Research on the Design and Development of a Secure Cold Wallet Device for Blockchain

DOI: https://doi.org/10.54654/isj.v2i25.1140

Vu Ta Cuong*, Nguyen Thanh Trung

***Abstract*— Cold Wallet is a specialized hardware device designed to store private keys and perform digital signing in an isolated (offline) environment, ensuring maximum security for digital assets. This paper presents a comprehensive design of a secure cold wallet device for Blockchain, focusing on security requirements and cryptographic processing capabilities.**

***Tóm tắt*— Ví lạnh (Cold Wallet) là thiết bị phần cứng chuyên biệt nhằm lưu trữ khóa riêng và thực hiện ký số trong môi trường cách ly mạng, đảm bảo an toàn tối đa cho tài sản số. Bài báo này trình bày một thiết kế tổng thể thiết bị ví lạnh an toàn cho Blockchain, tập trung vào các yêu cầu bảo mật, khả năng xử lý mật mã.**

Keywords*— *Cold wallet, blockchain, transaction signature.

Từ khóa*— *Ví lạnh, chuỗi khối, ký giao dịch.

I. INTRODUCTION

The rapid development of blockchain technology and digital assets such as Bitcoin and Ethereum has led to an increasing demand for private key protection. In blockchain systems, private keys play a key role in authenticating transactions and determining ownership of digital assets. If this key is exposed, users may lose all their assets and cannot recover them. In the context of increasingly sophisticated cyber attacks, malware and online fraud, securely storing private keys has become a major challenge.

A cold wallet is one of the most effective most effective security solutions, operating completely separate from the Internet to avoid remote threats. Unlike hot wallets, cold wallets store private keys offline, only connecting when needed to sign transactions. However, many commercial cold wallets today still have risks such as lack of software transparency, potential hardware tampering, and are not suitable for security requirements in the domestic environment, especially in the field of defense research, government, or large enterprises.

Therefore, designing and manufacturing a secure cold wallet device, fully mastering both software and hardware, having an active security mechanism with low cost and easy integration is a necessary research direction, especially in the following cases:

- The need for digital asset security is increasing among both individual and institutional users;

- Vietnam is promoting policies to develop digital financial technology and a safe digital ecosystem;

- The need to master the entire source code, ensure localization, and maintain independence, suitable for research, training and deployment in specific environments.

This article presents the design and fabrication of a dedicated cold wallet device, focusing on the following elements:

- Complete control over hardware design, firmware and software for the device;

- Secure firmware design with local private key management;

- Protection mechanism against unauthorized extraction.

This manuscript was received on September 15, 2025. It was reviewed on September 23, 2025, revised on October 10, 2025 and accepted on October 29, 2025.

* Corresponding author

Thereby, the research aims at a customizable cold wallet platform, serving both research, education and practical deployment purposes at domestic organizations.

II. Overview of cold wallet technology and existing solutions

A cold wallet is a device or method of storing private keys offline, completely isolated from the Internet for most of its use. The main goal of a cold wallet is to prevent remote attacks, especially:

- Malware on your computer or phone;

- Email/phishing scams;

- Attack via Wi-Fi, USB or browser.

General operating principle of cold wallet:

- Private keys are generated and stored directly in the cold wallets, never leaked to the outside.

- When a transaction needs to be made, the cold wallet receives unsigned transaction data from the computer via a connection port (USB, QR code, Bluetooth...).

- The device signs the transaction with its internal private key, without revealing the key, and then sends back the signed data.

- The computer or network device will broadcast the signed transaction to the blockchain.

This approach ensures that the private key never appears on a networked system, significantly reducing the risk of remote theft.

A cold wallet device typically includes:

- Microcontroller (MCU): Controls all operations, processes data, and manages keys.

- Physical control button: Ensures that the confirmation action cannot be simulated remotely.

- Flash Memory: Stores private keys, seeds, system configuration.

- Connectivity (USB, NFC, Bluetooth): For communication with host devices.

- Security Mechanism: Protects private keys from unauthorized reading or physical extraction.

