BlackBerry Dynamics Security White Paper

Version 1.8

Overview

This document provides detail about the security provided by BlackBerry Dynamics. The intended audience is CIOs, IT managers, software architects, and people with a similar level of technical knowledge.

This document assumes some knowledge about the features and purpose of the BlackBerry Dynamics Platform. For background, you might want to read the BlackBerry Dynamics Administrator and Developer Overview.

BlackBerry Dynamics enables mobile enterprise applications to employ the industry-leading security features such as:

- Network operation center (NOC) and proxy infrastructure that enables connections between mobile clients and application servers that are behind the enterprise firewall. There is no need for a VPN, or for ports to be opened in the enterprise firewall.
- End-to-end encryption of data in transit between mobile clients and application servers.
- Storing enterprise data on the device in a separate secure container that can be remotely wiped by an administrator.
- Encrypting with AES 256-bit cipher. This technology is used to protect data at rest, in the secure container, and to protect data in transit between client and server.
- Enforcing password and device compliance policies, when accessing enterprise data.

BlackBerry has developed few applications using BlackBerry Dynamics such as BlackBerry Work, BlackBerry Access etc. Many ISVs have also built applications using BlackBerry Dynamics. Enterprises can build their own applications using BlackBerry Dynamics as well.

Components

The BlackBerry Dynamics platform has the following components and interfaces:

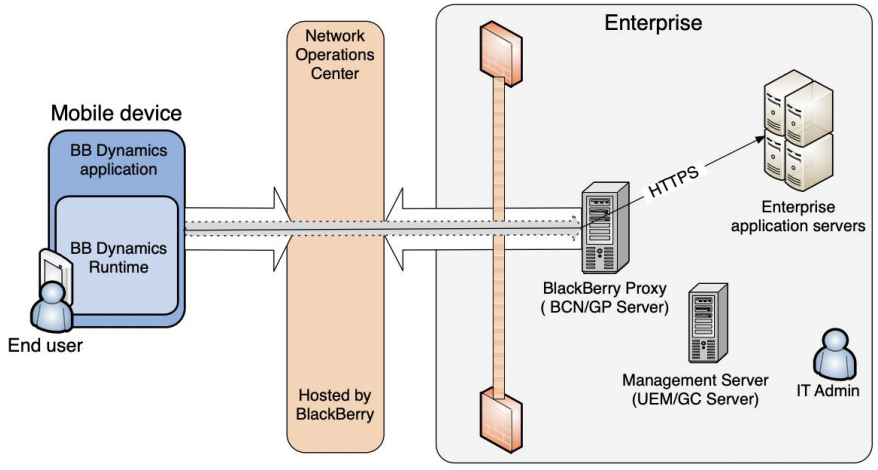


Figure 1 - BlackBerry Dynamics Platform Architecture

- **BlackBerry Dynamics Application**: An application with embedded calls to the BlackBerry Dynamics runtime which provides services/features to the user. Sometimes also referred as Dynamics application.
- **BlackBerry Dynamics Runtime:** Every BlackBerry Dynamics application includes an instance of the BlackBerry Dynamics runtime. The runtime has APIs that give the application access to user authentication, secure communications, secure storage, and communication behind the firewall. The runtime also handles enforcement of security policies on behalf of the application. An instance of the BlackBerry Dynamics runtime may sometimes be referred to as a Container or Dynamics Runtime.
- **Network Operation Center**: The BlackBerry Dynamics Network Operation Center (NOC) provides the secure communications infrastructure between the BlackBerry Dynamics runtime on the device, and the BlackBerry Dynamics enterprise servers behind the firewall.
- **Enterprise Servers**: There are two BlackBerry Dynamics components installed behind the enterprise firewall.

    1. **Management Server**: This server provides management of the enterprise's users, applications and security policies. In a standalone Dynamics deployment this is called Good Control (GC) server. In this document this server is referred as **GC server**. In BlackBerry UEM deployment this service is part of **BlackBerry UEM server**/core. BlackBerry Dynamics applications can be activated against Good Control server or BlackBerry UEM server. BlackBerry UEM server supports MDM and enterprise app store functionality in addition to BlackBerry Dynamics applications.
    2. The **BlackBerry Proxy** service provides the secure communications infrastructure between the NOC and application servers that are behind the enterprise firewall. This service is run inside **GP server** in standalone Dynamics deployment. In a BlackBerry UEM deployment this service is part of **BlackBerry Connectivity Node** (BCN).

Protocols

- **Good Relay Protocol (GRP)**: Protocol over TCP for end-to-end secure communications between the BlackBerry Dynamics app and the GP server. See Good Relay Protocol.
- **Good Notification Push (GNP):** Protocol over HTTPS which enables notification messages to be pushed from an application server to BlackBerry Dynamics app. See Good Notification Push.
- **BlackBerry Dynamics Auth token:** Protocol which enables an application server to query the GP server over HTTPS to determine the user identity of an incoming request. See BlackBerry Dynamics Authentication Token.

Security Features

The following table lists the major features of the BlackBerry Dynamics platform and the specific security capability associated with that feature.

Security Element	Features
Container Access	• Different policies for different users • Remote lock • Compliance verification • Auto-Lock & User access authentication
Container Data Storage	• Secured & Managed container to protect enterprise data • User data encrypted with AES-256 • Remote erase & lock • FIPS140-2 certified crypto module.
Data Transmission	• TLS connections • AES-256 encryption • FIPS140-2 certified crypto module in the BlackBerry Dynamics runtime. • Connection monitor
Enterprise Resource Protection	• No holes in the firewall • Role based administration • No enterprise credentials outside the firewall • Centralized and cross platform security control over customer base • Connections to permitted application servers, or permitted domains.

How Data Is Protected

Data can be grouped into two categories:

- On-device: Data already stored on a device
- In-transit: Data in process of being communicated

On-Device Data

These tables provide a summary of how on-device data is protected by the BlackBerry Dynamics platform.

How data is protected…	Answer
Enterprise data is saved inside the BlackBerry Dynamics app	Encrypted with AES-CBC using with 256 bit key.
BlackBerry Dynamics application encryption key	Protected with user password/secret. Password strength and requirement is set from the GC console.

How data is protected…	Answer
After uninstallation	Files are deleted.
Device stolen	User is asked for password. Data on device is encrypted.
User is no longer entitled to the application	BlackBerry Dynamics app locked or data deleted. Initiated by IT administrator using a GC console.

In-Transit Data

These tables provide a summary of how in-transit data is protected by the BlackBerry Dynamics platform.

How data is protected…	Answer
Data in transit between the BlackBerry Dynamics app and GP server.	Encrypted with AES-CTR by GRP.
Data in transit between application client and application server	Application developer controls this. The BlackBerry Dynamics runtime library provides HTTPS option. Data in transit between application client and GP server is encrypted by GRP regardless of what application developer uses at application layer.
Data pushed by the application server to the BlackBerry Dynamics application using GNP (sent from the GP server)	Encrypted with AES-CBC.
GC server to BlackBerry Dynamics NOC	TLS
GP server to BlackBerry Dynamics NOC	TLS
GC server to GP server and GP server to GC server	TLS

BlackBerry Dynamics App

The BlackBerry Dynamics SDK includes a client library, which is used to build the BlackBerry Dynamics app for mobile devices. The BlackBerry Dynamics SDK provides the APIs to perform app activation, data pushes, enterprise connectivity, and user authentication. For management, the BlackBerry Dynamics runtime supports remote app and container management, including the ability to wipe app data. The BlackBerry Dynamics SDK is available to developers for iOS and Android platforms to build applications. In addition, BlackBerry supports cross-platform frameworks such as Cordova, HTML5 and Xamarin. Features supported for each platform can be found at BlackBerry developer site and in the API Reference Documentation. In addition, on Windows and MacOS BlackBerry provides BlackBerry Access application which is built using BlackBerry Dynamics SDK.

Application activation is the process by which a BlackBerry Dynamics runtime receives initial provisioning and configuration data from the NOC and the GC server. Security aspects of application activation are described in App Activation.

Once activation is completed a BlackBerry Dynamics app may:

- Establish an HTTP, HTTPS, and TCP connection to an application server behind the enterprise firewall by using APIs exposed in the BlackBerry Dynamics SDK. Security aspects of enterprise connectivity are described in Enterprise Connectivity. The BlackBerry Dynamics runtime supports TLS 1.0, TLS1.1 and TLS1.2. The BlackBerry Dynamics runtime also supports NTLM v2 and Kerberos authentication for HTTP/S connections.
- Set up a push service with its app server. The push service enables the app server to push data to the BlackBerry Dynamics app when the BlackBerry Dynamics app is not directly connected to the application server. Security aspects of the push service are described in Good Notification Push section.
- Use the encrypted file and encrypted database support provide by the BlackBerry Dynamics runtime.

Data Storage on the Device

A BlackBerry Dynamics app is recommended to store all user and management data in encrypted files and databases using APIs provided by BlackBerry Dynamics runtime. These files and databases are encrypted by the Dynamics runtime with AES (CBC mode), using a 256-bit random key (Data Encryption Key), and random IV's.

The Data Encryption Key itself is encrypted with a key based on the user provided secret and stored in the Startup file (see next section). The mechanism used to protect Data Encryption Key varies based on the user authentication mode enabled by the administrator. The startup file is disabled from being backed up. The startup file is used during the app startup.

An encrypted copy of Data Encryption Key is also saved in the Recovery/Restore file (as described in the section Application Unlock and Restore). Recovery file could be backed up if app data backup is enabled on the device. This file is used to restore the app on a different device.

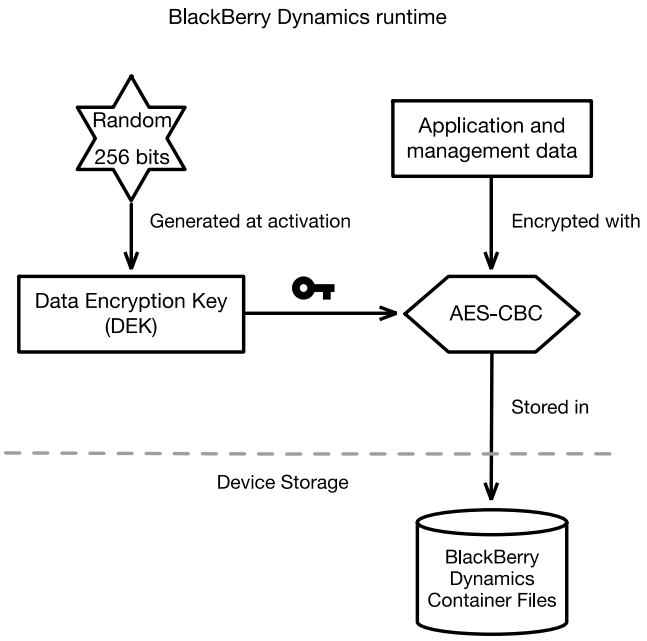


Figure 2a: BlackBerry Dynamics runtime data persistence

User Authentication and Key Storage

A BlackBerry Dynamics app can only be used by the user who has activated the app. After app activation is completed, the user is asked to set a password. This password (User Secret) is used to secure the Data Encryption Key and authenticate the user on subsequent app startup. When the app delegates authentication to another BlackBerry Dynamics app, the User Secret is provided by another BlackBerry Dynamics app (see the Authentication Delegation section).

