CSL759: Cryptography and Computer Security

Ragesh Jaiswal

CSE, IIT Delhi

Stream Ciphers: Indistinguishability Vs Unpredictability

- Definition ((t,ϵ)-indistinguishable PRG): A function $G:\{0,1\}^k \rightarrow \{0,1\}^n$ is said to be (t,ϵ)-secure Pseudorandom Generator if for all algorithms that run in time at most t, we have:

$$Adv_{PRG}(A,G) \leq \epsilon$$

- Definition ((t,ϵ)-unpredictable PRG): A function $G:\{0,1\}^k \rightarrow \{0,1\}^n$ is called (t,ϵ)-unpredictable pseudorandom generator of for all algorithms A that run in time at most t and for all $i \in \{1,\dots,n-1\}$, we have:
$\Pr[A(G(K)[1 \dots i]) = G(K)[i+1]] \leq \frac{1}{2} + \epsilon$.

- Theorem(indistinguishability implies unpredictability): Let $G:\{0,1\}^k \rightarrow \{0,1\}^n$. If G is a $(t+1,\epsilon)$-indistinguishable PRG, then G is also a (t,ϵ)-unpredictable PRG.
- Proof: We show the contrapositive.
  - Suppose A is an algorithm that runs in time at most t and i be the index such that
  $\Pr[A(G(K)[1 \dots i]) = G(K)[i+1]] > \frac{1}{2} + \epsilon$.
  - Consider algorithm $B(x)$: If $(A(x[1 \dots i]) = x[i+1])$, then output 1 else 0.

Stream Ciphers: Indistinguishability Vs Unpredictability

- Theorem(indistinguishability implies unpredictability): Let $G: \{0,1\}^k \rightarrow \{0,1\}^n$. If G is a (t, ϵ)-indistinguishable PRG, then G is also a (t, ϵ)-unpredictable PRG.
- Proof: We show the contrapositive.
  - Suppose A is an algorithm that runs in time at most t and i be the index such that

    $$\Pr\left[A(G(K)[1 \dots i]) = G(K)[i+1]\right] > \frac{1}{2} + \epsilon.$$

  - Consider algorithm $B(x)$: If $(A(x[1 \dots i]) = x[i+1])$, then output 1 else 0.
  - Claim 1: B runs in time $t + 1$.
  - Claim 2: $Adv_{PRG}(B, G) > \epsilon$.
    - Claim 2.1: $\Pr_{K \leftarrow \{0,1\}^k}\left[B\left(G(K)\right) = 1\right] > \frac{1}{2} + \epsilon$.
    - Claim 2.2: $\Pr_{R \leftarrow \{0,1\}^n}[B(R) = 1] = \frac{1}{2}$.

Stream Ciphers: Indistinguishability Vs Unpredictability

- Theorem(unpredictability implies indistinguishability): Let $G: \{0,1\}^k \rightarrow \{0,1\}^n$. If G is a $(t', \epsilon/n)$-unpredictable PRG, then G is also a (t, ϵ)-indistinguishable PRG.
  - This is known as Yao's theorem and is proved using an idea known as the *Hybrid* argument. We will discuss this proof in some time.

Stream Ciphers: Pseudorandom generator

- What are the desirable properties of a pseudorandom generator $G: \{0,1\}^k \rightarrow \{0,1\}^n$ for Cryptographic purposes:
  - Stretch: $n > k$
  - Efficient: G should be efficiently computable.
  - Indistinguishability: No bounded resource algorithm should be able to distinguish the output of the generator $G(x)$ (for random x) from a random n bit string.
  - Unpredictability: The output of the generator should not be predictable.
- Suppose there is a generator G that is efficient, that stretches the input, that is (t, ϵ)-indistinguishable. When can we call it secure?

Stream Ciphers: Pseudorandom generator

- What are the desirable properties of a pseudorandom generator $G: \{0,1\}^k \rightarrow \{0,1\}^n$ for Cryptographic purposes:
  - Stretch: $n > k$
  - Efficient: G should be efficient
  - Indistinguishability: No bounded resource algorithm should be able to distinguish the output of the generator $G(x)$ (for random x) from a random n bit string.
  - Unpredictability: The output of the generator should not be predictable.
- Suppose there is a generator G that is efficient, that stretches the input, that is (t, ϵ)-indistinguishable. When can we call it secure?
  - t is large. How large?
  - ϵ is small. How small?

