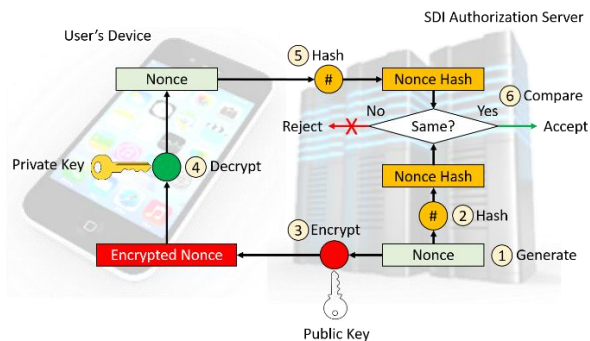


Secure Digital Identity

Genesis

The idea of the Secure Digital Identity (SDI) was inspired by the Health Insurance Portability and Accountability Act (HIPAA) enacted in 1996, which mandated that the patient, not the healthcare provider, was the sole owner of their medical record. The patient was supposed to be in control of who could see that record. SDI would provide robust authentication of the patient/owner of the record, who would decide who could have access to it. As the basis for strong authentication, the Public Key Infrastructure (PKI) was chosen for two reasons: it's nearly impossible to break the proof provided by the private key and the nonrepudiation it affords.

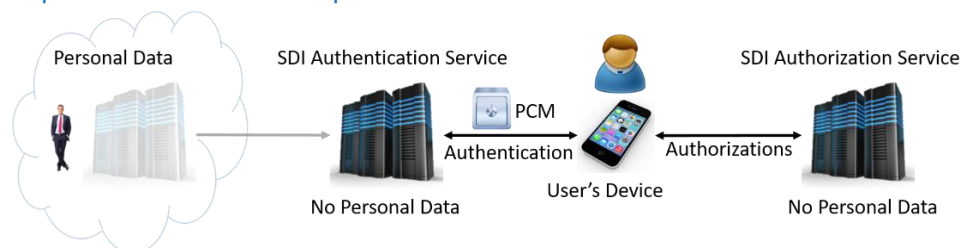
Functional Concept



1. The SDI authorization server generates a nonce
2. Hashes the nonce and caches it
3. Encrypts it with the user's public key and sends it to the user's device
4. The user's device decrypts the received encrypted nonce with the corresponding private key and sends it back to the SDI authorization server
5. The SDI authorization server hashes the received decrypted nonce
6. And compares against the cached nonce hash. A match means identity approved; if not, rejected

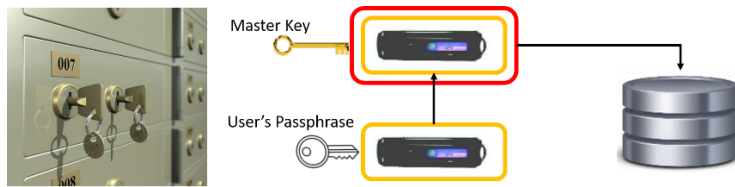
SDI provides security assertions based on verification of a nonce, bundled with a directive that contains essential information required by the transaction, such as payment confirmation or rejection, or a command for a machine to act, like unlocking the door or starting the engine. SDI does not transfer any business data, but it assists the service in verifying the sender or recipient.

