PACS Test Card and Fault Path User Guide

VERSION 1.0.4

FIPS 201/FICAM TESTING PROGRAM

August 16, 2018

Office of Government-wide Policy
Office of Technology Strategy
Identity Management Division
Washington, DC 20405

Contents

1 Overview

The General Services Administration (GSA) is responsible for supporting the adoption of interoperable and standards-based Identity, Credential, and Access Management (ICAM) technologies throughout the Federal Government. As part of that responsibility, GSA operates and maintains the Federal Information Processing Standard (FIPS) Publication 201 Approved Products List (APL) , as well as services for Federal ICAM (FICAM) conformance and compliance.

The Federal Government's emphasis on strong authentication for physical access to federal agencies contributes to the growing need to support agency implementers. Accordingly, the FIPS 201/FICAM Testing Program has produced a set of test cards available for loan to agencies and vendors for the purpose of testing Physical Access Control Systems (PACS) in advance of submitting their products to the Program for testing. The cards may also be used by security professionals and integrators to ensure their system was installed correctly and is able to handle security threats and interoperability issues.

The FICAM testing program uses a variety of methods to determine whether a PACS is qualified for listing on the APL. The techniques and procedures for establishing if a PACS should be listed are detailed in GSA's PACS Functional Requirements and Test Cases (FRTC). The FRTC uses an assortment of procedures to confirm Certificate Path Validation, Card Authenticity, OCSP, Policies, and other nuances to ensure certified systems comply with FIPS 201-2.

2 ICAM Testing

2.1 ICAM Test Cards

The ICAM Test cards provide for a mixture of positive and negative test cases. As functional requirements evolve the test cases necessary for listing on the GSA APL may require the addition or deprecation of cards, and test cases. For this reason, it is important to ensure that the validation solution is built and tested to the most current versions of the FRTC and its associated ICAM Test Cards. GSA has a limited number of PACS Test Card Loaner Sets available on a first come-first serve time limited basis. Alternatively, vendors can build their own ICAM Test Cards using the GSA Card Builder tool.

2.1.1 Generation 1 and 2 ICAM Test Cards

Generation 1and 2 cards are ICAM Test Cards 1 – 24 and were developed for testing the PACS systems against FRTC 1.2 requirements. In this iteration of ICAM Test Cards, Card 1 and Card 2 were the "Golden" PIV and PIV-I cards. The Generation 1-2 golden cards have since been superseded by Generation 3 "Golden" cards but are still required for Time of Access (FRTC §5) testing purposes. Cards 3 – 24 are basically clones of either Card 1 or 2 except faults have been engineered into them as described in Table 1. For Time of Access testing, Cards 1 and 2 must be registered into the validation system to determine if the validation system can detect when a previously "Golden" card is tainted and deny access

2.1.2 Generation 3 ICAM Test Cards

Cards 25 – 55 were developed in response to the more stringent requirements of FRTC 1.3.3 and 1.3.3. Each of these cards is unique and contains its own FASC-N and name.

2.1.3 Tabular List of ICAM Test Cards

Table 1 shows the ICAM Test Cards and their configuration. They are used in conjunction with the ICAM PKI as prescribed in the FICAM Testing Program Functional Requirements and Test Cases.

Table 1 - ICAM Test Cards and Descriptions

ICAM Test Card	Valid/Invalid	Description	Threat Type
1	Valid	Golden PIV	None
2	Valid	Golden PIV-I	None
3	Invalid	Substituted keypair in PKI-AUTH certificate (AKID/SKID mismatch)	Manipulated Data
4	Invalid	Tampered CHUID	Manipulated Data
5	Invalid	Tampered PIV and Card Authentication Certificates	Manipulated Data

ICAM Test Card	Valid/Invalid	Description	Threat Type
6	Invalid	Tampered PHOTO	Manipulated Data
7	Invalid	Tampered FINGERPRINT	Manipulated Data
8	Invalid	Tampered SECURITY OBJECT	Manipulated Data
9	Invalid	Expired CHUID signer	Invalid Date
10	Invalid	Expired certificate signer	Invalid Date
11	Invalid	PIV Authentication Certificate expiring after CHUID	Invalid Date
12	Invalid	Authentication certificates valid in future	Invalid Date
13	Invalid	Expired authentication certificates	Invalid Date
14	Invalid	Expired CHUID	Invalid Date
15	Invalid	Valid CHUID copied from one card to another **(PIV)**	Copied Credential
16	Invalid	Valid Card Authentication Certificate copied from one card to another **(PIV)**	Copied Credential
17	Invalid	Valid PHOTO copied from one card to another **(PIV)**	Copied Credential
18	Invalid	Valid FINGERPRINT copied from one card to another **(PIV)**	Copied Credential
19	Invalid	Valid CHUID copied from one card to another **(PIV-I)**	Copied Credential
20	Invalid	Valid Card Authentication Certificate copied from one card to another **(PIV-I)**	Copied Credential
21	Invalid	Valid PHOTO copied from one card to another **(PIV-I)**	Copied Credential
22	Invalid	Valid FINGERPRINT copied from one card to another **(PIV-I)**	Copied Credential
23	Invalid	Private and Public Key mismatch	Manipulated Keys
24	Invalid	Revoked authentication certificates	Revoked Credential

ICAM Test Card	Valid/Invalid	Description	Threat Type
25	Valid	Discovery object is not present	Only Application PIN is present and shall be used.
26	Valid	Discovery object tag 0x5F2F is present First byte: 0x40, Second byte 0x00	Only Application PIN is present and shall be used.
27	Valid	Discovery object tag 0x5F2F is present First byte: 0x60, Second byte: 0x10	Application and Global PINs are present. Application PIN is primary.
28	Valid	Discovery object tag 0x5F2F is present First byte: 0x60, Second byte: 0x20	Application and Global PINs are present. Global PIN is primary.
29	Valid	Deprecated: Discovery object is present and tag 0x5F2F is not populated	Only Application PIN is present and shall be used.
30	Valid	Future: Card with PPS F=372, D=1 (13,440 baud)	ISO Standards Conformance
31	Valid	Future: Card with PPS F=372, D=2 (26,881 baud)	ISO Standards Conformance
32	Valid	Future: Card with PPS F=372, D=4 (53,763 baud)	ISO Standards Conformance
33	Valid	Future: Card with PPS F=372, D=12 (161,290 baud)	ISO Standards Conformance
34	Valid	Future: Card with PPS F=512, D=8 (78,125 baud)	ISO Standards Conformance
35	Valid	Future: Card with PPS F=512, D=16 (156,250 baud)	ISO Standards Conformance
36	Valid	Future: Card with PPS F=512, D=32 (312,500 baud)	ISO Standards Conformance
37	Valid	Card with PPS F=512, D=64 (625,000 baud), ECC Card Auth Cert, and Pairing Code for Secure Messaging	ISO Standards Conformance

