[MS-GRVDYNM]:
Groove Dynamics Protocol

Revision Summary

Date	Revision History	Revision Class	Comments
04/04/2008	0.1		Initial Availability
06/27/2008	1.0	Major	Revised and edited the technical content
12/12/2008	1.01	Editorial	Revised and edited the technical content
07/13/2009	1.02	Major	Changes made for template compliance
08/28/2009	1.03	Editorial	Revised and edited the technical content
11/06/2009	1.04	Editorial	Revised and edited the technical content
02/19/2010	2.0	Editorial	Revised and edited the technical content
03/31/2010	2.01	Editorial	Revised and edited the technical content
04/30/2010	2.02	Editorial	Revised and edited the technical content
06/07/2010	2.03	Editorial	Revised and edited the technical content
06/29/2010	2.04	Editorial	Changed language and formatting in the technical content.
07/23/2010	2.05	Minor	Clarified the meaning of the technical content.
09/27/2010	2.05	No change	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	2.05	No change	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	2.05	No change	No changes to the meaning, language, or formatting of the technical content.
03/18/2011	2.05	No change	No changes to the meaning, language, or formatting of the technical content.
06/10/2011	2.05	No change	No changes to the meaning, language, or formatting of the technical content.
01/20/2012	2.05	No change	No changes to the meaning, language, or formatting of the technical content.
04/11/2012	2.05	No change	No changes to the meaning, language, or formatting of the technical content.
07/16/2012	2.05	No change	No changes to the meaning, language, or formatting of the technical content.
10/08/2012	2.05	No change	No changes to the meaning, language, or formatting of the technical content.
02/11/2013	2.05	No change	No changes to the meaning, language, or formatting of

Date	Revision History	Revision Class	Comments
			the technical content.
07/30/2013	2.05	No change	No changes to the meaning, language, or formatting of the technical content.
11/18/2013	2.05	No change	No changes to the meaning, language, or formatting of the technical content.
02/10/2014	2.05	No change	No changes to the meaning, language, or formatting of the technical content.
04/30/2014	2.05	No change	No changes to the meaning, language, or formatting of the technical content.
07/31/2014	2.05	No change	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction

This document specifies the Groove Dynamics Protocol, an application-layer distributed protocol for consistently ordering operations on an arbitrary number of peers. This protocol consists of encoded XML messages used to synchronize data in a shared space.

Sections 1.8, 2, and 3 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in [RFC2119]. Sections 1.5 and 1.9 are also normative but does not contain those terms. All other sections and examples in this specification are informative.

1.1 Glossary

The following terms are defined in [MS-GLOS]:

Unicode
unique identifier (UID)

The following terms are defined in [MS-OFCGLOS]:

account
block
canonical URL
delta
delta log
dependency graph
endpoint
engine
HMAC-SHA1
identity URL
record
SHA-1
shared space
Simple Symmetric Transport Protocol (SSTP)

The following terms are specific to this document:

async delta: A delta that is sent to only a subset of the endpoints (3) in a shared space. An async delta does not have any dependent deltas.

sequence: A unique identifier for a delta that includes the user identifier for the endpoint (3) that created the delta.

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as described in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

References to Microsoft Open Specification documents do not include a publishing year because links are to the latest version of the documents, which are updated frequently. References to other documents include a publishing year when one is available.

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information.

[BCMO800-38A] National Institute of Standards and Technology, "Recommendation for Block Cipher Modes of Operation: Methods and Techniques", NIST Special Publication 800-38A, December 2001, http://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf

[FIPS197] FIPS PUBS, "Advanced Encryption Standard (AES)", FIPS PUB 197, November 2001, http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[IEEE1363a] Institute of Electrical and Electronics Engineers, "IEEE Standard Specifications for Public-Key Cryptography﷯Amendment 1: Additional Techniques", 1363a-2004, September 2004, http://ieeexplore.ieee.org/iel5/9276/29460/01335427.pdf

[MS-GRVSSTP] Microsoft Corporation, "Simple Symmetric Transport Protocol (SSTP)".

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, http://www.rfc-editor.org/rfc/rfc2119.txt

[RFC3174] Eastlake III, D., and Jones, P., "US Secure Hash Algorithm 1 (SHA1)", RFC 3174, September 2001, http://www.ietf.org/rfc/rfc3174.txt

[RFC4634] Eastlake III, D., and Hansen, T., "US Secure Hash Algorithms (SHA and HMAC-SHA)", RFC 4634, July 2006, http://www.ietf.org/rfc/rfc4634.txt

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, October 2006, http://www.ietf.org/rfc/rfc4648.txt

[WBXML1.2] Martin, B., and Jano, B., Eds., "WAP Binary XML Content Format", W3C Note, June 1999, http://www.w3.org/1999/06/NOTE-wbxml-19990624

1.2.2 Informative References

[MS-GLOS] Microsoft Corporation, "Windows Protocols Master Glossary".

[MS-GRVRDB] Microsoft Corporation, "Groove RDB Commands Protocol".

[MS-OFCGLOS] Microsoft Corporation, "Microsoft Office Master Glossary".

[MSR-TR-2003-60] Saito, Y. and Shapiro, M., "Optimistic Replication", September 2003, http://research.microsoft.com/research/pubs/view.aspx?type=Technical%20Report&id=681

[RFC2045] Freed, N., and Borenstein, N., "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996, http://ietf.org/rfc/rfc2045.txt

[XML10] World Wide Web Consortium, "Extensible Markup Language (XML) 1.0 (Third Edition)", February 2004, http://www.w3.org/TR/2004/REC-xml-20040204/

[XMLNS] Bray, T., Hollander, D., Layman, A., et al., Eds., "Namespaces in XML 1.0 (Third Edition)", W3C Recommendation, December 2009, http://www.w3.org/TR/2009/REC-xml-names-20091208/

1.3 Protocol Overview (Synopsis)

1.3.1 Synchronization

This protocol is used to synchronize information among endpoints (3) participating in a **shared space**. A shared space consists of a set of zero or more tools. Each tool has zero or more **engines**. Engines define a set of operations, or commands. Changes to the synchronized tool data are made by executing these commands. One or more commands are grouped into a **delta**, which is the unit of transactional consistency in a shared space. Dynamics guarantees that all of the commands in a delta are executed sequentially. Dynamics synchronizes the shared space by causing all deltas to be executed in the same order on all endpoints.

A typical example would be a shared space with a threaded discussion tool that would allow multiple endpoints to contribute discussion topics and post replies. This tool could be built using the record database (RDB) engine. RDB has a command set for manipulating **records**, which includes commands for adding and deleting records, and setting fields on existing records. Data consistency across all endpoints is achieved by using this protocol to sequence the execution of the commands.

A shared space can have an arbitrary number of members, but all examples in this document use a space with three members, who are referred to as A, B, and C. In examples, deltas will be designated by a letter indicating the endpoint that created the delta and a number that is the **sequence** of the delta generated on that endpoint. So A3 would be the third delta created by endpoint A. This is a simplification of the delta sequence numbering scheme which will be described later.

A simple scenario in which dynamics is used starts with the user at endpoint A creating a new discussion topic. This causes the RDB engine to create a command to add a new record. The command includes the title and contents of the discussion topic. A creates delta A1 containing the add record command. Dynamics sends A1 as a delta message to endpoints B and C. When those endpoints receive A1 they execute the commands in the delta, which causes the new record to be added and appear in the tool so that it can be read by the users on those endpoints.

In the event that different endpoints make independent changes, it is impossible to get a completely consistent order of execution of deltas on all endpoints. Dynamics achieves convergence by getting a logically equivalent order of execution on all endpoints. All engines implementing this protocol support the ability to undo all of their commands. In the event that a newly received delta requires that the delta order be changed, the commands in the previously executed deltas that are being reordered will be undone and then executed again in the proper order.

For example, consider what happens when two endpoints, A and B, create new discussion entries at the same time. This results in the creation of deltas A2 and B1. Dynamics will define the ordering of those deltas. This ordering will be consistent on all endpoints. In this case assume that B1 is before A2. When B receives A2, it will have already executed B1. Because A2 comes after B1 this isn't a problem and it can execute A2. However A will have already executed A2 when B1 is received. To get the correct logical ordering, it will need to undo A2, execute B1 and then execute A2.

Dynamics guarantees causal consistency in the order of execution of deltas. Causal consistency is the property that when an endpoint, A, executes a delta created by a different endpoint, B, A must have previously executed all normal deltas that B had executed when it created the new delta. For any new delta, any deltas that the creator of that delta had executed prior to creating the delta need to be executed on all endpoints before they execute the new delta. So if A has executed B2 and then creates A3, but C receives A3 before receiving B2, C waits to execute A3 until B2 is received and executed.

1.3.2 Messages

There are two messages in the dynamics protocol. Delta messages are used to send deltas to other endpoints in a shared space. **Delta Ack** messages are used to acknowledge the receipt of deltas.

Dynamics messages consist of a wrapper in a format similar to MIME, as described in [RFC2045]. This wraps the compressed, secured payload. This is an encoding, using a subset of WBXML, as described in [WBXML1.2], of the secure XML, as described in [XML10].

The secure XML uses XML namespaces, as described in [XMLNS], and contains the message contents. These contents are encrypted using the AES algorithm, as described in [FIPS197] in CTR mode, as described in [BCMO800-38A], with a per-space symmetric key. Encryption prevents anyone who is not a member of the shared space from reading the message. The message contents are also signed using the ESIGN algorithm, as described in [IEEE1363a]. The signature private key is unique for a single space and a single member. Signing is used to guarantee both message integrity and message authenticity.

Once the secure XML has been decrypted, the structure and attributes of the decrypted XML determine how dynamics processes the **Delta** or **Delta Ack** message.

1.4 Relationship to Other Protocols

This protocol depends on the **Simple Symmetric Transport Protocol (SSTP)** , as described in [MS-GRVSSTP].

Engine command protocols, such as the Groove RDB Commands Protocol, as described in [MS-GRVRDB], use this protocol as the transport for their messages.

This protocol also depends on WBXML, as described in [WBXML1.2], which it uses to compress its messages before disseminating them to other endpoints.

1.5 Prerequisites/Preconditions

This protocol operates within a shared space. It assumes that the shared space has already been created and that all endpoints in the shared space are running compatible implementations of the dynamics protocol. All engines in the space have a known engine URL that can be used to address commands to the engine. All endpoints in the space have a known device URL, **identity URL** and **unique identifier (UID)** endpoint UID.

The following security keys for the space are known:

- **Per-space master key:** This key is used to encrypt all messages. The key identifier and key version are known.
- **Per-space per-member signature private key for the current member:** This key is used to sign messages sent by the current member.
- **Per-space per-member signature public keys for all members:** These keys are used to verify signatures for messages sent by other members.

1.6 Applicability Statement

This protocol can be used anytime that operation-transfer peer-to-peer optimistic replication is necessary. For most replication problems, state-transfer optimistic replication is likely preferable. See [MSR-TR-2003-60] for a survey of replication algorithms. This protocol is only appropriate for problems that warrant the high complexity of the protocol.

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

Protocol messages MUST be transported using the SSTP protocol, as specified in [MS-GRVSSTP]. Endpoints MUST be identified by their device URL and identity URL. All messages MUST be addressed to a set of one or more endpoints. The resource URL `grooveDynamics://Dynamics/;CanTelURL=grooveTelespace:%2f%2fTelespacePathAsync` is the asynchronous resource URL. It MUST be used to send and receive async deltas and identity-disseminated deltas. The resource URL `grooveDynamics://Dynamics/;CanTelURL=grooveTelespace:%2f%2fTelespacePath` is the normal resource URL. It MUST be used to receive and send all other dynamics messages. In both URLs, *TelespacePath* MUST be the path of the telespace **canonical URL**.

2.2 Message Syntax

This protocol specifies the following types as XML attribute values:

Binary: A base64-encoded string representation of the binary data, as defined in [RFC4648].

Hex String: A hexadecimal string attribute MUST consist of characters in the ranges 0 through 9 and "A" through "F". Hexadecimal strings MUST be compared as hexadecimal numbers.

Int: An **Int** attribute MUST be a decimal string representation of an integer in the range 0 through 2,147,483,647.

Null: A **Null** attribute that MUST be an empty string.

String: A **Unicode** string.

2.2.1 Delta Message

2.2.1.1 MIME-like Wrapper

The following table shows the format of this wrapper.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
Header (153 bytes)																															
...																															
Compressed Secured Payload (variable)																															
...																															
Epilogue (19 bytes)																															
...																															

Header (153 bytes): The header MUST comprise the following bytes:

```
0000  4d 49 4d 45 2d 56 65 72 73 69 6f 6e 3a 20 31 2e  MIME-Version: 1.
0010  30 20 28 47 72 6f 6f 76 65 20 32 29 0d 0a 43 6f  0 (Groove 2)..Co
0020  6e 74 65 6e 74 2d 54 79 70 65 3a 20 6d 75 6c 74  ntent-Type: mult
0030  69 70 61 72 74 2f 72 65 6c 61 74 65 64 3b 20 62  ipart/related; b
0040  6f 75 6e 64 61 72 79 3d 22 3c 3c 5b 5b 26 26 26  oundary="<<[[&&&
0050  5d 5d 3e 3e 22 0d 0a 3c 3c 5b 5b 26 26 26 5d 5d  ]]>>"..<<[[&&&]]
0060  3e 3e 0d 0a 43 6f 6e 74 65 6e 74 2d 54 79 70 65  >>..Content-Type
0070  3a 20 61 70 70 6c 69 63 61 74 69 6f 6e 2f 57 42  : application/WB
0080  58 4d 4c 3b 20 63 68 61 72 73 65 74 3d 22 75 73  XML; charset="us
0090  2d 61 73 63 69 69 22 0d 0a                       -ascii"..
```

Compressed Secured Payload (variable): Specified in 2.2.1.2. This field MUST NOT contain the following sequence of bytes:

```
0000  0d 0a 2d 2d 3c 3c 5b 5b 26 26 26 5d 5d 3e 3e 2d  ..--<<[[&&&]]>>-
0010  2d 0d 0a                                        -..
```

Epilogue (19 bytes): The epilogue MUST comprise the following bytes:

```
0000  0d 0a 2d 2d 3c 3c 5b 5b 26 26 26 5d 5d 3e 3e 2d  ..--<<[[&&&]]>>-
0010  2d 0d 0a                                        -..
```

2.2.1.2 Compressed Secured Payload

This payload MUST be WBXML as specified in [WBXML1.2]. In addition, the following constraints MUST be met:

The WBXML version number, as specified in [WBXML1.2] section 5.4, MUST be 1.2.

The Document Public Identifier, as specified in [WBXML1.2] section 5.5, MUST be encoded either as the well-known document type public identifier "Unknown or missing public identifier" (value 1) as specified in [WBXML1.2] section 7.2, or as a string "(null),0" in the string table. Either value can be used.

The **Charset**, as specified in [WBXML1.2] section 5.6, MUST be 3 (representing US-ASCII).

The following tokens MUST NOT be used in the encoding of any attribute value (referred to as *attrValue* in [WBXML1.2] section 5.3):

- EXT_I_1 (0x41)
- EXT_I_2 (0x42)
- EXT_T_0 (0x80)
- EXT_T_1 (0x81)
- EXT_T_2 (0x82)
- EXT_0 (0xC0)
- EXT_1 (0xC1)
- EXT_2 (0xC2)

The following tokens MUST NOT be used in the encoding of any content (referred to as *content* in [WBXML1.2] section 5.3):

- EXT_I_1 (0x41)
- EXT_I_2 (0x42)
- EXT_T_0 (0x80)
- EXT_T_1 (0x81)
- EXT_T_2 (0x82)
- EXT_0 (0xC0)
- EXT_1 (0xC1)
- EXT_2 (0xC2)
- OPAQUE (0xC3)

This MUST encode the secured XML element specified in 2.2.1.3.

2.2.1.3 Secured XML

Section 3.1.5.1 specifies how to convert the secured XML to the delta XML specified in section 2.2.1.4.

2.2.1.3.1 Element Structure

The secured XML MUST consist of an element with tag "urn:groove.net:Del". This is the delta element.

The delta element MUST have a content element with the tag "urn:groove.net:SE". This is the secured element.

The secured element MUST have two content elements. The first content element MUST have the tag "urn:groove.net:EC". This is the encrypted element. The second content element MUST have the tag "urn:groove.net:Auth". This is the authenticator element.

2.2.1.3.2 Delta Element Attributes

These are specified in 2.2.1.4.2.

2.2.1.3.3 Secured Element Attributes

Version (String): This attribute MUST be present and MUST be "3,0,0,0".

2.2.1.3.4 Encrypted Element Attributes

EC (Binary): This attribute MUST be present and MUST be the encrypted payload.

IV (Binary): This attribute MUST be present and MUST be the initialization vector (IV).

KID (String): This attribute MUST be present and MUST be the key identifier.

KV (Int): This attribute MUST be present and MUST be the key version.

2.2.1.3.5 Authenticator Element Attributes

PTSig (Binary): This attribute MUST be present and MUST be the signature of the message.

2.2.1.4 Delta XML

2.2.1.4.1 Element Structure

A decrypted delta message is an XML message that MUST consist of an element with the tag "urn:groove.net:Del". This is the delta element.

The delta element MUST have a content element with the tag "urn:groove.net:Cmds". This is the commands element. The delta element MUST NOT have any other content.

The commands element MUST have one or more content elements with the tag "urn:groove.net:Cmd". These are the command elements. The commands element MUST NOT have any other content.

2.2.1.4.2 Delta Element Attributes

The protocol defines the following attributes for the **delta** element.

Gp (Int): This attribute MUST be present and MUST be the delta group number. This is used for delta ordering. If the sequence of the last delta in the creator's **delta log** is higher than the sequence on the new delta, then this MUST be one more than the highest group number in the delta log. Otherwise this SHOULD <1> be the highest group number but can be one more than the highest group number in the creator's delta log. This ensures that the newly created delta is ordered at the end of the creator's delta log.

Version (String): This attribute MUST be present and MUST be "1,0,0,0".

Seq (Hex String): This attribute MUST be present on normal deltas. It MUST NOT be present on asynchronous or identity-disseminated deltas. It MUST be the delta sequence, which uniquely identifies the delta and is used for delta ordering. The value MUST be 24 characters. The first 12 characters MUST be the Endpoint UID for the creating endpoint. The next eight characters MUST be the creator identifier. The creator identifier MUST either be the same as on the delta most recently created by this endpoint or be a newly generated string that has not previously been used as a creator identifier by this endpoint. In the event that the final four characters of the sequence for the most recently created delta were "FFFF" the creator identifier MUST be newly generated. The final four characters are the hexadecimal representation of the sequence number. If a new creator identifier is used, the sequence number MUST be 0001. Otherwise the sequence number MUST be 1 more than the sequence number on the delta most recently generated by this endpoint.

SubSeq (Hex String): This attribute MUST be present on an asynchronous or identity-disseminated delta. It MUST NOT be present on a normal delta. This is the sequence of an asynchronous or identity-disseminated delta, which uniquely identifies the delta and is used for delta ordering. The value MUST be 32 characters. The first 12 characters MUST be the Endpoint UID for the creating endpoint. The next 8 characters MUST be the creator identifier. The creator identifier MUST either be the same as on the delta most recently created by this endpoint or be a newly generated string that has not previously been used as a creator identifier by this endpoint.

