Mobile Spam Reporting Technical Specification

Candidate Version 1.0 – 23 Nov 2010

Open Mobile Alliance

OMA-TS-SpamRep-V1_0-20101123-C

Contents

Figures

Tables

1. Scope

The scope of this document is to define the technical details of the SpamRep Enabler, including:

- SpamRep message format and characteristics
- SpamRep Client-Server protocol
- Client procedures
- Server procedures relevant to the SpamRep Client-Server interface

This document does not address technical areas that are outside the scope of the SpamRep 1.0 Enabler, such as server-to-server message forwarding, processing the SpamRep message, or server interaction with any external spam-mitigation infrastructure.

2. References

2.1 Normative References

[3GPP2-X.P0027-002] 3GPP2 X.P0027-002 "Presence Security", URL: http://3gpp2.org/Public_html/specs/index.cfm

Note: Work in progress, awaiting IETF drafts

[3GPP-TR_33.978] 3GPP TR 33.978 "Security aspects of early IP Multimedia Subsystem (Release 6)", URL: http://www.3gpp.org/ftp/Specs/archive/33_series/33.978/

[3GPP-TS 23.003] 3GPP TS 23.003 V9.2.0 (2010-03), "Numbering, addressing and identification"

URL: http://www.3gpp.org/ftp/Specs/archive/23_series/23.003/23003-920.zip

[3GPP-TS_23.040] 3GPP TS 123.040 version 9.2.0"Technical Specification Group Core Network and Terminals;Technical realization of the Short Message Service (SMS) (Release 9)",

URL: http://www.3gpp.org/ftp/Specs/archive/23_series/23.040/

[3GPP-TS_33.141] 3GPP TS 33.141 "Presence service; Security", URL: http://www.3gpp.org/ftp/Specs/archive/33_series/33.141/

[3GPP-TS_33.222] 3GPP TS 33.222 "Generic Authentication Architecture (GAA); Access to network application functions using Hypertext Transfer Protocol over Transport Layer Security (HTTPS)", URL: http://www.3gpp.org/ftp/Specs/archive/33_series/33.222/

[3GPP-TS_33.937] 3GPP TS 33.937, "Study of mechanisms for Protection against Unsolicited Communication for IMS (PUCI)", Release 9, June 2010, URL: http://www.3gpp.org/ftp/specs/archive/33_series/33.937/33937-920.zip

[E.164] ITU-T Recommendation E.164 (02/2005), "The international public telecommunication numbering plan"

URL: http://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-E.164-200502-I!!PDF-E&type=items

[E.212A] ITU; "LIST OF MOBILE COUNTRY OR GEOGRAPHICAL AREA CODES (POSITION ON 1 DECEMBER 2007)" or later version

URL: http://www.itu.int/dms_pub/itu-t/opb/sp/T-SP-E.212A-2007-PDF-E.pdf

[FIPS.180-2.2002] US Department of Commerce FIPS PUB 180-2, "Secure Hash Standard", August 2002.

[IANAADFAM] IANA; "Address Family Numbers," February 5, 2010

URL: http://www.iana.org/assignments/address-family-numbers

[OMA_MMS] "Multimedia Messaging Service Encapsulation Protocol", Version 1.3, Open Mobile Alliance™, OMA-TS-MMS-ENC-V1_3-20080128-C, URL: http://www.openmobilealliance.org/

[RFC1320] IETF RFC 1320 "The MD4 Message-Digest Algorithm", R. Rivest, April 1992,

URL: http://www.ietf.org/rfc/rfc1320.txt

[RFC1321] IETF RFC 1321 "The MD5 Message-Digest Algorithm", R. Rivest, April 1992,

URL: http://www.ietf.org/rfc/rfc1321.txt

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

[RFC2046] IETF RFC 2046 "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types," N. Freed et. al, November 1996., URL: http://www.ietf.org/rfc/rfc2046.txt

[RFC2047] IETF RFC2047 "MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions For Non-ASCII Text"; Moore, K.; November 1996 URL: http://www.ietf.org/rfc/rfc2047.txt

[RFC2119] IETF RFC2119 "Key words for use in RFCs to Indicate Requirement Levels", S. Bradner, March 1997, URL:http://www.ietf.org/rfc/rfc2119.txt

[RFC2616] IETF RFC 2616 "Hypertext Transfer Protocol – HTTP/1.1", R. Fielding, June 1999, URL: http://www.ietf.org/rfc/rfc2616.txt

[RFC2617] IETF RFC 2617 "HTTP Authentication: Basic and Digest Access Authentication", Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A. and L. Stewart, June 1999, URL: http://www.ietf.org/rfc/rfc2617.txt

[RFC2818] IETF RFC 2818 "HTTP Over TLS", Rescorla, E., May 2000, URL: http://www.ietf.org/rfc/rfc2818.txt

[RFC3339] IETF RFC3339 Date and Time on the Internet: Timestamps, RFC 3339

URL: http://tools.ietf.org/html/rfc3339

[RFC3462] IETF RFC 3462 "The Multipart/Report Content for the Reporting of Mail System Administrative Messages," G. Vaudreuil, January 2003.,

URL: http://www.ietf.org/rfc/rfc3462.txt

[RFC4234] IETF RFC4234 "Augmented BNF for Syntax Specifications: ABNF". D. Crocker, Ed., P. Overell. October 2005, URL:http://www.ietf.org/rfc/rfc4234.txt

[RFC4648] IETF RFC4648 "The Base16, Base32, and Base64 Data Encodings," RFC 4648

URL: http://tools.ietf.org/html/rfc4648

[RFC4871] IETF RFC4871 "DomainKeys Identified Mail (DKIM) Signatures", E. Allman, J. Callas, M. Delany, M. Libbey, J. Fenton; May 2007
URL http://www.ietf.org/rfc/rfc4871.txt

[RFC4880] IETF RFC4880 "OpenPGP Message Format", J. Callas, L. Donnerhacke, H. Finney, D. Shaw, R. Thayer; November 2007
URL http://www.ietf.org/rfc/rfc4880.txt

[RFC5246] IETF RFC 5246 "The Transport Layer Security (TLS) Protocol", Dierks, T., Rescorla, E., August 2008, URL: http://www.ietf.org/rfc/rfc5246.txt

[RFC5322] IETF RFC5322 "Internet Message Format"; Resnick, P. (editor); October, 2008.
URL: http://www.ietf.org/rfc/rfc5322.txt

[RFC5598] IETF RFC5598 "Internet Mail Architecture", D. Crocker; July 2009
URL http://www.ietf.org/rfc/rfc5598.txt

[RFC5751] IETF RFC5751 "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification", B. Ramsdell, S. Turner; January 2010
URL http://www.ietf.org/rfc/rfc5751.txt

[SCRRULES] "SCR Rules and Procedures", Open Mobile Alliance™, OMA-ORG-SCR_Rules_and_Procedures, URL:http://www.openmobilealliance.org/

[SMPP] The SMPP Forum, "Short Message Peer to Peer

Protocol Specification v3.4," Document Version:- 12-Oct-1999 Issue 1.2

[SpamRep_AD] "Mobile Spam Reporting Architecture", Version 1.0, Open Mobile Alliance™, OMA-AD-SpamRep-V1_0,
URL: http://www.openmobilealliance.org/

[SpamRep-RD] "SpamRep Requirements", Open Mobile Alliance™, OMA-RD-SpamRep-V1_0,

URL: http://www.openmobilealliance.org/

[XSD_spam_rep] "XML Schema Definition: "SpamRep Document", Version 1.0, Open Mobile Alliance™, OMA-SUP-XSD_spam_rep-V1_0,
URL: http://www.openmobilealliance.org/

2.2 Informative References

[DMBOOT] "OMA Device Management Bootstrap, Version 1.2". Open Mobile Alliance™. OMA-TS-DM_Bootstrap-V1_2. URL:http://www.openmobilealliance.org

[DMPRO] "OMA Device Management Protocol, Version 1.2". Open Mobile Alliance™ OMA-TS-DM_Protocol-V1_2. URL: http://www.openmobilealliance.org

[OMADICT] "Dictionary for OMA Specifications", Version x.y, Open Mobile Alliance™, OMA-ORG-Dictionary-Vx_y, URL:http://www.openmobilealliance.org/

[RFC3986] IETF RFC3986 "Uniform Resource Identifier: Generic Syntax", T. Berners-Lee, R. Fielding, L. Masinter; Januaryu 2005
URL http://www.ietf.org/rfc/rfc3986.txt

[RFC4288] IETF RFC4288 "Media Type Specifications and Registration Procedures"; Freed, N., Klensin, J.; December 2005
URL: http://www.ietf.org/rfc/rfc4288.txt

3. Terminology and Conventions

3.1 Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

All sections and appendixes, except “Scope” and “Introduction”, are normative, unless they are explicitly indicated to be informative.

3.2 Definitions

By-Fingerprint	See [SpamRep-RD]
By-Reference	See [SpamRep-RD]
By-Value	See [SpamRep-RD]
Content	See [SpamRep-RD]
IMEI	**I**nternational **M**obile **E**quipment **I**dentity is a globally unique number identifying a mobile device in GSM, WCDMA, and iDEN networks
Local Spam Box	See [SpamRep-RD]
MEID	**M**obile **E**quipment **I**dentifier is a globally unique number identifying a mobile device in CDMA networks
Network Spam Box	See [SpamRep-RD]
Reporter	See [SpamRep-RD]
Spam	See [SpamRep-RD]
Spam Report	See [SpamRep-RD]
SpamRep Client	See [SpamRep-RD]
SpamRep Document	An XML document conforming to the XML schema described in [XSD_spam_rep]. It is used inside SpamRep Statements in messages exchanged between SpamRep Clients and Servers.
SpamRep Statement	A multipart/report MIME part representing a Spam Report, Action Request, Status Query, Quarantined Messages Query, Report Status, Action Response, or Quarantined Messages List. SpamRep Statements contain human and machine readable text, and optionally a copy of Spam Content.
SpamRep Message	A message exchanged between SpamRep Client and Server. It consists of a SpamRep Document and optionally a message deemed as Spam.
Simple SpamRep Message	The more basic of two SpamRep Message formats containing a single SpamRep Statement
Complex SpamRep Message	A SpamRep Message format typically used to package multiple SpamRep Statements into a single message.
SpamRep Server	See [SpamRep-RD]

3.3 Abbreviations

OMA	Open Mobile Alliance

4. Introduction

The SpamRep Enabler provides a capability wherein messages or content that is received can be designated, either by the User or autonomously by the SpamRep Client, as Spam and a report containing information about the Spam is sent to a Server for further processing. The report typically contains information about the origin(s) and content of the Spam message, and this information may be used to blacklist the originator or block future transport of the same or similar messages. The SpamRep 1.0 Enabler only specifies the Client-Server interface and message format; any processing of the received Spam Report by the Server is out of scope.

In addition to the basic capability described above, the SpamRep Enabler provides a mechanism for the SpamRep Client (e.g., the device User)to request the blocking or unblocking of specific Spam message originators, as well as an ability to indicate the type of abusive message received (e.g., malware, phishing attack, etc.).

4.1 Version 1.0

The SpamRep 1.0 Technical Specification addresses the requirements targeted for this release. Some features may require updates in later releases, as some requirements have been deferred to a future release.

Please refer to [SpamRep-RD] for those features that are addressed in this release. The architecture, functional components and the high level call flows are described in [SpamRep_AD].

5. Messages and Message Formats

The SpamRep Enabler defines the SpamRep Message format and the request-response protocol used by a SpamRep Client and a SpamRep Server for exchanging messages related to messaging abuse.

Within each SpamRep Message, information is conveyed by one or more SpamRep Statements. A SpamRep Statement may, for example, contain a User's report of Spam received on a mobile device, action requests, a SpamRep Server's response, or another type of communication. There are two SpamRep Message formats:

- Simple SpamRep Message (containing one SpamRep Statement) and
- Complex SpamRep Message (containing one or more SpamRep Statements).

The Simple SpamRep Message format is identical to the SpamRep Statement format; no encapsulation is needed. The Complex SpamRep Message format, discussed later in this section, uses two extra Multipart Internet Mail Extensions (MIME, [RFC2045]) layers of encapsulation.

If it contains only one SpamRep Statement, a SpamRep Message SHOULD be formatted as a Simple SpamRep Message, as illustrated in Figure 1. Otherwise, or if it contains multiple SpamRep Statements, a SpamRep Message MUST be formatted as a Complex SpamRep Message as discussed later in this section and as illustrated in Figure 2 below.

