Finance and Economics Discussion Series

Federal Reserve Board, Washington, D.C.
ISSN 1936-2854 (Print)
ISSN 2767-3898 (Online)

A Robust Risk Framework for Offline Payments

Bikash Poudel, Sarah Carey, Robert Flynn, Chakrapani Narayan, Richard Payne, Eshwar Satrasala, Seaira Spooney, and James Lovejoy

2025-105

Please cite this paper as:
Poudel, Bikash, Sarah Carey, Robert Flynn, Chakrapani Narayan, Richard Payne, Eshwar Satrasala, Seaira Spooney, and James Lovejoy (2025). "A Robust Risk Framework for Offline Payments," Finance and Economics Discussion Series 2025-105. Washington: Board of Governors of the Federal Reserve System, https://doi.org/10.17016/FEDS.2025.105.

A Robust Risk Framework for Offline Payments

Bikash Poudel, Sarah Carey, Robert Flynn, Chakrapani Narayan, Richard Payne, Eshwar Satrasala, Seaira Spooney, and James Lovejoy[1]

Abstract

The capability to make offline digital payments is emerging as a vital component of the broader payments ecosystem, especially in scenarios in which internet connectivity is unavailable such as during a crisis or natural disaster. Offline digital payment services offer a secure and reliable alternative to cash. Even so, there are a limited number of viable offline payment protocols in production today. Our work introduces a comprehensive model for a secure, end-to-end offline payment experience using consumer-grade smartphones. Our model employs a robust cryptographic protocol and tamper-evident secure element (SE) hardware to prevent double-spending and counterfeiting while preserving user privacy.

Introduction

Offline payment functionality is an emerging digital payment technology that enables users to transact without an internet connection. Currently, physical cash is the standard offline payment solution. However, in times of crisis, reliable access to cash presents operational and logistical challenges for central banks, depository institutions, and consumers alike. By not relying on internet access, offline payment functionality can replicate some features of cash (e.g., can be used without a connection) while minimizing its downsides, such as the need for physical handling, storage, and physical security. This offline capability significantly enhances payment system resiliency, especially during situations like a pandemic or widespread network outages, such as those caused by natural disasters. It also provides policymakers with flexible options to scale currency distribution in exigent circumstances and enables merchants and consumers to continue making transactions without the need for internet access.

Currently, there is a lack of an end-to-end offline digital payments systems in production[2]. Our work addresses the offline payments challenges of double-spending and counterfeiting by proposing an end-to-end offline model that could ensure the secure exchange of an offline token during operational outages of varying types.

Beyond the general benefits to the broader ecosystem, offline payments could offer significant advantages to merchants and consumers. Presently, consumers lack a digital alternative to cash that functions without internet access. U.S. payments system operators are also limited in their ability to continue operating payments systems without internet connectivity. Therefore, developing robust offline payment capability is crucial to addressing these vulnerabilities and providing a reliable digital option when cash is not available.

Background

Presently, offline payments are primarily facilitated using physical cash. Although consumers are increasingly adopting mobile payment methods, consumers have expressed a clear desire to use cash in times of crisis (Judson 2024) or when there is a prolonged loss of internet connectivity. Current trends show that the growth in mobile payments usage is outpacing the decline in cash-based payments. For example, according to the Atlanta Fed's 2024 Survey and Diary of Consumer Payment Choice,[3] while the use of cash declined by 2 percentage points, from 16 percent in 2023 to 14 percent in 2024, the use of mobile payments increased by 4 percentage points during that same period, from 28 to 32 percent. Digital wallets, which facilitate mobile payments, store a variety of transaction data to process and settle payments. Given the trends in consumer behavior, our work provides additional options to obtain cash-like features while securing digital transaction data with advanced cryptographic protocols.

Offline Data Transfer Technology

Offline payments could utilize either near field communication (NFC) or Bluetooth technology as data transport methods to securely exchange tokens from one phone to another over an encrypted channel. In our model, to execute a transaction, the user would bring two smartphones into close proximity to initiate the data transfer. The two wallets would then initiate a three-phase offline transfer protocol to perform the token transfer. Additional details on token transfer can be found in "Offline Token Transfer Protocol," pg. 13.

Benefits of Offline Payments

Offline payment technologies offer several benefits to consumers due to their privacy, accessibility, security, and safety features:

1. **Privacy:** Offline payments are cash-like, ensuring transaction privacy within the digital wallet. For example, in our model, digital wallets do not contain any personally identifiable information (PII), only wallet addresses and proofs.
2. **Accessibility:** Payments can be made with a smartphone or at a point-of-sale terminal without the need for internet connectivity,
3. **Security:** Offline tokens can be managed and secured even if a device is offline, and
4. **Safety:** Offline functionality reduces the need to carry physical cash, minimizing the risk of theft.

Our Proposed Solution

Our solution introduces an offline token transfer model, leveraging our novel SignOnce semantics-based cryptographic protocol implemented in a tamper-resistant secure element (SE) hardware chip.[4] The SE, integrated within a smartphone's system-on-chip, executes critical cryptographic operations and stores cryptographic keys essential for our device-to-device offline token transfer protocol. We implemented the offline protocol in a digital wallet and tested the offline token transfer protocol in iOS smartphones using Bluetooth as the transport method. For more information on SignOnce, refer to "Offline Applet," pg. 6.

Our protocol is also distinct in that it is operating system-agnostic and addresses the challenges of double-spending and counterfeiting, even when the wallet remains offline for a prolonged period. Our work ensures that the offline wallet can verify the authenticity of a received token without requiring an internet connection and provides a secure and reliable offline payment model.

Solution Scope

We focused our work on device-to-device transactions between two smartphones in an offline state. In this case, offline payments could enable users to make payments instantly with their phones in the event of a critical outage where internet-based and other forms of payment may not be available. Each wallet in our model would have pre-loaded offline tokens and an online balance. The recharge function in the wallet enables users to convert the desired amount of their online balance to offline tokens. Then, when offline (i.e., without internet connectivity), an offline token transfer would occur after the following steps:

1. **Offline Token Transfer**: First, users must securely access their offline wallet account. Our model uses a password-protected wallet. However, the specific user authorization protocol could vary depending upon the digital wallet provider and the user's personal preferences, similar to the various ways to access a digital wallet today (e.g., biometric data or user-specific pin codes). Once the sender and receiver access their respective wallets, the sender's wallet discovers the receiver's wallet and the offline token of desired value is transferred.