If this endpoint had previously created a normal delta with the same creator identifier, then the next 4 characters MUST be the sequence number from the last normal delta created by this endpoint. Otherwise the next four characters MUST be "0000". In either case, the final 8 characters are the sub-sequence number. This MUST be the hexadecimal representation of a number. This MUST be "00000001" for the first asynchronous or identity-disseminated delta created and for the first

asynchronous or identity-disseminated delta which is created since a normal delta was created. It MUST be one more than the previous value sub-sequence value for all subsequent asynchronous and identity-disseminated deltas.

AssimilationPriority (Int): This attribute MUST be present if the delta has an explicit assimilation priority. It MUST NOT be present if the delta does not have an explicit assimilation priority. It MUST NOT be present on asynchronous or identity-disseminated deltas. The assimilation priority is used in delta ordering, as specified in section 3.1.5.2. The value of this attribute MAY be 0.

BlkNum (Int): This attribute MUST be present if the delta has an explicit assimilation priority. It MUST NOT be present if the delta does not have an explicit assimilation priority. This is the **block** number. This is used for delta ordering. It MUST be set to one more than the block number of the highest block in the delta log. See section 3.1.5.2.

DLS (String): This attribute MUST be present if the delta has an explicit assimilation priority. It MUST NOT be present if the delta does not have an explicit assimilation priority. This is the delta log state. The value MUST be a comma-delimited string. There MUST be one field in the string for each endpoint in the space. Each field in the string MUST be a 32 character hexadecimal string. The first 8 characters of the field MUST be a hexadecimal representation of the group number of the last normal delta in the delta log created by that endpoint. The next 24 characters MUST be the sequence number of the last normal delta in the delta log created by that endpoint.

Async (Null): This attribute MUST be present if the delta is an **async delta**. This attribute MUST NOT be present if the delta is not an asynchronous delta.

DepSeq (String): This attribute MUST be present if the delta has explicit dependencies. It MUST NOT be present if the delta does not have explicit dependencies. The explicit dependencies are computed as follows. Find the set of all sources in the **dependency graph** (see section 3.1.1). These are the immediate dependencies. If the last four characters of the sequence of this delta are not 0001 then remove from the set of immediate dependencies the most recently created delta from this endpoint. The resulting set contains the explicit dependencies. If this set is not empty, then this attribute MUST be a comma-delimited string. The fields in the string MUST be the sequences of the explicit dependencies.

IdDiss (Null): This attribute MUST be present if the delta is an identity-disseminated delta. It MUST NOT be present if this is not an identity-disseminated delta.

2.2.1.4.3 Commands Element Attributes

The protocol defines the following attributes for the commands element.

PurGrp (Int): This attribute MUST be present and MUST be the purge group. The value specifies that the delta creator is willing to purge all deltas with groups less than or equal to this value. This MUST NOT be set to a number higher than 0 unless it is guaranteed that all endpoints, including the local endpoint, in the shared space have all deltas up to that group number.

Rank (Int): This attribute MUST be present and MUST be one more than the highest rank of all received and previously created normal deltas.

SenderMinDep (Int): This attribute MUST be present and SHOULD <2> be equal to the smallest group number of all of the immediate dependencies of this delta. It MAY be any smaller number. This is used to calculate which deltas are available to be purged.

PurNot (Null): This attribute MUST be present if at least one of the command elements has the PurNot attribute. It MUST NOT be present if none of the command elements has the PurNot attribute.

SpStSet (String): This attribute MUST be present if there is new space state information that has not been sent on a previous delta. It MAY<3> be set if there is no new space state information. This is the space state set. The space state for an endpoint consists of the following information:

- **Rank:** The highest rank of the deltas executed on the endpoint.
- **Min Dependency Group:** The minimum dependency group declared by the endpoint.
- **Purge Group:** The group number that the endpoint is willing to purge.
- **Dependencies:** The sources of the dependency graph on the endpoint.

This attribute MUST contain updated space states for all endpoints, excluding the local endpoint and any states that have previously been sent on a delta. States for endpoints that have been previously sent on deltas MAY <4> be sent, but such information is redundant. The value MUST be a semi-colon delimited string. The string MUST end with a semi-colon. The fields are paired into SpaceStates and EndpointSets. For example:

```
SpaceState1;EndpointSet1;SpaceState2;EndpointSet2;…SpaceStateN;EndpointSetN
```

Within each pair, the endpoints in the endpoint set have the matching space state. In the example, the endpoints in EndpointSet2 have SpaceState2.

The **SpaceState** MUST be a semi-colon delimited string with four fields. The first MUST be the rank of the space state. The second MUST be the minimum dependency group of the space state. The third SHOULD <5> be the purge group of the space state but can be zero. The fourth MUST be the set of dependencies of the space state. The dependencies MUST be a comma-delimited string of sequence numbers.

The **EndpointSet** MUST be a comma-delimited string of endpoint identifiers. The endpoint identifiers are 16 character hexadecimal strings. The first 12 characters MUST be the Endpoint UID. The last 4 characters MUST be "0000".

TimeCreated (String): This attribute MAY <6> be present and MAY be any value.

2.2.1.4.4 Command Element Attributes

The protocol defines the following attributes for the command element.

EngineURL (String): This attribute MUST be present and MUST be the identifier of the engine that is used to execute the command.

Nested (Hex String): This attribute MUST be present if this command was created as part of a creation-nested delta. A creation-nested delta is a delta that is created after another (containing) delta has been created, but before any commands in the containing delta have been executed. The commands of the creation-nested delta are included in the containing delta. These commands are marked to indicate that they were part of the creation-nested delta and may be treated differently by the engine. This attribute MUST NOT be present if this command was not created as part of a creation-nested delta. It is the creation nested sequence of the delta. This MUST be 16 characters. All commands that were part of the same creation-nested delta MUST have the same value for this attribute. If there were multiple layers of nested deltas, this MUST be the same only for commands that were part of the same innermost nested delta.

NOrd (Int): This attribute MUST be present if this command was created as part of a creation-nested delta. This attribute MUST NOT be present if this command was not created as part of a creation-nested delta. This is the nested ordinal of the command. This MUST be set to the 0-based

ordinal of the command in the nested delta. If there were multiple layers of nested deltas, this MUST be the 0-based ordinal of the command in the outermost nested delta.

PurNot (Null): This attribute MUST be present if the engine that executed this command is notified when the command is purged.

urn:groove.net:CmdAsyncLocalOnly (Null): This attribute MUST be present if this command is only to be executed on the endpoint that created the delta.

urn:groove.net:CmdIdDiss (Null): This attribute MUST be present if this command is only to be executed on endpoints with the same Identity URL as the delta creator.

2.2.2 Delta Ack Message

2.2.2.1 MIME-like Wrapper

This wrapper MUST be as specified in section 2.2.1.1. The payload MUST be as specified in section 2.2.2.2.

2.2.2.2 Compressed Secured Payload

This payload MUST be WBXML, as specified in [WBXML1.2], which conforms to the constraints specified in 2.2.1.2. This MUST encode the secured XML specified in section 2.2.2.3.

2.2.2.3 Secured XML

Section 3.1.5.1 specifies how to process the secured XML. The result of this processing is the **Delta Ack** XML specified in section 2.2.2.4.

2.2.2.3.1 Element Structure

The secured XML MUST consist of an element with the tag **DelAck**. This is the delta acknowledgement element.

The **DelAck** element MUST have a content element with the tag **urn:groove.net:SE**. This is the secured element. It is specified in section 2.2.1.3.

2.2.2.3.2 DelAck Element Attributes

These attributes are as specified in section 2.2.2.4.2.

2.2.2.4 DelAck XML

2.2.2.4.1 Element Structure

This MUST consist of an element with the tag **DelAck**. This is the delta acknowledgement element.

The **DelAck** element MUST have a content element with the tag **DelAckBody**. This is the delta acknowledgement body element. The **DelAck** element MUST NOT have any other content.

The **DelAckBody** element MUST NOT have any content.

2.2.2.4.2 DelAck Element Attributes

The protocol defines the following attributes of the **DelAck** element.

ContactURL (String): This attribute MUST be present and MUST be the Identity URL of the endpoint that created the **DelAck**.

DepSeq (String): This attribute MUST be present and MUST have a value as defined in 2.2.1.4.2 except that the sequence of the previously generated delta from this endpoint is included.

DeviceURL (String): This attribute MUST be present and MUST be the device URL of the endpoint that created the **DelAck**.

Gp (Int): This attribute MUST be present and MUST be the highest group number of all the deltas executed by the endpoint that created the **DelAck**.

2.2.2.4.3 DelAckBody Element Attributes

The protocol defines the following attributes on the **DelAckBody** element.

PurGrp (Int): This attribute MUST be present and MUST have a value as specified in 2.2.1.4.3.

SenderMinDep (Int): This attribute MUST be present and MUST have a value as specified in 2.2.1.4.3.

SenderRank (Int): This attribute MUST be present and MUST be the highest rank of all deltas executed by the creator of this **DelAck**.

SpStSet (String): Specified in 2.2.1.4.3.

3 Protocol Details

3.1 Common Details

All endpoints in this protocol behave identically. There are no separate roles for clients and servers.

3.1.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

Delta Log: The delta log contains the history of all deltas that have been executed. The deltas should be organized sequentially by order of execution.

Dependency Graph: The set of dependencies on the deltas is representable as a directed acyclic graph. The deltas are vertices. The edges are the immediate dependencies of a delta. Edges are added so that there is at most one path between any two vertices. A delta A1 depends on a different delta B1 if and only if there is a path from A1 to B1 in the dependency graph. This is shown in the following diagram.

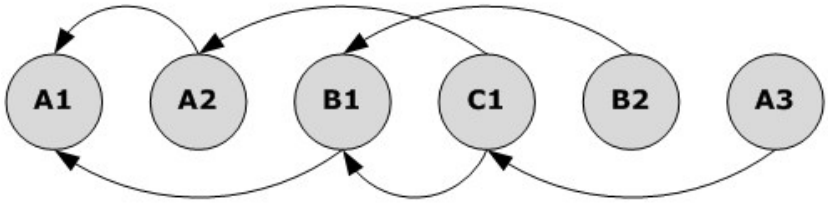

Figure 1: Sample dependency graph

The preceding graph results from the following steps:

1. A creates A1.
2. B, C receive A1.
3. A creates A2.
4. C receives A2.
5. B creates B1.
6. A,C receive B1.
7. C creates C1.
8. A receives C1.
9. B creates B2.
10. A creates A3.

Endpoint Space State: The last known space state for each endpoint. The following information is kept for each endpoint:

- **Rank:** The highest rank of the deltas executed on the endpoint.
- **Min Dependency Group:** The minimum dependency group declared by the endpoint.
- **Purge Group:** The group number that the endpoint is willing to purge.
- **Dependencies:** The sources of the dependency graph on the endpoint.

3.1.2 Timers

None.

3.1.3 Initialization

3.1.3.1 Per-Space Encryption Key

The per-space encryption key MUST be used to encrypt all messages. It is derived from the per-space master key described in section 1.5. The per-space encryption key MUST be the result of a Groove-specific key derivation function. The parameters MUST be:

i_Key: The **SHA-1** hash, as specified in [RFC3174], of the concatenation of the per-space master key and the Unicode string "MaskStringForTelespaceSecurityCipherKeys". This string MUST be hashed as a byte array. The zero terminator MUST NOT be included in the hash.

i_KeySizeInBytes: Size of **i_Key** in bytes.

i_DerivedKeySizeInBytes: MUST be the same as the size of the per-space master key.

3.1.3.1.1 Pseudo-code for Groove-specific Key Derivation Function

This function makes use of **HMAC-SHA1**, as specified in [RFC4634].

```
-- Data types:
--   Byte: 8-bit unsigned integer.
--   ByteArray: array of Bytes. Index is always zero based.
--   Int32: 32-bit signed integer.
--   UInt32: 32-bit unsigned integer.
--
-- Input:
--   i_Key as ByteArray: Data to derive the key from.
--   i_KeySizeInBytes as Int32: Number of bytes in i_Key.
--   i_DerivedKeySizeInBytes as Int32: Number of bytes for the derived key.
--
-- Output:
--   o_DerivedKey as ByteArray: Derived key.

DEFINE GrooveSpecificPBKDF2(i_Key, i_KeySizeInBytes, i_DerivedKeySizeInBytes, o_DerivedKey)
AS
     CONST Int32 hlen = 20
     CONST Int32 max_dkeylen = 0x7fffffff - 2 * hlen
     CONST Int32 maxkeysize = 64
     CONST Int32 maxpwdlen = maxkeysize
     CONST UInt32 maxblock = (i_DerivedKeySizeInBytes + (hlen - 1)) / hlen
```

```
VAR t as ByteArray[hlen]
VAR u as ByteArray[hlen]
VAR Index as Int32

FOR Index = 0 To hlen-1
     SET t[Index] = 0
     SET u[Index] = 0
ENDFOR

VAR hpwd as ByteArray[maxpwdlen]
VAR hpwdlen as Int32

IF i_KeySizeInBytes <= maxpwdlen THEN
     FOR Index = 0 To i_KeySizeInBytes-1
         SET hpwd[Index] = i_Key[Index]
     ENDFOR
     SET hpwdlen = i_KeySizeInBytes
ELSE
     VAR hash as SHA1
     CALL hash.Update(i_Key, i_KeySizeInBytes)
     SET hpwd = hash.Final()
     SET hpwdlen = 20
ENDIF

VAR hmac as HMAC_SHA1
SET hmac.Key = hpwd
SET hmac.KeySize = hpwdlen

VAR k_ipad as ByteArray[maxkeysize]
FOR Index = 0 To maxkeysize-1
     SET k_ipad[Index] = 0x36
ENDFOR

FOR Index = 0 To hpwdlen-1
     SET k_ipad[Index] = k_ipad[Index] XOR hpwd[Index]
ENDFOR

CALL hmac.Update(k_ipad, maxkeysize)

VAR accum as Int32
SET accum = 0

VAR block as UInt32
FOR block = 1 To maxblock
     VAR block_be As ByteArray[4]
     SET block_be[0] = RIGHT_SHIFT_BITS(block, 24)
     SET block_be[1] = RIGHT_SHIFT_BITS(block, 16)
     SET block_be[2] = RIGHT_SHIFT_BITS(block, 8)
     SET block_be[3] = RIGHT_SHIFT_BITS(block, 0)

     CALL hmac.Update(block_be, 4)
     SET u = hmac.Final()

     FOR Index = 0 To hlen-1
         SET t[Index] = u[Index]
     ENDFOR

     VAR want As Int32
     SET want = i_DerivedKeySizeInBytes - accum
```

```
            VAR got As Int32
            IF want > hlen THEN
                SET got = hlen
            ELSE
                SET got = want
            ENDIF

            FOR Index = 0 To got-1
                SET o_DerivedKey[accum+Index] = t[Index]
            ENDFOR

            SET accum = accum + got
        ENDFOR
    ENDDEFINE
```

3.1.3.2 Account Login

When the user logs into the **account** that contains the shared space the implementation MUST register as a resource handler for the asynchronous resource URL and the normal resource URL specified in section 2.1.

3.1.4 Higher-Layer Triggered Events

3.1.4.1 Normal Delta Created

The XML structure of a delta is specified in section 2.2.1.4. When a higher layer finishes creating a delta, the delta MUST be placed at the end of the delta log and executed. The delta MUST be secured and serialized as specified in section 3.1.4.3. The resulting message MUST be sent to all endpoints in the space.

3.1.4.2 Async or Identity-disseminated Delta Created

The XML structure of a delta is specified in section 2.2.1.4. When a higher layer finishes creating an async or identity-disseminated delta, the delta MUST be stored at the end of the delta log and executed. The delta MUST be secured and serialized as specified in section 3.1.4.3. An async delta MUST be sent to endpoints specified by the delta creator. An identity-disseminated delta MUST be sent to all endpoints in the space that have the same identity as the creating endpoint.

3.1.4.3 Securing and Serializing a Message

A delta message (section 2.2.1) and a **Delta Ack** message (section 2.2.2) are secured and serialized as follows.

3.1.4.3.1 Header and Payload

The content is removed from the root XML element. The root element with no content is the header element. In a delta message the header element MUST be the **Delta** element (section 2.2.1.4.2). In a **Delta Ack** message the header element MUST be the **DelAck** element (section 2.2.2.4.2). The header element MUST NOT have any content. The removed content is the payload element. In a delta message the payload element is the commands element (section 2.2.1.4.3) with its contents. In a **Delta Ack** message the payload element is the **DelAckBody** element (section 2.2.2.4.3).

3.1.4.3.2 Encrypted Payload

The payload element MUST be encoded in WBXML, as specified in [WBXML1.2], and in section 2.2.1.2. The order of the attributes MUST be sorted by Unicode code point. The resulting binary representation of the payload element MUST be encrypted using the AES algorithm, as specified in [FIPS197] in CTR mode, as specified in [BCMO800-38A]. The current per-space encryption key (section 3.1.3.1) MUST be used. An initialization vector MUST be used to encrypt each message.

3.1.4.3.3 Message Signature

The message digest MUST be computed using SHA-1, as specified in [RFC3174]. The input to the digest MUST consist of three values. First is the telespace canonical URL. Second is the header element encoded in WBXML, as specified in [WBXML1.2] and in section 2.2.1.2. The order of the attributes MUST be sorted by Unicode code point. Third is the encrypted payload specified in section 3.1.4.3.2.

The message digest MUST be signed using ESIGN, as specified in [IEEE1363a]. The per-space per-member signature private key (section 1.5) MUST be used. The result is the message signature.

3.1.4.3.4 Secured XML

The Secured XML (sections 2.2.1.3 and 2.2.2.3) MUST be created by adding the secured element as the only content of the header element. The encrypted element MUST be the first content element of the secured element. The authenticator element MUST be the second content element of the secured element.

The attributes on the encrypted element MUST be set as follows:

EC: The encrypted payload specified in 3.1.4.3.2.

IV: The initialization vector for the encrypted payload specified in 3.1.4.3.2.

KID: The key identifier of the per-space master key (section 1.5).

KV: The key version of the per-space master key (section 1.5).

The attributes on the authenticator element MUST be set as follows:

PTSig: The message signature specified in section 3.1.4.3.3.

3.1.4.3.5 Serialized Message

The secured XML MUST be compressed as specified in section 2.2.1.2 and included in a MIME-like wrapper as specified in section 2.2.1.1.

3.1.5 Message Processing Events and Sequencing Rules

Section 3.1.5.1 specifies message processing common to all messages. Sections 3.1.5.2, 3.1.5.3 and 3.1.5.4 specify the processing for specific message types.

3.1.5.1 Common Message Processing

The following steps MUST be used to process all messages.

3.1.5.1.1 Secure XML Deserialization

The compressed payload MUST be read from the MIME-like wrapper (section 2.2.1.1). The payload MUST be decompressed using WBXML, as specified in [WBXML1.2] and in section 2.2.1.2. The result is the secured XML (section 2.2.1.3). If the message does not match the message specification then the message MUST be ignored.

The secure element MUST be removed from the header element.

3.1.5.1.2 Signature Verification

The message digest MUST be computed using SHA-1, as specified in [RFC3174]. The input to the digest MUST consist of three values. First is the telespace canonical URL. Second is the header element encoded in WBXML, as specified in [WBXML1.2] and in section 2.2.1.2. The order of the attributes MUST be sorted by Unicode code point. Third is the encrypted payload which is the value of the EC attribute of the encrypted element.

