Secure and guarantee QoS in a video sequence: a new approach based on TLS protocol to secure data and RTP to ensure real-time exchanges.

[1]HAMZA TOUIL, [1,2]NABIL EL AKKAD, [1]KHALID SATORI
[1]LISAC, Faculty of Sciences, Dhar-Mahraz (FSDM), Sidi Mohamed Ben Abdellah University, Fez, MOROCCO
[2]Laboratory of Engineering, Systems and Applications (LISA), National School of Applied Sciences (ENSA), Sidi Mohamed Ben Abdellah University, Fez, MOROCCO

Abstract: The continued development of networks has significantly contributed to increasing the quantity of information available to replace old intelligence-gathering methods faster and more efficiently. For this, it is necessary to implement services that meet the consumers' requirements and measure precisely the factors that can generate obstacles to any communication, among these causes we can cite strong security and high quality of services. In this work, we implement a secure approach useful in continuous communications in a time axis (video sequence, VOIP call...), the process consists in establishing a well-secured connection between two interlocutors (the server that broadcasts the video sequence and a client) using an AES encryption key of size 256. A step of jitter check (latency variation) periodically is essential for the customer in order to make a decision: If the jitter is within the standards (compared to the tolerable value), we continue to encrypt with the AES256 key, if no, both ends must go through an automatic and uninterrupted fast renegotiation of the video to switch to a small AES key (192,128) to reduce the bandwidth on the channel, this operation must be repeated in an alternative way until the end of the communication.

Keywords: QoS, security, AES, cypher suite, RTP, compromise, Sniffing. DDOS.

Received: January 1, 2021. Accepted: March 15, 2021. Published: April 2, 2021.

1. Introduction

Security is a significant challenge in network management and the ever-increasing number of individuals connecting to the Internet. The transmission of sensitive information and the desire to ensure this information's confidentiality has become an essential point in establishing computer networks [1-5]. Therefore, it is crucial to provide a stable technical and legal framework that guarantees adequate data protection. This new trend tends to become more than a rule of competitiveness; it is becoming a genuine legal obligation to protect personal data using adequate and sufficient security measures [8].

The recent strengthening of regulatory requirements has highlighted the security issues of systems (standard, sophisticated, intelligent...), applications, etc. The latter define and implement security policies, sometimes formalized, sometimes empirical, not only to cover the purpose of the system (authentication, prevent unauthorized disclosure of data, prevent unauthorized modification of data, prevent unauthorized use of network or computer resources in general ...). But also, at the level of choice of optimal and efficient algorithms, compatible with other solutions. Furthermore, the quality-of-service strategy specifies several network attributes such as clients or applications' priority and the actions for processing different traffic categories. However, in our case, we will deal more specifically with the QOS related to multimedia. The process consists of establishing a secured connection between two interlocutors (the server that broadcasts the video sequence and a client) using an AES encryption key of 256. A step of verification of the jitter (latency variation) periodic is essential on the part of the client to make a decision:

If the jitter is within the standards [6,7], the system must keep the encryption with the AES256 key, if not, both ends must go through an automatic and uninterrupted fast renegotiation of the video to switch to a small size AES key (192,128) to reduce the bandwidth on the channel, this operation must be repeated hastily until the end of the communication. This provides a full grasp of the security parameters to be addressed to the QoS objectives. To assess the needs in terms of security and quality of service, the proposed solution allows a compromise was found between better security and a better quality of service. Depending on the different test scenarios, the dimensions of this solution can be evaluated. However, in any case, the requirements are more critical, as they directly impact users [9-14]. In the rest of this document, we will dissect the related works. Then we will simulate the problem that led us to realize this solution and the added value of our work [28].

2. Related Works

A set of studies carried out in this context.[15], Proposed a framework for the quality of protection that corresponds to security and QoS requirements using a multi-attribute decision-making model. In other words, the algorithm puts the encryption keys in order of performance; then if there is degradation at the QOS level, the algorithm replaces the key with another performing month. In [16], Deals with service attacks in telecom networks are widespread and particularly severe. It treats security and QoS in an integrated way using the concept of Quality of Security Service where security is considered a parameter of quality of service. This solution works very well against service attacks.

Some protocols are part of the RTP family, which can ensure a certain level of security. SRTP [17] protocol provides encryption, authentication and integrity of messages and protects against the replay of RTP data. SRTP works in both unicast and multicast mode. In addition to preventing unauthorized eavesdropping on an RTP session, users can also limit the amount of personal information they provide. It recommended that applications do not issue RTCP source description packets without first informing the user. This protocol is robust in terms of security, and this is not the case in QOS, as it performs key changes at a given time interval and does not check the QOS parameters. ZRTP [18] describes a mechanism that allows two communicating parties to exchange encryption keys securely. In order to be able to encrypt traffic using SRTP. Although it is based on using the public key encryption algorithm, it does not require PKI or any particular infrastructure. The dialogue between the two parties carried out using the RTP protocol using specific extensions. Being independent of the signalling protocol is potentially compatible with all VOIP protocols (SIP, H323, Megaco...). A client that does not support ZRTP will ignore these extensions, without impacting communications. The key exchange is done peer to peer and does not require any central server. Infrastructure independence has been a priority in the design of this protocol.

3. Overview of Protocol SSL/TLS.

The purpose of the protocol is to provide secure data transmission. In this case, asymmetric encryption algorithms are used for authentication (a public-private key pair), and symmetric encryption algorithms (secret key) are used to maintain confidentiality. When a user visits a website, the browser requests certificate information from the server, and the server sends a copy of the SSL certificate together with the public key. Then, the browser checks the certificate, which must match the website's name, the validity date of the certificate, and the presence of a root certificate issued by a trusted certificate authority. If the browser trusts the certificate, it generates a session pre-master secret based on the public key using the highest level of encryption supported by both parties(figure1).

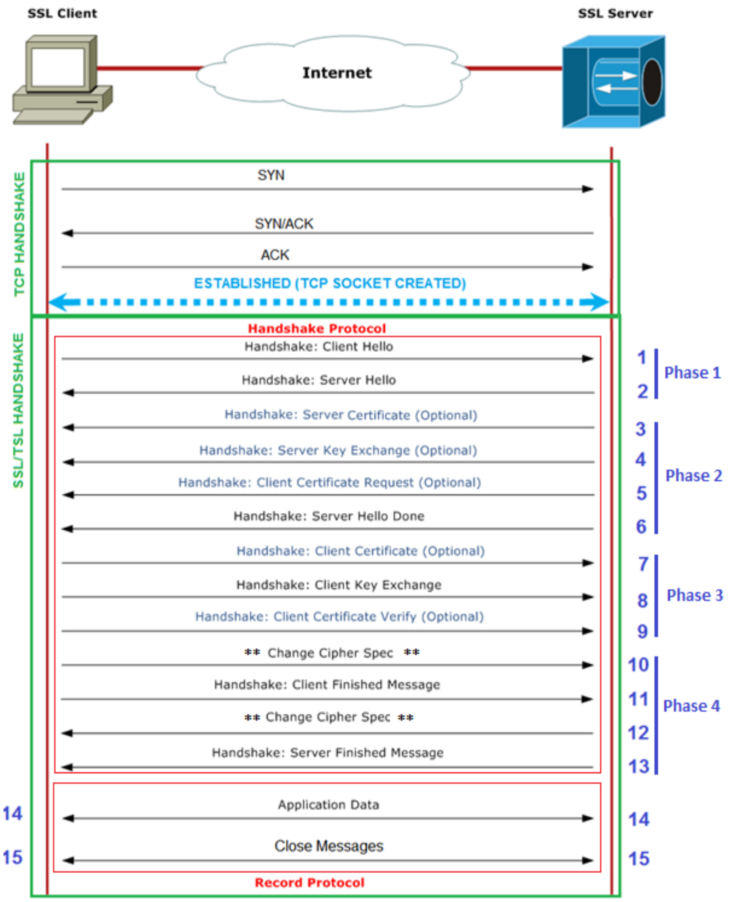

SSL Client
SSL Server
Internet
TCP HANDSHAKE
SYN
SYN/ACK
ACK
ESTABLISHED (TCP SOCKET CREATED)
SSL/TSL HANDSHAKE
Handshake Protocol
Handshake: Client Hello
Handshake: Server Hello
Handshake: Server Certificate (Optional)
Handshake: Server Key Exchange (Optional)
Handshake: Client Certificate Request (Optional)
Handshake: Server Hello Done
Handshake: Client Certificate (Optional)
Handshake: Client Key Exchange
Handshake: Client Certificate Verify (Optional)
** Change Cipher Spec **
Handshake: Client Finished Message
** Change Cipher Spec **
Handshake: Server Finished Message
Application Data
Close Messages
Record Protocol
1
2
Phase 1
3
4
5
6
Phase 2
7
8
9
Phase 3
10
11
12
13
Phase 4
14
15
14
15

Fig. 1. The steps of SSL communication.

