Unbound Crypto-of-Things

Release Notes

Version 1.9.2103.39335

May 2021

Table of Contents

1. About This Document

This document specifies new features, enhancements, and improvements in the CoT solution that spans Android and iOS mobile devices, CoT server, and optional CoT proxy.

2. CoT Release 1.9.2103.39335

The following changes were made in the **CoT Android SDK**.

2.1. Fixed Issues

- **Authentication Flags**

  An issue was fixed that was causing a problem on some specific devices (such as Xiaomi Mi A1). When doing enrollment, devices with this issue seem to complete successfully, however, when the user tries to use the token it fails (*userAuthenticationRequired* fails). The issue only occurs if KeyGuard is set (through PIN/pattern or fingerprint).

  To fix the issue, when enrolling new keys on an Android device, an authentication key is generated with the following flags disabled: *setUserAuthenticationRequired* and *setUserAuthenticationValidityDurationSeconds*. These flags can be enabled for authentication by using **enableUserAuthenticationCheck()** during token creation. For example:

```
DYSign.getInstance().createSignToken("label", MainActivity.USERNAME, new
DYNoCredentials().enableUserAuthenticationCheck() , params, new
DYSignTokenFactory.DYInitTokenListener() {…}
```

- **Enrollment and Signing Issues**

  When running on iOS devices, signing was not working. In addition, on devices without a passcode, there was an issue causing enrollment to fail.

3. CoT Release 1.9.2103

This release of CoT has the following updates.

3.1. New Features and Enhancements

This release contains the following enhancements:

- **Last Modified Database Update**

  A new column is now in the token table for the CoT database that shows the last modified time of the tokens. This column enables searching for tokens that were not used for a very long period in order to clean up the database.

  - In the table **dy_tokens**, the new column is called **last_modified**.
  - The column type is **TIMESTAMP WITH TIME ZONE**.

- **Device without Passcode**

  You can now use iOS devices that do not have a passcode. Previously, only devices with a passcode could be used for key generation and usage.

  The relevant line as shown in bold in the example must be passed in **initParams** to disable the passcode requirement.

  For example:

```
NSDictionary* initParams = [[NSDictionary alloc] init];
[initParams setValue:[NSNumber numberWithBool:NO] forKey:@"passcode-required"];
BOOL b = [[DYMobile sharedDYMobile] initDYMobileWithURL:[NSURL URLWithString:
SERVER_URL]
    domain:domain
    deviceUID:[UIDevice currentDevice].identifierForVendor.UUIDString
    serverCertificate:certificate
    urlParams:URL_PARAMS
    initParams:initParams
    error:&error];
```

3.2. Fixed Issues

The following issues were fixed in this release.

- Using a specific Android device, CoT failed to generate a key.

3.3. Upgrade Information

The CoT Server must be upgraded prior to upgrading the mobile app.

The new CoT Server is backwards compatible with the previous CoT SDK.

3.3.1. Database Update

For this release, an upgrade script must be run before installing the RPM. This script adds a new column to a database table.

1. Download the scripts archive (*mobile-schema.1.9.2103.39327*).
2. Run the script *upgrade.sql* contained in that archive.
3. Install the RPM in the usual way.

3.4. Documentation

This release has these associated documents:

- CoT Admin Guide
- CoT Developers Guide

4. CoT Release 1.5.1706.12949

This release of CoT has the following updates.

4.1. Fixed Issues

- The upgrade from 1.4.1706 to 1.5.1706.12949 now works.

> **Note**
> This version does not work on the Apple simulator for iOS 13.1, 13.2 or 13.3.

4.2. Documentation

This release has these associated documents:

- CoT Admin Guide
- CoT Developers Guide
- CoT Developers Guide for iOS

5. CoT Release 1.5.1706.12947

This release of CoT has the following updates.

5.1. Fixed Issues

- If a user removed the device passcode, their token was invalidated and new token could not be enrolled.

  Notes:

  When the device passcode is removed, all enrolled tokens become invalid. The user must then re-enroll a new token.

