A Security Based Multimedia Web Service Framework
Supporting Multimedia Content Streaming

Department of CSE
Christ College of Engg. & Tech.
Puducherry, India

Abstract—At Present web services standards is not well supported the transfer of streaming data. To incorporate multimedia streaming sustain in the Web services domain, this manuscript introduces a new multimedia streaming Web services framework for the transfer of streaming multimedia content with security in the form of XML encryption and Digital Signatures. Initially, the framework incorporates an accomplishment of a query service for publishing a depiction of the multimedia content specifically input to or output from a multimedia Web service. This query service is precise in an expansion of WSDL. Utilizing MPEG-7 metadata, content descriptions could be queried previous to the invocation of a multimedia Web service. Next, two new MEPs and their SOAP HTTP bindings are formed for the swap of streaming data between two SOAP endpoints. The implementations of these new MEPs utilize the MIME Multipart/Related structure and MTOM packaging when transmitting the multimedia packets as SOAP messages. To diminish the transfer overhead introduced by the packaging method, this manuscript analysed expansively the application of a variety of compression schemes for the SOAP messages in addition to the packaging of the binary packet data. Analysis illustrate that the proposed framework can attain a performance similar to a simple HTTP multimedia streaming method.

Keywords—Binary XML, Multimedia streaming, Service oriented architecture, Web services, XML Encryption, XML Digital Signature.

I. INTRODUCTION
Service oriented architecture (SOA) is a set of principles related to the growth of systems using abilities offered by services in a distributed environment. SOA give emphasis to the reusability, growth and interoperability of solutions developed as a composition of the distributed services. These facilitate better scalability and manageability, and a reduced amount of cost for the development of large-scale systems such as multimedia systems. One of the familiar implementations of the SOA representation is Web services. Web services permit machines or software to communicate over the network on different platforms in a consistent manner, by means of a set of XML-based standards. Data is usually encapsulated and exchanged using SOAP, which is a lightweight protocol for transmitting structured information in an XML-based format. In the Web service architecture a specified task can be attained by consuming a set of services from different service providers.

Multimedia systems are typically monolithic systems that are expensive and challenging to maintain and develop. Given the advances of SOA the development of complex multimedia systems should be expected to reap the benefits of using SOA. Using SOA concepts a complex multimedia system can be built from the composition of multiple small reusable components. In the Web services architecture these components are multimedia Web services, which process and/or provide multimedia data as a service. As opposed to a monolithic system this approach sup-ports reusability of the individual services and flexibility in the development of a multimedia applications. In particular, real-time applications such as multimedia streaming systems are not well supported by Web services standards. A key problem is the lack of support for the transfer of streaming data between two Web service end points [1].

Because more and more business organizations have been accepting Web services technology to publish their distributable data to make their services accessible to more other organizations, it is achievable that the data to be available take account of multimedia information, e.g. audio and video. Due to the reality that multimedia data displays exclusive features that need specific handling, it is essential to scrutinize the existing Web services techniques to better support multimedia Web services. When the web service framework is applied to maintain a multimedia Web service, two major factors influence the success of a multimedia Web service: (1) the transfer of multimedia information over the Web, and (2) to manage the composite devices for multimedia contents. Multimedia content caching and streaming are two vital resolutions to the first problem. Web caching at a service provider site can mostly increase the service provider’s capability to sustain a huge quantity of service requesters. The streaming paradigm facilitates a media file be take part in at the service requester’s site as it is being transferred over the Web; as a result, the burden of the service provider can be improved. concerning the second
issue, at the service requester side, multimedia data is usually divide over multiple devices with suitable multimedia capabilities [4].

A. Objective

This paper proposes a novel Web services framework supporting multimedia streaming. The proposed framework permits multimedia Web services to be published, composed and consumed just like other Web services in secured way. Development of great multimedia systems can be attained using a composition of services contained by the Web services framework. This allocated component-based development has the benefits of the SOA standard such as maintainability, reliability, reusability and scalability of the systems and their components. To maintain multimedia streaming in the Web services framework numerous basic problems have to be examined, which are the major focal point of this manuscript.

