# **Summary of EncrypSSI: A User-Centric SSI-Based Account Authentication System**

**Introduction:**  
EncrypSSI is a Self-Sovereign Identity (SSI) system tailored for secure, decentralized account-based authentication. EncrypSSI was developed to address the growing concerns over data breaches caused by the centralized storage of account credentials by organizations. These breaches expose sensitive user information, leading to financial and personal security risks. EncrypSSI reimagines account-based authentication by allowing users to own and control their credentials, shifting from centralized storage to a decentralized, self-sovereign model. By leveraging Verifiable Credentials (VCs), Decentralized Identifiers (DIDs), and blockchain technology, EncrypSSI enhances security, privacy, and user autonomy in account management.

**Core Components of EncrypSSI**

**1. Verifiable Credentials (VCs):**

VCs are the cornerstone of the EncrypSSI system, securely representing account credentials in a privacy-preserving manner.

* **Metadata Only:**
  + VCs do not store sensitive data like usernames or passwords.
  + Instead, they contain metadata such as:
    - User DID
    - Account ID
    - Issuer signature
    - VC unique identifier
    - Date of issuance
* **Verification Processes:**
  + **Ownership:** The system ensures the user presenting the VC owns and controls the associated DID.
  + **Authenticity:** Verifies the VC was issued by the application using the issuer’s digital signature.
  + **Integrity:** Ensures the VC has not been altered by comparing its hash against a blockchain-stored reference.
* Upon successful verification, the user is seamlessly logged into their account.

**2. Decentralized Identifiers (DIDs):**

DIDs are unique, blockchain-anchored identifiers that establish a link between users and their credentials without exposing personal data.

* Each DID is associated with a cryptographic key pair, enabling secure verification of user ownership.
* DIDs are integral to verifying the user’s identity during account login.

**3. User Wallet:**

The **EncrypSSI Wallet** is the central hub for managing DIDs and VCs.

* **Local Storage:**
  + VCs and DIDs are securely stored on the user’s device, eliminating reliance on centralized servers.
* **Secure Interaction:**
  + During authentication, the wallet presents the correct VC and cryptographic proofs to the verifier.
* **Lifecycle Management:**
  + The wallet allows users to view, organize, and manage their credentials effortlessly.

**4. DID Management Features:**

EncrypSSI provides advanced DID management capabilities:

* **Creation:** Users can generate multiple DIDs in their wallet and assign names for easy identification.
* **Export and Import:**
  + DIDs and associated VCs can be exported in a standardized JSON format and imported into other wallets or devices.
  + DIDs generated by other wallets can be seamlessly imported into EncrypSSI user wallet.
* **Deletion:** Users can remove or permanently delete DIDs, along with their associated VCs.

**Account Registration Process**

1. **DID Selection:**
   * Users select an existing DID from their wallet to associate with the account they are registering.
   * The DID is verified to belong to the user by the application (see ownership verification below).
   * Depending on the application, it may also ask users for additional details such as contact details: email address / phone number.
2. **VC Issuance:**
   * The application (e.g., TikTok, Netflix) issues a Verifiable Credential (VC) to the user.
   * The VC contains metadata such as the user’s DID, account ID, issuer signature, and a unique identifier.
3. **VC Storage:**
   * The user securely stores the VC in their wallet.
4. **VC Hashing:**
   * A cryptographic hash of the VC is generated and stored on the blockchain alongside the VC’s unique identifier to ensure integrity.

**Login and Verification Process**

**Step 1: VC Presentation:**

* The user selects the appropriate VC from their wallet and presents it to the application.

**Step 2: Ownership Verification:**

* The application verifies that the user owns the DID associated with the VC by:
  1. Sending a one-time random **nonce** to the user.
  2. The user’s wallet hashes the nonce, signs it with the private key linked to the DID, and sends back the signed hash.
  3. The application retrieves the user’s **DID Certificate** from the blockchain, containing the public key.
  4. Using the public key, the application decrypts the signature, verifies the hash, and confirms the user’s ownership.

**Step 3: Authenticity Verification:**

* The application verifies the issuer’s signature in the VC, confirming it was issued by the application.

**Step 4: Integrity Verification:**

* The application hashes the received VC and compares it to the hash stored on the blockchain. A match confirms the VC has not been tampered with.

**Step 5: Login Confirmation:**

* Upon successful verification, the user is granted access to their account.

**Core Security Features**

1. **Blockchain Integration:**
   * **DID Certificates:** Immutable records of user DIDs, enabling decentralized verification.
   * **VC Hashing:** Ensures the integrity of credentials by storing cryptographic hashes on the blockchain.
2. **Privacy-Preserving Design:**
   * VCs contain only metadata, avoiding exposure of sensitive user data.
3. **Robust Cryptographic Verification:**
   * Ownership, authenticity, and integrity of VCs are verified using cryptographic methods.
4. **User-Centric Control:**
   * The wallet empowers users to manage their credentials independently, including creating, exporting, importing, and deleting DIDs and VCs.

**Why EncrypSSI Stands Out**

* **Passwordless Authentication:** VCs store no passwords, enhancing security and reducing vulnerabilities.
* **Decentralization:** Eliminates reliance on central servers for user identity and credential storage.
* **Usability:** The wallet streamlines credential management, making SSI accessible and convenient.
* **Security and Privacy:** Ensures user ownership and data integrity while protecting sensitive information.