TTCN-3 Based Conformance Testing of Mobile Broadcast Business Management System in 3G networks

Zhiliang Wang[1,3], Xia Yin[2,3], Yang Xiang[2,3], Ruiping Zhu[4], Shirui Gao[2,3],
Xin Wu[2,3], Shijian Liu[2,3], Song Gao[5], Li Zhou[2,3], Peng Li[2,3]

[1]Network Research Center, Tsinghua University
[2]Department of Computer Science and Technology, Tsinghua University
[3]Tsinghua National Laboratory for Information Science and Technology (TNList)
[4]China Mobile Research Institute
[5]School of Software and Microelectronics, Peking University
Beijing, P. R. China
{wzl,yxia,xiangy08,gaoshirui,wuxin,lsj,gaosong,zhouli,lipeng}@csnet1.cs.tsinghua.edu.cn
[4]zhuruiping@chinamobile.com

Abstract. Mobile broadcast service is one of the emerging most important new services in 3G networks. To better operate and manage mobile broadcast services, mobile broadcast business management system (MBBMS) should be designed and developed. Such a system, with its distributed nature, complicated XML data and security mechanism, faces many challenges in testing technology. In this paper, we study the conformance testing methodology of MBBMS, and design and implement a MBBMS protocol conformance testing tool based on TTCN-3, a standardized test description language that can be used in black-box testing of reactive and distributed system. In this methodology and testing tool, we present a semi-automatic XML test data generation method of TTCN-3 test suite and use HMSC model to help the design of test suite. In addition, we also propose an integrated testing method for hierarchical MBBMS security architecture. This testing tool has been used in industrial level's testing.

1 Introduction

Mobile broadcast service, e.g. mobile TV, is one of the emerging most important new services in 3G networks. 3GPP (3rd Generation Partnership Project) has defined a new transport technology, called MBMS (Multimedia Broadcast/Multicast Services)[1,2]. MBMS, using 3G mobile networks as its bearer network, is a point-to-multipoint service in which data is transmitted from a single source entity to multiple recipients[2]. MBMS focuses on the transport aspects of mobile broadcast and multicast services[3], while for service layer technologies, OMA (Open Mobile Alliance) has standardized the Mobile Broadcast Services 1.0 (BCAST 1.0) specification[4,5]. BCAST 1.0 defines some functions such as Service Provisioning, Service Guide, Service and Content protection, File and Stream distribution, etc[5], and can be used over several different broadcast distribution systems, including 3GPP MBMS.

With the rapid development of bearer technologies of mobile broadcast services and the deployment of real services, operation and management of these services are also becoming very important. To better operate and manage mobile broadcast services, Mobile Broadcast Business Management System (MBBMS) has been designed and developed[6]. Referring to the related specifications of 3GPP and OMA, MBBMS uses both mobile communication network and broadcast network to make use of advantages of both kinds of network. Via mobile communication network, mobile terminals interact with MBBMS to implement the following functions: authentication and authorization, service subscription and unsubscription, service guide delivery and service key delivery, etc. Via broadcast network, demanded traffic data can be delivered from content sources to mobile terminals.

MBBMS is a very complicated system so that many challenges are faced in its design, development and testing. This paper focuses on the testing technology of MBBMS. Software testing and protocol testing techniques are widely used in order to ensure the quality of network protocol and communication system implementations. Conformance testing is a basic method of protocol testing, which can be used to test whether an implementation conforms to its protocol specification. In the field of conformance testing, many literatures have been published, such as international standard ISO/IEC 9646[7]. TTCN-3 (Testing and Test Control Notation)[8,9] is a new test specification language standardized by ETSI (European Telecommunications Standards Institute) and can be used in black-box testing of reactive and distributed system. TTCN-3 has been widely applied in many fields of testing, especially the testing of communication systems.

In this paper, we study the conformance testing methodology of MBBMS, and design and implement a MBBMS protocol conformance testing tool based on TTCN-3. In this methodology and testing tool, we present a semi-automatic XML test data generation method of TTCN-3 test suite and use HMSC (High-Level Message Sequence Charts)[24,25] model to help the design of test suite. In addition, we also propose an integrated testing method for hierarchical security architecture of MBBMS. This testing tool has been used in industrial level's testing.

The rest of the paper is structured as follows. Section 2 introduces related work. Section 3 gives an overview of MBBMS architecture. Section 4 gives test architecture for MBBMS. In Section 5, we present a framework of designing MBBMS TTCN-3 test suite. Section 6 proposes the testing method of MBBMS security mechanism. Conclusion and future work are given in Section 7.

2 Related Work

TTCN-3 has been applied in many areas. For example, in conformance testing, [10,11] applied TTCN-3 in RIPng and Mobile IPv6 respectively. In performance testing, [12] studied the performance testing of a mobile management protocol BCMP, and [13] applied TTCN-3 into distributed load testing. In our previous work [14], we designed and implemented a TTCN-3 based protocol integrated test system called PITSv3, and applied it in BGPv4+ (Border Gateway Protocol) conformance testing and OSPFv2 (Open Shortest Path First routing protocol) robustness testing.

Especially, [15,16] used TTCN-3 in web service and web application testing respectively. [17,18] developed the TTCN-3-based conformance testing tools for OMA MMS (Multimedia Messaging Service) and Push-to-talk service over Cellular in 3G or 4G networks respectively.