  The CoT SDK must be reinitialized after a new device passcode is set. The user may need to restart the app, depending on the specific application implementation.

5.2. Documentation

This release has these associated documents:

- CoT Admin Guide
- CoT Developers Guide
- CoT Developers Guide for iOS

6. CoT Release 1.5.1706.12940

This release of CoT has the following updates.

6.1. New Features and Enhancements

This release contains the following enhancements:

- **Log generation**
  CoT generates logs every hour so that smaller files are created. Previously, it was generating logs on a daily basis.

6.2. Fixed Issues

The following issues were fixed in this release.

- An issue occurred with key rotation when more than one server was using the database.
- There was a memory leak that was fixed.

6.3. Upgrade Information

The CoT Server must be upgraded prior to upgrading the mobile app.

The new CoT Server is backwards compatible with the previous CoT SDK.

6.4. Documentation

This release has these associated documents:

- CoT Admin Guide
- CoT Developers Guide
- CoT Developers Guide for Android
- CoT Developers Guide for iOS

7. CoT Release 1.5.1706

This release of CoT has the following updates.

7.1. New Features and Enhancements

This release contains the following enhancements:

- **Secondary Database**
  CoT can be configured to use a secondary database. This database is used if the primary database is not available. Subsequently, if the secondary database becomes unavailable, CoT tries to switch back to the primary database.

  > **Note**
  > When using PIM Connector, it is assumed that both databases are sharing the same password.

- **iOS Vendor ID**
  CoT uses an AES key to generate a token on iOS devices. Previously, this token was based on the vendor ID, which under certain conditions could change.
- **Mobile SDK Encryption**
  The CoT mobile SDK encryption was changed to use AES/GCM.
- **Open Source Components**
  Open source components used by CoT were updated to newer versions.

7.2. Fixed Issues

The following issues were fixed in this release.

- COT-162: CoT was failing to get a database connection.
- COT-143: Under certain conditions with a heavy load, the CoT server was not able to communicate with the database.

7.3. Upgrade Information

The CoT Server must be upgraded prior to upgrading the mobile app.

The new CoT Server is backwards compatible with the previous CoT SDK.

7.4. Documentation

This release has these associated documents:

- CoT Admin Guide
- CoT Developers Guide
- CoT Developers Guide for Android
- CoT Developers Guide for iOS

8. CoT Release 1.4.2007

This release of CoT has the following updates.

8.1. New Features and Enhancements

This release contains the following enhancements:

- **CoT dependencies**
  All open source projects used by CoT were updated to the latest versions.
- **CoT smart proxy**
  The smart proxy sample code, as well as the CoT SDK and CoT Server, were enhanced to support the encryption protocol/token. You can now encrypt data on the client, start the decryption on the client, and then get the decrypted value on the smart proxy.
- **mobilecl log files**
  For added security, log files are written to the home directory instead of the *tmp* directory.
- **Server API vs. Client SDK API**
  The CoT Server API is available separately from the Client SDK API (Note that the old API is still available for backwards compatibility).
- **Tokens required for all access**
  Access tokens are now required for all CoT Server REST API calls. The proxy sample code was updated to use the access token (a placeholder was added for it).

8.2. Documentation

This release has these associated documents:

- CoT Admin Guide
- CoT Developers Guide
- CoT Developers Guide for Android
- CoT Developers Guide for iOS

9. CoT Release 1.4.1706

This release of CoT has the following updates.

9.1. New Features and Enhancements

This release contains the following enhancements:

- **CoT refresh API**
  A new API was added to trigger refresh settings on demand. This API removes the need to periodically run a refresh, and as a result, avoids potential performance issues.
- **Performance monitoring**
  Supports the AppDynamics performance monitoring tool.
- **Log files**
  Log files now contain the hostname in the filename. The documentation was updated to reflect that you can create domain specific log files.
- **Android SDK - RSAkey.class**
  The RSAkey class was updated to remove an insecure import of *java.util.random*.
- **OpenShift support**
  CoT can now be run as a container in the OpenShift platform. Refer to the CoT Admin Guide for more information.

