Secrets Management

caution

Chainlink Functions is still in BETA. The use of secrets in your requests is an experimental feature that may not operate as expected and is subject to change. Use of this feature is at your own risk and may result in unexpected errors, possible revealing of the secret as new versions are released, or other issues.

Chainlink Functions enables users to fetch data from HTTP(s) APIs and perform custom computation. To enable access to APIs that require authentication credentials, Chainlink Functions allows users to provide encrypted secret values, which can be used when the DON executes a Functions request. These values can only be decrypted via a multi-party decryption process called threshold decryption. This means that every node can only decrypt the secrets with participation from other DON nodes.

These encrypted secret values can either be uploaded to the DON or hosted at a remote URL. Then, a reference to the encrypted secrets can be used when making a Functions request.

Threshold encryption

Using a single private key to decrypt user secrets poses a considerable security risk. If any node with this key becomes malicious or compromised, it can decrypt all accessible user secrets. To address this vulnerability, we have integrated a feature known asthreshold encryption. The characteristics of this enhanced security measure include:

1. Decentralized Nodes Deployment: The system operates on a Decentralized Oracle Network (DON) with N nodes.
2. Master Secret Key Distribution: Instead of centralizing the private key, the master secret key (MSK) is partitioned into shares. Each node within the DON possesses a distinct share of this MSK.

3. The system enforces two thresholds:

4. Liveness: This feature ensures the system's availability and fault tolerance. It can tolerate up to F Byzantine nodes (nodes that can act arbitrarily or maliciously) while remaining operational. The system adheres to the mathematical relationship$3F+1 \leq N$. This means the total number of nodes N must be at least three times the number of faulty or malicious nodes F, plus one more.

5. Decryption security: Considering that the MSK is distributed across multiple nodes, safeguarding it is essential. To recover the master secret key (MSK), an attacker must corrupt a number of K nodes, adhering to the relationship:

$F<K<=2F+1$. 4. Encryption Process: A public key is associated with the MSK. Users use this public key to encrypt their secrets, generating ciphertext.Note: The encrypted secrets are never stored onchain. 5. Threshold Decryption Mechanism: Decrypting any given ciphertext mandates the collaboration of at least K out of N key shares, where:

$F<K<=2F+1$

For decryption to initiate, a minimum of K nodes must concurrently process a user request bearing the identical ciphertext.

Hosting methods

There are two methods for sharing encrypted secrets with the DON:

- DON-hosted (Location.DONHosted): Users send the encrypted secrets (ciphertext) to the DON through the secrets endpoint, as depicted in therequest and receive diagram .Note: To use this capability, one of the subscriptions owned by your externally-owned account (EOA) must maintain a minimum balance. Read theminimum balance for uploading encrypted secrets section to learn more.
- User-hosted (Location.Remote): Some users prefer to manage the storage and lifecycle of their secrets, such as deleting the encrypted secrets once their request is fulfilled. In this setup, users host the encrypted secrets (ciphertext) on a publicly accessible HTTP(s) endpoint and then encrypt the URL with the DON public key before sharing the encrypted URL with the DON.Note: After the request has been addressed, the user is responsible for removing their stored ciphertext.