WIRELESS POWER
CONSORTIUM

Qi Specification

Authentication Protocol

Version 1.3

January 2021

COPYRIGHT

The *Qi Specification, Authentication Protocol* is published by the Wireless Power Consortium and has been prepared by the members of the Wireless Power Consortium. Reproduction in whole or in part is prohibited without express and prior written permission of the Wireless Power Consortium.

may not otherwise be disclosed in any form to any person or organization without express and prior written permission of the Wireless Power Consortium.

DISCLAIMER

The information contained herein is believed to be accurate as of the date of publication, but is provided "as is" and may contain errors. The Wireless Power Consortium makes no warranty, express or implied, with respect to this document and its contents, including any warranty of title, ownership, merchantability, or fitness for a particular use or purpose. Neither the Wireless Power Consortium, nor any member of the Wireless Power Consortium will be liable for errors in this document or for any damages, including indirect or consequential, from use of or reliance on the accuracy of this document. For any further explanation of the contents of this document, or in case of any perceived inconsistency or ambiguity of interpretation, contact: info@wirelesspowerconsortium.com.

RELEASE HISTORY

Specification Version	Release Date	Description
1.3	January 2021	Initial release of this document.

Table of Contents

1 General 5
1.1 Structure of the Qi Specification. 5
1.2 Scope 6
1.3 Compliance. 6
1.4 References. 6
1.5 Conventions 7
1.6 Power Profiles. 9

2 Overview 10
2.1 References. 11
2.2 Cryptographic methods 13
2.3 Security overview 13
2.4 Impact to existing ecosystem 13
2.5 Support for revocation 14

3 Certificates and private keys 15
3.1 Certificate Chains 15
3.2 Certificates. 17
3.3 Certificate Chain slots 26
3.4 Power Transmitter private keys 26
3.5 Other private keys. 27

4 Authentication protocol. 28
4.1 Digest query 28
4.2 Certificate Chain read 28
4.3 Authentication challenge. 29
4.4 Errors and alerts. 29

5 Authentication messages. 30
5.1 Authentication message header 30
5.2 Authentication requests. 31
5.3 Authentication responses 34

6 Timing requirements 39
6.1 Power Receiver timing requirements. 39
6.2 Power Transmitter timing requirements. 40

7 Protocol flow examples 41
7.1 Simple flow 41
7.2 Flow with caching 42
7.3 Flow with caching and revocation 43
7.4 Challenge first flow. 44

1 General

The Wireless Power Consortium (WPC) is a worldwide organization that aims to develop and promote global standards for wireless power transfer in various application areas. A first application area comprises flat-surface devices such as mobile phones and chargers in the Baseline Power Profile (up to 5 W) and Extended Power Profile (above 5 W).

1.1 Structure of the Qi Specification

General documents

- Introduction
- Glossary, Acronyms, and Symbols

System description documents

- Mechanical, Thermal, and User Interface
- Power Delivery
- Communications Physical Layer
- Communications Protocol
- Foreign Object Detection
- NFC/RFID Card Protection
- Authentication Protocol

Reference design documents

- Power Transmitter Reference Designs
- Power Receiver Design Examples

Compliance testing documents

- Power Transmitter Test Tools
- Power Receiver Test Tools
- Power Transmitter Compliance Tests
- Power Receiver Compliance Tests

NOTE: The compliance testing documents are restricted and require signing in to the WPC members' website. All other specification documents are available for download on both the WPC public website and the WPC website for members.

1.2 Scope

The *Qi Specification, Authentication Protocol* (this document) defines the architecture and application-level messaging for the Authentication of a Power Transmitter Product by a Power Receiver to ensure that the Power Transmitter Product is both Qi certified and the product of a registered manufacturer.

1.3 Compliance

All provisions in the *Qi Specification* are mandatory, unless specifically indicated as recommended, optional, note, example, or informative. Verbal expression of provisions in this Specification follow the rules provided in ISO/IEC Directives, Part 2.

Table 1: Verbal forms for expressions of provisions

Provision	Verbal form
requirement	"shall" or "shall not"
recommendation	"should" or "should not"
permission	"may" or "may not"
capability	"can" or "cannot"

1.4 References

For undated references, the most recently published document applies. The most recent WPC publications can be downloaded from http://www.wirelesspowerconsortium.com. In addition, the *Qi Specification* references documents listed below. Documents marked here with an asterisk (*) are restricted and require signing in to the WPC website for members.

- Product Registration Procedure Web page*
- Qi Product Registration Manual, Logo Licensee/Manufacturer*
- Qi Product Registration Manual, Authorized Test Lab*
- Power Receiver Manufacturer Codes,* Wireless Power Consortium
- The International System of Units (SI), Bureau International des Poids et Mesures
- Verbal forms for expressions of provisions, International Electotechnical Commission

For regulatory information about product safety, emissions, energy efficiency, and use of the frequency spectrum, visit the regulatory environment page of the WPC members' website.

1.5 Conventions

1.5.1 Notation of numbers

- Real numbers use the digits 0 to 9, a decimal point, and optionally an exponential part.
- Integer numbers in decimal notation use the digits 0 to 9.
- Integer numbers in hexadecimal notation use the hexadecimal digits 0 to 9 and A to F, and are prefixed by "0x" unless explicitly indicated otherwise.
- Single bit values use the words ZERO and ONE.

1.5.2 Tolerances

Unless indicated otherwise, all numeric values in the *Qi Specification* are exactly as specified and do not have any implied tolerance.

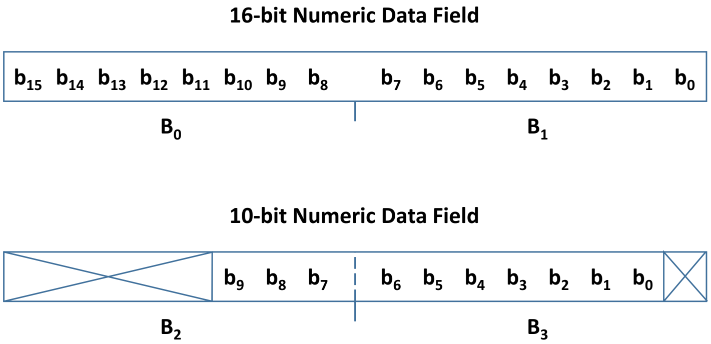
1.5.3 Fields in a data packet

A numeric value stored in a field of a data packet uses a big-endian format. Bits that are more significant are stored at a lower byte offset than bits that are less significant. Table 2 and Figure 1 provide examples of the interpretation of such fields.

Table 2: Example of fields in a data packet

<table>
<tr><th></th><th>b_7</th><th>b_6</th><th>b_5</th><th>b_4</th><th>b_3</th><th>b_2</th><th>b_1</th><th>b_0</th></tr>
<tr><th>B_0</th><td colspan="8" rowspan="2">(msb)
16-bit Numeric Data Field
(lsb)</td></tr>
<tr><th>B_1</th></tr>
<tr><th>B_2</th><td colspan="5">Other Field</td><td colspan="3">(msb)</td></tr>
<tr><th>B_3</th><td colspan="7">10-bit Numeric Data Field (lsb)</td><td>Field</td></tr>
</table>

Figure 1. Examples of fields in a data packet


16-bit Numeric Data Field
b_{15} b_{14} b_{13} b_{12} b_{11} b_{10} b_9 b_8 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0
B_0
B_1
10-bit Numeric Data Field
b_9 b_8 b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0
B_2
B_3

1.5.4 Notation of text strings

Text strings consist of a sequence of printable ASCII characters (i.e. in the range of 0x20 to 0x7E) enclosed in double quotes ("). Text strings are stored in fields of data structures with the first character of the string at the lowest byte offset, and are padded with ASCII NUL (0x00) characters to the end of the field where necessary.

EXAMPLE: The text string “WPC” is stored in a six-byte fields as the sequence of characters 'W', 'P', 'C', NUL, NUL, and NUL. The text string “M:4D3A” is stored in a six-byte field as the sequence 'M', ':', '4', 'D', '3', and 'A'.

1.5.5 Short-hand notation for data packets

In many instances, the *Qi Specification* refers to a data packet using the following shorthand notation:

<MNEMONIC>/<modifier>

In this notation, <MNEMONIC> refers to the data packet's mnemonic defined in the *Qi Specification, Communications Protocol*, and <modifier> refers to a particular value in a field of the data packet. The definitions of the data packets in the *Qi Specification, Communications Protocol*, list the meanings of the modifiers.

For example, EPT/cc refers to an End Power Transfer data packet having its End Power Transfer code field set to 0x01.

1.6 Power Profiles

A Power Profile determines the level of compatibility between a Power Transmitter and a Power Receiver. Table 3 defines the available Power Profiles.

- *BPP PTx*: A Baseline Power Profile Power Transmitter.
- *EPP5 PTx*: An Extended Power Profile Power Transmitter having a restricted power transfer capability, i.e. $\mathrm{P}_{\mathrm{L}}^{(\mathrm{pot})}$ = 5 W.
- *EPP PTx*: An Extended Power Profile Power Transmitter.
- *BPP PRx*: A Baseline Power Profile Power Receiver.
- *EPP PRx*: An Extended Power Profile Power Receiver.

