Inheritor: On-Chain Digital Inheritance

Author: aernoud@inheritor.com
Website: www.inheritor.com

Abstract

As the world becomes increasingly digitized, the value and prevalence of digital assets have grown significantly. Private keys to crypto wallets, online banking passwords, access to social media accounts, and personal video messages are examples of assets that may be lost upon an individual's unexpected death or incapacitation. We propose a peer-to-peer solution—Inheritor—that creates digital wills embedded in smart contracts to support event-driven, time-locked transfer of digital assets. This on-chain digital will serves as a crucial component of modern estate planning, ensuring that digital legacies are securely passed on to designated beneficiaries under predefined conditions.

Introduction

In today's digital era, individuals accumulate various digital assets that hold personal and financial value. Traditional inheritance processes involve creating a will with a notary who acts as the custodian of confidential documents and assets to be transferred. However, notaries can sometimes be compromised, leading to breaches of trust and security; numerous documented cases highlight instances where notarial misconduct has resulted in the mishandling of wills and estates. Additionally, access to reliable notaries can be a significant challenge in less developed countries, where the notarial system may be under-resourced or lack stringent regulatory oversight. Many people postpone creating formal inheritance arrangements due to ongoing life commitments and these systemic issues, resulting in a lack of secure mechanisms for asset transfer. This situation is exacerbated when individuals are geographically distant from their families or have unshared digital and physical assets, increasing the risk of total loss without a secure inheritance mechanism.

The Challenge of Digital Inheritance

Unexpected death or incapacitation can lead to the permanent loss of digital assets, as security protocols discourage sharing passwords and private cryptographic keys. Without knowledge of or access to these credentials, beneficiaries cannot retrieve important digital assets. Ensuring that digital assets remain inaccessible to beneficiaries until the inheritor's passing presents a unique challenge. Modern cryptographic systems typically require beneficiaries to possess private keys, granting them the ability to decrypt assets at any time, which conflicts with the requirement for conditional access based on the inheritor's status.

A Peer-to-Peer Solution

A peer-to-peer inheritance model is more intuitive, enabling individuals to directly appoint beneficiaries and allocate assets via a digital platform, thus eliminating reliance on third-party custodians. This approach addresses concerns about entrusting sensitive information—such as crypto private keys—to external entities. Existing solutions like multi-signature wallets and private server storage still require trust in third parties and their assurances of ethical practices and perpetual infrastructure.

Blockchain technology offers a trustless, distributed, peer-to-peer framework for transactions and data storage. This facilitates the development of solutions that empower individuals to act as their own custodians of digital assets designated for beneficiaries, effectively disrupting the traditional notary model.

Smart Contracts as Digital Wills

Smart contracts are conceptually ideal for guaranteeing the transfer of digital assets to designated beneficiaries under predefined conditions (e.g., within one month after the inheritor's death). Deployed on a perpetual and distributed blockchain, they provide cryptographic assurance that the "will" is executed precisely according to the inheritor's intentions, without the possibility of tampering or reliance on any third party.

A dead-man switch mechanism can be utilized to activate the transfer of access to the digital asset from owner to beneficiary. By storing the asset on the blockchain and controlling access via a smart contract, we enable a secure and conditional transfer.

Why Ethereum?

Initially, we planned to implement the digital inheritance concept on Cardano, recognized for its advanced conceptual and technical blockchain capabilities. However, the complexity and steep learning curve of Cardano development postponed this approach. Integrating Inheritor with the Cardano blockchain remains on our roadmap (Charles, my digits).

After evaluating various options, we decided to build Inheritor on Ethereum, the original smart contract blockchain. Working directly on Ethereum's Layer 1 provides optimal security, decentralization, and censorship resistance, which are crucial for applications with low transaction volumes that require high trust. Layer 2 solutions and alternative blockchains, such as Solana and Arbitrum, offer higher throughput and lower costs. However, they also introduce additional operational complexities, potential security vulnerabilities, and censorship resistance due to their reliance on off-chain computations and less mature infrastructure.

