CryptoVault - A Secure Hardware Wallet for Decentralized Key Management

***Abstract*—In traditional centralized internet services, third parties authenticate the transactions of the users. An important property of decentralized blockchain networks is the unrestricted and secured access to the private keys of users, which may often be threatened for several reasons. One considerable problem in systems based on blockchain technology is when users lose access to their keys due to e.g. a broken or lost device. This paper, firstly, introduces an implementation that generates and maintains the private key in an Intel SGX enclave. The implementation allows using the private key in a process isolated from all other processes running on the same system. Secondly, the paper provides a method that enables the secure storage and recovery of a back-up key to and from an external repository, using an end-to-end secure connection. One proposed application, with which this technology could be exploited, is the social wallet.**

I. Introduction

Internet transactions have changed the way people do their daily business. For example, purchases are often made using a mobile payment application or a debit/credit card instead of cash. Internet has also introduced tendering and comparison services that make complex topics such as household insurance or electricity contracts easier to purchase while also providing support to the making of insurance claims or monitoring your electricity consumption via an online form or a mobile application. Finally, third party authentication services guarantee the validation of your identity so that you can securely authenticate yourself to the service provider, thus generating a trust relationship right from the beginning. Although major changes in the aforementioned elements and routines have occurred, one thing has remained the same: almost every transaction is authenticated by some trusted notary, who keeps record of each of the transactions of participants within that industry. Blockchain technology has introduced a way to publish timestamped transactions without trusted parties and it can change the current centralized trusted-third-party policy.

While blockchain technology is decentralized, key management has been left as a sole responsibility of the user. This is different from centralized approaches that have some kind of administrator, IT support or similar that can reset the password or create new credentials in case you lose them. Now that this support is missing, the situation has lead to the loss of assets in case the users' keys are lost due to broken equipment, lost password or some other mishap. In many cases, the users outsource the management of their keys to a *wallet application* that can be either software, hardware or a combination of these. The New York Times published a recent article claiming losses as large as $220 million due to the forgotten password of an encrypted hard drive that contains the private keys to a Bitcoin digital wallet [1]. According to Chainanalysis, around 20% of all Bitcoin which is roughly $210 billion have not been moved during the last five years or longer and are therefore considered lost [2].

This paper presents a hardware wallet that securely stores and manages the sensitive keys of the user, and a reliable method for backing up these keys as independent shares stored in multiple locations. The research answers the question, how to design a novel hardware wallet with more appropriate backup method compared to the currently used solutions. The paper is structured as follows. We begin by introducing the background, starting from blockchain transactions and wallet applications to the possible risks and threats targeted against them as well as the risk mitigation methods. Then we describe the CryptoVault concept and how it provides a solution to the challenge, providing short insights also to the security assumptions. Finally we discuss about the potential threats to the concept in addition to introducing the social wallet concept that might be an interesting application for CryptoVault before concluding remarks.

II. Background

Blockchain technology is based on the principle that any accepted transaction is seen and verified by each of the nodes participating in the network. The transaction is an atomic data structure in a block, and any new block is linked to the previous one. These blocks together form a blockchain. The miner or sealer is an actor that combines these transactions and adds a new block to the blockchain network. The consensus protocol determines which blocks define the current state of this common database.

A. Blockchain transaction

The transaction contains the information to be decentralized and secured with the blockchain. For example, in Bitcoin, the information consists of value transfers between different users identified by addresses. Each transaction contains a nonce value to prevent double-spend attacks, and a signature. The signature consists of a transaction hash signed with a private key that verifies the origin of the transaction. The elliptic curve digital signature algorithm and the private key are used to calculate a signature for all blockchain transactions. The blockchain protocol, agreed by the majority of nodes, defines the transactions that are considered as a valid ones and can be used in valid blocks.

B. Wallet Applications

Blockchain transactions typically contain information on the receiver, value, fee, address specific signature and nonce. Users interact with the blockchain using a *wallet* application. Wallets are used to store the private keys, generate signatures and encode the transactions on behalf of the user. There are various different wallet implementations making use of different technologies. A good survey on this topic can be found in [3]. Below we present some of the most important features of wallets that are relevant to this paper.

As in [3], we can divide wallets into a few categories. The first are online (web) wallets, which can be used through any web browser. These offer great convenience and ease of use, but require an internet connection, a fairly secure environment (browser, operating systems etc.) and the security of the user's keys is many times based on trust to the service provider. The second type of wallet is a mobile wallet, which is an application on a mobile device. The convenience of mobile wallets is also good and they enable better possibilities for the user to retain control of their keys (e.g. on their device). On the other hand, the overall security of the mobile device becomes critical for these types of wallets.