The message signature, which is the value of the PTSig attribute of the authenticator element, MUST be verified using ESIGN, as specified in [IEEE1363a]. The per-space per-member signature public key (section 1.5) MUST be used. For delta messages the **Endpoint UID** in the first 12 bytes of the Seq or SubSeq attribute of the header element (section 2.2.1.4.2) MUST be used to determine which member's key to use. For **Delta Ack** messages the **ContactURL** attribute of the header element (section 2.2.2.4.2) MUST be used to determine which member's key to use. If the message signature is not the valid signature of the message digest the message MUST be ignored.

3.1.5.1.3 Payload Decryption

The encrypted payload, which is the value of the EC attribute of the encrypted element, MUST be decrypted using the AES algorithm, as specified in [FIPS197] in CTR mode, as specified in [BCMO800-38A]. The per-space encryption key (section 3.1.3.1) matching the key identifier and key version that are values of the KID and KV attributes of the encrypted element MUST be used. The value of the IV attribute on the encrypted element MUST be used as the initialization vector.

The decrypted payload MUST be decompressed using WBXML, as specified in [WBXML1.2], and in section 2.2.1.2. The resulting XML element MUST be set as the content of the header element. The message is now ready for processing specific to the message type. The message type is determined by the tag of the header element.

3.1.5.2 Normal Delta Received

The dependencies on the delta MUST be checked. If the last four characters of the delta sequence are not "0001", there is an implicit dependency on the previously created delta from the same endpoint. The previously created delta would have the same endpoint UID and creator identifier. The sequence number of the previously created delta would be one less than the sequence number of this delta. The explicit dependencies are the comma-separated fields in the **DepSeq** attribute. If any of the dependencies are not in the delta log, the new delta MUST NOT be ordered and executed. Instead it SHOULD be kept and reprocessed if the missing dependencies are added to the delta log.

When a delta is received it MUST be ordered in the delta log. The remainder of this section specifies how the new deltas and all deltas in the delta log MUST be ordered. Implementations are not required to adhere to all steps in this algorithm, as long as the final ordering is consistent with that described by this algorithm.

3.1.5.2.1 Computing Independent Deltas

To properly order deltas, it is necessary to compute whether two deltas, A1 and B1, are independent. This is done using the dependency graph described in section 3.1.1. If there is no path from A1 to B1 and no path from B1 to A1, the deltas are independent.

3.1.5.2.2 Ordering Into Blocks

The set of deltas are first ordered into blocks. Each block has one priority delta that defines the block and is the block delta. The **BlkNum** attributes on the block deltas are consecutive and define the ordering of the blocks. In the event of independent delta creation, there could be multiple priority deltas with the same block number, so it is necessary to determine which priority delta is the block delta.

Ordering deltas into blocks consists of the following steps, which are subsequently described in more detail:

1. Find all priority deltas for consideration.
2. Find the highest priority delta and make it a block delta.
3. Remove independent priority deltas from consideration.
4. Repeat steps 2 and 3 until there are no more priority deltas to consider.
5. Assign all remaining deltas to a block.

Here are the details for each step of the process:

1. Priority deltas are those that have the **AssimilationPriority** attribute set on the delta element. The algorithm starts by finding all of these deltas and putting them into consideration for being the block delta.
2. From the set of priority deltas in consideration, find the one that has the highest priority. In the event of a tie on priority, the winner is the one with the lower group number. If both priority and group number are tied, then the one with the lower sequence number is the winner. Sequence numbers are compared by treating the sequence string as a hexadecimal number with the first character being the most significant digit. The winning delta is considered a block delta and is removed from the set.
3. Remove from consideration all priority deltas that were created independently from the winning delta from step 2. The process described in section 3.1.5.2.1 finds these independent deltas.
4. Repeat steps 2 and 3 on the remaining deltas for consideration until there are no more deltas left for consideration.
5. Create a block for each block delta found in the preceding algorithm and order the blocks in increasing order by block number. All of the remaining deltas are assigned to blocks as follows. The normal deltas go into the highest block such that the block delta does not depend on the delta being ordered. The asynchronous and identity-disseminated deltas MUST be assigned to a block as described in 3.1.5.3.

3.1.5.2.3 Ordering Within Blocks

Within each block, the deltas in that block are divided into groups based on the **Gp** attribute on the delta element. The groups are ordered by number in increasing order.

Within each group the deltas are ordered in increasing order by sequence number specified in the Seq or **SubSeq** attribute on the delta element. Sequence and sub-sequences are compared by first treating all sequences as sub-sequences by appending "00000000". The resulting sub-sequences are compared by treating the sequence string as a hexadecimal number with the first character being the most significant digit.

3.1.5.2.4 Delta Execution

Once the new ordering of deltas has been determined it is necessary to undo and execute deltas to achieve a logically consistent ordering. The old ordering in the delta log and the new ordering MUST be compared to determine the point of divergence: the first position in the orderings that is not the same in both orderings. The deltas, starting with the last delta in the old ordering and proceeding in reverse order until the point of divergence, MUST be undone. Then the deltas in the new ordering, starting at the point of divergence and proceeding until the end, MUST be executed. The delta log MUST be replaced with the new ordering.

3.1.5.2.5 Space State Update

The space state of the endpoint that created the delta MUST be updated. The new space state has the rank from the **Rank** attribute, the purge group from the **PurGrp** attribute, and the dependencies of the delta sequence.

The space state for other endpoints MUST be updated if the delta contains a more recent space state for that endpoint. Space states are compared by first looking at the rank. The space state with the higher rank is more recent. If the ranks are the same, then the space state with the larger set of dependencies is more recent.

3.1.5.2.6 Delta Ack

After the new delta has been received a **Delta Ack** MUST be sent to the endpoint that created the delta. The **Delta Ack** message is specified in section 2.2.2. The message MUST be secured and serialized as specified in 3.1.4.3.

3.1.5.3 Async or Identity-disseminated Delta Received

The dependencies on the delta MUST be checked. If the sequence number of the delta sub-sequence is not "0000" then there is an implicit dependency on the previously created delta from the same endpoint. Its sequence would be the first 24 characters of the sub-sequence. The explicit dependencies are the comma-separated fields in the **DepSeq** attribute. If any of the dependencies are not in the delta log, the new delta MUST NOT be ordered and executed. Instead it SHOULD be kept and reprocessed if the missing dependencies are added to the delta log.

Ordering of asynchronous and identity-disseminated deltas is similar to ordering normal deltas. Asynchronous and identity-disseminated deltas MUST NOT have an assimilation priority, so these deltas do not play a role in the ordering of deltas into blocks. Async and identity-disseminated deltas MUST be assigned to the highest block in which they have a dependency. Within that block the ordering and execution proceeds as described in section 3.1.5.2.

3.1.5.4 Delta Ack Received

The dependencies on the **Delta Ack** MUST be checked. The dependencies are the comma-separated fields in the **DepSeq** attribute. If any of the dependencies are not in the delta log, the new **Delta Ack** MUST NOT update the space state. Instead the **Delta Ack** SHOULD be ignored.

If the dependencies do exist, the space state MUST be updated. This is done as specified in section 3.1.5.2.5 with the exception that the dependency sequence for the endpoint that created the **Delta Ack** is set to the dependencies of the **Delta Ack**.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

4 Protocol Examples

4.1 Processing an Incoming Delta Message

This example illustrates four stages in the decoding of an incoming Delta Message (see section 2.2.1). In this example, the incoming message represents an Add Record command invoked on another endpoint. The record has a Name field with the value "TestName" and an Age field with the value 123. The command element also contains the attributes "TableDefID" and "CMD", which are Groove RDB Commands Protocol attributes.

4.1.1 MIME-like Wrapper

```
0000  4d 49 4d 45 2d 56 65 72 73 69 6f 6e 3a 20 31 2e  MIME-Version: 1.
0010  30 20 28 47 72 6f 6f 76 65 20 32 29 0d 0a 43 6f  0 (Groove 2)..Co
0020  6e 74 65 6e 74 2d 54 79 70 65 3a 20 6d 75 6c 74  ntent-Type: mult
0030  69 70 61 72 74 2f 72 65 6c 61 74 65 64 3b 20 62  ipart/related; b
0040  6f 75 6e 64 61 72 79 3d 22 3c 3c 5b 5b 26 26 26  oundary="<<[[&&&
0050  5d 5d 3e 3e 22 0d 0a 3c 3c 5b 5b 26 26 26 5d 5d  ]]>>"..<<[[&&&]]
0060  3e 3e 0d 0a 43 6f 6e 74 65 6e 74 2d 54 79 70 65  >>..Content-Type
0070  3a 20 61 70 70 6c 69 63 61 74 69 6f 6e 2f 57 42  : application/WB
0080  58 4d 4c 3b 20 63 68 61 72 73 65 74 3d 22 75 73  XML; charset="us
0090  2d 61 73 63 69 69 22 0d 0a 02 00 00 03 8a 5d 28  -ascii".......](
00a0  6e 75 6c 6c 29 2c 30 00 75 72 6e 3a 67 72 6f 6f  null),0.urn:groo
00b0  76 65 2e 6e 65 74 3a 44 65 6c 00 56 65 72 73 69  ve.net:Del.Versi
00c0  6f 6e 00 31 2c 30 2c 30 2c 30 00 44 65 70 53 65  on.1,0,0,0.DepSe
00d0  71 00 36 42 31 36 43 34 34 45 39 37 45 37 33 46  q.6B16C44E97E73F
00e0  36 43 46 39 45 35 30 30 30 31 00 53 65 71 00 31  6CF9E50001.Seq.1
00f0  38 37 30 31 39 43 33 45 32 33 36 36 39 39 44 32  87019C3E236699D2
0100  33 31 31 30 30 30 32 00 47 70 00 32 31 00 75 72  3110002.Gp.21.ur
0110  6e 3a 67 72 6f 6f 76 65 2e 6e 65 74 3a 53 45 00  n:groove.net:SE.
0120  33 2c 30 2c 30 2c 30 00 75 72 6e 3a 67 72 6f 6f  3,0,0,0.urn:groo
0130  76 65 2e 6e 65 74 3a 45 43 00 4b 49 44 00 5f 54  ve.net:EC.KID._T
0140  4b 49 44 00 4b 56 00 31 00 49 56 00 42 46 72 48  KID.KV.1.IV.BFrH
0150  58 63 43 75 52 44 6c 76 37 30 51 72 36 31 79 68  XcCuRDlv70Qr61yh
0160  6b 51 3d 3d 00 45 43 00 79 71 73 73 41 55 33 2b  kQ==.EC.yqssAU3+
0170  6a 6a 56 61 43 43 47 52 39 4d 5a 68 59 79 64 41  jjVaCCGR9MZhYydA
0180  43 75 73 44 5a 68 6e 5a 32 57 4b 71 71 4d 4e 69  CusDZhnZ2WKqqMNi
0190  5a 38 6a 70 38 73 68 4d 74 6d 37 72 44 78 63 42  Z8jp8shMtm7rDxcB
01a0  51 68 47 73 2f 34 6f 58 53 64 39 54 57 63 4c 55  QhGs/4oXSd9TWcLU
01b0  5a 6b 4c 4a 6e 30 4f 38 6e 59 45 4a 53 45 31 7a  ZkLJn0O8nYEJSE1z
01c0  6d 63 75 6f 59 41 6d 64 6e 58 57 72 66 47 4e 58  mcuoYAmdnXWrfGNX
01d0  6e 36 78 39 43 78 4e 6c 4a 7a 6a 4b 77 61 47 61  n6x9CxNlJzjKwaGa
01e0  72 73 30 4d 48 48 5a 65 64 41 44 48 39 6b 33 45  rs0MHHZedADH9k3E
01f0  64 78 69 75 48 6f 4d 30 44 69 71 4a 57 79 7a 33  dxiuHoM0DiqJWyz3
0200  46 69 65 4a 6b 45 4a 7a 38 67 53 58 56 6c 41 6a  FieJkEJz8gSXVlAj
0210  4e 55 37 44 57 49 76 57 30 70 76 4d 62 37 67 6f  NU7DWIvW0pvMb7go
0220  4d 57 38 78 70 63 72 4a 6c 48 68 41 57 56 4d 72  MW8xpcrJlHhAWVMr
0230  77 39 30 58 30 44 38 35 75 52 66 32 62 51 34 42  w90X0D85uRf2bQ4B
0240  51 39 69 33 70 55 35 56 51 48 48 75 69 53 61 47  Q9i3pU5VQHHuiSaG
0250  6f 34 31 66 4d 74 31 49 72 52 38 44 67 52 72 74  o41fMt1IrR8DgRrt
0260  4c 6d 69 65 43 6e 4a 62 6e 6b 36 46 2b 31 31 54  LmieCnJbnk6F+11T
0270  75 50 63 78 46 46 7a 30 59 72 70 65 68 54 79 78  uPcxFFz0YrpehTyx
0280  37 2f 73 73 35 75 6d 67 4b 61 42 50 32 43 6e 58  7/ss5umgKaBP2CnX
0290  36 35 51 7a 45 30 64 45 39 34 2f 35 45 7a 38 32  65QzE0dE94/5Ez82
02a0  4d 4c 37 6d 30 41 78 36 48 6d 6b 78 45 72 59 78  ML7m0Ax6HmkxErYx
02b0  78 72 6e 52 7a 62 78 45 33 45 50 37 41 57 43 55  xrnRzbxE3EP7AWCU
02c0  7a 41 2b 54 65 6f 53 58 4c 46 43 77 36 75 72 5a  zA+TeoSXLFCw6urZ
```

```
02d0  67 56 6f 66 49 4d 79 4c 37 64 69 6d 42 5a 47 76  gVofIMyL7dimBZGv
02e0  43 47 2f 49 2b 67 62 62 32 64 36 63 53 67 38 4d  CG/I+gbb2d6cSg8M
02f0  62 6a 43 2f 64 70 66 78 38 75 58 61 4a 6d 6c 67  bjC/dpfx8uXaJmlg
0300  68 30 79 54 4d 47 73 4e 42 46 35 47 66 61 46 74  h0yTMGsNBF5GfaFt
0310  5a 56 6b 46 42 6c 64 75 37 32 6f 57 43 42 59 4e  ZVkFBldu72oWCBYN
0320  41 46 6f 52 55 77 37 48 53 45 6c 6f 68 5a 70 34  AFoRUw7HSElohZp4
0330  72 65 67 4b 79 38 35 36 72 70 53 71 47 49 62 4c  regKy856rpSqGIbL
0340  41 54 4a 46 69 70 54 4d 4e 4f 6b 68 6f 37 30 6c  ATJFipTMNOkho70l
0350  63 48 65 4c 6e 69 59 73 38 74 5a 67 41 68 2b 63  cHeLniYs8tZgAh+c
0360  74 50 2b 6b 58 48 4d 78 31 63 57 31 4c 31 6c 37  tP+kXHMx1cW1L1l7
0370  4b 70 79 71 77 64 79 42 44 6e 5a 4e 43 49 47 2f  KpyqwdyBDnZNCIG/
0380  6c 7a 79 4a 6e 6b 63 66 5a 32 56 4d 50 33 48 45  lzyJnkcfZ2VMP3HE
0390  52 36 30 62 48 38 56 47 79 2f 58 6e 4d 54 46 33  R60bH8VGy/XnMTF3
03a0  38 52 79 77 63 74 74 6a 50 52 35 67 50 33 33 33  8RywcttjPR5gP333
03b0  43 36 70 54 71 38 30 34 6d 75 7a 36 35 66 39 6b  C6pTq804muz65f9k
03c0  45 43 57 38 35 39 71 77 76 51 4f 53 74 42 4f 56  ECW859qwvQOStBOV
03d0  49 42 6f 7a 30 65 61 74 41 74 77 78 4b 49 32 76  IBoz0eatAtwxKI2v
03e0  45 31 50 65 79 56 52 38 49 5a 76 4c 33 51 39 6d  E1PeyVR8IZvL3Q9m
03f0  6a 33 71 65 42 4c 6f 6f 6f 4e 58 4e 75 2b 77 4a  j3qeBLoooNXNu+wJ
0400  35 55 6f 53 2b 4b 4d 34 58 70 6e 32 62 41 33 59  5UoS+KM4Xpn2bA3Y
0410  39 69 59 32 35 6a 52 72 31 4e 69 5a 7a 4f 6f 2f  9iY25jRr1NiZzOo/
0420  4e 4f 67 6e 70 4f 4f 59 73 49 52 46 74 42 51 33  NOgnpOOYsIRFtBQ3
0430  42 6f 43 32 56 42 37 42 76 6c 44 30 74 36 6b 4d  BoC2VB7BvlD0t6kM
0440  39 79 5a 45 43 4e 4d 4f 35 45 45 73 52 73 6a 41  9yZECNMO5EEsRsjA
0450  4e 31 33 61 4c 43 34 6e 6e 32 31 4b 30 39 48 6d  N13aLC4nn21K09Hm
0460  77 63 74 53 49 50 6e 45 42 59 6b 51 2f 69 39 67  wctSIPnEBYkQ/i9g
0470  65 55 2f 32 62 5a 6c 52 70 79 72 6f 4d 34 79 69  eU/2bZlRpyroM4yi
0480  78 38 74 36 4c 38 41 77 51 2b 4b 54 4b 57 67 77  x8t6L8AwQ+KTKWgw
0490  59 57 45 50 56 77 6a 36 6c 73 34 34 79 65 73 31  YWEPVwj6ls44yes1
04a0  59 4f 64 52 48 4e 35 6d 41 47 68 53 69 2b 4d 58  YOdRHN5mAGhSi+MX
04b0  6b 33 68 4e 47 79 78 42 69 6d 6e 4f 68 44 41 39  k3hNGyxBimnOhDA9
04c0  66 4e 36 53 78 4e 34 45 6c 49 47 6b 46 56 52 47  fN6SxN4ElIGkFVRG
04d0  48 45 6a 6b 2f 74 54 56 72 61 75 62 32 61 6b 3d  HEjk/tTVraub2ak=
04e0  00 75 72 6e 3a 67 72 6f 6f 76 65 2e 6e 65 74 3a  .urn:groove.net:
04f0  41 75 74 68 00 50 54 53 69 67 00 52 4f 53 5a 36  Auth.PTSig.ROSZ6
0500  63 42 68 54 6b 41 50 38 4f 78 5a 4e 62 64 31 53  cBhTkAP8OxZNbd1S
0510  76 73 6f 41 65 4e 73 54 39 37 57 74 76 56 67 6f  vsoAeNsT97WtvVgo
0520  39 44 42 2b 4b 72 6d 31 36 65 4c 52 48 7a 78 33  9DB+Krm16eLRHzx3
0530  37 41 38 35 48 43 64 70 38 38 7a 51 65 4b 74 5a  7A85HCdp88zQeKtZ
0540  45 6d 57 45 74 30 58 58 53 50 42 6c 41 73 31 44  EmWEt0XXSPBlAs1D
0550  73 36 37 4e 41 7a 77 48 39 55 4b 78 6d 70 41 42  s67NAzwH9UKxmpAB
0560  78 67 69 56 44 4b 59 45 51 52 6f 31 64 65 76 54  xgiVDKYEQRo1devT
0570  5a 6f 5a 50 6c 4c 51 4b 34 58 55 37 76 78 64 6c  ZoZPlLQK4XU7vxdl
0580  6d 69 6a 52 6d 59 33 42 51 31 77 30 47 48 5a 43  mijRmY3BQ1w0GHZC
0590  54 66 35 69 38 58 39 4a 37 59 54 56 76 52 38 34  Tf5i8X9J7YTVvR84
05a0  76 4b 44 38 33 6a 39 65 59 53 4c 5a 31 79 54 42  vKD83j9eYSLZ1yTB
05b0  65 37 51 4d 36 5a 70 36 68 61 6a 31 36 6d 4b 6c  e7QM6Zp6haj16mKl
05c0  6f 47 68 48 4f 6a 67 4a 78 47 2b 56 73 4c 77 68  oGhHOjgJxG+VsLwh
05d0  6c 5a 58 7a 38 68 6d 66 57 5a 61 36 53 4d 56 57  lZXz8hmfWZa6SMVW
05e0  57 30 7a 64 67 68 6b 48 72 6b 73 62 63 36 36 4e  W0zdghkHrksbc66N
05f0  2f 55 33 79 64 62 58 4e 70 79 77 00 c4 09 04 1c  /U3ydbXNpyw.....
0600  83 24 04 2c 83 33 04 4c 83 50 04 69 83 6c 01 c4  .$.,.3.L.P.i.l..
0610  6f 04 1c 83 81 01 01 84 81 09 04 81 1b 83 81 1f  o...............
0620  04 81 25 83 81 28 04 81 2a 83 81 2d 04 81 46 83  ..%..(..*..-..F.
0630  81 49 01 84 88 42 04 88 56 83 88 5c 01 01 01 0d  .I...B..V..\....
0640  0a 2d 2d 3c 3c 5b 5b 26 26 26 5d 5d 3e 3e 2d 2d  .--<<[[&&&]]>>--
0650  0d 0a                                            ..
```

