Advanced XML Security

Framework for Building Secure XML Management System (SXMS)

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

***Abstract*— The eXtensible Markup Language (XML) has been widely adopted for information exchange across various networks due to flexibility providing common syntax for messaging systems. Excessive use of XML as a communication medium created an aligned interest in the level of security provided for those XML-Based systems. Many security concerns have been tackled by the World Wide Consortium (W3C) creating the "XML Encryption Syntax and Processing" Recommendation [4].**

This study presents a new architecture to handle received XML messages from various systems, on one hand suggested architecture is focused on classifying received XML messages (sensitive and non-sensitive) data to define which parts of the XML document to be encrypted and which to be forwarded to another module in suggested system to handle message composition, On the other hand the architecture is focused on securing XML messages by encrypting flagged XML parts each with different type of encryption depending on data sensitivity and importance level defined, this stage uses an approach based on W3C "XML Encryption Syntax and Processing" Recommendation.

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; data cleansing; xml security; encryption standards

I. INTRODUCTION

For the past few years XML [10] has become the leading standard for data exchange between various systems due to its nature of using plain text to encode a hierarchical set of information using verbose tags to allow the XML document to be understood without any special reader or interpreter. Many businesses tend to manage systems communication using XML as the standard medium [2, 3]. Due to data sensitivity exchanged in those systems, the concern of XML security has been raised to a significant level focusing on methods and approaches to secure XML messages exchanged. Many XML security models have been recommended [5, 6, 7] to handle security issues like distribution of secure XML documents and control access for published data using encryption standards. XML is redundant in its nature with large overhead because of how XML is structured. Yet, integrating security standards will deny the main idea of XML being a flexible language. Many models have been proposed to secure XML messages either by XML Encryption or XML Signature proving a robust mechanism securing XML messages in general; however those results were involved in performance issues showing a delay in files transmission due to security standards applied.

To overcome performance limitations this research suggests a new architecture to handle bulk XML messages with ability to encrypt each message using different type of encryption along with message optimization for better performance. Approach will handle both data cleansing and encryption in two separate phases.

The rest of the paper is categorized as follows. In Section two we show what contribution is being offered, Section three focus on system characteristics and features, data sensitivity classification, measurement of success, and how to measure performance, section four will describe system components and lifecycle, then Section five will focus on how to deploy suggested system in real life, finally section six will show conclusion and future work.

II. CONTRIBUTION OF THE STUDY

Suggested framework is defined by the formation of two major components; each component is well structured and operates as an independent unit providing set of operations needed to implement overall desired system. These components are designed based on a level of pre-defined priority to form the life cycle of the proposed architecture.
The following describes in details the new contribution:

- Create an algorithm for data cleansing (Classifier), ability to classify XML data submitted from various channels like (Banking applications, credit card, treasury, stock markets) into two major groups (sensitive, non-sensitive) based on data classification matrix described in table 3.0.
- Create an algorithm to find the importance level for data blocks classified by "Classifier Algorithm" based on the weight of each data block submitted.
- An advanced optimization level for the stream-based implementation of XML encryption proposed by apache and IBM "Xerces2" to overcome:
  1- Achieve high optimization level regardless of XML file size "before encryption", their

approach depends on specific file range (100kb – 200kb) to achieve best performance

2- Overcome their (0.7 – 26%) reduction of encryption processing time

- Enhance W3C "XML Encryption Recommendations" to perform multiple encryption standard on same document based on importance level defined by the proposed classification algorithm.
- Bulk messages handler, mechanism handle bulk XML messages submitted by different channels to redirect each message to its own path for further processing.

III. PROPOSED SYSTEM CHARACTERISTICS

A. *Suggested System Characterstics*

- Efficiency: System is designed to handle bulk of XML messages based on algorithm to distinguish between sensitive and non-sensitive data in each message according to importance level, handling unit should produce an optimized secure messages.
- Compatibility: System compatibility is a key factor in system design to fit in any environment; First algorithm which is used to decompose the original message and fetch sensitive data is based on a set of mathematical notations and relational algebra. Second algorithm is based on W3C "XML Encryption Recommendations" approach.
- Security: Communication between sender/ receiver, message handling process, encryption process, data cleansing process, and message assembly process have a suggested security measures in the proposed system architecture.
- Flexibility: System designed to act as a plug-and-play system, whereby minimal customization is required to handover the system to any new environment. a middleware to be suggested in the future to act as system assembler for adding new message types and modify existing ones.
- High Availability: An embed protocol within proposed system to ensure high percentage of overall system availability, operational continuity is achieved by using parallel components ready and attached to system back-bone.