The third type of solution is a desktop wallet, which operates on a personal computer. Here the convenience is somewhat diminished and not all wallets are available for different operating systems. Again, the security of the desktop environment becomes critical, but the hosting of users keys is much more likely to be local than hosted by the service provider or a third party.

Finally, there are hardware wallets, which are dedicated devices that host and/or operate the user's keys on their behalf. The convenience of hardware wallets is not that good, but the security is better as the user is in control of the keys. The interested reader can check the evaluations in [3] for details on specific implementations.

C. Possible risks

As mentioned above, there are many risks concerning the security of the private keys that are used in blockchain transactions. These risks are related to the possible hacking of the device and/or software that is used in the device. For example, if passwords are used to protect the keys, guessing and keylogging attacks can be used to recover these passwords. After that, the attacker can use the passwords to gain access to the user's private keys.

In addition, it is possible that there is poor randomness in the generation of the keys [4] or flaws in the hardware/software used to generate the keys [5]. These types of issues in cryptography and the repercussions have been reported in e.g. [6]. An attacker could use this information to gain access to the user's assets (by finding out the private key).

One issue, that has been traditionally poorly dealt with the different wallet applications, is the loss of keys or destruction of the device containing the keys. These types of problems either render the user unable to access their cryptocurrencies, or require the user to place trust into some service provider (e.g. wallet operator) to back up their keys and to have a secure protocol in place to recover the lost keys.

The aim of our CryptoVault solution is to enable the user to backup their keys securely and privately without the need to trust third parties (except in possibly storing some encrypted information). The users of CryptoVault should also be free from the risks related to the lack of entropy, unauthorized access to random access memory or data storage, real-time side-channel attacks, phishing, denial of service and a single point of failure.

D. Cryptography

RSA is a widely used public key cryptosystem created and published by Rivest, Shamir and Adleman [7]. There are fours steps in the RSA system: key generation, key distribution, encryption and decryption. The key generation step produces two keys, public and private. During key distribution, the user's public key is delivered in some reliable but not necessarily secret way to the other communicating party - meanwhile the private key stays secret with the user. Then the other party can use the public key for encryption and the user can decrypt with their private key. Besides ensuring the secrecy of messages, the RSA scheme can also be used to sign messages. In such a case the private key of the sender is used as a signature and the public key is used to verify it.

Shamir's Secret Sharing (SSS) [8] is a scheme that can be used to split a secret into parts, or shares, that can then be distributed between the participants of the scheme. The shares are recombined whenever the original secret is needed. In the simplest case all the shares are needed for reconstructing the original secret, but there are modifications in which a number of shares above a threshold value are needed for reconstruction. SSS can be used to e.g. secure and distribute keys.

E. Secure Hardware

In traditional systems the operating system protects applications and resources such as memory and processor from malicious events. The purpose of secure hardware is to isolate applications and resources to protect sensitive information from malfunction. Trusted Execution Environments (TEE) can run sensitive code with higher security level the operating systems. Intel Software Guard Extensions (SGX) [9] is a TEE that isolates the running code and data from the untrusted environment.

III. CryptoVault

Our CryptoVault concept and implementation aims at combining the best features of different wallet types, while minimizing the related risks in these wallets. The main secure functionalities, that are achieved within CryptoVault are generating keys with high entropy, end-to-end protected key backup, signing transactions inside a trusted execution environment and running key recovery without a single point of failure.

The implementation utilizes three main security technologies. First of all, secret keys are split into shares for secure

