Virtual and Distributed Hardware Security Module for Secure Key Management

Diogo Novo, Robin Vassantlal, Alysson Bessani, and Bernardo Ferreira

LASIGE, Faculdade de Ciências, Universidade de Lisboa, Portugal
dnovo@lasige.di.fc.ul.pt,
{rvassantlal,anbessani,blferreira}@ciencias.ulisboa.pt

Abstract. Hardware Security Modules (HSMs) play a crucial role in enterprise environments by safeguarding sensitive cryptographic keys and performing essential cryptographic operations. These devices, however, are expensive and difficult to manage, making them inaccessible to startups and small organizations. This work presents the development of a Virtual and Distributed HSM that can be practically deployed in real-world environments while providing robust security guarantees comparable to those of physical HSMs. Our approach leverages efficient protocols from the field of threshold cryptography, specifically distributed key generation, threshold signatures, and threshold symmetric encryption, which are the key operations performed by HSMs. By distributing trust among multiple parties and ensuring that no single entity has full control over cryptographic keys, our solution enhances security and resilience against breaches for a fraction of the cost of real HSMs. Additionally, we explore whether our system can support crypto-wallets for securely managing cryptocurrencies, such as Bitcoin and Ethereum. This will demonstrate the flexibility and applicability of our solution, namely in the growing field of digital finance, providing a secure alternative to manage digital assets. Experimental results reveal promising performance with low latency and acceptable scalability as server numbers increase, especially for Schnorr-based operations.

Keywords: Hardware Security Modules · Distributed Key Generation · Threshold Signatures · Threshold Symmetric Encryption

Context. As the business landscape continually evolves, the imperative to address cybersecurity risks becomes critical for organizations of all sizes. Despite substantial investments by large enterprises, smaller businesses often lack awareness of these threats or have not made protecting their information systems a top priority, leaving them vulnerable. The 2022 IBM Security report [10] reveals the consequence of these practices, a global average cost of data breaches reaching an all-time high of $4.35 million in 2022 (compared with $4.24 million in 2021).

Traditional security approaches involve the use of Hardware Security Modules (HSMs), physical devices that process cryptographic operations and safeguard cryptographic keys by hiding and protecting these cryptographic materials. These devices must always be relied upon, having internationally recognized certifications that vouch for their security guarantees [7].

Problem & Solution. While effective, HSMs are costly and often impractical for smaller companies [9]. This work proposes a virtual and distributed HSM solution, enabling smaller businesses to develop early-stage security strategies by employing a cheaper and more practical infrastructure without compromising on security. This is made possible because of the lower prices when comparing hosting a distributed service versus a service that needs to replicate their system through a set of physical and expensive Hardware Security Modules, in order to secure the availability of the infrastructure. Our approach is self-hostable and acts almost as a Software-as-a-Service (SaaS) solution, since it is ready to use and easily adaptable to the client's needs and environment, requiring less effort to put it into practice.

Related Work. Our virtual HSM addresses the limitations of existing attempts to virtualize HSMs, such as SoftHSM [12], pmHSM [5], and a hardware-backed Virtual HSM [13]. The first is used only for testing purposes since it offers no security guarantees, the second improves some properties that were lacking in the first, including security and availability, by adapting it to a distributed solution, and the third achieves a solution using both software and hardware, namely using Intel SGX [6], a hardware-based Trusted Execution Environment (TEE). These attempts do not meet the expectations we want to achieve with this work, since unlike previous solutions, our approach aggregates the required properties all in one, ensuring availability, integrity, and confidentiality without dependency on TEEs, but instead, we rely on a distributed system in order to achieve the same security levels of a physical device, allowing our HSM to be tolerant to asynchrony and faults/intrusions, which none of the referred attempts had in consideration. Specifically, we employ a Byzantine Fault-Tolerant State Machine Replication (BFT SMR) system, provided by BFT-SMaRt [2], since this is the state-of-the-art for implementing this type of system realistically and practically.