Around the world, manufacturers have also invested in research and launched a number of products such as:

- Ledger Nano S/X [1]: uses STM32 MCU with Secure Element ST31, supports many types of coins, relatively closed software;

- Trezor (Model T, One) [2]: open source, STM32 MCU, no Secure Element; physically exploitable.

- Coldcard [3]: emphasizes physical intrusion resistance, supports communication via SD card, open-source design.

- Keystone [4], Passport [5]: use camera instead of USB port to transfer data using QR code to limit connection attacks.

Most of these devices are closed source, making it difficult to verify the security of both software and hardware. Some devices have even been found to have information leaks or software vulnerabilities [6, 7]. In addition, research [8-11] also points out security vulnerabilities related to cold wallets.

Through a survey of related products and research projects, it can be seen that the above products often find it difficult to meet the requirements of technological autonomy and development of applied technology according to the orientation and characteristics of Vietnam. This is because the above products still have some limitations:

- Dependence on foreign hardware with unverifiable backdoors;

- Not understanding security solutions, potential risk of being attacked;

- Difficult to customize, does not support specific cryptographic algorithms, domestic standards.

Based on the above analysis, the research in this paper aims to design and manufacture a cold wallet with the following characteristics:

- Completely autonomous from hardware design, firmware, software, can be verified, modified and integrated as required.

- Implement security mechanisms to securely protect keys and cryptographic parameters.

III. HARDWARE ARCHITECTURE, FIRMWARE DESIGN AND KEY SECURITY SOLUTIONS

A. Hardware architecture

The cold wallet device is designed with the goal of: absolute protection of internal private keys, secure communication with the external environment, and protection against reading keys and cryptographic parameters out. The overall architecture of the system is shown in Figure 1.

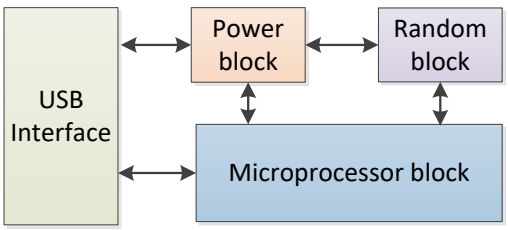

Figure 1. Block diagram of cold wallet device

In which, the main components in architecture are:

- Main MCU: expected to use STM32 chip, responsible for controlling all operations, processing transactions, signing, and memory management.

- Random number generator: uses physical random number generator circuit.

- USB HID Interface: Connects to the computer as a secure peripheral (no drive emulation to reduce the risk of attacks via file protocols).

B. Firmware design and key security solutions

1. General design solution

The firmware of a cold wallet device is a key component that determines the security of the entire system. The main goal is to ensure that the private key is never leaked under any conditions, including firmware failure or unauthorized physical access to the device.

The firmware is developed following the following safety principles:

- The private key is encrypted with the Data Encryption Key (DEK) before being saved to device memory.

- The private key only exists in RAM when needed, and is erased from memory immediately after signing.

- Never transmit your private key or Data Encryption Key (DEK) out, under any protocol or mode.

- Use user PIN authentication before performing signing.

- Separate communication function and encryption/signing function to avoid logic errors.

- Protect critical memory areas with the microcontroller's hardware configuration.

- All random values needed during device operation are taken from the true random number generator of the STM32 chip (The true random number generator has beentested using German BSI statistical tests of AIS-31 (T0 to T8), and NIST SP800-90B statistical test suite [12]).

2. Key security solution

To ensure the safety of the device, all private keys are encrypted before being stored in the device's memory. In addition, after the device is loaded with firmware, the STM32 chip's readout protection will be enabled to prevent data from being read back.

The key is encrypted/decrypted using the DEK (Data Encryption Key) master key, which is generated using the chip's true random number generator when the device is initialized. The DEK is encrypted using a key derived (using PBKDF2 with 10000 rounds [13]) from the user PIN (or administrator PIN) and a random SALT value (20 bytes). Thus, the device will store 02 encrypted copies of the Data Encryption Key (DEK): 01 copy for the user, 01 copy for the administrator. In addition, to ensure the integrity of the Data Encryption Key (DEK), we use HMAC-SHA256 [14].