A cryptographic hash of the *User Secret*, called the User Key, is used to encrypt Data Encryption Key (DEK). Encrypted Data Encryption Key is kept inside the Startup file. SHA512 hash of User Key is also saved in the Startup file and on subsequent app startup is used to authenticate the user. Depending on the device OS and hardware security capabilities, some cryptographic operations are completed inside the secure enclave or device key store as shown in the following diagrams.

Android

The figure below shows how Data Encryption Key is secured on an Android device.

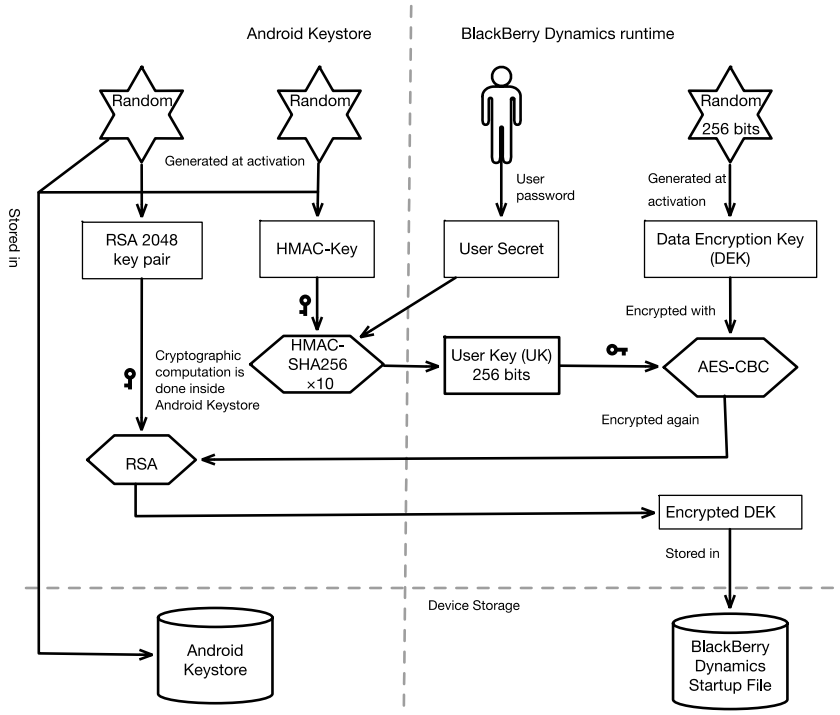


Figure 3a: Securing Data Encryption Key on Android

iOS

The figure below shows how Data Encryption Key is secured on an iOS device. Secure Enclave support was added in SDK version 4.2.x.

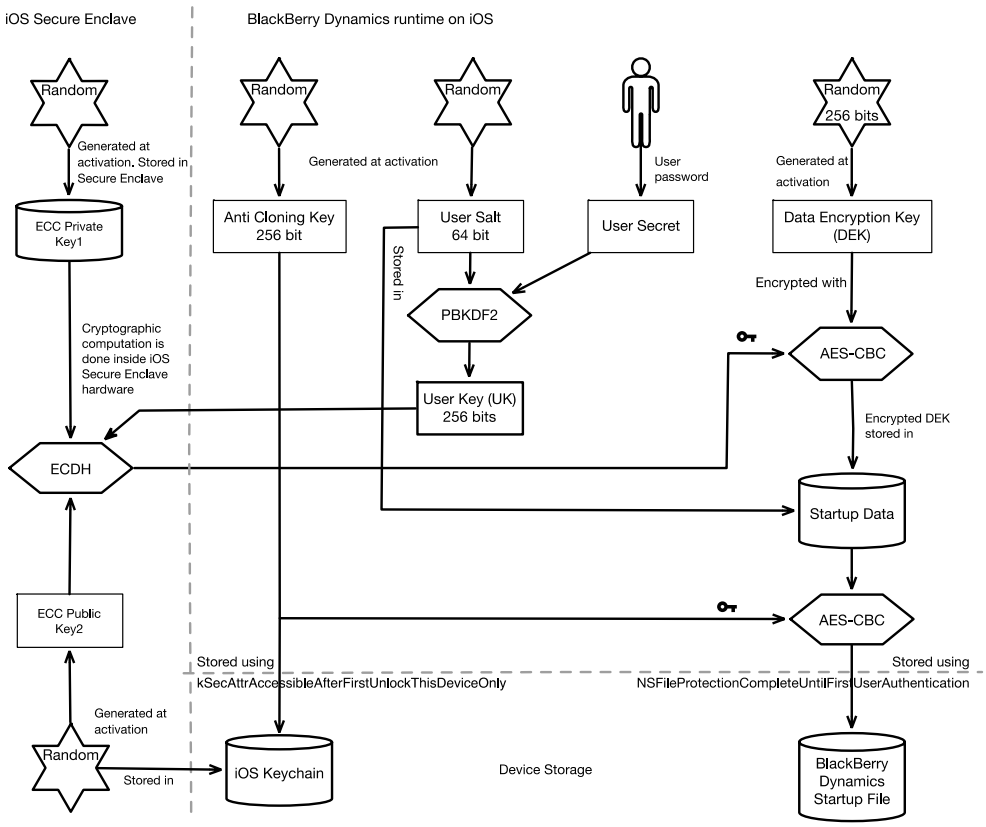
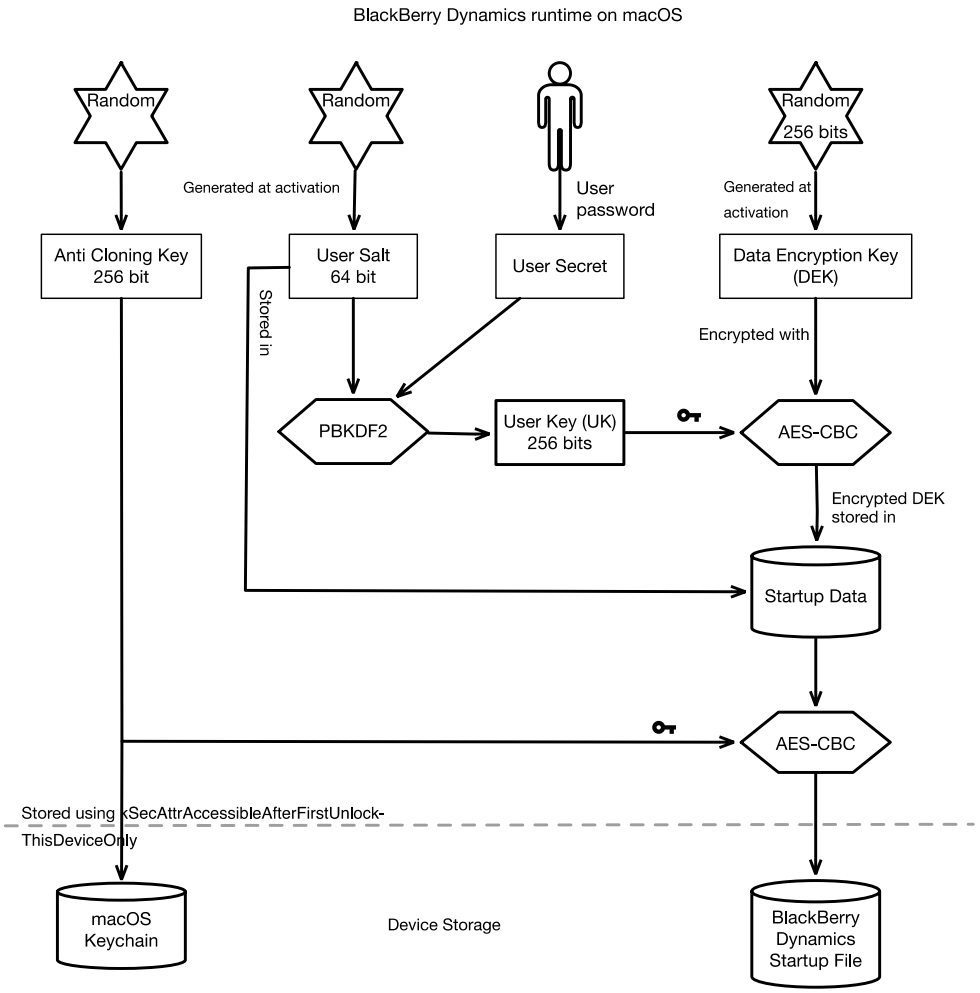