Each SpamRep Statement SHALL be formatted as [RFC3462] multipart/report MIME content of report-type=vnd.oma.spamrep+xml. A SpamRep Statement SHALL consist of two mandatory and an optional third parts:

- First part: Human-readable message text providing some information on the conditions that caused the SpamRep Document to be generated per [RFC3462].
- Second part: A SpamRep Document, which is the machine-readable XML payload information, formatted as "application/vnd.oma.spamrep+xml" MIME content.
- Optional third part: Spam Content, the possibly-abridged and/or anonymized message being reported as Spam, formatted as a MIME object, with the corresponding content type consistent with [RFC2045].
- Where a third MIME part is included, a Content-ID field MUST be added in accordance with [RFC2045].

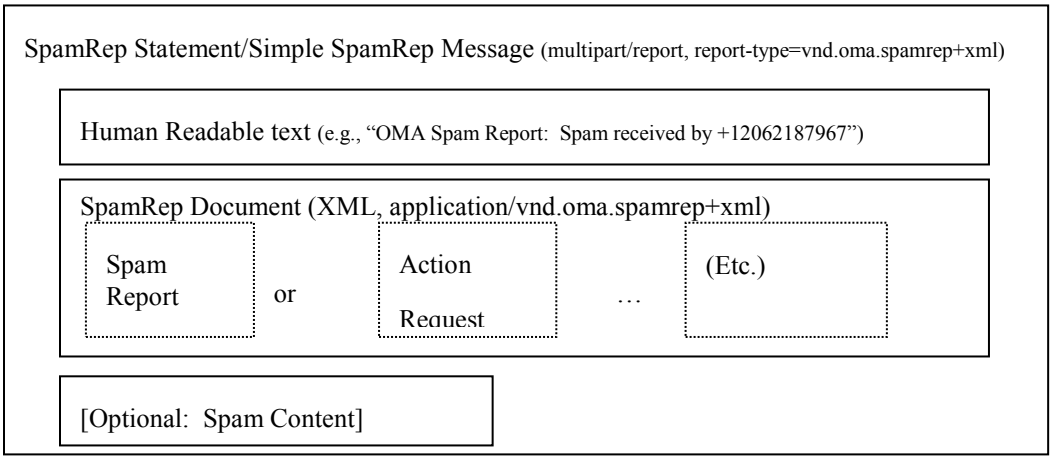


Figure 1: SpamRep Statement and Simple SpamRep Message Structure

A SpamRep Document SHALL be an XML document containing a SpamRep Message Element. There are several types of SpamRep Message Elements; the following types of complex XML elements SHALL be SpamRep Message Elements:

- Spam Report
- Action Request
- Status Query
- Quarantined Messages Query
- Report Status
- Action Response
- Quarantined Messages List

The XML schema for the SpamRep Document and SpamRep Message Elements is described in section 5.3.

As mentioned above, to optimize communications, the SpamRep Enabler allows SpamRep Messages to contain one or more SpamRep Statements using the Complex SpamRep Message format. Formally, a SpamRep Message SHALL either be a Simple SpamRep Message with MIME parts as in Figure 1 above, or SHALL be a collection of one or more SpamRep Statements contained within an outermost [RFC3462] multipart/report MIME part of report-type=mixed and inner multipart/mixed part as in Figure 2 below.

A Complex SpamRep Message SHALL have the following two parts:

- First part: Human-readable message text providing some information on the conditions that caused the message to be generated per [RFC3462] (e.g., "This is a collection of OMA spam reports")
- Second part: [RFC2046] message/vnd.oma.spamrep.multipart.mixed MIME part containing one or more SpamRep Statements
    - 1st SpamRep Statement
    - …
    - Nth SpamRep Statement

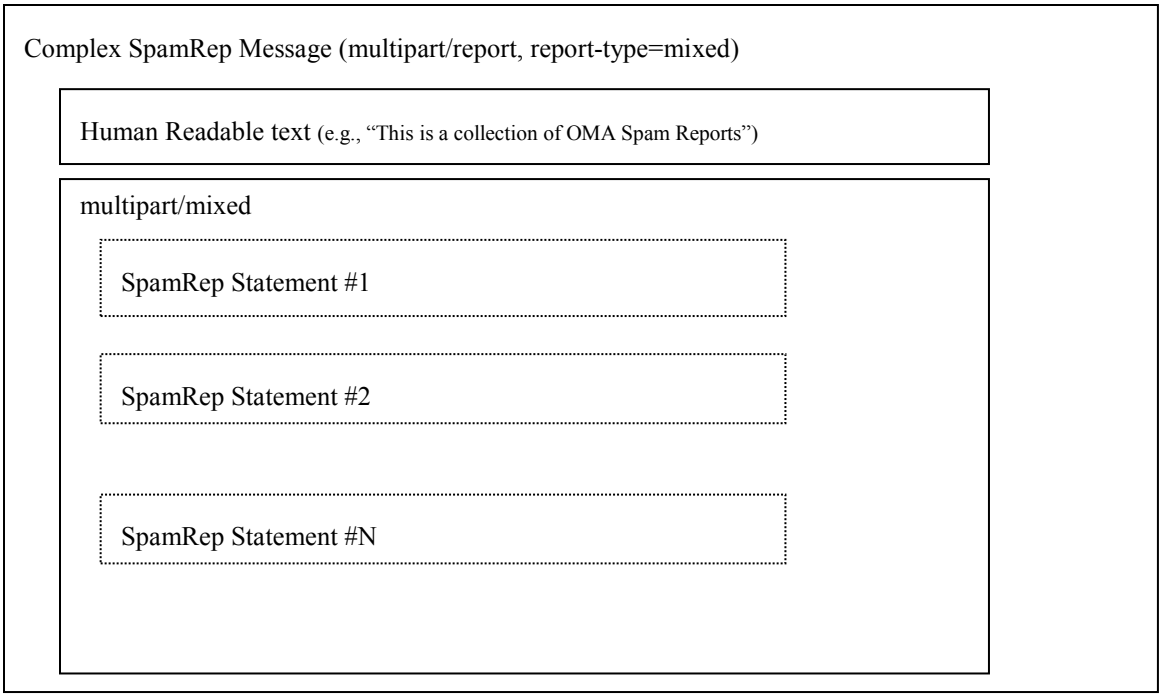


Figure 2: Complex SpamRep Message Structure

5.1 SpamRep Client Originated Messages

The following SpamRep Message Elements are legal in and only in messages sent from SpamRep Client to SpamRep Server:

1. Spam report;
2. Action request;
3. Status query, and
4. Quarantined messages query.

5.1.1 Spam Report Message Element

A Spam Report message containing one or more Spam Report Message Elements is sent by the SpamRep Client, either automatically or manually initiated by the User, to report a message deemed to be Spam. The structure and format of the Spam Report Message Element is defined by the XML schema as described in section 5.3. The Spam Report Message Element is implemented as the <spam-report> element of the <spam-rep-document> root element with the following clarifications:

The following table specifies parameters in a Spam Report Message Element:

Parameter name	Data type	Parameter cardinality	Description
SpamRepMessageID	Integer	1	Contains the unique id for this SpamRep Message Element
SpamRepClientID	String	1	Identifies the SpamRep Client, assigned and provisioned by the MNO.
ReportType	Enumerated	1..3	Indicates the type of spam reporting: By-Value, By-Reference, By-Fingerprint. Multiple ReportType parameters indicate a report containing a combination of reporting methods.
MessageType	String	1	Indicates the message type, e.g.: Email, SMS, MMS, IM, etc
ValueType	String	1	When ReportType is set to By-Value, indicate whether the full or partial Content is included in the Spam Report. Possible values for the "ValueType" attribute are: "full" and "partial"
MessageReference	String	1	When reporting By-Reference, this parameter is a reference to the Spam message.
HashingFunction	String	1	Indicate the hashing function applied to the reference when ReportType is set to By-Reference.
MessageFingerprint	Data Structure	0..n	When reporting By-Fingerprint, each MessageFingerprint parameter contains a fingerprint of the message.
ReportedMessageProtocol	String	0..1	The protocol used to transport the Spam message to the reporting node. RFC821, SMPP, GSM_MAP, GSM SMS, etc
MessageAttributes	Data Structure	0..1	Contains additional information about the reported abuse message, typically included if there is no or inadequate Content, such as when ReportType is By-Reference or By-Fingerprint. Each MessageType has its own set of attributes. The attributes are generally transmitted in, and extracted from message headers.
SubmissionTime	String	0..1	Contains the date and time of a Spam Report's original submission, formatted as per [RFC3339] Section 5.6, "Internet Date/Time Format".

OriginatingAddress	String	0..1	See Table 16 (Description of Frequently Used Parameters)
ForwardStatus	Boolean	0..1	See Table 16 (Description of Frequently Used Parameters)
AbuseType	Integer	0..1	Indicates the type of abuse: 0 : Spam, 1 : Phishing, 2 : Malware (e.g., Virus/Spyware), 3 : Not Spam, 4 : Miscategorized, 5 : Unauthorized Message (violation of a security policy), 6 : Sender Authentication Failure, 7: Invalid Message Format, 8: Other 9-255: reserved for future use.
SharePermission	Data Structure	0..n	Indicates Reporter's permission to share Spam Reports with third parties which reside outside of SpamRep Server's network. See Section 5.2.4
Version	String	1	See Table 16 (Description of Frequently Used Parameters)
Detection Information	Structure	0..n	Contains information on the algorithm or method used to trigger the spam report.

Table 1: Parameters in a Spam Report Message Element

The SpamRep Client SHALL generate the SpamRepMessageID parameter and SHALL ensure its uniqueness locally. This parameter SHALL be used to correlate this Message Element with a Message Element contained in the subsequent SpamRep Server response.

SpamRepClientID parameter SHALL contain the IMEI of the reporting device when the SpamRep Client is deployed on a mobile device in a GSM network, or the MEID of the reporting device when the SpamRep Client is deployed on a mobile device in a CDMA network, or the value of a provisioned identifier otherwise.

The value of the ReportType parameter SHALL be determined by Mobile Network Operator's policies. For example, based on MessageType, SMS messages are reported By-Value, MMS messages are reported By-Fingerprint and Email messages are reported By-Reference, or based on the message size, messages smaller then certain size are reported By-Value and all others are reported By-Reference.

When ReportType is set to By-Value, attribute "ValueType" SHALL be provided to indicate whether the full or partial Content is included in the Spam Report. Possible values for the "ValueType" attribute are "full" and "partial".

When ReportType is set to By-Reference, attribute "HashingFunction" SHALL be provided to indicate the hashing function applied to the reference. Possible values for the "HashingFunction" attribute are specified in section 5.1.1.2.

When ReportType is set to By-Fingerprint, attribute "FingerprintAlgID" of the MessageFingerprint data structure SHALL be provided to indicate the fingerprinting function applied to the Content. Possible values for the "FingerprintAlgID" attribute are specified in section 5.1.1.3.

The value of MessageType parameter SHALL indicate the type of the messaging system which delivered the message being reported as Spam. The Message Type parameter SHALL have one of the following values: EMAIL, SMS, MMS, IM or OTHER.

When the Spam Report is By-Reference, MessageReference SHALL be included, and SHOULD contain a message reference in a format which can be processed by the SpamRep Server (e.g., a mutually-agreed format per service provider policy).

When Spam Report is By-Fingerprint, one or more MessageFingerprint attributes SHALL be included. At least one MessageFingerprint attribute SHOULD be in a format and based on an algorithm which can be processed by the SpamRep Server (e.g., a mutually-agreed format per service provider policy).

Note that it is permissible for a single Spam Report to report both By-Reference and By-Fingerprint.

MessageAttributes SHALL contain the additional information about the message being reported as Spam as specified in section 5.1.1.1.

The SpamRep Client SHOULD report the date and time of a Spam Report's original submission.

The SpamRep Client SHALL set the ForwardStatus parameter to "1" if it is forwarding a Spam Report. By default, if the parameter is not present, this is not a forwarded report.

The SpamRep Client SHOULD set the AbuseType parameter if available.

The value of AbuseType parameter SHALL indicate the type of abuse or lack of abuse in the case when the value is set to "Not Spam". The AbuseType parameter SHALL have one of the following values: Spam, Phishing, Malware, Not Spam, Miscategorized, Unauthorized Message, Sender Authentication Failure, Invalid Message Format, Other or Unspecified. Spam indicates an unsolicited, usually commercial, message. Phishing indicates a message trying to fraudulently obtain confidential information. Malware indicates a message containing a virus or adware. Not Spam indicates that a message previously identified as Spam is not Spam and that it should be released from the Network or Local Spam Box. Miscategorized indicates that an incorrect message content classification has been applied. Unauthorized indicates that the message was sent without permission. Sender Authentication Failure indicates that the identity of the sender failed authentication including signalling discrepancies such as seen in SMS spoofing or faking. Invalid Message Format – indicates a message that is encoded incorrectly. Other indicates other types of abuse not covered by predefined types. Unspecified indicates that no specific type of abuse is specified.

