

1 Lecture Plan

- Complete proof of “perfect secrecy \iff perfect indistinguishability”.
- Motivate computational security
- Define computationally secure encryption
- Define pseudorandom generators

2 Perfect secrecy \iff Perfect Indistinguishability

- Giving a better explanation of the conditional independence of M and $\mathcal{A}(C)$ given $C = c$.
- Completing the last part of the proof.

3 Computational Security

- To avoid the limitations of perfect secrecy, the weaker notion of computational secrecy is used in modern cryptography.
- Concrete guarantees of security are difficult to provide
- In the asymptotic approach, the cryptographic schemes as well as the involved parties are parametrized by an integer-valued *security parameter* n (typically the key length)
- Computational security allows two relaxations
 - Security is only guaranteed against adversaries with randomized attack algorithms with running time which is polynomial in n .
Note: An algorithm A runs in polynomial time if there exists a polynomial p such that, for every input $x \in \{0, 1\}^*$, the computation of $A(x)$ terminates within at most $p(|x|)$ steps where $|x|$ denotes the length of x . An algorithm with polynomial running time is said to be **efficient**.
 - The adversary is allowed to succeed with *negligible* probability, i.e. the success probability is *asymptotically smaller than any inverse polynomial in n* .

Definition (Page 48 of KL). A function f from the natural numbers to the non-negative real numbers is **negligible** if for every positive polynomial p there is an N such that for all integers $n > N$ it holds that $f(n) < \frac{1}{p(n)}$.

Examples: 2^{-n} , $2^{-\sqrt{n}}$, $n^{-\log n}$

- Negligible success probabilities obey certain closure properties.

Proposition (Page 49 of KL). *Let negl_1 and negl_2 be negligible functions. Then,*

1. *The function negl_3 defined by $\text{negl}_3(n) = \text{negl}_1(n) + \text{negl}_2(n)$ is negligible.*
2. *For any positive polynomial p , the function negl_4 defined by $\text{negl}_4(n) = p(n) \cdot \text{negl}_1(n)$ is negligible.*

- The second part of the proposition implies that if a certain event occurs with only negligible probability in a certain experiment, then the event occurs with negligible probability even if the experiment is repeated polynomially many times.
- *General framework of any computational security definition:* A scheme is secure if for every probabilistic polynomial-time adversary \mathcal{A} carrying out an attack of a formally specified type, the probability that \mathcal{A} succeeds in the attack (where success is also formally specified) is negligible.
- Necessity of the relaxations
 - Exclude brute-force attackers
 - Exclude pure-guess attackers who succeed with exponentially small probability

3.1 Defining Computationally Secure Encryption

- We need to introduce the security parameter n in our syntax of private-key encryption.
- We assume $\mathcal{M} = \{0, 1\}^*$.
- We allow the decryption algorithm to output an error in case it is presented with an invalid ciphertext.

Definition. *A **private-key encryption scheme** is a tuple of probabilistic polynomial-time algorithms $(\text{Gen}, \text{Enc}, \text{Dec})$ such that:*

1. *The key-generation algorithm takes 1^n as input and gives key k , i.e. $k \leftarrow \text{Gen}(1^n)$.*
2. *For $m \in \{0, 1\}^*$, $c \leftarrow \text{Enc}_k(m)$.*
3. *For ciphertext c , $\text{Dec}_k(c) = m$ or error indicator \perp .*

It is required that for every n, c, k , we have $\text{Dec}_k(\text{Enc}_k(m)) = m$.

3.2 Indistinguishability in the presence of an eavesdropper

- We consider the ciphertext-only attack where the adversary observes a single ciphertext.
- Our definition will resemble the perfect adversarial indistinguishability definition except for two differences:
 - The experiment is parametrized by n
 - We require the adversary to output equal length messages m_0, m_1 . (See exercise 3.2 of KL)
- Consider the following experiment $\text{PrivK}_{\mathcal{A}, \Pi}^{\text{eav}}(n)$:
 1. The adversary \mathcal{A} is given 1^n and outputs a pair of arbitrary messages $m_0, m_1 \in \mathcal{M}$ with $|m_0| = |m_1|$.
 2. A key k is generated using Gen , and a uniform bit $b \in \{0, 1\}$ is chosen. Ciphertext $c \leftarrow \text{Enc}_k(m_b)$ is computed and given to \mathcal{A} . This ciphertext c is called the *challenge ciphertext*.
 3. \mathcal{A} outputs a bit b' .
 4. The output of the experiment is defined to be 1 if $b' = b$, and 0 otherwise. We write $\text{PrivK}_{\mathcal{A}, \Pi}^{\text{eav}}(n) = 1$ if the output of the experiment is 1 and in this case we say that \mathcal{A} succeeds.

Definition. A private-key encryption scheme $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ has *indistinguishable encryptions in the presence of an eavesdropper*, or is **EAV-secure**, if for all probabilistic polynomial-time adversaries \mathcal{A} there is a negligible function negl such that, for all n ,

$$\Pr [\text{PrivK}_{\mathcal{A}, \Pi}^{\text{eav}}(n) = 1] \leq \frac{1}{2} + \text{negl}(n).$$

4 Pseudorandom Generators

- It is not known how to construct computationally secure encryption schemes without making any assumptions. We need to assume the existence of pseudorandom generators.
- A pseudorandom generator is an efficient (polynomial-time), deterministic algorithm for transforming a short, uniform bitstring called the *seed* into a longer, “uniform-looking” or “pseudorandom” output string.

5 References and Additional Reading

- Sections 3.1, 3.2, 3.3 from Katz/Lindell