Figure 3b: Securing Data Encryption Key on iOS

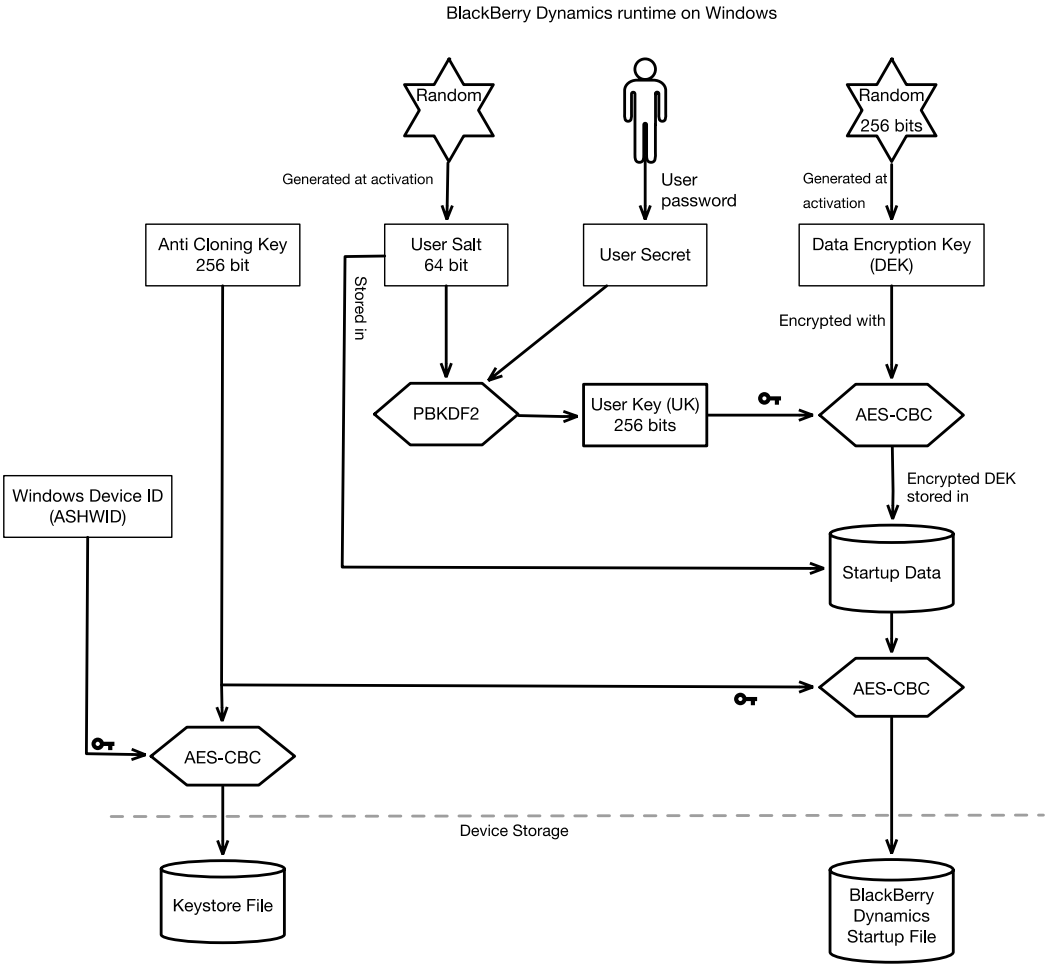
macOS

The figure below shows how Data Encryption Key is secured on a macOS device.



Windows

The figure below shows how Data Encryption Key is secured on a Windows device.



Biometric User Authentication

Blackberry Dynamics runtime supports user authentication based on fingerprint recognition system from Apple Touch ID and Face ID on iOS devices, Android Fingerprint and Samsung Pass on Android devices. Collectively these features are called biometric authentication below. Biometric authentication is not supported on macOS and Windows OS. When this feature is enabled, user can perform biometric authentication instead of providing password. Blackberry Dynamics runtime support of biometric authentication supplements user password.

No work is required from the Blackberry Dynamics app developers. Operating system's standard behavior of biometric authentication is not changed by BlackBerry Dynamics runtime. This feature also does not impact Authentication Delegation feature. Administrator can control this feature from the management console in the

security policy settings. No biometric meta-data is received by Blackberry Dynamics. Operating system only informs Blackberry Dynamics SDK if the biometric authentication has succeeded or failed.

Administrators can

- Enable biometric authentication. This setting allows Dynamics runtime to use biometric authentication when app requires re-authentication (i.e. app is already running) such as idle lock, restore from background, easy activation etc.
- Enable biometric authentication after cold start. This is the case when device is shut down and started or app is started for the first time. If this is not enabled, user will be prompted for password in these situations.
- Provide a time interval after which user will be prompted for password (even when user has already performed biometric authentication)

If biometric authentication is enabled by the administrator, user is still required to set password after the app activation. In addition, if the policy allows, users can choose to enable fingerprint authentication. If the device passcode is changed, or when the fingerprint registered by the device OS is changed, Blackberry Dynamics runtime will ask for the user password. For more information about fingerprint support in Blackberry Dynamics and security recommendations see Blackberry Dynamics Fingerprint Authentication document in References.

Android

On Android devices when biometric authentication is enabled for cold start, a new 256 bit Cold Start Key (CSK) is generated in the Android keystore with userAuthenticationRequired property and saved in the keystore. This CSK is used as depicted in the diagram below. This is in addition to how user secret is handled and used as shown in the User Authentication and Key Storage section (Figure 3a: Securing Data Encryption Key on Android).

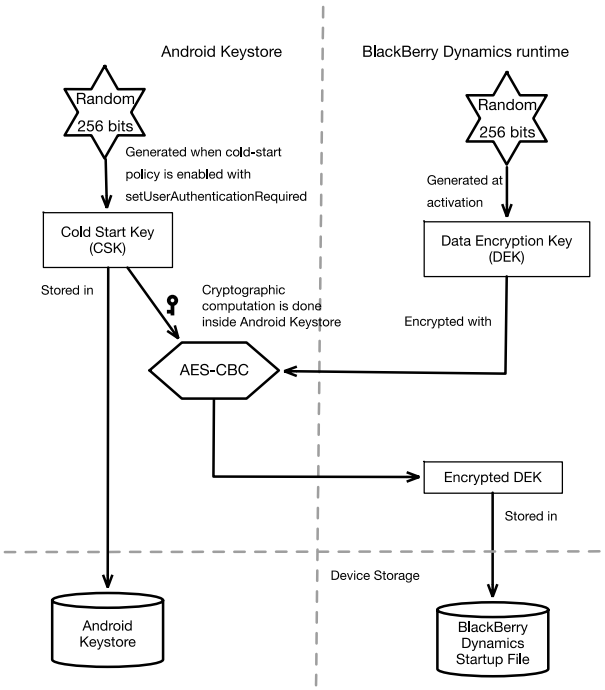


Figure 4: Securing Data Encryption Key on Android for Cold Start with biometric authentication

iOS

When fingerprint authentication is enabled for cold start on iOS devices, the User key is saved inside the device keychain with attribute kSecAttrAccessibleWhenPasscodeSetThisDeviceOnly. If the passcode is removed/disabled, User key is removed from the device keychain.

No Password

Administrators also have the option to not require the password to be set by BlackBerry Dynamics app users. When this "No Password" feature is enabled, the BlackBerry Dynamics runtime creates 256 random key called User Secret Replacement (USR) and is used in the same way as user password. USR is saved inside the Startup file.

Idle Lock

The BlackBerry Dynamics runtime supports idle lock feature. When the user has not interacted with the app for a specified amount of time as set by the administrator, the BlackBerry Dynamics runtime will display a BlackBerry Dynamics unlock screen to the user. The BlackBerry Dynamics unlock screen is super imposed on the app user interface screen. The app is said to be idle locked and user must authenticate to see the app's user interface again.

While BlackBerry Dynamics unlock screen is displayed, the app is still running; app data is present in the runtime memory; the app has access to the data in the file system, and it can connect to the app server.

Bypass Unlock

Bypass unlock feature allows the app to display certain user interface screens to the user while idle lock is in effect. This feature can be useful to the apps that require swift user response such as inbound call in a telephony app or immediate storage of external data such as picture or note to the secure storage.

Bypass unlock is only supported on iOS and Android.

The user interface screens that are enabled for Bypass Unlock are specified by the developer when the app is built. Access to Bypass Unlock is restricted and must be requested from BlackBerry. Apps that are granted access to Bypass Unlock are issued a unique signed registration token by BlackBerry. The token must be embedded in the app declaration at build-time.

Developers are required to display the list of user interface screens that are enabled for Bypass Unlock feature to the administrator using app specific policies in the GC console. In addition, developer is recommended to provide control to the administrator to turn on/off this Bypass Unlock feature via the same app specific policy.

For more information on how to use Bypass Unlock feature and how to get registration token see Bypass Unlock: Application Developer Guide

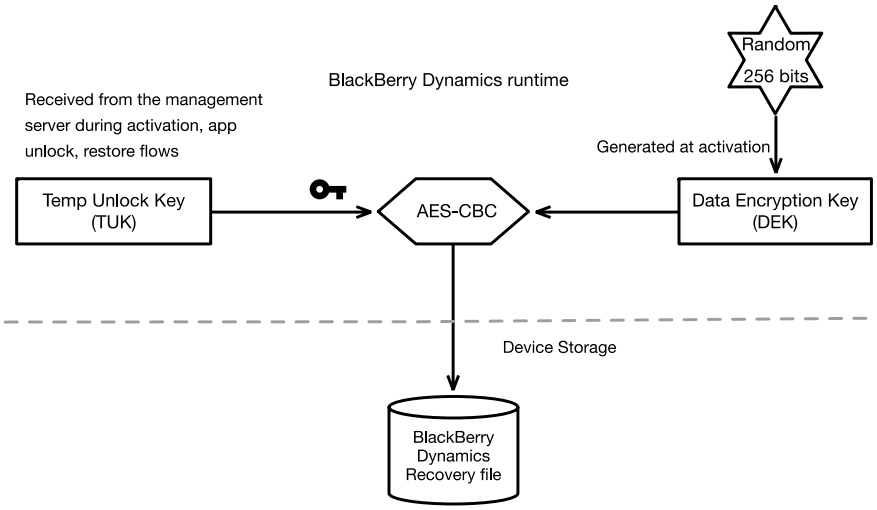
App Unlock and Restore

The BlackBerry Dynamics app can be unlocked if a user has forgotten the password or the app has been remotely locked by the enterprise IT administrator. To unlock the app user needs an app unlock key issued by the management server.

This app unlock key can only be used to unlock the container for which it was created and will expire in 24 hours. The key is used to authenticate the user and identify the instance of the BlackBerry Dynamics app to the management server in the unlock process.

The BlackBerry Dynamics app then uses the same process as described in App Activation. It authenticates itself to the management server using app unlock key and gets the Temp Unlock Key (TUK, 256bit) from the management server and unlocks the BlackBerry Dynamics app (i.e. decrypt the Data Encryption Key, DEK).

After the initial enterprise activation, a copy of DEK encrypted with TUK is saved in the *Recovery* file. TUK is created for each container by the management server during the activation process. The same TUK is sent to the client during the app unlock/restore process.



App Restore on a New Device

A BlackBerry Dynamics app supports data being restored on a new device. A user who is using a BlackBerry Dynamics app on one device can backup data for the BlackBerry Dynamics app using OS provided services such as iCloud or equivalent. This data can then be restored on a different device running the same OS. However, BlackBerry Dynamics app data is encrypted inside the container. To unlock and restore the data inside the container user needs to get an app unlock key from the management server and input this on the BlackBerry Dynamics app. This is the same key and process as described above.

Background Authorize

Background Authorize enables a recently locked application to utilize the principal Dynamics APIs like secure storage and secure communications when the application is running in the background. This feature is helpful when an application may have stopped (because the operating system unloaded it from runtime memory, or the application has crashed). On an iOS device an application may be started in the background in response to having received an APNS message (for example, a new email has been received). In this scenario, if the Background Authorize feature is enabled, the application is able to download the new messages and store them in the secure container even though the user did not unlock the Dynamics container. When the user brings the application to the foreground, they are required to authenticate to see their new messages.