1) Multimedia metadata description: To sustain the interoperability of services the service interfaces have to be finely explained and depicted. These explanations and descriptions are characteristically enclosed inside WSDL which are published in service registries. Thus these WSDL interface descriptions could also be applied to multimedia Web services. Nonetheless, XML Schema, the language used to describe data in WSDL, is not enough for the description of multimedia resources. For the reason that, a dedicated metadata scheme [3] such as MPEG-7 is used so that an improved explanation of multimedia data can be make possible.

One of the problems of with multimedia metadata schemes is to the metadata formed by these schemes are moderately better than the XML Schema of regular Web service operation. Additionally, Ever since the inputs and outputs of various Web service operations may have their own separate metadata the development of the size of WSDL will make it uncontrollable if all metadata information is stored within WSDL. To solve this problem the proposed framework provides a simple query mechanism so that service requesters can obtain multimedia metadata proficiently.

2) Streaming Data Content Transfer between Two Web Services Endpoints: In multimedia applications, while the multimedia data is bulky or includes live data, streaming is characteristically make use of the delivery method. Though, present Web services standards do not sustain streaming data transfer among two Web service endpoints. That involves multimedia Web services have to transfer multimedia data in its entirety, which diminishes the efficiency of the application and every so often cannot be achieved because the total size of the data is not well-known at the establishment.

A variety of studies [3][4][5][6][7][8] have examined the improvement of multimedia applications in the perspective of SOA. These studies reflect on different traditions to separate a multimedia application into a control path and a data path. Messages for controlling the course of the application are transmitted using typical Web services implementation. The multimedia data is transmitted in another channel such as raw TCP connection, which are not part of the Web services architecture. One of the advantages of these approaches is that they can take advantage of some readily available technologies for multimedia transfer and at the same time a service-oriented development can be employed. Though, the main inadequacy of these approaches is that, the need to maintain external entities, the interoperability, maintainability and scalability of the application are compromised. Furthermore, various WS-* specifications such as WS-Security are not valid to the multimedia data, which delays the extensibility of the fundamental multimedia system.

So as to have a streaming data transmit compatible with the WS-* extensions the data has to be passed within the XML infosets of the SOAP messages. However, streaming data by description is group of time-dependent small data packets. To transmit these small packets the Web services communication desires to have the ability to transmit multiple objects. Ruth et al. [9] recommended that a proxy service can be used to include a Single-Request/Multiple-Response message exchange prototype in Web services. Though, this approach is not intended for time-critical message transformation. When several messages are needed a separate request has to be completed for every message, which is time-consuming. In this paper the proposed structure eliminates the dependency on any external entity by streaming multimedia data encapsulated in SOAP messages. A grouping of MIME structure and MTOM attachment scheme is employed to transmit the streaming data in an only one connection.

This transmission arrangement is generated in a novel HTTP SOAP binding. Since the streaming data is transmitted inside the Web services architecture, the multimedia Web services in the proposed framework are able to publish, compose and consume like any other Web services [1]. In addition, WS extensions can be applied to the transmitted data because the data is encapsulated in the XML infoset. Here WS-Security in the form of any other Web services [1].

Before I give explanation what WS-Security is, I consider that it is significant to realize why WS-Security exists in any way. A lot of people new to Web services perceive SOAP as a method to exchange messages between two endpoints over HTTP. Over HTTP, one could authenticate the caller, sign the message, and encrypt the contents of the message. This creates the message secure in some dimensions: the caller is well-known; the receiver of the message can make sure that the message did not change in transfer, and entities scrutiny the wire traffic cannot form out what data is being exchanged. Intended for those looking at SOAP messaging to resolve bigger problems, though, HTTP-based security only isn't sufficient. Many of the superior problems occupy in sending the message all along a path more complex than request/response or over a transport that does not engage HTTP. The identity, integrity, and security of the message in addition to the the caller need to be conserved over multiple hops. More than one encryption key possibly will be used along
the route. Trust domains would be crossed. HTTP as well as its security mechanisms just addresses point-to-point security. More difficult solutions require end-to-end security baked in. WS-Security addresses how to preserve a secure context over a multi-point message path. The following figure describes the above explained mechanism.

Figure 1: SOAP Envelope with WS Security

3) Ineffectiveness of XML-based Data Communication: Web services, for the reason of their dependence on XML-based protocols, have experiences from performance issues for time-critical applications. Heinzl et al [11].

