Network Working Group
Internet-Draft
Intended status: Experimental
Expires: August 31, 2012

Z. Ordogh
Research In Motion Limited

February 28, 2012

Spam reporting using IMAP: SREP
draft-ordogh-spam-reporting-using-imap-01

Abstract

This document defines an IMAP extension which allows a client to report spam by reference and allows an IMAP server to perform any action on the reported messages, including leaving the action at the client's discretion.

In addition, this document discusses how an IMAP server can tap into spam aggregator services, ultimately allowing the IMAP server to improve its decision-making process.

Conventions Used In This Document

In examples, "C:" and "S:" indicate lines sent by the client or the server, respectively.

The key words "MUST", "MUST NOT", "SHOULD", "SHOULD NOT", and "MAY" in this document are to be interpreted as described in [RFC2119].

This specification follows the recommendations in [XDASH].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 31, 2012.

Copyright Notice

Table of Contents

1. Introduction

The Internet Message Access Protocol [IMAP4] does not support reporting spam on its own. There are a number of solutions available based on the multipart/report content type defined in [OLD-REPORT] and its revision, [REPORT]. However, these solutions require including the message contents and hence, consume bandwidth to transmit the entire message. In bandwidth-constrained environments - such as mobile networks - it is highly desirable to send only a minimum set of information - a reference - instead of the entire message. Solutions that exist today employ manipulating proprietary flags in the IMAP storage to achieve the bare minimum, however more advanced solutions cannot be developed by using flags only; the IMAP server needs to be involved actively in the spam reporting process.

Furthermore, it is highly desirable to permit individual server implementations to handle spam in any way these systems choose to: do nothing, flag, perform deletion or relocation, recommend deletion or relocation to the client, or, leave the decision at the client's discretion as a whole. However, in order to make such a decision on the server side, a spam aggregator service, such as [OMA-SPAMREP], needs to be involved in the decision-making process.

This document specifies a new IMAP command, SREP, along with its syntax, which allows a client to inform the server that the user considered a message (or parts thereof) spam, or, that the user no longer considers a message (or parts thereof) spam. Since all information about the message is readily available on the server, the command also allows the server to implement a more intelligent and accurate decision logic, which may be invoked when the spam is reported and the server can respond with its decision to the client.

Additionally, this document contains example flows, illustrating various decisions that the server may choose to evaluate, including invoking an aggregator service, such as one based on [OMA-SPAMREP].

The SREP command allows:

- reporting spam; i.e. set the spam condition, and,
- reporting that a message (that was reported spam earlier) is no longer spam; i.e. clear the spam condition.

2. Scope

This document focuses only on the client-server interactions and the scope is limited to messages that either exist on the IMAP server, or, exist elsewhere and the IMAP server is configured to access them. Consequently, deposit-time filtering, messages that have been deleted, and messages that exist in an external storage but are

accessible only via an access protocol unknown to the IMAP server are out of scope.

3. The SREP command

The SREP command follows the conventions of [IMAP4].

Arguments:
- directive; see Section 3.1
- OPTIONAL abuse type; see Section 3.2
- reference; see Section 3.3
- OPTIONAL list of part identifiers; see Section 3.4
- OPTIONAL request action; see Section 3.5

Responses; see Section 3.6:
- OPTIONAL OK response: RELOCATE
- OPTIONAL OK response: RELOCATED
- OPTIONAL OK response: DELETE
- OPTIONAL OK response: DELETED
- OPTIONAL OK response: KEYWORD

Result:
- OK - command completed successfully
- NO - the server cannot access one or more messages (deleted or unauthorized)
- BAD - there was an error during processing the command (syntax or unsupported parameter)

The formal syntax of the SREP command is defined in Section 3.7.

The SREP command allows:
- reporting spam; i.e. set the spam condition, and,
- reporting that a message (that was reported spam earlier) is no longer spam; i.e. clear the spam condition.

The SREP command may be used with any IMAP4 server implementation that returns "SREP" as one of the supported capabilities in response to the CAPABILITY command. If the server does not indicate support for the SREP capability, the client MUST NOT use the SREP command.

