Symmetric-Key Encryption: constructions

Lecture 4

PRG, Stream Cipher

Story So Far

- We defined (passive) security of Symmetric Key Encryption (SKE)
  - **SIM-CPA** = **IND-CPA** + **almost perfect correctness**
    - Restricts to PPT entities
    - Allows negligible advantage to the adversary
- Today: Constructing one-time SKE from Pseudorandomness
- Next time:
  - Pseudorandomness from One-Way Permutations
  - Multi-message SKE

Constructing SKE schemes

- Basic idea: “stretchable” pseudo-random one-time pads (kept compressed in the key)
  - (Will also need a mechanism to ensure that the same piece of the one-time pad is not used more than once)
- Approach used in practice today: complex functions which are conjectured to have the requisite pseudo-randomness properties (stream-ciphers, block-ciphers)
- Theoretical Constructions: Security relies on certain computational hardness assumptions related to simple functions

Pseudorandomness Generator (PRG)

- Expand a short random **seed** to a "random-looking" string
- First, PRG with fixed stretch: $G_k: \{0,1\}^k \rightarrow \{0,1\}^{n(k)}$, $n(k) > k$
- How does one define random-looking?
  - Next-Bit Unpredictability: PPT adversary can't predict i^{th} bit of a sample from its first $(i-1)$ bits (for every $i \in \{0,1,\ldots,n-1\}$)
  - A "more correct" definition:
    - PPT adversary can't distinguish between a sample from $\{G_k(x)\}_{x \leftarrow \{0,1\}^k}$ and one from $\{0,1\}^{n(k)}$
  - **Turns out they are equivalent!**

$$| \Pr_{y \leftarrow PRG}[A(y)=0] - \Pr_{y \leftarrow rand}[A(y)=0] |$$

is negligible for all PPT A

Coming up

Recall

Computational Indistinguishability

- Two distribution ensembles $\{X_k\}$ and $\{X'_k\}$ are said to be **computationally indistinguishable** if
  - $\forall$ (non-uniform) PPT distinguisher D, $\exists$ negligible $\nu(k)$ such that $|\ Pr_{x \leftarrow X_k}[D(x)=1] - Pr_{x \leftarrow X'_k}[D(x)=1]\ | \leq \nu(k)$

$X_k \approx X'_k$

- cf.: Two distribution ensembles $\{X_k\}$ and $\{X'_k\}$ are said to be **statistically indistinguishable** if $\forall$ functions T, $\exists$ negligible $\nu(k)$ s.t. $|\ Pr_{x \leftarrow X_k}[T(x)=1] - Pr_{x \leftarrow X'_k}[T(x)=1]\ | \leq \nu(k)$
  - Equivalently, $\exists$ negligible $\nu(k)$ s.t. $\Delta(X_k,X'_k) \leq \nu(k)$ where $\Delta(X_k,X'_k) := \max_T |\ Pr_{x \leftarrow X_k}[T(x)=1] - Pr_{x \leftarrow X'_k}[T(x)=1]\ |$

Pseudorandomness Generator (PRG)

- Takes a short seed and (deterministically) outputs a long string
  - $G_k: \{0,1\}^k \rightarrow \{0,1\}^{n(k)}$ where $n(k) > k$
- Security definition: Output distribution induced by random input seed should be "pseudorandom"
  - i.e., Computationally indistinguishable from uniformly random
  - $\{G_k(x)\}_{x \leftarrow \{0,1\}^k} \approx U_{n(k)}$
  - Note: $\{G_k(x)\}_{x \leftarrow \{0,1\}^k}$ cannot be statistically indistinguishable from $U_{n(k)}$ unless $n(k) \leq k$ (Exercise)
    - i.e., no PRG against unbounded adversaries

Equivalent definitions

$| \Pr_{y \leftarrow PRG}[B(y_1^{i-1}) = y_i] - ½ |$ is negligible for all i, all PPT B

$| \Pr_{y \leftarrow PRG}[A(y)=0] - \Pr_{y \leftarrow rand}[A(y)=0] |$ is negligible for all PPT A

- Next-Bit Unpredictable ⇔ Pseudorandom
- Pseudorandom ⇒ NBU:
  - Reduction: Given a PPT adversary B (for NBU), will show how to turn it into a PPT adversary A (for Pseudorandomness) with similar advantage. Hence the advantage must be negligible.