SharePermissionList is provisioned initially by the Mobile Network Operator (see section 5.2.4). The SharePermission parameter SHALL be included in a Spam Report, when the CurrentPermission element of the SharePermissionList is changed in the SpamRep Client (.e.g. by external entity such as Reporter or any other system). If there is no change, the SharePermission parameter SHOULD be empty, indicating to the SpamRep Server to use previously stored version of SharePermission. The following table defines the SharePermission structure.

Parameter name	Data type	Parameter cardinality	Description
Permission	string	1	Indicates the amount of personal information to be shared or the ban on sharing the Spam Report with third party. The value is one of these: "Entire message" "Email / phone number" "Anonymous" "Deny" Email / phone number: Share only the email address or phone number Anonymous: Do not share any personal information Deny: Revocation of sharing of Spam Report
ThirdPartyID	string	1	The ID of third party to share the Spam Report.

Table 2: The SharePermission Structure

The Version parameter SHALL indicate the version of specification that the SpamRep Client is using to generate the report. The version number in this specification is set to "1.0".

The following table defines the parameters in the DetectionInformation structure:

Parameter name	Data type	Parameter cardinality	Description
DetectionMethod	String	1	Detection method that caused this message to be classified as spam (flooding, faking, content etc.)
PolicyName	String	0..1	If this message was classified as spam due to operator or subscriber setting, the name of the individual filter involved
AbuseScore	String	0..1	As per [3GPP-TS_33.937] (Prevention of Unsolicited Communications for IMS) recommendations

Table 3 Parameters in Detection Information Structure

5.1.1.1 Message Attributes

When preparing a Spam Report and the MessageType Parameter is set to “Email”, the SpamRep Client SHALL include the following data in the MessageAttributes parameter:

Parameter name	Data type	Parameter cardinality	Description
MessageHeaderField	String	0..n	The contents of a header field as defined in [RFC5322]. These MUST be included in the order in which they appeared in the message about which a SpamRep report is being generated and with no changes such as space compression or comment removal. If the header field is wrapped (i.e. contains newlines), it MUST be encoded using one of the mechanisms defined in [RFC2047]. All header fields on the message SHOULD be included; the From, To, Cc, Subject and Date fields MUST be included if present except if specifically disallowed by privacy or anonymization policies.
HeaderFrom	String	0..1	Value of the From party (subscriber identifier) referenced in upper HTTP layer

Table 4: Email message attributes

When preparing an SMS Spam Report the SpamRep Client may include SMS parameters from the following table with the indicated cardinality in the MessageAttributes parameter. All parameter values SHALL be represented as printable character strings.

In many client implementations, it is expected that only a small subset of these parameters will be utilized. For subscriber-generated abuse reports from device-based clients of mobile-terminated spam, there is a recommended subset of parameters which should be included in Spam Reports if available and consistent with policy. These parameters are designated ‘(R)’; they in the common case of subscriber-generated abuse reports representing SMS messages received by a mobile device. It is also recommended that the entire User Data (TP-UD) field be included as Content in SpamRep Messages.

Parameter Name	Data Type	Parameter cardinality (R) = Recommended for mobile reportin	Description
DCS	Integer	0..1 (R)	Data Coding Scheme (TP-DCS) per [3GPP-TS 23.040]
OriginationAddress	String	0..1 (R)	Spammer's purported address. [3GPP-TS23.040] "International number" and "ISDN/telephone numbering plan" formats shall be preferred, and assumed if the TON and NPI are omitted. Syntax is: <TP-Origination-Address per [3GPP-TS23.040] in decimal string format> ['‚' <TON in integer format {0-7} per [3GPP-TS 23.040]> '‚' <NPI in integer format per [3GPP-TS23.040>]
Destination Address	String	0..1 (R)	Address that the abusive message was addressed to. [3GPP-TS23.040] "International number" and "ISDN/telephone numbering plan" formats shall be preferred and assumed if the TON and NPI are omitted. Syntax is: <TP-Destination-Address per [3GPP-TS23.040] in decimal string format> ['‚' <TON in integer format {0-7} per [3GPP-TS 23.040]> '‚' <NPI in integer format per [3GPP-TS23.040>]
SCA	String	0..1 (R)	Address of SMS Service Centre in delivery path. Contained in SMS but not necessarily stored in device. Syntax is: Up to 15 characters as defined in [E.164]
ServiceCenterTimestamp	String	0..1 (R)	In 3GPP services this is the timestamp added by SMSC. Syntax is: [3GPP-TS 23.040] Service Center Timestamp in [RFC 3339] "Internet Date/Time Format" In 3GPP services this is the timestamp added by SMSC
DeviceTimestamp	String	0..1 (R)	'Received at device' timestamp in [RFC 3339] "Internet Date/Time Format."

PID	Integer	0..1 (R)	[3GPP-TS 23.040] TP-PID (Protocol ID) in decimal string format
UDL	Integer	0..1 (R)	User data length per [3GPP-TS 23.040] UDL. Note that this may represent a number of octets or septets, depending on the DCS value.

UDIndicator	String	0..1	Indication as to whether the Content of the SpamRep Message (the UD) is a verbatim copy of the spam, or a decoded (e.g., rendered into a different character set) version, or is removed entirely.
			Legal values are:
			`RAW,`
			`DECODED,`
			`REMOVED`

UDHI	String	0..1	Indication that UDH is present in body of an SMS message.
			Legal values are(if possible) the [3GPP-TS 23.040] TP-UDHI:
			`Present,`
			`Absent,`
			`Unknown`
			If this attribute is not specified, the assumed value is Unknown.

UDHAttached	Enumerated	0..1	Indicates whether the Content of the SpamRep Message (the UD) includes UDH, or only the body of the UD. [Note: It is desirable to include UDH in the Content; however this is not always possible.]
			Legal values are:
			`True,`
			`False,`
			`Unknown`
			If this attribute is not specified, the assumed value is True.

UDH	String	0..1	Attribute may be used only if a Universal Data Header (UDH) is present in the spam message's body and the UDL attribute is present. This attribute includes the [3GPP-TS 23.040] UDH, beginning with UDHL and excluding any trailing

			Fill bits, in [RFC 4648] Base64 format. Note: This information should also be incorporated in any raw UD Content included in the SpamRep Message.
MTI	String	0..1	SMS Message Type Indicator (MTI) field. Value should be [3GPP-TS 23.040] TP-MTI if possible, otherwise 'UNKNOWN' or 'OTHER' Legal values are: SMS-DELIVER, SMS-DELIVER-REPORT, SMS-SUBMIT, SMS-SUBMIT-REPORT, SMS-STATUS-REPORT, SMS-COMMAND, UNKNOWN, OTHER If this attribute is not specified, the assumed value is SMS-DELIVER, by the SpamRep server.
DeliveryNetwork	Data Structure	0..1	Indicates how SMS was or would be delivered (e.g., Circuit Switched or Packet Switched)
InterfaceType	Enumerated	0..1	Used to convey information about type of interface / nature of the transaction in the messaging node for which the spam event occurs The legal values are: 0 - Mobile Originated/Mobile Terminated 1 – Mobile Originated/Application Terminated 2 – Application Originated/Mobile Terminated 3 – Mobile Originated/Destination Unknown 4 – Origination Unknown/Mobile Terminated 5 – Application Originated/Application Terminated 6 – Inter-carrier Application Originated/Mobile Terminated 7 – Mobile Originated/Inter-carrier Application Terminated 8 – Mobile Originated/Inter-carrier Application Terminated
OriginationIMSI	String	0..1	Origination IMSI or MT-Correlation-ID. There are security risks in disclosing IMSI, and given MSISDN, IMSI can be

			retrieved by an authorized party. The syntax follows the definition of IMSI as per [3GPP-TS 23.040].
DestinationIMSI	String	0..1	Destination IMSI or MT-Correlation-ID. There are security risks in disclosing IMSI, and given MSISDN, IMSI can be retrieved by an authorized party. The syntax follows the definition of IMSI as per [3GPP-TS 23.040].
IMEI	String	0..1	ReceivingReceiving device IMEI per [3GPP-TS 23.003]. Effectively a serial # of the associated device.
CellID	String	0..1	Cell ID associated with the Spam Report. A mobile device likely will report the current rather than the spam's forward path Cell ID. SC Timestamp vs. Report Timestamp may provide confidence.
LAC	String	0..1	Cellular 'Location Area Code' associated with the Spam Report. This may be the current rather than the forward path LAC.
MCC	String	0..1	Mobile Country Code per ITU-T [E.212A] associated with the report.
MNC	String	0..1	Mobile Network Code per ITU-T [E.212A] associated with the report.
MessageFlood	String	0..1	Used to collectively describe a flood of messages. Format is: <Count of messages> [',' <receipt time period in seconds>]]
VPF	Integer	0..1	TP-Validity-Period-Format – a single-digit integer per [3GPP-TS 23.040]. Legal values are: {0-3}.
VP	String	0..1	Must not be present if VPF = '0'. Presence and legal values depend on VPF per [3GPP-TS 23.040]. Legal values: Integer (if VPF =1) , SpamRep Time Format (if VPF =2), Enhanced VP in [RFC 4648] Base64 format (if VPF = 3)
MR	Integer	0..1	TP-Message-Reference per [3GPP-TS 23.040]. Legal values are: {0-255}.
SR	Integer	0..1	1 if status report indicated by TP-SRI or requested by TP-SRR per [3GPP-TS 23.040]. Legal values are: {0,1}.

ConcatenatedMessage Segments	String	0..1	Indicates that the SpamRep represents a concatenation of two or more SMS segments per [3GPP-TS 23.040]. Legal values are: <integer - # of segments>, CONCATENATED, UNKNOWN Note that unless this parameter is set to '1' indicating a single segment, other parameters (e.g., delivery times) may represent the values from any one or more message segments. Default (if attribute absent) is 'UNKNOWN.'
MMS	Boolean	0..1	Indicates more messages to send per [3GPP-TS 23.040] TP-MMS.
RD	Boolean	0..1	Reject duplicate per [3GPP-TS 23.040] TP-RD.
SRQ	Integer	0..1	For Status messages only. Status report qualifier per [3GPP-TS 23.040] SRQ. Legal values are: `{0,1}`.
DT	String	0..1	For Status messages only. Discharge time per [3GPP-TS 23.040] DT in [RFC 3339] "Internet Date/Time Format"
ST (status)	Integer	0..1	For Status messages only. Status per [3GPP-TS 23.040] ST
MSC_E164	String	0..1	[E.164] format addressa of associated MSC.

ReportingNode	String	0..1	If populated, indicates the type of element that reported the spam. If not present, "Mobile_Device" assumed.
			Legal values:
			`SMS_Router,`
			`IP-SM-GW,`
			`SMS-SC,`
			`Mobile_Device,`
			`Subscriber,`
			`Email_Gateway,`
			`Unknown,`
			`Other`

ReportingNodeAddress	String	0..1	SCCP or IP address of node which reported the SMS. Address type indicated as per [IANAADFAM]
OriginationNodeAddress	String	0..1	SCCP or IP address of node which was the originator of the SMS. Address type indicated as per [IANAADFAM]
DestinationNodeAddress	String	0..1	SCCP or IP address of node which was the recipient of the SMS.. Address type indicated as per [IANAADFAM]
ESM_Class	Integer	0..1	Indicates presence or absence of User Data Header Indicator (UDHI) and Reply Path (RP) in spam per [SMPP 3.4]. Legal values: 0 (0x00) Normal SMS 64 (0x40) Normal SMS with UDHI set; i.e., short_message contains UDH 128 (0x80) Normal SMS with reply path set 192 (0xC0) Normal SMS with UDHI reply path set i.e. short_message contains UDH and reply path set

Table 5: SMS message attributes

Parameter name	Data type	Parameter cardinality	Description
Network	Enumerated	0..1	Indicates the type of network via which the SMS was or would be delivered. The legal values are: GSM, UMTS, LTE, 3GPP, CDMA, Other, Unknown
Sender	String	0..1	Indicates how SMS was or would be delivered (e.g., Circuit Switched or

			Packet Switched). The syntax is: [Circuit|Packet]

Table 6: Parameters of the DeliveryNetwork structure

When preparing a Spam Report and the MessageType Parameter is set to "MMS", the SpamRep Client SHALL include the following data in the MessageAttributes parameter:

Parameter name	Data type	Parameter cardinality	Description
MessageType	String	1	Value of the X-Mms-Message-Type parameter as specified in [OMA_MMS]
MessageID	Integer	1	Value of the Message-ID parameter as specified in [OMA_MMS]
TransactionID	String	1	Value of the X-Mms-Transaction-ID parameter as specified in [OMA_MMS]
To	String	0..1	Value of the To parameter as specified in [OMA_MMS]
From	String	0..1	Value of the From parameter as specified in [OMA_MMS]

Table 7: MMS message attributes

When preparing a Spam Report and the MessageType Parameter is set to "IM", the SpamRep Client SHALL include the following data in the MessageAttributes parameter:

Parameter name	Data type	Parameter cardinality	Description
ServiceType	String	1	Identifier of the IM service, e.g. ICQ, MSN, etc
To	String	0..1	Address of the recipient of the message (e.g. OMA IM UserID)
From	String	0..1	Purported address of the sender of the message

Table 8: IM message attributes

5.1.1.2 Message Reference

The SpamRep Enabler supports reporting spam messages By-Reference.

