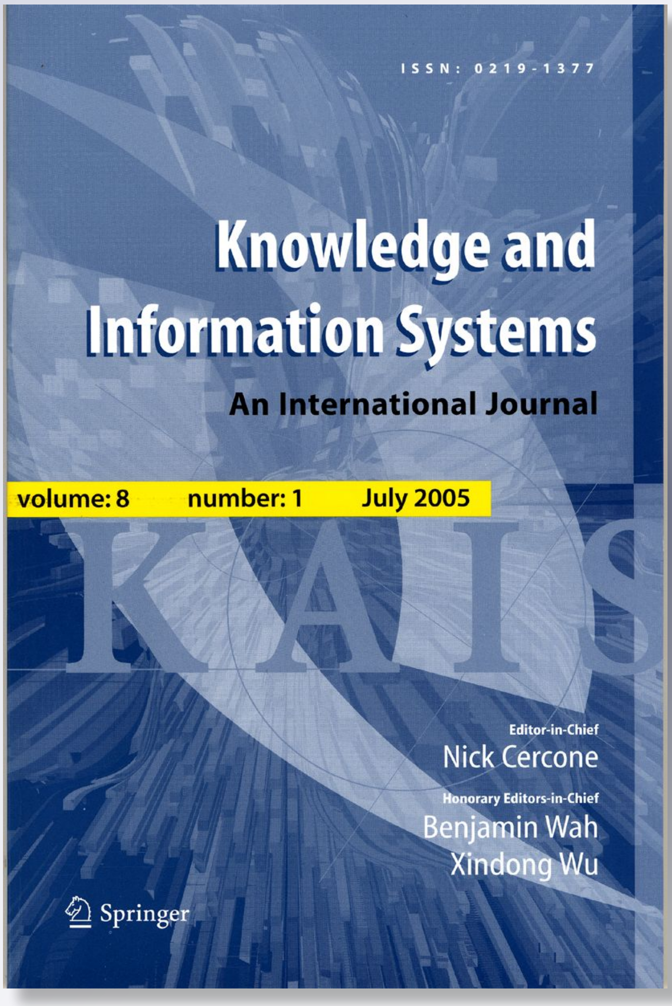
An energy-conserving approach for data formatting and trusted document exchange in resource-constrained networks

P. P. Abdul Haleem & M. P. Sebastian

Knowledge and Information Systems
An International Journal

ISSN 0219-1377
Volume 32
Number 3

Knowl Inf Syst (2012) 32:559-587
DOI 10.1007/s10115-011-0450-0



Knowl Inf Syst (2012) 32:559–587
DOI 10.1007/s10115-011-0450-0

REGULAR PAPER

An energy-conserving approach for data formatting and trusted document exchange in resource-constrained networks

P. P. Abdul Haleem · M. P. Sebastian

Received: 12 October 2009 / Revised: 6 May 2011 / Accepted: 22 October 2011 /
Published online: 26 November 2011

Abstract Lightweight data formatting and document exchanging schemes are of interest in conserving energy while exchanging documents, particularly in the resource constrained networks (RCNs). This paper presents an energy-conserving, lightweight method for data representation and trusted document exchange. It is based on a format derived from the YAML Ain't Markup Language (YAML), a lightweight data serialization language and includes a schema extraction process to separate data from its structure. The resultant schema is useful for thinning the data format and also for checking the rewriting attacks. The rewriting attack is checked using a two-tiered approach. It is observed that the proposed document format is less verbose and more energy conserving than XML and other popular non-binary formats. The format can be used to represent non-document data also.