9.2. Upgrade Information

These are important notes about upgrading:

- These minimum versions are now required:
  - Oracle Java 1.8
  - Tomcat 8
  - Android 4.4
- This version was specifically tested to work with Oracle 19c.

9.3. Documentation

This release has these associated documents:

- CoT Admin Guide
- CoT Developers Guide
- CoT Developers Guide for Android
- CoT Developers Guide for iOS

10. CoT Release 1.3.1706.34340

10.1. Fixed Issues

The following issue was fixed in this release:

- There was an issue that caused the CoT server database connection pool to ignore updated credentials.

11. CoT Release 1.3.1706.33065

11.1. Fixed Issues

The following issue was fixed in this release:

- There was an issue with the offline OTP refresh. After executing a refresh, if the server payload was not returned to the client, it was causing subsequent refreshes to fail. This issue also caused a synchronization problem between the offline and online refresh. The refresh now works correctly.

11.2. Upgrade Information

When upgrading from a previous version of CoT, the Offline OTP feature needs to be enabled for the relevant domain.

12. CoT Release 1.3.1706.32494

CoT version 1.3.1706 has the following updates:

- The product version number was changed to 1.**3**.1706.
- The driver for Oracle for Java Database Connectivity (JDBC) version 8 is included in the package.

12.1. Fixed Issues

The following issues were fixed in this release:

- The encryption test was failing after upgrading. It now works correctly.
- The sample for Offline OTP for iOS was missing files (internal to the Xcode project). These files are included in the current distribution.
- On Android, there was an issue where the verification of the MF for Offline OTP was skipped. The MF is now verified on the client.
- There was an issue where a corrupted QR code could potentially cause a DoS for an Offline OTP token.

13. CoT Release 1.2.1706.31382

This version of CoT support for Offline-OTP. Offline-OTP provides the capability to use the mobile device for 2-factor authentication, even when the device is offline.

The OTP is accomplished by scanning in a QR code from a PC screen using your mobile app. In addition, the OTP is implemented using MPC protocols, which adds another layer of security in that the full OTP seed is never exposed.

14. CoT Release 1.2.1706.26227

14.1. iOS SDK

Added static framework (DYMobileCore and DYMobileSign)

15. CoT Release 1.2.1706.19135

15.1. Android SDK

15.1.1. Issues Fixed

Fixed bug in Android 8.1 (API 27) caused by an undocumented API change.

When creating a signing key in the *AndroidKeyStore*, a number of seconds for user authentication validity must be specified using *setUserAuthenticationValidityDurationSecondsscenarios*().

In Android 8.1 (API 27), the *setUserAuthenticationValidityDurationSecondsscenarios*() method throws an exception when passed values are greater than 4,294,900.

The default value for *setUserAuthenticationValidityDurationSecondsscenarios*() is now set to 4,294,900 (was 2,147,483,647) seconds.

Added an API for using a custom value (see the Android Developer's Guide 1.2.1706.19135).

16. CoT Release 1.2.1706.18784

16.1. EKP Server

16.1.1. Issues Fixed

- EKP server no longer fails to resolve the EKP domain list after losing the Oracle connection.
    - The cause: internal domain list was cleared after losing the Oracle connection. Solution: the list is maintained up to the next successful refresh from the database.

17. CoT Release 1.2.1706. 11(18263)

17.1. EKP Server

17.1.1. Enhancements

Request for confirmation when changing a setting

By default, the system now prompts to confirm a change. To bypass a prompt, append `-f true` to the command.

Example:

- `mobilecl enablepow -d mydomain` command prompts for confirmation
- `mobilecl enablepow -d mydomain -f true` runs without prompt

Server Key Expiration

- Added command to retrieve the server key expiration date:

```
mobilecl prn-server-key-exp-date
```

- Added SERVER_KEY_EXPIRED warning log.

17.1.2. Issues Fixed

EKP server no longer crashes when a wrong keystore password is provided in the settings file. The system now logs the LOAD_STORAGE_KEY error.

