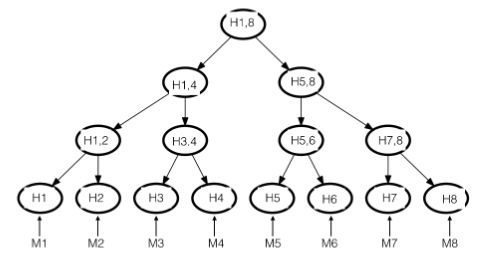
David Hatcher

U74842093

HW 1

1. One-Time Pad Basics [5]
   1. **One Time Pad (OTP) encryption is highly secure, but why do we not see it much in practice? Explain at least two reasons (some discussed in-class) [3p]**
      1. One of the reasons is that to achieve secure communication using one-time pad, the key used to generate the cyphertext can only be used once. Thus, requiring the sender and receiver to use a new key for each message they send. If the key is used more than once the adversary can intercept those messages and there will be an information leak. **(basics.pptx slide 10)**
      2. Another reason that OTP is not used very much is because the key used to generate the cyphertext must be a True Random Number and these are difficult to acquire. **(basics.pptx slide 10)**
   2. **Why does an ideal OTP encryption achieve information-theoretical security? [2pt]**
      1. OTP achieves information-theoretical security because the cyphertext gives no information about what the actual message contains. So even if an adversary intercepts the cyphertext they are left with the following equation to solve:  
         As the adversary only has the cyphertext, they would be unable to solve this even with unlimited computation power. **(basics.pptx slide 10)**
2. **Security of cryptographic hash function[10p]:**
   1. **What is the birthday attack against hash functions? Given an n-bit hash output, mathematically formulate the upper bound of collision probability of it. Specifically, write how birthday paradox is formulated. You can use approximations if needed.**
      1. The birthday attack against hash functions is the idea that if you were trying to achieve a 50% chance that anyone in a single room has the same birthday as you then you’d need to have 253 people in that room. But if you just wanted to have a 50% chance to find 2 random people in a room that share a birthday you’d only need 23 people to do this. This follows the strong-collision resistance idea, that if you needed to find a collision with a specific hash output you’d need to try a much larger amount of hash outputs before finding one that matches, however, if you were just trying to find two random hash outputs that match you’d likely need far fewer to accomplish that. **(CryptoHashFunctions.ppt slide 8)**
      2. Birthday Paradox Formula:  
         Where n is the number of people in the room.  
         Where n is the length of your hash output and h is the number of tries to find a collision. Based on this, the probability would approach 1 as the denominator approaches infinity, however the denominator is bound by (2n - h)!, thus the highest value h can have is 2^n. Thus, the upper bound of collision probability would occur as h -> 2^n. As such the upper bound is the following equation:
3. **Suppose a sender S uses this Merkle Hash Tree to authenticate these messages to receiver R.** 
   1. **How would one authenticate message M4?**
      1. **Write which elements must be transmitted from sender to the receiver [3p]**
         1. M4, H3, H12, H58
      2. **And write the correct verification equation as in the slides [2p]:**
   2. **Merkle hash trees have an additional level of hash in the leaves. Why? [2p]**
      1. This extra level of hashing means that there is less data being exposed across the network. Basically M3 could be sent over as well as M4 but that would allow an adversary to intercept both M3 and M4, while sending over H3 and M4 means that if an adversary were to intercept both of these they would only have M4 and the hash of M3 which cannot be reverted to M3. This means that less useful information can be leaked to adversaries.
   3. **What are two necessary conditions for a set of data to be authenticated by Merkle-tree? Two 2pt each total [4p]**
      1. The first necessary condition is that the Root node must be sent ahead of time over the secure channel. This is done so that the reciever has the value to check against the path and message being sent over the unsecure channel at a later time.
      2. The second condition is that all the leaf values must be known ahead of time. Since the root must be calculated and sent to the reciever before the verification can take place all of the leaf values are required to calculate that value. Since the sender will be transmitting the message to be verified along with the path of nodes they will have to calculate those nodes by hashing the different messages.
   4. **Describe how Merkle-hash trees are used for memory integrity protection in computer systems and in Cryptocurrencies. [4p]**
      1. In computers memory integrity protection can use Merkle-hash trees by having a value stored in the secure processors persistent storage which can be combined with values found on the untrusted storage location, when these values are hashed together they will produce a root that is also known by the secure processor and this will allow the memory integrity to be tested.
      2. With cryptocurrencies the Merkle-hash trees are used to generate the Root Hash of each block, with the nodes being the different transactions on that block. They can verify the integrity of the block by checking the an individual transaction against the different hashes and comparing the root of the hash path given with the root hash of the block.
4. **HMAC and Client Server Puzzles: [20p]**
   1. **What is HMAC [3p]** 
      1. **Write its formula [1p].**   
         *k’* = the key, padded on the right with 0’s to 512 bits  
         *m* = the message  
         *v1* = 0x363636…36  
         *v2* = 0x5c5c5c…5c  
         *h* = the hash function being used   
         + = concatenation
      2. **Why it is important to follow a specific padding mechanism (as opposed to basic concatenation of the key) in HMAC construction [2 p] ?**
         1. Without padding the the key it opens up the outer hash function to allow for extension attacks. An extension attack is where an adversary will pad the message with extra information. This allows the adversary to add more data into the message that will pass the hash test without knowing the secret key. By padding the initial key, K, with zeros to the length of what the hash function accepts as a block it forces any data being sent as the message to be hashed as it own block rather than concatenated along with the key and set the state of the registers, by doing this is prevents the length extension attacks from occurring.
   2. **In Partial Image-based Puzzles, what is the advantage to using sub-puzzles over one large puzzle, provide formulate and it’s rationale [8pt]**
      1. As the number of steps needed to solve a single puzzle of k bits difficulty is expected to take 2k/2 or 2k-1 steps to solve by having sub-puzzles this allows the main puzzle to be broken up into small steps, each step requiring the same expected number of steps to solve, 2k-1, which comes to m \* 2k-1 steps to solve the entire puzzle. This is done in order to allow the difficulty of the puzzle to be changed in case of an attack so that the attackers will be slowed down and hopefully not be successful in their DoS attack.
   3. **In slide 19, we describe a new puzzle mechanism that premits pre-computation of puzzles and puzzle sharing (i.e. the same puzzle can be used by different clients).**
      1. **Please elaborate at least three design rationales (discussed in class) that premits this feature [9p]**
         1. The usage of the Client ID and a server nonce is one of the rationales that allows for puzzle sharing. Having these two fixed values included in the solution to the puzzle means that even if one client were to find the solution to the puzzle it could not be used by a different client. This is because the second client, the adversary, would have a different client ID and server nonce, thus the values would not hash to the solution.
         2. A rationale that allows for pre-computation of puzzles is the fact that they are requiring the client to find an X value that when hashed with the other 3 values will have a set number of leading zeros in the hash. This means that rather than having to compute the hash of some pre-image based on the secret key, a time, and the request message from the client, then asking the client to solve for a specific set of bits such as in the previous client-server puzzles the server can just tell the client, that given these values k, Ts, and Ns, find an X value that will have k leading zeros. Then when a possible solution is given to the server the only requirement is to hash those values.
         3. Another design rationale that enables the ability to puzzle sharing is the server broadcasting the digitally signed k, Ts, and Ns. By doing this it allows the server to let any client get the required information for to solve the puzzle and not have to constantly make connections to different clients and provide them individually with the information.
5. **What is the Kerckhoff’s principal in modern cryptography? Write 2-3 paragraph of explanantion and why it is important [4p]**
   1. The main idea of kerchkhoff’s principal as it stands today is for a cryptographic system to work and be effective it must be completely secure even when the system it self and all of the information pretaining to it, with the exception of a secret key, are public knowledge. When applied to todays techonology this can be equated to the idea that algorithm used by SHA-256 to generate hash functions is public knownledge and can be found by anyone with access to the Google Search Engine. However, to actually generate the same hash using that algorithm as someone else with an unknown secret key it practically impossible with current computational power.  
        