Moreover, these Layer 2 solutions often exhibit a less decentralized network of validators, conflicting with Ethereum's core principles of trustlessness and censorship resistance. By leveraging Ethereum's Layer 1, we ensure maximum security, decentralization, and longevity for the smart contracts governing digital inheritance.

While deploying smart contracts on Ethereum's Layer 1 can be relatively costly due to higher gas fees, we consider this expense to be a worthwhile investment in security and trustlessness. The additional costs are easily offset when compared to the operational expenses and risks associated with centralized solutions like notaries or solutions acting as or depending on third-party custodians.

Traditional methods often involve ongoing fees, the potential for human error, and vulnerabilities to fraud or malpractice. By utilizing Ethereum's robust and decentralized network, Inheritor eliminates intermediaries and leverages blockchain technology to provide a secure, one-time setup cost that ensures the integrity and longevity of digital wills. This guaranteed execution of a will is priceless, offering peace of mind that your final wishes will be honored without the uncertainties and delays associated with traditional processes.

Arbitrum One: A Cost-Efficient Complement to Ethereum's Unmatched Security

While Ethereum Layer 1 remains the gold standard for high-value digital wills—offering unmatched decentralization, censorship resistance, and instant finality—Arbitrum One provides a pragmatic, low-cost alternative for users prioritizing affordability. By leveraging Ethereum's security through Optimistic Rollups, Arbitrum slashes transaction fees by over 90%, making it ideal for smaller estates or frequent updates to inheritance plans.

However, Arbitrum's security model introduces trade-offs. Its validation process relies on a smaller set of validators to detect and contest fraudulent transactions during a 7-day dispute window. While Ethereum ultimately enforces finality, beneficiaries must temporarily trust that validators act diligently during this period—a contrast to Ethereum Layer 1, where transactions finalize irreversibly without intermediaries.

Censorship risks also differ. Arbitrum's sequencer, a centralized component responsible for ordering transactions, could theoretically delay or exclude transactions. Though mitigations like forced inclusion via Ethereum exist, Arbitrum's validator network remains less decentralized than Ethereum's global node distribution, marginally increasing exposure to collusion or regulatory pressure.

For high-value digital wills, these compromises warrant caution. While Arbitrum's fraud proofs and Ethereum-backed finality provide robust security, Ethereum Layer 1 stands alone for assets demanding absolute assurance against censorship or centralization.

In summary, while Arbitrum One presents an attractive solution for its lower costs and increased throughput, its reduced censorship resistance and greater centralization necessitate careful consideration—especially for high-value digital wills. Inheritor's integrated approach ensures users can tailor their digital inheritance strategy, optimizing for either cost efficiency or uncompromising security.

Inheritor's integrated approach ensures that users can optimise their digital inheritance strategy to meet both financial and security objectives.

Digital Asset Storage with Arweave

Storing large digital assets directly on the Ethereum blockchain is impractical due to inherent limitations. Ethereum imposes high gas costs for data storage, rendering large file uploads economically unfeasible. The network enforces strict block size and transaction data limits to maintain performance and decentralization.

Arweave offers significant advantages for secure, long-term inheritance solutions. It is a decentralized, add-only storage network utilizing a blockchain-like structure called the Blockweave. Its distributed nature ensures data redundancy and resistance to censorship or data loss.

The add-only architecture guarantees immutability; once data is stored, it cannot be altered or deleted. This is crucial for inheritance, as it ensures the digital asset remains intact and untampered until the beneficiary accesses it.

Moreover, Arweave's fee structure significantly enhances the practicality of Inheritor for digital inheritances. Arweave operates on a one-time payment model that ensures permanent data storage, eliminating the need for recurring fees. This "pay once, store forever" approach perfectly aligns with the requirements of digital inheritances, where assets must remain accessible and intact indefinitely. By leveraging Arweave, Inheritor provides a cost-effective solution that guarantees the enduring

preservation of digital assets without the burden of ongoing operational expenses. This ensures that beneficiaries can reliably access their inherited assets long into the future, without concerns about storage sustainability or escalating costs.

