Intelligent XML Tag Classification Techniques for XML Encryption Improvement

Faisal T Ammari
School of Computing and Engineering
University of Huddersfield
Huddersfield, UK
f.ammari@hud.ac.uk

Dr.Joan Lu
School of Computing and Engineering
University of Huddersfield
Huddersfield, UK
j.lu@hud.ac.uk

Dr. Maher Abur-rous
Faculty of Information Technology
Zarqa University
Zarqa, Jordan
maher@zpu.edu.jo

Abstract— **Flexibility, friendliness, and adaptability have been key components to use XML to exchange information across different networks providing the needed common syntax for various messaging systems. However excess usage of XML as a communication medium shed the light on security standards used to protect exchanged messages achieving data confidentiality and privacy.**

This research presents a novel approach to secure XML messages being used in various systems with efficiency providing high security measures and high performance. system model is based on two major modules, the first to classify XML messages and define which parts of the messages to be secured assigning an importance level for each tag presented in XML message and then using XML encryption standard proposed earlier by W3C [3] to perform a partial encryption on selected parts defined in classification stage.

As a result, study aims to improve both the performance of XML encryption process and bulk message handling to achieve data cleansing efficiently.

Keywords: xml encryption; xml classification; xml security; encryption standard; data classification

1. Introduction

XML [1] is considered the leading standard for major data exchange among systems due to its flexible nature using a plain text to encode a set of information by using combination of tags allowing the XML document to be read and understood without the need for special reader or interpreter. Such flexibility encouraged businesses with different niches to adopt this communication medium in their communication messaging [2]. Increasing demand on using XML as a communication medium raised the necessity of having a proper and solid security measures surrounding exchanged XML messages, many recommendations have been proposed to secure the XML messages either by using XML Encryption [3], XML Signature [4], and XML Key Management [5]. Proposed security models tend to add a major security measure successfully; however results deploying these models involved in performance issues, presentation problems, and high resources usage which make the process of securing the message not efficient in high volume transactions.

To overcome such limitations especially the performance issues this research proposed a new model to handle large chunks of XML messages in an efficient and fast way improving the performance, speed, and resources utilization when integrating existing systems with any of the security models proposed. To achieve model goal the structure built taking into consideration two major phases before reaching the final step securing the XML message, one of the phases is to classify incoming XML messages using a hybrid set of rules extracted from a fuzzy inference system [6] and set of associative classification algorithms [7, 8, 9] to find out the importance level for each of the tags within every message received. Once classified by the classification module each tag is assigned an importance level (Low, Medium, and High) attribute which is used in phase two performing a partial encryption depending on the previous importance attribute assigned by classification module. Proposed framework tends to secure the necessary part of XML message without the needs to perform a full encryption which eventually save resources and achieve high performance level.

The rest of the paper is categorized as follows. In Section two we show previous work and literature review, Section three focus on our contribution in this research, Section four focus on system model and design, section five we illustrates a case study describing system usability, finally sections six will show conclusion and future work.

1.1 Extensible Markup Language (XML)

W3C first recommended XML as the standard for data representation over the web, however due to its flexible nature many businesses started to move their business communication messaging over networks to XML eliminating the complexities of standard communication messaging. Whether the transfer of XML messages is over the web or other networks this created an aligned interest to secure XML content being transferred and forbid any unauthorized access by public users.

Many security models have been proposed either on the network level [10, 11] or on XML level itself [3, 4, 5], among proposed models the World Wide Consortium (W3C) came up with the XML Encryption standard which gives the ability to perform a full or partial encryption on XML message content. Although W3C Recommendation provided the confidentiality needed but few issues rose using this security model concerning performance and inefficient memory usage which gives a clear space for improvements.

1.2 XML SECURITY

Extensive use of XML as a communication medium triggered an alert to have a proper and solid security models attached to the exchanged messages. As many recommendations and model have been proposed, W3C played a major role providing security models using different approached like XML encryption [3], XML signature [4], and XML Key Management [5] to provide a standardized way representing data and information in a secure manner. XML Encryption is one of their leading security standards to provide the required data confidentiality by encrypting selected parts or the entire document. Flexibility of selecting specific parts of the document to be encrypted gives us the advantage to select which parts to be processed saving time and resources, However, some publications included criticism focusing on performance and representation issues [12, 13] which will be discussed in literature review section

1.3 XML CLASSIFICATION

Due to XML nature being a structural language, classification can be done in three ways, first by using the textual content of the message and then use any standard text categorization technique to fetch the data, second by using only the structure of the XML document, and third by using a hybrid approach using both the textual content and the structure. Few classification approaches tend to use the hybrid approach in their classification models [14, 15]. We tend to classify XML messages based on their content rather than structure because it gives us more flexibility in case of any missing DTD elements, classification tend to categorize documents into classes based on a set of algorithms deployed on a pre-defined set of data.

Upon classification we should have a clear set of categories describing message contents enabling us to perform the proposed security model.

