3GPP TSG SA WG3 Security — S3#31 **S3-030694**
18 - 21 November 2003
Munich, Germany

MMS TF Doc 263/03

MMS Security Considerations

Version 1.0.0

Document Source:	**Stefan Andersson, Engineering Support MMS** **(anderssons@malmo.mail.telia.com)**	
Document Creation Date:	**2003-11-04**	
Document Status:	**For Approval**	**X**
	For Information	
Associated Knowledge Basis:		
Circulation Restricted[1]:	**GSM Association**	
	Members	**X**
	Associate Members	**X**

Document History:

0	24/09/03 (MMS TF #14)

[1] **Restricted – Confidential Information**

Access to and distribution of this document is restricted to the persons listed under the heading Circulation Restricted. This document is confidential to the Association and is subject to copyright protection. This document is to be used only for the purposes for which it has been supplied and information contained in it must not be disclosed or in any other way made available, in whole or in part, to persons other than those listed under Security Restrictions without the prior written approval of the Association. The GSM MoU Association ("Association") makes no representation, warranty or undertaking (express or implied) with respect to and does not accept any responsibility for, and hereby disclaims liability for the accuracy or completeness or timeliness of the information contained in this document. The information contained in this document may be subject to change without prior notice.

1	20/10/03 draft
2	04/11/03 Approved

Contents

1 Introduction

1.1 Background

The task *MMS Security and Fraud Protection (including Spam)* is Work Area 41.2 of the MMS Task Force (cf. Doc MMS TF 005/03 in its latest revision).

The purpose of the task is to study the security related issues of MMS and to produce input to 3GPP SA3, GSMA Security group and OMA that motivates work on countermeasures. It is also a part of the task to suggest countermeasures and security requirements where applicable. The detailed scope is defined in [6].

Security requirements in the current stage 1 document [4] are the baseline for the current security mechanisms in MMS. These requirements are recapitulated below:

> "*The user shall be able to use and access MM in a secure manner. It shall be possible for the contents of MMs to be read only by the intended recipient(s). A Recipient shall be informed of the reliability of the sender in case the sender has authorized his identity to be transmitted.*
>
> *The integrity of MMs during transit shall be assured to the extent of the network capabilities.*
>
> *The MMS shall be intrinsically resistant to attempts of malicious or fraudulent use.*
>
> *The " Security Threats and Requirements" in 22.133 shall not be compromised*"

Section 5.1.29 in the 3GPP2 stage 1 document [1] has some additional requirements:

> "*The MMS shall have the ability to authenticate the user regardless of access technology*
>
> *The MMS shall support data transport in a secure manner between the user and MMS*
>
> *The MMS authentication scheme shall use access specific information.*"

These requirements will be used later in this document as a part of the analysis of threats and countermeasures.

1.2 References

[1] Multimedia Messaging Services, Stage 1 Requirements. 3GPP2 S.R0064-0 V 1.0

[2] Security in MMS standardization, MMS TF Doc 36/03

[3] Proposed minimum handset security requirements for MMS v2.0, MMS TF Doc 113/03

[4] Multimedia Messaging Service, Stage 1. 3GPP TS 22.140

[5] Multimedia Messaging Service, functional description, stage 2. 3GPP TS 23.140 V6.2.0

[6] Work plan for Work Area 41.2: MMS Security, MMS TF Doc 142/03

[7] RFC 2595 “Using TLS with IMAP, POP3 and ACAP”

[8] RFC 2487 “SMTP service extensions for secure SMTP over TLS”

[9] Open Mobile Alliance; OMA-MMS-CTR-v1_1, Multimedia Messaging Service, Client Transactions, Version 1.1, URL: http://www.openmobilealliance.org/

[10] Open Mobile Alliance; WAP Transport Layer End-To-End Security, WAP-187-TransportE2Esec, URL: http://www.openmobilealliance.org/

[11] WAP Push Security Concerns, Jagjeet Sondh, Vodafone Group Research and Development, Lung Wan Vodafone UK Core Network Development

[12] XML-Signature syntax and processing, W3C Recommendation 12 February 2002, URL: www.w3.org/TR/2002/REC-xmldsig-core/

[13] XML-Encryption syntax and processing, W3C Recommendation 10 December 2002, URL: www.w3.org/TR/2002/REC-xmlenc-core/

[14] “Unsolicited bulk email: Definitions and problems”. Paul Hoffman. Internet mail consortium report: UBE-DEF IMCR-004, October 5, 1997

[15] “Unsolicited bulk email: Mechanisms for control”. Paul Hoffman, Dave Crocker. Internet mail consortium report: UBE-SOL IMCR-008, May 4, 1998

[16] Open Mobile Alliance; WAP-182-ProvArch-20010314, Provisioning Architecture Overview, URL: http://www.openmobilealliance.org/

[17] Open Mobile Alliance; WAP-182-ProvBot-20010314, Provisioning Bootstrap, URL: http://www.openmobilealliance.org/

[18] RFC 2806, ”URLs for telephone calls”

[19] Response to LS from CPWP on MMS read-reply reports, MMS TF 104/03

[20] LS to MMS TF on MMS read-reply reports, MMS TF 181/03

[21] RFC 2368, ”The mailto URL scheme”

[22] RFC 2505 “Anti-Spam Recommendations for SMTP MTAs”

[23] Network domain security; IP network layer security. 3GPP TS 33.210

[24] Network domain security; MAP application layer security. 3GPP TS 33.200

[25] RFC 2535 “Domain name system security extensions”

[26] RFC 2845 “Secret key transaction authentication for DNS”

1.3 Definitions and abbreviations

For the purpose of this document the following definitions and abbreviations apply:

CDR	Charging Data Record
DoS	Denial of Service
IPMM	IP Multimedia
ISIM	IP Multimedia Services Identity Module
MAPSec	MAP Security
MM1	Interface between the MMS UA and the MMSC
MM3	Interface between the MMSC and external systems, e.g. e-mail systems
MM4	Interface between MMSCs
MM5	Interface between the MMSC and the HLR
MM7	Interface between the MMSC and VAS applications
MM8	Interface between the MMSC and the billing system
MMS	Multimedia messaging service
MMS broker	A proxy MMSC shared between several operators
MMSC	MMS center, MMS Relay/Server
OTA	Over The Air
PPG	Push Proxy Gateway
TLS	Transport Layer Security
SEG	Security Gateway
SIP	Session Initiation Protocol
SIR	Service Initiation Request
SL	Service loading
SMIL	Synchronized Multimedia Integration Language
SMTP	Simple Mail Transfer Protocol
SSL	Secure Socket Layer
UA	User Agent
VAS	Value Added Service
WAP GW	WAP gateway
WTLS	Wireless Transport Layer Security

2 MMS System architecture

2.1 System overview

Figure 1 below describes the high-level system architecture of the Multimedia messaging, MMS, service environment. The architecture is fetched from the 3GPP2 stage 1 document [1] since this contains a superset of the 3GPP architecture. The addition in the 3GPP2 document versus the 3GPP stage 1 specification is the possibility to let a third party service provider run the MMS. This scenario is however not further dealt with in this report as no such implementations exist.