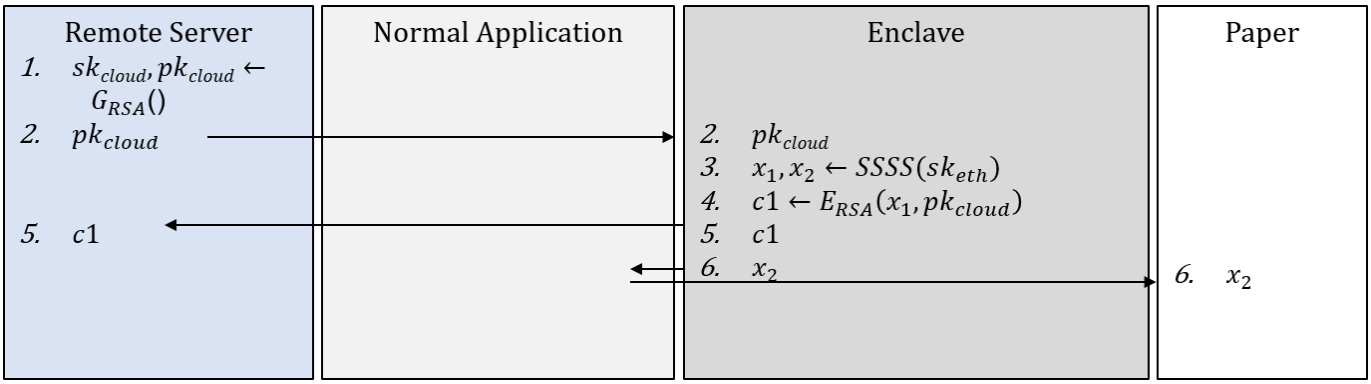

Fig. 1. Secret key backup procedure

backup. The shares are stored in paper (i.e. printed out by the user) and remote servers using the SSS Scheme. Second, processes and storage related to the keys are isolated from the operating system by using the Intel SGX environment. Third, RSA is used to provide an end-to-end encrypted information channel between the SGX-enclave and remote servers when distributing key shares.

Our implementation secures Ethereum keys. Ethereum uses the secp256k1 [10] elliptic curve for digital signatures. The secret key is a randomly selected positive integer below constant `secp256k1n - 1`. The corresponding public key is calculated by elliptic curve multiplication with secp256k1 generator point. Finally, the Ethereum address can be derived by calculating the keccak256-hash from the formatted public key. The secret key, public key and address are generated inside enclave, and the address is stored in plain text into the file system.

When the address is calculated, the next step is to make a backup of the secret key. Figure 1 presents how a secret key can be divided into 2 shares which are stored on a remote server and on paper backup (printed out by the user). In a real-world use case, more than one remote-server shares are recommended. The user can then also choose a threshold number of shares that are needed to reconstruct the original key (e.g. 2 out of 3 shares).

Below are the steps of the backup algorithm.

1) Remote server generates RSA-2048 keypair
2) Public key of remote server is transferred into Enclave
3) Shamir Secret Sharing is used to divide Ethereum secret key into 2 shares
4) Share is encrypted using Public key of remote server
5) Encrypted share is transferred into remote server.
6) Finally second share is transferred into normal application and can printed into paper as a plain text.

Our assumption is that the required minimum amount of shares is always available for recovery and the number of malicious parties is smaller than the required minimum amount of shares. This ensures that the secret cannot be recovered outside of the system. For example, to ensure trustworthiness, the server used to back up users keys may run an instance of CryptoVault.

Figure 2 presents the recovery of a secret key. The recovery process is performed in reverse order. Using the RSA-keypair of the local enclave to encrypt and decrypt the remote-share, and finally the Ethereum secret key is reconstructed.

The same cryptographic functions are used in both local and remote instances, meaning that the same application and enclave can be used in remote servers as in local computers.

A. Security Assumptions

The security of the CryptoVault system is based on the following set of assumptions

1) Intel SGX environment is considered to be safe
2) Shamir's Secret Sharing Scheme implementation is considered to be safe
3) RSA-2048 implementation is considered to be safe

Many different practical attack vectors for Intel SGX have been found over the last few years. Latest known attacks have been categorized in [11]. Some of the attacks are more theoretical and are not feasible for all targets, but it is also important to consider that all possible attacks might not have been discovered yet.

IV. Discussion

Although there are some attacks against the Intel SGX system, it should be noted that the concept of CryptoVault can be used with any other (future) TEE that is available. Furthermore, it is important to note that the number of security assumptions and required trusted parties are very few. This is in itself already a great benefit as the potential risks can be more easily understood by and communicated to the users. This should lead into far fewer lost keys for the users.

A. Threat analysis

CryptoVault provides a fairly secure environment, but there are still some possible threats that should be considered. If an attacker gains physical access to the user's device, but cannot access the operating system, he will not be able to steal the key from the device. If the device is stolen, the user cannot use his account until he recovers the keys from the backup storage.

