© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 12 Issue: 03 | July - September 2021
Paper is available at http://www.jrps.in | Email : info@jrps.in

Study of Cloud authentication

Mohit Malik, M. Tech.
Ms. Shalini Bhadola, Kirti Bhatia, Assistant professor
Department of CSE, Sat Kabir Institute Of Technology And Management
Maharishi Dayanand University

Abstract

Security is one of the most pressing issues in the field of information engineering. It is critical to keep authoritative or client information safe. Moving from traditional technology to cloud engineering is pointless if the information of the organisation is not protected on the cloud. There are a number of non-profit organisations that are assisting and raising awareness about cloud computing security issues. One such organisation is the Cloud Security Alliance, which publishes a report every year on the most well-known security vulnerabilities in cloud computing. According to CSA research from 2013, there are eight well-known cloud security threats that might compromise a client's personal information without their knowing.

Key Words: Security, Organisation, Technology VOD etc.

Introduction

Real live video streaming has become one of the most popular Internet applications in the last decade. All researchers have had many successful commercial deployments using CDN or peer-to-peer engines. Formerly obtain high availability and low start-up latencies, but the expense of installing specific servers is exorbitant. This is especially severe when user requirements vary considerably and servers must be over-supplied with maximum loads. Peer-to-peer solution usually involves cheaper implementation costs and is more scalable, but it is not possible to ensure reliability or service quality. Efforts were also made to synergize dedicated servers with peer-to-peer. Unfortunately, the overall objective continues to be a scalable, dependable, responsive and cost-effective solution.

Demand video

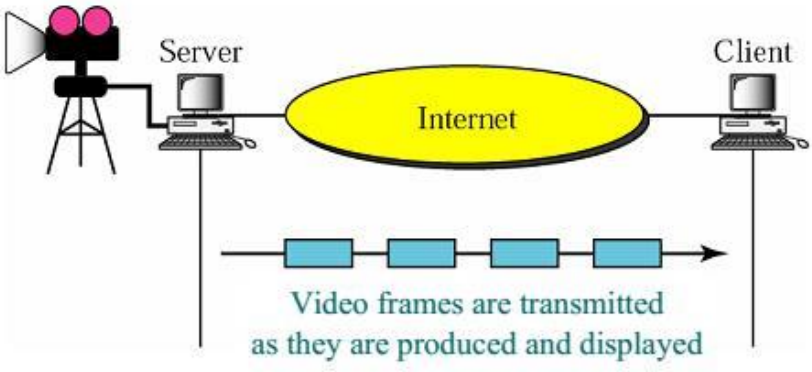
Video-on-demand (VoD) is now a popular internet service in the cloud. Usually the ISP provider nowadays charges the VOD to utilise the 95 percentile criterion for bandwidth, which works: Average server bandwidth is monitored every 5 minutes per month. These bandwidth measurements throughout a month create a set of values and a 95th percentile value is the lowest number in the set which is higher than 95% of the values. As the demand for VOD services changes over time within a day, however, providing servers with distinct property for a value of 95 percentiles should last a few hours a day leads to bandwidth failure at other times. In the case of PPLive , for example, the tariff cycle is less than 20%,

© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 12 Issue: 03 | July - September 2021
Paper is available at http://www.jrps.in | Email : info@jrps.in

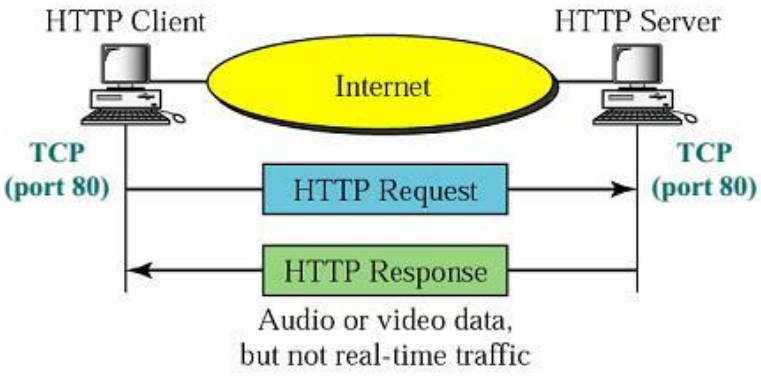
above 50%, with an average of 40%. Ninety-fifth percentile value is five times lower. Furthermore, a multiplicity of flashes is very costly even if flash mob may be anticipated.

Streaming of real time and non-real time

Video streaming is described as sending a video file via a network connection from the server to the client. It's two kinds: The one is real-time streaming, where live streaming is performed, i.e. a living event is given to customer while the other is non-real-time streaming, i.e. an archived lecture or movie is sent to customers on demand.



