Registry System Testing

Data Escrow Test Area Specification

Version A

File name: Test Area DataEscrow.docx
Last saved: 2017-07-27

Document control

Document information and security

Made by	Responsible for fact	Responsible for document
Lennart Bonnevier	Mats Dufberg	Mats Dufberg

Security class	File name
External	Test Area DataEscrow.docx

Revisions

Date	Version	Name	Description
2017-07-27	A	Mats Dufberg	First release version.

LIST OF CONTENTS

1. INTRODUCTION 5

1.1 SCOPE 5
1.2 REFERENCES 5
1.2.1 External 5
1.2.2 Internal 5
1.2.3 Document Hierarchy 6
1.3 EARLIER DOCUMENTS 6
1.4 LEVEL IN THE OVERALL SEQUENCE 6
1.5 TEST CLASSES AND OVERALL TEST CONDITIONS 6

2. TEST REQUIREMENTS 7

2.1 TEST ITEMS AND THEIR IDENTIFIERS 7
2.1.1 Statement of Work 7
2.1.2 Applicant Guidebook 7
2.1.3 Specification 2 8
2.1.4 Algorithms 9
2.2 FEATURES TO BE TESTED 10
2.3 FEATURES NOT TO BE TESTED 10
2.4 APPROACH 10
2.5 ITEM PASS/FAIL CRITERIA 10
2.6 SUSPENSION CRITERIA AND RESUMPTION REQUIREMENTS 10
2.7 TEST DELIVERABLES 10

3. TEST TRACEABILITY MATRIX 11

4. TEST MANAGEMENT 12

5. TEST CASE DATAESCROWFILENAME01: VERIFY FILE NAME, FULL ESCROW 13

5.1 TEST CASE IDENTIFIER 13
5.2 OBJECTIVE 13
5.3 INPUTS 13
5.4 OUTCOME(S) 13
5.5 ENVIRONMENTAL NEEDS 13
5.6 SPECIAL PROCEDURAL REQUIREMENTS 13
5.7 INTERCASE DEPENDENCIES 13
5.8 ORDERED DESCRIPTION OF STEPS TO BE TAKEN TO EXECUTE THE TEST CASE 14

6. TEST CASE DATAESCROWFILENAME02: VERIFY FILE NAME, DIFFERENTIAL ESCROW 15

6.1 TEST CASE IDENTIFIER 15
6.2 OBJECTIVE 15
6.3 INPUTS 15
6.4 OUTCOME(S) 15
6.5 ENVIRONMENTAL NEEDS 15
6.6 SPECIAL PROCEDURAL REQUIREMENTS 15
6.7 INTERCASE DEPENDENCIES 15
6.8 ORDERED DESCRIPTION OF STEPS TO BE TAKEN TO EXECUTE THE TEST CASE 16

7. TEST CASE DATAESCROWVERIFY01: VERIFY SIGNATURES, FULL ESCROW 17

7.1 TEST CASE IDENTIFIER 17
7.2 OBJECTIVE 17
7.3 INPUTS 17
7.4 OUTCOME(S) 17
7.5 ENVIRONMENTAL NEEDS 17
7.6 SPECIAL PROCEDURAL REQUIREMENTS 17
7.7 INTERCASE DEPENDENCIES 17
7.8 ORDERED DESCRIPTION OF STEPS TO BE TAKEN TO EXECUTE THE TEST CASE 18

8. TEST CASE DATAESCROWVERIFY02: VERIFY SIGNATURES, DIFFERENTIAL ESCROW 19

8.1 TEST CASE IDENTIFIER 19
8.2 OBJECTIVE 19
8.3 INPUTS 19
8.4 OUTCOME(S) 19
8.5 ENVIRONMENTAL NEEDS 19
8.6 SPECIAL PROCEDURAL REQUIREMENTS 19
8.7 INTERCASE DEPENDENCIES 19
8.8 ORDERED DESCRIPTION OF STEPS TO BE TAKEN TO EXECUTE THE TEST CASE 20