4.1.2 Compressed Secured Payload

The MIME-like wrapper header and epilogue are stripped, leaving the following Compressed Secured Payload, which is a WBXML stream, as specified in [WBXML1.2]:

```
0000  02 00 00 03 8a 5d 28 6e 75 6c 6c 29 2c 30 00 75  .....](null),0.u
0010  72 6e 3a 67 72 6f 6f 76 65 2e 6e 65 74 3a 44 65  rn:groove.net:De
0020  6c 00 56 65 72 73 69 6f 6e 00 31 2c 30 2c 30 2c  l.Version.1,0,0,
0030  30 00 44 65 70 53 65 71 00 36 42 31 36 43 34 34  0.DepSeq.6B16C44
0040  45 39 37 45 37 33 46 36 43 46 39 45 35 30 30 30  E97E73F6CF9E5000
0050  31 00 53 65 71 00 31 38 37 30 31 39 43 33 45 32  1.Seq.187019C3E2
0060  33 36 36 39 39 44 32 33 31 31 30 30 30 32 00 47  36699D23110002.G
0070  70 00 32 31 00 75 72 6e 3a 67 72 6f 6f 76 65 2e  p.21.urn:groove.
0080  6e 65 74 3a 53 45 00 33 2c 30 2c 30 2c 30 00 75  net:SE.3,0,0,0.u
0090  72 6e 3a 67 72 6f 6f 76 65 2e 6e 65 74 3a 45 43  rn:groove.net:EC
00a0  00 4b 49 44 00 5f 54 4b 49 44 00 4b 56 00 31 00  .KID._TKID.KV.1.
00b0  49 56 00 42 46 72 48 58 63 43 75 52 44 6c 76 37  IV.BFrHXcCuRDlv7
00c0  30 51 72 36 31 79 68 6b 51 3d 3d 00 45 43 00 79  0Qr61yhkQ==.EC.y
00d0  71 73 73 41 55 33 2b 6a 6a 56 61 43 43 47 52 39  qssAU3+jjVaCCGR9
00e0  4d 5a 68 59 79 64 41 43 75 73 44 5a 68 6e 5a 32  MZhYydACusDZhnZ2
00f0  57 4b 71 71 4d 4e 69 5a 38 6a 70 38 73 68 4d 74  WKqqMNiZ8jp8shMt
0100  6d 37 72 44 78 63 42 51 68 47 73 2f 34 6f 58 53  m7rDxcBQhGs/4oXS
0110  64 39 54 57 63 4c 55 5a 6b 4c 4a 6e 30 4f 38 6e  d9TWcLUZkLJn0O8n
0120  59 45 4a 53 45 31 7a 6d 63 75 6f 59 41 6d 64 6e  YEJSE1zmcuoYAmdn
0130  58 57 72 66 47 4e 58 6e 36 78 39 43 78 4e 6c 4a  XWrfGNXn6x9CxNlJ
0140  7a 6a 4b 77 61 47 61 72 73 30 4d 48 48 5a 65 64  zjKwaGars0MHHZed
0150  41 44 48 39 6b 33 45 64 78 69 75 48 6f 4d 30 44  ADH9k3EdxiuHoM0D
0160  69 71 4a 57 79 7a 33 46 69 65 4a 6b 45 4a 7a 38  iqJWyz3FieJkEJz8
0170  67 53 58 56 6c 41 6a 4e 55 37 44 57 49 76 57 30  gSXVlAjNU7DWIvW0
0180  70 76 4d 62 37 67 6f 4d 57 38 78 70 63 72 4a 6c  pvMb7goMW8xpcrJl
0190  48 68 41 57 56 4d 72 77 39 30 58 30 44 38 35 75  HhAWVMrw90X0D85u
01a0  52 66 32 62 51 34 42 51 39 69 33 70 55 35 56 51  Rf2bQ4BQ9i3pU5VQ
01b0  48 48 75 69 53 61 47 6f 34 31 66 4d 74 31 49 72  HHuiSaGo41fMt1Ir
01c0  52 38 44 67 52 72 74 4c 6d 69 65 43 6e 4a 62 6e  R8DgRrtLmieCnJbn
01d0  6b 36 46 2b 31 31 54 75 50 63 78 46 46 7a 30 59  k6F+11TuPcxFFz0Y
01e0  72 70 65 68 54 79 78 37 2f 73 73 35 75 6d 67 4b  rpehTyx7/ss5umgK
01f0  61 42 50 32 43 6e 58 36 35 51 7a 45 30 64 45 39  aBP2CnX65QzE0dE9
0200  34 2f 35 45 7a 38 32 4d 4c 37 6d 30 41 78 36 48  4/5Ez82ML7m0Ax6H
0210  6d 6b 78 45 72 59 78 78 72 6e 52 7a 62 78 45 33  mkxErYxxrnRzbxE3
0220  45 50 37 41 57 43 55 7a 41 2b 54 65 6f 53 58 4c  EP7AWCUzA+TeoSXL
0230  46 43 77 36 75 72 5a 67 56 6f 66 49 4d 79 4c 37  FCw6urZgVofIMyL7
0240  64 69 6d 42 5a 47 76 43 47 2f 49 2b 67 62 62 32  dimBZGvCG/I+gbb2
0250  64 36 63 53 67 38 4d 62 6a 43 2f 64 70 66 78 38  d6cSg8MbjC/dpfx8
0260  75 58 61 4a 6d 6c 67 68 30 79 54 4d 47 73 4e 42  uXaJmlgh0yTMGsNB
0270  46 35 47 66 61 46 74 5a 56 6b 46 42 6c 64 75 37  F5GfaFtZVkFBldu7
0280  32 6f 57 43 42 59 4e 41 46 6f 52 55 77 37 48 53  2oWCBYNAFoRUw7HS
0290  45 6c 6f 68 5a 70 34 72 65 67 4b 79 38 35 36 72  ElohZp4regKy856r
02a0  70 53 71 47 49 62 4c 41 54 4a 46 69 70 54 4d 4e  pSqGIbLATJFipTMN
02b0  4f 6b 68 6f 37 30 6c 63 48 65 4c 6e 69 59 73 38  Okho70lcHeLniYs8
02c0  74 5a 67 41 68 2b 63 74 50 2b 6b 58 48 4d 78 31  tZgAh+ctP+kXHMx1
02d0  63 57 31 4c 31 6c 37 4b 70 79 71 77 64 79 42 44  cW1L1l7KpyqwdyBD
02e0  6e 5a 4e 43 49 47 2f 6c 7a 79 4a 6e 6b 63 66 5a  nZNCIG/lzyJnkcfZ
02f0  32 56 4d 50 33 48 45 52 36 30 62 48 38 56 47 79  2VMP3HER60bH8VGy
0300  2f 58 6e 4d 54 46 33 38 52 79 77 63 74 74 6a 50  /XnMTF38RywcttjP
0310  52 35 67 50 33 33 33 43 36 70 54 71 38 30 34 6d  R5gP333C6pTq804m
0320  75 7a 36 35 66 39 6b 45 43 57 38 35 39 71 77 76  uz65f9kECW859qwv
0330  51 4f 53 74 42 4f 56 49 42 6f 7a 30 65 61 74 41  QOStBOVIBoz0eatA
0340  74 77 78 4b 49 32 76 45 31 50 65 79 56 52 38 49  twxKI2vE1PeyVR8I
0350  5a 76 4c 33 51 39 6d 6a 33 71 65 42 4c 6f 6f 6f  ZvL3Q9mj3qeBLooo
```

```
0360  4e 58 4e 75 2b 77 4a 35 55 6f 53 2b 4b 4d 34 58  NXNu+wJ5UoS+KM4X
0370  70 6e 32 62 41 33 59 39 69 59 32 35 6a 52 72 31  pn2bA3Y9iY25jRr1
0380  4e 69 5a 7a 4f 6f 2f 4e 4f 67 6e 70 4f 4f 59 73  NiZzOo/NOgnpOOYs
0390  49 52 46 74 42 51 33 42 6f 43 32 56 42 37 42 76  IRFtBQ3BoC2VB7Bv
03a0  6c 44 30 74 36 6b 4d 39 79 5a 45 43 4e 4d 4f 35  lD0t6kM9yZECNMO5
03b0  45 45 73 52 73 6a 41 4e 31 33 61 4c 43 34 6e 6e  EEsRsjAN13aLC4nn
03c0  32 31 4b 30 39 48 6d 77 63 74 53 49 50 6e 45 42  21K09HmwctSIPnEB
03d0  59 6b 51 2f 69 39 67 65 55 2f 32 62 5a 6c 52 70  YkQ/i9geU/2bZlRp
03e0  79 72 6f 4d 34 79 69 78 38 74 36 4c 38 41 77 51  yroM4yix8t6L8AwQ
03f0  2b 4b 54 4b 57 67 77 59 57 45 50 56 77 6a 36 6c  +KTKWgwYWEPVwj6l
0400  73 34 34 79 65 73 31 59 4f 64 52 48 4e 35 6d 41  s44yes1YOdRHN5mA
0410  47 68 53 69 2b 4d 58 6b 33 68 4e 47 79 78 42 69  GhSi+MXk3hNGyxBi
0420  6d 6e 4f 68 44 41 39 66 4e 36 53 78 4e 34 45 6c  mnOhDA9fN6SxN4El
0430  49 47 6b 46 56 52 47 48 45 6a 6b 2f 74 54 56 72  IGkFVRGHEjk/tTVr
0440  61 75 62 32 61 6b 3d 00 75 72 6e 3a 67 72 6f 6f  aub2ak=.urn:groo
0450  76 65 2e 6e 65 74 3a 41 75 74 68 00 50 54 53 69  ve.net:Auth.PTSi
0460  67 00 52 4f 53 5a 36 63 42 68 54 6b 41 50 38 4f  g.ROSZ6cBhTkAP8O
0470  78 5a 4e 62 64 31 53 76 73 6f 41 65 4e 73 54 39  xZNbd1SvsoAeNsT9
0480  37 57 74 76 56 67 6f 39 44 42 2b 4b 72 6d 31 36  7WtvVgo9DB+Krm16
0490  65 4c 52 48 7a 78 33 37 41 38 35 48 43 64 70 38  eLRHzx37A85HCdp8
04a0  38 7a 51 65 4b 74 5a 45 6d 57 45 74 30 58 58 53  8zQeKtZEmWEt0XXS
04b0  50 42 6c 41 73 31 44 73 36 37 4e 41 7a 77 48 39  PBlAs1Ds67NAzwH9
04c0  55 4b 78 6d 70 41 42 78 67 69 56 44 4b 59 45 51  UKxmpABxgiVDKYEQ
04d0  52 6f 31 64 65 76 54 5a 6f 5a 50 6c 4c 51 4b 34  Ro1devTZoZPlLQK4
04e0  58 55 37 76 78 64 6c 6d 69 6a 52 6d 59 33 42 51  XU7vxdlmijRmY3BQ
04f0  31 77 30 47 48 5a 43 54 66 35 69 38 58 39 4a 37  1w0GHZCTf5i8X9J7
0500  59 54 56 76 52 38 34 76 4b 44 38 33 6a 39 65 59  YTVvR84vKD83j9eY
0510  53 4c 5a 31 79 54 42 65 37 51 4d 36 5a 70 36 68  SLZ1yTBe7QM6Zp6h
0520  61 6a 31 36 6d 4b 6c 6f 47 68 48 4f 6a 67 4a 78  aj16mKloGhHOjgJx
0530  47 2b 56 73 4c 77 68 6c 5a 58 7a 38 68 6d 66 57  G+VsLwhlZXz8hmfW
0540  5a 61 36 53 4d 56 57 57 30 7a 64 67 68 6b 48 72  Za6SMVWW0zdghkHr
0550  6b 73 62 63 36 36 4e 2f 55 33 79 64 62 58 4e 70  ksbc66N/U3ydbXNp
0560  79 77 00 c4 09 04 1c 83 24 04 2c 83 33 04 4c 83  yw......$.,.3.L.
0570  50 04 69 83 6c 01 c4 6f 04 1c 83 81 01 01 84 81  P.i.l..o........
0580  09 04 81 1b 83 81 1f 04 81 25 83 81 28 04 81 2a  .........%..(..*
0590  83 81 2d 04 81 46 83 81 49 01 84 88 42 04 88 56  ..-..F..I...B..V
05a0  83 88 5c 01 01 01                                ..\...
```

4.1.3 Secured XML

The WBXML stream, as specified in [WBXML1.2], comprising the Compressed Secured Payload is decoded into the following Secured XML:

```
<urn:groove.net:Del Gp="21" DepSeq="6B16C44E97E73F6CF9E50001" Version="1,0,0,0"
Seq="187019C3E236699D23110002">
    <urn:groove.net:SE Version="3,0,0,0">
        <urn:groove.net:EC
EC="yqssAU3+jjVaCCGR9MZhYydACusDZhnZ2WKqqMNiZ8jp8shMtm7rDxcBQhGs/4oXSd9TWcLUZkLJn0O8nYEJSE1zm
cuoYAmdnXWrfGNXn6x9CxNlJzjKwaGars0MHHZedADH9k3EdxiuHoM0DiqJWyz3FieJkEJz8gSXVlAjNU7DWIvW0pvMb7
goMW8xpcrJlHhAWVMrw90X0D85uRf2bQ4BQ9i3pU5VQHHuiSaGo41fMt1IrR8DgRrtLmieCnJbnk6F+11TuPcxFFz0Yrp
ehTyx7/ss5umgKaBP2CnX65QzE0dE94/5Ez82ML7m0Ax6HmkxErYxxrnRzbxE3EP7AWCUzA+TeoSXLFCw6urZgVofIMyL
7dimBZGvCG/I+gbb2d6cSg8MbjC/dpfx8uXaJmlgh0yTMGsNBF5GfaFtZVkFBldu72oWCBYNAFoRUw7HSElohZp4regKy
856rpSqGIbLATJFipTMNOkho70lcHeLniYs8tZgAh+ctP+kXHMx1cW1L1l7KpyqwdyBDnZNCIG/lzyJnkcfZ2VMP3HER6
0bH8VGy/XnMTF38Rywcttj PR5gP333C6pTq804muz65f9kECW859qwvQOStBOVIBoz0eatAtwxKI2vE1PeyVR8IZvL3Q9
mj3qeBLoooNXNu+wJ5UoS+KM4Xpn2bA3Y9iY25jRr1NiZzOo/NOgnpOOYsIRFtBQ3BoC2VB7BvlD0t6kM9yZECNMO5EEs
RsjAN13aLC4nn21K09HmwctSIPnEBYkQ/i9geU/2bZlRpyroM4yix8t6L8AwQ+KTKWgwYWEPVwj6ls44yes1YOdRHN5mA
GhSi+MXk3hNGyxBimnOhDA9fN6SxN4ElIGkFVRGHEjk/tTVraub2ak=" IV="BFrHXcCuRDlv70Qr61yhkQ==" KV="1"
KID="_TKID"/>
```

```
        <urn:groove.net:Auth
PTSig="ROSZ6cBhTkAP8OxZNbd1SvsoAeNsT97WtvVgo9DB+Krm16eLRHzx37A85HCdp88zQeKtZEmWEt0XXSPBlAs1Ds
67NAzwH9UKxmpABxgiVDKYEQRo1devTZoZPlLQK4XU7vxdlmijRmY3BQ1w0GHZCTf5i8X9J7YTVvR84vKD83j9eYSLZ1y
TBe7QM6Zp6haj16mKloGhHOjgJxG+VsLwhlZXz8hmfWZa6SMVWW0zdghkHrksbc66N/U3ydbXNpyw"/>
    </urn:groove.net:SE>
</urn:groove.net:Del>
```

4.1.4 Delta XML

The binary content of the Secured XML, embedded in the EC attribute of the element with the name urn:groove.net:EC, is Base64-decoded and decrypted, producing the following Delta XML:

```
<urn:groove.net:Del Gp="21" DepSeq="6B16C44E97E73F6CF9E50001" Version="1,0,0,0"
Seq="187019C3E236699D23110002">
    <urn:groove.net:Cmds PurGrp="18" TimeCreated="1203007644862" SenderMinDep="19" Rank="95"
PurNot="" SpStSet="94;20;0;6B16C44E97E73F6CF9E50001;6B16C44E97E70000;">
        <urn:groove.net:Cmd TableDefID="-1" CMD="0"
EngineURL="ToolContainer/e5wk3rqetj6vg/RDBManager" PurNot="" NOrd="0" DBName="RDB"
Nested="700AAC32E2C61404">
            <urn:groove.net:Record3 Forms_Tool_grooveFormID="1.063395895012627E-043"
_Created="1203007639620" _RecordID="-7.5191736700565293E-050" Age="123"
_CreatedByURL="grooveIdentity://ke8xy5aqrzcwf5kief35drj82e5xmvt8@" RecDefID="12684112"
Name="TestName" _Modified="1203007639620"
_ModifiedByURL="grooveIdentity://ke8xy5aqrzcwf5kief35drj82e5xmvt8@"/>
        </urn:groove.net:Cmd>
    </urn:groove.net:Cmds>
</urn:groove.net:Del>
```

The command element also contains the attributes **TableDefID** and **CMD**, which are Groove RDB Commands Protocol attributes.

4.2 Producing an Outgoing Delta Ack Message

This example follows from Section 4.1, where the incoming **Delta** message was decoded. After receiving this message, an outgoing **Delta Ack** message (section 2.2.2) is generated. This example illustrates four stages in the encoding of this message.