ICAM Test Card	Valid/Invalid	Description	Threat Type
38	Invalid	Hash value within the Security Object does not match hash value of its corresponding data group buffer.	Manipulated Data
39	Valid	Federally issued PIV-I card using FASC-N with the agency's Agency Code plus System Code, Credential Number, Credential Series Code, and Issue Code.	Incorrect Identifier
40	Valid	Deprecated (replaced by Card 54): Federally-issued PIV-I. Valid: Federally issued PIV-I card using fourteen 9s.	Incorrect Identifier
41	Invalid	Public key on card does not match public key previously registered to the system.	Copied container
42	Invalid	Certificates on the card refer to an OCSP responder that uses an expired response signing certificate.	Invalid Date
43	Valid	Valid certificates on the card refer to an OCSP responder that uses a response signing certificate that is revoked but contains the *id-pkix-ocsp-nocheck* OID.	Invalid Credential
44	Invalid	Certificates on the card refer to an OCSP responder that uses a response signing certificate that is revoked, and the *id-pkix-ocsp-nocheck* OID is not present.	Invalid Credential
45	Invalid	Certificates on the card refer to an OCSP responder that uses a response signing certificate with an invalid signature.	Manipulated Data
46	Valid	Valid: FIPS 201-2 card with card UUIDs in the SubjectAltName extensions are sequentially after the FASC-Ns (replaces Card 1).	None
47	Valid	Golden FIPS 201-2 PIV card with card UUIDs in the SubjectAltName extensions are sequentially before the FASC-N.	SP 800-73-4 Standards Conformance
48	Valid	Future: Golden FIPS 201-2 PIV card with ISO 7816-compliant PPS LEN value greater than zero.	ISO Standards Conformance

ICAM Test Card	Valid/Invalid	Description	Threat Type
49	Invalid	FIPS 201-2 PIV card profile with exception that Cardholder Facial Image CBEFF has expired.	Invalid Date
50	Valid	Golden FIPS 201-2 PIV card profile with exception that Cardholder Facial Image CBEFF will expire before CHUID expiration date.	Invalid Date
51	Invalid	FIPS 201-2 PIV card profile with exception that Cardholder Fingerprints CBEFF has expired.	Invalid Date
52	Valid	Golden FIPS 201-2 PIV card profile with exception that Cardholder Fingerprints CBEFF will expire before CHUID expiration date.	Invalid Date
53	Valid	Golden FIPS 201-2 PIV card profile with slightly larger than recommended Card Authentication Certificate (2160 bytes).	SP 800-73-4 Standards Conformance
54	Valid	Golden FIPS 201-2 Non-Federally Issued PIV-I card (replaces Card 2).	None
55	Invalid	FIPS 201-2 PIV card missing its Security Object	Tampered Data

3 ICAM Fault Bridge

The ICAM Fault Bridge emulates a Bridged PKI Environment containing 4 operational groups of purposefully altered certificates including: Standards Conformant Processing, Crypto/Compatibility, Manipulated Certification Authority, and Revoked Certificate Authority.

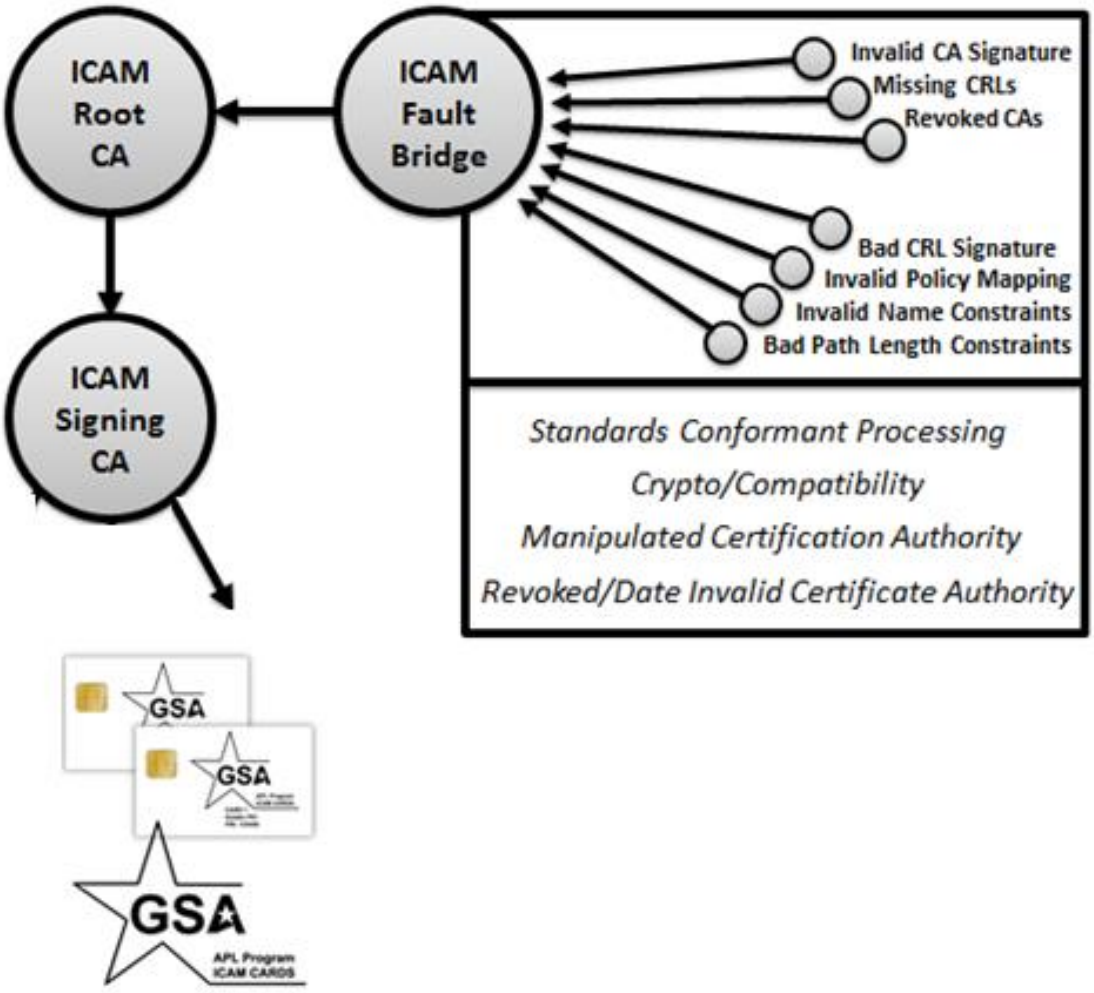
Each of the 4 operational groups presents a different risk to deployed systems that are as follows:

- *Standards Conformant Processing* – a system accepting medium assurance credentials must be able to correctly process the certificate path between an end entity certificate and a server or workstation's trusted root. The Standards Conformance area focuses on certificate extensions that are both security relevant to relying parties & widely deployed in the Federal PKI environment. High assurance systems unable to correctly process security relevant extensions leave themselves vulnerable to wrongly accepting invalid credentials.
  (e.g., Inhibit Policy Mappings, Path Length Constraints, or Name Constraints extensions)
- *Crypto/Compatibility* – NIST 800-78-31, Table 3-1 lays out the time periods for use of Cryptographic Algorithms and Key Sizes. Systems accepting medium assurance credentials must stay up to date with Federal and Commercial cryptographic standards. Systems unable to keep 'ahead of the curve' will be slowly obsoleted as technology progresses, *(e.g., ECDSA (Curve P-256), Large RSA Key sizes)*
- *Manipulated Certification Authority* – In the event that a Certification Authority is compromised the Manipulated Certificate Authority group simulates what known attacks would look like and allow system owners to identify ways to enhance their detection methods.
  (e.g., Invalid CA/CRL Signatures, Path Length Constraint, Invalid Subject Domain)
- *Revoked or Date Invalid Certificate Authority* – an attacker or regular user may try to gain access using a credential that has been revoked by the issuer or that has expired. Revocation also occurs in a Bridged environment and not just at the end entity level and even CAs have a set lifetime their signing keys are valid for. Systems not checking for revocation information or expiration of Certificate Authorities or Cross-Certificates could be vulnerable to larger scale invalid access use cases if an entire organization becomes compromised.
  (e.g., Revoked Certification Authority, Invalid CA notAfter Date)

Each of the 30 cross-certified CAs falls into a single operational group and all have very specific security related concerns. It should be noted that the Fault Bridge only provides test cases for security threats related to the Federal and Commercial cross-certified space. The Fault Bridge is designed to enhance testing of GSA ICAM Test Cards with known positive and negative certificate paths.

Figure 1 Fault Bridge Conceptual Architecture

1 NIST SP800-78-4 Cryptographic Algorithms and Key Sizes for Personal Identity Verification
https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-78-4.pdf



3.1.1 Fault Bridge Testing Procedure

The testing methodology to use with the Fault Bridge is that only a single Trusted Root certificate is installed at the time of each test. Any number of intermediate certificates may be installed during testing, but only a single Fault Bridge root certificate should be trusted.

Following that targeted methodology, the Fault Bridge is able to interface with any product that uses Public Key Certificates; however, individual products might each have different configuration settings for their use. Most commercial validation engines will directly make calls to the Microsoft Crypto API (CAPI) trust stores for either certificate path building, certificate validation, or both.

Although the Fault Bridge is compatible with any system that accepts digital certificates, instructions for testing are provided for only Microsoft CAPI at this time. Please contact Fault Bridge support for additional information or help regarding specific systems.

3.1.1.1 Disable Automation Root Store Updates

Disable automatic root store updates. For information on how to do this, follow the steps in APPENDIX A: Turn off Automatic Root Certificate Updates.

3.1.1.2 Install the Fault Bridge Intermediate Certificates

Install all intermediate cross certificates associated with the Fault Bridge. These certificates will be used for path building but are not trust anchors themselves. For information on how to install the intermediate certificates, follow the steps in APPENDIX C: Intermediate Certificate Installation.

3.1.1.3 Install the Root Certificate for the Fault Under Test

Install the trusted root certificate for the fault path under test. Refer to APPENDIX D: Trusted Root Certificate Installation. There should not be any other ICAM Test card root certificates installed - APPENDIX B: Deleting a Root or clearing the Trusted Root Certification Authorities store. The validation engine will have a single path, allowing the specific fault to be targeted in the certificate path.

3.1.1.4 Clear Cached URLS

Windows will cache URLs for OCSP and CRL servers. To ensure accurate test results, those URLs should be cleared when fault paths are changed. This is done from the Windows Command prompt with administrator privilege. The command strings are:

```
certutil -setreg chain\ChainCacheResyncFiletime @now
certutil -urlcache * delete
```

3.1.2 Fault Bride Tabular Description

ICAM PKI Path Number	Fault description	Operational Group
1	ICAM Invalid CA Signature	Manipulated Data
2	ICAM Invalid CA *notBefore* Date	Revoked/Date Invalid
3	ICAM Invalid CA *notAfter* Date	Revoked/Date Invalid
4	ICAM Invalid Name Chaining	Standards Conformant Processing
5	ICAM Missing Basic Constraints	Standards Conformant Processing
6	ICAM Invalid CA False Critical	Manipulated Data
7	ICAM Invalid CA False not Critical	Standards Conformant Processing
8	ICAM Invalid Path Length Constraint	Standards Conformant Processing
9	ICAM keyUsage keyCertSign False	Standards Conformant Processing
10	ICAM keyUsage Not Critical	Standards Conformant Processing
11	ICAM *keyUsage* Critical *CRLSign* False	Standards Conformant Processing
12	ICAM Invalid inhibitPolicyMapping	Standards Conformant Processing
13	ICAM Invalid DN *nameConstraints*	Standards Conformant Processing
14	ICAM Invalid SAN *nameConstraints*	Standards Conformant Processing
15	ICAM Invalid Missing CRL	Standards Conformant Processing
16	ICAM Invalid Revoked CA	Revoked/Date Invalid
17	ICAM Invalid CRL Signature	Manipulated Data
18	ICAM Invalid CRL Issuer Name	Standards Conformant Processing
19	ICAM Expired CRL *nextUpdate*	Revoked/Date Invalid
20	ICAM Invalid CRL *notBefore*	Revoked/Date Invalid
21	ICAM Invalid CRL Distribution Point	Standards Conformant Processing
22	ICAM Valid requiredExplicitPolicy	Standards Conformant Processing
23	ICAM Invalid requiredExplicitPolicy	Standards Conformant Processing

ICAM PKI Path Number	Fault description	Operational Group
24	ICAM Valid GeneralizedTime	PKI/Crypto Compatibility
25	ICAM Invalid GeneralizedTime	Standards Conformant Processing
32	ICAM Invalid SKID	Standards Conformant Processing
33	ICAM Invalid AKID	Standards Conformant Processing
34	ICAM Invalid CRL format	Standards Conformant Processing
35	ICAM 4096bit RSA key	PKI/Crypto Compatibility
36	ICAM Invalid CRL Signer	Standards Conformant Processing
37	ICAM Valid PIV-I	Standards Conformant Processing

Table 2 Fault Path Tabular Description

3.1.3 Fault Bridge Detailed Descriptions

3.1.3.1 ICAM Invalid CA Signature

Fault Description: Root 1, *Invalid CA Signature*, has a path that contains a cross-certificate that has an invalid CA signature.

Threat Overview: When building a path to a trusted root, applications must verify the signature of each certificate in the path. If a CA has an invalid signature then the public key of the invalid certificate cannot be properly linked to the issuing CA. Not performing signature verification presents attackers with the opportunity to substitute any bad certificate to your software, regardless of issuer, and have it validated correctly.