Table 3: Capabilities included in a Power Profile

Feature	BPP PTx	EPP5 PTx	EPP PTx	BPP PRx	EPP PRx
A*x* or B*x* design	Yes	Yes	No	N/A	N/A
MP-A*x* or MP-B*x* design	No	No	Yes	N/A	N/A
Baseline Protocol	Yes	Yes	Yes	Yes	Yes
Extended Protocol	No	Yes	Yes	No	Yes
Authentication	N/A	Optional	Yes	N/A	Optional

2 Overview

The *Qi Specification, Authentication Protocol* (this document) defines a protocol for a Power Receiver to authenticate a Power Transmitter. In this context, Authentication is a tamper-resistant method to establish and verify the identity of the Power Transmitter, enabling the Power Receiver to trust the Power Transmitter to operate within the bounds of the *Qi Specification*. This Authentication protocol version 1.0 makes use of Data Transport Streams between the Power Receiver and Power Transmitter as defined in the *Qi Specification, Communications Protocol*.

Authentication allows an organization to set and enforce a policy with regard to acceptable products. This will permit useful security assurances in real world situations. For example, a mobile phone manufacturer concerned about product damage or safety hazards resulting from substandard wireless charging devices can set a policy limiting the power drawn from an untrusted wireless charger.

This document aims to be closely aligned with the USB Authentication specification, particularly as it is likely that products will exist in the market that support both.

2.1 References

Unless specified otherwise, all standards specified, including those from ISO, ITU, and NIST refer to the version or edition which is more recent, as of 1 January 2018.

ECDSA

- ANSI X9.62-2005; Public Key Cryptography for the Financial Services Industry, The Elliptic Curve Digital Signature Algorithm (ECDSA) (available at www.global.ihs.com or https://www.techstreet.com)
- NIST-FIPS-186-4, Digital Signature Standard (DSS), Section 6, Federal Information Processing Standards Publication, July 2013 (available at: http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf)
- ISO/IEC 14888-3 Digital signatures with appendix—Part 3: Discrete logarithm based mechanisms (Clause 6.6)

NIST P-256, secp-256r1

- Certicom-SEC-2 (available at: http://www.secg.org/sec2-v2.pdf)
- NIST-FIPS-186-4, Digital Signature Standard (DSS), Appendix D: Recommended Elliptic Curves for Federal Government Use, Federal Information Processing Standards Publication, July 2013 (available at: http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf)
- ISO/IEC 15946 Cryptographic techniques based on elliptic curves (NIST P-256 is included as example)

  NOTE: The ISO/IEC 15946 series treats elliptic curves differently from FIPS 186- 4. ISO/IEC 15946-5 is about elliptic curve generation. That is, based on the method in part 5, each application and implementation can generate its own curves to use. In other words, there are no ISO/IEC recommended curves. P-256 is considered an example in ISO/IEC 15946. In addition, Elliptic Curve signatures and key establishment schemes have been moved to ISO/IEC 14888 and ISO/IEC 11770 respectively, together with other discrete-log based mechanisms. Test vectors (examples) using P-256 are included for each of those mechanisms.

SHA-256

- NIST-FIPS-180-4 (available at: http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180 -4.pdf)
- ISO/IEC 10118-3 Hash-functions—Part 3: Dedicated hash-functions (Clause 10)

SP800-90A

- NIST-SP-800-90A Revision 1 (available at: http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90Ar1.pdf.)

SP800-90B

- NIST-SP-800-90B (available at: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90B.pdf)

USB Authentication

- Universal Serial Bus Type-C™ Authentication Specification (available at http://www.usb.org/developers/docs/ as part of the USB 3.2 Specification download package)

X.509

- ITU-T-X.509, Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks (available at: https://www.itu.int/rec/T-REC-X.509/en)
- ISO/IEC 9594-8, Information technology—Open Systems Interconnection—The Directory—Part 8: The Directory: Public-key and attribute certificate frameworks (available at: https://www.iso.org/standard/80325.html)
- RFC5280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile

2.2 Cryptographic methods

This specification targets a 128-bit security level for all cryptographic primitives. The cryptographic methods used by this specification are shown in Table 5 (in Section 3.1).

2.2.1 Random number generators

The generation of cryptographic keys and the cryptographic protocol exchanges rely on cryptographic quality random numbers. Random numbers are defined as numbers that are distinguishable from random by no algorithm with an algorithmic complexity of less than $O(2^{128})$.

The output of a NIST SP800-90A-compliant PRNG seeded with a 256-bit full SP800-90B entropy value is sufficient to meet this standard.

2.3 Security overview

Security of the Authentication protocol depends on the protection of private keys, and that protection is supported by security evaluation.

2.3.1 Methodology

This specification defines a Certificate-based method for Authentication that allows a Power Receiver to authenticate a Power Transmitter and, by policy, choose how to interact with that Power Transmitter. For example, a Power Receiver may choose not to use the full advertised capabilities of an unauthenticated Power Transmitter.

2.3.2 Periodic re-Authentication

The Power Receiver can optionally perform periodic re-Authentications to verify that an authenticated Power Transmitter has not been replaced by a different one.

2.4 Impact to existing ecosystem

The impact to existing Power Transmitter Products depends largely on individual Policy decisions regarding legacy Power Transmitters (i.e. Power Transmitters that predate this specification). For example, a Power Receiver with a Policy that allows full functioning of legacy Power Transmitters will have a minimal impact on the current ecosystem, while a Power Receiver with a Policy that limits or refuses to use functionalities exposed by a legacy Power Transmitter will have a more significant impact.

2.5 Support for revocation

If the Power Receiver supports Authentication functionality, then the Power Receiver should support Revocation.

Revocation Lists are normally updated off line, e.g. as part of a software or firmware update or during Product manufacturing. Manufacturers are not required to provide on-line maintenance of Revocation Lists or to make Revocation Lists field-updatable in their Products, however, the Power Receiver should frequently update its revocation list and not only during firmware updates. For some Power Receiver Products with Internet connectivity (e.g. smartphones), frequent Revocation List updates should be relatively easy. Other Power Receiver Products without direct Internet connectivity, such as fitness trackers, may not be able to update their Revocation List as frequently, but manufacturers should consider ways to keep the Revocation List current in their product designs.

NOTE: Support for updatable Revocation Lists implies field-updatable non-volatile memory.

NOTE: The WPC Authentication License Administrator (WPC-ALA) will issue Revocation Lists from time to time. The format of Revocation Lists and how they are communicated is outside the scope of this specification.

3 Certificates and private keys

3.1 Certificate Chains

The requirements in this section apply to Certificate Chains that are populated in slots 0 and 1. The Power Transmitter Product manufacturer determines the requirements for Certificates that are populated in slots 2 and 3. For further information about slots, see Section 3.3, *Certificate Chain slots*.

A Certificate Chain consists of three or four Certificates; see Table 4. The first Certificate in the chain is the Root Certificate identifying the WPC Root Certificate Authority. It is a self-signed Certificate (i.e. signed by the WPC Root Certificate Authority). As an optimization, the Certificate Chain contains a hash of the Root Certificate rather than the Root Certificate itself. The second Certificate is a Manufacturer CA Certificate identifying the product's manufacturer. It is signed by the WPC Root Certificate Authority. The last Certificate in the chain is a Product Unit Certificate identifying the individual Power Transmitter product. It is signed by the Manufacturer Certificate Authority.

In addition to identification data, the Manufacturer CA Certificate contain a public key for verifying the authenticity of the Product Unit Certificate. The Product Unit Certificate contains a public key for verifying the authenticity of the Power Transmitter Product Unit. See Section 4, *Authentication protocol*, for details.

The following requirements apply to the Certificates in a Certificate Chain.

- Each Product Unit Certificate shall identify a unique Power Transmitter Product Unit by using, for example, the product unit's RSID number (wpc-qi-rsid).
- Each Power Transmitter Product Unit shall have a unique public key.
- Product Unit Certificates contained in different slots of the same Power Transmitter Product Unit may contain the same identity and public key, provided that this does not introduce a security vulnerability.
- A manufacturer may have multiple Manufacturer CA Certificates to identify, for example, different production facilities or product families. Each of these Manufacturer CA Certificates shall contain a unique public key relating to a unique private key.

  NOTE: Using multiple Manufacturer CA Certificates is encouraged to partition volumes of product units to limit the collateral impact caused in the event that a Manufacturer CA Certificate is revoked (collateral impact of revoking all product units signed by that Manufacturer CA Certificate).
- A manufacturer shall use the private key associated with a Manufacturer CA Certificate to sign Product Unit Certificates.
- The WPC Root Certificate Authority shall ensure that all Manufacturer CA Certificates contain unique public keys.
- A manufacturer's internal Certificate Authority shall ensure that all of the manufacturer's Secondary Certificates contain unique public keys.

The Certificate Chains populated in slots 0 and 1 shall use X.509 Certificates. See Section 3.3, *Certificate Chain slots*, for further information about slots 0 and 1.

A Power Receiver should not trust a Power Transmitter if it finds any Certificate Chain in slot 0 or slot 1 to be revoked, regardless of whether another Certificate Chain is present that is not revoked. Product manufacturers determine the requirements for Certificate Chains populated in slots 2 and 3. For further information, see Section 3.3, *Certificate Chain slots*.

Table 4: Certificate Chain (X.509 Certificates)

	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
B_0...B_1	msb	Length						
								lsb
B_2...$B_{(1+N\_RH)}$	Root Certificate Hash (length N_RH)							
$B_{(2+N\_RH)}$...$B_{(1+N\_RH+N\_MC)}$	Manufacturer CA Certificate (length N_MC)							
$B_{(2+N\_RH+N\_MC)}$... $B_{(1+N\_RH+N\_MC+N\_PUC)}$	Product Unit Certificate (length N_PUC)							
Note: The Root Certificate is not included in the chain, only its hash is included in the chain.								

