Why SafePwd.net ?

As a full remote IT team at beginning, we **DSYS LTD**, a georgian based company needed a password manager to conduct our(s) activit(ies).

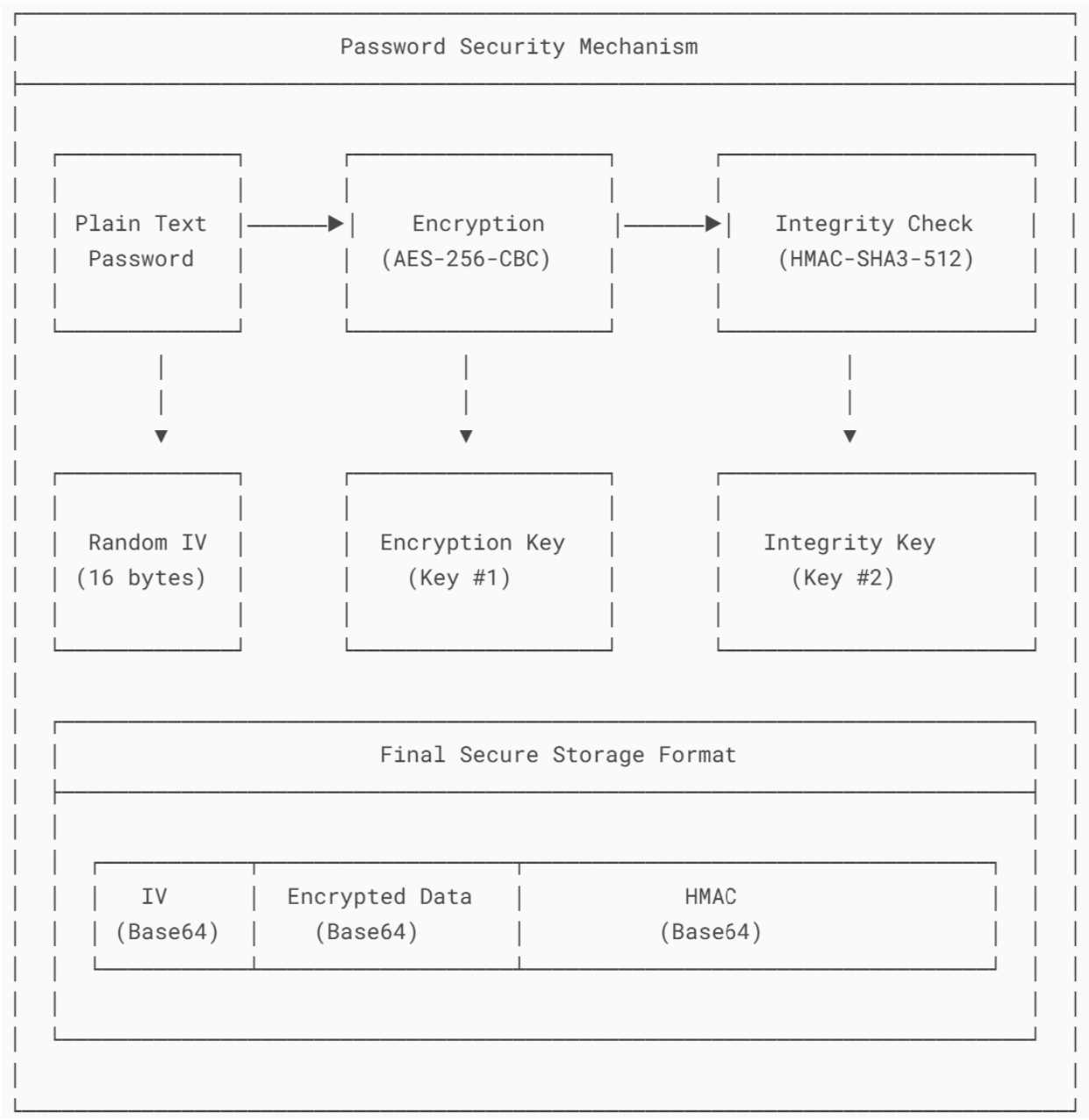
We created the first safepwd.net to allow us to exchange securely between us. We tested securesafe.net which was too slow to be honest ; with 500 password this swiss apps was totally struggling to give back results quickly, We used also in a corporate account the known lastpass.com : too expensive with no flexible plan and a support quite rude during our(s) contact(s) then we decided to create our own tool “for us”.

After used extensively and internally the message crypted system, we designed the password manager to get out this 2 last tool and we are now here for you

1. **How your password are secured ?**

The application secures passwords using a dual-layered mechanism that combines encryption with integrity verification. It begins by generating two distinct cryptographic keys—one dedicated to encrypting the password and the other for verifying its integrity. The AES-256-CBC algorithm is employed to encrypt the plain password using the first key along with a securely generated initialization vector (IV). This process ensures that the encrypted data remains unreadable without the correct key and IV, while the randomness of the IV prevents patterns from forming even when encrypting identical passwords.