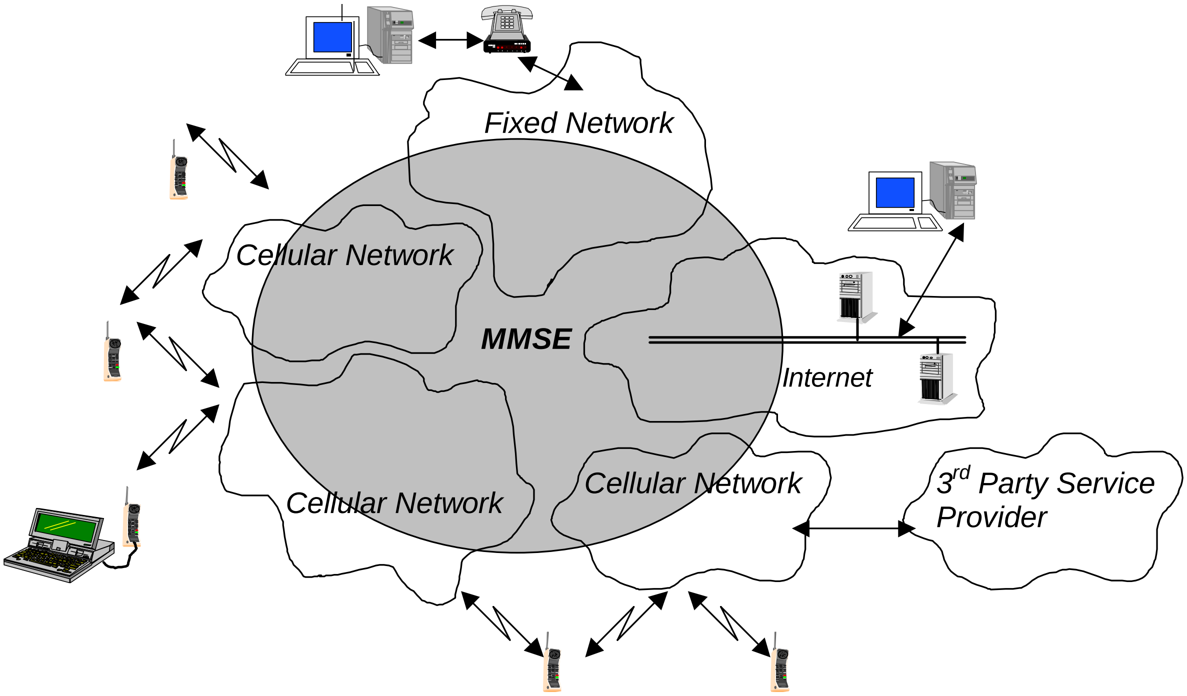

Figure 1 MMS Architecture

From the architecture we derive the more detailed scenarios described in subsequent sections of this document.

It is assumed that the reader is somewhat familiar with the MMS architecture and protocols.

2.2 Scenario 1

In the first scenario two MMS UAs that reside on different cellular networks communicate. The communication between MMS end users roughly uses the following path:

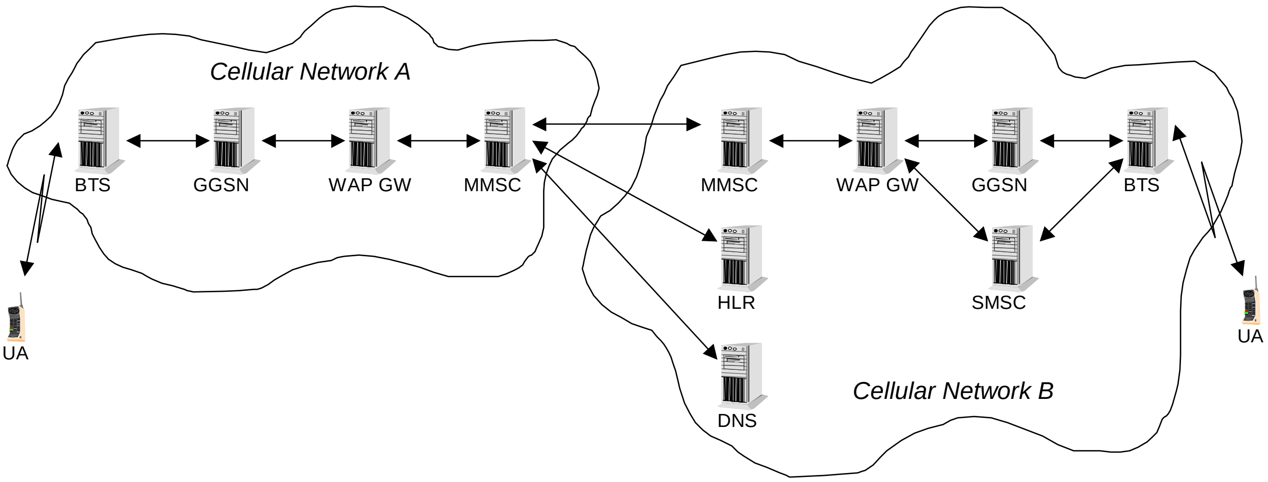

Figure 2 Scenario 1

When using the WAP 2.0 protocol stack the WAP gateway, GW, may be omitted from the communication path. In this case the communication between the UA and the MMSC will be http-based end to end.

However it can be expected that, even with WAP 2.0 terminals, most operators will deploy a WAP GW 2.0 (that is a dual stack proxy being able to relay both WSP and WP-HTTP onto normal HTTP).

In the figure, between BTS and GGSN, the intermediate nodes in the chain

BTS – BSC – SGSN – GGSN

are not shown in order not to overload the figure.

The originator UA (left) may be roaming. In this case the BTS, BSC and SGSN (on the left) belong to the visited PLMN whereas GGSN, WAP GW and MMSC belong to the HPLMN. In this case, between SGSN (vPLMN) and GGSN (HPLMN), typically GRX is used.

The recipient UA (right) may be roaming. In this case the BTS, BSC and SGSN (on the right) belong to the visited PLMN whereas GGSN, WAP GW, HLR, SMSC and MMSC belong to the HPLMN. In this case, between GGSN (HPLMN) and SGSN (vPLMN), typically GRX is used.

The communication between Originator MMSC and Recipient HLR can be performed via an intermediate signaling entity in the Originator PLMN:

oMMSC – intermediate signaling entity - rHLR

The communication between Originator MMSC and Recipient DNS typically is done via the home DNS:

oMMSC – oDNS - rDNS

We consider scenarios where the UAs reside on the same network as a special case of our first scenario and they will therefore not be described separately.

In scenario 1 we can identify the following existing security measures:

- The communication between UA[2] and SGSN is encrypted and integrity protected
- The interface between MMSCs that reside on different networks, MM4, is SMTP-based and uses typically GRX or similar networks or public IP with IPSec or 'nailed through connections' (leased line etc.)
- The communication between the MMSC and the HLR can be protected using MAP security
- DNS traffic may be protected using DNS security as specified in [25] and [26]
- The communication between the UA and the WAP server may be protected using WTLS. This may provide mutual authentication, integrity protection and confidentiality
- If WAP 2.0 is used the communication between the UA and MMSC may be protected using TLS. This may provide mutual authentication, integrity protection and confidentiality
- SSL or TLS may protect the communication between the WAP server and the MMSC. This may provide mutual authentication, integrity protection and confidentiality
- The communication for SMS (WAP push) is encrypted and integrity protected between BTS and UA

[2] More exactly: the mobile station. Here and in the following, a distinction within the terminal is not yet made.

In some configurations the communication between networks is protected on the IP layer as described in [23]. Here a new logical entity is introduced, the security gateway (SEG), the figure below.

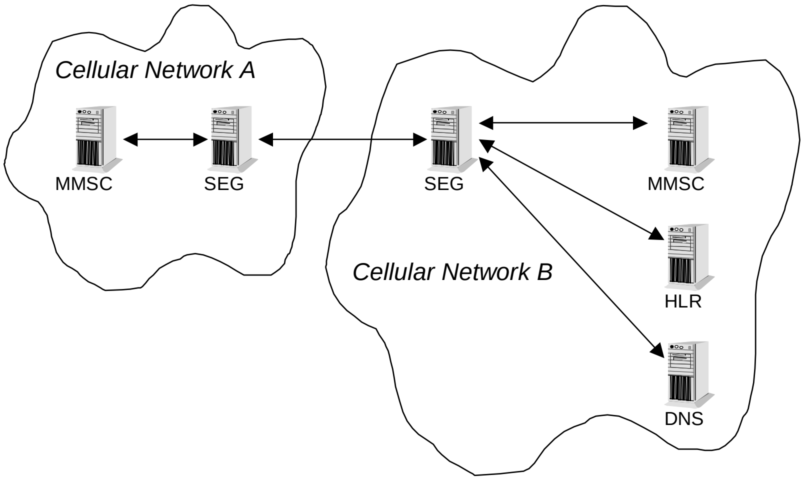

Figure 3 Security gateways

This approach may protect MM4 traffic, MM5 traffic as well as DNS transactions. Manual key management governed by roaming agreements is expected between SEGs.