System Functionalities. Therefore, to achieve our goal, we studied state-of-the-art efficient protocols from the field of threshold cryptography, specifically for the operations of distributed key generation [11, 16], threshold signatures [8, 14, 3], and threshold symmetric encryption [1], since these are the distributed versions of the most important functionalities of an HSM, in addition to safe-guarding cryptographic keys. These threshold protocols correspond to cryptographic algorithms where multiple parties are needed to perform cryptographic operations, contrasting with relying on a single trusted device. This alternative requires that a certain threshold of devices be compromised for an adversary to be able to violate the system's security. Based on this study, we developed an efficient and robust Virtual and Distributed HSM to provide the security guarantees enumerated above.

Implementation. Our implementation is built on top of COBRA [16], a protocol stack for dynamic proactive secret sharing that allows implementing confidentiality in practical BFT SMR systems. This framework allowed us to protect

and secure the generated cryptographic keys, through its distributed polynomial generation protocol, and to adapt the same protocol to perform the distributed key generation algorithm for an arbitrary number of different elliptic curves. This protocol is based on (dynamic proactive) secret sharing, one of the branches of threshold cryptography. This scheme protects the confidentiality of a stored secret by splitting it into n shares, where a portion of these can later be combined to recover the initial secret, but combining fewer shares than required does not reveal any information about the secret. Besides having these characteristics, the employed scheme is also dynamic and proactive, allowing changes to the set of shareholders, providing share-renewal, and enabling verifiability to the shares, ensuring its integrity. The BFT SMR system and its features such as fault and asynchrony tolerance, crash recovery, and group reconfigurations are provided by BFT-SMaRt [2], upon which COBRA was built.

Regarding the implementation of the threshold signatures and the threshold symmetric encryption, both protocols are similar when taking a high-level view of the algorithms. A group of parties collectively compute a partial signature or a partial result without disclosing any information about the private key, and then a trusted entity, in our case the client, receives these results and aggregates them into the final signature or encryption/decryption.

The chosen signatures, and also key generation algorithms, were Schnorr [4] and BLS [3], since these are much easier to implement in a threshold manner than others, and are supported by two of the most significant blockchains, Bitcoin and Ethereum, respectively. For encryption and decryption, we chose the most recognized protocol, the Distributed Symmetric-key Encryption (DiSE) proposal, which consists of a generic construction of threshold authenticated encryption based on any distributed pseudorandom function (DPRF). The DPRF is the algorithm's most important component, and its responsibility is to generate partial results deterministically. Being deterministic is the key factor for making possible a later decryption of a ciphertext previously encrypted by this same protocol.

Another Use Case. Even though we have emphasized small businesses and the usage of an HSM, our system can also be used as a cryptocurrency wallet [15]. Although it is not integrated directly into a blockchain and does not have some basic features, such as viewing the account balance or the transactions made, it supports the most important ones, namely, the key generation, which is done when creating the account; the safe storage of the private key, by distributing its shares among the available servers; and also the signing of transactions, since it implements blockchain compatible algorithms.

Results. The results of the experimental evaluation in terms of the performance of our system show that the Schnorr implementation (both for key generation and signatures) is at least four times faster and scales better when compared with BLS and, regarding the encryption protocol, as the number of used replicas increases, its performance reduces by about half. Specifically, having 4 servers

and tolerating 1 faulty, the Schnorr algorithm got 48.76 and 46.43 operations per second (op/s) when generating keys and issuing signatures, respectively, while the BLS algorithm got 7.69 and 12.23 op/s. The encryption algorithm achieved, for the same configuration, 17.7 op/s. When scaling up the number of replicas, that is, for 7 replicas and at most 2 faulty, Schnorr operations reduced their performance by around 20%, while BLS and encryption/decryption operations reduced by around 50%.

Acknowledgments. This work was supported by FCT through project SMaRtChain, ref. 2022.08431.PTDC (https://doi.org/10.54499/2022.08431.PTDC), and the LASIGE Research Unit, ref. UIDB/00408/2020 (https://doi.org/10.54499/UIDB/00408/2020) and ref. UIDP/00408/2020 (https://doi.org/10.54499/UIDP/00408/2020).