9. TEST CASE DATAESCROWCONTENT01: VALIDATE CONTENT, FULL ESCROW 21

9.1 TEST CASE IDENTIFIER 21
9.2 OBJECTIVE 21
9.3 INPUTS 21
9.4 OUTCOME(S) 21
9.5 ENVIRONMENTAL NEEDS 21
9.6 SPECIAL PROCEDURAL REQUIREMENTS 21
9.7 INTERCASE DEPENDENCIES 21
9.8 ORDERED DESCRIPTION OF STEPS TO BE TAKEN TO EXECUTE THE TEST CASE 21

10. TEST CASE DATAESCROWCONTENT02: VERIFY CONTENT, DIFFERENTIAL ESCROW ... 22

10.1 TEST CASE IDENTIFIER 22
10.2 OBJECTIVE 22
10.3 INPUTS 22
10.4 OUTCOME(S) 22
10.5 ENVIRONMENTAL NEEDS 22
10.6 SPECIAL PROCEDURAL REQUIREMENTS 22
10.7 INTERCASE DEPENDENCIES 22
10.8 ORDERED DESCRIPTION OF STEPS TO BE TAKEN TO EXECUTE THE TEST CASE 22

11. GENERAL 23

11.1 GLOSSARY 23
11.2 DOCUMENT CHANGE PROCEDURES 23

1. Introduction

This document describes the Data Escrow Level Tests within the Registry System Testing framework.

1.1 Scope

The Registry System Testing Provider will validate the format of the data escrow deposit as provided by the Registry Operator.