Keywords Knowledge representation · Mobile computing · Verbosity reduction · XML · Rewriting attacks · Data formatting · Document exchange · PBM · YAML

1 Introduction

XML is widely used as a medium for representing and exchanging documents among heterogeneous types of applications and devices. It is a meta-language that allows the developers to create their own markup languages [1]. XML is particularly useful in exchanging information due to its flexibility and generality over the binary and proprietary formats, developer productivity, provisions for interoperability between the applications and widespread tool availability [2]. It is an essential entity in the generation of programmable web services [3] [4]. Recently, XML finds application as a message syntax format in Simple Object Access

P. P. Abdul Haleem (✉)
Department of Computer Science and Engineering, MES College of Engineering, Thrikkanapuram PO,
Kuttippuram, Kerala 679573, India
e-mail: ppabdulhaleem@gmail.com

M. P. Sebastian
Indian Institute of Management Kozhikode, IIMK Campus PO, Calicut 673510, India

Protocol (SOAP). The power of XML lies in its extensibility, where the developers can define a set of domain-specific tags that carry the semantics of the documents [1]. Despite its universal acceptance in multiple applications, the verbose nature of XML (due to the abundant use of tags) is a cause of concern in RCNs. This may seriously hinder the future use of XML (especially in RCNs) for exchanging, parsing, and querying documents, because of the fact that the document size grows much faster than the communication bandwidth [5]. RCNs include ad hoc networks, sensor networks, mobile networks, and pervasive systems.

1.1 Constrained mobile wireless environment

The abundant growth of Internet services and increased number of wireless mobile users resulted in the penetration of wireless mobile devices into the realm of wired networks. Wireless mobile devices (with limited resources such as battery power, memory, processing power, and input and screen) and wireless networks (with constraints such as unreliable channels, low bandwidth, increased latency, increased rate of retransmission of lost packets, and weak security features) together impose a number of limitations. Users also expect to access a wide range of applications due to the popularity and advancements in wireless and mobile computing technologies [6].

Fewer number of characters for data representation is well suited for an environment with limited input and screen. The largest chunk of power in a mobile device is consumed for the transmission and reception of the messages [7]. Packet losses and subsequent retransmissions through relatively unreliable channels can result in further resource crunch. Security protocols for the constrained mobile wireless environment also prefer reduced size of document and less number of packets to be transmitted/ received [8]. All of these constrains have direct links to the size of the document to be transmitted.

Power limitation is a bottleneck in the design of wireless mobile devices [9]. The power limitation is mainly due to the space constraints to accommodate battery within a mobile device [7]. The protocols used for wireless networking such as Bluetooth and Wireless Local Area Network (WLAN) also consume battery power considerably [10]. Data communication in the wireless mobile environment has to address issues originating from insufficient battery power integrated to the wireless mobile devices. The biggest consumer of energy in a mobile device is the RF part of cellular engine, which is responsible for transmission and reception of messages [7]. Another point of concern is the possibility of power drain due to packet losses in the wireless mobile environment. The packet loss in WLAN is estimated to be 30%, which results in heavy drain of battery power [7]. The orthogonal frequency-division multiplexing (OFDM) technology shipped with the IEEE 802.11a and 802.11g standards is known to have high peak-to-average-power ratio (PAPR), expensive transmitter circuitry requirement, and poor power efficiency [11].

Security protocols in the constrained wireless mobile environment must conserve energy to the maximum extent. Among the main sources of power drain during a secure wireless session are the number of packets to be transmitted or received and the size of the messages required for establishing a session. Analysis on the energy consumption for processing the various cryptographic algorithms underlines the need to have messages with reduced size and small key sizes [8].

The huge verbosity (a difference of about one-third) of an XML-ized document compared to the same information listed in a standard document format [5] underlines the need for size reduction of messages, thereby achieving energy conservation. Reduction of message size results in reduction in the number of transmissions, which in turn reduces the data transmission cost and also handles the shortage of storage space [12]. It is observed that in the wireless world, reducing the message size is one of the most important concern than achieving the processing efficiency gains through an alternative format [13]. Hence, it is clear that the size of document format is the directly influencing factor on all constraints, particularly on the energy consumption, of wireless mobile devices and networks.

