

DIN EN 62680-1-4



ICS 29.220.01; 33.120.01; 35.200

Einsprüche bis 2018-08-29

Entwurf

**Schnittstellen des Universellen Seriellen Busses für Daten und Energie –
Teil 1-4: Gemeinsame Bauteile –
Festlegung für USB-Typ-C™-Authentifizierungs-Spezifikation
(IEC 100/2981/CDV:2017);
Deutsche und Englische Fassung prEN 62680-1-4:2017**

Universal Serial Bus interfaces for data and power –
Part 1-4: Common Components –
USB Type-C™ Authentication Specification
(IEC 100/2981/CDV:2017);
German and English version prEN 62680-1-4:2017

Anwendungswarnvermerk

Dieser Norm-Entwurf mit Erscheinungsdatum 2018-06-29 wird der Öffentlichkeit zur Prüfung und Stellungnahme vorgelegt.

Weil die beabsichtigte Norm von der vorliegenden Fassung abweichen kann, ist die Anwendung dieses Entwurfs besonders zu vereinbaren.

Stellungnahmen werden erbeten

- vorzugsweise online im Norm-Entwurfs-Portal von DIN unter www.din.de/go/entwuerfe bzw. für Norm-Entwürfe der DKE auch im Norm-Entwurfs-Portal der DKE unter www.entwuerfe.normenbibliothek.de, sofern dort wiedergegeben;
- oder als Datei per E-Mail an dke@vde.com möglichst in Form einer Tabelle. Die Vorlage dieser Tabelle kann im Internet unter www.din.de/go/stellungnahmen-norm-entwuerfe oder für Stellungnahmen zu Norm-Entwürfen der DKE unter www.dke.de/stellungnahme abgerufen werden;
- oder in Papierform an die DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE, Stresemannallee 15, 60596 Frankfurt am Main.

Die Empfänger dieses Norm-Entwurfs werden gebeten, mit ihren Kommentaren jegliche relevanten Patentrechte, die sie kennen, mitzuteilen und unterstützende Dokumentationen zur Verfügung zu stellen.

Gesamtumfang 123 Seiten

DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE

Nationales Vorwort

Die Deutsche Fassung des europäischen Dokuments prEN 62680-1-4:2017 „Schnittstellen des Universellen Seriellen Busses für Daten und Energie – Teil 1-4: Gemeinsame Bauteile – Festlegung für USB-Typ-C™-Authentifizierungs-Spezifikation“ (Entwurf in der Umfrage) ist unverändert in diesen Norm-Entwurf übernommen worden.

Die Internationale Elektrotechnische Kommission (IEC) und das Europäische Komitee für Elektrotechnische Normung (CENELEC) haben vereinbart, dass ein auf IEC-Ebene erarbeiteter Entwurf für eine Internationale Norm zeitgleich (parallel) bei IEC und CENELEC zur Umfrage (CDV-Stadium) und Abstimmung als FDIS (en: Final Draft International Standard) bzw. Schluss-Entwurf für eine Europäische Norm gestellt wird, um eine Beschleunigung und Straffung der Normungsarbeit zu erreichen. Dem entsprechend ist das internationale Dokument IEC 100/2981/CDV:2017 „Universal Serial Bus interfaces for data and power – Part 1-4: Common Components – USB Type-C™ Authentication Specification“ unverändert in den Entwurf prEN 62680-1-4:2017 übernommen worden.

Da die Deutsche Fassung noch nicht endgültig mit der Englischen und Französischen Fassung abgeglichen ist, ist die englische Originalfassung des IEC-CDV entsprechend der diesbezüglich durch die IEC erteilten Erlaubnis beigefügt. Die Nutzungsbedingungen für den deutschen Text des Norm-Entwurfes gelten gleichermaßen auch für den englischen IEC-Text.

Das internationale Dokument wurde vom TC 100 „Audio, video and multimedia systems and equipment“ der Internationalen Elektrotechnischen Kommission (IEC) erarbeitet.

Bei der Abstimmung zu dem Europäischen Schluss-Entwurf bei CENELEC und dem Internationalen Schluss-Entwurf bei IEC [Final Draft International Standard (FDIS)] sind jeweils nur „JA/NEIN“-Entscheidungen möglich, wobei „NEIN“-Entscheidungen fundiert begründet werden müssen. Dokumente, die bei CENELEC als Europäische Norm angenommen und ratifiziert werden, sind unverändert als Deutsche Normen zu übernehmen.

Für dieses Dokument ist das nationale Arbeitsgremium K 742 „Audio-, Video- und Multimediasysteme, -geräte und -komponenten“ der DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE (www.dke.de) zuständig.

Für den Fall einer undatierten Verweisung im normativen Text (Verweisung auf ein Dokument ohne Angabe des Ausgabedatums und ohne Hinweis auf eine Abschnittsnummer, eine Tabelle, ein Bild usw.) bezieht sich die Verweisung auf die jeweils aktuellste Ausgabe des in Bezug genommenen Dokuments.

Für den Fall einer datierten Verweisung im normativen Text bezieht sich die Verweisung immer auf die in Bezug genommene Ausgabe des Dokuments.

Der Zusammenhang der zitierten Dokumente mit den entsprechenden deutschen Dokumenten ergibt sich, soweit ein Zusammenhang besteht, grundsätzlich über die Nummer der entsprechenden IEC-Publikation. Beispiel: IEC 60068 ist als EN 60068 als Europäische Norm durch CENELEC übernommen und als DIN EN 60068 ins Deutsche Normenwerk aufgenommen.

Es wird auf die Möglichkeit hingewiesen, dass einige Elemente dieses Dokuments Patentrechte berühren können. IEC ist nicht dafür verantwortlich, einige oder alle diesbezüglichen Patentrechte zu identifizieren.

Das Original-Dokument enthält Bilder in Farbe, die in der Papierversion in einer Graustufen-Darstellung wiedergegeben werden. Elektronische Versionen dieses Dokuments enthalten die Bilder in der originalen Farbdarstellung.

**Schnittstellen des Universellen Seriellen Busses für Daten und Energie –
Teil 1-4: Gemeinsame Bauteile –
Festlegung für USB-Typ-C™-Authentifizierungs-Spezifikation**

	Inhalt	Seite
1	Einleitung.....	6
1.1	Anwendungsbereich.....	6
1.2	Überblick	6
1.3	Verwandte Dokumente.....	6
1.4	Begriffe und Abkürzungen.....	8
1.5	Konventionen	9
1.5.1	Prioritäten	9
1.5.2	Schlüsselwörter	9
1.5.3	Zahlen	11
1.5.4	Byte-Reihenfolge.....	11
2	Überblick	11
2.1	Topologie.....	11
2.2	Verschlüsselungsverfahren	12
2.2.1	Zufallszahlen	13
2.3	Überblick über die Sicherheit	13
2.3.1	Periodische Wiederholung der Authentifizierung	13
2.3.2	Speicherung und Schutz geheimer Schlüssel	13
2.3.3	Kriterien für die Sicherheitsbewertung	13
2.4	Auswirkungen auf das bestehende Ökosystem.....	14
2.4.1	Proxy-Fähigkeiten (die Hub-Topologie durchziehende PD)	14
3	Authentifizierungs-Architektur	14
3.1	Zertifikate.....	14
3.1.1	Format.....	14
3.1.2	Textformat	14
3.1.3	Attribute und Erweiterungen.....	14
3.2	Zertifikatsketten	16
3.2.1	Bereitstellung	17
3.3	Private Schlüssel.....	17
4	Authentifizierungs-Protokoll	17
4.1	Digest-Abfrage	17
4.2	Lesen einer Zertifikatskette	17
4.3	Authentifizierungs-Challenge	18
4.4	Fehler und Warnungen	18
4.4.1	Invalid Request	18

	Seite
4.4.2 Unsupported Protocol Version	18
4.4.3 Busy.....	18
4.4.4 Unspecified.....	18
5 Authentifizierungsnachrichten	18
5.1 Kopf	19
5.1.1 USB-Typ-C-Authentifizierungsprotokollversion	19
5.1.2 Nachrichtentyp.....	19
5.1.3 Param1	19
5.1.4 Param2	19
5.2 Authentifizierungsanforderungen	19
5.2.1 GET_DIGESTS	20
5.2.2 GET_CERTIFICATE	20
5.2.3 CHALLENGE.....	21
5.3 Authentifizierungsantworten	21
5.3.1 DIGESTS	22
5.3.2 CERTIFICATE	22
5.3.3 CHALLENGE_AUTH.....	23
5.3.4 ERROR.....	24
6 Authentifizierung von PD-Produkten	25
6.1 Übertragungen mit einer Länge kleiner oder gleich <i>MaxExtendedMsgLen</i>	25
6.2 Übertragungen mit einer Länge größer als <i>MaxExtendedMsgLen</i>	25
6.3 Zeitvorgaben für erweiterte PD-Sicherheitsnachrichten.....	30
6.3.1 Authentifizierungs-Initiator	30
6.3.2 Authentifizierungs-Beantworter	31
6.4 Context Hash.....	32
7 Authentifizierung von USB-Produkten	32
7.1 Deskriptoren	32
7.1.1 Deskriptor für Authentifizierungsfähigkeiten.....	32
7.2 Zuordnung von Authentifizierungsnachrichten in USB.....	33
7.2.1 Authentication IN	33
7.2.2 Authentication OUT	34
7.3 Authentifizierungs-Protokoll.....	34
7.3.1 Digest-Abfrage.....	35
7.3.2 Lesen eines Zertifikats	35
7.3.3 Authentifizierungs-Challenge	35
7.3.4 Fehler	36
7.4 Zeitvorgaben für USB	36
7.4.1 Zeitvorgaben für USB-Hosts	36

	Seite
7.4.2 Zeitvorgaben für USB-Geräte	37
7.5 Context Hash.....	38
8 Protokollkonstanten.....	38
A ACD	38
A.1 ACD-Formatierung	38
A.1.1 Versions-TLV.....	39
A.1.2 XID-TLV.....	40
A.1.3 Stromquellenfähigkeiten-TLV	40
A.1.4 Stromquellenzertifizierungs-TLV	41
A.1.5 Kabelfähigkeiten-TLV	41
A.1.6 Sicherheitsbeschreibungs-TLV	42
A.1.6.1 FIPS-/ISO-Bezeichner.....	42
A.1.6.2 Common-Criteria-Bezeichner.....	43
A.1.6.3 Sicherheitsanalyse-Bezeichner.....	45
A.1.7 Playpen-TLV.....	46
A.1.8 Anbietererweiterungs-TLV	47
A.1.9 Erweiterungs-TLV.....	47
A.2 ACD eines PD-Produkts.....	47
A.3 ACD eines USB-Produkts	48
B Verschlüsselungsbeispiele.....	49
B.1 Beispiel für eine Authentifizierungssequenz	49
B.2 Beispiel einer Zertifikatsketten-Topologie	50
B.2.1 Zertifikatskette	50
B.2.1.1 Zwischenzertifikat.....	51
B.2.1.2 Leaf-Zertifikat	52
B.2.2 Root-Zertifikat.....	54
B.2.3 Schlüsselpaare.....	55
B.2.3.1 Root-Schlüsselpaar	55
B.2.3.2 Zwischenschlüsselpaar	55
B.2.3.3 Leaf-Schlüsselpaar	56
B.3 Beispiel für die Überprüfung einer Authentifizierungssignatur.....	56
B.3.1 CHALLENGE-Anforderung	56
B.3.2 CHALLENGE_AUTH-Antwort	56
C Potenzielle Angriffsvektoren.....	57
Bilder	
Bild 2-1 – Beispieltopologie	12
Bild 6-1 – Beispiel für einen Sicherheitsübertragungsvorgangs vonseiten eines Authentifizierungs- Initiators.....	28

	Seite
Bild 6-2 – Beispiel für einen Sicherheitsübertragungsvorgangs vonseiten eines Authentifizierungs-Beantworters	29
Bild 6-3 – Beispiel für das Lesen einer 612-Byte-Zertifikatskette	30
Bild A-1 – Bitmap der Versions-TLV-Daten	39
Bild A-2 – Bitmap der Common-Criteria-Bezeichner	43
Bild A-3 – Bitmap der Sicherheitsanalyse-Bezeichner	45
Tabellen	
Tabelle 1-1 – Begriffe und Abkürzungen	8
Tabelle 2-1 – Zusammenfassung der Verschlüsselungsverfahren	12
Tabelle 3-1 – Format einer Zertifikatskette	16
Tabelle 5-1 – Authentifizierungsnachrichtenkopf	19
Tabelle 5-2 – USB-Typ-C-Authentifizierungsprotokollversion	19
Tabelle 5-3 – Arten von Authentifizierungsanforderungen	20
Tabelle 5-4 – Kopf einer GET_DIGESTS-Anforderung	20
Tabelle 5-5 – Kopf einer GET_CERTIFICATE-Anforderung	20
Tabelle 5-6 – Nutzdatum einer GET_CERTIFICATE-Anforderung	21
Tabelle 5-7 – Kopf einer CHALLENGE-Anforderung	21
Tabelle 5-8 – Nutzdatum einer CHALLENGE-Anforderung	21
Tabelle 5-9 – Arten von Authentifizierungsantworten	21
Tabelle 5-10 – Kopf einer DIGESTS-Antwort	22
Tabelle 5-11 – Nutzdatum einer DIGESTS-Antwort	22
Tabelle 5-12 – Kopf einer CERTIFICATE-Antwort	22
Tabelle 5-13 – Nutzdatum einer CERTIFICATE-Antwort	23
Tabelle 5-14 – Kopf einer CHALLENGE_AUTH-Antwort	23
Tabelle 5-15 – Nutzdatum einer CHALLENGE_AUTH-Antwort	23
Tabelle 5-16 – Nachrichteninhalt für digitale ECDSA-Signaturen	24
Tabelle 5-17 – Kopf einer ERROR-Antwort	24
Tabelle 5-18 – ERROR-Codes	25
Tabelle 6-1 – Timeout-Werte für einen PD-Authentifizierungs-Initiator	31
Tabelle 6-2 – Zeitvorgaben für PD-Authentifizierungs-Beantworter	31
Tabelle 7-1 – Deskriptor für Authentifizierungsfähigkeiten	33
Tabelle 7-2 – Deskriptortypen für Authentifizierungsfähigkeiten	33
Tabelle 7-3 – bRequest-Werte in Authentifizierungsnachrichten	33
Tabelle 7-4 – Felder einer Authentication-IN-Steuerungsanforderung	34
Tabelle 7-5 – Zuordnung des Authentifizierungsnachrichtenkopfes	34
Tabelle 7-6 – Felder einer Authentication-OUT-Steuerungsanforderung	34
Tabelle 7-7 – Felder einer GET_DIGESTS-Authentication-IN-Steuerungsanforderung	35
Tabelle 7-8 – Felder einer GET_CERTIFICATE-Authentication-OUT-Steuerungsanforderung	35
Tabelle 7-9 – Felder einer CERTIFICATE-Authentication-IN-Steuerungsanforderung	35

