Pseudorandom Functions: A pseudorandom function is a family of functions with the property that the input-output behavior of a random instance of the family is "computationally indistinguishable" from that of a random function.

Let $F: K \times D \to R$ be a family of functions. $K = \{0,1\}^k$, $D = \{0,1\}^l$, and $R = \{0,1\}^L$ for some integers $k, l, L \geq 1$. Let A be an algorithm (adversary) that takes an oracle for a function $g: D \to R$, and returns a bit. We consider two different ways in which g can be chosen, giving rise to two different worlds:

World 0: The function g is drawn at random from $Func(D,R)$.

World 1: The function g is drawn at random from F. A key K is randomly chosen and g is set to F_K.

The objective of A is to tell whether g belongs to world 0 or world 1. A is allowed to be a randomized algorithm. We consider two experiments:

$Exp_F^{prf-1}(A)$

{ Randomly select K;
$b \leftarrow A^{F_K}$;
Return b;
}

$Exp_F^{prf-0}(A)$

{ Randomly select g from $Func(D,R)$;
$b \leftarrow A^{g}$;
Return b;
}

A^{F_K} means A is given F_K as oracle: $X \to [F_K] \to F_K(X)$ (oracle)

A^g means A is given g as oracle: $X \to [g] \to g(X)$ (oracle)

The prf-advantage of A is defined as:

$$Adv_F^{prf}(A) = Pr[Exp_F^{prf-1}(A) = 1] - Pr[Exp_F^{prf-0}(A) = 1].$$

Intuitively, F is "secure" if the value of the advantage function is "low" for all adversaries whose resources are "practical" (for example: polynomial time algorithms).

Pseudorandom Permutations (PRP): A family of functions $F: K \times D \to D$ is a pseudorandom permutation if the input-output behavior of a random instance of the family is "computationally indistinguishable" from that of a random permutation on D.

PRP under Chosen Plaintext Attack (CPA): We consider an adversary A that has oracle access to a function g chosen in one of two ways:

World 0: The function g is drawn at random from Perm(D).

World 1: The function g is drawn at random from F. A key k is randomly chosen and g is set to F_k.

A has to decide the world in which g belongs to.

Let $F: K \times D \to D$ be a family of functions, and let A be an algorithm that takes an oracle for a function $g: D \to D$, and returns a bit. We consider two experiments:

$Exp_F^{prp\text{-}cpa\text{-}1}(A)$

{ k is randomly selected key;
  $b \leftarrow A^{F_k}$;
  Return b;
}

$Exp_F^{prp\text{-}cpa\text{-}0}(A)$

{ g is randomly chosen from Perm(D);
  $b \leftarrow A^{g}$;
  Return b;
}

The prp-cpa advantage of A is defined as:

$$Adv_F^{prp\text{-}cpa}(A) = Pr[Exp_F^{prp\text{-}cpa\text{-}1}(A) = 1] - Pr[Exp_F^{prp\text{-}cpa\text{-}0}(A) = 1]$$

Intuitively, a family F is a secure PRP under CPA if $Adv_F^{prp-cpa}$ is "small" for all adversaries using a "practical" amount of resources.

PRP under Chosen Ciphertext Attack (CCA):

We consider an adversary A that has oracle access to two functions, g and its inverse g^{-1} chosen in one of two ways:

World 0: The function g is drawn at random from Perm(D).

World 1: The function g is drawn at random from F.

Let $F: K \times D \to D$ be a family of permutations, and let A be an algorithm that takes an oracle for a function $g: D \to D$, and also an oracle for the function $g^{-1}: D \to D$, and returns a bit. Objective of A is to decide in which world g belongs to. We consider two experiments:

$Exp_F^{prp-cca-1}(A)$

{ Randomly select a key k;
$b \leftarrow A^{F_k, F_k^{-1}}$;
Return b;
}

$Exp_F^{prp-cca-0}(A)$

{ Randomly select g from Perm(D);
$b \leftarrow A^{g, g^{-1}}$;
Return b;
}

The prp-cca-advantage of A is defined as:

$$Adv_F^{prp-cca}(A) = Pr[Exp_F^{prp-cca-1}(A) = 1] - Pr[Exp_F^{prp-cca-0}(A) = 1]$$

Intuitively, a family F is a secure PRP under CCA if $Adv_F^{prp-cca}(A)$ is "small" for all adversaries using a "practical" amount of resources.