This key management scheme can also be used in configurations without SEGs. One such scenario would be to use symmetric key security for DNS [26] based on keys exchanged as a part of roaming agreements between operators.

2.3 Scenario 2

In our next scenario one of the UAs is replaced with an e-mail client that resides on a fixed network. Here the communication path would be as follows:

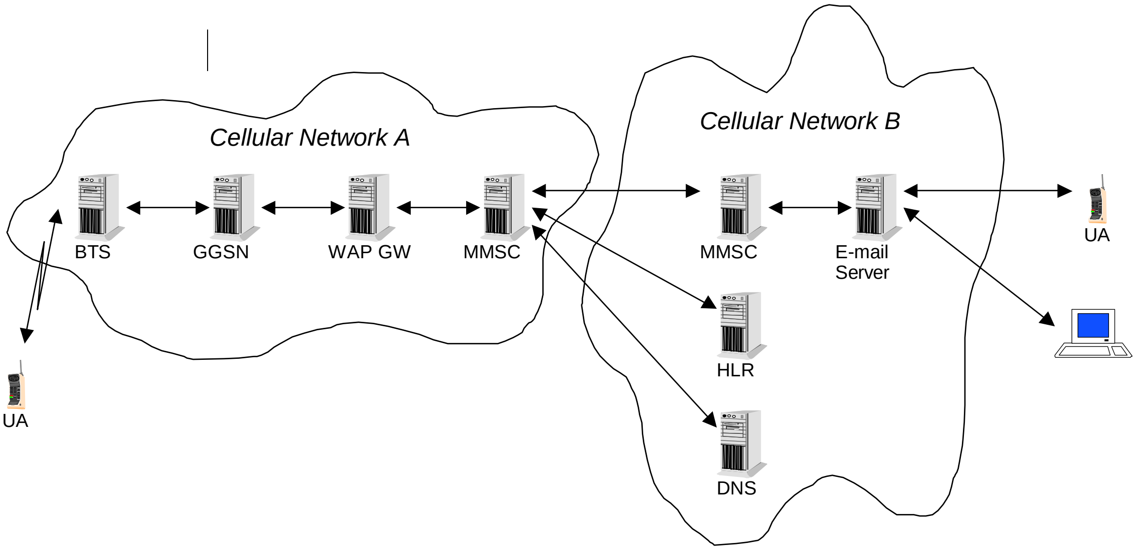

Figure 4 Scenario 2

The same explanations as for the previous figure apply.

In scenario 2 we can identify the following security measure in addition to the ones available in scenario 1:

- TLS/SSL may be used to protect the communication between the recipient UA (right) and the e-mail server as described in [7], [8]

If the e-mail server reside outside cellular network B the interface between it and the MMSC may be protected using the same mechanisms that are used on MM4, i.e. IPSec etc.

2.4 Scenario 3

In our next scenario, scenario 3, a MMSC is shared between several operators. The shared MMSC is defined as an MMS broker. The figure below elaborates on the architecture.

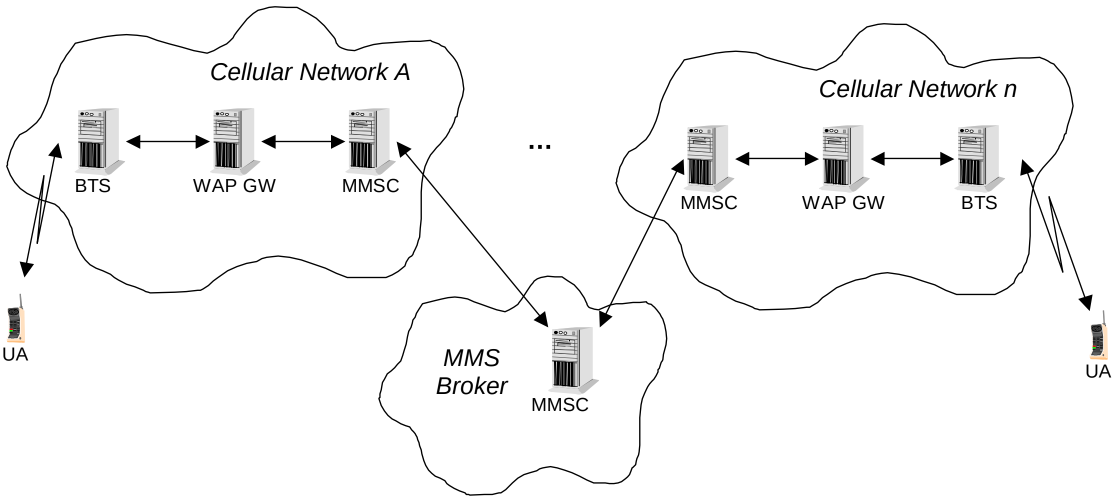

Figure 5 Scenario 3

From an architectural perspective this scenario is quite similar to the previous. The main difference is in the trust model, where this scenario assumes that the authentication is done by the radio access network. The architecture also implies that hop-by hop security must be used. In this scenario we can utilize the following security mechanisms:

- SSL or TLS may protect the communication between the MMSCs and the MMS broker. This may provide mutual authentication, integrity protection and confidentiality

- The interface between MMSCs and the MMS broker may be protected using IPSec. This may provide mutual authentication, integrity protection and confidentiality.

Key management between the MMS broker and the operators require special attention to create a balanced trust model.

Furthermore the MMS broker will be a weak link in the architecture since it will contain plaintext. This is quite similar to the situation in the WAP GW, which is also the part of the original WAP security architecture that has received most critical feedback.

2.5 Scenario 4

In our last scenario, scenario 4, a value added service provider is connected to the MMSC in cellular network A as described in the figure below. The VAS will communicate with the MMSC using SOAP transported over http. Apart from that the protocol has many similarities with the protocol between a UA and the MMSC. In other words the VAS has several similarities with a UA since it is capable of receiving, sending and forwarding MMs

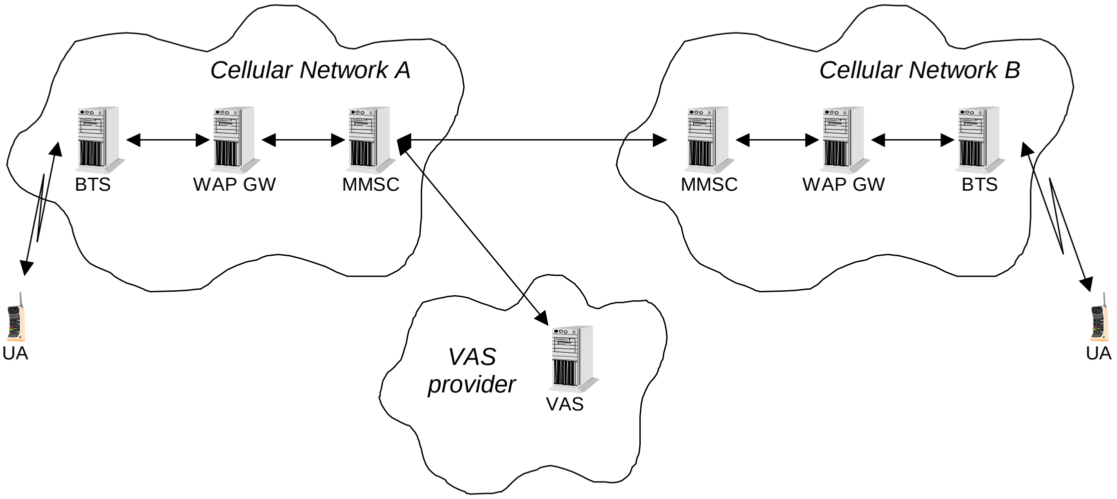

Figure 6 Scenario 4

Communication between the VAS and the MMSC can utilize the following security mechanisms:

- SSL or TLS may protect the communication between the MMSC and the VAS. This may provide mutual authentication, integrity protection and confidentiality
- The interface between MMSC and the VAS may be protected using IPSec. This may provide mutual authentication, integrity protection and confidentiality.
- SOAP security can be used to protect the messages between the MMSC and the VAS. This may provide mutual authentication, integrity protection and confidentiality.