	Seite
Tabelle 7-10 – Felder einer CHALLENGE-Authentication-OUT-Steuerungsanforderung	36
Tabelle 7-11 – Felder einer CHALLENGE_AUTH-Authentication-IN-Steuerungsanforderung	36
Tabelle 7-12 – Timeout-Werte für Authentifizierungs-Initiatoren.....	36
Tabelle 7-13 – Antwortzeiten eines Authentifizierungs-Beantworters.....	37
Tabelle 8-1 – Protokollkonstanten	38
Tabelle A-1 – Allgemeines TLV-Format	39
Tabelle A-2 – TLV-Typen	39
Tabelle A-3 – Felder des Versions-TLV	39
Tabelle A-4 – Codierung der ACD-Version	40
Tabelle A-5 – Felder des XID-TLV	40
Tabelle A-6 – Felder des Stromquellenfähigkeiten-TLV	40
Tabelle A-7 – Daten des Stromquellenfähigkeiten-TLV	40
Tabelle A-8 – Felder des Stromquellenzertifizierungs-TLV.....	41
Tabelle A-9 – Felder des Kabelfähigkeiten-TLV.....	41
Tabelle A-10 – Daten des Kabelfähigkeiten-TLV	42
Tabelle A-11 – Felder des Sicherheitsbeschreibungs-TLV.....	42
Tabelle A-12 – Sicherheitsdaten	42
Tabelle A-13 – Bezeichner der FIPS-/ISO-Stufe.....	42
Tabelle A-14 – Gefährdungsbewertung	43
Tabelle A-15 – EAL-Codierung.....	44
Tabelle A-16 – Schutzprofil-Codierung	44
Tabelle A-17 – Sicherheit bei der Entwicklung.....	45
Tabelle A-18 – Certification Maintenance	45
Tabelle A-19 – Codierung des Prüfverfahrens	46
Tabelle A-20 – Gefährdungsbewertung	46
Tabelle A-21 – Felder des Playpen-TLV	46
Tabelle A-22 – Felder des Anbietererweiterungs-TLV	47
Tabelle A-23 – Daten des Anbietererweiterungs-TLV.....	47
Tabelle A-24 – Felder des Erweiterungs-TLV	47
Tabelle A-25 – ACD-TLVs für PD-Produkte (1 von 2).....	47
Tabelle A-26 – ACD-TLVs für USB-Produkte.....	48
Tabelle B-1 – Felder des Versions-TLV	53
Tabelle B-2 – Felder des XID-TLV	53
Tabelle B-3 – Felder des Stromquellenfähigkeiten-TLV	53
Tabelle B-4 – Felder des Sicherheitsbeschreibungs-TLV.....	54
Tabelle B-5 – Felder des Playpen-TLV	54
Tabelle B-6 – Felder des Anbietererweiterungs-TLV	54

1 Einleitung

Diese Spezifikation beschreibt eine Möglichkeit zur Authentifizierung von Produkten hinsichtlich Identifikation und Konfiguration. Die Authentifizierung findet mittels USB-Power-Delivery-Nachrichtenkommunikation und/oder mittels USB-Datenbus-Kontrolltransaktionen statt.

Die USB-Typ-C™-Authentifizierung ermöglicht es Organisationen, Richtlinien für zulässige Produkte festzulegen und umzusetzen. Dadurch wird die Sicherheit in realen Situationen zweckdienlich sichergestellt. Zum Beispiel:

- Ein Anbieter, der seine Geräte vor Produktschäden durch minderwertige Ladegeräte schützen möchte, kann eine Richtlinie festlegen, nach der das Laden nur mit zertifizierten PD-Produkten möglich ist.
- Ein Anwender, der Bedenken bezüglich des Ladens seines Telefons an öffentlichen Ladestationen hat, kann eine Richtlinie festlegen, nach der das Laden des Telefons nur mit zertifizierten PD-Produkten möglich ist.
- Eine Organisation, die verhindern möchte, dass nicht identifizierbare Speichergeräte auf die PC-Infrastruktur des Unternehmens zugreifen können, können in ihren PCs eine Richtlinie festlegen, nach der nur verifizierte und von der internen IT-Abteilung signierte USB-Speichergeräte verwendet werden können.

1.1 Anwendungsbereich

Diese Spezifikation legt die Architektur und Methodologie für die einseitige Produktauthentifizierung fest. Bei der Erarbeitung wurde eine uneingeschränkte Kompatibilität mit und Erweiterung der bestehenden PD- und USB-Infrastruktur angestrebt. Die Spezifikation enthält Informationen zur Ermöglichung der Anwendung von Richtlinien; einzelne Richtlinienentscheidungen werden jedoch nicht festgelegt.

Die Authentifizierung von USB-Typ-C-Produkten, die Alternativmodi unterstützen, ist erlaubt. Die entsprechenden Verfahren liegen jedoch außerhalb des Anwendungsbereichs dieser Spezifikation.

1.2 Überblick

Diese Spezifikation enthält Primitive für die einseitige Authentifizierung. Das in dieser Spezifikation festgelegte Sicherheitsmodell ermöglicht die Sicherstellung, dass ein Produkt:

- von einer bestimmten Art von einem bestimmten Hersteller ist und bestimmte Eigenschaften aufweist;
- sich im Besitz und unter der Kontrolle einer bestimmten Organisation befindet.

Die lokale Richtlinie legt fest, welche Eigenschaften ein verbundenes Produkt aufweisen muss, bevor es auf eine Ressource zugreifen oder eine Ressource zur Verfügung stellen kann (z. B. Energie, Speicher usw.).

Produktanbieter können Sicherheitsmerkmale hinzufügen, die über die in dieser Spezifikation aufgeführten Merkmale hinausgehen. Die Festlegung und Implementierung dieser Merkmale liegt jedoch in der Verantwortung des Anbieters. Zusätzliche Merkmale dürfen die in diesem Dokument beschriebenen grundlegenden Festlegungen nicht verändern.

1.3 Verwandte Dokumente

- **USB 2.0** – Universal-Serial-Bus-Spezifikation, Revision 2.0, (einschließlich Errata und ECNs vom 11. August 2014) (im vorliegenden Dokument als USB-2.0-Spezifikation bezeichnet) (verfügbar unter: <http://www.usb.org/developers/docs>).
- **USB 3.1** – Universal-Serial-Bus-3.1-Spezifikation, Revision 1.0, (einschließlich Errata und ECNs vom 11. August 2014) (im vorliegenden Dokument als USB-3.1-Spezifikation bezeichnet) (verfügbar unter: <http://www.usb.org/developers/docs>).

- **USBDP** – Universal-Serial-Bus-Power-Delivery-Spezifikation, Revision 3, Version 1.0a, 25. März 2016 (in diesem Dokument als USB-PD-Spezifikation bezeichnet) (verfügbar unter: <http://www.usb.org/developers/docs>).
- **USBTYPEC** – Spezifikation für Universal-Serial-Bus-Typ-CTM-Kabel und Steckverbinder, Revision 1.2, 25. März 2016 (in diesem Dokument als USB-Typ-C-Spezifikation bezeichnet) (verfügbar unter: <http://www.usb.org/developers/docs>).
- **USBTYPEC BRIDGE** – Universal-Serial-Bus-Typ-C-Brücken-Spezifikation, Revision 1.0, 25. März 2016, (verfügbar unter <http://www.usb.org/developers/docs>).
- **ASN.1** – ISO-822-1-4;
 - ITU-T X.680 (verfügbar unter: https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-X.680-201508-!PDF-E&type=items);
 - ITU-T X.681 (verfügbar unter: https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-X.681-201508-!PDF-E&type=items);
 - ITU-T X.682 (verfügbar unter: https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-X.682-201508-!PDF-E&type=items);
 - ITU-T X.683 (verfügbar unter: https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-X.683-201508-!PDF-E&type=items).
- **DER** – ISO-8825-1; ITU-T X.690 (verfügbar unter: https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-X.690-201508-!PDF-E&type=items).
- **X509v3** - ISO-9594-8; ITU-T X.509 (verfügbar unter: https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-X.509-201210-!PDF-E&type=items).
- **Common Criteria**:
 - Common Criteria for Information Technology Security Evaluation, Teile 1-3, Version 3.1, Revision 4, September 2010 (verfügbar unter: <https://www.commoncriteriaportal.org/cc/#supporting>);
 - ISO/IEC 15408 Evaluationskriterien für IT-Sicherheit, Teile 1-3.
- **ECDSA**:
 - ANSI X9.62; NIST-FIPS-186-4, Abschnitt 6 (verfügbar unter: <http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf>);
 - ISO/IEC 14888-3 Digitale Signaturen mit Anhang – Teil 3: Mechanismen basierend auf diskreten Logarithmen (6.6).
- **NIST P256, secp256r1**:
 - Certicom-SEC-2 (verfügbar unter: <http://www.secg.org/sec2-v2.pdf>); NIST-Recommended-EC (verfügbar unter: <http://csrc.nist.gov/groups/ST/toolkit/documents/dss/NISTReCur.pdf>).
 - ISO/IEC 15946 Kryptographische Techniken auf Basis von elliptischen Kurven (enthält NIST P-256 als Beispiel)
 - *Anmerkungen: Die Reihe ISO/IEC 15946 behandelt elliptische Kurven anders als FIPS 186-4. In ISO/IEC 15946-5 geht es um die Generierung elliptischer Kurven. Das bedeutet, dass alle Anwendungen und Implementierungen ihre eigenen Kurven auf Grundlage der in Teil 5 beschriebenen Methode generieren können. In anderen Worten handelt es sich dabei nicht um von ISO/IEC empfohlene Kurven. P-256 wird in ISO/IEC 15946 als Beispiel betrachtet. Signaturen auf Basis von elliptischen Kurven und Schlüsselerstellungsverfahren wurden gemeinsam mit anderen auf diskreten Logarithmen basierenden Mechanismen in ISO/IEC 14888 bzw. ISO/IEC 11770 verschoben. Alle Teile dieser Mechanismen enthalten auf P-256 basierende Prüfvektoren (Beispiele).*

- **SHA256:**
 - NIST-FIPS-180-4 (verfügbar unter: <http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf>).
 - ISO/IEC 10118-3 Hash-Funktionen – Teil 3: Dedizierte Hash-Funktionen (Abschnitt 10).
- **OID** – ITU-T X.402 (verfügbar unter: <https://www.itu.int/rec/T-REC-X.402-199906-I/en>).
- **SP800-90A:**
 - NIST-SP-800-90A (verfügbar unter: <http://csrc.nist.gov/publications/nistpubs/800-90A/SP800-90A.pdf>).
 - *Anmerkung: NIST-SP-800-90A wurde im Juni 2015 zurückgezogen und durch NIST-SP-800-90A Revision 1 ersetzt*
<http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90Ar1.pdf>.
- **SP800-90B:**
 - NIST-SP-800-90B (verfügbar unter: http://csrc.nist.gov/publications/nistpubs/800-90/sp800-90b_second_draft.pdf)¹.

¹ Dieses Dokument befindet sich noch in der Entwurfsphase.

² Sofern nichts anderes angegeben ist, beziehen sich alle angegebenen Normen, einschließlich der ISO-, ITU- und NIST-Normen, auf die aktuellste Version oder Ausgabe mit Stand vom 1. Januar 2016.

1.4 Begriffe und Abkürzungen

Für die Anwendung dieses Dokuments gelten die folgenden Begriffe und Abkürzungen.