The videoconferencing system talk about by Oh et al. [12] used SOAP messages to summarize and transmit a sequence of pictures, which is transferred as frames of a video stream. To get better efficiency of the system the authors used Fast Web Services [13], a binary encoding scheme for SOAP messages based on ASN.1, when transmitting the SOAP messages. Nevertheless, in their investigation they do not have a thorough comparison of performances based on the use of the ASN.1 representation.

The transmission overhead of the streaming data transfer contains both the transmitted SOAP message content and the overhead in the packaging method. If a suitable compression scheme is applied to single or together of these the efficiency of the data transfer will be enhanced. In this manuscript a complete investigation has been approved out on the application of compression schemes to the streaming data transmit in the proposed framework. A wide range of compression schemes have been experienced. They take in generic text compressors, XML compressors, SOAP compressors and binary XML representations. The compression techniques are applied to a particular package, which contains the packaging overhead, the SOAP message and the binary data, of a packet. Certain the results, the finest compression scheme is applied to a multimedia Web service in the proposed framework. The performance analysis of the proposed framework, mutually with and without using a compression scheme, is compared in opposition to a simple HTTP streaming method.

II. RELATED WORK

Yu and Lin [3] investigation make use of a quality of service (QoS) broker to build decisions in service selection and service composition in multimedia Web services. During this work, multimedia data is transferred throughout a separate socket connection established among the client and the server. The authors of this work think it is best to use a separate socket connection as the size overhead is vast under the current Web service standards. Zhang et al proposed a SOAP-oriented component-based framework for multimedia Web service [4][5]. In this work a multimedia Web service is divided into a control flow and a data flow. The control flow is transferred via ordinary Web services while the data flow is controlled by the offered multimedia infrastructure. The study proposed a SOAP enhancement so that multimedia data can be alternatively transferred using SOAP. The enrichment contains the use of ‘boxcarring’ and batching of messages.

Though, as the transfer of each part of a large chunk of multimedia data needs a separate Web service request the transfer may be too costly and ineffective when streaming multimedia is transmitted using one separate request per packet.
Decneut et al converse the use of multimedia Web services on varied multimedia environments [6]. The work is targeted at the allocation of multimedia data for a diversity of client configurations and network conditions. The transformation of multimedia content is attained using the now outdated DIME attachment scheme over HTTP, TCP or UDP channels. A multimedia Web service is merely allowed to send multimedia content in one single package. That means streaming data is not sustained, or it have to be transferred as if the packets are a large chunk of multimedia content.

Two studies [7][8] seems into the determination of service workflow and system performance in multimedia applications provided that personalized adaptation of multimedia content. In these studies the multimedia Web services are responsible for the control, notification and monitoring of multimedia streams whereas the multimedia content is exchanged in a separate TCP connection.

III. DESIGN OF THE SECURITY BASED MULTIMEDIA STREAMING WEB SERVICE FRAMEWORK

A multimedia Web service must be able to be published, discovered and consumed like other Web service. Though there is a main difference between a multimedia Web service and an ordinary Web service in the means that the data is transferred between the two end-points of a service. A multimedia Web service engages multimedia content. If the size of the multimedia content is small it can be distributed as a binary attachment inside the response. If the size of the content is huge it will not be logical to transfer it in its entirety in the response. For huge multimedia content it is suitable to have a streaming ability between the two endpoints of the multimedia Web service.

In this paper, the design of our proposed multimedia Web service framework has two main parts. First, in the publishing and discovery of a multimedia Web service a description of the input and output of multimedia content has to be supplied by the service provider. This is to ensure that the service consumers understand the expected qualities and information of the content that they need to send to or receive from the provider. In a multimedia Web service this is necessary because different devices may require different delivering qualities. It is highly desirable that a service consumer can select a multimedia Web service with the most suitable qualities for the underlying device. Second, after selecting the appropriate multimedia Web service the multimedia content has to be transferred between the service provider and the service consumer. In the case of streaming multimedia the two endpoints of the service connection have to be capable of transmitting the streaming multimedia data between themselves. The proposed framework provides a mechanism for the transmission of streaming multimedia between two service endpoints with security.

In the proposed framework the multimedia streams are sent in their entirety. This does not mean the service requester has to wait for the whole stream before processing its data. On the contrary, the service requester can process any chunk of data while the entire stream is arriving at the destination. This paper does not consider streaming control such as RTSP. Nevertheless, an application-specific Web service can be added on top of the multimedia Web service, which can be used to control the underlying streaming data transfer transparently. The main advantage of this approach is that the Web services are open, loosely coupled way and it is primarily through web services that the visions and promises of SOA are realized in its fullness then this will provide the scalability and reliability of the provided services.