4.2.1 DelAck XML

```
<DelAck DepSeq="187019C3E236699D23110002"
DeviceURL="dpp:///ta9mpajfckvk8yhmi39m3zq76v5wrypn5xhbtza"
ContactURL="grooveIdentity://ytbefxy2gge7svbtqi473ctfgg6dz895@" Gp="21">
    <DelAckBody SpStSet="95;19;18;187019C3E236699D23110002;187019C3E2360000;" PurGrp="18"
SenderMinDep="20" SenderRank="95"/>
</DelAck>
```

4.2.2 Secured XML

The DelAck XML is encrypted, Base64-encoded, and embedded in a Secure XML element:

```
<DelAck DepSeq="187019C3E236699D23110002"
DeviceURL="dpp:///ta9mpajfckvk8yhmi39m3zq76v5wrypn5xhbtza"
ContactURL="grooveIdentity://ytbefxy2gge7svbtqi473ctfgg6dz895@" Gp="21">
    <urn:groove.net:SE Version="3,0,0,0">
```

```
        <urn:groove.net:EC KID="_TKID" KV="1" IV="DCpmzlzs/hiz6+Alr5DvXQ=="
EC="N3UNlLWzZzEF+0mC6cink9gRzoNYw/PEmXplL+zlWLr1niuRjeIq9uyflxzu3kE+MgC7TOulbeg+M2Bc3+vixXZ7Y
H+2OmnepnatgimnGDk+9OA0jvHd8M9JDzf2TcP1jZ3tHkceF50q2gda5n9Qnu9U/tRmInn2PxZU91zsawR7FdejsWk="/
>
        <urn:groove.net:Auth
PTSig="kRCehuwNOqgUbrEsn4tKyQCurWYh7XqqzWsC4euk5gCAHUoMG+mNcea0XC7WtWEDDZgIOrqz6dwNzEul+RpfE/
uUugbIqjFRaevzlPYIqetCd62XDc/W+PiI+tQ9xsPm5r3wyq/+YdF+IUmumKa7722LWLQEifK8sKSqbtvOkjqnv4W+U7p
BHukKxOAIc32SqnWjKqr54xZzqfK2WwWkKgeKccPGYHMFHYHWqk3HV8ZvjkYJ6l9vA4yT/LbaRfdb"/>
    </urn:groove.net:SE>
</DelAck>
```

4.2.3 Compressed Secured Payload

The Secured XML is compressed using WBXML, as specified in [WBXML1.2], to form the Compressed Secured Payload:

```
0000  02 00 00 03 85 5f 28 6e 75 6c 6c 29 2c 30 00 44  ....._(null),0.D
0010  65 6c 41 63 6b 00 44 65 70 53 65 71 00 31 38 37  elAck.DepSeq.187
0020  30 31 39 43 33 45 32 33 36 36 39 39 44 32 33 31  019C3E236699D231
0030  31 30 30 30 32 00 44 65 76 69 63 65 55 52 4c 00  10002.DeviceURL.
0040  64 70 70 3a 2f 2f 2f 74 61 39 6d 70 61 6a 66 63  dpp:///ta9mpajfc
0050  6b 76 6b 38 79 68 6d 69 33 39 6d 33 7a 71 37 36  kvk8yhmi39m3zq76
0060  76 35 77 72 79 70 6e 35 78 68 62 74 7a 61 00 43  v5wrypn5xhbtza.C
0070  6f 6e 74 61 63 74 55 52 4c 00 67 72 6f 6f 76 65  ontactURL.groove
0080  49 64 65 6e 74 69 74 79 3a 2f 2f 79 74 62 65 66  Identity://ytbef
0090  78 79 32 67 67 65 37 73 76 62 74 71 69 34 37 33  xy2gge7svbtqi473
00a0  63 74 66 67 67 36 64 7a 38 39 35 40 00 47 70 00  ctfgg6dz895@.Gp.
00b0  32 31 00 75 72 6e 3a 67 72 6f 6f 76 65 2e 6e 65  21.urn:groove.ne
00c0  74 3a 53 45 00 56 65 72 73 69 6f 6e 00 33 2c 30  t:SE.Version.3,0
00d0  2c 30 2c 30 00 75 72 6e 3a 67 72 6f 6f 76 65 2e  ,0,0.urn:groove.
00e0  6e 65 74 3a 45 43 00 4b 49 44 00 5f 54 4b 49 44  net:EC.KID._TKID
00f0  00 4b 56 00 31 00 49 56 00 44 43 70 6d 7a 6c 7a  .KV.1.IV.DCpmzlz
0100  73 2f 68 69 7a 36 2b 41 6c 72 35 44 76 58 51 3d  s/hiz6+Alr5DvXQ=
0110  3d 00 45 43 00 4e 33 55 4e 6c 4c 57 7a 5a 7a 45  =.EC.N3UNlLWzZzE
0120  46 2b 30 6d 43 36 63 69 6e 6b 39 67 52 7a 6f 4e  F+0mC6cink9gRzoN
0130  59 77 2f 50 45 6d 58 70 6c 4c 2b 7a 6c 57 4c 72  Yw/PEmXplL+zlWLr
0140  31 6e 69 75 52 6a 65 49 71 39 75 79 66 6c 78 7a  1niuRjeIq9uyflxz
0150  75 33 6b 45 2b 4d 67 43 37 54 4f 75 6c 62 65 67  u3kE+MgC7TOulbeg
0160  2b 4d 32 42 63 33 2b 76 69 78 58 5a 37 59 48 2b  +M2Bc3+vixXZ7YH+
0170  32 4f 6d 6e 65 70 6e 61 74 67 69 6d 6e 47 44 6b  2OmnepnatgimnGDk
0180  2b 39 4f 41 30 6a 76 48 64 38 4d 39 4a 44 7a 66  +9OA0jvHd8M9JDzf
0190  32 54 63 50 31 6a 5a 33 74 48 6b 63 65 46 35 30  2TcP1jZ3tHkceF50
01a0  71 32 67 64 61 35 6e 39 51 6e 75 39 55 2f 74 52  q2gda5n9Qnu9U/tR
01b0  6d 49 6e 6e 32 50 78 5a 55 39 31 7a 73 61 77 52  mInn2PxZU91zsawR
01c0  37 46 64 65 6a 73 57 6b 3d 00 75 72 6e 3a 67 72  7FdejsWk=.urn:gr
01d0  6f 6f 76 65 2e 6e 65 74 3a 41 75 74 68 00 50 54  oove.net:Auth.PT
01e0  53 69 67 00 6b 52 43 65 68 75 77 4e 4f 71 67 55  Sig.kRCehuwNOqgU
01f0  62 72 45 73 6e 34 74 4b 79 51 43 75 72 57 59 68  brEsn4tKyQCurWYh
0200  37 58 71 71 7a 57 73 43 34 65 75 6b 35 67 43 41  7XqqzWsC4euk5gCA
0210  48 55 6f 4d 47 2b 6d 4e 63 65 61 30 58 43 37 57  HUoMG+mNcea0XC7W
0220  74 57 45 44 44 5a 67 49 4f 72 71 7a 36 64 77 4e  tWEDDZgIOrqz6dwN
0230  7a 45 75 6c 2b 52 70 66 45 2f 75 55 75 67 62 49  zEul+RpfE/uUugbI
0240  71 6a 46 52 61 65 76 7a 6c 50 59 49 71 65 74 43  qjFRaevzlPYIqetC
0250  64 36 32 58 44 63 2f 57 2b 50 69 49 2b 74 51 39  d62XDc/W+PiI+tQ9
0260  78 73 50 6d 35 72 33 77 79 71 2f 2b 59 64 46 2b  xsPm5r3wyq/+YdF+
0270  49 55 6d 75 6d 4b 61 37 37 32 32 4c 57 4c 51 45  IUumumKa7722LWLQE
0280  69 66 4b 38 73 4b 53 71 62 74 76 4f 6b 6a 71 6e  ifK8sKSqbtvOkjqn
0290  76 34 57 2b 55 37 70 42 48 75 6b 4b 78 4f 41 49  v4W+U7pBHukKxOAI
```

```
02a0  63 33 32 53 71 6e 57 6a 4b 71 72 35 34 78 5a 7a  c32SqnWjKqr54xZz
02b0  71 66 4b 32 57 77 57 6b 4b 67 65 4b 63 63 50 47  qfK2WwWkKgeKccPG
02c0  59 48 4d 46 48 59 48 57 71 6b 33 48 56 38 5a 76  YHMFHYHWqk3HV8Zv
02d0  6a 6b 59 4a 36 6c 39 76 41 34 79 54 2f 4c 62 61  jkYJ6l9vA4yT/Lba
02e0  52 66 64 62 00 c4 09 04 10 83 17 04 30 83 3a 04  Rfdb........0.:.
02f0  69 83 74 04 81 27 83 81 2a 01 c4 81 2d 04 81 3f  i.t..'..*...-..?
0300  83 81 47 01 84 81 4f 04 81 61 83 81 65 04 81 6b  ..G...O..a..e..k
0310  83 81 6e 04 81 70 83 81 73 04 82 0c 83 82 0f 01  ..n..p..s.......
0320  84 83 44 04 83 58 83 83 5e 01 01 01              ..D..X..^...
```

4.2.4 MIME-like Wrapper

The MIME-like wrapper header and epilogue are added to the Compressed Secured Payload to produce the final **Delta Ack** message:

```
0000  4d 49 4d 45 2d 56 65 72 73 69 6f 6e 3a 20 31 2e  MIME-Version: 1.
0010  30 20 28 47 72 6f 6f 76 65 20 32 29 0d 0a 43 6f  0 (Groove 2)..Co
0020  6e 74 65 6e 74 2d 54 79 70 65 3a 20 6d 75 6c 74  ntent-Type: mult
0030  69 70 61 72 74 2f 72 65 6c 61 74 65 64 3b 20 62  ipart/related; b
0040  6f 75 6e 64 61 72 79 3d 22 3c 3c 5b 5b 26 26 26  oundary="<<[[&&&
0050  5d 5d 3e 3e 22 0d 0a 3c 3c 5b 5b 26 26 26 5d 5d  ]]>>"..<<[[&&&]]
0060  3e 3e 0d 0a 43 6f 6e 74 65 6e 74 2d 54 79 70 65  >>..Content-Type
0070  3a 20 61 70 70 6c 69 63 61 74 69 6f 6e 2f 57 42  : application/WB
0080  58 4d 4c 3b 20 63 68 61 72 73 65 74 3d 22 75 73  XML; charset="us
0090  2d 61 73 63 69 69 22 0d 0a 02 00 00 03 85 5f 28  -ascii"........_(
00a0  6e 75 6c 6c 29 2c 30 00 44 65 6c 41 63 6b 00 44  null),0.DelAck.D
00b0  65 70 53 65 71 00 31 38 37 30 31 39 43 33 45 32  epSeq.187019C3E2
00c0  33 36 36 39 39 44 32 33 31 31 30 30 30 32 00 44  36699D23110002.D
00d0  65 76 69 63 65 55 52 4c 00 64 70 70 3a 2f 2f 2f  eviceURL.dpp:///
00e0  74 61 39 6d 70 61 6a 66 63 6b 76 6b 38 79 68 6d  ta9mpajfckvk8yhm
00f0  69 33 39 6d 33 7a 71 37 36 76 35 77 72 79 70 6e  i39m3zq76v5wrypn
0100  35 78 68 62 74 7a 61 00 43 6f 6e 74 61 63 74 55  5xhbtza.ContactU
0110  52 4c 00 67 72 6f 6f 76 65 49 64 65 6e 74 69 74  RL.grooveIdentit
0120  79 3a 2f 2f 79 74 62 65 66 78 79 32 67 67 65 37  y://ytbefxy2gge7
0130  73 76 62 74 71 69 34 37 33 63 74 66 67 67 36 64  svbtqi473ctfgg6d
0140  7a 38 39 35 40 00 47 70 00 32 31 00 75 72 6e 3a  z895@.Gp.21.urn:
0150  67 72 6f 6f 76 65 2e 6e 65 74 3a 53 45 00 56 65  groove.net:SE.Ve
0160  72 73 69 6f 6e 00 33 2c 30 2c 30 2c 30 00 75 72  rsion.3,0,0,0.ur
0170  6e 3a 67 72 6f 6f 76 65 2e 6e 65 74 3a 45 43 00  n:groove.net:EC.
0180  4b 49 44 00 5f 54 4b 49 44 00 4b 56 00 31 00 49  KID._TKID.KV.1.I
0190  56 00 44 43 70 6d 7a 6c 7a 73 2f 68 69 7a 36 2b  V.DCpmzlzs/hiz6+
01a0  41 6c 72 35 44 76 58 51 3d 3d 00 45 43 00 4e 33  Alr5DvXQ==.EC.N3
01b0  55 4e 6c 4c 57 7a 5a 7a 45 46 2b 30 6d 43 36 63  UNlLWzZzEF+0mC6c
01c0  69 6e 6b 39 67 52 7a 6f 4e 59 77 2f 50 45 6d 58  ink9gRzoNYw/PEmX
01d0  70 6c 4c 2b 7a 6c 57 4c 72 31 6e 69 75 52 6a 65  plL+zlWLr1niuRje
01e0  49 71 39 75 79 66 6c 78 7a 75 33 6b 45 2b 4d 67  Iq9uyflxzu3kE+Mg
01f0  43 37 54 4f 75 6c 62 65 67 2b 4d 32 42 63 33 2b  C7TOulbeg+M2Bc3+
0200  76 69 78 58 5a 37 59 48 2b 32 4f 6d 6e 65 70 6e  vixXZ7YH+2Omnepn
0210  61 74 67 69 6d 6e 47 44 6b 2b 39 4f 41 30 6a 76  atgimnGDk+9OA0jv
0220  48 64 38 4d 39 4a 44 7a 66 32 54 63 50 31 6a 5a  Hd8M9JDzf2TcP1jZ
0230  33 74 48 6b 63 65 46 35 30 71 32 67 64 61 35 6e  3tHkceF50q2gda5n
0240  39 51 6e 75 39 55 2f 74 52 6d 49 6e 6e 32 50 78  9Qnu9U/tRmInn2Px
0250  5a 55 39 31 7a 73 61 77 52 37 46 64 65 6a 73 57  ZU91zsawR7FdejsW
0260  6b 3d 00 75 72 6e 3a 67 72 6f 6f 76 65 2e 6e 65  k=.urn:groove.ne
0270  74 3a 41 75 74 68 00 50 54 53 69 67 00 6b 52 43  t:Auth.PTSig.kRC
0280  65 68 75 77 4e 4f 71 67 55 62 72 45 73 6e 34 74  ehuwNOqgUbrEsn4t
0290  4b 79 51 43 75 72 57 59 68 37 58 71 71 7a 57 73  KyQCurWYh7XqqzWs
```

```
02a0  43 34 65 75 6b 35 67 43 41 48 55 6f 4d 47 2b 6d  C4euk5gCAHUoMG+m
02b0  4e 63 65 61 30 58 43 37 57 74 57 45 44 44 5a 67  Ncea0XC7WtWEDDZg
02c0  49 4f 72 71 7a 36 64 77 4e 7a 45 75 6c 2b 52 70  IOrqz6dwNzEul+Rp
02d0  66 45 2f 75 55 75 67 62 49 71 6a 46 52 61 65 76  fE/uUugbIqjFRaev
02e0  7a 6c 50 59 49 71 65 74 43 64 36 32 58 44 63 2f  zlPYIqetCd62XDc/
02f0  57 2b 50 69 49 2b 74 51 39 78 73 50 6d 35 72 33  W+PiI+tQ9xsPm5r3
0300  77 79 71 2f 2b 59 64 46 2b 49 55 6d 75 6d 4b 61  wyq/+YdF+IUmumKa
0310  37 37 32 32 4c 57 4c 51 45 69 66 4b 38 73 4b 53  7722LWLQEifK8sKS
0320  71 62 74 76 4f 6b 6a 71 6e 76 34 57 2b 55 37 70  qbtvOkjqnv4W+U7p
0330  42 48 75 6b 4b 78 4f 41 49 63 33 32 53 71 6e 57  BHukKxOAIc32SqnW
0340  6a 4b 71 72 35 34 78 5a 7a 71 66 4b 32 57 77 57  jKqr54xZzqfK2WwW
0350  6b 4b 67 65 4b 63 63 50 47 59 48 4d 46 48 59 48  kKgeKccPGYHMFHYH
0360  57 71 6b 33 48 56 38 5a 76 6a 6b 59 4a 36 6c 39  Wqk3HV8ZvjkYJ619
0370  76 41 34 79 54 2f 4c 62 61 52 66 64 62 00 c4 09  vA4yT/LbaRfdb...
0380  04 10 83 17 04 30 83 3a 04 69 83 74 04 81 27 83  .....0.:.i.t..'.
0390  81 2a 01 c4 81 2d 04 81 3f 83 81 47 01 84 81 4f  .*...-..?..G...O
03a0  04 81 61 83 81 65 04 81 6b 83 81 6e 04 81 70 83  ..a..e..k..n..p.
03b0  81 73 04 82 0c 83 82 0f 01 84 83 44 04 83 58 83  .s.........D..X.
03c0  83 5e 01 01 01 0d 0a 2d 2d 3c 3c 5b 5b 26 26 26  .^.....--<<[[&&&
03d0  5d 5d 3e 3e 2d 2d 0d 0a                          ]]>>--..
```

4.3 Producing an Outgoing Delta Message

This example illustrates four stages in the encoding of an outgoing **Delta** message (see section 2.2.1). In this example, the outgoing message represents an Add Record command invoked on the local endpoint. The record has a **Name** field with the value "TestName", and an **Age** field with the value 123.