Tabelle 1-1 – Begriffe und Abkürzungen (1 von 2)

Begriff/Abkürzung	Definition
ACD (en: Additional Certificate Data for a Product)	Zusätzliche Zertifikatsdaten für ein Produkt
Authentifizierung	Verfahren zur Bestätigung, ob eine Instanz das ist, was sie vorgibt zu sein
Authentifizierungs-Initiator	Produkt, das die Authentifizierung einleitet
Authentifizierungs-Beantworter	Produkt, bei dem ein Authentifizierungs-Initiator versucht, sich zu authentifizieren
Zertifikat (en: Certificate; Cert)	Digitale Form der Identifikation, die Informationen über die Instanz enthält und die Eigentümerschaft eines bestimmten öffentlichen Schlüssels zertifiziert
Zertifizierungsinstanz (en: Certificate Authority; CA)	Instanz, die Zertifikate ausstellt
Zertifikatskette	Reihe von zwei oder mehr Zertifikaten, in der jedes Zertifikat vom vorhergehenden Zertifikat in der Kette signiert ist
Instanz	Produkt oder mit diesem Produkt verbundene(r) Organisation, Anbieter oder Hersteller
Vertrauenswürdigkeitsstufe (en: Evaluation Assurance Level; EAL)	Numerische Bewertung, die die Tiefe und Genauigkeit einer Sicherheitsbewertung beschreibt
Zwischenzertifikat	Zertifikat, das weder ein Root- noch ein Leaf-Zertifikat ist
Leaf-Zertifikat	Letztes Zertifikat in einer Zertifikatskette

Tabelle 1-1 (2 von 2)

Begriff/Abkürzung	Definition
LSB (en: Least Significant Byte)	Niedrigstwertiges Byte
MSB (en: Most Significant Byte)	Höchstwertiges Byte
Nonce	Zahl, die in einem bestimmten Schlüsselkontext nur einmal verwendet wird. Kann als N-Once (einmalig verwendete Zahl) interpretiert werden
OID (en: Object Identifier)	Objektbezeichner. Weitere Informationen hierzu siehe OID
PD (en: Power Delivery)	USB-Stromversorgung
PD-Produkt	Quelle, Verbraucher oder Kabel nach USBPD
PDUSB-Produkt	USB-Host oder USB-Gerät, das sowohl PD- als auch USB-Kommunikation unterstützt
Richtlinie	Eine Richtlinie definiert das Verhalten von Produkten. Sie legt die für das Produkt angegebenen Fähigkeiten, seine Authentifizierungsanforderungen und die Ressourcenverfügbarkeit hinsichtlich nicht authentifizierter Produkte fest
Produkt	Bezieht sich auf ein PD-Produkt, USB-Produkt und/oder PDUSB-Produkt
Pseudozufallszahlengenerator (en: Pseudorandom Number Generator; PRNG)	Deterministischer Algorithmus zur Erzeugung einer Zahl oder Zahlenfolge, die rechnerisch von echt zufälliger Erzeugung nicht zu unterscheiden ist. Weitere Informationen hierzu siehe SP800-90A
Root-Zertifikat	Erstes Zertifikat in einer Zertifikatskette. Dieses Zertifikat signiert sich selbst
TLV (en: Type, Length, Value)	Typ, Länge, Wert
USB-Gerät	Gerät (einschließlich Hub) wie in USB2.0 und USB3.1 definiert
USB-Host	Host wie in USB2.0 und USB3.1 definiert
USB-Hub	Hub wie in USB2.0 und USB3.1 definiert
USB-Produkt	USB-Host oder USB-Gerät

1.5 Konventionen

1.5.1 Prioritäten

Besteht ein Widerspruch zwischen Text, Bildern und Tabellen, haben Tabellen Vorrang vor Bildern und Bilder Vorrang vor dem Text.

1.5.2 Schlüsselwörter

Anhand der folgenden Schlüsselwörter wird zwischen verschiedenen Anforderungs- und Optionsstufen unterschieden.

1.5.2.1 Bedingt normativ

Bedingt normativ ist ein Schlüsselwort, das eine Eigenschaft bezeichnet, die dann verpflichtend wird, wenn eine andere verwandte Eigenschaft implementiert wurde. Wurden die abhängigen Eigenschaften implementiert, sind Konstrukteure dazu verpflichtet, alle derartigen Anforderungen zu implementieren, um die Interoperabilität mit anderen konformen Produkten sicherzustellen.

1.5.2.2 Abgelehnt

Abgelehnt ist ein Schlüsselwort, das eine Eigenschaft bezeichnet, die in früheren Ausgaben der Spezifikation unterstützt wurde, nun jedoch nicht mehr unterstützt wird.

1.5.2.3 Informativ

Informativ ist ein Schlüsselwort, das Informationen zu dieser Spezifikation beschreibt, die der Erörterung und Erklärung von Anforderungen und Merkmalen dienen, ohne sie vorzuschreiben.

1.5.2.4 Darf

Darf ist ein Schlüsselwort, das eine Wahlmöglichkeit angibt, ohne einen Vorrang zu implizieren.

1.5.2.5 N. z.

N. z. ist ein Schlüsselwort, das angibt, dass ein Feld oder Wert nicht zutrifft, daher keinen festgelegten Wert hat und nicht vom Empfänger überprüft oder verwendet werden darf.

1.5.2.6 Normativ

Normativ ist ein Schlüsselwort, das Merkmale beschreibt, die von dieser Spezifikation vorgeschrieben werden.

1.5.2.7 Optional/Optional normativ

Optional und **Optional normativ** sind gleichbedeutende Schlüsselwörter, die Eigenschaften beschreiben, die nicht von dieser Spezifikation vorgeschrieben werden. Wird ein **optionales** Merkmal implementiert, muss es jedoch auf die in dieser Spezifikation festgelegte Weise implementiert werden.

1.5.2.8 Reserviert

Reserviert ist ein Schlüsselwort, das reservierte Bits, Bytes, Wörter, Felder und Codewerte bezeichnet, die für eine zukünftige Normierung vorgemerkt werden. Ihre Verwendung und Interpretation darf in zukünftigen Erweiterungen dieser Norm festgelegt werden und, soweit nicht anders angegeben, dürfen sie nicht in der Anbieterimplementierung verwendet oder verändert werden. Ein **reserviertes** Bit, Byte, Wort oder Feld muss vom Absender auf null gesetzt und vom Empfänger ignoriert werden. **Reservierte** Feldwerte dürfen nicht vom Absender gesendet werden. Sollten sie empfangen werden, muss der Empfänger sie ignorieren.

1.5.2.9 Muss/Normativ

Muss und **Normativ** sind Schlüsselwörter, die eine verbindlichen Anforderung angeben. Konstrukteure sind dazu verpflichtet, alle derartigen Anforderungen zu implementieren, um die Interoperabilität mit anderen konformen Produkten sicherzustellen.

1.5.2.10 Sollte

Sollte ist ein Schlüsselwort, das eine Wahlmöglichkeit mit einer bevorzugten Alternative angibt. Gleichwertig mit der Formulierung „es wird empfohlen, dass“.

1.5.3 Zahlen

Zahlen, auf die unmittelbar der Kleinbuchstabe „b“ folgt (z. B. 01b) sind Binärwerte. Zahlen, auf die unmittelbar der Kleinbuchstabe „h“ folgt (z. B. 3Ah) sind Hexadezimalwerte. Zahlen, auf die unmittelbar weder „b“ noch „h“ folgt sind Dezimalwerte.

1.5.4 Byte-Reihenfolge

Soweit nicht anders angegeben, werden alle Mehrfach-Byte-Werte in dieser Spezifikation in Little-Endian-Reihenfolge, d. h. vom LSB zum MSB, interpretiert und über den Bus übertragen.

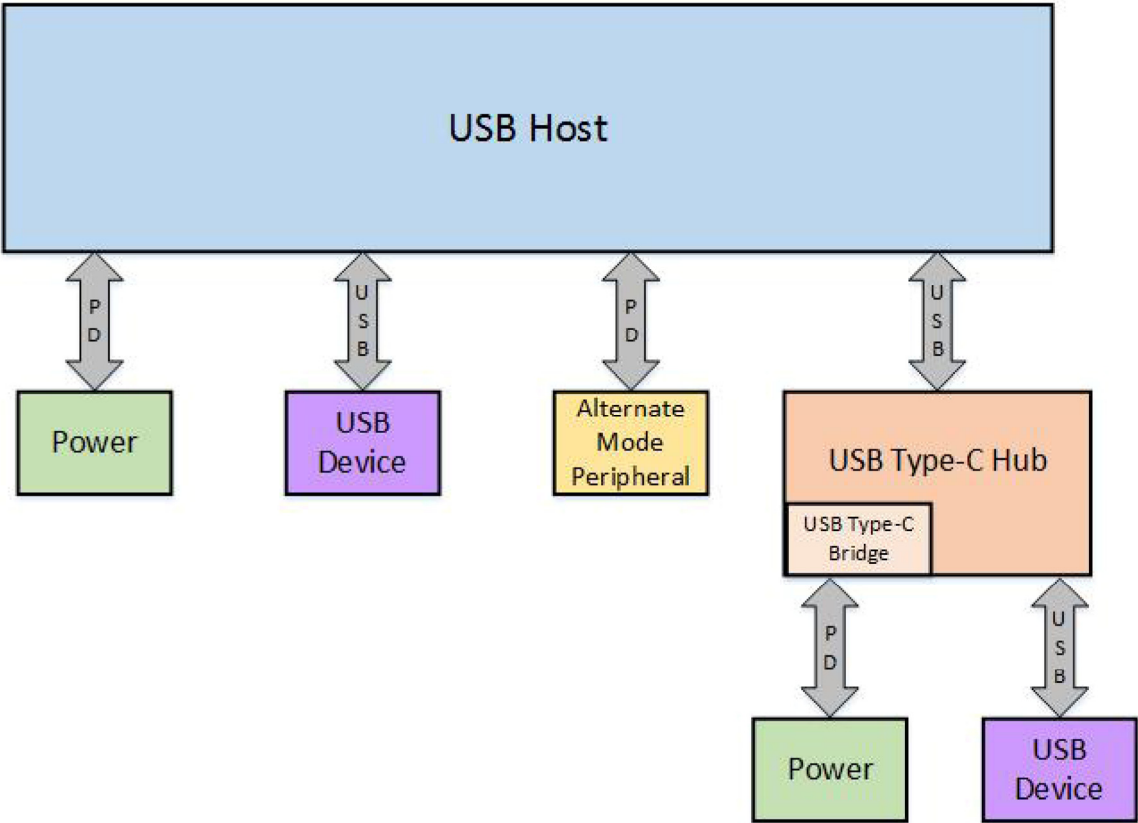
Die Reihenfolge, in der einzelne Bits über einen Bus übertragen werden, ist für PD-Produkte in **USBPD** und für USB-Produkte in **USB2.0** und **USB3.1** festgelegt.

2 Überblick

Dieser Abschnitt enthält keine normativen Anforderungen.

2.1 Topologie

In Bild 2-1 ist eine Beispieltopologie von USB-Produkten und PD-Produkten dargestellt, die USB-Typ-C Authentifizierung unterstützen. Diese Darstellung erhebt keinen Anspruch auf Vollständigkeit oder auf die Einbeziehung aller möglichen Topologie-Konfigurationen.



Text im Bild:

englisch

USB Host
PD
USB
Power
USB Device
Alternate Mode Peripheral
USB Type-C Hub
USB Type-C Bridge

deutsch

USB-Host
PD
USB
Energie
USB-Gerät
Peripheriegerät mit Alternativmodus
USB-Typ-C-Hub
USB-Typ-C-Brücke

Bild 2-1 – Beispieltopologie

2.2 Verschlüsselungsverfahren

Diese Spezifikation strebt eine Sicherheitsstufe mit 128-Bit-Verschlüsselung für alle Verschlüsselungsverfahren an. Die in dieser Spezifikation verwendeten Verschlüsselungsverfahren sind in Tabelle 2-1 dargestellt.

Tabelle 2-1 – Zusammenfassung der Verschlüsselungsverfahren

Verfahren	Verwendung
Verschlüsselung nach X509v3, DER	Zertifikatsformat
ECDSA mit NIST P256, secp256r1 -Kurve, nicht komprimiertes Punktformat	Digitale Signierung von Zertifikaten und Authentifizierungsnachrichten
SHA256	Hash-Algorithmus

2.2.1 Zufallszahlen

Die Erzeugung kryptografischer Schlüssel und der Austausch kryptografischer Protokolle basieren auf kryptografisch sicheren Zufallszahlen. Zufallszahlen sind Zahlen, deren Randomisierung von keinem Algorithmus mit einer algorithmischen Komplexität von weniger als $O(2^{128})$ erkannt werden kann.

Die erzeugten Werte eines mit NIST **SP800-90A** konformen PRNG mit einem Seed mit voller 256-Bit-Entropie nach **SP800-90B** ist ausreichend, um diese Norm zu erfüllen.

2.3 Überblick über die Sicherheit

Diese Spezifikation legt ein zertifikatbasiertes Authentifizierungsverfahren fest, das es einem Produkt ermöglicht, ein anderes angeschlossenes Produkt zu authentifizieren und auf Grundlage einer Richtlinie über die Art der Interaktion mit diesem Produkt zu entscheiden. So kann zum Beispiel ein PD-Verbraucher entscheiden, nicht alle angegebenen Fähigkeiten einer nicht authentifizierten PD-Quelle zu nutzen. Die Authentifizierung kann entweder von einem PD-Verbraucher, einer PD-Quelle oder einem USB-Host eingeleitet werden.

2.3.1 Periodische Wiederholung der Authentifizierung

Produkte können optional die Authentifizierung periodisch wiederholen. Die Wiederholung der Authentifizierung dient der Überprüfung, dass ein authentifiziertes Produkt nicht durch ein anderes Produkt ersetzt wurde.

2.3.2 Speicherung und Schutz geheimer Schlüssel

Eine Gefahr in Bezug auf die USB-Typ-C-Authentifizierung ist die Extraktion von sicheren Schlüsseln aus Produkten. Im schlimmsten Fall könnte ein Angreifer durch Extraktion eines einzigen Schlüssels Produkte in unbegrenztem Umfang klonen. Ein solches Szenario würde das Vertrauen in das gesamte Ökosystem der USB-Typ-C-Authentifizierung erschüttern.

Aus diesem Grund sei es Produkthanbietern empfohlen, geeignete Maßnahmen zum Schutz der Ausführung des USB-Typ-C-Authentifizierungsprotokolls und aller privaten Schlüssel zu ergreifen. Die Produkte sollten einen geschützten, manipulationssicheren Betrieb bieten und die Speicherung der privaten Schlüssel sollte so ausgelegt sein, dass diese weder ganz noch teilweise gelesen, kopiert oder auf andere Art offengelegt werden können. Hierzu gehört der Schutz vor Seitenkanalattacken und Fehlereinstreuungsangriffen, einschließlich Software-Exploits und physikalischen Angriffen wie Lecks, Probing, Glitch-Attacks, Reverse Engineering und statistische Analyseverfahren. Beispiele für solche Angriffe sind Angriffe auf Grundlage differentieller Leistungsanalysen erster und höherer Ordnung, elektromagnetischer Analyse und Fehleranalyse. Weitere Beispiele für Angriffsvektoren sind in Anhang B aufgeführt.