In the field of MSC-based testing, [19] introduced a tool developed by UK Systems and Software Engineering Research group of Motorola Labs to automatically generate test cases from a combination of MSCs and PDUs, especially, this tool can generate TTCN-2 test cases from MSCs. [20] studied TTCN-3 Test Case Generation from MSCs, and gave translation rules of MSC into TTCN-3. [11] presented an approach of TTCN-3 test cases generation from SDL (Specification and Description Language) and MSC specifications and applied the approach in testing Mobile IPv6 protocol.

In MBBMS, XML messages are used in protocol interactions, thus the mappings from XML to TTCN-3 is important for TTCN-3 based MBBMS testing. [21] has standardized the mapping from XML schemas to TTCN-3 standard data types and provided a method of using XML in TTCN-3. However, as shown in [22], this method has many limitations: for example, it will increase the complexity of the implementation of TTCN-3 encoder and decoder. So this method is difficult to be used in practical testing. Different from the method of [21], this paper presents a simple XML TTCN-3 representation and a semi-automatic XML test data generation method of TTCN-3 test suite.

In this paper, we present a TTCN-3 based MBBMS protocol conformance testing methodology, and integrate various techniques (e.g., HMSC, XML) into the testing tool.

3 An Overview of MBBMS Architecture

Figure 1 shows the architecture of MBBMS network. MBBMS uses both mobile communication network and broadcast network to make use of advantages of both kinds of network.

Via mobile communication network, mobile terminals interact with mobile broadcast service platform (for short, platform), which is the core network element in MBBMS. The main functional modules of mobile broadcast service platform are as follows:

(1) Service Guide (SG) Server: its function is service guide generation and delivery.

(2) Bootstrapping Service Function (BSF) Server: this module interacts with mobile terminals and HLR (Home Location Register) to execute GBA (Generic Bootstrapping Architecture) procedure[23] in order to support user authentication and authorization.

(3) Network Application Function (NAF) Server: the main functions of this module are service/user management (e.g., service subscription and unsubscription) and service key management (e.g., service key generation and delivery).

Via broadcast network, demanded traffic data can be delivered from content sources to mobile terminals.

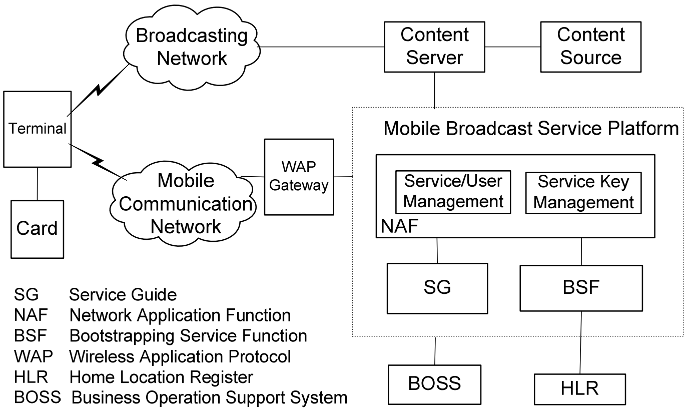

Broadcasting Network
Content Server
Content Source
Terminal
Mobile Broadcast Service Platform
WAP Gateway
Mobile Communication Network
Service/User Management
Service Key Management
NAF
Card
SG
BSF
SG Service Guide
NAF Network Application Function
BSF Bootstrapping Service Function
WAP Wireless Application Protocol
HLR Home Location Register
BOSS Business Operation Support System
BOSS
HLR

Fig. 1. MBBMS Architecture[6]

4 Test Architecture

In MBBMS, two important network elements, mobile terminal and mobile broadcast service platform, are required to be tested. Figure 2 shows the test environment for MBBMS in a test lab. In the test environment, terminals under test can be attached to the lab mobile communication network; mobile broadcast service platform under test is attached to the lab wired network; WAP (Wireless Application Protocol) gateway is used to connect the lab mobile communication network and wired network and translate and forward the messages between terminals and the platform. The test tool is located in the lab wired network. When testing terminals under test, the test tool should act as simulated platform; while when testing platform under test, the test tool should act as simulated terminal.

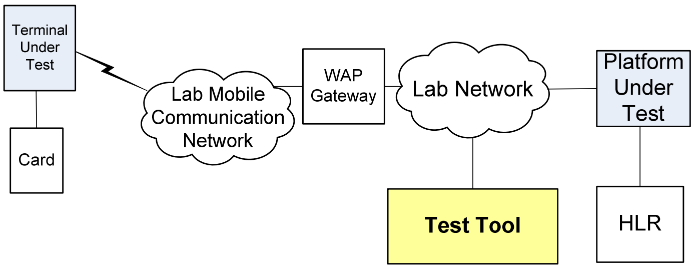
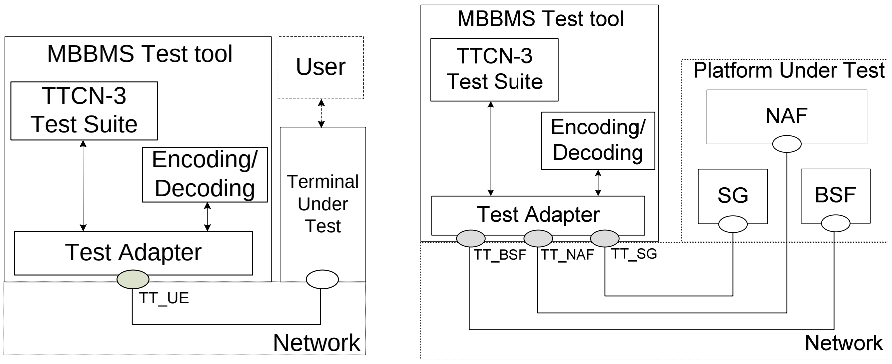


