

TELECOMMUNICATION
STANDARDIZATION SECTOR

TD 2366

STUDY PERIOD 2005-2008

English only

Original: English**Question(s):** 9/17

Jeju, Korea, 19-28 April 2006

TEMPORARY DOCUMENT

Source: Editors**Title:** Draft Text on X.websec-3, Security Architecture for Message Security in Mobile Web Services

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU

This TD is a follow-up of the WP2 plenary meeting.

Summary

This draft text of X.websec-3 describes security architecture and scenarios for message security in mobile Web Services.

Security services for messages are the most fundamental security requirement for mobile Web Services. Although components for message security such as WS-Security have been standardized, still there is no standard architecture and service scenarios for providing message security for mobile Web Services. Since SOAP messages use HTTP ports, they cannot be filtered by firewalls, and it is required to provide message filtering mechanism based on message contents in the architecture for secure mobile Web Services. It is also required to integrate security policy mechanisms suitable for Web Services message security and message filtering into the architecture. Since many mobile terminals do not have the enough processing power to fully support Web Services protocol stack, and many backend application servers are not based on Web Services, it is also necessary to provide interworking mechanisms and scenarios between mobile Web Services and legacy non-Web Services applications.

This draft text is to establish a guideline on security architecture and scenarios for message security in mobile Web Services that satisfies above requirements.

Contact:	Jae Seung Lee ETRI Korea	Tel: +82 42 860 1326 Fax: +82 42 860 5611 Email: jasonlee@etri.re.kr
-----------------	--------------------------------	--

Contact:	Ki Young Moon ETRI Korea	Tel: +82 42 860 6644 Fax: +82 42 860 5611 Email: kymoon@etri.re.kr
-----------------	--------------------------------	--

Contact:	Kyo-Il Chung ETRI Korea	Tel: +82 42 860 1920 Fax: +82 42 860 5611 Email: kyoil@etri.re.kr
-----------------	-------------------------------	---

Attention: This is not a publication made available to the public, but an **internal ITU-T Document** intended only for use by the Member States of ITU, by ITU-T Sector Members and Associates, and their respective staff and collaborators in their ITU related work. It shall not be made available to, and used by, any other persons or entities without the prior written consent of ITU-T.

1. Scope

The scope of this contribution deals with the security architecture and scenarios for secure mobile web services as follows:

- Integrated security architecture for message security in mobile web services that consist of various mobile terminals and networks
- Integrated security architecture that covers legacy network components or services that do not support full web services protocol stack
- Authentication, integrity and confidentiality of the message in mobile web services environment
- Integrated security architecture that utilizes security policy for message security on mobile web services environment
- Reference message security architecture and its usage model for mobile web services

2. References

The following ITU-T Recommendations and other references contain provisions, which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published.

The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

(To be developed)

3. Definitions

(To be developed)

4. Abbreviations

(To be developed)

CA	Certificate Authority
PKI	Public Key Infrastructure
SOAP	Simple Object Access Protocol
TLS	Transport layer security
TTP	Trusted Third Party
UDDI	Universal Discovery, Description, Integration
WSDL	Web Services Description Language
XML	eXtensible Markup Language

5. Overview of web services security

Web services are a set of protocols based on XML (eXtensible Markup Language). Fig.1 illustrates the base Web services protocols:

- SOAP (Simple Object Access Protocol): defines the message format in XML contains

the service request and response. SOAP is independent of any particular transport and implementation technology.

- WSDL (Web Services Description Language): describes a Web service. It provides a programmatic way to describe what a service does, paving the way for automation.
- UDDI (Universal Discovery, Description, Integration): is a cross industry initiative to create a standard for service discovery together with a registry facility that facilitates the publishing and discovery processes.

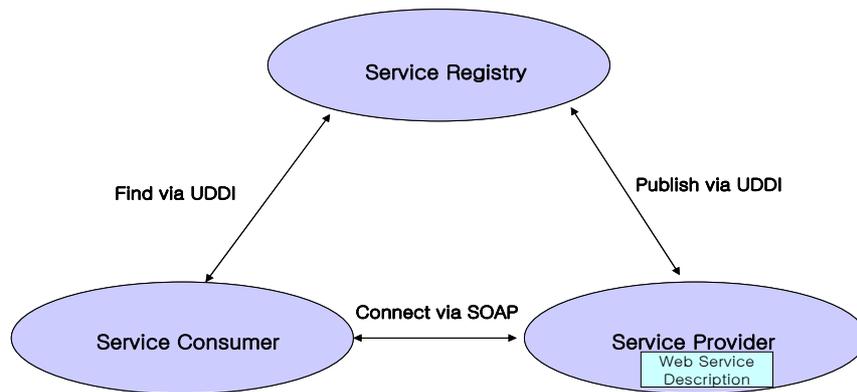


Fig.1 Base Web Services Protocols

Core Web services technologies such as SOAP, WSDL and UDDI do not directly provide the security mechanisms. Recognizing this, IBM, Microsoft and Verisign have jointly started to develop a comprehensive security model for Web services. The Web services security model introduces a collection of individual, inter-related specifications that describe an approach for layering security facilities into a Web service environment. The architecture is designed to allow mixing-and-matching of the specifications, enabling implementers to deploy only the specific parts they need.

The Web services security roadmap suggested by IBM and Microsoft is composed of a whole suite of specifications covering various facets of security (messaging, policies, trust, privacy, etc). Fig. 2 illustrates the roadmap.

The specifications build upon one another and are all build on top of a single specification, WS-Security (SOAP Messages Security) , that defines a message security model.

The summary of the specifications are as follows:

- WS-Security: describes how to attach signatures and encryption headers to SOAP messages. In addition, it describes how to attach security tokens, including binary security tokens such as X.509 certificates and Kerberos tickets, to messages.
- WS-Policy: describes the capabilities and constraints of the security and other business policies on intermediaries and endpoints (e.g. required security tokens, supported encryption algorithms, privacy rules).
- WS-Trust: describes a framework for trust models that enables Web services to securely interoperate.
- WS-Privacy: describes a model for how Web services and requesters state privacy preferences and organizational privacy practice statements.

- WS-SecureConversation: describes how to manage and authenticate message exchanges between parties including security context exchange and establishing and deriving session keys.
- WS-Federation: describes how to manage and broker the trust relationships in a heterogeneous federated environment including support for federated identities.
- WS-Authorization: describes how to manage authorization data and authorization policies.

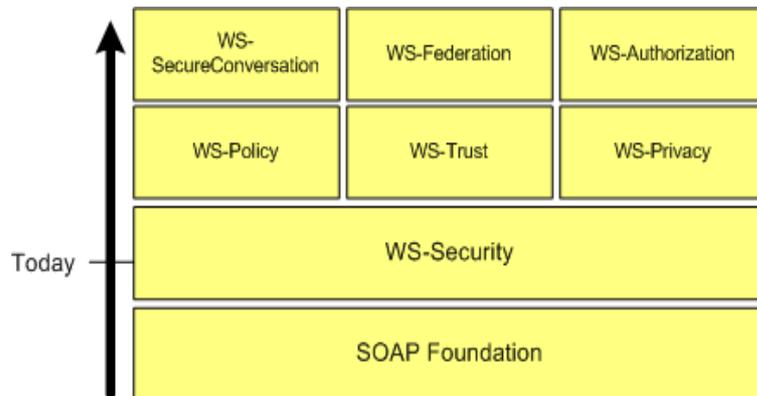


Fig.2 WS-Security Roadmap

WS-Security (SOAP Messages Security) is a base protocol in the Web services security architecture, and it is now an OASIS standard. WS-Security describes extensions to the SOAP protocol to provide secure messaging, specifically ensuring message integrity and message confidentiality.

To accomplish this, the specification describes how to attach signature and encryption headers, as well as binary encoded security tokens such as X.509 certificates and Kerberos tickets to SOAP messages.