2.3.3 Kriterien für die Sicherheitsbewertung

Der Bedarf an zuverlässigen Sicherheitsbewertungen mit messbaren Ergebnissen hat zu weltweit anerkannten Zertifizierungs- und Bewertungsverfahren geführt. Eines der umfangreichsten und flächendeckendsten Verfahren zur Sicherheitsbewertung ist **Common Criteria**, das eine globale Infrastruktur für allgemein anerkannte Zertifikate bietet. Diese Infrastruktur umfasst eine staatliche Überwachung von Zertifizierungsbehörden, akkreditierte und gleichberechtigte Evaluierungslaboratorien und weltweit einheitliche, internationale Anerkennungsverträge auf Gegenseitigkeit. Zudem stellt **Common Criteria** verschiedene Bewertungen zur Verfügung, aus denen Anbieter die entsprechende Stufe wählen können.

Produkthanbietern wird empfohlen, die für den jeweiligen Markt erforderliche Vertrauenswürdigkeitsstufe (en: evaluation assurance level; EAL) zu wählen und eine unabhängige Bewertung durch ein akkreditiertes Bewertungslabor durchführen zu lassen. Nach der Bewertung veröffentlicht die Zertifizierungsstelle das Ergebnis in Form eines Zertifikats, welches automatisch in vielen verschiedenen Ländern anerkannt wird.

2.4 Auswirkungen auf das bestehende Ökosystem

Die Auswirkungen auf bestehende Produkte sind in großem Maße von einzelnen Richtlinienentscheidungen zu Altprodukten abhängig (d. h. Produkte, die vor Veröffentlichung dieser Spezifikation eingeführt wurden). So hat zum Beispiel ein USB-Host mit einer Richtlinie, die den vollen Funktionsumfang von Altgeräten unterstützt, nur minimale Auswirkungen auf das bestehende Ökosystem, während ein USB-Host mit einer Richtlinie, die Funktionalitäten in Verbindung mit Altprodukten einschränkt oder ausschließt, größere Auswirkungen hat.

2.4.1 Proxy-Fähigkeiten (die Hub-Topologie durchziehende PD)

PD ist ein Port-zu-Port-Kommunikationsprotokoll. Um PD-Produkte authentifizieren zu können, die von einem USB-Hub gesehen in Abwärtsrichtung angeschlossen werden, muss sowohl im USB-Hub als auch im zugehörigen Treiber im USB-Host eine USB-Typ-C-Brückenfunktion implementiert sein. Die **USB-Typ-C-Brücke** wird verwendet, um PD-Anforderungen an die USB-Datendomäne zu übersetzen. Weitere Informationen hierzu siehe USBTYPEC BRIDGE.

3 Authentifizierungs-Architektur

3.1 Zertifikate

3.1.1 Format

Alle Zertifikate müssen entsprechend der **X509v3-ASN.1**-Struktur aufgebaut sein. Für alle Zertifikate muss eine binäre **DER**-Codierung für **ASN.1** verwendet werden. Alle Zertifikate müssen die in Tabelle 2-1 aufgeführten Verschlüsselungsverfahren anwenden. Für die weitere Beschreibung des Zertifikatsformats werden Kenntnisse der **X509v3**-Zertifikatsterminologie vorausgesetzt.

Leaf-Zertifikate dürfen die Länge *MaxLeafCertSize* nicht überschreiten. Zwischenzertifikate dürfen die Länge *MaxIntermediateCertSize* nicht überschreiten.

Zertifikate und die darin festgelegten Felder, Attribute und Erweiterungen sind im Big-Endian-Format dargestellt.

3.1.2 Textformat

Alle textuellen **ASN.1**-Objekte in Zertifikaten, darunter DirectoryString, GeneralName und DisplayText, müssen entweder als UTF8String, PrintableString oder IA5String festgelegt sein. Die Länge eines textuellen Objekts darf 64 Byte nicht überschreiten, ausschließlich der **DER**-Typen- und der **DER**-Längencodierung.

3.1.3 Attribute und Erweiterungen

Sofern zutreffend wird der Objektbezeichner (en: Object Identifier; OID) angegeben.

3.1.3.1 Ausgezeichneter Name

Der ausgezeichnete Name besteht aus verschiedenen Attributen, die die Instanz, die einen entsprechenden privaten Schlüssel enthält, eindeutig bezeichnen. Eine Zertifizierungsinstanz darf für verschiedene Instanzen keine Zertifikate mit dem gleichen ausgezeichneten Namen ausstellen. Die Eindeutigkeit des ausgezeichneten Namen kann durch Verwendung eines Attributs mit eindeutigen Werten, z. B. dem binären X500Uniquel-Identifier oder der textuellen Seriennummer (siehe 3.1.3.1.3), erreicht werden.

3.1.3.1.1 Gemeinsamer Name (OID 2.5.4.3)

Dieses Attribut muss in jedem Zertifikat enthalten sein und muss einen String beinhalten, der mit einem der drei folgenden Muster übereinstimmt:

- „USB:“
- „USB:<vid>“
- „USB:<vid>:<pid>“

Dabei steht <vid> für einen vierstelligen hexadezimalen String mit Kleinbuchstaben, der 16-Bit-Werte entsprechend der VID des Zertifikat-Betreffs verschlüsselt, und <pid> für einen vierstelligen hexadezimalen String mit Kleinbuchstaben, der 16-Bit-Werte entsprechend der PID des Zertifikat-Betreffs verschlüsselt. Wenn vorhanden, müssen <vid> und <pid> mit Nullen aufgefüllt werden und Big Endian sein. Die Hex-Codierung von VID und PID darf keine Großbuchstaben enthalten.

Das Attribut *gemeinsamer Name* im Leaf-Zertifikat einer Zertifikatskette muss sowohl eine VID als auch eine PID enthalten. Bei Nicht-Leaf-Zertifikaten ist die Angabe von VID und PID im Attribut *gemeinsamer Name* optional. Enthält jedoch ein Zertifikat der Kette einen VID-Wert, muss der gleiche VID-Wert auch in allen nachfolgenden Zertifikaten verwendet werden. Enthält ein Zertifikat der Kette einen PID-Wert, muss der gleiche PID-Wert auch in allen nachfolgenden Zertifikaten verwendet werden.

3.1.3.1.2 Organisationsname (OID 2.5.4.10)

Dieses Attribut muss in einem Root-Zertifikat enthalten sein und darf in anderen Zertifikaten enthalten sein. Wenn vorhanden, muss das *Organisationsname*-Attribut den visuell lesbaren Namen der Organisation enthalten, die im Besitz des zum Zertifikat passenden privaten Schlüssels ist.

3.1.3.1.3 Seriennummer (OID 2.5.4.5)

Dieses Attribut ist optional. Wenn vorhanden, darf es nur in einem Leaf-Zertifikat enthalten sein. Verschiedene PD-Produkte, auch solche aus ein und derselben Fertigungslinie, haben unterschiedliche Schlüssel und benötigen daher unterschiedliche *ausgezeichnete Namen*. Das Attribut *gemeinsamer Name* und das *Organisationsname*-Attribut bieten keine solche Eindeutigkeit. Der *ausgezeichnete Name* enthält mehrere Attribute, die verwendet werden können, um ihn eindeutig zu machen. Es wird empfohlen, das *Seriennummern*-Attribut mit einem hex-codierten Wert mit Kleinbuchstaben der für die Eindeutigkeit benötigten binären Daten (z. B. Wafer-Nummer, Losnummer, Fertigungslos usw.) zu befüllen.

3.1.3.2 Grundlegende Beschränkungen (OID 2.5.29.19)

Diese Erweiterung muss vorhanden und als kritisch markiert sein. In einem Leaf-Zertifikat muss die *cA*-Komponente „false“ sein. In einem Nicht-Leaf-Zertifikat muss die *cA*-Komponente „true“ sein. Es dürfen keine weiteren Komponenten enthalten sein, einschließlich *pathLenConstraint*.

3.1.3.3 Schlüsselverwendung (OID 2.5.29.15)

Diese Erweiterung muss vorhanden sein. Bei Leaf-Zertifikaten muss das *digitalSignature*-Bit gesetzt und alle anderen Bits müssen gelöscht sein. Bei Nicht-Leaf-Zertifikaten muss das *keyCertSign*-Bit gesetzt sein. Optional darf das *cRLSign*-Bit gesetzt sein. Alle anderen Bits müssen gelöscht sein.

3.1.3.4 Erweiterte Schlüsselverwendung (OID 2.5.29.37)

Diese Erweiterung muss vorhanden und als kritisch markiert sein. Sie muss den vom USB-IF vergebenen OID 2.23.145.1.1 für die erweiterte Schlüsselverwendung „USB-Auth“ enthalten und darf weitere OIDs enthalten.

3.1.3.5 Gültigkeit

Die Felder *notBefore* und *notAfter* geben die Zeitspanne an, in der die Informationen bezüglich der Gültigkeit eines Zertifikats aufrechterhalten werden. Bei Produkten sollten die Gültigkeitszeitspannen ignoriert werden.

Die *notBefore*- und *notAfter*-Gültigkeitszeitspannen müssen entweder in ASN.1 GeneralizedTime für ein beliebiges Jahr oder in ASN.1 UTCTime für Jahre vor 2050 angegeben werden.

Für das *notBefore*-Feld wird ein Wert von „19700101000000Z“ empfohlen (für 00:00 am 01. Jan. 1970 UTC, dem Beginn der POSIX-Epoche). Für das *notAfter*-Feld wird ein Wert von „99991231235959Z“ empfohlen (für 23:59:59 am 31. Dez. 9999 UTC, der wie in IETF-RFC-5280, 4.1.2.5, definiert für ein unbekanntes Ablaufdatum verwendet wird). Die Verwendung der empfohlenen *notBefore*- und *notAfter*-Werte sorgen für eine maximierte Kompatibilität mit Zertifikat-Verarbeitungstapeln.

3.1.3.6 USB-IF ACD (OID 2.23.145.1.2)

Die Erweiterung USB-IF ACD ist eine anbieterspezifische Zertifikaterweiterung für mit dieser Spezifikation übereinstimmende Produkte. Sie enthält ein Binärobjekt im ACD-Format, das in Anhang A.1 beschrieben wird. Das Binärobjekt ist als **ASN.1 DER** OCTET STRING mit einer maximalen Größe von *MaxACDSize* Byte verschlüsselt. Die Felder für die ACD eines PD-Produkts sind in Anhang A.2 festgelegt. Die Felder für die ACD eines USB-Produkts sind in Anhang A.3 festgelegt.

Leaf-Zertifikate müssen diese Erweiterung enthalten. Nicht-Leaf-Zertifikate dürfen diese Erweiterung nicht enthalten.

3.1.3.7 Weitere Attribute und Erweiterungen

Weitere in **X509v3** festgelegte Zertifikatattribute und -erweiterungen sind zulässig, sofern die maximale Zertifikatgröße *MaxLeafCertSize* für ein Leaf-Zertifikat bzw. *MaxIntermediateCertSize* für ein Nicht-Leaf-Zertifikat nicht überschreitet.

3.2 Zertifikatsketten

Zertifikate werden in Zertifikatsketten zusammengefasst. Eine Zertifikatskette ist die binäre Verkettung (von Bytes) der in Tabelle 3-1 dargestellten Felder. Eine Zertifikatskette darf *MaxCertChainSize* Byte nicht überschreiten.

Tabelle 3-1 – Format einer Zertifikatskette

Offset	Feld	Größe	Beschreibung
0	<i>Länge</i>	2	Gesamtlänge einer Zertifikatskette in Byte einschließlich aller in dieser Tabelle aufgeführten Felder. Dieses Feld ist Little Endian.
2	<i>Reserviert</i>	2	Auf null gesetzt
4	<i>RootHash</i>	32	32-Byte- SHA256 -Hashwert des Root-Zertifikats. Dieses Feld ist Big Endian.
36	<i>Zertifikate</i>	Länge – 36	Ein oder mehrere ASN.1-DER -codierte X509v3 -Zertifikate, von denen das erste vom Root-Zertifikat und jedes nachfolgende vom vorhergehenden Zertifikat signiert ist. Das letzte Zertifikat ist das Leaf-Zertifikat. Dieses Feld ist Big Endian.

Zertifikatsketten befinden sich in sogenannten „Slots“. Jeder Slot muss entweder leer sein oder eine vollständige Zertifikatskette enthalten. Ein Produkt darf nicht mehr als 8 Slots umfassen. Slots 0 bis 3 dürfen nur für Zertifikatsketten verwendet werden, die mit einem USB-IF-Root-Zertifikat verwurzelt sind, und dürfen

keine anderen Zertifikatsketten enthalten. Slots 4 bis 7 dürfen für weitere Zertifikatsketten verwendet werden. Die **ASN.1-DER**-Codierung jedes einzelnen Zertifikats kann zur Bestimmung seiner Länge analysiert werden.

3.2.1 Bereitstellung

Die Bereitstellung ist der Vorgang, durch den ein Produkt eine oder mehrere Zertifikatsketten erhält. Das Bereitstellungsverfahren für ein Produkt liegt außerhalb des Anwendungsbereichs dieser Spezifikation.

