eGov Products & Services

Cryptographic Key Management

Balancing Security and Usability in Key Management

Edited by **MOHAN DAS VISWAM**

The management of cryptographic keys is crucial for securing sensitive data and ensuring the integrity and confidentiality of communications in web applications. Cryptographic keys play a vital role in applications for various security functions, like encryption and digital signing. Proper management of these keys is essential to protect against data breaches, unauthorized access, and other cyber threats. This article evaluates three methods for the generation and storage of cryptographic keys:

- Generation and Storage of Private Keys within Application Server
- Software-Based HSM with a Dedicated Server and Isolated VLAN
- Hardware HSM

Background

Applications typically use both digital signature and encryption for data integrity and confidentiality. Traditionally data hashing was used for securing database records. Since a bad actor can easily tamper the data and replace the hashes, digital signature was introduced to generate and store signature of the data. The digital signature of an official is insisted to sign every critical transaction in the application. However internal processes within application could create large number of records which are practically impossible to sign using personal digital signature solutions. Another use case is the encryption or digital signing of data exchanged with other stakeholder applications through APIs as part of the business process.

The scalability problem with client based digital signatures could be resolved by using server based digital signatures for signing and encrypting of records at server side. This approach required generation and storing private keys within application servers or databases. This approach, while straightforward and cost-effective, poses significant security risks and trust issues. The following are some of the risks involved in generating and storing private keys within an application server:

The article explores three cryptographic key management approaches: storing keys in application servers, using Hardware Security Modules (HSMs), and implementing software-based HSMs on dedicated servers and isolated VLANs. It assesses each method's security, cost-effectiveness, and practicality. While storage in application/database server poses high security risks, hardware HSMs offer top-tier protection at a higher cost. Software-based HSMs strike a balance, providing enhanced security and scalability, making them an appealing option for secure and cost-effective key management.

- Application servers typically run in the demilitarized zone within the cloud or data centre. It is not advisable store private keys in this zone due to limited isolation policies.
- Private keys stored in application servers are vulnerable to various types of cyberattacks such as malware, unauthorized access, and insider threats.
- If an attacker gains access to the server, they can potentially compromise the private keys, leading to data breaches and loss of sensitive information.

Storing private keys in database is also common practice in private key management. However, this approach comes with several significant risks that could compromise the security of the cryptographic keys and the overall system.

- SQL Injection attacks on the database could reveal private keys
- Access to private key by the database administrator
- A data breach could also expose all the private keys
- Database backups that are not properly protected could be used to extract private keys

Hardware Security Module (HSM)

A Hardware Security Module (HSM) is a dedicated device designed to securely generate, store, and manage cryptographic keys. HSMs offer robust security features, including tamper resistance and strong access control mechanisms. The features and benefits of HSMs are given below:

Physical Security

- HSMs are housed in secure, tamper-resistant enclosures.
- They often include tamper-evident seals and self-destruct mechanisms that activate if unauthorized access is detected.

Secure Key Storage

- Keys are stored in a protected environment within the HSM and never leave the device in plain text.
- Cryptographic operations (e.g., signing, encryption) are performed within the HSM, minimizing the risk of key exposure.

Strong Access Controls

- HSMs implement multi-factor authentication (MFA) and role-based access controls (RBAC) to restrict access to cryptographic keys.
- Integration with identity and access management systems further enhances security.

Arun K Varghese
Scientist-D
arun.kv@nic.in

Table 13.1: Comparison of Cryptographic Key Management Methods

Feature	Application Server	Software-Based HSM	HSM
Security	Low	Moderate	High
Key Exposure Risk	High (keys Unprotected)	Moderate (keys protected)	Low (keys stored in dedicated Risk hardware)
Physical Security	None	None	Strong (tamper-resistant hardware)
Access Control	None	PIN based	Very Strong (MFA, RBAC, hardware tokens)
Compliance	Difficult to meet Standards	Easier to meet standards with secure implementation	Meets high security standards (FIPS, etc.)
Cost	None	Moderate (dedicated server/VM)	High (specialized hardware)
Deployment Complexity	Low	Moderate (requires dedicated server/VM)	High (requires specialized hardware setup)
Scalability	Limited (server resources)	High (virtualization/cloud)	Moderate (physical hardware limitations)

Compliance

- HSMs are often certified to meet stringent security standards such as FIPS 140-2/3, ensuring compliance with industry regulations.

Software-Based HSM

Software-based HSMs, also known as virtual HSMs, emulate the functionality of a hardware HSM within a software environment. A popular software based HSM is SoftHSM which is an implementation of a cryptographic store accessible through a PKCS#11 interface. SoftHSM was developed as a part of the OpenDNSSEC project. SoftHSM could be deployed on dedicated servers hosted in an isolated VLAN to enhance security. The features and benefits of Software-Based HSMs are given below:

Enhanced Security Compared to Application Servers

- Although not as secure as hardware HSMs, software-based HSMs provide a higher level of security than generating and storing keys within application servers.
- They utilize software encryption and access control mechanisms to protect private keys.

Cost-Effective

- Software-based HSMs are generally more affordable than hardware HSMs, making them accessible for organizations with budget constraints.
- They eliminate the need for specialized hardware, reducing capital expenditure.

Flexible Deployment

- Can be deployed on existing infrastructure, leveraging virtualization and cloud environments.
- Can be easily scaled to meet increase in usage requirements.

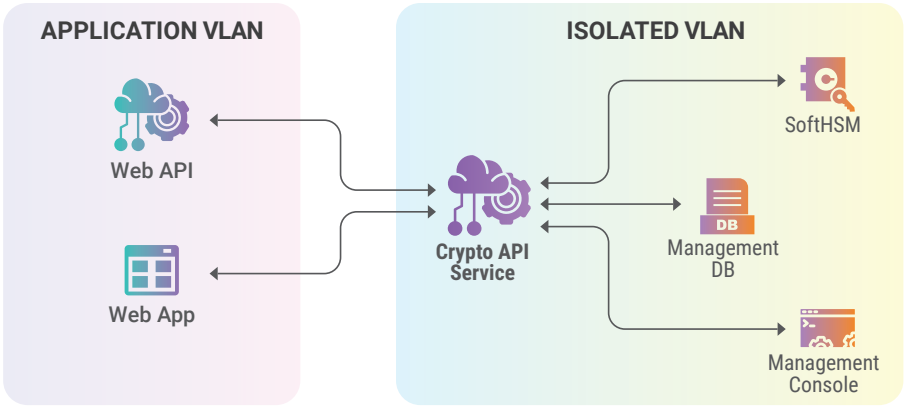
Access Controls

- Can be augmented to implement robust access controls, including MFA and RBAC.

Conclusion

The generation and storage of cryptographic keys within application/database servers poses significant security risks, including exposure to cyberattacks, lack of physical security, and compliance challenges. Hardware HSMs offer the highest level of security, providing robust physical protection, secure key storage, and strong access controls. They are compliant with stringent security standards, making them the preferred choice for organizations handling highly sensitive data. However, they come with higher hardware and licencing costs and deployment complexity. Software-based HSMs offer a balanced approach, providing enhanced security compared to application servers while being more cost-effective than hardware HSMs.

Fig 13.1 **SoftHSM based architecture**



Contact for more details

Manoj P A
Sr. Technical Director
NIC Kerala State Centre
CDAC Building, Vellayambalam
Thiruvananthapuram, Kerala - 695033
Email: manoj.pa@nic.in, Phone: 0471-2724529