1.2 References

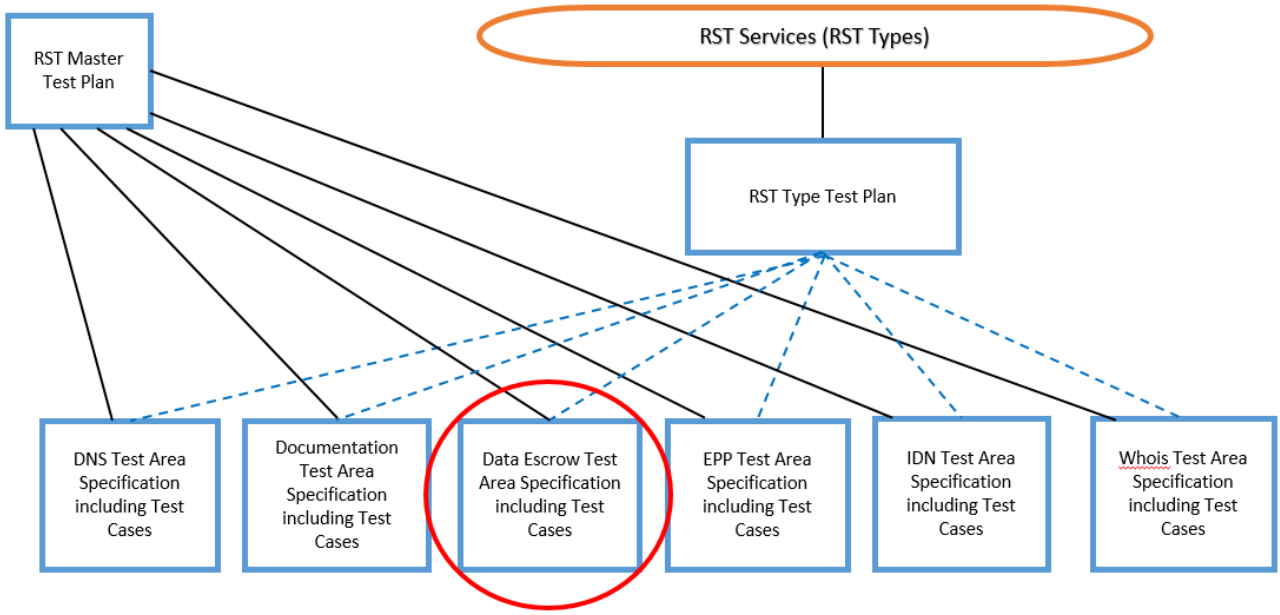
1.2.1 External

- IEEE 829-2008
- ICANN gTLD Applicant Guidebook, Version 2012-06-04
- RA, "Registry Agreement", https://newgtlds.icann.org/sites/default/files/agreements/agreement-approved-09jan14-en.pdf
- Internet-Draft, "Domain Name Registration Data (DNRD) Objects Mapping", http://tools.ietf.org/html/draft-arias-noguchi-dnrd-objects-mapping
- RFC 4880, "OpenPGP Message Format", https://tools.ietf.org/html/rfc4880
- OpenPGP-IANA-Registry, "Pretty Good Privacy", http://www.iana.org/assignments/pgp-parameters/pgp-parameters.xhtml
- draft-arias-noguchi-registry-data-escrow, "Registry Data Escrow Specification", https://tools.ietf.org/html/draft-arias-noguchi-registry-data-escrow
- draft-arias-noguchi-dnrd-objects-mapping, "Domain Name Registration Data (DNRD) Objects Mapping", https://tools.ietf.org/html/draft-arias-noguchi-dnrd-objects-mapping

1.2.2 Internal

- Pre-Delegation Testing, Statement of Work
- Registry System Testing, Master Test Plan

1.2.3 Document Hierarchy



This document is one of many Test Area Specifications for RST (circled in red in the above graphic). It defines the Test Cases for its Test Area.

1.3 Earlier documents

This document replaces, in contents, the following documents that were part of PDT (Pre-Delegation Testing):

- Pre-Delegation Testing: Data Escrow Test Plan (version E)
- Pre-Delegation Testing: Data Escrow Test Cases (Version E)

1.4 Level in the overall sequence

This Test Area and the associated Test Cases can be run in parallel with the other Test Areas.

1.5 Test classes and overall test conditions

Both full and incremental deposits of sample data will be tested with positive test cases.

2. Test Requirements

2.1 Test items and their identifiers

2.1.1 Statement of Work

The main requirements for testing data escrow are found in the Statement of Work:

[R21] Validate the format of one full and one incremental data escrow deposit as provided by the [Registry Operator] for compliance with the New gTLD Registry Agreement Specification 2 – *Data Escrow Requirements* set forth in Module 5 of the AGB.

[R22] Verify that the [Registry Operator's] data escrow profile is in compliance with section 3 of the New gTLD Registry Agreement Specification 2 – *Data Escrow Requirements* set forth in Module 5 of the AGB.

Requirements [R23] and [R24] in the Statement of Work are also about Data Escrow, but they are only about document reviewing and are handled by the Documentation Test Area. They are thus not included in this test area.

Note 1: As per the Global Base New gTLD Registry Agreement, ICANN allows the Registry Operators to only deliver full data escrow deposits each time. Incremental files will thus be tested only if the RO delivers such files to the RST Service Provider.

Note 2: The RST Service Provider shall validate the deposit against an archive of ICANN approved data escrow schemas. Requirement [R22] is thus more about checking that the Registry Operator is using XML schemas which have been agreed upon with ICANN.

2.1.2 Applicant Guidebook

Section 5.2 of the AGB states the following requirements:

> ***Escrow deposit --*** The […] samples of data deposit [provided by the Registry Operator] that include both a full and an incremental deposit showing correct type and formatting of content will bereviewed. Special attention will be given to the agreement with the escrow provider to ensure that escrowed data can be released within 24 hours should it be necessary. ICANN may, at its option, ask an independent third party to demonstrate the reconstitutability of the registry from escrowed data. ICANN may elect to test the data release process with the escrow agent.

The following requirements have been identified from the text above. Note that the requirements on the data escrow agreement are handled by the Documentation Test Area, as mentioned in 2.1.1.

[AGB1] One full deposit of sample data MUST be tested
[AGB2] One differential deposit of sample data deposit MUST be tested

Note 1: The word "differential" is used in the requirement above and not "incremental". This is because specification 2 uses this term.

2.1.3 Specification 2

Specification 2 of the Registry Agreement ("RA") will not be fully cited here, but a number of requirements have been identified, "RS" plus index referring to where in Specification 2 of RA that the requirement has its origin.

[RS3] Escrow Format Specification:

[RS3.1a] Registry objects, such as domains, contacts, name servers, registrars, etc. will be compiled into a file constructed as described in draft-arias-noguchi-registry-data-escrow, see [draft-arias-noguchi-registry-data-escrow] and draft-arias-noguchi-dnrd-objects-mapping, see [draft-arias-noguchi-dnrd-objects-mapping] (collectively, the "DNDE Specification").