XML Encryption and XML Signature illustrate the ways of encrypting and signing the contents of XML messages. XML Canonicalization explains ways of building the XML ready to be signed and encrypted. What WS-Security adds to obtainable specifications is a framework to set in these mechanisms into a SOAP message. This is made in a transport - neutral manner.

WS-Security describes a SOAP Header element to bring security-related data. If XML Signature is utilized, this header can have the information defined through XML Signature that communicates how the message was signed, the key to be used, and the ensuing signature value. Similarly, if an element in the message is encrypted, the encryption information for instance that communicated by XML Encryption can be contained inside the WS-Security header. WS-Security does not identify the format of the signature or encryption. As an alternative, it specifies how an individual would embed the security information lay out by other specifications in a SOAP message. WS-Security is mostly a specification for an XML-based security metadata container. WS-Security as well provides a place to offer binary authentication tokens for example Kerberos Tickets and X.509 Certifications: BinarySecurityToken.

The XML Encryption protocols specify that element or the entire body of a SOAP message can be encrypted. While XML encryption is used, an encryption algorithm is run next to the portion of the XML document being encrypted and this XML data is substitutes with the resulting encrypted data in an EncryptedData element. WS-Security, in which is based on XML Encryption, added stipulates that while XML encryption is used in a SOAP message, the location of the ensuing EncryptedData element is referenced from the Security header element. If many elements in the body of the message are encrypted, after that all is referenced by a separate ReferencedData element in the ReferenceList. Designed for an EncryptedData element, a suggestion at the encryption key used can be specific in the KeyInfo element, and the encryption algorithm can be used in the EncryptionMethod element. The KeyInfo element is defined in the XML Signature specification.

While a message is signed, it is almost not possible to interfere with the message. Message signing do not protects the message itself from external parties considering its contents. By means of the signature, the receiver of the SOAP message can recognize that the signed elements had not changed en route. You be supposed to use XML Signature to sign messages when possible. XML Signature previously handles an amount of the tougher items to form out. WS-
Security basically explains how to employ signing to establish that the message has not changed. All the WS-Security mechanisms mentioned above offers a way to sign the message thus that you can be certain of two things:

- The user identified by the X.509 certificate, ticket signed the message.
- The message has not been tampered with since it was signed.

Every of the methods make available a secret that can be utilized to sign the message. X.509 lets the sender to sign the message by their private key. The following figure describes the use of security tokens, signatures, and encryption.

Figure 2: Security based Multimedia Streaming Web Service Framework

IV. CONTENT DESCRIPTION FOR MULTIMEDIA WEB SERVICES

In a multimedia Web service it is significant to offer an explanation of the original multimedia content. Data such as the streaming format, compression codec, resolution and frames per second are aspects that have to be considered while a service consumer attempts to decide on the best possible multimedia service between a set of published services. Additionally, during service composition, the multimedia information is necessary in the determination of the service workflow. For instance, an added service such as a transcoding service may be requisite when the ideal multimedia Web service in terms of the preferred format and qualities is not accessible.

Multimedia content can be described by MPEG-7 [14]. MPEG-7 metadata includes an extensive range of languages for the explanation of different aspects of multimedia data. Since MPEG-7 is an XML-based language it can be simply inserted in the Web service stack. In our proposed framework an easy method is provided to query the depiction of the multimedia Web service using an extended WSDL of the service [1].

A. Metadata Query Services
MPEG-7 metadata know how to be used to depict the input and output multimedia content of a multimedia Web service. A clear-cut approach is to insert the metadata contained by the description of a WSDL operation. But, MPEG-7 metadata is moderately bulky compared to the size of a WSDL file. The size of a WSDL file will enlarge significantly if MPEG-7 metadata are inserted. As well, the multimedia content description may be modifying from time to time. If an item of metadata is altered the corresponding Web service has to evaluate and publish its WSDL to all interrelated service registries.

