19

CSCI 5440: Theory of Cryptography	2020-2021 Term 1

Assignment 3

Student: Yu Zheng — Student ID: 1155113945

Question 1

- If G is a (s,ϵ)-pseudorandom generator,the output of $G(X)$ is computationally indistinguishable to a uniform random string R. When $s=0$, $G(X)$ is indistinguishable to a random R; When $s=1$, $G(x)+R$ is also distinguishable to a random R. Before revealing, on the view of the receiver, the receiver can not know $s=0$ or $s=1$. This holds since both $G(X)$ and $G(X)+R$ are computationally indistinguishable to a random string. Now, let's work out the parameters. To know x by $G(X)+s\cdot R$, the receiver only needs to know the $G(X)$. Since G is a (s,ϵ)-pseudorandom generator, the commitment should be (s,ϵ)-hiding.

- Let's consider this question conversely, *i.e.*, finding out X and X' such that $G(X)+G(X')=R$. If $X=X'$, we then know $G(X)=G(X')$. The probability of this case is $\sum_{2^k}\dfrac{1}{2^k\cdot 2^k}=\dfrac{1}{2^k}$, where $2^k\cdot 2^k$ is the possibilities of XX'. If $X\neq X'$, the probability of $G(X)=G(X')$ is with complexity $O(\frac{1}{2^{3k}})$ (I am not sure about this case, but this case does not matter since the probability is much smaller than $1/2^k$). If $G(X)=G(X')$, $G(X)+G(X')=R$ exists when $s=0,s'=1$ or $s=1,s'=0$. This holds since either $G(X)+(G(X')+R)$ or $(G(X)+R)+G(X')$ is equal to R. Therefore, for the case of non-existing X,X', the probability is $1-2^{-k}$.

10

- The binding refers that the probability that a (malicious) sender S^* of size at most s can make receiver validate two different X,X' on the same commitment is negligible. This happens if the $G(X)=G(X')+R$ (or $G(X)+R=G(X')$), *i.e.*, for both cases $s=0$ and $s=1$. By Q1(b), we know that the probability is 2^{-k}, which is negligible. Therefore, the commitment is $(s,2^{-k})$-binding.

WHAT IS S?

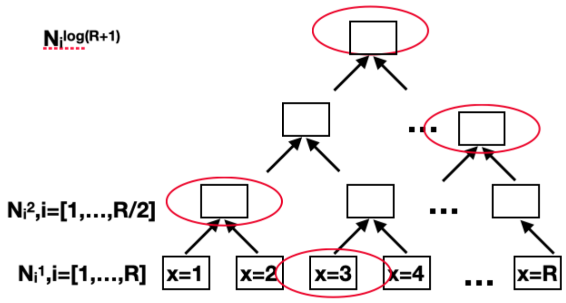
Question 2

- A database querying scheme works as defined in Question 2. Following the same definitions and notations, we define (s,ϵ)-security of this scheme under a malicious adversary:

  Definition 1. (s,ϵ)*-secure querying. The probability that a malicious Bob B^* of size at most s can pass Alice's verification by cheating with y^* and $cert^*$ such that $y^*\neq D(x)$ is at most ϵ.*

- Verification: If $com=h_K(cert)$, output *accept*; otherwise *reject*, where K is a public key.

  Proof: The security of this scheme is based on the commitment scheme. A commitment scheme needs to satisfy the properties of hiding and binding.



9

Figure 1: Merkle tree-based certification

WHY $1/2^{Rn}$?

Hiding: This requires that the commitment hides D from Alice before disclosure (receiving the *cert*). The one-way property of hash function h guarantees that Alice cannot retrieve D by only viewing the hash of D. The *cert* of size Rn is indistinguishable to a random sting. Then, we can easily know the commitment is $(s, 1/2^{Rn})$-hiding, otherwise h breaks.

Binding: This requires that a malicious Bob B^* can not find two different inputs to pass Alice's verification. If B^* succeeds to find two different D, D', which implies that B^* finds a collision of h_K. This contradicts the assumption that h is a collision-resistant hash function. Suppose h_K in the size of t is (s, ϵ)-collision-resistant. The construction of R copies is $(s-(t-1)R, R\epsilon)$-collision resistant. Thus, the commitment is $(s-(t-1)R, R\epsilon)$-binding.

By analysis above, we proved the protocol is secure.

- Construct a Merkle tree as Figure 1. Each node of the tree is denoted by N_i^l, where N_i^l is the i-th node at "level" l, $l \in [1, R+1]$. Here, I use "level" just for explanation, while the formal definition of level counting from the root is depth+1. The bit-length of each node is n. The leaf nodes at first level is $D(x)$, $x \in [1, R]$, and nodes at the level l is computed from the nodes at the level $l-1$ by a hash function $h_M : \{0,1\}^{2n} \to \{0,1\}^n$.

  Certification: all nodes on the path from the root to Alice's input query x. The level is $\log(R+1)$, and each node has n bits. Thus, the length of the certification is $n\log(R+1)$. For example, if Alice's query is $x=4$, the certification is constructed by the nodes in red circles (see Figure 1).

  Verification: By $D(x)$ and $D(x')$ such that $x' = x+1$ for odd x and $x' = x-1$ for even x, Alice verifies the node at $l=2$. By the nodes from the certification and the computed node above, Alice verifies the new node at $l=3$. Similar to above, Alice verifies the nodes on the path to input query towards the root node.

  Proof: The collision resistant hash guarantees the binding of the commitment since a malicious Bob can not find different inputs to pass Alice's verification. The one-way property guarantees that Alice cannot retrieve the inputs of *com* before Bob sends the *cert* and y to Alice. Now, let's work out the parameters. Suppose h_M is a (s, ϵ)-collision resistant hash in the size of t. By Theorem 4 in Lecture 6, we know that the Merkle tree with $O(\frac{1}{2}R(R-1))$ (*i.e.*, the number of nodes) copies is $(s - t\log\left(\frac{1}{2}R(R-1)\right), \frac{1}{2}R(R-1)\epsilon)$-collision resistant, which guarantees the binding property. As for hiding, the certificate of size $n\log(R+1)$ is indistinguishable to a random string, which implies $(s, \underbrace{1/2^{n\log(R+1)}})$-hiding. Therefore, the construction is secure.

HOW DO YOU GET THIS?