> In general, the EKP Server stops working when something goes wrong while loading encrypted storage data.

> The system logs LOAD_STORAGE_KEY error, and return an error to all EKP API calls, including the /api/status request.

After fixing the cause (such as invalid storage key, invalid storage key credentials, etc.), restart Tomcat on the server to reactivate its functionality.

17.2. Android SDK

17.2.1. Upgrade

The Android sample code was upgraded to use `OkHttp 3.8`.

18. CoT Release 1.2.1706

18.1. EKP Server

18.1.1. New Features

New REST API Endpoints for Smart Proxy Support

New REST API endpoints were added that supports enhanced proxy server functionality.

New HTTP health-check endpoints and updated HTTP responses

Endpoint	Positive Response	Negative Response
/api/status	{"status":"SYSTEM-OK"}	{"status":"ERROR"}
/api/status/database	{"status":"DATABASE-OK"}	{"status":"ERROR"}
/api/status/cryptoengine	{"status":"CRYPTO-OK}	{"status":"ERROR"}
/api/status/storageencryption	{"status":"ENC-ON"} {"status":"ENC-OFF"}	

18.1.2. Enhancements

Log Message Format

Mobilecl logs now contain OS `username` field.

New Sections in the Admin Guide

Added the following sections:

- Cloning an EKP server.
- Switching from PROD database to COB and back.
- Troubleshooting and updated log event list.
- Mobilecl lockout control commands.

18.2. EKP Proxy

18.2.1. New Features

Reimplemented of the Smart Proxy Architecture

The core of the smart-proxy has been uncoupled from communication framework.

New Endpoints for the Sample Implementation of the Communication

New endpoints were added to support the sample implementation of communication with the mobile apps utilizing smart-proxy capabilities.

18.3. EKP Mobile SDK

18.3.1. New Features

A New set API for Interworking with Smart Proxies

Classes supporting proxy-based applications were redesigned. In the current implementation, they:

a. Are uncoupled from the communication framework.

b. Support crypto industry-standard execution paradigm: create/update/finalize.

Redesign of the Proxy Samples

The code samples that demonstrate interworking with smart proxies were redesigned to:

a. To use the new paradigm.

b. To implement uncoupled communication framework with the proxy server.

19. CoT Release 1.2.1705

19.1. EKP Server

19.1.1. New Features

Support Smart EKP Proxy

EKP Server Software has been enhanced to support the functionality required by the smart EKP Proxy (see *Sample EKP Proxy* below).

Separate Audit-Log Files

Now you can modify the log4j/dymobile.xml to accumulate domain-specific audit logs in the separate (per-domain) files.

Configure LDAP Service Account Credentials Type

This feature allows adding LDAP-based authentication of mobile user credentials.

Configurable Out-of-Sync Threshold per Domain

Now it is possible to reject a crypto request from a mobile device that is not synchronized with its share on the Server.

> Mobile device synchronizes with its share on the server upon successful completion of the request-response cycle. A mobile enters "out-of-sync" state when the counter of the incomplete requests to the server passes the configured threshold.

Each domain can have a different threshold value for the counter of the incomplete requests. By default this value is unlimited.

19.1.2. Issues Fixed

Logs of the previous days are lost

Fixed.

Mobilecl can disable a domain feature without warning that it holds active tokens

Fixed. In such a case, Admin is prompted to confirm the action.

Admin Guide – Added description and use of $DYMOBILE_OPTS

$DYMOBILE_OPTS allows customizing Java options in certain commands. For example, it is used to specify the location (and password) for accessing Java Key Store while running the `mobilecl selftest` command to an HTTPS site.

19.2. Sample EKP Proxy

19.2.1. New Features

Application Proxy for EKP Server

This release introduces sample implementation of an application proxy for EKP Server that can be installed on the App Server.

Use of EKP Proxy is recommended in configurations where EKP server cannot be available for direct access from mobile applications, or there is a specific need to use the smart proxy options.

The sample EKP proxy provides two types of endpoints for the app developers:

- Simple endpoint (just for moving messages between mobile and dyadic server).
- Smart endpoint (for completing the computation and getting results, while forwarding the messages between the mobile and the EKP Server).

