JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 39, 473-487 (2023)
DOI: 10.6688/JISE.202305_39(3).0002

A Secure Cloud Gaming System

CHUN-I FAN, HSIN-NAN KUO, YUNG-SHENG TU, YUAN-CHI CHEI,
CHU-CHIA CHUANG, YU-CHUN TSENG AND ARIJIT KARATI+
Department of Computer Science and Engineering
National Sun Yat-sen University
Kaohsiung, 804 Taiwan
E-mail: cifan@mail.cse.nsysu.edu.tw; bluedunk@gmail.com;
{sam88213; taurus12280503; chiang310019}@gmail.com;
m083140005@g-mail.nsysu.edu.tw; arijit.karati@mail.cse.nsysu.edu.tw

Cloud gaming is a relatively new trend in gameplay nowadays. It sends the user's actions to the gaming server through the internet, having the data and computation of game software processed in the cloud server, and then sending the gaming screen to the user's computer. The advantage of such a method is that it prevents users from wasting money on unneeded hardware upgrades. The same game can run on many platforms simultaneously, decreasing hardware requirements. The 5G mobile network is becoming increasingly popular; it offers a better environment than the 4G mobile network, 10 to 100 times quicker internet speed, and reduced network energy usage, making cloud gaming a more delightful experience. However, safe data access in the cloud remains a primary priority in such a system. This paper proposes a robust system allowing users to access the game securely and authentically through cloud space. In addition, to defend against specific network assaults, we employ an intrusion prevention system. Our system is implemented in an ideal simulation scenario to demonstrate the functional advantages that improve the next-generation gaming experience.

Keywords: GamingAnywhere server, intrusion prevention system, denial-of-service attack, database, hash chain

1. INTRODUCTION

Cloud gaming, known as gaming on-demand or gaming-as-a-service, is a type of online gaming in which games are run on remote servers and expressly transmitted to the user's device, or more commonly, playing a game remotely from the cloud via Best-Effort Internet. Cloud gaming has become tremendously widespread due to its advantages to game players, developers, and service providers. It allows various entities to do the specific tasks mentioned in Table 1. However, it makes supporting real-time computer games more challenging. Two events have accelerated the growth of the cloud gaming market: the first is the acquisition of Gaikai, a powerful gaming console developing business, by Sony in 2012; and the second is the competition between Sony's PlayStation Now (PS Now) and Nvidia's GeForce Now. Although cloud gaming has enormous potential, game

Received January 5, 2022; revised February 14 & March 20, 2022; accepted April 3, 2022.
Commuicated by Po-Wen Chi.
+ Corresponding author.

Table 1. Activities of each entity in cloud gaming platform.

The plyaer	Game developer	Service provider
1. Visit their games at any time. 2. Purchase or rent games according to their needs. 3. Avoid them from updating their hardware periodically	1. Concentrate on a sole platform, which reduces the costs for porting and testing. 2. Bypass the retailers to obtain higher profits. 3. Contact more game players. 4. Avoid pirate, because the game software will never be downloaded to the client's computer.	1. Bring new business model. 2. Ask for more requests to the deployed cloud resources. 3. Demonstrate alternative potentials in remotely exercising application programs since cloud gaming imposes the most stringent requirements to calculation and network resource constraints.

developers and service providers must overcome various hurdles to achieve its potential to attract more game players. We list the most crucial points. First and foremost, the cloud gaming platform and test bench must be built for extensive performance evaluation. Measuring the quality of service (QoS) index [1] (*e.g.*, energy consumption and network index) and the quality of experience (QoE) index [2] are among the evaluations (*e.g.*, the perceptual experience of players). Developing a cloud gaming platform and test bench will be time-consuming, and understanding the complex relationships between QoS and QoE will be more difficult. Second, the cloud gaming platform and assessment process require researchers to optimize numerous factors such as the cloud server and communication channel. In practice, optimizing technologies were used to the following aspects:

- Better resource utilization and distributed architecture on the cloud server.
- Optimal content encoding and adaptive transmission in the communication channel.

Furthermore, computer games are classified into several types. These items can be divided into two categories: *viewpoint* and *theme*. Viewpoint refers to how a player perceives the game scene. It controls the variation of produced video on the screen. First-person, second-person, third-person, and omnipresent are the most prevalent viewpoints as illustrated in Table 2. How players engage with game content is determined by the game *theme*. Shooting, fighting, sports, role-playing games (RPGs), action role-playing games (ARPGs), turn-based strategy, real-time strategy (RTS), and management simulation are all common themes. Although the game's premise may limit the viewpoint, a game's genre may generally be defined by its viewpoint and theme, such as first-person shooting, third-person ARPG, omnipresent RTS, and so on. Fast-paced first-person shooting games have the most complex scenes, making them the most demanding titles for cloud gaming service providers. On the other hand, third-person turn-based RPG games are less sensitive to latency and hence more suited to cloud gaming.

