# 1 Security Analysis

We give security definitions and briefly demonstrate the security of SGX-Cube under two types of attacks: offline and online attacks.

## 1.1 Data Confidentiality under Offline Attacks

As the records of credential databases are encrypted by secret keys, attackers cannot derive any sensitive information from encrypted records without knowing the keys. To demonstrate the data confidentiality of entire databases under offline attacks, we adopt the real world versus ideal world formalization [1] to define the column confidentiality under offline attacks. It is parameterized by a stateful leakage function $\mathcal{L}_1$ describing what information leaks in the protocols. More precisely, we define two games $Real_{\mathcal{A}}$ and $Ideal_{\mathcal{A}}$ with a simulator $\mathcal{S}$ [2] and an adversary $\mathcal{A}$. The simulator $\mathcal{S}$ can simulate real protocols and data using a leakage collection, and the adversary $\mathcal{A}$ has the server's view and can interact with real (or simulated) protocols. If $\mathcal{A}$ cannot distinguish the simulated column data from the real column data, then we can say the column achieves $\mathcal{L}_1$-confidentiality under offline attacks. Note that the columns of usernames, passwords, and user information are encrypted by the cryptographic tools HMAC and AES. Therefore, if the adversary $\mathcal{A}$ has finite computational resources and has not held the secret subkeys of each column, it cannot launch chosen plaintext attacks to distinguish the simulated column data from real column data.

– $\mathbf{Real}^{\lambda}_{\mathcal{A}_1}(\mathbf{k})$: First, $\mathcal{A}_1$ chooses users to send a number of registration requests. Then, $\mathcal{A}_1$ obtains a set of encrypted data in the column $\lambda$ from the SGX enclave. Finally, $\mathcal{A}_1$ outputs a bit $b \in \{0, 1\}$.
– $\mathbf{Ideal}^{\lambda}_{\mathcal{A}_1,\mathcal{S}}(\mathbf{k})$: First, $\mathcal{S}$ simulates encrypted data in the column $\lambda$ with the leakage function $\mathcal{L}_1$. Finally, $\mathcal{A}_1$ outputs a bit $b \in \{0, 1\}$.

***Definition 1 (Column Confidentiality under Offline Attacks)*** *Let $neg(k)$ be a negligible function. A column $\lambda$ achieves $\mathcal{L}_1$-confidentiality under offline attacks, if for any polynomial-time attacker $\mathcal{A}_1$, there exists a simulator $\mathcal{S}$ such that the following equation holds:*

$$|Pr[Real^{\lambda}_{\mathcal{A}_1}(k) = 1] - Pr[Ideal^{\lambda}_{\mathcal{A}_1,\mathcal{S}}(k) = 1]| \leq neg(k) \tag{1}$$

Based on the above definition, we give the following theorem and sketch a proof to demonstrate the data confidentiality under offline attacks.

***Theorem 1*** *If HAMC_SHA256 is a secure pseudo-random function and AES is a secure symmetric encryption scheme, then the username column, password column, and information column achieve $\mathcal{L}_1$-confidentiality, where the leakage function collection $\mathcal{L}_1$ is defined as $L_1(D) = \{(C_u, C_p, C_i)|(C_u, C_p, C_i) \in D\}$*

Note that in this theorem, $D$ represents the credential database, and $(C_u, C_p, C_i)$ represents a record of three columns.

***Proof 1*** Here, we construct a simulator $\mathcal{S}$ with the leakage function $\mathcal{L}_1$ to prove column confidentiality. If $\mathcal{S}$ can simulate encrypted data in a column that $\mathcal{A}$ has a negligible advantage to distinguish from real data, then we can say the column achieves $\mathcal{L}_1$-confidentiality. First, $\mathcal{S}$ initiates $|D|$ empty list as a credential database. For each record, $\mathcal{S}$ selects three random strings $C_u^*$, $C_p^*$, and $C_i^*$ to simulate encrypted data in the username column, password column, and information column. Particularly, the lengths of ($C_u^*$, $C_p^*$, $C_i^*$) are same to the lengths of $(C_u, C_p, C_i) \in \mathcal{L}_1$. Recall that data in three columns are encrypted by *HAMC_SHA256* and *AES*. If *HAMC_SHA256* is a secure pseudorandom function and *AES* is a secure symmetric encryption scheme, then $\mathcal{A}$ has a negligible advantage to distinguish the simulated database from the real database.

### 1.2 Data Confidentiality under Online Attacks

To demonstrate the data confidentiality of entire databases under online attacks, we also follow real world versus ideal world to define column confidentiality under online attacks. Here, we present two games with a simulator $S$ and a stronger attacker $\mathcal{A}_2$, where $\mathcal{A}_2$ can compromise the server and a subset of users, and it can launch chosen-plaintext attacks to learn ciphertexts.

- $\mathbf{Real}^{\lambda}_{\mathcal{A}_2}(\mathbf{k})$: First, $\mathcal{A}_2$ chooses an honest user to send a request for login or registration. Second, $\mathcal{A}_2$ obtains encrypted data in the column $\lambda$ from the SGX enclave. Finally, $\mathcal{A}_2$ outputs a bit $b \in \{0, 1\}$.
- $\mathbf{Ideal}^{\lambda}_{\mathcal{A}_2,\mathcal{S}}(\mathbf{k})$: First, $\mathcal{A}_2$ chooses an honest user to send a request for login or registration. Second, $\mathcal{A}_2$ simulates encrypted data in the column $\lambda$ with the leakage function $\mathcal{L}_1$. Finally, $\mathcal{A}_2$ outputs a bit $b \in \{0, 1\}$.