Inheritor in tandem with Arweave employs Merkle trees to handle large file uploads efficiently and to verify data integrity. By breaking files into chunks and hashing them into a Merkle tree structure, it enables quick detection of any tampering attempts. This cryptographic method ensures that even with large datasets, the integrity verification process remains efficient.

By encrypting digital assets before storage, confidentiality is maintained—only the intended beneficiary with the correct decryption key can access the content. Combined with Arweave's permanent and tamper-proof storage, this approach provides a secure and reliable method for preserving digital assets intended for future inheritance.

Overcoming Blockchain Challenges

Blockchains are public and transparent; all data and transactions are visible. For inheritance purposes, we require a method to store digital assets such that no one—not even the beneficiary with the private key—can access them prematurely. Implementing a time-release mechanism allows for the conditional release of the decryption key. While the decryption key can be stored on the public blockchain (since only the beneficiary can use it), it should remain hidden from the beneficiary until the appropriate time, especially for sensitive messages intended to be disclosed posthumously.

Additionally, interacting with blockchains can be complex for users, particularly due to the need to manage digital wallets, private keys, and gas fees associated with transactions. These complexities can be a significant barrier to entry for individuals who are not familiar with blockchain technology. Inheritor simplifies this process by leveraging ERC-4337 compliant contracts, which enable account abstraction on Ethereum. This allows users to interact with the blockchain without the need to set up traditional externally owned accounts (EOAs) or manage private keys directly.

ERC-4337, also known as Ethereum Improvement Proposal 4337, introduces a new standard for account abstraction at the protocol level without requiring consensus-layer changes. By using smart contract wallets that comply with ERC-4337, Inheritor can provide users with a seamless experience where the complexities of gas fees and key management are abstracted away.

With account abstraction, users can perform blockchain transactions using familiar authentication methods, such as biometric data(face-id), instead of dealing with private keys. Gas fees, which are typically paid in Ether, can be handled within the smart contract and are subsidized by Inheritor, removing the need for users to hold Ether just to pay for transaction fees.

By integrating ERC-4337 compliant smart contract wallets, Inheritor enables:

- Gas Abstraction: The smart contract can handle gas fees internally or utilize gas relayers, allowing users to perform transactions without needing to understand or pay gas fees directly.
- Seamless Interaction: Users can interact with the blockchain through the Inheritor app without installing browser extensions or specialized wallet software.

By adopting ERC-4337 account abstraction, Inheritor positions itself as one of the first iOS apps to seamlessly leverage blockchain technology without the typical complexities associated with it. By abstracting away the need for users to manage private keys or handle gas fees, Inheritor integrates

blockchain functionalities into the iOS platform in a user-friendly manner. This allows users to interact with the app just like any other iOS application, without needing to understand the underlying blockchain mechanics.

As a result, Inheritor not only simplifies the process of setting up a digital will but also demonstrates how blockchain technology can be made accessible to the general public through a intuitive mobile application.

Cryptography Without the Crypto

For the encryption and decryption of digital assets, Inheritor employs a robust cryptographic scheme that combines well-established primitives to ensure that only the designated beneficiary can access the encrypted content under predefined conditions. Each encryption operation begins with an ephemeral ECDH key exchange using the secp256k1 curve, which is aligned with Ethereum's cryptographic standards. This approach guarantees forward secrecy, as a fresh ephemeral key pair is generated for every session, ensuring that even if a long-term key is compromised, past sessions remain secure.

The raw ECDH shared secret is then processed through HKDF (as specified in RFC 5869) with SHA-256. A 32-byte random salt is generated using a secure random number generator (SecRandomCopyBytes), and a fixed, domain-specific string is used for domain separation. This two-step derivation transforms the raw shared secret into a uniformly random symmetric key, thereby mitigating any non-uniformity in the ECDH output.