The SREP command may result in ambiguity, therefore the client MUST NOT send any commands before the result of the SREP command has been received, see Section 5.5 in [IMAP].

The command MAY be issued on one or more messages at a time, in the currently selected mailbox.

The command MAY be extended in the future with new parameters

(actions, directives, reference types, etc). Servers MUST be able to recognize parameters unknown to them and respond with a BAD response in case they encounter such a parameter.

3.1. Directives

The directive argument tells the server whether a message is being reported as a spam, or, it is being reported as no longer a spam. The SREP command MUST include the directive. To report a spam, the directive MUST be SET. To report that a message is no longer considered to be a spam, the directive MUST be CLEAR.

Extensions are permitted, as defined in Section 3.7.

3.2. Abuse types

The client may have additional information about the spam regarding the nature of the abuse. When such information is available, the client SHOULD include the abuse type argument in the request. When such information is not available, the client MUST omit the abuse type argument from the request. When the directive argument is CLEAR, the client MUST omit the abuse type argument from the request. This specification defines the following abuse types:

1. Phishing (forgery, link manipulation, etc.): an attempt to divulge information from the recipient by masquerading the sender and/or the content(s) of the message as a trustworthy form of communication.
2. Malware (virus, spyware, etc.): a malicious piece of software code embedded or attached to the message specifically designed to disrupt normal operation, gather sensitive information, gain unauthorized access, and/or perform other abusive behavior upon execution.

Extensions are permitted, as defined in Section 3.7.

3.3. References

The reference argument consists of a reference type and a reference value. In general, the reference type MUST indicate the format of the reference while the reference value MUST contain a value corresponding to the indicated reference format. To use a unique identifier specified in [IMAP4], the reference type MUST be UID and the reference value MUST be a number expressing the unique identifier of the message. To use a sequence set specified in [IMAP4], the reference type MUST be SEQ and the reference value MUST be sequence numbers corresponding to the specified message sequence number set. To use an authorized URL specified in [URLAUTH], the reference type

MUST be URLAUTH and the reference value MUST be an URLAUTH-authorized URL, authorizing the entire message.

Extensions are permitted, as defined in Section 3.7.

3.4. Part identifiers

When the reference identifies one and only one message, the list of part identifiers MAY be included to improve the accuracy of spam detection. When the reference identifies more than one message, the list of part identifiers MUST be omitted.

The list of part identifiers is a parenthesized list of part identifiers. Part identifiers MAY identify header fields or bodies. Header field identifiers MUST be prefixed with the word 'header' and the dot ('.') character MUST be used as the separator character. Header fields MUST be identified by the name of the header field.

Example:
The 'From' header field is identified as 'header.from'.

Body identifiers MUST be prefixed with the word 'body' and the dot ('.') character MUST be used as the separator character. Bodies MUST be identified by their positions within the message hierarcy, where the first position is 1 and the main level is 1. To refer the entire body of a message (or all bodies of a multipart message), the separator character, the position MUST be omitted.

Examples:
- The entire body of a message (or all bodies of a multipart message) is identified as 'body'.
- Considering a simple multipart message, the part following the first boundary is identified as 'body.1'.
- Considering a multipart message that includes an email attachment following the second boundary, and the email attachment containing text following the first boundary, the text within the email message is identified as 'body.2.1'.

The formal syntax of the part-id-list is defined in Section 3.7.

3.5. Request Action

The request action argument explicitly tells the server what to do with the the message. To request a specific action from the server explicitly, the SREP command MUST include the request action argument. To not request a specific action, the SREP command MUST NOT include the request action argument; in this case, the server MUST decide the course of action. The client MAY specify either one of the following actions:

- The KEYWORD the client requests that only keyword(s) should be added to the message. The server MUST add the appropriate keyword.
- The RELOCATE the client requests that the message should be relocated. The server MUST relocate the message by copying the message to the destination mailbox removing the original as if another connected client requested this action.
- The DELETE the client requests that the message should be deleted. The server MUST delete the message as if another client performed this action.