Message integrity is established through a combination of XML Digital Signatures and security tokens. By pairing digital signatures with security tokens that either contain or imply key data, the recipient of a message can be assured that the message has been transmitted without modification by a trust party. Digital signatures support multiple signatures from multiple actors on the same document, as well as multiple signature formats.

The security token mechanism defined by WS-Security is sufficiently extensible to support multiple security token formats (username-password pairs, X.509 Certificates, etc.) and allow for the explicit inclusion of a token, or an assertion regarding a security token that exists elsewhere.

Message confidentiality is established through a combination of XML Encryption and security tokens. As is the case with digital signatures, the encryption mechanisms defined in WS-Security are designed to support a wide variety of encryption technologies, processes, and operations by multiple actors. An encrypted element may also reference a security token.

WS-Security specification also defines a mechanism for encoding binary security tokens, such as X.509 Certificates, Kerberos tickets, and opaque encrypted keys, and transmitting them with a SOAP message.

Fig. 3 illustrates the structure of a message secured by WS-Security.



Fig.3 Structure of a message secured by WS-Security

XML Signature for a part of the SOAP message is generated and inserted into Security Header element. A part of the SOAP message is encrypted using XML Encryption and its header information is inserted into Security Header element, and the encrypted part is replaced by its cipher data. Security token(s) which is related to digital signature or encryption is added to the Security Header element. A Timestamp element, which is used to determine the freshness of the message, may be inserted into the Security Header element. The Security Header element is inserted into SOAP Header part of the SOAP message.

WS-Security is a standard set of SOAP extensions that can be used when building secure Web services to implement message level integrity and confidentiality. It is a building block that can be used in conjunction with other Web services extensions and higher-level application-specific protocols. In the Web services scenarios, SOAP messages may be exchanged via intermediary nodes, and WS-Security can provide end-to-end security to the messages. The intermediary nodes may perform additional security processing to the SOAP messages if necessary.

6. Security Architecture for Message Security in Mobile Web Services

This section describes a security architecture model for message security in mobile web services that consist of various mobile terminals and networks.

Usage model based on this architecture, including detailed processing flows, is explained in the next section.

6.1 Reference Security Architecture for Message Security in Mobile Web Services

Fig.4 illustrates reference security architecture for message security in mobile web services. It consists of following components:

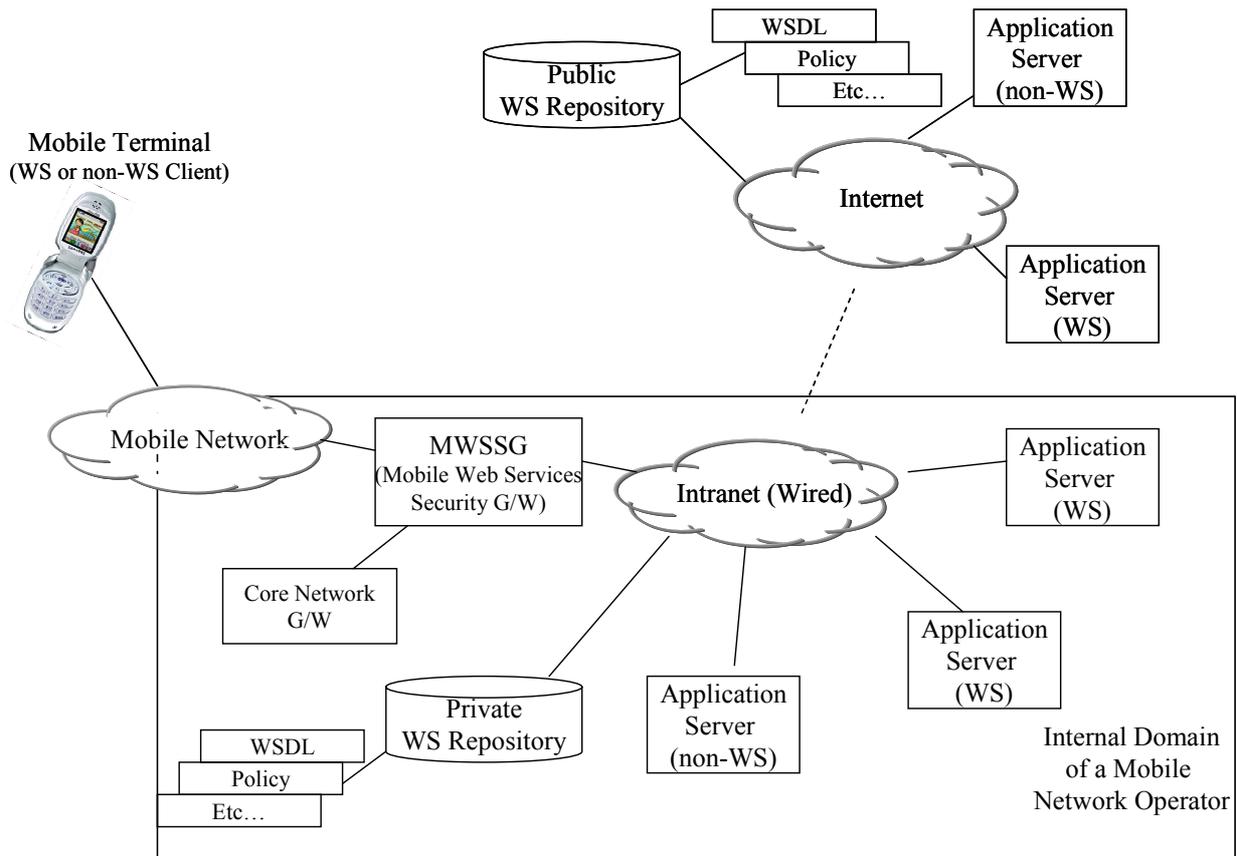


Fig.4 reference security architecture for message security in mobile web services

- Mobile terminal: A mobile terminal is a client of mobile services. It may be a cellular phone or a PDA. It may support full web services protocol stack or not. It is required to support TCP/IP and SSL/TLS (or WTLS) for this proposed architecture.
- MWSSG (Mobile Web Services Security Gateway): MWSSG is the most important component in this reference architecture. All requests from mobile clients are sent to MWSSG. It provides a single access point to all application servers.

If the request is in the form of SOAP message (that is, client is web services enabled), then it check the tags and validates the schema. It also verifies that the SOAP message has been processed according to the given security policy. If the validation of the message is successful, then the message is routed to the destination service, by referencing the actual endpoint from the private web services repository. If the verification of the message fails, then the request is rejected. This feature is helpful for reducing the load of the system and security.

If the incoming request is not a SOAP message and it is encrypted using SSL or TLS, MWSSG decrypts it and validates the contents to check whether it is a malicious message or not. MWSSG also plays the role of a message converter. That is, if the request message is a SOAP message and the target service is not based on web services, then the SOAP message is converted into the legacy message that can be understood by the target service. In this case, application specific knowledge is required for the message conversion. It is possible to make a request to backend JMS or EJB servers, for example, by using WSIF (Web Services Invocation Framework) by IBM. It makes JMS

or EJB services, whose interfaces are described in WSDL, to be invoked by web services clients. The converted message may be secured again by SSL or TLS. If the incoming message is a legacy message (that is, the client is not web services enabled) and the target service expects a SOAP message, then MWSSG converts the message into SOAP and properly secure it if it is required, by using WSDL and security policy from the Private WS Repository.

If both mobile client and target web service provide full web services protocol stack, then the request and response SOAP messages are secured by WS-Security, and they may be exchanged without decryption by intermediaries.

- Application server provides services to mobile clients and it may or may not support full web services protocol stack. It publishes its interface and address information to Private Web Services Repository.
- Only the interface exposed by MWSSG is published into the Public Web Services repository. Subset of the interfaces that is required to invoke the web service via MWSSG by the client is published by MWSSG to the Public WS Repository. In this case, endpoint of the target web service is replaced by the endpoint of MWSSG. So, the client only knows the endpoint of the MWSSG.
- Core Network Gateway provides access to the network elements of the network operator. An example of the Core Network Gateway is Parlay/OSA Gateway, which used to link applications using the Parlay/OSA APIs with the existing network elements. The Parlay/OSA Gateway is under the control of the network operator or service provider, and is a single point through which all Parlay/OSA interactions pass. MWSSG may use Core Network Gateway to use functions provided by network elements that are needed for security management. Parlay/OSA Gateway may be accessed using Parlay X Web Services. Since Parlay X Web Services are Web Services based, these services also can be protected by MWSSG.