   To prevent an attacker from intercepting the transaction, our model incorporates two features, a challenge-response protocol—a question-and-answer only available to the sender and receiver—and a "time-out clock." If the receiver does not respond with the correct challenge response in the designated amount of time, the transaction will not be completed.

   Upon successful completion, the transaction is reflected in the offline transaction history, or balance, of both the users. The transaction settles instantly in both wallets. Additionally, our protocol includes a token verification phase in which both parties can verify the authenticity of the offline token when offline.

2. **Online Deposit**: Once the wallets are back online, users can deposit excess offline tokens into their bank accounts, which will be reflected in their online balance. This step is optional but desirable, as users may want to limit the number of offline tokens they hold.

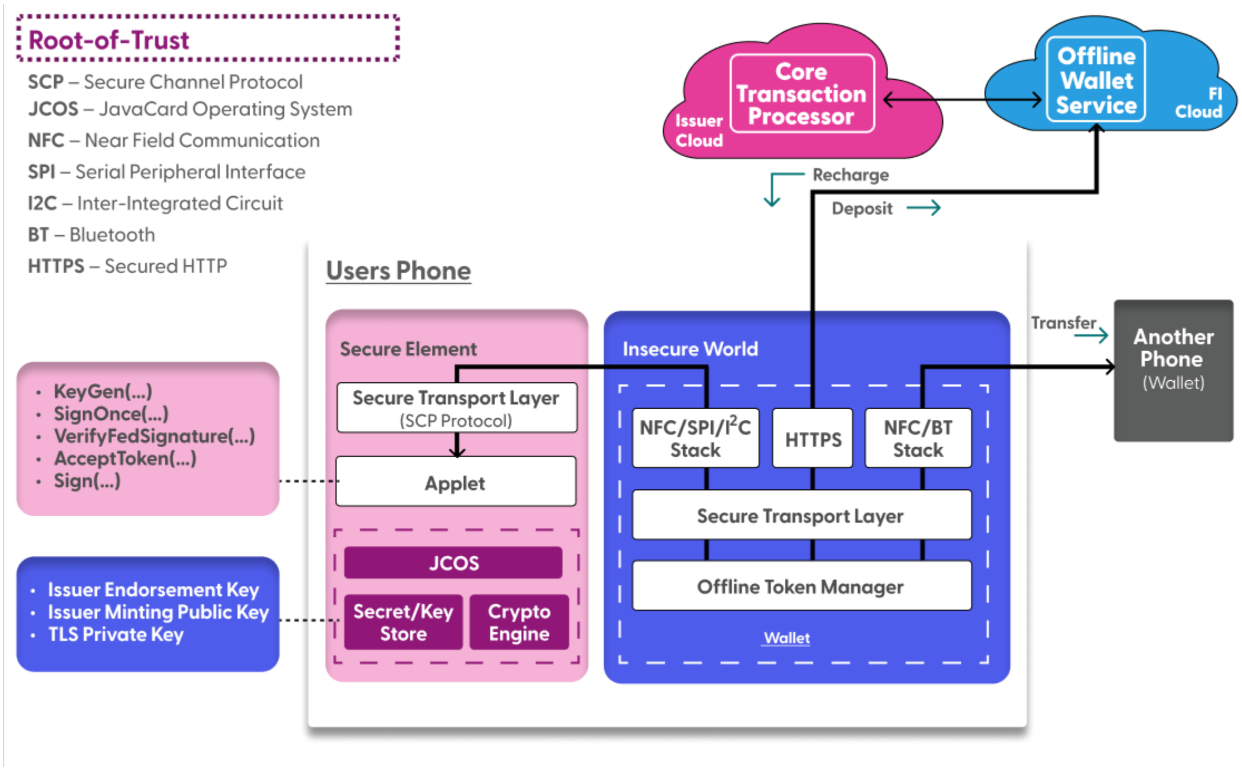
We do not specify the issuer in our model. However, to provide additional clarity on how the solution could operate, the following text outlines the role an issuer could play. Potential issuers could include U.S. payments infrastructure operators with joint accounts[5] at Federal Reserve Banks, a cooperative of depository institutions, or an entirely new, fit-for-purpose, private-sector arrangement. In all cases, the tokens should be pegged or convertible to the U.S. dollar and as similar as possible to a digital bearer instrument, functioning as cash would in a consumer's physical wallet. The issuer should be interoperable with, and offer cloud-based services to, U.S. depository institutions. It should also possess the operational sophistication necessary to facilitate tokenized transactions and supervise mitigation strategies for counterfeiting and double-spending.

For more information on offline token creation and exchange, refer to "Solution Technical Design," pg. 4.

Solution Technical Design

The high-level architecture of the offline wallet and the interaction with the services running in the issuer's cloud is depicted in Figure 1.

Figure 1: Architecture of an offline wallet with a secure element hardware chip



Offline Wallet

The offline wallet has two major components, the offline wallet application (app) itself and the offline applet. The offline wallet app is designed to run on a user's smartphone as any app does today. We define the environment containing apps that are not in the secure element as the insecure world, as illustrated in Figure 1. That is because the apps hosted there could become infected with malware or be subject to various classes of firmware, system software, and application-level attacks.[6] The offline applet is insulated from the insecure world because it runs on the SE, effectively creating a "secure world." The SE is a tamper-evident chip and is integrated in the phone's system-on-chip. Because all cryptographic key and sensitive operations happen inside the offline applet running on the SE, our protocol remains secure against the simple hardware, firmware, system software, unprivileged software, and network adversaries.

The offline wallet performs all offline token management and secure session management activities. The offline token manager is the main component of the offline wallet application. It handles the recharge, merge, split, transfer, and deposit of offline tokens:

- The recharge function refills the wallet with the token(s) of desired value. It is essentially an online to offline token conversion, where all or part of the user's online account balance is converted into offline tokens. The recharge function increases the offline token holdings while decreasing the online balance.
- The merge and split operations help users create distinct token denominations. These operations are akin to creating change from physical dollars.
    - A merge function combines two or more tokens into one whose value is the sum of the values of the combining tokens.
    - A split function generates two tokens out of one token, where one of the tokens holds the desired value and the other holds the change.
- A transfer function conveys the ownership of a token from one wallet to another and transports the token to another wallet.
- A deposit function enables the wallet to convert the offline tokens to an online account balance in the user's bank account. A deposit increases the online balance while decreasing the total value of the offline token(s).