The device is designed with 01 user and 01 administrator with different permissions. The user will have the right to view key information and perform digital signature. The administrator will have the right to manage keys (add, delete keys), unlock users (when locked), but will not have the right to perform digital signatures. Each

user has a separate password, and there is a regulation on the number of times the password is entered incorrectly. The PIN initialization process will be performed on the basis of the master key DEK and store the ciphertext in memory. The PIN authentication process will be performed on the basis of decoding the master key DEK from the ciphertext read from the device memory.

Specifically, the PIN initialization process is shown in Figure 2, including the following steps:

Step 1: The device randomly generates the SALT value and the master key DEK.

Step 2: From the SALT value and PIN through the PBKDF2 scheme to derive parameters including the key and IV used to encrypt DEK (KEY||IV) and the key for the HMAC-SHA256 scheme (KEY_HMAC).

Step 3: Perform master key encryption DEK using the KEY and IV values derived in step 2. The result of this step will be the master key ciphertext ENC_DEK.

Step 4: Calculate the HMAC-SHA256 value of the master key DEK using the key KEY_HMAC. The result of this step will be the HMAC_DEK value.

Step 5: Save the SALT||ENC_DEK||HMAC_DEK value to memory.

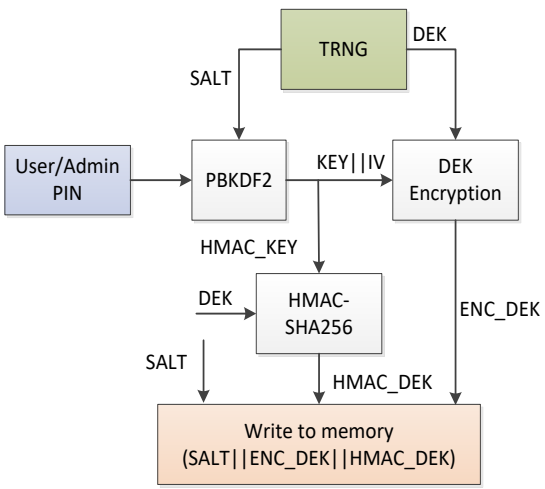

Figure 2. PIN generation flowchart

The PIN authentication process is performed as shown in Figure 3, including the following steps:

Step 1: Read the SALT||ENC_DEK||HMAC_DEK values from memory.

Step 2: From the SALT value and PIN through the PBKDF2 scheme to derive parameters including the key and IV used to decrypt DEK (KEY||IV) and the key for the HMAC-SHA256 scheme (KEY_HMAC).

Step 3: Decrypt the master key DEK using the KEY and IV values derived in step 2. The result of this step will be the decrypted version DEC_DEK.

Step 4: Calculate the HMAC-SHA256 value of DEC_DEK using the key KEY_HMAC. The result of this step will be the HMAC_DEC_DEK value.

Step 5: Compare HMAC_DEK and HMAC_DEC_DEK to determine the outcome of the PIN authentication process the result of the PIN authentication process.

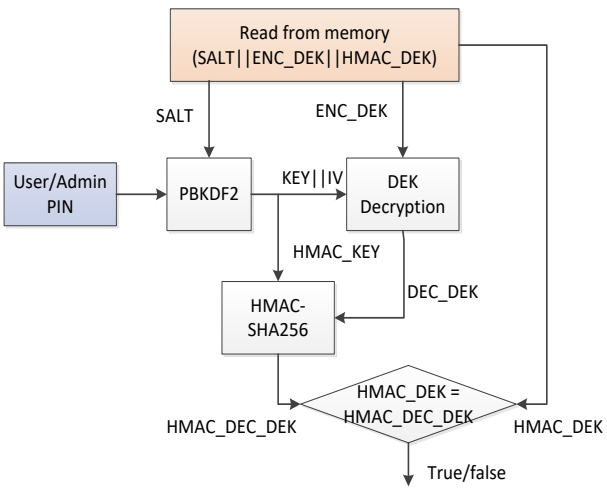

Figure 3. PIN authentication flowchart

IV. TESTING AND EVALUATION

After researching and coming up with a design solution, we have built a cold wallet prototype with the following specifications:

- Using STM32H723 chip;

- USB 2.0 interface;

- Supported algorithms: ECDSA (secp256k1), HMAC-SHA256, MKV, PBKDF2, BIP-32/39.