Fig. 2. Test Environment for MBBMS

Figure 3 shows the test architecture to test terminal (Figure 3(a)) and platform (Figure 3(b)). In TTCN-3 test tool, when testing a new protocol or system, TTCN-3 test suite should be designed and developed. In addition to test suite, two modules related to specific protocol under test, Test Adapter and Encoding/Decoding, should be developed in order to execute test suite. In Figure 3(a), one test system interface TT_UE is defined to interact with terminal under test; and in Figure 3(b), three test system interfaces TT_SG, TT_NAF, TT_BSF are defined to interact with SG, NAF, BSF servers respectively.



(a) Test Architecture to Terminal (b) Test Architecture to Platform

Fig. 3. Test Architecture

5 Design of TTCN-3 Test Suite

TTCN-3 test suite is the core in TTCN-3 based conformance testing. Figure 4 shows the design framework of MBBMS conformance test suite. The starting point of test suite design is the MBBMS protocol specification, from which we can get XML schemas of protocol messages and the HMSC model of protocol control flow. In this paper, we use HMSC (High-Level Message Sequence Charts)[24,25] model to help the design of test suite (see Section 5.1) and present a semi-automatic XML test data generation method in TTCN-3 test suite (see Section 5.2).

5.1 HMSC Model of MBBMS and Test Suite Organization

Message Sequence Charts (MSC) is a language to describe the interaction between a number of independent message-passing instances[24]. High-Level MSC (HMSC) [24] is an extended model of MSC. In HMSC model, a set of MSCs can be combined to form a directed graph, in which nodes can be represented as MSCs and edges indicate the execution flows of MSCs. We give the formal definition of HMSC as Definition 1.

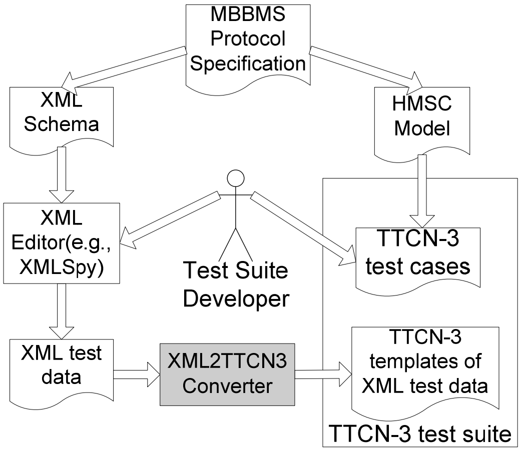


Fig. 4. Design framework of MBBMS conformance test suite

Definition 1. *HMSC* (*High-Level Message Sequence Charts*)[26,27]

An HMSC is a tuple $H=(S,\rightarrow,s^0,s^f,\lambda)$, where $(S,\rightarrow,s^0,s^f)$ is a transition system with set of nodes *S*, transition relation $\rightarrow\subseteq S\times S$, initial nodes s^0 and set of final nodes s^f. Each node $s\in S$ is labeled by an MSC, denoted by $\lambda(s)$.

An *execution* of *H* is the labeling $\lambda(s_0)\ \lambda(s_1)\ldots\lambda(s_k)$ for some path s_0, s_1, …, s_k in *H*, where, $s_0 = s^0$, $s_k= s^f$, $(s_i, s_{i+1})\in\rightarrow$ (i=0,1,2,…,k-1). □

According to its protocol specification, we can get the HMSC model of MBBMS. Figure 5 shows the simplified HMSC model, in which ▽ is the start symbol and △ is the end symbol. This model contains the following 8 MSC nodes: GBA (Generic Bootstrapping Architecture) procedure; Service Activation, meaning to activate the mobile broadcast service; SG (Service Guide) Retrieval; Service Subscription; Subscription Information Update; Account Inquiry, meaning to inquire the specific services; Service Unsubscription and SK (Service Key) Retrieval, meaning to unsubscribe a service and retrieve the service key respectively after a service has been subscribed. Figure 5 also gives the MSC of Service Subscription, in which two instances Terminal and NAF are contained. The terminal firstly sends a Service Request message to the NAF server to subscribe a specific service, and then the NAF server sends a Service Response back to the terminal to reply the subscription result.

With the help of MBBMS HMSC model, we give the strategy of test suite design. We implement one test group (i.e., a TTCN-3 module) for each MSC node. Some normal (i.e., to test the normal behaviors of the system) and abnormal (e.g., to test how to handle invalid inputs) test cases are contained in each test group. The method proposed in [20] can be used to map a MSC to a TTCN-3 test case. The organization of test suite we design is shown as table 1.