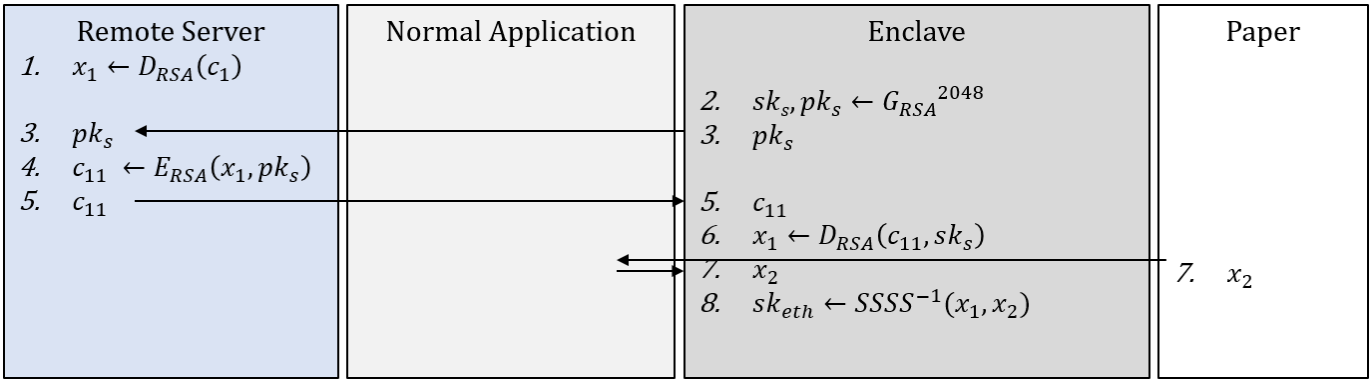

Fig. 2. Sercet key recovery procedure

In case the attacker gains access to the operating system, it may result in exploiting of the trusted software. If the attacker is able to read the memory space, in which the printable share of the private key is temporarily located, he naturally gains part of the secret. However, in case the attacker gains access rights to initiate trusted environment, he is able to make a backup copy of the key, which eventually leads to obtaining the key. If the attacker can alter the communication between CryptoVault and remote server, he is able to conduct an active man-in-the-middle attack and by passing own encryption key to enclave or the remote server attacker is able to gain access to secret share.

B. Social Wallet

Considering the functionality of our CryptoVault solution, there is an interesting new concept of *social wallets* [12], [13], that also takes into account the need for users' wallet backups. Social wallets are an interesting potential application for CryptoVault and a point of recent discussion for users to backup their wallets. Social wallets offer users the possibility to distribute shares of their keys within their social circle(s). Then some part (as in SSS) of these shares could be combined to recover the lost keys. CryptoVault could be used to generate and store these shares either in digital form on the users' devices (TPMs even) or in physical form as in printed QR codes.

V. Conclusion

In this paper, we have outlined the risks related to the current blockchain key management environments and introduced an implementation of the distributed key backup, based on Shamir's Secret Sharing. In the implementation, secure hardware (Intel SGX) was used to avoid the various risks that are present in modern computational environments. The most fundamental problem addressed by this CryptoVault solution is key back-ups and key recovery in the case that a device and the keys within it are lost or broken. Our solution can also be used as a part of future key backup mechanisms, such as social wallets.

References

[1] N. Popper, "Lost passwords lock millionaires out of their bitcoin fortunes," The New York Times, 2021, https://www.nytimes.com/2021/01/12/technology/bitcoin-passwords-wallets-fortunes.html.

[2] C. Team, "60% of bitcoin is held long term as digital gold. what about the rest?" Chainanalysis Blog, 2020, https://blog.chainalysis.com/reports/bitcoin-market-data-exchanges-trading.

[3] S. Suratkar, M. Shirole, and S. Bhirud, "Cryptocurrency wallet: A review," in *2020 4th International Conference on Computer, Communication and Signal Processing (ICCCSP)*, 2020, pp. 1–7.

[4] N. Heninger, Z. Durumeric, E. Wustrow, and J. A. Halderman, "Mining your ps and qs: Detection of widespread weak keys in network devices," in *21st {USENIX} Security Symposium ({USENIX} Security 12)*, 2012, pp. 205–220.

[5] M. Nemec, M. Sys, P. Svenda, D. Klinec, and V. Matyas, "The return of coppersmith's attack: Practical factorization of widely used rsa moduli," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 16311648. [Online]. Available: https://doi.org/10.1145/3133956.3133969

[6] A. Parsovs, "Estonian electronic identity card: security flaws in key management," in *29th {USENIX} Security Symposium ({USENIX} Security 20)*, 2020, pp. 1785–1802.

[7] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[8] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, p. 612613, Nov. 1979. [Online]. Available: https://doi.org/10.1145/359168.359176

[9] "Intel software guard extensions," https://software.intel.com/sgx-sdk.

[10] S. SEC, "2: Recommended elliptic curve domain parameters," *Standards for Efficient Cryptography Group, Certicom Corp*, 2000.

[11] A. Nilsson, P. N. Bideh, and J. Brorsson, "A survey of published attacks on intel sgx," 2020.

[12] V. Buterin, "Why we need wide adoption of social recovery wallets," https://vitalik.ca/general/2021/01/11/recovery.html, 2021.

[13] S. He, Q. Wu, X. Luo, Z. Liang, D. Li, H. Feng, H. Zheng, and Y. Li, "A social-network-based cryptocurrency wallet-management scheme," *IEEE Access*, vol. 6, pp. 7654–7663, 2018.