B. *Data Sensitivity Classification*

A significant part of this study is based on data sensitivity level determined by data classification process; data submitted by surrounding channels should be processed by this process, purpose is to assure an appropriate deployment of security measures through system lifecycle. Confidentiality, integrity, and availability are the attributes which determines classification basics. Table I illustrates sample data classification matrix.

TABLE I. DATA CLASSIFICATION BASED ON SENSITIVITY LEVEL

	Non-Sensitive Data	**Sensitive Data**		
Criteria	Information can be available without any exception, data being exposed will not affect overall message integrity, one of the three attributes (availability, integrity, confidentiality) should be achieved in the message	Information which must be available and delivered intact to assure message integrity and secrecy, Data is restricted and can't be accessed unless in final stage (decryption process), high level of availability, confidentiality, and integrity must be availed		
Handler	- Message Forward by "Forwarder" Module without any additional handling (Encryption)	- Message Encrypt by "Encryptor" to deploy encryption standard based on sensitivity level of each tag forwarded		
Attributes Scale (High, Medium, Low)	Confidentiality: **Low** Integrity: **High** Availability: **Medium**	Confidentiality: **High** Integrity: **High** Availability: **High**		
Risks	Financial Risk: **Low** Operational Risk: **Medium** Continuity Risk: **High**	Financial Risk: **High** Operational Risk: **High** Continuity Risk: **High**		
Importance Level	**N/A**	**High**	**Medium**	**Low**
Sample Channel (Credit Card)	- Credit Card Expiry Date - Issuer Bank - Credit Card Type - Issue Date - Credit Card Issue Place	- Credit Card Number - Credit Card CCV2 - Holder Name		

C. *Measurements of Success*

Proposed system should meet the following success indicators to prove system authenticity and validity:

1) An efficient mechanism to distinguish between sensitive and non-sensitive data using "Decision Algorithm" system module

Success Factor: Ability to receive bulk of messages and extract sensitive data out of each message based on the Decision Algorithm module. Variables involved in module testing:

- XML Message Type (Transactional message, informative message, purchase message, etc.).
- XML message size and content (message size, message richness, message architecture).
- Sending Channel (Financial institution, bank, Educational institution, etc.).

2) A feasible way using multiple encryptions for the XML messages produced from "Decision Algorithm" module, this approach is based on "W3C XML Encryption Recommendation" with customization whereby it uses multiple encryption standards within same document according to importance level

Success Factor: Ability to deploy multiple encryption standards within same document efficiently each according to importance level and data sensitivity, Variables involved in module testing:

- XML Message Size
- Complexity of Data
- Type of Encryption Standard (RSA, AES, etc.)
- Importance Level (for submitted data blocks)

3) High Utilization measures: Message Assembler capability to compose forwarded encrypted message and other chunks in efficient and optimized way.

D. Performance Measurement

SXMS was designed to improve performance of two major entities, the encryption of XML messages (Individual or in Bulk) using multiple encryption standards and data classification process, Purpose of performance measurement is the continuous monitoring and reporting of overall system achievements, in specific to monitor pre-defined goals and what have been achieved.

Why Measure Performance:

- Identifying progress against pre-defined goals
- Discover potential improvements
- Compare performance against other standards

Goals of Performance Measurement:

- Qualitative Goals
  1. Encrypt XML message using multiple encryption standard within same document.
  2. Data segregation: fetch sensitive data from the original XML message for each message delivered depending on sensitivity level defined by data classification process.

- Quantitative Goals
  1- Bulk encrypt XML messages using multiple standards, goal is to encrypt simultaneous messages from different channels.
  2- Data segregation in Bulk, goal is to segregate sensitive data within same XML document for multiple messages received simultaneously from different channels.