[RS3.1b] If not already an RFC, Registry Operator will use the most recent draft version of the DNDE Specification available at the Effective Date. Registry Operator may at its election use newer versions of the DNDE Specification after the Effective Date. Once the DNDE Specification is published as an RFC, Registry Operator will implement that version of the DNDE Specification, no later than one hundred eighty (180) calendar days after.

[RS3.1c] UTF-8 character encoding will be used.

[RS3.2] If a Registry Operator offers additional Registry Services that require submission of additional data, not included above, additional "extension schemas" shall be defined in a case by case basis to represent that data. These "extension schemas" will be specified as described in [draft-arias-noguchi-dnrd-objects-mapping]. Data related to the "extensions schemas" will be included in the deposit file described in [RS3.1a]. ICANN and the respective Registry Operator shall work together to agree on such new objects' data escrow specifications.

[RS4] Processing of Deposit files:

[RS4a] Acceptable algorithms for Public-key cryptography, Symmetric-key cryptography, Hash and Compression are those enumerated in RFC 4880, not marked as deprecated in OpenPGP IANA Registry, see [OpenPGP-IANA-Registry], that are also royalty-free.

[RS4-1] The XML file of the deposit as described in [draft-arias-noguchi-registry-data-escrow] must be named as the containing file as specified in [RS5] but with the extension xml.

[RS4-2] The data file(s) are aggregated in a tarball file named the same as [RS4-1] but with extension tar.

[RS4-3a] The suggested algorithm for compression [of the tarball file] is ZIP as per RFC 4880.

[RS4-3b] The compressed data [the tarball file] will be encrypted using the escrow agent's public key. The suggested algorithms for Public-key encryption are Elgamal and RSA as per RFC 4880. The suggested algorithms for Symmetric-key encryption are TripleDES, AES128 and CAST5 as per RFC 4880.

[RS4-4] The file may be split as necessary if, once compressed and encrypted, it is larger than the file size limit agreed with the escrow agent.

[RS4-5a] A digital signature file will be generated for every processed file using the Registry Operator's private key. […] The suggested algorithms for Digital signatures are DSA and RSA as per RFC 4880. The suggested algorithm for Hashes in Digital signatures is SHA256.

[RS4-5b] The digital signature file will be in binary OpenPGP format as per RFC 4880 Section 9, reference 3, and will not be compressed or encrypted.

[RS4-6] The processed files and digital signature files will then be transferred to the Escrow Agent through secure electronic mechanisms […].

[RS4-7] The Escrow Agent will then validate every (processed) transferred data file using the procedure described in [RS8].

[RS5] Files will be named according to the following convention: {gTLD}_{YYYY-MM-DD}_{type}_S{#}_R{rev}.{ext} […]

[RS6] Distribution of public keys:

[RS6a] Each of Registry Operator and Escrow Agent will distribute its public key to the other party (Registry Operator or Escrow Agent, as the case may be) via email to an email address to be specified.

[RS6b] Each party will confirm receipt of the other party's public key with a reply email, and the distributing party will subsequently reconfirm the authenticity of the key transmitted via offline methods […].

[RS8] Verification procedure**:**

[RS8-1] The signature file of each processed file is validated.

[RS8-2] If processed files are pieces of a bigger file, the latter is put together.