Stream Ciphers: Pseudorandom generator

- Suppose there is a generator G that is efficient, that stretches the input, that is (t, ϵ)-indistinguishable. When can we call it secure?
  - t is large. How large?
  - ϵ is small. How small?
- Example values of k, n, t, and ϵ:
  - $k = 128$
  - $n = 2^{20}$
  - $t = 2^{80}$
  - $\epsilon = 2^{-40}$
- Are these figures good enough for all scenarios (present and future)?
  - We will discuss this in sometime.

Stream Ciphers: Security of Stream Ciphers

- Formalizing notion of security:
  - When can you say that your protocol has been broken?
    - Adversary is able to figure out the secret key.
    - Adversary is able to figure out the entire message.
    -
    -
  - Think of a security property that implies all other relevant security properties.
- Perfect Secrecy (Information Theoretic Security):
  - Let the message space be $\{0,1\}^n$.
  - For any two message M_0, M_1, and Ciphertext C

    $$\Pr[E_K(M_0) = C] = \Pr[E_K(M_1) = C]$$

    where the probability is over uniformly random K in the Keyspace.

Stream Ciphers: Security of Stream Ciphers

- Perfect Secrecy (Information Theoretic Security): An encryption scheme (E, D) is said to be perfectly secure if for any two message M_0, M_1, and Ciphertext C

$$\Pr[E_K(M_0) = C] = \Pr[E_K(M_1) = C]$$

- Definition ((t, ϵ)-message indistinguishability): An encryption scheme (E, D), is said to be (t, ϵ)-message indistinguishable if for any two messages M_0, M_1, and any algorithm A that runs in time at most t we have:

$$\mathrm{Adv}_{IND}(A, E) = \left|\Pr\left[A\left(E_K(M_0)\right) = 1\right] - \Pr\left[A\left(E_K(M_1)\right) = 1\right]\right| \leq \epsilon$$

- The above definition is with respect to one time encryption only.

Stream Ciphers: Security of Stream Ciphers

- Why is message indistinguishability a strong notion of security?
  - Theorem (informal): If a scheme is secure w.r.t. the message indistinguishability notion, then it is secure w.r.t. the key recovery notion.
    - Proof: ?
  - Theorem (informal): If a scheme is secure w.r.t. the message indistinguishability notion, then it is secure w.r.t. 100^{th} bit known notion.
  -
  -
  -

Stream Ciphers: Security of Stream Ciphers

- One Time Pad:
  - Let the key space and message space be $\{0,1\}^n$.
  - $E_K(M) = K \oplus M$ and $D_K(C) = K \oplus C$.
  - Is the OTP encryption scheme secure w.r.t. notion of message indistinguishability? What is the advantage of any Algorithm for this encryption scheme?
- OTP using PRG:
  - Let $G: \{0,1\}^k \rightarrow \{0,1\}^n$ be PRG.
  - $E_K(M) = G(K) \oplus M$ and $D_K(C) = G(K) \oplus C$.
  - Theorem (informal): If G is a secure PRG w.r.t. indistinguishability notion, then (E, D) is secure w.r.t. message indistinguishability.

Stream Ciphers: Security of Stream Ciphers

- OTP using PRG:
  - Let $G: \{0,1\}^k \rightarrow \{0,1\}^n$ be PRG.
  - $E_K(M) = G(K) \oplus M$ and $D_K(C) = G(K) \oplus C$.
  - Theorem (informal): If G is a secure PRG w.r.t. indistinguishability notion, then (E, D) is secure w.r.t. message indistinguishability.
  - Theorem: For every algorithm A that "attacks" (E, D) against the notion of message indistinguishability, there is an algorithm B that "attacks" G against the notion of indistinguishability such that