For the purposes of the SpamRep Enabler, the message reference SHALL consist of full or partial message metadata (e.g. message headers), and optionally partial message data, to which a hashing function has been applied.

The applied hashing function SHALL be specified in the "HashingFunction" parameter of the Spam Report Message Element. Example values for this attribute are:

1. "null" – indicates that no hashing function is applied to the message reference
2. "MD4" – indicates that MD4 hashing function as defined in [RFC1320] is applied to the message reference.

3. "MD5" – indicates that MD5 hashing function as defined in [RFC1321] is applied to the message reference.
4. "SHA-1" – indicates that SHA-1 hashing function (Secure Hash Algorithm) [FIPS.180-2.2002] is applied to the message reference.
5. "SHA-2" – indicates that SHA-2 hashing function [FIPS.180-2.2002] is applied to the message reference.
6. Other hashing functions can be used depending on the messaging server.

The default value for the "hashing-function" attribute is "MD5".

The SpamRep Server SHALL support both MD4 and MD5 hashing functions. The SpamRep Client SHOULD support both MD4 and MD5 hashing functions. The preferred hashing function SHALL be determined by operator's policies.

When preparing a Spam Report and the MessageType Parameter is set to "Email" and ReportType is set to "By-Reference", the SpamRep Client SHALL include the following data in the message reference:

1. All message headers in the order in which they were received, with no modifications whatsoever to spacing, line wrapping, line termination, removal of duplicates, or any other changes. The structure of the email message is defined in [RFC5322]

When preparing a Spam Report and the MessageType Parameter is set to "SMS" and ReportType is set to "By-Reference", the SpamRep Client SHALL include the following data in the message reference:

1. The entire SMS-DELIVER PDU as defined in [3GPP-TS_23.040] except for the last field (TP-UD).

When preparing a Spam Report and the MessageType Parameter is set to "MMS" and ReportType is set to "By-Reference", the SpamRep Client SHALL include the following data in the message reference:

1. The entire M-Notification.ind PDU as defined in [OMA_MMS].

When preparing a Spam Report and the MessageType Parameter is set to "IM" and ReportType is set to "By-Reference", the SpamRep Client SHALL include the following data in the message reference:

1. Entire message including both Content and metadata.

5.1.1.3 Message Fingerprint

The SpamRep Enabler supports reporting spam messages By-Fingerprint.

Cryptographic hash algorithms such as MD5, SHA-1, SHA-256 and also proprietary algorithms can be applied to part of or the entire spam message to generate a message "fingerprint". Some keywords can also be identified and extracted to form a message fingerprint. A recent standardized robust fingerprinting algorithm for image is the MPEG-7 Image Signature Tools [MPEG-7-IMG-SIG], which can be applied to an image in a message.

A given message may have more than one fingerprint. For example, for a URL fingerprinting algorithms, there may be one fingerprint for each URL contained in a Spam message. Therefore, the SpamRep Enabler supports the inclusion of multiple fingerprints. When reporting By-Fingerprint, for each reported fingerprint there SHALL be one "MessageFingerprint" structure in the Spam Report Message Element.

Each fingerprint is represented by a MessageFingerprint data structure, as specified in the following table:

Parameter name	Data type	Parameter cardinality	Description
FingerprintAlgID	String	1	ID of the fingerprint algorithm.
Fingerprint	Opaque Data	1	Opaque Message Fingerprint which is extracted by the SpamRep Client using the algorithm identified by the parameter FingerprintAlgID.
Range	String	0..1	Algorithm-dependent, identifies the part of the message or message characteristic to which the fingerprint algorithm has been applied.

Table 9: Parameters in the MessageFingerprint structure

A fingerprint algorithm identifier, FingerprintAlgID, identifies the particular fingerprint algorithm used to generate the message fingerprint. Within each MessageFingerprint structure, the applied fingerprint algorithm(s) SHALL be specified in the "FingerprintAlgID" parameter. Example values for this parameter are:

1. "MD5" – indicates that MD5 hashing function as defined in [RFC1321] is applied to the message.
2. "SHA-1" – indicates that SHA-1 hashing function is applied to the message.
3. "SHA-256" – indicates that SHA-562 hashing function is applied to the message.
4. "KEYWORD" - indicates that the fingerprints extracted are keywords from the message body.
5. "MPEG7-IMG-SIG" - indicates that the MPEG-7 Image Signature Tools algorithm is applied to the message.

Other fingerprint algorithms may be used.

Within each MessageFingerprint structure, the "Fingerprint" parameter SHALL be specified, and its value set to the result of the particular fingerprint algorithm specified in the "FingerprintAlgID", using the Range parameter value if appropriate.

Within each MessageFingerprint structure, the optional "Range" parameter MAY be used to identify the part of the message body or transport characteristic the fingerprint algorithm is applied to. The meaning of the range parameter is dependent on the particular fingerprint algorithm. For MD5, SHA-1 and SHA-256 algorithms, the absence of the "Range" parameter SHALL indicate that the algorithm has been applied to the entire message body, including headers if any.

5.1.2 Action Request

An Action Request message is initiated by the User and sent by the SpamRep Client, to request an action to be performed by the SpamRep Server. SpamRep Server SHALL and SpamRep Client SHOULD support the following actions:

1. Block Sender;
2. Unblock Sender.

SpamRep Server and SpamRep Client MAY additionaly support the following action:

1. Release Quarantined Message.

The structure and format of the Action Request Message Element is defined by the XML schema as described in section 5.3. The Action Request Message Element is implemented as the <action-request> element of the <spam-rep-document> root element with the following clarifications:

The following table specifies parameters in an Action Request Message Element:

Parameter name	Data type	Parameter cardinality	Description
ActionType	String	1	The type of the action to be performed. Possible values are: "BlockSender", "UnblockSender" and "ReleaseQuarantinedMessage".
Sender	String	0..n	The address of the sender to be blocked or unblocked. This element is mandatory for Block and Unblock Sender requests
Quarantined MessageID	String	0..n	The Id of the message to be released from quarantine. This Id is returned in the quarantined messages list as the response to quarantined messages query.

Table 10: Action Request Parameters

The value of the ActionType parameter SHALL indicate the type of the action being requested. The ActionType parameter SHALL have one of the following values: "BlockSender", "UnblockSender", or "ReleaseQuarantinedMessage".

The Sender parameter SHALL be mandatory when ActionType is either "BlockSender" or "UnblockSender", otherwise it SHALL be optional. The value of the Sender parameter SHALL contain the address of the sender to be blocked or

unblocked. The sender address MAY be one of the following: MSISDN or SIP URI in case of SMS or MMS, email address in case of email messages, or a specific sender URI applicable to the IM service.

The QuarantinedMessageID parameter SHALL contain the value of the identically named parameter in the Quarantined Messages List returned as the response to the Qurarntined Messages Query corresponding to the message which is requested to be released from quarantine.

5.1.3 Status Query

Status Query message is initiated by the User and sent by the SpamRep Client in order to obtain status of a previously submitted Spam Report from the SpamRep Server. The structure and format of the Status Query Message Element is defined by the XML schema as described in section 5.3. The entire Message Element is represented by the <status-query> element of the <spam-rep-document> root element with the following clarifications:

The following table specifies parameters in a Spam Report Status Query Message Element:

Parameter name	Data type	Parameter cardinality	Description
SpamReportID	String	1..n	List of SpamReportID's that identify the Spam Reports whose status is queried.

Table 11: Parameters in a Status Query message

5.1.4 Quarantined Messages Query

Quarantined Messages Query message is initiated by the User and sent by the SpamRep Client in order to obtain a list of quarantined messages from the SpamRep Server. The structure and format of the Quarantined Messages Query Message Element is defined by the XML schema as described in section 5.3. The Quarantined Messages Query Message Element is implemented as the <quarantined-messages-query> element of the <spam-rep-document> root element with following the clarifications:

There are no query-specific parameters.

5.2 SpamRep Server Originated Messages

The following SpamRep Message Elements are legal in and only in messages sent from SpamRep Server to SpamRep Client:

1. Report status;
2. Action response, and
3. Quarantined messages list.

5.2.1 Report Status

Report Status message is sent by the SpamRep Server as the initial response to the Spam Report message received from the SpamRep Client. This message is also sent in response to the Status Query message, and as an asynchronous status notification message. The structure and format of the Report Status Message Element is defined by the XML schema as described in section 5.3. The entire Message Element is represented by the <report-status> element of the <spam-rep-document> root element with the following clarifications:

The following table specifies parameters in a Report Status Message Element

Parameter name	Data type	Parameter cardinality	Description
SpamReportID	String	1	See Table 16 (Description of Frequently Used Parameters)
StatusCode	Integer	1	Status of Spam Report identified by SpamReportID. See Section 8
StatusText	String	0..1	Information pertaining to the status. See Section 8.
SpamRepMessageID	String	0..1	When Report Status is sent in response to a Spam Report this parameter SHALL specify the same value as the parameter with the same name in the Spam Report. When Report Status is sent in response to a Status Query this parameter SHALL NOT be present.
AbuseType	Integer	0..1	The SpamRep Server MAY return its own AbuseType classification of the Spam Report.

Table 12: Information elements in the Spam_Report_Status Message Element

The SpamReportID parameter SHALL contain the newly generated ID when Report Status is sent in response to the initial Spam Report, and the value of the SpamReportID parameter of the received Status Query Message Element when Report Status is sent in response to a Status Query.

The StatusCode parameter SHALL contain either one of the predefined values in Section 8.

Upon receipt of a "By Value Required" response, a SpamRep Client SHOULD re-generate its original report message, using a ReportType of "By-Value" and including the entire offending message in the report's Content. The SpamRep Client SHALL NOT generate any other reply.

StatusText parameter SHOULD contain optional textual information pertaining to the Spam Report status.

SpamRepMessageID parameter SHALL contain the value of the SpamRepMessageID parameter of the received Spam Report Message Element when Report Status is sent in response to the initial Spam Report. When Report Status is sent in response to a Status Query Message Element this parameter SHALL NOT be present.

If the AbuseType is not included in Spam Report (e.g. Unauthorized Message, Sender Authentication Failure), the SpamRep Server or external anti-spam structure MAY insert the proper value for the AbuseType. A SpamRep Server MAY include this attribute containing its conclusion on the correct classification of the Spam Report.

5.2.2 Action Response

Action Response message is sent by the SpamRep Server in response to the Action Request message received from the SpamRep Client. The structure and format of the Action Response Message Element is defined by the XML schema as described in section 5.3. The Action Response Message Element is implemented as the <action-response> element of the <spam-rep-document> root element with the following clarifications:

Parameter name	Data type	Parameter cardinality	Description
SpamRepServerID	String	1	Identifies the SpamRep Server. See Table 16
StatusCode	Integer	1	Provides information about the status of an Action Response. See Section 8.
StatusText	String	0..1	Information pertaining to the status. See Section 8.

Table 13: Parameters in an Action Response Message Element

5.2.3 Quarantined Messages List

Quarantined Messages List message is sent by the SpamRep Server in response to the Quarantined Messages Query message received from the SpamRep Client. The list of quarantined messages is obtained from an entity external to the SpamRep Server (e.g. message box located in the Messaging System). The structure and format of the Quarantined Messages List Message Element is defined by the XML schema as described in section 5.3. The entire Message Element is represented by the <quarantined-messages-list> element of the <spam-rep-document> root element with the following clarifications:

The following table specifies parameters in a Quarantined Messages List Message Element.

Parameter name	Data type	Parameter cardinality	Description
QuarantinedMessage	Data Structure	0..n	A quarantined message structure.
StatusCode	Integer	1	Provides information about the status of Query. See Section 8.
StatusText	String	0..1	Information pertaining to the status. See Section 8.

Table 14: Parameters in a Quarantined Messages List Message Element

Parameter name	Data type	Parameter cardinality	Description
QuarantinedMessageID	String	1	ID of a quarantined message, assigned by the SpamRep Server or an external entity such as a Messaging System.
QuarantinedMessageAddInfo	Opaque	0..1	Contains additional information about the quarantined message, typically extracted from the message header, to allow a user to judge if the quarantined message is a mis-blocked message.