For authenticated encryption, Inheritor utilises AES-GCM, which not only encrypts the digital asset but also produces an authentication tag to ensure both data integrity and confidentiality. This same mechanism is applied when encrypting the symmetric key itself, using the derived key from the HKDF process. The final encrypted package is self-contained, including the ephemeral public key (needed for rederiving the shared secret), the salt, the nonce for AES-GCM, the ciphertext, and the authentication tag. This comprehensive packaging allows the beneficiary to rederive the encryption key and securely decrypt the digital asset.

By combining ephemeral ECDH, HKDF for robust key derivation, and AES-GCM's authenticated encryption, Inheritor provides strong guarantees of forward secrecy, integrity, and confidentiality, ensuring that only the intended beneficiary can access the digital asset under the correct conditions.

Off-Chain Storage of the Decryption Key

Storing the encrypted symmetric key package on the Ethereum blockchain could allow the beneficiary to access the digital asset prematurely, before the conditions are fulfilled. Even designating variables as 'private' within a smart contract does not prevent external access, as all on-chain data is visible to network participants leveraging more sophisticated methods.

To address this, we store the package off-chain. This brings us to a different kind of blockchain trilemma involving decentralization, privacy, and security. While blockchain technology offers decentralization and security through its transparent and immutable ledger, this transparency can conflict with the need for privacy—particularly when handling sensitive information like decryption keys in a digital inheritance scenario.

Balancing Decentralization, Privacy, and Security

Public blockchains like Ethereum are inherently transparent; all data and transactions are publicly accessible. This transparency ensures trustlessness and verifiability but poses a challenge for applications requiring confidentiality. Storing sensitive data directly on-chain could expose it to unauthorized parties, undermining privacy and security.

This creates a trilemma where enhancing privacy may compromise decentralization or security:

- Enhancing Privacy: Utilizing off-chain storage or encryption can protect sensitive data but may introduce centralized components or reduce transparency.
- Maintaining Decentralization: Keeping all data on-chain supports decentralization but risks exposing sensitive information due to the blockchain's transparent nature.
- Ensuring Security: Robust security measures must protect against unauthorized access without introducing central points of failure or sacrificing decentralization.

Inheritor's Approach to the Trilemma

Inheritor addresses this trilemma by strategically balancing these aspects:

- Off-Chain Storage for Privacy: Sensitive data, such as the encrypted symmetric key, is stored off-chain using Cloudflare's distributed network. While Cloudflare is a privately owned entity (introducing a degree of centralization), it offers robust security and global data availability. This enhances privacy by keeping decryption keys off the public blockchain.
- On-Chain Verification for Decentralization and Security: Critical verification processes, such as enforcing inheritance conditions, are managed on-chain using Ethereum's immutable smart contracts. This maintains decentralization and leverages blockchain security for condition enforcement.
- Robust Cryptographic Techniques for Security: Advanced encryption methods ensure that even if off-chain data storage is centralized, the data remains secure and inaccessible without the proper cryptographic keys.

By accepting a slight compromise in decentralization for significant gains in privacy and security, Inheritor effectively balances this trilemma. The use of off-chain storage does not undermine the overall trustlessness of the system because the essential verification and state management occur on the decentralized blockchain.

An Innovative Time-Lock Mechanism

Traditional time-lock encryption methods, such as time-lock puzzles and Verifiable Delay Functions (VDFs), have limitations when applied to secure digital inheritance systems. Time-lock puzzles are inefficient due to high computational burdens and susceptibility to advancements in computing power. VDFs, while more secure against parallelization, still impose significant computational demands and do not adapt to external conditions like the owner's status.