The server decrypts the pre-master secret using its private key, agrees to continue communication, and creates a master secret using encryption. Both parties now use a symmetric key that is only valid for that session. Once completed, the key is destroyed, and the next time you visit the site, the contact process begins again [19,23,24,25].

4. Quality of Service.

Three main actors have essential stakes in designing and provisioning the Internet-based on the Internet Protocol (IP) [20]. These are the sender, the receiver and the Internet Service Provider (ISP). These actors compose the triangle of services (Fig.2). The sender wants to submit any form of traffic at any time (high load, saturation), while the receiver expects to receive all this sent traffic intact, with little delay (short delay, jitter, and packet loss). Also, the third player, the provider, wants to use the minimum possible network capacity per customer (whether sender or receiver) in order to be able to accommodate more customers on its network, resulting in higher profits.

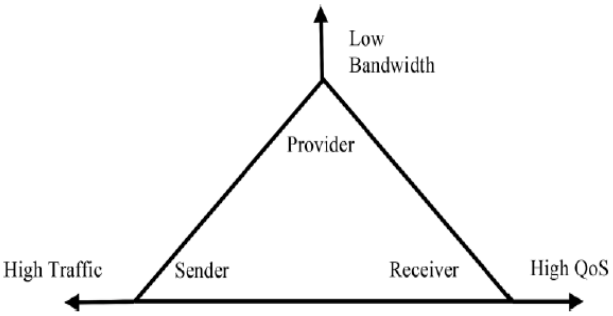


Fig. 2. QoS triangle

The first objective of QoS is to give priority services, including bandwidth, jitter, and latency. It can also say that QoS represents the set of techniques needed to manage network bandwidth, delay, jitter, and packet loss. Another relevant term that will use shortly is a network flow or stream. A flow can be defined in several ways. One of them refers to a combination of source and destination addresses, source and destination socket numbers, and session identifiers. It can also be defined more broadly as any packet from a particular application or an inbound interface.

Real-Time Protocol (RTP): The first formal effort to support end-to-end, real-time transfer of stream data over network IP. RTP is a session layer protocol, which runs above the Datagram Protocol (UDP) user layer and is therefore transparent to network routers. This is an essential distinction from later technologies and architectures where routers have a key role in providing QoS differentiation. To get a better idea of how a video sequence [21] operating in standards affects traffic flow's bandwidth requirements, we classify them into one primary and two secondary constraints. The primary constraint is that the packet loss rate must be less than 1%. The secondary is that the 95th percentile of the end-to-end delay should be less than 50 ms, and the second constraint that jitter should be less than 30 ms.

5. Possibilities of Attacks

This part aims to see the attacks that can cause a blockage on a communication channel such as the sniffing techniques [22] used by malware or hackers to exploit data that passes through a public network. Furthermore, it aims at identifying packets that circulate between communicators. This technique will make it possible to distinguish packets on routers or a communication channel thanks to dedicated tools (Wireshark in our case) connected to a database containing the attack model. If the sniffing system is detected as attacks, the firewall separates the Internet Protocol (IP) address. Then the communication between the attacker's host and the target will be interrupted.

To capture confidential information from the flow of data packets over the network, an attacker must install an appropriate "sniffer" (network protocol analyzer) on the victim's system, e.g., Wireshark, Ettercap, Bettercap, Tcpdump, WinDump. It may not be just software. Sometimes the monitoring is done from a hardware device connected to the system.

DOS/DDOS: Attacks target corporate servers in companies and websites, much less often - individuals' personal computers. The aim of these actions, as a rule, is one: to cause economic damage to those attacked and to remain in the shadows. In some cases, DoS and DDoS attacks are steps in server hacking and are aimed at stealing or destroying information. In fact, a company or website belonging to anyone can become a victim of cybercriminals. Generally, we can distinguish several types: In the case of a massive (volume-based) DDoS attack, many requests are often used, often sent from legitimate IP addresses, so that the site "drowns" in traffic. These attacks aim to "block" all available bandwidth and block legitimate traffic [26]. In a protocol-level attack (such as UDP or ICMP), the goal is to deplete system resources. To do this, open requests are sent, e.g., TCP/IP requests with a fake IP, and due to the exhaustion of network resources, it becomes impossible to process legitimate requests. Typical representatives are DDoS attacks, known in narrow circles as Smurf DDos, Ping of Death, and SYN flood. Another type of DDoS attack at the protocol level involves sending many fragmented packets that the system cannot handle. Layer 7 DDoS attacks are the sending of seemingly harmless requests that appear to result from normal user activity. Botnets and automated tools are generally used to implement them. Notable examples are Slowloris, Apache Killer, Cross-site scripting, SQL-injection, Remote file injection [27].

6. The Proposed Approach

The field of intervention of our method is wide; however, we focus on studying the transmission of a video sequence from a server (broadcaster) to a simple client (consumer). The client sends information to the server, such as the SSL protocol version, session ID, and encryption suites, and then the information such as the cryptographic algorithms and keys supported. The server chooses the best encryption suite supported by it and the client, and sends it to the client (Certificate (Public Key, Data)), and then requests the client to send its certificate if necessary. After the client verifies the certificate, it sends the encryption key used to encrypt messages; this is done once and for all in regular communication. However, in our case, we will modify it to be dynamic and automatic and linked to the channel and QoS status. To start with better security, we need to use a more secure key for this, and we need to start the encryption with the AES_256 key. The system will then automatically control the channel status through the existing parameters (latency, jitter...). If abnormal behavior is observed (network saturation, congestion ...), the system must intervene and change the key quickly to a smaller size than the one used initially and then continue the procedure. If stabilization is observed after, the system will change to a large key (figure 4).

6.1 Concept of our method.

The diagram below (Fig.3) describes the steps followed in our method.

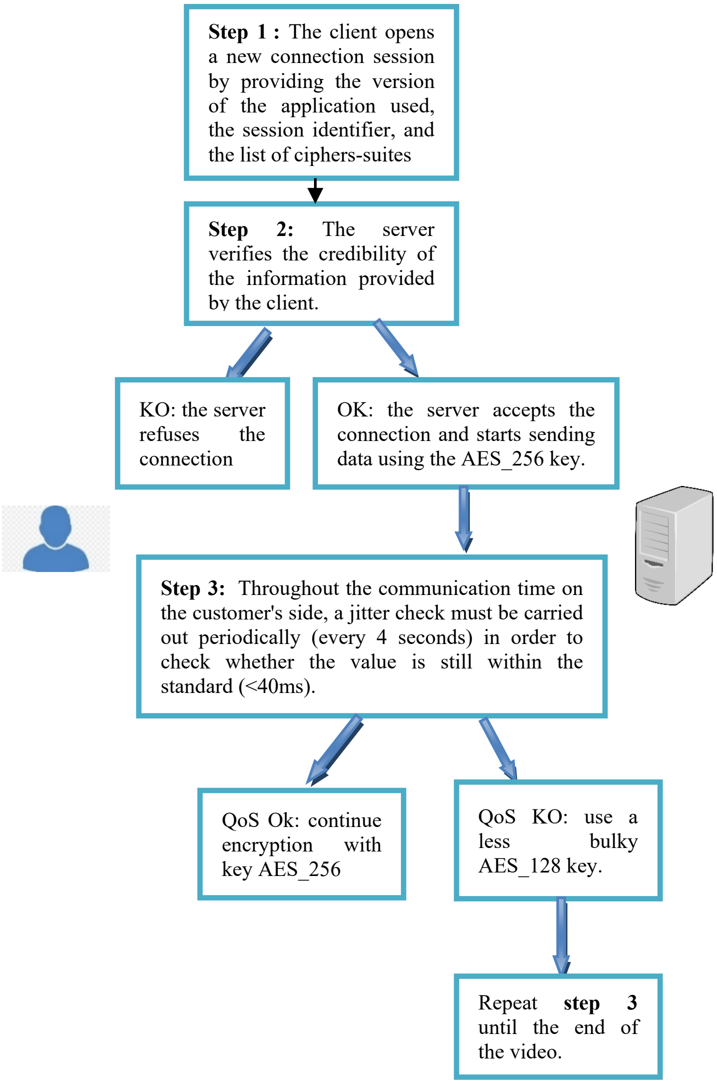

Step 1 : The client opens a new connection session by providing the version of the application used, the session identifier, and the list of ciphers-suites
Step 2: The server verifies the credibility of the information provided by the client.
KO: the server refuses the connection
OK: the server accepts the connection and starts sending data using the AES_256 key.
Step 3: Throughout the communication time on the customer's side, a jitter check must be carried out periodically (every 4 seconds) in order to check whether the value is still within the standard (<40ms).
QoS Ok: continue encryption with key AES_256
QoS KO: use a less bulky AES_128 key.
Repeat step 3 until the end of the video.

Fig. 3. Operating principle of our method.