2. LITERATURE REVIEW

Flexibility, expressiveness, and usability of XML itself formed an interesting motive for researchers to shed more light on XML functionalities and deployment either on commercial level or scientific level, Researchers diverted their interest in securing exchanged XML messages due to extensive usage of XML in general, efficient models have been proposed [3, 4, 5, 16, 17, 18]. Models tend to protect XML messages in a proper way ensuring data confidentiality and authenticity, but due to the variations of XML threats [19] illustrated below:

1. Oversized Payload: Creating XDoS attack
2. Schema Change: Change WS schema to change and modify data processed by the used application
3. XML Routing: Redirecting sensitive data inside XML path
4. Entity Attack (External): Parsing XML input from various sources using incorrect XML parser
5. XML Parameter Tampering: Script injection into XML parameters
6. Coercive Parsing: Injection of suspicious content into XML itself
7. Recursive Payload: Creating XDos attack agains XML parser by sending mass amount of data

To cover all mentioned threats it was challenging for proposed security models, however it can be achieved by XML Encryption [3] which can secure the whole XML document or parts within the XML message covering sensitive information from being exposed to achieve data confidentiality. Some publications showed a performance issues [13], presentation problem [21], and use of canonicalization [20] "Continue"

3. CONTRIBUTION OF THE STUDY

In this research we proposed a new model to act as an Intelligent XML tag classification model for XML encryption improvement enabling businesses and institutions to take advantage of using it to provide a secure and efficient XML messaging system. A combined set of modules formed the system main functionality, each unit act as an independent functional unit to perform set of operations delivering desired output, Main model to provide both secure and efficient messaging operated by two independent units forming core of the system model, first unit main functionality is to classify XML messages using a hybrid approach of data mining classification techniques [22] and fuzzy logic techniques [23] to provide a mixture of future behavioral detection and existing pattern recognition, main output of this model is to extract the importance level and assign the value to tag attribute already defined in each XML message, once classified it will be forwarded to second module which perform the encryption based on importance level for each assigned XML tag, by doing so we ensure that only sensitive data of each message is encrypted which gives us the advantage of message optimization and utilization.

3.1 System Characterstics

System built to provide an efficient and secure XML messaging using various techniques to handle small or large quantity of messages, here are the basic characteristics:

- Security: Plays a major role in system design delivering a trustworthy communication channel, system security has more than one dimension protecting outgoing XML messages by encrypting sensitive information embedded within each message, three dimensions are used to cover system security, the first is the encryption standard for specific set of XML tags, second is the communication medium security where we use secure transmission over VPN to exchange encrypted messages, the third dimension is the authentication of messages whereby originator needs to be member of the system domain to be able to handshake with the main model for identification purposes.
- Flexibility: System is built to scale either vertically by adding new domain members, system services, and functionalities or horizontally by enhancing or amending existing techniques, such flexibility will give the option for other contribution to improve main model achieving the best usability.
- Efficiency: Main purpose of system design not only to handle system messages but to handle them in an efficient and robust way, whether system is handling large chunks of messages or small ones it is designed to distribute large amount of messages over many instances distributed in system farm, this achieved by using a load balancer to act as an intelligent distributor to different system units, each unit can perform number of incoming messages, using this technique will balance the process and achieve high performance and efficient distribution.
- Usability: System is designed to act as an independent unit where it does not pay much attention to the environmental factors like operating system, database used, and communication medium.
- High Availability: By using system farm which contains multiple entities of system acting as a complete separate system unit performing all functionalities.

3.2 Measurements of Success

System designed to achieve set of goals ensuring secure and efficient exchange of XML messages among different systems, following measures are key factor for overall success:

1) XML Classification: ability to classify XML messages by using a combination of system automation classification and human input classification to achieve highest credability

Success of this measure relies on the following:

- Flatten XML message to perform classification on message content rather than structure
- Classify XML messages to find the importance level value and assign it to the existing XML attribute within each message
- Define set of XML Tags to act as message identifier where they are categorized into three equal layers for ease of classification
- Content forwarding after performing system classification to either security module or to message assembler for final

2) XML Encryption: Upon XML classification process this module is designed to perform message encryption based on the assigned importance level attribute processed by the XML classification module, multiple encryption on the same document to be achieved based on importance level tag which gives the indeication whether to use high level key encryption or medium level. this approach is based on "W3C XML Encryption Recommendation"

Success of this measure relies on the following:

- Partial encryption to be deployed upon importance level assignment
- Symmetric cipher to be deployed on selected XML tags and based on their importance level assigned by the XML classification module
- Selection of block cipher or stream based cipher

3) Message Utilization: By performing XML classification and then securing only sensetive parts of each XML message system tend to achieve high performance due to decreased usage of resources and memory.

4) System Load Balancer: This module to ahieve highest availability and best performance where it distribute incoming bulk of XML messages into different entities in system farm whereby each entity in system farm present the complete system model, this will ensure the best performance in case of high usage and large amount of XML messages.

3.3 Goals and Objectives:

- Qualitative Goals
  1. Multiple encryption standards to be performed on the same document depending on importance level assigned during system process
  2. Data Classification based on a combination of intelligent data mining classification techniques and fuzzy logic using a simple rule-based approach 'IF X AND Y THEN Z'

- Quantitative Goals
  1- Bulk XML messages handling and encryption using a load balancer to achieve a simultaneous classification
  2- Parallel processing for both message classification and message encryption

4. System Model

4.1 Proposed Model to Secure and Manage XML messages

This research presents a novel approach to secure XML messages by using an intelligent data mining classification techniques to perform a partial encryption for specific parts of XML message overcoming the high usage of resources in inefficient traditional methods, early stage is to balance incoming XML messages and distribute to a different system entities using a load balancer, then to classify incoming XML messages using intelligent techniques to prepare for the final stage which is the encryption of specific XML tags based on importance level attribute assigned previously by the classification module. Figure 1.0 illustrates system model and components involved to deliver the XML secure management system.

