Pre-Delegation Testing

Data Escrow Test Cases

Version E

File name: PDT_DataEscrow_TC.docx
Last saved: 2013-07-01

Document control

Document information and security

Made by	Responsible for fact	Responsible for document
Rickard Bellgrim	Rickard Bellgrim	Rickard Bellgrim

Security class	File name
External	PDT_DataEscrow_TC.docx

Revisions

Date	Version	Name	Description
2013-01-17	PA1	Rickard Bellgrim	Initial document
2013-01-20	PA2	Rickard Bellgrim	Update after external review
2013-01-22	PA3	Rickard Bellgrim	Update after external review
2013-01-24	PA4	Rickard Bellgrim	Update text
2013-02-06	PA5	Rickard Bellgrim	Add Document Hierarchy and final chapter
2013-02-07	PA6	Rickard Bellgrim	Updated requirements
2013-03-24	PB1	Rickard Bellgrim	Moved test case DataEscrowProfile Allows ryde-csv.tar as file extension Rewrite the test case DataEscrowProfile
2013-04-08	B	Staffan Hagnell	Delivery D2 for production
2013-04-17	PC1	Rickard Bellgrim	File names are case insensitive. Removed .ryde-csv.tar as an extension. All deposits are now a tar archive.
2013-04-19	C	Mats Dufberg	Released
2013-04-29	PD1	Rickard Bellgrim	Changes from ICANN: • Updated external references • Check file names on tarball and XML
2013-05-03	D	Mats Dufberg	Released
2013-06-12	PE1	Rickard Bellgrim	Clarified the test description for each test case.
2013-06-24	PE2	Rickard Bellgrim	Clarified pass/fail criteria. Removed test case DataEscrowProfile. Now part of DataEscrowContent.
2013-07-01	E	Mats Dufberg	Released.

LIST OF CONTENTS

1. Introduction

1.1 Scope

All of the test cases for the data escrow can be found in this document.

1.2 References

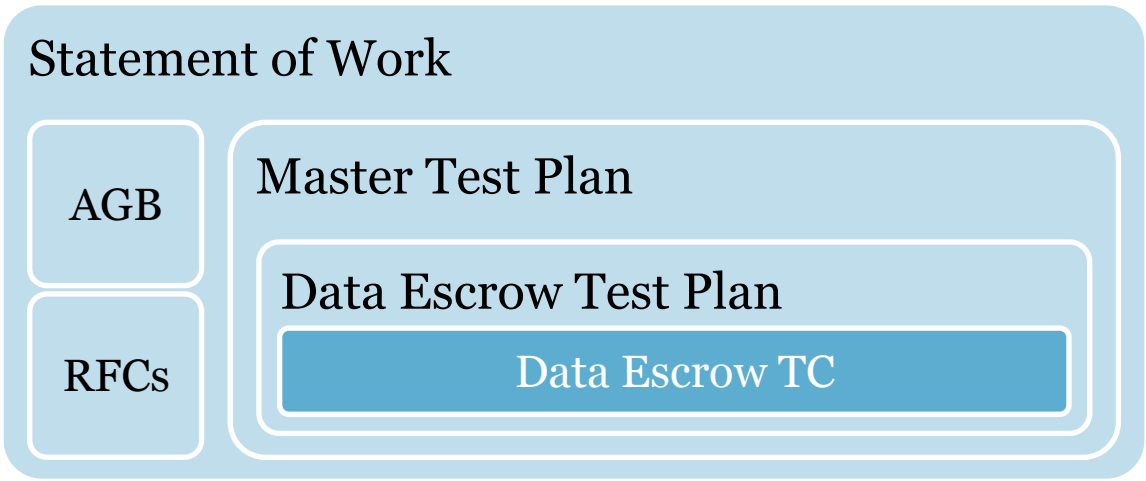
1.2.1 External

- IEEE 829-2008
- ICANN gTLD Applicant Guidebook, Version 2012-06-04
- http://tools.ietf.org/html/draft-arias-noguchi-registry-data-escrow
- http://tools.ietf.org/html/draft-arias-noguchi-dnrd-objects-mapping
- http://tools.ietf.org/html/draft-gould-thippeswamy-dnrd-csv-mapping

1.2.2 Internal

- Pre-Delegation Testing, Statement of Work
- Pre-Delegation Testing, Master Test Plan
- Pre-Delegation Testing, Data Escrow Test Plan

1.2.3 Document Hierarchy



1.3 Context

All tests are performed locally in the test environment.