This Algorithm will allow us to find compromises between various objectives:

➢ Automate renegotiations: this procedure did not exist before, because the negotiation is done once and for all, at the start of communication, but with this the algorithm we can have renegotiations if necessary, it all depends on the state of the channel, jitter, latency ...

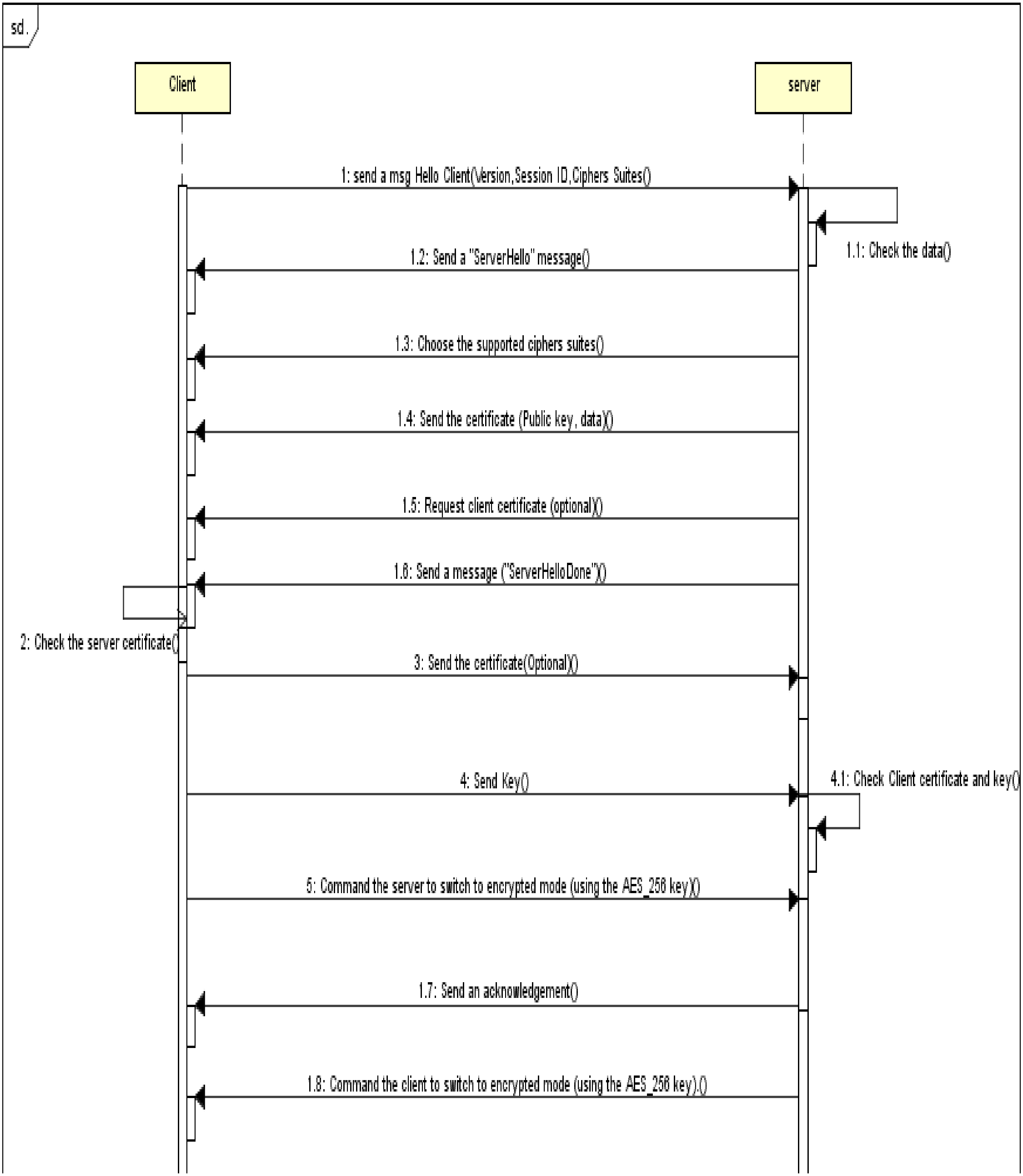
➢ Change the key in an alternative way: this option has two major advantages:

✓ The key used in the session is temporary, and therefore it will be more difficult for a hacker to attack the canal.

✓ There is no need to allocate significant resources to use more keys secure. Alternatively, to implement more or less weak keys to save costs resources. Because thanks to this solution we can make changes between keys in a flexible way.

➢ Optimization of resources: instead of allocating significant resources to implement solutions in the worst cases, thanks to this algorithm, we can reserve resources compatible with the current situation.

➢ An equilibrium between security and service quality: i.e., if the channel is loaded, then the latency is important. The algorithm chooses by default the cyphers suites with encryption keys and lightweight hashing to ensure a better quality of service possible.


sd.
Client
server
1: send a msg Hello Client(Version,Session ID,Ciphers Suites()
1.1: Check the data()
1.2: Send a "ServerHello" message()
1.3: Choose the supported ciphers suites()
1.4: Send the certificate (Public key, data)()
1.5: Request client certificate (optional)()
1.6: Send a message ("ServerHelloDone")()
2: Check the server certificate()
3: Send the certificate(Optional)()
4: Send Key()
4.1: Check Client certificate and key()
5: Command the server to switch to encrypted mode (using the AES_256 key)()
1.7: Send an acknowledgement()
1.8: Command the client to switch to encrypted mode (using the AES_256 key).()

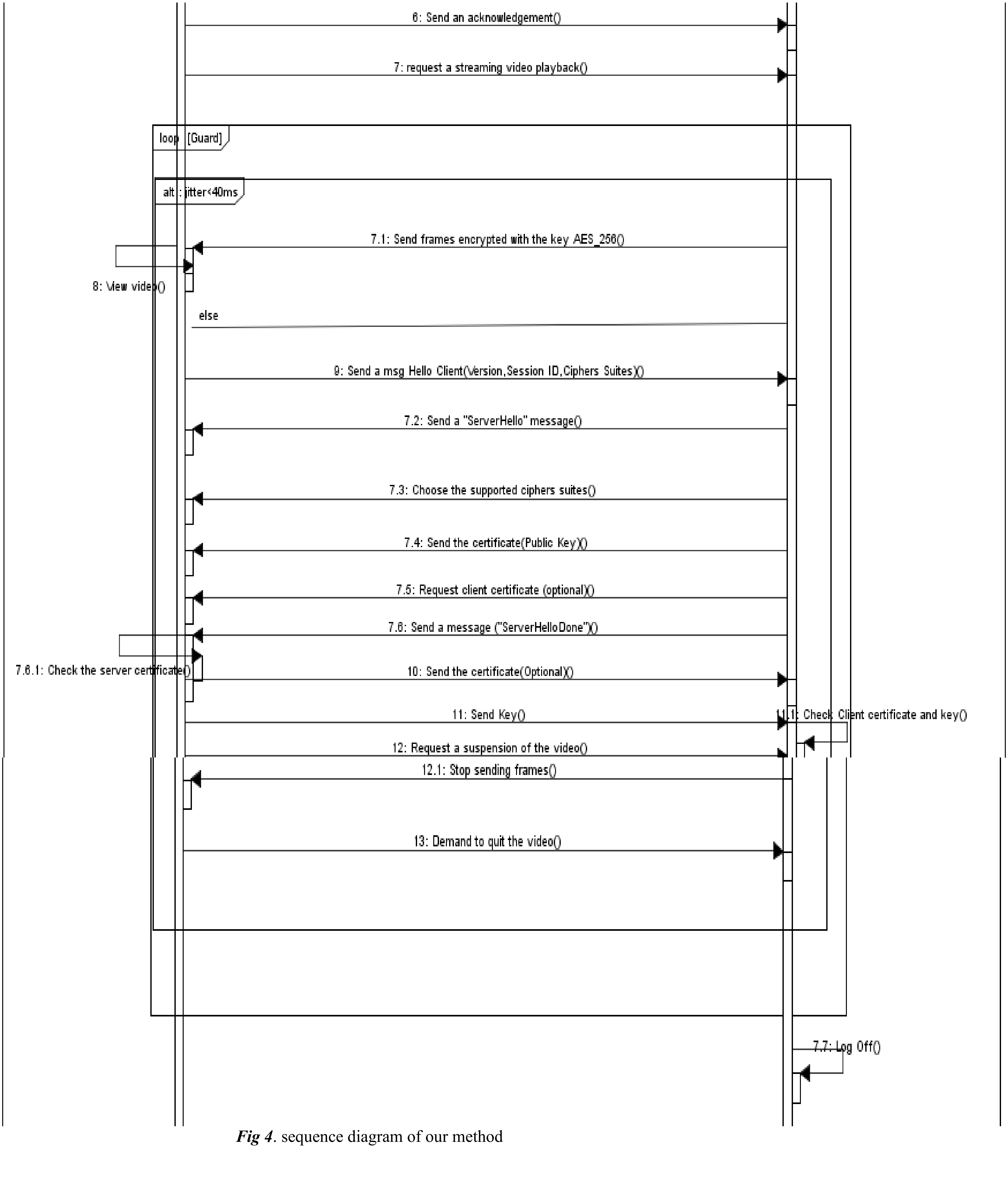
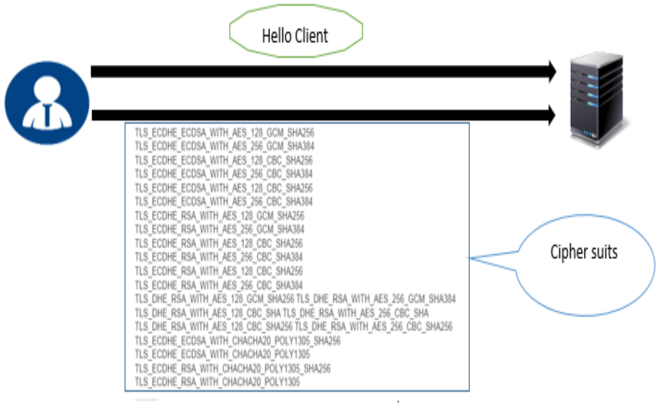

