Internet Infrastructure Security (COMP444)
A3

Due at 11:55pm on 24 February 2015
Submission site: `https://submit.comp.polyu.edu.hk/`

Rocky K. C. Chang

February 12, 2015

1. `[6 marks]` (Oracle attack) Consider the oracle attack workshop on exploiting a web server's response to an attacker's input. Consider that the attacker tries to decode the intermediate value for the first block (i.e., $D(c_1)$). Note that the block size is 64 bits.

   (a) `[3 marks]` The attacker first tries to obtain the last 8 bits of $D(c_1)$ by entering different IVs. Finally, the server returns a "200 OK" message for an IV whose last 8 bits are `0x59`. What are the last 8 bits in $D(c_1)$?

   (b) `[3 marks]` The attacker then moves on to guess the last 16 bits in $D(c_1)$ by entering different IVs. What are the last 8 bits in the IVs that he tries?

2. `[6 marks]` (Coin flipping over the network) Consider that Alice and Bob make a decision based on flipping of a fair coin. They have devised a protocol that allows them to do so over a network, and they have to agree on a hash function $f(x)$ and that a head (tail) corresponds to an even (odd) x beforehand. The protocol works as follows. Alice selects a random number. Bob tries to guess whether the number is even or odd. If Bob's guess is correct, the coin outcome is given by his guess.

   The detail protocol works as follows:

   (a) Alice picks a random number x and computes $f(x)$, and sends $f(x)$ to Bob.

   (b) Bob tells Alice his guess of x as even or odd.

   (c) Alice sends x to Bob.

   (d) Bob verifies $f(x)$ and checks whether his guess is correct or not.

   Answer the following questions:

   (a) `[3 marks]` How does the collision resistant property of $h(x)$ prevent Alice from cheating?

   (b) `[3 marks]` How does the pre-image resistance property of $h(x)$ prevent Bob from cheating?

3. `[6 marks]` (CBC-MAC) As discussed in the slides on block ciphers, CBC can be used for generating an MAC (message authentication code) for a message. More specifically, the IV is set to 0, and the last ciphertext block is used for the MAC. However, it is known that this simple CBC-MAC is insecure for variable-length messages.

   As a simple example, consider two messages m and m' whose length is just one block each. An attacker first obtains the MACs for the two messages: $MAC(k, m) = E_k(m)$ and $MAC(k, m') =$

$E_k(m')$. Then he constructs a new message whose length is two blocks: $m||(m' \oplus E_k(m))$ (i.e., m is concatenated with $m' \oplus E_k(m)$). Show that the attacker can obtain the correct MAC for this new message. (Hint: use the CBC encryption to obtain $E_k(m' \oplus E_k(m))$.)

4. `[6 marks]` (Padding for iterated hashing) As discussed in the slides on hashing, a message to be hashed is usually preprocessed to become an integral number of blocks. The preprocessing usually adds a padding of additional data which must satisfy the one-to-one mapping from a message to a padded message. Otherwise, the two different messages will collide after the preprocessing step.

   (a) (3 marks) A simple padding mechanism is to pad the message with a minimal number of bit 0 at the end of the message. Explain why this padding mechanism fails to satisfy the one-to-one mapping requirement. (Hint: consider a message m and another message is m concatenated with a bit 0.)

   (b) (3 marks) A modified padding mechanism is to always pad with a bit 1 and then followed by a minimal number of bit 0. Does this padding mechanism meet the one-to-one mapping requirement?