1.4 Notation for description

Each test case for the data escrow is described in their own chapter. The test procedures are described directly in the test case.

2. Data Escrow File Name 01

2.1 Test case identifier

DataEscrowFileName01

2.2 Objective

The test will receive one full deposit of sample data. The objective is to verify file names.

Requirements from the test plan: [R21], [AGB1], [REG4]

2.3 Inputs

The following information will be needed as input for this test case:

Id	Description	Type
TLD	The ASCII compatible name of the TLD	String
DataFileFull-[1..n]	The files containing the full deposit	Files
DataSigFull-[1..n]	The files containing the signature	Files

2.4 Outcome(s)

Files MUST be named according to the following convention:
{gTLD}_{YYYY-MM-DD}_{type}_S{#}_R{rev}.{ext}

2.5 Environmental needs

This test has no environmental needs.

2.6 Special procedural requirements

This test has no special procedural requirements.

2.7 Intercase dependencies

This test has no intercase dependencies.

2.8 Ordered description of steps to be taken to execute the test case

All of the checks are case insensitive.

The data files MUST follow this format {gTLD}_{YYYY-MM-DD}_{type}_S{#}_R{rev}.{ext}
For each **<DataFileFull>**, check that:

1. {gTLD} is equal to **<TLD>**. If it is an IDN-TLD, then this MUST be the A-label.
2. {YYYY-MM-DD} is equal to year, month, and day. The file MUST be maximum 40 days old.
3. {type} is equal to “full”.
4. {#} is a number greater than or equal to 1. Leading zeroes are not allowed.
5. {rev} is a number greater than or equal to 0. Leading zeroes are not allowed.
6. {ext} is equal to “ryde”.

The signature files MUST follow this format {gTLD}_{YYYY-MM-DD}_{type}_S{#}_R{rev}.{ext}
For each **<DataSigFull>**, check that:

1. {gTLD} is equal to **<TLD>**. If it is an IDN-TLD, then this MUST be the A-label.
2. {YYYY-MM-DD} is equal to year, month, and day. The file MUST be maximum 40 days old.
3. {type} is equal to “full”.
4. {#} is a number greater than or equal to 1. Leading zeroes are not allowed.
5. {rev} is a number greater than or equal to 0. Leading zeroes are not allowed.
6. {ext} is equal to “sig”.

3. Data Escrow File Name 02

3.1 Test case identifier

DataEscrowFileName02

3.2 Objective

This test is optional and will only be performed if the applicant has supplied a differential deposit.

The test will receive one differential deposit of sample data. The objective is to verify file names.

Requirements from the test plan: [R21], [AGB2], [REG4]

3.3 Inputs

The following information will be needed as input for this test case:

Id	Description	Type
TLD	The ASCII compatible name of the TLD	String
DataFileDiff-[1..n]	The files containing the differential deposit	Files
DataSigDiff-[1..n]	The files containing the signature	Files

3.4 Outcome(s)

Files MUST be named according to the following convention:
{gTLD}_{YYYY-MM-DD}_{type}_S{#}_R{rev}.{ext}

3.5 Environmental needs

This test has no environmental needs.

3.6 Special procedural requirements

This test has no special procedural requirements.

3.7 Intercase dependencies

This test has no intercase dependencies.

3.8 Ordered description of steps to be taken to execute the test case

All of the checks are case insensitive.

The data files MUST follow this format {gTLD}_{YYYY-MM-DD}_{type}_S{#}_R{rev}.{ext}
For each **<DataFileDiff>**, check that:

1. {gTLD} is equal to **<TLD>**. If it is an IDN-TLD, then this MUST be the A-label.
2. {YYYY-MM-DD} is equal to year, month, and day. The file MUST be maximum 40 days old.
3. {type} is equal to “diff”.
4. {#} is a number greater than or equal to 1. Leading zeroes are not allowed.
5. {rev} is a number greater than or equal to 0. Leading zeroes are not allowed.
6. {ext} is equal to “ryde”.

The signature files MUST follow this format {gTLD}_{YYYY-MM-DD}_{type}_S{#}_R{rev}.{ext}
For each **<DataSigDiff>**, check that:

1. {gTLD} is equal to **<TLD>**. If it is an IDN-TLD, then this MUST be the A-label.
2. {YYYY-MM-DD} is equal to year, month, and day. The file MUST be maximum 40 days old.
3. {type} is equal to “diff”.
4. {#} is a number greater than or equal to1. Leading zeroes are not allowed.
5. {rev} is a number greater than or equal to 0. Leading zeroes are not allowed.
6. {ext} is equal to “sig”.