6: Send an acknowledgement()
7: request a streaming video playback()
loop [Guard]
alt : jitter<40ms
7.1: Send frames encrypted with the key AES_256()
8: View video()
else
9: Send a msg Hello Client(Version,Session ID,Ciphers Suites)()
7.2: Send a "ServerHello" message()
7.3: Choose the supported ciphers suites()
7.4: Send the certificate(Public Key)()
7.5: Request client certificate (optional)()
7.6: Send a message ("ServerHelloDone")()
7.6.1: Check the server certificate()
10: Send the certificate(Optional)()
11: Send Key()
11.1: Check Client certificate and key()
12: Request a suspension of the video()
12.1: Stop sending frames()
13: Demand to quit the video()
7.7: Log Off()

Fig 4. sequence diagram of our method

Our method is set up to satisfy users' needs by minimizing the workload due to the different treatments, i.e., to invent a dynamic algorithm that adapts to the different situations of the channel without any external intervention.

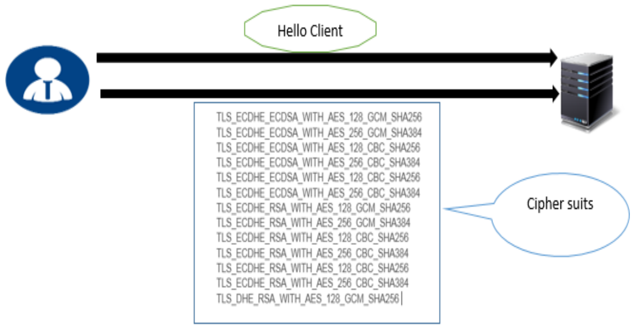
As already evoked the first phase consists of passing by a standard negotiation, the customer sends a hello + the list of cyphers suites that he supports as shown below (figure 5).



TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
TLS_DHE_RSA_WITH_AES_128_CBC_SHA TLS_DHE_RSA_WITH_AES_256_CBC_SHA
TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256
TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305
TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256
TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305

Fig. 5. Starting an SSL Negotiation

A modification will be made to the previous phase by applying a filter at the cipher's suits list to support only the AES encryption key and eliminate the DES,3DES keys since they are too much and are not compatible with this kind of exchange. The new list is as illustrated Figure 6.



TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256

Fig. 6. Parametric SSL negotiation

If you want to focus on a cipher's components, they usually consist of four parts, as shown in Figure 7.

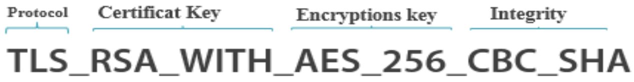


Fig. 7. Components of a cipher suits

After the server receives the client request the second filter is going to be applied this time on the server-side to tolerate that the AES key size 256 in the suggestions the goal is to start with a higher security level, using a more secure key of ample size for that we must start the encryption with the key AES_256(we do not take into account the robustness of the hash key in our study)(figure 8).

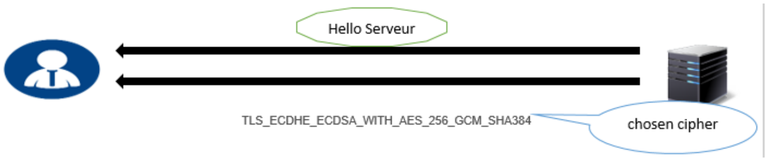


Fig. 8. Parametric SSL negotiation_ AES256.

The next step is to ensure confidentiality, so the server must send a certificate validated by a CA (Certification Authority))(figure 9).

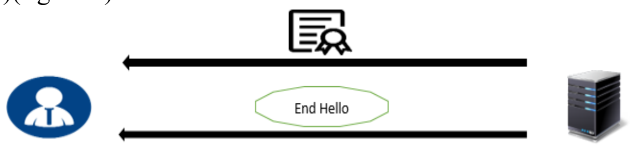


Fig. 9. Final round of negotiations

After the certificate verification phase on the part of the client, it sends the encryption key used to encrypt the messages; this phase is carried out once and for all in regular communication. However, in our case, we will modify it so that it is dynamic and automatic is linked to the channel and the QoS state. The certificate's sending on the client-side remains optional so that the two interlocutors can securely exchange data.

At this point, we have managed to provide favorable security but assuming we are facing a DDOS attack?

Our method can detect this attack on our flag for the security measures, which can make our task more manageable. The "timestamp" field is available on the RTP frames for our service at the calculation level to the latency variation (JITTER). We remind you that the tolerable value for videos estimated at 40ms.

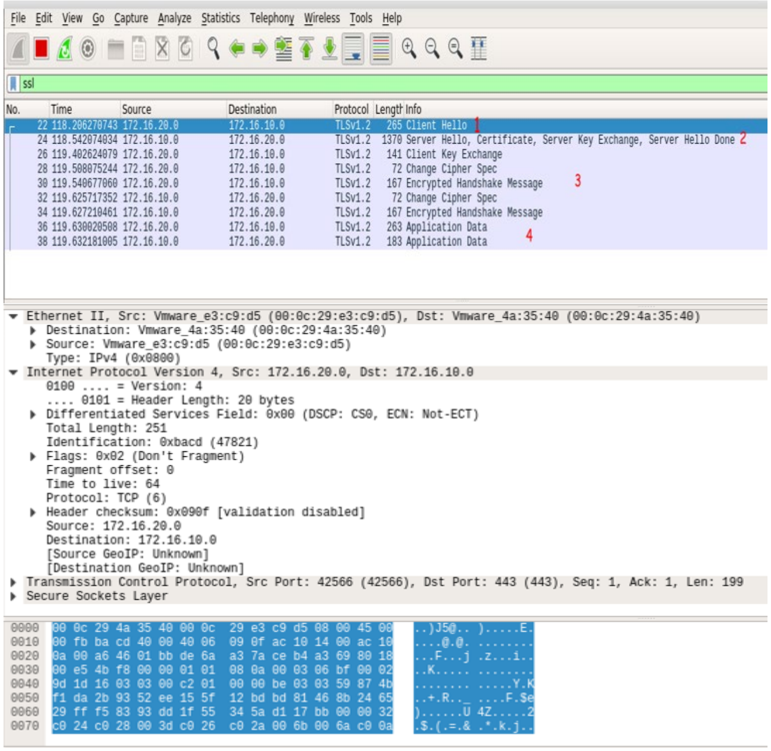
After every 4 seconds, a check of the different QOS parameters (packet loss, latency, jitter) is done automatically if the system detects one or more abnormal things (e.g., jitter exceeds the tolerable threshold of 40 ms as a result of a DOS attack), i.e., the channel is well loaded and can cause service degradation. So a switch to another small key is essential to reduce the size of the frames sent.

In time, an axis that does not exceed one second, the client must request a small key through a quick renegotiation with the server. However, this time he must propose the AES128 keys in the list of cipher suites.

In the same way, after every 4 seconds, a jitter calculation is done automatically. If the jitter is above the tolerable threshold, we always keep the same key if we move to the next size to increase the security.

6.2 Experiments

To simulate our method, we have used two virtual machines that use a Linux operating system, and the first machine will be the client and the second a secure broadcast server, which can stream videos on demand in a secure mode. The server uses RTP to transport the data in real-time, and SSL to secure the exchanges. One or more clients can retrieve and manipulate the video remotely using the RTSP protocol. To retrieve a video sequence, the client sends a request to create the channel and initialize the session. At this step, the client and the server make exchanges (Certificate, encryption key, cyphers suites, ...) We used the Wireshark tool to capture different interactions. The client sends to the server information such as SSL protocol version, session id, and cypher suites information such as cryptographer algorithms and supported keys. Then the server selects the cypher suite supported by it and the client. Client and server exchange certificates with each other; each certificate contains a public key plus data specific to the certificate. After the certificate verification phase, the client sends the encryption key used to encrypt messages; this phase is performed once and for all in regular communication. However, in our case, we will modify it to be dynamic and automatic is linked to the channel status and the QoS. A standard IP frame with the essential elements (source and destination address, version, flags, fragments, TTL, total length...) encapsulates the data sent (figure10).

