IERG4130: Introduction to Cyber Security **Fall 2017**

Assignment 1

Deadline: HKT 15:30, Oct 23, 2017

Please answer all of the questions. Submit an A4 size hard copy to the designated homework box on or before the deadline. You are reminded that you are not allowed to copy from any sources without proper citations and acknowledgements. All submissions must include a declaration of academic honesty contained in the website http://www.cuhk.edu.hk/policy/academichonesty/.

Question 1: True or False. Explain your answer.

(1) By marking stack and heap as non-executable, we can eliminate buffer overflow attack completely.

(2) In UNIX system, the password file contains all users' encrypted passwords. If two users choose the same password, the corresponding two entries in the password file will be identical.

(3) Chrome/Firefox's private browsing mode will prevent web sites from discovering the user's name or IP address.

(4) XSRF attack is caused by confusing the system into interpreting content as code.

(5) It is impossible to recover the message encrypted by one-time pad via brute force attack.

(6) Using the same AES key to encrypt more than one messages is not secure as AES encryption is deterministic.

(7) One is likely to find out collisions of a k bits output hash function by trying out k^2 random inputs.

(8) A typical RSA cryptosystem implementation uses small decryption key to allow efficient decryption.

Question 2: Message Integrity in Counter (CTR) mode

CTR alone does not ensure message integrity. To add integrity, some non-cryptographer Carol proposes the following variant of CTR, called Encrypt-then-Hash CTR (hCTR) ($\mathsf{Enc}^{\mathrm{hCTR}}$, $\mathsf{Dec}^{\mathrm{hCTR}}$). It consists of a normal CTR encryption scheme ($\mathsf{Enc}^{\mathrm{hCTR}}, \mathsf{Dec}^{\mathrm{hCTR}}$) whose block size is ℓ, and a hash function $h(\cdot)$ whose output length is ℓ bits. Formally, $\mathsf{Enc}^{\mathrm{hCTR}}$ and $\mathsf{Dec}^{\mathrm{hCTR}}$ are defined as follows (assuming the message M is $\ell \times n$ bit long for some integer n):

Encryption $\mathsf{Enc}_k^{\mathrm{hCTR}}(M)$

1 : $C \leftarrow \mathsf{Enc}_k^{\mathrm{CTR}}(M)$
2 : $\sigma \leftarrow h(C)$
3 : **return** $C' = (C||\sigma)$

Decryption $\mathsf{Dec}_k^{\mathrm{hCTR}}(C')$

1 : Parse C' as $C||\sigma$
2 : **if** $\sigma \neq h(C)$ **then**
3 : **return** $\perp$
4 : **endif**
5 : $M \leftarrow \mathsf{Dec}_k^{\mathrm{CTR}}(C)$
6 : **return** M

Carol claims that hCTR is super secure in the sense that no one can come up with a ciphertext C' that does not decrypt to $\perp$ (denoting the input C' is an invalid ciphertext) without knowing the secret key k. As a clever IERG4130 student, you need to show that this claim is false.

To make your life easier, you are provided with an "oracle" machine that given M, returns $C' = \mathsf{Enc}_k^{\mathrm{hCTR}}(M)$. You can use this machine ONCE only.

(a) Show that for any $\ell \times n$ bit long message M^*, you can come up with a ciphertext C^* such that $\mathsf{Dec}_k^{\mathrm{hCTR}}(C^*) = M^*$ after obtaining $C' = \mathsf{Enc}_k^{\mathrm{hCTR}}(M)$ for an M of your choice.

To fix the attack in (a), Carol proposes Hash-then-Encrypt CTR (hCTR') defined as follows:

Encryption $\mathsf{Enc}_k^{\mathrm{hCTR'}}(M)$

1 : $\sigma \leftarrow h(M)$
2 : $C \leftarrow \mathsf{Enc}_k^{\mathrm{CTR}}(M||\sigma)$
3 : **return** C

Decryption $\mathsf{Dec}_k^{\mathrm{hCTR'}}(C)$

1 : Parse $\mathsf{Dec}_k^{\mathrm{CTR}}(C)$ as $M||\sigma$
2 : **if** $\sigma \neq h(M)$ **then**
3 : **return** $\perp$
4 : **endif**
5 : **return** M

(b) Show that hCTR' is as insecure as hCTR.

(c) Propose a fix to the attack you mentioned above and explain how does it work briefly.

Question 3: RSA Cryptosystem

(a) Given $N = pq = 31 \times 127 = 3937$, $\mathsf{ek} = 11$. Find the value of the Euler's totient evaluated at N, and hence the decryption key dk (in smallest positive integer) of the RSA encryption scheme.

(b) Decrypt $c = 413$ (in smallest positive integer) by repeated squaring. You can use Chinese Remainder Theorem (CRT) to speed up your calculation. Show your steps.

(c) Denote the decrypted plaintext as m. Show that one can produce the ciphertext of $m \cdot m'$ for arbitrary m' WITHOUT the knowledge of m and dk.

(d) Propose a fix to avoid the above attack.

(e) Denote the decrypted plaintext of ciphertext c' as m', what can you tell about m' if $c' \neq c$.