3.3 Private Schlüssel

Jeder Zertifikatskette in einem Produkt ist ein privater Schlüssel zugeordnet, dessen entsprechender öffentlicher Schlüssel im Leaf-Zertifikat dieses Slots zertifiziert ist. Das Produkt muss Zugriff auf diesen privaten Schlüssel haben. Private Schlüssel müssen auf eine Art und Weise generiert, bereitgestellt und gespeichert werden, durch die die Vertraulichkeit des Schlüssels ausreichend geschützt wird.

Die Eindeutigkeit eines privaten Schlüssels wird durch Einhaltung folgender Regeln sichergestellt:

- Ein privater Schlüssel, der von einem Produkt verwendet wird, darf nicht von einem anderen Produkt verwendet werden.
- Ein privater Schlüssel, der von einem Produkt auf einem Slot verwendet wird, darf nicht auf einem anderen Slot verwendet werden.

4 Authentifizierungs-Protokoll

Ein Authentifizierungs-Initiator kann drei Vorgänge durchführen:

- Zertifikatsketten-Digests von einem Authentifizierungs-Beantworter abfragen;
- eine Zertifikatskette von einem Authentifizierungs-Beantworter lesen;
- eine Aufforderung an einen Authentifizierungs-Beantworter senden, um dessen Authentizität zu überprüfen.

Ein Authentifizierungs-Initiator darf so viele oder wenige dieser Vorgänge durchführen wie benötigt werden, um die gewünschte Authentifizierungslatenz zu erreichen. Ein Authentifizierungs-Initiator darf die Vorgänge außerdem in einer beliebigen Reihenfolge einleiten. Zum Beispiel darf ein Authentifizierungs-Initiator, der nur Slot 0 für die Authentifizierung verwendet, zuerst eine Anforderung an ein PD-Produkt senden, und dann das Lesen einer Zertifikatskette einleiten, wenn die Ziel-Zertifikatskette noch nicht zwischengespeichert wurde.

Ein Produkt darf nur dann Authentifizierungs-Beantworter sein, wenn es eine Zertifikatskette in Slot 0 enthält.

4.1 Digest-Abfrage

Um Zertifikatsketten-Digests von einem Authentifizierungs-Beantworter abzufragen, sendet ein Authentifizierungs-Initiator wie in 5.2.1 festgelegt eine GET_DIGESTS-Anforderung. Tritt ein Fehler auf, muss der Authentifizierungs-Beantworter wie in 4.4 festgelegt mit der entsprechenden ERROR-Antwort antworten. Anderenfalls muss der Authentifizierungs-Beantworter wie in 5.3.1 festgelegt mit einer DIGESTS-Antwort antworten. Nach Erhalt einer DIGESTS-Antwort kann ein Authentifizierungs-Initiator überprüfen, ob er eine der Zertifikatsketten des Authentifizierungs-Beantworters zwischengespeichert hat. Dadurch kann der Authentifizierungs-Initiator eventuell den Lesevorgang der Zertifikatskette überspringen und somit Zeit sparen.

4.2 Lesen einer Zertifikatskette

Um eine Zertifikatskette oder einen Teil von ihr zu lesen, sendet ein Authentifizierungs-Initiator wie in 5.2.2 festgelegt eine GET_CERTIFICATE-Anforderung.

Erhält ein Authentifizierungs-Beantworter eine GET_CERTIFICATE-Anforderung, die an einen Offset außerhalb der Zertifikatskette gerichtet ist (d. h. Offset > Länge) oder versucht, über die Länge der Ziel-Zertifikatskette hinaus zu lesen (d. h. Offset + Länge > Länge der Zertifikatskette), so muss der Authentifizierungs-Beantworter eine ERROR-Authentifizierungsantwort mit *Param1* gleich INVALID_REQUEST und *Param2* gleich 00h zurücksenden.

Tritt ein Fehler auf, muss der Authentifizierungs-Beantworter wie in 4.4 festgelegt mit der entsprechenden ERROR-Antwort antworten. Anderenfalls muss der Authentifizierungs-Beantworter wie in 5.3.2 beschrieben mit einer CERTIFICATE-Antwort antworten.

4.3 Authentifizierungs-Challenge

Für eine Challenge an einen Authentifizierungs-Beantworter sendet ein Authentifizierungs-Initiator wie in 5.2.3 festgelegt eine CHALLENGE-Anforderung. Tritt ein Fehler auf, muss der Authentifizierungs-Beantworter wie in 4.4 festgelegt mit der entsprechenden ERROR-Antwort antworten. Anderenfalls muss der Authentifizierungs-Beantworter wie in 5.3.3 beschrieben mit einer CHALLENGE_AUTH-Antwort antworten.

4.4 Fehler und Warnungen

4.4.1 Invalid Request

Erhält ein Authentifizierungs-Beantworter eine Authentifizierungsanforderung mit einem oder mehreren ungültigen Feldern, muss er auf diese Authentifizierungsanforderung mit einer ERROR-Antwort mit *Param1* gleich INVALID_REQUEST und *Param2* gleich 00h antworten.

4.4.2 Unsupported Protocol Version

Erhält ein Authentifizierungs-Beantworter eine Authentifizierungsanforderung mit einer nicht unterstützten Sicherheitsprotokoll-Version im *Protocol/Version*-Feld, so muss er auf die Authentifizierungsanforderung mit einer ERROR-Antwort antworten, in der *Protocol/Version* auf die niedrigste unterstützte Sicherheitsprotokollversion, *Param1* auf UNSUPPORTED_PROTOCOL und *Param2* auf die höchste unterstützte Sicherheitsprotokollversion gesetzt ist.

4.4.3 Busy

Erhält ein Authentifizierungs-Beantworter eine Authentifizierungsanforderung, ist jedoch nicht in der Lage, die in 6.3 (für PD-Produkte) oder 7.4 (für USB-Produkte) angegebenen Zeitvorgaben einzuhalten, so muss er auf diese Authentifizierungsanforderung mit einer ERROR-Antwort mit *Param1* gleich BUSY und *Param2* gleich 00h antworten.

4.4.4 Unspecified

Erhält ein Authentifizierungs-Beantworter eine Authentifizierungsanforderung und tritt dann ein Fehler auf, der nicht unter die oben beschriebenen Bedingungen fällt, so muss er auf diese Authentifizierungsanforderung mit einer ERROR-Antwort mit *Param1* gleich UNSPECIFIED und *Param2* gleich 00h antworten.

5 Authentifizierungsnachrichten

Authentifizierungsnachrichten werden verwendet, um authentifizierungsbezogene Informationen zu übermitteln. Eine Authentifizierungsnachricht besteht aus einem Nachrichtenkopf, gefolgt von einem Nutzdatum mit veränderlicher Länge (einschließlich null). Das Format eines Nachrichtenkopfes ist in 5.1 festgelegt.

Es gibt zwei Arten von Authentifizierungsnachrichten: Authentifizierungsanforderungen und Authentifizierungsantworten. Authentifizierungsanforderungen sind in 5.2 festgelegt. Authentifizierungsantworten sind in 5.3 festgelegt.

Abschnitt 6 beschreibt die Zuordnung von Authentifizierungsnachrichten in der PD-Nachrichtenübertragung. Abschnitt 7 beschreibt die Zuordnung von Authentifizierungsnachrichten in USB-Übertragungen.

5.1 Kopf

Alle Authentifizierungsnachrichten müssen mit dem in Tabelle 5-1 festgelegten 4-Byte-Kopf beginnen.

Tabelle 5-1 – Authentifizierungsnachrichtenkopf

Offset	Feld	Größe	Verweisung
0	<i>ProtocolVersion</i>	1	Abschnitt 5.1.1
1	<i>MessageType</i>	1	Abschnitt 5.1.2
2	<i>Param1</i>	1	Abschnitt 5.1.3
3	<i>Param2</i>	1	Abschnitt 5.1.4

5.1.1 USB-Typ-C-Authentifizierungsprotokollversion

Dieses Feld legt fest, welche Version der USB-Typ-C-Authentifizierungsspezifikation verwendet wird. In Tabelle 5-2 sind die gültigen Werte für dieses Feld angegeben. Ein Produkt darf für die Version des USB-Typ-C-Authentifizierungsprotokolls keinen Wert verwenden, der einer von ihm nicht unterstützten Spezifikationsüberarbeitung entspricht.

Tabelle 5-2 – USB-Typ-C-Authentifizierungsprotokollversion

Name	Wert	Bedeutung
Reserviert	00h	Reserviert
V1.0	01h	USB-Typ-C-Authentifizierungsprotokollversion 1.0
Reserviert	02h-ffh	Reserviert

Künftig sollen Produkte eine fortlaufende Reihe von USB-Typ-C-Authentifizierungsprotokollversionen unterstützen.

5.1.2 Nachrichtentyp

Dieses Feld legt den Authentifizierungsnachrichtentyp fest und muss einen der in Tabelle 5-3 oder Tabelle 5-9 aufgeführten Authentifizierungsnachrichtentypen enthalten.

5.1.3 Param1

Dieses Feld wird zur Übermittlung eines ersten 1-Byte-Parameters verwendet. Der Inhalt dieses Parameters ist veränderlich und wird durch den Authentifizierungsnachrichtentyp festgelegt.

5.1.4 Param2

Dieses Feld wird zur Übermittlung eines zweiten 1-Byte-Parameters verwendet. Der Inhalt dieses Parameters ist veränderlich und wird durch den Authentifizierungsnachrichtentyp festgelegt.

5.2 Authentifizierungsanforderungen

Authentifizierungsanforderungen werden von einem Authentifizierungs-Initiator verwendet, um einen Befehl an einen Authentifizierungs-Beantworter zu senden und/oder um Daten abzufragen. Die Arten von Authentifizierungsanforderungen sind in Tabelle 5-3 aufgeführt.

Solange ein Authentifizierungs-Initiator weder eine Antwort auf die zuvor versendete Authentifizierungsanforderung erhalten hat, noch ein Timeout ausgelöst hat, darf er keine neue Authentifizierungsanforderung senden.

Tabelle 5-3 – Arten von Authentifizierungsanforderungen

Wert	Beschreibung
00h – 7Fh	Darf nur für Authentifizierungsantworten verwendet werden
80h	Reserviert
81h	GET_DIGESTS
82h	GET_CERTIFICATE
83h	CHALLENGE
84h – FFh	Reserviert

5.2.1 GET_DIGESTS

Diese Anforderung wird zur Abfrage von Zertifikatsketten-Digests verwendet. Der Kopf einer GET_DIGESTS-Anforderung ist in Tabelle 5-4 festgelegt. Eine GET_DIGESTS-Anforderung enthält keine Nutzdaten.

Tabelle 5-4 – Kopf einer GET_DIGESTS-Anforderung

Offset	Feld	Größe	Wert
0	<i>ProtocolVersion</i>	1	V1.0
1	<i>MessageType</i>	1	GET_DIGESTS
2	<i>Param1</i>	1	Reserviert
3	<i>Param2</i>	1	Reserviert

5.2.2 GET_CERTIFICATE

Diese Anforderung wird verwendet, um ein Segment einer Ziel-Zertifikatskette zu lesen. Der Kopf einer GET_CERTIFICATE-Anforderung ist in Tabelle 5-5 festgelegt. Das Nutzdatum einer GET_CERTIFICATE-Anforderung ist in Tabelle 5-6 festgelegt.

Tabelle 5-5 – Kopf einer GET_CERTIFICATE-Anforderung

Offset	Feld	Größe	Wert
0	<i>ProtocolVersion</i>	1	V1.0
1	<i>MessageType</i>	1	GET_CERTIFICATE
2	<i>Param1</i>	1	Slotnummer der Ziel-Zertifikatskette, die gelesen werden soll. Der Wert dieses Felds muss zwischen 0 und 7 (einschließlich) liegen.
3	<i>Param2</i>	1	Reserviert

Tabelle 5-6 – Nutzdatum einer GET_CERTIFICATE-Anforderung

Offset	Feld	Größe	Wert
4	<i>Offset</i>	2	Offset in Byte ab dem Beginn der Zertifikatskette bis zu dem Byte, an dem die Leseanforderung beginnt. Dieses Feld ist Little Endian.
6	<i>Länge</i>	2	Länge der Leseanforderung in Byte. Dieses Feld ist Little Endian.

5.2.3 CHALLENGE

Diese Anforderung wird verwendet, um die Authentifizierung eines Produkts einzuleiten. Der Kopf einer CHALLENGE-Anforderung ist in Tabelle 5-7 festgelegt. Das Nutzdatum einer CHALLENGE-Anforderung ist in Tabelle 5-8 festgelegt.

Tabelle 5-7 – Kopf einer CHALLENGE-Anforderung

Offset	Feld	Größe	Wert
0	<i>ProtocolVersion</i>	1	V1.0
1	<i>MessageType</i>	1	CHALLENGE
2	<i>Param1</i>	1	Slotnummer der Zertifikatskette des Empfängers, die für die Authentifizierung verwendet wird
3	<i>Param2</i>	1	Reserviert

Tabelle 5-8 – Nutzdatum einer CHALLENGE-Anforderung

Offset	Feld	Größe	Beschreibung
4	<i>Nonce</i>	32	Zufällige 32-Byte-Nonce, die vom Authentifizierungs-Initiator gewählt wird. Dieses Feld ist Little Endian.

5.3 Authentifizierungsantworten

Authentifizierungsantworten werden von einem Authentifizierungs-Beantworter verwendet, um auf eine Authentifizierungsanforderung zu antworten. Die Arten von Authentifizierungsantworten sind in Tabelle 5-9 aufgeführt.

Tabelle 5-9 – Arten von Authentifizierungsantworten

Wert	Beschreibung
00h	Reserviert
01h	DIGESTS
02h	CERTIFICATE
03h	CHALLENGE_AUTH
04h-7Eh	Reserviert
7Fh	ERROR
80h – FFh	Darf nur für Authentifizierungsanforderungen verwendet werden

5.3.1 DIGESTS

Diese Antwort wird von einem Produkt verwendet, um Zertifikatsketten-Digests zu senden und mitzuteilen, welche Slots gültige Zertifikatsketten-Digests enthalten. Der Kopf einer DIGESTS-Antwort ist in Tabelle 5-10 festgelegt. Das Nutzdatum einer DIGESTS-Antwort ist in Tabelle 5-11 festgelegt.