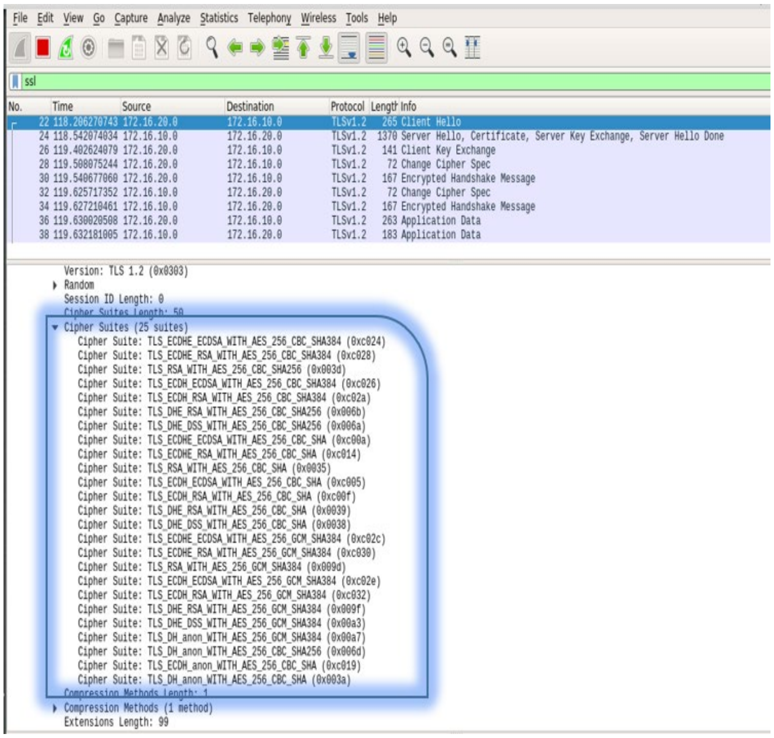

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
ssl
1
2
3
4

```
No.   Time          Source       Destination  Protocol Length Info
   22 118.206270743 172.16.20.0  172.16.10.0  TLSv1.2   265 Client Hello
   24 118.542074034 172.16.10.0  172.16.20.0  TLSv1.2  1370 Server Hello, Certificate, Server Key Exchange, Server Hello Done
   26 119.402624079 172.16.20.0  172.16.10.0  TLSv1.2   141 Client Key Exchange
   28 119.508075244 172.16.20.0  172.16.10.0  TLSv1.2    72 Change Cipher Spec
   30 119.540677060 172.16.20.0  172.16.10.0  TLSv1.2   167 Encrypted Handshake Message
   32 119.625717352 172.16.10.0  172.16.20.0  TLSv1.2    72 Change Cipher Spec
   34 119.627210461 172.16.10.0  172.16.20.0  TLSv1.2   167 Encrypted Handshake Message
   36 119.630020508 172.16.20.0  172.16.10.0  TLSv1.2   263 Application Data
   38 119.632181005 172.16.10.0  172.16.20.0  TLSv1.2   183 Application Data

▼ Ethernet II, Src: Vmware_e3:c9:d5 (00:0c:29:e3:c9:d5), Dst: Vmware_4a:35:40 (00:0c:29:4a:35:40)
  ▶ Destination: Vmware_4a:35:40 (00:0c:29:4a:35:40)
  ▶ Source: Vmware_e3:c9:d5 (00:0c:29:e3:c9:d5)
    Type: IPv4 (0x0800)
▼ Internet Protocol Version 4, Src: 172.16.20.0, Dst: 172.16.10.0
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes
  ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 251
    Identification: 0xbacd (47821)
  ▶ Flags: 0x02 (Don't Fragment)
    Fragment offset: 0
    Time to live: 64
    Protocol: TCP (6)
  ▶ Header checksum: 0x090f [validation disabled]
    Source: 172.16.20.0
    Destination: 172.16.10.0
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
▶ Transmission Control Protocol, Src Port: 42566 (42566), Dst Port: 443 (443), Seq: 1, Ack: 1, Len: 199
▶ Secure Sockets Layer

0000  00 0c 29 4a 35 40 00 0c  29 e3 c9 d5 08 00 45 00   ..)J5@.. ).....E.
0010  00 fb ba cd 40 00 40 06  09 0f ac 10 14 00 ac 10   ....@.@. ........
0020  0a 00 a6 46 01 bb de 6a  a3 7a ce b4 a3 69 80 18   ...F...j .z...i..
0030  00 e5 4b f8 00 00 01 01  08 0a 00 03 06 bf 00 02   ..K..... ........
0040  9d 1d 16 03 03 00 c2 01  00 00 be 03 03 59 87 4b   ........ .....Y.K
0050  f1 da 2b 93 52 ee 15 5f  12 bd bd 81 46 8b 24 65   ..+.R.._ ....F.$e
0060  29 ff f5 83 93 dd 1f 55  34 5a d1 17 bb 00 00 32   )......U 4Z.....2
0070  c0 24 c0 28 00 3d c0 26  c0 2a 00 6b 00 6a c0 0a   .$.(.=.& .*.k.j..
```

Fig .10. Initialization of the SSL session under Wireshark.

As marked in red, in the preceding figure the number (1) the components of the hello-client frame, (2) the server response, (3) the session encryption, and then (4) the data exchanges.

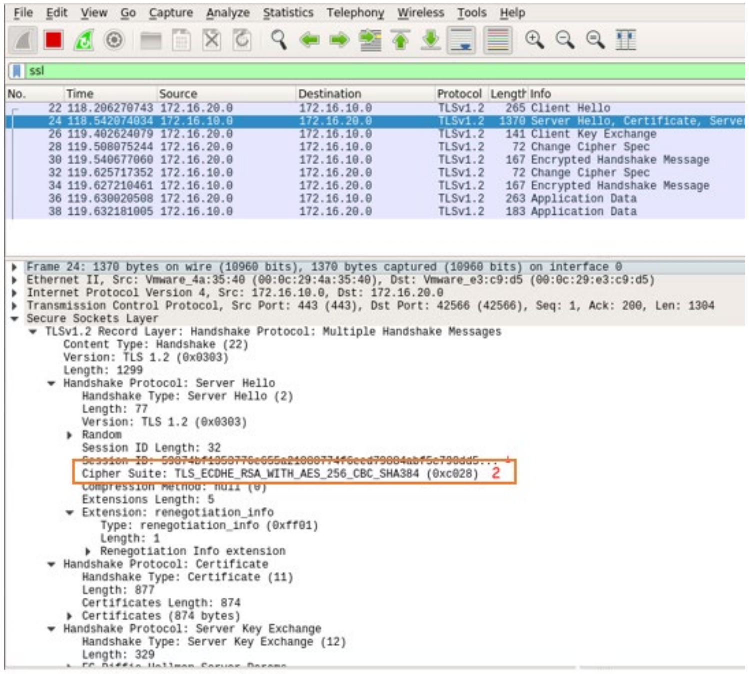
2) The customer must send that the ciphers-suites whose encryption key is AES256 (as described in Figure 11):