Length The length is the total number of bytes in the Certificate Chain including the Length field.
Root Certificate Hash A SHA-256 hash of the Root Certificate. The length of N_RH is 32 bytes. See Section 2.1, *References*.

Manufacturer Certificate This Certificate identifies the manufacturer.

Product Unit Certificate This Certificate identifies the individual product unit.

3.2 Certificates

3.2.1 Format of Certificates

All Certificates shall use:

- the X.509 v3 ASN.1 structure,
- binary DER encoding for ASN.1, and
- the cryptographic methods listed in Table 5.

Table 5: Summary of cryptographic methods

Method	Use
X.509 v3, DER encoding	Certificate format
ECDSA using the NIST P256, secp256r1 curve, uncompressed point OR compressed point format as specified by Sec 2.2. of RFC5280	Digital signing of Certificates and Authentication Messages
SHA-256	Hash algorithm, used in the ECDSA calculation and in creating digests of Certificates and Certificate Chains.

The further description of the Certificate format assumes that the reader is familiar with X.509 v3 Certificate terminology.

NOTE: Certificates and the fields, attributes, and extensions defined therein are Big Endian.

Product unit certificates shall not exceed MaxProdCertSize in length. A Manufacturer certificate or Secondary certificate shall not exceed MaxManufacturerCertSize in length.

3.2.1.1 Textual format

All textual ASN.1 objects contained within X.509 Certificates, including DirectoryString, GeneralName, and DisplayText, shall be specified as a UTF8String. The length of any textual object shall not exceed 64 bytes excluding the DER type and DER length encoding.

3.2.1.2 Attributes and Extensions

Where applicable, the Object Identifier (OID) is provided.

Basic Constraints (OID 2.5.29.19)

This extension shall be present in a Manufacturer or Secondary certificate. When Basic Constraints is included it shall be marked as critical, and the cA component shall be true.

This extension shall not be present in a Product Unit Certificate.

Validity

The notBefore and notAfter fields indicate the time interval during which information regarding a Certificate's validity is maintained. For Product Units, the validity times should be ignored.

Certificate notBefore and notAfter validity times shall be specified using ASN.1 GeneralizedTime for any year, or ASN.1 UTCTime for years prior to 2050.

For a Manufacturer CA Certificate it is recommended that the notBefore field be "19700101000000Z" (for 00:00 on 01-Jan-1970 UTC, which is POSIX epoch time). It is recommended that the notAfter field be "99991231235959Z" (for 23:59:59 on 31-Dec-9999 UTC, which is used for an unknown expiration time as defined in IETF-RFC-5280, Section 4.1.2.5). Use of the recommended notBefore and notAfter values will maximize compatibility with certificate processing stacks.

Qi policy extension (wpc-qi-policy)

The WPC Qi policy extension is a custom Manufacturer CA Certificate extension for use with products compliant to this specification. The value is a binary object and is reserved in this version of the specification. The value of wpc-qi-policy is listed on The Qi Specifications page of the WPC members' website.

The binary object is encoded as an ASN.1 DER OCTET STRING with a fixed size of QiPolicySize bytes.

Manufacturer CA Certificates shall contain this extension. Product Unit Certificates shall not contain this extension.

Qi RSID extension (wpc-qi-rsid)

The WPC Qi revocation sequential identifier (RSID) extension is a custom Product Unit Certificate extension for use with products compliant to this specification. The value is a binary object and is provided to store a sequential identifier unique to each product unit that enables range-based revocation of batches. The value of wpc-qi-rsid is listed on The Qi Specifications page of the WPC members' website.

The binary object is DER encoded as an ASN.1 OCTET STRING with a maximum size of MaxQiRSIDSize bytes. The RSID value is unique to each product unit, from a minimum of 1 up to MaxQiRSIDSize bytes.

Product Unit Certificates shall contain this extension. Manufacturer CA Certificates shall not contain this extension.

3.2.1.3 Certificate content

A Manufacturer CA or Product Unit Certificate uses the ASN.1 structure defined in Rec. ITU-T X.509 (10/2016), sec 7.2 Public-key Certificate.

A Manufacturer CA Certificate shall implement only the ASN.1 field values as specified Table 6.

Table 6: Certificate profile for a Manufacturer CA Certificate

	Field	OID	Data type	Value	Example
TBSCertificate	Version		INTEGER	0x02 {=X.509v3}	0x02
	SerialNumber		INTEGER	A unique number up to 9 bytes in length	0x01 10 20 30 40 50 60 70
	signature	1.2.840.10045.4.3.2 {ecdsa-with-SHA256}		N/A	
	issuer	2.5.4.3 {Common Name}	UTF8String	"WPCCA"+ one character suffix denoting different root CA instances	"WPCCA1"
	validity.notBefore		Either GeneralizedTime for any year, or UTCTime for years prior to 2050	Any value (not used by PRx)	2000-01-01 00:00:00
	validity.notAfter		Either GeneralizedTime for any year, or UTCTime for years prior to 2050	Any value (not used by PRx)	9999-12-31 23:59:59
	subject	2.5.4.3 {Common Name}	UTF8String	7 byte string consisting of: ▪ four upper-case characters containing a PTMC hex value ▪ one character containing a dash ▪ two arbitrary alpha-numeric characters	“CACA-1A”

Table 6: Certificate profile for a Manufacturer CA Certificate (Continued) (Continued)

	Field	OID	Data type	Value	Example
TBSCertificate	subjectPublicKeyInfo.algorithm	1.2.840.10045.2.1 {ecPublicKey}		N/A	
		1.2.840.10045.3.1.7[*] {secp256r1}		N/A	
	subjectPublicKeyInfo.subjectPublicKey		BIT STRING	(Public Key value; optionally may use compressed point representation)[†]	0x04:64:68:34:24:4e:1a:37:b2:f8:3a:d5:30:68:73: 8a:65:9b:e2:a6:d2:c6:f3:c9:3f:90:5e:8c:7d:a3: 59:79:27:6b:a7:fb:d4:30:83:42:a9:90:a7:c1:92: 90:98:20:7d:90:77:f2:97:f3:f5:3a:77:25:01:3c: 55:44:19:e8:4a
	Extensions.1	2.5.29.19 {basicConstraints}		N/A	
	Extensions.1.critical		BOOLEAN	TRUE	TRUE
	Extensions.1.extnValue.cA		BOOLEAN	TRUE	TRUE
	Extensions.1.extnValue.pathLenConstraint		INTEGER	0	0x00
	Extensions.2	2.23.255.1.1 {wpc-qi-policyFlags}		N/A	
	Extensions.2.critical		BOOLEAN	TRUE	TRUE
	Extensions.2.extnValue		OCTET STRING	4 octets (bytes) reserved for future use	0x00 00 00 00

[*] 1.2.840.10045 refers to the named curve defined equivalently by three organizations as:
NIST (FIPS186-4): "P-256"
SECG (SEC 2): "secp256r1"
IETF (RFC 3279): "prime256v1"

[†] A certificate verifier can unambiguously distinguish compressed from uncompressed point representations by evaluating the value of the first byte (0x02 and 0x03 for compressed points and 0x04 for an uncompressed point).

A Product Unit Certificate shall have the fields shown in Table 7.

Table 7: Certificate profile for a Product Unit Certificate

	Field	OID	Data Type	Value	Example
TBSCertificate	Version		INTEGER	0x02 {=X.509v3}	0x02
	serialNumber		INTEGER	A unique number up to 9 bytes in length	0x01 10 20 30 40 50 60 70
	signature	1.2.840.10045.4.3.2 {ecdsa-with-SHA256}		N/A	
	issuer	2.5.4.3 {Common Name}	UTF8String	7 bytes string consisting of: ▪ Four upper-case characters containing a PTMC hex value ▪ one character containing a dash ▪ two arbitrary alpha-numeric characters ▪ this value shall match exactly byte-for-byte the value from the Manufacturer CA Certificate Subject common name	"CACA-1A"
	validity.notBefore		Either GeneralizedTime for any year, or UTCTime for years prior to 2050	Should use the current time when issuing the certificate (not used by PRx)	2020-10-01 00:00:00
	validity.notAfter		Either GeneralizedTime for any year, or UTCTime for years prior to 2050	Should use one day after NotBefore (not used by PRx)	2020-10-02 00:00:00

Table 7: Certificate profile for a Product Unit Certificate (Continued)

	Field	OID	Data Type	Value	Example
TBSCertificate	subject.attribute1	2.5.4.3 {Common Name}	UTF8String	String consisting of: ▪ The Qi ID encoded as a six-character text string, left-padded with UTF-8 "0" (zero) characters as necessary. For example, the Subject ID for a product with Qi ID 6386 is encoded as "006386" ▪ Optionally one character containing a dash followed by up to 28 arbitrary characters to improve readability e.g. product name	"006386-Model5"
	subject.attribute2 {optional}	2.5.4.92 {tagAFI}	OCTET STRING	Optional: up to 32 arbitrary bytes	0x20 20 09 23 F4
	subject.attribute3 {optional}	0.9.2342.19200300.100.1.1 {userId}	UTF8String	Optional: up to 32 arbitrary characters	"SEPT-23"
	subjectPublicKeyInfo.algorithm	1.2.840.10045.2.1 {ecPublicKey}		N/A	
		1.2.840.10045.3.1.7* {secp256r1}		N/A	

Table 7: Certificate profile for a Product Unit Certificate (Continued)

	Field	OID	Data Type	Value	Example
TBSCertificate	subjectPublicKeyInfo.subjectPublicKey		BIT STRING	(Public Key value; optionally may use compressed point representation)[†]	0x04:64:68:34:24:4e:1a:37:b2:f8:3a:d5:30:68:73: 8a:65:9b:e2:a6:d2:c6:f3:c9:3f:90:5e:8c:7d:a3: 59:79:27:6b:a7:fb:d4:30:83:42:a9:90:a7:c1:92: 90:98:20:7d:90:77:f2:97:f3:f5:3a:77:25:01:3c: 55:44:19:e8:4a
	Extensions.1	2.23.255.1.2 {wpc-qi-auth-RSID}		N/A	
	Extensions.1.critical		BOOLEAN	TRUE	TRUE
	Extensions.1.extnValue		OCTET STRING	RSID	0x00 00 00 00 00 00 00 00 01

[*] 1.2.840.10045 refers to the named curve defined equivalently by three organizations as:
NIST (FIPS186-4): "P-256"
SECG (SEC 2): "secp256r1"
IETF (RFC 3279): "prime256v1"

[†] A certificate verifier can unambiguously distinguish compressed from uncompressed point representations by evaluating the value of the first byte (0x02 and 0x03 for compressed points and 0x04 for an uncompressed point).