3.1.3.2 ICAM Invalid notBefore Date

Fault Description: Root 2, *Invalid CA notBefore Date*, is root certificate that has a cross-certificate that will be not valid until sometime in the future: Sunday, December 31, 2034 8:00:00 PM

Threat Overview: Software must be able to recognize a certificate's life-span. The notBefore date is used by organizations to plan future PKI segments that will turn on at a certain date, usually to replace an expiring CA certificate. Trusting a CA certificate before it is valid presents a risk to your software if the certificate contains updated security settings that do not yet apply to your application – such as a change in the certificates assurance levels, or name constraints that have been added or removed.

3.1.3.3 ICAM Invalid notAfter Date

Fault Description: Root 3, *Invalid CA notAfter Date*, has a path that contains a cross-certificate that has expired.

Threat Overview: Trusting CA and cross-certificates after their validity date has ended presents the risk of outdated information being accepted by your application. An expired CA certificate may represent a business relationship that has not been renewed or security settings that are no longer acceptable.

3.1.3.4 ICAM Invalid Name Chaining

Fault Description: Root 4, *Invalid Name Chaining*, has a path that contains an invalid issuer/subject name combination with the cross-certificate.

Threat Overview: RFC 5280 path building requirements state certificate name chaining must be done. Ensuring that a path contains the proper issuer/subject names through mitigates the risk that a key is being misused by someone not authorized to have access to that key. Name chaining also removes ambiguity of which key belongs to which organization or individual.

3.1.3.5 ICAM Missing Basic Constraints

Fault Description: Root 5, *Missing Basic Constraints*, contains a cross-certificate without the Basic Constraints extension.

Threat Overview: A Certificate without the Basic Constraints extension presents two risks to your application. The first is that the path length is not being constrained at that level and any number of subordinate certificates may be issued under the certificate. The second, and larger risk, is that your application cannot differentiate whether the certificate is an End Entity or a CA certificate. Other than relying on keyUsage to determine what the certificate may be used for, your application will not know whether the issuer intended that certificate's key to sign other certificates or CRLs.

3.1.3.6 ICAM Invalid CA False Critical

Fault Description: Root 6, *Invalid CA False Critical*, contains a certificate path that flags unintended certificate extension criticality issues.

Threat Overview: The purpose of this certificate path is to test an applications ability to be selective in the extensions that are processed. A critical extension in a certificate must be processed according to RFC 5280 – but there is no way to force an application to process a Subject Key Identifier. Use this path to test either "Graceful Failure" or your applications ability to perform selective extension processing.

3.1.3.7 ICAM Invalid CA False not Critical

Fault Description: Root 7, *Invalid CA False not Critical*, contains no critical extensions.

Threat Overview: The purpose of this certificate path is to test an applications ability to be selective in the extensions that are processed regardless of criticality. A non-critical extension in a certificate is not required to be processed according to RFC 5280. Use this path to test either "Graceful Failure" or your applications ability to perform selective extension processing.

3.1.3.8 ICAM Invalid pathLengthConstraint

Fault Description: Root 8, *Invalid Path Length Constraint*, contains a cross-certificate in its path that has a basic constraint length that is too short, making the certificate path invalid for all cross-certified test infrastructures.

Threat Overview: The Path Length component of Basic Constraints is to allow issuers the ability to dictate how far a certificate path can travel from your trusted root before it must reach an end entity. This protects your application from accepting potential rogue certificates from issuers that are "too many hops" away from your root. Often times entities that are too far away from your root will not have known issuance and security processes, making the trustworthiness of the PKI questionable.

3.1.3.9 ICAM keyUsage keyCertSign False

Fault Description: Root 9, *keyUsage keyCertSign not set*, does not have the Certificate Signing Key Usage set in the root certificate. This makes the root certificate invalid for signing itself – or any other certificates.

Threat Overview: Not checking CA certificates for the Certificate Signing Key Usage means any certificate could potentially sign a new certificate and have that be accepted by your application. This presents a risk because any user could extend a certificate to themselves or a potential attacker that would pass validation.

3.1.3.10 ICAM keyUsage Not Critical

Fault Description: Root 10, *keyUsage Not Critical*, has a cross-certificate in its path where Basic Constraints are critical and Key Usage is not.

Threat Overview: The purpose of this certificate path is to test an applications ability to be selective in the extensions that are processed regardless of criticality. A non-critical extension in a certificate is not required to be processed according to RFC 5280. This Certificate path differs from the previous non-critical path because the Key Usage extension is non-critical and the Basic Constraints are critical – in a typical cross-certificate both should be critical. Use this path to test either "Graceful Failure" or your applications ability to perform selective extension processing.

3.1.3.11 ICAM CRLSign False

Fault Description: Root 11, *keyUsage Critical CRLSign not set*, is a root certificate that does not have the CRL Signing key usage, making it unable to issue CRLs.

Threat Overview: Accepting certificate paths from CAs that are not authorized to issue CRLs, but are doing so anyway, present a risk to your application. If any issue is found with revocation information or capability in a certificate path, high assurance security sensitive applications should ignore those paths or risk exposure to being unable to prevent blacklisted certificates from access.

3.1.3.12 ICAM Invalid inhibitPolicyMapping

Fault Description: Root 12, *Invalid inhibitPolicyMapping*, contains a cross-certificate that contains the Inhibit Policy Mapping constraint causing no further policy mappings to occur, causing the path to become invalid for any policy identifier.

Threat Overview: Systems relying on specific levels of assurance such as medium software, medium hardware, PIV, or PIV-I must be able to process policy mappings all certificates in any given path. In addition to processing mappings policy constraints such as the inhibit policy mapping indicator protect roots from additional mappings to unknown entities.

3.1.3.13 ICAM Invalid DN nameConstraints

Fault Description: Root 13, *Invalid DN nameConstraints*, provides a cross-certificate that has permitted subtrees to the subject DN of an end entity certificate to test if the application will allow other name forms through, unconstrained.

Threat Overview: Windows XP and Server 2003 based machines have a known bug with disallowing name forms when they fall outside of the permitted list of names but are not explicitly excluded. Use this certificate path to ensure that your application is correctly allowing name constraints that are neither permitted nor excluded.

3.1.3.14 ICAM Invalid SAN DN Name

Fault Description: Root 14, *Invalid SAN DN name*, provides a cross-certificate that specifies a single permitted subtree. The end entity certificate includes a subjectAltName with a DN that falls outside that subtree to test if the application will allow that name through, unconstrained.

Threat Overview: Windows XP and Server 2003 based machines have a known bug with disallowing name forms when they fall outside of the permitted list of names but are not explicitly excluded. Use this certificate path to ensure that your application is correctly allowing name constraints that are neither permitted nor excluded.

