One-time pad\[ \text{keys} = M = C = \{0, 1\}^d \quad (\text{there are } 2^d \text{ keys}) \]

**Thm:** The one-time pad has perfect secrecy

**Problem:** The one-time pad is not practical

**Proof:**
\[
E(k, m) = k \oplus m = c \\
D(k, c) = k \oplus c
\]

**Goal:** Prove that \( Pr_k [E(k, m) = c] = Pr_k [E(k, m') = c] \) for all \( m, m', c \)

Suppose \( E(k, m) = c \)

then \( k \oplus m = c \)

\( k = m \oplus c \)

so, \( Pr_k [E(k, m) = c] = \frac{1}{2^d} \)

This is also true for \( m' \)

**Thm:** Every encryption scheme \( E: \text{keys} \times M \rightarrow C \) with perfect secrecy has \( |\text{keys}| \geq |M| \)

**Proof:** Fix \( c \in C \) s.t. \( c \in \text{Range}(E) \)

Let \( D = \{ D(k, c) : k \in \text{keys} \} \)

then \( |D| \geq |\text{keys}| \)

Also, \( \exists k, m \in M: E(k, m) = c \)

so, \( Pr_k [E(k, m) = c] \leq \frac{1}{2^d} \)

By perfect secrecy,
\[
Pr_k [E(k, m) = c] = 0 \\
\forall m' \in M
\]
Hence, $D = M$

So, $|M| = |D| = |\text{keys}|$

Conclusion: Perfect secrecy is impractical

**Stream Ciphers**

$\text{keys} = \{0, 1\}^k$ -- $G$ is a deterministic

$M = C = \{0, 1\}^n$ -- $n$-bit number generator

$\text{key} \rightarrow G \rightarrow \text{key}$

$\oplus$

plaintext $\rightarrow$ cipher-text $\rightarrow$

$G$:  
- is deterministic
- must have bigger output than input

Example:

unsigned int state;  
int rand(void) {
    state = 322349 * state + 45656749;
    return state % 2;
}  

A. E.
even → odd → even → odd →...

So, the output of rand() is one of
\[ \{01010101... ; 10101010... \} \]

Example: Linear Feedback Shift Register (LFSR)

\[ \begin{array}{cccccc}
X_0 & X_1 & X_2 & X_3 & X_4 & output \\
\hline
100110 & \rightarrow 1 \\
110011 & \rightarrow 1 \\
\end{array} \]

\[ 111101 \]

Attack on LFSR

output \( b_n = x_2 \oplus x_4 \)

\( b_0 = x_1 \oplus x_3 \)

\( b_{n+1} = x_0 \oplus x_n \)

\( b_2 = (x_0 \oplus x_2) \oplus x_3 \)

\( b_3 = (x_0 \oplus x_3) \oplus x_2 \)

\( b_4 = (x_0 \oplus x_4) \oplus x_3 \)

\( b_5 = (x_0 \oplus x_5) \oplus (x_3 \oplus x_5) \)

can recover key by solving linear equations

solution: \( O(n^3) \)

What do we want from a good PRG?

- look random
- be uniformly distributed
- can't compute key from output
- can't predict future outputs from output
- can't compute past outputs from output

Consider scenario where the sender sends either

"Let's attack."

"Don't attack."

Encrypted

Adversary sees: \( c = G(r) \oplus m \).

So, attacker computes

\[ S_1 = c \oplus m, \]

\[ S_2 = c \oplus m. \]

Suppose \( G \) always outputs a prime number

Suppose \( S_1 = 17 \ast 3042 \)

\[ \Rightarrow \text{Sender sent } m. \]

Goal: output of \( G \) is indistinguishable from randomly chosen bit string.

Distinguishability

A distribution \( D \) on set \( S \) is a function

\[ D : S \rightarrow [0,1] \]

where \( D(x) \) is the probability of choosing \( x \)

We say \( X \leftarrow D \) to mean \( x \) is chosen according to \( D \).
Ex. The uniform distribution $U$ on $\mathbb{Z}_1, \ldots, 10^3$

$P_r [X \in U; X = 4] = \frac{1}{10^3}$

Ex. Let $B$ be the distribution of the number of heads after flipping 2 coins

$P_r [X \in B; X = 0] = \frac{1}{4}$
$P_r [X \in B; X = 1] = \frac{1}{2}$