There is open-source software in cloud gaming systems – GamingAnywhere, a modular solution that works across numerous platforms, such as Windows, Linux, Android. Its primary goal is to connect users and cloud gaming servers to the same local area network for data transmission and processing.

Further, it offers extendibility, portability, and reconfigurability, allowing academics to test concepts and construct cloud services while allowing players to design their private cloud gaming systems [3]. Because of its distributed real-time interactive design,

Table 2. Most prevalent viewpoints.

Type	Description	Example
First-person	Graphical views of the characters in game.	Counter-Strike
Second-person	In-game characters are represented from the back, allowing the user to see the characters on the screen.	Grand Theft Auto
Thurd-person	Fix the players' vision when viewing 3D scenes in 2D surroundings. The sky perspective, often known as God's vision, is commonly used.	Diablo, Command & Conquer, FreeStyle, *etc*.
Omnipresent	players can examine the region of interest (RoI) from various angles and distances	Age of Empires 3, Stronghold 2, Warcraft III

clients may play games in the cloud without upgrading their hardware to match the game's minimum requirements. Its users could play cloud-based games without upgrading their hardware to match the game's minimum requirements. Overview, platform, optimization, and commercialization are the four components of cloud gaming. The overview can be divided into general cloud gaming and specialized cloud gaming, which includes mobile devices. In addition, the platform would deliver QoS and extended QoE evaluations, allowing for client satisfaction. There are two ways to optimize, and both can be done from the server's architecture or by enhancing data compression for data transfer [4]. The GamingAnywhere system's efficiency can be tested by running it in a virtual environment, such as VMware or VirtualBox, and deploying the virtual kit in the system. The server's energy consumption impacts the overall efficiency of the system. According to [5], a better scheduling algorithm would increase the virtual machine's performance since the Energy-Aware scheduling method can change the virtual machine's energy usage. However, the data sent between the game user and the server via the Internet may be modified with or even dropped on purpose by the attackers. Thus, the security and integrity of such communication are critical in cloud gaming.

2. LITERATURE REVIEW

Due to the rapid expansion of E-commerce and the internet, *secure database access* is one of the most crucial elements [6]. This can be achieved using NIDS Snort to analyze the network attack process [7] and categorize event records by labeling the original IP address in three statuses as *suspected*, *threatened*, and *malicious*, and saving them in the IP address status database. The network administrator will be given information about records classified into distinct record kinds as well as the status of IP addresses [8], which solves the problem of massive records. The data shall be encrypted before transmitting to the cloud service provider (CSP). When CSP processes data, users must supply a private key to the server for decrypting data, which may jeopardize the secrecy of data kept in the cloud server [9, 10]. Homomorphic encryption, executes specific functions on encrypted data without decrypting it, can be used in such a system. *User authentication*, in addition to encryption, is a critical attribute of cloud-based gaming [11]. Some works concerning to the user authentication are surveyed [12–14].

Distributed Denial-of-Service (DDoS) is a significant issue in a cloud-based gaming environment, with far more severe consequences due to cloud structures than traditional

network security. When a vulnerability in a cloud server is uncovered, it is not the same as a security issue that affects a single machine. The user's connection will trigger the identical problem in every client computer, resulting in the vulnerability being widely reproduced throughout the computing structure and posing a danger to traditional network security [15]. In addition, a new DDoS attack discovery approach is primarily directed at the virtual machine (VM)-based application and the CSP intranet: the vulnerable area's multi-tenancy cloud structure. When confronted with a DDoS attack, traditional detection and defensive procedures fail to preserve service level agreements (SLAs). This is due to their slow response time after an attack is detected, the high cost of CSP, and their ability to resolve intellectual attack stratagems [16]. Due to inefficiency, large storage capacity, and other concerns, traditional defense tactics cannot simply be applied to cloud security. A filtering strategy based on reliability known as the Confidence-Based Filtering (CBF) method is being investigated for the cloud computing environment to mitigate such an issue [17]. The authors introduced two cycles, namely attack and non-attack. Legal data packets are gathered throughout the non-attack cycle, and their attributes are retrieved to generate confidence set files according to their values. The supplied data packet is considered abandoned or not based on the standards of the confidence set files by computing its score. After testing, CBF scored well in speed, storage space requirements, and filtering accuracy, indicating that it is suitable for real-time filtering in a cloud context.