3.1.3.15 ICAM Invalid Missing CRL

Fault Description: Root 15, *Invalid Missing CRL*, contains a cross-certificate that has a CRL distribution point that points to nothing.

Threat Overview: The risk an application faces with CRL related issues – particular issues where revocation information cannot be found is that an attacker or terminated employee is using a credential that should have been blacklisted to gain access to application or network resources that should no longer be available to them.

3.1.3.16 ICAM Invalid Revoked CA

Fault Description: Root 16, *Invalid Revoked CA*, is a certificate path back to a root that contains a revoked Certificate Authority

Threat Overview: End entities are revoked with much greater frequency than Certificate Authorities. Some applications check revocation information for end entities but neglect to check the validity of Signing or Intermediate CAs in every certificate path. The risk here is that when a Certificate Authority is revoked every credential issued by that CA is now invalid and should be denied access to your application, regardless of the reason code for the CA's revocation.

3.1.3.17 ICAM Invalid CRL Signature

Fault Description: Root 17, *Invalid CRL Signature*, has a certificate present in the cert path that has a CRL that has an invalid signature.

Threat Overview: Applications that do not check the signature of CRLs they retrieve for revocation information are at risk if the revocation list becomes the attack scenario. Without performing signature verification on revocation lists they are susceptible to having their data contents altered after they are issued and reposted for use by an attack who has potentially acquired a revoked credential they would like to make valid again.

3.1.3.18 ICAM Invalid CRL Issuer DN

Fault Description: Root 18, *Invalid CRL Issuer Name*, is a certificate path where the CRL for the cross-certificate has a name other than the issuing CA but is signed with the proper CA key.

Threat Overview: When building paths and checking revocation information applications must use name chaining to remove ambiguity in the ownership of CA and end entity keys. A CRL with an invalid issuer name could also be an indicator that another entity has illegitimately gained control of the CA server and keys to issuer their own end certificates or CRLs.

3.1.3.19 ICAM Expired CRL

Fault Description: Root 19, *Expired CRL* is a certificate path where the CRL for the cross-certificate has an expired *nextUpdate* value (an expired CRL).

Threat Overview: A system that relies on an expired CRL can be fooled into validating a revoked certificate, granting unauthorized access to controlled areas.

3.1.3.20 ICAM Invalid CRL notBefore

Fault Description: Root 20, *Invalid CRL notBefore* is a certificate path where the CRL for the cross-certificate is not yet valid.

Threat Overview: Software must be able to recognize a CRL's life-span. The notBefore date is used by organizations to plan future PKI segments that will turn on at a certain date, usually to replace an expiring CA certificate. Trusting a CA's CRL before it is valid presents a risk to your software if the certificate contains updated security settings that do not yet apply to your application – such as a change in the certificates assurance levels, or name constraints that have been added or removed.

3.1.3.21 ICAM Invalid CRL Distribution Point

Fault Description: Root 21, *Invalid distributionPoint*, is a root cert with a path that contains a CRL distribution point that is present, but invalid and unreadable.

Threat Overview: The risk to applications is that with revocation checking being a key component of certificate validation, an invalid distribution point could crash or cause unintended behavior, the worst being allowing access to potentially bad credentials.

3.1.3.22 ICAM Valid requiredExplicitPolicy

Fault Description: Root 22, *Valid requiredExplicitPolicy*, is a certificate path where the indicator for explicit policy is set. Your application or system must require a certificate policy in order for this path to be valid.

Threat Overview: There is an interoperability threat to applications and systems that cannot process certificate paths that require explicit policies. Currently many enterprise PKI’s do not utilize this extension in their certificates, but the market is slowly shifting to one where knowing and requiring a specific assurance level for internet transactions and access is desired across the board.

3.1.3.23 ICAM Invalid requiredExplicitPolicy

Fault Description: Root 23, *Invalid requiredExplicitPolicy*, is a certificate path where the indicator for explicit policy is set, however it is set incorrectly.

Threat Overview: There is an interoperability threat to applications and systems that cannot process certificate paths that require explicit policies. Currently many enterprise PKI's do not utilize this extension in their certificates, but the market is slowly shifting to one where knowing and requiring a specific assurance level for internet transactions and access is desired across the board. This test root will give you a negative test case to pair with Root 21 for testing this certificate extension.

3.1.3.24 ICAM Valid GeneralizedTime

Fault Description: Root 24, *Valid GeneralizedTime*, is a root certificate with a valid path that is very long lived and uses the proper date/time encoding for certificates after the year 2049.

Threat Overview: At the end of the year 2049 the way date and time is done for certificates will change to Generalized Time. Very long-lived applications or systems such as network devices or databases that will stay online will need to be able to consume Generalized Time.

3.1.3.25 ICAM Invalid GeneralizedTime

Fault Description: Root 25, *Invalid GeneralizedTime*, is a root certificate with a valid path that is uses Generalized Time before the year 2049, which is an incorrect encoding.

Threat Overview: Very long-lived applications or systems such as network devices or databases that will stay online will need to be able to consume Generalized Time, however, certificates should not be encoded using Generalized Time until after 2049. This root certificate allows for a way to test whether your application or system can identify certificates that have dates that are encoded incorrectly.

3.1.3.26 Reserved for Future Use

3.1.3.27 Reserved for Future Use

3.1.3.28 Reserved for Future Use

3.1.3.29 Reserved for Future Use

3.1.3.30 Reserved for Future Use

3.1.3.31 Reserved for Future Use

3.1.3.32 ICAM Invalid SKID

Fault Description: Root 32, *Invalid SKID*, is a path that contains a cross-certificate with an invalid Subject Key ID (SKID).

Threat Overview: An invalid subject key identifier in a certificate is an indication that issuance was done improperly or something is wrong with the public key of the certificate. Systems should be able to invalidate a certificate path by examining a certificates public key against the SKID.

3.1.3.33 ICAM Invalid AKID

Fault Description: Root 33, *Invalid AKID*, is a path that contains a cross-certificate with an invalid Authority Key ID (AKID).

Threat Overview: An invalid authority key identifier in a certificate is an indication that issuance was done improperly or something is wrong with the public key of the certificate issuer. Systems should be able to invalidate a certificate path by examining the certificate issuer's public key against the AKID

3.1.3.34 ICAM Invalid CRL format

Fault Description: Root 34, *Invalid CRL Format*, contains a CRL located in the path that is not formatted correctly. The CRL contains incorrect extensions and data and could potentially be unreadable by the system or application under test.

Threat Overview: The risk to applications is that with revocation checking being a key component of certificate validation, an invalid CRL format could crash or cause unintended behavior, the worst being allowing access to potentially bad credentials.

3.1.3.35 ICAM 4096bit RSA key

Fault Description: Root 35, *4096bit RSA key*, is a valid SHA1 root with a key size of 4096 to use for performance benchmarking a system. A 4096-bit key size can substantially increase signature validation time.

