First Look: The InterSystems Public Key Infrastructure (PKI)

Version 2018.1
2018-01-31

InterSystems Corporation 1 Memorial Drive Cambridge MA 02142 www.intersystems.com

First Look: The InterSystems Public Key Infrastructure (PKI)
InterSystems Version 2018.1 2018-01-31
Copyright © 2018 InterSystems Corporation
All rights reserved.

InterSystems, InterSystems Caché, InterSystems Ensemble, InterSystems HealthShare, HealthShare, InterSystems TrakCare, TrakCare, InterSystems DeepSee, and DeepSee are registered trademarks of InterSystems Corporation.

InterSystems IRIS Data Platform, InterSystems iKnow, Zen, and Caché Server Pages are trademarks of InterSystems Corporation.

All other brand or product names used herein are trademarks or registered trademarks of their respective companies or organizations.

This document contains trade secret and confidential information which is the property of InterSystems Corporation, One Memorial Drive, Cambridge, MA 02142, or its affiliates, and is furnished for the sole purpose of the operation and maintenance of the products of InterSystems Corporation. No part of this publication is to be used for any other purpose, and this publication is not to be reproduced, copied, disclosed, transmitted, stored in a retrieval system or translated into any human or computer language, in any form, by any means, in whole or in part, without the express prior written consent of InterSystems Corporation.

The copying, use and disposition of this document and the software programs described herein is prohibited except to the limited extent set forth in the standard software license agreement(s) of InterSystems Corporation covering such programs and related documentation. InterSystems Corporation makes no representations and warranties concerning such software programs other than those set forth in such standard software license agreement(s). In addition, the liability of InterSystems Corporation for any losses or damages relating to or arising out of the use of such software programs is limited in the manner set forth in such standard software license agreement(s).

THE FOREGOING IS A GENERAL SUMMARY OF THE RESTRICTIONS AND LIMITATIONS IMPOSED BY INTERSYSTEMS CORPORATION ON THE USE OF, AND LIABILITY ARISING FROM, ITS COMPUTER SOFTWARE. FOR COMPLETE INFORMATION REFERENCE SHOULD BE MADE TO THE STANDARD SOFTWARE LICENSE AGREEMENT(S) OF INTERSYSTEMS CORPORATION, COPIES OF WHICH WILL BE MADE AVAILABLE UPON REQUEST.

InterSystems Corporation disclaims responsibility for errors which may appear in this document, and it reserves the right, in its sole discretion and without notice, to make substitutions and modifications in the products and practices described in this document.

For Support questions about any InterSystems products, contact:

InterSystems Worldwide Response Center (WRC)
Tel: +1-617-621-0700
Tel: +44 (0) 844 854 2917
Email: support@InterSystems.com

Table of Contents

First Look: The InterSystems Public Key Infrastructure (PKI)..... 1
1 Why a PKI Is Important 1
1.1 The Fundamentals of How Public-Key Cryptography and Certificate Authorities (CA) Work Together 3
2 About the InterSystems PKI 3
3 Trying the InterSystems PKI for Yourself 3
3.1 Configure Instance #1 as a CA Server 4
3.2 Configure Instance #2 as a CA Client 4
3.3 On Instance #2, Submit a Certificate Signing Request (CSR) to the CA Server 5
3.4 On Instance #1, Process the CSR 5
3.5 On Instance #2, Download Its Certificate and CA Server Certificate from the CA Server 6
3.6 Wrapping Up and Next Steps 6
4 For More Information about the InterSystems PKI 7

First Look: The InterSystems Public Key Infrastructure (PKI)

This First Look guide introduces you to the InterSystems public key infrastructure (PKI), which can play an important role in your organization's security strategy. It presents information about public-key cryptography, certificate authorities, and PKI. It then walks through some initial tasks associated with using the InterSystems PKI. Once you've completed this guide, you will have created a certificate authority (CA), and then requested and received a certificate from it for a CA client.

This First Look uses InterSystems IRIS™ default settings, which allows you to acquaint yourself with the fundamentals of the PKI without having to deal with other details that are important when performing an implementation. For the full documentation on database encryption, see "The InterSystems Public Key Infrastructure" in the *InterSystems IRIS Security Administration Guide*.