In tune with the widespread use and popularity of wireless mobile devices, complex operations in distributed and collaboration technologies that allow people to move across organizational boundaries and to collaborate among/in organizations and communities are possible with mobile devices. As a result, user communities demand for increased flexibility, inter-connectivity, and autonomy of involved systems as well as new coordination and interaction styles for collaboration among people. Three emerging trends that are visible in this context are (i) providing virtual web communities, (ii) web recommender systems that are adaptive to the ubiquitous devices, and (iii) adaptive content generation and delivery.

Recently, virtual communities and the so-called social networks have become very popular. Hence, the idea of providing a virtual web community enabling virtual collaborations among the users is gaining momentum. In such an environment, the issues of describing data in a compact format to efficiently exploit the limited resources in the mobile environments (by supporting better ways of providing data relevant to the user and enabling improved interoperability with the environment and with other mobile users) are crucial. Research works such as context-aware data sharing service to enable spontaneous collaboration between mobile workers [14] and providing collaborative services for ad hoc and spontaneous communities [15] underline the need for a less verbose data formatting and document exchange scheme, which can improvise the performance of the resource-conserving virtual collaboration schemes among the user communities.

Web recommender systems help users make decisions in the complex information space where a huge volume of information is available to them. Recommendation systems are expected to provide multidimensional recommendations considering items, users, and the exploited device. Recently, a number of web page recommender systems (such as [16–18]) have been developed to extract the user behavior from the users navigational path and to predict the next request as s/he visits a web page.

The popularity of wireless mobile devices also pose a different kind of challenge. A web server usually delivers the content without considering the heterogeneity problem in terms of the different types of access devices, network bandwidth, preferences/characteristics of the user, etc. As a result, the devices with limited resources may also be delivered with pages that are rich in images and video. This may lead to slow content delivery or even making it impossible to visualize some pages. To address this problem, adaptive content generation and delivery [19–22] and context-aware computing to fuse mobile, sensor, and social data [23] were developed.

In many of the above approaches, an XML-based dynamic content creation mechanism is employed to create, maintain, and provide multiple variants of the content depending on the type of devices from which the request is originated. The device (the capacity of the device) and information (the principle of poly-representation based on the document surrogates and the data source characteristics) "contexts" play decisive roles in the modeling of effective contextual information retrieval systems [24]. Clearly, an alternative scheme with the merits of XML and with less verbosity can be a catalyst in the performance gain of the data management and sharing services, which is one of the most important and challenging part of a middle-ware layer in a distributed storage system that allows group members to share data in collaborative working environments (CWEs).

There is an argument that the designers of wireless mobile devices are competing to pack more facilities into the devices, and hence there is no scope for energy-conserving measures, especially for messaging. But in reality, these features are used only by the top 10% of the customers. Also, the wireless revolution that transformed telecoms in emerging markets such as India, China, and Africa is constrained by the fact the electricity market remains underdeveloped [25]. More than 2.5 billion people (over 40% of the planet's population) live in rural and remote areas of developing countries [26]. More than 1.5 billion people live without access to electricity, and another billion have access to only unreliable electricity [27]. An estimated 80% of the world's population could use a mobile phone (thanks to broad wireless network coverage), but only about 20% actually subscribe to a wireless service, mainly because mobile phones are too expensive, according to the GSM Association. It is also interesting to note that the biggest penetration of wireless networking and devices are from these emerging markets as opposed to developed nations [26,28].