As an alternative of counting the entire MPEG-7 metadata an associated metadata query service is provided for each operation of a multimedia Web service. The exclusive purpose of the metadata query service is to offer a channel for the allocation of the metadata associated with the multimedia Web service. As referred from [1] For example, Fig. 2 shows a description of an operation ‘GetMedia’ together with the specification of its metadata query service ‘Get-MediaDesc’.

In addition, the metadata query service provides decision makers such as service registries and service brokers the capability to develop appropriate algorithms so that multimedia Web services can be selected based on the information of the multimedia content [1].

### V. STREAMING DATA TRANSFER

In the proposed framework streaming data transfer is accomplished by refining the approach described in [15] and [16]. This engages extending two present Web service standards: the message exchange prototype (MEP) and the SOAP binding of a MEP [17]. The method that Web service messages are swap between two service end-points is restricted by a MEP. To maintain streaming data transfer, two novel types of MEP are created for handling streaming data. This new MEPs can follow by used via a communication protocol such as UDP or TCP. Especially, this paper considers the application of the new MEPs to HTTP, the mainly used protocol for Web services [1].

#### A. Message Exchange prototype

There are two MEPs defined in the SOAP specification at present, the request-response MEP and the response MEP. Together of these MEPs employ a single request and a single response structure. In this study two new MEPs are formed for two types of multimedia streaming Web services. A single request-multiple response (SRMR) MEP provides a Web service delivering streaming data in response to a single request. A multiple request-multiple response (CRMR) MEP is developed for a Web service receiving a stream of data as input and delivering one more stream of data as output with key.

1) **SRCR MEP:** The difference in the novel MEP compared to the request-response MEP is means that data is transferred once communication is established. Following the request is transferred from the requesting SOAP node to the responding SOAP node; together the nodes are locked in looping states in which the streaming data transfer takes.
place with security in the form of XML encryption and Digital signature. Each chunk of data is transmitted and received separately and continuously between the two nodes. This process lasts until the finish of the streaming data transfer.

2) **CRCR MEP**: In the CRCR MEP, although a particular request is through the requesting node the request body contains the request information in addition of a stream of data, which can be a multimedia data stream as an alternative of a single chunk of information. To generate a state diagram for this MEP the request is dividing into the request information and its associated data stream. Fig. 4 represents the CRCR MEP. The processing states handle the constant transmission of the streaming data to and from the SOAP nodes. In these states the sending and receiving processes happen in parallel with security. They can run asynchronously or one later than another depending on the characteristics of the provided Web service.

### B. SOAP HTTP Binding

In segment 5.1 two new MEPs are cleared for multimedia Web services. The implementation of a MEP in a Web service program is illustrated by a SOAP binding. The present SOAP binding in the Web service standards is the SOAP HTTP binding and so it is the main normally used protocol in the implementation of Web services.

In the recent SOAP HTTP binding a SOAP message is place in the HTTP entity body in every of the requests and responses of the service. This two SOAP message contains the request and the equivalent response of a Web service. A SOAP message in each the request or response can merely be used to store one single item, though the item is able to any type such as a number or a data record. Nevertheless, as exposed in Fig streams of multimedia data are sent and/or received by the SOAP nodes in the recently created MEPs. That way using a single SOAP message is not enough for handling these two MEPs. The subsequent sections first illustrate the packaging method used for the data packets. Then new SOAP HTTP bindings are launched for transferring streaming data in the SRCR MEP and the CRCR MEP.

#### 1) SOAP Packaging for a Multimedia Stream:

Multimedia streaming is a constant transmission of data packets. To transmit a multimedia stream from one SOAP endpoint to a further the stream has to be encapsulated in various form of SOAP packaging. As each data packet represents one piece of information within the stream a SOAP message is used to enclose one single packet during the transmission. To do that two issues have to be resolved. First, the data packet enclosed by a SOAP message requires a way to identify itself. This is because a SOAP message can be transmitted in a variety of bindings and therefore SOAP messages can arrive at a SOAP endpoint asynchronously. If a SOAP message can be identified the SOAP processor will be able to forward the SOAP message to an appropriate handler. In this work the mechanism of Security and addressing is achieved by using Web Services Security (WS-Security) and Web Service addressing (WS-Addressing) [18]. WS-Security and WS-Addressing allows a SOAP message to carry header information with the encryption and digital signature such as the destination of the message and the identity of the data packet with Encryption message and Digital Signature message. Next, SOAP is an XML-based protocol. So as to means binary data, for example multimedia packets cannot be openly inserted into the body of the message. However, binary data can be included in a SOAP message with binary-to-text encodings such as base64 encoding.