SQL injection has a significant influence on the cloud gaming environment. Several efforts in preventing SQL assaults are offered for cloud systems, including [18–20]. The authors [21] proposed a method combining Dynamic Taint Analysis and filtered SQL to address the Web service problem of SQL Injection in the cloud computing environment. In addition, entrenching Web application programs in the cloud environment ensures their security. This method examined the vocabulary rules of SQL statements. After that, decision trees are built by examining the syntax of SQL statements. Furthermore, it developed a 3-ary tree on the model based on the syntactic principles of SQL statements for attack detection. According to the empirical results, this strategy is realistic and feasible. On the other hand, a detecting module can be added to improve accuracy.

3. MOTIVATION AND PURPOSE OF THE RESEARCH

Playing cloud games in a suitable environment is inextricably linked to the internet. A gaming PC system must meet specific requirements to play triple-A video games. Cloud gaming also allows gamers to play high-quality games without requiring high-end hardware. Because cloud gaming necessitates a large amount of bandwidth, employing a 5G mobile network for internet access may be conceivable in the future. Low-latency connections for large devices, giving players a better experience than a 4G mobile network configuration, could improve cloud gaming efficiency even further. Using a pre-built gaming system as the foundation for cloud gaming, on the other hand, emphasizes the need for network structure security. Due to the cloud system's reliance on networks, an attacker might steal, alter, or install malware on users' workstations, execute a denial-of-service assault over a network, or even exploit the cloud system's fundamental weaknesses. Instead of addressing a secure and robust environment, most of the discussed works in the literature focus on the effects of rapid communication and gameplay experience. In addi-

tion, incorporating current cryptographic protocols causes a slew of problems, including inefficiency in integration, security vulnerability exposure, and QoE reduction. We presented a dependable secure protocol for the existing open-source cloud gaming platform. Under this framework we made the following contributions:

- The proposed protocol uses hash chaining with salt approach to improve user authentication. Specifically, we protect users' information using the hashed password [22], design security approaches for dealing with network threats and implement the algorithm to create a safe and user-friendly cloud gaming system. Besides, an entropy based approch is considered to resist DDoS attack.
- We improve key game elements, including network connection stability, game operation fluency, and accuracy. Furthermore, we safeguard personal information, such as account and credit card information, critical when paying for game content.

4. PRELIMINARIES

4.1 Cloud Gaming Model Overview

The functionality of cloud gaming is depicted in Fig. 1. Typically, the cloud gaming provider owns more data centers and/or servers from which gamers can access the game. Moreover, the cloud gaming platform executes computer games, which can be separated into two primary parts [23]:

- executing the player's instruction to modify the gaming logic, and
- the render engine, which generates the real-time game scene

The cloud gaming platform includes the instruction interpreter, image capturing device, and video encoder. The Player's commands are provided by the instruction interpreter, while the game scene is taken as an image by the image capture device and compressed

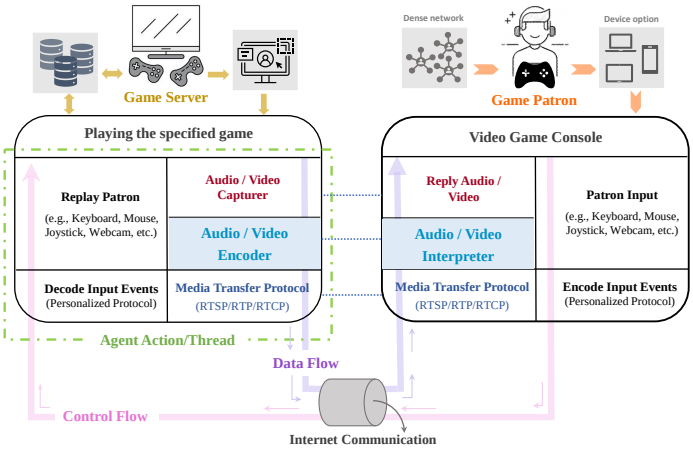


Fig. 1. The operation of cloud gaming.

by the video encoder. The cloud gaming platform both sends and receives picture data packets from customers. It is a client device with only two simple components:

1. The receiver links to the gaming controller (*e.g.*, joystick, keyboard, or mouse).
2. Video decoder accomplished by mass-produced decoder IC chips.

4.2 Hash Chaining Operation

A hash chain is formed by repeatedly applying a cryptographic hash function $H(\cdot)$ on a string $x \in \{0,1\}$. The repeatation continues until it matches with the length of chain l. Therefore for given l, the message digest is computed as

$$\delta = H^l(x) = \underbrace{H(H(\cdots H(x)))}_{l}$$

The hash chain has the following two interesting properties:

- Given $H^{l-1}(x)$, computation of $H^l(x)$ is feasible. Thus, we have $H^{l-1}(x) \rightarrow H^l(x)$.
- Given $H^l(x)$, computation of $H^{l-1}(x)$ is infeasible, *i.e.*, $H^{l-1}(x) \nleftarrow H^l(x)$.