$$Adv_{IND}(A, E) \leq 2 \cdot Adv_{PRG}(B, G)$$

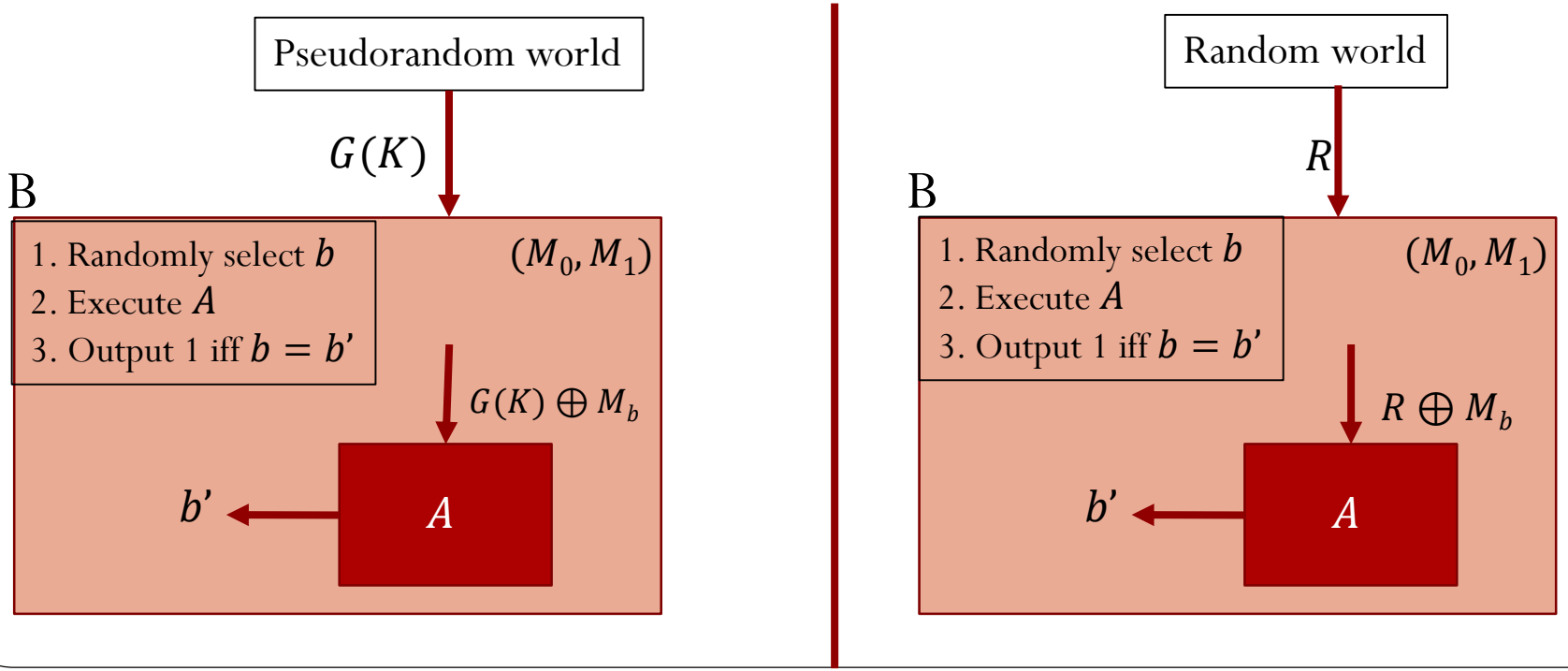
Moreover, if the running time of B is at most twice the running time of A.

Stream Ciphers: Security of Stream Ciphers

- Theorem: For every algorithm A that "attacks" (E, D) against the notion of message indistinguishability, there is an algorithm B that "attacks" G against the notion of indistinguishability such that