![Figure 4: The state Transition diagram of the compound request –compound response message exchange prototype](image-url)
The drawback is that a binary-to-text encoding brings a significant increase in data size and processing time. Otherwise binary data can be transferred together with a SOAP message using the SOAP Message Transmission Optimization Mechanism (MTOM) [19]. In this example the identity of the packet is ‘da-ta5608@example.org’, the related message, which can be the request message of the multimedia stream, msg2056@example.org’.

The associated action of the Web service is http://example.org/getmedia’. As referred from the MTOM Serialization of a data packet of a multimedia stream in [1]. WS-Addressing is used to indicate the destination and the identity of the packet.

2) SOAP HTTP Binding for the SRCR MEP: In distinguish to the request-response MEP, the SRCR MEP sends the response from the requesting SOAP node to the requesting SOAP node like a stream of data in its place a single item. This stream of data is transferred in the looping state in the responding node as shown in Fig. 4. Within the request-response MEP its SOAP HTTP binding transfers the response as a SOAP message in the body of a HTTP response.

If a multimedia stream by way of multiple items is to be endlessly transmitted along a HTTP response the SOAP HTTP binding will require to be modified. In this assessment a MIME Multipart/Related pack-age is used to place the multimedia stream in a HTTP response body. Prior to including attachment content in a signature reference hash calculation, that MIME attachment is supposed to be canonicalized. The motive is that signature verification requires an identical hash of content as while signing occurred. The result is shown in Fig. 6. Using this association the response can take as many SOAP messages as required. The requesting SOAP node is then in charge to read each SOAP message constantly when the SOAP messages are expected from the responding node, as shown in the looping state in the requesting SOAP node in Fig. 4.

The subsequent example message demonstrates the use of security tokens, signatures, and encryption.

```xml
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope
 xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
 <wsu:Timestamp>
  <wsu:Created>
   2002-08-22T00:26:15Z
  </wsu:Created>
  <wsu:Expires>
   2002-08-22T00:31:15Z
  </wsu:Expires>
 </wsu:Timestamp>
 <wsse:Security soap:mustUnderstand="1">
  <xenc:ReferenceList>
   <xenc:DataReference URI="#EncryptedContent-f6f50b24-3458-41d3-aac4-390f476f2e51" />
  </xenc:ReferenceList>
  <xenc:ReferenceList>
   <xenc:DataReference URI="#EncryptedContent-666b184a-a388-46cc-a9e3-06583b9d43b6" />
  </xenc:ReferenceList>
 </wsse:Security>
</soap:Envelope>
```
3) SOAP HTTP Binding for the CRCR MEP: The CRCR MEP engages two channels of data streaming. Individual is from the requesting SOAP node to the SOAP responding node and one more one in the opposite direction. The SOAP HTTP binding for the SRCR MEP depicted in the earlier section permits a data stream to be expected by a message, Rosu proposed A st body, again using the MIME Multiple/Related package. However, HTTP is afer mechanism can be employed when transmitting
ciency of SOAP transmission
text compressors by separating and compressingely a small part of the transmitted message.
ult. The valuation is based on
-ot of namespaces used by a
E Multi
tinde of a single packet, which contains the SOAP message, the packet data and a
message, Rosu proposed A st body, again using the MIME Multiple/Related package. However, HTTP is afer mechanism can be employed when transmitting

ciency of SOAP transmission
text compressors by separating and compressingely a small part of the transmitted message.
ult. The valuation is based on
-ot of namespaces used by a
E Multi

VI. COMPRESSION PROCESSES OF SOAP MESSAGES
The SOAP HTTP binding described in Section 5.2 introduces a considerable amount of transfer overhead to the multimedia transfer. The overall structure of the streaming data transmission mechanism described in Section 5.2.2 for the SRCR MEP, as referred in the [1] in the case when MTOM is used, is shown in Fig. 7. A first level of MIME Multi-part/Related structure includes the SOAP messages of the data packets. Per SOAP message a second level of MIME Multipart/Related structure is used to divide the textual SOAP message and the binary attachment of the message. For every MIME packages and headers are required to identify information for example, the length and the type of content related with the package. Other transfer overhead, compared to the unique multimedia stream, is too required when every data packet is transmitted collectively with a SOAP message.