The secure transport layer is another component of the wallet that manages the secure communication channels between the wallet and three key stakeholders:

1. Another offline wallet,
2. The offline applet running in the SE, or
3. Services running in a financial institution's or issuer's cloud.

From the retail user's perspective, the experience could be similar to the current tap-to-pay process. Technologically speaking, the wallet-to-wallet communication can happen using NFC or Bluetooth transport. The offline wallet application to SE communication can happen using NFC, serial peripheral communication (SPI), or inter-integrated circuit (I^2C) buses. Because the offline wallet resides in the insecure world, communication between the offline wallet and the cloud services in the issuer's cloud is secured using HTTPS. The HTTPS would use the latest version of the transport layer security (TLS) protocol, TLSv1.3 or higher. The private key needed to initiate the secure transfer of information for the TLS (TLS handshake) would be stored securely inside the SE.

Offline Applet

The SE is a specialized chip in a user's smartphone containing state-of-the-art security countermeasures for hardware and software vulnerabilities. It would provide all the cryptographic functions needed for the offline wallet and applet, as well as a secret or key store[7] for sensitive information like a TLS private key. An applet's private key enables it to sign a transaction and prove offline token ownership, while its public key is used to receive and verify the transaction. The API calls and operations are detailed below.

1. Key Generation

The first cryptographic operation implemented in the offline applet is called KeyGen and is invoked by an offline token-receiving wallet. During an offline token transfer, the sender's wallet needs to transfer ownership of the candidate token to the receiver. This is done by embedding the receiver's token ownership public key (KTO_{pub}) and the issuer's endorsement (digital signature) on KTO_{pub} into the candidate token. For this, the receiver's wallet must invoke the KeyGen API in the offline applet running in the SE to generate the token ownership key pair (KTO_{prv}, KTO_{pub}) and the issuer's endorsement for KTO_{pub}. The issuer's endorsement is an important component in the key generation as it separates a valid KTO_{pub} from a forged or an invalid KTO_{pub}.

The issuer's key endorsement private key (KIE_{prv}) lives in the SE key store and signs the KTO_{pub} key as an issuer's endorsement. The KIE_{prv} is the root-of-trust of the proposed offline protocol because the security of the proposed protocol hinges on the confidentiality and integrity of this key. The private part of the token ownership key (KTO_{prv}) is stored in the SE secret store and is not exposed to the offline wallet in any way. The issuer's key endorsement private key (KIE_{prv}) is configured in the SE key store during the manufacturing and factory provisioning process.[8]

2. SignOnce

The second cryptographic operation in the offline applet is called SignOnce. SignOnce is used during token transfer to convey the ownership of a token from the sender's wallet to the receiver's wallet. SignOnce generates the proof-of-transfer-of-ownership. This is done by signing the candidate token using the token ownership private key (KTO_{prv}) corresponding to the token ownership public key (KTO_{pub}) in the token. After the successful proof generation, the KTO_{prv} is deleted from the key store so that the same token cannot be spent or transferred again. Thus, the sender's wallet uses the SignOnce API to transfer the ownership of a token to the receiver where, at the end of the operation, the signing key (KTO_{prv}) gets deleted from the key store. The atomicity[9] of successful signing and successful deletion of the key is guaranteed by the fault injection-hardened offline applet. This is how we prevent double-spending.

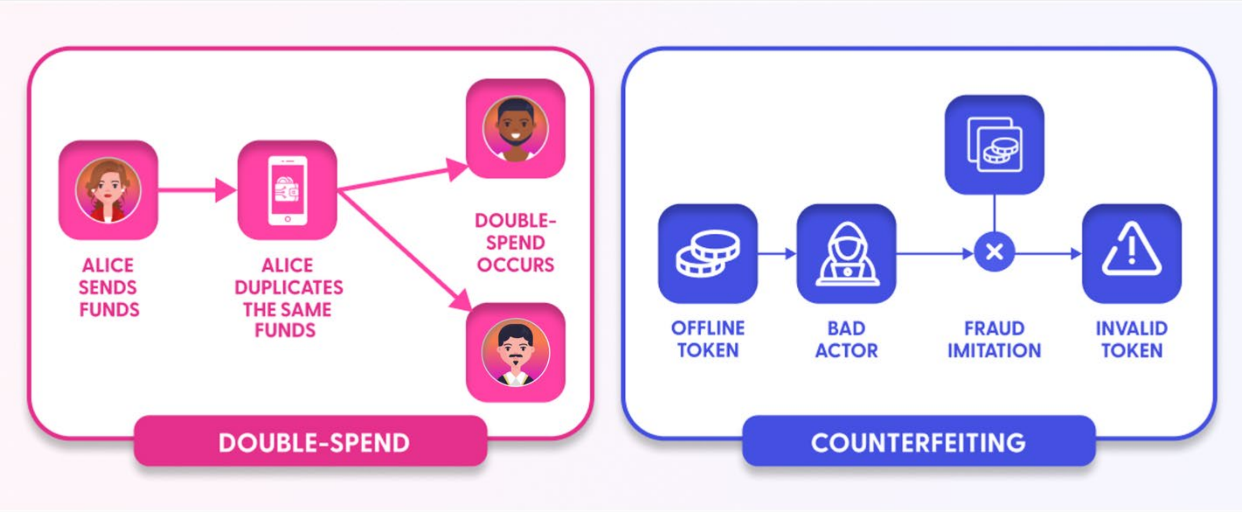
3. TLS Payload Signing

The third function in the offline applet is TLS payload Sign, which is used by the HTTPS connection between the offline wallet and services running in the financial institution's or issuer's cloud. Signing occurs during the TLSv1.3 handshake, where the elliptic curve or RSA cryptographic algorithm signs the TLSv1.3 handshake payload.

Threat Modeling

The two main vulnerabilities intrinsic to cash include counterfeiting and theft. To maintain parity or improve upon the security and safety standards of cash, our prototype addresses and solves for the vulnerabilities inherent to digital offline payments, including double-spending and counterfeiting.