The Inheritor smart contract overcomes these limitations by leveraging blockchain technology to create a conditionally accessible inheritance mechanism based on the owner's activity. The contract defines an InheritanceState with states such as Active, Claimable, Claimed, and Cancelled. The owner must periodically call the checkIn() function to reset the lastCheckIn timestamp, keeping the inheritance in the Active state. If the owner fails to check in within the specified checkInInterval plus a gracePeriod, the contract's state transitions to Claimable, allowing the beneficiary to call claimInheritance().

This mechanism ensures that the inheritance becomes accessible to the beneficiary under the correct conditions—specifically, when the owner has not checked in within the allotted time frame, indicating possible incapacitation or death. Ethereum's immutable and transparent blockchain enforces these conditions reliably and without the possibility of tampering.

The Role of Cloudflare Workers

The Cloudflare Worker acts as an intermediary that securely manages the conditional release of the encrypted symmetric key based on the state of the Ethereum smart contract. When a digital will is created, the Inheritor app sends a POST request to the Worker with the contractAddress and encryptedSymmetricKey. The Worker validates the input and stores the encrypted key in Cloudflare's Key-Value (KV) storage, using the contract address as the key.

When a beneficiary attempts to retrieve the key, Inheritor sends a GET request to the Worker. Before releasing the key, the Worker verifies whether the conditions for its release have been satisfied by checking the contract's state on the Ethereum blockchain. It uses ethers.js and connects through Infura to interact with the blockchain.

To optimize performance, the Worker implements a caching mechanism for the contract's state, reducing latency and minimizing blockchain network load. If the contract state confirms that the key should be released, the Worker provides the encrypted symmetric key. If not, it informs the requester accordingly.

Robust error handling and retry mechanisms ensure that transient network issues do not prevent the key from being released when appropriate. Security considerations are central to this design; the encrypted symmetric key is safe in KV storage because, without the appropriate private key, it cannot be decrypted.

Integrating the Components

Integrating the smart contract with components like Arweave and an off-chain Cloudflare Worker enhances the system's security and efficiency. The encrypted digital asset is stored on Arweave, ensuring data permanence and integrity. The Cloudflare Worker acts as an off-chain oracle, determining if the encrypted symmetric key should be released based on the smart contract's state.

By utilizing Cloudflare's distributed edge computing infrastructure, the system benefits from robust security and high availability, despite a slight trade-off in decentralization. This integrated solution represents a state-of-the-art method for secure, distributed time-lock implementation in digital wills and inheritances. It overcomes traditional limitations by eliminating reliance on computational delays and instead using time-based conditions and user activity to control access. The smart contract provides an immutable record of conditions, while off-chain services enforce access control efficiently.

By combining blockchain smart contracts with distributed storage and edge computing, this approach ensures that digital assets are accessible only to rightful beneficiaries under predefined conditions, maintaining privacy and security throughout the process. It sets a new standard for secure, conditional access to digital inheritances.

Ensuring Accessibility Beyond the Inheritor App

Inheritor is fundamentally designed to ensure that beneficiaries can retrieve their inherited digital assets even without the use of the Inheritor app or reliance on platforms like Apple's ecosystem. Recognizing

that the longevity and accessibility of digital inheritance should not depend on any single application or company, Inheritor utilizes open-source technologies and public domain protocols. All the cryptographic methods, smart contract codes, and blockchain data used are based on widely adopted, publicly available standards. This transparency ensures that the process is not tied to proprietary software or closed systems that could become obsolete or inaccessible.

To facilitate independent access, Inheritor will provide open-source scripts and comprehensive documentation in the public domain. These resources will guide beneficiaries on how to retrieve and decrypt their inherited assets without relying on the Inheritor app. The provided tools will include scripts for interacting directly with the Ethereum blockchain to verify the state of the inheritance smart contract and to invoke necessary functions like claimInheritance(). Additionally, decryption utilities will be available to perform the cryptographic operations required to recover the encrypted symmetric key and decrypt the digital asset stored on Arweave.