Tabelle 5-10 – Kopf einer DIGESTS-Antwort

Offset	Feld	Größe	Wert
0	<i>ProtocolVersion</i>	1	V1.0
1	<i>MessageType</i>	1	DIGESTS
2	<i>Param1</i>	1	Fähigkeitsfeld; muss für diese Spezifikation auf 01h gesetzt sein. Alle anderen Werte sind reserviert.
3	<i>Param2</i>	1	Slot-Maske. Das Bit auf Position K dieses Bytes muss dann und nur dann auf 1b gesetzt werden, wenn Slot-Nummer K eine Zertifikatskette für die im <i>ProtocolVersion</i> -Feld angegebene Protokollversion enthält. (Bit 0 ist das niedrigstwertige Bit des Bytes.) Die Anzahl der zurückgesendeten Digests muss der Anzahl der in diesem Byte gesetzten Bits entsprechen. Die Digests müssen in aufsteigender Reihenfolge der Slot-Nummern zurückgesendet werden.

Tabelle 5-11 – Nutzdatum einer DIGESTS-Antwort

Offset	Feld	Größe	Wert
4	<i>Digest[0]</i>	32	32-Byte-SHA-256-Digest der ersten Zertifikatskette. Dieses Feld ist Big Endian.
...	...	...	...
4 + (32 * (n - 1))	<i>Digest[n-1]</i>	32	32-Byte-SHA-256-Digest der letzten (n-ten) Zertifikatskette. Dieses Feld ist Big Endian.

5.3.2 CERTIFICATE

Diese Anforderung wird von einem Produkt verwendet, um die angeforderten Segmente einer Zertifikatskette zu senden. Der Kopf einer CERTIFICATE-Antwort ist in Tabelle 5-12 festgelegt. Das Nutzdatum einer CERTIFICATE-Antwort ist in Tabelle 5-13 festgelegt.

Tabelle 5-12 – Kopf einer CERTIFICATE-Antwort

Offset	Feld	Größe	Wert
0	<i>ProtocolVersion</i>	1	V1.0
1	<i>MessageType</i>	1	CERTIFICATE
2	<i>Param1</i>	1	Slotnummer der zurückgesendeten Zertifikatskette
3	<i>Param2</i>	1	Reserviert

Tabelle 5-13 – Nutzdatum einer CERTIFICATE-Antwort

Offset	Feld	Größe	Wert	Beschreibung
4	<i>CertChain</i>	<i>Länge</i>	Daten	Angeforderter Inhalt der Ziel-Zertifikatskette. Die Endianness einer Zertifikatskette ist in Tabelle 3-1 festgelegt.

5.3.3 CHALLENGE_AUTH

Diese Antwort wird von einem Produkt verwendet, um auf eine CHALLENGE-Anforderung zu antworten. Der Kopf einer CHALLENGE_AUTH-Antwort ist in Tabelle 5-14 festgelegt. Das Nutzdatum einer CHALLENGE_AUTH-Antwort ist in Tabelle 5-15 festgelegt.

Tabelle 5-14 – Kopf einer CHALLENGE_AUTH-Antwort

Offset	Feld	Größe	Wert
0	<i>ProtocolVersion</i>	1	V1.0
1	<i>MessageType</i>	1	CHALLENGE_AUTH
2	<i>Param1</i>	1	Muss im Param1-Feld die Slot-Nummer der entsprechenden CHALLENGE-Anforderung enthalten
3	<i>Param2</i>	1	Slot-Maske. Das Bit auf Position K dieses Bytes muss dann und nur dann auf 1b gesetzt werden, wenn Slot-Nummer K eine Zertifikatskette für die im ProtocolVersion-Feld angegebene Protokollversion enthält. (Bit 0 ist das niedrigstwertige Bit des Bytes.)

Tabelle 5-15 – Nutzdatum einer CHALLENGE_AUTH-Antwort

Offset	Feld	Größe	Wert
4	<i>MinProtocolVersion</i>	1	Niedrigste von diesem Gerät unterstützte Protokollversion
5	<i>MaxProtocolVersion</i>	1	Höchste von diesem Gerät unterstützte Protokollversion
6	<i>Fähigkeiten</i>	1	Für diese Spezifikation auf 01h gesetzt. Alle anderen Werte sind reserviert
7	<i>Reserviert</i>	1	Reserviert
8	<i>CertChainHash</i>	32	32-Byte- SHA256 -Hashwert der zur Authentifizierung verwendeten Zertifikatskette. Dieses Feld ist Big Endian.
40	<i>Salt</i>	32	32-Byte-Wert, der vom Authentifizierungs-Beantworter gewählt wird. Dieses Feld ist Little Endian. <i>Anmerkung: Das Salt kann ein zufälliger, ein fester oder irgendein anderer Wert sein.</i>
72	<i>Context Hash</i>	32	Siehe 6.4 für PD-Produkte oder 7.5 für USB-Produkte. Dieses Feld ist Big Endian.
104	<i>Signatur</i>	64	Siehe 5.3.3.1. Dieses Feld ist Little Endian.

5.3.3.1 Signatur

Das *Signatur*-Feld in einer CHALLENGE_AUTH-Antwort enthält eine 64 Byte lange digitale **ECDSA**-Signatur zum in Tabelle 5-16 aufgeführten Nachrichteninhalt. Die **ECDSA**-Signatur wird unter Verwendung von Werten (r,s) mit Little-Endian-Codierung generiert, wobei r bei Offset 0 und s bei Offset 32 beginnt. Jeder Wert hat eine Länge von 32 Byte, wobei die rechte Seite bei Bedarf mit Nullen aufgefüllt wird.

Tabelle 5-16 – Nachrichteninhalte für digitale ECDSA-Signaturen

Offset	Feld	Größe	Wert
0	<i>ReqMsg</i>	36	Vollständiger Inhalt (d. h. Kopf und Nutzdatum) der entsprechenden CHALLENGE-Anforderung
36	<i>RespMsg</i>	104	Inhalt der zu signierenden CHALLENGE_AUTH-Antwort außer Signatur-Feld

Ein Nachrichtensignierer mit sicherem RNG kann nicht-deterministische **ECDSA** verwenden. Ein Nachrichtensignierer ohne sicheren RNG kann deterministische **ECDSA** verwenden. In deterministischen **ECDSA** wird der zufällige „k“-Wert vom Hash der zu signierenden Nachricht und einem privaten Schlüssel abgeleitet.

ANMERKUNG Sowohl in deterministischen als auch in nicht-deterministischen ECDSA ist die Generierung des „k“-Wertes anfällig für Fehler, die zur Offenlegung des privaten Schlüssels führen können. Für weitere Informationen siehe RFC 6979 (verfügbar unter: <https://tools.ietf.org/html/rfc6979>).

5.3.4 ERROR

Diese Antwort wird von einem Produkt verwendet, um Fehlerinformationen zu übertragen. Der Kopf einer ERROR-Antwort ist in Tabelle 5-17 festgelegt. Eine ERROR-Antwort enthält keine Nutzdaten.

Tabelle 5-17 – Kopf einer ERROR-Antwort

Offset	Feld	Größe	Beschreibung
0	<i>ProtocolVersion</i>	1	V1.0 ¹
1	<i>MessageType</i>	1	ERROR
2	<i>Param1</i>	1	Fehlercode. Siehe Tabelle 5-18.
3	<i>Param2</i>	1	Fehlerdaten. Siehe Tabelle 5-18.

Tabelle 5-18 – ERROR-Codes

Fehlercode	Wert	Beschreibung	Fehlerdaten
Reserviert	00h	Reserviert	Reserviert
INVALID_REQUEST	01h	Ein oder mehrere Anforderungsfelder sind ungültig	00h
UNSUPPORTED_PROTOCOL	02h	Die angeforderte Sicherheitsprotokollversion wird nicht unterstützt	Höchste unterstützte Sicherheitsprotokollversion ¹
BUSY	03h	Das Gerät kann derzeit nicht antworten, wird aber in der Zukunft in der Lage sein zu antworten	00h
UNSPECIFIED	04h	Ein unbestimmter Fehler ist aufgetreten	00h
Reserviert	05h- EFh	Reserviert	Reserviert
Vom Anbieter festgelegt	F0h- FFh	Vom Anbieter festgelegt	Vom Anbieter festgelegt

¹ANMERKUNG Die niedrigste unterstützte Sicherheitsprotokollversion wird im ProtocolVersion-Feld im Nachrichtenkopf zurückgesendet.

6 Authentifizierung von PD-Produkten

USBPD beschreibt den Mechanismus für das Senden von Authentifizierungsnachrichten über PD. Die Zuordnung eines Sicherheitstransfers zu PD-Nachrichten ist abhängig davon, ob die Übertragung *MaxExtendedMsgLen* überschreitet oder nicht. *MaxExtendedMsgLen* ist in **USBPD** festgelegt.

6.1 Übertragungen mit einer Länge kleiner oder gleich *MaxExtendedMsgLen*

Für Sicherheitsübertragungen, bei denen sowohl die Länge der Authentifizierungsanforderung als auch die Länge der entsprechenden Authentifizierungsantwort kleiner oder gleich der *MaxExtendedMsgLen* ist, wird nur ein einziger PD-Nachrichtenaustausch benötigt. Um eine Sicherheitsübertragung einzuleiten, sendet der Authentifizierungs-Initiator eine Security_Request-PD-Nachricht, die im Datenblock der PD-Authentifizierungsanforderung (SRQDB) eine Authentifizierungsanforderung enthält. Um die Sicherheitsübertragung abzuschließen, sendet ein Authentifizierungs-Beantworter eine Security_Response-PD-Nachricht, die im Datenblock der PD-Authentifizierungsantwort (SRPDB) eine Authentifizierungsantwort enthält.

Um zum Beispiel Zertifikatsketten-Digests abzufragen, sendet ein Authentifizierungs-Initiator eine Security_Request-PD-Nachricht, die eine GET_DIGESTS-Authentifizierungsnachricht enthält. Der Authentifizierungs-Beantworter antwortet dann mit einer Security_Response-PD-Nachricht, die eine DIGESTS-Authentifizierungsnachricht (oder zutreffende ERROR-Authentifizierungsnachricht) enthält.

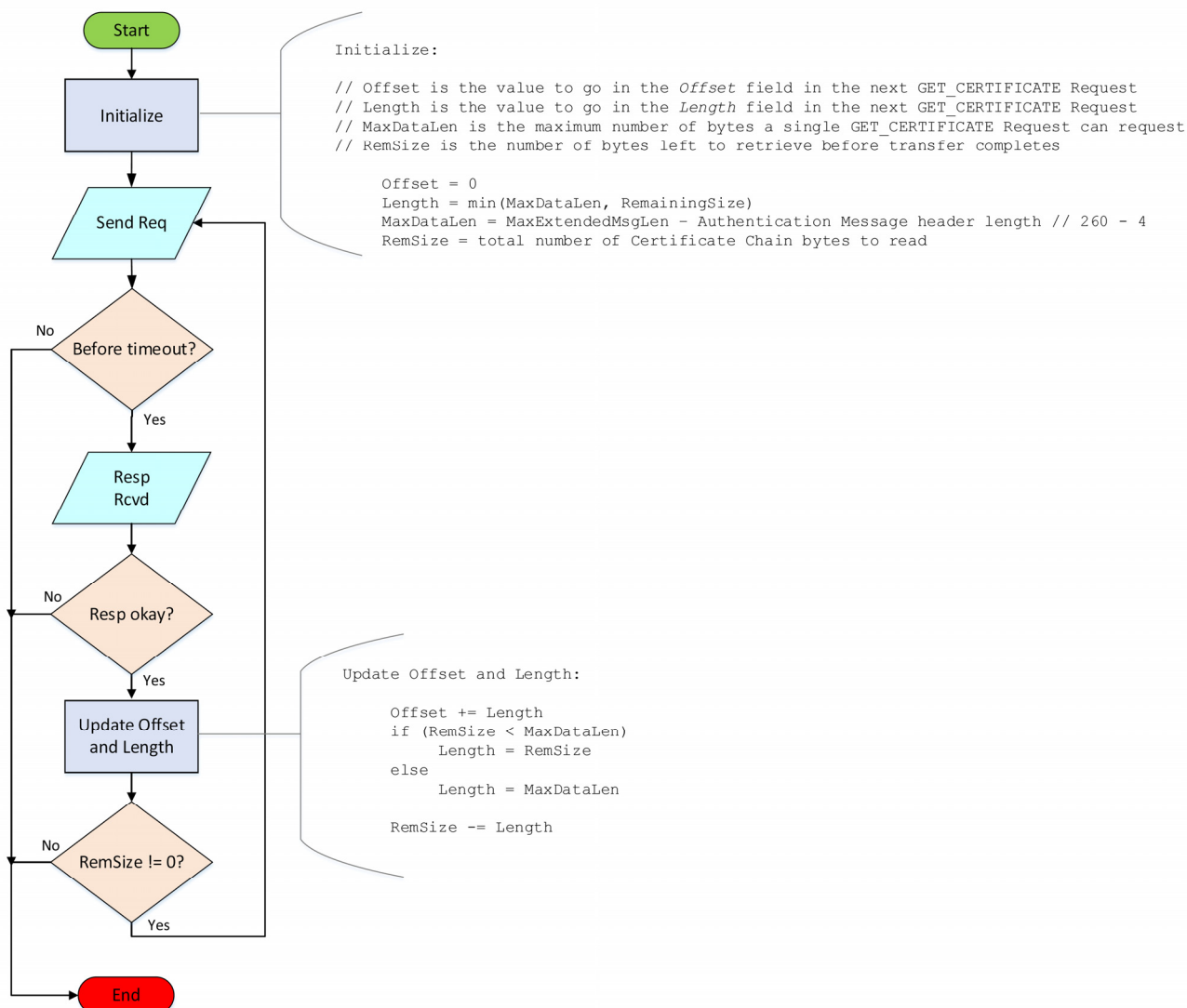
6.2 Übertragungen mit einer Länge größer als *MaxExtendedMsgLen*

Für Sicherheitsübertragungen, bei denen die Länge der Authentifizierungsantwort größer als *MaxExtendedMsgLen* ist, werden mehrere PD-Nachrichtenaustausche benötigt, um in das PN-Nachrichtensystem zu passen.