4.3.1 Delta XML

```
<urn:groove.net:Del DepSeq="6B16C44E97E73F6CF9E50002" Version="1,0,0,0" Gp="23"
Seq="6B16C44E97E7011B33C40001">
    <urn:groove.net:Cmds SpStSet="98;22;0;6B16C44E97E73F6CF9E50002;187019C3E2360000;"
PurNot="" PurGrp="21" SenderMinDep="23" Rank="99" TimeCreated="1203089515333">
        <urn:groove.net:Cmd DBName="RDB" PurNot=""
EngineURL="ToolContainer/e5wk3rqetj6vg/RDBManager" NOrd="0" Nested="8AE984B938ED8DDB"
TableDefID="-1" CMD="0">
            <urn:groove.net:Record3
_CreatedByURL="grooveIdentity://ytbefxy2gge7svbtqi473ctfgg6dz895@"
_ModifiedByURL="grooveIdentity://ytbefxy2gge7svbtqi473ctfgg6dz895@" Name="TestName"
Forms_Tool_grooveFormID="1.063395895012627E-043" RecDefID="12684112" _Created="1203089510896"
_Modified="1203089510896" Age="123" _RecordID="-2.4200873997467265E+048"/>
        </urn:groove.net:Cmd>
    </urn:groove.net:Cmds>
</urn:groove.net:Del>
```

4.3.2 Secured XML

The Delta XML is encrypted, Base64-encoded, and embedded in a Secure XML element:

```
<urn:groove.net:Del Version="1,0,0,0" DepSeq="6B16C44E97E73F6CF9E50002"
Seq="6B16C44E97E7011B33C40001" Gp="23">
    <urn:groove.net:SE Version="3,0,0,0">
        <urn:groove.net:EC KID="_TKID" KV="1" IV="7zAst91gDirqtuliaKdcvg=="
EC="Zg3jhWshQEryhS4bEEiPa4KGUxtNl1L+J2JyrLghGJ2NR6ORY05Szypknq5YhiC9AXxzFFOhR/IjfJbqPIH57prjt
qhAw8h3Ow4nrVe82lAOAX02S4mRFKiGE2v+k24j8wPaRtXLZRQPBrWOFLLpVSe5XSqKncJB2B/hwNXMEfIYIKeyIpnnKh
```

```
H/f6wEwKzScOQ46+2lBfolvfmnwP3hOT64RMGMGYyrdELGKazQ4Zatufaku6y5k9KdwTgfKN4iZajy7N3Dpw2OrjPWPnN
iRRu8Cv0OQjJZW7ecs/tde/K1N0dkAbPJYwqYCKXJcrSxwR1nT5tbNKq7NJdAS3ULe/Wayud/xy6fTuzhNc63i0OvQoLt
OeFyzr7tnWMUKRrOOUPqhc+dUFODiF5TKCb/bHobCnPUEcvw8X78Zf5lmoEyLYSkh4fiz35r6/I0rayLFJW1gHg+ObGgt
gJpGC3CFc7toEwpvNdw5tMA4BRB1B7OwLzC4IFGK/z+s7moGDqB+DrLt/muIuv6HRzAnHKGzNcLfDdazbIP7PAXFvFCVK
QPkZ6E1OEailDK6DgON6pN97OJVCD5h2171yJuzlV7arrRqnjRvY0hU6BM+wjzayfjREnGOegqRmzGpOPZoYeNlIdRAIF
8i2fq3vnp2zekoLXTukBFULWbTNaHm2DXAWR68C488tbfQRck0eXAvasbg+kGSm6rashZZDHa50YSbT65uRU4MBjJWlTW
NGohCwJvPAl5NM7SadEqeck4RF6viv7CMzEq2R72sBYWRzoowlJLoUOMqZwOQWNvizL0/0H22/28EENYYp6lQKVIZVxtP
ysIfFBF6RBn3hNQu7tQoGPhiwD1KzQ0NVTkbq7EQwMA7k9FH1QtjjE="/>
        <urn:groove.net:Auth
PTSig="mkLU9YXSukHfi0NOLgaOC7b3ib/gtkEME2HAXGGQviFiAPTkwKg2cLpILDIWcSE8r0aDFOVuKuRQ9PC7D+2/+c
FLe19XIsfs73BIvihJL8rUTJat+BMvBuPWGsPGknI8ohsxYWmETy1ub05gzaQrx30ZN+HpTZjijINvk3+gklWiUE87H5r
xYAM67SCymbvyZwbtm/aqqf7ZaH4sDN/3dVDUsbnPg0tgzRc0An2PeUwLirtC2iYY1hBfnl9k5m5+"/>
    </urn:groove.net:SE>
</urn:groove.net:Del>
```

4.3.3 Compressed Secured Payload

The Secured XML is compressed using WBXML, as specified in [WBXML1.2], to form the Compressed Secured Payload:

```
0000  02 00 00 03 8a 5d 28 6e 75 6c 6c 29 2c 30 00 75  .....](null),0.u
0010  72 6e 3a 67 72 6f 6f 76 65 2e 6e 65 74 3a 44 65  rn:groove.net:De
0020  6c 00 56 65 72 73 69 6f 6e 00 31 2c 30 2c 30 2c  l.Version.1,0,0,
0030  30 00 44 65 70 53 65 71 00 36 42 31 36 43 34 34  0.DepSeq.6B16C44
0040  45 39 37 45 37 33 46 36 43 46 39 45 35 30 30 30  E97E73F6CF9E5000
0050  32 00 53 65 71 00 36 42 31 36 43 34 34 45 39 37  2.Seq.6B16C44E97
0060  45 37 30 31 31 42 33 33 43 34 30 30 30 31 00 47  E7011B33C40001.G
0070  70 00 32 33 00 75 72 6e 3a 67 72 6f 6f 76 65 2e  p.23.urn:groove.
0080  6e 65 74 3a 53 45 00 33 2c 30 2c 30 2c 30 00 75  net:SE.3,0,0,0.u
0090  72 6e 3a 67 72 6f 6f 76 65 2e 6e 65 74 3a 45 43  rn:groove.net:EC
00a0  00 4b 49 44 00 5f 54 4b 49 44 00 4b 56 00 31 00  .KID._TKID.KV.1.
00b0  49 56 00 37 7a 41 73 74 39 31 67 44 69 72 71 74  IV.7zAst91gDirqt
00c0  75 6c 69 61 4b 64 63 76 67 3d 3d 00 45 43 00 5a  uliaKdcvg==.EC.Z
00d0  67 33 6a 68 57 73 68 51 45 72 79 68 53 34 62 45  g3jhWshQEryhS4bE
00e0  45 69 50 61 34 4b 47 55 78 74 4e 6c 31 4c 2b 4a  EiPa4KGUxtNl1L+J
00f0  32 4a 79 72 4c 67 68 47 4a 32 4e 52 36 4f 52 59  2JyrLghGJ2NR6ORY
0100  30 35 53 7a 79 70 6b 6e 71 35 59 68 69 43 39 41  05Szypknq5YhiC9A
0110  58 78 7a 46 46 4f 68 52 2f 49 6a 66 4a 62 71 50  XxzFFOhR/IjfJbqP
0120  49 48 35 37 70 72 6a 74 71 68 41 77 38 68 33 4f  IH57prjtqhAw8h3O
0130  77 34 6e 72 56 65 38 32 6c 41 4f 41 58 30 32 53  w4nrVe82lAOAX02S
0140  34 6d 52 46 4b 69 47 45 32 76 2b 6b 32 34 6a 38  4mRFKiGE2v+k24j8
0150  77 50 61 52 74 58 4c 5a 52 51 50 42 72 57 4f 46  wPaRtXLZRQPBrWOF
0160  4c 4c 70 56 53 65 35 58 53 71 4b 6e 63 4a 42 32  LLpVSe5XSqKncJB2
0170  42 2f 68 77 4e 58 4d 45 66 49 59 49 4b 65 79 49  B/hwNXMEfIYIKeyI
0180  70 6e 6e 4b 68 48 2f 66 36 77 45 77 4b 7a 53 63  pnnKhH/f6wEwKzSc
0190  4f 51 34 36 2b 32 6c 42 66 6f 6c 76 66 6d 6e 77  OQ46+2lBfolvfmnw
01a0  50 33 68 4f 54 36 34 52 4d 47 4d 47 59 79 72 64  P3hOT64RMGMGYyrd
01b0  45 4c 47 4b 61 7a 51 34 5a 61 74 75 66 61 6b 75  ELGKazQ4Zatufaku
01c0  36 79 35 6b 39 4b 64 77 54 67 66 4b 4e 34 69 5a  6y5k9KdwTgfKN4iZ
01d0  61 6a 79 37 4e 33 44 70 77 32 4f 72 6a 50 57 50  ajy7N3Dpw2OrjPWP
01e0  6e 4e 69 52 52 75 38 43 76 30 4f 51 6a 4a 5a 57  nNiRRu8Cv0OQjJZW
01f0  37 65 63 73 2f 74 64 65 2f 4b 31 4e 30 64 6b 41  7ecs/tde/K1N0dkA
0200  62 50 4a 59 77 71 59 43 4b 58 4a 63 72 53 78 77  bPJYwqYCKXJcrSxw
0210  52 31 6e 54 35 74 62 4e 4b 71 37 4e 4a 64 41 53  R1nT5tbNKq7NJdAS
0220  33 55 4c 65 2f 57 61 79 75 64 2f 78 79 36 66 54  3ULe/Wayud/xy6fT
0230  75 7a 68 4e 63 36 33 69 30 4f 76 51 6f 4c 74 4f  uzhNc63i0OvQoLtO
0240  65 46 79 7a 72 37 74 6e 57 4d 55 4b 52 72 4f 4f  eFyzr7tnWMUKRrOO
0250  55 50 71 68 63 2b 64 55 46 4f 44 69 46 35 54 4b  UPqhc+dUFODiF5TK
0260  43 62 2f 62 48 6f 62 43 6e 50 55 45 63 76 77 38  Cb/bHobCnPUEcvw8
```

```
0270  58 37 38 5a 66 35 6c 6d 6f 45 79 4c 59 53 6b 68  X78Zf5lmoEyLYSkh
0280  34 66 69 7a 33 35 72 36 2f 49 30 72 61 79 4c 46  4fiz35r6/I0rayLF
0290  4a 57 31 67 48 67 2b 4f 62 47 67 74 67 4a 70 47  JW1gHg+ObGgtgJpG
02a0  43 33 43 46 63 37 74 6f 45 77 70 76 4e 64 77 35  C3CFc7toEwpvNdw5
02b0  74 4d 41 34 42 52 42 31 42 37 4f 77 4c 7a 43 34  tMA4BRB1B7OwLzC4
02c0  49 46 47 4b 2f 7a 2b 73 37 6d 6f 47 44 71 42 2b  IFGK/z+s7moGDqB+
02d0  44 72 4c 74 2f 6d 75 49 75 76 36 48 52 7a 41 6e  DrLt/muIuv6HRzAn
02e0  48 4b 47 7a 4e 63 4c 66 44 64 61 7a 62 49 50 37  HKGzNcLfDdazbIP7
02f0  50 41 58 46 76 46 43 56 4b 51 50 6b 5a 36 45 31  PAXFvFCVKQPkZ6E1
0300  4f 45 61 69 6c 44 4b 36 44 67 4f 4e 36 70 4e 39  OEailDK6DgON6pN9
0310  37 4f 4a 56 43 44 35 68 32 31 37 31 79 4a 75 7a  7OJVCD5h2171yJuz
0320  6c 56 37 61 72 72 52 71 6e 6a 52 76 59 30 68 55  lV7arrRqnjRvY0hU
0330  36 42 4d 2b 77 6a 7a 61 79 66 6a 52 45 6e 47 4f  6BM+wjzayfjREnGO
0340  65 67 71 52 6d 7a 47 70 4f 50 5a 6f 59 65 4e 6c  egqRmzGpOPZoYeNl
0350  49 64 52 41 49 46 38 69 32 66 71 33 76 6e 70 32  IdRAIF8i2fq3vnp2
0360  7a 65 6b 6f 4c 58 54 75 6b 42 46 55 4c 57 62 54  zekoLXTukBFULWbT
0370  4e 61 48 6d 32 44 58 41 57 52 36 38 43 34 38 38  NaHm2DXAWR68C488
0380  74 62 66 51 52 63 6b 30 65 58 41 76 61 73 62 67  tbfQRck0eXAvasbg
0390  2b 6b 47 53 6d 36 72 61 73 68 5a 5a 44 48 61 35  +kGSm6rashZZDHa5
03a0  30 59 53 62 54 36 35 75 52 55 34 4d 42 6a 4a 57  0YSbT65uRU4MBjJW
03b0  6c 54 57 4e 47 6f 68 43 77 4a 76 50 41 6c 35 4e  lTWNGohCwJvPAl5N
03c0  4d 37 53 61 64 45 71 65 63 6b 34 52 46 36 76 69  M7SadEqeck4RF6vi
03d0  76 37 43 4d 7a 45 71 32 52 37 32 73 42 59 57 52  v7CMzEq2R72sBYWR
03e0  7a 6f 6f 77 6c 4a 4c 6f 55 4f 4d 71 5a 77 4f 51  zoowlJLoUOMqZwOQ
03f0  57 4e 76 69 7a 4c 30 2f 30 48 32 32 2f 32 38 45  WNvizL0/0H22/28E
0400  45 4e 59 59 70 36 6c 51 4b 56 49 5a 56 78 74 50  ENYYp6lQKVIZVxtP
0410  79 73 49 66 46 42 46 36 52 42 6e 33 68 4e 51 75  ysIfFBF6RBn3hNQu
0420  37 74 51 6f 47 50 68 69 77 44 31 4b 7a 51 30 4e  7tQoGPhiwD1KzQ0N
0430  56 54 6b 62 71 37 45 51 77 4d 41 37 6b 39 46 48  VTkbq7EQwMA7k9FH
0440  31 51 74 6a 6a 45 3d 00 75 72 6e 3a 67 72 6f 6f  1QtjjE=.urn:groo
0450  76 65 2e 6e 65 74 3a 41 75 74 68 00 50 54 53 69  ve.net:Auth.PTSi
0460  67 00 6d 6b 4c 55 39 59 58 53 75 6b 48 66 69 30  g.mkLU9YXSukHfi0
0470  4e 4f 4c 67 61 4f 43 37 62 33 69 62 2f 67 74 6b  NOLgaOC7b3ib/gtk
0480  45 4d 45 32 48 41 58 47 47 51 76 69 46 69 41 50  EME2HAXGGQviFiAP
0490  54 6b 77 4b 67 32 63 4c 70 49 4c 44 49 57 63 53  TkwKg2cLpILDIWcS
04a0  45 38 72 30 61 44 46 4f 56 75 4b 75 52 51 39 50  E8r0aDFOVuKuRQ9P
04b0  43 37 44 2b 32 2f 2b 63 46 4c 65 31 39 58 49 73  C7D+2/+cFLe19XIs
04c0  66 73 37 33 42 49 76 69 68 4a 4c 38 72 55 54 4a  fs73BIvihJL8rUTJ
04d0  61 74 2b 42 4d 76 42 75 50 57 47 73 50 47 6b 6e  at+BMvBuPWGsPGkn
04e0  49 38 6f 68 73 78 59 57 6d 45 54 79 31 75 62 30  I8ohsxYWmETy1ub0
04f0  35 67 7a 61 51 72 78 33 30 5a 4e 2b 48 70 54 5a  5gzaQrx30ZN+HpTZ
0500  6a 69 6a 49 4e 76 6b 33 2b 67 6b 6c 57 69 55 45  jijINvk3+gklWiUE
0510  38 37 48 35 72 78 59 41 4d 36 37 53 43 79 6d 62  87H5rxYAM67SCymb
0520  76 79 5a 77 62 74 6d 2f 61 71 71 66 37 5a 61 48  vyZwbtm/aqqf7ZaH
0530  34 73 44 4e 2f 33 64 56 44 55 73 62 6e 50 67 30  4sDN/3dVDUsbnPg0
0540  74 67 7a 52 63 30 41 6e 32 50 65 55 77 4c 69 72  tgzRc0An2PeUwLir
0550  74 43 32 69 59 59 31 68 42 66 6e 6c 39 6b 35 6d  tC2iYY1hBfnl9k5m
0560  35 2b 00 c4 09 04 1c 83 24 04 2c 83 33 04 4c 83  5+......$.,.3.L.
0570  50 04 69 83 6c 01 c4 6f 04 1c 83 81 01 01 84 81  P.i.l..o........
0580  09 04 81 1b 83 81 1f 04 81 25 83 81 28 04 81 2a  .........%..(..*
0590  83 81 2d 04 81 46 83 81 49 01 84 88 42 04 88 56  ..-..F..I...B..V
05a0  83 88 5c 01 01 01                               ..\...
```