3.2.1.4 Root CA Certificate

A Root CA certificate shall implement the ASN.1 field values as specified in Table 8.

Table 8: Certificate profile for a Root CA Certificate

	Field	OID	Data Type	Value	Example
TBSCertificate	Version		INTEGER	0x02 {=X.509v3}	0x02
	serialNumber		INTEGER	A unique number up to 9 bytes in length	0x01 10 20 30 40 50 60 70
	signature	1.2.840.10045.4.3.2 {ecdsa-with-SHA256}		N/A	
	issuer	2.5.4.3 {Common Name}	UTF8String	"WPCCA"+ one character suffix denoting different root CA instances	"WPCCA1"
	validity.notBefore		Either GeneralizedTime for any year, or UTCTime for years prior to 2050	Any value (not used by PRx)	2000-01-01 00:00:00
	validity.notAfter		Either GeneralizedTime for any year, or UTCTime for years prior to 2050	Any value (not used by PRx)	9999-12-31 23:59:59
	subject	2.5.4.3 {Common Name}	UTF8String	Same as "issuer"	"WPCCA1"
	subjectPublicKeyInfo.algorithm	1.2.840.10045.2 .1 {ecPublicKey}		N/A	
		1.2.840.10045.3.1.7 {secp256r1}		N/A	

Table 8: Certificate profile for a Root CA Certificate (Continued)

	Field	OID	Data Type	Value	Example
TBSCertificate	subjectPublicKeyInfo.subjectPublicKey		BIT STRING	(Public Key value; optionally may use compressed point representation)[*]	0x04:64:68:34:24:4e:1a:37:b2:f8:3a:d5:30:68:73: 8a:65:9b:e2:a6:d2:c6:f3:c9:3f:90:5e:8c:7d:a3: 59:79:27:6b:a7:fb:d4:30:83:42:a9:90:a7:c1:92: 90:98:20:7d:90:77:f2:97:f3:f5:3a:77:25:01:3c: 55:44:19:e8:4a
	Extensions.1	2.5.29.19 {basicConstraints}		N/A	
	Extensions.1. critical		BOOLEAN	TRUE	TRUE
	Extensions.1. extnValue.cA		BOOLEAN	TRUE	TRUE
	Extensions.1. extnValue. pathLenConstraint	this shall not be present	N/A	N/A	N/A

[*] A certificate verifier can unambiguously distinguish compressed from uncompressed point representations by evaluating the value of the first byte (0x02 and 0x03 for compressed points and 0x04 for an uncompressed point).

3.2.1.5 Additional attributes and extensions

Additional Certificate attributes and extensions defined in X.509 v3 are not allowed in a Manufacturer CA certificate or in a Product Unit CA certificate..

3.2.1.6 Constraints

QiPolicySize = 0x4 bytes

MaxQiRSIDSize = 0x9 bytes

MaxManufacturerCertSize = 0x200 bytes

MaxProdCertSize = 0x200 bytes

3.3 Certificate Chain slots

Certificate Chains reside in positions called slots. Each slot shall either be empty or contain one complete Certificate Chain. A Power Transmitter shall not contain more than 4 slots. Slots 0 and 1 shall only be used for Certificate Chains rooted with a WPC Root Certificate and shall not contain any other Certificate Chains. The manufacturer may use Slots 2 and 3 for any additional Certificate Chains as long as that manufacturer is identified by the Manufacturer Code in the Manufacturer Certificate of the Certificate Chain stored in slot 0.

Manufacturer-dependent Certificate Chains do not need to be structured as defined in Table 4. For example, they may be in an X.509v3 format with ASN.1 DER encoding, and/or they may be signed in a manufacturer-dependent manner. Slots 2 and 3 shall be populated in order, starting at slot 2. Manufacturer-dependent Certificate Chains that share a public/private key pair with a Certificate Chain in one or more of slots 0 and 1 shall be constructed to provide at least the same level of security to the public/private key pair as is required for the Certificate Chains stored in slots 0 and 1. Protection for manufacturer-dependent Certificate Chains and the formats and protocol they use shall match or exceed the protection for Certificate Chains, formats, and protocols defined in the *Qi Specification, Authentication Protocol* (this document) and stored in slots 0 and 1.

3.3.1 Provisioning

Provisioning is the process by which a Power Transmitter private key acquires one or more Certificate Chains. This procedure is outside the scope of the *Qi Specification*.

3.4 Power Transmitter private keys

Each Certificate Chain in a Power Transmitter corresponds to a Power Transmitter private key whose corresponding public key is certified in the Product Unit Certificate of the slot associated with the Certificate Chain. The Power Transmitter shall have access to that private key. The Power Transmitter shall store private keys in a manner that adequately protects the confidentiality of the key.

A manufacturer shall not use a private key associated with one Power Transmitter Product Unit in any other Power Transmitter Product Unit, Manufacturer Certificate, or Secondary Certificate.

3.5 Other private keys

The WPC Root Certificate Authority has a single public/private key pair. It shall generate and store its private key and use it for signing Manufacturer Certificates in a manner that adequately protects the confidentiality of the key.

Each Manufacturer Certificate and Secondary Certificate is associated with a unique public/ private key pair. The private key associated with a Manufacturer Certificate or Secondary Certificate is used for signing the next Certificate in a Certificate Chain. Manufacturers shall generate, provision, and store private keys in a manner that adequately protects the confidentiality of the key. See Section 2.4, *Impact to existing ecosystem* for detailed recommendations.

A Manufacturer shall not use a private key associated with one Manufacturer Certificate or Secondary Certificate in any other Manufacturer Certificate, Secondary Certificate, or Product Unit Certificate.

4 Authentication protocol

The Power Receiver can perform three operations:

- Query a Power Transmitter for Certificate Chain digests
- Read a Certificate Chain from a Power Transmitter
- Challenge a Power Transmitter in order to verify its authenticity

A Power Receiver may initiate as many or as few of these operations as are needed to achieve the desired Authentication latency. In addition, a Power Receiver may initiate the operations in any order. For example, a Power Receiver may first challenge a Power Transmitter, and then initiate a Certificate Chain read if the essential information from a target Certificate Chain is not already cached. Chapter 7, *Protocol flow examples*, provides some flow examples.

4.1 Digest query

To query a Power Transmitter for Certificate Chain digests, a Power Receiver sends a GET_DIGESTS request as defined in Section 5.2.1, *GET_DIGESTS*. If an error condition is encountered, the Power Transmitter shall respond with the appropriate ERROR response as defined in Section 5.3.4, *Error*. Otherwise, the Power Transmitter shall respond with a DIGESTS response as defined in Section 5.3.1, *DIGESTS*. After receiving a DIGESTS response, the Power Receiver can check to see if it has any of the Power Transmitter's Certificate Chains cached. This allows the Power Receiver to potentially skip reading a Certificate Chain and thus save time.

4.2 Certificate Chain read

To read a Certificate Chain, or portion thereof, a Power Receiver shall send a GET_CERTIFICATE request as defined in Section 5.2.2, *GET_CERTIFICATE*.

The Power Receiver may rely on the fact that valid Certificate Chains stored in slots 0 and 1 contain at least two Certificates (i.e. are at least 276 Bytes long). The Power Receiver can read and parse the first two Certificates in order to determine if the last Certificate in the Certificate Chain is not a Product Unit Certificate, and, if so, read the further Certificates one at a time.

If a Power Transmitter receives a GET_CERTIFICATE request that targets an offset that is outside the Certificate Chain (i.e. offset > length) or attempts to read beyond the length of the target Certificate Chain (i.e. (offset + length) > Certificate Chain length), then the Power Transmitter shall return an ERROR response of INVALID_REQUEST.

The Power Transmitter shall respond to error conditions with the appropriate ERROR response as defined in Section 5.3.4, *Error*. Otherwise, the Power Transmitter shall respond with a CERTIFICATE response as described in Section 5.3.2, *CERTIFICATE*.

4.3 Authentication challenge

To challenge a Power Transmitter, the Power Receiver sends a CHALLENGE request as defined in Section 5.2.3, *CHALLENGE*. If an error condition is encountered, the Power Transmitter shall respond with the appropriate ERROR response as defined in Section 5.3.4, *Error*. Otherwise, the Power Transmitter shall respond with a CHALLENGE_AUTH response as described in Section 5.3.3, *CHALLENGE_AUTH*.

NOTE: A Power Receiver that supports Authentication Protocol Versions greater than 1 should ensure that the version of the CHALLENGE request and the version of the response match. The purpose of this is to resist a downgrade attack.