4. Data Escrow Verify 01

4.1 Test case identifier

DataEscrowVerify01

4.2 Objective

The test will verify the signatures of the received files. If it is a multi-part transmission, then the files are put together. Decrypt and uncompress the result.

Requirements from the test plan: [R21], [AGB1], [REG3], [REG6.1], [REG6.2], [REG6.3], [ALGO]

4.3 Inputs

The following information will be needed as input for this test case:

Id	Description	Type
DataFileFull-[1..n]	The files containing the full deposit	Files
DataSigFull-[1..n]	The files containing the signature	Files
DataRegPubKey	The public key used for verification	File

4.4 Outcome(s)

- The signature, encryption, and compression are done in accordance with RFC 4880.
- The files MUST be signed using RSA, DSA, or ECDSA with SHA1, RIPEMD160, SHA224, SHA256, SHA384, or SHA512.
- If multi-part files, then all files MUST be present.
- The files MUST be encrypted using RSA, Elgamal, or ECDH with IDEA, TripleDES, CAST5, Blowfish, AES128, AES192, AES256, or Twofish.
- The decrypted and uncompressed file will be used in upcoming test.

4.5 Environmental needs

This test has no environmental needs.

4.6 Special procedural requirements

This test has no special procedural requirements.

4.7 Intercase dependencies

DataEscrowFileName01 must first have been executed successfully.

4.8 Ordered description of steps to be taken to execute the test case

All operations are done in accordance with RFC 4880.

For each **<DataSigFull>**:

1. Validate the signature. It MUST be possible to validate the **<DataFileFull>** using the signature and the **<DataRegPubKey>**.
2. Check the properties of the signature:
    a. Digest algorithm SHA1, RIPEMD160, SHA224, SHA256, SHA384, or SHA512 MUST be used. MD5 is deprecated and MUST NOT be used.
    b. Public key algorithm RSA, DSA or ECDSA MUST be used.

If there is more than one **<DataFileFull>**:

1. All file parts MUST be present. See {#} in the file name and that they form a sequence of numbers starting with 1.
2. Concatenate the files in order.

Decrypt and uncompress the (concatenated) file:

1. Decrypt the file using the private test key. The file will be uncompressed automatically by the client software.
2. Check the properties of the encrypted file:
    a. Symmetric algorithm IDEA, TripleDES, CAST5, Blowfish, AES128, AES192, AES256, or Twofish MUST be used.
    b. Public key algorithm RSA, Elgamal or ECDH MUST be used. (Note that this will always be RSA because of the PDT Provider's public key.)
3. Check the original file name of the unencrypted file. It MUST be the same as the encrypted deposit but with extension tar.

Untar the decrypted archive and check that there is an XML file. It MUST be named as the deposit but with the extension xml. The XML file MUST NOT be placed in any subdirectory within the archive.

5. Data Escrow Verify 02

5.1 Test case identifier

DataEscrowVerify02

5.2 Objective

This test is optional and will only be performed if the applicant has supplied a differential deposit.

The test will verify the signature of the received files. If it is a multi-part transmission, then the files are put together. Decrypt and uncompress the result.

Requirements from the test plan: [R21], [AGB2], [REG3], [REG6.1], [REG6.2], [REG6.3], [ALGO]

5.3 Inputs

The following information will be needed as input for this test case:

Id	Description	Type
DataFileDiff-[1..n]	The files containing the differential deposit	Files
DataSigDiff-[1..n]	The files containing the signature	Files
DataRegPubKey	The public key used for verification	File

5.4 Outcome(s)

- The signature, encryption, and compression are done in accordance with RFC 4880.
- The files MUST be signed using RSA, DSA, or ECDSA with SHA1, RIPEMD160, SHA224, SHA256, SHA384, or SHA512.
- If multi-part files, then all files MUST be present.
- The files MUST be encrypted using RSA, Elgamal, or ECDH with IDEA, TripleDES, CAST5, Blowfish, AES128, AES192, AES256, or Twofish.
- The decrypted and uncompressed file will be used in upcoming test.

5.5 Environmental needs

This test has no environmental needs.

5.6 Special procedural requirements

This test has no special procedural requirements.

