Spring 2016 Cryptography and Network Security

Homework 1

Release Date: 3/17/2016

Due Date: 3/31/2016, 14:20

Instruction

- **Submission Guide:** Please turn in your homework in class and submit your codes to CEIBA. For the codes, you need to put all of them in a folder named by your student id, compress it to hw1_{student_id}.zip. For example, hw1_r04922456.zip.
- You may encounter new concepts that haven't been taught in class, and thus you're encouraged to discuss with your classmates, search online, ask TAs, etc. However, you must write your own answer and code. Violation of this policy leads to serious consequence.
- You may need to code some programs in some programming problems. Since you can use any language you like, we will use pseudo extension **code.ext** (e.g., code.py, code.c) when referring to the file name.
- You are recommended to provide a brief usage of your code in **readme.txt** (e.g., how to compile, if needed, and execute). You may lose points if TAs can't run your code.
- In the programming part, your programs need to connect to the given service and get a flag which is a message in the format `CNS{...}` as your answer. (Services only allow connections from 140.112.0.0/16 and 140.118.0.0/16.)

Handwriting

1. CIA (9%)

Please explain three major security requirements: confidentiality, integrity and availability. For each security requirement, please give an example in the real world.

2. Hash Function (9%)

Please explain three properties of a cryptographic hash function: one-wayness, weak collision resistance and strong collision resistance.
For each property, please give an example applied in the real world.

3. MAC and Digital Signature (12%)

Suppose that students are asked to submit their homework to a trusted online submission system. Each student has one secure account, and he/she can use it to submit his/her homework. The homework consists of 100 multiple choice questions and students only need to submit a sequence of answers to the online submission system. However, all submissions

are public: that is, students can read each other's submissions at any time! Now, the following approaches attempt to prevent students from copying answers on the submission system:

(a) (4%) The submission system asks student i to submit SHA256(x_i) before the deadline, where x_i is student i's answer. After the deadline, the student submits x_i. The system can calculate the hash value and verify whether it is the same as the one submitted before the deadline. Can this approach prevent students from copying answers? Explain your answer.

(b) (4%) Each student i shares a secret key K_i with the system and submits $\text{MAC}_{K_i}(x_i)$ before the deadline. Student i submits answer x_i after the deadline, and the system can verify it by the shared secret key K_i. Can this approach prevent students from copying answers? Explain your answer.

(c) (4%) Each student generates a RSA key pair (K_i, K_i^{-1}) and securely publishes the public key such that every student including the system knows K_i. Then, student i generates a signature which consists of $(x_i, Sign_{K_i^{-1}}(SHA256(x_i)))$. Students submit $Sign_{K_i^{-1}}(SHA256(x))$ to the system before the deadline and submit answers x_i after the deadline. The system can verify the signature by the public key K_i. Can this approach prevent students from copying answers? Explain your answer.

4. Symmetric Cryptography with KDC (14%)

Suppose that each client is assigned a shared secret key with a trusted Key Distribution Center (KDC). Each client i can communicate securely with the KDC by the secret key K_{Si}, and secret keys are securely stored on the server. Now, Alice wants to send a secret message M to Bob via the following protocol:

$A \to KDC : ID_A||ID_B||E_{K_{SA}}(K)$, where K is a session key generated from Alice
$KDC \to A : E_{K_{SB}}(K)||ID_B$
$A \to B : E_{K_{SB}}(K)||E_K(M)$

Because Bob known K_{SB}, he can decrypt $E_{K_{SB}}(K)$ to get K and subsequently decrypt $E_K(M)$ to get M. Assume that the attacker has a secret key with the KDC and can eavesdrop packets of this protocol.

(a) (7%) How can the attacker get the message M? Please explain clearly.

(b) (7%) Please fix the protocol to prevent an attacker from getting the message M. Please explain clearly.

Programming

5. MD5 of MD5 (6%)

MD5 is a general hash function. You need to find x_1 and x_2 such that the leftmost 20 bits are all 1 in $MD5(x_2||MD5(x_1))$. You can access the service by `nc soc12.csie.ntu.edu.tw`

`20080` for more information. What is the flag? Please explain how you find x_1 and x_2 and save your code as `code5.ext`.

6. Hash Collision (11%)

In the class, we learned about the cryptographic hash function and it should satisfy weak collision resistance. The TAs create their own hash function in `prob6.py`. Given x_1, you need to find $x_2 \neq x_1$ s.t. $H(x_1) = H(x_2)$. You can access the service by `nc soc12.csie.ntu.edu.tw 20090` for more information. What is the flag? Explain how you collide the hash function and save your code as `code6.ext`.

7. What Does The Fox Say (11%)

You are given a session log `prob7.pcap`. (You can open it by Wireshark.) You need to observe the session information and login as **fox**. You can access the service by `nc soc12.csie.ntu.edu.tw 20100`. What is the flag? Explain the vulnerability and how you attack. Save your code as `code7.ext`.

8. Login As Admin (11%)

We discussed entity authentication (identification) in the class. The verifier can check the claimed identity of the prover, and the prover usually provides a password as a proof of the identity. Given the server `prob8.py`, you have one account **guest** and the password **passwdpasswdpass**. But you don't have the administrator password. Can you login as **admin**? You can access the service by `nc soc12.csie.ntu.edu.tw 20110`. What is the flag? Explain the vulnerability and how you attack. Save your code as `code8.ext`.

9. Shop (17%)

Here is a shopping service `prob9.py`. When you purchase items M, the shop gives you $\text{HMAC}_K(M)$ and you can checkout by providing (M, $\text{HMAC}_K(M)$) if the MAC verification is correct. Now, you need to purchase the flag. You can access the service by `nc soc12.csie.ntu.edu.tw 20120`. What is the flag? Explain what cause the vulnerability and how you attack. Save your code as `code9.ext`.
Note: $HMAC_K(M) = H((K \oplus opad)||H((K \oplus ipad)||M))$, further details please refer to the course slide and wiki.