    Could be seen as showing the contrapositive: ¬NBU ⇒ ¬Pseudorandom
  - For any PPT B and i, consider PPT A which uses it to predict ith bit and then checks if the prediction was correct
  - Formally, A(y) outputs $B(y_1^{i-1}) \oplus y_i$ (i as specified by B). Then:

    $| \Pr_{y \leftarrow PRG}[A(y)=0] - \Pr_{y \leftarrow rand}[A(y)=0] | = | \Pr_{y \leftarrow PRG}[B(y_1^{i-1}) = y_i] - ½ |$

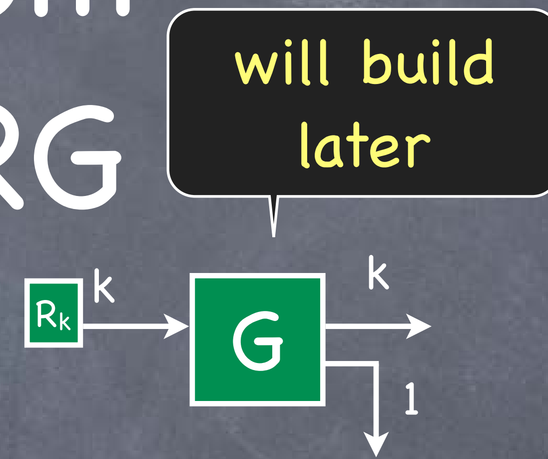
Equivalent definitions

$| Pr_{y \leftarrow PRG}[B(y_1^{i-1}) = y_i] - ½ |$ is negligible for all i, all PPT B

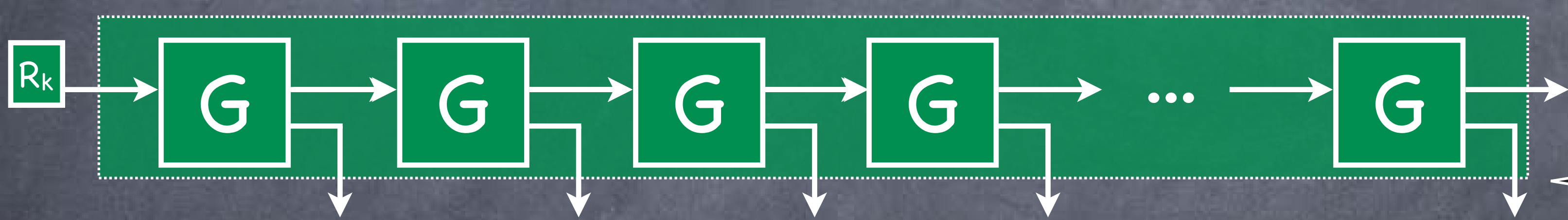
$| Pr_{y \leftarrow PRG}[A(y)=0] - Pr_{y \leftarrow rand}[A(y)=0] |$ is negligible for all PPT A

- Next-Bit Unpredictable ⇔ Pseudorandom
- NBU ⇒ Pseudorandom: Using a **Hybrid Argument**
  - Define distributions H_i over n-bit strings: $y \leftarrow$ PRG. Output $y_1^i \| r$ where r is n-i independent uniform bits. H_0 = rand, H_n = PRG.
  - NBU ⇒ $H_i \approx H_{i+1}$: Given a PPT distinguisher A, let PPT predictor B be as follows: On input $z \in \{0,1\}^{i-1}$, pick $b \leftarrow \{0,1\}$, $r \leftarrow \{0,1\}^{n-i}$ and output $A(z \| b \| r) \oplus b$. Then [Exercise] : $|Pr_{y \leftarrow PRG}[B(y_1^{i-1}) = y_i] - ½| = |Pr_{y \leftarrow H_i}[A(y)=0] - Pr_{y \leftarrow H_{i+1}}[A(y)=0]|$
  - Then [Exercise] : $H_0 \approx H_n$ (for n(k) that is polynomial)

General PRG from 1-Bit Stretch PRG



- One-bit stretch PRG, $G_k: \{0,1\}^k \rightarrow \{0,1\}^{k+1}$
- Increasing the stretch
  - Can use part of the PRG output as a new seed