TTP (Trusted Third Party) such as CA or an XKMS (XML Key Management Specification) server may be used to provide certificate issuance services. OCSP (Online Certificate Status Protocol) server or XKMS server may be used for certificate validation services

By using WSDL and security policy from Public WS Repository, clients send request message to MWSSG, and then the request is verified and is routed to the actual target web service by referencing Private Web Services Repository.

The proposed architecture covers legacy network components and services that do not support full web services protocol stack. It utilizes security policy and key management for message security on mobile web services environment. Illegal request messages are filtered out before they are transmitted to the application server, and it helps to reduce the load of the entire system. This architecture provides a single access point to all application servers, and it hides the deployment structure of the servers from clients. Therefore, it allows changes of deployment structure of the servers without changing the clients. It also improves security, since access control and other security processing can be done at the single access point.

6.2 Components of the Mobile Web Services Security Gateway

Fig.5 illustrates components of MWSSG.

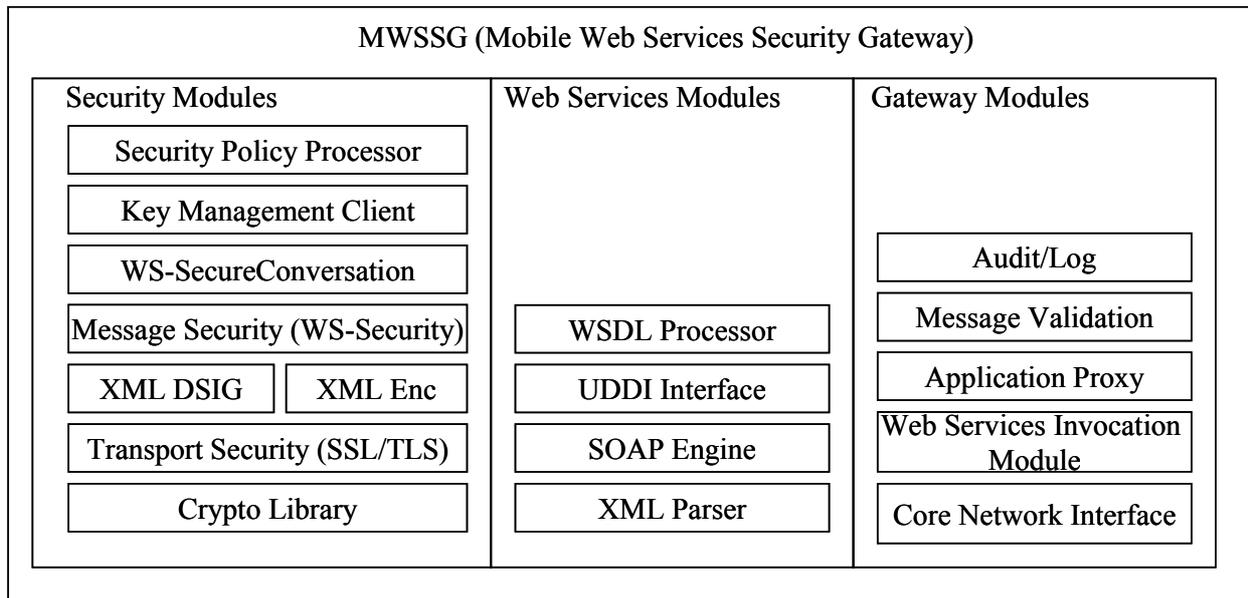


Fig.5 components of the MWSSG

6.2.1 Security Modules

These modules provide security functionality to MWSSG.

- Crypto library provides encryption, digital signature, and certificate processing functions needed for message security.
- Transport Security module provides SSL and TLS for transport security.
- XML DSIG module provides XML digital signature function for web services security.
- XML Enc module provides XML encryption function for web services security.
- Message Security module implements WS-Security specification for SOAP message security
- WS-SecureConversation module implements WS-SecureConversation specification and provides security context negotiation.
- Key Management Client module provides functions needed for requesting and verifying certificates. CA client or XKMS client may be used for requesting certificates. OCSP client or XKMS client may be used for validating certificates.
- Security Policy Processor processes security policy accessed from web services repository. It is based on WS-SecurityPolicy specification.

6.2.2 Web Services Modules

These modules provide web services related functionality to MWSSG.

- XML Parser supports parsing and manipulation of XML documents.
- SOAP engine processes web services request and response, and other web services related processing.

- UDDI interface is used to access UDDI. WSDL, security policy, and other interface information are retrieved from UDDI.
- WSDL processor retrieves WSDL and processes it.
- WS-Addressing module provides routing of SOAP messages.

6.2.3 Gateway Modules

These modules provide application proxy related and message validation functionality to MWSSG.

- Web Services Invocation module is used for invoking services that are not based on web services (e.g., JMS or EJB) requested by mobile clients using web services.
- Application Proxy is used for invoking web services from legacy clients that do not support web services. It converts the message into a SOAP message, and secures it by using WS-Security if it is required.
- Message Validation module is used to validate SOAP messages. The validation includes validation of schema, XML tags, and security policy conformance.
- Core Network Interface is used to access network elements in network operator through Core Network Gateway. This interface is usually used for monitoring the core networks. Core Network Interfaces may be implemented using Parlay X APIs. Core network elements can be accessed through Parlay Gateway and Parlay X gives application developers access to the Parlay gateways using Web Services.
- Audit/Log module is used to audit and log the system events.

6.3 Comparison with other Technologies

6.3.1 Parlay X

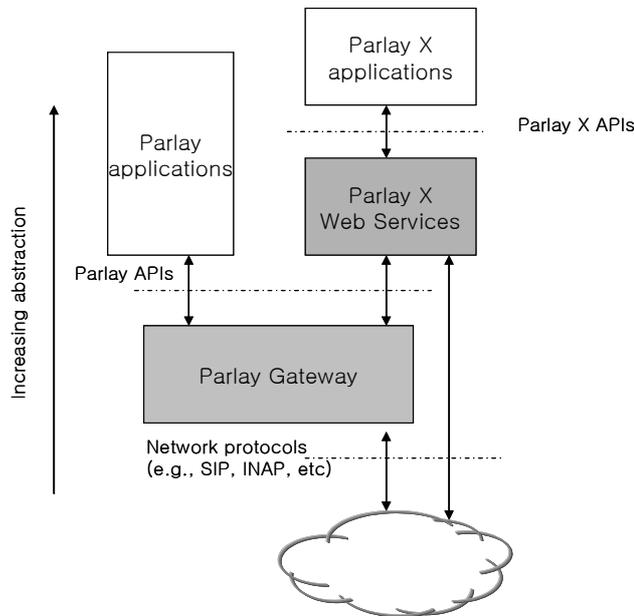


Fig.5.1 Parlay X Architecture

Parlay X is one of the specifications that are currently being worked on by the working groups of Parlay. Parlay describes a network API that allows developers to create applications that access the public network. It is specified in a number of forms including interface description language (IDL), Web Services description language (WSDL) and Java. One of the main strengths of Parlay has been its network and technology independence. Parlay achieved this through their gateway technology.

Figure 6.3 gives a functional representation of the Parlay X architecture and shows relationship between Parlay X Web Services and Parlay APIs. A Parlay gateway typically implements the Parlay APIs. Parlay X Web Services represent an abstraction and simplification of the Parlay APIs.

MWSSG is a gateway used to secure Web Services at the message level, and is different from such gateways. MWSSG may use Parlay Gateway to use functions provided by network elements that are needed for security management. Parlay/OSA Gateway may be accessed using Parlay X Web Services. Since Parlay X Web Services are Web Services based, these services also can be protected by MWSSG. Parlay Gateway and MWSSG are not duplicated gateway, and they can be combined together for better mobile web services.

