Study Security Cloud with SHA-2 Algorithm

Arnes Yuli Vandika[1], Tia Tanjung[2],

[1]Program Studi Informatika, Universitas Bandar Lampung
[2]Program Studi Sistem Informasi, Universitas Bandar Lampung

e-mail: arnes@ubl.ac.id[1], tia.tanjung@ubl.ac.id[2]

Abstrak

Studi ini mengeksplorasi hubungan antara cloud computing dan keamanan, berkonsentrasi pada penggunaan algoritma kriptografi SHA-2. Karena sifat pengaturan awan yang didistribusikan dan dibagi, komputasi awan telah mengubah cara informasi dan layanan diakses dan disediakan. Namun, itu juga menciptakan masalah keamanan baru. Sebagai metode kriptografi yang andal untuk menjamin integritas dan autentikasi data, Secure Hash Algorithm 2 (SHA-2) semakin populer. Studi ini bermaksud untuk menyelidiki bagaimana SHA-2 diintegrasikan ke dalam kerangka kerja keamanan cloud dan mengevaluasi seberapa baik SHA-2 melindungi data dan komunikasi dalam infrastruktur cloud. Studi ini bertujuan untuk menawarkan wawasan untuk meningkatkan keseluruhan postur keamanan sistem berbasis cloud dengan mempelajari penerapan SHA-2 di berbagai jenis layanan cloud dan situasi penerapan. Untuk mengevaluasi efek dunia nyata dari penggunaan algoritme SHA-2 untuk mengamankan lingkungan cloud, proyek ini menggabungkan analisis teoretis dengan eksperimen langsung. Dalam penelitian tersebut, overhead komputasional SHA-2 dalam operasi cloud dievaluasi sambil mempertimbangkan latensi dan penurunan kinerja. Makalah ini juga melihat kemungkinan kelemahan dan metode serangan yang dapat memanfaatkan kelemahan SHA-2 saat diterapkan dalam skenario cloud. Riset ini berkontribusi pada pengetahuan yang lebih besar tentang pertukaran antara keamanan dan kinerja dalam sistem cloud yang memanfaatkan teknik SHA-2 dengan mengatasi masalah ini. Hasil penelitian ini dapat menjadi referensi bagi arsitek cloud, pakar keamanan, dan peneliti saat memutuskan apakah akan menggunakan metode kriptografi yang kuat seperti SHA-2 untuk melindungi kerahasiaan dan integritas data dalam aplikasi berbasis cloud.

Kata kunci: SHA-2, Kriptografi, Cloud.

Abstract

This study explores the relationship between cloud computing and security, concentrating on the use of the SHA-2 cryptographic algorithm. Due to the distributed and shared nature of cloud settings, cloud computing has transformed how information and services are accessed and provided. However, it has also created new security issues. As a reliable cryptographic method for guaranteeing data integrity and authentication, the Secure Hash Algorithm 2 (SHA-2) has grown in popularity. This study intends to investigate how SHA-2 is integrated into cloud security frameworks and evaluate how well it protects data and communication inside cloud infrastructures. This study aims to offer insights into improving the overall security posture of cloud-based systems by studying the application of SHA-2 across various cloud service types and deployment situations.To evaluate the real-world effects of using the SHA-2 algorithm to secure cloud environments, this project combines theoretical analysis with hands-on experimentation. In the research, the computational overhead of SHA-2 in cloud operations is evaluated while taking into account latency and performance deterioration. The paper also looks into possible flaws and attack methods that could take advantage of SHA-2's shortcomings when applied in cloud scenarios. The research contributes to a greater knowledge of the trade-offs between security and performance in cloud systems utilizing the

SHA-2 technique by addressing these issues. The results of this study may serve as a reference for cloud architects, security experts, and researchers when deciding whether to use strong cryptographic methods like SHA-2 to protect the confidentiality and integrity of data in cloud-based applications.

Keywords: *SHA-2,* cryptography, cloud.

