Statistical and Computational Security

Cryptography and Protocols

Andrei Bulatov

Symmetric Encryption Scheme

- A symmetric encryption scheme is a triple of algorithms (K,E,D)
  - K keys generation
  - E encryption algorithm
  - D decryption algorithm
- For simplicity assume that k $\leftarrow$ K uniformly at random, $k \in \{0,1\}^l$ or $k \in U_l$
- $P \in \{0,1\}^m$ plaintext

  $E:\{0,1\}^l \times \{0,1\}^m \to \{0,1\}^* \quad | \quad E_k(P) = C$

  $D:\{0,1\}^l \times \{0,1\}^* \to \{0,1\}^m \quad | \quad D_k(C) = P$

- In general, E (and possibly D) are randomized

Perfect Security

- Let (K,E,D) be a symmetric encryption scheme. It is said to be perfectly secure if for any two plaintexts P_1, P_2 and a ciphertext C

$$\Pr[E_k(P_1) = C] = \Pr[E_k(P_2) = C],$$

where the probability is over the random choice k ← K, and also over the coins flipped by E

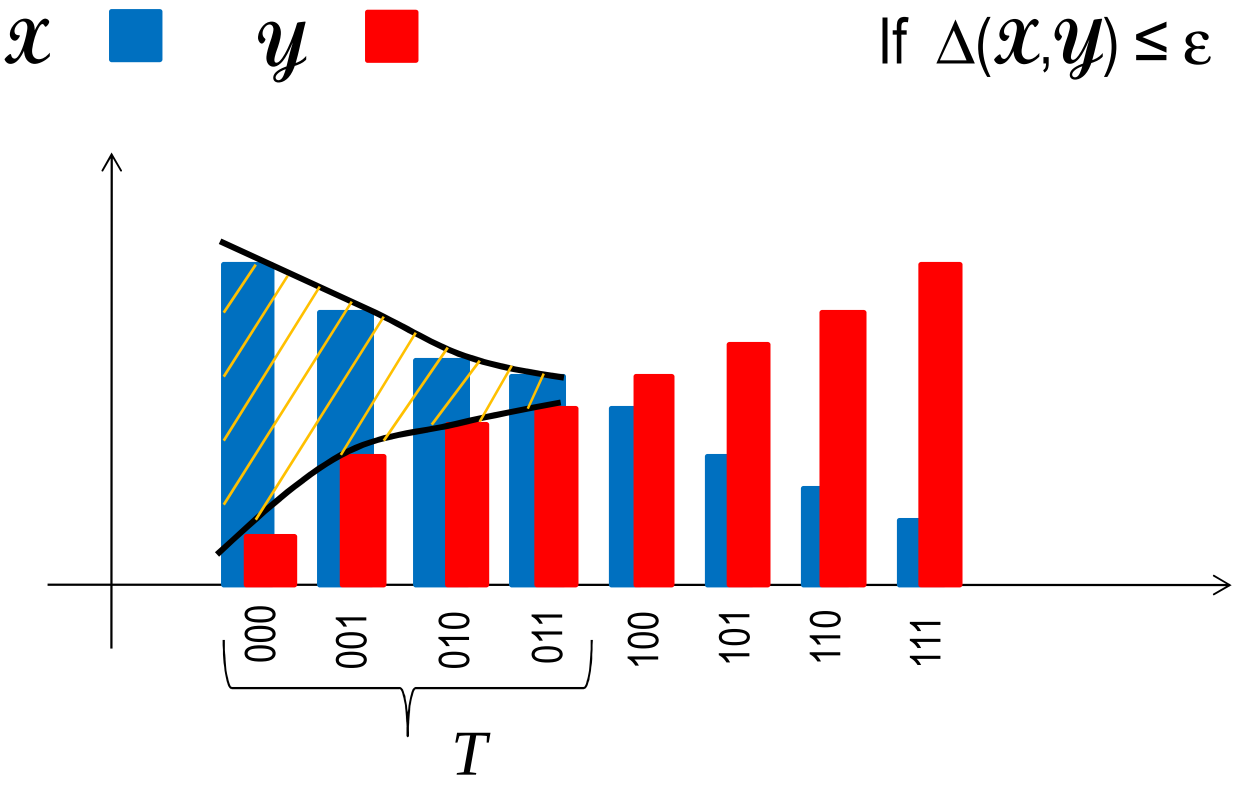
Statistical Distance

- Let $\mathcal{X}$ and $\mathcal{Y}$ be two distributions over $\{0,1\}^m$ The statistical distance between $\mathcal{X}$ and $\mathcal{Y}$, denoted $\Delta(\mathcal{X},\mathcal{Y})$ is