Table 15: The QuarantinedMessage Structure

5.2.4 Share Permission List

A list of Share Permissions is provisioned to the SpamRep Client. The way it is provisioned is out of scope and dependent on implementation (e.g. downloaded from a web site, supplied via OMA DM, etc) and service provider's policy.

A Share Permission contains attributes for a third party, its name and current permission. Even if the SpamRep Client does not want to share the Spam Report, depending on the third party it can be limited (e.g. government agency, DenyAllowed is set to "False").

The following table specifies the attributes of the Share Permission element:

Element name	Data type	Parameter cardinality	Description
ThirdPartyID	String	1	The ID of third party to share the Spam Report.
ThirdPartyName	String	1	The name of third party. The SpamRep Client can display this name instead of the ID to the Reporter to get the confirmation of sharing the Spam Report.
CurrentPermission	String	1	Indicates the current permission for sharing with the related third party. If Share Permission element is not included in the Spam Report, the value of CurrentPermission is assumed by default. Allowed values are the same as for the SharePermission element of the Spam Report: "Entire message" "Email / phone number" "Anonymous"

			" Deny" If CurrentPermission is "Entire message", the SpamRep Client can insert all of elements. If CurrentPermission is "Email / phone number", the SpamRep Client can insert three of them except "Entire message". If CurrentPermission is "Anonymous", the SpamRep Client can insert "Anonymous" and "Deny".
DenyAllowed	Boolean	1	Indicates to allow the SpamRep Client to deny sharing with the related third party. If the value is "True", the SpamRep Client can revoke sharing the Spam Report with the third party as it inserts "Deny" in Permission parameter of Spam Report.

Table 16: SharePermissionList Elements

5.3 SpamRep XML Document

SpamRep XML document SHALL be used as the container for the SpamRep Message Element in both SpamRep Server and Client originated messages.

5.3.1 Mime Type

The MIME type for SpamRep XML document SHALL be "application/vnd.oma.spamrep+xml".

5.3.2 XML Schema

The SpamRep XML document SHALL conform to the XML schema described in [XSD_spam_rep].

5.3.3 Structure

The SpamRep XML document SHALL conform to the XML schema described in section 5.3.2 "XML Schema", with the following clarifications:

- The root element of the SpamRep XML document SHALL be <spam-rep-document>
- The root element of the SpamRep XML document SHALL contain one message.

5.4 Description of Frequently Used Parameters

The following table contains descriptions of frequently used parameters throughout the document:

Parameter name	Data type		Description
SpamRepClientID	String		Identifies the SpamRep Client, assigned and provisioned by the MNO.
SpamRepServerID	String		Identifies the SpamRep Server, assigned and provisioned by the MNO.
SpamReportID	String		Identifies the SpamRep Report, assigned by the SpamRep Server after receiving the report and is communicated back to the client in the response.
SpamRepMessageID	Integer		Contains a unique identifier that is given by the SpamRep Enabler (i.e. SpamRep Client, or SpamRep Server) to each SpamRep Message. The response to a given message SHALL contain the same

			SpamRepMessageID.
AbuseType	String		Indicates the type of abuse: Spam, Phishing, Malware (e.g., Virus/Spyware), Not Spam, Miscategorized, Unauthorized Message (violation of a security policy), Sender Authentication Failure, Invalid Message Format, Other, Unspecified.
SubmissionTime	String		The date and time of a Spam Report's submission, per the format defined by [RFC3339] Section 5.6, "Internet Date/Time Format".
OriginatingAddress	String		Identifies the actual or purported originating address of the abusive message. In the case of a report of an email message, this is the first address taken from the content of the From: header field.
ForwardStatus	Boolean		Specifies whether or not the report is a forwarded report.
SharePermission	Data Structure		Indicates Reporter's permission to share Spam Reports with third parties that reside outside of SpamRep Server's network.
Version	String		Indicates the SpamRep version.

Table 17: Frequently used parameters

6. Procedures

6.1 Common Procedures

A SpamRep Client and SpamRep Server interact through a single interface exposed by the SpamRep Server, using SpamRep Messages defined in Section 5. For information about SpamRep call flows for each of the functions described below, refer to Appendix B. Flows (informative) of [SpamRep_AD].

6.2 Client Procedures

The procedures described in the following subsections are performed by the SpamRep Client.

6.2.1 Spam Report Submission

For information about the spam reporting call flow, refer to the informative Appendix B.1 of [SpamRep_AD].

When reporting messaging abuse, the SpamRep Client:

1. SHALL assemble a SpamRep Message (i.e., either a Simple SpamRep Message or Complex SpamRep Message) containing one or more Spam Report Message Elements as described in section 5, and
2. SHALL send the SpamRep Message to the SpamRep Server as described in section 7.

6.2.2 Requesting an Action

For information about the call flow for requesting an action, refer to the informative Appendix B.4 of [SpamRep_AD].

When requesting an action, the SpamRep Client:

1. SHALL assemble an Action Request message as described in section 5.1.2;
2. SHALL assemble a SpamRep Document containing the assembled Action Request Message Elemente as described in section 5.3; and,
3. SHALL send the SpamRep Message to the SpamRep Server as described in section 7. The SpamRep Message SHALL contain the assembled SpamRep Document.

6.2.3 Spam Report Status Query

For information about the call flow for requesting a Spam Report status, refer to the informative Appendix B.2 of [SpamRep_AD].

When requesting a Spam Report status, the SpamRep Client:

1. SHALL assemble a Status Query message as described in section 5.1.3;
2. SHALL assemble a SpamRep Document containing the assembled Status Query Message Element as described in section 5.3; and,
3. SHALL send the SpamRep Message to the SpamRep Server as described in section 7. The SpamRep Message SHALL contain the assembled SpamRep Document.

6.2.4 Quarantined Messages Query

For information about the call flow for querying for quarantined messages, refer to the informative Appendix B.4 of [SpamRep_AD].

When querying for quarantined messages, the SpamRep Client:

1. SHALL assemble a Quarantined Messages Query message as described in section 5.1.4;
2. SHALL assemble a SpamRep Document containing the assembled Quarantined Messages Query Message Element as described in section 5.3; and,
3. SHALL send the SpamRep Message to the SpamRep Server as described in section 7. The SpamRep Message SHALL contain the assembled SpamRep Document.

6.3 Server Procedures

The procedures described in the following subsections are performed by the SpamRep Server.

6.3.1 Receiving a SpamRep Message

After receiving a SpamRep Message from a SpamRep Client, the SpamRep Server:

1. SHALL extract and parse the SpamRep Document(s) in order to extract the SpamRep Message Element enclosed within each SpamRep Document.
2. SHALL process each of the extracted SpamRep Message Elements as described in sections 6.3.1.1-6.3.1.4
3. SHALL send a response to the requesting SpamRep Client as described in section 6.3.2

6.3.1.1 Processing a Received Spam Report

After receiving a Spam Report Message Element, the SpamRep Server:

1. SHALL verify that the message being reported as spam is either included in the Spam Report Message Element when reporting is By-Value, or can be uniquely identified if reporting is By-Reference or By-Fingerprint.
2. SHALL generate the unique SpamReportID, create a Status Report Message Element and set the StatusCode element to “110” and/or StatusText to “Received” if the step 1 was successful.
3. SHALL create a Status Report Message Element. Depending on the Spam Report, the SpamRep Server set the StatusCode element, (e.g. StatusCode to “425” and/or StatusText to “By Value Required” if the step 1 was not successful because SpamRepServer wants the whole content and not a fingerprint).
4. SHALL update the SharePermissionList for the Reporter if SharePermission element is present in the Spam Report.

6.3.1.2 Processing a Received Action Request

After receiving an Action Request Message Element, the SpamRep Server:

1. SHALL attempt to perform the requested action by accessing external systems.
2. SHALL create an Action Response Message Element including StatusCode and/or StatusText, and enclose the response received from the external systems.

6.3.1.3 Processing a Received Spam Report Status Query

After receiving a Spam Report Status Query Message Element, the SpamRep Server:

1. SHALL retrieve the status for each SpamReportID contained in the Spam Report Status Query Message Element, either from its internal records or by accessing external systems.
2. SHALL create a Status Report Message Element for each SpamReportID and set the StatusCode element and/or StatusText to the value of the retrieved status.

6.3.1.4 Processing a Received Quarantined Messages Query

After receiving a Quarantined Messages Query Message Element, the SpamRep Server:

1. SHALL attempt to retrieve a list of quarantined messages by accessing external systems, e.g., using existing Messaging System interfaces to access the message box located in the Messaging System.
2. SHALL create a Quarantined Messages List Message Element and populate it with the list of quarantined messages obtained in step 1 and set the StatusCode element as “220” and/or StatusText as “Success”. If there are no quarantined messages, the Quarantined Messages List Message Element SHALL be empty and set the StatusCode element as “404” and/or StatusText as “Not Found”.

6.3.1.5 Preparing an Asynchronous Status Notification Message

If specified by policy, SpamRep Server MAY send asynchronous Spam Report status notification messages to the SpamRep Client. In such a case, the SpamRep Server:

1. SHALL retrieve the status for each intended SpamReportID, either from its internal records or by accessing external systems.
2. SHALL create a Status Report Message Element for each intended SpamReportID and set the StatusCode element and/or StatusText to the value of the retrieved status.

6.3.2 Sending a Response

After processing all of the SpamRep Message Elements received in the SpamRep Message, the SpamRep Server:

1. SHALL encapsulate all of the generated response SpamRep Message Elements in a new SpamRep Message.
2. SHALL send response to the requesting SpamRep Client containing the newly generated SpamRep Message as described in section 7.

7. Protocol

SpamRep Clients and Servers SHALL communicate via a transaction protocol. Each transaction SHALL consist of a SpamRep Client transmission of a SpamRep Message to a SpamRep Server, followed by a SpamRep Server transmission of a SpamRep Message response to the same SpamRep Client.

SpamRep Client SHALL initiate each transaction by sending an [RFC2616] HTTP POST request to the URI of the SpamRep Server. The body of this HTTP request SHALL contain a SpamRept Client-initiated SpamRep Message.

After receiving and processing the HTTP POST request, the SpamRep Server SHALL respond with an [RFC2616] HTTP response. The body of this HTTP response SHALL be a SpamRep Server-intiatied SpamRep Message.

This section covers only error cases specific to SpamRep. When there is no SpamRep message to send in response to a SpamRep Client request, the SpamRep Server SHALL send a 204 No Content response. Other allowed status codes, reflecting the outcome of the HTTP POST request, are defined in [RFC2616]. The Status-Line header MUST be included in the response to reflect the outcome of the HTTP POST submission.

8. Status Code and Text

The transaction of SpamRep is structured as a pair of request and response. A SpamRep Client sends a request or report message to the SpamRep Server in the form of a SpamRep Message (e.g. Spam Report, Status query, Action Request, Quarantined messages query) and the SpamRep Server responses in the form of a SpamRep Message(e.g. Report Status, Action Response, Quarantined message list).

Request SpamRep Message		Response SpamRep Message
Spam Report	↔	Report Status
Status Query	↔	Report Status
Action Request	↔	Action Response
Quarantined messages query	↔	Quarantined message list

Upon the SpamRep Message from the SpamRep Client, the SpamRep Server SHALL response with a StatusCode over “200” and under “400” if the request goes through the normal process.

When the SpamRep Server encounters an error while processing a SpamRep Message from the SpamRep Client, the SpamRep Server SHALL respond with a “4xx” or “5xx” StatusCode for each error encountered. Upon receiving a response with the StatusCode, the SpamRep Client MAY send another SpamRep Message.

The StatusText attribute value MAY be any text. It SHOULD contain human-readable text describing the status, and MAY contain additional helpful information.

The table describes which SpamRep Messages contain which error code (NOTE: These are not HTTP status codes).