The server MUST ignore the destination mailbox in case it is nil, or, the request action is KEYWORD or DELETE. It is assumed that the server is pre-configured with the location where user's spam messages are stored. If the server is not configured with such information and the destination mailbox in a RELOCATE action is nil, or, destination mailbox in a RELOCATE action is otherwise inaccessible to the user (does not exist, insufficient permission, etc) the the server MUST reject the request (see BAD response in Section 3.6). NOTE: While the DELETE action does not seem appropriate in case the directive argument is CLEAR, it is permitted. The formal syntax of the request action argument is defined in Section 3.7.

3.6. Responses and Results

The SREP command MAY result in system flag changes, keyword changes, message relocation, message removal, or a combination of these.

The result of the command MUST be either OK, NO or BAD:

- The OK result MUST be returned only in case the server parsed and completed the command successfully.
- The NO result MUST be returned only in case the server parsed the command successfully, but there is a problem with the referenced message(s) that prevents the server from completing the requested actions, such as one or more messages do not exist on the server, one or more messages are not properly authorized by URLAUTH, etc.
- The BAD result MUST be returned only in case the server cannot parse the command, or a configuration error is preventing the server from completing the requested actions.

When the result is OK, the response to a SET directive MUST be either KEYWORD, RELOCATE, RELOCATED, DELETE, or DELETED.

When the result is OK, the response to a CLEAR directive MUST be either KEYWORD, RELOCATE, or RELOCATED.

The server responses are:

- The KEYWORD response occurs in case this specific action has been explicitly requested by the client, or, in case the server decided that only the keywords should be updated either because it does not wish to give any recommendation to the client (RELOCATE or DELETE), or, because it does not have sufficient information either internally, or, from the spam aggregator service that it is configured to use.
- The RELOCATE response occurs in case the server decided that the message should be relocated, however leaves this action to the client. The client MAY decide what to do with the message.
- The RELOCATED response occurs in case this specific action has been explicitly requested by the client, or, in case the server decided that the message should be relocated and it performed relocation of the message to the appropriate location before the response was sent. The server MUST relocate the message by copying the message to the appropriate location and removing the original as if another connected client requested this action.
- The DELETE response occurs in case the server decided that the message should be deleted, however leaves this action to the client. The client MAY decide what to do with the message.
- The DELETED response occurs in case this specific action has been explicitly requested by the client, or, in case the server decided that the message should be deleted, and performed deletion of the message before the response was sent. The server MUST delete the message as if another client performed this action.

The KEYWORD, RELOCATE and DELETE responses MUST include the list of flags/keywords that have been added or removed. Added keywords MUST be prefixed with a plus sign ('+'), while removed keywords MUST be prefixed with a minus sign ('-'). The RELOCATED and DELETED responses MUST NOT include keywords. The formal syntax of the actions is defined in Section 3.7.