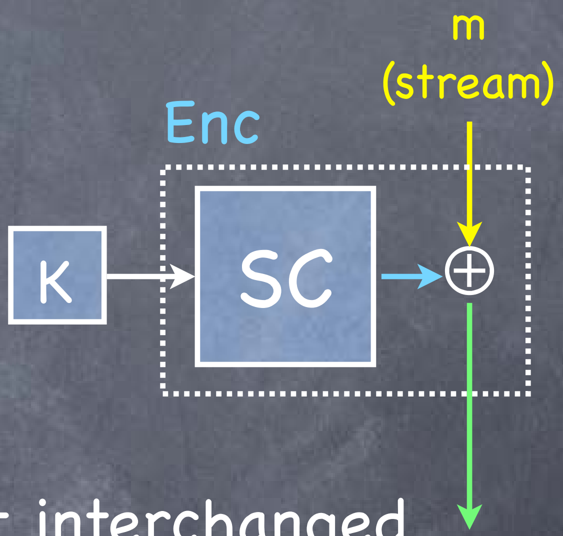


Why is this a PRG?

A "hybrid argument"

  - If intermediate seeds are never output, can keep stretching on demand (for any "polynomial length")
  - A stream cipher

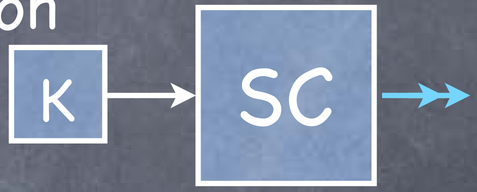
One-time CPA-secure SKE with a Stream-Cipher



- One-time Encryption with a **stream-cipher**:
  - Generate a one-time pad from a short seed
  - Can share just the seed as the key
  - Mask message with the pseudorandom pad
- Decryption is symmetric: plaintext & ciphertext interchanged
- SC can spit out bits on demand, so the message can arrive bit by bit, and the length of the message doesn't have to be a priori fixed
- Security: indistinguishability from using a truly random pad (coming up)

Stream Ciphers

- Stream ciphers in practice
  - Naturally useful for onetime (stream) encryption, in protocols where a key is established per session



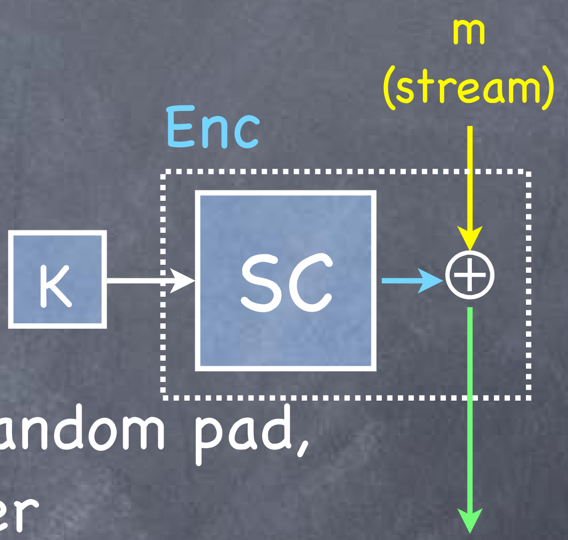
- Many popular candidates:
  - **RC4**: **Obsolete** (but popular). Designed in 1987. Leaked (and broken) in 1994. Still used in BitTorrent, and supported as an option in some protocols.
  - **eSTREAM** portfolio:

Profile 1 (software)	HC-128, Rabbit, Salsa20/12, SOSEMANUK	128 bit keys
Profile 2 (hardware)	Grain, MICKEY, Trivium	80 bit keys

  - **NIST** recommendation: AES in an appropriate mode (later)

One-time CPA-secure SKE with a Stream-Cipher

- In IDEAL experiment, consider simulator that uses a truly random string as the ciphertext
- To show REAL ≈ IDEAL
- Consider an intermediate world, HYBRID:
    - Like REAL, but Enc/Dec use a (long) truly random pad, instead of the output from the stream-cipher
    - HYBRID = IDEAL (recall perfect security of one-time pad)
    - Claim: REAL ≈ HYBRID
        - Consider the experiments as a system that accepts the pad from outside (R' = SC(K) for a random K, or truly random R) and outputs the environment's output. This system is PPT, and so can't distinguish pseudorandom from random.



One-time CPA-secure SKE with a Stream-Cipher