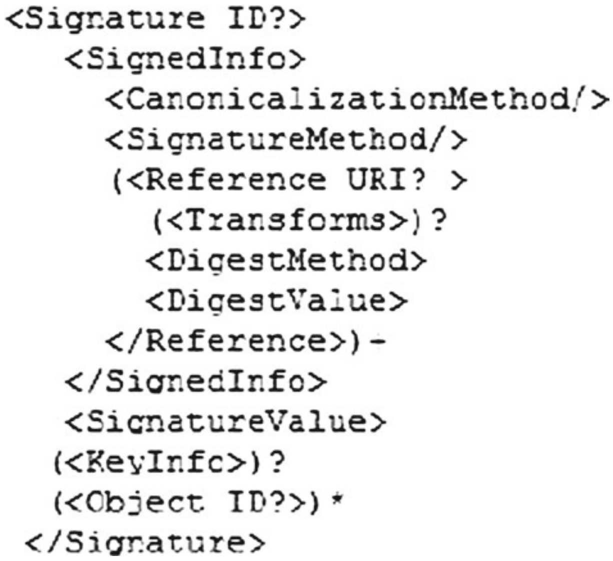
These facts reveal the importance of low-cost wireless mobile devices that are low on features and do not come with many features other than the SMS support. Low-cost devices are steadily increasing their market share, especially in developing countries. The projected shipments of ultra-low-cost mobile phones is expected to grow to more than 360 million by 2015, representing a 2010–2015 compound annual growth rate (CAGR) of over 22%, while shipments of low-cost handsets will reach 249 million by the same time period [29]. Mobile phone penetration in developing countries now stands at 68%, whereas in developed countries, it is reaching saturation levels with an average of 116 subscriptions per 100 inhabitants [30].

The relationship between lack of infrastructure such as power availability, per capital income, and demand for low-cost entry-level devices, especially in emerging markets, indicates wide scope for energy-conserving applications. The biggest consumer of energy in a mobile device is the RF part of cellular engine, which is responsible for transmission and reception of messages [7]. The trend from voice to (mobile) data applications is reflected in the growing number of text messages sent in 2010, which are close to 200,000 text messages per second [30]. Hence, it is clear that both for the entry-level devices and powerful new-generation devices, a less verbose, energy-conserving document exchange scheme is very relevant.

1.2 XML rewriting attacks

XML documents are prone to attacks that are classified in the literature as XML rewriting attacks (replay, man-in-the-middle, redirection, and dictionary attacks) [31]. Attackers make use of a loophole in the XML Signature specification [32] to modify the message [31,33]. XML Signature specification (Fig. 1) allows a non-contiguous object of an XML dataset to be signed separately. Each signed object is referenced by a Uniform Resource Identifier (URI) indirection from the Reference element of the signature. This indirect referencing is by means of an ID, without giving any clues about the actual location of the signed element. This leads to rewriting attacks, where an object can be relocated. The two types of attacks that are common are (i) generation and insertion of new elements into the messages, and (ii) copying parts of a message into other part of the same message, or into fabricated messages that may be generated using the previous pattern [34]. These modifications result in either a replay attack or a redirection attack. Figures 2 and 3 show the steps in a sample XML rewriting attack.

The signed object referenced by URI = "1" is moved into a bogus header and the modified portions are inserted (instead of the moved parts). As long as the reference indirection remains the same, this change does not cause any problem because (i) the message contains a signature provided by an authorized requester, (ii) the value of the element referenced by



```
<Signature ID?>
   <SignedInfo>
     <CanonicalizationMethod/>
     <SignatureMethod/>
     (<Reference URI? >
       (<Transforms>)?
       <DigestMethod>
       <DigestValue>
     </Reference>)+
   </SignedInfo>
   <SignatureValue>
  (<KeyInfo>)?
  (<Object ID?>)*
</Signature>
```