A. Compression Techniques for SOAP Messages
A SOAP message is an XML-based communication protocol. By means of headers and namespace specifications reside in a considerable segment of the entire message, the real payload is merely a small part of the transmitted message. In the perspective of a time critical application, for instance streaming multimedia transmission, the size overhead is enormous compared to other dedicated multimedia transmission protocols. To get better efficiency of SOAP transmission a variety of compression and encoding techniques can be used.

One of the simplest approaches is to utilize a generic text compressor for example gzip, PPM or bzip2 to reduce the size of a SOAP message. Though, generic text compressors do not reflect on the structural information of XML messages, which can be exploited for augmented compression rates. XML compression methods such as XMLPPM [20] and XMill [21] can attain a slightly improved compression rate more than generic text compressors by separating and compressing the content and hierarchical structure. A synopsis of general XML compression techniques is exposed in [22].

SOAP messages, alternatively, have a highly recurring structure. For illustration, a lot of namespaces used by a succession of SOAP messages are potentially an identical. In addition, the structure of the header and payload is probable to be an alike across a similar kind of Web services. To obtain advantage of these redundancies, Werner et al. [23] have established a SOAP message compression using differential encoding. Their idea is to send out only the variation between a SOAP message and skeleton messages generate using the WSDL of the Web service. As an alternative of using a skeleton message, Rosu proposed A-SOAP [24], which is a vocabulary based method for the compression of SOAP messages. A vocabulary of usually used tags is build up incrementally among the sender and the receiver. A SOAP message is then transferred as an encoded entity based on the existing dictionary.

An additional class of techniques encodes XML messages by means of a more compact binary encoded format for example; Fast Infoset [25] and Efficient XML interchange (EXI) [26]. They intend at minimizing the encoding size and at the similar time maximize the processing speed of an XML message. A different advantage of the binary XML techniques is that binary data can be openly encoded in the SOAP message with no need for whichever packaging method such as the MIME package. This as a result it will eliminate the need to employ additional headers for the depiction of the package.

B. Analysis of Compression Techniques
In this segment the performance of a preferred set of compression techniques are calculated when they are applied to the configuration proposed in Section 5.2. The outcome is shown in Table 1 which is referred in [1]. The packet size is assumed to be the volume of the payload when the packet is transmitted in a TCP connection using a MTU of 1500 bytes. The overhead size as a percentage of the packet size is exposed in the last column of the result. The valuation is based on the compression applied to the transmission of a single packet, which contains the SOAP message, the packet data and a

Fig 6: Example of a SOAP Message containing WS-Security information in the SOAP Header and Body
variety of MIME headers. The first two rows have the result when no compression is used on a base64 encoded message and a MTOM package. For all that compression techniques, aside from the two binary XML methods listed at the bottom two rows of the table, the transformation of the packet data is attained using MTOM. For binary XML methods they are capable of directly encode base64 encoded content in their binary form. So they do not have a MTOM header size in their end result.

<table>
<thead>
<tr>
<th>First level MIME Multipart/Related headers</th>
<th>First multimedia data packet</th>
</tr>
</thead>
<tbody>
<tr>
<td>First level MIME boundary</td>
<td></td>
</tr>
<tr>
<td>First multimedia data packet</td>
<td></td>
</tr>
<tr>
<td>Second level MIME Multipart/Related headers</td>
<td></td>
</tr>
<tr>
<td>Second level MIME boundary</td>
<td>The SOAP message of a packet</td>
</tr>
<tr>
<td>Second level MIME boundary</td>
<td>The binary data of a packet</td>
</tr>
<tr>
<td>End of second level MIME boundary</td>
<td></td>
</tr>
<tr>
<td>First level MIME boundary</td>
<td></td>
</tr>
<tr>
<td>Second multimedia data packet</td>
<td></td>
</tr>
<tr>
<td>Second level MIME Multipart/Related headers</td>
<td></td>
</tr>
<tr>
<td>Second level MIME boundary</td>
<td>The SOAP message of a packet</td>
</tr>
<tr>
<td>Second level MIME boundary</td>
<td>The binary data of a packet</td>
</tr>
<tr>
<td>End of second level MIME boundary</td>
<td></td>
</tr>
<tr>
<td>First level MIME boundary</td>
<td></td>
</tr>
</tbody>
</table>

Fig: 7 The SOAP response structure of the streaming data transmission mechanism in the SRMR MEP.