IV. SYSTEM COMPONENTS AND LIFECYCLE

A. System components

- Decision Algorithm (SXMS Decider Module):

Is the key component in system architecture, component core is the mathematical equations and relational algebra used to fetch and classify "Breakdown" each message into multiple portions for further processing. Goal of message decomposition is to identify which parts to be directed to encryption module and which to be forwarded to message aggregator. Relational algebra is involved to fetch in-depth details for precise judgment and classification, each chunk of the main XML message should be treated differently as per classification; module will add set of new values to identify block importance and overall chunk sensitivity.

- Encrypt Algorithm (SXMS Encrypt Module):

This module is dependant on SXMS Decider and act as chunks receiver, module main responsibility is to receive chunks and re-classify them according to importance level assigned by decision algorithm. Re-classification is an essential part to decide what type of encryption to be used for each chunk, Encryption type is selected based on assigned importance level, and the relationship is direct correlation with importance level. Below table illustrates the direct correlation

Module consist of many more parts to accomplish overall goal, following components are involved in this module:

- Chunk re-classifier: (to re-classify submitted chunks) based on importance level and data sensitivity
- Chunk forwarder: after initial classification this sub-module will forward chunks without further encryption
- Encryption Identifier: to decide which type of encryption to be used in the specified chunk
- Function Encrypt(): for final chunk encryption and encapsulation/wrapping

- Message Optimizer (SXMS Optimizer):

Although this step is included in Encrypt Algorithm but the functionality is rather different and can be classified as independent sub-unit. Main purpose of this sub-module is to ensure maximum utilization of the shredded message. Utilization of message is calculated upon chunks arrival from Decision algorithm; once chunks have been classified SXMS Optimizer can perform inner functions for finding expected message utilization and final message destination. Below is sample message utilization:

Original Message Size = **12k**

Message size after classification: **4k / 5k / 3k (Forward)**

% Utilization (Output Gap) = (Actual Output – Potential Output) / Potential Output * 100

% OG = (12 – 9) / 9 * 100 = **33% Utilization**

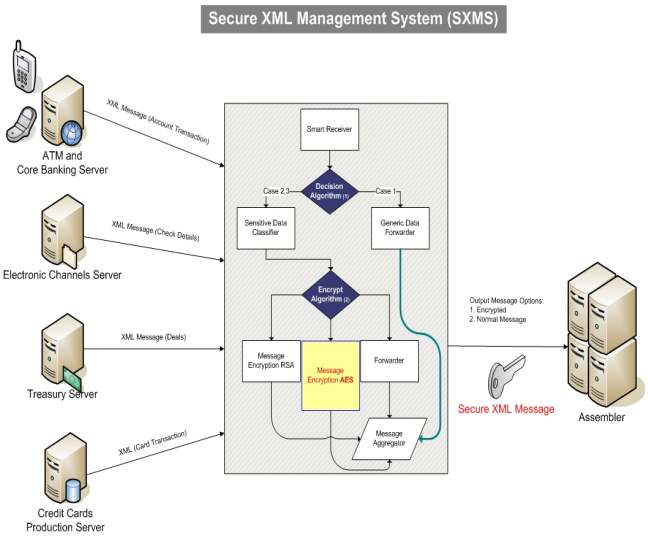

Figure 1. System lifecycle

Figure 1 illustrates how system works and phases included in system life cycle, following steps will describe in details how system operates using the core and assistant modules:

Step 1: (Message Composition) SXMS system can be used either within the organization or between different organizations, flexibility is a key factor in system design enabling easy adoption in any system environment. Source is creating the XML message based on the type of message, content, and final goal of sending message; in above scenario Treasury Department is sending one of the deals through SXMS to the final assembler which is clearing department. Application within the treasury server is constructing the message based on deal content and purpose, once composed; a connection will be established with SXMS engine to initiates and alert "Smart Receiver" sub-module. Once the message is submitted, smart receiver will accept the connection and XML message accordingly whereby it seals message validation and handshake confirmation.

Step 2: (Message Delivery by Smart Receiver) this sub-module act as a validator to check and confirm message validity, checking includes:

- Origin of the message (Sender authenticity)
- Message Tag, Handshake flags
- Message Type and content
- Message Description and destination
- Determine next phase and path

Once checked and confirmed, receiver will send the message to next phase which is "Decision Algorithm" phase.

Step 3: (Data Cleansing by Decision Algorithm) the core module in the system, purpose is to classify "Breakdown" each message into multiple portions for further processing. Classification is based on the importance level of each tag identified using this phase.