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
ssl

```
No.   Time          Source       Destination  Protocol Length Info
   22 118.206270743 172.16.20.0  172.16.10.0  TLSv1.2   265 Client Hello
   24 118.542074034 172.16.10.0  172.16.20.0  TLSv1.2  1370 Server Hello, Certificate, Server Key Exchange, Server Hello Done
   26 119.402624079 172.16.20.0  172.16.10.0  TLSv1.2   141 Client Key Exchange
   28 119.508075244 172.16.20.0  172.16.10.0  TLSv1.2    72 Change Cipher Spec
   30 119.540677060 172.16.20.0  172.16.10.0  TLSv1.2   167 Encrypted Handshake Message
   32 119.625717352 172.16.10.0  172.16.20.0  TLSv1.2    72 Change Cipher Spec
   34 119.627210461 172.16.10.0  172.16.20.0  TLSv1.2   167 Encrypted Handshake Message
   36 119.630020508 172.16.20.0  172.16.10.0  TLSv1.2   263 Application Data
   38 119.632181005 172.16.10.0  172.16.20.0  TLSv1.2   183 Application Data

    Version: TLS 1.2 (0x0303)
  ▶ Random
    Session ID Length: 0
    Cipher Suites Length: 50
  ▼ Cipher Suites (25 suites)
      Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 (0xc024)
      Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xc028)
      Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA256 (0x003d)
      Cipher Suite: TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384 (0xc026)
      Cipher Suite: TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384 (0xc02a)
      Cipher Suite: TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 (0x006b)
      Cipher Suite: TLS_DHE_DSS_WITH_AES_256_CBC_SHA256 (0x006a)
      Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0xc00a)
      Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)
      Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
      Cipher Suite: TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA (0xc005)
      Cipher Suite: TLS_ECDH_RSA_WITH_AES_256_CBC_SHA (0xc00f)
      Cipher Suite: TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x0039)
      Cipher Suite: TLS_DHE_DSS_WITH_AES_256_CBC_SHA (0x0038)
      Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c)
      Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)
      Cipher Suite: TLS_RSA_WITH_AES_256_GCM_SHA384 (0x009d)
      Cipher Suite: TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02e)
      Cipher Suite: TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384 (0xc032)
      Cipher Suite: TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (0x009f)
      Cipher Suite: TLS_DHE_DSS_WITH_AES_256_GCM_SHA384 (0x00a3)
      Cipher Suite: TLS_DH_anon_WITH_AES_256_GCM_SHA384 (0x00a7)
      Cipher Suite: TLS_DH_anon_WITH_AES_256_CBC_SHA256 (0x006d)
      Cipher Suite: TLS_ECDH_anon_WITH_AES_256_CBC_SHA (0xc019)
      Cipher Suite: TLS_DH_anon_WITH_AES_256_CBC_SHA (0x003a)
    Compression Methods Length: 1
  ▶ Compression Methods (1 method)
    Extensions Length: 99
```

Fig .11. The list of AES_256 suite ciphers provided by TLS1.2.

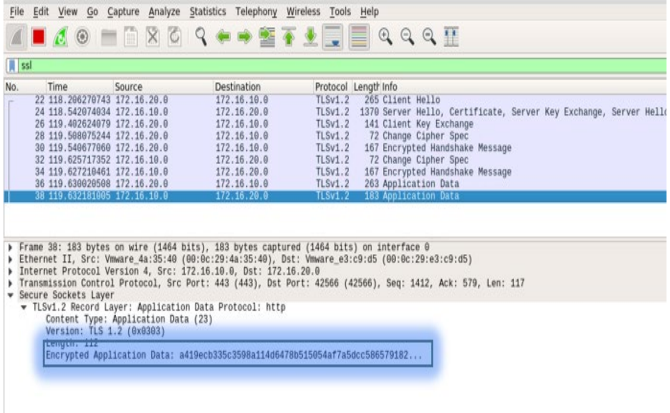
3) The server chooses the first support cipher on its part, as shown below with the number (2) in red (Figure 12).


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
ssl

```
No.   Time          Source        Destination   Protocol Length Info
   22 118.206270743 172.16.20.0   172.16.10.0   TLSv1.2   265 Client Hello
   24 118.542074034 172.16.10.0   172.16.20.0   TLSv1.2  1370 Server Hello, Certificate, Server
   26 119.402624079 172.16.20.0   172.16.10.0   TLSv1.2   141 Client Key Exchange
   28 119.508075244 172.16.20.0   172.16.10.0   TLSv1.2    72 Change Cipher Spec
   30 119.540677060 172.16.20.0   172.16.10.0   TLSv1.2   167 Encrypted Handshake Message
   32 119.625717352 172.16.10.0   172.16.20.0   TLSv1.2    72 Change Cipher Spec
   34 119.627210461 172.16.10.0   172.16.20.0   TLSv1.2   167 Encrypted Handshake Message
   36 119.630020508 172.16.20.0   172.16.10.0   TLSv1.2   263 Application Data
   38 119.632181005 172.16.10.0   172.16.20.0   TLSv1.2   183 Application Data

▸ Frame 24: 1370 bytes on wire (10960 bits), 1370 bytes captured (10960 bits) on interface 0
▸ Ethernet II, Src: Vmware_4a:35:40 (00:0c:29:4a:35:40), Dst: Vmware_e3:c9:d5 (00:0c:29:e3:c9:d5)
▸ Internet Protocol Version 4, Src: 172.16.10.0, Dst: 172.16.20.0
▸ Transmission Control Protocol, Src Port: 443 (443), Dst Port: 42566 (42566), Seq: 1, Ack: 200, Len: 1304
▾ Secure Sockets Layer
  ▾ TLSv1.2 Record Layer: Handshake Protocol: Multiple Handshake Messages
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 1299
    ▾ Handshake Protocol: Server Hello
        Handshake Type: Server Hello (2)
        Length: 77
        Version: TLS 1.2 (0x0303)
      ▸ Random
        Session ID Length: 32
        Session ID: 59874bf1950776e655a21000774f6eed79004abf5e790dd5...
        Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xc028)  2
        Compression Method: null (0)
        Extensions Length: 5
      ▾ Extension: renegotiation_info
          Type: renegotiation_info (0xff01)
          Length: 1
        ▸ Renegotiation Info extension
    ▾ Handshake Protocol: Certificate
        Handshake Type: Certificate (11)
        Length: 877
        Certificates Length: 874
      ▸ Certificates (874 bytes)
    ▾ Handshake Protocol: Server Key Exchange
        Handshake Type: Server Key Exchange (12)
        Length: 329
      ▸ EC Diffie-Hellman Server Params
```

Fig .12. The choice of Cipher suit

4) After starting the video playback, the resulting packets are encrypted using the denial function during the negotiation and write key. The algorithms used for encryption are AES_256(figure 13).


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
ssl

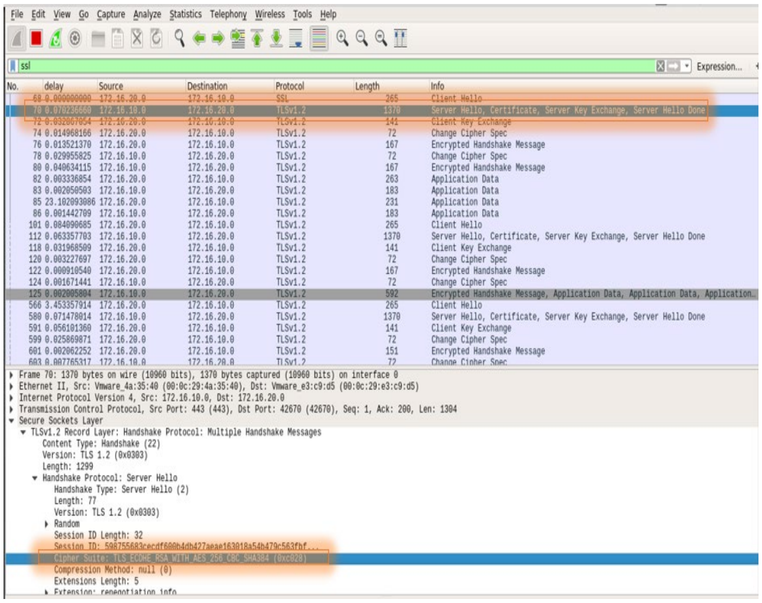
```
No.   Time          Source        Destination   Protocol Length Info
   22 118.206270743 172.16.20.0   172.16.10.0   TLSv1.2   265 Client Hello
   24 118.542074034 172.16.10.0   172.16.20.0   TLSv1.2  1370 Server Hello, Certificate, Server Key Exchange, Server Hello
   26 119.402624079 172.16.20.0   172.16.10.0   TLSv1.2   141 Client Key Exchange
   28 119.508075244 172.16.20.0   172.16.10.0   TLSv1.2    72 Change Cipher Spec
   30 119.540677060 172.16.20.0   172.16.10.0   TLSv1.2   167 Encrypted Handshake Message
   32 119.625717352 172.16.10.0   172.16.20.0   TLSv1.2    72 Change Cipher Spec
   34 119.627210461 172.16.10.0   172.16.20.0   TLSv1.2   167 Encrypted Handshake Message
   36 119.630020508 172.16.20.0   172.16.10.0   TLSv1.2   263 Application Data
   38 119.632181005 172.16.10.0   172.16.20.0   TLSv1.2   183 Application Data

▸ Frame 38: 183 bytes on wire (1464 bits), 183 bytes captured (1464 bits) on interface 0
▸ Ethernet II, Src: Vmware_4a:35:40 (00:0c:29:4a:35:40), Dst: Vmware_e3:c9:d5 (00:0c:29:e3:c9:d5)
▸ Internet Protocol Version 4, Src: 172.16.10.0, Dst: 172.16.20.0
▸ Transmission Control Protocol, Src Port: 443 (443), Dst Port: 42566 (42566), Seq: 1412, Ack: 579, Len: 117
▾ Secure Sockets Layer
  ▾ TLSv1.2 Record Layer: Application Data Protocol: http
      Content Type: Application Data (23)
      Version: TLS 1.2 (0x0303)
      Length: 112
      Encrypted Application Data: a419ecb335c3598a114d6478b515054af7a5dcc586579182...
```

Fig. 13. Data Encryption.