For those not familiar with SOAP security it is based on XML-dsig[12] and XML-encrypt[13].

It should also be noted that the VAS may have the capability to recall messages and to generate CDRs. Thereby they pose a greater threat than a normal UA.

3 Threats and countermeasures on the communication path

3.1 Threat categories

To make the threat analysis more systematic we use following the attack definitions described in [2]:

- Protocol attack, an MM is submitted to/retrieved from the MMSC not adhering to the protocol as defined in the standard causing the MMSC to malfunction.

- Data attack, an MM is submitted to the MMSC not adhering to the data format as defined in the standard causing the MMSC or the receiving MMS UA to malfunction.

- Service attack, a MM is submitted/retrieved from to the MMSC adhering to all current standards but misusing the service (e.g., unsolicited and spam messages, service theft, identity theft, loss of confidentiality).

We will use these attack definitions when we analyze the interfaces on the communication path in the rest of this chapter.

3.2 Threats and countermeasures on MM1

3.2.1 Http/WSP

When we analyze the threats that arise from protocol and data attacks the important issue is to determine if such attacks are possible not to create an extensive list of detailed attack descriptions. It would be impossible to find all attacks and in general that approach would lead us into a never-ending spiral of attacks and countermeasures. Another reason for not choosing this approach is that many attacks will be implementation dependent and they will therefore not have general applicability. We would like to draw this even further by stating that it is enough that an attacker can modify the protocol since it is difficult or even impossible to build a MMSC/UA that can withstand all types of attacks on the signaling messages. Instead we should require integrity protection on MM1 if protocol or data attacks are possible.

Scenario 1 is as secure as the access technology security as long as the entire communication path is under operator control. Although there are some theoretical attacks on 2G security we can conclude that we in all practical cases have a secure system.

If on the other hand one of the communication interfaces between the GGSN, WAP GW or MMSC would be publicly accessible this would make us susceptible to protocol and data attacks. Natural countermeasures are IPSec on the interface between the GGSN and the WAP GW and SSL/TLS on the interface between the WAP GW and MMSC. One potential weakness with this approach is the fact that the communication is in plaintext in the GW. Therefore the physical protection of the WAP GW is quite important. WTLS can be used as an alternative to IPSec but again the communication will be in plaintext in the GW.

Co-locating the WAP GW and the MMSC can reduce this threat. In addition gateway navigation [10] can be utilized if the operator choose to have a dedicated GW co-located with the MMSC. As described in the figure below GW navigation enables network operator to redirect the communication to a dedicated GW.

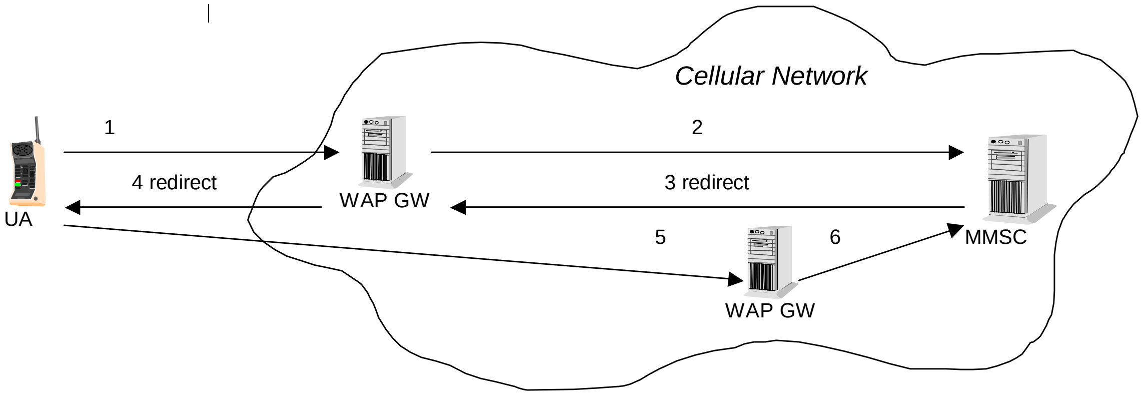

Figure 7 GW navigation flow of events

The approach described above may also be used in scenario 3, i.e. the MMSC broker scenario even though hop-by-hop security is the natural choice.

In a WAP 2.0 architecture these countermeasures would not be available since there may not be a WAP GW involved. Instead end to end TLS would be in use between the UA and the MMSC.

When using TLS the standards specify that the URI of the server is verified against information in the server certificate. This mechanism is designed to prevent rouge servers from masquerading as legitimate. In wireless clients with limited display capabilities it is even more important since the URI is generally not visible to the user. This mechanism could typically also be used to verify the address of the MMSC in the UA against its server certificate.

There may be a subtle threat to this mechanism that stems from the way the server certificate is verified. The verification will be done against a predefined root certificate on the client. These root certificates may be preconfigured on the UA or on the SIM/WIM and they may even be downloaded by the user. It is this last option that may introduce a threat. If the root of a rouge CA is introduced on the client this CA will be able to issue server certificates that circumvent the URI verification mechanism. The countermeasure would be not to use user downloaded roots for this purpose.

Yet another reason for treating the roots with care is the fact that they define the trust model and control which server a UA can communicate securely with. If these roots can be modified by the user it is he or she that is in control of the trust model not the operator.

If we look at the portions of the MM protocol that can be protected by TLS we find that TLS or SSL can protect the following PDUs:

- M-Send.req
- M-Send.conf
- M-Retrieve.conf
- M-NotifyResp.ind
- M-Acknoledge.ind
- M-Read-Rec.ind
- M-Forward.req
- M-Forward.conf

A triggering mechanism can be implemented as a configurable parameter in the client or some other means integrated in the protocol. The latter possibility can be compared to how IPSec is triggered from SIP in IPMM.

TLS can also be applied to the retrieval of a MM, triggered by the content location URI in the M-Notification.ind. Here the standards specify the use of the https URI scheme to trigger TLS.

3.2.2 Push

As described in the previous section several parts of the MM protocol can be protected by TLS. Unfortunately this is not as straightforward for the PDUs carried over unconfirmed push:

- M-Notification.ind
- M-Delivery.ind
- M-Read-Orig.ind

In the WAP push architecture, Figure 8 below, push messages are relayed through a push proxy gateway, PPG. Service providers and push initiators, access the push proxy gateway using the push access protocol. For MMS the MMSC would act as a push initiator. If the client doesn't have a WAP session the PPG sends a SMS containing the URI from which the client can download the message.