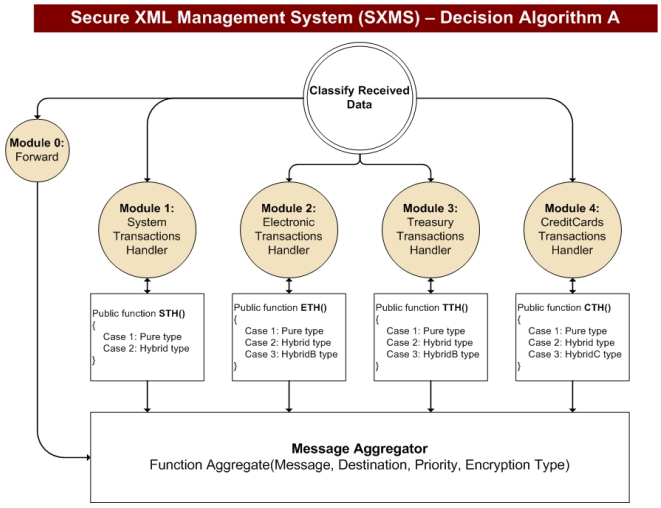

Figure 2. Decision Algorithm Structure

Step 4: (Message Optimization by Optimizer) main purpose of this step is to find optimal message size by calculating utilization percentage to find out feasibility of the encryption algorithm, final message utilization is calculated by finding output gap:

% Utilization (Output Gap) = (Actual Output – Potential Output) / Potential Output * 100

Message optimizer will keep record of how each chunk utilized and sum up overall message utilization, this step is required to measure performance of each message handled via SXMS. Figure 3 illustrates how module functions with example.

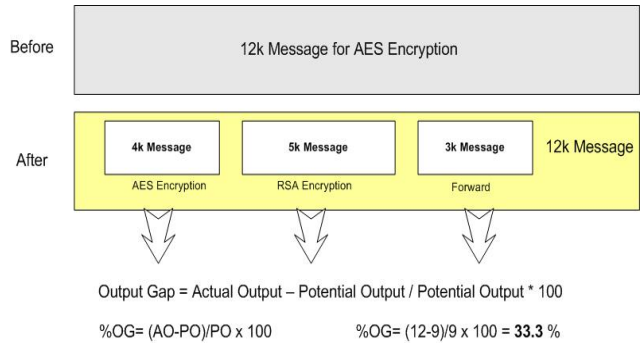

Figure 3. Message Optimizer

Step 5: (Data Encryption by Encrypt Algorithm) Step3 and Step4 jointly presents the core of SXMS operation, this algorithm is to encrypt extracted / forwarded data from pervious step whereby it handles each block separately to define the level of encryption and standard to be used. Encryption standard is dependant on importance level flagged and forwarded by Decision Algorithm, currently suggested is to have two types of encryption each to be used upon block flag defined earlier.

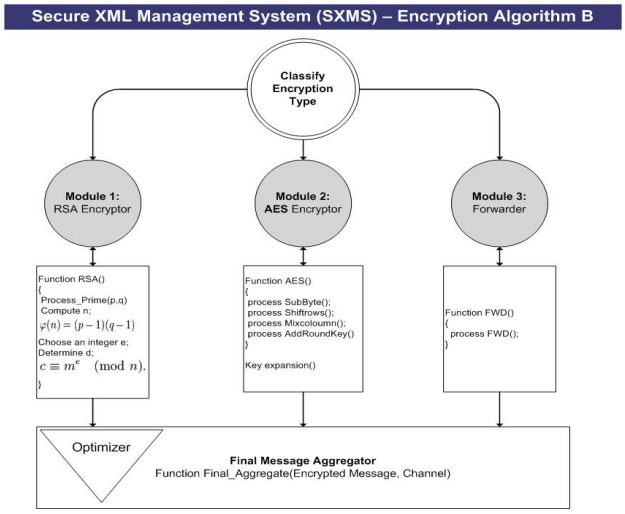

Figure 4. Encryption Algorithm Structure

Step 6: (Assembly by Message Assembler) final stage of system lifecycle is to compose the final encrypted message; assembly is based on many sub-functions including:

- Chunks assembly
- Chunks validation and verification
- Utilization checksum
- Chunks source
- Final message composition

V. SXMS IN REAL LIFE

Scandals and money laundry cases occurred in many banks and financial institutions world wide revealed the vulnerability of those institutions to abuse, many of them depends in their systems on XML as communication medium or for data storage [8], SXMS designed to fit in any system to act as complete and secure management system for XML messages they handle within their systems or between other organizations. As an example of real life implementation of SXMS I take one of Jordanian leading banks and how SXMS can help improve quality, security, and productivity of daily operations.