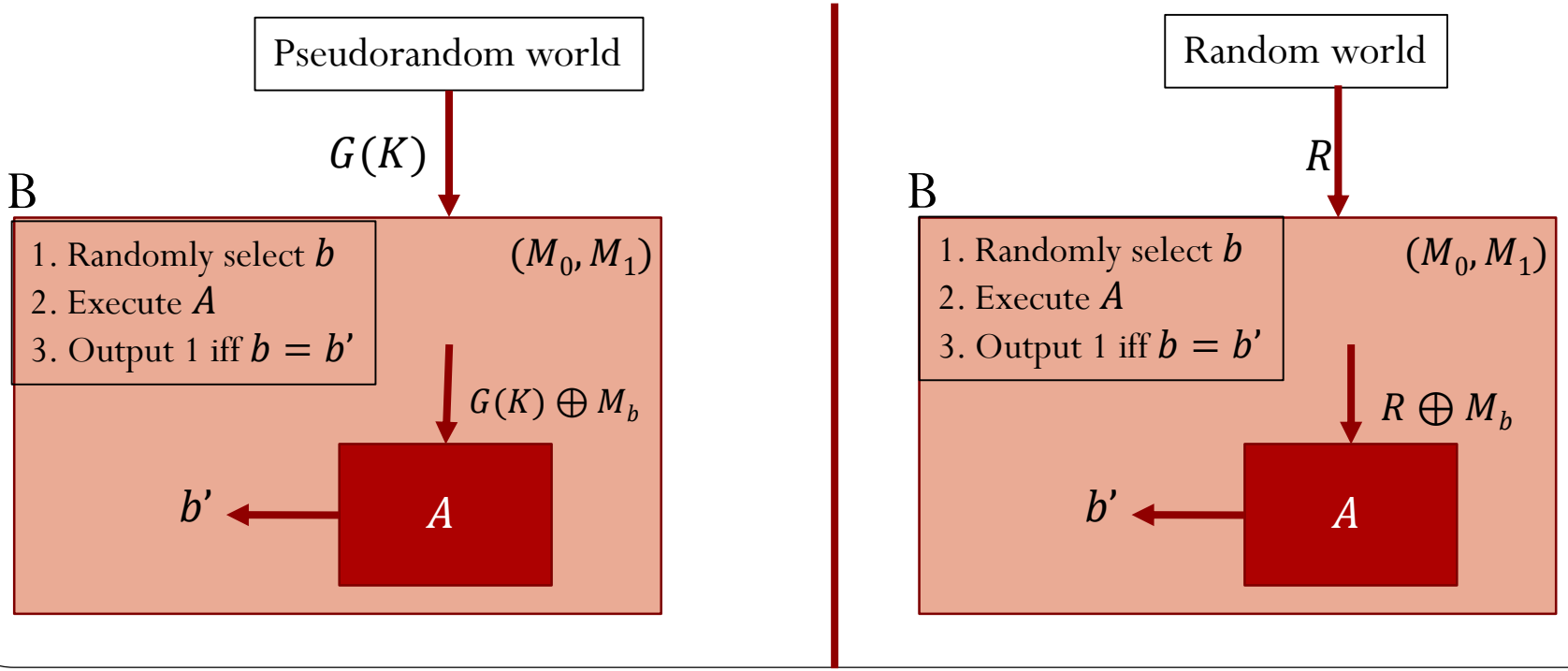
$$Adv_{IND}(A, E) \leq 2 \cdot Adv_{PRG}(B, G)$$

Moreover, if the running time of B is at most twice the running time of A.



Stream Ciphers: Security of Stream Ciphers

- $Adv_{PRG}(B, G) = |\Pr[B(G(K)) = 1] - \Pr[B(R) = 1]|$
- Lemma 1: $\Pr[B(R) = 1] = \frac{1}{2}$.
- Lemma 2: $\Pr[B(G(K)) = 1] = \frac{1}{2} \pm Adv_{IND}(A, E)$



Hybrid Argument

Hybrid Argument: Indistinguishability Vs Unpredictability

- Theorem(unpredictability implies indistinguishability): Let $G: \{0,1\}^k \to \{0,1\}^n$. If G is a $(t', \epsilon/n)$-unpredictable PRG, then G is also a (t, ϵ)-indistinguishable PRG.
- Proof: We prove the contrapositive. Let A be an algorithm that runs in time at most t such that the following holds:

$$\left| \Pr_{K \leftarrow \{0,1\}^k}[A(G(K)) = 1] - \Pr_{R \leftarrow \{0,1\}^n}[A(R) = 1] \right| \leq \epsilon$$

we will show that there exists an algorithm B that predicts the output of the generator.

  - Let the output of the generator be denoted by $y_1 y_2 \dots y_n$ and let $r_1 r_2 \dots r_n$ denote independent random bits.

Hybrid Argument : Indistinguishability Vs Unpredictability

- Theorem(unpredictability implies indistinguishability): Let $G: \{0,1\}^k \to \{0,1\}^n$. If G is a $(t', \epsilon/n)$-unpredictable PRG, then G is also a (t, ϵ)-indistinguishable PRG.

- Proof: We prove the contrapositive. Let A be an algorithm that runs in time at most t such that the following holds:

$$\left| \Pr_{K \leftarrow \{0,1\}^k}[A(G(K)) = 1] - \Pr_{R \leftarrow \{0,1\}^n}[A(R) = 1] \right| \leq \epsilon$$

we will show that there exists an algorithm B that predicts the output of the generator.