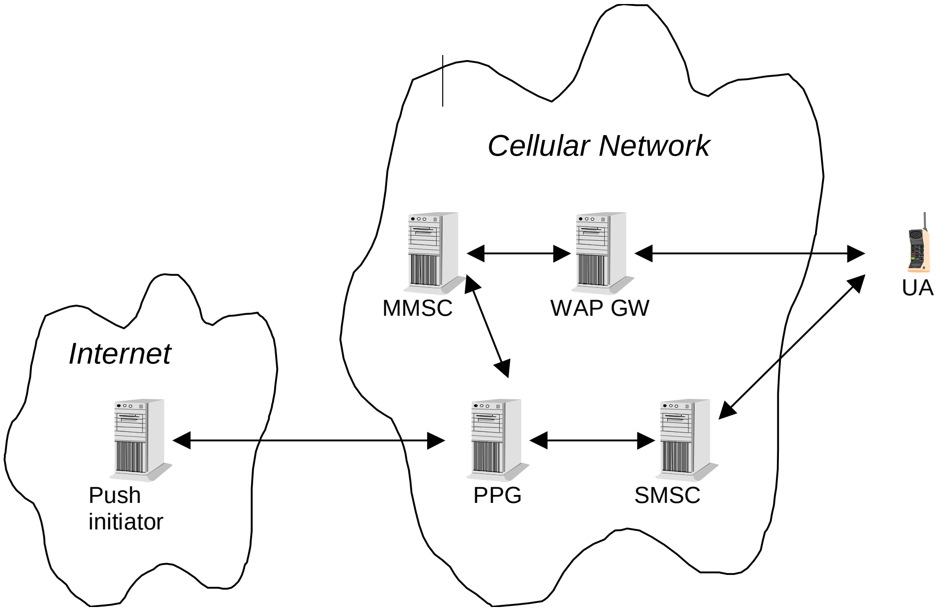

Figure 8 WAP push architecture

Two push mechanisms are defined in OMA:

- Service loading, SL. SMS with URI.
- Service Initiation Request, SIR. SMS with URI, GW address and other connection parameters.

Several OMA members have raised security concerns and proposed counter measures. The results in [11] are used as a base for much of the remainder of this section. With our notation the attacks related to push would be of the type service attacks.

It should be noted that several of the attacks related to push are generic and not specifically tied to MMS.

The following attacks are identified in [11]:

- With service loading it is possible for an attacker to force the client to download content from the Internet if the WAP GW allows that. If the GW only accepts addresses on the MMSC the request will be rejected and the terminal will prompt “failed” which will also cause inconvenience for the user.

- Using SIR it is possible for an attacker to make the client connect to another gateway instead of the one configured on the client. Furthermore since the SIR may contain a CSD phone number this can be used for immediate fraud against the user.

A first countermeasure would be to only allow SIR against a predefined set of GWs. Similar mechanisms can protect SL, here the PPG source address would be verified against a predefined set of allowed addresses. The proposed mechanism introduces a white list that holds the details of allowed PPGs. This white list may be stored on the terminal or on the SIM. Solving the generic push security problem through the introduction of a white list mechanism has some deployment difficulties. New fields on the SIM must be specified and introduced or the white lists must be provisioned to the UAs.

There is a simpler solution in the case of MMS based on the configuration information that is already available in the UAs. The idea is to only download messages located at the terminals provisioned MMS server. All control messages (e.g. acknowledgement, reject, etc.) and originate MMS messages are sent from the terminal to a specific URL that are stored in the terminal. If the MMS server stores all messages on locations that start with the correct hostname prefix compared to the URL address that is stored in the terminal, then the terminal could easily check if the MMS message is stored at the MMS server and not somewhere else.

Unfortunately neither of these mechanisms is resistant against SMS spoofing. Therefore work is ongoing in OMA to define more robust push security mechanisms. Currently the push security requirements are being drafted in OMA. At this point it is difficult to guess what security mechanisms will be developed, but it can be concluded that it will take some time before the specifications are ready.

3.2.3 User identification

MM1 assumes that an underlying authentication scheme is used and that the identity information can be retrieved by the MMSC through RADIUS. In this report it is assumed that the authentication scheme is SIM based. In other words it is assumed that the user id password authentication towards the MMS APN is not used for MMS user identification and billing. The use of password based schemes would clearly impose a security threat on the MMS system.

In scenarios where the entire MMS is under operator control this is definitely the most efficient and secure approach. Assuming that the underlying authentication mechanism is SIM or USIM based.

This is not true in scenarios where the MMS is not under operator control or where SIM/USIM based authentication is not available, e.g. scenario 2. Here mechanisms in the MMS layer would be preferable, again assuming that they are SIM, USIM or ISIM based. Furthermore it is not obvious that the trust model in scenario 2 is such that the current approach is preferable.

3.3 Threats and countermeasures on MM2

MM2 is not specified by the current MMS release. Therefore the analysis of this interface is very brief. If this interface is publicly accessible, the system is threatened by the same type of attacks as the ones that are possible on MM1. As consequence authentication and integrity protection would be required.

3.4 Threats and countermeasures on MM3

MM3 enables interworking with existing e-mail servers, i.e. scenario 2 described earlier.

Since the MMS protocols are not extended over MM3 protocol and data attacks seem unfeasible on that interface. Unfortunately this is not true for service attacks in general and spam in particular. The main reason for this is that user authentication and charging mechanisms between the e-mail server and the UAs don't match the security level of the mechanisms available in a cellular network. The authentication mechanism is a good example of this mismatch. In a cellular network smart card based mechanisms are used to authenticate the users whereas password based schemes generally used to authenticate the users towards the e-mail server.

The lack of robust charging and authentication mechanisms opens the possibility to introduce spam in MMS through MM3. It should also be noted that spam introduced on MM3 would not only affect the operator connected to MM3 but it can spread through MM4 to other operators as well.

Current best practice is not to allow inbound MMs on MM3. This eliminates the spam threat. If this is to be changed there is definitely a need to enhance the existing MM3 security mechanisms.

3.5 Threats and countermeasures on MM4

MM4 is susceptible to protocol, data and service attacks and it therefore requires relevant security mechanisms. As described earlier there are several mechanisms that can be used, i.e. IPSec etc. Since these mechanism assume direct routing without intermediate MMSCs it is important to architect the networks accordingly

Although MM4 may be the most dangerous interface in MMS it is also the one that is most straightforward to protect using existing technology, IPSec, if we use direct routing between the MMSCs. If SMTP proxies are allowed the use of end to end IPSec is impossible. This is the case in Scenario 3 where hop-by-hop security must be applied. In configurations where the MMS broker can be trusted the hop-by-hop approach is a viable option. Nevertheless end to end security solutions on SMTP level are worth studying in the future.

A final remark on MM4 is that the MMSCs should be configured to only accept SMTP mail messages to reduce the potential threats from the rest of the SMTP message set.

3.6 Threats and countermeasures on MM5

The HLR interface, MM5, may rely on MAP security, MAPsec, as defined in [24]. MAPsec will provide:

1. Integrity protection
2. Origin authentication
3. Replay protection
4. Confidentiality may also be provided as an optional feature.

The first three parts constitute protection mode 1, which is also, the option that should be used to protect the MM5 traffic. Fallback to unprotected mode is deprecated since it may introduce a security risk.

Security association establishment and key agreement between operators will be handled manually as a part of normal roaming agreements.

3.7 Threats and countermeasures on MM6

MM6 is another interface that is not specified by the current MMS release. If this interface is publicly accessible, the operator may loose valuable information about his customers. Furthermore the user privacy aspects of the information should also be taken into careful consideration. As consequence authentication, integrity protection and confidentiality would be required.

3.8 Threats and countermeasures on MM7

On an abstract level MM7 is similar to MM1, where the VAS takes on the role of the UA. In other words the VAS is capable of receiving, sending and forwarding MMs. The two major differences in a security perspective are:

- The VAS communicates with the MMSC over open IP networks
- The VAS is not expected to hold a SIM or USIM for authentication.