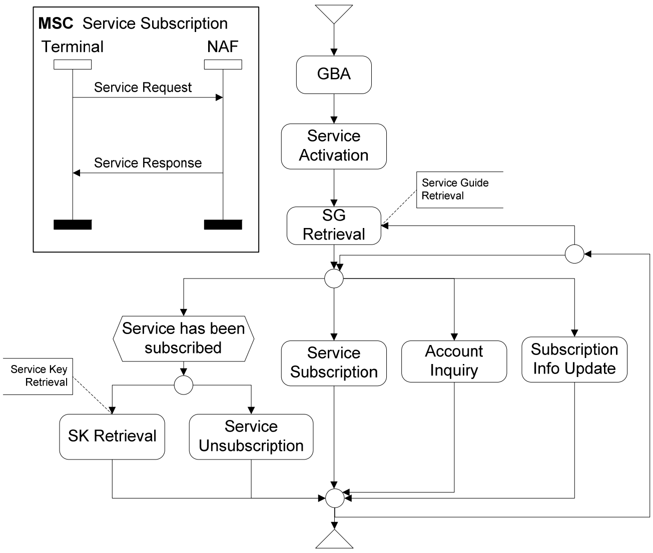


Fig. 5. The simplified HMSC model of MBBMS

Table 1. Test suite organization

No	Test Groups	Test cases for terminal under test			Test cases for platform under test		
		# of Normal Test Cases	**# of Abnormal Test Cases**	**# of Total Test Cases**	**# of Normal Test Cases**	**# of Abnormal Test Cases**	**# of Total Test Cases**
1	GBA	1	5	6	2	3	5
2	Service Guide Retrieval	5	2	7	9	10	19
3	Subscribe Info Update	3	5	8	2	3	5
4	Service Activation	4	2	6	5	6	11
5	Service Subscription	2	3	5	2	4	6
6	Service Unsubscription	2	3	5	2	3	5
7	Account Inquiry	3	2	5	3	3	6
8	Service Key Retrieval	3	1	4	4	2	6
	Total Number	23	23	46	29	34	63

To test a complete flow of MBBMS, a test path (i.e., an execution of HMSC) should be firstly extracted from HMSC model. For example, in Figure 5, a test path covering all 8 MSC nodes is as shown in Figure 6.

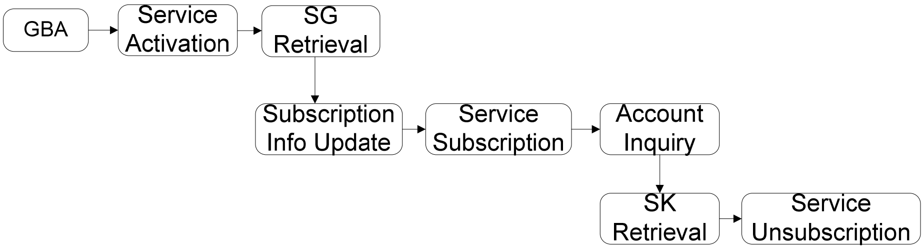


Fig. 6. A test path of HMSC covering all 8 MSC nodes

Such a test path can be implemented by the control part of a TTCN-3 module (see Figure 7). The statement **execute**, defined in TTCN-3 [9], is used to execute test cases in the TTCN-3 module control. In the TTCN-3 code of Figure 7, line 9 executes the test case Subscribe_1Service() and gets its verdict. In line 10, if the verdict of test case Subscribe_1Service() is **pass**, i.e., the service has been subscribed successfully, the following three test cases will be executed, else the three test cases will not be executed.

5.2 XML Test Data Representation and Generation

In the MBBMS protocol, HTTP+XML is used as protocol message format. How to describe the XML test data in TTCN-3 is a difficult problem. In this paper, we present a semi-automatic XML test data generation method in TTCN-3 test suite to simplify the development process of test suite. The framework of this method is shown in Figure 4. Firstly, XML schemas of protocol messages can be extracted from protocol specification; and then an XML editor, such as XMLSpy[28], can be used to generate the XML test data from XML schemas; we present a method of TTCN-3 XML test data definition and also implement a XML2TTCN3 converter tool to convert the XML test data to TTCN-3 templates automatically.

```
module MBBMS{
   ......
  control{
    execute(GBA_Basic() );       //execute basic GBA procedure
    execute(Activation_Open() ); //activate service
    execute(SGRetrieval_all() ); //retrieve all SG info
    execute(SubscriptionUpdate_NoSub() );
          //subscription info update (when no subscribed service)
    verdicttype sub_verdict;
    sub_verdict := execute(Subscribe_1Service() );
                                          //subscribe one service
    if(sub_verdict == pass) {
      //service subscription should be successful
      execute(AccountInquiry_1Service() ); //account inquiry
      execute(SKRetrieve_Normal() );
                 //retrieve service key of the subscribed service
      execute(Unsubscribe_1Service() );
                             //unsubscribe the subscribed service
    }
  }
}
```

Fig. 7. The control part of a TTCN-3 module corresponding to Figure 6

According to the characteristic of XML data, we define a generic recursive TTCN-3 type of XML messages as shown in Figure 8.

```
type record NameValueType {
  universal charstring name optional,
  universal charstring val optional
}   //Name-Value pair in XML
type record of NameValueType NameValueTypeArray;  //can be an array
type record XMLBeginType{
  universal charstring prefix optional,
  NameValueTypeArray attributes optional,
  universal charstring suffix optional
}
type record XMLContentType {
  XMLBeginType begin optional,
  XMLContentTypeArray content optional,
  universal charstring end optional
}
type record of XMLContentType XMLContentTypeArray; //can be an
array
```

