AryanNath - Assignment 4

Aryan Nath

April 2025

1. Let $\Pi = (\mathcal{M}, \mathcal{K}, \mathcal{C}, \texttt{Enc}, \texttt{Dec})$ be a symmetric-key encryption scheme. Suppose $\xrightarrow{K} \mathcal{K}$ and $\xrightarrow{M} \mathcal{M}$ be two random variables. Π is called **perfectly secure** if for all $m \in \mathcal{M}$ and $c \in \mathcal{C}$ we have

$$\mathbb{P}[M = m \mid C = c] = \mathbb{P}[M = m]$$

Prove or refute the following:

(a) Assume $|\mathcal{K}| = |\mathcal{C}| = |\mathcal{M}|$. Π provides perfect secrecy if and only if every key is used with equal probability $\frac{1}{|\mathcal{K}|}$, and for every $m \in \mathcal{M}$ and every $c \in \mathcal{C}$, there is a unique key k such that $\mathrm{Enc}(m, k) = c$.

(b) Π provides perfect secrecy if and only if for every probability distribution $\mathbb{P}_M$ over $\mathcal{M}$, every $m_0, m_1 \in \mathcal{M}$, and every $c \in \mathcal{C}$, we have:

$$\mathbb{P}[C = c \mid M = m_0] = \mathbb{P}[C = c \mid M = m_1].$$

10 marks

1. The proofs for part (a) and (b) are as follows:

(a) Given $|\mathcal{K} = |\mathcal{C}| = |\mathcal{M}|$, we prove this statement as follows:

Forward Implication:

We used Baye's theorem on $P[M = m|C = c]$ to get:

$$P[M = m|C = c] = \frac{P[C = c|M = m] \times P[M = m]}{P[C = c]}$$

Given $P[M = m|C = c] = P[M = m]$, we simplify this as:

$$P[C = c|M = m] = P[C = c]$$

Hence, for all messages m, the probability $P[C = c|M = m]$ is constant.

Key uniqueness: If there was no key k such that $Enc(m, k) = c$ then $P[C = c|M = m] = 0$. But $P[C = c] > 0$ as c is in the ciphertext space, so there is some message for which c is produced. Now if there are multiple keys such that $Enc(m,k) = c$, then using the pigeonhole principle ($|\mathcal{K}| = |\mathcal{C}| = |\mathcal{M}|$) there is an (m,c) pair which does not have a corresponding key k, which we have already shown to arise a contradiction. Hence, for each $m \in \mathcal{M}$ and $c \in \mathcal{C}$, there is a unique key k such that $Enc(m,k) = c$.

Uniform probability for choosing the key: Given that $P[C = c|M = m] = \frac{1}{|\mathcal{K}|}$, using key uniqueness we can state that $P[k = K] = P[C = c|M = m]\frac{1}{|\mathcal{K}|}$. Since each m, c pair has a unique k associated to it, for every $k \in \mathcal{K} P[K = k] = \frac{1}{|\mathcal{K}|}$.

Reverse implication:
Given there is exactly one key k with $Enc(m,k) = c$ and each key is chosen with probability $\frac{1}{|\mathcal{K}|}$, we simplify as follows:

$$P[M = m|C = c] = \frac{P[C = c|M = m] \cdot P[M = m]}{P[C = c]}$$

$$\Rightarrow P[M = m|C = c] = \frac{P[K = k] \cdot P[M = m]}{P[C = c]} = \frac{\frac{1}{\mathcal{K}} \cdot P[M = m]}{P[C = c]}$$

$$\Rightarrow P[M = m|C = c] = \frac{\frac{1}{|\mathcal{C}|} \cdot P[M = m]}{P[C = c]}$$

Since,

$$P[C = c] = \sum_m P[C = c|M = m] \times P[M = m] = \frac{1}{|\mathcal{C}|} \times \sum_m P[M = m] = \frac{1}{|\mathcal{C}|}$$

Therefore,

$$P[M = m|C = c] = \frac{\frac{1}{|\mathcal{C}|} \cdot P[M = m]}{\frac{1}{|\mathcal{C}|}}$$

$$\Rightarrow P[M = m|C = c] = P[M = m]$$

(b) **Forward Implication:**

Supose the $\prod$ provides perfect secrecy, then we can simplify $P[C = c|M = m_i]$ as:

$$P[C = c|M = m_i] = \frac{P[M = m_i|C = c] \times P[C = c]}{P[M = m_i]} = \frac{P[M = m_i] \times P[C = c]}{P[M = m_i]} = P[C = c]$$