3.7. Formal Syntax

This document extends the formal syntax defined in [IMAP4] using the Augmented Backus-Naur Form (ABNF) notation specified in [ABNF]. Note: all string literals are case insensitive.

```
   srep-command        = "SREP" SP directive *1(SP abuse-type) SP \
                         reference *1(SP part-id-list) \
                         *1(SP request-action)
   directive           = "SET" / "CLEAR" / directive-ext
   directive-ext       = atom
                         ; It is not required that new directives
                         ; begin with "X-", see [XDASH]
   abuse-type          = "AT" SP abuse-type-id
   abuse-type-id       = 1*DIGIT
                         ; no leading zeroes or signs
                         ; New abuse types MUST be registered with
                         ; IANA as standard or standards-track
   reference           = reference-type SP reference-value
   reference-type      = "UID" / "SEQ" / "URLAUTH" / reference-type-ext
   reference-type-ext  = atom
                         ; It is not required that new reference types
                         ; begin with "X-", see [XDASH]
   reference-value     = uniqueid /           ; see [IMAP]
                         sequence-set /       ; see [IMAP]
                         authorized-url /     ; see [URLAUTH]
                         reference-value-ext
   authorized-url      = authimapurlfull /    ; see [URLAUTH]
                         authimapurlrump      ; see [URLAUTH]
   reference-value-ext = atom
                         ; New reference values MUST correspond to
                         ; reference-type-ext
   part-id-list        = "(" part-id *(SP part-id) ")"
   part-id             = header-id / body-id
   header-id           = "header." header-fld-name
                         ; see header-fld-name in [IMAP]
   body-id             = "body" *("." 1*DIGIT)
                         ; no leading zeroes or signs in numberic part
   request-action      = "DO" SP req-action *1(SP destination-box)
   req-action          = "KEYWORD" /
                         "RELOCATE" /
                         "DELETE"
   destination-box     = mailbox / nil
   resp-text-code      = resp-spam-actions    ; responses specific to
                                              ; this command, extending
                                              ; existing resp-text-code
                                              ; defined in [IMAP]
   resp-spam-actions   = "KEYWORD" flag-list / ; see flag-list in [IMAP]
                         "RELOCATE" flag-list /; see flag-list in [IMAP]
                         "RELOCATED" /
                         "DELETE" flag-list /  ; see flag-list in [IMAP]
                         "DELETED"
```

3.8. Examples to report spam

Report single message as spam; no identified parts; server only flags the message and hints that it should be moved:

```
C: Z020 SREP SET SEQ 10
S: Z020 OK [RELOCATE +$OMAEVVM10-spam-user-identified] SREP
Completed.
```

Report single message as spam; header and body identified; server adds the appropriate flags and hints that it should be deleted:

```
C: Z040 SREP SET SEQ 9 (header.from body.2)
S: Z040 OK [DELETE (+$OMAEVVM10-spam-user-identified-field.from
+$OMAEVVM10-spam-user-identified-body.2)] SREP Completed.
```

Report single message as spam; no identified parts; server moves the message (may clear/set flags too, but that is irrelevant because the client will need to reconcile anyway).

```
C: Z060 SREP SET SEQ 8
S: Z060 OK [RELOCATED] SREP Completed.
```

Report single message as spam; no identified parts; server deletes the message.

```
C: Z080 SREP SET SEQ 6
S: Z080 OK [DELETED] SREP Completed.
```

Report single message as spam; no identified parts; client requests explicitly to delete the message, server deletes the message as the client requested.

```
C: Z100 SREP SET SEQ 4 DO DELETE NIL
S: Z100 OK [DELETED] SREP Completed.
```

3.9. Examples to report messages as no longer spam

Report single message as no longer spam; no identified parts; server only clears the appropriate flags from the message.

```
C: Z020 SREP CLEAR SEQ 10
S: A020 OK [KEYWORD -$OMAEVVM10-spam-user-identified] SREP
Completed.
```

Report single message as no longer spam; earlier the header and body were identified as spam; the server clears the appropriate flags.

```
C: Z040 SREP CLEAR SEQ 9
S: A040 OK [KEYWORD (-$OMAEVVM10-spam-user-identified-field.from
-$OMAEVVM10-spam-user-identified-body.2)] SREP Completed.
```

Report single message as no longer spam; no identified parts; server moves the message (may set/clear flags, too but that is irrelevant because the client will need to reconcile anyway).

```
C: Z060 SREP CLEAR SEQ 8
S: A060 OK [RELOCATED] SREP Completed.
```