Fig. 8. Generic recursive TTCN-3 type definition of XML messages

We take the Service Request message[1] in MBBMS for example. From XML schema of this message and with the help of XML editor tool, we can get a XML test data as shown in Figure 9. We divide a XML data into three parts: begin, content and end part. In the example of Figure 9, the begin part is "<ServiceRequest requestID="1">" and the end part is "</ServiceRequest>", and the content part is just located between the begin and end parts, which is a nested XML message (e.g., the example of Figure 9) or an array of several nested XML messages. Thus we define a generic record type of "XMLContentType" (see Figure 8). The begin part can be further defined as a record type of "XMLBeginType", which is comprised of the

[1] Service Request message structure can be found in section 5.1.5.2.1 of [5]

prefix, attribute and suffix part. In the example of Figure 9, the prefix of the begin part is a universal charstring “<ServiceRequest ”; the suffix part is a universal charstring “>”; the attribute part is an array of Name-Value pairs, which is defined as a record type “NameValueType”.

```
<ServiceRequest requestID="1">
  <PurchaseItem gloalIDRef="10"></PurchaseItem>
</ServiceRequest>
```

Fig. 9. An example of Service Request XML test data

In TTCN-3, templates are the test data definitions, which are used to either transmit a set of values or to test whether a set of received values matches the template specification[9]. The TTCN-3 template definition of ServiceRequestPayload, corresponding to the XML test data in Figure 9, is shown in Figure 10. This template is a parameterized one with two formal parameters, requestid and globalid, thus the XML test data in Figure 9 can be represented as ServiceRequestPayload("1", "10"), in which two formal parameters are replaced by actual parameters "1" and "10" respectively. In addition, we also implement a TTCN-3 function “findInXML”, by which we can easily extract the data value of the field we are interested in.

```
template XMLContentType ServiceRequestPayload(
   universal charstring requestid, universal charstring globalid):={
  begin := {
    prefix := "ServiceRequest",
    attribute := {
      [0] := { name := "requestID", val := int2unichar(34) & requestid & int2unichar(34) }//Name-Value pair: requestID="requestid"
    },
    suffix := omit
  }, //begin = <ServiceRequest requestID="1">
  content := {
    [0] := {
      begin := {
        prefix := "PurchaseItem",
        attribute := {
          [0] := { name := "globalIDRef", val := int2unichar(34) & globalid & int2unichar(34) }  //Name-Value pair: gloalIDRef="globalid"
        },
        suffix := omit
      },
      content := omit,
      end := omit
    }
  }, //content = <PurchaseItem gloalIDRef="globalid "></PurchaseItem>
  end := omit
}
```

Fig. 10. TTCN-3 template definition of XML test data in Figure 9

Compared with the method in [21], using this method of TTCN-3 XML test data definition and semi-automatic XML test data generation is beneficial for test developers in the following aspects:

(1) This method of TTCN-3 XML test data representation is simple and easy to understand. The recursive XML data type definition is generic, which simplifies the test data definitions when developing TTCN-3 test suite manually.

(2) This method can simplify the development of encoding/decoding module. When encoding, some symbols, such as “>”, “<” and “=”, should be added; and when decoding, these symbols should be taken off. Though this requires the additional operations of encoding/decoding module, the complexity of the encoder/decoder is much smaller than the method in [21].

(3) The method of semi-automatic XML test data generation can lighten test developers’ workload of defining large numbers of TTCN-3 templates. Especially, in MBBMS, some XML messages, e.g., Service Guide related messages[29], have complex structures. Using automatic XML2TTCN-3 converter simplifies the development of the TTCN-3 test data largely.

6 Testing MBBMS Security Mechanism

Security in mobile broadcast service is important to transmit data from server to a set of users. 3GPP MBMS security architecture has been proposed to provide the mechanism of authentication, key distribution and data protection for a MBMS User Service[30]. MBBMS system makes use of 3GPP MBMS security architecture in service/ user management, key management and content delivery. This security architecture, for its complexity, faces many challenges in testing technology. In this section, we propose an integrated testing method to hierarchical MBBMS security architecture.

6.1 MBBMS Security Architecture

Hierarchical MBBMS security architecture is shown in Figure 11.

Firstly, by using HTTP Digest AKA (Authentication and Key Agreement) protocol[31], terminal and BSF server in platform execute GBA[23] procedure to establish share keys (Ks) in order to protect the point-to-point communication between terminal and platform. From Ks, two kinds of keys, MRK (MBMS Request Key) and MUK (MBMS User Key)[30], can be derived both in terminal and platform according to the same algorithm.

Secondly, the interactions of service/user management process between terminal and NAF server should be protected by HTTP digest authentication as defined in RFC 2617 [32]. In HTTP digest authentication, MRK will be used as password.

Thirdly, in the process of key management, MSK (MBMS Service Key) [30] is generated in platform and delivered to terminals. In this delivery, MIKEY (Multimedia Internet KEYing) [33] protocol message is transmitted to carry encrypted MSK and MUK is used to encrypt MSK to protect it.