5. OUR RESEARCH METHOD

This section discusses our security mechanisms and how they are used. We go over the setup, then the various connections between the server, client, and database, as well as the intrusion protection system.

5.1 GamingAnywhere Setup

As shown in Fig. 2, the GamingAnywhere cloud system is installed on a virtual machine on the server. Now we will go over the many steps that have been mentioned. Install 64-bit Ubuntu in VMware and decompress the GamingAnywhere source code. After that, create GamingAnywhere. Install the dependency first, update and compile the configuration file; finally, mount the dynamic dependent library file stored in system variables. Prepare a cloud game client as well as a suitable Linux game next. Then, open the GamingAnywhere server, change the configuration file, and run the cloud game. We use the terminal to run particular commands, change the configuration file, and launch the GamingAnywhere server. Last but not least, enable the client connection.

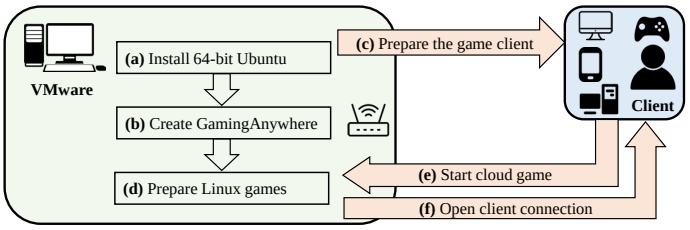


Fig. 2. Process of installing GamingAnywhere.

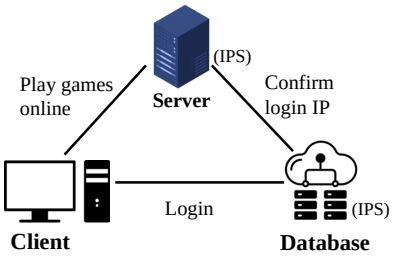


Fig. 3. The connection among client, server and database.

5.2 The Connection Among Server, Client and Database

The relationship between client, server, and database is represented in Fig. 3.

1. **Client:** Use the Qt C++ API for GUI programming, which offers users a graphical user interface. The database must be connected before the registration and login; once the user has been identified as a legitimate user, he or she can connect to the server and play the game.

2. **Server:** The server is written in C++ and is in charge of obtaining the legal IP address from the Database. When a legal Client connects to the server, the cloud game begins. The Intrusion Prevention System will be used to prevent an attack.

3. **Database:** Set up a database for storing user information using the MySQL C API. This is used for identification to prevent unregistered users from connecting to the server directly, which could cause the server to run out of resources, crash, or become unavailable. To avoid an assault on the database, the Intrusion Prevention System will be employed.

4. **Between Database and Client:** The client sends hashed data to the database via a hash chain [24], the frequency decreases each time, which is constructed by combining the data with salt generated by the database. After receiving the data from the client, the database generates a random salt and hashes the data before storing it. Fig. 4 shows the flow chart for storing Client data in the database. Here, the user chooses a password (*e.g.*, pwd) and decides a hash chain index (*e.g.*, i). Now, the user with the help of Salt generates a strong password$_i$ = Hash^i (pwd, Salt), and sends password$_i$ to the database. Certainly, to enhance security, the database stores hashed password as HPW = Hash(password$_i$, Salt). Thus, on next run, user computes password$_{(i-1)}$ = Hash$^{(i-1)}$ (pwd, Salt) and checks whether HPW $\stackrel{?}{=}$ Hash(Hash(password$_{(i-1)}$), Salt). If it holds, the database updates as HPW = Hash (password$_{(i-1)}$, Salt). Once the hash chain is about to finish, the user reinitiates index by i' during ith session.

5. **Between Database and Server:** The TLS protocol first performs the necessary negotiation and certification to establish the encrypted channel. The Database sends the lawful user's IP address to the Server through a TLS protocol encrypted channel to preserve communication privacy. After that, the server saves the legal user's IP