3.10. Example flows

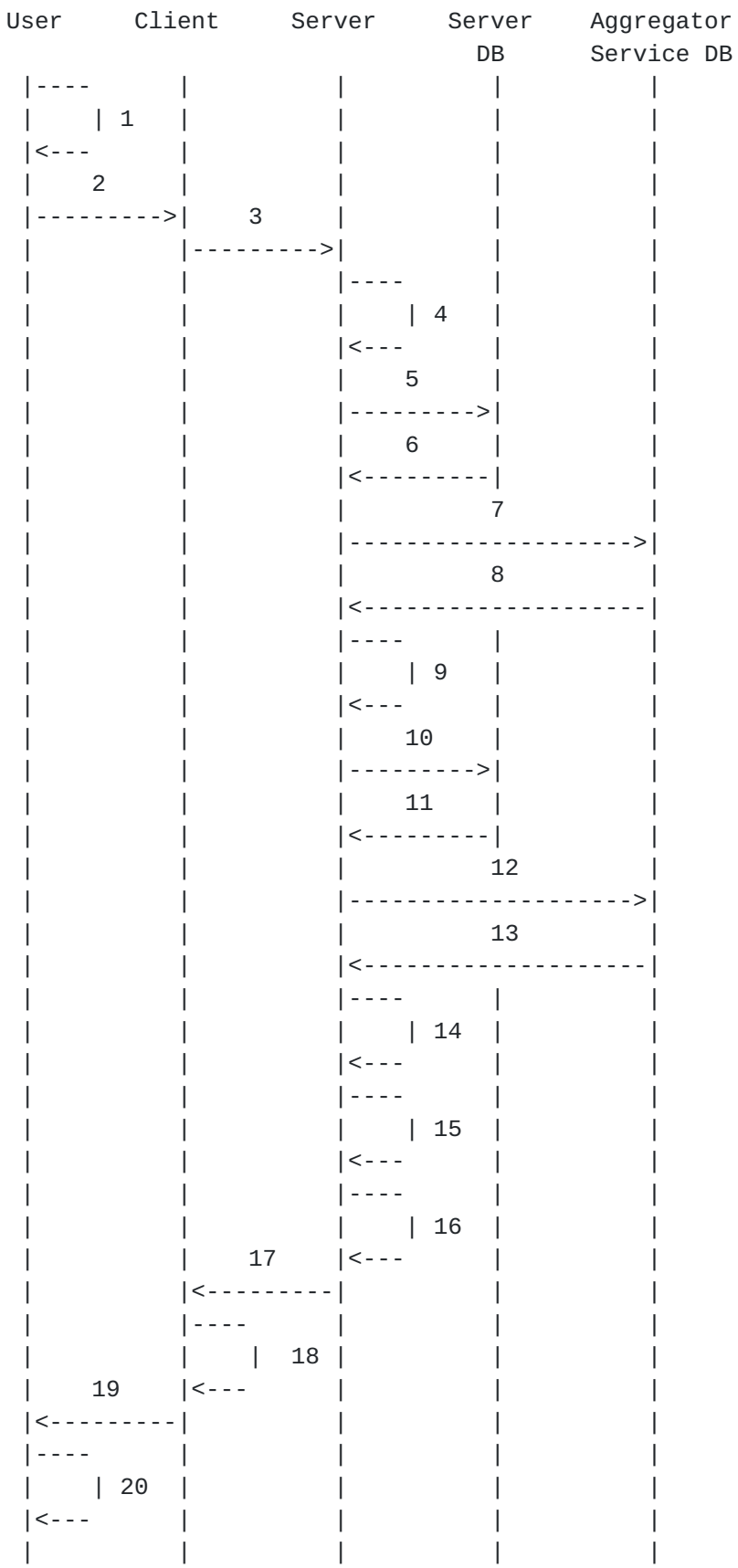
The new command specified in this document allows a client to save significant bandwidth by sending only reference(s) instead of full-blown spam reports that most if the time include the entire message. By doing so, the responsibility of recording metadata and handling the message designated as spam has been shifted to the server side. It is not the purpose of IMAP servers to deal with various aspects of spam reporting, such as creating and storing metadata, making the metadata available when new messages arrive, etc. IMAP servers should take advantage of an aggregator service and perform the exchanges related to spam reporting in the background, delegating the work to the aggregator service. Spam aggregator services are expected to collect and store metadata in very large volumes, evaluate the stored metadata and support queries to decide whether an incoming message is a spam or not. Open Mobile Alliance (OMA) specified the [OMA-SPAMREP] enabler release that supports deploying such aggregator services.

The flows in the following sub-sections illustrate informative examples for various scenarios that may be triggered by the SREP command defined in Section 3.