By evaluating the overhead size in Table I it is obvious that the binary XML methods are considerably better than the further compression techniques. As the SOAP-specific compression techniques perform a only some percent better than the general XML compressors, all of them still have to bring the MTOM headers, which engage a main segment of the transfer overhead. With no need to have the MTOM headers the binary XML methods, EXI in particular, are the clear choices in the proposed framework to maintain compression in the transmission. From the proposed System analysis, The WS-Security signed message analysis added three smaller binary components to the original messages, in addition to another hardly any hundred bytes of XML tags. A number of this added binary data was cryptographic hashes and keys. On the whole, the prologue of WS-Security signatures added about 3K bytes to the text-based XML SOAP messages. Analysis showed that the MTOM message sizes were all better than their text-based XML counterparts for the signed messages. The signed message has three miniature binary fields, Digest Value, Digital Signature and Binary Security Token (X.509 certificate) and the reserves coming from not using Base64 encoding are fewer than the fixed additional costs of the MIME headers and XML protocol elements desirable by MTOM.

The encrypted message tests add innovative XML header elements, including some binary-encoded ones, and a binary message body that augments in size as go from simple to complex time critical applications. The text-based message will take all of these binary components as Base64 strings. The encrypted message body is efficiently a single random binary string that will be essentially uncompressible. All of its internal structures and tags had gone as well, captivating away much of the improvement of compression and binary encoding, and send-off only the overhead introduced by Base64 encoding to be eradicated. This is reflect in the analysis results, with most options to text-based encoding contribution up to about 20% reduction in message size, far away from the 89% reduction achieved by compression for the complex message. In review, those tests showed that together the compression and binary encoding obtainable important reductions in messages sizes compared to text-based XML. Compression, binary encoding and MTOM all handle a large binary attachment proficiently, undoing the outcome of Base64 encoding. MTOM normally works well for large binary attachments, other than the overhead of using MIME makes it less suitable for small binary inclusions such as cryptographic hashes and X.509 certificates. The overhead of Base64 encoding made small difference to the experiential latencies.

These analysis results only reproduce the performance observed on a lightly-loaded high-bandwidth network and related tests run over a lower-bandwidth or busier network would show increased latencies for the larger multimedia content SOAP messages, particularly those with Base64-encoded binary elements. In explanation was that using gZip compression slowdowns the overall performance in every test messages by at least 50%, indicative of that compression was not effectual in this environment for some of our test message types. The new processor load of performing compression destined that the utilizing of gZip in the Web Services stack tainted response times in the entire cases on our extremely fast network.
VII CONCLUSION

This manuscript has obtained a well-organized Security based Web service framework for multimedia Web services. A trouble-free method for publishing and discovering multimedia Web services is explained. A metadata query service is formed to help the smooth progress of the description of the fundamental multimedia content for an exacting Web service. The transformation of multimedia streams is the end result of two innovative streaming data MEPs and the implementations of the MEPs with HTTP SOAP bindings with WS-Security. The HTTP SOAP bindings utilize the MIME Multipart/Related structure to output the multimedia packets. Every packet is then transferred as a package of a SOAP message in addition to its binary data. The novel MEPs and SOAP bindings acquire added transfer overhead in the transformation of multimedia streaming data contrast to a simple HTTP transmission method. To diminish the effect of this overhead, a variety of SOAP compression techniques for the packaging of SOAP messages are analyzed.

The proposed system analysis that using one of the binary XML schemes is the nearly all efficient amongst all the compression techniques that have been discovered. Finally an analysis of the proposed system demonstrates that the Web service transformation method has a performance similar to an HTTP multimedia streaming transmission scheme. At the instant the multimedia data is transferred utilizing the original quality. As a probable upcoming extension, the multimedia Web service would be more proficient if the quality of the transformation is flexible and could be controlled by a few QoS parameters. For instance, a SOAP request can be packaged with the necessary QoS parameter in the type of Composite Capability/Preference Profiles (CC/PP) [29]. Certain the depiction in the CC/PP, a multimedia stream in a suitable quality would be return by the QoS-aware multimedia Web service.

REFERENCES