Secure Cloud and SHA-2

Many people have inquired as to what distinguishes SHA 2 from SHA 256. Part of SHA 2 is this 256-bit SHA. The United States government's development of the SHA 2 project led to the creation of roughly 4 SHA, namely SHA-224, SHA-384, SHA-512/224, and SHA-512/256, which are now included in the SHA 2 category. SHA-512/256, often known as SHA 256, is SHA 2 with a hash length of 256 bits. The most widely used version of SHA 2 at the moment is SHA 256 bits, which is required for acquiring an extremely secure A+ SSL configuration on SSL Qlabs. Because of this, SHA 2 and SHA 256 are identical. Part of SHA 2 is SHA 256. This SHA-2 hashing algorithm will be connected to the SSL root certificate's hashing algorithm.

The SHA-2 (Secure Hash Algorithm 2) algorithm and cloud computing are related because they both aim to maintain the security and integrity of data within cloud-based systems. Cloud computing offers a flexible and scalable approach to data processing, storage, and services, allowing businesses to efficiently manage their resources. However, because data is sent and kept on remote servers that can be accessed over the internet, this convenience also raises security issues. The SHA-2 algorithm is used in this situation.

A fixed-size hash value (digest) generated by the cryptographic hash function SHA-2 from an input (piece of data). This hash value acts as the input data's digital fingerprint. The SHA-2 algorithm is used in cloud computing to secure data integrity and authentication. Data can be SHA-2 hashed before being delivered or stored in the cloud. The hash value serves as a special identification for the data, ensuring that it was not altered during storage or transport. Data breaches, unauthorized access, and cyberattacks are possible threats to cloud infrastructure. In cloud security protocols, the use of SHA-2 algorithms offers a cryptographic layer that can prevent these dangers. By ensuring that various data inputs do not result in the same hash output, SHA-2's collision resistance attribute guards against unauthorized changes. Furthermore, because of the avalanche effect, even a small change in the input data causes a noticeable difference in the hash output, making it very difficult for malicious actors to corrupt the data undetected.

Purpose SHA-2 Testing

Furthermore, following the model provides a more thorough mathematical breakdown of the steps that make up the SHA-256 hash function. Remember that this is a condensed form that leaves out some of the implementation-specific details and optimizations.

Let's mathematically dissect the key parts of SHA-256:

Message Padding: Add a single "1" bit to the end of a message M of length L bits.

'0' bits should be added until the length reaches 448 modulo 512.

Add the 64-bit big-endian integer L, the initial message length, to the end.

Constants (K): The first 64 prime integers are divided into a set of 64 constant 32-bit words (K[0], K[1],..., K[63]) that make up the first 32 bits of the fractional parts of the cube roots.

Initial Hash Values (H):

The first eight prime numbers are used to generate the first eight 32-bit initial hash values (H[0], H[1],..., H[7]) for the SHA-256 algorithm.

Processing in Chunks:

The padded message should be split into 512-bit chunks (M[0], M[1], etc.).

Main Compression Loop:

Apply the following formula to each chunk M[i] to expand it into 64 32-bit words (W[0], W[1],.., W[63]):
W[t] equals M[t] when t is zero to fifteen.
For 16 t 63, W[t] = 1(W[t-2]) + W[t-7] + 0(W[t-15]) + W[t-16], where 0 and 1 are SHA-256-defined bitwise operations (rotations and XORs).
Update on Working Variable:
Set the initial hash values (H[0], H[1],.., H[7])
for the eight working variables (a, b, c, d, e, f, g, and h).
For t = 0 to 63, update the working variables in a loop:

T1 = h plus 1(e) plus Ch(e, f, g) plus K[t] plus W[t]
T2 = Maj(a, b, c) + 0(a)
In the equation h = g g = f f = e e = d + T1 d = c c = b
b = a a = T1 + T2

Final Hash

After processing each piece, add the working variables a, b, c, d, e, f, g, and h together to create the final hash value.

