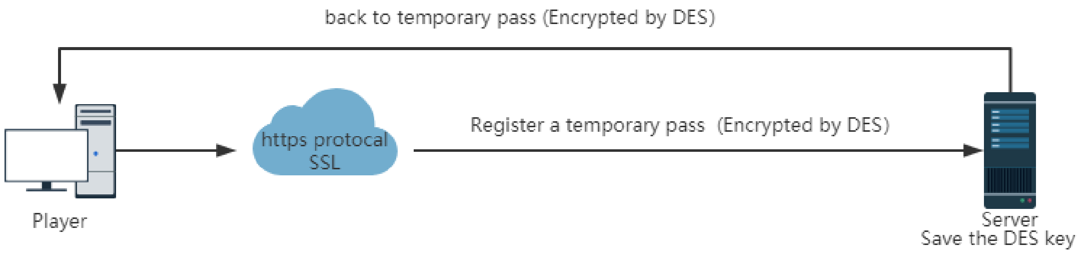
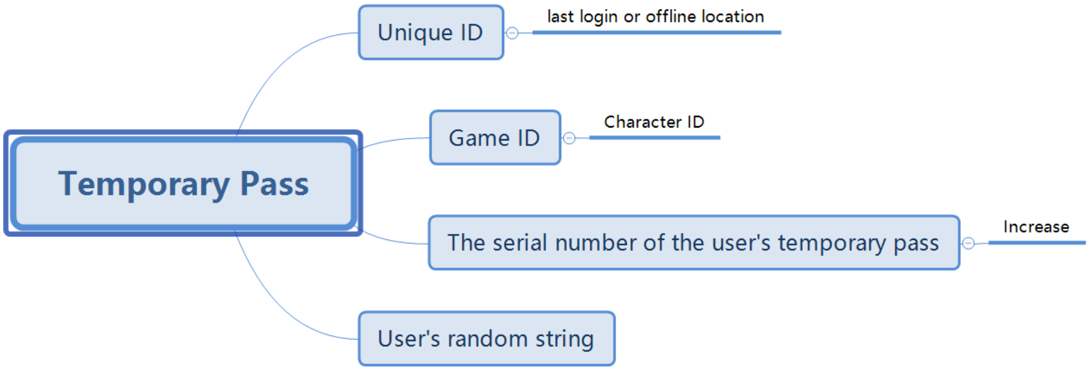
User security

1. User authentication

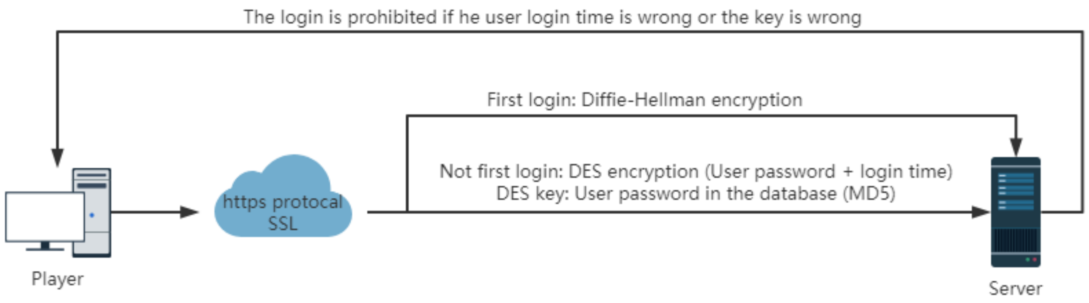
   Solution: Register a temporary pass on the server when the user connects to the server. The server returns the user's temporary pass, and the user enters the game through the temporary pass. The temporary pass needs to contain the user's unique ID (last login or offline location), game ID, the serial number of the user's temporary pass, and the user's random character string, etc., and then use des encryption. The encrypted key of the temporary pass is stored in the server and does not need to be returned to the user. When the user logs out, a new ordered temporary pass serial number and unique ID are generated in the database. The user can only see a string of meaningless strings all the time, and the temporary pass is one-time and cannot be forged.


back to temporary pass (Encrypted by DES)
https protocal
SSL
Register a temporary pass (Encrypted by DES)
Player
Server
Save the DES key


Temporary Pass
Unique ID
last login or offline location
Game ID
Character ID
The serial number of the user's temporary pass
Increase
User's random string

2. Password secure submission

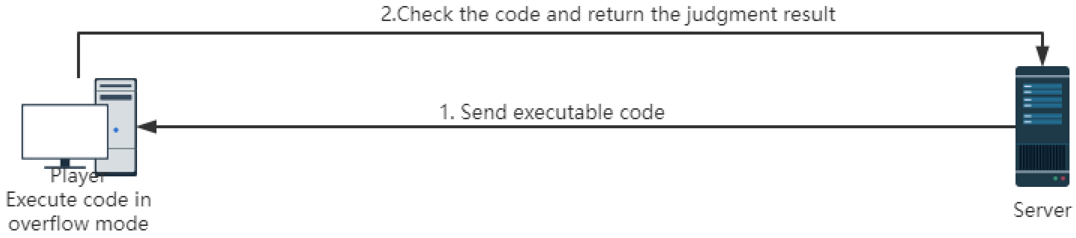
   Solution: When the user connects to the server, if the user's password is not found in the database, the user's first password is encrypted and transmitted to the server by Diffie Hellman and stored in the database, in order to compare the user's login password with the password at the time of registration. The next time the user logs in, the user's password and the user's login time are spliced together and then encrypted with des. The login time will be mentioned in the session key in the Trojan horse security protection later. Use the hash value of the user's password in the database to decrypt the encrypted content to the server to obtain the user's password and login time. If the user login time is incorrect or the key is wrong, the user is denied login.