Threat Overview: NIST 800-78-3, Cryptographic Algorithms and Key Sizes for Personal Identity Verification, dictates key sizes to be used within the Federal PKI. Currently 2048-bit keys are required and used. The increase in size can cause applications to slow down, public key operations naturally take longer with larger keys. Testing with Root 1 will allow you to benchmark your systems processing speed verifying signatures that used larger keys.

3.1.3.36 ICAM Invalid CRL Signer

Fault Description: Root 36, *Invalid CRL Signer*, is a certificate path that includes a CRL with an invalid signature.

Threat Overview: When building a path to a trusted root, applications must verify the signature of each certificate in the path. If a CRL has an invalid signature then the public key of the invalid CRL cannot be properly linked to CRL signer. Not performing signature verification presents attackers with the opportunity to substitute any bogus CRL to your software, regardless of issuer, and have it validated correctly.

3.2 Certificate Policy OIDS

3.2.1 ICAM PIV Test Card Policy OIDS

The certificate policies OIDS for FIPS 201-2 PIV cards that this project uses are below. They use the NIST test OIDs that are designated to mimic production OIDs. Validation systems should configure their initial policy sets as follows:

Certificate Name	EE Certificate Policy OID
PIV Authentication	2.16.840.1.101.3.2.1.48.11
Card Authentication	2.16.840.1.101.3.2.1.48.13
Card Authentication EKU KPID	2.16.840.1.101.3.6.8
Digital Signature	2.16.840.1.101.3.2.1.48.9
Key Management	2.16.840.1.101.3.2.1.48.9
Content Signing	2.16.840.1.101.3.2.1.48.86
Content Signing EKU KPID	2.16.840.1.101.3.6.7

Table 3 Test PIV OIDs

3.2.2 ICAM PIV-I Test Card Policy OIDS

The certificate policies for PIV-I cards that this project uses are below. With PIV-I cards, the certificate policies on the certificates must correctly map to an initial policy on the validation system. The PIV-I Signing CA cert contains the Subject Domain policies below, which map to the Issuer Domain policies.

Relying parties (validation systems) should be configured for the Issuer Domain initial policy set.

Certificate Name	Subject Domain Policy OID	Issuer Domain Policy OID
Authentication	2.16.840.1.101.3.2.1.48.248	2.16.840.1.101.3.2.1.48.78
Card Authentication	2.16.840.1.101.3.2.1.48.249	2.16.840.1.101.3.2.1.48.79
Key Management	2.16.840.1.101.3.2.1.48.250	2.16.840.1.101.3.2.1.48.3
Digital Signature	2.16.840.1.101.3.2.1.48.251	2.16.840.1.101.3.2.1.48.4
Content Signing	2.16.840.1.101.3.2.1.48.252	2.16.840.1.101.3.2.1.48.80

Table 4 Test PIV-I OIDs

3.2.3 Federal PIV Card Certificate Policy OIDs

The certificate policy OIDs used by the Federal Government are found on Federally-issued PIV cards.

Certificate Name	EE Certificate Policy OID
PIV Authentication	2.16.840.1.101.3.2.1.3.13
Card Authentication	2.16.840.1.101.3.2.1.3.17
Card Authentication EKU KPID	2.16.840.1.101.3.6.8
Digital Signature	2.16.840.1.101.3.2.1.3.7*
Key Management	2.16.840.1.101.3.2.1.3.6*
Content Signing	2.16.840.1.101.3.2.1.3.39
Content Signing EKU KPID	2.16.840.1.101.3.6.7

* These can actually be one or more of:

2.16.840.1.101.3.2.1.3.6 (id-fpki-common-policy)

2.16.840.1.101.3.2.1.3.7 (id-fpki-common-hardware)

2.16.840.1.101.3.2.1.3.16 (id-fpki-common-High)

Table 5 Federal PIV Certificate Policy OIDs

4 PIV-I Card Certificate Policy OIDs

Certificate policies for PIV-I cards below. With PIV-I cards, the certificate policies on the certificates must correctly map to an initial policy on the relying party validation system. The PIV-I Signing CA cert and bridge certificates contain mappings from Subject Domain policies which map to the Issuer Domain policies below.

More information can be found at NIST's Computer Security Register

Relying parties (validation systems) should be configured for the Issuer Domain initial policy set.

Certificate Name	EE Certificate Policy OID
Authentication	2.16.840.1.101.3.2.1.3.18
Card Authentication	2.16.840.1.101.3.2.1.3.19
Card Authentication EKU KPID	2.16.840.1.101.3.6.8
Digital Signature	2.16.840.1.101.3.2.1.3.18
Key Management	2.16.840.1.101.3.2.1.3.18*
Content Signing	2.16.840.1.101.3.2.1.3.20*
Content Signing EKU KPID	2.16.840.1.101.3.8.7

Table 6 PIV-I Card Certificate Policy OIDs

Appendix A: Turn Off Automatic Root Certificate Updates

To perform this procedure, you must be a member of the local **Administrators** group, or you must have been delegated the appropriate authority.

To turn off Automatic Root Certificates Update:

1. Click *Start*, and then click *Run*.
2. Type **gpedit.msc**, and then click *OK*.
3. If the *User Account Control* dialog box appears, confirm that the action it displays is what you want, and then click *Continue*.
4. Under *Computer Configuration*, double-click *Administrative Templates*, double-click *System*, double-click *Internet Communication Management*, and then click *Internet Communication* settings.
5. Double-click Turn off *Automatic Root Certificates Update*, click *Enabled*, and then click *OK*.
6. Close the Local Group Policy Editor.

Note: You can use Group Policy to set policy settings that apply across a given site, domain, or organizational unit in Active Directory Domain Services.