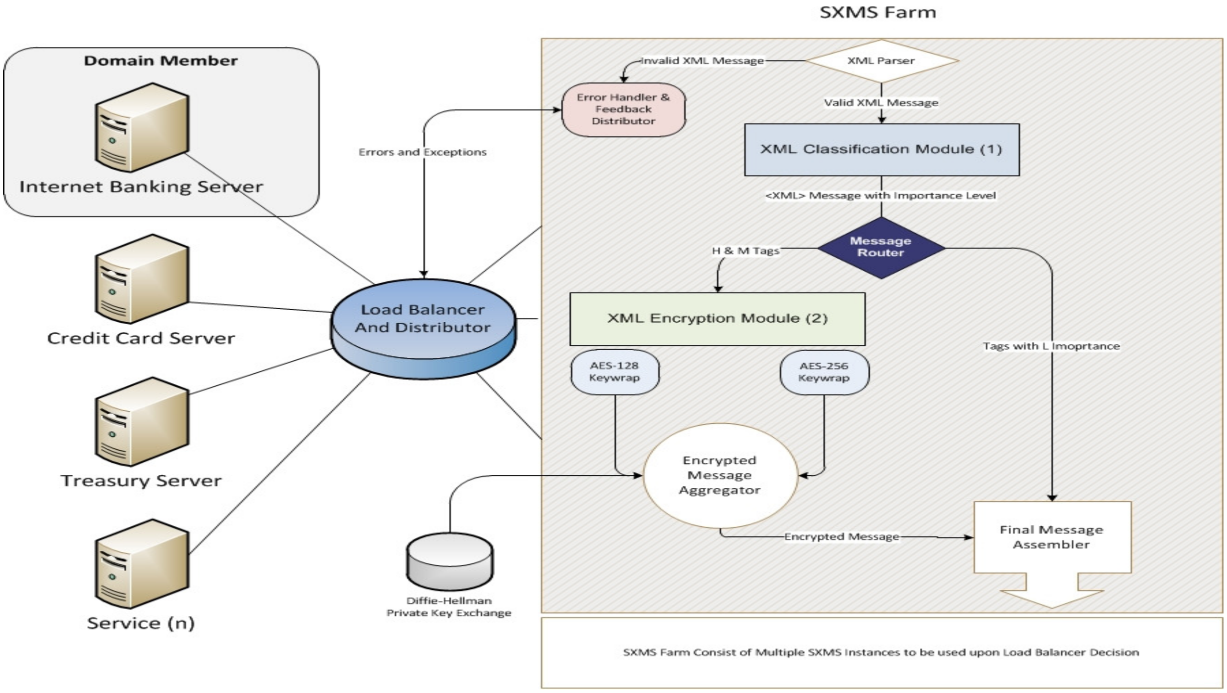

SXMS Farm
Domain Member
Internet Banking Server
Credit Card Server
Treasury Server
Service (n)
Load Balancer And Distributor
Errors and Exceptions
Invalid XML Message
XML Parser
Error Handler & Feedback Distributor
Valid XML Message
XML Classification Module (1)
<XML> Message with Importance Level
Message Router
H & M Tags
XML Encryption Module (2)
Tags with L Imoprtance
AES-128 Keywrap
AES-256 Keywrap
Encrypted Message Aggregator
Diffie-Hellman Private Key Exchange
Encrypted Message
Final Message Assembler
SXMS Farm Consist of Multiple SXMS Instances to be used upon Load Balancer Decision

Figure1.0 System Design

4.2 System Design

System has been built based on two main components forming system model core, main idea is to perform multiple tasks ensuring both high security standard for exchanged XML messages and efficiency securing them with robust security standards. First module is the classification module which is responsible for classifying incoming XML messages based on XML tags within each message, XML message is being flattened into text mode before the classification stage and later classified using a combination of automation and human expertise to take place in message classification using fuzzy logic approach and associative classification. Once tags have been classified, Encryption module is the second basic module to take place after message classification performing a partial encryption based on W3C recommendation [3] only to those tags flagged with specific importance level.

4.2.1 Classification Module: This module perform a set of intelligent techniques to assign a new value which is the importance level for each XML tag, main idea is distinguish which parts of the message to be secured using high encryption standard and which to be ignored and forwarded directly to message assembler, Module is using fuzzification techniques of a set of input variables based on 10 characteristics extracted from the incoming XML message depending on previous knowledge experience and expertise backgrounds as following:

1- Transaction Amount: financial institutions tend to set transactions limits, limits allow users to perform that amount on daily basis based on local policies, ranges are defined based on policy setting security measures for each range

2- Transaction Currency: Well defined list of allowed currencies to be used within online/offline systems is a must to have more control over transferred transactions; each currency has its own set of risk variables depending on usage and importance.

3- Account Type: To have a pre-defined list of account types placing segments on customer accounts allowing special handling for each account type

4- Transaction Notes: When performing a financial transaction the source account should have a clear status whereby it has no specific notes registered before, notes are recorded based on historical status along with current status, such notes gives a clear indication about account status

5- Profile ID: A unique identified for the destination account owner, this value is set during system integration and profile registration giving the mutual trust.

6- Account Tries: When the destination account is used more than ones this will give an indication that it is a trustworthy account paying less attention during data classification

7- Incorrect Password Tries: Number of times users tried to enter the password incorrectly to complete the financial transaction.

8- Time Spent on Service: Time spent navigating the service before performing the transaction, time range is set based on bank's policy defining each range to be critical time performing transactions or not

9- Daily Transactions: How many transactions performed before doing the final financial transaction

10- Transaction Time: financial day is categorized in three periods: peak period, normal hours, and dead zone. Periods are defined separately by each financial institution based on local policy and historical transactions range.