Figure 2: Threats applicable to offline protocol



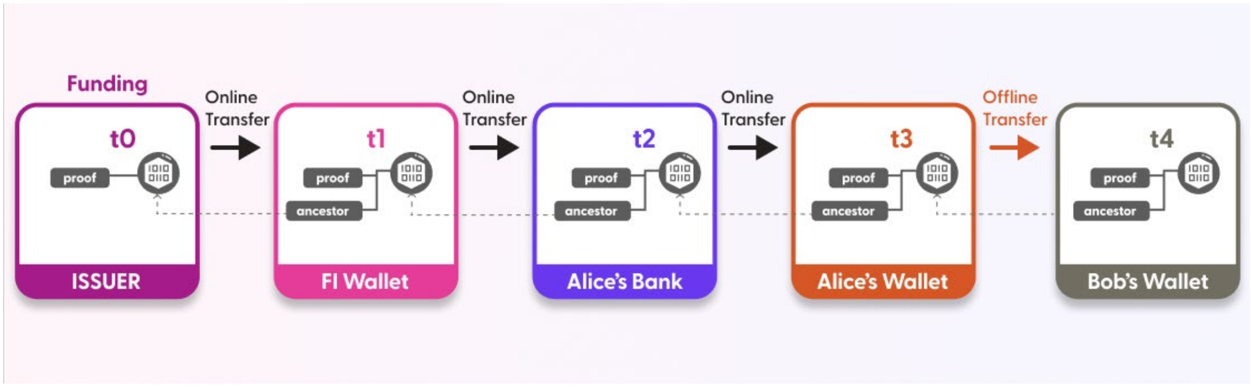
Double-spending

Double-spend is the risk that a sender could use the same offline token to pay two or more distinct recipients (See left panel of Figure 2). Offline wallets are not connected to any cloud environment when the offline token transaction occurs. Therefore, the wallet must be self-sufficient and able to prevent itself from receiving a token that has already been spent.

Our model's SignOnce semantic-based offline protocol could effectively mitigate the double spend problem. SignOnce enforcement would occur inside the fault-injection hardened offline applet running inside the SE, thus rendering the protocol robust against network, system software, and simple hardware adversaries.

In addition to SignOnce, the offline token data structure keeps track of all of a token's ancestors from the time it was minted. The token history verification process checks and validates all the fields of the token and its ancestors, ensuring that the same token is not repeated in the ancestor chain. This is depicted in Figure 3 for token t4. Token t4 ended up in Bob's wallet after three online transfers and one offline transfer. Thus, {t0, t1, t2, t3} are the ancestors of t4. The diagram shows that the tokens are chained by the ancestor field such that we could verify all the proofs until we hit the root or freshly minted token (t0). All digital wallets can verify all the proofs in the token because the public key is part of the token itself. For the last proof generated by the issuer's minting key, the wallet can use the issuer's minting public key stored locally in the wallet for signature verification. This means one requirement for our offline payment model is that the token minter's public key and certificate must be provisioned in the offline wallet. Another is that a strong key should be used for signing the minted token so that the lifetime of the key can be longer.

Figure 3: Ancestors of token t4 are tracked all the way back to the freshly minted token from the issuer (t0).



Counterfeiting

Another important attack to consider is the counterfeiting of a token (See right panel of Figure 2). A counterfeiting attack is an attempt to fraudulently imitate a valid offline token by tampering with various token fields. Tampering attacks can include:

- Increasing the value of a token by tampering with the value field,
- Changing the ownership of a token by tampering with the ownership field
- Forging a valid proof in the proof of ownership field, or
- Modifying the ancestor tokens field to remove a double-spending incident.

The token history verification process would check and validate all the fields of the token data structure and repeat this for all of the tokens in the ancestor chain. This process would detect any evidence of token tampering anywhere in the ancestor chain.

The only way to successfully attack an offline wallet is to physically break or clone an SE, because the cryptographic key used for the SignOnce mechanism is stored there. Since our implementation would use a tamper-evident SE, engineered to maintain its integrity against most known hardware and software vulnerabilities, only a well-funded, skilled hardware adversary could break the protocol. This threat is pertinent to all state-of-the-art digital wallets, such as ApplePay, GooglePay, and SamsungPay, because all of these platforms rely on an SE as the trusted computing base (TCB) or secure key storage (i.e., key vault) solution.

Our early exploration of these threat models and their implications aimed to provide a framework for threat detection, potential mitigation, and containment strategies in the event of a security breach. This effort drove our technical design decisions, protocol development, and implementation. In the following section, we detail the architecture of the offline wallet and its implementation.