  - Let the output of the generator be denoted by $y_1 y_2 \dots y_n$ and let $r_1 r_2 \dots r_n$ denote independent random bits.
  - Consider the following distributions on n bit strings.
    - $D_0 = r_1 r_2 \dots r_n$
    - $D_1 = y_1 r_2 \dots r_n$
    - $D_2 = y_1 y_2 r_3 \dots r_n$
    - .
    - $D_n = y_1 y_2 \dots y_n$

Hybrid Argument : Indistinguishability Vs Unpredictability

- Theorem(unpredictability implies indistinguishability): Let $G: \{0,1\}^k \to \{0,1\}^n$. If G is a $(t', \epsilon/n)$-unpredictable PRG, then G is also a (t, ϵ)-indistinguishable PRG.

- Proof: We prove the contrapositive. Let A be an algorithm that runs in time at most t such that the following holds:

$$\left| \Pr_{K \leftarrow \{0,1\}^k}[A(G(K)) = 1] - \Pr_{R \leftarrow \{0,1\}^n}[A(R) = 1] \right| \leq \epsilon$$

we will show that there exists an algorithm B that predicts the output of the generator.

  - Let the output of the generator be denoted by $y_1 y_2 \dots y_n$ and let $r_1 r_2 \dots r_n$ denote independent random bits.
  - Consider the following distributions on n bit strings.
    - $D_0 = r_1 r_2 \dots r_n$
    - $D_1 = y_1 r_2 \dots r_n$
    - $D_2 = y_1 y_2 r_3 \dots r_n$
    - .
    - $D_n = y_1 y_2 \dots y_n$
  - Claim 1: $\left| \Pr_{R \leftarrow D_n}[A(R) = 1] - \Pr_{R \leftarrow D_0}[A(R) = 1] \right| \leq \epsilon$

Hybrid Argument : Indistinguishability Vs Unpredictability

- Theorem(unpredictability implies indistinguishability): Let $G: \{0,1\}^k \to \{0,1\}^n$. If G is a $(t', \epsilon/n)$-unpredictable PRG, then G is also a (t, ϵ)-indistinguishable PRG.

- Proof: We prove the contrapositive. Let A be an algorithm that runs in time at most t such that the following holds:

$$\left| \Pr_{K \leftarrow \{0,1\}^k}[A(G(K)) = 1] - \Pr_{R \leftarrow \{0,1\}^n}[A(R) = 1] \right| > \epsilon$$

we will show that there exists an algorithm B that predicts the output of the generator.

  - Let the output of the generator be denoted by $y_1 y_2 \dots y_n$ and let $r_1 r_2 \dots r_n$ denote independent random bits.
  - Consider the following distributions on n bit strings.
    - $D_0 = r_1 r_2 \dots r_n$
    - $D_1 = y_1 r_2 \dots r_n$
    - $D_2 = y_1 y_2 r_3 \dots r_n$
    - .
    - $D_n = y_1 y_2 \dots y_n$
  - Claim 1: $\left| \Pr_{R \leftarrow D_n}[A(R) = 1] - \Pr_{R \leftarrow D_0}[A(R) = 1] \right| > \epsilon$
  - Claim 2: $\exists i, \left| \Pr_{R \leftarrow D_i}[A(R) = 1] - \Pr_{R \leftarrow D_{i+1}}[A(R) = 1] \right| > \epsilon/n$

Hybrid Argument : Indistinguishability Vs Unpredictability

- Theorem(unpredictability implies indistinguishability): Let $G: \{0,1\}^k \to \{0,1\}^n$. If G is a $(t', \epsilon/n)$-unpredictable PRG, then G is also a (t, ϵ)-indistinguishable PRG.

- Proof:
  - Claim 2: $\exists i, \left| \Pr_{R \leftarrow D_i}[A(R) = 1] - \Pr_{R \leftarrow D_{i+1}}[A(R) = 1] \right| > \epsilon/n$
  - How do we use the above claim to design an algorithm B that predicts the $(i+1)^{th}$ bit of the generator, given the first i bits?

Hybrid Argument : Indistinguishability Vs Unpredictability

- Theorem(unpredictability implies indistinguishability): Let $G: \{0,1\}^k \to \{0,1\}^n$. If G is a $(t', \epsilon/n)$-unpredictable PRG, then G is also a (t, ϵ)-indistinguishable PRG.