Fig. 1 XML digital signature specification

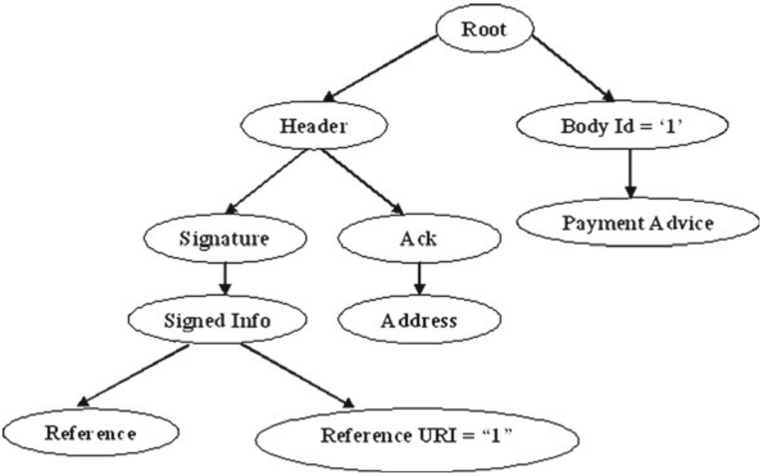


Fig. 2 Structure before the XML rewriting attack

the signature is unchanged, and (iii) the reference uses a position independent mechanism. Formulation of a thinned data format with less number of bytes alone may not be sufficient to withstand the common attacks prevailing on XML messages.

1.2.1 Policy-based messaging (PBM)

Messaging has become a killer application of the new, flexible, open, dynamic and service-aware network paradigm. Hence, messages that are sent over the network become critical documents when they contain sensitive information. So there is a need to augment the

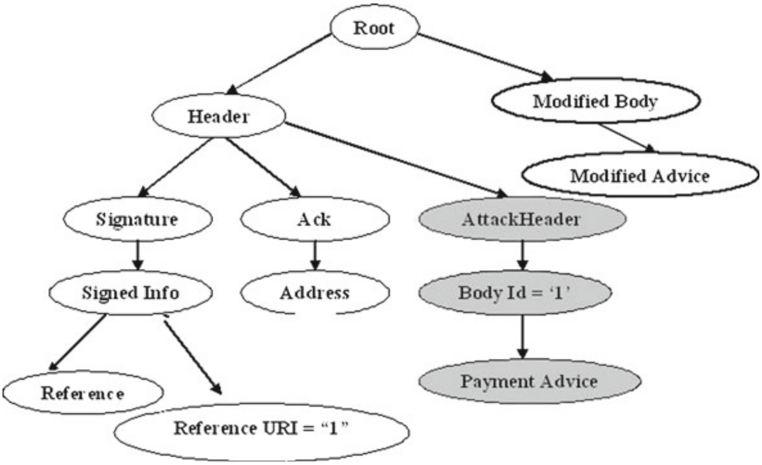
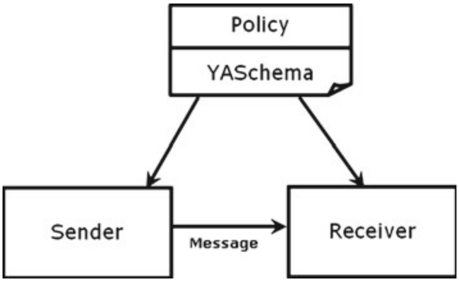


Fig. 3 Structure after the XML rewriting attack

Fig. 4 A PBM system



messaging systems with policy-based management enabled infrastructures so as to provide more flexible and secure messaging services.

Policy based messaging (PBM) aims at enforcing security policies with messages and for the recipient systems, thus promoting a distributed mechanism for secure messaging as shown in Fig. 4. Secure messaging is achieved by defining security-related policies and confining the messaging systems behavior to defined security constraints [35].

The steps needed for a PBM system as outlined in Eggenberger et al. [35] and Zhao and Chadwick [36] are as follows:

1. Compose the messages and assign recipients to the messages.
2. Identify what constraints and rules are needed to be applied to access the messages. All constraints and rules are then defined by a policy.
3. Attach the policy to the message and send with the message. Deliver, store, and protect the policy as an integral part of the message.
4. Retrieve the enclosed policy before allowing the user to access the contents of the message by the target messaging system.
5. Perform action on the message in accordance with the policy by the target system. Refuse requests of action in violation of the policy.

A thinned document exchange format that is not prone to common XML rewriting attacks and enhanced with a lightweight PBM trust management will have a wider scope in the constrained wireless mobile environment.