5.7 Intercase dependencies

DataEscrowFileName02 must first have been executed successfully.

5.8 Ordered description of steps to be taken to execute the test case

All operations are done in accordance with RFC 4880.

For each **<DataSigDiff>**:

1. Validate the signature. It MUST be possible to validate the **<DataFileDiff>** using the signature and the **<DataRegPubKey>**.
2. Check the properties of the signature:
    a. Digest algorithm SHA1, RIPEMD160, SHA224, SHA256, SHA384, or SHA512 MUST be used. MD5 is deprecated and MUST NOT be used.
    b. Public key algorithm RSA, DSA or ECDSA MUST be used.

If there is more than one **<DataFileDiff>**:

1. All file parts MUST be present. See {#} in the file name and that they form a sequence of numbers starting with 1.
2. Concatenate the files in order.

Decrypt and uncompress the (concatenated) file:

1. Decrypt the file using the private test key. The file will be uncompressed automatically by the client software.
2. Check the properties of the encrypted file:
    a. Symmetric algorithm IDEA, TripleDES, CAST5, Blowfish, AES128, AES192, AES256, or Twofish MUST be used.
    b. Public key algorithm RSA, Elgamal or ECDH MUST be used. (Note that this will always be RSA because of the PDT Provider's public key.)
3. Check the original file name of the unencrypted file. It MUST be the same as the encrypted deposit but with extension tar.

Untar the decrypted archive and check that there is an XML file. It MUST be named as the deposit but with the extension xml. The XML file MUST NOT be placed in any subdirectory within the archive.

6. Data Escrow Content 01

6.1 Test case identifier

DataEscrowContent01

6.2 Objective

This test will validate the full deposit against the profile.

Requirements from the test plan: [R21], [R22], [AGB1], [REG2], [REG6.4]

6.3 Inputs

The following information will be needed as input for this test case:

Id	Description	Type
DataFileFull	The unencrypted file containing the full deposit	File
DataProfile	The data escrow profile described using W3C XML Schema. Provided by ICANN.	XML file

6.4 Outcome(s)

The full deposit MUST have valid XML and contain required and valid attributes.

6.5 Environmental needs

This test has no environmental needs.

6.6 Special procedural requirements

This test has no special procedural requirements.

6.7 Intercase dependencies

DataEscrowVerify01 must first have been executed successfully.

6.8 Ordered description of steps to be taken to execute the test case

1. Check if it is an XML or CSV deposit.
    a. If CSV deposit, then the corresponding XML in the tests below.
2. Validate the **<DataFileFull>** XML file against the **<DataProfile>** XML schema provided by ICANN. The applicant MUST use extensions which have been agreed upon with ICANN.
3. Check the content of the XML:
    a. The type MUST be "FULL".
    b. The date part of the watermark MUST match the date in the file name.
    c. There MUST NOT be a "deletes" element in the file.

7. Data Escrow Content 02

7.1 Test case identifier

DataEscrowContent02

7.2 Objective

This test is optional and will only be performed if the applicant has supplied a differential deposit.

This test will validate the differential deposit against the profile.

Requirements from the test plan: [R21], [R22], [AGB2], [REG2], [REG6.4]

7.3 Inputs

The following information will be needed as input for this test case:

Id	Description	Type
DataFileDiff	The unencrypted file containing the differential deposit	File
DataProfile	The data escrow profile described using W3C XML Schema. Provided by ICANN.	XML file

7.4 Outcome(s)

The differential deposit MUST have valid XML and contain required and valid attributes.

7.5 Environmental needs

This test has no environmental needs.

7.6 Special procedural requirements

This test has no special procedural requirements.

7.7 Intercase dependencies

DataEscrowVerify02 must first have been executed successfully.

7.8 Ordered description of steps to be taken to execute the test case

1. Check if it is an XML or CSV deposit.
    a. If CSV deposit, then the corresponding XML in the tests below.
2. Validate the **<DataFilediff>** XML file against the **<DataProfile>** XML schema provided by ICANN. The applicant MUST use extensions which have been agreed upon with ICANN.
3. Check the content of the XML:
    a. The type MUST be “DIFF”.
    b. The prevId attribute MUST be present.
    c. The date part of the watermark MUST match the date in the file name.

8. Global

8.1 Glossary

The glossary is available in the Master Test Plan.

8.2 Document change procedures

Document change procedures are documented in the Master Test Plan.