As already mentioned, the algorithm must work in different scenarios. It must also be able to automate the change of keys; for this, we will discuss both solutions:
Free channel: in ordinary cases using two virtual machines connected, a server and a client, then observe the results under Wireshark.
Saturated channel: in this case, we used the Hping3 tool to apply a DOS attack, to load the channel and see the behavior of the algorithm.

5) Free Channel

After starting the video at the client, the calculation of the different QOS parameters (latency, jitter.) is done automatically every 4 seconds. If one or more parameters exceed the recommended thresholds, the key must be changed. The encryption is carried out with the AES_256 key (Figure 14).

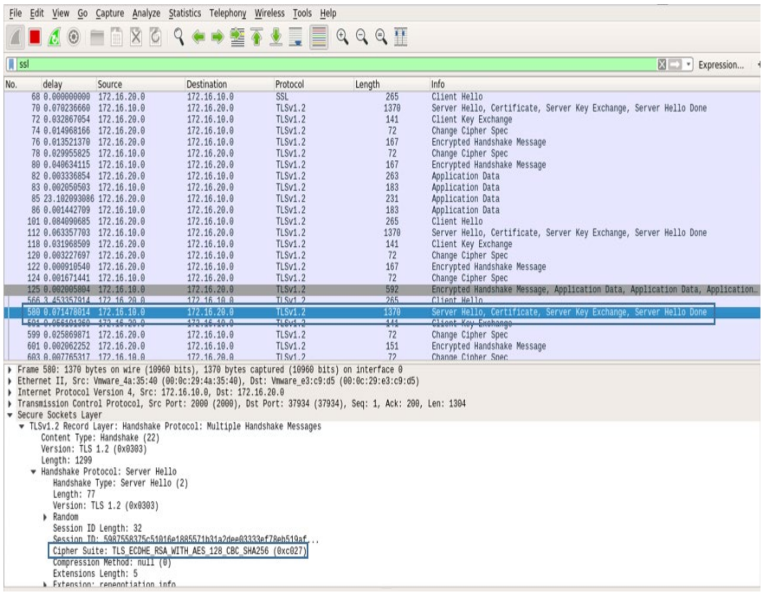

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
ssl
Expression...

```
No.  delay        Source       Destination  Protocol  Length  Info
  68 0.000000000  172.16.20.0  172.16.10.0  SSL        265    Client Hello
  70 0.070236660  172.16.10.0  172.16.20.0  TLSv1.2   1370    Server Hello, Certificate, Server Key Exchange, Server Hello Done
  72 0.032067054  172.16.20.0  172.16.10.0  TLSv1.2    141    Client Key Exchange
  74 0.014968166  172.16.20.0  172.16.10.0  TLSv1.2     72    Change Cipher Spec
  76 0.013521370  172.16.20.0  172.16.10.0  TLSv1.2    167    Encrypted Handshake Message
  78 0.029955825  172.16.10.0  172.16.20.0  TLSv1.2     72    Change Cipher Spec
  80 0.040634115  172.16.10.0  172.16.20.0  TLSv1.2    167    Encrypted Handshake Message
  82 0.003336854  172.16.20.0  172.16.10.0  TLSv1.2    263    Application Data
  83 0.002050503  172.16.10.0  172.16.20.0  TLSv1.2    183    Application Data
  85 23.102093086 172.16.20.0  172.16.10.0  TLSv1.2    231    Application Data
  86 0.001442709  172.16.10.0  172.16.20.0  TLSv1.2    183    Application Data
 101 0.084090685  172.16.20.0  172.16.10.0  TLSv1.2    265    Client Hello
 112 0.063357703  172.16.10.0  172.16.20.0  TLSv1.2   1370    Server Hello, Certificate, Server Key Exchange, Server Hello Done
 118 0.031968509  172.16.20.0  172.16.10.0  TLSv1.2    141    Client Key Exchange
 120 0.003227697  172.16.20.0  172.16.10.0  TLSv1.2     72    Change Cipher Spec
 122 0.000910540  172.16.20.0  172.16.10.0  TLSv1.2    167    Encrypted Handshake Message
 124 0.001671441  172.16.10.0  172.16.20.0  TLSv1.2     72    Change Cipher Spec
 125 0.002005804  172.16.10.0  172.16.20.0  TLSv1.2    592    Encrypted Handshake Message, Application Data, Application Data, Application…
 566 3.453357914  172.16.20.0  172.16.10.0  TLSv1.2    265    Client Hello
 580 0.071478014  172.16.10.0  172.16.20.0  TLSv1.2   1370    Server Hello, Certificate, Server Key Exchange, Server Hello Done
 591 0.056101360  172.16.20.0  172.16.10.0  TLSv1.2    141    Client Key Exchange
 599 0.025869871  172.16.20.0  172.16.10.0  TLSv1.2     72    Change Cipher Spec
 601 0.002062252  172.16.20.0  172.16.10.0  TLSv1.2    151    Encrypted Handshake Message
 603 0.007765317  172.16.10.0  172.16.20.0  TLSv1.2     72    Change Cipher Spec

▸ Frame 70: 1370 bytes on wire (10960 bits), 1370 bytes captured (10960 bits) on interface 0
▸ Ethernet II, Src: Vmware_4a:35:40 (00:0c:29:4a:35:40), Dst: Vmware_e3:c9:d5 (00:0c:29:e3:c9:d5)
▸ Internet Protocol Version 4, Src: 172.16.10.0, Dst: 172.16.20.0
▸ Transmission Control Protocol, Src Port: 443 (443), Dst Port: 42670 (42670), Seq: 1, Ack: 200, Len: 1304
▾ Secure Sockets Layer
  ▾ TLSv1.2 Record Layer: Handshake Protocol: Multiple Handshake Messages
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 1299
    ▾ Handshake Protocol: Server Hello
        Handshake Type: Server Hello (2)
        Length: 77
        Version: TLS 1.2 (0x0303)
      ▸ Random
        Session ID Length: 32
        Session ID: 598755683cecdf600b4db427aeae163018a54b479c563fbf...
        Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xc028)
        Compression Method: null (0)
        Extensions Length: 5
      ▸ Extension: renegotiation_info
```

Fig .14. encrypt with the AES256 key

6) Channel saturated

As explained, Denial of Service is a technique that consists of sending a data stream that is too large concerning what the target can receive and process. If someone has an IP address and wants to deny you access to the Internet or block access to your site, they will be able to do so if they have a sufficiently large connection. It will then flood and saturate upload bandwidth, which will cause a massive disruption to Internet Traffic in both directions. After performing a DOS attack with the Hping3 tool to saturate the channel, it turns out that the algorithm only uses the AES_128 key to minimize the packet size and manage network congestion.
The client starts the encryption with the AES_256 key to having a reliable security level. However, as soon as the jitter exceeds the 30ms threshold, it is necessary to automatically change the key to AES_128 to reduce the frames' size and facilitate the communication as mentioned (figure 15).



```
No.  delay        Source       Destination  Protocol  Length  Info
  68 0.000000000  172.16.20.0  172.16.10.0  SSL       265     Client Hello
  70 0.070236660  172.16.10.0  172.16.20.0  TLSv1.2   1370    Server Hello, Certificate, Server Key Exchange, Server Hello Done
  72 0.032867054  172.16.20.0  172.16.10.0  TLSv1.2   141     Client Key Exchange
  74 0.014968166  172.16.20.0  172.16.10.0  TLSv1.2   72      Change Cipher Spec
  76 0.013521370  172.16.20.0  172.16.10.0  TLSv1.2   167     Encrypted Handshake Message
  78 0.029955825  172.16.10.0  172.16.20.0  TLSv1.2   72      Change Cipher Spec
  80 0.040634115  172.16.10.0  172.16.20.0  TLSv1.2   167     Encrypted Handshake Message
  82 0.003336854  172.16.20.0  172.16.10.0  TLSv1.2   263     Application Data
  83 0.002050503  172.16.10.0  172.16.20.0  TLSv1.2   183     Application Data
  85 23.102093086 172.16.20.0  172.16.10.0  TLSv1.2   231     Application Data
  86 0.001442709  172.16.10.0  172.16.20.0  TLSv1.2   183     Application Data
 101 0.084090685  172.16.20.0  172.16.10.0  TLSv1.2   265     Client Hello
 112 0.063357703  172.16.10.0  172.16.20.0  TLSv1.2   1370    Server Hello, Certificate, Server Key Exchange, Server Hello Done
 118 0.031968509  172.16.20.0  172.16.10.0  TLSv1.2   141     Client Key Exchange
 120 0.003227697  172.16.20.0  172.16.10.0  TLSv1.2   72      Change Cipher Spec
 122 0.000910540  172.16.20.0  172.16.10.0  TLSv1.2   167     Encrypted Handshake Message
 124 0.001671441  172.16.10.0  172.16.20.0  TLSv1.2   72      Change Cipher Spec
 125 0.002005804  172.16.10.0  172.16.20.0  TLSv1.2   592     Encrypted Handshake Message, Application Data, Application Data, Application…
 566 3.453357914  172.16.20.0  172.16.10.0  TLSv1.2   265     Client Hello
 580 0.071478014  172.16.10.0  172.16.20.0  TLSv1.2   1370    Server Hello, Certificate, Server Key Exchange, Server Hello Done
 581 0.056101360  172.16.20.0  172.16.10.0  TLSv1.2   141     Client Key Exchange
 599 0.025869871  172.16.20.0  172.16.10.0  TLSv1.2   72      Change Cipher Spec
 601 0.002062252  172.16.20.0  172.16.10.0  TLSv1.2   151     Encrypted Handshake Message
 603 0.007765317  172.16.10.0  172.16.20.0  TLSv1.2   72      Change Cipher Spec

▸ Frame 580: 1370 bytes on wire (10960 bits), 1370 bytes captured (10960 bits) on interface 0
▸ Ethernet II, Src: Vmware_4a:35:40 (00:0c:29:4a:35:40), Dst: Vmware_e3:c9:d5 (00:0c:29:e3:c9:d5)
▸ Internet Protocol Version 4, Src: 172.16.10.0, Dst: 172.16.20.0
▸ Transmission Control Protocol, Src Port: 2000 (2000), Dst Port: 37934 (37934), Seq: 1, Ack: 200, Len: 1304
▾ Secure Sockets Layer
  ▾ TLSv1.2 Record Layer: Handshake Protocol: Multiple Handshake Messages
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 1299
    ▾ Handshake Protocol: Server Hello
        Handshake Type: Server Hello (2)
        Length: 77
        Version: TLS 1.2 (0x0303)
      ▸ Random
        Session ID Length: 32
        Session ID: 5987558375c51016e1885571b31a2dee03333ef78eb519af...
        Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xc027)
        Compression Method: null (0)
        Extensions Length: 5
      ▸ Extension: renegotiation_info
```