Thus, it is desirable to have a document exchange format with the following features: (i) simple and flexible (both for the user and the application programmer), (ii) less verbosity for the message without compression/binary encoding (without affecting the readability), (iii) a schema definition that helps reduce the overheads and with added functionalities for trusted data exchange, (iv) mechanisms to check rewriting attacks, and (v) support for non-document data.

In this paper, we propose a data formatting and document exchange scheme that consumes less number of bytes for data representation. The desirable features of the proposed format include the readability and user friendliness as in XML, but with less number of bytes for data representation. We also propose a lightweight scheme for protecting the data from rewriting attacks. This work is motivated by the well-known resource constraints in the wireless mobile environment [37] and the importance of data size over processing efficiency in conserving the resources in the constrained wireless mobile environment [3].

The rest of the paper is organized as follows: Sect. 2 surveys the related work. Section 3 presents the proposed scheme. Section 4 evaluates the performance of the proposed scheme and Sect. 5 concludes the paper.

2 Literature survey

Reduction in the bytes for document exchange was a topic of interest of several research papers. The methods adopted in these papers include adoption of compression techniques [38,39] and alternative serialization methods including binary [40,41] and non-binary. The additional overheads for the compression–decompression processes and neutrality to natural languages make these methods less attractive for the RCNs.

Several alternative non-binary serialization formats were proposed to reduce the verbosity of XML, which include YAML [42], JavaScript Object Notation (JSON) [43], Open Node Syntax (ONX) [44], and Simple Outline XML [45]. These formats suggest methods for conserving space and tend to maintain the agility of XML.

It is not sufficient to have a sliced serialization format alone for the data when there is a concern about the trusted transmission of messages. As we stated earlier, XML documents are vulnerable to rewriting attacks. Many solutions were proposed to overcome this attack, which include XML Digital Signature [32], WS Policy [46], WS Security [47], SOAP Account [34], and WS Policy Advisor [48].

Though these methods suggest novel and innovative solutions, they are not free from shortcomings and overheads. The security loophole reported for XML Digital Signature is inherited by WS Policy and WS Security, as both of them make use of the XML Digital signature specification. The improper framing of the policy rules allows trapdoors for malicious attackers. Also, the WS Security standard includes XML Digital Signature, XML Encryption, X.509 certificate, and Kerberos ticket, making it complex and heavyweight, especially for the RCNs. The WSE Policy Advisor shows performance degradation when the policy configuration file is complex. Also, the queries run by the WS Policy Advisor cannot detect the possible existence of signed element reordering attack [31,33,34,49]. The in-line approach [34,49] is a novel method to check the rewriting attack by keeping the account information of the signed objects. This is done with the introduction of an account structure consisting of the following information: (i) the number of child elements under the root element, (ii) the number of header elements in the message, (iii) the number of references in each signature, and (iv) the successor and predecessor relationship of each signed object as parent element and sibling elements. A flaw in this method is highlighted in Gajek et al. [50].

Thus, many research proposals are available in the literature as XML alternative data formats. However, none of them meet the twin requirements of reduced verbosity and security. Hence, there is a need for additional research to formulate a data format that performs better than the existing ones. In our previous work [51], we have chosen YAML as the base format for proposing an alternative to XML. The format needed a schema extraction process and two phases of thinning. We have created a schema definition and have taken measures for reducing the verbosity of the messages. In this paper, we refine the schema extraction process and make use of the schema definition for ensuring additional trust for the transmitted data.