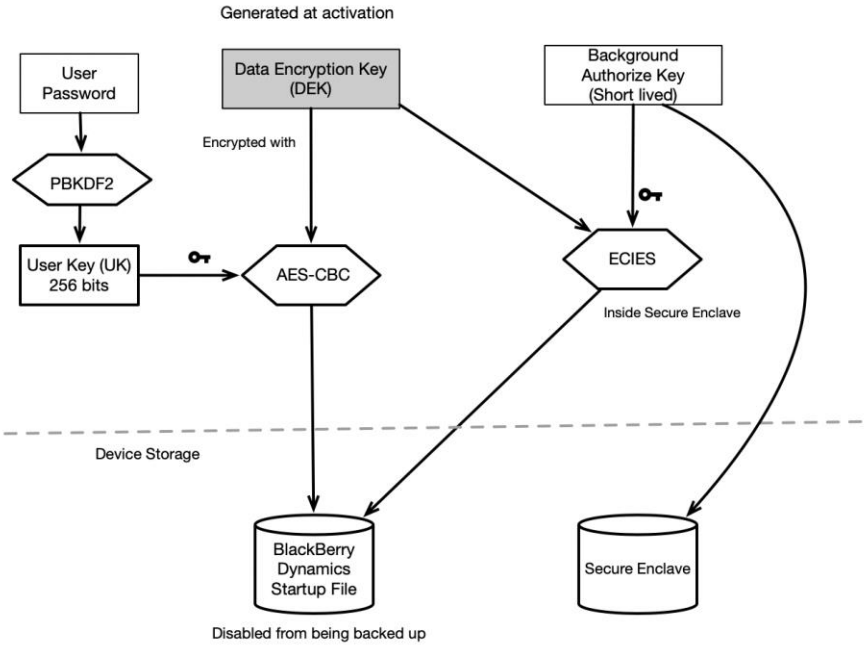
This feature is supported on iOS platform in the BlackBerry Dynamics SDK 6.0.

For a Dynamics application to support Background Authorize developer needs to:

- Request to use the feature from BlackBerry and embed a signed authorization inside their application.

- Define an application specific policy which enables an administrator to turn on this feature and specify duration for which this feature is enabled. Duration is the time since the user last authenticated to the Dynamics container. For example: 12 hours, or 1 day.

To unlock a Dynamics container the user key is required as described in "User Authentication and Key Storage" section. User key is derived from user password or provided by Auth Delegate application. When Background Authorize feature is allowed and the container is unlocked, a short-lived Background Auth key is generated in the iOS Secure Enclave. Now Background Authorize feature is active. Background Auth key is used to encrypt Data Encryption Key (DEK) and this encrypted DEK is saved in the BlackBerry Dynamics Startup file. When Background Authorize time has elapsed (from the time user authenticated the container), Background Auth key is removed from the iOS Secure Enclave and Background Authorize feature becomes inactive. The security risk of this feature is same as No Password feature during the time this feature is enabled.



Sharing Data between BlackBerry Dynamics Apps on the same Device

Using Shared Services, a BlackBerry Dynamics app may securely send and receive data with another BlackBerry Dynamics app installed on same device. The security model is described in Shared Services Framework.

Data Leak Prevention

Using policy rules in their management server, administrators can prevent users from copying data from BlackBerry Dynamics app to non-BlackBerry Dynamics apps on the device, as well as prevent dictation, custom keyboards, and screen capture. Capabilities available on each platform vary and can be seen the management console. In the GC console, the **Data Leakage Prevention** section of the security policy controls these capabilities. In the BlackBerry UEM console, these settings are managed in the BlackBerry Dynamics profile.

Secure cut-copy-paste is accomplished by encrypting the data saved in the copy/paste buffer (using AES 256, CBC) along with keyed hash (HMAC-SHA512). All containers belonging to a user are provided same copy-paste key by the management server.

Device Integrity

BlackBerry Dynamics runtime performs various checks to verify the device and application integrity. On Android and iOS, admin can enable rooted/jailbreak detection using the compliance policies. Similar jailbreak detection policies are not available on Windows and Mac platforms as users on these platforms most likely to have administrative privileges. Therefore, desktop applications lack some security protections offered by closed mobile platforms. Please check the desktop application admin guide for available mitigations. BlackBerry Dynamics runtime also supports Android SafetyNet attestation checks and iOS application integrity verification. Rules for various compliance checks are pushed to clients by the management server and updated dynamically.

FIPS Compliance

Federal Information Processing Standards (FIPS) are U.S. government regulations regarding computing and computing security (see FIPS Pub 140-2 by NIST for more information). FIPS compliance is supported on Android and iOS platforms and can be enabled in a security policy. BlackBerry Dynamics SDK uses FIPS validated crypto module called OpenSSL FIPS Object Module.

When an admin enables FIPS compliance in a policy, the major effect is on associated apps. Enabling FIPS compliance enforces the following constraints in conformance with FIPS:

- App must be built with FIPS enabled as documented in the references below. Apps that do not have FIPS enabled will not conform to the security policy and are blocked on user devices. Users must contact an administrator to be unblocked. Administrators can unblock the user by disabling FIPS compliance in the policy at either the user level or the app level.

- MD4 and MD5 are prohibited by FIPS, which means that access to NTLM- or NTLM2-protected web pages and files is blocked. Wrapped apps are blocked. In secure socket key exchanges with ephemeral keys, with servers that are not configured to use Diffie-Hellman keys of sufficient length, BlackBerry Dynamics retries with static RSA cipher suites.

For information about how to build apps using BlackBerry Dynamics SDK that will be FIPS compliant, see the BlackBerry Dynamics SDK for Android Development Guide or the BlackBerry Dynamics SDK for iOS Development Guide.

Enterprise Servers

There are two types of BlackBerry Dynamics enterprise servers installed inside the enterprise firewall: A Management server (UEM or GC server) and a BlackBerry Proxy (BCN or GP server). These servers only make outbound connections to the NOC, and do not require inbound ports to be opened in the firewall. Both these servers work with outbound proxy servers if required.

Note: Although this document explains the standalone Dynamics solution using GC server, but comments for the most parts apply to the BlackBerry UEM deployment also. If you require MDM or MAM capabilities, you must manage BlackBerry Dynamics apps using BlackBerry UEM. When you upgrade from Good Control to BlackBerry UEM, you not only get to use the great features that Good Control provides but you also get to take advantage of an enhanced feature set such as:

- Support for more devices and OS platforms.
- Better app management. Fine grained policy management.
- Improved administration and provisioning
- Advanced connectivity and networking
- Expanded compliance and integrity checking
- Additional ways to enroll user certificates

For more information on why you should upgrade to BlackBerry UEM, see Benefits of Upgrading from Good Control to BlackBerry UEM. A future release of this document will include updated BlackBerry UEM information.

Good Control Server

The GC server manages users, devices, policies, BlackBerry Dynamics apps, and GP servers. Collection of all GC servers and GP servers constitutes a BlackBerry Dynamics deployment. A BlackBerry Dynamics deployment may have multiple GC servers.

The IT administrator uses the GC server to create and manage users, provision access keys, manage access, policies, and app entitlements. The administrator can create different policies based on security needs for different users. Within a security policy, the administrator controls various requirements and conditions such as user authentication, user password strength, auto lock, copying of data outside of the secure BlackBerry Dynamics container, compliance with OS version, hardware manufacturer or models, jailbreak and rooted detection. To find more information about security and compliance policies please refer to admin guide in references section.

The IT administrator can also use the GC server to define and deploy apps to groups of users and set app-specific configuration data. Capability also exists to allow or deny apps to individual users.

Using a GC server an IT administrator has precise control over which app servers and domains a BlackBerry Dynamics app can establish a connection with. The GC server also manages all GP servers installed inside the enterprise. GC server communicates with the GP Server over authenticated HTTPS connection.

An IT administrator may install multiple GC servers for load balancing and fault tolerance. The list of available GC servers is sent to the BlackBerry Dynamics runtime by the GC server.

Certificate Authority

At the time first GC server is installed, GC server generates a 2048bit RSA key pair (GD CA). The public key for this key pair is self-signed and acts as certificate authority for the BlackBerry Dynamics deployment. Each instance of the GC server also creates an intermediate certificate authority (GC ICA), which is signed by the GD CA. See section on Certificates below to see other places where GD CA is used.

Administrator Roles and Rights

With Role-Based Access Control (RBAC), your organization can easily restrict access to GC functions and offload a group of tasks to certain administrators or help desk support specialists or without compromising internal policies and requirements. Role privileges are enforced globally across all GC servers in your deployment, so administrators have the same rights and access for any GC server they log into. Custom roles can be created by giving them a smaller set of rights.

An administrator can have multiple roles. In this case, the administrator inherits the cumulative rights granted to all roles to which the administrator's account belongs.

Perhaps the most critical group of rights is the Container Management. For example, **View Full Access Keys for All User** right allows administrators to view complete access key for any user.

GC server administrators are authenticated against enterprise Microsoft Active Directory (AD). Administrators have option of providing their AD password or use Kerberos single sign-on. Instructions for enabling and setting-up Kerberos authentication and Kerberos Constrained Delegation in BlackBerry Dynamics can be found at docs.blackberry.com.

Installation

Installation of the first GC server requires a serial number and license key issued by Blackberry. The GC server will register with the NOC, over HTTPS, using the serial number and license key. During this process, a binary login key is negotiated. The binary login key is used in future sessions, along with the serial number and license key, to authenticate to the NOC.

Each GC creates a 2048bit RSA key pair for hosting TLS connection (GC TLS cert). This certificate is signed by the GC ICA. For example: this TLS certificate is used to host GC console that admin/user login to. This TLS certificate key pair can be changed by the administrator by installing certificate issued from a well-known certificate authority (CA).

During installation, an overall administrator (super-admin) is assigned. The super-admin must already be present in the corporate directory. Additional administrators, with different roles, can be added by the super-admin, but

they must also be present in the corporate directory. Only designated administrators can log in to the GC console and manage users, containers and apps. All actions taken by administrators are logged for audit purposes.

All policy, configuration, and container information are saved by the GC server in the database provided by the IT administrator during the GC server installation. Access to the database server is authenticated.

In order to install additional GC servers in the same BlackBerry Dynamics deployment, admin can get the license key from the any of the existing GC servers. If a completely new and independent BlackBerry Dynamics deployment is desired, then a new license key needs to be requested from BlackBerry Developer Network portal.

Good Proxy server

The Good Proxy (GP) server enables a BlackBerry Dynamics app to connect to app servers that are inside the enterprise firewall.

GP servers can be grouped in a cluster and associated to app servers or sub-domains. An administrator can specify which GP cluster will be used to complete connection at sub-domain level or at individual app server level.

The GP server only allows connections from BlackBerry Dynamics apps to the app servers that have been permitted by the IT administrator through the GC console. A GP server enforces the app server connection request at the user level.