1 Why a PKI Is Important

The frequency of news about security breaches at many organizations makes clear the need to secure their communications. Organizations need to protect data that is traveling from one site to another or when there needs to be some kind of verifiable, legally binding digital signature. Powerful, effective, and pervasive tools to address these and other needs are *public-key cryptography* and *public key infrastructure (PKI)*.

Public-key cryptography enables the encryption and decryption of data. This provides a means of performing various actions related to securing data. These include protecting data in transit on an unsecured network (such as the Internet) or establishing the provenance of a document. It thereby enables crucial technologies, such as Transport Layer Security (TLS), which is the successor to the Secure Sockets Layer (SSL) and is the means by which browsers protect our connections to web sites.

Public-key cryptography operates on data controlled by distinct entities, where entities can be people, applications, organizations, and so on. However, public-key cryptography alone does not provide sufficient confidence of the identity of these entities in activities, particularly if they do not know each other personally. To achieve that level of confidence, there needs to be a larger structure that also provides trustworthy and verifiable identification information for entities involved. Such a structure is known as a PKI.

A PKI establishes a means for entities to be confident of each other's identities, even without any direct personal knowledge of or contact with each other. This requires each entity to trust a third party — called a *certificate authority (CA)* — that vouches for the identity of the other entity (also known as the other *peer*). With a PKI, entities can perform meaningful and legally binding cryptographic operations, which include encryption, decryption, and digital signing and signature verification.

InterSystems provides a public key infrastructure (PKI) that uses an instance of InterSystems IRIS as a CA, allows you to create key pairs, and allows you to create certificates that are associated with these key pairs. The InterSystems CA is suitable for use within organizations internally and in non-production environments. It is not recommended for use as a production or commercial CA; while its certificates are cryptographically sound, a commercial CA requires a level of organizational and legal infrastructure in additional to technological infrastructure.

Public-key cryptography with a PKI supports a number of vital secure activities:

- Digitally signing electronic documents
- Verifying the signature of electronic documents

- Encrypting communications between parties
- Encrypting documents

1.1 The Fundamentals of How Public-Key Cryptography and Certificate Authorities (CA) Work Together

When using public-key cryptography, each entity has a *private key*, which is a closely held secret, and a *public key*, which is made widely available. If you perform an action with one key, you can perform the complementary action with the other key; for example, if you encrypt data with your private key, then only your public key can decrypt that data. If someone else encrypts content with your public key, only your private key – and therefore, only you – will be able to decrypt it. This means that public-key cryptography provides a means for secure and private communications between two entities.

For public-key cryptography to be useful among entities who do not know each other and who cannot easily verify each other's identity, there needs to be a third party that both entities trust. This third party is the certificate authority (CA). Certificate authorities create certificates, which are digital documents that bind a public key to a set of identifying information for the public key's holder. Since the public key and the private key are inextricably tied to each other, a certificate also binds the identifying information to the private key. Some organizations have in-house CAs, which they use to support internal activities; other CAs operate as independent organizations, usually providing certificates as a commercial service. Commercial CAs typically offer certificates based on varying degrees of identity verification; with sufficient verification, a certificate can create a legally binding tie between an organization or person and a public-private key pair. The use of CAs allows entities in an unsecured environment to have sufficient confidence to use public-key cryptography in meaningful and legally binding ways.

Entities that are communicating with each other do not need to use the same CA. Rather, each one simply needs to trust the other's CA. This relationship of trusting a CA is usually established without any user intervention, such as by having a browser ship with a set of pre-approved CA certificates. In fact, an entity can trust one CA because it has a certificate from a second CA that is already trusted; in this scenario, the first CA is known as an intermediate CA — and there can be multiple intermediate CAs.

When an entity obtains a certificate from a CA, a number of events have occurred – frequently without being visible to the user. First, the CA client uses an algorithm to generate the key pair; the CA client then obtains necessary information to describe the entity using the key pair, which has to do with the entity's location, organization, and so on. Taken all together, this identifying information comprises a *distinguished name (DN)*. The entity provides the public key and DN information to the CA in the form of a *certificate signing request (CSR)*; it does not provide the private key because, again, this is a closely held secret.