Real Time Video Streaming



Non-Real Time Video Streaming (Video- on Demand)

Cloud authentication

Research findings show that any online and cloud authentication method should offer high security, ease of use of interfaces and user mobility support. Customers like accessing their apps from various

places and devices including desktop, laptop, PDA, smartphones, mobile phones etc. These criteria pose major safety concerns for applications. A broad variety of user needs provides a broad range of cloud attack vectors that make cloud security a thought-provoking issue. Cloud service providers have to guarantee that their services are accessed by only genuine users, which indicates a robust system for user authentication. However, many attacks exist to establish authentication loopholes and thus to find the most secure and user-acceptable authentication method are a major problem in the cloud context. A thorough understanding of authenticity assaults and appropriate mitigation methods is therefore needed to design a foolish cloud environment authentication system.

High-level description of algorithm

1. **Key Expansions**—round keys are derived from cipher key using Rijndael's key schedule. AES requires a separate 128-bit round key block for each round plus one more.

2. **Initial Round**

- Add Round Key—each byte of state is combined with a block of round key using bitwise xor.

3. **Rounds**

- Sub Bytes—a non-linear substitution step where each byte is replaced with another according to a lookup table.
- Shift Rows—a transposition step where last three rows of state are shifted cyclically a certain number of steps.
- Mix Columns—a mixing operation which operates on columns of state, combining four bytes in each column.

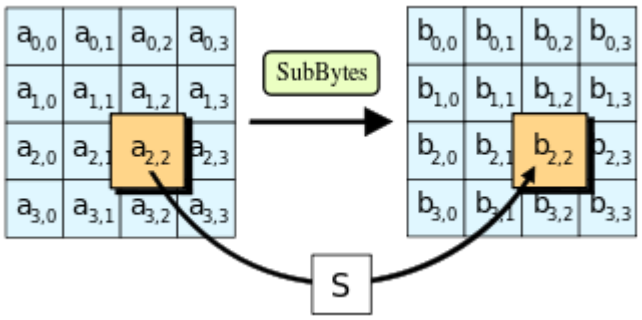
4. **Add Round Key**

5. **Final Round (no Mix Columns)**

- Sub Bytes
- Shift Rows
- Add Round Key.

© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 12 Issue: 03 | July - September 2021
Paper is available at http://www.jrps.in | Email : info@jrps.in

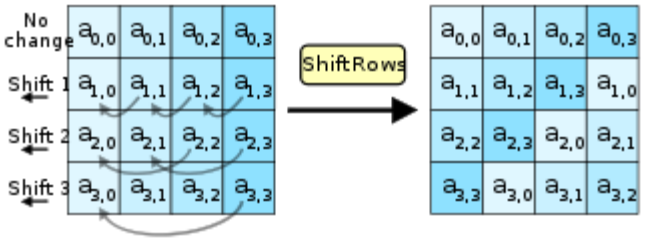
The Sub Bytes step



In Sub Bytes step, each byte in state is replaced with its entry in a fixed 8-bit lookup table, *S*; $b_{ij} = S(a_{ij})$.

In SubBytes step, each byte $a_{i,j}$ in *state* matrix is replaced with a SubByte $S(a_{i,j})$ using an 8-bit substitution box, Rijndael S-box. This operation provides non-linearity in cipher. S-box used is derived from multiplicative inverse over **GF**(2^8), known to have good non-linearity properties. To avoid attacks based on simple algebraic properties, S-box is constructed by combining inverse function with an invertible affine transformation. S-box is also chosen to avoid any fixed points (and so is a derangement), i.e., $S(a_{i,j}) \neq a_{i,j}$, and also any opposite fixed points, i.e., $S(a_{i,j}) \oplus a_{i,j} \neq \text{0xFF}$. While performing decryption, Inverse SubBytes step is used, which requires first taking affine transformation and then finding multiplicative inverse (just reversing steps used in SubBytes step).

The Shift Rows step



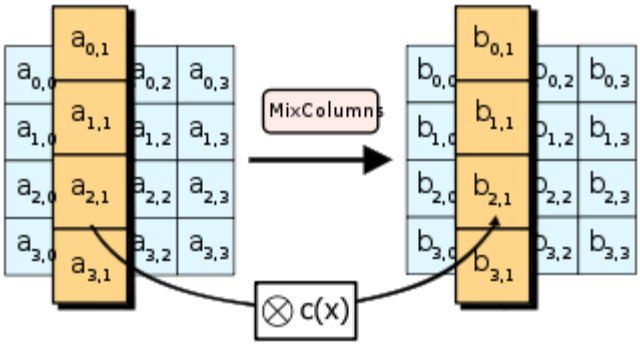
In Shift Rows step, bytes in each row of state are shifted cyclically to left. number of places each byte is shifted differs for each row.