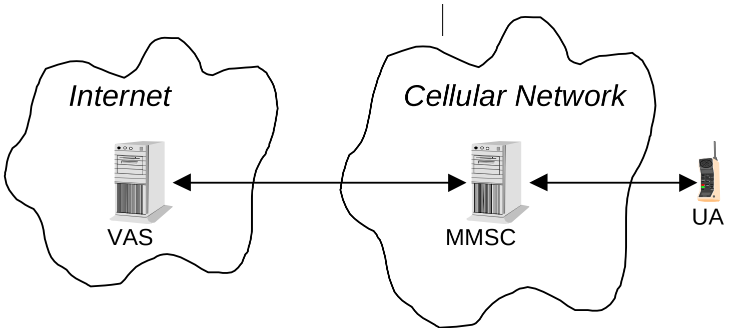

Figure 9 MM7 VAS MMSC interaction

The MMS functional description recommends the use of Http authentication and TLS. In other words it is recommended that we use TLS for integrity protection, confidentiality and server authentication. The VAS authentication will be performed using Http authentication.

A possible weakness with this approach stems form the use of Http authentication. It would be preferable to utilize TLS for client authentication but that would introduce a key management problem. The operator would have to issue TLS client certificates to the VAS.

IPSec could also be used to protect MM7 the same way as it can be used to protect MM4. Since IPSec key management generally is considered as cumbersome this approach is most suited for scenarios with rather static operator-VAS relations.

Since MM7 may be SOAP based there is a third alternative, SOAP security. From a security and key management perspective SOAP security and TLS are quite similar since they are both PKI based. The main difference is that SOAP security is a quite new technology when compared to TLS, which is widely deployed.

Regardless of the choice of security mechanism we must assume close co-operation between the VAS and operator. Basically the operator must be able to trust the VAS. Much of the trust issues can be covered in the business agreement between the VAS and the operator.

If on the other hand the VAS can not be trusted the system will face the following threats:

- Charging abuse as defined in [5]

- The same type of fraud introduced by malicious UAs described in section 4.4.

A malicious VAS has many similarities with a malicious UA. We will elaborate further on malicious UAs later in this document and we will leave that discussion for now. Nevertheless a malicious VAS has pose some additional threats since it may abuse the charging mechanisms.

3.9 Threats and countermeasures on MM8

MM8 is the third interface that is not specified by the current MMS release. Even more so than in the case of MM8 the operator is at risk. If this interface is standardized and used the security aspects and the trust model must be carefully considered.

4 Threats and countermeasures in the UA

4.1 Malicious content

The malicious content attack defined in [2] is directly applicable to threats on the UA. The definition is as follows: a well-formatted MM is sent to the MMS but it contains malicious content that will cause harm on the UA.

Since MMS can be considered as both a client and a bearer we have two scenarios related to content rendering. The first is auto rendering of known content types. In this case the UA would automatically render (display, play, execute…) the content from the SMIL presentation. The second is unknown content types, which will not be automatically rendered; instead they may be stored on the UA to be invoked later by the user.

The first scenario is the potentially most dangerous one since it opens up for automatic attacks that the user can’t protect himself against.

Viruses are often defined as malicious software inserted into another application to attack the host and spread to other systems. MMS has the potential of becoming a channel for viruses since:

- MMS supports auto rendering
- MMS supports superdistribution
- MMS may at some point support more dangerous content types such as, script languages, Java etc.

A majority of the supported media types seem to pose no threat when auto rendered. The media types that fall into this category are:

- Text
- Speech
- Audio
- Syntethic audio
- Bitmap graphics
- Video
- Vector graphics

Depending on implementation some of these content types may be susceptible to buffer overflow attacks. Currently the majority of handsets are based on closed OS systems built on more or less proprietary hardware. To some extent this act as protection against buffer overflow attacks that try to execute a malicious application. But it does not protect against buffer overflow attacks targeted at disrupting the UA, e.g. DoS attacks. This is a threat that must be taken seriously since similar bugs have been observed in existing SMS implementations.

The presentation and synchronization formats SMIL and XHTML have some features that could be misused if not implemented correctly.

In SMIL the timing module can potentially be used to perform a denial of service attack against the user. The attack would involve a SMIL presentation with exceptionally long delays between the elements. A MMS UA can easily prevent this attack by always providing the user the possibility to jump to the next slide in the presentation or to cancel the ongoing SMIL presentation. This threat can be also be diminished by adding sanity checks to the timing elements of the SMIL presentation.

If the XHTML implementation allows either the: tel, vtel, mailto, smsto or mmsto URI scheme this can be used to commit fraud against the user if the user can be tricked to select one of these links. Things get even worse if the implementation allows automatic interaction with the address book in the UA. This would open the possibility for “I love you” type viruses.

A countermeasure would be to graphically indicate the purpose of a URI. That option seems to be difficult from a usability perspective. A better option is to follow the Java approach and clearly warn the user before initiating a chargeable event. The latter is also the option that is recommended in [18] and [21]. Furthermore [21] also recommends that the “From” address is not set by the URL, instead it should be provided by the native mail client.

For MMS, the terminal will to offer more control of the user interface than for other services. For example, the screen may be faked, and the user may be misled to accept actions of the terminal without realizing what he does. A generic countermeasure is to make the MMS client application visibly distinguishable from the SMIL presentation. This can be done by only allowing the MMS application to use a portion of the display, the rest would be reserved for system status information, softkeys etc. This would preferably be combined with mechanisms that always allow the user to stop the rendering of a MMS. Technically this can be achieved by giving the MMS application the right priority in the system, i.e. lower than the UI and system threads.

The best generic policy is to always warn the user of any potentially chargeable event triggered by a command embedded in MMS. Furthermore the user should have to confirm any such action before it is performed.

As new media types are added the risk of automatically rendering them must be considered. If MMS end up in the same situation as e-mail is today we face a situation where virus checkers, MMS filters, and firewalls must be introduced at great cost to protect the system.

It should be noted that automatic rendering of streaming content may be associated with some risk since it is similar to push as it automatically connects to a URL that may reference a malicious server.

In the second scenario MMS is merely used as a transport and the rendering client in the UA must handle the threat. This would for example be the case for Java applications distributed using MMS as a bearer. In this case the Java security framework and domain model protects the terminal against malicious Java applications.

4.2 Spam and DoS attacks

We derive our definition of MMS spam and the problems it may cause from [14]. The definition of MMS spam would be MMS messages that are sent to a group of recipients who have not requested it.

Spam is so dangerous because the majority of the costs related to the message are passed onto the recipient and the recipient operator. Several such costs can be identified. Here are some examples:

- Network traffic costs to the destination operator, e.g. bandwidth and MMSC storage capacity
- Time lost deleting unwanted messages
- Loss of revenue due to lowered MMS usage

The thing that makes e-mail spam so devastating is the fact that it is virtually free to send e-mail. This means that nearly all costs are shifted to the recipient. In the case of MMS this is not true as long as MMs are only allowed into the system on channels that support user authentication and charging. So one thing that currently protects the MMS systems from spam is the charging model.

Looking at the MMS reference architecture we find four potential entry points for spam, MM1, MM3, MM4 and MM7. As described earlier in this report some configurations of MM1 and MM3 may lack robust authentication and charging mechanisms. This is the kind of environment in which spam can flourish.

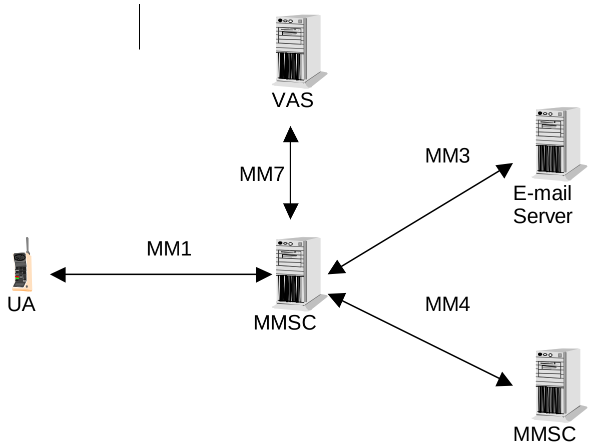

Figure 10 Potential spam entry points

If spam is introduced in the system there are some countermeasures that can be deployed [15]:

- Origin based filtering, at the MMSC or in the UA
- Message based filtering, at the MMSC or in the UA
- Originator accountability

Further countermeasures that can be introduced in the MMSCs are suggested in [22], here are some examples:

- Only allow traffic from authorized (authenticated) UAs or MMSCs
- Keep log information to make it possible to analyze attacks and identify the attackers
- Introduce control mechanisms to control the rate with which messages are sent and received

One countermeasure that can't directly be translated to MMS is the following. The idea is to add enough information in the "Received:" field to make it possible to trace the message path and to take action towards the spam originator.