Ein Authentifizierungs-Initiator muss eine Sicherheitsübertragung in mehrere Authentifizierungsnachrichten unterteilen, die *MaxExtendedMsgLen* jeweils nicht überschreiten. Dies geschieht, indem ein Authentifizierungsinitiator eine Reihe von Authentifizierungsanforderungen sendet, die auf ein fortlaufendes Segment der zu lesenden Sicherheitsdaten abzielen. Alle Authentifizierungsanforderungen werden im SRQDB einer

Security_Request-PD-Nachricht gesendet. Alle Authentifizierungsantworten werden im SRPDB einer Security_Response-PD-Nachricht gesendet.

Bild 6-1 ist beispielhaft dargestellt, wie ein Authentifizierungs-Initiator eine Sicherheitsübertragung unterteilen und durchführen kann, wenn die Anzahl der in der Zertifikatskette eines Authentifizierungs-Beantworters gelesenen Bytes MaxExtendedMsgLen überschreitet.



Text im Bild:

englisch

Start

Initialize

Send Req

No

Before timeout?

Yes

Resp Rcvd

Resp okay?

Update Offset and Length

RemSize != 0?

End

Initialize :

// Offset is the value to go in the Offset field in the next GET_CERTIFICATE Request

deutsch

Start

Initialisieren

Send Req

Nein

Vor Timeout?

Ja

Resp Rcvd

Resp okay?

Offset und Länge aktualisieren

RemSize != 0?

Ende

Initialisieren:

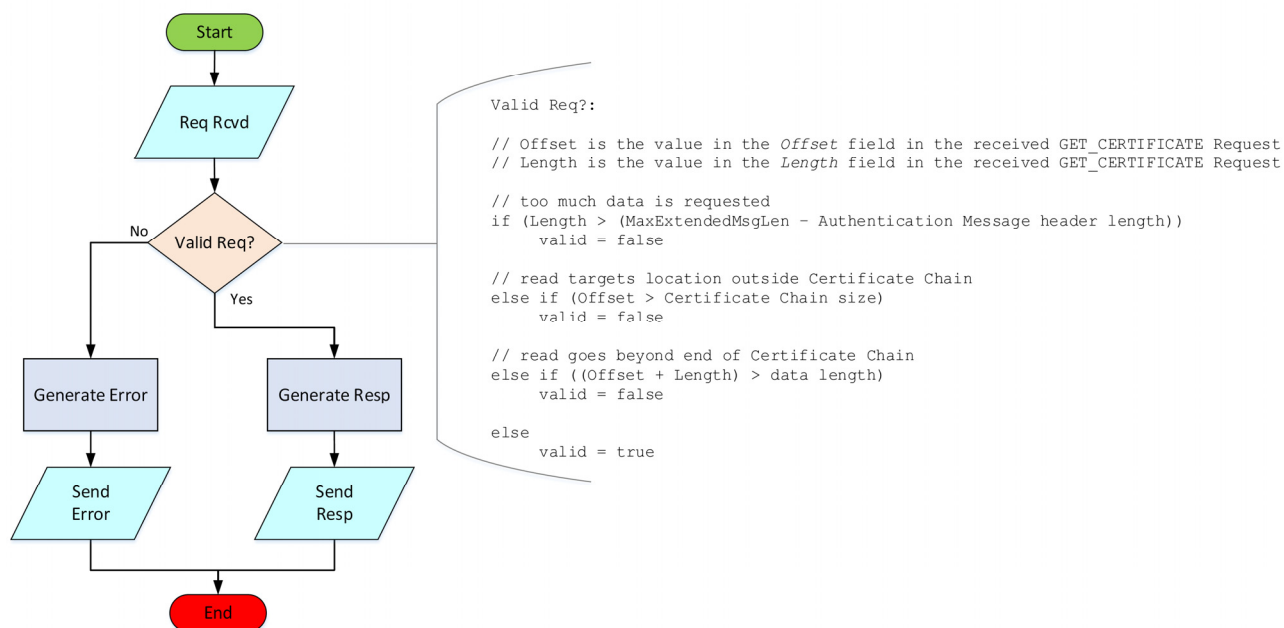
// Offset ist der Wert, mit dem das Offset-Feld der nächsten GET_CERTIFICATE-Anforderung befüllt wird

<pre>// Length is the value to go in the Length field in the next GET_CERTIFICATE Request // MaxDataLen is the maximum number of bytes a single GET_CERTIFICATE Request can request // RemSize is the number of bytes left to retrieve before transfer completes Offset = 0 Length = min(MaxDataLen, RemainingSize) MaxDataLen = MaxExtendedMsgLen - Authentication Message header length // 260 - 4 RemSize = total number of Certificate Chain bytes to read Update Offset and Length: Offset += Length if (RemSize < MaxDataLen) Length = RemSize else Length = MaxDataLen RemSize -= Length</pre>	<pre>// Length ist der Wert, mit dem das Length-Feld der nächsten GET_CERTIFICATE-Anforderung befüllt wird // MaxDataLen ist die höchste Anzahl von Bytes, die eine einzelne GET_CERTIFICATE- Anforderung anfordern kann // RemSize ist die Anzahl von Bytes, die vor Abschluss der Übertragung noch abgefragt werden müssen Offset = 0 Length = min(MaxDataLen, RemainingSize) MaxDataLen = MaxExtendedMsgLen - Authentication Message header length // 260 - 4 RemSize = total number of Certificate Chain bytes to read Offset und Length aktualisieren: Offset += Length if (RemSize < MaxDataLen) Length = RemSize else Length = MaxDataLen RemSize -= Length</pre>
---	--

ANMERKUNG Die Timeout-Werte sind in Tabelle 6-1 aufgeführt.

Bild 6-1 – Beispiel für einen Sicherheitsübertragungsvorgangs vonseiten eines Authentifizierungs-Initiators

In Bild 6-2 ist beispielhaft dargestellt, wie ein Authentifizierungs-Beantworter auf eine GET_CERTIFICATE-Anforderung antworten kann. Die Kriterien, anhand derer ein Authentifizierungs-Beantworter bestimmt, ob eine bestimmte GET_CERTIFICATE-Anforderung gültig ist, sind im Pseudocode rechts neben dem Bild aufgeführt.



Text im Bild:

englisch

Start

Req Rcvd

No

Valid Req?

Yes

Generate Error

Send Error

End

Generate Resp

Send Resp

Valid Req?:

// Offset is the value in the Offset field in the received GET_CERTIFICATE Request

// Length is the value in the Length field in the received GET_CERTIFICATE Request

// too much data is requested

// read targets location outside Certificate Chain

else if (Offset > Certificate Chain size)

valid = false

// read goes beyond end of Certificate Chain

deutsch

Start

Req Rcvd

Nein

Valid Req?

Ja

Fehler generieren

Fehler senden

Ende

Generate Resp

Send Resp

Valid Req?:

// Offset ist der Wert im Offset-Feld der erhaltenen GET_CERTIFICATE-Anforderung

// Länge ist der Wert im Length-Feld der erhaltenen GET_CERTIFICATE-Anforderung

// zu viele Daten angefordert

// es soll eine Stelle außerhalb der Zertifikatskette gelesen werden

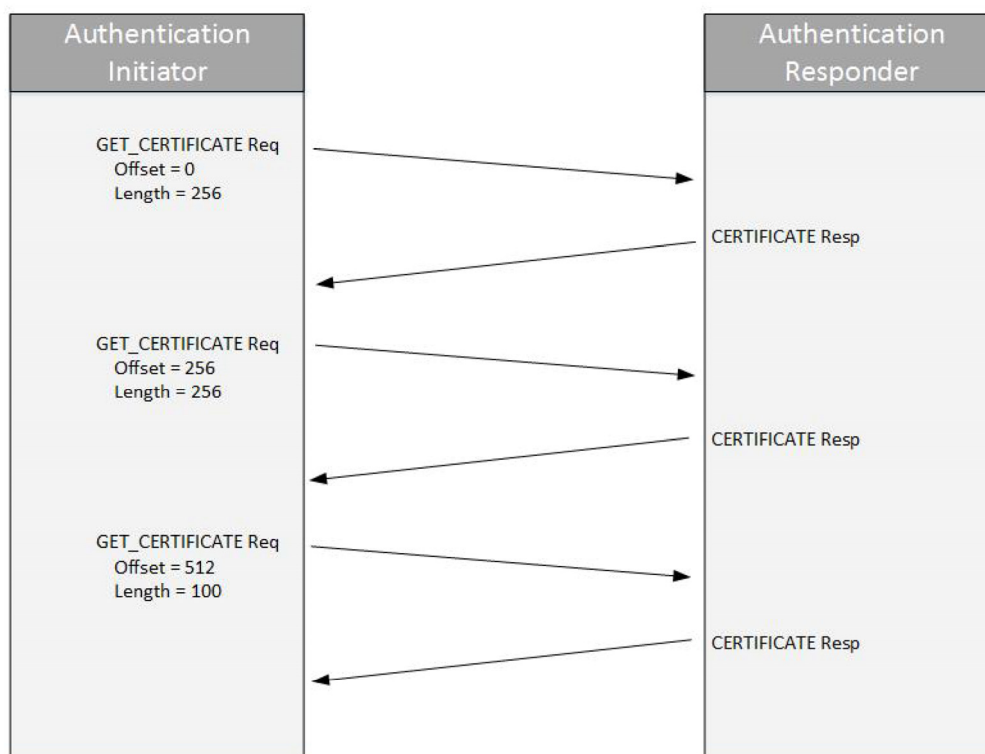
else if (Offset > Certificate Chain size)

valid = false

// es soll über das Ende der Zertifikatskette hinaus gelesen werden

In Bild 6-3 ist ein Beispiel für eine GET_CERTIFICATE-Anforderung für eine 612-Byte-Zertifikatskette unter Verwendung der Prozesse aus Bild 6-1 und Bild 6-2 dargestellt.

Bild 6-2 – Beispiel für einen Sicherheitsübertragungsvorgangs vonseiten eines Authentifizierungs-Beantworters



Text im Bild:

englisch

Authentication Initiator

GET_CERTIFICATE Req

Offset = 0

Length = 256

Authentication Responder

CERTIFICATE Resp

deutsch

Authentifizierungs-Initiator

GET_CERTIFICATE-Anforderung

Offset = 0

Länge = 256

Authentifizierungs-Beantworter

CERTIFICATE-Antwort

Bild 6-3 – Beispiel für das Lesen einer 612-Byte-Zertifikatskette

6.3 Zeitvorgaben für erweiterte PD-Sicherheitsnachrichten

6.3.1 Authentifizierungs-Initiator

In Tabelle 6-1 sind die Timeout-Werte für einen Authentifizierungs-Initiator angegeben. Die Timeout-Werte müssen ab dem Zeitpunkt gemessen werden, zu dem das letzte Bit des *EOP* der *GoodCRC*-Nachricht, die der Authentifizierungsanforderungs-Nachricht entspricht, von der PD-Bitübertragungsschicht empfangen wurde, bis zu dem Zeitpunkt, zu dem das letzte Bit des *EOP* der Authentifizierungsantwort-Nachricht von der PD-Bitübertragungsschicht empfangen wurde.

Tabelle 6-1 – Timeout-Werte für einen PD-Authentifizierungs-Initiator

Parameter	Timeout-Wert		Beschreibung
	Nicht segmentiert	Segmentiert	
<i>tDigestRcvd</i>	40 ms	200 ms	Timeout für eine GET_DIGESTS-Authentifizierungsanforderung.
<i>tCertRcvd</i>	40 ms	200 ms	Timeout für eine GET_CERTIFICATE-Authentifizierungsanforderung.
<i>tChallengeAuthRcvd</i>	1000 ms	1200 ms	Timeout für eine CHALLENGE-Authentifizierungsanforderung.

Erhält ein Authentifizierungs-Initiator innerhalb von *tDigestRcvd* nach Absenden einer GET_DIGESTS-Authentifizierungsanforderung keine Authentifizierungsantwort, wird von einem Fehler ausgegangen. Die anschließende Fehlerbehandlung liegt außerhalb des Anwendungsbereichs dieser Spezifikation.

Erhält ein Authentifizierungs-Initiator innerhalb von *tCertRcvd* nach Absenden einer GET_CERTIFICATE-Authentifizierungsanforderung an einen Authentifizierungs-Beantworter keine Authentifizierungsantwort, wird von einem Fehler ausgegangen. Die anschließende Fehlerbehandlung liegt außerhalb des Anwendungsbereichs dieser Spezifikation.

Erhält ein Authentifizierungs-Initiator innerhalb von *tChallengeAuthRcvd* nach Absenden einer CHALLENGE-Authentifizierungsanforderung keine Authentifizierungsantwort, wird von einem Fehler ausgegangen. Die anschließende Fehlerbehandlung liegt außerhalb des Anwendungsbereichs dieser Spezifikation.