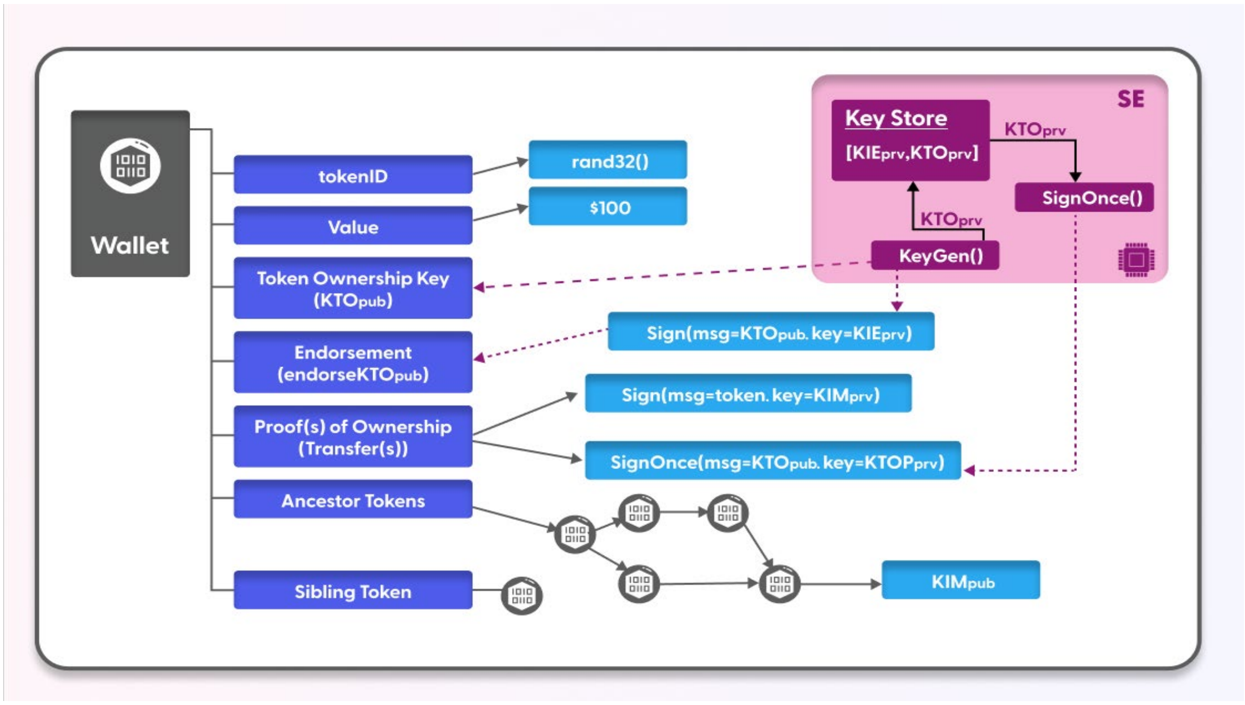
Implementation

This section outlines how an offline token could be represented as a digital form of cash and describes its technical lifecycle, from minting to redemption.

Representing an Offline Token

The offline token data structure and associated fields are depicted in Figure 4.

Figure 4: Offline token data structure


Wallet
tokenID
Value
Token Ownership Key (KTOpub)
Endorsement (endorseKTOpub)
Proof(s) of Ownership (Transfer(s))
Ancestor Tokens
Sibling Token
rand32()
$100
SE
Key Store
[KIEprv,KTOprv]
KTOprv
SignOnce()
KTOprv
KeyGen()
Sign(msg=KTOpub. key=KIEprv)
Sign(msg=token. key=KIMprv)
SignOnce(msg=KTOpub. key=KTOPprv)
KIMpub

A detailed description of each of the fields is described below:

1. **Token Identifier (ID)** is a unique number that identifies a token. The offline protocol ensures that no two valid tokens have the same token ID.

2. **Value** is a big integer number that represents the worth of the offline token(s), measured in U.S. dollars.

3. **Token Ownership Key (KTO_{pub})** is a public part of the token ownership key pair (KTO) that identifies the specific wallet to which the token belongs. In our offline protocol, each token must belong to a specific wallet. This is done by storing the public part of the KTO in the token itself while the private part of the KTO is stored in the SE key store. KTO_{prv} is only accessible to the offline applet during the SignOnce operation.

4. **Endorsement** refers to a digital signature of the KTO_{pub} signed by the issuer's key endorsement private key (KIE_{prv}). The endorsement field ensures that the token ownership public key is valid. Because all wallets have access to the KIE_{pub}, they will be able to verify the endorsement field. The private part of the endorsement key (KIE_{prv}) is stored in the SE secret store. This is done during the manufacturing provisioning phase of the smartphone itself, similar to how Apple or Samsung programs their keys in their respective secure elements. This ensures that the wallets with an issuer-endorsed secure element can generate valid token ownership keys.

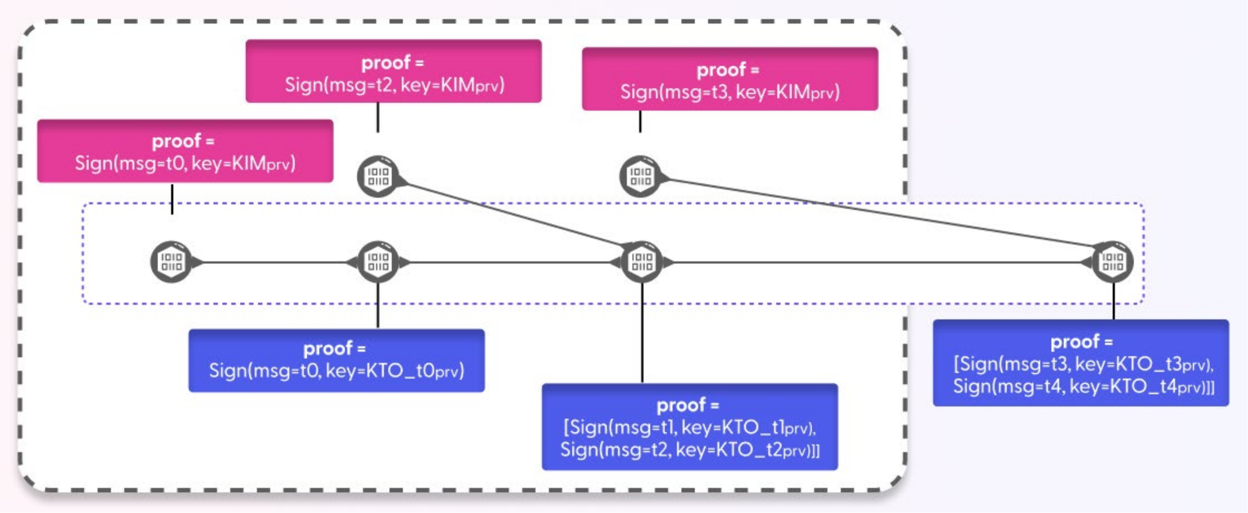
5. **Proof of a Valid Ownership Transfer (aka Proof):** Proof generation happens in two scenarios—during token minting in the issuer's cloud and during token transfer from one party to another. In the issuer's cloud, there is a cloud-based service called a core transaction processor (CTP) that implements the token minting function. During token minting, the token is signed by the token minting private key (KIM_{prv}) and the signature is stored in the token's proof field. In the token transfer scenario, the sender's KTO_{prv} is used to perform SignOnce on the modified token (i.e., a token whose token ownership key and endorsement are replaced with that of the receiver's KTO_{pub} and endorsement) which generates a digital signature. The proof field is populated with this signature.