According on the above definition, we give the following security theorem and sketch a proof to demonstrate data confidentiality.

***Theorem 2*** *If HAMC_SHA256 is a secure pseudo-random function and AES is a secure symmetric encryption scheme, then the username column does not achieve $\mathcal{L}_1$-confidentiality, but the password column and information column achieve $\mathcal{L}_1$-confidentiality.*

Note that in this theorem, $\mathcal{L}_1$ specifically refers to the leakage function defined in Theorem 1.

***Proof 2*** Here, we construct a simulator $\mathcal{S}$ with the leakage function $\mathcal{L}_1$ to prove column confidentiality. If $\mathcal{S}$ can simulate encrypted data in a column that $\mathcal{A}_2$ has a negligible advantage to distinguish from real data, then we can say the column achieves $\mathcal{L}_1$-confidentiality. In the *Ideal* game, $\mathcal{S}$ simulates encrypted data in three columns as follows.

- *Simulating encrypted usernames.* For each login or registration request from an honest user. $\mathcal{S}$ selects a random string $C_u^*$ to simulate the encrypted username in the request. Note that $\mathcal{A}_2$ can launch chosen-plaintext attacks to query the real encrypted username $C_u$. Therefore, $\mathcal{A}_2$ has non-negligible advantage to distinguish $C_u^*$ from $C_u$.

– *Simulating encrypted passwords and user information.* As encrypted passwords are concealed in the SGX enclave during login requests, $\mathcal{S}$ only simulates encrypted passwords and user information in the registration request. First, $\mathcal{S}$ chooses an honest user to send a registration request. Then, $\mathcal{S}$ selects two random string $C_p^*$ and $C_i^*$ to simulate the encrypted passwords and user information output by the SGX enclave. Recall that the real password $C_p$ and user information $C_i$ are encrypted by *HAMC_SHA256* and *AES*, and their generating keys are related to a unique username. Therefore, if *HAMC_SHA256* is a secure pseudorandom function and *AES* is a secure symmetric encryption scheme, then $\mathcal{A}_2$ has a negligible advantage to distinguish $C_p^*$ and $C_i^*$ from $C_p$ and $C_i$ without generating keys. ■

Theorem 2 shows that attackers cannot derive any information from passwords and user information from credential databases. Although the username can be extracted by chosen-plaintext attacks, the confidentiality of whole databases is still much higher than existing credential databases.

### 1.3 Data Integrity under Online Attacks

Since data can only be manipulated by online attacks, we only demonstrate data integrity under online attacks.

**Data Integrity in Memory.** The data integrity in memory is guaranteed by the security of the SGX enclave. The whole authentication procedure is completed inside the enclave. The authentication results are delivered to the requester from the enclave directly via secure communication channels. The results are not revealed in the server memory and cannot be tampered. Hence, the attacker can't compromise the data integrity in the memory.

**Data Integrity on Disk.** The data integrity on disk means that credential databases cannot be manipulated to authenticate a user without a correct pair of username and password. Here, we consider an attacker $\mathcal{A}_3$ who can corrupt both authentication server and a subset of users. Particularly, $\mathcal{A}_3$ can control a user $u_1$ to generate a password $p_1$ from a known string, and then replace an honest user $u_2$'s password with $p_1$ in the credential database. Next, $\mathcal{A}_3$ may attempt to impersonate $u_2$ by sending the known string as the password. Recall that the HMAC value of each user's password is generated by a unique secret key. Therefore, attackers have a non-negligible advantage to impersonate an honest user if *HAMC_SHA256* is collision-resistant. Specifically, we give the following security theorem to demonstrate data integrity.

***Theorem 3*** *SGX-Cube achieves data integrity on the disk if HAMC_SHA256 and SHA256 is collision-resistant.*

***Proof 3*** First, $\mathcal{A}_3$ can utilize a generating key $K^*$ and a known string $s$ to manipulate an encrypted password $ct_s^* = HMAC\_SHA256(k^*, s)$. Then, $\mathcal{A}_3$ may attempt to send the password $s$ with the username to impersonate a user. On receiving the username and password, the SGX enclave generates the HMAC value $ct_s = HMAC\_SHA256(K_u, s)$ and tests if $ct_s$ equals to $ct_s^*$ to authenticate the user, where $K_u = SHA256(K_p||username)$. Note that $\mathcal{A}_3$ does not hold $K_p$, and each username in our credential database is unique. If *HAMC_SHA256* and *SHA256* is collision-resistant, there is negligible possibility that $K_s$ equals to $K^*$ and $ct_s$ equals to $ct_s^*$. Therefore, $\mathcal{A}_3$ cannot bypass the authentication with non-negligible advantage. ■

## References

1. Curtmola, R., Garay, J., Kamara, S., Ostrovsky, R.: Searchable symmetric encryption: improved definitions and efficient constructions. Journal of Computer Security **19**(5), 895–934 (2011)
2. Lindell, Y.: How to simulate it–a tutorial on the simulation proof technique. In: Tutorials on the Foundations of Cryptography, pp. 277–346. Springer (2017)