4.3.4 MIME-like Wrapper

The MIME-like wrapper header and epilogue are added to the Compressed Secured Payload to produce the final Delta message:

```
0000  4d 49 4d 45 2d 56 65 72 73 69 6f 6e 3a 20 31 2e  MIME-Version: 1.
0010  30 20 28 47 72 6f 6f 76 65 20 32 29 0d 0a 43 6f  0 (Groove 2)..Co
0020  6e 74 65 6e 74 2d 54 79 70 65 3a 20 6d 75 6c 74  ntent-Type: mult
0030  69 70 61 72 74 2f 72 65 6c 61 74 65 64 3b 20 62  ipart/related; b
0040  6f 75 6e 64 61 72 79 3d 22 3c 3c 5b 5b 26 26 26  oundary="<<[[&&&
0050  5d 5d 3e 3e 22 0d 0a 3c 3c 5b 5b 26 26 26 5d 5d  ]]>>"..<<[[&&&]]
0060  3e 3e 0d 0a 43 6f 6e 74 65 6e 74 2d 54 79 70 65  >>..Content-Type
0070  3a 20 61 70 70 6c 69 63 61 74 69 6f 6e 2f 57 42  : application/WB
0080  58 4d 4c 3b 20 63 68 61 72 73 65 74 3d 22 75 73  XML; charset="us
0090  2d 61 73 63 69 69 22 0d 0a 02 00 00 03 8a 5d 28  -ascii".......](
00a0  6e 75 6c 6c 29 2c 30 00 75 72 6e 3a 67 72 6f 6f  null),0.urn:groo
00b0  76 65 2e 6e 65 74 3a 44 65 6c 00 56 65 72 73 69  ve.net:Del.Versi
00c0  6f 6e 00 31 2c 30 2c 30 2c 30 00 44 65 70 53 65  on.1,0,0,0.DepSe
00d0  71 00 36 42 31 36 43 34 34 45 39 37 45 37 33 46  q.6B16C44E97E73F
00e0  36 43 46 39 45 35 30 30 30 32 00 53 65 71 00 36  6CF9E50002.Seq.6
00f0  42 31 36 43 34 34 45 39 37 45 37 30 31 31 42 33  B16C44E97E7011B3
0100  33 43 34 30 30 30 31 00 47 70 00 32 33 00 75 72  3C40001.Gp.23.ur
0110  6e 3a 67 72 6f 6f 76 65 2e 6e 65 74 3a 53 45 00  n:groove.net:SE.
0120  33 2c 30 2c 30 2c 30 00 75 72 6e 3a 67 72 6f 6f  3,0,0,0.urn:groo
0130  76 65 2e 6e 65 74 3a 45 43 00 4b 49 44 00 5f 54  ve.net:EC.KID._T
0140  4b 49 44 00 4b 56 00 31 00 49 56 00 37 7a 41 73  KID.KV.1.IV.7zAs
0150  74 39 31 67 44 69 72 71 74 75 6c 69 61 4b 64 63  t91gDirqtuliaKdc
0160  76 67 3d 3d 00 45 43 00 5a 67 33 6a 68 57 73 68  vg==.EC.Zg3jhWsh
0170  51 45 72 79 68 53 34 62 45 45 69 50 61 34 4b 47  QEryhS4bEEiPa4KG
0180  55 78 74 4e 6c 31 4c 2b 4a 32 4a 79 72 4c 67 68  UxtNl1L+J2JyrLgh
0190  47 4a 32 4e 52 36 4f 52 59 30 35 53 7a 79 70 6b  GJ2NR6ORY05Szypk
01a0  6e 71 35 59 68 69 43 39 41 58 78 7a 46 46 4f 68  nq5YhiC9AXxzFFOh
01b0  52 2f 49 6a 66 4a 62 71 50 49 48 35 37 70 72 6a  R/IjfJbqPIH57prj
01c0  74 71 68 41 77 38 68 33 4f 77 34 6e 72 56 65 38  tqhAw8h3Ow4nrVe8
01d0  32 6c 41 4f 41 58 30 32 53 34 6d 52 46 4b 69 47  2lAOAX02S4mRFKiG
01e0  45 32 76 2b 6b 32 34 6a 38 77 50 61 52 74 58 4c  E2v+k24j8wPaRtXL
01f0  5a 52 51 50 42 72 57 4f 46 4c 4c 70 56 53 65 35  ZRQPBrWOFLLpVSe5
0200  58 53 71 4b 6e 63 4a 42 32 42 2f 68 77 4e 58 4d  XSqKncJB2B/hwNXM
0210  45 66 49 59 49 4b 65 79 49 70 6e 6e 4b 68 48 2f  EfIYIKeyIpnnKhH/
0220  66 36 77 45 77 4b 7a 53 63 4f 51 34 36 2b 32 6c  f6wEwKzScOQ46+2l
0230  42 66 6f 6c 76 66 6d 6e 77 50 33 68 4f 54 36 34  BfolvfmnwP3hOT64
0240  52 4d 47 4d 47 59 79 72 64 45 4c 47 4b 61 7a 51  RMGMGYyrdELGKazQ
0250  34 5a 61 74 75 66 61 6b 75 36 79 35 6b 39 4b 64  4Zatufaku6y5k9Kd
0260  77 54 67 66 4b 4e 34 69 5a 61 6a 79 37 4e 33 44  wTgfKN4iZajy7N3D
0270  70 77 32 4f 72 6a 50 57 50 6e 4e 69 52 52 75 38  pw2OrjPWPnNiRRu8
0280  43 76 30 4f 51 6a 4a 5a 57 37 65 63 73 2f 74 64  Cv0OQjJZW7ecs/td
0290  65 2f 4b 31 4e 30 64 6b 41 62 50 4a 59 77 71 59  e/K1N0dkAbPJYwqY
02a0  43 4b 58 4a 63 72 53 78 77 52 31 6e 54 35 74 62  CKXJcrSxwR1nT5tb
02b0  4e 4b 71 37 4e 4a 64 41 53 33 55 4c 65 2f 57 61  NKq7NJdAS3ULe/Wa
02c0  79 75 64 2f 78 79 36 66 54 75 7a 68 4e 63 36 33  yud/xy6fTuzhNc63
02d0  69 30 4f 76 51 6f 4c 74 4f 65 46 79 7a 72 37 74  i0OvQoLtOeFyzr7t
02e0  6e 57 4d 55 4b 52 72 4f 4f 55 50 71 68 63 2b 64  nWMUKRrOOUPqhc+d
02f0  55 46 4f 44 69 46 35 54 4b 43 62 2f 62 48 6f 62  UFODiF5TKCb/bHob
0300  43 6e 50 55 45 63 76 77 38 58 37 38 5a 66 35 6c  CnPUEcvw8X78Zf5l
0310  6d 6f 45 79 4c 59 53 6b 68 34 66 69 7a 33 35 72  moEyLYSkh4fiz35r
0320  36 2f 49 30 72 61 79 4c 46 4a 57 31 67 48 67 2b  6/I0rayLFJW1gHg+
0330  4f 62 47 67 74 67 4a 70 47 43 33 43 46 63 37 74  ObGgtgJpGC3CFc7t
0340  6f 45 77 70 76 4e 64 77 35 74 4d 41 34 42 52 42  oEwpvNdw5tMA4BRB
0350  31 42 37 4f 77 4c 7a 43 34 49 46 47 4b 2f 7a 2b  1B7OwLzC4IFGK/z+
0360  73 37 6d 6f 47 44 71 42 2b 44 72 4c 74 2f 6d 75  s7moGDqB+DrLt/mu
0370  49 75 76 36 48 52 7a 41 6e 48 4b 47 7a 4e 63 4c  Iuv6HRzAnHKGzNcL
0380  66 44 64 61 7a 62 49 50 37 50 41 58 46 76 46 43  fDdazbIP7PAXFvFC
0390  56 4b 51 50 6b 5a 36 45 31 4f 45 61 69 6c 44 4b  VKQPkZ6E1OEailDK
03a0  36 44 67 4f 4e 36 70 4e 39 37 4f 4a 56 43 44 35  6DgON6pN97OJVCD5
```

```
03b0  68 32 31 37 31 79 4a 75 7a 6c 56 37 61 72 72 52  h2171yJuzlV7arrR
03c0  71 6e 6a 52 76 59 30 68 55 36 42 4d 2b 77 6a 7a  qnjRvY0hU6BM+wjz
03d0  61 79 66 6a 52 45 6e 47 4f 65 67 71 52 6d 7a 47  ayfjREnGOegqRmzG
03e0  70 4f 50 5a 6f 59 65 4e 6c 49 64 52 41 49 46 38  pOPZoYeNlIdRAIF8
03f0  69 32 66 71 33 76 6e 70 32 7a 65 6b 6f 4c 58 54  i2fq3vnp2zekoLXT
0400  75 6b 42 46 55 4c 57 62 54 4e 61 48 6d 32 44 58  ukBFULWbTNaHm2DX
0410  41 57 52 36 38 43 34 38 38 74 62 66 51 52 63 6b  AWR68C488tbfQRck
0420  30 65 58 41 76 61 73 62 67 2b 6b 47 53 6d 36 72  0eXAvasbg+kGSm6r
0430  61 73 68 5a 5a 44 48 61 35 30 59 53 62 54 36 35  ashZZDHa50YSbT65
0440  75 52 55 34 4d 42 6a 4a 57 6c 54 57 4e 47 6f 68  uRU4MBjJWlTWNGoh
0450  43 77 4a 76 50 41 6c 35 4e 4d 37 53 61 64 45 71  CwJvPAl5NM7SadEq
0460  65 63 6b 34 52 46 36 76 69 76 37 43 4d 7a 45 71  eck4RF6viv7CMzEq
0470  32 52 37 32 73 42 59 57 52 7a 6f 6f 77 6c 4a 4c  2R72sBYWRzoowlJL
0480  6f 55 4f 4d 71 5a 77 4f 51 57 4e 76 69 7a 4c 30  oUOMqZwOQWNvizL0
0490  2f 30 48 32 32 2f 32 38 45 45 4e 59 59 70 36 6c  /0H22/28EENYYp6l
04a0  51 4b 56 49 5a 56 78 74 50 79 73 49 66 46 42 46  QKVIZVxtPysIfFBF
04b0  36 52 42 6e 33 68 4e 51 75 37 74 51 6f 47 50 68  6RBn3hNQu7tQoGPh
04c0  69 77 44 31 4b 7a 51 30 4e 56 54 6b 62 71 37 45  iwD1KzQ0NVTkbq7E
04d0  51 77 4d 41 37 6b 39 46 48 31 51 74 6a 6a 45 3d  QwMA7k9FH1QtjjE=
04e0  00 75 72 6e 3a 67 72 6f 6f 76 65 2e 6e 65 74 3a  .urn:groove.net:
04f0  41 75 74 68 00 50 54 53 69 67 00 6d 6b 4c 55 39  Auth.PTSig.mkLU9
0500  59 58 53 75 6b 48 66 69 30 4e 4f 4c 67 61 4f 43  YXSukHfi0NOLgaOC
0510  37 62 33 69 62 2f 67 74 6b 45 4d 45 32 48 41 58  7b3ib/gtkEME2HAX
0520  47 47 51 76 69 46 69 41 50 54 6b 77 4b 67 32 63  GGQviFiAPTkwKg2c
0530  4c 70 49 4c 44 49 57 63 53 45 38 72 30 61 44 46  LpILDIWcSE8r0aDF
0540  4f 56 75 4b 75 52 51 39 50 43 37 44 2b 32 2f 2b  OVuKuRQ9PC7D+2/+
0550  63 46 4c 65 31 39 58 49 73 66 73 37 33 42 49 76  cFLe19XIsfs73BIv
0560  69 68 4a 4c 38 72 55 54 4a 61 74 2b 42 4d 76 42  ihJL8rUTJat+BMvB
0570  75 50 57 47 73 50 47 6b 6e 49 38 6f 68 73 78 59  uPWGsPGknI8ohsxY
0580  57 6d 45 54 79 31 75 62 30 35 67 7a 61 51 72 78  WmETy1ub05gzaQrx
0590  33 30 5a 4e 2b 48 70 54 5a 6a 69 6a 49 4e 76 6b  30ZN+HpTZjijINvk
05a0  33 2b 67 6b 6c 57 69 55 45 38 37 48 35 72 78 59  3+gklWiUE87H5rxY
05b0  41 4d 36 37 53 43 79 6d 62 76 79 5a 77 62 74 6d  AM67SCymbvyZwbtm
05c0  2f 61 71 71 66 37 5a 61 48 34 73 44 4e 2f 33 64  /aqqf7ZaH4sDN/3d
05d0  56 44 55 73 62 6e 50 67 30 74 67 7a 52 63 30 41  VDUsbnPg0tgzRc0A
05e0  6e 32 50 65 55 77 4c 69 72 74 43 32 69 59 59 31  n2PeUwLirtC2iYY1
05f0  68 42 66 6e 6c 39 6b 35 6d 35 2b 00 c4 09 04 1c  hBfnl9k5m5+.....
0600  83 24 04 2c 83 33 04 4c 83 50 04 69 83 6c 01 c4  .$.,.3.L.P.i.l..
0610  6f 04 1c 83 81 01 01 84 81 09 04 81 1b 83 81 1f  o...............
0620  04 81 25 83 81 28 04 81 2a 83 81 2d 04 81 46 83  ..%..(..*..-..F.
0630  81 49 01 84 88 42 04 88 56 83 88 5c 01 01 01 0d  .I...B..V..\....
0640  0a 2d 2d 3c 3c 5b 5b 26 26 26 5d 5d 3e 3e 2d 2d  .--<<[[&&&]]>>--
0650  0d 0a                                            ..
```

4.4 Processing an Incoming Delta Ack Message

This example follows from section 4.3, where the outgoing **Delta** message was encoded and disseminated. After receiving this message, another endpoint will generate and send a **Delta Ack** message (section 2.2.2) to the local endpoint. This example illustrates four stages in the decoding of this message.

4.4.1 MIME-like Wrapper

```
0000  4d 49 4d 45 2d 56 65 72 73 69 6f 6e 3a 20 31 2e  MIME-Version: 1.
0010  30 20 28 47 72 6f 6f 76 65 20 32 29 0d 0a 43 6f  0 (Groove 2)..Co
0020  6e 74 65 6e 74 2d 54 79 70 65 3a 20 6d 75 6c 74  ntent-Type: mult
0030  69 70 61 72 74 2f 72 65 6c 61 74 65 64 3b 20 62  ipart/related; b
```

```
0040  6f 75 6e 64 61 72 79 3d 22 3c 3c 5b 5b 26 26 26  oundary="<<[[&&&
0050  5d 5d 3e 3e 22 0d 0a 3c 3c 5b 5b 26 26 26 5d 5d  ]]>>"..<<[[&&&]]
0060  3e 3e 0d 0a 43 6f 6e 74 65 6e 74 2d 54 79 70 65  >>..Content-Type
0070  3a 20 61 70 70 6c 69 63 61 74 69 6f 6e 2f 57 42  : application/WB
0080  58 4d 4c 3b 20 63 68 61 72 73 65 74 3d 22 75 73  XML; charset="us
0090  2d 61 73 63 69 69 22 0d 0a 02 00 00 03 85 0b 28  -ascii".........(
00a0  6e 75 6c 6c 29 2c 30 00 44 65 6c 41 63 6b 00 44  null),0.DelAck.D
00b0  65 70 53 65 71 00 36 42 31 36 43 34 34 45 39 37  epSeq.6B16C44E97
00c0  45 37 30 31 31 42 33 33 43 34 30 30 30 31 00 44  E7011B33C40001.D
00d0  65 76 69 63 65 55 52 4c 00 64 70 70 3a 2f 2f 2f  eviceURL.dpp:///
00e0  77 37 78 68 6a 72 72 72 34 74 73 35 32 61 71 33  w7xhjrrr4ts52aq3
00f0  39 38 62 78 72 37 68 7a 6b 74 70 64 35 70 6d 72  98bxr7hzktpd5pmr
0100  34 6e 62 74 78 61 69 00 43 6f 6e 74 61 63 74 55  4nbtxai.ContactU
0110  52 4c 00 67 72 6f 6f 76 65 49 64 65 6e 74 69 74  RL.grooveIdentit
0120  79 3a 2f 2f 6b 65 38 78 79 35 61 71 72 7a 63 77  y://ke8xy5aqrzcw
0130  66 35 6b 69 65 66 33 35 64 72 6a 38 32 65 35 78  f5kief35drj82e5x
0140  6d 76 74 38 40 00 47 70 00 32 33 00 75 72 6e 3a  mvt8@.Gp.23.urn:
0150  67 72 6f 6f 76 65 2e 6e 65 74 3a 53 45 00 56 65  groove.net:SE.Ve
0160  72 73 69 6f 6e 00 33 2c 30 2c 30 2c 30 00 75 72  rsion.3,0,0,0.ur
0170  6e 3a 67 72 6f 6f 76 65 2e 6e 65 74 3a 45 43 00  n:groove.net:EC.
0180  4b 49 44 00 5f 54 4b 49 44 00 4b 56 00 31 00 49  KID._TKID.KV.1.I
0190  56 00 54 69 67 49 77 44 6c 78 37 57 38 4f 70 73  V.TigIwDlx7W8Ops
01a0  64 54 31 79 54 46 45 77 3d 3d 00 45 43 00 52 69  dT1yTFEw==.EC.Ri
01b0  47 66 69 4c 73 64 41 2f 6d 69 69 35 73 4b 6e 50  GfiLsdA/mii5sKnP
01c0  33 66 77 46 37 62 2b 68 34 59 2f 75 57 30 35 38  3fwF7b+h4Y/uW058
01d0  69 52 77 41 6f 67 66 62 31 77 67 58 43 6f 37 45  iRwAogfb1wgXCo7E
01e0  42 58 34 55 4d 4a 74 69 66 75 74 41 77 6e 6f 54  BX4UMJtifutAwnoT
01f0  69 42 72 53 36 6e 64 63 4d 2f 56 43 5a 74 67 49  iBrS6ndcM/VCZtgI
0200  6b 57 45 51 6d 4c 46 51 78 62 59 51 3d 3d 00 75  kWEQmLFQxbYQ==.u
0210  72 6e 3a 67 72 6f 6f 76 65 2e 6e 65 74 3a 41 75  rn:groove.net:Au
0220  74 68 00 50 54 53 69 67 00 62 4d 61 64 49 75 35  th.PTSig.bMadIu5
0230  37 79 77 68 36 42 65 66 49 77 4e 78 30 68 52 31  7ywh6BefIwNx0hR1
0240  51 6e 51 41 56 44 59 48 77 76 72 4a 73 50 65 67  QnQAVDYHwvrJsPeg
0250  2b 4f 7a 47 6e 72 58 44 2f 58 30 61 52 43 45 77  +OzGnrXD/X0aRCEw
0260  6a 65 4c 77 6e 2f 48 52 57 49 43 6b 32 46 51 4d  jeLwn/HRWICk2FQM
0270  68 71 48 46 7a 58 49 49 41 56 51 6e 57 4a 2f 55  hqHFzXIIAVQnWJ/U
0280  50 7a 4e 55 44 65 76 4c 52 6b 47 6a 6b 77 78 38  PzNUDevLRkGjkwx8
0290  31 59 73 6f 66 66 42 5a 67 6c 64 36 36 76 6b 36  1YsoffBZgld66vk6
02a0  69 6f 67 5a 51 72 6b 76 65 5a 72 59 51 52 64 39  iogZQrkveZrYQRd9
02b0  53 47 61 5a 6a 4a 79 49 6d 66 50 65 42 54 67 4e  SGaZjJyImfPeBTgN
02c0  69 57 70 79 4e 6d 32 46 6e 5a 75 79 55 51 39 78  iWpyNm2FnZuyUQ9x
02d0  6e 32 6c 68 30 2b 4f 6a 2b 4c 70 6c 38 69 49 38  n2lh0+Oj+Lpl8iI8
02e0  57 72 58 61 52 44 54 71 64 76 46 79 4e 4f 54 46  WrXaRDTqdvFyNOTF
02f0  71 6e 2b 38 34 57 79 50 4b 79 50 57 39 54 75 64  qn+84WyPKyPW9Tud
0300  58 30 65 30 79 65 4e 7a 59 45 6b 71 6d 51 4e 55  X0e0yeNzYEkqmQNU
0310  72 6d 59 71 52 37 6f 68 78 34 41 57 71 53 4f 7a  rmYqR7ohx4AWqSOz
0320  77 59 71 42 74 7a 79 64 6d 00 c4 09 04 10 83 17  wYqBtzydm.......
0330  04 30 83 3a 04 69 83 74 04 81 27 83 81 2a 01 c4  .0.:.i.t..'..*..
0340  81 2d 04 81 3f 83 81 47 01 84 81 4f 04 81 61 83  .-..?..G...O..a.
0350  81 65 04 81 6b 83 81 6e 04 81 70 83 81 73 04 82  .e..k..n..p..s..
0360  0c 83 82 0f 01 84 82 70 04 83 04 83 83 0a 01 01  .......p........
0370  01 0d 0a 2d 2d 3c 3c 5b 5b 26 26 26 5d 5d 3e 3e  ...--<<[[&&&]]>>
0380  2d 2d 0d 0a                                      --..
```