4.4 Errors and alerts

4.4.1 Invalid request

If a Power Transmitter receives an Authentication request with one or more invalid fields, it shall respond to that Authentication request with an ERROR response of INVALID_REQUEST as described in Section 5.3.4, *Error*.

4.4.2 Unsupported protocol version

If a Power Transmitter receives an Authentication request that contains an unsupported Authentication Protocol Version in the Qi Authentication Protocol Version field, it shall respond to that Authentication request with an ERROR response of UNSUPPORTED_PROTOCOL that has the Authentication Protocol Version set to the maximum Authentication Protocol Version it supports and the Error Data field also set to the maximum Authentication Protocol Version it supports as described in Section 5.3.4, *Error*.

4.4.3 Busy

If a Power Transmitter receives an Authentication request but is unable to meet the timing requirements listed in Chapter 6, *Timing requirements*, it shall respond to that Authentication request with an ERROR response of BUSY as described in Section 5.3.4, *Error*, within the required response time.

4.4.4 Unspecified

If a Power Transmitter, upon receiving an Authentication request, encounters an error that is not covered by the conditions above, it shall respond to that Authentication request with an ERROR response of UNSPECIFIED as described in Section 5.3.4, *Error*.

5 Authentication messages

Authentication messages are used to convey information related to Authentication. Neither a Power Receiver nor a Power Transmitter shall add any padding after an Authentication message, i.e. the first byte of the Authentication message shall be the first byte of the data transport stream and the last byte of the Authentication message shall be the last byte of the data transport stream.

There are two types of Authentication messages: Authentication requests and Authentication responses. Authentication requests are defined in Section 5.2, *Authentication requests*. Authentication responses are defined in Section 5.3, *Authentication responses*.

5.1 Authentication message header

All Authentication messages start with a one-byte header, which shall have the format shown in Table 9. The header may be followed by a payload, as defined in Section 5.2, *Authentication requests*, and Section 5.3, *Authentication responses*.

Table 9: Authentication message header

	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
B_0	Authentication Protocol Version				Message Type			

Authentication Protocol Version This field identifies which version of the Authentication Specification is being used. Table 10 shows the valid values for this field. A Product shall not use an Authentication Protocol Version value corresponding to a specification revision that it does not support.

Table 10: Authentication Protocol Version 1.0

Name	Value	Meaning
Reserved	0x0	Reserved value
V1.0	0x1	Authentication Protocol Version 1.0
Reserved	0x2 - 0xF	Reserved value

The Power Receiver shall indicate the maximum protocol version that it supports in the Authentication Protocol Version field of the first Authentication request that it sends to the Power Transmitter on a new connection. If the Power Receiver receives an ERROR response of UNSUPPORTED_PROTOCOL, it shall use the version indicated in the Error Data field of the ERROR response in all subsequent Authentication requests. If a Power Transmitter receives an Authentication request that contains an Authentication Protocol Version in the Authentication Protocol Version field that is lower than the highest version it supports, it shall use the lower version in all Authentication responses.

In all cases, the format of the message shall match that specified for the version of the Authentication Protocol Version given in the Authentication Protocol Version field.

Message Type This field identifies Authentication Message type from one of the values in Table 11 (authentication requests) or Table 15 (authentication responses).

5.2 Authentication requests

Authentication requests are used by a Power Receiver to send a command to a Power Transmitter and/or to retrieve data. Authentication request types are listed in Table 11.

A Power Receiver shall not send another Authentication request until it has either received a response for or timed out the previously-sent Authentication request.

Table 11: Authentication requests

Value	Description
0x0 - 0x7	Used for Authentication responses
0x8	Reserved
0x9	GET_DIGESTS
0xA	GET_CERTIFICATE
0xB	CHALLENGE
0xC...0xF	Reserved

5.2.1 GET_DIGESTS

This request is used to retrieve Certificate Chain digests. The Power Receiver may request any number of digests at a time. The format for a GET_DIGESTS request is defined in Table 12.

Table 12: GET_DIGESTS request

	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
B_0	Authentication Message Header							
B_1	Reserved			Reserved	Slot Mask			

A Power Receiver shall set all reserved bits to ZERO. A Power Transmitter shall ignore all reserved bits.

Authentication Message Header This field identifies the authentication protocol version and message type, as defined in Section 5.1, *Authentication message header*. It shall be set to 0x19.

Slot Mask This field identifies the slots (see Section 3.3, *Certificate Chain slots*) in which the requested Certificate Chains are stored. b_0 is set to ONE to request the Certificate Chain stored in slot 0, b_1 is set to ONE to request the Certificate Chain stored in slot 1, etc.

NOTE: Multiple certificate chains can be requested with a single command.

5.2.2 GET_CERTIFICATE

This request is used to read a segment of a target Certificate Chain. The format for a GET_CERTIFICATE request is defined in Table 13.

Table 13: GET_CERTIFICATE request

	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
B_0	Authentication Message Header							
B_1	OffsetA8			LengthA8			Slot Number	
B_2	Offset70							
B_3	Length70							

A Power Receiver shall set all reserved bits to ZERO. A Power Transmitter shall ignore all reserved bits.

Authentication Message Header This field identifies the authentication protocol version and message type, as defined in Section 5.1, *Authentication message header*. It shall be set to 0x1A.

OffsetA8, Offset70 These two fields combine to form the offset in bytes from the start of the Certificate Chain to where the read request begins. The offset value is OffsetA8 × 256 + Offset70 and is denoted as GET_CERTIFICATE/Offset. Attempting to read beyond the end of a Certificate Chain shall result in the Power Transmitter returning an ERROR response of INVALID_REQUEST.

An offset value at or above 0x600 shall be adjusted by the PTx to return data relative to the first byte of the Product Unit Certificate (i.e. as if the Product Unit Certificate begins at 0x600).

An offset value in the range 0x600-0x7FF thus means the PTx sends bytes from the Certificate Chain at an offset of:

2 +N_RH + N_MC + GET_CERTIFICATE/Offset - 0x600

The above means that (in an example implementation) the offset value is adjusted as follows:

```
if (offset >= 0x600) offset = 2 +N_RH + N_MC + offset – 0x600;
```

LengthA8, Length70 These two fields combine to form the length in bytes of the read request. The length value is LengthA8 × 256 + Length70 and is denoted as GET_CERTIFICATE/Length. Attempting to read beyond the end of a Certificate Chain shall result in the Power Transmitter returning an ERROR response of INVALID_REQUEST.

A length value of 0x000 means the PTx shall “send as many remaining bytes as possible.”

The above means that (in an example implementation) the length value is adjusted as follows:

```
if (length == 0) length = 2 +N_RH + N_MC + N_PUC – offset;
```

Slot Number This is the slot number in which the target Certificate Chain is stored. The value in this field shall identify one of the Certificate Chains identified as being present in the DIGESTS response message or in the CHALLENGE_AUTH response message.

5.2.3 CHALLENGE

This request is used to initiate Authentication of a Power Transmitter Product Unit. The format for a CHALLENGE request is defined in Table 14.

Table 14: CHALLENGE request

	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
B_0	Authentication Message Header							
B_1	Reserved					Reserved	Slot Number	
B_2…B_{17}	Nonce							

A Power Receiver shall set all reserved bits to ZERO. A Power Transmitter shall ignore all reserved bits.

Authentication Message Header This field identifies the authentication protocol version and message type, as defined in Section 5.1, *Authentication message header*. It shall be set to 0x1B.

Slot Number This is the Slot number in which the Certificate Chain that will be used for Authentication is stored. The value in this field shall identify one of the Certificate Chains identified as being present in the DIGESTS response message or the CHALLENGE_AUTH response message.

NOTE: A Certificate Chain is always present in Slot 0.

Nonce This is a 128-bit binary Random Number chosen by the Power Receiver. The RNG used to generate the Nonce should meet the requirements of SP800-90A and SP800-90B (see Section 2.2.1, *Random number generators*).

5.3 Authentication responses

Power Transmitters use the response types listed in Table 15 to respond to Authentication requests.

Table 15: Authentication responses

Value	Description
0x0	Reserved
0x1	DIGESTS
0x2	CERTIFICATE
0x3	CHALLENGE_AUTH
0x4...0x6	Reserved
0x7	ERROR
0x8...0xF	Used for Authentication requests

5.3.1 DIGESTS

Power Transmitters use the DIGESTS response to send Certificate Chain digests and to report which slots contain valid Certificate Chain digests. The format for a DIGESTS response is defined in Table 16.

Table 16: DIGESTS response

	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
B_0	Authentication Message Header							
B_1	Slots Populated Mask				Slots Returned Mask			
$B_{2 + 32 \times (n - 1)} \cdots$ $B_{33 + 32 \times (n - 1)}$	Digests Returned							

Authentication Message Header This field identifies the authentication protocol version and message type, as defined in Section 5.1, *Authentication message header*. It shall be set to 0x11.

Slots Populated Mask The bit in position b_{K+4} of this field shall be set to ONE if and only if slot number K contains a Certificate Chain for the protocol version in the Authentication Protocol Version field.

NOTE: A Certificate Chain is always present for slot 0, and therefore bit b_4 in this mask is always set to ONE.

Slots Returned Mask The bit in position b_K of this field shall be set to ONE if and only if:

- slot number K contains a Certificate Chain for the protocol version in the Authentication Protocol Version field, and
- slot number K was set to 1 in the corresponding GET_DIGEST request.

If there are no such slots, then all bits in this mask shall be set to ZERO.

The number of digests returned shall be equal to the number of bits set in this byte. The digests shall be returned in order of increasing slot number.