3 The proposed scheme

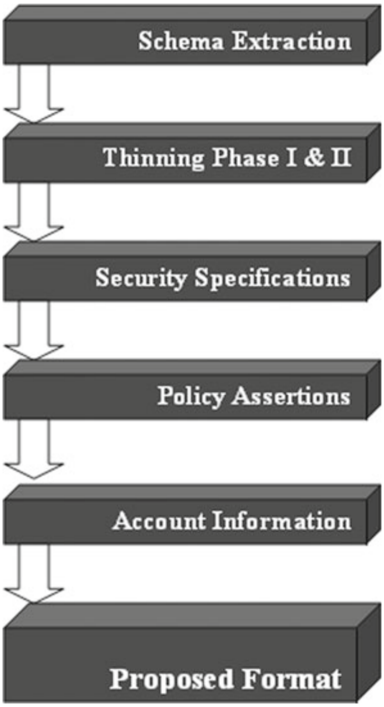
In this section, we propose refinements to the schema creation process and propose modifications to the Signature specifications [51] to make the data safe from the threats of rewriting attacks. We propose a two-tier mechanism to make the data in transit free from rewriting attacks. The proposed scheme is as shown in Fig. 5.

Our scheme can be subdivided into two stages: (i) creation of YASchema for verbosity reduction, and (ii) employing YASchema for a lightweight policy-based messaging (PBM).

It is to be noted that the emphasis of this paper is on the representation of data with less verbosity for better document exchange. Data include both text and non-text (images, sound, and video). As both XML and YAML are text-based formats, it is interesting to check whether all these types of data can be represented using these formats.

Since it is not possible to embed a raw binary graphics file (or any other non-text data) directly into an XML/YAML file, any of the following methods can be used for this purpose:

Fig. 5 The proposed method



(i) by giving the exact location of the file as a url, or (ii) by encoding the non-textual file to base64 format and representing it as textual data being represented. Non-textual data are not embedded as such directly in any of these formats.

Hence, it can be concluded that YAML can be used to exchange non-textual data as in XML. But the overall verbosity will be lesser in YAML due to its serialization methods. Since the method of representation is identical for text and non-textual data, we have only considered text data for our experiments.

3.1 Creation of YASchema for verbosity reduction

3.1.1 Message creation in YAML format

Messages of varying size and complexity are prepared in the YAML format. Based on the discussions in Qin and Taffet [52] and Lanka and Parikh [53], five categories of messages are identified for message creation [51] as shown in Table 1.

As XML supports structured data, simple to highly structured data must undergo the verbosity reduction measures. The *Short* category represents typical messages that are exchanged between the users (a simple messaging format with only text). The *Small* category represents a sophisticated version of the *Short* category. It represents a typical e-business message of an invoice or an inquiry form. The *Medium* category typically holds the details of about 25 customer records. The *Large* and *Composite* categories are constituted as invoice records with higher degrees of complexity. These categories are designed to provide consistent test data. Among these categories, the *Medium*, *Large* and *Composite* categories contain repeated occurrences of data with the same structure. In the case of *Small*, *Short* and *Medium* categories, all elements are subjected to size reduction measures, whereas in the case of *Large* and *Composite*, only parts of the message are subjected to size reduction measures.

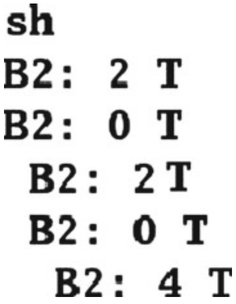
Together with the document to be transmitted, user needs to prepare a brief description about the structure of the message. The description, which we call as "metafile" includes details such as category of the message, type of each nodes, and whether the node contents can be squeezed or not. This arrangement is proposed to simplify the schema creation and trust management processes.

A sample metafile is as shown in Fig. 6. The first entry of the metafile indicates the category of the message. The next entries indicate the message structure—type of the nodes, the number of nodes, and whether it can be squeezed or not. In Fig. 6, the entry "B2: 2 T" indicates that the first node in the message is a mapping that can be squeezed. An outline of the document structure is available in the metafile.

The advantages of introducing a metafile is that the schema extraction process gets a gist about the structure of the document. This eliminates the overhead to scan the entire message.