4.4.2 Compressed Secured Payload

The MIME-like wrapper header and epilogue are stripped, leaving the following Compressed Secured Payload, which is a WBXML stream, as specified in [WBXML1.2]:

```
0000  02 00 00 03 85 0b 28 6e 75 6c 6c 29 2c 30 00 44  ......(null),0.D
0010  65 6c 41 63 6b 00 44 65 70 53 65 71 00 36 42 31  elAck.DepSeq.6B1
0020  36 43 34 34 45 39 37 45 37 30 31 31 42 33 33 43  6C44E97E7011B33C
0030  34 30 30 30 31 00 44 65 76 69 63 65 55 52 4c 00  40001.DeviceURL.
0040  64 70 70 3a 2f 2f 2f 77 37 78 68 6a 72 72 72 34  dpp:///w7xhjrrr4
0050  74 73 35 32 61 71 33 39 38 62 78 72 37 68 7a 6b  ts52aq398bxr7hzk
0060  74 70 64 35 70 6d 72 34 6e 62 74 78 61 69 00 43  tpd5pmr4nbtxai.C
0070  6f 6e 74 61 63 74 55 52 4c 00 67 72 6f 6f 76 65  ontactURL.groove
0080  49 64 65 6e 74 69 74 79 3a 2f 2f 6b 65 38 78 79  Identity://ke8xy
0090  35 61 71 72 7a 63 77 66 35 6b 69 65 66 33 35 64  5aqrzcwf5kief35d
00a0  72 6a 38 32 65 35 78 6d 76 74 38 40 00 47 70 00  rj82e5xmvt8@.Gp.
00b0  32 33 00 75 72 6e 3a 67 72 6f 6f 76 65 2e 6e 65  23.urn:groove.ne
00c0  74 3a 53 45 00 56 65 72 73 69 6f 6e 00 33 2c 30  t:SE.Version.3,0
00d0  2c 30 2c 30 00 75 72 6e 3a 67 72 6f 6f 76 65 2e  ,0,0.urn:groove.
00e0  6e 65 74 3a 45 43 00 4b 49 44 00 5f 54 4b 49 44  net:EC.KID._TKID
00f0  00 4b 56 00 31 00 49 56 00 54 69 67 49 77 44 6c  .KV.1.IV.TigIwDl
0100  78 37 57 38 4f 70 73 64 54 31 79 54 46 45 77 3d  x7W8OpsdT1yTFEw=
0110  3d 00 45 43 00 52 69 47 66 69 4c 73 64 41 2f 6d  =.EC.RiGfiLsdA/m
0120  69 69 35 73 4b 6e 50 33 66 77 46 37 62 2b 68 34  ii5sKnP3fwF7b+h4
0130  59 2f 75 57 30 35 38 69 52 77 41 6f 67 66 62 31  Y/uW058iRwAogfb1
0140  77 67 58 43 6f 37 45 42 58 34 55 4d 4a 74 69 66  wgXCo7EBX4UMJtif
0150  75 74 41 77 6e 6f 54 69 42 72 53 36 6e 64 63 4d  utAwnoTiBrS6ndcM
0160  2f 56 43 5a 74 67 49 6b 57 45 51 6d 4c 46 51 78  /VCZtgIkWEQmLFQx
0170  62 59 51 3d 3d 00 75 72 6e 3a 67 72 6f 6f 76 65  bYQ==.urn:groove
0180  2e 6e 65 74 3a 41 75 74 68 00 50 54 53 69 67 00  .net:Auth.PTSig.
0190  62 4d 61 64 49 75 35 37 79 77 68 36 42 65 66 49  bMadIu57ywh6BefI
01a0  77 4e 78 30 68 52 31 51 6e 51 41 56 44 59 48 77  wNx0hR1QnQAVDYHw
01b0  76 72 4a 73 50 65 67 2b 4f 7a 47 6e 72 58 44 2f  vrJsPeg+OzGnrXD/
01c0  58 30 61 52 43 45 77 6a 65 4c 77 6e 2f 48 52 57  X0aRCEwjeLwn/HRW
01d0  49 43 6b 32 46 51 4d 68 71 48 46 7a 58 49 49 41  ICk2FQMhqHFzXIIA
01e0  56 51 6e 57 4a 2f 55 50 7a 4e 55 44 65 76 4c 52  VQnWJ/UPzNUDevLR
01f0  6b 47 6a 6b 77 78 38 31 59 73 6f 66 66 42 5a 67  kGjkwx81YsoffBZg
0200  6c 64 36 36 76 6b 36 69 6f 67 5a 51 72 6b 76 65  ld66vk6iogZQrkve
0210  5a 72 59 51 52 64 39 53 47 61 5a 6a 4a 79 49 6d  ZrYQRd9SGaZjJyIm
0220  66 50 65 42 54 67 4e 69 57 70 79 4e 6d 32 46 6e  fPeBTgNiWpyNm2Fn
0230  5a 75 79 55 51 39 78 6e 32 6c 68 30 2b 4f 6a 2b  ZuyUQ9xn2lh0+Oj+
0240  4c 70 6c 38 69 49 38 57 72 58 61 52 44 54 71 64  Lpl8iI8WrXaRDTqd
0250  76 46 79 4e 4f 54 46 71 6e 2b 38 34 57 79 50 4b  vFyNOTFqn+84WyPK
0260  79 50 57 39 54 75 64 58 30 65 30 79 65 4e 7a 59  yPW9TudX0e0yeNzY
0270  45 6b 71 6d 51 4e 55 72 6d 59 71 52 37 6f 68 78  EkqmQNUrmYqR7ohx
0280  34 41 57 71 53 4f 7a 77 59 71 42 74 7a 79 64 6d  4AWqSOzwYqBtzydm
0290  00 c4 09 04 10 83 17 04 30 83 3a 04 69 83 74 04  ........0.:.i.t.
02a0  81 27 83 81 2a 01 c4 81 2d 04 81 3f 83 81 47 01  .'..*...-..?..G.
02b0  84 81 4f 04 81 61 83 81 65 04 81 6b 83 81 6e 04  ..O..a..e..k..n.
02c0  81 70 83 81 73 04 82 0c 83 82 0f 01 84 82 70 04  .p..s.........p.
02d0  83 04 83 83 0a 01 01 01                          ........
```

4.4.3 Secured XML

The WBXML stream, as specified in [WBXML1.2], comprising the Compressed Secured Payload is decoded into the following Secured XML:

```
<DelAck Gp="23" ContactURL="grooveIdentity://ke8xy5aqrzcwf5kief35drj82e5xmvt8@"
DeviceURL="dpp:///w7xhjrrr4ts52aq398bxr7hzktpd5pmr4nbtxai" DepSeq="6B16C44E97E7011B33C40001">
    <urn:groove.net:SE Version="3,0,0,0">
```

```
        <urn:groove.net:EC
EC="RiGfiLsdA/mii5sKnP3fwF7b+h4Y/uW058iRwAogfb1wgXCo7EBX4UMJtifutAwnoTiBrS6ndcM/VCZtgIkWEQmLF
QxbYQ==" IV="TigIwDlx7W8OpsdT1yTFEw==" KV="1" KID="_TKID"/>
        <urn:groove.net:Auth
PTSig="bMadIu57ywh6BefIwNx0hR1QnQAVDYHwvrJsPeg+OzGnrXD/X0aRCEwjeLwn/HRWICk2FQMhqHFzXIIAVQnWJ/
UPzNUDevLRkGjkwx81YsoffBZgld66vk6iogZQrkveZrYQRd9SGaZjJyImfPeBTgNiWpyNm2FnZuyUQ9xn2lh0+Oj+Lpl
8iI8WrXaRDTqdvFyNOTFqn+84WyPKyPW9TudX0e0yeNzYEkqmQNUrmYqR7ohx4AWqSOzwYqBtzydm"/>
    </urn:groove.net:SE>
</DelAck>
```

4.4.4 DelAck XML

The binary content of the Secured XML, embedded in the EC attribute of the element with the name urn:groove.net:EC, is Base64-decoded and decrypted, producing the following **DelAck** XML:

```
<DelAck Gp="23" ContactURL="grooveIdentity://ke8xy5aqrzcwf5kief35drj82e5xmvt8@"
DeviceURL="dpp:///w7xhjrrr4ts52aq398bxr7hzktpd5pmr4nbtxai" DepSeq="6B16C44E97E7011B33C40001">
    <DelAckBody SenderRank="99" SenderMinDep="23" PurGrp="22"/>
</DelAck>
```

4.5 Simple Delta Ordering

The following describe the six deltas generated by the example from the delta graph in section 3.1.1. The endpoint UID for A is "E9641419D18C", for B is "6401C37EFB36" and for C is "E2D20DF7D85D". This results in a final ordering of A1, A2, B1, B2, C1, A3. All of these deltas contain one command from the engine with Engine URL of "Dynamics". This engine is responsible for the **CMD** and **TestId** attributes on the **urn:groove.net:Cmd** elements. The engine requires notification of purge for all of its commands.

Delta A1: This is not the first delta generated in the space, which is why it has an explicit **DepSeq**, a **Gp** of 3, a sequence number of 7, Rank of 11 and explicit SpStSet. The Seq attribute was constructed by concatenating A's endpoint UID ("E9641419D18C"), A's current creator identifier ("02B9495F") and the sequence number (0007).

```
<urn:groove.net:Del DepSeq="E2D20DF7D85D3E419CCD0002" Gp="3" Seq="E9641419D18C02B9495F0007"
Version="1,0,0,0">
    <urn:groove.net:Cmds PurGrp="0" PurNot="" Rank="11" SenderMinDep="1"
SpStSet="8;2;0;E9641419D18C02B9495F0006;6401C37EFB360000;" TimeCreated="1201037037430">
        <urn:groove.net:Cmd CMD="7" EngineURL="Dynamics" PurNot=""
TestId="759EF7B5C21DCB62"/>
    </urn:groove.net:Cmds>
</urn:groove.net:Del>
```

Delta A2: A has not received any deltas since creating delta A1, so A1 is the only immediate **dependency** of this delta. Because A1 is the previously generated delta from this **endpoint** and A1 and A2 have the same creator identifier, A1 is an implicit dependency and the DepSeq attribute is not set. A2 can have the same **group** as A1, so the **Gp** attribute is set to 3. The sequence number for A2 is 8, one more than the sequence number of A1. The rank is set to 12, one higher than the previous highest rank. A's space state information has not changed, so this delta doesn't have a SpStSet attribute. The new space state for A can be computed from other attributes on the delta. Even though the only immediate dependency of this delta has group 3, A chooses to only set the **SenderMinDep** to 1.

```
<urn:groove.net:Del Gp="3" Seq="E9641419D18C02B9495F0008" Version="1,0,0,0">
```

```
    <urn:groove.net:Cmds PurGrp="0" PurNot="" Rank="12" SenderMinDep="1"
TimeCreated="1201037037649">
        <urn:groove.net:Cmd CMD="7" EngineURL="Dynamics" PurNot=""
TestId="182C6C2419CE089F"/>
    </urn:groove.net:Cmds>
</urn:groove.net:Del>
```

Delta B1: This delta depends on A1. Because B's endpoint UID is less than A's, this delta could not go in the same group. As a result the **Gp** attribute is 4. This caused this delta to be ordered after A2. The **SpStSet** duplicates the information sent with A1. It specifies that endpoint A has sent a delta with rank 11, min dependency group of 1, purge group of 0, and that A's next dependency will be on A1. Because this information had already been sent on the delta, there is no value in having it sent again.

```
<urn:groove.net:Del DepSeq="E9641419D18C02B9495F0007" Gp="4" Seq="6401C37EFB366A87F4210003"
Version="1,0,0,0">
    <urn:groove.net:Cmds PurGrp="0" PurNot="" Rank="12" SenderMinDep="2"
SpStSet="11;1;0;E9641419D18C02B9495F0007;E9641419D18C0000;" TimeCreated="1201037037992">
        <urn:groove.net:Cmd CMD="7" EngineURL="Dynamics" PurNot=""
TestId="48369E7BE594B678"/>
    </urn:groove.net:Cmds>
</urn:groove.net:Del>
```

Delta B2: This delta has an implicit dependency on B1.

```
<urn:groove.net:Del Gp="4" Seq="6401C37EFB366A87F4210004" Version="1,0,0,0">
    <urn:groove.net:Cmds PurGrp="0" PurNot="" Rank="13" SenderMinDep="2"
TimeCreated="1201037038242">
        <urn:groove.net:Cmd CMD="7" EngineURL="Dynamics" PurNot=""
TestId="7FC378554217F394"/>
    </urn:groove.net:Cmds>
</urn:groove.net:Del>
```

Delta C1: This delta has explicit dependencies on both A2 and B1. At the time C generated this delta, B1 was the last delta in its delta log. Because the endpoint UID for C is greater than the endpoint UID of B, this delta could go in the same group, so Gp is set to 4. The space state set includes information for both A and B. This duplicates information that had already been set on the deltas that they created, so it is not necessary to set it on this delta.

```
<urn:groove.net:Del DepSeq="E9641419D18C02B9495F0008,6401C37EFB366A87F4210003" Gp="4"
Seq="E2D20DF7D85D3E419CCD0003" Version="1,0,0,0">
    <urn:groove.net:Cmds PurGrp="0" PurNot="" Rank="13" SenderMinDep="3"
SpStSet="12;1;0;E9641419D18C02B9495F0008;E9641419D18C0000;12;2;0;6401C37EFB366A87F4210003;640
1C37EFB360000;" TimeCreated="1201037038117">
        <urn:groove.net:Cmd CMD="7" EngineURL="Dynamics" PurNot=""
TestId="6BC67CB8D94B31CD"/>
    </urn:groove.net:Cmds>
</urn:groove.net:Del>
```

Delta A3: This delta has an explicit dependency on C1. Because A's endpoint UID is greater than C's this delta can go in the same group. The space state set includes information for both B and C. This duplicates information that had already been set on the deltas that they created, so it is not necessary to set it on this delta.

```
<urn:groove.net:Del DepSeq="E2D20DF7D85D3E419CCD0003" Gp="4" Seq="E9641419D18C02B9495F0009"
Version="1,0,0,0">
    <urn:groove.net:Cmds PurGrp="0" PurNot="" Rank="14" SenderMinDep="1"
SpStSet="12;2;0;6401C37EFB366A87F4210003;6401C37EFB360000;13;3;0;E2D20DF7D85D3E419CCD0003;E2D
20DF7D85D0000;" TimeCreated="1201037038352">
        <urn:groove.net:Cmd CMD="7" EngineURL="Dynamics" PurNot=""
TestId="AC571FA90B2ED5B8"/>
    </urn:groove.net:Cmds>
</urn:groove.net:Del>
```

4.6 Priority Delta Ordering

This example is the same as in section 4.1 except that deltas C1 and A3 have an explicit assimilation priority. The endpoint UIDs are the same, but there are new creator identifiers. There are no significant changes to deltas A1, A2, B1 or B2. Because of the priority on deltas C1 and A3 the order of assimilation is different. The ordering is A1, A2, B1, C1, B2, A3. The assimilation priorities cause two additional blocks to be created. The first contains A1, A2 and B1. The second contains C1. The final block contains B2 and A3. C1 and A3 are block deltas. B2 is ordered in the last block because A3 does not depend on it.

Delta A1:

```
<urn:groove.net:Del DepSeq="E2D20DF7D85D27460B3E0002" Gp="3" Seq="E9641419D18C367218970007"
Version="1,0,0,0">
  <urn:groove.net:Cmds PurGrp="0" PurNot="" Rank="11" SenderMinDep="1"
SpStSet="8;2;0;E9641419D18C367218970006;6401C37EFB360000;" TimeCreated="1201119771044">
    <urn:groove.net:Cmd CMD="7" EngineURL="Dynamics" PurNot="" TestId="07BABB29CA877BF0"/>
  </urn:groove.net:Cmds>
</urn:groove.net:Del>
```

Delta A2:

```
<urn:groove.net:Del Gp="3" Seq="E9641419D18C367218970008" Version="1,0,0,0">
  <urn:groove.net:Cmds PurGrp="0" PurNot="" Rank="12" SenderMinDep="1"
TimeCreated="1201119771184">
    <urn:groove.net:Cmd CMD="7" EngineURL="Dynamics" PurNot="" TestId="635E6781992C71B5"/>
  </urn:groove.net:Cmds>
</urn:groove.net:Del>
```

Delta B1:

```
<urn:groove.net:Del DepSeq="E9641419D18C367218970007" Gp="4" Seq="6401C37EFB36712340A30003"
Version="1,0,0,0">
  <urn:groove.net:Cmds PurGrp="0" PurNot="" Rank="12" SenderMinDep="2"
SpStSet="11;1;0;E9641419D18C367218970007;E9641419D18C0000;" TimeCreated="1201119771294">
    <urn:groove.net:Cmd CMD="7" EngineURL="Dynamics" PurNot="" TestId="D83DCCA04C620A1F"/>
  </urn:groove.net:Cmds>
</urn:groove.net:Del>
```

Delta C1: This is a priority delta. AssimilationPriority is set to 1. This delta goes in the fourth block (three other priority deltas were created before A1). DLS is the delta log state, which contains information about the last delta from each endpoint, in this case B1, C0, and A2. Each sequence in the delta log state is prefixed by the group number, in this case 4, 3, and 3.

```
<urn:groove.net:Del AssimilationPriority="1" BlkNum="4"
DLS="000000046401C37EFB36712340A30003,00000003E2D20DF7D85D27460B3E0002,00000003E9641419D18C36
7218970008" DepSeq="E9641419D18C367218970008,6401C37EFB36712340A30003" Gp="4"
Seq="E2D20DF7D85D27460B3E0003" Version="1,0,0,0">
  <urn:groove.net:Cmds PurGrp="0" PurNot="" Rank="13" SenderMinDep="3"
SpStSet="12;1;0;E9641419D18C367218970008;E9641419D18C0000;12;2;0;6401C37EFB36712340A30003;640
1C37EFB360000;" TimeCreated="1201119771434">
    <urn:groove.net:Cmd CMD="7" EngineURL="Dynamics" PurNot="" TestId="823BAA1446F63381"/>
  </urn:groove.net:Cmds>
</urn:groove.net:Del>
```

Delta B2: This is ordered after C1 despite being in the same group and having a lower sequence because it is in the next block.

```
<urn:groove.net:Del Gp="4" Seq="6401C37EFB36712340A30004" Version="1,0,0,0">
  <urn:groove.net:Cmds PurGrp="0" PurNot="" Rank="13" SenderMinDep="2"
TimeCreated="1201119771559">
    <urn:groove.net:Cmd CMD="7" EngineURL="Dynamics" PurNot="" TestId="4EFE4ED0DFE7D3D0"/>
  </urn:groove.net:Cmds>
</urn:groove.net:Del>
```

Delta A3: This is another priority delta. This delta goes in the fifth block. The delta log state is B1, C1, A2, with groups 4, 4, 3.

```
<urn:groove.net:Del AssimilationPriority="1" BlkNum="5"
DLS="000000046401C37EFB36712340A30003,00000004E2D20DF7D85D27460B3E0003,00000003E9641419D18C36
7218970008" DepSeq="E2D20DF7D85D27460B3E0003" Gp="4" Seq="E9641419D18C367218970009"
Version="1,0,0,0">
  <urn:groove.net:Cmds PurGrp="0" PurNot="" Rank="14" SenderMinDep="1"
SpStSet="12;2;0;6401C37EFB36712340A30003;6401C37EFB360000;13;3;0;E2D20DF7D85D27460B3E0003;E2D
20DF7D85D0000;" TimeCreated="1201119771638">
    <urn:groove.net:Cmd CMD="7" EngineURL="Dynamics" PurNot="" TestId="1859740D3380D0E9"/>
  </urn:groove.net:Cmds>
</urn:groove.net:Del>
```

5 Security

5.1 Security Considerations for Implementers

5.1.1 Use of Semi-weak Algorithms

The current protocol uses SHA-1, as described in [RFC3174], when computing the message digest. While there are no known practical attacks against SHA-1 at this point, it is showing signs of weakness.

5.1.2 Use of Non-standard/Suspect Algorithms

The current protocol uses ESIGN, as described in [IEEE1363a], for public key signature. ESIGN is not standard and has not been scrutinized as much as some other public key signature algorithms.

5.1.3 Insufficient Encryption of Delta Messages

The current protocol does not encrypt attributes on the delta element itself. This allows a passive attacker to read all the attributes on the delta element.

5.2 Index of Security Parameters

Security Parameter	Section
Per-space master key	1.5
Per-space encryption key	1.5, 3.1.3
Per-space per-member signature private key	1.5, 3.1.4.3.3
Per-space per-member signature public key	1.5, 3.1.5.1.2
Encryption algorithm	1.3.2, 3.1.4.3.2, 3.1.5.1.3
Signature algorithm	1.3.2, 3.1.4.3.3, 3.1.5.1.2
Hash algorithm	3.1.4.3.3, 3.1.5.1.2
Initialization vector	3.1.4.3.2, 3.1.5.1.3
Message signature	3.1.4.3.3, 3.1.5.1.2

6 Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs:

- Microsoft Office 2010 suites
- Microsoft Office Groove 2007
- Microsoft Office Groove Server 2007
- Microsoft Groove Server 2010
- Microsoft SharePoint Workspace 2010

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms SHOULD or SHOULD NOT implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that the product does not follow the prescription.

<1> Section 2.2.1.4.2: Office Groove 2007 and SharePoint Workspace 2010 sometimes set the group number to one higher than the highest group number when not required to limit the number of deltas within a group.

<2> Section 2.2.1.4.3: Office Groove 2007 and SharePoint Workspace 2010 sometimes set a number smaller than the smallest group number of all of the dependencies to prevent the purging of deltas which are required to enable the user to undo updates.

<3> Section 2.2.1.4.3: Office Groove 2007 and SharePoint Workspace 2010 sometimes set this when there is no new information.

<4> Section 2.2.1.4.3: Office Groove 2007 and SharePoint Workspace 2010 sometimes set this when there is no new information.

<5> Section 2.2.1.4.3: Office Groove 2007 and SharePoint Workspace 2010 sometimes set this to zero if the purge group has already been purged.

<6> Section 2.2.1.4.3: Office Groove 2007 and SharePoint Workspace 2010 set TimeCreated to a representation of the time the delta was created. This is not required by the protocol, and is used to simplify debugging.

7 Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.

8 Index

P

R

S

T

V