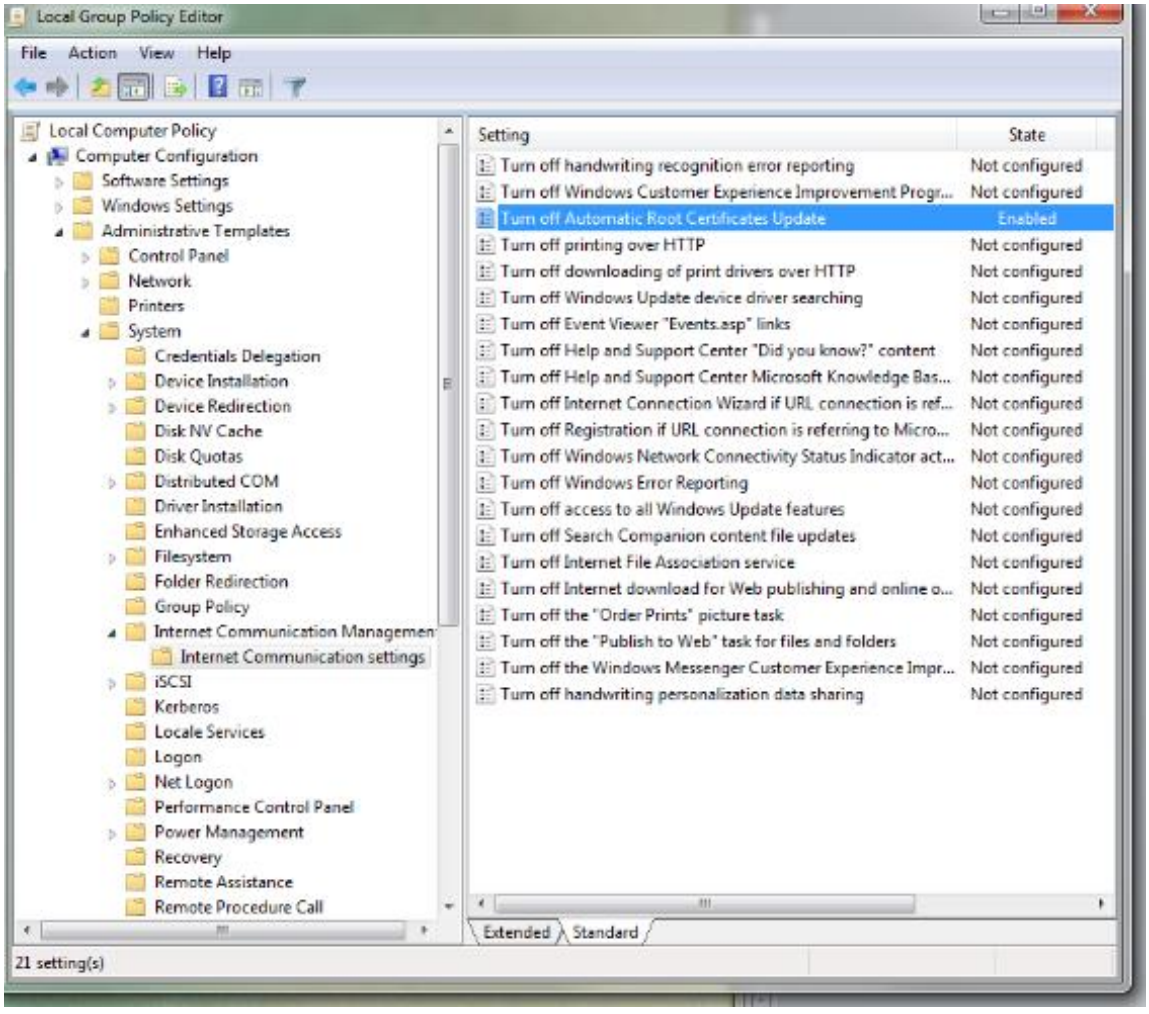


Figure 2 Group Policy Editor

Appendix B: Deleting a Root or clearing the Trusted Root Certification Authorities store

To delete certificates from the Trusted Root Certification Authorities, store for a user account:

1. Click *Start*, click *Start Search*, type **mmc**, and then press *Enter*.
2. On the *File* menu, click *Add/Remove Snap-in*.
3. Under *Available Snap-ins*, click *Certificates*, and then click *Add*.
4. Under *This snap-in will always manage certificates for*, click *Local Computer*, and then click *Next*.
5. If you have no more snap-ins to add to the console, click *OK*.
6. In the console tree, double-click *Certificates*.
7. Browse to the *Trusted Root Certification Authorities* store.
8. Select all available certificates in the Trusted Root Certification Authorities store, right click and select *Delete2*.

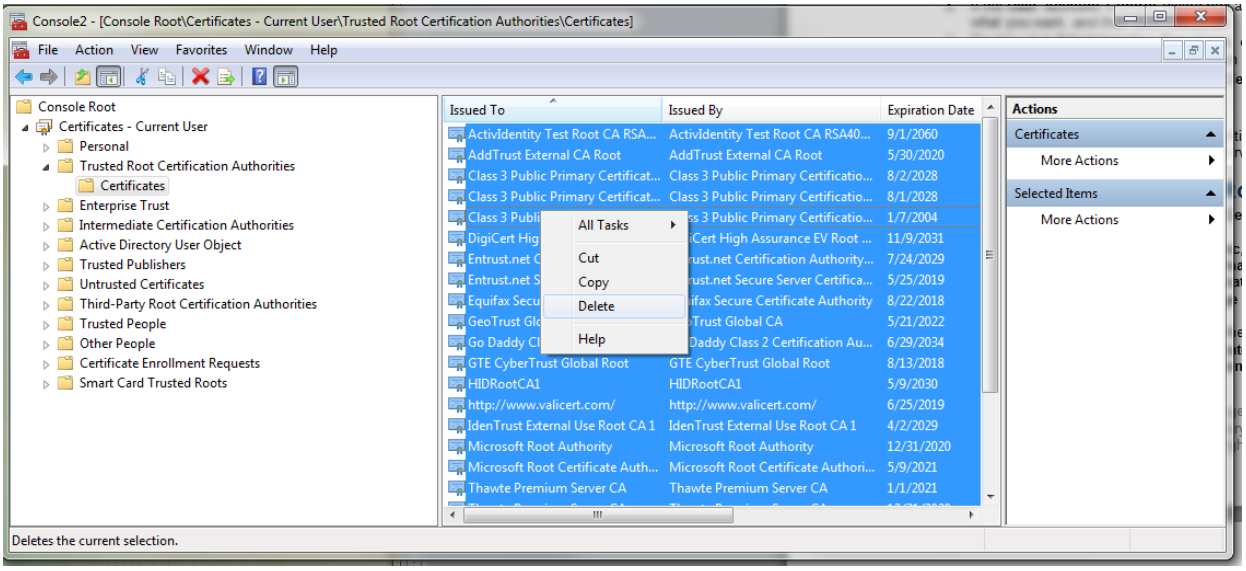


Figure 3 Trusted Root Certificate Store

Appendix C: Intermediate Certificate Installation

If Root Certificates are installed correctly then path building should occur automatically. These steps are provided in case additional trouble shooting is required or automatic path discovery is turned off on the system under test.

STEP 1: Download the intermediate .p7b bundle. The file location on the web is:

http://http.apl-test.cite.fpki-lab.gov/bridge/Intermediate_Bridge_Certs.p7b

2 There may be certain Trusted Root settings that are necessary for the application you are testing to run – leave those settings intact while ensuring only a single ICAM Fault Bridge root is present in the **Trusted Root Certification Authorities** store.

The .p7b file is a collection of all the intermediate certificates needed to operate the ICAM Fault Bridge.

NOTE File dialog boxes may appear different on each operating system, but the overall steps and file names are the same.

STEP 2: Right click the Intermediate Certificate's .p7b file and select “Install Certificate”

When you select install certificate a dialog will appear showing you the Certificate Import Wizard, following the steps in the wizard, automatically import the Intermediate's .p7b file. This will install every intermediate certificate to and from the PKI Fault Bridge.