6. **Ancestor Token(s):** As described earlier (Figure 3), the ancestor field keeps track of all the tokens that are involved in the creation of a given token. For token t4 in Figure 3, the set {t0, t1, t2, t3} are the ancestors where t0 is the root token that was minted. For example, say a token was formed by this sequence—mint, transfer, merge, and merge, as displayed in Figure 5, the ancestor field of token t5 would track all the parent tokens going up to the root tokens (i.e., tokens that are minted and transferred by the CTP). In Figure 5, tokens t0, t2, and t3 are the root tokens. The proof field of a root token is signed by the KIM_{prv} which can be verified by the KIM_{pub}. The proof field of transferred, merged, or split tokens is signed by their respective parent tokens' KTO_{prv} using SignOnce.

Because the ancestor chain length and token size are positively correlated, transactions may take longer to verify, and therefore settle, with each additional transaction. Our prototype includes a logic that automatically deposits offline tokens with a long ancestor chain (i.e., more than 30)[10] when the wallet comes back online to help mitigate transaction latency. This logic ensures a token with a long ancestor chain is deposited and burned in the issuer's cloud and is replaced by a new token with just the issuer's proof and no ancestor. While this optimization (i.e., the long ancestor chain replacement) maintains the prototype's security guarantees, it also requires the wallet to be online.

7. **Sibling Token(s):** This field is a record that tracks the sibling of a token formed by a split operation. Splitting a token produces two tokens, a desired token and a change token. The desired and change tokens use the sibling field to track their sibling. This field is also used during the token history verification function to check if double-spending occurred while splitting a token.

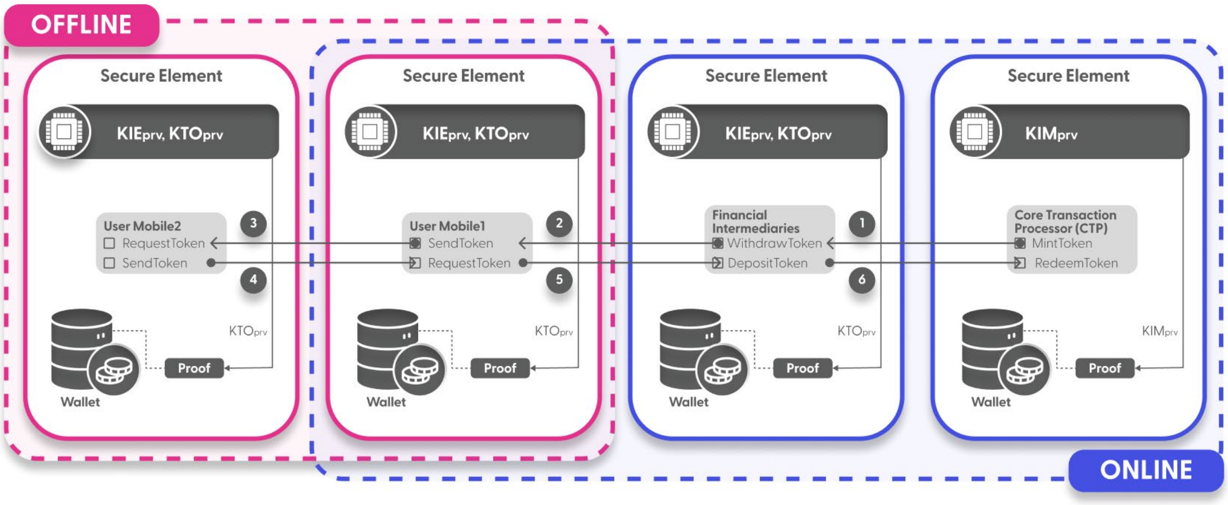
Figure 5: Schematic of t5: Ancestors of token t5 which are formed by mint, transfer, merge, and merge operations


proof =
Sign(msg=t2, key=KIMprv)
proof =
Sign(msg=t3, key=KIMprv)
proof =
Sign(msg=t0, key=KIMprv)
proof =
Sign(msg=t0, key=KTO_t0prv)
proof =
[Sign(msg=t1, key=KTO_t1prv),
Sign(msg=t2, key=KTO_t2prv)]
proof =
[Sign(msg=t3, key=KTO_t3prv),
Sign(msg=t4, key=KTO_t4prv)]

Offline Token Lifecycle

The lifecycle of an offline token includes token mint, transfer, redeem, withdraw, recharge, and deposit phases or events. The lifecycle starts in the CTP, where a token issuer's minting private key (KIM_{prv}) signs a freshly minted offline token. This digital signature is embedded in the token's proof field. Any entity in the offline token-based payments ecosystem can verify the signature using the KIM_{pub}. The token minting can happen only in the CTP as illustrated in Figure 6. The CTP keeps track of all the tokens it has minted. The KIM_{prv} is stored in the secure element, which could be a cloud-hosted hardware security module (HSM) or some form of online secure vault.

Figure 6: Offline token mint and transfer flows



In our model, a financial institution (FI) would send a mint request to the CTP using the API MintToken (depicted as step 1 of Figure 6). The MintToken API would convert the FI's online account balance into offline tokens. There is also a reverse flow that enables an FI to convert excess offline tokens back to an online account balance. This can be accomplished using the RedeemToken API, as shown in step 6 of Figure 6. Because the CTP tracks all the minted tokens, it can check whether a token presented for redemption has been previously redeemed or not.

Once FIs have offline tokens in their offline wallets, users can request offline tokens using the WithdrawToken API. This is shown in step 2 of Figure 6. WithdrawToken converts the user's online account balance into offline tokens. The FI's offline wallet would simply check whether the user has the required account balance, then transfer the tokens to the user's wallet. A reverse flow, the DepositToken API, enables a user to convert excess offline tokens in their phone's wallet into an online account balance. This is illustrated as step 5 of Figure 6.