Operational Concept



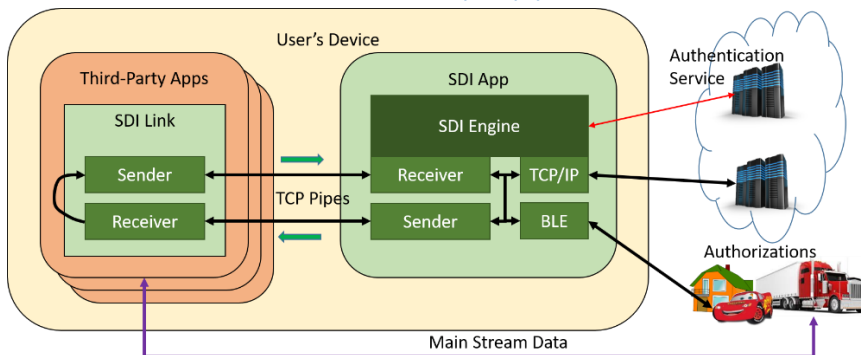
At a higher level, SDI consists of two services and the user's device hosting the SDI app between them. To operate, the user must authenticate with the SDI authentication service to obtain their Personal Cryptographic Matter (PCM) digital container, which contains the necessary data/keys to perform authorizations across multiple services. The PCM container comes encrypted with the user's passphrase, which the user must provide to unlock it. The process of authentication is very similar to what is presented in the Functional Concept, where the user must have a private key to decrypt the nonce. All in all, SDI is already a two-factor authentication: know the passphrase (what you know) and have the key (what you have) verified by approved services, besides the ability to unlock a device with the SDI app.

PCM Protection Concept



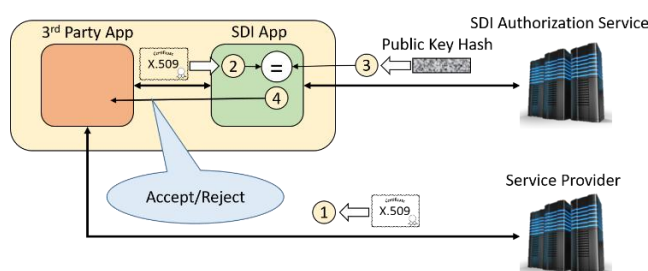
To ensure that mass breaches or insider hacking do not affect users' identities, each PCM is encrypted with the user's passphrase on the user's device and then encrypted by the server with the master key before storing it in the user's record. This process is analogous to a bank safety deposit box.

Collaboration with 3rd Party Apps



The SDI app can work as an independent, standalone authenticator or forward authenticated authorizations triggered by 3rd party apps. A 3rd party app maintains data connectivity directly with a service while triggering the SDI app whenever the user's verification is required, like with ZTA (Zero Trust Architecture) systems.

Dynamic Certificate Pinning



1. The 3rd party app retrieves the TLS X.509 certificate for a site it is connected to
2. And forwards it to the SDI app with a request for verification
3. The SDI app retrieves the hash of that certificate from the SDI authorization, which should be configured to have it
4. Compares the hashes and returns accept/reject to the 3rd party app

Static pinning of certificates is not simple, especially with their high rotation rate. The SDI service can be configured to do it dynamically on demand. A 3rd party app can retrieve the certificate for a site it is connected to, forward it to the SDI app, and receive a quick decision, accepted or rejected, and thus avoid sending sensitive information to a spoofing server. The SDI app receives the hash of a public key from the SDI authorization service encrypted with its registration public key, so the SDI app is the only one that can decrypt it. The SDI app can also verify its own connection the same way if configured.

Summary

The SDI platform can be characterized as follows:

1. Two-factor authentication in one: what you know (the passphrase to decrypt PCM), what you have (the authorization private key to access a resource)
2. No exchange of secrets or personal data: communication is with anonymized identifiers (hashed concatenation of an email and a secret key) and a nonce (a big number used once)
3. Single Sign-On (SSO): the user needs one authentication to gain access to multiple services/sites, and the authentication is without sending a secret known to both sides (e.g., a password). Instead, a nonce is exchanged, and it changes with every transaction
4. Authorizations are fully password-less: authorization is with the exchange of a nonce
5. Ready for post quantum era: uses 256-bit nonce, SHA 256 hashed, AES 256, and ECC up to 521 bits NIST or 512 bits Brain Pool
6. Device independence: all users' devices share the same set of keys
7. No re-registration security hole: once the user registers with a site, no more registrations can be performed
8. Dynamic certificate pinning: certificates arrive in two different paths
9. The kill switch on both services: authentication and authorization. On the SDI authentication service, the user needs to log in with a panic mode (from a different device) to trigger the temporary panic state until the next normal login. On the SDI authorization, the hosting service can permanently disable the account without removing it. The account is dysfunctional, and nobody can access it, including the account owner. Re-registration is not possible either.