We conducted an experiment to evaluate the correctness of encrypting the private key in the device before saving it to the flash memory. In

this experiment, we used the Test Vector parameters in [15]. The experimental results shown in Figure 4, Figure 5 and Figure 6 show that the algorithm works correctly and according to the standard.

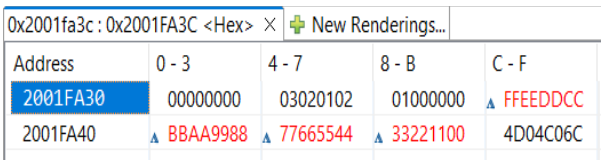

Figure 4. 128-bit plaintext data

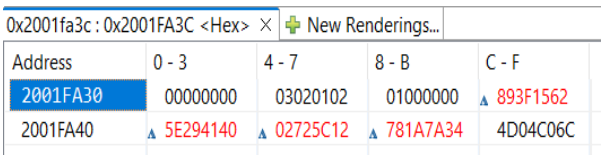

Figure 5. Ciphertext data

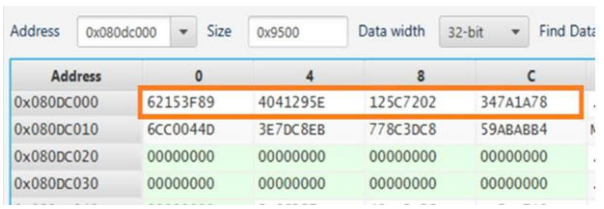

Figure 6. Key data written in flash memory

We tested the device's signing speed, the results are shown in Figure 7.

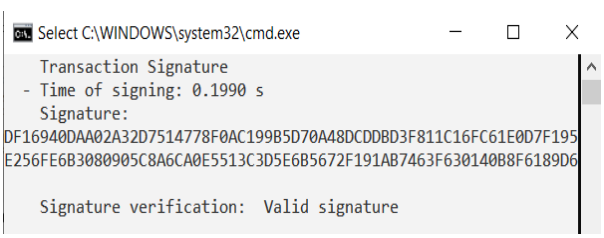

Figure 7. Transaction signature speed test results

Additionally, we conducted other tests and obtained the following results:

TABLE I. TEST RESULTS

No.	Test	Result
1	Generate master key DEK	Success, each time generates a different key
2	DEK master key storage	Check correct decryption authentication using user PIN
3	Change PIN	Works correctly, the DEK master key is decrypted with the old PIN and re-encrypted with the new PIN
4	Communicate with computer	Stable operation via USB HID, receiving and executing commands properly
5	Sign the transaction	Receive and execute transactions in correct format, PIN verification required before signing
6	Software attack	The device does not respond to any commands that are not part of the firmware-controlled protocol.
7	Key reading attack	The key cannot be read, all operations related to the key are performed on the device
8	PIN brute force attack	When the user PIN is entered incorrectly more than 10 times, the device is put into lock mode. To unlock, the administrator PIN must be authenticated. When the administrator PIN code is entered incorrectly more than 10 times, the device is completely locked and needs to reload the firmware for the device.
9	Digital signature performance evaluation (Execute 1000 times with varying random inputs).	Average signature speed: ~200 ms

In addition, we evaluated the security level of the device according to the standards in FIPS 140-3 [16]. The evaluation results are shown in Table II.