The CA receives the CSR and then processes it according to its procedures. The CA then signs a document that binds the public key to the DN information, thereby creating a certificate (specifically, a certificate that conforms to X.509 standard). Finally, the CA client obtains the certificate from the CA, and then can use it for various activities, such as establishing a TLS connection.

When two entities need to authenticate each other, they use their certificates and the CA's trusted relationship to them. Hence, when Alice and Bob attempt to communicate via, say, TLS, the TLS handshake performs authentication for each of them as follows:

- Alice ends up with Bob's certificate. Alice can trust this certificate because Bob's CA has signed it and Bob's CA is a trusted CA.
- The same is true for Bob with Alice's certificate.

2 About the InterSystems PKI

Taken all together, the activities of a CA and of public-key cryptography are part of what is called a *public key infrastructure (PKI)*. Hence, a PKI provides a means of creating and managing key pairs and certificates, and can support cryptographic operations including encryption, decryption, and digital signing and signature verification. InterSystems IRIS includes a PKI.

With the InterSystems PKI, you can set up a Certificate Authority (CA), a CA client, and start sending secured data between users in a matter of steps.

When an instance of InterSystems IRIS is acting as a CA, it is known as a CA server; when an instance is using a CA's services, it is known as a CA client. A single instance can be both a CA server and a CA client.

When establishing itself as a CA server, an instance of InterSystems IRIS either creates a key pair and then embeds the public key in a self-signed X.509 certificate or it uses a private key and X.509 certificate signed by an outside CA. X.509 is an industry-standard certificate structure that associates a public key with a Distinguished Name (DN).

3 Trying the InterSystems PKI for Yourself

It's easy to set up and use the InterSystems PKI. In this example, you will use two instances of InterSystems IRIS. You are going to perform a series of initial operations with instance #1 as a CA (here primarily known as a CA server) and instance #2 as a CA client. The steps are:

1. Configure Instance #1 as a CA Server
2. Configure Instance #2 as a CA Client
3. On the CA Client, Submit a Certificate Signing Request (CSR) to the CA Server
4. On the CA Server, Process the CSR
5. On the CA Client, Download Its Certificate and CA Server Certificate from the CA Server

Before You Begin

To try using the InterSystems PKI, you need two licensed instances of InterSystems IRIS. For basic information on installing InterSystems IRIS, the InterSystems IRIS Installation quick start.

Important: The example in this First Look simplifies the process of setting up and using a CA in ways that are not appropriate for a production system. For example, we provide a suggested password that is used to encrypt and decrypt the CA server's private key. On a production system (or anything other than a demo system such as this one), *never* use a publicly known password, as this could jeopardize the security of your CA's private key and therefore your entire PKI; if this key is exposed or compromised, *all* of a CA's certificates become untrustworthy.

Similarly, the directory for Certificate Authority's certificate and private key files is on the same machine as the instance of InterSystems IRIS that you used for this First Look's exercises. For a production system, this directory should always be on an external device (not a local hard drive or a network server), preferably on an encrypted external device. This is because the directory holds the private key of the CA.

When setting up your production system follow the instructions in the "InterSystems Public Key Infrastructure (PKI)" chapter of the *InterSystems IRIS Security Administration Guide*.

3.1 Configure Instance #1 as a CA Server

To configure instance #1 as a CA server:

1. In the instance #1 Management Portal, go to the **Public Key Infrastructure** page (**System Administration** > **Security** > **Public Key Infrastructure**).
2. On the **Public Key Infrastructure** page, under **Certificate Authority Server**, select **Configure Local Certificate Authority server**. This displays two fields:
   - **File name root for Certificate Authority's Certificate and Private Key files (without extension)** — Enter `FLCA` (First Look Certificate Authority). This uses `FLCA` as the name of the private key file and certificate file, so the private key is in FLCA.key and the certificate is in FLCA.cer.
   - **Directory for Certificate Authority's Certificate and Private Key files** — Enter `flca`. This places creates the <install-dir>/mgr/flca directory and places the FLCA CA certificate and private key files there.
3. Click **Next** to continue.
4. In the fields that appear, enter the following values:
   - **Password to Certificate Authority's Private Key file** and **Confirm Password** — Enter the password to encrypt and decrypt the CA's private key file. We recommend that you use `myflcapw`, so that you have a copy of the password here.
   - Under **Certificate Authority Subject Distinguished Name**, in the **Common Name** — Enter `First Look CA`, which identifies this CA.

