# MAS433 Cryptography: Tutorial 5
# Hash Function and MAC
# 14.10.2010

**Instructions.**

1. Submission of tutorial solution is compulsory.
2. Submission deadline: 13 October 2011, 6PM
3. Please submit your solution by sending email to wuhj@ntu.edu.sg (If your solution is handwritten, you can scan and convert it into digital format, such as .pdf document.)
4. Tutorial solution will not be provided after the tutorial class.

**Question 1.** Birthday Attack

1.1 Randomly select four persons. What is the probability that two of them have the same birthday?

1.2 Randomly select 32 persons. What is the probability that two of them have the same birthday?

1.3 Randomly select four NTU students. What is the probability that two of them have the same Matriculation number?

**Question 2.** Hash Function

2.1 For a hash function based on a modified Merkle-Damgard construction in which only '0' bits are padded to the end of the message, how to find a collision with low complexity?

2.2 For SHA-1, the message length is formatted as a 64-bit word, and it is padded to the message. If the message length is 200 bits, how many compression function operations are needed to compress this message? How many compression function operations are needed if the message lengthens are 0 bit, 1 bit, 447 bits, 448 bits, 511 bits, 512 bits, 960 bits?

2.3 Find the message digest of the file "Tutorial5.tex" (the file is given in edventure), suppose that SHA-1 is used.
(Hint: If you are using Windows operating system, you may read the following link:
http://lists.gnupg.org/pipermail/gnupg-announce/2004q4/000184.html
If you are using Linux, you may use the command "sha1sum" directely to compute the message digest of a file.)

2.4 Change the above file name to "Tutorial6.tex", compute the message digest of the file "Tutorial6.tex".

2.5 Change the submission deadline in "Tutorial5.tex" to 23 October 2011, compute the message digest.

**Question 3.** (Bonus Question) Here is a method to protect the login passwords on a computer: when we create a computer user account in Windows, Linux or UNIX, the login password of a user is hashed, and the hashed value (instead of the password) is stored on the computer. When a user login to the system, the computer hashes the password, and compares the hashed value with the stored hash value. With this approach, it becomes difficult to recover the passwords if an attacker (or even administrator) gains access to the computer.

3.1 What is the property of hash function being used in this password protection scheme?

3.2 It is normally required that a password is hashed together with a random number (called 'salt' here), and store the hashed value together with the salt on the computer. Explain why. (Hint: consider that there are a number of users.)

3.3 Suppose that SHA-1 is used in this password protection scheme, what is the complexity to recover an equivalent password if the hashed value of an password is known to an attacker? (Here the equivalent password means that this password is hashed to the same value as the hash value being stored on a computer, but this password may or may not be the same as the original password.)

**Question 4.** (Bonus Question) Bit-Commitment
Alice and Bob are playing the coin tossing game over the phone: Alice tosses a coin, then Bob guesses whether it is head or tail. Without using cryptography, it is difficult to prevent Alice from cheating Bob (suppose that real-time video communication is not used). Now Bob requires Alice to use a strong hash function $H$ in the game: After tossing a coin, Alice generates a random number $r$, if the tossed coin shows head on the top, Alice computes

$H(1 \parallel r)$, and sends the message digest to Bob; if it is tail, Alice computes $H(0 \parallel r)$, and sends the message digest to Bob. After receiving the message digest, Bob guesses whether the coin is head or tail: if Alice agrees with the guessed value, then Bob wins; otherwise, if Alice claims that Bob guessed wrongly, Alice needs to provide the random number $r$ to Bob to show that Bob guessed wrongly (Bob can compute the message digest and compare it with the one received from Alice). Briefly explain why a random number $r$ is needed in this game. And what is the proper size of $r$?

**Question 5.** (Bonus Question) Hash Function Construction

5.1 In a hash function based on Merkle-Damgard construction, the compression function is given as $H_i = E_{m_i}(H_{i-1})$, where $E_k()$ denotes the encryption of an ideal block cipher with $n$-bit block size.

5.1.1 What is the complexity to find a preimage of this hash function? (Hint: meet-in-the-middle attack)

5.1.2 What is the complexity to find a second-preimage of this hash function?

5.1.3 What is the complexity to find a collision of this hash function?

5.2 In a hash function based on a modified Merkle-Damgard construction in which the IV is not fixed (i.e., an user is allowed to set the value of IV arbitrarily), the compression function is given as $H_i = E_{m_i}(H_{i-1})$, where $E_k()$ represents the encryption of an ideal block cipher with $n$-bit block size.

5.2.1 What is the complexity to find a preimage of this hash function?

5.2.2 What is the complexity to find a second-preimage of this hash function?

5.2.3 What is the complexity to find a collision of this hash function?

**Question 6.** Message Authentication Code

6.1 After the CBC encryption, denote the last ciphertext block as $C_n$. An MAC algorithm generates the authentication tag as: $t = E_{K_A}(C_n)$, where $K_A$ is the secret key for MAC algorithm, and it is different from the encryption key. The message authentication tag is sent together with the ciphertext. How to attack this MAC algorithm?

6.2 Denote the message as $M = m_1 \| m_2 \| m_3 \| \cdots \| m_n$. An MAC algorithm generates the authentication tag as: $t = E_K(m_1) \oplus E_K(m_2) \oplus \cdots \oplus E_K(m_n)$. How to attack this MAC algorithm?

**Question 7.** CMAC

7.1 What are the differences between CBC encryption and CBC-MAC?

7.2 Why is CBC-MAC insecure?

7.3 What are the differences between CMAC and CBC-MAC?

7.4 In CMAC, why is the value of $K1$ not simply set as $E_k(0)$?

**Question 8.** HMAC

8.1 Consider an MAC algorithm constructed from a hash function using the key-prefix method. How to attack this MAC algorithm?

8.2 HMAC-SHA-1 is the HMAC constructed from SHA-1. It is now widely used for secure internet communication. For HMAC-SHA-1, how many compression function operations are needed to generate the authentication tag of an 800-bit message? How many compression function operations are needed if the message lengths are 0 bit, 960 bits?

**Question 9.** Initialization Vector

9.1 How to choose the initialization vectors for CBC mode, CFB mode, OFB mode, CTR mode, synchronous stream cipher, asynchronous stream cipher, hash function, CBC-MAC and CMAC?

9.2 How would the security be affected if identical IVs are used with the same secret key in the cryptosystems in Question 9.1?