6.3.2 AAA

AAA is an architectural framework for configuring a set of three independent security functions in a consistent manner. AAA provides a modular way of performing the following services:

- Authentication: Provides the method of identifying users, including login and password dialog, challenge and response, messaging support, and, depending on the security protocol you select, encryption.
- Authorization: Provides the method for remote access control, including one-time authorization or authorization for each service, per-user account list and profile, user group support, and support of IP, IPX, ARA, and Telnet.
- Accounting—Provides the method for collecting and sending security server information used for billing, auditing, and reporting, such as user identities, start and stop times, executed commands (such as PPP), number of packets, and number of bytes.

MWSSG is focused on security of messages, such as authentication and access control of Web Services messages. MWSSG assumes that user authentication and authorization are performed using existing security services such as AAA in mobile operator domain, and does not provide such security services.

7. Usage Model for Message Security in Mobile Web Services

In this section, usage model of the proposed message security architecture and its processing flows are explained. Usage model is classified into four categories.

Following conditions are assumed in the usage model:

- Mobile clients or application servers may or may not support full web services protocol stack.
- Application servers are located in the internal domain of a mobile network operator or in the external domain that is connected to Internet.
- Mobile clients, MWSSG, and application servers have obtained certificates issued by TTP (Trusted Third Party) such as CA or XKMS server.

7.1 Both the mobile client and the application server support web services protocol stack

7.1.1 The target web service is located in the internal domain of a mobile network operator

Fig.6 illustrates the usage model of message security between a web services enabled mobile client and a web services enabled application server. In this case, the target web service is located in the internal domain of a mobile network operator.

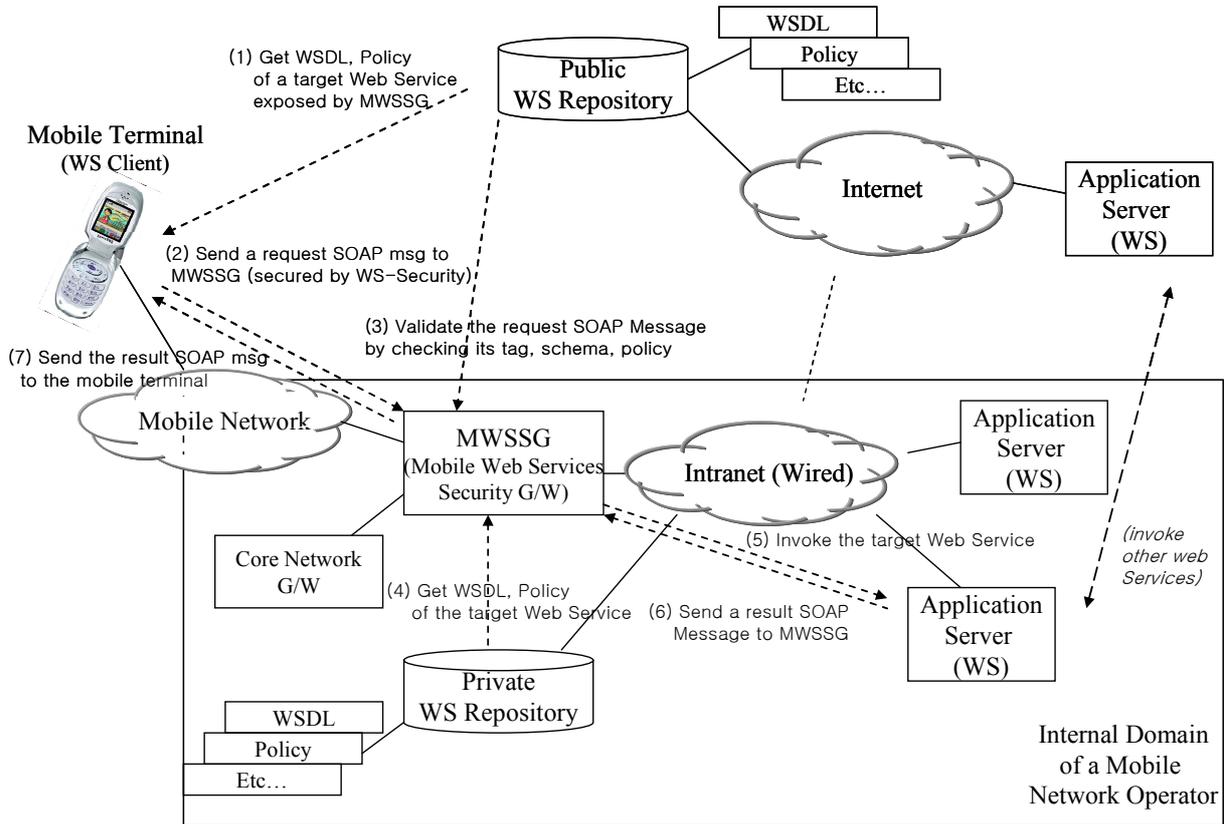


Fig.6 message security between web services enabled mobile clients and application servers (1)

- (1) A mobile client accesses WSDL and security policies of the target service from the public web services repository. They had been exposed by MWSSG. WSDL contains endpoint of MWSSG (not the endpoint of the actual service). The accessed security policies and WSDL contains security policies and interfaces that are required to invoke target service via MWSSG. The actual endpoint, interfaces and security policies are stored in the private web services repository in the internal domain of a mobile network operator, and the repository cannot be accessed directly by the clients.
- (2) The client sends a request SOAP message that has been secured by applying WS-Security, according to the security policies to MWSSG. The SOAP message had been generated by referencing WSDL.
- (3) MWSSG receives the request SOAP message and validates the tags and schema. It also checks that the message has been secured according to the security policy. WSDL and security policies accessed from the public web services repository are used. If the validation fails, MWSSG rejects the request.

- (4) MWSSG accesses WSDL and security policies of the target web service from the private web services repository. The actual endpoint of the target service can be obtained from the WSDL.
- (5) MWSSG invokes the target web service by sending the received SOAP message to the target web service, using the actual endpoint information. If the SOAP message is not decrypted at the MWSSG and is sent to the target web service, then end-to-end security between the client and the web service is achieved. If MWSSG and the target web service can trust each other, MWSSG may decrypt the message and validate the signature of it if the target web service has poor processing power or it does not have web services security functionality. The target web service may invoke other web services to perform the requested operations.
- (6) The response from the target web service is in SOAP message, and it is secured by applying WS-Security, according to security policies of the target service, and it is sent to MWSSG. If MWSSG and the target web service can trust each other, the target service may omit the security processing of the message. In this case, MWSSG secures the message before it sends the message back to the mobile client, according to the security policies accessed from the private web services repository.
- (7) MWSSG sends the secured SOAP response message to the mobile client.

7.1.2 The target web service is located in the external domain of a mobile network operator

Fig.7 illustrates the usage model of message security between a web services enabled mobile client and a web services enabled application server. In this case, the target web service is located in the external domain of a mobile network operator.