   In a production environment, when you configure a CA server, you need to complete the fields in this section to include the email account of the user responsible for signing requests for the CA. For this First Look, we can skip it.
5. Click **Save**. InterSystems IRIS displays a message indicating success, such as:

```
Certificate Authority server successfully configured.
Created new files: C:\InterSystems\CAserver\mgr\flca\FLCA.cer .key, and .srl.
Certificate Authority Certificate SHA-1 fingerprint:
E3:FB:30:09:53:90:9A:31:30:D3:F0:07:8F:64:65:CD:11:0A:1A:A2
```

   This indicates that InterSystems IRIS has performed the following actions:
   - Created a key pair.
   - Saved the private key to a file to the location you specified and with the root name that you specified.
   - Created a self-signed CA certificate containing the public key.
   - Saved the certificate to a file to the location you specified and with the root name that you specified.
   - Created a counter of the number of certificates issued and stored it in an SRL (serial) file in the same directory as the certificate and the private key. (Each time the CA issues a new certificate, InterSystems IRIS gives the certificate a unique serial number based on this counter and then increments the value in the SRL file.)

3.2 Configure Instance #2 as a CA Client

To configure instance #2 as a CA client:

1. In the instance #2 Management Portal, go to the **Public Key Infrastructure** page (**System Administration** > **Security** > **Public Key Infrastructure**).

2. On the **Public Key Infrastructure** page, under **Certificate Authority Client**, select **Configure Local Certificate Authority Client**, which displays several fields on this page.
3. Complete them as follows, leaving other fields blank or with their defaults:
   - **Certificate Authority server hostname** — The host where the CA server is running. It's visible in the URL of instance #1's Management Portal.
   - **Certificate Authority WebServer port number** — The port number of the instance that is the CA server. It's visible in the URL of instance #1's Management Portal.
   - **Name** — The name `First Look CA client`.

   Note: If this were a CA client in a production system, in the **Local technical contact** area, you would also complete the **Phone number** and **Email address** fields. This would then allow the CA administrator to contact you to verify your identity. This is an important step in a production system, because the CA server should only provide certificates to clients where there is a verifiable identity and the clients need to provide this contact information to begin that process.
4. Click **Save**.

InterSystems IRIS acknowledges success through a message such as "Certificate Authority client successfully configured."

3.3 On Instance #2, Submit a Certificate Signing Request (CSR) to the CA Server

Next, on instance #2, submit a certificate signing request (CSR) to the CA server:

1. Still on the **Public Key Infrastructure** page (**System Administration** > **Security** > **Public Key Infrastructure**), under **Certificate Authority Client**, select **Submit Certificate Signing Request to Certificate Authority Server**, which displays several new fields.
2. Complete them as follows, leaving other fields blank or with their defaults:
   - **File name root for local Certificate and Private Key files (without extension)** — Enter `FLCAclient` (First Look Certificate Authority client). This uses `FLCAclient` as the name of the private key file and certificate file, so the private key is in FLCAclient.key and the certificate will soon be in FLCAclient.cer.
   - Under **Subject Distinguished Name**, in the **Common Name** field — Enter `FL CA client`.
3. Complete these fields as required and click **Save**. If successful, InterSystems IRIS then displays a message such as:

   ```
   Certificate Signing Request FLCAclient successfully submitted to the Certificate Authority
   at instance CASERVER on node FLCATEST.SAMPLE.COM.
   SHA-1 Fingerprint: C2:B0:D6:0D:D6:AB:43:DF:7F:B1:22:AE:14:D7:45:FF:CC:0C:20:D0
   ```

   In a production environment, you would need to make a copy of the SHA-1 fingerprint that InterSystems IRIS displays, as you — as the CA client — would need this information as part of the verification process for issuing a certificate.
4. At this point, you have used InterSystems IRIS to create and submit the CSR.