The ShiftRows step operates on rows of state; it cyclically shifts bytes in each row by a certain offset. For AES, first row is left unchanged. Each byte of second row is shifted one to left. Similarly, third and fourth rows are shifted by offsets of two and three respectively. For blocks of sizes 128 bits and 192 bits, shifting pattern is same. Row n is shifted left circular by n-1 bytes. In this way, each column of output state of ShiftRows step is composed of bytes from each column of input state. (Rijndael variants with a larger block size have slightly different offsets). For a 256-bit block, first row is unchanged and

© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 12 Issue: 03 | July - September 2021
Paper is available at http://www.jrps.in | Email : info@jrps.in

shifting for second, third and fourth row is 1 byte, 3 bytes and 4 bytes respectively—this change only applies for Rijndael cipher when used with a 256-bit block, as AES does not use 256-bit blocks. Importance of this step is to avoid columns being linearly independent, in which case, AES degenerates into four independent block ciphers.

The Mix Columns step



In MixColumns step, each column of state is multiplied with a fixed polynomial *c(x)*.

In MixColumns step, four bytes of each column of state are combined using an invertible linear transformation. MixColumns function takes four bytes as input and outputs four bytes, where each input byte affects all four output bytes. Together with ShiftRows, MixColumns provides diffusion in cipher.

During this operation, each column is transformed using a fixed matrix (matrix multiplied by column gives new value of column in state):

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}$$

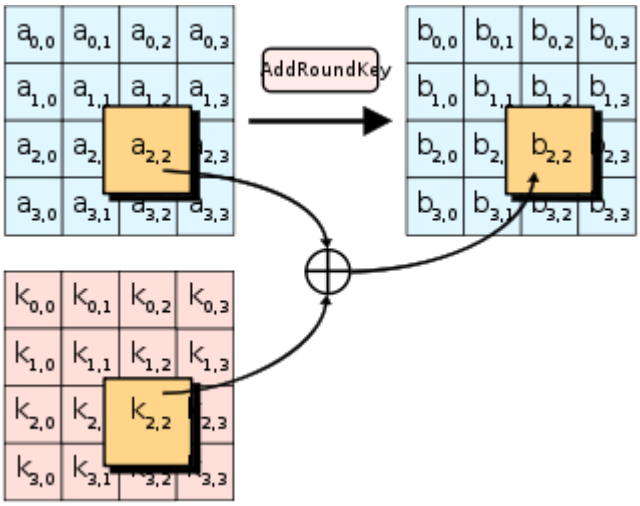
Matrix multiplication is composed of multiplication and addition of entries. Entries are 8 bit bytes treated as coefficients of polynomial of order x^7. Addition is simply XOR. Multiplication is modulo irreducible polynomial $x^8+x^4+x^3+x+1$. If processed bit by bit then after shifting a conditional XOR with 0x1B should be performed if shifted value is larger than 0xFF (overflow must be corrected by subtraction of generating polynomial). These are special cases of usual multiplication in **GF**(2^8).

In more general sense, each column is treated as a polynomial over **GF**(2^8) and is then multiplied modulo x^4+1 with a fixed polynomial $c(x) = 0x03 \cdot x^3 + x^2 + x + 0x02$. Coefficients are displayed in their hexadecimal equivalent of binary representation of bit polynomials from **GF**(2)[x]. MixColumns

© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 12 Issue: 03 | July - September 2021
Paper is available at http://www.jrps.in | Email : info@jrps.in

step can also be viewed as a multiplication by shown particular MDS matrix in finite field **GF**(2^8). This process is described further in article Rijndael mix columns.

The AddRoundKey step



In AddRoundKey step, each byte of state is combined with a byte of round subkey using XOR operation (⊕).

In AddRoundKey step, subkey is combined with state. For each round, a subkey is derived from main key using Rijndael's key schedule; each subkey is same size as state. subkey is added by combining each byte of state with corresponding byte of subkey using bitwise XOR."

Post-compression Encryption Algorithm

The Secure Real-time Transport Protocol, or naive approach encrypts com-pressed bitstream by packetizing multimedia data and individually encrypting every packet using AES. Although it is secure, it has huge computational overheads and it is not conducive to different desired properties of compressed bitstreams in general, owing to encryption of compressed data.