It is assumed that none of these methods will be able to fully eliminate spam once it is in circulation. It should also be noted that originator accountability is a prerequisite for robust charging mechanisms.

Operators must consider the risk that other operators deploy a charging or authentication model that introduces SPAM. In other words if one operator opens a channel for spam, messages from this channel can be routed to other operators networks.

To prevent this operators may introduce spam filters on MM4. Another way of preventing this is to have a suitable model for operator-operator charging for MMS passed over MM4. The main measure is that the receiving operator must get some revenue from the sending operator, this way the receiver will not bear the entire cost of the messages.

It is also important to use secure MM7 connections (and MM3 in case that is used for VASPs). If authentication and integrity protection is not enabled an attacker can potentially get access to the MM7 interface. The attacker would then be able to send MO or MT charged messages and consequently launching spam attacks on a network.

The potential spam entry points described earlier are also the interfaces most likely to be subject to denial of service attacks. Another potential source of DoS attacks is the auto rendered malicious content described earlier.

Spam can also be used to launch DoS attacks on MMS level. If an attacker can get into the MMS infrastructure, for example through push, he can spam a large number of UAs with M-notification.ind messages so that they will try to connect to a single MMSC in order to overload it with Http Get requests.

In general the wireless world is inherently less resistant to DoS attacks due to bandwidth, memory and CPU limitations on the clients.

4.3 OTA configuration

OMA defines push based over the air, OTA, configuration mechanisms in [16][17]. The configuration data is a XML document that can contain settings for, the browser, e-mail, MMS etc. The OTA configuration can be performed over several bearers including, SMS, USSD, SIM and cell broadcast. The architecture is further described in Figure 11 below.

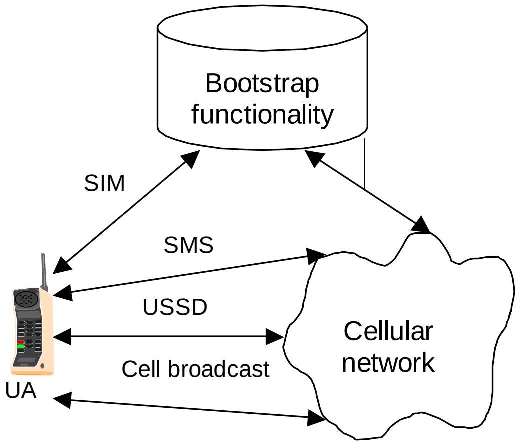

Figure 11 OTA configuration architecture

Adequate security mechanisms are important since OTA configuration easily can be used for malicious purposed. Two methods have been defined to meet the threats:

- Bootstrap security by means of a shared secret
- Bootstrap security by means of an out of band delivery of authentication information.

In the first method the shared secret is used to calculate an SHA-1 based HMAC which integrity protects and authenticates the configuration data. The HMAC value is carried as a parameter to the media type in the content type header. The integrity key can either be a value entered by the user, a value defined by the network or a combination of both.

As the name of the second method implies the MAC value is not carried in the OTA message. Instead the MAC is incorporated in a PIN which is delivered to the user by an out of band mechanism. This pin is entered by the user and compared to a value calculated by the UA.

In GSM the IMSI will be used as the network value for the shared secret.

If the security of the OTA configuration mechanism is insufficient it can be used to commit fraud against the users. A possible scenario is that someone creates a fake configuration message where the MMSC address and the access parameters are spoofed. Once the configuration is installed in the client the attacker could send a normal which would then make a CSD connection to a malicious MMSC. The MMSC could keep the connection open for monetary fraud or it could launch buffer overflow attacks on the UA.

The security level of these mechanisms can be questioned mainly due to limitations in the effective key length of the user entered PIN. This can be compared to the security level that could be achieved if for example the SIM would be used to carry the symmetric key.

Currently OTA configuration messages use a special UDH. It should be recommended that UAs limit the acceptance of OTA configuration messages to SMS with a dedicated UDH sent from a trusted entity. Especially a UA should reject UDHs originating from mobile terminals like GSM modems.

4.4 Malicious UAs

Focus in the previous chapters of this paper has been on communication security or threats against the UA. In this section we will elaborate on threats on the system that arise from malicious UAs.

Let’s start by assessing the difficulty to create a malicious UA. The first possibility is that the phone manufacturer implements a malicious UA. This clearly feasible from a technical perspective but it is definitely prevented by the business environment. If this should happen operators can simply decide not to do business with that manufacturer. This trust model is not very different from what is in place concerning the GSM protocols.

A second option is that then MS UA is implemented in Java. This is technically not possible using the standard MIDP 2.0 APIs since a Java MIDLet can’t access the generic push inbox. In devices that implement proprietary APIs for accessing the push inbox the situation may be very different.

Another way to guarantee that the Java untrusted domain couldn’t be used to implement a malicious UA is to integrate information in the protocol that can’t be accessed from Java. One such possibility is to include SIM based authentication in the MMS protocol.

It should also be noted that it seems impossible to prevent Java MIDlets from implementing a parallel MMS system based on SMS and http outside of operator control.

A third possibility for malicious UAs is on open OS terminals based on Symbian OS, Microsoft OS, Linux etc. Here a third party can implement a malicious MMS UA. In this case it is expected that all APIs in the handset are available to the third party provider. Therefore it will be difficult to prevent this type of MMS UA by relying on SIM based authentication.

Finally we have the most obvious implementation of a malicious UA, the MMS UA implemented on a PC. In this case the UA on the PC uses the terminal as a modem when communicating with the MMSC.

The conclusion is that it is clear that malicious UAs can appear in MMS systems already today.

Assuming that it is possible to implement a malicious UA what will be the consequences? One observation is that the threat is not as dangerous as if MM1 would lack security mechanisms. If there was no user authentication or integrity protection on MM1 any hacker on the Internet could inject malicious messages into the system. A malicious UA would still need to perform authentication before it can access the MMS. This means that malicious UAs can be detected and blocked from accessing the system.

Although it is possible to track malicious UAs the system must still be robust enough to minimize the impact of DoS attacks and fraud.

To prevent DoS attacks from malicious UAs the MMSC must:

- Resist Buffer overflow attacks on the UA generated MM1 messages
- Resist DoS attacks where the UA tries to flood the MMSC with MM1 messages.
- Implement mechanisms to resist misuse of the protocol flow, e.g. the UA doesn't send M-Acknowledge.ind in a MMS retrieval transaction with confirmation.
- Be robust enough to cope with UAs that modify the MMSC generated transaction identifier in messages such as the M-NotifyResp.Ind and the M- Acknoledge.Ind
- Have sanity checks on the time parameters EarliestDeliveryTime and TimeOfExpiry

A potential source of fraud can be found in the information generated by the UA that may be included in the CDR for a MM. The information we believe have the greatest potential for misuse is:

- ContentInfo, i.e. audio, video, text etc
- MM status, i.e. delivered, rejected etc

Cooperating malicious UAs can circumvent content type based charging by agreeing to use a lower charge content type to carry high value content. For example they would set the content type text and still let the message carry a picture.