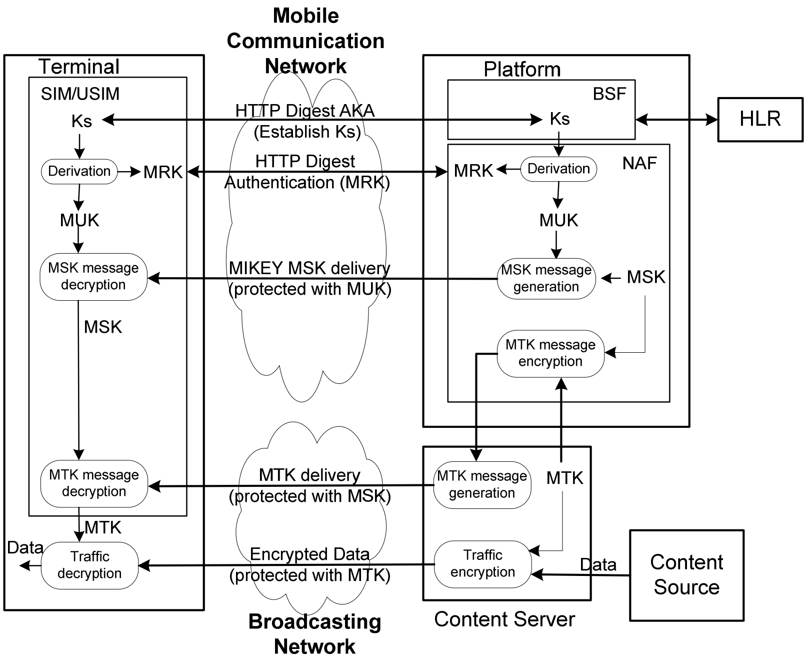


Fig. 11. MBBMS Security Architecture[30,6]

Fourthly, MTK (MBMS Traffic Key) [30] is generated in the content server, and the encrypted MTK with MSK is delivered to terminal in the broadcast network. At last, the data from content source can be encrypted with MTK and transmitted in the broadcasting network securely.

In this security architecture, mobile communication network will be used in GBA procedure, HTTP digest authentication and MSK delivery. Since the test tool focuses on the interactions between terminal and platform in the mobile communication network, we only consider these three mechanisms in this paper.

6.2 Testing Method of MBBMS Security Mechanism

Since various MBBMS security mechanisms are related to each others very closely as shown in Figure 11, we present an integrated testing method in this paper. The framework of testing method is shown in Figure 12. We use the specific test cases to test the three levels of MBBMS security mechanisms and the three kinds of test cases can be generated from the MBBMS HMSC models. In this framework, two configuration files are used: static config file contains some static configurable parameters such as IP-Port information and the parameters of GBA algorithms, and dynamic config file contains the dynamically generated parameters in GBA procedure such as the share key Ks. TTCN-3 external function mechanism[9] is used to implement complex GBA related algorithms and HTTP Digest Authentication algorithms in the test adapter module. In this way, complex calculations in TTCN-3 test suite can be avoided.

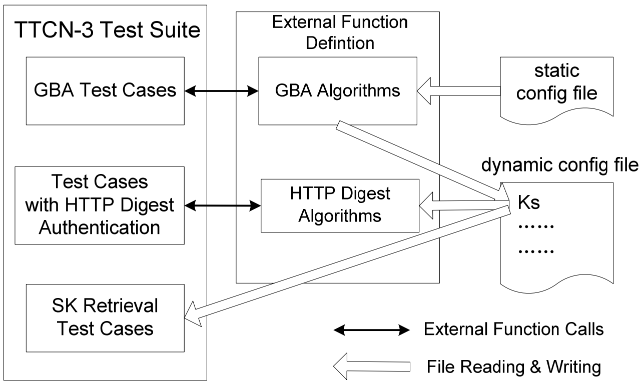
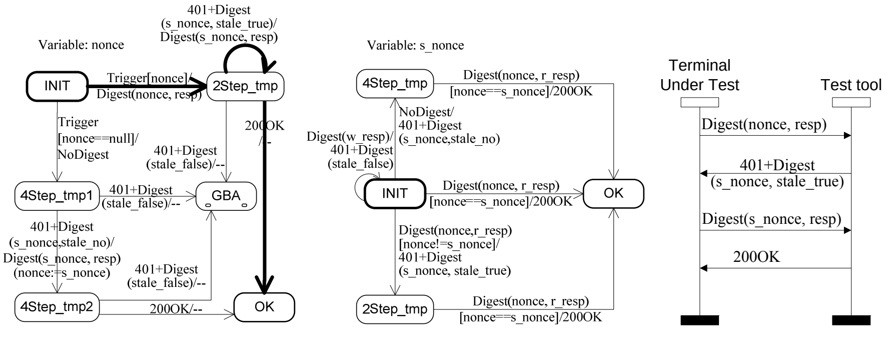


Fig. 12. The framework of testing methods to MBBMS security architecture

To test HTTP Digest Authentication mechanism, we use EFSM (Extended Finite State Machine) to model the protocol behaviors of both terminal and platform. Figure 13 (a) and (b) show the EFSM models of terminal and platform respectively. The INIT state is the initial state of both the two EFSMs. The OK state means the successful ending of HTTP Digest Authentication. From EFSM models, test sequences can be generated by using existing EFSM based test generation method. For example, Figure 13(c) shows an example of test sequence generated from Figure 13(a), which corresponds to the heavy lines of Figure 13(a). This test sequence is easy to be mapped to a TTCN-3 test case, which we have implemented in the test tool.