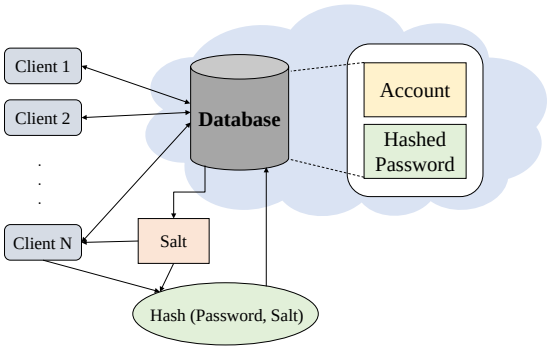
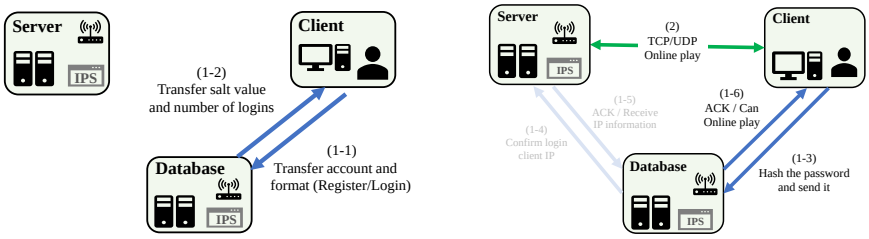


Fig. 4. Process of storing client data in the database.



(a) Step 1 (b) Step 2

Fig. 5. Registration/Login phase.

address in Set, then waits for the user's connection and compares it to the information in Set to see if the user is legitimate.

6. **Between Client and Server:** The client sends a request to the server first. After that, the server checks to see if the client's IP address is recorded in Set, and if it is, the cloud game will be opened for play, or the connection will be closed if not.

5.3 Intrusion Protection System

To avoid the attack, use an intrusion detection system, then create additional security rules to guard against other assaults based on the attack records. Create software that compares the characteristics of attack records; when an attack is discovered, it is logged in the IPS rules, and the IPS deletes data packets from unusual IPs.

6. RESEARCH RESULTS

This experiment allows players to play various games without the need for cutting-edge equipment. An attacker cannot steal user information, attack the database, or attack the server, making the cloud gaming system more secure. The analysis for each part of the experiment is as follows:

1. **Client:** The communication for client registration and login operations is depicted

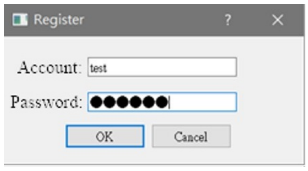
in Fig. 5. As exhibited in Fig. 5a, the Database is informed of the client's request for registration/login by broadcasting the account and format, then waits for the Salt and number of login times from the Database before transmitting the Hash Chain processed client's password back to the Database. The client can connect to the server after the Database sends authorization, as shown in Fig. 5b.

2. **Database:** When two different activities are performed, this entity is used.

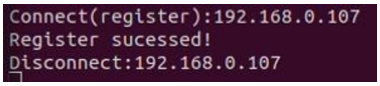
   (a) **Registration step:** It waits for the Client to establish a connection, then generates a random Salt of no more than 15 bytes in length and sends it to the Client for password hashing. The result is presented on the screen if the registration is successful, as shown in Figs. 6a-6c. User's number, account, hashed password, login time, logout time, Salt, and remaining login times are the Client's information formats stored in MySQL by Database in the sequence.

   (b) **Login Step:**The IP address is transmitted to the server through an encrypted TLS connection when the Client is confirmed correct. The Client login page is presented in Fig. 6d. It contains the content of the certificate used to establish an encrypted TLS channel with the server. After that, the Client receives the result to connect the server properly. Fig. 7a shows when the maximum number of Client logins is reached, the Database generates a random Salt and sends it to the Client, who then repeats the password hash process and returns. The data transmission mechanism during the active connection between Client and Database is depicted in Fig. 7b.

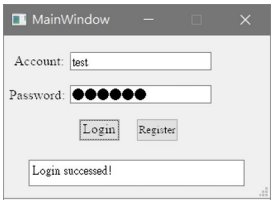
3. **Server with Database:** TLS protocol is used for the negotiation and certificate required for establishing an encrypted channel. After connecting to the database, the server sends the necessary certificate content and waits for data. When the client's IP address is received, store it in the IP database and mark it as a legal client to determine whether the client requesting connection is legal.



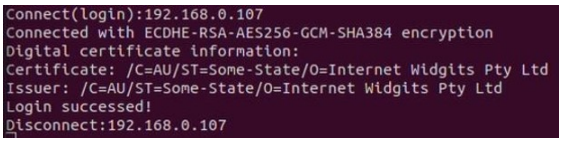
(a) Client UI registration interface.



(c) Database registration.