Hence, $P[C = c|M = m_i] = P[C = c]$ is only dependent on c and not M. Therefore, for every $m_0, m_1 \in \mathcal{M}$, and every $c \in \mathcal{C}$, we have:

$$P[C = c|M = m_0] = P[C = c|M = m_1]$$

Reverse Implication:

Given that for every $m_0, m_1 \in \mathcal{M}$, and every $c \in \mathcal{C}$, we have:

$$P[C = c|M = m_0] = P[C = c|M = m_1] = \tau$$

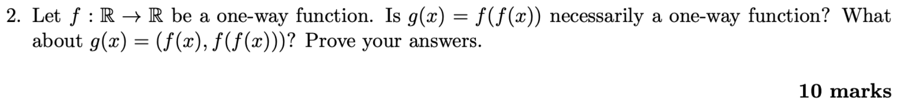
Therefore,

$$P[M = m|C = c] = \frac{P[C = c|M = m] \times P[M = m]}{P[C = c]} = \frac{\tau \times P[M = m]}{P[C = c]}$$

Now, $P[C = c] = \sum_m P[C = c|M = m] \times P[M = m] = \tau \sum_m P[M = m] = \tau$. Substituting this in the above equation gives us:

$$P[M = m|C = c] = \frac{\tau \times P[M = m]}{\tau}$$
$$\Rightarrow P[M = m|C = c] = P[M = m]$$

for every $m \in \mathcal{M}$ and $c \in \mathcal{C}$. Therefore, $\prod$ provides perfect secrecy.



2. Let $f : \mathbb{R} \to \mathbb{R}$ be a one-way function. Is $g(x) = f(f(x))$ necessarily a one-way function? What about $g(x) = (f(x), f(f(x)))$? Prove your answers.

10 marks

2. Given a one-way function $f : \mathcal{R} \to \mathcal{R}$, we know that $f = f^{-1}$ as then computing $f^{-1}(f(x)) = x$ which contradicts the property of one way functions that it is hard to find x given $f(x)$. So we only consider the case that $f(x) \neq f^{-1}(x)$.

   (a) Given $g(x) = f(f(x))$. Since $f(x) \in dom(f)$ we know that it is hard to find $f(x)$ given $f(f(x))$ based on the one wayness of $f(x)$. So given $g(x) = f(f(x))$, it is hard to find $f(x)$ and thus is it is hard to x. Therefore, $g(x)$ is a one way function.

   (b) Given g(x) = (f(x),f(f(x))). Since $f(x)$ is a one-way function, we know that it is hard to find x given $f(x)$, and we proved in part (a) that given $f(f(x))$ it is hard to find x. Hence, given $g(x) = (f(x), f(f(x)))$ it is hard to find x. Therefore, g is a one-way function.

3. Given the subroutine $\mathcal{A}$, the algorithm for finding a non-trivial factor of N is as follows:

3. Let N be an odd composite integer. Suppose there exists an algorithm $\mathcal{A}$ that, given an integer x coprime to N, computes the order of x modulo N. Show that there exists an algorithm to find a non-trivial factor of N, using $\mathcal{A}$ as a subroutine, with only polynomial overhead. You may use the following theorems:

   (a) If $x^2 \equiv 1 \pmod N$ and $x \not\equiv \pm 1 \pmod N$, then at least one of $\gcd(x-1, N)$ or $\gcd(x+1, N)$ is a non-trivial factor of N.

   (b) If N has m distinct prime factors, the probability that a random x coprime to N has even order r and $x^{r/2} \not\equiv \pm 1 \pmod N$ is at least $1 - \frac{1}{2^m}$.

   Your solution should describe the algorithm, justify its correctness using the theorems above, and analyze its running time.