(a) EFSM model of terminal (b) EFSM model of platform (c) A test case example

Fig. 13. EFSM models of HTTP Digest Authentication

We integrate the security testing supports into the test adapter and encoding/decoding modules as follows in Figure 14. The test adapter module contains two threads, the main thread and the receiving thread, which are responsible for sending and receiving test data respectively. In encoding/decoding module, gzip function is integrated to support gzip compression and decompression, which will be used in service guide delivery[29]. Calculating center is also implemented to provide some external function definitions, mainly about HLR/AuC (GBA) related algorithms and HTTP Digest Authentication algorithms.

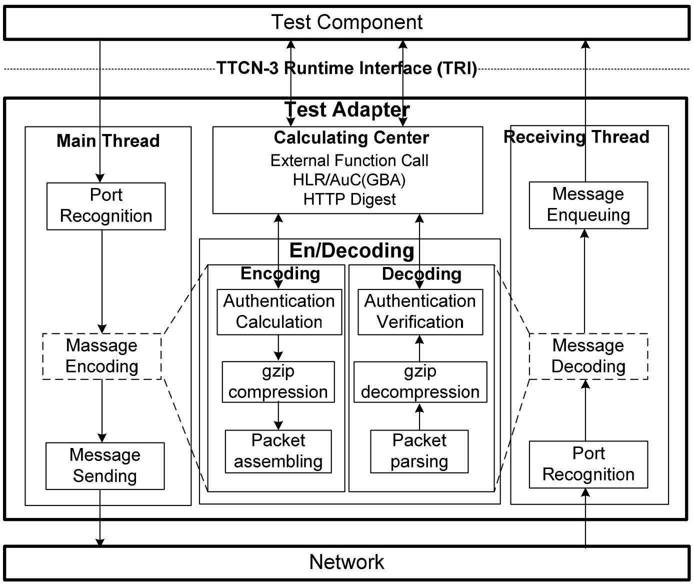


Fig. 14. Test adapter and Encoding/Decoding modules structure

7 Conclusions

In this paper, we study the conformance testing methodology of MBBMS. Many challenges are faced in MBBMS testing, for its distributed nature, complicated XML data and security mechanism. We integrate various techniques in the testing methodology. For the design framework of MBBMS conformance test suite, we present a semi-automatic XML test data generation method of TTCN-3 test suite and use HMSC model to help the design of test suite. In addition, we also propose an integrated testing method for hierarchical security architecture of MBBMS. We have implemented the testing tool based on a commercial TTCN-3 tool telelogic tester [34] and this testing tool has been used in industrial level's testing.

Currently, only one test component, i.e., MTC (Main Test Component), is used in test cases. In the future work, distributed test architecture with multiple test components can be used to enhance the testing capability. Remote testing for terminals can be another possible future direction.

Acknowledgments. This work is partially supported by the National Natural Science Foundation of China under Grant (No. 60572082), Young Faculty Research Funding of Ministry of Education of China (No. 200800031063) and National Key Technology R&D Program of China (No. 2008BAH37B04). The authors thank Yong Li, Wei Zhang, Jing Wang, Hui Wang, etc., for their discussions and helps.

References

1. 3GPP TS 23.246(v9.0.0): Multimedia Broadcast/Multicast Service (MBMS); Architecture and Functional Description (Release 9) (March 2009)
2. 3GPP TS 26.346(v8.2.0): Multimedia Broadcast/Multicast Service (MBMS); Protocols and Codecs (Release 8) (March 2009)
3. Hartung, F., Horn, U., Huschke, J., Kampmann, M., Lohmar, T., Lundevall, M.: Delivery of Broadcast Services in 3G Networks. IEEE Transactions on Broadcasting. 53(1), 188--199 (2007)
4. Open Mobile Alliance: Mobile Broadcast Services Architecture (OMA-AD-BCAST-V1_0-20090212-A) (February 2009)
5. Open Mobile Alliance: Mobile Broadcast Services (OMA-TS-BCAST_Services-V1_0-20090212-A) (February 2009)
6. Zhang, H., Chang, J., Tang, J., Yang, Z.: Analysis and Design of Mobile Broadcast Business Management System. Telecom Engineering Techniques and Standardization. 2, 1--4 (2008) (In Chinese)
7. ISO/IEC: ISO/IEC 9646: Information technology, open systems interconnection, conformance testing methodology and framework. Geneva, Switzerland (1991)
8. Grabowski, J., Hogrefe, D., Rethy, G., et al: An introduction to the testing and test control notation (TTCN-3). Computer Networks. 42(3), 375--403 (2003)
9. ETSI: ETSI standard ES 201 873-1 V3.4.1: The Testing and Test Control Notation version 3; Part 1: TTCN-3 Core Language. European Telecommunications Standards Institute (ETSI), Sophia-Antipolis, France (September 2008)
10. Floch, A., Roudaut, F., Sabiguero, A., et al: Some lessons from an experiment using TTCN-3 for the RIPng testing. In: Khendek, F., Dssouli, R. (eds.) Proc. of the 17th IFIP International Conference of Testing of Communicating Systems (TestCom 2005). LNCS, vol. 3502, pp. 318--332. Springer, Heidelberg (2005)
11. Noudem, F., Viho, C.: Modeling, verifying and testing mobility protocol from SDL language. In: Prinz, A., Reed, R., Reed J. (eds.) Proc. of the 12th International SDL Forum (SDL 2005). LNCS, vol. 3530, pp. 198--209. Springer, Heidelberg (2005)
12. Dibuz, S., Szabo, T., Torpis, Z.: BCMP performance test with TTCN-3 mobile node emulator. In: Groz, R., Hierons, R.M., (eds.) Proc. of the 16th IFIP International Conference of Testing of Communicating Systems (TestCom 2004). LNCS, vol. 2978, pp. 50--59. Springer, Heidelberg (2004)
13. Din, G., Tolea, S., Schieferdecker, I.: Distributed load tests with TTCN-3. In: Uyar, M.Ü., Duale, A.Y., Fecko, M.A. (eds.) Proc. of the 18th IFIP International Conference on Testing of Communicating Systems (TestCom 2006). LNCS, vol. 3964, pp. 177--196. Springer, Heidelberg (2006)
14. Yin, X., Wang, Z., Jing, C., Shi, X.: A TTCN-3-based protocol testing system and its extension. Science in China Series F: Information Sciences. 51(11), 1703--1722 (2008)
15. Werner, E., Grabowski, J., Troschutz, S., Zeiss, B.: A TTCN-3-based Web Service Test Framework. In: Software Engineering 2008 Workshops, pp. 375--382. (2008)