By making these resources publicly available, Inheritor ensures that beneficiaries can access their inheritance regardless of future uncertainties surrounding specific technologies or organizations. This approach aligns with the decentralized principles of blockchain technology, emphasizing user empowerment and resilience.

It guarantees that the inheritance mechanism remains operational indefinitely, free from dependencies on any single application or entity. Beneficiaries can be confident that they will be able to retrieve their inherited digital assets securely and efficiently, using publicly available tools and protocols, even if Inheritor or other platforms cease to exist.

Conclusion

Inheritor implements a robust methodology to facilitate the transfer of digital assets to beneficiaries at a future point using smart contracts. By leveraging Ethereum's immutable blockchain for condition enforcement, Arweave's permanent storage for secure asset preservation, and Cloudflare's distributed network for conditional key release, Inheritor effectively balances the trilemma of decentralization, privacy, and security.

This integrated system ensures that digital assets are securely managed and only accessible to designated beneficiaries under predefined conditions, without reliance on traditional intermediaries. By addressing the challenges of digital inheritance through a seamless integration of on-chain and off-chain technologies, our solution represents a significant advancement in digital inheritance management. It provides a practical and secure framework for preserving and transferring digital legacies in the modern world, ensuring that individuals can confidently pass on their digital assets to future generations.

References

1. **Merkel Trees**
   - Wikipedia: Merkle Tree. Available at: https://en.wikipedia.org/wiki/Merkle_tree

2. **Ethereum**
   - Buterin, V. (2013). "Ethereum Whitepaper: A Next-Generation Smart Contract and Decentralized Application Platform". Available at: https://ethereum.org/en/whitepaper/
   - Official Ethereum Website. Available at: https://ethereum.org/

3. **Arweave**
   - Arweave Documentation. Available at: https://docs.arweave.org/
   - Williams, S. "Arweave: A Protocol for Economically Sustainable Information Permanence" Available at: https://www.arweave.org/yellow-paper.pdf

4. **Symmetric Key Encryption (AES-GCM)**
   - NIST Special Publication 800-38D: "Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC". Available at: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38d.pdf
   - Wikipedia: Galois/Counter Mode (GCM). Available at: https://en.wikipedia.org/wiki/Galois/Counter_Mode

5. **Elliptic Curve Diffie-Hellman (ECDH)**
   - Standards for Efficient Cryptography Group (SECG): "SEC 1: Elliptic Curve Cryptography". Available at: https://www.secg.org/sec1-v2.pdf
   - Wikipedia: Elliptic-curve Diffie–Hellman. Available at: https://en.wikipedia.org/wiki/Elliptic-curve_Diffie–Hellman

6. **Verifiable Delay Functions (VDFs)**
   - Boneh, D., Bonneau, J., Bünz, B., & Fisch, B. (2018). "Verifiable Delay Functions". Available at: https://eprint.iacr.org/2018/601.pdf

7. **Cloudflare Workers**
   - Cloudflare Workers Documentation. Available at: https://developers.cloudflare.com/workers/
   - Cloudflare Blog: "Introducing Cloudflare Workers". Available at: https://blog.cloudflare.com/introducing-cloudflare-workers/

8. **ECDSA and secp256k1 Curve**
   - Standards for Efficient Cryptography Group (SECG): "SEC 2: Recommended Elliptic Curve

9. **Time-Lock Encryption**
   - Rivest, R. L., Shamir, A., & Wagner, D. A. (1996). "Time-lock Puzzles and Timed-release Crypto". Available at: https://people.csail.mit.edu/rivest/pubs/RSW96.pdf
   - Wikipedia: Time-lock Puzzle. Available at: https://en.wikipedia.org/wiki/Time-lock_puzzle

10. **Edge Computing**
    - Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). "Edge Computing: Vision and Challenges". IEEE Internet of Things Journal, 3(5), 637-646. Available at: https://ieeexplore.ieee.org/document/7488250
    - Wikipedia: Edge Computing. Available at: https://en.wikipedia.org/wiki/Edge_computing**Elli**ptic