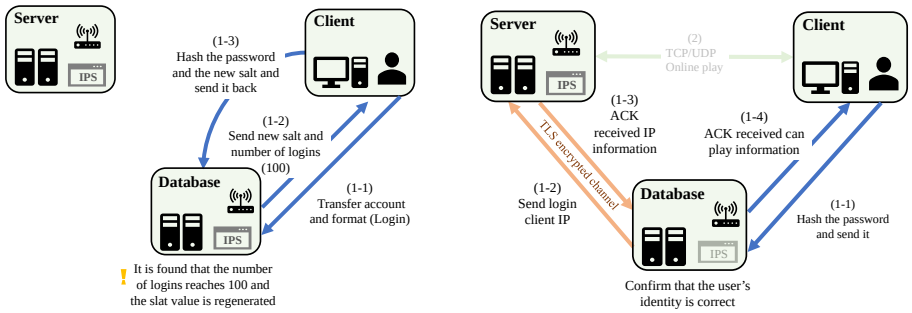


(b) Client UI login interface.



(d) Login screen of database.

Fig. 6. Execution result.



(a) Database re-generates and transmits *Salt*. (b) Data transmission between Database and Client.

Fig. 7. Strategies to handle client login.

4. **IPS:** As shown in Fig. 8, it functions by computing 500 network traffic data, then decides whether the variance in network traffic of each IP over a while is expected. Integrating the Fast Entropy Approach and the Adaptive Threshold Algorithm, it identifies DoS attacks and adds the attacker IP to the IPS Rule for discarding future data packets. The following is a description of how program variables are used:

```
for auto x : mapper do
    double temp;
    if premapper.find(x.first) ≠ premapper.end() then
        tmp = [premapper[x.first]]
        if x.second > temp then
            H = log(tmp/x.second) − log(x.second/sum)
        else H = log(x.second/temp) − log(x.second/sum)
        end if
        if H > 1.5 ∗ mean then
            B++
        end if
        if H < (0.5 ∗ mean) then
            B−−
        end if
        if fabs(mean − H) > (B × stdDeviation) then
            cout << x.first << endl;
        end if
    end if
end for
```

Note: x.first: IP Name to be computed. **x.second:** It is the number of the IP to be computed appeared in 500 datum. **temp:** The stored value is the number of appeared x.first in the last 500 datum. **H:** Entropy. **sum:** The number of appeared x.first in this computing. **mean:** The average of the times that IP appeared during this computing (sum/total number of appeared IP). **B:** threshold. **stdDeviation:** Standard Deviation.

Fig. 8. Entropy approach.

7. FURTHER DISCUSSION

Users can quickly complete the registration and login to begin playing the game in a safe environment, but they will still confront the following issues:

- **Latency during play:** Because of the limited bandwidth and delay seen on the 4G mobile network, the network latency is asynchronous between client and server, as shown in Fig. 9. The network latency produces image lag during transmission. 1The 5G mobile network has gained traction, and the issue is expected to be remedied after the 5G network replaces the 4G network.

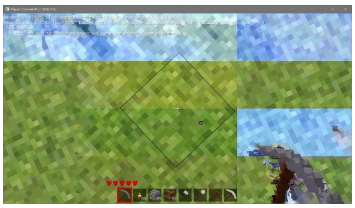


Fig. 9. UDP packet transmission latency.

- **Intranet server transmission congestion for massive data packet:** After a DoS attack on the server, many malicious data packets are left in the network, causing the connection between server and client to be impacted, causing the FPS of the user's gaming screen to be reduced and display delayed. It is believed that flow cleaning from upstream (done by an outsourcing firm or by programming machine learning oneself) and the defense mechanism downstream will lessen such concerns.
- **Problem of account safety:** The security of a user's information is protected only in the part of password hash transmission, which may not be enough to withstand a malicious attack by someone with bad intentions. Therefore, the method of safety protection presented in Fig. 10 is expected to be adopted, *i.e.*, iris recognition or fingerprint recognition, and so on, to achieve more rigorous protection. Despite its complexity, the procedure has a significant impact on safety.
- **Unavoidable DDoS attack:** It is not capable of completely preventing DDoS attacks, as shown in Fig. 11, but it is expected to be avoided by setting up more virtual servers to shift the target or detecting and handling DDoS attacks as efficiently as possible by machine learning to ensure the quality when users play the game.

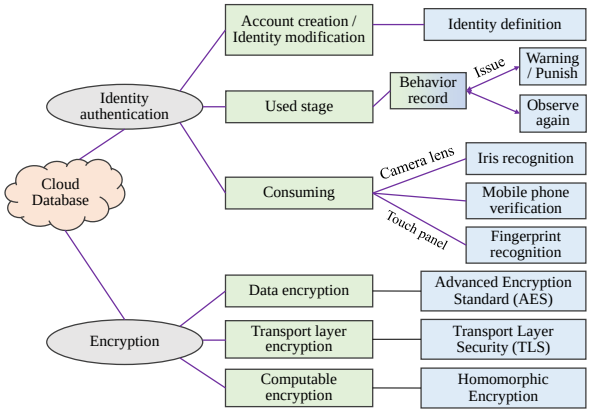


Fig. 10. Safety protection with enhanced identification.

