OSCI-Bibliothek (JAVA) – Versionshistorie

Version	Datum	Änderungen gegenüber Vorversion
0.96 Beta	24.11.2003	./.
0.97 Beta	08.12.2003	Überarbeitung der Schnittstellen der Bibliothek (kleinere Ergänzungen, Entfernung der `DialogFinder`-Schnittstelle und der Archivierungsfunktionen durch die `OSCIDataSource`-Schnittstelle), div. Bugfixes
0.98 Beta	15.12.2003	Implementierung MIME-Parser, Überarbeitung Schemadefinition, AES-Einbindung, `StoreMessage`-Klasse hinzugefügt, div. Bugfixes
1.00	09.03.2004	Verbesserte Unterstützung des Sprachinterface, Unterstützung der `Progress`-Schnittstelle, Fehlerbehandlung verbessert, Dokumentation qualitätsgesichert, div. Bugfixes
1.01	02.04.2004	ID-Schreibweise in XML-Dokumenten korrigiert; bei paralleler Mehrfachsignatur mit identischem Zertifikat wird nun jede Signatur geprüft; kleinere Korrekturen vorgenommen
1.02	30.04.2004	Probleme bei Benutzung ohne Debug-Einstellung beseitigt
1.03	23.06.2004	Korrigenda vom 10.06.2004 zur OSCI-Spezifikation 1.2 eingearbeitet, `ContentPackageIInterface` erweitert, Zertifikatsreferenzen vereinheitlicht, Bugfixes, ProcessCard-Operationen erweitert
1.04	08.07.2004	Dialogeröffnung mit Test-Intermediär nur via `InitDialog`-Nachricht möglich, `FeedbackObject`-Klasse eingeführt, die Methode `ContentContainer.checkAllSignatures()` wirft nun bei Nichtvorhandensein einer Signatur eine Exception, `MediateDelivery` erlaubt nun auch Message-IDs ohne Subject (laut Spezifikation), Default bei `QualityOfTimeStamp`-Eigenschaften ist nun `plain`
1.1	06.12.2004	Diverse Bugfixes, z.B. ein Problem mit Attachments, die in verschachtelten Content-Containern referenziert werden, beseitigt. Laufzettelabholaufträge für mehrere Message-IDs erweitert. Diverse Anpassungen und Änderungen mit dem Ziel der Kompatibilität mit Apache-XML Implementierungen. Base64-Codierung der Attachments als Transfer-Encoding. Transport-Interface um Methode `newInstance()` erweitert (Thread-Sicherheit in „automatischen“ Clients)
1.1.1	14.07.2005	Das Schließen von HTTP-Streams wurde verbessert. Schließen der Input-Streams in `Content` wurde überarbeitet. Ein Problem beim Parsen von `ResponseToProcessDelivery` bei fehlendem `Originator` o. `Addressee` wurde beseitigt und das Lesen von SOAP-Error-Nachrichten (mit `StoreInputStreams`), die keine Schema-Definition enthalten, wurde verbessert. Es wurde ein Problem mit abgeleiteten Rollenklassen behoben. `Inspections` codieren die DNs nun XML-konform (nur für Intermediär). Es wird nur noch die Key-Usage `non-repudiation` gecheckt und nun auch die Key-Usage `digital signature` zugelassen. Kollisionen von Referenz-IDs (z.B. mehrere `author0_*`) durch importierte

Version	Datum	Änderungen gegenüber Vorversion
		Content-Container wurden beseitigt.
1.2	03.02.2006	Es wurden zahlreiche Änderungen mit dem Ziel der Kompatibilität zu kommerziellen XML-Implementierungen vorgenommen. Hervorzuheben ist hier die Entfernung von Leerzeilen in dem Attachment MIME-Container. Es wurden Änderungen vorgenommen, um die Namespace-Präfixe variabel zu gestalten. Grenzen in der Realisierung haben sich durch XML-Signature ergeben, da ansonsten Signaturen unverschlüsselter Inhaltsdaten bei Änderungen der Präfixe ungültig würden. Temporäre Dateien werden nun beim Beenden der JRE gelöscht. Die Java-Bibliothek unterstützt nun den Längenparameter, der an die `getConnection`-Methode des Transport-Interfaces übergeben wird. Die Länge der Nachricht wird vorher berechnet. Die Base64-Codierung für den Transportumschlag und Attachments ist nun optional, bleibt jedoch Default. Diese Default-Belegung sollte nur bedacht geändert werden, da sich hieraus Inkompatibilitäten bei der Kommunikation mit vorherigen Versionen der OSCI-Bibliothek (und Intermediären) ergeben. Im `DialogHandler` können nun mehrere Default-Supplier, also mehrere Privatschlüssel, für die Entschlüsselung eingehender Nachrichten (und ggf. Signatur der Antworten) gesetzt werden.
1.2.1	03.03.2006	Kleinere Probleme der Version 1.2 beseitigt (Längenberechnung verschlüsselter Nachrichten, Reihenfolge der Content-Container-/`EncryptedData` –Tags, Namespace-Deklarationen in zusätzlichen SOAP-Headern)
1.2.2	14.06.2006	Problem mit signierten verschachtelten Content-Containern, die Attachments enthalten, behoben. Verlust des Base64-Transformers der Inhaltsdatensignatur beim Laden gespeicherter Nachrichten beseitigt. Neue Methode zum Laden gespeicherter Nachrichten mit Prüfung der Nachrichtensignatur in der Klasse `StoredMessage`. Neue Methoden in der Klasse `OSCIMessage` zum Zugriff auf Content-Container und Content-Objekte anhand des `refID`-Attributs.
1.2.3	06.12.2006	Methode `SwapBuffer.setTmpDir(String)` zum Setzen des Verzeichnisses für temporäre Dateien hinzugefügt. Neue abstrakte Subklasse von `OSCIDataSource (OSCIDataSourceExt123)` zur Unterstützung der Verschlüsselung temporärer ggf. vertraulicher Inhaltsdaten.
1.2.4	27.02.2007	Zusätzliches Attribut `mimeHeaders` in der Attachment-Klasse für weitere Header-Einträge des umschließenden MIME-Boundary-Abschnitts
1.2.5	17.04.2007	Überarbeitung der Verwendung gesetzter Security-Provider (durchgängige Nutzung des Providers für alle kryptographischen Operationen außer der Zufallszahlenerzeugung). Kleinere Änderungen, z.B. Exception-Handling der XML-Parser.
1.2.6	05.09.2007	Begrenzung der Attachment-Größe auf 2 GB beseitigt. Problem mit IBM-JDK behoben. `Base64InputStream.available()`-