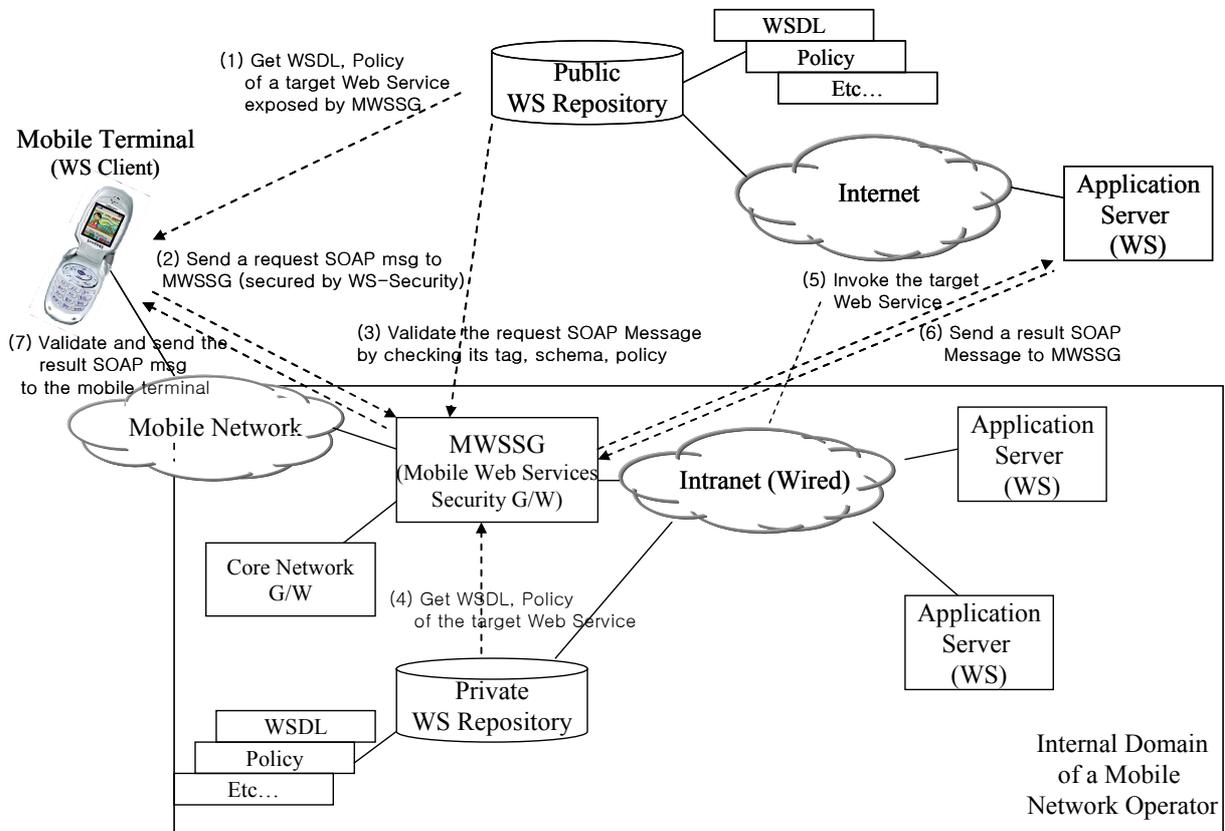


Fig.7 message security between web services enabled mobile clients and application servers (2)

- (1) ~ (4) is same as (1)~(4) of 4.1.1
- (5) is same as (5) of 4.1.1 except that the SOAP request message must not be decrypted at MWSSG, since the request is sent to the external domain. There may be another MWSSG at the target domain and it may validate the request message in the same manner as MWSSG of the source site does.
- (6) is same as (6) of 4.1.1 except that the target web service must secure the SOAP message by WS-Security. (If there is another MWSSG at the target domain, then it may secure the SOAP message instead of the target service.)
- (7) MWSSG validates the response SOAP message by checking its tag, schema, and security polices. If it is successful, it is sent to the mobile client.

7.2 The mobile client does not fully support web services protocol stack and the target service supports web services protocol stack

7.2.1 The target web service is located in the internal domain of a mobile network operator

Fig.8 illustrates the usage model of message security between legacy mobile terminal and web services enabled application server. This case covers two cases: the mobile client does not support web services at all; and the mobile client support web services but does not support WS-Security functionality.

It is assumed that the mobile client has the interface and security policies information to access the target service via MWSSG before it sends the request.

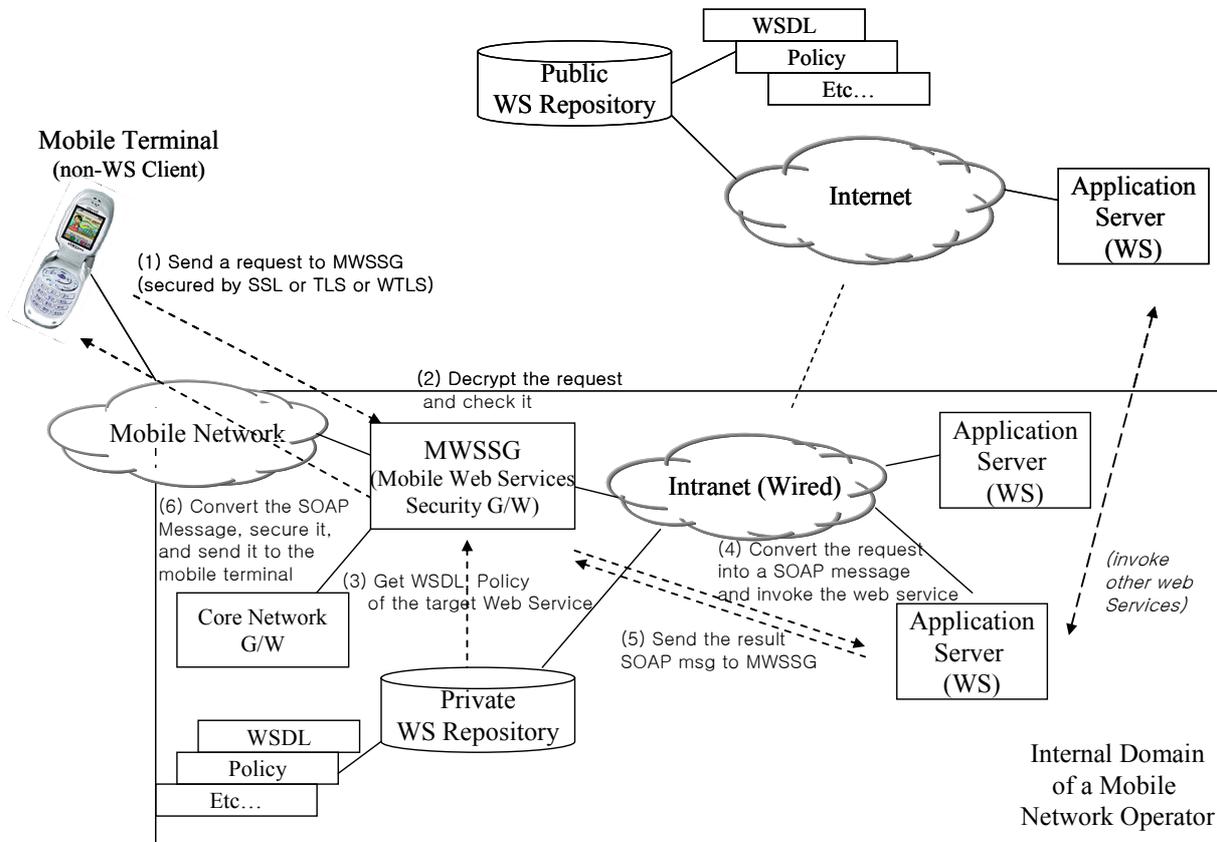


Fig.8 message security between legacy mobile client and web services enabled application server (1)

- (1) The client sends a request message that has been secured by applying SSL or TLS or WTLS, according to the security policies to MWSSG. If the client does not support web services at all, then the message is not in SOAP. If the client does not support WS-Security but supports basic web services, then the message may be in SOAP and the whole message is encrypted by SSL or TLS or WTLS. In this case, WSDL from the public web services repository may be used to generate the request message.
- (2) MWSSG decrypts the request and checks it. If the message is in SOAP, then its tags and schema are validated. Otherwise, the message is checked to see that it is not a malicious code.
- (3) MWSSG accesses WSDL and security policies of the target web service from the private web services repository. The actual endpoint of the target service can be obtained from the WSDL.
- (4) If the request message is not in SOAP, then it is converted into a SOAP message, and sent to the target web service by MWSSG. If the request message is already in SOAP, then conversion is not required. The request SOAP message may be secured by WS-Security if it is required by security policies. The target web service may invoke other web services to perform the requested operations.
- (5) The response from the target web service is in SOAP message, and it may have been secured by applying WS-Security or not. It is sent to MWSSG.
- (6) MWSSG decrypts the response message if it had been secured. If the request message was in SOAP, then secure the SOAP message by SSL or TLS, and send it to the mobile client. If the request message was not in SOAP, then convert the SOAP message, secure it by applying SSL or TLS, and send it back to the client.