11. Curve Digital Signature Algorithm (ECDSA)

- NIST FIPS PUB 186-4: “Digital Signature Standard (DSS)”. Available at: https://doi.org/10.6028/NIST.FIPS.186-5

12. Symmetric Key Encryption

- Schneier, B. (1996). Applied Cryptography: Protocols, Algorithms, and Source Code in C. John Wiley & Sons.
- Wikipedia: Symmetric-key Algorithm. Available at: https://en.wikipedia.org/wiki/Symmetric-key_algorithmInfura
- Infura Documentation. Available at: https://infura.io/docs

13. Ethers.js

- Ethers.js Documentation. Available at: https://docs.ethers.io/v5/
- GitHub Repository: ethers-io/ethers.js. Available at: https://github.com/ethers-io/ethers.js/

Acknowledgments

We would like to thank the open-source and AI community and the developers of the technologies utilized in this project. Their contributions have been invaluable in enabling the development of Inheritor.

Future Work

Looking ahead, we plan to:

- Expand Blockchain Compatibility: Integrate Inheritor with Cardano to offer users more options based on their preferences and needs.
- Enhance User Experience: Continue simplifying the user interface to make the process of setting up digital wills even more accessible to non-technical users.
- Security Audits: Undergo comprehensive security audits by third-party experts to validate and enhance the security measures implemented.
- Legal Compliance: Collaborate with legal experts to ensure that digital wills created with Inheritor are recognized and enforceable under various jurisdictions.

Glossary

- Smart Contract: A self-executing contract with the terms of the agreement directly written into code, running on a blockchain.
- Blockchain Trilemma: The challenge in achieving decentralization, security, and scalability (or privacy) simultaneously in blockchain systems.
- Dead-Man Switch: A mechanism that triggers an action if a required input is not received within a set time, used here to transition the contract state based on the owner's activity.
- AES-GCM: Advanced Encryption Standard in Galois/Counter Mode, a symmetric encryption algorithm providing confidentiality and integrity through combined encryption and authentication.
- ECDH: Elliptic Curve Diffie-Hellman, a key agreement protocol that allows two parties to establish a shared secret over an insecure channel.
- Ephemeral Key: A temporary key used for a single session or transaction to enhance security through forward secrecy.
- Arweave Blockweave: A blockchain-like data structure used by Arweave to store data permanently and efficiently.
- KV Storage: Key-Value Storage, a simple database that stores data as a collection of key-value pairs.
- HKDF: HMAC-based Key Derivation Function; a method for deriving cryptographically strong keys from a shared secret. It employs a random salt and a domain-specific info string to ensure the derived key is uniformly random and context-bound.
- Salt (Cryptographic): A randomly generated value used in key derivation to ensure that even identical inputs produce unique derived keys, preventing vulnerabilities from key reuse.
- Nonce: A number used only once in encryption processes, such as in AES-GCM, to guarantee that each encryption operation produces a unique ciphertext even when the same plaintext is encrypted multiple times.
- Domain Separation (Info String): A context-specific value used in key derivation (e.g., within HKDF) to bind the derived key to a particular application or purpose, ensuring that keys derived in different contexts remain distinct.
- Forward Secrecy: A security property that ensures the compromise of long-term keys does not jeopardise the security of past session keys, protecting previous communications even if current keys are compromised..

Contact Information

For more information about Inheritor or to contribute to the project, please contact:

- Email: aernoud@inheritor.com
- Website: www.inheritor.com

Disclaimer

This white paper is for informational purposes only and does not constitute legal, financial, or investment advice. Users should consult with professional advisors before making any decisions based on the information provided.

Note: The technologies and protocols referenced in this document are subject to ongoing development and improvement. Readers are encouraged to refer to the latest documentation and research for the most up-to-date information.