Digests Returned The sequence of 32-byte SHA-256 digests of the Certificate Chain(s) for which the bit(s) in the Slots Returned Mask are non-zero, in order of increasing slot number. If all bits in Slots Returned are set to zero, then Digests Returned is not present. In the byte offset in Table 16, *n* represents the n^{th} digest returned.

5.3.2 CERTIFICATE

This response is used by a Power Transmitter to send the requested segment of a Certificate Chain. The format for a CERTIFICATE response is defined in Table 17.

Table 17: CERTIFICATE response

	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
B_0	Authentication Message Header							
B_1 … $B_{(1 + \text{length} - 1)}$	Certificate Chain Segment							

Authentication Message Header This field identifies the authentication protocol version and message type, as defined in Section 5.1, *Authentication message header*. It shall be set to 0x12.

Certificate Chain segment

The Certificate Chain segment shall contain a part of the requested segment (or the entire segment) of the requested Certificate Chain starting at the Offset requested in the corresponding GET_CERTIFICATE request. The length of this field shall be less than or equal to the length (Length98 × 256 + Length70) requested in the corresponding GET_CERTIFICATE request, and greater than or equal to 1.

5.3.3 CHALLENGE_AUTH

Power Transmitters use CHALLENGE_AUTH to respond to a CHALLENGE request. The format for a CHALLENGE_AUTH response is defined in Table 18.

Table 18: CHALLENGE_AUTH response

	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
B_0	Authentication Message Header							
B_1	Maximum Authentication Protocol Version				Slots Populated Mask			
B_2	Certificate Chain Hash LSB							
$B_3 \ldots B_{34}$	msb Signature *r* value lsb							
$B_{35} \ldots B_{66}$	msb Signature *s* value lsb							

Authentication Message Header This field identifies the authentication protocol version and message type, as defined in Section 5.1, *Authentication message header*. It shall be set to 0x13.

Maximum Authentication Protocol Version The maximum Authentication Protocol Version supported by the Power Transmitter.

Slots Populated Mask The bit in position b_K of this byte shall be set to ONE if and only if slot number K contains a Certificate Chain for the protocol version in the Authentication Protocol Version field.

A Certificate Chain is always present for slot 0, and therefore bit b_0 in this mask is always set to ONE.

Certificate Chain Hash LSB The LSB of the 32-byte SHA-256 hash of the Certificate Chain for which a challenge response is being provided.

Signature *r* value This field shall contain the 256-bit *r* value of a non-deterministic ECDSA digital signature, generated from the SHA-256 hash of TBSAuth as specified in Table 19. See ANSI X9.62-2005 for details.

Signature *s* value This field shall contain the 256-bit *s* value of a non-deterministic ECDSA digital signature, generated from the SHA-256 hash of TBSAuth as specified in Table 19. See ANSI X9.62-2005 for details.

Table 19: TBSAuth (for signature calculation)

	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
B_0	Prefix							
$B_1 \ldots B_{32}$	Certificate Chain Hash							
$B_{33} \ldots B_{50}$	Challenge Request							
$B_{51} \ldots B_{53}$	Copy of $B_0 \ldots B_2$ fields in the CHALLENGE_AUTH response in Table 18							

Prefix The Prefix field shall contain the ASCII representation of 'A', i.e. 0x41.

Certificate Chain Hash The Certificate Chain Hash field shall hold the SHA256 hash of the Certificate Chain for which a challenge response is being provided.

Challenge Request The Challenge Request field shall contain B0 … B17 of the CHALLENGE request message (see Table 14) to which the CHALLENGE_AUTH is a response.

5.3.4 Error

This response is used by a Power Transmitter to transmit error information. The format for an ERROR response is defined in Table 20.

Table 20: ERROR response

	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
B_0	Authentication Message Header							
B_1	Error Code							
B_2	Error Data							

Authentication Message Header This field identifies the authentication protocol version and message type, as defined in Section 5.1, *Authentication message header*. It shall be set to 0x17.

Error Code Defined in Table 21

Error Data Defined in Table 21

Table 21: ERROR Codes and Data.

Error Code	Value	Description	Error Data
Reserved	0x00	Reserved value	Reserved
INVALID_REQUEST	0x01	One or more request fields are invalid	0x00
UNSUPPORTED_ PROTOCOL	0x02	Requested protocol version is not supported	Maximum supported version
BUSY	0x03	Device cannot respond now but will be able to respond in the future	0x00
UNSPECIFIED	0x04	Unspecified error has occurred	0x00
Reserved	0x09...0xEF	Reserved value	Reserved
Manufacturer defined	0xF0...0xFF	Manufacturer defined (defined by the owner of the Manufacturer Code in the Manufacturer Certificate)	Manufacturer defined

Table 22 gives the recommended behavior of a Power Receiver when it receives an ERROR response to an Authentication request.

Table 22: Recommended ERROR Behavior

Error Code	Received in response to request	Recommended behavior
Reserved INVALID_REQUEST UNSPECIFIED	GET_DIGESTS	Treat as Authentication failure
	GET_CERTIFICATE	Treat as Authentication failure
	CHALLENGE	Treat as Authentication failure
UNSUPPORTED_PROTOCOL	GET_DIGESTS GET_CERTIFICATE CHALLENGE	Retry using protocol version not exceeding Maximum supported version as reported in Error Data; if already using a protocol version not exceeding Maximum supported version, then retry once, then treat as Authentication failure
BUSY	GET_DIGESTS	Retry after waiting t_{Retry}
	GET_CERTIFICATE	Retry after waiting t_{Retry}
	CHALLENGE	Retry after waiting t_{Retry}
Manufacturer defined	GET_DIGESTS GET_CERTIFICATE CHALLENGE	Manufacturer defined

6 Timing requirements

6.1 Power Receiver timing requirements

Table 23 shows the timeout values a Power Receiver should apply for Authentication request messages. For the data transport layer defined in the *Qi Specification, Communications Protocol*, the timings start at the begin of the final ADC/end data packet sent in the data transport stream carrying the Authentication request.

Table 23: Power Receiver wait time values

Parameter	Wait Time Value	Description
$t_{\text{DigestTimeout}}$	43 seconds	The timeout for a GET_DIGESTS Authentication request
$t_{\text{CertTimeout}}$	(N × 0.3 + 4) seconds, with a minimum of 5 seconds	The timeout for a GET_CERTIFICATE Authentication request, where *N* represents the number of bytes requested
$t_{\text{ChallengeAuthTimeout}}$	23 seconds	The timeout for a CHALLENGE Authentication request

Note: The timeout values are based on the following considerations.

- The Power Transmitter requires some time to prepare its response (see Section 6.2, *Power Transmitter timing requirements*) for the associated intervals
- The Power Transmitter uses ADT/1 data packets to send its response
- The Power Receiver sends at least one DSR data packet every 250 ms
- The Authentication response includes a 20% overhead for communications errors

If the Power Receiver does not receive an Authentication response within the timeout of the first Authentication request, it should retry that request up to five times. If a timeout still occurs after the last retry, the Power Receiver should follow its policy corresponding to the Power Transmitter not supporting Authentication functionality.

If the Power Receiver does not receive an Authentication response within the timeout of a subsequent Authentication request (i.e. not a retry of the first Authentication request), it should consider that as an error. In this case, it should not retry the subsequent Authentication request but immediately follow its policy corresponding to the Power Transmitter not supporting Authentication.

6.2 Power Transmitter timing requirements

A Power Transmitter shall meet the timing requirements defined in Table 24. For the data transport layer defined in the Qi Specification, Communications Protocol, the timings start at the beginning of the final ADC/end data packet sent in the data transport stream carrying the Authentication request.

Table 24: Power Transmitter Timing Requirement

Parameter	Wait Time Value	Description
$t_{\text{DigestReady}}$	3 seconds	The maximum time between sending a GET_DIGESTS Authentication request and being ready to start sending a DIGESTS Authentication response
$t_{\text{CertReady}}$	3 seconds	The maximum time between sending a GET_CERTIFICATE Authentication request and being ready to start sending a CERTIFICATE Authentication response
$t_{\text{ChallengeAuthReady}}$	3 seconds	The maximum time between sending a CHALLENGE Authentication request and being ready to start sending a CHALLENGE_AUTH Authentication response

A Power Receiver should not attempt to retrieve an Authentication response by sending a DSR/poll data packet before the corresponding timing value has expired. Doing so can cause the Power transmitter to send a BUSY Authentication response. The time it takes to send the latter delays retrieval of the expected Authentication response.

NOTE: A Power Transmitter fails a timing requirement if it cannot complete sending its Authentication response within the associated Power Receiver timeout value defined in Section 6.1, *Power Receiver timing requirements*, provided the Power Receiver sends at least one DSR data packet every 250 ms.