Fig .15. The exchange of the key to AES_128

7. Conclusions

This study has allowed us to move on to a more important phase that citing the different needs, dysfunctions, and challenges we have encountered. Afterward, we carried out studies on the requirements and the different possible approaches to realize this hybrid algorithm based on RTP and SSL protocols.

The security provided by standard RTP is insufficient because it does not support authentication, and its default encryption algorithm (DES) is fragile at present and simpler to hack.

The biggest security challenge is the management of security keys, how to distribute them, how to store and update them, how to protect them from hackers. For this purpose, we thought about realizing a dynamic and automatic security solution.

References

[1] Mohammed A. Al-Maqri, Ali Mohammed Mansoor Aznul Qalid Sabri3 Sri Devi Ravana4 Hussein Soubhi Yaseein; High performing multimedia transmission approach based on QoS support and admission control over IEEE 802.11e networks , International Journal of Communication Systems,march 2020. https://doi.org/10.1002/dac.4193

[2] Serhrouchni. ;Integration of the digital signature in the protocol SSL/TLS Intégration de la signature numérique au protocole SSL/TLS; Annales des Telecommunications/Annals of Telecommunications, P 522-541,May/June 2006.

[3] Hu, Fan, Zhang; An effective differential power attack method for advanced encryption standard, P (58-61); International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, CyberC 2019.

[4] Nakasone, T., Li, Y., Yu, S; Key-Dependent Weakness of AES-Based Ciphers under Clockwise Collision Distinguisher; International Conference on Information Security & Cryptology. (2012).

[5] Douglas_R._Stinson,_Maura_B._Paterson; Cryptography théorie and pratices ; International Standard Book Number-13: 978-1-1381-9701-5,2019.

[6] Pisheh, M.A.Z., Sheikhi, A; DETECTION AND COMPENSATION OF IMAGE SEQUENCE JITTER DUE TO AN UNSTABLE CCD CAMERA FOR VIDEO TRACKING OF A MOVING TARGET; Proceedings - 2nd International Symposium on 3D Data Processing, Visualization, and Transmission. Pages 258-261,2004.

[7] Yang, C.,Ling, Y .,Li, X ; Information encryption algorithm in power network communication security model; IOP Conference Series: Materials Science and Engineering,december 2019.

[8] Kambourakis, G., Rouskas, A., Gritzalis, S. ; Using SSL/TLS in authentication and key agreement procedures of future mobile networks (2002) 2002 4th International Workshop on Mobile and Wireless Communications Network, MWCN 2002, art. no. 1045713, pp. 152-156. doi: 10.1109/MWCN.2002.1045713.

[9] Acharya, Bibhudendra, et al; Image encryption using advanced hill cipher algorithm. International Journal of Recent Trends in Engineering 1.1 (2009): 663-667.

[10] Ali Mansouri1 · Xingyuan Wang1,2 ; Image encryption using shuffled Arnold map and multiple values manipulations; Springer-Verlag GmbH Germany, part of Springer Nature 2020

[11] Hofmann, G.R; The modelling of images for communication in multimedia environments and the evolution from the image signal to the image document; Vis. Comput. 9(6), 303–317 (1993).

[12] Lin, C.-H., Chao, M.-W., Liang, C.-Y., Lee, T.-Y; A novel semi-blind-and-semi-reversible robust watermarking scheme for 3D polygonal models; Vis. Comput. 26(6), 1101–1111 (2010).

[13] Tu, S.-C., Tai,W.-K Isenburg,M., Chang, C.-C ; An improved data hiding approach for polygon meshes; Vis. Comput. 26(9), 1177–1181 (2010).

[14] Li, G., Wang, L ; Double chaotic image encryption algorithm based on optimal sequence solution and fractional transform; Vis. Com- put. 35(9), 1267–1277 (2019).

[15]Tarik Taleb ; Yassine Hadjadj Aoul ; Abderrahim Benslimane; Integrating Security with QoS in Next Generation Networks 2010 IEEE Global

Telecommunications Conference GLOBECOM 2010, 6-10 Dec. 2010.

[16] Aiash, Mahdi, Mapp, Glenford E. and Lasebae, Aboubaker; Security and QoS integration for protecting service providers in hterogeneous environments; International Journal of Computer Science, 38 (4). pp. 384-393. ISSN 1819-656X,2011.

[17] (Bud) Bates, Regis J; Securing VOIP || Other protocols SRTP, ZRTP, and SIPS; 10.1016/B978-0-12-417039-1.00006-1;2015.

[18] Riccardo Bresciani and Andrew Butterfield; A formal security proof for the ZRTP Protocol; International Conference for Internet Technology and Secured Transactions, ICITST 2009 5402595

[19] Jonathan_Katz,_Yehuda_Lindell ; INTRODUCTION TO MODERN CRYPTOGRAPHY Second Edition 500; International Standard Book Number-13: 978-1-4665-7027-6,2015

[20] Bushra Anjum and Harry Perros ; Bandwidth Allocation for Video under Quality of Service Constraints ; Great Britain and the United States by ISTE Ltd and John Wiley & Sons, Inc;2015

[21]Seetha, S., Francis, S.A.J., Kanaga, E.G.M., Daniel, E., Durga, S. A framework for multi-constraint multicast routing in wireless mesh networks. In 2019 Fifth international conference on advanced computing & communication systems (ICACCS) (2019) IEEE, Mar, 2019, pp. 445-451.

[22] Atoum, Y., Liu, Y., Jourabloo, A., Liu, X. Face anti-spoofing using patch and depth-based CNNs (2018) IEEE International Joint Conference on Biometrics, IJCB 2017, 2018-January, pp. 319-328.

[23] F. Elazzaby, N. El Akkad and S. Kabbaj. A new encryption approach based on four squares and Zigzag. The 1st international conference on Embedded Systems and Artificial Intelligence, ESAI (2019).

[24] M. Es-sabry, N. El akkad , M. Merras, A.Saaidi and K.Satori. A New Color Image Encryption Using Random Numbers Generation And Linear Functions. The 1st international conference on Embedded Systems and Artificial Intelligence, ESAI (2019).

[25] Y, Qiao, Yu, F. Richard, Distributed Denial of Service Attacks in Software-Defined Networking with Cloud Computing, IEEE Communications Magazine 53(4), 52-59 (2015).

[26] R.V. Deshmukh, K.K. Devadkar; Understanding DDoS Attack & Its Effect in Cloud Environment; Procedia Comput. Sci 49(1), 202–210 (2015).

[27] M. Monika, Y. Singh, A Review; DoS and DDoS Attacks, International Journal of Computer Science and Mobile Computing 4(6), 260-265 (2015)

[28] Hamza TOUIL; Nabil EL AKKAD; Khalid SATORI: Text Encryption: Hybrid cryptographic method using Vigenere and Hill Ciphers: International Conference on Intelligent Systems and Computer Vision (2020).