      This is important because if you encrypt information or generate hashes with an algorithm that is no secured in this way it will become easy to break as soon as that algorithm is publically available. With today’s technology, especially with web-based systems becoming the new norm, it is increasingly difficult to hide from the client what your system is doing. This increases the possibility that at some point the algorithm you’re using for encryption or to generate hashes will at some point become public knowledge. When this happens, if you are not following Kerckhoff’s principal you will be opening your system up to adversaries.

**References**

**Miessler, Daniel. “The Birthday Attack.” *Daniel Miessler*, 18 Dec. 2019, danielmiessler.com/study/birthday\_attack/#:%7E:text=The%20birthday%20attack%20is%20a,253%20people%20in%20the%20room.**

**Nakamoto, Satoshi. “Bitcoin - Open Source P2P Money.” *Bitcoin*, bitcoin.org/bitcoin.pdf. Accessed 2 Feb. 2021.**

**Team, Finjan. “The Origins of Kerckhoff’s Principle and How It Is Used in Encryption Today.” *Finjan Blog*, 11 Sept. 2017, blog.finjan.com/the-origins-of-kerckhoffs-principle-and-how-it-is-used-in-encryption-today.**

**Whitehat Security. “Hash Length Extension Attacks.” *WhiteHat Security*, 14 Dec. 2020, www.whitehatsec.com/blog/hash-length-extension-attacks.**

**Wikipedia contributors. “HMAC.” *Wikipedia*, 2 Feb. 2021, en.wikipedia.org/wiki/HMAC#:%7E:text=HMAC%20uses%20two%20passes%20of,result%20and%20the%20outer%20key.**

**Lecture Slides**