$$\max_{T\subseteq\{0,1\}^m} |\Pr[\mathcal{X}\in T]-\Pr[\mathcal{Y}\in T]|$$

If $\Delta(\mathcal{X},\mathcal{Y})\le\varepsilon$ we write

$$\mathcal{X}\equiv_\varepsilon\mathcal{Y}$$



Statistical Security

- A symmetric encryption scheme is said to be ε-statistically secure, if for any two plaintexts P_1, P_2 distributions $E_k(P_1), E_k(P_2)$ are ε-equivalent

- **Theorem**.

  Let (K,E,D) be a SES with m-bit messages and m-1 –bit keys.

  Then there are plaintexts P_1, P_2 with $\Delta(E_k(P_1), E_k(P_2)) \geq 1/2$

Statistical Security (cntd)

- **Observation**.

  If $\mathbb{E}[X] \le \mu$ then $\Pr[X \le \mu] > 0$.

- **Proof** (of the theorem).

  Let $P_1 = 0^m$ and $S = \{E_k(0^m) \mid k \in \{0,1\}^{m-1}\}$ Then $|S| \le 2^{m-1}$

  Experiment:

  Choose a random plaintext $P \in \{0,1\}^m$ define 2^{m-1} random variables: for every $k \in \{0,1\}^{m-1}$ we set

  $T_k(P) = 1$ if $E_k(P) \in S$ and 0 otherwise

  For every k, E_k is one-to-one, hence, $\Pr[T_k = 1] \le \frac{1}{2}$

  Therefore $\mathbb{E}[T_k] \le \frac{1}{2}$

Statistical Security (cntd)

- **Proof** (cntd)

  Set $T = \sum_k T_k$ Then $\mathbb{E}[T] = \mathbb{E}\left[\sum_k T_k\right] = \sum_k \mathbb{E}[T_k] \leq \frac{2^{m-1}}{2}$

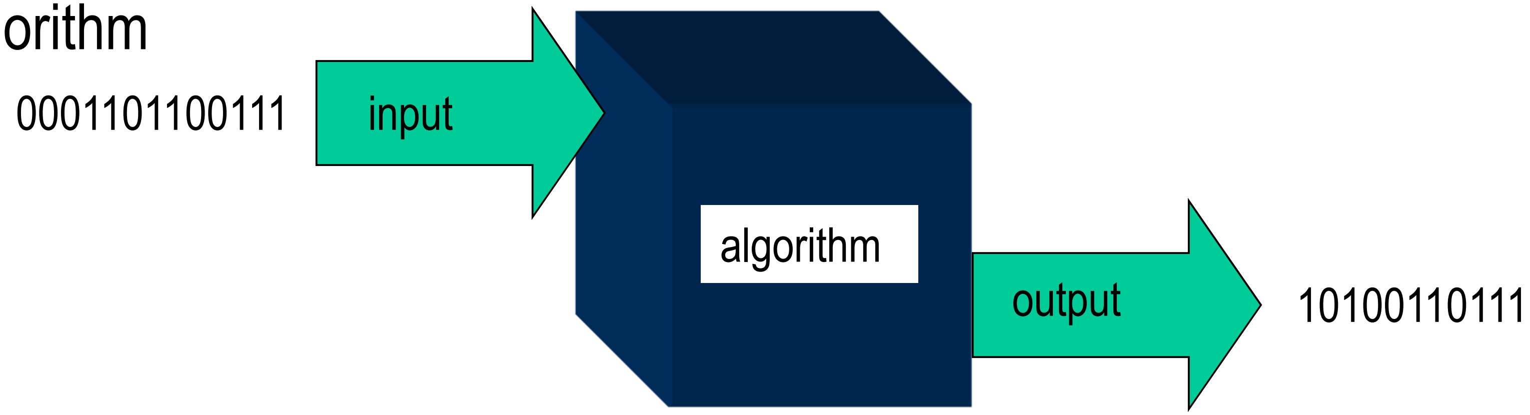
  By Observation, $\Pr[T \leq \frac{2^{m-1}}{2}] > 0$, or in other words, there exists P such that $\sum_k T_k(P) \leq \frac{2^{m-1}}{2}$