16. Stepien, B., Peyton, L., Xiong, P.: Framework testing of web applications using TTCN-3. International Journal on Software Tools for Technology Transfer. 10(4), 371--381 (2008)
17. Lin, Y.-B., Liang, C.-F., Chen, K.-H., Liao, H.-Y.: NTP-SIOT: a test tool for advanced mobile services. IEEE Network. 21(1), 21--26 (2007)
18. Lin, Y.-B., Wang, C.-C., Lu, C.-H., Hsu, M.-R.: NTP-PoCT: a conformance test tool for push-to-talk over cellular network. Wireless Communications and Mobile Computing. 8(5), 673--686 (2007)
19. Baker, P., Bristow, P., Jervis, C., King, D.J., Mitchell, B.: Automatic Generation of Conformance Tests from Message Sequence Charts. In: SAM 2002. LNCS, vol. 2599, pp. 170--198. Springer, Heidelberg (2002)
20. Ebner M.: TTCN-3 Test Case Generation from Message Sequence Charts. In: IEEE International Symposium on Software Reliability Engineering (ISSRE 2004) Workshop on Integrated-reliability with Telecommunications and UML Languages (ISSRE 2004: WITUL 2004) (2004)
21. ETSI: ETSI standard ES 201 873-9 V3.3.1: The Testing and Test Control Notation version 3; Part 9: Using XML schema with TTCN-3. European Telecommunications Standards Institute (ETSI), Sophia-Antipolis, France. (July 2008)
22. Using XML Schema with TTCN-3, http://wiki.openttcn.com/media/index.php/OpenTTCN/Working_documents/Reviews/Using_XML_Schema_with_TTCN-3
23. 3GPP TS 33.220(v8.6.0): Generic Authentication Architecture (GAA); Generic bootstrapping architecture (Release 8) (March 2009)
24. ITU-TS Recommendation Z.120: Message Sequence Chart (MSC). ITU-TS, Geneva (2004).
25. Mauw, S., Reniers, M.A.: High-level message sequence charts. In: Cavalli, A.R., Sarma A. (eds.) Proc. of the 8th International SDL Forum (SDL 1997). pp. 291--306. Elsevier (1997)
26. Genest, B., Hélouët, L., Muscholl, A.: High-Level Message Sequence Charts and Projections. In: Amadio, R.M., Lugiez, D. (eds.) 14th international conference on concurrency theory(CONCUR 2003). LNCS, vol. 2761, pp. 311--326. Springer, Heidelberg (2003).
27. Genest, B., Muscholl, A., Seidl, H., Zeitoun, M.: Infinite-state high-level MSCs: Model-checking and realizability. Journal of Computer and System Sciences. 72(4), 617--647 (2006)
28. XMLSpy, http://www.altova.com/products/xmlspy/xmlspy.html
29. Open Mobile Alliance: Service Guide for Mobile Broadcast Services(OMA-TS-BCAST_Service_Guide-V1_0-20090212-A) (February 2009)
30. 3GPP TS 33.246(v8.3.0): 3G Security; Security of Multimedia Broadcast/Multicast Service (MBMS) (Release 8) (March 2009)
31. Niemi, A., Arkko, J., Torvinen, V.: Hypertext Transfer Protocol (HTTP) Digest Authentication Using Authentication and Key Agreement (AKA). IETF RFC 3310. (September 2002)
32. Franks, J., Hallam-Baker, P., Hostetler, J., et al: HTTP Authentication: Basic and Digest Access Authentication. IETF RFC 2617. (June 1999)
33. Arkko, J., Carrara, E., Lindholm, F., et al: MIKEY: Multimedia Internet KEYing. IETF RFC 3830. (August 2004)
34. Telelogic tester, http://www.telelogic.com/Products/tester/index.cfm