TABLE II. DEVICE EVALUATION RESULTS ACCORDING TO FIPS 140-3

No.	Standard group	Request	Proposed Device
1	Cryptographic module specification	The module must have a brief specification describing its functions, interfaces, and encryption protocols.	Have
2	Roles and services	User authorization and access authentication	Yes (PIN authentication)
3	Cryptographic algorithms	Uses FIPS-Approved Algorithms	Yes (ECDSA, SHA256, …)
4	Key management	Key lifecycle protection, key generation, storage, key deletion	Yes (key is encrypted before saving, not exported)
5	Physical security	Hardware access protection,	Partial (with anti-reverse mechanism)

		physical attack detection	
6	Logical security	Prevent sensitive data leakage during transmission	Yes (communication via standard protocol)

Based on the security levels in FIPS 140-3, it can be assessed that the device currently meets FIPS 140-3 level 2 and has the potential to approach level 3 if additional physical components such as anti-tamper epoxy and FIPS-compliant Secure Element are deployed.

TABLE III. COMPARISON RESULTS OF THE PROPOSED DEVICE AND COMMERCIAL DEVICES

Device	**Signing speed (ECDSA secp256k1)**
Proposed Device	~200 ms
Ledger Nano X	~300–400 ms (depending on Bluetooth/USB communication type)
Trezor Model T	~170–300 ms
Cold card	~200–400 ms
Keystone	~200–600 ms (QR-based, adds camera/scan delay)
Passport	~200–800 ms (QR-based, depending on camera UX)

A comparative assessment was conducted between the proposed device and several commercial security-oriented systems (Table III). The results demonstrate that the proposed device meets all required performance benchmarks, exhibiting security and operational efficiency comparable to leading commercial solutions. Furthermore, the proposed design consolidates key advantages found in commercial devices–such as transparent architecture, strict access control, and robust hardware-level security–while introducing a significant advantage in terms of design autonomy and in-house fabrication, which enhances trust, customizability, and supply chain security.

V. CONCLUSION

The paper presents the process of designing and manufacturing a secure cold wallet device for blockchain, from requirement analysis, hardware architecture, firmware, to private key security mechanisms and attack prevention. The test results show that the device operates stably, functions properly, consumes low power, and can operate with USB power.

The proposed device in this paper represents a novel approach to the design and fabrication of secure cold wallets, featuring several key characteristics: the integration of transparency and hardware-based security, autonomous manufacturing capability, flexible customizability, and alignment with international standards.

The results of the article also confirm that the authors have completely mastered the design, especially the cryptographic solution, successfully integrating the MKV algorithm to protect the security of the cryptographic parameters in the device (Data Encryption Key (DEK), private key). The device is capable of achieving the security level according to the FIPS 140-3 standard and ensuring readiness when deployed in environments with strict security requirements such as finance and government.

The above assessment of the device's security level according to FIPS 140-3 is only a subjective assessment of the author team. In the future, the research team will continue to research and improve the device's security level in the following directions:

- Integrate stronger hardware security, such as secure elements to store private keys;

- Build biometric user authentication mechanisms (e.g. fingerprints);

- Physical protection: circuit covered with anti-tamper epoxy.

- Firmware security: built-in firmware integrity verification mechanism at boot

- Logging and monitoring: Built-in memory area for error logging, can combine external memory.

- Expand security testing to include fault injection, and side-channel analysis.

ACKNOWNLEDGMENT

This work was researched under the support of the Science and Technology Project No. 03/2025/KCM of the Vietnam Government Information Security Commission. We would like to sincerely thank the project leader for creating conditions and supporting us during the research process.

REFERENCES

[1] Ledger, Accessed: 07-Jun-2025, *https://www.ledger.com/*.

[2] Trezor, Accessed: 07-Jun-2025, *https://trezor.io/*.

[3] Coldcard, Accessed: 07-Jun-2025, *https://coldcard.com/*.

[4] Keystone, Accessed: 07-Jun-2025, *https://keyst.one/*.

[5] Passport, Accessed: 07-Jun-2025, *https://bitcoin.org/en/wallets/hardware/passport/*.