The choice function is denoted by "Ch," the majority function is denoted by "Maj," and the bitwise functions "0" and "1" are used in the preceding explanation. The algorithm's security depends on the particular bitwise operations and mathematical operations utilized, which include bitwise rotations, shifts, XORs, ANDs, and ORs.

This mathematical explanation gives a summary of the SHA-256 processes. To guarantee the security of the hash function, however, the precise implementation necessitates careful consideration of low-level bit manipulation, endianness, and other issues. I strongly advise reviewing the official specification and well-known cryptographic libraries if intend to utilize or implement SHA-256.

Purpose Flow Model

Starting with: | v User Data Input | v Client-Side Encryption (Key 1) | v Secure Transmission to Cloud | v Authentication and Authorization | v Data Storage in Encrypted Form | v Data Retrieval | v Data Decryption (Key 2) | v Data Preparation for SHA-2 | v Preprocessing (Padding, Length Appending) | v Message Schedule Generation | v SHA-2 Computation

RESULT

The way individuals and corporations manage and analyze data has been changed by cloud computing. Providing for the integrity and security of data is a crucial component of data management, which is frequently accomplished via cryptographic methods. This project benefits greatly from the Secure Hash Algorithm 2 (SHA-2) family's dependable and powerful hashing capabilities. SHA-2 hashing is a key tool for data verification, digital signatures, and authentication procedures in the context of cloud computing. The use of SHA-2 in cloud systems is explored in this abstract, along with its advantages and drawbacks.

Cryptographic services are provided by cloud service providers including Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP) as a component of their security offerings. Users can create SHA-2 hashes using these services to ensure the integrity of their data. Through APIs or administrative panels, users communicate with these services, entering data and choosing the desired SHA-2 variation, such as SHA-256 or SHA-512. The cloud service uses the selected SHA-2 variant to process the supplied data and generates a fixed-length hash value. This hash value serves as a cryptographic signature of the input data's integrity and serves as a one-of-a-kind representation of the data. These hashes can be used by organizations to validate the veracity of transferred files, set up safe authentication procedures, and guarantee the general reliability of their data.

When implementing cloud-based SHA-2 hashing, security concerns must come first. To protect cryptographic credentials, data transmission to and from the cloud service must be secure, and appropriate key management procedures must be followed. The generated hashes themselves should also be stored in a secure manner because they are essential for guaranteeing the validity of the data.

In conclusion, SHA-2 hashing is crucial for data security and integrity in cloud computing contexts. Utilizing cloud cryptographic services allows users to easily create SHA-2 hashes, improving their capacity to validate data accuracy and set up safe digital procedures. The effectiveness of SHA-2 in upholding the integrity and dependability of data hosted in the cloud is ensured by following best practices and remaining updated about the services provided by certain cloud service providers.

REFERENCES

Akshita Bhandari, Ashutosh Gupta, Debasis Das, 2016 6th International Conference - Cloud System and Big Data Engineering, Secure Algorithm for Cloud Computing and Its Applications

K. Prathapkumar1, Dr. A. Thirumurthi Raja, Nat. Volatiles & Essent. Oils, 2021; 8(5):9535-9541, DOUBLE SIGNATURE BASED CRYPTOGRAPHY USING DS-SHA256 IN CLOUD COMPUTING

Lefebvre, F., Czyz, J., & Macq, B. (2003, September). A robust soft hash algorithm for digital image signatures. In Proceedings 2003 International Conference on Image Processing (Cat. No. 03CH37429) (Vol. 2, pp. II-495). IEEE.

Steinberger, J. (2010, May). Stam's collision resistance conjecture. In Annual International Conference on the Theory and Applications of Cryptographic Techniques (pp. 597-615). Springer, Berlin, Heidelberg.

Contini, S., & Yin, YL (2006, December). Forgery and partial key-recovery attacks on HMAC and NMAC using hash collisions. In International Conference on the Theory and Application of Cryptology and Information Security (pp. 37-53). Springer, Berlin, Heidelberg.