7.2.2 The target web service is located in the external domain of a mobile network operator

Fig.9 illustrates the usage model of message security between legacy mobile terminal and web services enabled application server.

- (1) is same as (1) in 4.2.1
- (2) MWSSG bypasses the request to the MWSSG located at the target domain, since the local MWSSG and the target web service are not trusted. The local MWSSG is not allowed to decrypt and see the contents of the message. The address of the target MWSSG had been published to the local private web services repository, and they may be used by local MWSSG to access the destination MWSSG.
- (3) The target MWSSG decrypts the request and checks it. If the message is in SOAP, then its tags and schema are validated. Otherwise, the message is checked to see that it is not a malicious code.
- (4) The target MWSSG accesses WSDL and security policies of the target web service from the private web services repository in its domain. The actual endpoint of the target service can be obtained from the WSDL.
- (5) If the request message is not in SOAP, then it is converted into a SOAP message, and sent to the target web service by the target MWSSG. If the request message is already in SOAP, then conversion is not required. The request SOAP message may be secured by WS-Security if it is required by security policies. The target web service may invoke other web services to perform the requested operations.

- (6) The response from the target web service is in SOAP message, and it may have been secured by applying WS-Security or not. It is sent to the target MWSSG.
- (7) The target MWSSG decrypts the response message if it had been secured. If the request message was in SOAP, then secure the SOAP message by SSL or TLS, and send it to the source MWSSG. If the request message was not in SOAP, then convert the SOAP message, secure it by applying SSL or TLS, and send it back to the source MWSSG.
- (8) The source MWSSG sends the response back to the mobile client.

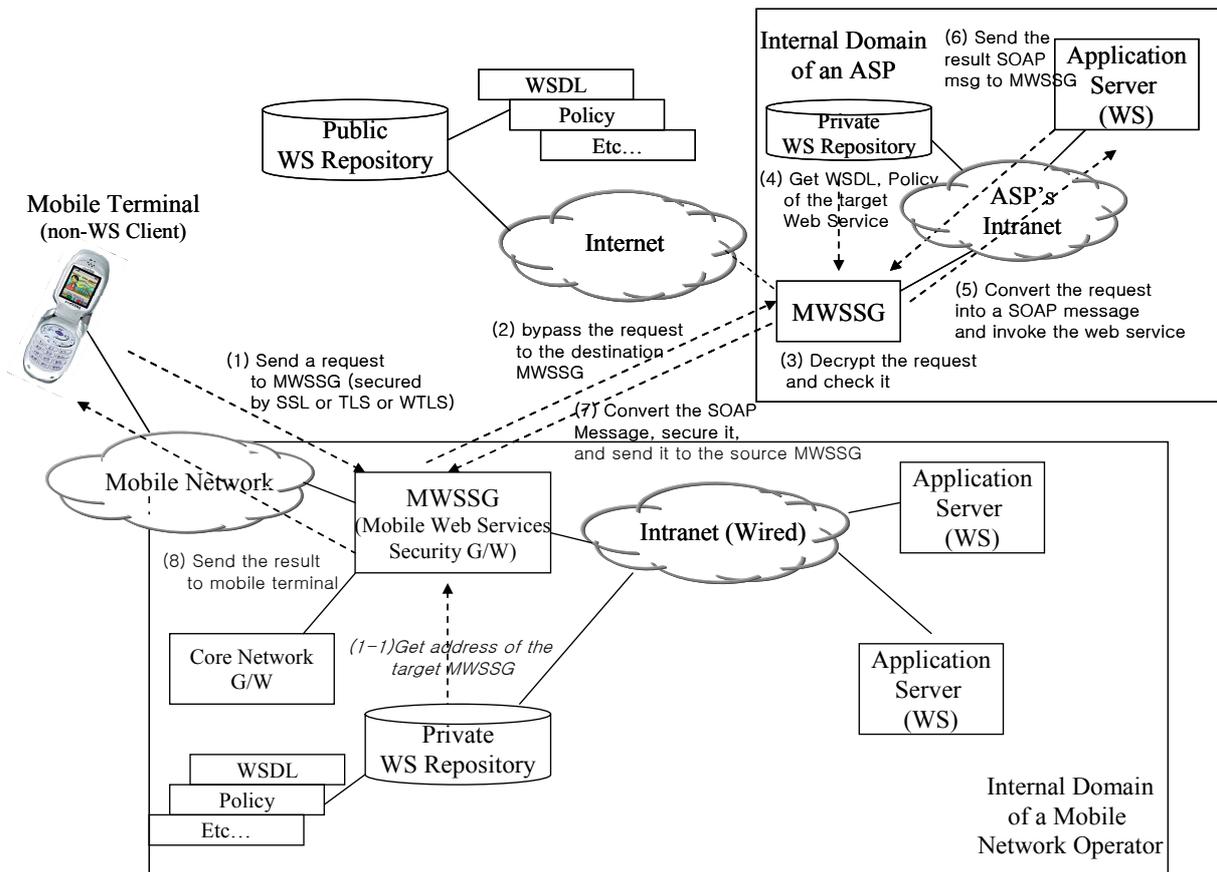


Fig.9 message security between legacy mobile client and web services enabled application server (2)

7.3 The mobile client supports web services protocol stack and the target service does not support web services protocol stack

7.3.1 The target service is located in the internal domain of a mobile network operator

Fig.10 illustrates the usage model of message security between web services enabled mobile terminal and legacy application server.

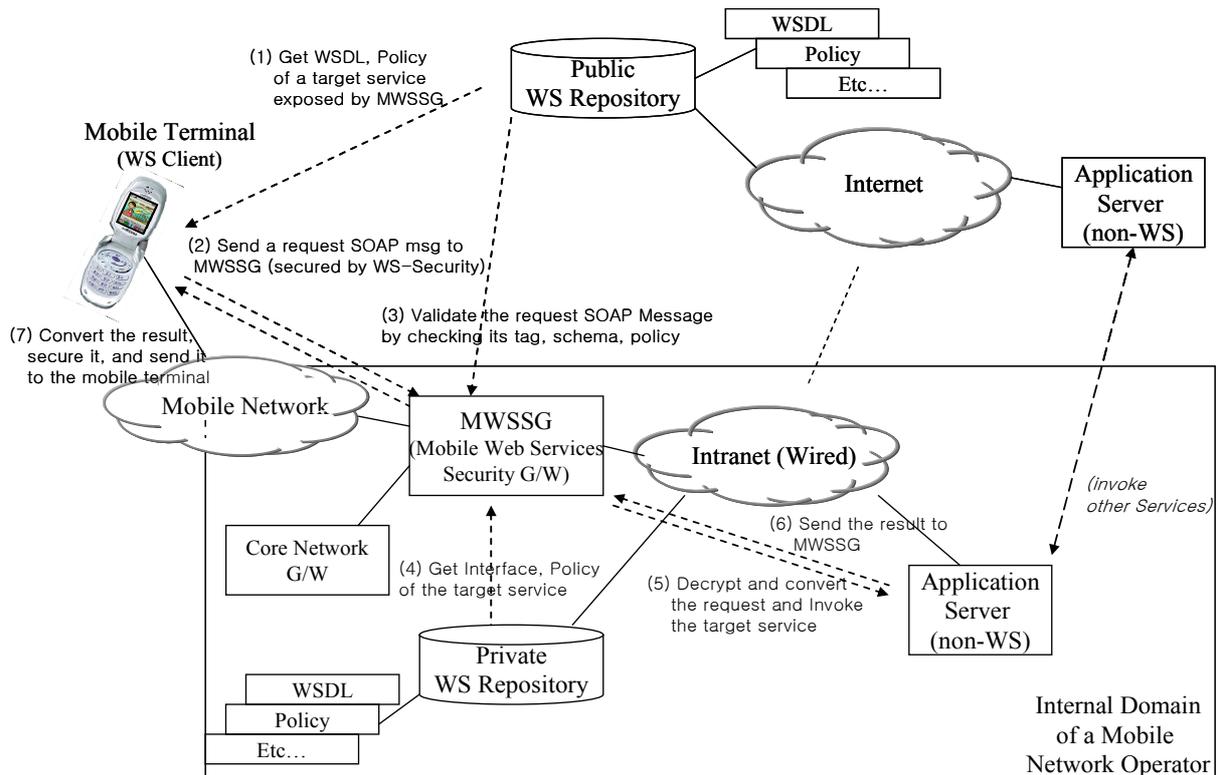


Fig.10 usage model of message security between web services enabled mobile client and legacy application server (1)

