PESU Center for
Information Security,
Forensics and
Cyber Resilience

Welcome to

PES University

Ring Road Campus, Bengaluru

PESU Center for
Information Security,
Forensics and
Cyber Resilience

PES
UNIVERSITY

APPLIED CRYPTOGRAPHY

Private Key System

Lecture 13

Private key cryptography

One key between sender and receiver

Perfect secrecy (formal)

- Encryption scheme (Gen, Enc, Dec) with message space $\mathcal{M}$ and ciphertext space $\mathcal{C}$ is *perfectly secret* if for every distribution over $\mathcal{M}$, every m $\in$ $\mathcal{M}$, and every c $\in$ $\mathcal{C}$ with Pr[C=c] > 0, it holds that

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

- I.e., the distribution of M does not change conditioned on observing the ciphertext

Perfect secrecy

- Requires that ***absolutely no information*** about the plaintext is leaked, even to eavesdroppers *with* ***unlimited computational power***

  - Has some inherent drawbacks
  - Seems unnecessarily strong

Computational security

- **Small probability of information leakage to eavesdroppers with bounded computational resources**
- Led to two relaxation to notion of security

  to achieve key reusability

  1. Security is only guaranteed against efficient adversaries that run for some feasible amount of time.
  2. Adversaries can break the scheme with small probability as most system does not require ever-lasting security.

Small probability?

- Consider the following four events:

1. Winning a lottery with 1 million contestants
2. Winning a lottery with 1 million contestants 5 times in a row
3. Winning a lottery with 1 million contestants 6 times in a row.
4. Winning a lottery with 1 million contestants 7 times in a row.

- What is the order of these events from most likely to least likely?

Computational security

- Two approaches
  - Concrete security
  - Asymptotic security

Concrete Approach

- A scheme is (t, ε)-secure if any adversary running for time at most t succeeds in breaking the scheme with probability at most ε .
- Disadvantages:
  - Will not consider the progress in the computing speed.

Asymptotic Approach

- Measure algorithms efficiency with respect to basic step depending on input size.
- Various notations are big O, omega and theta

Asymptotic in context of cryptography

- Security parameter: n (Size of secret key)
- All algorithms are expressed as function of security parameter
- Running time of user (encryption/decryption function)
- Running time of adversary
- Success rate of attacker

- AES security parameter n= 128, 192, 256

Polynomial-time algorithm

- Defining efficient algorithm

"Algorithm A has a polynomial running time, if there exists a polynomial p(.), such for every input $x \in \{0,1\}^*$, the computation of A(x) terminates within p(|x|) steps, where |x| denotes the length of the string x"

A polynomial-time algorithm is an algorithm whose execution time is either given by a polynomial on the size of the input or can be bounded by such a polynomial.

Negligible functions

- " A negligible function Is one that is asymptotically smaller than any inverse polynomial function"
- Definition 1:
  - ***A function f from the natural number to a non negative real number is negligible if every positive polynomial p there is an N such that for all integers*** $n > N$ ***it holds that*** $f(n) < \frac{1}{p(n)}$

Security parameter n

Negligible function

- Definition 2:

For every constant c, there exists some N such that $f(n) < n^{-c}, \ for\ all\ n > N$

Therefore

$$f(n) < \frac{1}{p(n)} < \frac{1}{n^c}$$

Example negligible functions

- $2^{-n}, 2^{-\sqrt{n}}, n^{-\log n}$ are all negligible functions
- Consider a function $f(n) = 2^{-n}$

  then $2^{-n} < 1/n^5$ for all n > 23

Carefully select the value of n for meaningful security

Closure properties

- For polynomials

  $if\ p_1(n) and\ p_2(n)\ are\ polynomial\ function\ then$

  $p_1(n) + p_2(n)$ and $p_1(n) X\ p_2(n)$ are polynomial

- for negligible functions

  $negl_3(n) = negl_1(n) + negl_2(n)$ is negligible

  $negl_4(n) = negl_1(n).\mathrm{p}(\mathrm{n})$ is negligible

Private key system

- The key-generation algorithm Gen takes as input 1^n and outputs a key k; we write k ← Gen(1^n)(emphasizing that Gen is a randomized algorithm). We assume without loss of generality that any key k output by Gen(1^n) satisfies |k|>= n.
- 2. The encryption algorithm Enc takes as input a key k and plaintext message m $\in\{0,1\}^*$, and outputs a ciphertext c. Since Enc may be randomized, we write this as c ← Enc_k(m).
- 3. The decryption algorithm Dec takes as input a key k and a ciphertext c, and outputs a message m or an error. We assume that Dec is deterministic, and so write m := Dec_k(c)