Version	Datum	Änderungen gegenüber Vorversion
		Methode implementiert. Kleinere Änderungen, z.B: Behandlung der Namespace-Deklaration von Signaturen beim Speichern von Nachrichten
1.3	12.2.2008	Erweiterung auf neue Hashalgorithmen gemäß Korrigenda der OSCI 1.2 Transport Spezifikation v. 11.2.2008. Hierzu Erweiterung des Signer-Interfaces um die Methode `getAlgorithm()`. Übergangsweise wird SHA-1 weiter unterstützt. Zur Kontrolle der Signaturstärke empfangener Signaturen zwei neue Methoden `OSCIMessage.hasWeakSignature(Date)` und `ContentContainer. hasWeakSignature(Role, Date)`
1.3.1	15.10.2008	Workaround f. Bug #6219755 im JDK 1.5 (Ansammlung von Threads) im Serverbetrieb. Methode `ContentContainer.getSignatures()` zum Ermitteln der Signatur- und Hashalgorithmen von Inhaltsdatensignaturen. Neues Attribut `OSCIMessage` in `OSCIErrorException`. Problem mit ungültigen Nachrichtensignaturen bei seriell verschlüsselten Inhaltsdaten behoben. Default-Algorithmus für symmetrische Verschlüsselung auf AES-256 umgestellt.
1.3.2	03.06.2010	Anpassung an Ablauf des Hashalgorithmus RIPEMD-160 zum 31.12.2010. **Die Prüfmethoden** `hasWeakSignature(...)` **liefern bei Prüfung ohne Datumsangabe für diesen Algorithmus** `true`. Methoden für frei formulierte Feedback-Texte in synchronen Szenarien auf Empfängerseite. Kleinere Überarbeitungen.
1.4	10.11.2010	Vorbereitung für die Aufnahme von Signaturzeitpunkten in Inhaltsdatensignaturen (rudimentäre XAdES-Unterstützung). Diverse kleinere Überarbeitungen.
1.5	19.11.2010	Unterstützung elliptischer Kurven als Signaturalgorithmen (Korrigenda v. Oktober 2011). Kleinere Überarbeitungen.
1.6	16.04.2014	Unterstützung von OAEP-Algorithmen für Verschlüsselung und Signaturen gemäß Korrigenda der OSCI 1.2 Transport Spezifikation v. 20.2.2014. **Implementierungen der Schnittstelle** `de.osci.osci12.extinterfaces.crypto.Decrypter` **müssen für die Verwendung von RSAES-OAEP-ENCRYPT um eine entsprechende Methode erweitert werden.**
1.6.1	30.09.2015	Bug-Fixes in der OSCI-Bibliothek, da in bestimmten Situationen OSCI-Fehlermeldungen auf Nachrichtenebene ausgegeben wurden (OSCI_9300).
1.7	22.02.2017	Umstellung auf die neuere Version des BouncyCastle-Security-Providers 1.55 und Einbinden von neueren Testzertifikaten. Es wird unterbunden, dass für mehrere Inhaltsdaten gleiche Referenz-IDs verwendet werden oder nachträglich dahingehend geändert werden können. Einbindung der SHA3-Hash-Algorithmen **Sicherheits-Update:** Anpassungen an den Prüfungen von XML-Dokumenten und Aufrufen des verwendeten XML-Parsers. Für die Verwendung des CBC-Modus sind serverseitig zusätzliche