- (1) A mobile client accesses WSDL and security policies of the target service from the public web services repository. They had been exposed by MWSSG.
- (2) The client sends a request SOAP message that has been secured by applying WS-Security, according to the security polices to MWSSG. The SOAP message had been generated by referencing WSDL.
- (3) MWSSG receives the request SOAP message and validates the tags and schema. It also checks that the message has been secured according to the security policy. WSDL and security policies accessed from the public web services repository are used. If the validation fails, MWSSG rejects the request.
- (4) MWSSG accesses interfaces and security policies of the target service from the private web services repository. The actual endpoint of the target service can be obtained from the web services repository.
- (5) MWSSG decrypts and verifies the signature of the request message. Then the SOAP request message is converted into the legacy request message that can be understood by the target service. In this case, application specific knowledge is required for the message conversion. It is possible to make a request to backend JMS or EJB servers, for example, by using WSIF (Web Services Invocation Framework) by IBM. It makes JMS or EJB services, whose interfaces are described in WSDL, to be invoked by web services clients. The converted message may be secured again by SSL or TLS if it is required.

- (6) The target service receives the request, decrypts it if it had been encrypted, and sends the response back to the MWSSG. The response may be encrypted using SSL or TLS if it is required.
- (7) MWSSG decrypts the message if it had been encrypted, and converts the message back into SOAP, and sends the response back to the mobile client.

7.3.2 The target service is located in the external domain of a mobile network operator

Fig.11 illustrates the usage model of message security between web services enabled mobile terminal and legacy application server.

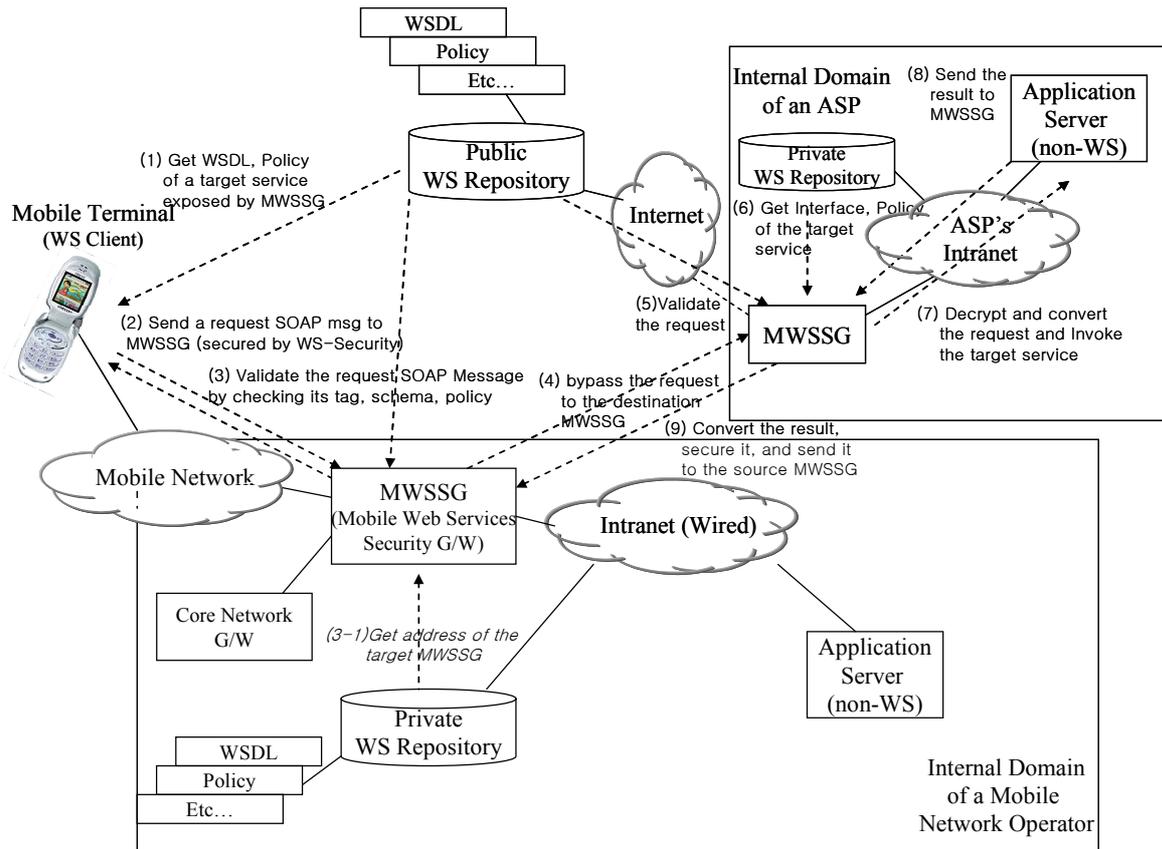


Fig.11 usage model of message security between web services enabled mobile client and legacy application server (2)

- (1) A mobile client accesses WSDL and security policies of the target service from the public web services repository. They had been exposed by MWSSG. They contain interfaces and security polices required to access the target service in the external domain via destination MWSSG, but the endpoint information contains the address of the local MWSSG.
- (2) The client sends a request SOAP message that has been secured by applying WS-Security, according to the security polices to the destination MWSSG. The SOAP message had been generated by referencing WSDL.
- (3) MWSSG receives the request SOAP message and validates the tags and schema. It also checks that the message has been secured according to the security policy. WSDL and security policies accessed from the public web services repository are used. If the validation fails, MWSSG rejects the request.

- (4) MWSSG bypasses the request to the MWSSG located at the target domain, since the local MWSSG and the target service are not trusted. The local MWSSG is not allowed to decrypt and see the contents of the message. The address of the target MWSSG had been published to the local private web services repository, and they may be used by local MWSSG to access the destination MWSSG.
- (5) The destination MWSSG validates the tags and schema, and security policy conformance, since the request is from the external domain.
- (6) The destination MWSSG accesses interfaces and security policies of the target service from the private web services repository in its domain. The actual endpoint of the target service can be obtained from this web services repository.
- (7) The destination MWSSG decrypts and verifies the signature of the request message. Then the SOAP request message is converted into the legacy request message that can be understood by the target service. The converted message may be secured again by SSL or TLS if it is required. Then the destination MWSSG sends the message to the destination service.
- (8) The target service receives the request, decrypts it if it had been encrypted, and sends the response back to the destination MWSSG. The response may be encrypted using SSL or TLS if it is required.
- (9) The destination MWSSG decrypts the message if it had been encrypted, and converts the message back into SOAP, secures it by applying WS-Security according to the security policies, and then sends the response back to the local MWSSG.
- (10) The local MWSSG validates the tags and schema, and security policy conformance, since the request is from the external domain. Then the response message is sent back to the mobile client.

7.4 Both the mobile client and the requested service do not support web services protocol stack

7.4.1 The target service is located in the internal domain of a mobile network operator

Fig.12 illustrates the usage model of message security between legacy mobile terminal and legacy application server.