The GP server connects to the app server over TCP to complete the connection to the app server. The last leg of the connection between the GP server and the app server is entirely inside the enterprise firewall and is not encrypted by default. If private communication is required over this connection, the BlackBerry Dynamics app can use TLS, which is supported by the BlackBerry Dynamics runtime.

The GP server also receives the BlackBerry Dynamics app activation requests from the BlackBerry Dynamics NOC and forwards the request to the GC server.

The GP server also provides an encrypted push-service to the app servers.

There is no user interface for the GP and neither does it save any data to permanent memory. An IT administrator may install multiple GP servers for load balancing and fault tolerance. A list of available GP servers is sent to the BlackBerry Dynamics runtime by the GC server.

Installation

Only a GC administrator can install a GP server. During installation, the GP server registers with the GC server and is assigned a license key and server name. With these credentials, the GP server authenticates itself to the NOC and negotiates a login key. Subsequent connections to the NOC require both license key and login key for authentication. The GP server connects to the NOC using TLS. The GP server's connection to the GC server is also authenticated with the authentication token established during the installation process.

During installation, the GP server generates a 2048bit RSA key pair. The public certificate for this key pair is signed by the GC server using GC ICA private key. This key pair is used to host HTTPS connection for the GC server, and the app servers (for push/GNP service). This key pair can be changed by the administrator by installing certificate issued from a well-known CA.

BlackBerry Dynamics Network Operations Center (NOC)

The BlackBerry Dynamics NOC has multiple types of servers running in the cloud, hosted by Blackberry, along with a database server.

Mobile Data Conduit (MDC) Server

All BlackBerry Dynamics apps maintain a persistent connection to this component of the NOC by using TLS over TCP. This connection is authenticated. It activates the BlackBerry Dynamics apps, delivers GNP push messages to the BlackBerry Dynamics runtime, and sends enterprise connection set-up and enterprise activation messages to the GP server.

Enterprise Gateway

The GC and GP servers which are running inside the enterprise firewall connect to this component of the NOC. The GP server is connected constantly, whereas the GC server establishes a connection as required. All connections are authenticated using the license key and login key established during server registration.

Relay Server

BlackBerry Dynamics apps establish a connection to their enterprise GP server through this component. Both BlackBerry Dynamics apps and GP servers connect to this component by using the Good Relay Protocol (GRP) over TCP. See Good Relay Protocol.

Catalog Server

This component provides an app life-cycle and management service to the BlackBerry Dynamics app and to the GC server.

The GC server uploads app metadata and sets permissions for users or groups of users. The GC server uses its license and login keys to authenticate to the Catalog server over HTTPS. All actions taken by the IT administrator (using GC) on the Catalog server are logged for audit purposes.

The BlackBerry Dynamics runtime checks for app entitlement for its user and wipes or locks the app when entitlement for the installed app is revoked. The BlackBerry Dynamics runtime uses a GNP token to authenticate itself to the Catalog server over HTTPS.

Dynamics Runtime Connections to the BlackBerry Hosted Services

All connections from Dynamics runtime to BlackBerry hosted services use only TLS v1.2 with strong cipher suites and root certificate authorities are pinned to a small set included in the SDK.

App Activation

App activation is the process by which a BlackBerry Dynamics app is activated with both the enterprise's GC server and the NOC. As a result of activation, the BlackBerry Dynamics app can be managed by the GC server and has permission to connect to the appropriate app servers inside the enterprise firewall. The BlackBerry Dynamics runtime has the keys to securely communicate with the NOC and GP servers in the enterprise. It receives initial provisioning and configuration data from the GC server and the NOC on successful completion of this process.

To start the activation process, the user needs an access key. The access key is random 15-digit alpha-numeric data [a-z, 0-9] (same strength as random 72bits) and is delivered in an email when the GC administrator adds the user in the GC console. The access key expires after a set time as configured on the GC server and can be used to activate only one BlackBerry Dynamics app.

When the user is added to the GC console and issued an access key a PBKDF2 hash (HMAC-SHA512, 16384 iterations, access key also acts as salt, 32 bytes long output) of the access key is sent to the NOC over an HTTPS connection. The access key is only available to the user and the GC server.

Activation Process

When a user installs a BlackBerry Dynamics app and performs the set-up, the activation process starts. First, the user is asked to enter an email address and an access key. The BlackBerry Dynamics runtime will then:

1. Activate against the NOC.
2. Establish end-to-end secure channel with GC server by performing authenticated ECDH parameter exchange. This eliminates the possibility of someone in the BlackBerry Dynamics NOC being able to execute a MITM attack.
3. Receive encrypted provisioning data from the GC server.

Step 1: Activate against the NOC

The BlackBerry Dynamics runtime performs activation by sending the email address and hash of the access key over an TLS connection to the NOC. If this credential (email address and hash of access key) is active, then the NOC sends provisioning data such as the Master Link Key, the Relay server, a list of GP servers and the GC server, to the BlackBerry Dynamics runtime.

Step 2: Establish secure channel with GC server

The BlackBerry Dynamics runtime establishes an ECDH shared secret with a GC server belonging to the enterprise by sending secure channel setup messages to the NOC over TLS. The NOC then forwards this message to the GP server over an HTTPS connection. The GP server, in turn, forwards this message to a GC server over HTTPS (See Figure 5). Secure channel setup message contains a user identifier (email address), ephemeral ECDH public key (from nist-curve P521), salt, GNP token, and MAC of the message to authenticate the sender, and integrity of the message. MAC is computed using HMAC-SHA512, with key (MAC key) derived from the access key using PBKDF2 (HMAC-SHA512, 16384 iterations, salt exchanged in the message, 32 bytes long output). The GC server returns the

response to the BlackBerry Dynamics runtime by using the GNP mechanism. The GC server's response contains its own ephemeral ECDH public key and MAC computed in the same way as the request above.

Step 3: Receive encrypted provisioning data

The BlackBerry Dynamics runtime requests enterprise provisioning data from the GC server (in the same way as step two above by sending the request message to the GC server via the NOC and GP server). The GC server returns encrypted provisioning data such as the Master Session key, app configuration data and a list of GP servers, over the GNP. The GC server encrypts the provisioning data response using an AES CBC cipher with an activation encryption key (256 bit long) derived from the ECDH shared secret established from the exchanges in step two. An ANSI X9.63 key derivation function is used to derive the activation encryption key. The provisioning data request and response also has a MAC computed in the same way as in Step Two above.

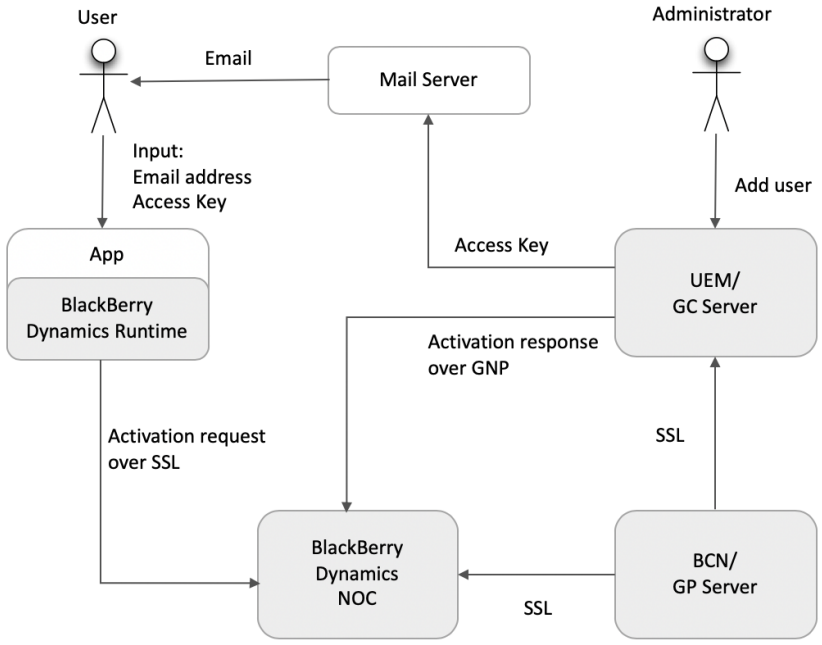


Figure 5: Components and protocols involved in app activation

Easy Activation

Easy Activation simplifies the provisioning process by allowing a BlackBerry Dynamics app to "hand off" activation to an app already installed on the device that can act as the activation delegate. The user has to retrieve and manually enter an access key for the first BlackBerry Dynamics app activated on a device.

Easy Activation is supported for Android and iOS platforms.

When a user installs a new BlackBerry Dynamics app and starts the provisioning process, the new app will check for the availability of a suitable installed activation delegate app. If an activation delegate is not discovered, the user will be prompted to use the standard provisioning process with an access key. If a suitable activation delegate app is detected, the "Easy Activation" setup option is also presented to the user.

When "Easy Activation" option is selected by the user, activation delegate app is launched. User is asked to provide the container password to authenticate the request. After authenticating the user, activation delegate app requests an access key from the GC. This access key is returned back to the app that requested easy activation. App being activated will now complete the activation process in the same way as if user entered the access key.

The new BlackBerry Dynamics app and the activation delegate app exchange the request and response over secure ICC channel (as described in Secure ICC Handshake). In its request for an access key, the new BlackBerry Dynamics app also sends a randomly generated nonce (term in cryptography meaning "a number or bit string used only once "). The access key received in this process can only be used along with the nonce used to create it. Thus, only the BlackBerry Dynamics app that requested the access key can complete the activation using it.

Enterprise Connectivity

The following protocols provide the capability for a BlackBerry Dynamics app to connect to the app servers behind the enterprise firewall and provide app server capability to push notification messages back to the BlackBerry Dynamics app.

Good Relay Protocol

The Good Relay Protocol (GRP) provides a secure, two-way, communication-channel between the BlackBerry Dynamics runtime on the device and the GP server inside the enterprise firewall. The GRP is a two-layer protocol which is discussed here in terms of a lower layer and an upper layer.

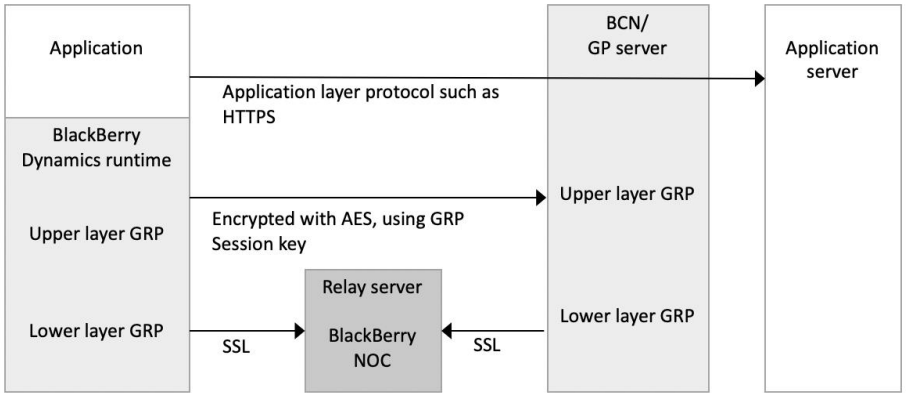


Figure 6: Encryption and keys used in GRP

The lower GRP layer enables BlackBerry Dynamics runtime and GP server to connect to the Relay server hosted inside BlackBerry NOC. The lower GRP layer has two segments both of which use a TLS connection over TCP:

- The first segment is between the BlackBerry Dynamics runtime and the relay server. This connection is authenticated with the Master Link Key that was set up when the client was provisioned.
- The second segment is between the relay server and the GP server. The GP server license key is used to authenticate this connection.