Following encryption, the output is further secured by generating a hash using HMAC with the SHA3-512 algorithm and the second key. This hash acts as a cryptographic seal that detects any tampering. The final encrypted output is a base64-encoded string containing the IV, the HMAC, and the encrypted password. By assigning separate keys for encryption and integrity, the application adheres to robust cryptographic practices, ensuring both confidentiality and protection against unauthorized modifications when storing or sharing passwords.

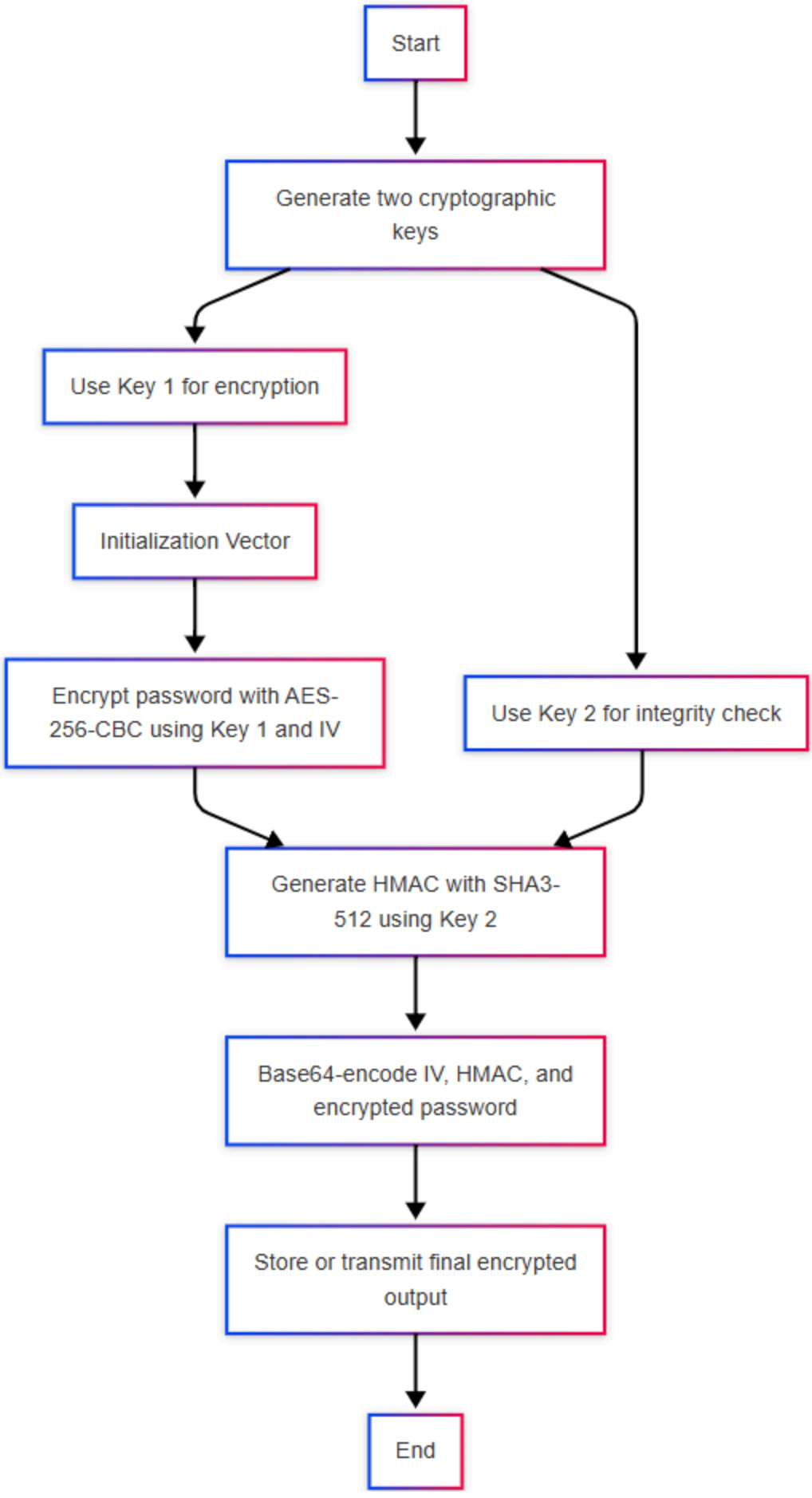

Password Security Mechanism
Plain Text Password
Encryption (AES-256-CBC)
Integrity Check (HMAC-SHA3-512)
Random IV (16 bytes)
Encryption Key (Key #1)
Integrity Key (Key #2)
Final Secure Storage Format
IV (Base64)
Encrypted Data (Base64)
HMAC (Base64)