3.10.1. The server simply flags a message

1. The user finds a voicemail that he/she deems to be spam.
2. He invokes the appropriate functions on the client to report the message as spam.
3. The client reports the spam using the appropriate command to the server: SREP SET SEQ 10
4. The server prepares the message referenced by the command for inquiry.
5. The server queries its internal database for precedence.
6. The server gets a 'not found' response from the internal database.
7. The server queries an external database for precedence, such as an aggregator service based on [OMA-SPAMREP].
8. The server gets a 'not found' response from the external database as well.
9. No records of the message are found; the server prepares the message to be recorded for future reference.
10. The server reports the message as spam to its internal database.
11. The server gets an 'ok' response from the internal database.

12. The server reports the message as spam to external database, such as an aggregator service based on [OMA-SPAMREP].
13. The server gets an 'ok' response from the external database.
14. The server checks the user's preferences and finds no guidance about handing the spam, so
15. ... it checks the service provider policies - and yet again, finds no instructions.
16. In the end, lacking any sort of guidance, the end the server stores a keyword for the message, and
17. ... informs that client about that using the appropriate response: OK [KEYWORD +$OMAEVVM10-spam-user-identified] Completed.
18. The client updates its representation of the voicemail repository and
19. ... the message turns red on the user interface.
20. The user cheers.


User
Client
Server
Server DB
Aggregator Service DB
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20

4. Acknowledgements

The author acknowledges and appreciates the work and comments from Josh Soref, Gaelle Martin-Cocher, Suresh Chitturi, Clara Severino and Christophe Le Thierry D'Ennequin.

5. IANA Considerations

This document constitutes registration of the SREP capability in the imap4-capabilities registry.

SREP command directives are registered by publishing a standards track or IESG-approved experimental RFC. The registry is currently located at: http://www.iana.org/assignments/spam-directive-registry

SREP command abuse types are registered by publishing a standards track or IESG-approved experimental RFC. The registry is currently located at: http://www.iana.org/assignments/spam-abuse-type-registry

SREP command reference types are registered by publishing a standards track or IESG-approved experimental RFC. The registry is currently located at:
http://www.iana.org/assignments/spam-reference-type-registry

All registries are case insensitive.

This document constitutes the following registrations with IANA:

IMAP SREP Directive Registry

Directive	Reference
SET	[this document]
CLEAR	[this document]

IMAP SREP Abuse Type Registry

Abuse Type	Reference
1 - Phishing	[this document]
2 - Malware	[this document]

IMAP SREP Reference Type Registry

Reference Type	Reference
UID	[this document]
SEQ	[this document]
URLAUTH	[this document]

Note to RFC Editor: replace "[this document]" with the RFC number before publication.

6. Security Considerations

When an aggregator service is actively involved in a deployment, the service provider MUST ensure that:

- a mutual trust relation is in place between the IMAP server and the aggregator service, and,
- the aggregator service does not leak any information.

See additional security considerations in [IMAP4] and [URLAUTH], respectively.

7. References

7.1. Normative References

[ABNF] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 5234, January 2008.

[IMAP4] Crispin, M., "INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1", RFC 3501, March 2003.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[URLAUTH] Crispin, M., "Internet Message Access Protocol (IMAP) - URLAUTH Extension", RFC 4467, May 2006.

[XDASH] Saint-Andre, P., Crocker, D., and M. Nottingham, "Deprecating Use of the "X-" Prefix in Application Protocols", October 2011.

7.2. Informative References

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

[OMA-SPAMREP] Open Mobile Alliance, "Mobile Spam Reporting 1.0, OMA-ERP-SpamRep-V1_0".

[REPORT] Kucherawy, M., "The Multipart/Report Content Type for the Reporting of Mail System Administrative Messages", RFC 3462, January 2012.

Author's Address

Zoltan Ordogh
Research In Motion Limited
1875 Buckhorn Gate
Mississauga, Ontario L4W 5P1
Canada

Phone: +19056294746x15674
Fax: +12892615950
EMail: zordogh@rim.com