Classification Methodology

Model is based on Mamdani [23] fuzzy inference performing the basic four steps shown in figure 2.0

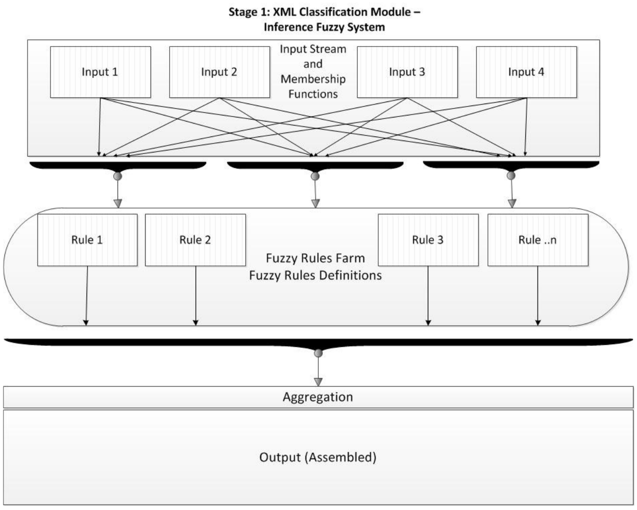


Figure 2.0: Mamdani fuzzy inference system

As shown the system consists of four basic steps as following:

- Step 1: Fuzzification
  Taking the crisp input X, Input Y, and determines the degree to which of these inputs belong to and where to fit in the fuzzy set. Figure 3.0 represents an example of sample of linguistic variable used representing one factor which is the transaction currency. The x-axis represents the range of transaction amount. The y-axis represents the degree of each value in linguistic descriptor

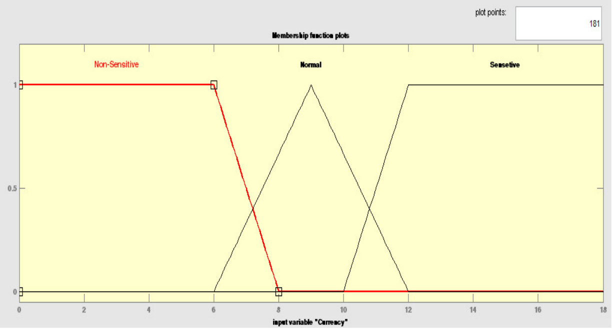


Figure 3.0: Input variable

Transaction Currency (Non-Sensitive, Normal, Sensitive)
Variable used: Transaction Amount
Ranges:

Non-Sensitive: [0, 0, 6, 8]
Normal: [6, 9, 12]
Sensitive: [10, 12, 18, 18]

- Step 2: Rule Evaluation
  Where we take the fuzzy inputs and apply to the qualified fuzzy rules, fuzzy operators (AND / OR) are used in case of any uncertainty to get a single value, outcome value is called "Truth Value" which will be applied to the membership function for rule evaluation
- Step 3: Aggregation of the Rule Outputs
  Process of unification of the outputs of all the rules, combining scaled rules into a single fuzzy set for each variable
- Step 4: Transforming the fuzzy output into a crisp output, figure 4.0 showing an example of expected crisp output [Low, Medium, High]

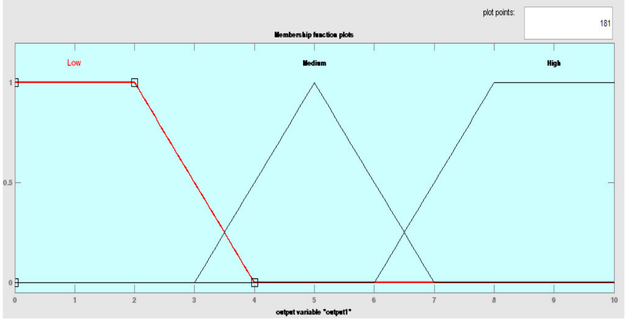


Figure 4.0: Sample output for classification rate “Importance Level”

Output should have a clear crisp value where it will be assigned to each tag classified

Low: which means importance level is low and should not pay more attention, root element and child tags to be forwarded directly to message assembler skipping encryption phase

Medium: Tag is somehow important; tag attribute is assigned the value of medium to be encrypted in next phase but with low key encryption

High: To be handled with high importance and encrypted in next phase with high key encryption “AES 256 Key Encryption”

Detection Model

To be able to perform the fuzzy inference system we have categorized the XML tags within each message into 10 characteristics distributed into three layers each has its own weight and criteria, Layers are Account Layer, Details Layer, and Environment Layer, Figure 5.0 represent the layers distribution

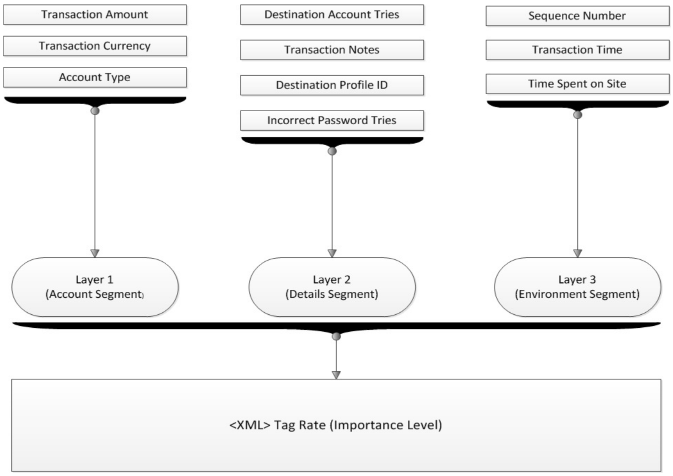


Figure 5.0: Layers Distribution

By giving a weight for each layer the calculation of overall weight is based on the following criteria:
Importance Level: **Sum** (Layer Weight * Layer Member)

