Encryption Key Management

Last Modified on 02/08/2021 4:13 pm EST

Overview

SAP Cloud Platform Open Connectors utilizes software- and hardware-based encryption to protect sensitive customer data and other sensitive company information from unauthorized access and misuse. The company also considers encryption key “pass phrases” and shared “root administrator credentials” as credentials that should be protected in accordance with Encryption Key Management Guidelines. This document establishes the guidelines to manage encryption keys to meet company risk mitigation goals and regulatory compliance with Payment Card Industry Data Security Standards (PCI DSS). The following guidelines are provided to set a standard for management of the encryption keys. These keys are vital to the protection of transactions and stored data. Management will be deployed at a level commensurate with the critical function that these keys serve.

Utilized Encryption

We have deployed the following critical applications and systems that utilize encryption:

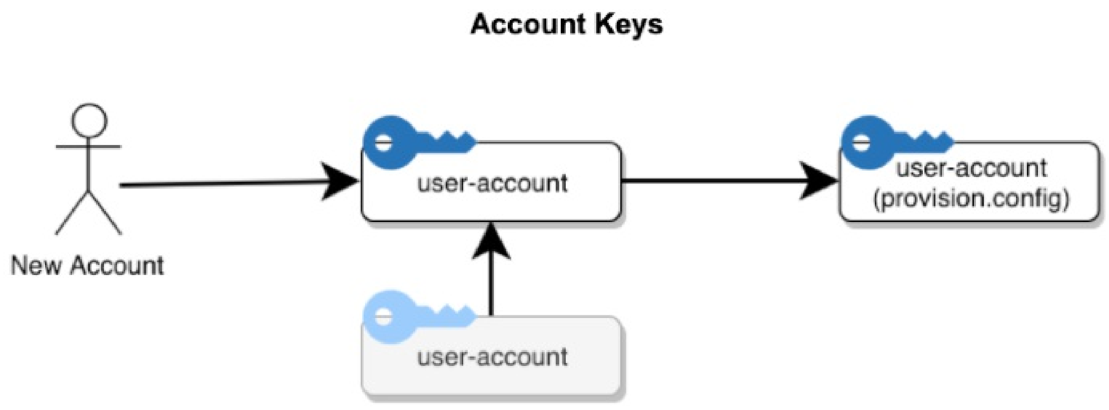
Application	Function	Type Encryption	Vendor
Encrypted network tunnels	VPN	256-bit AES	Pritunl
Database - field level		256 bit AES CBC mode; PKCS 5	Postgres
API servers/application servers	SSLPassword hashing	TLS 1.2 SSL with SHA-256 and 2048-bit public keys PBKDF2 (SHA1 + HMAC) for key derivation with 150000 iterations for password hashing	Comodo and AWS
Marketing website	SSL	TLS 1.2 SSL with SHA-256 and 2048-bit public keys	Comodo
Personal use files and laptop encryption	Encryption at rest	All hard drives on company laptops will have FileVault enabled. Only Apple computers are valid for the company.	Apple Filevault
Remote access	Administrators	SSH is available for our VPC for admins only – no user access. 4096-bit public keys and SHA-256 password hashing, with multi-factor authentication required using Google Authenticator.	
Terminal encryption keys for merchants	N/A	Not applicable – SAP Cloud Platform Open Connectors is not a merchant and does not have a merchant ID.	

Encryption Key Lifecycle

There are two main types of encryption key lifecycles: application-level and user-level encryption key lifecycles. Application-level keys are keys that secure data related to the operations and networking of the SAP Cloud Platform Open Connectors products. They are created during as part of the installation of a CE environment or application server, and exist entirely internally to the application. They exist for the lifetime of the environment or server that are are meant to secure, and are deleted only when an environment or server is destroyed.

User-level keys are keys that secure data provided by user of the product, and which may be kept at the user, account, or organization level, and include **Account Keys**, **User Keys**, and **Instance Keys**. They have an internal lifecycle that is attuned to the resource(s) which they are designed to secure. During normal operation, these keys are accessible only via user login credentials (e.g. a password or authorization token), and are created either during the creation of a new user, or when a user creates certain types of resources that contain sensitive information, such as sub-accounts or connector instances. These keys exist exactly as long as the user or resource exists and is in active use, and are deleted when the user or resource is destroyed.