[6] N. Ivanov, Q. Yan. "EthClipper: A Clipboard Meddling Attack on Hardware Wallets with Address Verification Evasion", in *2021 IEEE Conference on Communications and Network Security (CNS)*, IEEE, 2021. DOI: 10.1109/CNS53000.2021.9705033.

[7] Ledger Receive Attack. , Accessed: 07-Jun-2025, *https://www.docdroid.net/Jug5LX3/ledger-receive-address-attack.pdf*.

[8] Y. Erinle, Y. Kethepalli, Y. Feng, J. Xu. SoK: Design, Vulnerabilities, and Security Measures of Cryptocurrency Wallets (2025) , Accessed: 07-Jun-2025, *https://arxiv.org/pdf/2307.12874*.

[9] M. Guri. BeatCoin: Leaking Private Keys from Air-Gapped Cryptocurrency Wallets (2018), Accessed: 07-Jun-2025, *https://arxiv.org/pdf/1804.08714*.

[10] Phong, T. Q., Chi, D. D., Huy , T. D., & Diep, N. N, "On some issues affecting the security of RSA and ECDSA digital signature schemes", *Journal of Science and Technology on Information Security*, vol. 1, no. 18, pp. 38-46. June. 2023. DOI: https://doi.org/10.54654/isj.v1i18.884.

[11] Cam, N. T., Duy, P. N., & Phuc, H. M. T, "Large-scale Android malware detection by integrating Blockchain and IPFS for secure virus signature distribution", *Journal of Science and Technology on Information Security*, vol. 1, no. 24, pp. 72-87. June. 2025. DOI: https://doi.org/10.54654/isj.v1i24.1085.

[12] STMicroelectronics, "STM32H723 datasheet", Accessed: 22-Sep-2025, *https://www.st.com/resource/en/datasheet/stm32h723zg.pdf*.

[13] NIST Special Publication 800-132: "Recommendation for Password-Based Key Derivation", 2010.

[14] International Organization for Standardization, ISO/IEC 9797–2, Information technology – Security techniques – Message Authentication Codes (MACs) – Part 2: Mechanisms using a hash-function, 2011.

[15] National standard TCVN 14263:2024: Information technology - Security techniques - MKV block cipher algorithm.

[16] FIPS PUB 140-3: Security Requirements for Cryptographic Modules. NIST, 2019.

ABOUT THE AUTHOR

Vu Ta Cuong

Workplace: Institute of Cryptography Science and Technology, Vietnam Government Information Security Commission.

Email: vutacuong109@gmail.com

Education: Received bachelor's degree in 2011, master's degree in 2013, and PhD in 2016 from Kharkov Aerospace University, Ukraine.

Recent research direction: cryptographic techniques.

Tên tác giả: **Vũ Tá Cường**

Cơ quan công tác: Viện Khoa học – Công nghệ mật mã, Ban Cơ yếu Chính phủ, Việt Nam

Email: vutacuong109@gmail.com

Quá trình đào tạo: Tốt nghiệp cử nhân năm 2011, Thạc sĩ năm 2013 và Tiến sĩ năm 2016 tại Đại học Hàng không vũ trụ Kharkov, Ucraina.

Hướng nghiên cứu hiện nay: Kỹ thuật mật mã.

Nguyen Thanh Trung

Workplace: Institute of Cryptography Science and Technology, Vietnam Government Information Security Commission.

Email: trungbcy@gmail.com

Education: Received bachelor's degree in 1995, master's degree in 2005, from Academy of Cryptography Techniques.

Recent research direction: cryptographic techniques.

Tên tác giả: **Nguyễn Thành Trung**

Cơ quan công tác: Viện Khoa học – Công nghệ mật mã, Ban Cơ yếu Chính phủ, Việt Nam

Email: trungbcy@gmail.com

Quá trình đào tạo: Tốt nghiệp cử nhân năm 1995, Thạc sĩ năm 2005 tại Học viện Kỹ thuật mật mã.

Hướng nghiên cứu hiện nay: Kỹ thuật mật mã.