Status Code	StatusText	SpamRep Message from SpamRep Client	Description
Normal Status Code			
210	Received	Spam Report Status Query	The Spam Report is received normally
211	Inspecting	Spam Report Status Query	The sender or message in Spam Report is under examination.
212	Applied	Spam Report Status Query	The server process is finished. (e.g. the sender is registered on the block list or get the warning point, the message in the Spam Report is recorded in the blocked message, virus is detected.)
213	Forwarding	Spam Report Status Query	The server process is finished and the Spam Report is forwarding to the third parties.
214	Completed	Spam Report Status Query	The all of process related with the Spam Report is completely finished.
215	Rejected	Spam Report Status Query Action Request	The Spam Report (e.g. in case of MNO notification or emergency message.) or Action Request is rejected.(e.g. depending on the MNO policy, ”Unblock Sender” has failed)
220	Success	Action Request Quarantined messages query	The Action Request or query is successful.
Error Code			
400	Bad Request	Spam Report Status Query	The SpamRep Message could not be understood by the SpamRep Server due to malformed syntax. The SpamRep

Status Code	StatusText	SpamRep Message from SpamRep Client	Description
		Action Request Quarantined messages query	Client SHOULD NOT repeat the request or report without modification.
401	Unauthorized Client	Spam Report Status Query Action Request	The SpamRep Client is not authorized by SpamRep Server.
404	Not Found	Spam Report Status Query Quarantined messages query	The SpamRep Server has not found anything matching the content in Spam Report, previous Spam Report or quarantined message.
409	Conflict	Spam Report Action Request	The request could not be completed due to a conflict with previous request.
410	Gone	Action Request Quarantined messages query	The requested resource is no longer available at the SpamRep Server
420	Unsupported Report Type	Spam Report	The SpamRep Server does not support the ReportType in the SpamRep Message.
421	Unsupported Abuse Type	Spam Report	The SpamRep Server does not support the AbuseType in the SpamRep Message.
422	Unsupported Message Type	Spam Report	The SpamRep Server does not support the MessageType in the SpamRep Message.
423	Unsupported Hashing function	Spam Report	The SpamRep Server does not support the value in "hashing-function" attribute in the SpamRep Message. If the SpamRep Server sends this value, the SpamRep Client MAY send the Spam Report once more using different hashing function.
424	Unsupported Third Party	Spam Report	The SpamRep Server does not support the ThirdPartyID in the SpamRep Message.
425	By Value Required	Spam Report	The SpamRep Server wants the whole content and not a fingerprint. If the SpamRep Server sends this value, the SpamRep Client SHALL send the Spam Report once more including whole content.
Additional Error Code			
500	Internal Server Error	Spam Report Status Query Action Request	The SpamRep Server encountered an unexpected condition which prevented it from fulfilling the request. This indicates that a later retry is also not expected to succeed.
503	Service Unavailable	Spam Report Status Query Action Request	The server is currently unable to handle the request due to a temporary overloading or maintenance of the server. The implication is that this is a temporary condition which will be alleviated after some delay. The SpamRep Client MAY retry the request after certain amount of time.
51x	(Now Defined)	Spam Report Status Query Action Request	SpamRep Server can assign additional error depending on its cases e.g. Resource Depletion

Table 18 Status Code

9. System Concepts

9.1 Authentication

The SpamRep interface SPR-1exposed by the SpamRep Server (see [SpamRep_AD]) SHALL provide SpamRep Client authentication, and SHOULD provide SpamRep Server authentication.

For a 3GPP IMS or 3GPP2 MMD realisation, the SPR-1 interface corresponds to the Ut Reference Point. In this case the authentication between the SpamRep Client and Server SHALL be performed according to [3GPP-TS_33.141] / [3GPP2-X.P0027-002].

If the Generic Authentication Architecture (GAA) as defined in [3GPP-TS_33.222] is not used, the SpamRep Client SHOULD and the SpamRep Server SHALL support [RFC2617] HTTP Digest Authentication.

The HTTP Digest authentication performed between SpamRep Client and Server SHALL conform to [RFC2617] with the following clarifications:

1. The HTTP "401 Unauthorized" error response SHALL be used;
2. The "username" parameter SHALL have the value of the SIP or Tel URI identifying the user (the Public User Identity) or the value of a parameter explicitly provisioned for the purpose of authentication;

The SpamRep Server SHALL support HTTP over TLS as specified in [RFC2818] for server authentication over the SPR-1 interface.

The SpamRep Client SHOULD support HTTP over TLS as specified in [RFC2818] for server authentication over the SPR-1 interface.

The Transport Layer Security (TLS) protocol is defined by [RFC5246].

An HTTP "403 Forbidden" error response SHALL be sent to the SpamRep Client after a number of successive failed responses to a challenge with the number of challenge failures is decided by local policy.

9.2 Authorization

The SpamRep Server SHALL support authorization of the Reporter based on Service Provider policies.

9.3 Privacy

The Spam Reporter's privacy SHOULD be protected. The SpamRep Message SHOULD be encrypted in order to protect privacy.As a best practice, the SpamRep server would be expected to obscure or delete personal information based on the Reporter's Share Permissions selection in the Spam Report in case it is forwarded by the SpamRep Server to other parties

Sharing of Spam Reports is allowed with the permission of the Spam Reporter, and the third parties on the list with whom the SpamRep Server is allowed to share the Spam Reports SHALL be consistent with that in the SpamRep Client. This SHALL be supported through the use of the Share Permission parameter in the Spam Report Message.

9.4 Security Considerations (Informative)

This section contains an informative discussion about security issues that developers should consider when making implementation choices. It contains no normative requirements.

9.4.1 Email Content

SpamRep supports the generation and exchange of reports about messages that Users or filters find objectionable for whatever reason, and specifies its transport. One of the message formats about which a report may be generated is electronic mail, or email. The format of an Internet message ("email"), independent of its transport, is defined in [RFC5322].

The Internet has long been a much more open network than the one used in general telecommunications. By virtue of its open research roots, the email ecosystem operates on a basis of little or no security and a general trust that actors will behave honourably. As the Internet grew and became more public, this layer of trust began to suffer widespread abuse from bad actors. Because of the absence of security in its transport, email actors are trivially able to change apparent identities by registering new domain names at low cost or in high volume, or creating new accounts, freely available from account providers. It is possible to send email nearly free of charge, in huge volumes, and with no requirement of a sustained single identity taking responsibility for the message. Moreover, as email has evolved, it has become possible to associate a plurality of identities, all different yet all possibly true, to a message. This has created a very complex message processing ecosystem, only recently properly documented in [RFC5598] and still evolving beyond that.

By contrast, access to the mobile messaging infrastructure is far less open, and actors can almost always be identified reliably and consistently as they are more bound to a single identity such as a MSISDN, and these identities are harder to acquire and change. As the network is less open, it is much more difficult to introduce forged data into the system.

Some mechanisms for messaging and identity security have evolved to address this issue, but to date they have not yet been widely deployed. Some of these include PGP [RFC4880], S/MIME [RFC5751] and DKIM [RFC4871], although the latter only verifies a domain name and not a complete user identity,

Software developers for mobile devices may therefore be unaware of these security issues involving handling of Internet messaging. A discussion of these issues, based on experience relayed in [RFC5598] and acquired from experts in the world of spam fighting on the Internet, and how they might play into development decisions of SpamRep packages is included below.

These considerations might also apply to other messaging protocols such as SMS and MMS if the mobile infrastructure evolves to have similar kinds of openness that the current Internet messaging infrastructure has.

9.4.1.1 Forgery

Both the content of an email message and the "envelope" used to move it via the typical transport mechanisms (usually, but not always, SMTP as defined in [RFC5321]) lack any integral security mechanisms. The implications for this are potentially severe; in the absence of such mechanisms, a bad actor can trivially forge a message with false content. This is a common tactic of bad actors, whose messages currently comprise the vast majority of email traffic. Moreover, not only can the message body be forged, but the portions of the message that contain identities and other meaningful protocol information (the "header") can also be forged, which leads to the ability to misrepresent the message's origin. This tactic is used by bad actors for multiple purposes:

1. To deceive receivers into trusting the body of the message;
2. To claim that a message comes from a particular trusted origin, bypassing naive filtering systems;
3. To divert responsibility for the message to a party (that may or may not actually exist) so that a meaningful response of any kind is not possible.

A common form of the first item listed above is what has come to be known as "phishing", wherein a recipient receives a message claiming to be from some organization with which the user has some sort of existing relationship asking the user to divulge sensitive personal information under the guise of its purported identity. The attacker is then able to use that data to gain unwarranted access to the victim's resources. An example of this is a claim from a bank that the user is being asked to verify its account credentials, redirecting the victim to a web site that appears legitimate but is hosted by the attacker. In 2004 the Federal Trade Commission held a workshop involving numerous messaging service providers and messaging security companies to address specifically this problem as fraudulent consumer losses passed well into nine figures of annual financial damage.

In the second case a list of known bad actors might be maintained by an operator, but since the bad actor can trivially change its identity, the efficacy of such a list is limited. Alternately, the bad actor can claim to be a user of the service attempting to secure itself, and in the absence of a widely deployed authentication mechanism the operator is unable to distinguish between a forged identity and a legitimate one.

In the third case, a reply of objection or other punitive action from a user or operator cannot be effective as it is directed to a nonexistent destination. Even worse, the falsified identity might be a legitimate address but one that had nothing to do with the transmission of the message, meaning complaints or punitive action are then directed to an innocent third party.

Such forgeries can sometimes be spotted by an adept user with access to some of the trace information included in the message header, but even automation of such analysis does not offer a high enough degree of accuracy to have warranted large scale deployment.

Given these implications, a SpamRep implementer, especially one developing a SpamRep Server. is encouraged not to take any automatic action based on any unverified identity associated with the message. For example:

1. Consider a SpamRep Server implementation that, in response to a SpamRep Report about an email message, generates a complaint message to the address found in the "From" field about that message. An attacker aware of this property of the Server can attack a victim by sending a large number of mobile users some junk email claiming to come from the victim; as recipients complain via SpamRep Reports, the Servers will bombard the victim with these generated complaint messages.
2. In the same scenario, the Server's complaint message might be sent to a concocted, non-existent address, which will itself generate a bounce message back to the Server. This could happen in high volumes, wasting processing power at a number of Internet nodes.
3. A SpamRep server that blocks a sender about whom it receives a SpamRep Report can be caused to block legitimate mail by a bad actor that forges email to claim to come from legitimate senders. This would constitute a denial-of-service attack.

9.4.1.2 Multiple Identities

A further complication is the fact that a single message may have many associated identities (and, as discussed above, none of them can actually be trusted). For example, a single delivered message will have a "From" header field (the only one that is mandatory) but can legitimately also have a "Sender" field, a "Reply-To" field and/or a "Return-Path" field. Moreover, some or all of these fields are permitted to contain multiple identities.

Thus, there can be multiple identities associated with a message. There is, however, no established heuristic to indicate, given any combination of them, which should be considered more likely to be correct. Thus selecting an identity on which to base some action cannot be reasonably accomplished.

Absent the development of such conventions endorsed by the email community, designs involving such heuristics should be avoided.

9.4.1.3 High Volumes

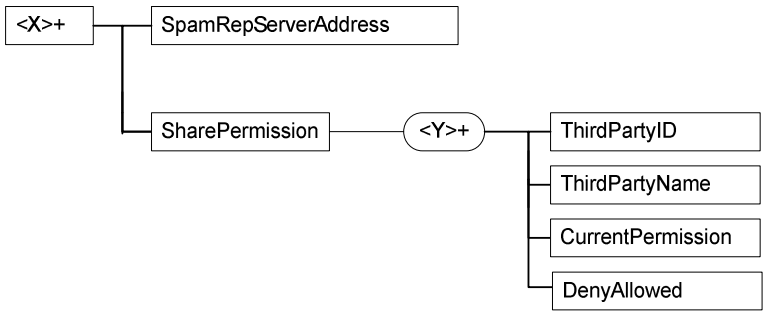
Since the cost of sending email is nearly zero, it is common for attackers to send large amounts of mail to a large distribution of users. This naturally results in a large number of complaints as Users process their mailboxes.

Server implementers would be well advised to design to accommodate large, bursty complaint volumes.

10. Management Object

10.1 Management Object Tree

The following figure shows the OMA SpamRep Management Object.



10.2 Management Object parameters

This section describes the parameters for the OMA SpamRep Management Object.

10.2.1 Node: /<X>

Status	Occurrence	Format	Min. Access Types
Mandatory	OneOrMore	node	Get

This interior node acts as a placeholder for one or more accounts of services and/or SpamRep Clients.

10.2.2 Node: /<X>/SpamRepServerAddress

Status	Occurrence	Format	Min. Access Types
Mandatory	One	chr	Add, Delete, Replace, Get,

This interior node provides the address of the SpamRep Server with which the SpamRep Client instance communicates.

10.2.3 Node: /<X>/SharePermission

Status	Occurrence	Format	Min. Access Types
Mandatory	One	chr	Add, Delete, Replace, Get,

This parameter acts as a parent node for all SharePermission objects.

10.2.4 Node: /<X>/SharePermission/<Y>

Status	Occurrence	Format	Min. Access Types
Required	OneOrMore	node	Add, Delete, Replace, Get,

This interior node acts as a placeholder node for a SharePermission object and its parameters.