STEP 3: Follow the steps for automatic import using the Certificate Import Wizard.

Select *Next* to continue with the import, and *Next* to have the wizard automatically place the certificates in the correct store, then Finish.

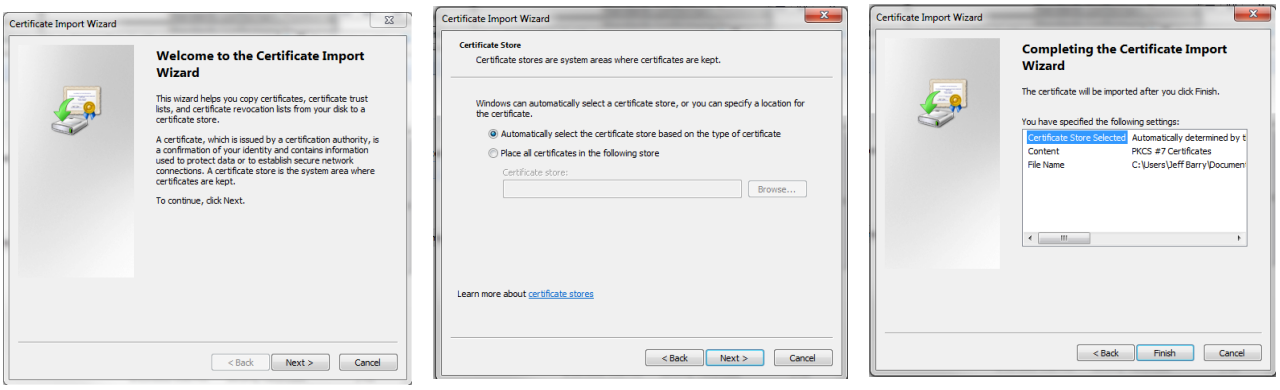


Figure 4 Certificate Import Wizard

STEP 4: You are finished installing the ICAM Fault Bridge Intermediate Certificates

APPENDIX D: Trusted Root Certificate Installation

Microsoft Windows Operating Systems

Option A

STEP 1: Download a Root Certificate you intend to test with from the links provided in the above descriptions (testing should be repeated for each Root Certificate). You may also download the complete root certificate bundle and import them into an untrusted certificate store and move them to and from the untrusted to the trusted root certification store as needed: The fault path trust anchors are at:

http://http.apl-test.cite.fpki-lab.gov/roots/Trust_Anchors.p7b

NOTE: File dialog boxes may appear different on each operating system, but the overall steps and file names are the same.

STEP 2: Double click to open the Root Certificate

When you double click the Root certificate a dialog will appear showing you the General Information for the certificate you just opened. Verify that the *Issued to:* and *Issued by:* fields both should contain a descriptive Test Case name such as: *Invalid CA Signature*.

STEP 3: Select "Install Certificate" at the bottom center of the certificate dialog box

This will launch the Certificate Import Wizard, the tool we will use to install the Root Certificate to your local computer.

Select Next to continue with the import.

Figure 5 Certificate Import Wizard Step 3

STEP 4: Choose *Place all certificate in the following store* from the radio button selection

We will be placing the Root Certificate in a specific store in order to establish trust with the GSA ICAM Test cards. After selecting *Place all certificates in the following store* click the *Browse* button.

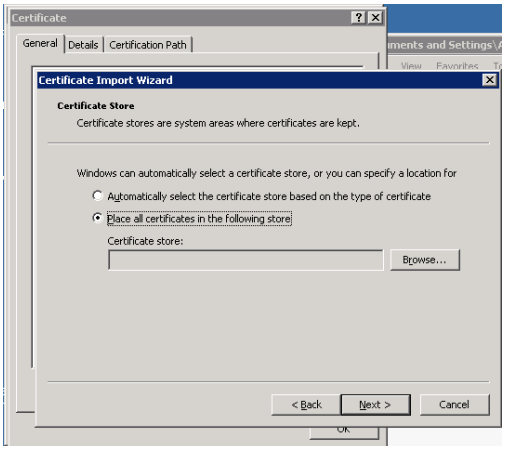


Figure 6 Certificate Import Wizard Step 4

STEP 5: Select Trusted Root Certification Authorities from the list of Certificate Stores

Upon selecting the Trusted Root Certification Authorities store click *Next* to continue.

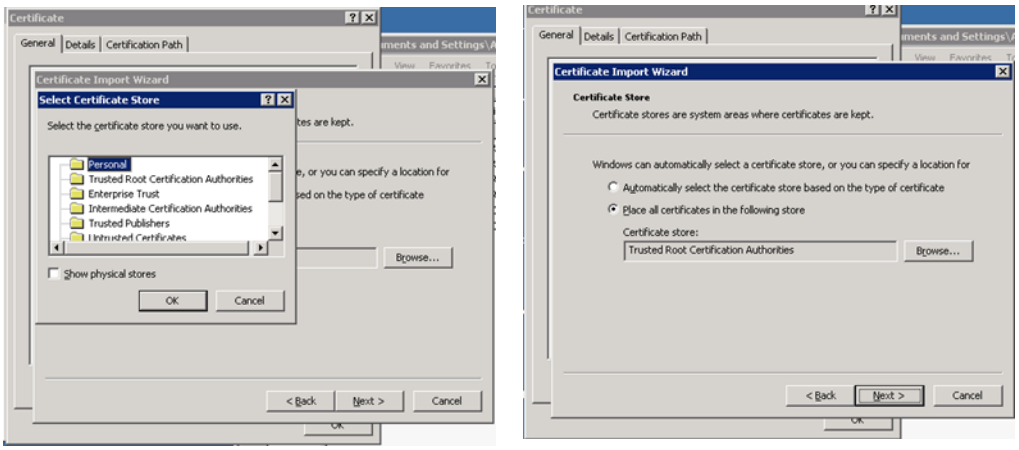


Figure 7 Certificate Import Wizard Step 5

STEP 6: Select *Finish*

You have now successfully imported a Fault Root Certificate for testing with the GSA ICAM Test Cards, you may see a Security Warning appear – if this occurs simply select yes.

Figure 8 Certificate Import Wizard Step 6

Option B

Alternatively, certificates can be imported directly from the Microsoft Management Console certificate store.

1. Click *Start*, click *Start Search*, type **mmc**, and then press ENTER.
2. On the *File* menu, click *Add/Remove Snap-in*.
3. Under *Available snap-ins*, click *Certificates*, and then click *Add*.
4. Under *This snap-in will always manage certificates for*, click *Local Computer*, and then click *Next*.
5. If you have no more snap-ins to add to the console, click *OK*.
6. In the console tree, double-click *Certificates*.

7. Double Click *Trusted Root Certification Authorities* store.
8. Right Click on *Certificates*
9. Click *All Tasks*
10. Click on *Import*
11. Follow operations performed in Option A