The upper GRP layer is the end-to-end protocol between the BlackBerry Dynamics runtime and the GP server that carries app data. This layer is encrypted with a session key that is not known to the relay server. In this layer BlackBerry Dynamics runtime informs GP server which app server to connect to.

The GRP authentication service is used to negotiate a GRP authentication token, and a GRP session key for the upper GRP layer. The GRP authentication token is used by the BlackBerry Dynamics runtime to authenticate itself to the GP server for each connection. The GRP session key is used for encrypting data for each upper GRP layer

connection. A GRP authentication token and GRP session key can be used for 24 hours; after which they expire. The GRP authentication token and GRP session key are exchanged securely over GRP using a key (GRP Key Exchange key) derived from the Master Session Key (the key exchanged during app activation). HMAC-SHA512 message authentication code is used to ensure integrity of the GRP authentication service.

GRP Key Exchange key is iterated HMAC-SHA512 of Master Session key plus some fixed constants.

Direct Connect

The BlackBerry Dynamics Direct Connect feature (referred to as Direct Connect) provides BlackBerry Dynamics clients ability to connect to GP servers without connecting through the BlackBerry Dynamics NOC.

This feature benefits following entities:

- Users whose BlackBerry Dynamics apps experience long connection establishment latency and low throughput due to large TCP round trip time (RTT) between BlackBerry Dynamics apps and the NOC, or between GP servers and the NOC.
- Organizations which have additional data privacy requirements which restrict user data leaving from their networks and do not want their user and enterprise data to be routed via the BlackBerry Dynamics NOC.

An enterprise administrator can enable this feature by configuring Direct Connect for each GP server in the server settings section in the GC console. When this feature is enabled, BlackBerry Dynamics clients will make connection to the GP directly instead of connecting via the BlackBerry Dynamics NOC to connect to an app server inside the enterprise network.

Deployment Models

There are two deployment configurations for Direct Connect.

1. A GP server is deployed in the DMZ where it is reachable from the Internet. DMZ is also known as perimeter network.
2. A HTTP Proxy server in the DMZ which forwards the connection from the BlackBerry Dynamics client to the GP server. This server is referred to as a DMZ proxy.

In both configurations a BlackBerry Dynamics client establishes an TLS connection to the GP server and authenticates to the GP server over GRP as documented in Good Relay Protocol. However, in the case of Direct Connect, no encryption is performed by the GRP. The TLS connection uses TLS 1.2 and negotiates ECDHE-RSA-AES256-SHA384 cipher suite. The GP server then uses the TLS certificate signed by the GC server (see Installation) to authenticate to the BlackBerry Dynamics client.

Connections to the enterprise app server will never be attempted via BlackBerry Dynamics NOC when configured for Direct Connect.

When connecting to the GC server, if the BlackBerry Dynamics client fails to connect to the GP server directly, its failover operation is to attempt to connect through the Relay server in the BlackBerry Dynamics NOC.

The figure below shows the protocols in use for Direct Connect.

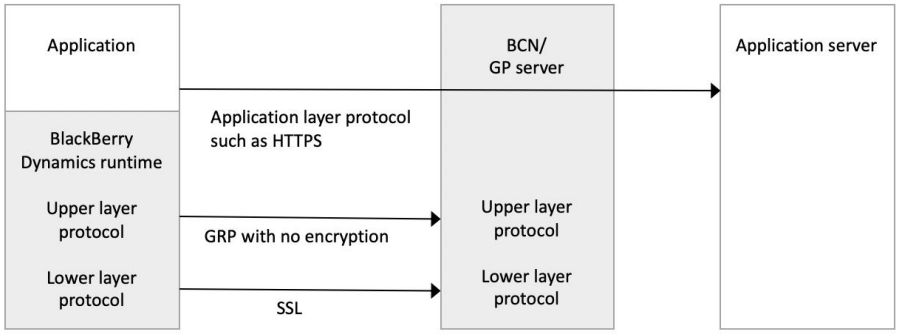


Figure 7: Layers and protocols in Direct Connect

GP in the DMZ

In this configuration, an enterprise admin installs a GP server in the DMZ. The GP server must be reachable from the Internet. GC servers and all enterprise app servers used in the BlackBerry Dynamics deployment must also be reachable from the GP servers in the DMZ.

BlackBerry Dynamics clients will then establish TLS connections to the GP as documented above in **Deployment Models**. Externally reachable GP server hostname can be set on the GC.

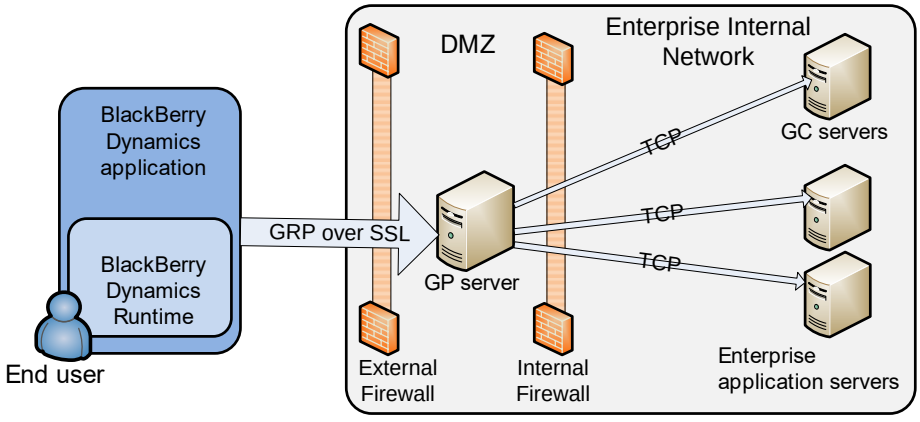


Figure 8: GP in the DMZ

DMZ Proxy

In this configuration, an enterprise admin installs an HTTP forward proxy server in the DMZ that supports HTTP Connect. The GP server would remain inside the internal network as in a typical BlackBerry Dynamics deployment. In this model, only GP servers would be reachable from the DMZ proxy as opposed to multiple app servers being exposed to the DMZ in the previous model.

The enterprise admin must provide the FQDN of the DMZ proxy in the GC console and associate the DMZ proxy with the GP server on the server settings screen.

BlackBerry Dynamics clients will first make an HTTP Connect request to the DMZ proxy, and request a connection to the GP server. No authentication is done against the DMZ proxy. The DMZ proxy completes this connection to the GP server, and the BlackBerry Dynamics client then establishes an TLS connection to the GP, as described in **Deployment Models**). The BlackBerry Dynamics client authenticates to the GP server over GRP. If the BlackBerry Dynamics client fails to authenticate, connection is terminated.

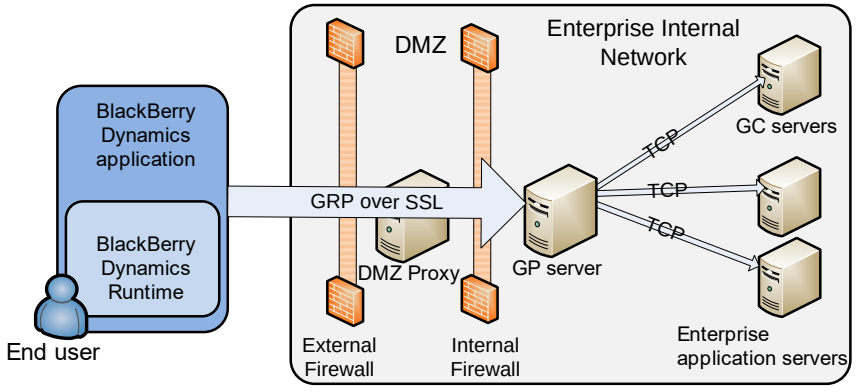


Figure 9: DMZ Proxy

Table below compares the three different connection models supported by the BlackBerry Dynamics.

Connection Model	Authentication	Encryption	DMZ connection requirements	Intranet connection requirements
Via BlackBerry Dynamics NOC	By the BlackBerry Dynamics NOC and GP server	AES 256 by GRP	Outbound	Outbound
Direct Connect with GP server in the DMZ	By the GP server. Authentication done in the DMZ.	AES 256 by TLS protocol	One inbound IP address per GP server	Multiple inbound IP address, one per app server

Direct Connect with DMZ Proxy	By the GP server. Authentication done inside the internal network.	AES 256 by TLS protocol	One inbound IP address per DMZ proxy	One inbound IP address per GP server

Good Notification Push

The Good Notification Push (GNP) service allows notification messages to be pushed from an app server to a BlackBerry Dynamics app. An app server needs a valid GNP token to send the message either to the MDC server inside BlackBerry Dynamics NOC or to the GP server inside enterprise firewall. The notification messages are transmitted via the MDC server (see Mobile Data Conduit (MDC) Server) to which the BlackBerry Dynamics runtime maintains a persistent TCP connection. When the app server sends notification messages to the MDC server, the messages are visible to the BlackBerry Dynamics NOC.

After a BlackBerry Dynamics app has completed app activation, its app server can use the encrypted GNP service provided by the GP server to push a notification message. The GP server will encrypt the messages pushed by the app server so that only intended BlackBerry Dynamics runtime can decrypt the message. In this case notification message is not visible to the BlackBerry Dynamics NOC.

App servers may send GNP commands to any one of their enterprise's GP servers. The GP server then encrypts the notification message, with an AES CBC (256 bit) cipher using the GRP session key and sends the encrypted notification message to the MDC server. An HMAC-SHA512 digest is used to ensure the integrity of the notification message. The encrypted notification message is then delivered by the MDC server to the BlackBerry Dynamics runtime. The BlackBerry Dynamics runtime decrypts this message, and delivers the pushed content to the BlackBerry Dynamics app.