Daily operations involved many submissions from various channels like treasury department, credit cards department, and electronic channels like ATM, internet banking, and phone banking. Huge traffic generated because of large number of submissions and size of each file submitted. Securing those messages is the main target of SXMS; efficiency is a key role in SXMS concept as well.

Large banks deal with large number of transactions on daily basis from their ATM machines, ATM plays an essential role in the client/bank relationship [9], and it means customer is concerned with quality and speed of each transaction requested. At the backend, banking systems are working under high pressure to achieve customer needs but without neglecting security issues to wrap each transaction coming from ATM machines in a secure way. SXMS can act as an intermediate between ATM machines and bank's backbone to deliver secure, optimized messages in timely manner.

Case Study (Jordan Ahli Bank): High demand on ATM over the years created a good awareness level to provide a better and faster ATM service, such enhancements lead to use XML as flexible and adaptable language replacing old techniques. In year 2007 the bank successfully replaced old communication with XML technology and became fully dependant on XML to handle ATM Transactions. Number of XML messages exchanged between ATM machines and bank's backbone increased dramatically as table II shows the increase.

Type	2007/2008	% Change	Weight
Balance Inquiry	7,257,874	13.7%	**34.4%**
Cash Withdrawal	9,730,076	20.3%	**42.2%**
Summary of Balances	3,830,659	34.3%	**16.4%**
Utility Payments	919,012	4.8%	**2.6%**
Internal Transfer	533,012	0.9%	**0.2%**
Other	1,180,366	4.2%	**4.1%**

TABLE II. ATM TRANSACTION BASED ON XML MESSAGES

SXMS can handle high volume of daily transactions in a secure and fast way providing XML message encryption and optimization based on data sensitivity**.**

VI. CONCLUSION AND FUTURE WORK

This paper suggested a framework for building secure XML management system; efforts were made to illustrate system functionality and adaptability with examples in real life. Practical testing to take place to prove theory and efficiency of suggested system. SXMS is a new approach and many contributions can take place to enhance and enrich overall system functionality, the following contributions can be achieved in the future:

- Network optimizer: Checking packet sizes before submitting and route each message depending on communication medium of specific submitting channel

- Load balancer: to control expected load to achieve high system performance

- Ability to use multiple communication medium (ADSL, ISDN, Wireless, ATM, Fiber Optic)

- On-the-fly scan: This is an advanced module to be added on the communication medium and detect XML message content while it is being transferred to SXMS

- Messages reconciliation: Audit tool to check and validate sent/received XML messages from each channel

- Middleware for message composition and easy customization

Feasibility study will take place for adding more modules during practical testing and deployment.

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] Rabhi, F.A. and Benatallah, B.: An Integrated Service Architecture for Managing Capital Market Systems. IEEE Network, 16(1):15-19, 2002.

[4] http://lists.w3.org/Archives/Public/xml-encryption/

[5] G. Miklau and D. Suciu. Controlling access to published data using cryptography. In Proceedings of the International Conference on Very Large Data Bases (VLDB), pages 898–909, September 2003.

[6] S. Cho, S. Amer-Yahia, L. Lakshmanan, and D. Srivastava. Optimizing the secure evaluation of twig queries. In Proceedings of the International Conference on Very Large Data Bases, 2002.

[7] E. Bertino and E. Ferrari. Secure and selective dissemination of XML documents. ACM Transactions on Information and System Security, 5(3):290–331, 2002.

[8] RANDAZZO, M. R., KEENEY, M. M., KOWALSKI, E. F., CAPPELLI, D. M., AND MOORE, A. P. 2004. Insider Threat Study: Illicit Cyber Activity in the Banking and Finance Sector. U.S. Secret Service and CERT Coordination Center/Software Engineering Institute, Philadelphia, PA, 25.

[9] 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.

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