10.2.5 Node: /<X>/SharePermission/<Y>/ThirdPartyID

Status	Occurrence	Format	Min. Access Types
Mandatory	One	chr	Add, Delete, Replace, Get,

The ThirdPartyID leaf indicates the linkage to SharePermission parameters. This parameter provides the ID for the third party.

10.2.6 Node: /<X>/SharePermission/<Y>/ThirdPartyName

Status	Occurrence	Format	Min. Access Types
Mandatory	One	chr	Add, Delete, Replace, Get,

The ThirdPartyName leaf indicates the linkage to SharePermission parameters. This parameter provides the name for the third party (e.g. government agency, operators).

10.2.7 Node: /<X>/SharePermission/<Y>/CurrentPermission

Status	Occurrence	Format	Min. Access Types
Mandatory	One	chr	Add, Delete, Replace, Get,

The Permission indicates the linkage to CurrentPermission parameters. This parameter represents the default permissible degree of sharing personal information with the third party (e.g. government agency, operators). It is expected to be the current degree of personal information (“Entire message”, “Email / phone number”, “Anonymous”, “Deny”).

10.2.8 Node: /<X>/SharePermission/<Y>/DenyAllowed

Status	Occurrence	Format	Min. Access Types
Mandatory	One	bool	Add, Delete, Replace, Get,

The DenyAllowed indicates whether the sharing the Spam Report can be revoked by the SpamRep Client or not. If the value is true, the Spam Report can revoke sharing its Spam Report with the third party.

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
n/a	n/a	No prior version –or- No previous version within OMA
OMA-xxyyz-V1_0-20021001-A	01 Oct 2002	Initial document to address the basic starting point Ref TP Doc# OMA-TP-2002-1234-xxyyzForApproval
OMA-xxyyz-V1_1-20030405-A	05 Apr 2003	description of changed Ref TP Doc# OMA-TP-2003-0321-xxyyzV1_1forApproval

A.2 Draft/Candidate Version 1.0 History

Document Identifier	Date	Sections	Description
Draft Versions OMA-TS-SpamRep-V1_0	09 Sep 2009	All	Initial Draft
	13 Nov 2009	5+	Incorporates input to committee: OMA-MWG-SpamRep-2009-0048-CR_TS_Detailed_Outline
	11 Feb 2010	5, 5.1, 5.2, 5.1.1, 5.1.2, 5.1.3,5.1.4, 5.2.1, 5.2.2, 5.2.3, 5.3, 5.3.1, 5.3.2, 5.3.3, 5.4	Incorporates input to committee: OMA-MWG-SpamRep-2010-0006-CR_Messages OMA-MWG-SpamRep-2010-0008R01-CR_ClientMessages OMA-MWG-SpamRep-2010-0015R01-CR_ServerMessages OMA-MWG-SpamRep-2010-0011R01-CR_MessageSchema OMA-MWG-SpamRep-2010-0001R02-CR_TS_Parameters OMA-MWG-SpamRep-2010-0012R01-CR_Parameters_SpamReport_Status_Query OMA-MWG-SpamRep-2010-0013R01-CR_Para__Retrieving_list_quarantined_messages
	06 Mar 2010	5.2.1, 5.2.2, 5.2.3	OMA-MWG-SpamRep-2010-0015R01-CR_ServerMessages
	17 Mar 2010	2.1, 9.1, 9.2	OMA-COM-SpamRep-2010-0022R03-CR_Authentication
	05 Apr 2010	5.1,5.2, 7, 2.1, 3.2, 6.2, 6.3, 5.3.1,	OMA-MWG-SpamRep-2010-0021-CR_MessageNameChange OMA-MWG-SpamRep-2010-0027R02-CR_Protocol OMA-COM-SpamRep-2010-0029R02-CR_ClientProcedures OMA-COM-SpamRep-2010-0030R03-CR_ServerProcedures OMA-COM-SpamRep-2010-0038R01-CR_MessageElements
	21 Apr 2010	5.1.1, 5.1.2, 5.2.1, 5.2.3, 6.3.1.4, 6.3.1.5, 2.1, 5.1.1.1, 5.1.1.2	OMA-COM-SpamRep-2010-0041R03-CR_SpamRepMsg OMA-COM-SpamRep-2010-0043R01-CR_ActionRequest OMA-COM-SpamRep-2010-0046-CR_Server_Msg_Fixes OMA-COM-SpamRep-2010-0047-CR_Add_Server_Procedure OMA-COM-SpamRep-2010-0049R01-CR_MessageAttributes OMA-COM-SpamRep-2010-0051R01-CR_MessageReference
	02 May 2010	5.1.1.3, 5.1.1.2	OMA-COM-SpamRep-2010-0054R01-CR_Msg_Fingerprint OMA-COM-SpamRep-2010-0063R01-CR_MessageReference
	01 Jun 2010	5.1.4, 5.2.1, 5.2.2, 3.2, 5, 5.3, 5.4, 5.1.1, 6.1, 6.2.1, 6.2.2, 6.2.3, 6.2.4	OMA-COM-SpamRep-2010-0068-CR_MessageElementFixes OMA-COM-SpamRep-2010-0052R04-CR_Messages OMA-COM-SpamRep-2010-0045R01-CR_SpamReport_Msg_Opt_Para OMA-COM-SpamRep-2010-0061R02-CR_Share_Permission OMA-COM-SpamRep-2010-0071R01-CR_SpamRep_Msg_MIME_Def_ OMA-COM-SpamRep-2010-0072R01-CR_Procedure_Clarification_
	21 Jun 2010	5.4, 5.2.4, 2.2, 5.1.1, 6.2, 6.3, 4.1, 5.1, 2.1, 3.2	OMA-COM-SpamRep-2010-0082R02-CR_Share_Permission_Comment OMA-COM-SpamRep-2010-0083R02-CR_Share_Permission_Provisioning OMA-COM-SpamRep-2010-0084R02-CR_Share_Permission_SpamReport OMA-COM-SpamRep-2010-0088R01-CR_section_4.1 OMA-COM-SpamRep-2010-0076R02-CR_AbuseType
	22 Jun 2010	All, 5.1.1.1, 2.1, 5.1.1	Editorial changes, OMA-COM-SpamRep-2010-0059R05-CR_SMS_Attributes
	24 Jun 2010	5.2.2, 5.4, 7.1, 7.2, 7.3, 9.2	OMA-COM-SpamRep-2010-0077R01-CR_TS_Cleanup

Document Identifier	Date	Sections	Description
	07 Jul 2010	2.1 3.1 3.2 Appendix B	Minor editorial fixes Sorting of references, abbreviations and definitions Contents page updated Removal of empty example sections OMA-COM-SpamRep-2010-0095R01-CR_Opt_out_Definition OMA-COM-SpamRep-2010-0094R02-CR_SCR_Table
	27 Aug 2010	5.2.1, 5.2.2, 5.2.3, 4, 4.1, 9.4, 2.2, 5.1.1.3, 5.1.1, 7, 5.1.2, 9.3	OMA-COM-SpamRep-2010-0109R01-CR_Report_Status OMA-COM-SpamRep-2010-0111R01-CR_Intro_non_UA_client OMA-COM-SpamRep-2010-0119R01-CR_SpamRep_Security_Considerations OMA-COM-SpamRep-2010-0122R01-CR_TS_Comment_C051 OMA-COM-SpamRep-2010-0123R01-CR_Same_vesion_number OMA-COM-SpamRep-2010-0129R01-CR_Comments_C086_C088 OMA-COM-SpamRep-2010-0134R01-CR_Comment_C053_to_TS OMA-COM-SpamRep-2010-0135R01-CR_comments_C092
	10 Sept 2010	2.1, 2.2, 3.2, 4, 5, 5.1, 5.1.1, 5.1.1.1, 5.1.1.2, 5.1.1.3, 5.1.2, 5.1.3, 5.1.2, 5.2, 5.2.1, 5.2.3, 5.3.3, 5.4, 6.2.1, 7, Appendix C	Editorial AIs: SpamRep-2010-A002 SpamRep-2010-A003 Correction of 9.2 heading OMA-COM-SpamRep-2010-0112R03-CR_SpamRep_MIME_types_and_structure OMA-COM-SpamRep-2010-0126R02-CR_Examples_Section OMA-COM-SpamRep-2010-0139-CR_Comments_C007_C008 OMA-COM-SpamRep-2010-0141-CR_Comment_C032 OMA-COM-SpamRep-2010-0142R01-CR_Comment_C043 OMA-COM-SpamRep-2010-0143R01-CR_Comment_C055 OMA-COM-SpamRep-2010-0151R01-CR_Report_Type OMA-COM-SpamRep-2010-0152R01-CR_Submit_Reporting OMA-COM-SpamRep-2010-0153R01-CR_Quarantined_Messages_List_Bugfix OMA-COM-SpamRep-2010-0154-CR_Fingerprint_Tweak
	13 Sept 2010	2, 5, 6, 8, 10, Appendix C, D, E	Editorial: Corrected revision numbers in previous change history OMA-COM-SpamRep-2010-0089R04-CR_Share_Permission_MO OMA-COM-SpamRep-2010-0103R03-CR_Email_Specific_Revisions OMA-COM-SpamRep-2010-0105R03-CR_Status_Code OMA-COM-SpamRep-2010-0114R01-CR_Spam_Detection_Information_Attribute
	03 Oct 2010	All	Editorial: Remaining CONRR comments C016, C021, C038, C039, C040, C044, C045, C049, C052, C054, C057, C058, C059, C060, C061, C063, C065, C066, C067, C068, C070, C071, C073, C074, C077, C081, C082, C083, C084, C085 General grammar, spelling, sentence structure clean up, message structure clarification OMA-COM-SpamRep-2010-0159R01-CR_Quarantined_Messages_List_Bugfix OMA-COM-SpamRep-2010-0161R02-CR_Msg_Structure_Related_Description_Bugfix OMA-COM-SpamRep-2010-0162R02-CR_Remove_Opt_Out_Def OMA-COM-SpamRep-2010-0163R02-CR_Fix_Reporting_Example OMA-COM-SpamRep-2010-0164R02-CR_Section_5.4_Bug_Fixes OMA-COM-SpamRep-2010-0165R01-CR_Section_5.1.1_Bug_Fixes
Candidate Version OMA-TS-SpamRep-V1_0	23 Nov 2010	All	Status changed to Candidate by TP: OMA-TP-2010-0465-INP_SpamRep_V1_0_ERP_for_Candidate_Approval

Appendix B. Static Conformance Requirements (Normative)

The notation used in this appendix is specified in [SCRRULES].

B.1 SCR for SpamRep Client (Protocol)

Item	Function	Reference	Requirement
SPAM-PROT-C-001-M	Support HTTP protocol	Section 7	
SPAM-PROT-C-002-M	Support exchange of SpamRep Messages	Section 7	SPAM-PROT-C-003-M
SPAM-PROT-C-003-M	Creation of SpamRep Messages	Section 5	SPAM-MESS-C-001-M AND (SPAM-MESS-C-002-M OR SPAM-MESS-C-003-M OR SPAM-MESS-C-004-M)

B.2 SCR for SpamRep Server (Protocol)

Item	Function	Reference	Requirement
SPAM-PROT-S-001-M	Support HTTP protocol	Section 7	
SPAM-PROT-S-002-M	Support exchange of SpamRep Messages	Section 7	SPAM-PROT-S-003-M
SPAM-PROT-S-003-M	Creation of SpamRep Messages	Section 5	SPAM-MESS-S-001-M

B.3 SCR for SpamRep Client (Messages)

Item	Function	Reference	Requirement
SPAM-MESS-C-001-M	Support creation of SpamRep Documents	Section 5.1	SPAM-DOCS-C-001-M OR SPAM-DOCS-C-002-M OR SPAM-DOCS-C-003-M OR SPAM-DOCS-C-004-M
SPAM-MESS-C-002-M	Support inclusion of Content	Section 5	
SPAM-MESS-C-003-M	Support Message Referencing	Section 5.1.1.2	
SPAM-MESS-C-004-M	Support Message Fingerprinting	Section 5.1.1.3	

B.4 SCR for SpamRep Server (Messages)

Item	Function	Reference	Requirement
SPAM-MESS-S-001-M	Support creation of SpamRep Documents	Section 5.2	SPAM-DOCS-S-001-M OR SPAM-DOCS-S-002-M OR SPAM-DOCS-S-003-M

B.5 SCR for SpamRep Client (Documents)

Item	Function	Reference	Requirement
SPAM-DOCS-C-001-M	Support creation of Spam Report	Section 5.1	
SPAM-DOCS-C-002-O	Support creation of Action Request	Section 5.1	
SPAM-DOCS-C-003-O	Support creation of Status Query	Section 5.1	
SPAM-DOCS-C-004-O	Support creation of Quarantined Message	Section 5.1	