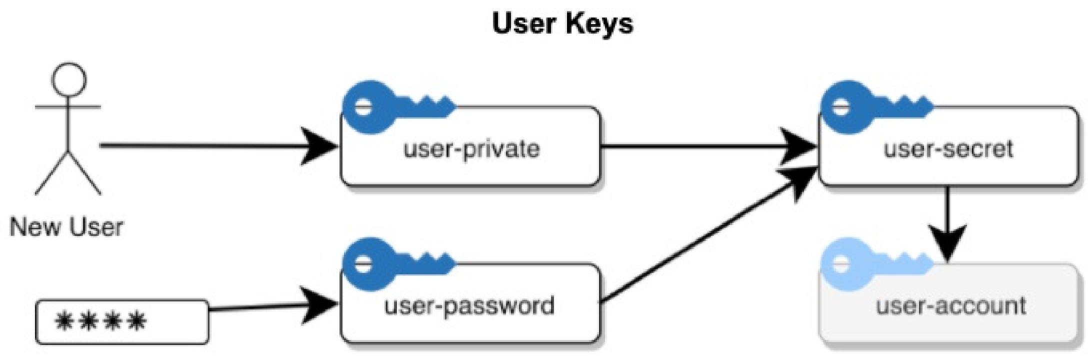
Account Keys



The above figure shows the creation of the "user-account" and "user-account (provision)" keys and their relationship (in these diagrams, an arrow from key A to key B represents a "secured by" or "accesses" relationship, indicating that key B can be accessed from key A). This occurs at the same time as the creation of the account and account structure. Additionally, if the account is created as the sub-account of an existing super account, the super account is given access to the sub-account via the existing super-account key (existing keys are shown in gray). If and when the account is deleted, the "user-account" and "user-account (provision)" keys are destroyed.

The "user-account" key is used to encrypt information any data specific to the account, and the "user-account (provision)" is used to encrypt data that belongs to resources owned by users in the account.

User Keys



The above figure shows the creation of the "user-private", "user-password", and "user-secret" keys, which are used to encrypt user-specific data, and which provide access to that user's existing account. When a user is created in a new account, the account and its keys are created first (as shown in 6.1), and then the user is created.

The "user-password" is created as a unique key generated from the user's password using an industry-standard password derivation function, and no other key is given access to it; therefore, if a user loses their password, that key must be re-generated. When the user supplies a password, the "user-password" key is recreated for that user session, and supplies access to the other user keys.

The "user-private" key is supplied as a backup mechanism. SAP Cloud Platform Open Connectors applications may run in a privileged mode which supplies access to all user-private keys, thereby gaining access to users' sensitive data. This is necessary for off-session tasks, or if (for example), the user loses their password to re-generate the "user-password" key.

The "user-secret" key is the main encryption gateway to a user's data. It is used to encrypt user-specific data, as well as providing access to the the account key for that user's account, giving them access to all resources in that account. These keys exist for the duration of user data, and are destroyed when that user is deleted from the system.

Customer private keys for data encryption are held by Cloud Elements, but only in encrypted form. There are only a limited number of ways that those keys can be unencrypted and used via the user password; this is provided by the user during login to our GUIs. We never save this value, and can’t retrieve it if the user loses it.

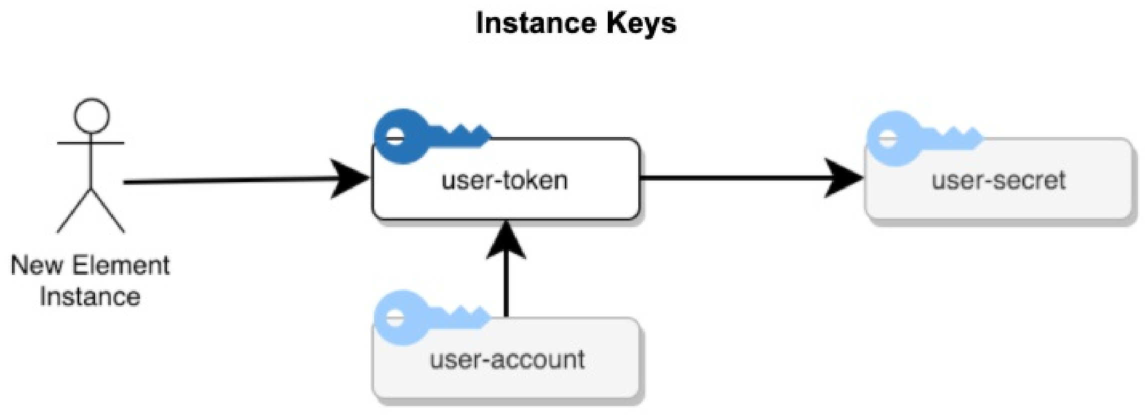
Via the user secret key: This is provided by the user when using our APIs. We don’t save this value, except in encrypted form--it can only be decrypted and shown via one of the other methods (for example, if the user is logged in via their password).