  For such P at most half of the keys satisfy $E_k(P) \in S$ or, equivalently, $\Pr[E_k(P) \in S] \leq 1/2$

  Since $\Pr[E_k(0^m) \in S] = 1$, we get $\Delta(E_k(0^m), E_k(P)) \geq 1/2$

Algorithms

- Algorithm



- Algorithm performs a sequence of `elementary steps' that can be:
  - arithmetic operations
  - bit operations
  - Turing machine moves
  - ………. (but not quantum computing!!)
- We allow probabilistic algorithms, that is flipping coins is permitted

Complexity

- The time complexity of algorithm A is function $f(n)$ that is equal to the number of elementary steps required to process the most difficult input of length n
- We do not distinguish algorithms of complexity $2n^2$ and $100000n^2$
- A computational problem has time complexity at most $f(n)$ if there is an algorithm that solves the problem and has complexity $O(f(n))$

  \- problem solvable in linear time: there is an algorithm that on input of length n performs at most Cn steps

  \- problem solvable in quadratic time: there is an algorithm that on input of length n performs at most Cn^2 steps

Complexity (cntd)

- Polynomial time solvable problems:

  There is a polynomial p(n) such that the problem is solvable in time O(p(n)

- P - class of problems solvable in poly time by a deterministic algorithm
- BPP - class of problems solvable in poly time by a probabilistic algorithm
- An algorithm is superpolynomial if its time complexity f(n) is not in O(p(n)) for any polynomial p(n)
- A function $\varepsilon: N \to [0,1]$ is polynomially bounded if $\varepsilon(n) \geq \frac{1}{p(n)}$ for some polynomial p(n)

Computational Security

- Let (K,E,D) be a SES that uses n-bit keys to encrypt m(n)-bit messages. It is computationally secure if for any polynomial time algorithm Eve: $\{0,1\}^* \to \{0,1\}$, any polynomially bounded ε: $\{0,1\}^* \to [0,1]$, n, and $P_1, P_2 \in \{0,1\}^{m(n)}$

$$|\Pr[\text{Eve}(E_{U_n}(P_1)) = 1] - \Pr[\text{Eve}(E_{U_n}(P_2)) = 1]| < \varepsilon(n)$$

- **Conjecture**.

  A computationally secure SES exists for $m(n) = n^{100}$ (may be even for $m(n) = 2^{0.9n}$)

Computational Indistinguishability: Difficulties

- It is useful to define computational security in a similar way as statistical one: define distance or equivalence of distributions and then say that $E_{U_n}(P_1)$ and $E_{U_n}(P_1)$ are `equivalent'. However, there are problems
- For computational definitions we need algorithms, not events

  Solution: Instead of saying $X \in S$ we use the characteristic function f of S. So we say f(X) = 1 instead.

  Distance between distributions can then be defined as

$$\max_f |\Pr[f(X)=1]-\Pr[f(Y)=1]|$$

  over all `easily' computable functions f
- Computational complexity does not make sense for fixed distributions.

  Solution: Use collections or sequences of random variables

Computational Indistinguishability: Definition

- Let T(n) and ε(n) be functions on natural numbers. Collections of random variables $\{X_n\}$ and $\{Y_n\}$ such that $X_n, Y_n \in \{0,1\}^n$ are said to be computationally (T,ε)-indistinguishable , if for any probabilistic algorithm Alg with time complexity at most T(n)

$$|\Pr[\text{Alg}(X_n) = 1] - \Pr[\text{Alg}(Y_n) = 1]| \le \varepsilon(n)$$

  Denoted $\{X_n\} \approx_{T,\varepsilon} \{Y_n\}$

- For example:

  Let (K,E,D) be a SES that uses n-bit keys to encript m(n)-bit messages. It is computationally secure if for any $P_1, P_2 \in \{0,1\}^{m(n)}$ distributions $E_{U_n}(P_1)$ and $E_{U_n}(P_1)$ are (T,ε)-indistinguishable for any polynomial T and any polynomially bounded ε