Example 1: We define a family of functions $F: \{0,1\}^k \times \{0,1\}^\ell \to \{0,1\}^L$ as follows: we let $k = L\ell$ and view a k-bit key K as specifying an L row by ℓ column matrix of bits. The input string $X = X[1]\ldots X[\ell]$ is viewed as a sequence of bits, and the value of $F(K,x)$ is the corresponding matrix vector product:

$$F_K(X) = \begin{bmatrix} K[1,1] & K[1,2] & \cdots & K[1,\ell] \\ K[2,1] & K[2,2] & & K[2,\ell] \\ \vdots & \vdots & & \vdots \\ K[L,1] & K[L,2] & & K[L,\ell] \end{bmatrix} \cdot \begin{bmatrix} X(1) \\ X(2) \\ \vdots \\ X(\ell) \end{bmatrix} = \begin{bmatrix} Y[1] \\ Y[2] \\ \vdots \\ Y(L) \end{bmatrix}$$

where $Y(1) = K[1,1]\cdot x[1] \oplus K[1,2]\cdot x(2) \oplus \cdots \oplus K[1,\ell]\cdot x[\ell]$

$Y(2) = K[2,1]\cdot x(1) \oplus K[2,2]\cdot x(2) \oplus \cdots \oplus K(2,\ell)\cdot x(1)$

$\vdots$

$Y(L) = K[L,1]\cdot x(1) \oplus K[L,2]\cdot x(2) \oplus \cdots \oplus K[L,\ell]\cdot x(1).$

Here the bits in the matrix are the bits in the key, and arithmetic is modulo two.
We observe that for any key K we have $F_K(0^\ell) = 0^L$.
This is a weakness since a random function of ℓ-bits to L-bits is very unlikely to return 0^L on input 0^ℓ.

Adversary D^g
{ $y \leftarrow g(0^\ell)$;
if $y = 0^L$ then return 1 else return 0;
}

$$Adv_F^{prf}(D) = Pr[Exp_F^{prf\text{-}1}(D) = 1] - Pr[Exp_F^{prf\text{-}0}(D) = 1]$$

$$= 1 - 2^{-L}.$$

Complexity of D is $O(\ell^2 L)$.

$\Rightarrow$ F is insecure PRF.

Example 2: Suppose we are given a secure PRF $F: \{0,1\}^k \times \{0,1\}^l \to \{0,1\}^L$. We want to use F to design a PRF $G: \{0,1\}^k \times \{0,1\}^1 \to \{0,1\}^{2L}$. Consider G_k:

$G_k(x) = F_k(x) \| F_k(\bar{x})$, where $\|$ denotes concatenation.

G is not secure. Consider adversary D:

Adversary D^g
{ $y_1 \leftarrow g(1^l)$;
$y_2 \leftarrow g(0^l)$;
Parse y_1 as $y_1 = y_{1,1} \| y_{1,2}$ with $|y_{1,1}| = |y_{1,2}| = L$;
Parse y_2 as $y_2 = y_{2,1} \| y_{2,2}$ with $|y_{2,1}| = |y_{2,2}| = L$;
if $y_{1,1} = y_{2,2}$ then return 1 else return 0;
}

$$Adv^{prf}_G(D) = Pr[Exp^{prf-1}_G(D) = 1] - Pr[Exp^{prf-0}_G(D) = 1]$$
$$= 1 - 2^{-L}$$

Complexity of D: $O(l+L)$ + time for 4 computations of F.

$\Rightarrow$ G is not secure PRF.

Security Against Key Recovery: Let $F: K \times D \to R$ be a family of functions, and let B be an algorithm that takes an oracle for a function $g: D \to R$ and outputs a string. We consider the experiment:

$Exp^{kr}_F(B)$
{ Randomly select a key k;
$k' \leftarrow B^{F_k}$;
If $k = k'$ then return 1 else return 0;
}

The kr-advantage of B is defined as:
$Adv_F^{kr}(B) = Pr[Exp_F^{kr}(B) = 1]$
For F to be secure, we should have $Adv_F^{kr}(B)$ "small" for all adversaries B with "reasonable" resources.

Example 3: Let $F: \{0,1\}^k \times \{0,1\}^l \to \{0,1\}^L$ be the family of functions from example 1. Its prf-advantage was very high. Now we will compute its kr-advantage. The following adversary B recovers the key. We let e_j be the l-bit binary string having a 1 in position j and zeroes everywhere else.

Adversary B^{F_K}
{ $K' \leftarrow \epsilon$; // ϵ is empty string
  for $j = 1, \ldots, l$ do
  { $y_j \leftarrow F_K(e_j)$;
    $K' \leftarrow K' \| y_j$;
  }
  return K';
}

$Adv_F^{kr}(B) = 1$

Time complexity is $O(l^2 L)$.

The Birthday Attack: Let $E: \{0,1\}^k \times \{0,1\}^l \to \{0,1\}^l$ be a family of permutations. Suppose q satisfies $2 \le q \le 2^{(l+1)/2}$. Then there is an adversary A, making q oracle queries and having running time that to do q computations of E, such that $Adv_E^{prf}(A) \ge (0.3)\frac{q(q-1)}{2^l}$.

Adversary A is given an oracle $g: \{0,1\}^l \to \{0,1\}^l$.

Adversary A^g

{ for $i = 1, \ldots, q$ do
  { Let x_i be the i-th l-bit string in lexicographic order.
    $y_i \leftarrow g(x_i)$;
  }
  if $y_1, \ldots, y_q$ are all distinct then return 1, else return 0.
}

$$Adv_E^{prf}(A) = Pr[Exp_E^{prf-1}(A)=1] - Pr[Exp_E^{prf-0}(A)=1]$$
$$= 1 - [1 - C(N,q)] = C(N,q) \ge (0.3)\frac{q(q-1)}{2^l}.$$

Here $C(N,q)$ is the probability that some bin gets two or more balls in the experiment of randomly throwing q balls into N bins.