Version	Datum	Änderungen gegenüber Vorversion
		Sicherheitsmaßnahmen nötig. Daher wurde ein neuer Modus (GCM) hinzugefügt, der langfristig den CBC-Modus ablösen wird.
1.7.1	03.03.2017	Die Überprüfung von mehrfach verwendeten Referenz-IDs kann durch das Setzen einer System Property gesteuert werden. **Bug-Fix**, da es unter bestimmten Umständen einen Fehler beim Abholen von OSCI-1.2-Nachrichten gab.
1.8	08.08.2017	Es wurden verschiedene Änderungen vorgenommen, um die paketierte Übertragung von Nachrichten, die in der Erweiterung der OSCI-Spezifikation bzgl. effizienter Datenübertragung beschrieben ist, zu ermöglichen. Es gibt Methoden um die zusätzlichen Header, wie die `FeatureDescription` und die `ChunkInformation`, zu erstellen. Außerdem gibt es Methoden um die neuen Nachrichtentypen `PartialStoreDelivery` und `PartialFetchDelivery` aufzubauen sowie die Response Nachrichten zu verarbeiten. Die Aufteilung einer großen Nachricht in Pakete wird an einem Beispiel demonstriert.
1.8.1	07.03.2018	Umstellung auf die aktuelle Version 1.59 des BouncyCastle-Security-Providers. Kleinere Verbesserungen und Fehlerbehebungen für die Funktionen zur Übertragung großer Nachrichten.
1.8.2	23.10.2018	Umstellung auf die neuere Version des BouncyCastle-Security-Providers 1.60 **Sicherheits-Update:** Es wurden Verbesserungen im MIME-Parser und bei der Strukturprüfung der Signaturen vorgenommen. Bei der Verwendung des CBC-Modus wird im Client-Dialog automatisch auf GCM gewechselt, wenn Client und Server diesen Modus unterstützen. Diese Funktion ist clientseitig abschaltbar.
1.8.3	20.11.2018	Problem beim gleichzeitigen parallelen Versenden von Nachrichten behoben. **Sicherheits-Update:** Weitere Verbesserungen bei der Strukturprüfung der OSCI-Nachricht und der Signaturen.
1.9.0 1.8.4-internal-2	29.01.2020	Verwendung von AES-256-GCM anstelle von AES-256-CBC als Standardmodus für die symmetrische Verschlüsselung Verwendung von RSA-OAEP anstelle von RSAES-PKCS1-v1_5 als Standardmodus für die asymmetrische Verschlüsselung Der Initialisierungsvektor bei Verwendung von AES-GCM lässt sich anpassen: 12 Byte bzw. 96 Bit, empfohlen gemäß XML-Encryption-Standard, oder 16 Byte bzw. 128 Bit Als Standardwert wird aus Gründen der Abwärtskompatibilität zu den Versionen 1.8.x ein Initialisierungsvektor von 16 Byte bzw. 128 Bit verwendet Umstellung auf die neuere Version des BouncyCastle-Security-Providers 1.62

Version	Datum	Änderungen gegenüber Vorversion
2.0.1	18.01.2021	Umstellung des Standardwerts für die Länge des Initialisierungsvektors bei der Verwendung von AES-GCM von 16 Byte bzw. 128 Bit auf den empfohlenen Wert 12 Byte bzw. 96 Bit Umstellung auf die neuere Version des BouncyCastle-Security-Providers 1.66 Unterstützung von JDK 11 (Oracle JDK, OpenJDK und Zulu-JDK) Element-Reihenfolge im Nachrichtentyp „ResponseToAcceptDelivery“ entspricht jetzt vollständig der Spezifikation Kleinere Optimierungen des Codes zur Verbesserung der Qualität
2.1.0	03.03.2021	Methode zum nachträglichen Anpassen der Länge des Initialisierungsvektors in Attachments hinzugefügt Verbesserungen einzelner Logmeldungen
2.1.1	25.04.2022	OSCI-Schema-Dateien hinzugefügt Verbesserung von zwei Textmeldungen Umstellung auf die neuere Version des BouncyCastle-Security-Providers 1.69
2.3.0	13.12.2022	Veralteter Algorithmus RSAES-PKCS1-v1_5 ist jetzt als „Deprecated“ markiert. Kleinere Optimierungen des Codes zur Verbesserung der Qualität. Neuer Beispiel-Code für das direkte Verarbeiten von EncryptedData- und NonIntermediaryCertificates-Elementen. Beispielzertifikate auf 4096-Bit-RSA-Schlüssel umgestellt. Umstellung auf die neuere Version des BouncyCastle-Security-Providers 1.71.
2.4.0	14.12.2022	Neue Methode in OSCI-Nachrichten eingefügt, um MessageMetaData-Objekte als Custom-Header hinzuzufügen (mit kommentiertem Beispiel-Code). Alle Quelltextdateien sind nun UTF-8-codiert.
2.4.1	10.10.2023	Datenpuffergröße für Schreibvorgänge an mehreren Stellen erhöht, um Performance zu verbessern (insbesondere für Clients, die auf Netzlaufwerke schreiben) OSCI-Nachrichten und Attachments werden standardmäßig ohne Base64-Codierung erzeugt Umstellung auf die neuere Version des BouncyCastle-Security-Providers 1.76