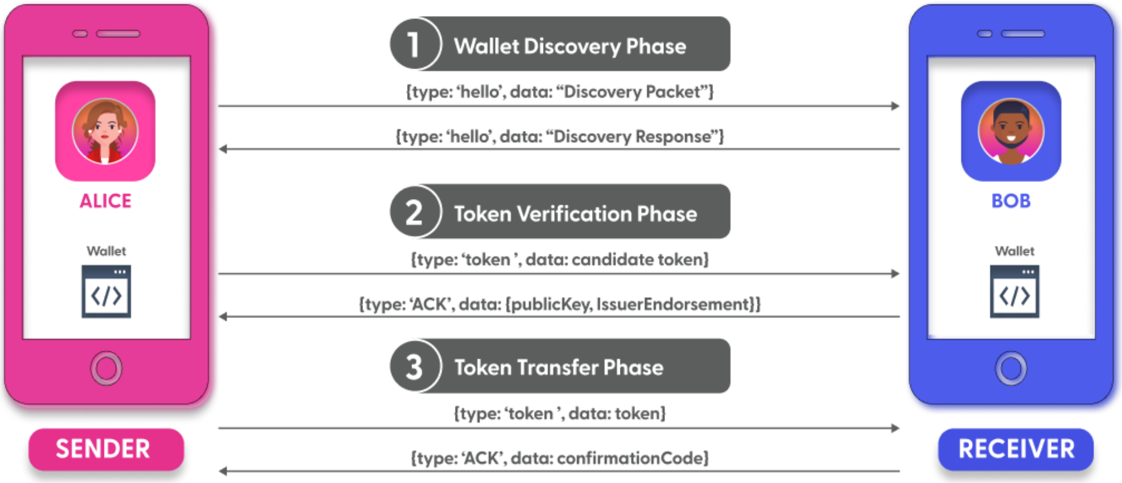
Users can only convert between online and offline tokens while connected to the internet. To access offline tokens in an emergency, they would need to either have pre-loaded their tokens as part of their preparation plan or come into physical proximity with a mobile internet hotspot.[11]

Once the user's wallet has offline tokens, the user does not need an internet connection to spend or transfer the tokens to another party. As depicted in steps 3 and 4 of Figure 6, two wallets can request or send offline tokens even when there is no internet connection. However, steps 1, 2, 5, and 6 require an online connection or internet access.

Offline Token Transfer Protocol

The offline token transfer is represented in steps 3 and 4 of Figure 6. To transfer an offline token, the sender and receiver's wallets go through a three-phase offline protocol, as described below and illustrated in Figure 7:

Figure 7: Offline token transfer protocol to nearby offline wallet


ALICE
Wallet
SENDER
1 Wallet Discovery Phase
{type: 'hello', data: "Discovery Packet"}
{type: 'hello', data: "Discovery Response"}
2 Token Verification Phase
{type: 'token', data: candidate token}
{type: 'ACK', data: {publicKey, IssuerEndorsement}}
3 Token Transfer Phase
{type: 'token', data: token}
{type: 'ACK', data: confirmationCode}
BOB
Wallet
RECEIVER

1. **Wallet Discovery:** In the wallet discovery phase, the sender's wallet scans and finds a nearby wallet, with a proximity of less than one foot, and executes a challenge-response protocol guaranteeing both wallets have a valid, issuer-endorsed SE. If either wallet does not have a valid SE engaged, the connection fails. This protocol ensures that an "innocent bystander," or unintended wallet, does not connect automatically and inadvertently receive the token. The users-in-the-loop mechanism requires both the sender and receiver to verify a 6-digit code generated by both the wallets and prevents any form of man-in-the-middle (MITM) attacks.

2. **Token Verification:** Once the two wallets are connected, the sender must prove to the receiver that it has a valid offline token with the expected value. In the token verification phase, the sender forwards its candidate token for the receiver to verify. If the receiver accepts the token as authentic, the receiver responds with its wallet address, which includes the KTO_{pub} and $\text{endorseKTO}_{\text{pub}}$. The sender then verifies the receiver's $\text{endorseKTO}_{\text{pub}}$ and modifies the candidate token such that the ownership key (KTO_{pub}) and endorsement fields are filled with the received KTO_{pub} and the $\text{endorseKTO}_{\text{pub}}$. Then, the receiver signs the modified token using SignOnce and embeds its signature into the token's proof field. Should an adversary use a previously leaked key to sign their KTO, our prototype's Key Provisioning Applet maintains a secure channel with the issuer's cloud to refresh the KIE_{prv} and prevent counterfeiting and double-spending.
3. **Token Transfer:** In this phase, the sender transfers the token to the receiver. The receiver then verifies the token history. After successful verification, the receiver generates a confirmation code and sends the code to the sender. The sender would then check whether it can generate the same confirmation code on its own. If the sender is able to, the transfer is considered successful. This constitutes the final phase in the three-phase offline protocol.

Potential Future Work

The following section explores future work that could enhance offline payment functionality. These areas include:

1. **Researching additional security feature configurations:** Offline payment functionality could be further configured based on product and policy decisions to limit counterfeiting risks, such as setting transaction amount limits, balance limits, hop counts (or token ancestry), or remote SE management.

2. **Verifying end-to-end feasibility:** Future work could also focus on integration efforts among FIs, particularly related to the issuing and redemption processes. This could provide a working model to clarify how FIs could request and reserve the optimal amount of offline tokens to have on-demand for future use. This verification and testing could be expanded by exploring interoperability with current payment rails, stablecoin networks, or other offerings. Given that the primary use case would be emergency scenarios, understanding the most efficient way to make offline payment functionality accessible for widespread use in a crisis could add to its feasibility analysis.

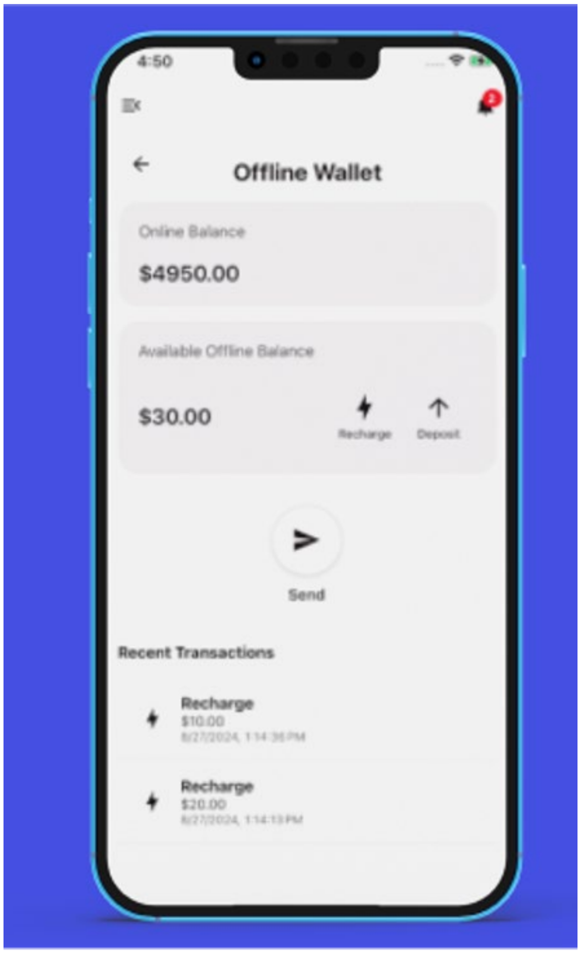
3. **Exploring the role of merchants:** Merchants could be key to amplifying the widespread, positive impact of offline payment functionality. Although our use case focused on person-to-person transfers, future work could explore how merchants might facilitate adoption, particularly if they could provide a user experience comparable to existing software-based point-of-sale or checkout experiences.