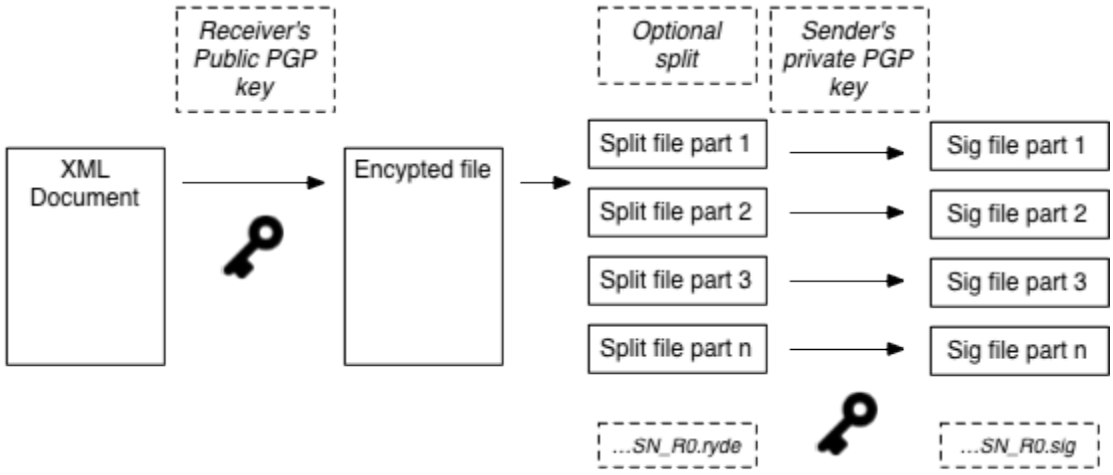
[RS8-3] Each file obtained in the previous step is then decrypted and uncompressed.

[RS8-4] Each data file contained in the previous step is then validated against the format defined in [draft-arias-noguchi-registry-data-escrow].

Note 1: [RS6, RS6a, RS6b] will not be tested, see section 2.4

Note 2: [RS5] is case insensitive.

Processing of files can be illustrated by this picture. The verification procedure is the reverse of this process.



2.1.4 Algorithms

RFC 4880 enumerates a number of algorithms used for Public-key cryptography, Symmetric-key cryptography, Hash and Compression. The algorithm must not be marked as deprecated in OpenPGP IANA Registry and must also be royalty-free.

[ALGO1] Public-key cryptography

[ALGO1.1] RSA used both for signing and encryption. RFC 4880 says that RSA Encrypt-Only and RSA Sign-Only are deprecated.

[ALGO1.2] DSA for signatures and Elgamal for encryption.

[ALGO1.3] ECDSA for signatures and ECDH for encryption. RFC 4880 has reserved code points for this. The full specification is in RFC 6637.

[ALGO2] Symmetric-key cryptography: IDEA, TripleDES, CAST5, Blowfish, AES128, AES192, AES256, or Twofish.

[ALGO3] Hash: SHA1, RIPEMD160, SHA224, SHA256, SHA384, or SHA512.

[ALGO3.1] MD5 is marked as deprecated in the IANA Registry and MUST not be used.

[ALGO4] Compression: ZIP, ZLIB, or BZip2.

Note 1: IDEA was patented in some countries, but the patents have now expired.

2.2 Features to be tested

- File names
- Processing of files
- Escrow profile
- Escrow format

2.3 Features not to be tested

It has been decided not to do a proper public key distribution with offline verification methods, as described in [RS6]. The tests are performed on dummy data and are thus not considered sensitive. Also, this will simplify the testing process. The Registry Operator can get the public key from a public website.

The ability to rebuild the registry from escrowed data will not be tested.

2.4 Approach

The Registry Operator must at least deliver full deposits of sample data including signature files and their own public PGP key. Encryption must be done using the public key provided by the Registry Testing System. The RO must deliver the differential files, if they are supporting this functionality.

The RST Service Provider will verify the files and their contents using manual inspection together with support tools and scripts.

The data escrow draft is still under development, which will impact this test plan and test cases. The test plan and test cases will be updated to match the current draft. However, multiple versions are needed to be supported.

ICANN will provide the XML schemas for any approved extensions made by the Registry Operator. This is what the deposit will be validated against.

ICANN may also elect to test the release process with the escrow agent. This is however described in a separate document because it is an extra service outside the normal Registry Service testing.

2.5 Item pass/fail criteria

The test will fail if the test item does not follow the requirement for each step in the procedure.

2.6 Suspension criteria and resumption requirements

The test cases need to run in the correct order, because the outcome from one test is used in the following test. If a test fails, then the subsequent test cases will not be run.

2.7 Test deliverables

The Data Escrow test area will produce:

- Level Test Logs (LTL)
- Anomaly Report (AR) in case of error
- Level Test Report (LTR)

3. Test Traceability Matrix

This table describes the different test cases and their mapping to the requirements.