The indistinguishability experiment

$$PrivK^{eav}_{A,\Pi}(n)$$

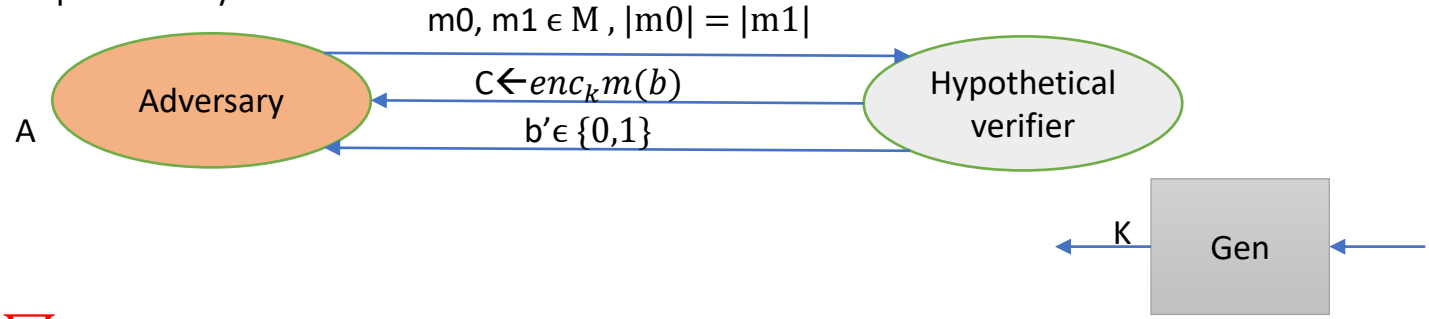
- PrivK denotes an experiment, in context of a private key or symmetric encryption,
- eav means we are considering an adversary who is an eavesdropper,
- A is the name of the adversarial algorithm
- Π is the name of the scheme,

In this experiment, the rules are as follows.

- Adversary can submit any pair of messages (m0, m1) from the plaintext space with the restriction that the size of the two plaintext should be same |m0| = |m1|
- The hypothetical verifier does the following:
  - It randomly generates a key by running a key generation algorithm and it randomly encrypts one of the messages (m0 | m1) using the key,
- The challenge for the adversary is to identify what plaintext has been encrypted in the challenge ciphertext c, whether it is m0 or m1.

The indistinguishability experiment

Computationally bounded



$\prod$ ***is perfectly indistinguishable if for every A***

$$pr(PrivK^{eav}_{A,\prod}(n)=1)=\frac{1}{2}$$

Adversary does

- Outputs a bit, namely its guess about what exactly has been encrypted in the challenge ciphertext.
- The scheme Π is perfectly secure or we say that a scheme is perfectly indistinguishable if the probability with which adversary could successfully Identify what message has been encrypted is upper bounded by half.

Semantic security in COA model

Enc is semantically secure, if the ciphertext does not reveal any additional information about the underlaying plaintext

- Adversary has access to abstract function h(m)
  - h(m): any prior information about plaintext obtained by other means
- Goal of the adversary is to compute f(m) of the underlying plaintext
  - f(m): Addition information that adversary wants to learn about message m

Semantic security in COA model

Semantic security is chances that adversary could compute f(m) using c and h(m) is same as adversary compute f(m) without c.

- $|pr(A(enc_k(m), h(m)) - pr(A'(h(m))| \leq negl(n)$

Let $R := \{0,1\}^4$ and consider the following PRF $F : R^5 \times R \to R$ defined as follows:

$$F(k,x) := \left\{ \begin{array}{l} t = k[0] \\ \text{for i=1 to 4 do} \\ \qquad \text{if } (x[i-1] == 1) \quad t = t \oplus k[i] \\ \text{output } t \end{array} \right.$$

That is, the key is $k = (k[0], k[1], k[2], k[3], k[4])$ in R^5 and the function at, for example, 0101 is defined as $F(k, 0101) = k[0] \oplus k[2] \oplus k[4]$.

For a random key k unknown to you, you learn that

$$F(k, 0110) = 0011 \text{ and } F(k, 0101) = 1010 \text{ and } F(k, 1110) = 0110\,.$$

What is the value of $F(k, 1101)$? Note that since you are able to predict the function at a new point, this PRF is insecure.

Thank You!

Next Class

- ☞ Mandatory reading for the next class
- ☞ https://en.wikipedia.org/wiki/Pseudorandom_number_generator

S Rajashree

Computer Science and Engineering

PES University, Bengaluru

PESU Center for Information Security, Forensics and Cyber Resilience

PESU IoT PESU Center for Internet of Things