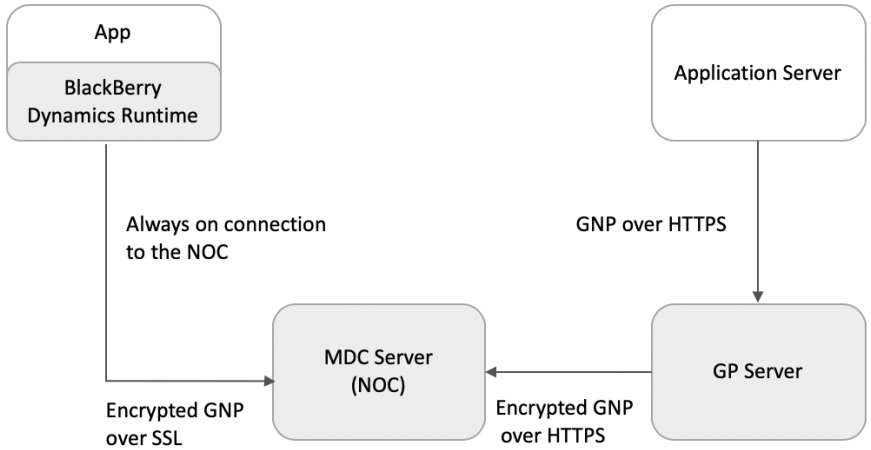


Figure 10: Push from App Server to the BlackBerry Dynamics app

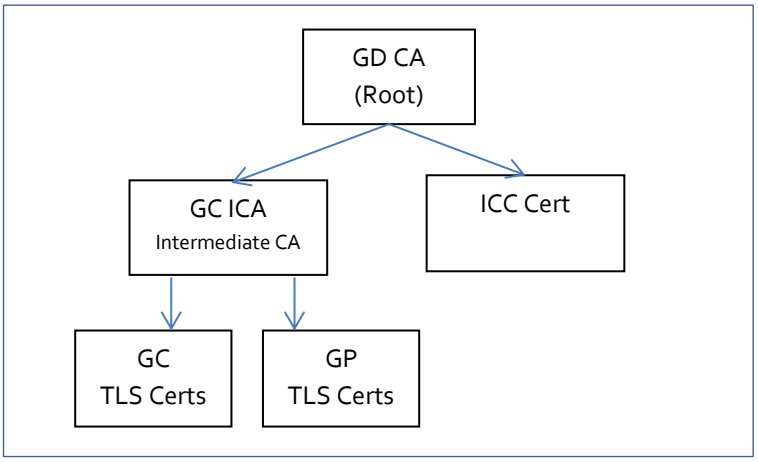
Certificates

X.509 certificates are used in the BlackBerry Dynamics deployments for various purposes such as in client and server authentication for TLS connections, SMIME emails. Certificates are used for the app layer logic and as well by the BlackBerry Dynamics runtime and GC and GP serves for the management purposes which are called Infrastructure Certificates.

Infrastructure Certificates

A BlackBerry Dynamics deployment by default creates a certificate authority (GD CA), which is internally used to issue TLS certificates for various BlackBerry Dynamics components. The BlackBerry Dynamics runtime also trusts this certificate authority and is provided with the GD CA public certificate at the time of activation.

This section describes the certificate hierarchy supported in the BlackBerry Dynamics deployment and various certificates in play and their purpose.



Certificates in Use

GC TLS Certificate is used in following connections as server-side certificate

1. TLS connections initiated by browsers to the GC server (GC Console UI).
2. TLS connections initiated by GP server to GC server.
3. TLS connections initiated by BlackBerry Dynamics apps to the GC server.

GP TLS Certificate is used in following connections as server-side certificate

1. TLS connections initiated by the GC server to the GP server
2. TLS connections initiated by BlackBerry Dynamics apps to GP servers (Direct Connect mode only)

3. TLS connections from the app servers to the GP server (when GNP push is used or when BlackBerry Dynamics Auth Token needs to be verified)

ICC Certificate

ICC certificate is issued to each activated BlackBerry Dynamics app.

1. ICC certificate is used for both client cert and server cert when connecting from one BlackBerry Dynamics app to another BlackBerry Dynamics app
2. ICC certificate may additionally be used as client cert in Direct Connect mode when configured by the IT admin to authenticate to an optional web proxy in the DMZ.

GC ICA

GC ICA is intermediate certificate authority that signs the GC and GP TLS certificates.

GD CA

GD CA is a root authority. It is self-signed. It is used to sign GC ICA and ICC certificates.

BlackBerry Dynamics Network Operation Center

Servers in the NOC use TLS certificate issued by Thawte to host the TLS connections.

Enterprise Certificates

An enterprise administrator can control certificates that are trusted for the app client to server connections and can issue certificates for each user device.

Trusted Certificate Authorities

An enterprise administrator can specify public certificates that should be trusted for the app client to server connection by the BlackBerry Dynamics runtime. In addition, administrator can control if the certificate authorities present in the device OS are to be trusted for the TLS connections. This feature is independent of the MDM control which allows administrator to add trusted certificate authority to the device key store.

User Certificate Usage

Sometimes a user may need a public-private key-pair to send and receive signed or encrypted emails (SMIME) or to authenticate to an app server instead of a user password.

BlackBerry Dynamics runtime supports user authentication on TLS connections using client (user) certificates. In addition, BlackBerry Dynamics runtime also supports Kerberos authentication in combination with user certificates for pre-authentication (PKINIT). Client App developer does not need to do any implementation work to use this feature.

Additional details about PKINIT and controlling which users can get certificates, can be found in the Good Control admin guide.

User Certificate Enrollment

An administrator can enroll the user certificates (pubic private key-pair) to a BlackBerry Dynamics runtime from many different sources. User certificates can be mandatory i.e. Dynamics runtime after activation will require and complete the user certificate enrollment. Only after user certificate is enrolled, user can start using the app. Once the certificate is enrolled in one Blackberry Dynamics app, all other BlackBerry Dynamics apps on the same device will share the same certificate. Sharing of the user certificate from one BlackBerry Dynamics app to another BlackBerry Dynamics app uses secure ICC framework. User's private key is never provided to the app layer.

Manual Enrollment The administrator or the user can upload their own pubic private key-pair in the pkcs12 formatted file from the management console, which is then sent to all user's BlackBerry Dynamics apps (runtime). In addition, user's pkcs12 file is removed from the management server after pre-configured time.

Dynamics PKI Connection

This supports creation of custom PKI Connector server by the customer to interoperate with their own certificate issuer infrastructure. Your PKI Connector server must implement the HTTP interfaces as documented in the Blackberry Dynamics user certificate management protocol link in References. When this provider is configured, BlackBerry Dynamics apps will automatically make a request to the GC server after activation. The GC server makes the request for credentials to the enterprise's BlackBerry Dynamics PKI connector. In addition, an administrator can require a user to input an additional one-time password (issued by the enterprise certificate issuer infrastructure) on the BlackBerry Dynamics app to get the certificate. This one-time password is sent to the enterprise BlackBerry Dynamics PKI connector for authentication. The Enterprise BlackBerry Dynamics PKI connector returns back credentials in pkcs12 format.

Microsoft NDES SCEP Connection

This is only supported in BlackBerry UEM deployment. Dynamics runtime uses SCEP to perform certificate enrollment directly against Microsoft NDES server. User is not prompted for password. SCEP password is never sent to the Dynamics runtime. Instead UEM server assists Dynamics runtime with creation of SCEP payloads (see reference: Assisted certificate enrollment).

Entrust SCEP Connection

This is only supported in BlackBerry UEM deployment. Dynamics runtime uses SCEP to perform certificate enrollment directly against Entrust CA server (for example Entrust IdentityGaurd server). User is not prompted for password. SCEP password is never sent to the Dynamics runtime. Instead UEM server assists Dynamics runtime with creation of SCEP payloads.

Entrust IdentityGuard based Smart Credentials

This is only supported BlackBerry UEM deployments. The BlackBerry UEM Client is required. The BlackBerry UEM Client interacts with Entrust IdentityGuard to enroll user credentials. Public key is sent to the Entrust server for signing. The user is required to enter or scan a password QR code. Entrust IdentityGuard supports derived credentials.

Purebred App based Derived Credentials

This is only supported in BlackBerry UEM deployments. The Purebred solution provides a client and a server. The Purebred client app receives user certificates of different types (signing, encryption, authentication) on the device from the Purebred server. On iOS platform, the BlackBerry UEM Client on the device requests certificates from the Purebred client app when triggered by the user.

However, on Android platform, the Purebred client app adds the user certificates (key-pair) to the device key store. The BlackBerry Dynamics runtime then calls out the platform APIs to perform cryptographic operations such as signing or encryption.

Additional Features

BlackBerry Dynamics provides the following additional features

- Delegate authentication and password security from one BlackBerry Dynamics app to another.
- Securely exchange data between two BlackBerry Dynamics apps.
- Enable IT administrators to manage app specific policies from the GC server.
- Securely provide a user's identity to an app server.
- Add security and manageability to an iOS app by wrapping.

Authentication Delegation

The authentication delegation feature allows one BlackBerry Dynamics app to hand off user authentication to either another BlackBerry Dynamics app (or to a Good for Enterprise app), which is running on the same device. This feature is supported on Android and iOS platforms.

To set up authentication delegation, the IT administrator must identify the app that will act as the authenticator from the GC console. Any BlackBerry Dynamics app if designated by the GC administrator can act as the authenticator.

Authentication Delegate: Any BlackBerry Dynamics app selected by the admin to perform user authentication.

Active Authentication Delegate: The BlackBerry Dynamics app that is performing the task of user authentication on the device. A device may have one or more active authentication delegates. Active authentication delegate app also acts as Easy Activation Delegate (see section: Easy Activation).

Process for Delegating

To perform authentication delegation, the designated authenticator app must be first installed and activated by the user. When the authenticator app has activated, the user sets a password for it. This password is then used to authenticate the user on all the BlackBerry Dynamics apps on the same device.

Setting Delegation

When the next BlackBerry Dynamics app is activated on the same device and requires a password to be set, it invokes the authenticator app to set user credential. The newly activated BlackBerry Dynamics app is informed about the apps that can act as authentication delegates during the activation process. In addition, GC provides the native app identifier (bundle ID for iOS, package name for Android OS) to be used, to send the request to the authenticator using secure ICC handshake (as described in Secure ICC Handshake).

The newly activated BlackBerry Dynamics app requests the authenticator app to provide the User Key (defined in User Authentication and Key Storage) instead of asking the user to set a security password. This User Key is then used to secure the contents of the BlackBerry Dynamics app as described in User Authentication and Key Storage.

The authenticator app may prompt the user for a password depending on its own state. The User Key is different for each BlackBerry Dynamics app that requests authentication delegation. The User Key is derived by the authenticator app using an authentication delegation key and a salt. The salt is the mobile OS specific app address (bundle ID/package name) of the app requesting authentication delegation. The authentication delegation key is sent by the GC server during activation.