Rule Base: each layer has a set of rules defined based on input variables within each layer, rule is based on “IF-THEN” rule, rule base should contains a number of entries depending on how many layer members exist, for example layer 1 has three members and we have three output expected so the entries should be calculated as (3□) = 27 entries presenting the rules for that layer. Table 1.0 represents a sample of rule base for layer1 (Account Layer), figure 6.0 and figure 7.0 illustrates surface and layout design.

Amount	Currency	Type	Account Layer
Non-Sensitive	Sensitive	Non-Sensitive	**Medium**
Non-Sensitive	Sensitive	Non-Sensitive	**Medium**
Non-Sensitive	Normal	Sensitive	**High**
Non-Sensitive	Normal	Sensitive	**High**
Non-Sensitive	Normal	Normal	**Medium**

Table 1: Rule base sample for layer 1

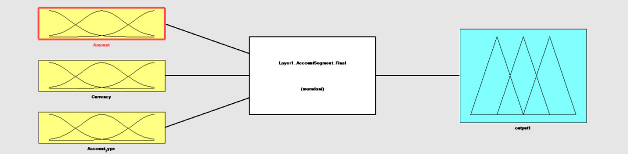
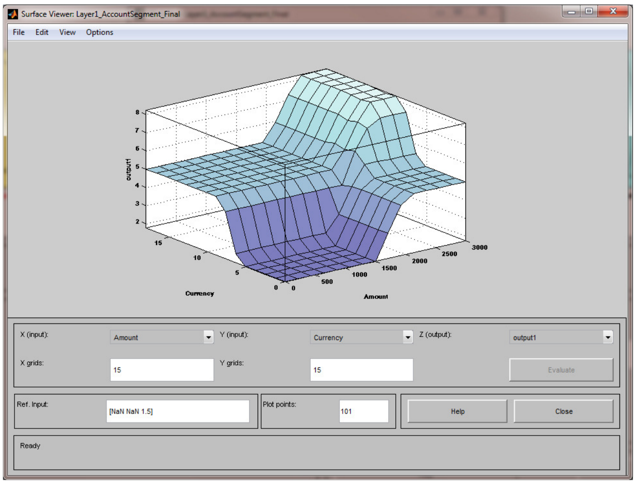


Figure 6.0: Layout of detection system



X (input): Amount
Y (input): Currency
Z (output): output1
X grids: 15
Y grids: 15
Ref. Input: [NaN NaN 1.5]
Plot points: 101
Ready

Figure 7.0 Surface views for the layer 1

Final evaluation is depending on finding centre of gravity as following equation:

$$COG = \frac{\int \mu_i(x)\ x\ dx}{\int \mu_i(x)\ dx}$$

µi(x): Aggregated membership function
x: Output variable.

After deploying the classification methodology on the three layers we will have a list of classified tags with importance level attribute defined and assigned based on the classification steps described above, rates to be taken to be used in next system level which is securing those XML tags.

4.2.2 Encryption Module

Upon classification stage, a list of XML tags have been assigned a value for one of their attribute which is the importance level, value should vary between (Low, Medium, High) which gives a clear indication to encryption module which parts to be encrypted and which to be forwarded directly to message assembler, We have chosen to perform the partial encryption on XML tags with importance level of (High, Medium) and forward any TAG with level of (Low) to message assembler where it will be aggregated with other XML parts at final stage. Figure 8.0 illustrate how the module operates.

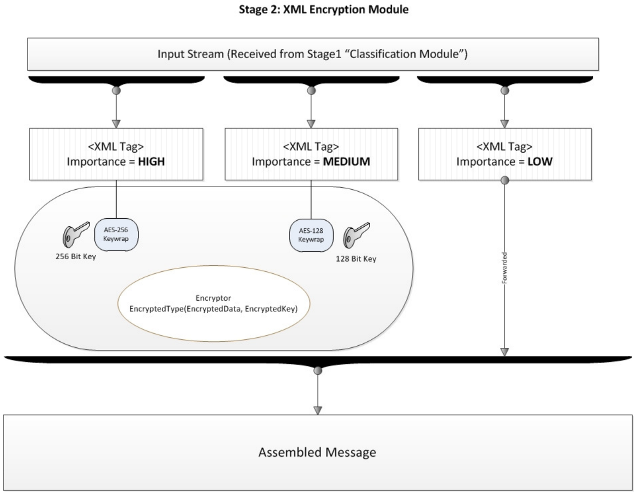


Figure 8.0 Surface views for the layer 1

Within encryption module and once the message received the input stream component will read tag importance level attribute and act accordingly where tags with importance level of (High, Medium) will be moved to final encryption process which uses AES encryption based on (128 bit key, and 256 bit key), below is sample of XML message received before the encryption and after the encryption:

