Pham Hoang Long Do
NetID: 2995060

# Homework 4

CSE 3400
Ghada Almashaqbeh

## Problem 1

1. Construct a new hash function $h'$ as follows: $h'(m) = h_1(m_0) \oplus h_2(m_1)$ such that $m = m_0 \| m_1$. If $h_2$ is a CRHF, will $h'$ be a CRHF? Justify your answer.
   - $h^{'}$ is CRHF in a sense that:
   (a) Given that $h_1(m_0)$ and $h_2(m_1)$ are CRHF and has the output $0, 1^n$. Therefore, according to the Birthday paradox, the probability for an attacker to find a collision $h_1(m) = h_1(m^{'})$ and $h_2(m) = h_2(m^{'})$ such that $m \neq m^{'}$ are equal to $2^{\frac{n}{2}}$.
   (b) Since the output of $h^{'}$ is $0, 1^n$. If there exist an attacker who tries to find a collision, he also has an equal chance of $2^{\frac{n}{2}}$ to find a collision which is equal to the probability of finding a collision for $h_1$ and $h_2$. $\epsilon^{CRHF}_{h',A} = \epsilon^{CRHF}_{h_1,A} = \epsilon^{CRHF}_{h_2,A}$
   (c) Since $h'$ yields the same probability as $h_1, h_2$ (CRHF). Therefore, $h'$ is also collision resistance.
2. Construct a new hash function $h''$ as follows: $h''(m) = m_0 \oplus h_1(m_1)$ such that $m = m_0 \| m_1$ and $|m_0| = n$. Is $h''$ a CRHF? Justify your answer.
   - $h''$ is **not CRHF** in a sense that:
   (a) An attacker could submit a first message $n = n_0 \| n_1$. The output after run this message to $h''$ will be: $\sigma = n_0 \oplus h_1(n_1)$
   (b) The second message that he can submit will be in the form: $n^{'} = n_0 \oplus h_1(n_1) \oplus h_1(\alpha) \| \alpha$. Where $\alpha$ could be anything of any size and $h_1(\alpha)$ has output of n.
   (c) After plug $n^{'}$ to $h''$ we have: $n_0 \oplus h_1(n_1) \oplus h_1(\alpha) \oplus h_1(\alpha) = n_0 \oplus h_1(n_1) = \sigma$. Which is a collision.
   (d) The probability to find the collision is always 1. Therefore, using this construction guarantee a zero percent chance collison resistance.
3. Construct a new hash function $h'''$ as follows: $h'''(m) = h_2(h_1(m))$. If $h_2$ is a OWF, will $h'''$ be a OWF? Justify your answer.
   - $h'''$ is a **OWF** since it's primarily constructed by $h_2$ which is another OWF function, and by the definition of OWF. An advesary can't come up with $m$ or $m'(m \neq m')$ such that $h(m) = h(m')$.
4. Construct a new hash function $h''''$ as follows: $h''''(m) = h_1(m_0) \| h_2(m_1)$ such that $m = m_0 \| m_1$. If $h_2$ is an SPR function, will $h''''$ be a CRHF? Justify your answer.
   - $h''''$ is not CRHF in a sense that:
   (a) The first part of the construction, $h_1(m_0)$ is CRHF, so the attacker would not focus on this. In other word, we will keep the first part of the message ($m_0$) the same.
   (b) The second part of the construction, $h_2(m_1)$ is SPR, which is a weaker security notation than CRHF. He will focus on this.
   (c) The two messages that he will submit will be like:
   $$m = m_0 \| m_1 \text{ and } m'' = m_0^{'} \| m_1^{'} \text{ which } (m_0 = m_0^{'} \text{ and } m_1 \neq m_1^{'})$$
   (d) the probability to find the collison of $h_2$ will be equal the probability of finding collison of the SPR function. That is:
   $$\epsilon^{CRHF}_{h_2,A} = \epsilon^{SPR}_{h_2,A}$$