4. **Examining operational resiliency in greater depth:** Offline functionality could lead to increased operational resiliency for payments rails. Future work could include expanding operational testing of basic offline payment functionality, extending the amount of time a user can be offline to minimize transaction delays, and exploring contingency experiences in the event of a power or network failure or outage. The testing results could inform product roadmaps or current offerings for existing payments rails and strengthen end-user confidence in payment systems.

Sample Offline Wallet User Interface

Although our offline protocol is described in detail above and may appear complex, the offline wallet user experience could be as simple as depicted in Figure 8 below, which includes the view of a user's mobile wallet app. In practice, a user would simply select an offline wallet option in their smartphone wallet and seamlessly recharge, deposit, or send offline tokens.

Figure 8: Offline wallet running in iOS device



Offline Wallet

Online Balance
$4950.00

Available Offline Balance
$30.00

Recent Transactions

Recharge
$10.00
8/27/2024, 1:14:36 PM

Recharge
$20.00
8/27/2024, 1:14:13 PM

The “Online Balance” section in the wallet screen shows the user’s online balance. The “Available Offline Balance” shows the total value of the offline tokens in the wallet. In this case, the wallet has an online account balance of $4,950 and $30 in offline token value. If the user wants to increase the offline token value by $100, the user could use the Recharge button to do so. After the recharge, the user’s online balance will be $4,850 and offline token value will be $130. Pressing the “deposit” up arrow would enable the user to decide how much to deposit. For example, a deposit request of $50 would increase the online account balance to $4,900 and decrease the total offline token value to $80.

Concluding Thoughts: Addressing the Offline to Cash Comparison

Based on our ongoing experimentation, collaboration, and concurrent threat assessments, we believe our model balances the requirements to make a privacy-protected, intermediated, and widely transferable system with the safety standards that consumers have come to expect from cash and electronic payment services. By leveraging applied cryptography, secure channels over existing data transport methods, and secure hardware on readily accessible smartphones, our work could limit the incidence of a potential attack while containing the broader impact, or blast radius, for the user and surrounding communities.

Our research demonstrates that although offline payment functionality may present some unique risks, offline payments can be as secure as other payment solutions when state-of-the-art hardware, software, and cryptographic protections are utilized. Our model could provide an option that considers the potential compromises necessary given the vulnerabilities inherent to offline transactions. When we tested our assumptions by exploring vulnerabilities synonymous with cash payments, not only were we able to meet the safety standards and primary functions of cash, but we also provided additional ways to mitigate the impact of pertinent attacks.

Our offline model could prevent some of the vulnerabilities associated with cash. The anonymity and lack of historical data available for physical cash transactions makes common crimes like counterfeiting and under-the-table theft less detectable, more difficult to trace, and harder to measure. Although cash is a trusted payment solution, it is less scalable in the event of an emergency. Through our proprietary, operating system-agnostic protocol, our work could provide optionality for policymakers and enable users to make offline digital transactions with confidence.

References

Aboulaiz, Laila, Bunmi Akintade, Hamzah Daud, Monique Lansey, Megan Rodden, Lucas Sawyer, and Matthew Yip. 2024. "Offline Payments: Implications for Reliability and Resiliency in Digital Payment Systems," FEDS Notes. Washington: Board of Governors of the Federal Reserve System, August 16. https://doi.org/10.17016/2380-7172.3456.

Foster, Kevin, Claire Greene, and Joanna Stavins. 2025. “2024 Survey and Diary of Consumer Payment Choice.” Federal Reserve Bank of Atlanta Research Data Report, no. 25-1. https://www.atlantafed.org/-/media/documents/banking/consumer-payments/survey-diary-consumer-payment-choice/2024/sdcpc_2024_report.pdfl.

Judson, Ruth. 2024. “Demand for U.S Banknotes at Home and Abroad: A Post-Covid Update,” International Finance Discussion Papers 1387. Washington: Board of Governors of the Federal Reserve System, https://doi.org/10.17016/IFDP.2024.1387.

[1] We also thank the Federal Reserve System’s Money and Payments Research and Development team for their contributions to this work.
[2] Aboulaiz et al. (2024).
[3] Foster, Greene and Stavins 2024.
[4] Our SignOnce operation is distinct from industry convention in that it is not merely signing something once. Instead, we apply this process to the wallet private key itself, which is maintained in the secure element.
[5] For more information on Federal Reserve joint account guidance, see Final Guidelines for Evaluating Joint Account Requests.
[6] The likelihood of your phone’s applications being infected with malware depends upon your operating system and the source of your applications.
[7] A secret store is a secure location or vault for sensitive data such as API tokens and keys, passwords, certificates and other sensitive data.
[8] The SE is not currently standard on smartphones. The SE manufacturing provisioning process could be done in collaboration with the SE vendor.
[9] An atomic action is one that effectively happens all at once. An atomic action cannot stop in the middle; it either happens completely, or it doesn't happen at all. No side effects of an atomic action are visible until the action is complete. See Oracle’s Java Concurrency courses https://docs.oracle.com/javase/tutorial/essential/concurrency/ for more information.
[10] In a production-grade prototype, we would define a long ancestor chain as one with more than 100 ancestors.
[11] Examples of mobile internet hotspots include portable, satellite-enabled internet stations and first responder command centers.