Test ID	Description	Requirement Point
DataEscrowFileName01	Receive one full deposit of sample data. Verify file names.	[R21], [AGB1], [RS5]
DataEscrowFileName02	Receive one differential deposit of sample data. Verify file names.	[R21], [AGB2], [RS5]
DataEscrowVerify01	Verify signature of files for full deposit. Put split files together. Decrypt and uncompress file.	[R21], [AGB1], [RS4], [RS8-1], [RS8-2], [RS8-3], [ALGO]
DataEscrowVerify02	Verify signature of files for differential deposit. Put split files together. Decrypt and uncompress file.	[R21], [AGB2], [RS4], [RS8-1], [RS8-2], [RS8-3], [ALGO]
DataEscrowContent01	Validate the full deposit against the profile provided by ICANN.	[R21], [R22] , [AGB1], [RS3], [RS8-4]
DataEscrowContent02	Validate the differential deposit against the profile provided by ICANN.	[R21], [R22], [AGB2], [RS3], [RS8-4]

4. Test management

The goal of these documents is to describe the test cases and how the new gTLDs are tested. This is just a part of a larger project and defining test management is not part of this subproject. However, some information can be found in the Master Test Plan.

5. Test Case DataEscrowFileName01: Verify file name, full escrow

5.1 Test case identifier

DataEscrowFileName01

5.2 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], [RS5]

5.3 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

5.4 Outcome(s)

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

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

This test has no intercase dependencies.

5.8 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".

6. Test Case DataEscrowFileName02: Verify file name, differential escrow

6.1 Test case identifier

DataEscrowFileName02

6.2 Objective

This test is optional and will only be performed if the Registry Operator 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], [RS5]

6.3 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

6.4 Outcome(s)

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

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

This test has no intercase dependencies.

6.8 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".

7. Test Case DataEscrowVerify01: Verify signatures, full escrow

7.1 Test case identifier

DataEscrowVerify01

7.2 Objective

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

Requirements from the test plan: [R21], [AGB1], [RS4], [RS8-1], [RS8-2], [RS8-3], [ALGO]

7.3 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

7.4 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.

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

DataEscrowFileName01 must first have been executed successfully.

7.8 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 RST 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.

8. Test Case DataEscrowVerify02: Verify signatures, differential escrow

8.1 Test case identifier

DataEscrowVerify02

8.2 Objective

This test is optional and will only be performed if the Registry Operator 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 reassembled. Decrypt and uncompress the result.

Requirements from the test plan: [R21], [AGB2], [RS4], [RS8-1], [RS8-2], [RS8-3], [ALGO]

8.3 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

8.4 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.

8.5 Environmental needs

This test has no environmental needs.

8.6 Special procedural requirements

This test has no special procedural requirements.

8.7 Intercase dependencies

DataEscrowFileName02 must first have been executed successfully.

8.8 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 RST 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.

9. Test Case DataEscrowContent01: Validate content, full escrow

9.1 Test case identifier

DataEscrowContent01

9.2 Objective

This test will validate the full deposit against the profile.

Requirements from the test plan: [R21], [R22], [AGB1], [RS3], [RS8-4]

9.3 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

9.4 Outcome(s)

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

9.5 Environmental needs

This test has no environmental needs.

9.6 Special procedural requirements

This test has no special procedural requirements.

9.7 Intercase dependencies

DataEscrowVerify01 must first have been executed successfully.

9.8 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 Registry Operator 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.

10. Test Case DataEscrowContent02: Verify content, differential escrow

10.1 Test case identifier

DataEscrowContent02

10.2 Objective

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

This test will validate the differential deposit against the profile.

Requirements from the test plan: [R21], [R22], [AGB2], [RS3], [RS8-4]

10.3 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

10.4 Outcome(s)

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

10.5 Environmental needs

This test has no environmental needs.

10.6 Special procedural requirements

This test has no special procedural requirements.

10.7 Intercase dependencies

DataEscrowVerify02 must first have been executed successfully.

10.8 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 Registry Operator 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.

11. General

11.1 Glossary

The glossary is available in the Master Test Plan.

11.2 Document change procedures

Document change procedures are documented in the RST Master Test Plan.