References

1. Agrawal, S., Mohassel, P., Mukherjee, P., Rindal, P.: DiSE: Distributed Symmetric-Key Encryption. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. p. 1993–2010. CCS '18, Association for Computing Machinery, New York, NY, USA (2018), https://doi.org/10.1145/3243734.3243774
2. Bessani, A., Sousa, J., Alchieri, E.E.: State Machine Replication for the Masses with BFT-SMART. In: 2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks. pp. 355–362 (2014). https://doi.org/10.1109/DSN.2014.43
3. Boneh, D., Gorbunov, S., Wahby, R.S., Wee, H., Wood, C.A., Zhang, Z.: BLS Signatures. Internet-draft, Internet Engineering Task Force (Jun 2022), https://www.ietf.org/archive/id/draft-irtf-cfrg-bls-signature-05.html
4. Brandão, L., Davidson, M.: Notes on threshold EdDSA/Schnorr signatures. https://nvlpubs.nist.gov/nistpubs/ir/2022/NIST.IR.8214B.ipd.pdf (2022)
5. Cifuentes, F., Hevia, A., Montoto, F., Barros, T., Ramiro, V., Bustos-Jiménez, J.: Poor Man's Hardware Security Module (pmHSM): A Threshold Cryptographic Backend for DNSSEC. In: Proceedings of the 9th Latin America Networking Conference. p. 59–64. LANC '16, Association for Computing Machinery, New York, NY, USA (2016). https://doi.org/10.1145/2998373.2998452
6. Costan, V., Devadas, S.: Intel SGX Explained. Cryptology ePrint Archive, Paper 2016/086 (2016), https://eprint.iacr.org/2016/086
7. Encryption-Consulting: https://www.encryptionconsulting.com/education-center/what-is-an-hsm/ (What is an HSM?), [Online - Accessed on 2024-06-14]
8. Gennaro, R., Goldfeder, S.: Fast Multiparty Threshold ECDSA with Fast Trustless Setup. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. p. 1179–1194. CCS '18, Association for Computing Machinery, New York, NY, USA (2018). https://doi.org/10.1145/3243734.3243859
9. Hagerstrand, B.: HSM Security with Cloud Like Economics. https://www.fortanix.com/blog/hsm-security-with-cloud-like-economics, [Online - Accessed on 2024-06-18]
10. Institute, P.: https://www.csoonline.com/article/573315/average-cost-of-data-breaches-hits-record-high-of-435-million-ibm.html (2022 IBM Security Report), [Online - Accessed on 2024-06-14]

11. Kate, A., Huang, Y., Goldberg, I.: Distributed Key Generation in the Wild. Cryptology ePrint Archive, Paper 2012/377 (2012), https://eprint.iacr.org/2012/377
12. OpenDNSSEC: SoftHSM. Version: 2.6.1. https://www.opendnssec.org/softhsm/ (2020), [Online - Accessed on 2024-06-14]
13. Rosa, M.: Virtual HSM: Building a Hardware-backed Dependable Cryptographic Store. Master's thesis, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa (2019)
14. Ruffing, T., Ronge, V., Jin, E., Schneider-Bensch, J., Schröder, D.: ROAST: Robust Asynchronous Schnorr Threshold Signatures. In: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security. p. 2551–2564. CCS '22, Association for Computing Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3548606.3560583
15. Suratkar, S., Shirole, M., Bhirud, S.: Cryptocurrency wallet: A review. In: 2020 4th International Conference on Computer, Communication and Signal Processing (ICCCSP). pp. 1–7 (2020). https://doi.org/10.1109/ICCCSP49186.2020.9315193
16. Vassantlal, R., Alchieri, E., Ferreira, B., Bessani, A.: COBRA: Dynamic Proactive Secret Sharing for Confidential BFT Services. In: 2022 IEEE Symposium on Security and Privacy (SP). pp. 1335–1353. IEEE Computer Society, San Francisco, CA, USA (2022). https://doi.org/10.1109/SP46214.2022.9833658