In this release, the smart EKP Proxy supports the following functionality:

- Password protection - using PQC encryption.
- Signing - using all variants of RSA.

19.3. EKP Mobile SDK

19.3.1. New Features

HTTP Parameters

Parameters may be forwarded to the EKP Server as a list of name-value pairs using one of the following methods:

- Appended to the URL.
- In the HTTP header.

HTTP Credentials

The credentials of the caller (client-id, client-secret) may be forwarded to the EKP in the header of HTTP (in the Authorization field) according to the Base Access Authentication specification (see *https://en.wikipedia.org/wiki/Basic_access_authentication*).

Added LDAP-based authentication of user credentials

LDAP-based authentication of user credentials (username, password) has been added. It supports the following username formats:

- Plain username.
- NetbiosName\sAMAccountName (domain\username).
- UPN (username@domain.com)
- Distinguished Name (cn=some,cn=ou,dc=domain,dc=com)

Connecting a Mobile Device to the EKP Server via EKP Proxy

The classes `DYPassword` and `DYSign` have been enhanced to support operations (Sign and Password Retrieve) that are implemented by the Smart EKP Proxy SDK.

Class	Method used by Direct Access to EKP server or simple proxy	Methods that replace it when using smart EKP proxy
DYPassword	retrievePassword	getDataForPasswordProxyRetrieve finalizePasswordProxyRetrieve
DYSign	Sign	getDataForRSAProxySign finalizeRSAProxySign

19.3.2. Issues Fixed

The TouchID prompt remains present after it expires.

Fixed. Response to the TouchID prompt is not time limited.

Integrity-TOKEN UUID is not bound to crypto message content in the case of password retrieval

Fixed.

We use token UID as AAD (Additional Authentication Data) when encrypting the payload sent to the mobile server.

When the min client version is lower than Rel 1.2, the server will try to decrypt with and without the AAD. When it is set to 1.2 or higher, the server will decrypt with the AAD only.

EC Signature verification fails on iOS8.1 and iOS8.2 platforms

Fixed for the sha256.

- sha384 and sha512 are not supported (iOS 8.1 and 8.2 limitations).

Access to a token does not verify that the token has been created in the domain that is being accessed

Fixed.

The software can only access tokens created in the currently addressed domain (and only if the feature list of the domain permits using the specified type of the token).

20. CoT Release 1.2.1703

20.1. EKP Server

20.1.1. New Features

Using Proof of Work (POW) to Mitigate DOS Attacks

Throttling of requests for new token can now be imposed on a mobile device by demanding it to solve the distinct challenge. Admin controls this feature by enabling/disabling and assigning the level of complexity of the challenge to each token domain. The complexity translates into time consumed by the mobile device to solve the challenge before issuing a new request.

20.1.2. Enhancements

Optimizing Backend Performance

Extensive use of caching to reduce the latency at the backend.

Enhancing Logging Capabilities

Logs now carry Domain attribute.

Log Archiving framework has been reorganized.

20.1.3. Issues Fixed

Failure to remove/delete the empty domain

Fixed.

Mitigating DoS Attacks using Client Refresh Setting

The minimum Refresh Interval is now set to 60 minutes.

Failure to rotate log files

Fixed and improved. Refer to "Enhancing Logging Capabilities" for more information.

Failure to run mobilecl self-test

There is no longer a requirement to create symbolic links to Openssl library files.

No validation against client-setting API for a non-existent domain

Fixed.

20.2. EKP Mobile SDK

20.2.1. New Features

Using Proof of Work (POW) to mitigate DOS attacks

See "Using Proof of Work to mitigate DOS attacks" in EKP Server.

Supporting Touch on pre-A7 iOS devices

Support for iOS devices before the A7 processor (e.g., iPhone5, iPhone 4s, iPad mini) with iOS9 and without secure–enclave.

20.2.2. Enhancements

Hiding API Keys and Secrets in HTTP header

Keys and Secrets used by APIs can now be sent as part of the HTTP header, or as parameterized query string of the URL.