1. **Input:**
    - The plain-text password is the input to the system.
2. **Encryption (AES-256-CBC):**
    - A **random IV (Initialization Vector)** is generated (16 bytes for AES-CBC).
    - The password is encrypted using **AES-256-CBC** with **Key #1** and the IV.
    - This ensures confidentiality by making the ciphertext unreadable without the key.
3. **Integrity Check (HMAC-SHA3-512):**
    - The encrypted data is hashed using **HMAC-SHA3-512** with **Key #2**.
    - This creates a cryptographic seal to detect tampering.
4. **Final Secure Storage Format:**
    - The **IV**, **encrypted password**, and **HMAC** are concatenated and **Base64-encoded**.
    - This ensures safe storage/transmission without corruption.

ey Security Features:

- **Separate Keys for Encryption & Integrity** → Prevents key reuse vulnerabilities.
- **Random IV for Encryption** → Ensures identical passwords produce different ciphertexts.
- **HMAC for Tamper Detection** → Protects against unauthorized modifications.

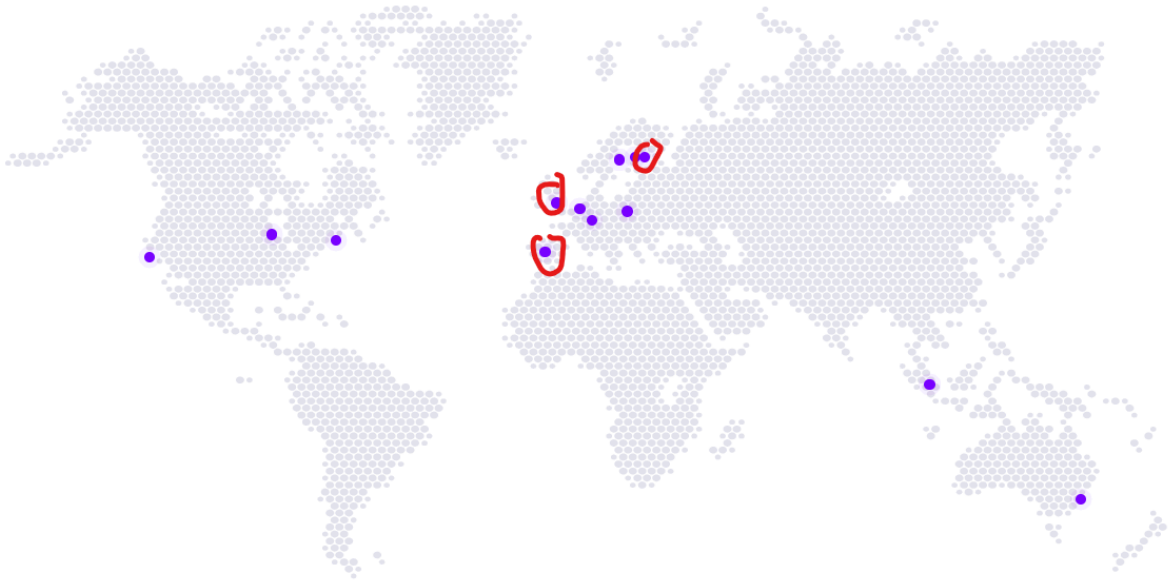
Another representation using Flowchart with Mermaid :


Start
Generate two cryptographic keys
Use Key 1 for encryption
Initialization Vector
Encrypt password with AES-256-CBC using Key 1 and IV
Use Key 2 for integrity check
Generate HMAC with SHA3-512 using Key 2
Base64-encode IV, HMAC, and encrypted password
Store or transmit final encrypted output
End

2. **How do we use redundancy**

Technically the server is “closed” to external request which is the port needed to connect to the website and access are strictly limited to

We are hosted by **Upcloud**, a **Finnish** based company, one the best platform for cloud hosting in Europe, in **Madrid (Spain)** as first node, second node will be deployed in Finland in Helsinki and third node in London (Uk) while we will deploy it on all node showed in the map for maximum redundancy :



3. **How do we secure your password. What can you lose ?**

You can lose access to the password during a period - we can’t guarantee if won’t be the case ever - but you won’t lose password themselves and you will be able to get them back to you asap

A cold storage is constantly “saving” your password to a third party server in our internal(s) network(s) which has no public access in London and Helsinki. Then your passwords are stored in Madrid, London and Helsinki at the same time. Users can only access to **Madrid** endpoint, London and Helsinki are closed to external traffic and only accessible by DSYS LTD administrators.

The recovery procedure will be to take this cold storage to re-instantiate a full active node in Madrid from London et/Or Helsinki. We are constantly checking if there is a difference between the 3 nodes.

Noted that the hard drive itself is encrypted too on top of the encryption we are making by upcloud which is keeping this keys **in this 3 nodes.**

Then your password are encrypted 2 times on our side and one time on the UpcLoud side. To encrypt it, a “robber” should have access to 2 sets of keys and one set of keys is not in our(s) server(s).

4. **Tracking**

We won’t put any third party tracking on these apps .We are excluding all non-EU companies like the US one and Chinese one. No external JS are loaded at all and won’t implement any third party tools.