- Proof:
  - Claim 2: $\exists i, \left| \Pr_{R \leftarrow D_i}[A(R) = 1] - \Pr_{R \leftarrow D_{i+1}}[A(R) = 1] \right| > \epsilon/n$
  - How do we use the above claim to design an algorithm B that predicts the $(i+1)^{th}$ bit of the generator, given the first i bits?
  - $B(y_1 y_2 \dots y_i)$:
    - Pick a random bit $r \leftarrow \{0,1\}$.
    - Pick independently random bits $r_{i+2}, r_{i+3}, \dots, r_n \leftarrow \{0,1\}$
    - Execute A on the input $(y_1 y_2 \dots y_i r r_{i+2} r_{i+3} \dots r_n)$, let b be the output of A.
    - If $(b = r)$, then output r else output $(1 - r)$.
  - Claim 3: $\Pr[B(y_1 y_2 \dots y_i) = y_{i+1}] > \epsilon/n$.

Security: Concrete Vs Asymptotic approach

Asymptotic security: Pseudorandom generator

- Suppose there is a generator G that is efficient, that stretches the input, that is (t, ϵ)-indistinguishable. When can we call it secure?
  - t is large. How large?
  - ϵ is small. How small?
- Example values of k, n, t, and ϵ:
  - $k = 128, n = 2^{20},\ t = 2^{80}, \epsilon = 2^{-40}$
- Are these figures good enough for all scenarios (present and future)?
  - Answer: No.
- Solution: Time of the adversary and error probability should not be concrete numbers but functions of a parameter of interest. This parameter is called the *security parameter*.

Asymptotic security: Pseudorandom generator

- Suppose there is a generator G that is efficient, that stretches the input, that is (t, ϵ)-indistinguishable. When can we call it secure?
  - t is large. How large?
  - ϵ is small. How small?
- Example values of k, n, t, and ϵ:
  - $k = 128, n = 2^{20},\ t = 2^{80}, \epsilon = 2^{-40}$
- Are these figures good enough for all scenarios (present and future)?
  - Answer: No.
- Solution: Time of the adversary and error probability should not be concrete numbers but functions of a parameter of interest. This parameter is called the *security parameter*.
- How does this help?

Asymptotic security: Pseudorandom generator

- Suppose there is a generator G that is efficient, that stretches the input, that is (t, ϵ)-indistinguishable. When can we call it secure?
  - t is large. How large?
  - ϵ is small. How small?
- Example values of k, n, t, and ϵ:
  - $k = 128, n = 2^{20},\ t = 2^{80}, \epsilon = 2^{-40}$
- Are these figures good enough for all scenarios (present and future)?
  - Answer: No.
- Solution: Time of the adversary and error probability should not be concrete numbers but functions of a parameter of interest. This parameter is called the *security parameter*.
- How does this help?
  - We can define security against all *polynomial time* adversaries i.e., algorithms that run in time polynomial in the security parameter.

Asymptotic security: Pseudorandom generator

- Suppose there is a generator G that is efficient, that stretches the input, that is (t, ϵ)-indistinguishable. When can we call it secure?
  - t is large. How large?
  - ϵ is small. How small?
- Example values of k, n, t, and ϵ:
  - $k = 128, n = 2^{20},\ t = 2^{80}, \epsilon = 2^{-40}$
- Are these figures good enough for all scenarios (present and future)?
  - Answer: No.
- Solution: Time of the adversary and error probability should not be concrete numbers but functions of a parameter of interest. This parameter is called the *security parameter*.
- How does this help?
  - We can define security against all *polynomial time* adversaries i.e., algorithms that run in time polynomial in the security parameter.
  - The success probability of such adversaries should be *negligible* in the security parameter.

Asymptotic security: Pseudorandom generator

- Solution: Time of the adversary and error probability should not be concrete numbers but functions of a parameter of interest. This parameter is called the *security parameter*.
- How does this help?
  - We can define security against all *polynomial time* adversaries i.e., algorithms that run in time polynomial in the security parameter.
  - The success probability of such adversaries should be *negligible* in the security parameter.
    - Definition (Negligible function): A function $negl(.)$ is said to be negligible if for every polynomial $p(.)$, there is an integer N such that for all integers $n > N, negl(n) < \frac{1}{p(n)}$.
    - Examples: $2^{-n}, 2^{-\sqrt{n}}, n^{-\log(n)}$

Asymptotic security: Pseudorandom generator

- Solution: Time of the adversary and error probability should not be concrete numbers but functions of a parameter of interest. This parameter is called the *security parameter*.
- How does this help?
  - We can define security against all *polynomial time* adversaries i.e., algorithms that run in time polynomial in the security parameter.
  - The success probability of such adversaries should be *negligible* in the security parameter.
    - Definition (Negligible function): A function $negl(.)$ is said to be negligible if for every polynomial $p(.)$, there is an integer N such that for all integers $\mathrm{k} > N, negl(k) < \frac{1}{p(k)}$.
    - Examples: $2^{-k}, 2^{-\sqrt{k}}, k^{-\log(k)}$
    - Properties:
      - $negl_1(k) + negl_2(k)$ is also negligible.
      - $p(k) \cdot negl(k)$ is also negligible.