Incoming XML message "after classification":

```
<TransactionDetails src='/paymentsystem.xml'>
 <Account ImportanceLevel="High">
        <AccHolder>Faisal Ammari</AccHolder>
        <AccountNumber>120130101144343401</AccountNumber>
        <Amount>32200</Amount>
        <Currency>USD</Currency>
        <Type>Indivisual</Type>
 </Account>
 <AccountDetails ImportanceLevel="Low">
        <AccountUsage>3</AccountUsage>
        <PasswordTries>1</PasswordTries>
        <ProfileID>0028827</ProfileID>
 </AccountDetails>
</TransactionDetails>
```

XML message after encryption

```
<TransactionDetails src='/paymentsystem.xml'>
      <Encrypted_Data src='xmlenc#'>
      <EncryptionMethod Algorithm='xml#AES'/>
       <Key_Info src='XML_Sig'>
         <Key_Name>AMD</Key_Name>
       </Key_Info>
       <Ci_Data>
         <Ci_Value>54544464fsdf?:#</Ci_Value>
       </Ci_Data>
     </Encrypted_Data>
 <AccountDetails ImportanceLevel="Low">
        <AccountUsage>3</AccountUsage>
        <PasswordTries>1</PasswordTries>
        <ProfileID>0028827</ProfileID>
 </AccountDetails>
</TransactionDetails>
```

Encryption is inherited from the parent Tag, As in above example the account tag has been classified with "High" importance level, Account tag represent one layer which is "account layer" consist of three basic components presented within layer main tag, once the parent tag has been assigned an importance level, tags are inherited with parent importance level value, which means they are encrypted using same encryption algorithm as their parent.

4.2.3 Message Utilizer

Classification module shed the light on which parts of the message to be encrypted; however message utilizer is performing two sets of operations to ensure maximum efficiency, first step is to utilize incoming messages from encryption module as following:

$$OG\ = \frac{ActualOutput - PotentialOutput}{PotentialOutput} * 100$$

Sample Utilization:
Original Message Size = **15k**
Classified Message: **4k / 7k / 4k**

$$OG\ = \frac{15-11}{11} * 100 = 36.6\%$$

In above example message has been utilized to achieve 36.6% out of original message processing time, Figure 9.0 illustrates how the utilization works.

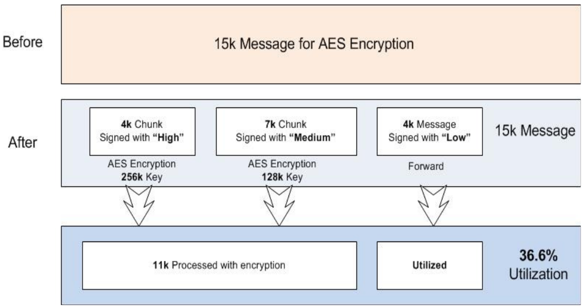


Figure 9.0 Message utilization process

Upon final utilization, Message will be ready for final submission to selected destination; However keys used for encryption should be transferred to decryptor using a secure and private way using Diffie-Hellman [24] key exchange for final message decryption. Figure 10.0 illustrates how to exchange keys between source and destination.

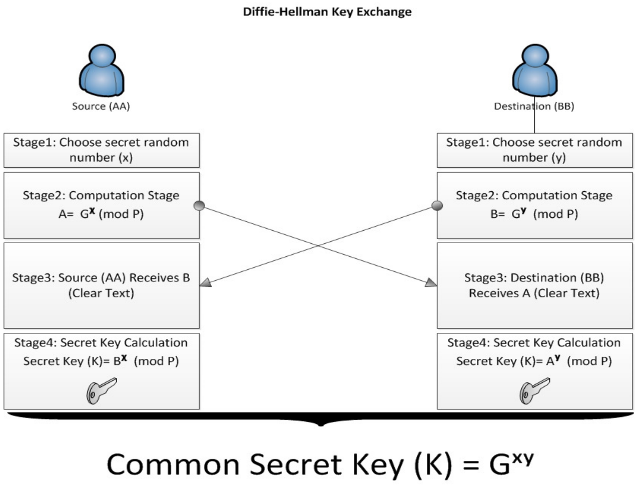

Diffie-Hellman Key Exchange
Source (AA)
Destination (BB)
Stage1: Choose secret random number (x)
Stage2: Computation Stage A= G^x (mod P)
Stage3: Source (AA) Receives B (Clear Text)
Stage4: Secret Key Calculation Secret Key (K)= B^x (mod P)
Stage1: Choose secret random number (y)
Stage2: Computation Stage B= G^y (mod P)
Stage3: Destination (BB) Receives A (Clear Text)
Stage4: Secret Key Calculation Secret Key (K)= A^y (mod P)
Common Secret Key (K) = G^xy

Figure 10.0 Diffie-Hellman Key Exchange

5. Case Study: Internet Banking Service

Banks and financial institutions tend to offer their services in a handy way serving their customers, one of the well-known services is the internet banking provided by many banks as a virtual service center enabling their clients to perform most of the financial transactions online and without the need to visit a physical branch. Providing such services will empower the relationship between financial institutions and their clients [25]. Most of the internet banking applications uses XML as a communication medium exchanging financial messages, However using XML should full fit customer needs like service efficiency and high performance which can be achieved by integrating our model in the existing internet banking application.

High usage of online services provided by banks and financial institutions created awareness for service providers to bundle these services with high security measures ensuring high data confidentiality and availability. Achieving high security measures requires not only high measures but an efficient approach to keep high security standard alongside high and efficient performance.

After taking the necessary approvals we have conducting our novel approach on internet banking service installed on a testing environment provided by Jordan Ahli Bank to measure the ease of integrating and performance improvement securing outgoing and incoming XML messages. System has been deployed as a middleware connected to the application backend. Few customizations have been placed to match XML message structure; mapping took place as well for final message classification.