20 marks

(a) Algorithm:

   i. Generate a random x in Z_N^*.

   ii. Use $\mathcal{A}$ to get the order of $x \pmod N$ as s.

   iii. Given theorem (b) we know that given x^s, we have that $x^{s/2} \not\equiv \pm 1 \pmod N$ with probability at least $1 - \frac{1}{2^m}$.

   iv. Now consider $x'^2 = x^s$ and $x' = x^{s/2}$. If $x^{s/2} \not\equiv \pm \pmod N$ then using theorem (a) we know that at least one of $\gcd(x-1, N)$ or $\gcd(x+1, N)$ is a non-trivial factor of N. If not, them repeat step *ii* with $x^{s/2} \pmod N$.

   v. We either terminate in step *iii* or when we reach $x \equiv \pm 1 \pmod N$. If we reach the second case then we start from step *i*.

(b) Correctness:

   - Since N is composite, we know that its non-trivial factor exists.
   - If the algorithm terminates in step *iii* then we know that by the correctness of theorem (a), the algorithm with terminate and give the correct output.
   - In case the algorithm does not terminate at step *iii* and we have to restart from step *i* with a different x, then we know that the number of times that we will have to restart will be very less, as for a large N (which has a large number of prime factors) we can use theorem (b) to claim that our possibility of repeating in an endless cycle is very small. Hence, for some value of $x \in Z_N^*$ we will terminate at step *iii*, which we have already claimed to return the correct output by the correctness of theorem (a).

(c) Time Complexity:

   The subroutines used and their corresponding complexities are:

   i. $\mathcal{A} : O(f(N))$.

   ii. $gcd : O(logN)$.

iii. Number of repetitions from step *iii* to step *ii*: $O(logN)$ (as the order of every element in Z_N^* is $|Z_N^*|$).

iv. Number of times we have to repeat from step v to step i is constant for any N with a large number of prime factors: O(1).

Hence, the total time complexity of the algorithm is $O(\log N(f(N)+\log N)) = O(\log N f(N) + \log N \log N$, which still has polynomial overhead in N.

4. Two standard dice have faces labeled 1 through 6. When rolled together, the possible sums range from 2 (if both show 1) to 12 (if both show 6), and every whole number in between is achievable. Now, imagine a twist: instead of regular dice, you have two blank cubes. You get to choose the numbers that go on each of the six faces of both dice. The two dice do not need to be identical, and you may use any positive integers you like (including repeats if necessary). Your task is to label these two dice such that, when rolled together, the sum of their faces yields all integers from 1 to 36.

   (a) Prove whether such a labeling is possible.

   (b) If yes, construct such a pair of dice and justify your design.

   (c) Explore whether other such dice exist that achieve the same 1–36 range. Can you generate a generalized formula or approach?

30 marks

4. (a) Since there are 6 faces on each of the two dice, the total number of unique combinations that we can get from the number of these faces is $6 \times 6 = 36$. Since this is equal to the number of integers that we want to generate by combining the numbers on the faces using sum, the sum of the numbers on any pair of faces has to be unique.

   To prove that we will be able to generate all integers $1 \ldots 36$ using the 6 faces of the two die, it is sufficient to prove that (1) we can generate $\min(\{1 \ldots 36\}) = 1$ and $\max(1 \ldots 36) = 36$, and (2) we each sum is unique.

   1) We can construct 1 using $1 + 0 = 1$, and we can construct 36 using $i + j = 36$ s.t. $i \geq 30 \,\&\, j \geq 0$.

   2) If we are able to create $D_1 : \{1, a_2, \ldots, a_6\}$ and $D_2 : \{0, b_2, \ldots, b_6\}$ then such that each sum $a_i + b_j$ is unique, with the minimum and maximum sums being 1 and 36 respectively, then since there are a total of $6 \times 6 = 36$ unique sums, will have generated the entire sequence of integers $1, \ldots, 36$.

   (b) D1: {1,2,3,4,5,6} and D2: {0,6,12,18,24,30}.

   Why this works:

i. $1 + 0 = 1$.

ii. $6 + 30 = 36$.

iii. Each sum is unique as for each $i \in D_1$, the sum of i with all terms in D_2 is $6q + r, 0 \leq q \leq 6, 1 \leq r \leq 6$. Hence, each $i \in D_1$ defines a unique increment i from the lower limit of each of the 6 intervals from $1, \ldots, 36$ when we take these integers in order from $1 \ldots 36$ and distribute them into the intervals $\{1, \ldots 6\}, \{7, \ldots, 12\}, \ldots, \{31, \ldots, 36\}$. Hence, when we pair each i with all values from D_2 we will get a unique sum every time. Therefore, we will generate the set $1, \ldots, 36$.

