Network Security

Project 2.2

Key Certificate and Binary Reversing

Instructor: Shiuhpyng Shieh

TA：E-Lin Ho, Jui-Chien Jao

1. Description

In this project, you need to finish two parts:

1) Make your RSA public key authenticated by CA (Certificate Authority) in order to communicate to the GameDownloader. To authenticate your RSA public key, you should follow the X.509 certificate format and use the certificate to prove who you are so as to download a game from the GameDownloader.
2) Find the logical error in the game you download from GameDownloader, and use the logical error to get your key in the game. You may need to use binary reversing tools to make this part much easier.

The purposes of this project are to let you know the importance of certificate, how a certificate is signed and verified, and binary reversing tools that can be used in next project.

The protocol is illustrated by Fig. 1.

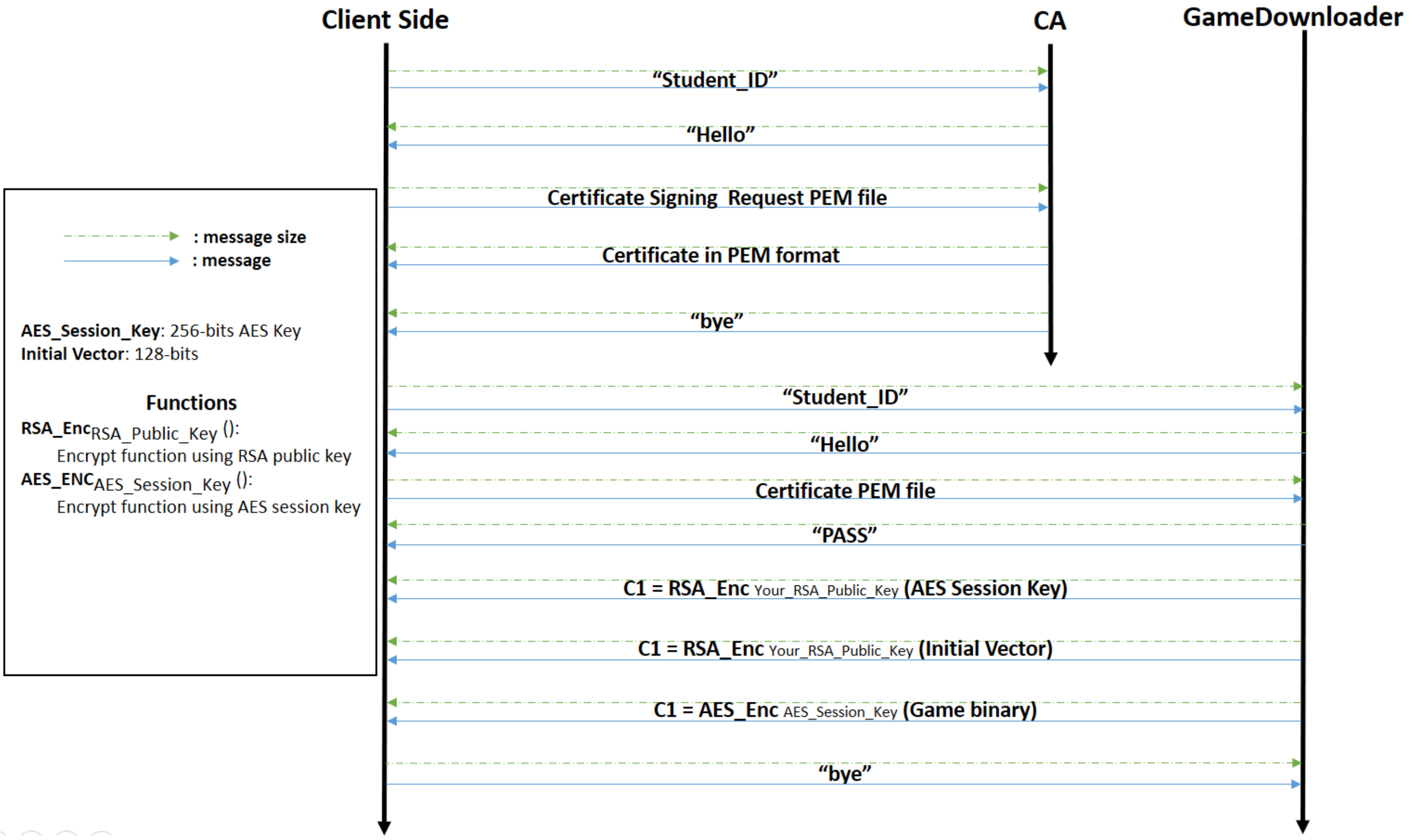
Client Side
CA
GameDownloader
"Student_ID"
"Hello"
Certificate Signing Request PEM file
Certificate in PEM format
"bye"
"Student_ID"
"Hello"
Certificate PEM file
"PASS"
C1 = RSA_Enc Your_RSA_Public_Key (AES Session Key)
C1 = RSA_Enc Your_RSA_Public_Key (Initial Vector)
C1 = AES_Enc AES_Session_Key (Game binary)
"bye"
: message size
: message
AES_Session_Key: 256-bits AES Key
Initial Vector: 128-bits
Functions
RSA_Enc RSA_Public_Key ():
Encrypt function using RSA public key
AES_ENC AES_Session_Key ():
Encrypt function using AES session key