Accelerated performance

Enrollment phase of the mobile device has been optimized to reduce the latency of the enrollment phase.

20.2.3. Issues Fixed

Crash due to OTP token enrollment while the domain feature is disabled

Fixed.

Inconsistent units of the refresh time on Android and IOS

Unified to minutes.

The inconsistency of iOS and Android sample program

Fixed.

21. CoT Release 1.2.1701

Release version 1.2.1701 corresponds with Dyadic Mobile Build number 1.2.11888.2062.

21.1. EKP Server

21.1.1. Upgrade

Upgrade from previous versions (1.0/1.1) is supported, for more information refer to the server installation guide.

21.1.2. Enhancements

The token key can be added at any time

Before this Release, the database had to be empty

You can decide to encrypt EKP data at any time, even after the system has been used for some time.

Token Key rotation, Key store keys rotation

Encryption keys of the different encrypted values in the EKP DB can now be rotated.

Added Mobilecl logging

Log Files are split into two major areas:

- Log files that reflect mobilecl command activity (New in this release).
- Logs generated by the mobile service.

Also, log files are split according to the level of the provided detail: "Audit" and "Trace."

- Logs generated by mobilecl commands have the following naming pattern:

```
mobilecl.<User-name>.<date>.log
```

```
mobilecl-trace.<User-name>.<date>.log
```

- Logs generated by the mobile service and have the following naming pattern

```
dymobile.<date>.log
```

```
dymobile-trace.<date>.log
```

Added domains for multi-tenancy support on the same server

The Admin of EKP Server can now specify "domains" for the more granular management of tokens. A Mobile App that is using Dyadic SDK can be further bound to particular "domain" of the Dyadic Crypto Service on the specified EKP Server.

21.1.3. Dependencies

New dependencies

org.apache.logging.log4j

- log4j-api version 2.8
- log4j-core version 2.8
- log4j-web version 2.8
- log4j-slf4j-impl version 2.8

21.2. EKP Mobile SDK

21.2.1. Enhancements

Introduced SDK refactoring option

You can now reduce the size of the Dyadic SDK that is included in your SW, by selecting only the required crypto components from the multitude provided by the Dyadic from the following set:

- Core
- Sign
- Password
- Encryption
- OTP

Utilize embedded Secure Element

Both iOS and Android SDK's are now utilizing the Secure Element (SE) on supported devices:

- Use the Secure Element to store token data
- SDK <–> EKP Server communication is encrypted using a key stored in SE.

Credentials API

The API to provide credentials to different SDK methods was modified to enable future enhancements. For more information refer to the Programmers Guide.

Client Settings

It is now possible to define the client's settings, such as "Refresh interval," and client applications that use the SDK will be able to fetch these settings from the Dyadic Mobile server.

Storage Encryption Configuration

Enabling storage key encryption is now done using a CLI command `mobilecl enable-storage-enc`, instead of a parameter in the `dymobile.conf` configuration file.

Use Domain-based addressing

During Initialization of the embedded SDK SW, you may specify the name of the domain that serves Dyadic APIs of your application.

The name of the "domain" (as the name of the Feature that you are using) is encoded into the URI of the HTTP method that implements the API.
If you choose not to use this option, SDK SW directs its requests to the "DEFAULT_ DOMAIN."

Android: Added fingerprint authentication

SDK for Android now supports fingerprint-based authentication on supporting devices.

22. CoT Release 1.1.1610

22.1. EKP Server

22.1.1. Enhancements

Management of DB Access Credentials

New option added to retrieve password (used by Dyadic Server to access database) from an external resource.

Migrate REST APIs from HTTP GET to HTTP POST

REST APIs of EKP v1.0 that used HTTP GET have been implemented using HTTP POST

Managed HTTP GET support

Starting with v1.1, the EKP Server rejects (by default) to process REST APIs that use HTTP GET.

> To support backward compatibility with the EKP 1.0 Clients, EKP Server must be explicitly configured to accept HTTP GET requests.