Via any of the user’s element instance tokens: We don’t save this value either, except in encrypted form--same deal for decrypting and showing to the user.

Via our “backup” key: this allows Cloud Elements itself to access a user’s data, in case we have an asynchronous process (such as a formula or bulk job) that needs to use that data while the user is logged out. We also use it to reset the user’s passwords or other keys in case they are lost.

Note that this is only for data saved on our system: instance oauth keys/tokens/passwords, secrets. Any data which is encrypted/saved elsewhere is solely their domain, and we don’t have access to it.

Instance Keys



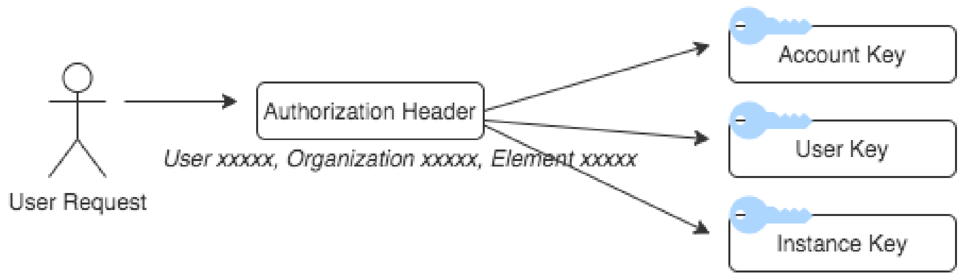
When a user provisions a new connector instance, a user-owned type of SAP Cloud Platform Open Connectors resource, a new "user-token" key is created for that Instance. This is shown in figure 6.3. That key is given access to that user's "user-secret"--and thereby access to the account and all resource data in that account (including the data belonging to itself).

Additionally, the user's account key is given access to the new token, allowing all users in the account to view that new connector instance. (Currently, our product and API only shows a user their own instances, even though, from a security perspective, all instances in their account and sub-accounts are available to them).

The "user-token" key lasts for the lifecycle of the connector instance, and is deleted when that connector instance is deleted.

Multi-Tenant

When an API call is made to SAP Cloud Platform Open Connectors, it requires an Authorization header. This header consists of the above-mentioned keys, which uniquely identifies the user and the connector instance that the user is requesting to access. Authenticate the request by including the User Secret and either your Organization Secret and Connector Instance Token, or both in the Authorization header



On using this Authorization header, SAP Cloud Platform Open Connectors identifies the tenant and executes the request accordingly.

Types of Tokens

There are four different "token" types that are transmitted to users that grant them access to SAP Cloud Platform Open Connectors resources.

User Token

The first is a simple "User Token", which is generated during the user creation process, and which is available to the user when they sign-on to SAP Cloud Platform Open Connectors services, and at any sign-in point later on. This token is sometimes called the "User Secret" in the nomenclature; although the token internally grants access to the "user-secret" key described above (and thus all the user's data), it has no other relationship to it. Users provide the User Secret as part of their "Authorization" header when accessing resources that they own, or which they are allowed to use, using a scheme similar to that of Basic Auth: see https://tools.ietf.org/html/rfc2617#section-3.2.2.

Instance Token

The second token is the "Instance Token", which is available when users create a new connector instance resource, and to any user which owns or has been granted permission to use that instance. This token is essentially an encoded version of the "user-token" key above, and therefore grants access to the instance resource that it is associated with. It also is expected on the "Authorization" header for requests that require access to that specific instance resource.

User JWT Token

The third token type used is the "User JWT". This is a JWT token type (see https://tools.ietf.org/html/rfc7519) which is made available via the UI console for all users, which acts similarly to the "User Token", but additionally includes the various authorization modes for that user. This token can be provided on any user's request using the "Authorization: Bearer" scheme which is standard for such token types (https://tools.ietf.org/html/rfc6750#section-2.1) instead of another scheme.

Temporary User JWT Token

The fourth and final token type is the "Temporary User JWT". Like the "User JWT", it provides access for a given user, but only for a limited time. The Temporary User JWT only grants user access only for the specific use of password reset, and is transmitted to the verified email address of that user upon request.