A sample of 300 records have been tested using our system, system were able to classify the messages into three basic layers previously described in system design. Table2, Table3, and Table4 illustrates sample of data provided segregated into three layers.

Transaction_Amount	Transaction_Currency	Account_Type	Account_Segment_Layer
Non-Sensitive	Non-Sensitive	Non-Sensitive	**Low**
Non-Sensitive	Non-Sensitive	Normal	**Low**
Non-Sensitive	Sensetive	Sensetive	**Low**
Non-Sensitive	Non-Sensitive	Sensetive	**Low**
Non-Sensitive	Non-Sensitive	Sensetive	**Low**
Non-Sensitive	Non-Sensitive	Non-Sensitive	**Low**
Non-Sensitive	Non-Sensitive	Non-Sensitive	**Low**
Sensetive	Sensetive	Normal	**High**
Non-Sensitive	Non-Sensitive	Non-Sensitive	**Low**
Non-Sensitive	Non-Sensitive	Normal	**Low**

Table 2: Sample of data received classified for Layer1

Transaction_Notes_CODE	Destination_ProfileID	Destination_Account_Tries	Incorrect_Password_Tries	Details Segment
Non-Sensitive	Sensitive	Normal	Non-Sensitive	**Low**
Non-Sensitive	Non-Sensetive	Normal	Non-Sensitive	**Low**
Non-Sensitive	Normal	Sensitive	Non-Sensitive	**Low**
Non-Sensitive	Normal	Normal	Non-Sensitive	**Low**
Non-Sensitive	Sensitive	Normal	Non-Sensitive	**Low**
Non-Sensitive	Sensitive	Normal	Non-Sensitive	**Low**
Non-Sensitive	Non-Sensetive	Normal	Non-Sensitive	**Low**
Non-Sensitive	Normal	Normal	Non-Sensitive	**High**
Normal	Normal	Normal	Non-Sensitive	**Low**
Non-Sensitive	Non-Sensetive	Normal	Non-Sensitive	**Low**
Normal	Sensitive	Normal	Non-Sensitive	**Low**
Non-Sensitive	Non-Sensetive	Normal	Non-Sensitive	**Low**
Non-Sensitive	Normal	Normal	Non-Sensitive	**Medium**
Non-Sensitive	Sensitive	Normal	Non-Sensitive	**Low**

Table 3: Sample of data received classified for Layer2

Time_On_Site	Daily_Transactions	Transaction_Time	Environment
Non-Sensetive	Sensitive	Normal	**Low**
Sensitive	Sensitive	Non-Sensitive	**Low**
Non-Sensetive	Normal	Normal	**Low**
Non-Sensetive	Sensitive	Normal	**Low**
Non-Sensetive	Normal	Normal	**Low**
Non-Sensetive	Sensitive	Normal	**Low**
Non-Sensetive	Normal	Non-Sensitive	**Low**
Sensitive	Sensitive	Non-Sensitive	**High**
Normal	Normal	Sensetive	**Low**
Normal	Normal	Sensetive	**Low**
Non-Sensetive	Normal	Sensetive	**Low**
Non-Sensetive	Sensitive	Sensetive	**Low**
Non-Sensetive	Sensitive	Normal	**Medium**
Non-Sensetive	Sensitive	Sensetive	**Low**

Table 4: Sample of data received classified for Layer3

The importance level has been assigned to corresponding node and tags forming each layer, importance level later has been used to perform the message partial encryption using encryption module using two different key-sized encryption algorithms, first was AES-128k encryption for tags marked with “Medium” importance level and AES-256k encryption for tags marked with “High” importance level. Following sample message illustrates how the message looks like after the classification process performed by classification module.

```
<TransactionDetails src='/paymentsystem.xml'>
 <Account ImportanceLevel="High">
        <AccHolder>Faisal Ammari</AccHolder>
        <AccountNumber>120130101144343401</AccountNumber>
        <Amount>32200</Amount>
        <Currency>USD</Currency>
        <Type>Indisivual</Type>
 </Account>
 <AccountDetails ImportanceLevel="Low">
        <AccountUsage>3</AccountUsage>
        <PasswordTries>1</PasswordTries>
        <ProfileID>0028827</ProfileID>
 </AccountDetails>
</TransactionDetails>
<EnvironmentDetails ImportanceLevel=”Medium”>
        <TimeOnSite>00:05:45</TimeOnSite>
        <Daily_Transactions>4</Daily_Transactions>
        <Transaction_Time>16:17:22</Transaction_Time>
</EnvironmentDetails>
```

Message after encryption:

```
<TransactionDetails src='/paymentsystem.xml'>
     <Encrypted_Data src='xmlenc#'>
     <EncryptionMethod Algorithm='xml#AES256'/>
      <Key_Info src='XML_Sig'>
        <Key_Name>AMD</Key_Name>
      </Key_Info>
      <Ci_Data>
        <Ci_Value>54544464fsdf?:#</Ci_Value>
      </Ci_Data>
    </Encrypted_Data>
 <AccountDetails ImportanceLevel="Low">
        <AccountUsage>3</AccountUsage>
        <PasswordTries>1</PasswordTries>
        <ProfileID>0028827</ProfileID>
 </AccountDetails>
     <Encrypted_Data src='xmlenc#'>
     <EncryptionMethod Algorithm='xml#AES128'/>
      <Key_Info src='XML_Sig'>
        <Key_Name>DEB</Key_Name>
      </Key_Info>
      <Ci_Data>
        <Ci_Value>2342223%34?:#</Ci_Value>
      </Ci_Data>
    </Encrypted_Data>
</TransactionDetails>
```

In above message grayed area represent the secured content using key encryption depending on importance level presented in yellow area.