To counter this threat the MMSC must perform semantic verification between the alleged content type and the data actually carried in the MMS. This type of automatic fraud detection can be done in various ways, here are two examples:

- The content type is checked against the size of the data. 500k text messages are quite unlikely.
- The content type is checked against known headers in the data. For example messages of the type texts are not likely to carry fields from the MP3 format

A malicious UA can also misuse the MMS status by actively sending error reports even if the content was delivered correctly. Here countermeasures are more difficult. One possibility is to monitor the error frequency to find suspicious behavior.

In another type of fraud malicious UAs carry end-to-end information in redundant fields of the MM protocols. This scenario related to read report messages has been studied in [19] and [20]. The read report PDU contains the following information:

- Recipient address
- Originator address
- Message ID
- Date and Time
- Read Status

The transaction flow for the read report is shown in the figure below.

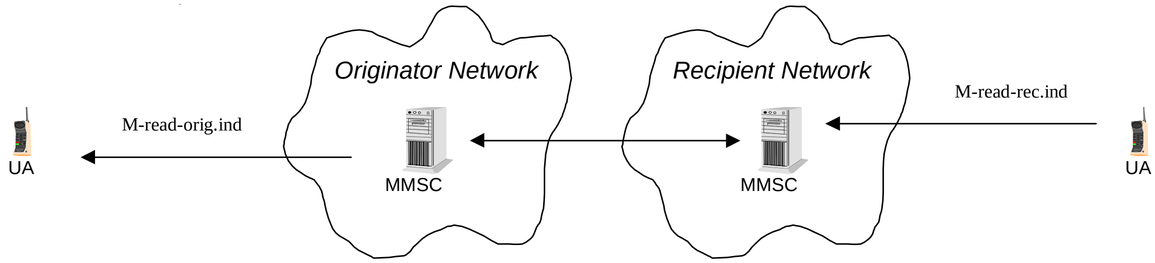

Figure 12 Read report

The potential for fraudulent usage of the fields in the read report PDU is as follows:

- If the recipient MMSC and the originator MMSC accept read-reply report PDUs without verifying them against earlier MM transfer or accept multiple read-reply report requests for the same MM, it is possible to send faked read-reply reports end-to-end.

- If the recipient MMSC and the originator MMSC don’t check the *Recipient address* and *Originator address* then those fields can be used to carry information end-to-end

- The *date and time* and *read status* fields can be used to carry information end-to-end

To prevent misuse the following countermeasures have been identified:

- A read report PDU shall be discarded if the messaged doesn't match the Message ID of a previous MM where the Read-Reply indicator is activated.
- A read report PDU shall be discarded if a Read-Reply report message with the same Message ID already has been received by the Originator MMSC.
- A read report PDU shall be discarded if the size exceeds a determined amount of bytes.

If these mechanisms are implemented only a few bytes of the read report PDU can be used to carry information end-to-end.

Inclusion of the *Recipient address* field in the read reply report may have another drawback this time related to routing. If the *Recipient address* field is used by the MMSC to route the read reply message then it may be possible for a malicious UA send the read reply to any UA. In other words this means that the read report could be re routed to a different UA than the one that sending the original MM.

The transaction flow for the re routed read report is shown in the figure below.

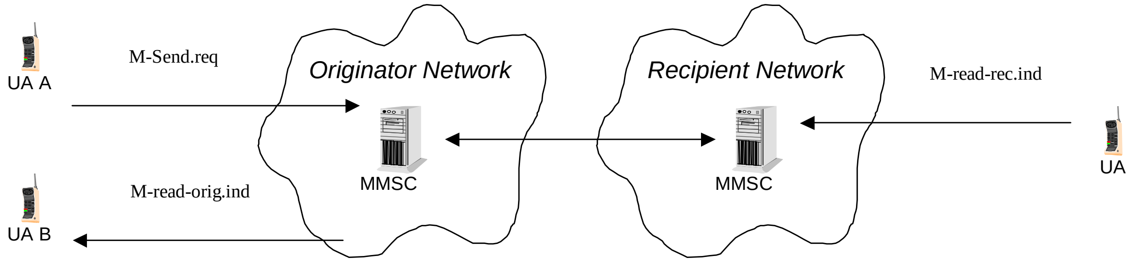

Figure 13 Read report

This exploit can be used by malicious UAs to send free of charge information but what is worse it can also be used to launch denial of service and spam attacks on MMSCs and UAs.

To prevent this another countermeasure must be introduced, the *Recipient address* of the M-read-rec.ind must not be used by the recipient MMSC for routing the message.

There are concerns that other transactions in MMS may suffer from similar side channel attacks. This seems not to be the case when studying the transaction flows. The only potential message is the delivery report but that message carries very little information end to end and therefore it poses no real threat. Furthermore the delivery report does not contain the *Recipient address* so it is not possible to reroute the message to another recipient then the message originator.

5 Conclusions and proposals for security enhancements

5.1 Conclusions

Our main conclusion is that most aspects of MMS as it is deployed currently seem to be secure with two exceptions, push and OTA configuration. Push seems to have the greatest vounerabilities, as it implements virtually no security scheme. OTA configuration is in a better position but the security mechanisms can and should be improved.

The security and authentication mechanisms in MMS are not sufficient in future configurations such as scenario 2 with incoming email or in scenarios with a MMS operator independent from the radio access operator.

If we compare MMS with IPMM we find that the security mechanisms are quite different. In IPMM authentication and integrity protection are mandatory on the first hop in the IPMM layer. In IPMM the following assumptions and properties has led to the requirement on integrity protection and authentication:

- signaling protocol in the user-plane

- access independence

MMS already fulfill the first property since the MM messages are relayed form the UA to the MMSC over WAP push or Http.

The second property can be derived from scenario 2 and from the security requirements defined in [1]. To recapitulate the requirement we are referring to: "*The MMS shall have the ability to authenticate the user regardless of access technology.* Therefore MMS layer security should be considered.

MMSC manufacturers must assume the existence of malicious UAs in the design of their products. As described earlier the MMSC must be robust against denial of service attacks on MM1; it must take countermeasures against CDR generation fraud as well as attempts on free end to end communication in redundant fields of the messages.

5.2 Security enhancements

As concluded in the previous section push security should be improved. The long-term solution is definitely to define generic push security mechanisms. With sufficiently strong mechanisms this would also improve the situation for OTA configuration of MMS parameters.

It is also possible to define a MMS specific short-term solution for the push security issue. This can be as simple as a whitelist mechanism based on the pre-provisioned MMSC address. It can also be based on cryptographic mechanism based on keys derived from the SIM. The latter approach should also be pursued for OTA configuration.

Whitelist, blacklist and filtering mechanisms should also be deployed on the MMSCs as a generic countermeasure to prevent fraud and SPAM.

An alignment with IPMM security should be considered. One obvious way to align the solutions would be to:

- Use EAP-AKA in http for MMS

- Run MMS over symmetric key TLS where they keys are derived from AKA.

Furthermore the S3 generic authentication architecture workitem should preferably also take MMS into account when defining the authentication mechanisms and architecture.

Definition of an end to end security solution for non-routing related information on SMTP level to enable a more secure deployment of MMS broker functionality should also be considered.

A sufficiently strong authentication and integrity protection solution should be defined for scenario 2 in order to allow incoming email on MM3.

Define robustness requirements on MMSC implementations to counter DoS and fraud from malicious UAs.

5.3 Final remarks

End-to-end security was not considered in this report since content adaptation was deemed to be an important MMS function, which would not be possible to implement on encrypted messages. This assumption may in the future be reconsidered since transcoding will effectively be prevented by the introduction of DRM in MMS.

Although this report is a systematic analysis of potential security threats as well as fraud possibilities in MMS it will by no means have found all weaknesses. It is therefore important to closely monitor the fraud attempts that arise as MMS becomes widely adopted.