7 Protocol flow examples

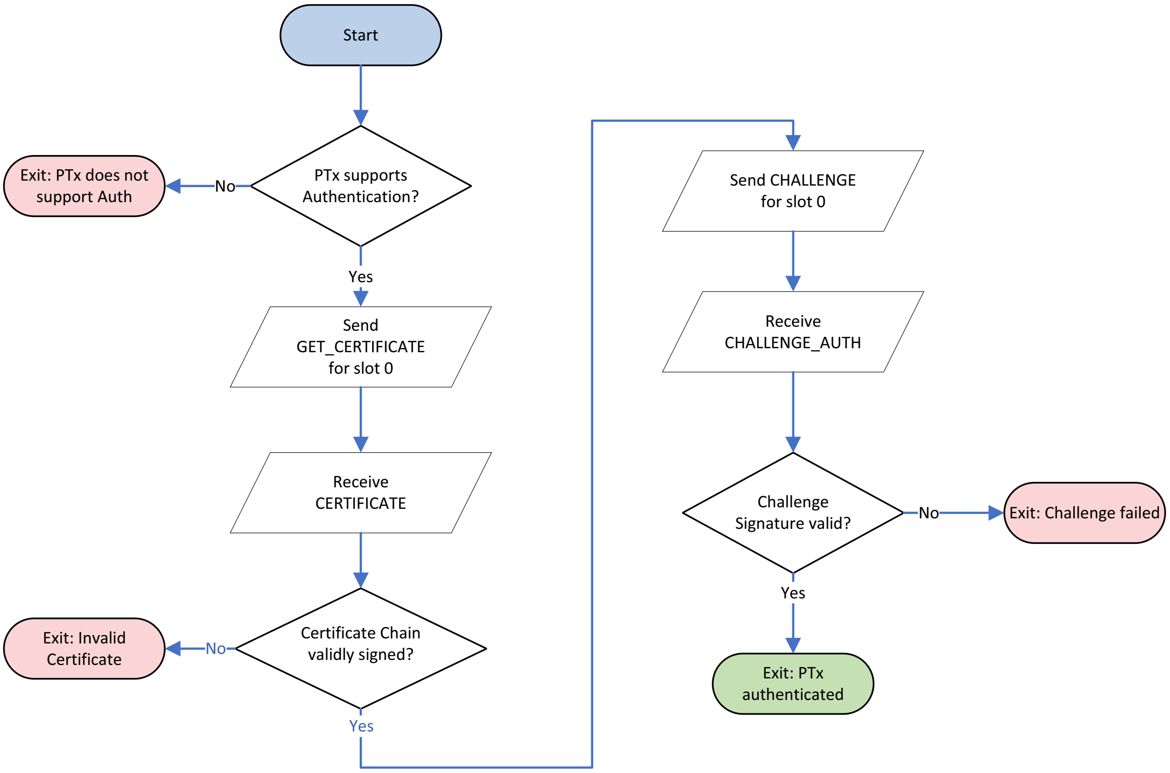
This chapter provides informative examples of suitable high level flows. It is not intended to be exclusive; other flows compliant with this specification are possible. These flows omit details such as behavior on timeouts while waiting for an Authentication response and the handling of ERROR responses. These examples also assume that the Power Receiver only processes the Certificate Chain in the Power Transmitter's slot 0. If a manufacturer intends for the Power Receiver to process Certificate Chains in other slots, the manufacturer should adapt these flows accordingly.

7.1 Simple flow

Figure 2 illustrates the behavior of a Power Receiver that neither caches the digests and public keys of previously authenticated Power Transmitters nor supports Revocation.

NOTE: This Power Receiver does not send a GET DIGESTS Authentication request because it does not cache digests.

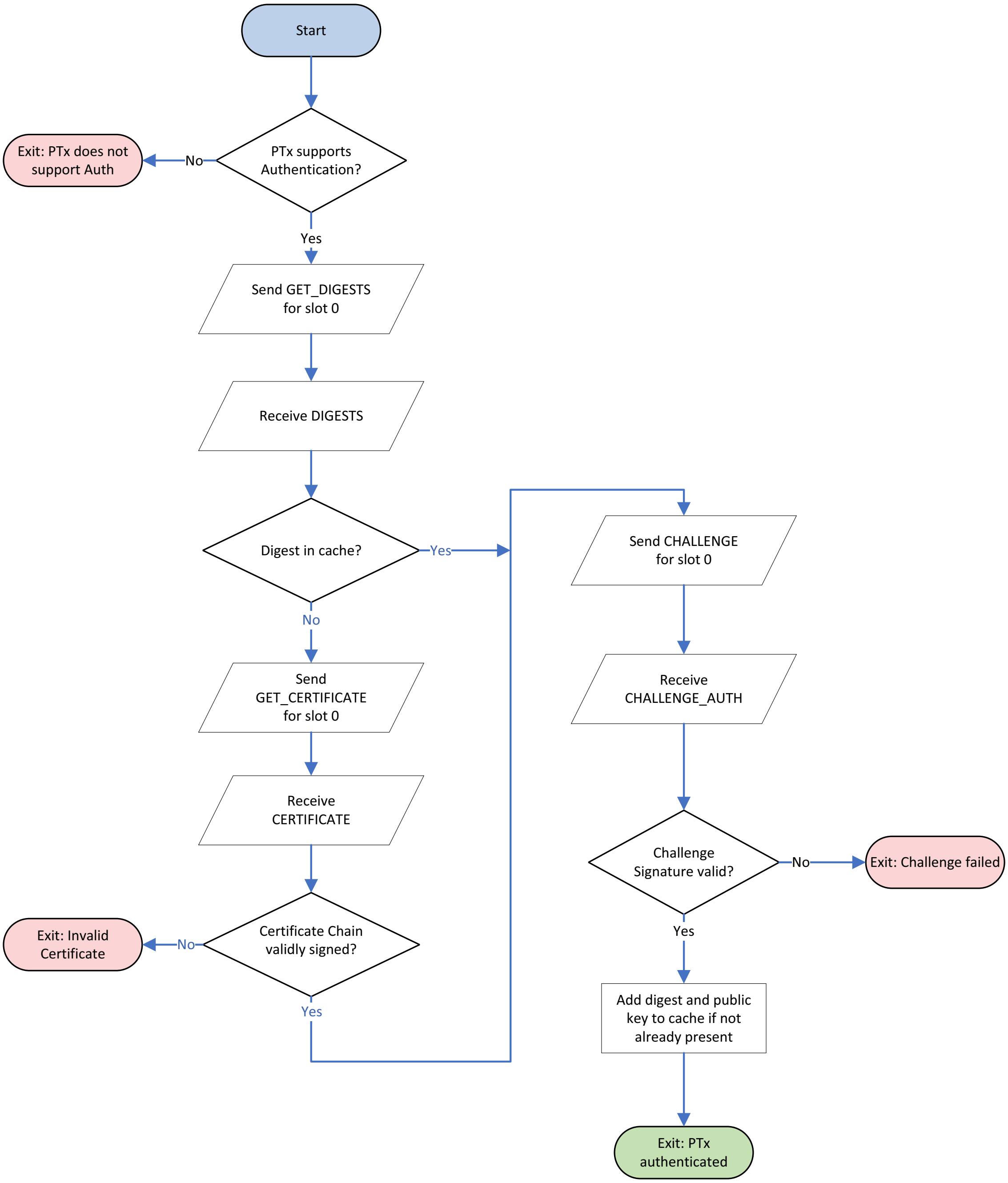
Figure 2. Simple Power Receiver


Start
PTx supports Authentication?
No
Exit: PTx does not support Auth
Yes
Send GET_CERTIFICATE for slot 0
Receive CERTIFICATE
Certificate Chain validly signed?
No
Exit: Invalid Certificate
Yes
Send CHALLENGE for slot 0
Receive CHALLENGE_AUTH
Challenge Signature valid?
No
Exit: Challenge failed
Yes
Exit: PTx authenticated

7.2 Flow with caching

Figure 3 illustrates the behavior of a Power Receiver that caches (in non-volatile memory) the digests and public keys of Power Transmitters that it has previously trusted. This allows the Power Receiver to skip sending the GET CERTIFICATES Authentication request and proceed directly to issuing the CHALLENGE Authentication request. If the Certificate Chain has been copied and deployed in a counterfeit Power Transmitter, the challenge will fail even though a match will be found in the cache.