Using Delegation

When a BlackBerry Dynamics app is started (post activation) or when it is in locked state, it typically asks the user to provide password to authenticate the user. A BlackBerry Dynamics app that has delegated authentication to another BlackBerry Dynamics app will invoke the authenticator app. The authenticator app authenticates the user by asking the authenticator app password. If the authentication is successful, the authenticator app returns the User Key, which is used to unlock the app's BlackBerry Dynamics container.

Authenticator app also ensures the correct BlackBerry Dynamics app gets the User Key by using native OS services as described below.

Multiple Authentication Delegates

GC server also allows administrator to specify more than one authentication delegates. This list of delegates is a prioritized list. Multiple authentication delegate feature allows administrator to

a. Migrate from the active authentication delegate app to a new authentication delegate app.
b. Support the case where a given authentication delegate is only available on one platform. Admin can provide authentication delegates available for each platform.

BlackBerry Dynamics runtime uses this list to pick the authentication delegate immediately after activation. BlackBerry Dynamics runtime will select the highest priority authentication delegate present at the time to get the User Key and will continue to use that app as active authentication delegate when container needs to be unlocked. When a new app is activated, which is a higher priority authentication delegate, it will ask the user to set a password. Subsequently, existing apps, when unlocked next time, will update their authentication delegate to the newly activated authentication delegate app. The active (current) authentication delegate must be present at this time and user may be required to authenticate (i.e. enter password) in the active authentication delegate before the switch to the new authentication delegate can be completed.
If the current authentication delegate app is removed by the user, then user will not be able to unlock other apps present on the device (which delegated authentication to the deleted app). In this case it is recommended that user reinstall and activate the deleted app. Alternately, it is possible for the user to request for temporary passwords to unlock each app present on the device and delegation authentication to an alternate app. Apps unlocked in this manner will then immediately delegate authentication to the highest priority authentication delegate present on the device.
Admin can also enable self-authentication. When this is enabled, it acts as lowest priority authentication delegate. If none of the higher priority authentication delegates are present at the time of activation, then app being activated will ask the user to set the password itself.

Secure ICC Handshake

Secure ICC handshake is used for Authentication Delegation, Easy Activation, and during App Kinetics. Secure ICC handshake establishes a secure channel between two apps on the same device when User Key is being requested for Authentication Delegation or when access key is being requested for Easy Activation.

This process uses following ciphers.

- An ECC curve P-521 is used for ECDH and an ANSI X9.63 key derivation function with SHA-512 as the underlying hash function for symmetric key derivation.
- Data exchanged by the ICC is encrypted by the AES-CBC cipher using 256bit key negotiated using ECDH key exchange.
- The IV is returned in the clear by the BlackBerry Dynamics Authenticator app.

iOS

Source app uses the openURL API to send the request to the destination app. The destination app uses the source app (iOS app identifier: bundle ID) information provided by the openURL iOS API to identify which app is requesting the service and to send the response back to it.

Android

On Android, Intent is used to request service from the other app. The source app uses Android's app identifier Package name (sent by the GC server/BlackBerry Dynamics NOC) to request service by using explicit Intent. Inside the request it sends its own Package Name. The destination app sends the response back to app identifier sent inside the request. BlackBerry Dynamics depends on the OS to send the response to the identified app.

Shared Services Framework

The Shared Services framework (formerly known as AppKinetics™) allows BlackBerry Dynamics apps or servers to expose their services/capabilities to other BlackBerry Dynamics apps. A service that is provided by one BlackBerry Dynamics app may be utilized by another BlackBerry Dynamics app. The BlackBerry Dynamics runtime provides API's for the BlackBerry Dynamics apps to discover services present on the device or on a server.

- For more information, see BlackBerry Dynamics Shared Services Framework

.

App-Based Services

The BlackBerry Dynamics runtime provides the means to securely exchange service requests and responses over a socket connection between service provider and service consumer BlackBerry Dynamics app on iOS and Android platforms.

The process of establishing secure socket connection between the two BlackBerry Dynamics apps is a two-step process as shown below:

1. Exchange certificates and establish a port number using secure ICC handshake.
   This process is described in Secure ICC Handshake. After this step both apps know the identity of the other app and possess ICC certificate of the other BlackBerry Dynamics app.
2. Establish TLS connection.
   The service provider app's certificate is used for the TLS connection. The service consumer app presents its certificate for client authentication. Both sides trust only certificates exchanged during secure ICC handshake. The cipher suite for the TLS connection is TLS_RSA_WITH_AES_256_CBC_SHA256. The RSA key length for the certificates is 3072 bits.

Server-Based Services

The Shared Services framework also enables any BlackBerry Dynamics apps to use services provided by the servers. For an example presence service exposed by presence server could be consumed by an email app or a document reader app built by some other developer.

Entitlement of server-based services is done at user level by binding the service to a BlackBerry Dynamics App ID. This BlackBerry Dynamics App ID need not have a corresponding client app. Any BlackBerry Dynamics app activated for a user can discover the server-side services it has entitlement too. In addition, the Shared Services framework provides the list of app servers that can be used to request the service. The service provider can require user-level authentication before providing the service.

App-specific policies

This feature allows the IT admin to set policies/configurations which are specific to an app within the GC console. Changes to an apps policy are tracked by the GC server and sent to the impacted BlackBerry Dynamics runtime's over an TLS connection. App policy definition is published by the app developer in an XML format from the GC console or BlackBerry Dynamics portal. The app policy definition file is saved in the BlackBerry Dynamics NOC so that it is available to all GC servers in an organization.

Additional information about this feature is available in the *Technical Brief: Application Policies*

.

BlackBerry Dynamics Authentication Token

The BlackBerry Dynamics platform includes rigorous authentication of the end user. This is used when, for example, identifying whether the user is entitled to run the current app, and when applying security policies. The BlackBerry Dynamics Authentication Token (BlackBerry Dynamics Auth) mechanism enables apps to take advantage of the authentication processes of the BlackBerry Dynamics platform.

BlackBerry Dynamics Auth tokens can be requested by the BlackBerry Dynamics app on the device after the app has completed activation with GC. During the app activation users identify is established. Once a token has been issued, the app on the device can send the token to the app server at the back end. The BlackBerry Dynamics Auth token can then be checked by the app server, using a verification service provided by the BlackBerry Dynamics infrastructure. If the token is verified, the user's identity (email address), app identifier, app server name (for which token was requested), an optional challenge string is returned to the app server.

Internally, the integrity of the BlackBerry Dynamics Auth token is checked with a security token. This is a keyed hash (HMAC-SHA512) of the contents of the BlackBerry Dynamics Auth token with a key (GRP Auth token) that is only known to the BlackBerry Dynamics runtime and the GP server.

This BlackBerry Dynamics Auth token is also used by the BlackBerry Dynamics runtime to authenticate to the GC server when it connects to the GC server to get the policies.

Kerberos Constrained Delegation

The Kerberos constrained delegation (KCD) feature allows BlackBerry Dynamics clients to support Kerberos based authentication without requiring users to enter their domain passwords. See the references section to learn about KCD.

An advantage of using KCD is that since the user is never asked for their domain password, the user's domain password cannot be stolen while user is typing it on the mobile device. Additionally, in some deployments, the user does not even have a domain password, since hardware-based authentication tokens are used.

The KCD feature must be enabled by an administrator on the GC console. Additionally, the administrator must configure the Kerberos service account for the GC server in Active Directory as trusted, for Kerberos constrained delegation, for all the app servers which are to be authenticated by the use of this mechanism. When this permission is set, GC is able to fetch Kerberos service tickets on behalf of all users for the app servers which are set for constrained delegation in AD. The only restriction is that the user account, the GC Kerberos service account, and the app service account all must belong to the same domain in order for the GC to be able to fetch the service ticket.

When a BlackBerry Dynamics client is challenged to authenticate to an app server using Kerberos over HTTP/S, the BlackBerry Dynamics client requests a service ticket for the app from the GC server. This request is authenticated using BlackBerry Dynamics Auth token. The GC server, if it is enabled for KCD, attempts to fetch the service ticket from the Ticket Granting Service running on the Kerberos Key Distribution Center (KDC) on behalf of the user, and it sends the service ticket and unencrypted Kerberos session key for the app server to the BlackBerry Dynamics client. The BlackBerry Dynamics client uses these tokens to authenticate to the app server. The service ticket is cached by the BlackBerry Dynamics client as specified in the service ticket itself and used in future requests to the same app server. If the BlackBerry Dynamics client is unable to get the service ticket from the GC server, it prompts the user to provide their domain password and completes Kerberos authentication to the app server. This feature can be applied selectively for some app servers, if desired.

References

- BlackBerry Dynamics Administrator and Developer Overview
- Good Control Admin Guide
- Technical Brief: Inter-Container Communication
- Technical Brief: Application Policies
- Blackberry Dynamics Fingerprint Authentication
- Bypass Unlock: Application Developer Guide
- KCD overview
- FIPS Pub 140-2 by NIST
- Blackberry Dynamics user certificate management protocol
- Assisted certificate enrollment
- BlackBerry Dynamics Shared Services Framework

Acronyms/Glossary

Term	Definition
AD	Active Directory
AES, AES-CBC	Advanced Encryption Standard, - Cipher Block Chaining mode
AppKinetics	Former name for BlackBerry Dynamics Shared Services
ARM	Popular RISC-oriented instruction set architecture (operating system)
CA	Certification Authority
DMZ	"De Militarized Zone" or perimeter network
ECC	Elliptic Curve Cryptography
EDEK	"Encrypt Decrypt Encrypt", also called Triple DES (Data Encryption Standard)
FIPS	Federal Information Processing Standard
FQDN	Fully Qualified Domain Name of a host
GC	Good Control server
GNP	Good Notification Push
GP	Good Proxy server
GRP	Good Relay Protocol
HMAC-SHA	Hash-based Message Authorization Code with Secure Hash Algorithm
HTTP/HTTPS	Hyper Text Transport Protocol/Secured
ICC	Inter-Container Communication protocol
KCD	Kerberos Constrained Delegation
MD4, MD5	Message Digest 4 and 5
MDC	Mobile Data Conduit protocol
NOC	Network Operations Center: Collection of servers and services hosted by BlackBerry

Term	Definition
PBKDF2	PBKDF2 is a key derivation function that is part of RSA Laboratories' Public-Key Cryptography Standards series, specifically PKCS #5 v2.0, also published as Internet Engineering Task Force's RFC 2898.
RSA	One of the widely used public-key cryptosystems and is widely used for secure data transmission; named after Ron Rivest, Adi Shamir, and Leonard Adleman.
SDK	Software Development Kit
SSL	Secure Sockets Layer, predecessor to TLS
TLS	Transport Layer Security, successor to SSL
TUK	Temp Unlock Key