Asymptotic security: Pseudorandom generator

- Solution: Time of the adversary and error probability should not be concrete numbers but functions of a parameter of interest. This parameter is called the *security parameter*.
- Asymptotic Security: A scheme is called secure if every PPT (Probabilistic Polynomial Time) adversary succeeds in *breaking* the scheme with only negligible probability.
- Security parameter: discussion
  - Security parameter is very closely related to the key size that is used. Usually it is the same as the key size. Asymptotic security implies that the larger the key size the more secure the scheme will be.
  - Example: Consider a PRG $G: \{0,1\}^k \rightarrow \{0,1\}^{l(k)}$. The deterministic algorithm G stretches arbitrary size seeds to longer strings.

Stream ciphers and PRGs

Summary

Stream Ciphers: Summary

- Stream ciphers are synonymous with pseudorandom generators (PRG).
- PRGs are algorithms that map $\{0,1\}^k$ to $\{0,1\}^{l(k)}$ with the following properties:
  - $\forall k, l(k) > k$.
  - The mapping algorithm G is deterministic and efficient.
  - Indistinguishability: For every PPT algorithm A (k here is the security parameter) and every polynomial $p(.)$, there is some integer N such that

    $\forall k > N\ , |\ Pr\left[A\left(G(K)\right) = 1\right] - \Pr[A(R) = 1]| \leq \frac{1}{p(k)}$.

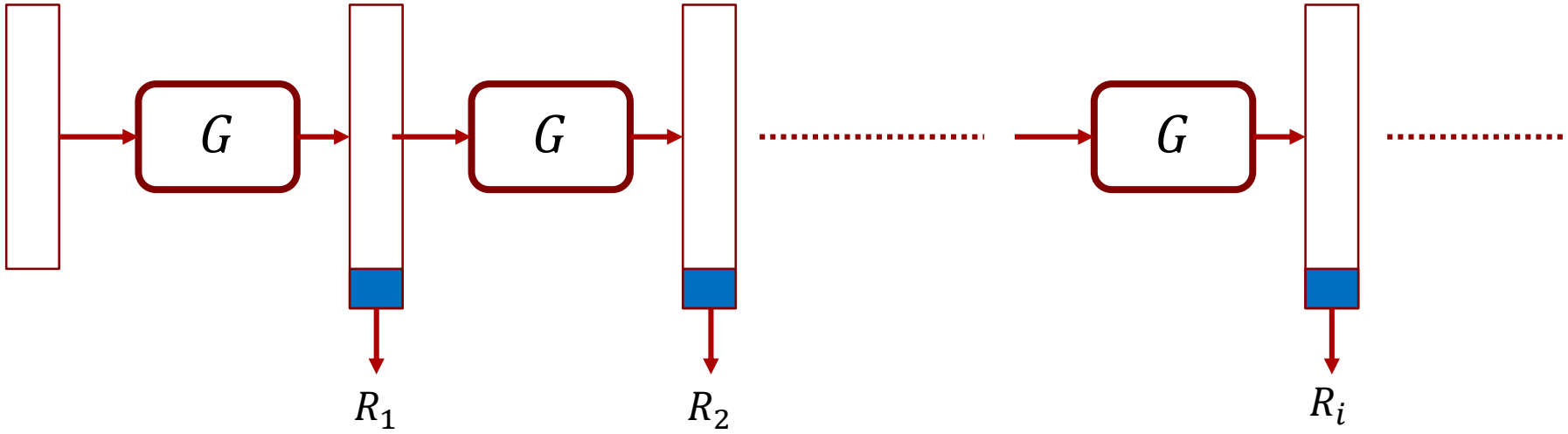
    In other words the success probability of all PPT algorithms should be negligible.