Many different algorithms have been proposed—which are format compliant, or have low computational requirements. Meyer and Gadegast proposed a selective video encryption scheme called Secure MPEG or SECMPEG for MPEG-1 video coding standard. It offers different levels of security by encoding different parts of compressed bitstream:

- Algorithm 1: It encrypts only headers from sequence layer to slice layer.

- Algorithm 2: It encrypts additionally low frequency DCT coefficients of all blocks in I-frames.

- Algorithm 3: It encrypts all I-frames and all I-blocks in P- and B-frames.

- Algorithm 4: It encrypts whole MPEG-1 sequence with naive algorithm."

"The approach has some notable limitations: computations savings are not significant because I-frames constitute 30–60 % of an MPEG video. Moreover Agi and Gong [**22agi96??**] demonstrated that some scene contents are still discernible by directly playing back selectively encrypted video stream on a conventional decoder. Maples and Spanos presented a similar approach called AEGIS".

Pre-compression Encryption Algorithm

Although it is possible to encrypt video content before compression it has some serious limitations which are crucial for mobile devices:

1. Pre-compression encryption implies encrypting raw or uncompressed bits which will waste lot of computational resources.

2. Encryption output is generally a random bitstream with lack of redundancy, making compression operation highly inefficient for general case. For example, consider encrypting a HD video at bare resolution of 480p (852 × 480) with AES. It would require 2.3 Million AES cycles per second to encode (and to decode) that video on a mobile device. Compression performance will be mostly lost as AES output bits will be nearly random with no possibility of lossless compression!

One known example is work of Pazarci and Diplin. scrambler, shown in above figure, is transparent to MPEG-2 compression. They encrypt video in RGB (red, green, blue) color space using four secret linear transformations before video coding. This scheme maintains compression efficiency of video codec but has been found unsafe against brute-force attacks"

Conclusion

Security for various multimedia applications such as video on-demand, video, video and media streaming is essential. A secured video transmission guarantees that the user does not get the information from the video while it is transmitted to the recipient, i.e. that only users who have paid for these services may see the videos and films. When the video is redundant the attacker may recreate the original video file simply. Data like text and computer code is less redundant than movies in their structure. All these characteristics make it more difficult to provide

security for an MPEG video. The security of the MPEG video transmission includes the encryption of portions of the MPEG bit stream or the whole bit stream.

References

1) Bergeron, C., Lamy-Bergot, C.: Complaint selective encryption for h.264/avc video streams. In: IEEE 7th Workshop on Multimedia Signal Processing, pp. 1–4 (2005). doi: 10.1109/ MMSP.2005.248641
2) Cheng, H., Li, X.: Partial encryption of compressed images and videos. IEEE Trans. Signal Process. 48(8), 2439–2451 (2000). doi: 10.1109/78.852023
3) Chiaraluce, F., Ciccarelli, L., Gambi, E., Pierleoni, P., Reginelli, M.: A new chaotic algorithm for video encryption. IEEE Trans. Consum. Electron. 48(4), 838–844 (2002)
4) Li, S., Zheng, X., Mou, X., Cai, Y.: Chaotic encryption scheme for real-time digital video. In: Real-Time Imaging VI. Proceedings of SPIE, vol. 4666, pp. 149–160 (2002)
5) Lian, S., Liu, Z., Ren, Z., Wang, H.: Secure advanced video coding based on selective encryp-tion algorithms. IEEE Trans. Consum. Electron. 52(2), 621–629 (2006)
6) Lian, S., Liu, Z., Ren, Z., Wang, H.: Commutative encryption and watermarking in video compression. IEEE Trans. Circuits Syst. Video Technol. 17(6), 774–778 (2007)
7) Logik Bomb: Hacker's Encyclopedia (1997)
8) Hafner, Katie; Markoff, John (1991). Cyberpunk: Outlaws and Hackers on the Computer Frontier. New York: Simon & Schuster. ISBN 0-671-68322-5.
9) Sterling, Bruce (1992). The Hacker Crackdown. Bantam. ISBN 0-553-08058-X.
10) Slatalla, Michelle; Joshua Quittner (1995). Masters of Deception: The Gang That Ruled Cyberspace. HarperCollins. ISBN 0-06-017030-1.
11) Dreyfus, Suelette (1997). Underground: Tales of Hacking, Madness and Obsession on the Electronic Frontier. Mandarin. ISBN 1-86330-595-5.
12) Verton, Dan (2002). The Hacker Diaries : Confessions of Teenage Hackers. McGraw-Hill Osborne Media. ISBN 0-07-222364-2.
13) Thomas, Douglas (2002). Hacker Culture. University of Minnesota Press. ISBN 0-8166-3345-2.