(c) Another possible combination of die pairs is $D_1 : \{0, 1, 2, 3, 4, 5\}, D_2 : \{1, 7, 13, 19, 25, 31\}$ as each of the sums are equal to the sums we achieved in the previous D_1, D_2 combination.

In general, if we take $D_1 : \{1 - z, 2 - z, 3 - z, 4 - z, 5 - z, 6 - z\}, D_2 : \{z, z + 6, z + 12, z + 18, z + 24, z + 30\}$ we will get the same sums as the original combination of D_1 and D_2. Since the numbers in the die faces have to all be positive, the only possible values of z are $z = 0, 1$. Without this constraint, it would work for any value of z.

5. Design a zero-knowledge proof protocol for the 3-coloring of graphs. Specifically, given a graph $G = (V, E)$ that is claimed to be 3-colorable, describe an interactive protocol in which a prover can convince a verifier that G is 3-colorable *without revealing* the actual coloring. Show that your protocol satisfies completeness, soundness, and zero-knowledge.

30 marks

5. Given a graph $\mathcal{G}$, the prover claims to the verifier that they have a 3-colouring of the graph. The prover has to prove their claim to the verifier without revealing the colouring itself. The zero knowledge interactive proof between the prover and the verifier will use the facts that if we have a valid 3-coloring of a graph, then after permuting the 3 colors, we will still have a valid 3-coloring and a 3-coloring is valid only if there exist no two adjacent vertices with the same color.

Interactive Proof:

1. **Prover:** Given a valid 3-coloring W of a graph, the prover randomly permutes the 3-colors to obtain a new 3-coloring for the graph. For each vertex $v_i \in V$, the prover uses a commitment scheme like Pedersen or polynomial commitments to commit to the colors of each vertex after the color permutation:

$$\forall i \in \{1, \ldots, |V|\}, c_i = com(v_i, \text{color of } v_i)$$

[which hides the color of each vertex and binds the vertex and its color to the commitment]

2. **Verifier:** The verifier then randomly picks an edge $e_{i,j} \in E$ and sends $e_{i,j}$ to the Prover.

3. **Prover:** The prover sends an opening proof for c_i and c_j to the verifier, and the verifier open c_i and c_j (the colors of vertices v_i and v_j) only if the proofs are correct.

4. **Verifier:** The verifier returns *acc* if $c_i \neq c_j$, and rejects otherwise.

5. If this interactive proof is repeated $O(|V||E|)$ times, then the probability that the prover has tricked the verifier is $\epsilon(|V|) = 2^{-|V|}$ (ϵ is a negligible function in $|V|$).

   The proofs for the completeness and soundness of the interactive proof are as follows:

   (a) **Completeness:** If the witness W is a valid 3-coloring of $\mathcal{G}$, then the regardless of what edge the verifier picks, the verifier will find that $c_i \neq c_j$ and will return *acc*.

   (b) **Soundness:** To show that the interactive proof is sound, we need to show that if the prover gives an invalid 3-coloring W of $\mathcal{G}$ then the verifier will accept the witness with probability:

   $$Prob[\text{Verifier returns } acc] \leq \epsilon(|V|)$$

   Since W is an invalid 3-coloring of $\mathcal{G}$, there has to exist an edge at least one edge $e_{i,j}$ such that $c_i = c_j$. Therefore,

   $$Prob[\text{Verifier returns } reject] \geq Prob[\text{Verifier picks } e_{i,j}] = \frac{1}{|E|}$$

   If we repeat the protocol t times, the probability that the verifier accepts an invalid W is:

   $$Prob[\text{Verifier returns } acc] \leq \left(1 - \frac{1}{|E|}\right)^t \leq \epsilon(|V|)$$

   Now, we prove that the interactive proof is zero-knowledge. A protocol is zero-knowledge if and only if the distributions

   $$\{view_{V^*}[P(x,y) \leftrightarrow V^*(x,z)]\}$$

   $$\{S(x,z)\}$$

   are distinguishable with probability at most $\epsilon(|V|)$.