## Problem 2

1. **(20 points)** Bob has a database composed of 16 files, each of which is of size 60 KB, and he constructed a Merkle tree over this database (i.e., the database files are the leaves of the tree). The files are named as $f_1, f_2, \ldots, f_{16}$, and when constructing the tree these files are ordered from left to right, i.e., $f_1$ is the leftmost leave and so on. Bob sent the tree root to Alice. Based on that answer the following:
   (a) How many levels does the resulting Merkle tree have?
   - There are **5 levels** for thie Merkle tree.
   (b) Assume Bob has used SHA384 for the hash function $h$ when constructing the tree. SHA384 has an output size of 48 bytes. What is the total size of the tree excluding the files?
   - Thereare **31 nodes** in this Three. Therefore, $31 nodes \times 48 = 1488 bytes$.
   (c) Bob wants to prove to Alice that file $f_{13}$ is a member of the tree. What is the proof of inclusion that Bob will send to Alice? what is the total size of that proof? How will Alice verify it?
   - Bob will send Alice the following:
      i. $h_1, h_2, ..., h_{12}$
      ii. The file $f_{13}$ itself.
      iii. The rest of $h_{14}, h_{15}, h_{16}$.
   - To verify, Alice will do the following:
      i. Hash $f_{13}$ on her side.
      ii. Hash $h_1$ and $h_2$ to obtain $h_{1-2}$, and keep do so for other hashes until she obtain $h_{3-4}, h_{5-6}, ..., h_{15-16}$.
      iii. Hash another level to obtain $h_{1-4}, h_{5-8}, h_{9-12}, h_{13-16}$.
      iv. Hash another level to get $h_{1-8}, h_{9-16}$.
      v. Hash the final level to get $h'_{1-16}$ and compare it with the orginal hash she received from Bob. If $h'_{1-16} = \sigma$, then $f_{13}$ is a member of the tree.
   (d) Will your answer change for part (c) if Bob wants to prove that file $f_4$ is a member of the tree? state all the changes.
   - Yes, my answer will change since now Bob need to send $f_4$, and $h_1, h_2, ..., h_{16}$ except $h_4$.
2. **(20 points)** Answer the following about blockchains:
   (a) Will mining be easier if the hash function used in proof of work is not a OWF? why? - Yes it is easier because, like what we discussed in class, if we don't have OWF, everyone can make a transaction by inverting some value $a$ that satisfies $a <$ difficulty tartget. That adversary can $invert(a)$ to get the value of the header.
   (b) Assume you want to use Bitcoin to buy a house (so you will issue a transaction $tx$ to buy the house). If you want the house to be bought when block number 100 is mined, what should you do? how can you later prove to someone that you bought the house when block 100 is mined? (here assume that any transaction you issue will be included

in the next block to be mined.)

i. First, a list of pointers to previous unspent transaction owned by the sender. This list is the input of our node.

ii. Second, The output of the node will be the address of the receiver or the hash of the output script.

iii. Lastly, sign the transaction using the private keys associated with the inputs.

- To verify, other miners check the validity of all transactions in a block, and then verify the solution of the hash puzzle.

(c) Will the mutability of blockchain break if the hash function we use to build it is only OWF? why?
- Yes it will break the chain since we need to include $H(prev - block)$ into the header of the current block, and basically hash of block = hash of header if our hash function $H$ is not CHRF then collisions will happen.

3. **(20 points)** Answer the following about the Merkle-Damgard Digest Function and its Extend function (shown in slides 9 and 11, lecture 7).

(a) Bob uses the Merkle-Damgard Digest Function for timestamping (to prove to Alice that he knows some document a year ago by posting only its hash then reveal the document to the public later). Will this application work if Alice and Bob are using different IV values? why? - No this application wont work if Bob and Alice use different IV since the construction, basically, chain together and IV is feeded into the first block, if different IV was used then the output of the first block will be different for Bob and Alice. This will lead to a chain effect which make the outputs of downstream blocks different.

(b) For the timestamping application, Mal claims that she knows Bob's document (while she does not) and one additional document she created by posting only one hash value covering both documents. How can Mal utilize the Merkle-Damgard extend function to achieve her goal and fool Alice that she knows both documents?
- The Merkle-Damgard construction is vulnerable to length extension attacks. Mal knows the output of a Merkle-Damgard hash function, and can easily compute the hash of new messages (The two that she claimed) that extend the original message without knowing the secret key used to compute the original hash. She can add her messages at the end and the Merkle-Damgrad exend fuction will keep do its job.