- (1) The client sends a request message that has been secured by applying SSL or TLS or WTLS, according to the security polices to MWSSG.
- (2) MWSSG decrypts the request and checks it to see that it is not a malicious code.
- (3) MWSSG sends the decrypted request to the target service.
- (4) The target service sends the response to the MWSSG.
- (5) MWSSG encrypt the response and sent it back to the mobile client.

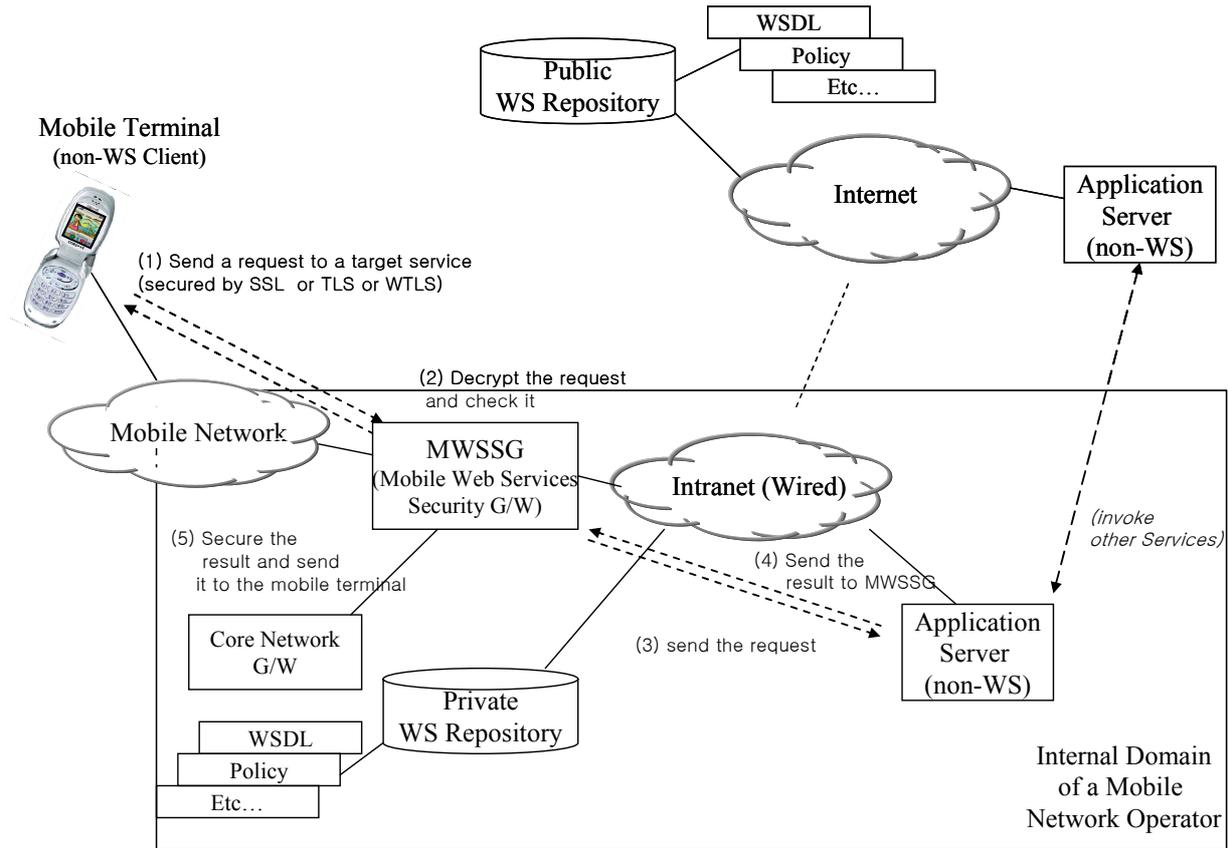


Fig.12 usage model of message security between web services enabled mobile client and legacy application server (1)

7.4.2 The target service is located in the internal domain of a mobile network operator

Fig.13 illustrates the usage model of message security between legacy mobile client and legacy application server.

- (1) The client sends a request message to the local MWSSG that has been secured by applying SSL or TLS or WTLS, according to the security policies to the destination service.
- (2) MWSSG just bypasses the request to the destination service or the destination MWSSG, since the local MWSSG and the target service are not trusted. The local MWSSG is not allowed to decrypt and see the contents of the message. The address of the target service or the destination MWSSG had been published to the local private web services repository, and they may be used by local MWSSG to access the target service.
- (3) The target service receives the request, decrypts it, generates response, secures it by applying SSL or TLS and sends it back to the local MWSSG.
- (4) The local MWSSG sends the response back to the mobile client.

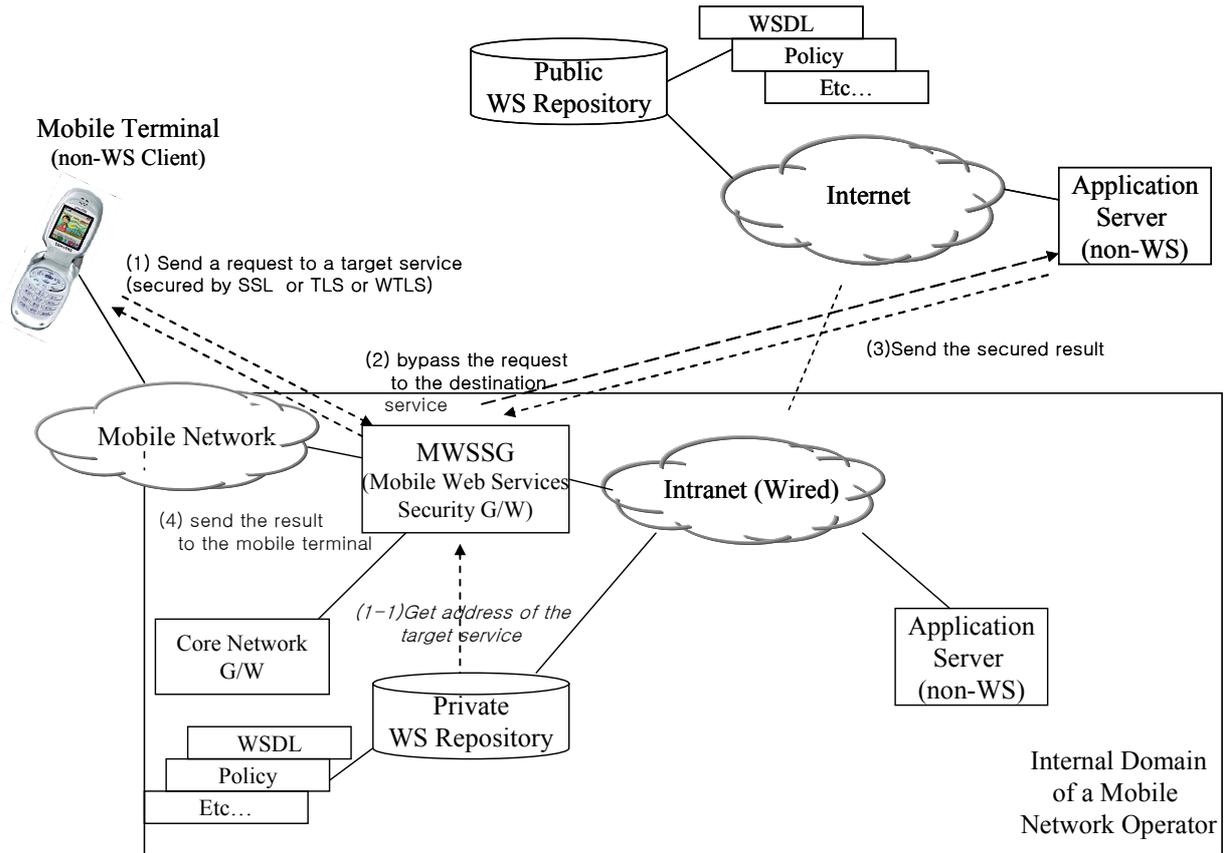


Fig.13 usage model of message security between web services enabled mobile client and legacy application server (2)