?

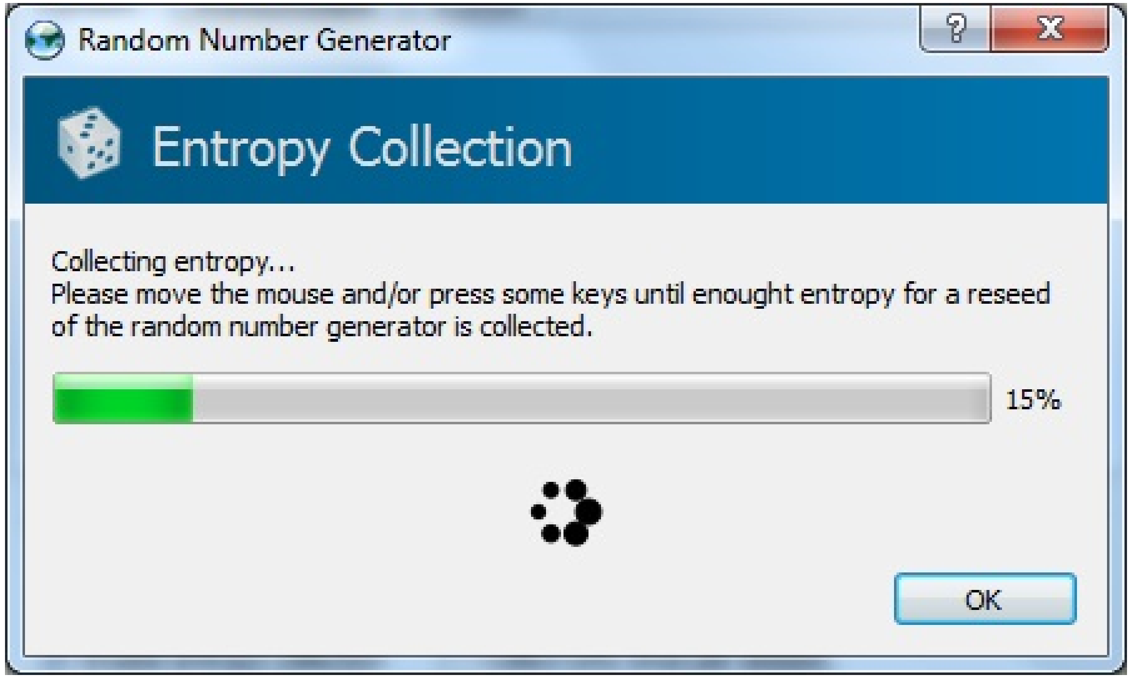
In Practice

- Hard to flip thousands/millions of coins
- Mouse-movements/keys
  - Uniform bits?
  - Independent bits?
- Use Randomness Extractors
  - As long as input has high entropy, we can extract (almost) uniform/independent bits
  - Hot research topic in theory



In Practice

- Hard to flip thousands/millions of coins
- Mouse-movements/keys
- Customized Randomness Chip?



Caveat: Don’t do this!

- Rand() in C stdlib.h is no good for cryptographic applications

- Source of many real

world flaws

Coming Up…

- MLK Day (No Class)

- Before Next Class (Wednesday)
  - Read: Katz and Lindell 3.1-3.2