Fig. 1 The protocol of project 2.2

Their IP addresses are below:
CA: “140.113.194.88:20000”
Server: “140.113.194.88:20500”
Response messages, “Hello” and “PASS”, may be other sentences if you do something wrong.

As shown in Fig. 1, there are two parts of the protocol:

Protocol between you and CA:

Sending your student ID to CA starts the communication between you and CA. At first, you construct and send the CSR (certificate signing request) to CA. Secondly, CA checks the contents of your CSR, and if there is no mistake about your information, CA will return your certificate signed by CA to you, otherwise you will receive nothing or error message. Finally, CA sends a “bye” in plaintext to you and the conversation ends.

You can only do this part once to get the certificate for entire project, and to simplify the protocol, the later CSR will replace the previous certificate with a new one.

Protocol between you and GameDownloader:

Sending your student ID to Gamedownloader starts the communication between you and GameDownloader. In the beginning, you should send your certificate signed by CA to GameDownloader in order to prove who you are. After verifying your certificate, if your certificate is correct, GameDownloader will use your RSA public key to encrypt AES key and Initial Vector for CBC mode and then send them to you, or you will get nothing. Next, GameDownloader sends a game encrypted by AES to you, and **you should continue receiving packets until the total size you received is equal to the size of the encrypted game since it’s a big-sized message.** In the end, to make sure you have received the encrypted game completely, you need to send the “bye” in plaintext to GameDownloader to close the connection.

Hints for the game

Since the boss is so strong that it is difficult to defeat it, we give you some hints:

1)

a) You can use binary reversing tool “objdump”, and argument “-d” of this command is helpful.

b) You can use “|” (pipe) as “X | Y” which means the **standard output** of X will be the **standard input** of Y. For example, “objdump -d game | less” is used to read assembly code in an easy way.

2) The key functions are “main”, “play_game” and “boss_move”.
3) Try to predict the boss’s moves, and boss will drop 10hp if you successfully attack it once.
4) You should know how it passes parameters and gets return values while doing a function call in Linux system.
   You can read the website:
   https://en.wikipedia.org/wiki/X86_calling_conventions#System_V_AMD64_ABI

2. Reminder

a) The common name in “subject” of your CSR must be your own student ID.
b) The game can only execute in Linux x86_64 based environment, and you can use CS Linux WorkStation if you have an account of it or build a virtual machine set with Linux x86_64 based OS.
c) Every message should be sent after the total number of bytes of it.
d) There will be a timeout setting for every connection.
e) The details of RSA and AES cryptosystem are the same as previous projects.

3. Deliverables

Each student must work individually and submit **a compress file named by your student ID, e.g., “<YOUR_STUDENT_ID.zip>”** containing:

a) All source files of your program named by your student ID.
b) A report, text, Word or PDF, named by your student ID includes descriptions of:
   - If you are the GameDownloader, how will you verify certificates?
   - Your key in the game.
   - The logic error in the game.
   - The way that you defeat the boss.
   - The commands or the steps for compiling and executing your program.

Copy or piracy is strictly prohibited.

If you have any question, please contact TA as soon as possible.

Any anomaly connection such DDoS will be traced for penalty.

Deadline: 2016/12/4 (Sun.) 23:59:59