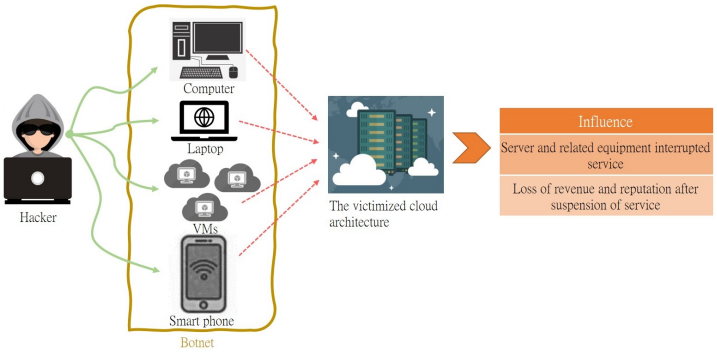


Fig. 11. The DDoS attack and influence on cloud system.

8. CONCLUSION

In this paper, we integrate an intrusion prevention approach that was previously unavailable in the open-source GamingAnywhere cloud gaming. The impact of an attack defense has been realized by including rules for detecting DoS attacks and banning malicious IPs into the intrusion prevention system (IPS) rule. Installing an IPS on the server defends the systems from malicious users who may have signed in. Secure data transmission to the database is further enhanced by employing an encrypted TLS channel between servers, or a password hash between client and server. Furthermore, the server and database are set up separately, allowing the server to only process data packets from the database's assigned IP address. The suggested methodology improves security while incurring adequate overheads, thereby enhancing the overall game experience.

ACKNOWLEDGMENT

This work was partially supported by the Ministry of Science and Technology of Taiwan under grants MOST 109-2221-E-110-044-MY2, MOST 110-2218-E-110-007-MBK, and MOST 110-2222-E-110-006-. It also was financially supported by the Information Security Research Center at National Sun Yat-sen University in Taiwan and the Intelligent Electronic Commerce Research Center from The Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE) in Taiwan.

REFERENCES

1. K.-T. Chen, Y.-C. Chang, H.-J. Hsu, D.-Y. Chen, C.-Y. Huang, and C.-H. Hsu, "On the quality of service of cloud gaming systems," *IEEE Transactions on Multimedia*, Vol. 16, 2013, pp. 480-495.
2. A. A. Laghari, H. He, K. A. Memon, *et al*., "Quality of experience (QoE) in cloud gaming models: A review," *Multiagent and Grid Systems*, Vol. 15, 2019, pp. 289-304.
3. C.-Y. Huang, D.-Y. Chen, C.-H. Hsu, and K.-T. Chen, "GamingAnywhere: an open-source cloud gaming testbed," in *Proceedings of the 21st ACM International Conference on Multimedia*, 2013, pp. 827-830.

4. W. Cai, R. Shea, C.-Y. Huang, K.-T. Chen, J. Liu, V. C.-M. Leung, and C.-H. Hsu, "A survey on cloud gaming: Future of computer games," *IEEE Access*, Vol. 4, 2016, pp. 7605-7620.
5. B. Vishnumolakala, "Performance evaluation of Gaming Anywhere server in a virtual environment," Thesis, Department of Communication Systems, Blekinge Institute of Technology, 2018.
6. A. Wibowo and T. N. B. Duong, "CloudNPlay: Resource optimization for a cloud-native gaming system," in *Proceedings of IEEE 30th Intl. Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises*, 2021, pp. 33-38.
7. D. Day and B. Burns, "A performance analysis of snort and suricata network intrusion detection and prevention engines," in *Proceedings of the 5th International Conference on Digital Society*, 2011, pp. 187-192.
8. E. Bertino and R. Sandhu, "Database security-concepts, approaches, and challenges," *IEEE Transactions on Dependable and Secure Computing*, Vol. 2, 2005, pp. 2-19.
9. M. Tebaa, S. El Hajji, and A. El Ghazi, "Homomorphic encryption applied to the cloud computing security," in *Proceedings of World Congress on Engineering*, Vol. 1, 2012, pp. 4-6.
10. C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, 2009, pp. 169-178.
11. D. Truong, *et al*., "Evaluating cloud-based gaming solutions," Master Thesis, Department of Communications and Humanities, The State University of New York Polytechnic Institute, 2021.
12. K. Alshouiliy and D. P. Agrawal, "Confluence of 4g lte, 5g, fog, and cloud computing and understanding security issues," in *Fog/Edge Computing For Security, Privacy, and Applications*, Vol. 83, 2021, pp. 3-32.
13. A. K. Murthy and R. Venkatesh, "Development of a reference architecture for streaming of cloud infotainment system to In-Car Thin clients," Master Thesis, Department of Computer Science and Engineering, Chalmers University of Technology, 2021.
14. A. E. Alchalabi, "Reinforcement learning based fair edge-user allocation for delay-sensitive edge computing applications," PhD Thesis, Electrical Engineering and Computer Science, University of Ottawa, 2021.
15. R. L. Neupane, T. Neely, N. Chettri, *et al*., "Dolus: cyber defense using pretense against DDoS attacks in cloud platforms," in *Proceedings of the 19th International Conference on Distributed Computing and Networking*, 2018, pp. 1-10.
16. A. Bhonde and S. Devane, "Impact of cloud attacks on service level agreement," in *Proceedings of International Conference on Communication Information and Computing Technology*, 2021, pp. 1-6.
17. Q. Chen, W. Lin, W. Dou, and S. Yu, "CBF: A packet filtering method for DDoS attack defense in cloud environment," in *Proceedings of IEEE 9th International Conference on Dependable, Autonomic and Secure Computing*, 2011, pp. 427-434.
18. H. Gu, J. Zhang, T. Liu, *et al*., "DIAVA: a traffic-based framework for detection of SQL injection attacks and vulnerability analysis of leaked data," *IEEE Transactions on Reliability*, Vol. 69, 2019, pp. 188-202.
19. I. Riadi, M. Kom, Y. Prayudi, and M. Kom, *et al*., "Investigasi bukti digital pada platform cloud gaming menggunakan framework FRED studi Kasus pada skyegrid cloud gaming services," Master Thesis, Universitas Islam Indonesia, 2021.