Stream Ciphers: Summary

- Stream ciphers are synonymous with pseudorandom generators (PRG).
- PRGs are algorithms that map $\{0,1\}^k$ to $\{0,1\}^{l(k)}$ with the following properties:
  - $\forall k, l(k) > k$.
  - The mapping algorithm G is deterministic and efficient.
  - Indistinguishability: The success probability of all PPT algorithms should be negligible.
- Question: Suppose we have a secure PRG where $l(k) = k + 1$, i.e., the PRG stretches the bits by 1. Can we construct a secure PRG with longer stretch?

Stream Ciphers: Summary

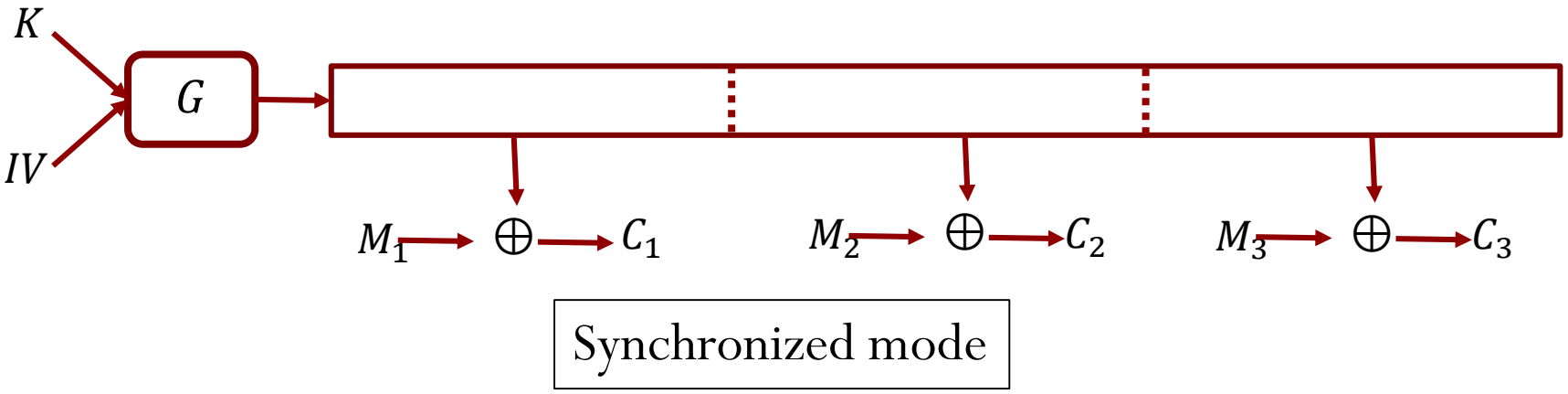
- Question: Suppose we have a secure PRG where $l(k) = k + 1$, i.e., the PRG stretches the bits by 1. Can we construct a secure PRG with longer stretch?



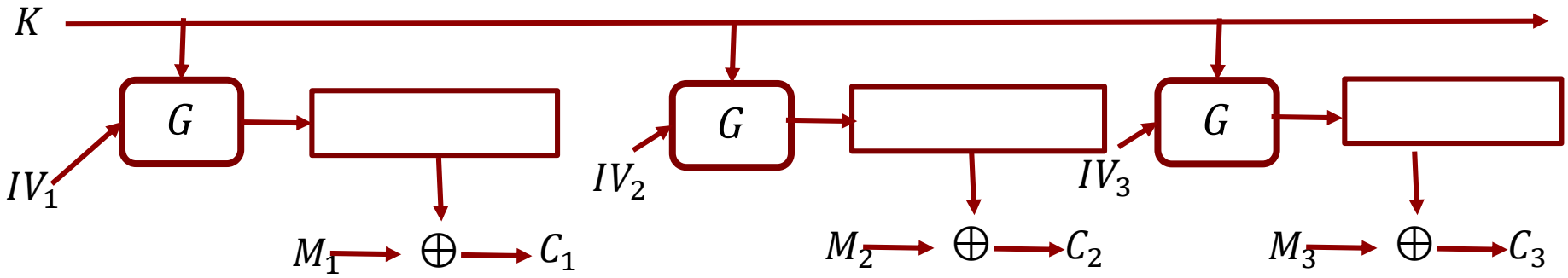
- Question: Why does the above construction give a secure PRG?

Stream Ciphers: Summary

- Multiple encryptions using Stream Ciphers.



Synchronized mode



Asynchronized mode

End