The login is prohibited if he user login time is wrong or the key is wrong
First login: Diffie-Hellman encryption
Not first login: DES encryption (User password + login time)
DES key: User password in the database (MD5)
https protocal
SSL
Player
Server

Question 2: The security of the game client

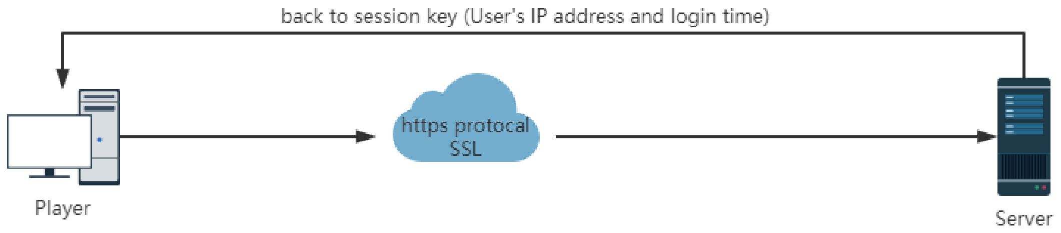
1. Dynamic anti-plugin

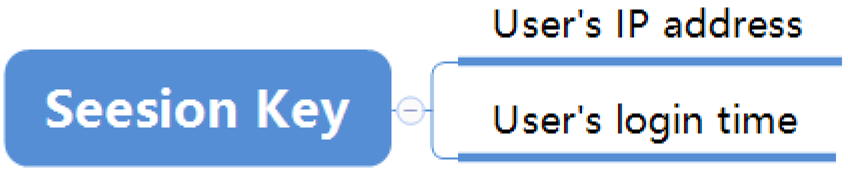
Solution: The server sends a piece of executable code, the client receives this piece of code and executes it, and the result is sent to the server. The server determines whether the returned result is correct. The sent executable code is mainly divided into detection code and return result. The detection code is any code edited by the maintainer. The purpose is to detect any information in the player's computer, such as whether the client's main program has been modified, verify the integrity of the client, send the program currently used by the player to the server, and check whether it is a plug-in. The returned result is for each test A tag is then returned to the server, so that the plug-in cannot return the correct tag, because the tag can be in any form and is very random. We can use the internal code of the function in the game to execute the executable code sent by the server in an overflow mode without judging the memory out-of-bounds access.


2.Check the code and return the judgment result
1. Send executable code
Player
Execute code in overflow mode
Server

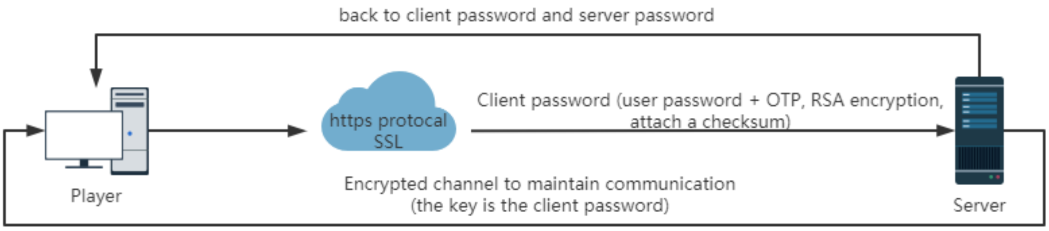
2. Trojan horse security protection

Solution: The user starts a built-in browser before logging in to the game, connects to the login server through the https protocol, and starts the game when the server returns the session key. The session key needs to include the user's IP address and login time. The temporary pass in the user authentication identity mentioned earlier can be used as another sign returned by the server to ensure that the content that the user sees is the official server, and to prevent the Trojan from maliciously modifying the game client file and causing the content to change. In the process of starting the game, in order to prevent the session key from being intercepted without the user’s knowledge, the built-in browser should be closed after the user’s game client has completely entered the game. If the game client on the user's computer gets the session key but fails to enter the game smoothly, a warning message will be issued to the user.





Solution: The game client generates a random number. You can use the secret (user password + OTP) known to the game client and the server. Use the RSA algorithm to encrypt the random number and send it to the server. You also need to attach a checksum to ensure that the server can Recognize this encrypted random number. After the server decrypts the random number, when the server verification is completed, the server generates a server random number, returns both random numbers to the user's game client, and then uses the game client's random number as a key to establish an encrypted channel Keep communication with the game client. In this way, the middleman cannot forge two identical random numbers.


back to client password and server password
Player
https protocal
SSL
Client password (user password + OTP, RSA encryption,
attach a checksum)
Encrypted channel to maintain communication
(the key is the client password)
Server

Question 3: The security of the game server

1. Prevent MITM Attack

Solution: The game client generates a random number. You can use the secret (user password + OTP) known to the game client and the server. Use the RSA algorithm to encrypt the random number and send it to the server. You also need to attach a checksum to ensure that the server can Recognize this encrypted random number. After the server decrypts the random number, when the server verification is completed, the server generates a server random number, returns both random numbers to the user's game client, and then uses the game client's random number as a key to establish an encrypted channel Keep communication with the game client. In this way, the middleman cannot forge two identical random numbers.