3.4 On Instance #1, Process the CSR

On instance #1 (the CA server), process the CSR, which turns it into a certificate:

1. In the Management Portal, go to the **Public Key Infrastructure** page (**System Administration** > **Security** > **Public Key Infrastructure**).

2. On the **Public Key Infrastructure** page, under **Certificate Authority Server**, select **Process pending Certificate Signing Requests**, which displays the pending CSR from the CA client.
3. Click **Process** to the right of the CSR, which displays the contents of the CSR, which displays the fields for processing the CSR. A few important points about these fields:
   - Because you are issuing a certificate for a CA client that can use security capabilities within InterSystems IRIS, under **Certificate Usage**, you can leave the default of **TLS/SSL, XML encryption and signature verification**.
   - In a production environment, you would need to verify the identity of the CA client. Under **Request Content**, the CA client's phone number and email would be displayed. This would allow you to contact them by phone or in person and verify their identity; their authority to hold a certificate containing the Subject Distinguished Name shown, signed by the CA for which you are responsible; and that the SHA-1 fingerprint is the one that was displayed when they created the CSR.
4. Click **Issue Certificate**, which causes the page to display the **Password for Certificate Authority's Private Key file** field.
5. In the **Password for Certificate Authority's Private Key file** field, enter `myflcapw`, which is the password you created when you configured the CA server.
6. Click **Finish** to create the certificate. IRIS displays a message such as

```
Certificate number 2 issued for Certificate Signing Request FLCAclient
```

InterSystems IRIS has now created the certificate. In a production system, it would also have notified the technical contact for the CA client by email that the certificate is available for download.

3.5 On Instance #2, Download Its Certificate and CA Server Certificate from the CA Server

The next and final step is for the CA client to download from the CA server both the CA server's certificate and its own certificate:

1. In the Management Portal, on instance #2, go to the **Public Key Infrastructure** page (**System Administration** > **Security** > **Public Key Infrastructure**).
2. On the **Public Key Infrastructure** page, under **Certificate Authority Client**, click **Get Certificate(s) from Certificate Authority server**.
3. In the fields that are displayed, there is a **Get Certificate Authority Certificate** button. Click it, which downloads the CA server certificate and displays a message such as:

```
Certificate Authority Certificate
(SHA-1 Fingerprint: 8A:38:C9:06:50:A0:4F:71:86:2B:69:4C:A2:42:E0:43:28:C8:70:EB)
saved in file "c:\intersystems\caclient\mgr\FLCA.cer"
```

4. Again, click **Get Certificate(s) from Certificate Authority server**.
5. The Issued Certificates table lists the CA client's certificate. Click the **Get** button next to its row. This downloads the CA client's certificate and displays a message such as:

```
Certificate number 2
(SHA-1 Fingerprint: 2E:82:27:73:72:38:BC:71:36:70:DC:9E:0D:EF:E6:BC:0D:A9:95:CD)
saved in file "c:\intersystems\caclient\mgr\FLCAclient.cer"
```

3.6 Wrapping Up and Next Steps

You have now:

1. Configured an instance of InterSystems IRIS as a CA server
2. Configured another instance of InterSystems IRIS as a CA client
3. Submitted a Certificate Signing Request (CSR) from the CA client to the CA server
4. Processed the CSR on the CA server
5. Downloaded the CA server's certificate and the CA client's own CA certificate from the CA server to the CA client

This means that you now have one InterSystems IRIS instance that is a functioning CA server and another InterSystems IRIS instance that is a functioning CA client. If you set up a CA client on another InterSystems IRIS instance and create TLS configurations for each instance, the two clients can exchange encrypted messages. This provides the basis for various secured activities.

Important: A final reminder: this example system does not help establish a secure environment because the CA's private key has been publicly published in this document. It is critical that you properly protect all private keys in production systems, and it is *most* important that you protect the private key of a CA. The exposure of a private key for use in a production system can result in security breaches, data exposure, financial losses, and legal vulnerability. Do not use this document's CA server private key for anything other than educating yourself about InterSystems IRIS features.

4 For More Information about the InterSystems PKI

For full documentation on the InterSystems PKI, see the "InterSystems Public Key Infrastructure" in the *InterSystems IRIS Security Administration Guide*.