6.3.2 Authentifizierungs-Beantworter

In Tabelle 6-2 sind die Zeitvorgaben angegeben, die ein Authentifizierungs-Beantworter einhalten muss. Die Zeitvorgaben müssen ab dem Zeitpunkt gemessen werden, zu dem das letzte Bit des **EOP** der Authentifizierungsanforderungs-Nachricht von der PD-Bitübertragungsschicht empfangen wurde, bis zu dem Zeitpunkt, zu dem das letzte Bit des **EOP** der **GoodCRC**-Nachricht, die der Authentifizierungsantwort-Nachricht entspricht, von der PD-Bitübertragungsschicht empfangen wurde.

Tabelle 6-2 – Zeitvorgaben für PD-Authentifizierungs-Beantworter

Parameter	Zeitvorgabewert		Beschreibung
	Nicht segmentiert	Segmentiert	
<i>tDigestSent</i>	20 ms	135 ms	Maximale Zeit zwischen dem Empfang einer GET_DIGESTS-Authentifizierungsanforderung und dem Versand einer Authentifizierungsantwort.
<i>tCertSent</i>	20 ms	135 ms	Maximale Zeit zwischen dem Empfang einer GET_CERTIFICATE-Authentifizierungsanforderung und dem Versand einer Authentifizierungsantwort.
<i>tChallengeAuthSent</i>	520 ms	635 ms	Maximale Zeit zwischen dem Empfang einer CHALLENGE-Authentifizierungsanforderung und dem Versand einer Authentifizierungsantwort.

Eine Authentifizierungsantwort muss innerhalb von *tDigestSent* nach Erhalt einer GET_DIGESTS-Authentifizierungsanforderung gesendet werden.

Eine Authentifizierungsantwort muss innerhalb von *tCertSent* nach Erhalt einer GET_CERTIFICATE-Authentifizierungsanforderung gesendet werden.

Eine Authentifizierungsanforderung muss innerhalb von *tChallengeAuthSent* nach Erhalt einer CHALLENGE-Authentifizierungsanforderung gesendet werden.

6.4 Context Hash

Das *Context Hash*-Feld in einer CHALLENGE_AUTH-Authentifizierungsantwort muss bei PD-Quellen, -Verbrauchern und -Kabelsteckern den Wert Null einnehmen. Auch bei PD-Geräten im Alternativmodus muss dieses Feld auf Null gesetzt sein.

7 Authentifizierung von USB-Produkten

Ein USB-Host kann ein USB-Gerät auf Grundlage der in dieser Spezifikation festgelegten Architektur und Methoden authentifizieren. Ein USB-Host, der ein USB-Gerät authentifiziert, übernimmt die Rolle des Authentifizierungs-Initiators und das USB-Gerät die Rolle des Authentifizierungs-Beantworters. Ein USB-Gerät darf nicht Authentifizierungs-Initiator sein.

In diesem Abschnitt werden die weiteren Voraussetzungen für die Authentifizierung von USB-Geräten beschrieben.

7.1 Deskriptoren

Die in diesem Abschnitt verwendeten Deskriptoren beschreiben die Authentifizierungsfähigkeiten eines USB-Geräts.

7.1.1 Deskriptor für Authentifizierungsfähigkeiten

Dieser Deskriptor wird verwendet, um die Authentifizierungsfähigkeiten und -eigenschaften zu beschreiben, die ein USB-Gerät unterstützt. Dieser Deskriptor muss als Teil des BOS-Deskriptorensatzes für ein USB-Gerät zurückgesendet werden, das Authentifizierung unterstützt.

Tabelle 7-1 – Deskriptor für Authentifizierungsfähigkeiten

Offset	Feld	Größe	Wert	Beschreibung								
0	<i>bLength</i>	1	Zahl	Größe des Deskriptors in Byte								
1	<i>bDescriptorType</i>	1	Konstante	Deskriptortyp: DEVICE CAPABILITY (16)								
2	<i>bDevCapabilityType</i>	1	Zahl	AUTHENTICATION								
3	<i>bmAttributes</i>	1	Bitmap	Bitmap-Codierung unterstützter Eigenschaften. <table><tr><th>Bit</th><th>Beschreibung</th></tr><tr><td>0</td><td>Muss auf 1 gesetzt sein, um anzugeben, dass Firmware aktualisiert werden kann. Anderenfalls muss es auf Null gesetzt sein.</td></tr><tr><td>1</td><td>Muss auf 1 gesetzt sein, um anzugeben, dass sich die Schnittstelle des Geräts bei der Aktualisierung ändert. Anderenfalls muss es auf Null gesetzt sein.</td></tr><tr><td>7:2</td><td>Reserviert. Muss auf Null gesetzt sein.</td></tr></table>	Bit	Beschreibung	0	Muss auf 1 gesetzt sein, um anzugeben, dass Firmware aktualisiert werden kann. Anderenfalls muss es auf Null gesetzt sein.	1	Muss auf 1 gesetzt sein, um anzugeben, dass sich die Schnittstelle des Geräts bei der Aktualisierung ändert. Anderenfalls muss es auf Null gesetzt sein.	7:2	Reserviert. Muss auf Null gesetzt sein.
Bit	Beschreibung											
0	Muss auf 1 gesetzt sein, um anzugeben, dass Firmware aktualisiert werden kann. Anderenfalls muss es auf Null gesetzt sein.											
1	Muss auf 1 gesetzt sein, um anzugeben, dass sich die Schnittstelle des Geräts bei der Aktualisierung ändert. Anderenfalls muss es auf Null gesetzt sein.											
7:2	Reserviert. Muss auf Null gesetzt sein.											
4	<i>bcdProtocolVersion</i>	1	Zahl	Muss auf den Wert der USB-Typ-C-Authentifizierungsprotokollversion gesetzt sein								
5	<i>bcdCapability</i>	1	Zahl	Muss auf den gleichen Wert gesetzt sein wie Param1 in einer DIGESTS-Authentifizierungsantwort								

Tabelle 7-2 – Deskriptortypen für Authentifizierungsfähigkeiten

Deskriptortyp für Authentifizierungsfähigkeiten	Wert
AUTHENTICATION	0Eh

7.2 Zuordnung von Authentifizierungsnachrichten in USB

Authentifizierungsnachrichten werden mittels der in diesem Abschnitt festgelegten Steuerungsanforderungen über USB übertragen. Zwei weitere zur USB-Authentifizierung verwendete USB-Standard-Anforderungscodes sind in Tabelle 7-3 festgelegt.

Tabelle 7-3 – bRequest-Werte in Authentifizierungsnachrichten

bRequest in Authentifizierungsnachrichten	Wert
AUTH_IN	18h
AUTH_OUT	19h

7.2.1 Authentication IN

Diese Steuerungsanforderung wird von einem USB-Host verwendet, um eine Authentifizierungsnachricht ohne Nutzdaten an ein USB-Gerät zu übertragen. Diese Anforderung leitet eine eingehende Steuerungsübertragung ein.

Die Felder *wValue* und *wIndex* eines Authentication-IN-Steuerungspakets enthalten den 4 Byte langen Authentifizierungsnachrichtenkopf, der in 5.1 beschrieben wird. In Tabelle 7-4 ist die Zuordnung eines Authentifizierungsnachrichtenkopfes zu den Feldern *wValue* und *wIndex* dargestellt.

Tabelle 7-4 – Felder einer Authentication-IN-Steuerungsanforderung

bmRequestType	bRequest	wValue	wIndex	wLength	Daten
10000000b	AUTH_IN	Authentifizierungsnachrichten-kopf		Variabel	Variabel

Tabelle 7-5 – Zuordnung des Authentifizierungsnachrichtenkopfes

Offset	Feld	Größe	Wert	Beschreibung
0	<i>ProtocolVersion</i>	1	Zahl	Ist dem High-Byte von <i>wValue</i> zugeordnet
1	<i>MessageType</i>	1	Zahl	Ist dem Low-Byte von <i>wValue</i> zugeordnet
2	<i>Param1</i>	1	Zahl	Ist dem High-Byte von <i>wIndex</i> zugeordnet
3	<i>Param2</i>	1	Zahl	Ist dem Low-Byte von <i>wIndex</i> zugeordnet

Das Verhalten eines USB-Geräts ist nicht festgelegt, wenn *wLength*, *wIndex* oder *wValue* für einen bestimmten Antworttyp nicht mit den in diesem Abschnitt festgelegten Werten übereinstimmen. Anderenfalls wird für den jeweiligen Zustand des USB-Geräts folgendes Verhalten erwartet:

Default-Wert	Nicht festgelegt.
Adresse	Anforderung ist gültig und Gerät muss antworten.
Konfiguriert	Anforderung ist ungültig und Gerät muss mit Anforderungsfehler antworten.

7.2.2 Authentication OUT

Diese Steuerungsanforderung wird von einem USB-Host verwendet, um eine Authentifizierungsnachricht mit Nutzdaten an ein USB-Gerät zu übertragen. Diese Anforderung leitet eine ausgehende Steuerungsübertragung ein.

Tabelle 7-6 – Felder einer Authentication-OUT-Steuerungsanforderung

bmRequestType	bRequest	wValue	wIndex	wLength	Daten
00000000b	AUTH_OUT	Authentifizierungsnachrichtenkopf		Variabel	Variabel

Die Felder *wValue* und *wIndex* eines Authentication-OUT-Steuerungspakets enthalten den 4 Byte langen Authentifizierungsnachrichtenkopf, der in 5.1 beschrieben wird. In Tabelle 7-5 ist die Zuordnung eines Authentifizierungsnachrichtenkopfes zu den Feldern *wValue* und *wIndex* dargestellt.

Das Verhalten eines USB-Geräts ist nicht festgelegt, wenn *wLength*, *wIndex* oder *wValue* für einen bestimmten Authentifizierungsanforderungstyp nicht mit den in diesem Abschnitt festgelegten Werten übereinstimmen. Anderenfalls wird für den jeweiligen Zustand des USB-Geräts folgendes Verhalten erwartet:

Default-Wert	Nicht festgelegt.
Adresse	Anforderung ist gültig und Gerät muss antworten.
Konfiguriert	Anforderung ist ungültig und Gerät muss mit Anforderungsfehler antworten.

7.3 Authentifizierungs-Protokoll

Ein USB-Host kann die in Abschnitt 4 beschriebenen Authentifizierungsvorgänge durchführen. In diesem Abschnitt wird beschrieben, wie die jeweiligen Vorgänge durchgeführt werden.

7.3.1 Digest-Abfrage

Ein USB-Host nutzt die in 5.2.1 festgelegte GET_DIGESTS-Anforderung, um Zertifikatsketten-Digests von einem USB-Gerät abzufragen. Um eine GET_DIGESTS-Anforderung zu senden, leitet ein USB-Host eine AUTH_IN-Steuerungsübertragung mit den in Tabelle 7-7 dargestellten Werten ein.

Tabelle 7-7 – Felder einer GET_DIGESTS-Authentication-IN-Steuerungsanforderung

bmRequestType	bRequest	wValue	wIndex	wLength	Daten
10000000b	AUTH_IN	Siehe Tabelle 5-4 bezüglich Werten und Tabelle 7-5 bezüglich Zuordnung.		260	DIGESTS-Antwort

Als Antwort auf eine GET_DIGESTS-Anforderung sendet ein USB-Gerät entweder eine DIGESTS-Antwort wie in 5.3.1 beschrieben oder eine ERROR-Antwort wie in 5.3.4 beschrieben zurück.

7.3.2 Lesen eines Zertifikats

Ein USB-Host nutzt eine AUTH_OUT-Steuerungsübertragung gefolgt von einer AUTH_IN-Steuerungsübertragung, um eine Zertifikatskette oder einen Teil davon aus einem USB-Gerät zu lesen. Die AUTH_OUT-Steuerungsübertragung enthält eine GET_CERTIFICATE-Anforderung wie in Tabelle 7-8 dargestellt.

Tabelle 7-8 – Felder einer GET_CERTIFICATE-Authentication-OUT-Steuerungsanforderung

bmRequestType	bRequest	wValue	wIndex	wLength	Daten
00000000b	AUTH_OUT	Siehe Tabelle 5-5 bezüglich Werten und Tabelle 7-5 bezüglich Zuordnung.		4	Siehe Tabelle 5-6

Nach erfolgreicher Durchführung der AUTH_OUT-Steuerungsübertragung leitet der USB-Host eine AUTH_IN-Steuerungsübertragung ein, um die CERTIFICATE-Antwort vom USB-Gerät abzufragen. Die AUTH_IN-Steuerungsübertragung enthält die in Tabelle 7-9 angegebenen Werte, wobei das High-Byte von *wValue* das *ProtocolVersion*-Feld enthält, das Low-Byte von *wValue* das in Tabelle 5-12 angegebene *MessageType*-Feld enthält und das *wLength*-Feld in der GET_CERTIFICATE-Antwort den in der vorherigen AUTH_OUT-Steuerungsübertragung enthaltenen *Length*-Wert plus 4 enthält (um den Authentifizierungsnachrichtenkopf zu berücksichtigen).

Tabelle 7-9 – Felder einer CERTIFICATE-Authentication-IN-Steuerungsanforderung

bmRequestType	bRequest	wValue	wIndex	wLength	Daten
10000000b	AUTH_IN	Siehe Tabelle 5-12	0	Länge + 4	CERTIFICATE-Antwort

Als Antwort auf die oben beschriebene AUTH_IN-Steuerungsübertragung sendet ein USB-Gerät entweder eine CERTIFICATE-Antwort wie in 5.3.2 beschrieben oder eine ERROR-Antwort wie in 5.3.4 beschrieben zurück.

7.3.3 Authentifizierungs-Challenge

Ein USB-Host nutzt eine AUTH_OUT-Steuerungsübertragung gefolgt von einer AUTH_IN-Steuerungsübertragung, um ein USB-Gerät zu authentifizieren. Die AUTH_OUT-Steuerungsübertragung enthält eine CHALLENGE-Anforderung wie in Tabelle 7-10 dargestellt.