97

[R1] Goldwasser and Bellare, Lecture Notes on Cryptography.

CHAPTER A

The birthday problem

A.1 The birthday problem

Some of our estimates in Chapters 6, 9 and 5 require precise bounds on the birthday probabilities, which for completeness we derive here, following [12].

The setting is that we have q balls. View them as numbered, $1, \ldots, q$. We also have N bins, where $N \geq q$. We throw the balls at random into the bins, one by one, beginning with ball 1. At random means that each ball is equally likely to land in any of the N bins, and the probabilities for all the balls are independent. A collision is said to occur if some bin ends up containing at least two balls. We are interested in $C(N, q)$, the probability of a collision.

The birthday phenomenon takes its name from the case when $N = 365$, whence we are asking what is the chance that, in a group of q people, there are two people with the same birthday, assuming birthdays are randomly and independently distributed over the 365 days of the year. It turns out that when q hits $\sqrt{365} \approx 19.1$ the chance of a collision is already quite high; for example at $q = 20$ the chance of a collision is at least 0.328.

The birthday phenomenon can seem surprising when first heard; that's why it is called a paradox. The reason it is true is that the collision probability $C(N, q)$ grows roughly proportional to q^2/N. This is the fact to remember. The following gives a more exact rendering, providing both upper and lower bounds on this probability.

Proposition A.1 Let $C(N, q)$ denote the probability of at least one collision when we throw $q \geq 1$ balls at random into $N \geq q$ buckets. Then

$$C(N, q) \leq \frac{q(q-1)}{2N}.$$

Also

$$C(N, q) \geq 1 - e^{-q(q-1)/2N},$$

and for $1 \leq q \leq \sqrt{2N}$

$$C(N, q) \geq 0.3 \cdot \frac{q(q-1)}{N}.$$

∎

In the proof we will find the following inequalities useful to make estimates.

Proposition A.2 For any real number $x \in [0,1]$—

$$\left(1-\frac{1}{e}\right)\cdot x \;\le\; 1-e^{-x} \;\le\; x\,.$$

∎

Proof of Proposition A.1: Let C_i be the event that the i-th ball collides with one of the previous ones. Then $\mathbf{Pr}[C_i]$ is at most $(i-1)/N$, since when the i-th ball is thrown in, there are at most $i-1$ different occupied slots and the i-th ball is equally likely to land in any of them. Now

$$\begin{aligned} C(N,q) &= \mathbf{Pr}[C_1 \vee C_2 \vee \cdots \vee C_q] \\ &\le \mathbf{Pr}[C_1] + \mathbf{Pr}[C_2] + \cdots + \mathbf{Pr}[C_q] \\ &\le \frac{0}{N} + \frac{1}{N} + \cdots + \frac{q-1}{N} \\ &= \frac{q(q-1)}{2N}\,. \end{aligned}$$

This proves the upper bound. For the lower bound we let D_i be the event that there is no collision after having thrown in the i-th ball. If there is no collision after throwing in i balls then they must all be occupying different slots, so the probability of no collision upon throwing in the $(i+1)$-st ball is exactly $(N-i)/N$. That is,

$$\mathbf{Pr}[D_{i+1} \mid D_i] \;=\; \frac{N-i}{N} \;=\; 1-\frac{i}{N}\,.$$

Also note $\mathbf{Pr}[D_1] = 1$. The probability of no collision at the end of the game can now be computed via

$$\begin{aligned} 1-C(N,q) &= \mathbf{Pr}[D_q] \\ &= \mathbf{Pr}[D_q \mid D_{q-1}]\cdot \mathbf{Pr}[D_{q-1}] \\ &\;\;\vdots \quad \vdots \\ &= \prod_{i=1}^{q-1} \mathbf{Pr}[D_{i+1} \mid D_i] \\ &= \prod_{i=1}^{q-1} \left(1-\frac{i}{N}\right)\,. \end{aligned}$$

Note that $i/N \le 1$. So we can use the inequality $1-x \le e^{-x}$ for each term of the above expression. This means the above is not more than

$$\prod_{i=1}^{q-1} e^{-i/N} \;=\; e^{-1/N-2/N-\cdots-(q-1)/N} \;=\; e^{-q(q-1)/2N}\,.$$

Putting all this together we get

$$C(N,q) \;\ge\; 1-e^{-q(q-1)/2N}\,,$$

which is the second inequality in Proposition A.1. To get the last one, we need to make some more estimates. We know $q(q-1)/2N \le 1$ because $q \le \sqrt{2N}$, so we can use the inequality $1-e^{-x} \ge (1-e^{-1})x$ to get

$$C(N,q) \;\ge\; \left(1-\frac{1}{e}\right)\cdot\frac{q(q-1)}{2N}\,.$$

A computation of the constant here completes the proof. ∎