   (c) **Zero-Knowledge:** Before we go through a formal proof for the zero-knowledge, we can observe that since the colors of the vertices are being permuted we never really get to know the original 3-coloring of the graph despite opening the commitments c_i and c_j. Hence, the graph coloring of the prover remains zero-knowledge to the verifier. Now, we give a formal proof for the interactive proof being zero-knowledge:

We construct a simulator S, which has the code for a p.p.t. adversary V^* and the Verifier, and it works as below:

i. $e'_{i,j} \overset{\$}{\leftarrow} V \times V$ and create 2 commitments c'_i, c'_j for 2 different random colors. For every other vertex $v_t, t \neq i, j$ let $c_k = 0$ (the zero string).

ii. Send the first message to V^* and get $e_{i,j}$ from V^*.

iii. If $e_{i,j} = e'_{i,j}$ then open c'_i, c'_j, otherwise go to step 1.

In order to prove that the protocol is zero-knowledge, we need to prove that the transcript the simulator S is indistinguishable from the transcript of the real-world protocol. Intuitively, the only difference between the real-world protocol and S is that in the real-world protocol, all of the vertices commit to a color, whereas in S most of the vertices have a zero string commitment. But all of these zero string commitments will never be opened, so by the hiding property of the commitment, all of the zero string commitments look identical to the commitments of the vertices of the real protocol.

Now, we will use a hybrid proof to show that the two transcripts are indistinguishable. Let H_0 be the description of the real-world protocol and let H_3 be the description of the simulator S.

- The algorithm S_0 from the description H_0 has the correct witness W and code of V^*; S_0 acts as an honest Prover and then interacts with V^* which results in:

  i. Commit the colors of the vertices and compute the first message honestly with witness W.

  ii. Get $e_{i,j}$.

  iii. Open c_i, c_j and if $c_i \neq c_j$ then return acc, else return reject.

  Output the transcript τ_0, here τ_0 has the same distribution as in the real protocol.

- Algorithm S_1 from the description H_1 has the correct W and code of V^*, S_1 guesses a random edge $e'_{i,j}$. S_1 acts as an honest prover and interacts with V^*, which results in:

  i. Commit to the colours of the vertices and compute the first message honestly with witness W.

  ii. Get $e_{i,j}$.

  iii. Now if $e_{i,j} \neq e'_{i,j}$ go back to Step 1, else open c_i, c_j and if $c_i \neq c_j$ then return acc, else return reject.

  Then output the transcript τ_1.

- The algorithm S_2 of the description H_2 has the correct witness and code of V^*, and S_2 guesses a random edge $e'_{i,j}$. S_2 then computes the first message using $e'_{i,j}$, which results in:

    i. S_2 commits to the coloring of every vertex to be zero for every $c_t, t \neq i, j$. c'_i and c'_j are being computed honestly using W.
    ii. Get $e_{i,j}$ from V^*.
    iii. If $e_{i,j} \neq e'_{i,j}$ then go back to step 1, else open c'_i, c'_j. If $c_i \neq c'_j$ then return *acc* else return reject.

  Output the transcript τ_2.

- The simulator algorithm S is from the description H_3 with code of V^* and S guesses a random edge $e'_{i,j}$, which results in:
    i. S commits to the coloring of all vertices to be zero where $k \neq i, j$. c'_i and c'_j are being computed honestly.
    ii. Get $e_{i,j}$ from V^*.
    iii. Now if $e_{i,j} \neq e_{i,j}$ go back to step 1, else open c'_i, c'_j and if $c'_i \neq c'_j$ then return *acc*, else return reject.

  Output the transcript τ_3.

Now, H_0 is indistinguishable from H_1 as the only difference between the two is that H_1 randomly chooses $e'_{i,j}$ until $e_{i,j}$ is found. Hence, their transcripts have identical distribution. Using the following lemma and short proof we can also get that H_1 is indistinguishable from H_2.

Lemma 2: The distribution of τ_1 = The distribution of τ_2.

Proof: Suppose there exists a distinguisher $\mathcal{D}$ that can successfully differentiate between τ_1 and τ_2; then they can break teh hiding property of the commitment scheme, as then all commitments which are not opened can be seen by this algorithm and this algorithm would output τ_1 or τ_2, however, this would result in a contradiction with the commitment scheme.

H_2 is indistinguishable from H_3 as the only difference between the two is the H_3 assigns a random coloring to the v_i and v_j instead of utilizing the witness W. However since we permute the colors in the first step, the witness W commits c_i and c_j randomly. Therefore, the distribution of τ_2 = distribution of τ_3.