6. CONCLUSION AND FUTURE WORK

We have presented in this paper a novel approach to secure XML messages used in many critical mission applications by using on-the-fly mechanism classifying XML messages creating three layers and apply fuzzy logic approach to determine which parts of the XML message to be secured using two level of XML encryption depending on importance level attribute. Efforts were made illustrating how system operates and mechanism behind it, sample deployment on a real –life system took place with success. System were built taking into consideration flexible mechanism where a future improvements can take place, the following are list of characteristics which can take place for further improvements:

- Fuzzy inference system: can be replaced or enhanced with mixture of associative classification rules that can work in combination with fuzzy inference system
- Integration layer to be placed between our system and any backend system to map XML messages for legacy system that are hard to customize
- Load Balancer improvements where we can deploy new technologies like packet filtering and hardware balancing
- System adaptability where it adapt itself depending on communication speed and traffic

7. REFERENCES

[1] T. Bray, J. Paoli, and C. M. Sperberg-McQueen. Extensible Markup Language (XML) 1.0. W3C, Feb. 1998

[2] Fan, M. Stallaert, J. and Whinston, A. B.: The Internet and the Future of Financial Markets, Communications of the ACM, 43(11):83-88, November 2000.

[3] XML Encryption Syntax and Processing (W3C Recommendation), 2003.

[4] XML-Signature Syntax and Processing (W3C/IETF Recommendation), February-2002

[5] XML Key Management Specification (XKMS 2.0). http://www.w3.org/TR/2005/PR-xkms2-20050502/, 2 May 2005

[6] Mamdami, E.H.; Assilina, S., "An experiment in linguistic synthesis with a fuzzy logic controller", International Journal of Man-Machine Studies, vol. 7(1), pp. 1-13, 1975

[7] William W. Cohen. Fast effective rule induction. In Proceedings of the Twelfth International Conference on Machine Learning, Lake Tahoe, California, 1995.

[8] J Cendrowska. PRISM: an Algorithm for Inducing Modular Rules. International Journal of Man-Machine Studies, 27:349–370, 1987.

[9] Frank, E., and, Witten, I. Generating accurate rule sets without global optimisation. Proceedings of the Fifteenth International Conference on Machine Learning, (pp. 144–151). Madison, Wisconsin, 1998

[10] S. Kent and R. Atkinson. RFC 2401: Security Architecture for the Internet Protocol. Internet Engineering Task Force,Nov. 1998.

[11] A. O. Freier, P. Karlton, and P. C. Kocher. The SSL Protocol Version 3.0. Netscape Communications, Nov. 1996.

[12] Shirasuna, S., Slominski, A., Fang, L., Gannon, D., 2004. Performance comparison of security mechanisms for grid services. In: Fifth IEEE/ ACM International Workshop on Grid Computing. IEEE Computer Society, pp. 360–364.

[13] Park, N., Kim, H., Chung, K., Sohn, S., Won, D., 2006. XMLsigncryption based lbs security protocol acceleration methods in mobile distributed computing. Computational Science and Its Applications – ICCSA 2006’, vol. 3984. Springer, Berlin/Heidelberg, pp. 251–259.

[14] L. Denoyer and P. Gallinari. Bayesian network model for semi-structured document classification. Information Processing and Management, 40(5):807–827, 2004.

[15] M. Mesiti, P. Rosso, and M. Merlo. A Bayesian Approach to WSD for the Retrieval of XML Documents. In In Proceedings of JOTRI, pages 11–18, 2002.

[16] “Oasis security services (saml) tc,” ”http://www.oasis-open.org/committees/security/”.

[17] Organization for the Advancement of Structured Information Standards (OASIS), Extensible Access Control Markup Language (XACML), V2.0, February 2005.

[18] ContentGuard. XrML: The digital rights language for trusted content and services. http://www.xrml.org/, 2001.

[19] SOA Approach to Integration, Packt Publising, M. Juric, P. Sarang, R. Loganathan, F. Jennings, 2007.

[20] Imamura, T., Clark, A., Maruyama, H., 2002. A stream-based implementation of XML encryption. In: XMLSEC 2002: Proceedings of the 2002 ACM Workshop on XML security. ACM Press, pp. 11–17.

[21] Boyer, J.M., 2003. Bulletproof business process automation: securing XML forms with document subset signatures. In: XMLSEC'03: Proceedings of the 2003 ACM Workshop on XML security. ACM Press, pp. 104–111.

[22] Kantardzic and Mehmed. "Data Mining: Concepts, Models, Methods, and Algorithms ,", John Wiley & Sons. ISBN 0471228524. OCLC 50055336,2003.

[23] M. Liu, D. Chen and C. Wu. The continuity of Mamdani method. International Conference on Machine Learning and Cybernetics, Page(s): 1680 - 1682 vol.3, 2002.

[24] DIFFIE,W. ANDHELLMAN,M. E. 1976. New directions in cryptography. IEEE Trans. on Information Theory IT-22, 6 (Nov.), 644–654.

[25] Baskerville, R. et al..: Extensible Architectures: The Strategic Value of Service-Oriented Architecture in Banking, In: ECIS'05: Proceedings of the 13th European Conference on Information Systems, Regensburg, (2005) pp. 761-772.