20. I. Abrar, S. N. Pottoo, F. S. Masoodi, and A. Bamhdi, "On IoT and its integration with cloud computing: Challenges and open issues," *Integration and Implementation of the Internet of Things Through Cloud Computing*, 2021, pp. 37-64.
21. K. Wang and Y. Hou, "Detection method of SQL injection attack in cloud computing environment," in *Proceedings of IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference*, 2016, pp. 487-493.
22. J. L. Carter and M. N. Wegman, "Universal classes of hash functions," *Journal of Computer and System Cciences*, Vol. 18, 1979, pp. 143-154.
23. R. Shea, J. Liu, E. C.-H. Ngai, and Y. Cui, "Cloud gaming: architecture and performance," *IEEE Network*, Vol. 27, 2013, pp. 16-21.
24. R. Anderson, F. Bergadano, B. Crispo, *et al*., "A new family of authentication protocols," *ACM SIGOPS Operating Systems Review*, Vol. 32, 1998, pp. 9-20.

Chun-I Fan received the MS degree in Computer Science and Information Engineering from the National Chiao Tung University, Hsinchu, Taiwan, in 1993, and the Ph.D. degree in Electrical Engineering from the National Taiwan University, Taipei, Taiwan, in 1998. From 1999 to 2003, he was an Associate Researcher and a Project Leader with Telecommunication Laboratories, Chunghwa Telecom Company, Ltd., Taoyuan, Taiwan. In 2003, he joined as a faculty in the Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan. He has been a Distinguished Professor since 2019. His research areas include applied cryptology, information and communication security.

Hsin-Nan Kuo was born in Pingtung, Taiwan. Presently, he is a Ph.D. scholar in the Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan. His current research includes cloud security, network/ information security, and applied cryptography.

Yung-Sheng Tu received the BSc degree in Computer Science and Engineering from National Sun Yat-sen University. His research interests include cloud computing, parallel computing, big data processing, and artificial intelligence.

Yuan-Chi Chei received the BS degree in Computer Science and Engineering from National Sun Yat-sen University. His research interests include artificial intelligence, intelligent information retrieval, and cloud computing.

Chu-Chia Chuang received the BS degree in Computer Science and Engineering from National Sun Yat-sen University. Her research interests include cloud computing, parallel computing, big data processing, and artificial intelligence.

Yu-Chun Tseng is a master's degree student in Information Security from National Sun Yat-sen University, Taiwan. His current research interests include cyber security and IoT network security.

Arijit Karati received his BS degree in Computer Applications from the University of Calcutta, West Bengal, India in 2011, and an MS degree in Computer Science from Pondicherry University, Puducherry, India in 2013. He received a Ph.D. degree in Computer Science and Engineering from the Indian Institute of Technology Dhanbad, India in 2018. He was a Postdoctoral Fellow in the Department of CSE, National Sun Yat-sen University, Kaohsiung, Taiwan. Presently, he is working as an Assistant Professor in the Department of CSE at NSYSU, Kaohsiung, Taiwan. His research areas include cryptology and cybersecurity.