Item	Function	Reference	Requirement
	Query		
SPAM-DOCS-C-005-M	Support receipt of Report Status	Section 5.2.1	
SPAM-DOCS-C-006-O	Support receipt of Action Response	Section 5.2.2	
SPAM-DOCS-C-007-O	Support receipt of Quarantined Message List document	Section 5.2.2	

B.6 SCR for SpamRep Server (Documents)

Item	Function	Reference	Requirement
SPAM-DOCS-S-001-M	Support creation of Report Status	Section 5.2.1	
SPAM-DOCS-S-002-M	Support creation of Action Response	Section 5.2.2	
SPAM-DOCS-S-003-M	Support creation of Quarantined Message List document	Section 5.2.2	
SPAM-DOCS-S-004-M	Support receipt of Spam Report	Section 5.1	
SPAM-DOCS-S-005-M	Support receipt of Action Request	Section 5.1	
SPAM-DOCS-S-006-M	Support receipt of Status Query	Section 5.1	
SPAM-DOCS-S-007-M	Support receipt of Quarantined Message Query	Section 5.1	

B.7 SCR for SpamRep Client (Procedures)

Item	Function	Reference	Requirement
SPAM-PRCD-C-001-M	Support submission of Spam Report	Section 6.2.1	
SPAM-PRCD-C-002-O	Support submission of Action Request	Section 6.2.2	
SPAM-PRCD-C-003-O	Support submission of Status Query	Section 6.2.3	
SPAM-PRCD-C-004-O	Support submission of Quarantined Message Query	Section 6.2.4	

B.8 SCR for SpamRep Server (Procedures)

Item	Function	Reference	Requirement
SPAM-PRCD-S-001-M	Support reception and processing of Spam Report	Section 6.3.1.1, 6.3.2	
SPAM-PRCD-S-002-M	Support reception and processing of Action Request	Section 6.3.1.2, 6.3.2	
SPAM-PRCD-S-003-M	Support reception and processing of Status	Section 6.3.1.3, 6.3.2	

Item	Function	Reference	Requirement
	Query		
SPAM-PRCD-S-004-M	Support reception and processing of Quarantined Message Query	Section 6.3.1.4, 6.3.2	
SPAM-PRCD-S-005-O	Support Asynchronous Status Messaging	Section 6.3.1.5, 6.3.2	

B.9 SCR for SpamRep Client (Authentication)

Item	Function	Reference	Requirement
SPAM-AUTH-C-001-O	Support Server authentication	Section 9.1	

B.10 SCR for SpamRep Server (Authentication)

Item	Function	Reference	Requirement
SPAM- AUTH-S-001-M	Support Client authentication	Section 9.1	

Appendix C. SpamRep Management Object (Informative)

SpamRep Management Object (MO) is a proposed object for SpamRep Enabler that allows provisioning of SpamRep server address and share permission in a device. The SpamRep Enabler may retrieve and manage connection information (e.g. correct SpamRep Server Address) and share permission (e.g. third party ID and name) using this MO.

The OMA SpamRep MO is defined using the OMA DM Device Description Framework and is compatible with OMA DM protocol version 1.2 [DMPRO] or any later compatible version. If SpamRep MO is to be configured during initial configuration (i.e. bootstrap) then the DM Profile, as described in [DMBOOT], can be used.

The proposed Management Object Identifier for the SpamRep Management Object is: ***urn:oma:mo:oma-spamrep:1.0***.

Appendix D. IANA Media Type Registration Template (Informative)

The following has been submitted to the IETF as a registration template for the MIME media type defined by this document:

To: <ietf-types@iana.org>
Subject: Registration of media type application/vnd.oma.spamrep+xml

Type name: application
Subtype name: vnd.oma.spamrep+xml
Required parameters: none
Optional parameters: none
Encoding considerations: May use 7bit, 8bit or binary
Security considerations: See Section 8 of [ID.IETF-MARF-BASE]
Interoperability considerations: none
Published specification: [this document]
Applications that use this media type: OMA SpamRep Enabler compliant applications
Additional information: none
Person & email address to contact for further information: Murray S. Kucherawy <msk@cloudmark.com>
Intended usage: COMMON
Restrictions on usage: none
Author: Open Mobile Alliance (OMA)
Change controller: Open Mobile Alliance (OMA)

The chair or his designee should submit the above registration as an Internet Draft to the IETF upon publication of the final version of the SpamRep TS (and I would be happy to act as said designee).

"[ID.IETF-MARF-BASE]" should be replaced by an RFC number if that memo is published before this document. "[this document]" should be replaced by the final name of this document upon publication.

Appendix E. Examples (Informative)

This section includes example uses for SpamRep, illustrating messages generated both by Client and Server and describing their interactions. These examples are informative only; where there are discrepancies between the examples and the earlier normative sections of this specification, the normative sections should be followed.

E.1 Email

This subsection illustrates the use of SpamRep in the case of a piece of unwanted email received at a mobile device, and the user of that device choosing to report the content to its provider.

E.1.1 Email spam

The following email message is received at a mobile device. The email format is defined in [RFC5322].

```
Received: from make.money.fast.example.com by mobile-dc.example.net
          via ESMTP; Thu 5 Aug 2010 11:28:09 -0700 (PDT)

From: John Q. Public <jqpublic-109231@example.com>

Message-Id: <msg91823@example.com>

To: Jane Doe <mobileUser@example.net>

Subject: Cheap pills!


Find your best pharmaceutical prices online!  Check out our web site:
http://pills.example.com
```

The receiving user objects to the content of this message and activates a spam reporting feature, such as pushing a "Report Spam" button on the user interface.

E.1.2 SpamRep Report from Client

Provisioned within the mobile device is the address of an HTTP [RFC2616] service, specified by a URL [RFC3986] to which the SpamRep Report will be posted. This URL identifies the SpamRep Server. Thus, the mobile device makes a connection to that URL and the following HTTP POST action is sent over that connection:

```
POST /spamrep HTTP/1.1

Content-Type:  multipart/report; report-type=vnd.oma.spamrep+xml;
   boundary="spamrepboundary12345"

Content-Length: 1142


--spamrepboundary12345
Content-Transfer-Encoding: 7bit

Content-Type: text/plain; charset=us-ascii


This is an OMA SpamRep spam report.  At 2010-08-10T19:08:50.52Z
mobileUser@example.net reported spam from jqpublic-109231@example.com


--spamrepboundary12345

Content-Type: application/vnd.oma.spamrep+xml
```

```
<spam-rep-document>
      <spam-report>
            <SpamRepMessageID> 9832751092741 </SpamRepMessageID>
            <SpamRepClientID> 4155551212 </SpamRepClientID>
            <ReportType> By-Reference </ReportType>
            <HashingFunction> MD5 </HashingFunction>
            <MessageReference> aHOxLLGVx8zMaqMhIp4UjQ6TdMw= </MessageReference>
            <MessageType> Email </MessageType>
            <MessageAttributes>
                  <Message-Id> &lt;msg91823@example.com&gt; </Message-Id>
                  <To> mobileUser@example.net </To>
                  <From> jqpublic-109231@example.com </From>
            </MessageAttributes>
            <SubmissionTime> 2010-08-10T19:08:50.52Z </SubmissionTime>
            <OriginatingAddress> jqpublic-109231@example.com </OriginatingAddress>
            <AbuseType> 0 </AbuseType>
            <Version> 1.0 </Version>
      </spam-report>
</spam-rep-document>

--spamrepboundary12345--
```

For the purpose of bandwidth conservation, the Client has elected to make its report using the By-Reference mechanism defined in Section 5.1.1.2. The hash has been computed using the SHA-1 hash algorithm, with the input being the entire header block of the spam shown in X.1.1 above. Per the email definition document, each line in the input message is terminated with a carriage return (ASCII 13) and a line feed (ASCII 10), and there is an empty line between the header and the body. The Received: field is continued across two lines, and the continued portion is indented by a horizontal tab character (ASCII 9). Therefore the total input of the example is 1142 bytes in length, and the resulting SHA1 hash expressed in hexadecimal byte form is 6873b12cb195c7cccc6aa321229e148d0e9374cc. The hash is encoded using base64 which is defined in [RFC2045].

E.1.3 SpamRep Report response from Server

The SpamRep Client retains the connection to the Server open until it receives a reply. In this example, the Server decides it has not received sufficient information from the Client to be able to act on the report, so it requests the complete original message instead of the reference. The reply thus received is:

```
Content-Type: multipart/related; boundary=”spamrepboundary34567”
Content-Length: 309

--spamrepboundary34567
Content-Type: application/vnd.oma.spamrep+xml
```

```
<spam-rep-document>
      <spam-report-status>
             <SpamReportID> 123456789 </SpamReportID>
             <SpamRepMessageID> 9832751092741 </SpamRepMessageID>
             <StatusCode> 425 </StatusCode>
             <StatusText> By Value Required </StatusText>
      </spam-report-status>
</spam-rep-document>

--spamrepboundary34567—
```

E.1.4 SpamRep Client re-submission

Here the SpamRep Client reissues its original request using the By-Value mechanism, as directed by the server.

```
POST /spamrep HTTP/1.1
Content-Type: multipart/related; boundary=”spamrepboundary02468”
Content-Length: 1049

--spamrepboundary02468
Content-Type: application/vnd.oma.spamrep+xml

<spam-rep-document>
      <spam-report>
             <SpamRepMessageID> 9832751092741 </SpamRepMessageID>
             <SpamRepClientID> 4155551212 </SpamRepClientID>
             <ReportType> By-Value </ReportType>
             <MessageType> Email </MessageType>
             <MessageReference> ref1123@example.net </MessageReference>
             <SubmissionTime> 2010-08-10T19:08:50.52Z  </SubmissionTime>
             <OriginatingAddress> jqpublic-109231@example.com </OriginatingAddress>
             <AbuseType> 0 </AbuseType>
             <Version> 1.0 </Version>
      </spam-report>
</spam-rep-document>

--spamrepboundary02468
Content-Type: application/octet-stream
Content-ID: ref1123@example.net
```

```
Received: from make.money.fast.example.com by mobile-dc.example.net
          via ESMTP; Thu 5 Aug 2010 11:28:09 -0700 (PDT)
From: John Q. Public <jqpublic-109231@example.com>
Message-Id: <msg91823@example.com>
To: Jane Doe <mobileUser@example.net>
Subject: Cheap pills!

Find your best pharmaceutical prices online!  Check out our web site:
http://pills.example.com

--spamrepboundary02468—
```

E.1.5 SpamRep Report response from Server

Now the Server has accepted the report from the Client.

```
Content-Type: multipart/related; boundary=”spamrepboundary56789”
Content-Length: 350

--spamrepboundary56789
Content-Type: application/vnd.oma.spamrep+xml

<spam-rep-document>
      <spam-report-status>
             <SpamRepMessageID> 9832751092741 </SpamRepMessageID>
             <SpamReportID> 811873119213 </SpamReportID>
             <StatusCode> 110 </StatusCode>
             <SpamReportStatus> Received </SpamReportStatus>
      </spam-report-status>
</spam-rep-document>

--spamrepboundary56789—
```

E.1.6 SpamRep Client requests a status report

In this instance some time has passed since the above spam report was made, and the user would like to know what the results are.

```
Content-Type: multipart/related; boundary=”spamrepboundary01010”
```

```
Content-Length: 296

--spamrepboundary01010
Content-Type: application/vnd.oma.spamrep+xml

<spam-rep-document>
      <status-query>
             <SpamReportID> 811873119213 </SpamReportID>
      </status-query>
</spam-rep-document>

--spamrepboundary01010—
```

E.1.7 SpamRep Server responds with a status report

The Server responds to the above request.

```
Content-Type: multipart/related; boundary=”spamrepboundary98765”
Content-Length: 389

--spamrepboundary98765
Content-Type: application/vnd.oma.spamrep+xml

<spam-rep-document>
      <spam-report-status>
             <SpamRepMessageID> 11112243890 </SpamRepMessageID>
             <SpamReportID> 811873119213 </SpamReportID>
             <StatusCode> 410 </StatusCode>
             <StatusText> Domain blocked </StatusText>
      </spam-report-status>
</spam-rep-document>

--spamrepboundary98765—
```

E.1.8 SpamRep Server error

The Server is unable to accept the spam report submitted in X.1.4 as it does not understand or support SHA-1 hashing.

```
Content-Type: multipart/related; boundary=”spamrepboundary98989”
```

```
Content-Length: 288

--spamrepboundary98989
Content-Type: application/vnd.oma.spamrep+xml

<spam-rep-document>
      <spam-report-status>
             <SpamReportID> 811873119213 </SpamReportID>
             <StatusCode> 423 </StatusCode>
      </spam-report-status>
</spam-rep-document>

--spamrepboundary98989—
```