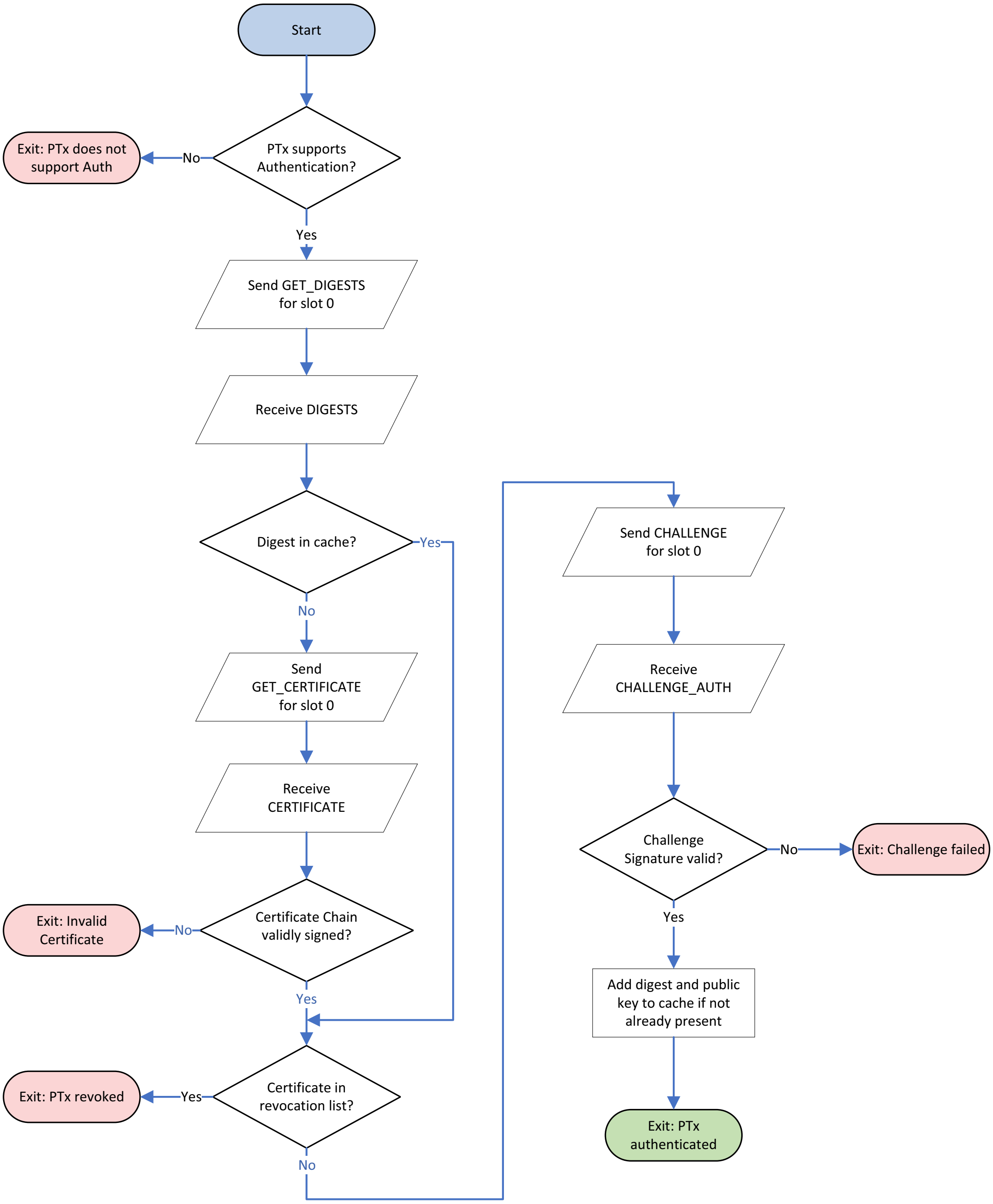
Figure 3. Power Receiver with caching


Start
PTx supports Authentication?
No
Exit: PTx does not support Auth
Yes
Send GET_DIGESTS for slot 0
Receive DIGESTS
Digest in cache?
Yes
No
Send GET_CERTIFICATE for slot 0
Receive CERTIFICATE
Certificate Chain validly signed?
No
Exit: Invalid Certificate
Yes
Send CHALLENGE for slot 0
Receive CHALLENGE_AUTH
Challenge Signature valid?
No
Exit: Challenge failed
Yes
Add digest and public key to cache if not already present
Exit: PTx authenticated

7.3 Flow with caching and revocation

Figure 4 illustrates the behavior of a Power Receiver that caches (in non-volatile memory) the digests and public keys of Power Transmitters that it has previously trusted and that support Revocation. This information typically includes the Manufacturer Code, the group code (if the Certificate Chain includes a Secondary Certificate), the product type, and the Product Unit Serial Number.

Figure 4. Power Receiver with caching and revocation


Start
PTx supports Authentication?
No
Exit: PTx does not support Auth
Yes
Send GET_DIGESTS for slot 0
Receive DIGESTS
Digest in cache?
Yes
No
Send GET_CERTIFICATE for slot 0
Receive CERTIFICATE
Certificate Chain validly signed?
No
Exit: Invalid Certificate
Yes
Certificate in revocation list?
Yes
Exit: PTx revoked
No
Send CHALLENGE for slot 0
Receive CHALLENGE_AUTH
Challenge Signature valid?
No
Exit: Challenge failed
Yes
Add digest and public key to cache if not already present
Exit: PTx authenticated

7.4 Challenge first flow

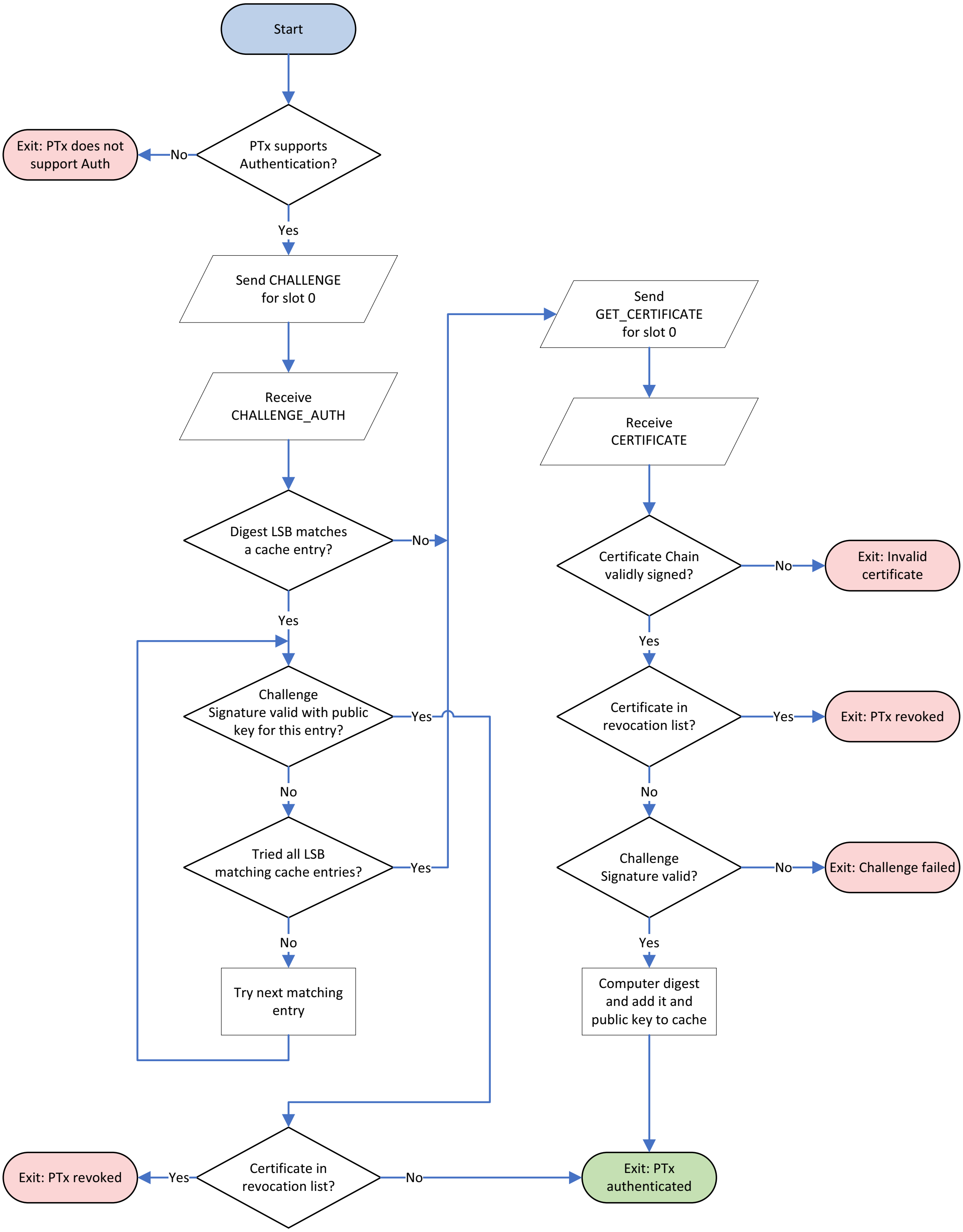
Figure 5 illustrates an alternative behavior of a Power Receiver that caches (in non-volatile memory) the digests and public keys of Power Transmitters that it has previously trusted and that support Revocation. This behavior shares the assumptions made in Section 7.3, *Flow with caching and revocation*, on the management of the cache and Revocation Lists.

This behavior starts when the Power Receiver issues a CHALLENGE Authentication request. The CHALLENGE AUTH response contains the LSB of the hash of the Certificate Chain. The Power Receiver examines its cache of digests to try to find a digest or digests that have a matching LSB. If there are no such digests, the Power Receiver proceeds on the basis that this is a first encounter with the particular Power Transmitter. If there is a match, the Power Receiver attempts to validate the CHALLENGE AUTH signature using the public key corresponding to the matched digest. If this validation succeeds, the Power Receiver checks that the Certificate has not been revoked. If the Certificate has not been revoked, the Power Receiver determines that the responder can be trusted. If this validation fails, then the Power Receiver retries with the next digest that matches the hash LSB returned in CHALLENGE AUTH response. If all such validations fail, then the Power Receiver proceeds on the assumption that this is the first encounter with the particular Power Transmitter.

When the Power Receiver determines that this is a first encounter with the particular Power Transmitter, it retrieves the Certificate Chain, validates it, checks that it has not been revoked, and then attempts to validate the previously returned CHALLENGE AUTH signature with the public key contained in the Product Unit Certificate.

The advantage of this method is that after the first encounter between the Power Receiver and the particular Power Transmitter, only one Authentication request and response communication takes place between the two devices.

Figure 5. Challenge first Power Receiver


Start
PTx supports Authentication?
No
Exit: PTx does not support Auth
Yes
Send CHALLENGE for slot 0
Receive CHALLENGE_AUTH
Digest LSB matches a cache entry?
No
Yes
Challenge Signature valid with public key for this entry?
Yes
No
Tried all LSB matching cache entries?
Yes
No
Try next matching entry
Send GET_CERTIFICATE for slot 0
Receive CERTIFICATE
Certificate Chain validly signed?
No
Exit: Invalid certificate
Yes
Certificate in revocation list?
Yes
Exit: PTx revoked
No
Challenge Signature valid?
No
Exit: Challenge failed
Yes
Computer digest and add it and public key to cache
Certificate in revocation list?
Yes
Exit: PTx revoked
No
Exit: PTx authenticated