Table 1 Message categories

Type	Remarks
Short	Size upto 50 bytes
Small	1 Record—string, float, DateTime
Medium	25 Records—string, float, DateTime
Large	1 Customer with 75 products
Composite	25 Customers with 10 products each



```
sh
B2:  2 T
B2:  0 T
  B2:  2T
  B2:  0 T
    B2:  4 T
```

Fig. 6 Metafile for "short" category

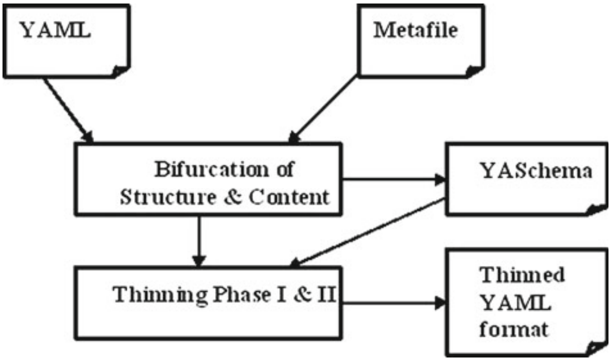


Fig. 7 Bifurcation of structure and content

3.1.2 Bifurcation of structure and content

Separation of data from content is very helpful in reducing the redundancy and verbosity of the message. When there is a set of messages to be transmitted with a common structure, impact of the bifurcation of structure and content will be higher [51]. The schema awareness is also helpful for a possible integration and data sharing among distributed, heterogeneous, and autonomous data sources across the web [54].

This process is done in two steps as shown in Fig. 7. In the first step, the schema details are extracted and written to a file. Both message and metafile are opened for processing. Category of the message and node type (of the message) are determined from the metafile. These details are recorded in the schema definition. Metafile is again checked to query about the structure of the message, the type of nodes, their numbers, and whether they can be squeezed or not. Then, a DFS is done on the first node of the message structure below the root to extract the details needed for schema creation. For each node, element name of the node, its data type, its node type, tag value, squeezable or not, and whether mandatory element or not are recorded in the schema definition.

In the second step, the derived information is verified with the metafile. Details like whether required or not and preserving the order of attributes are also added to the schema definition at this stage. For each entry in the metafile, the corresponding entries in the schema definition are compared for the node type, data type, and other related details.

This method has the following advantages over the method presented in Haleem et al. [51]: (i) an outline of the message structure is available (in advance, in the form of a metafile), (ii) verification of the schema details are possible with the help of the metafile (without the need to verify the entire message contents again), and (iii) identification of the squeezable elements is done in the metafile itself, simplifying the schema creation process.

3.2 Employing YASchema for a lightweight PBM

XML rewriting attacks may lead to the modification of (i) message, (ii) structure and expected hierarchy of the message, (iii) successor–predecessor relationship, and (iv) number of predecessor, successor, and sibling elements of the affected area.

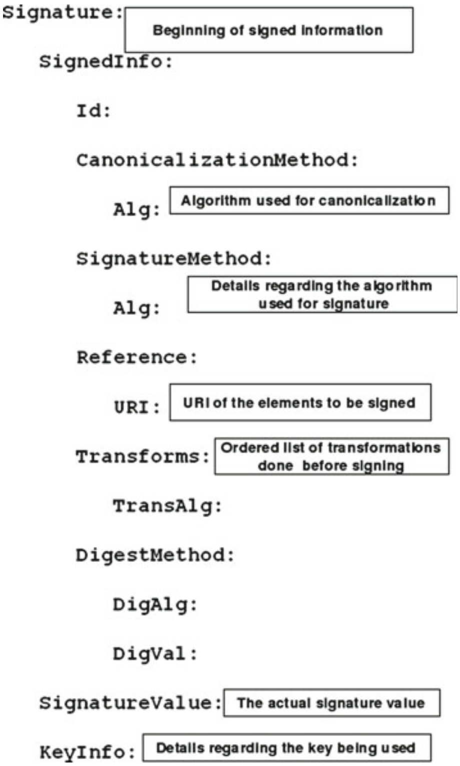
In this section, we propose mechanisms to check the rewriting attacks on YAML message with the help of YASchema. This involves restructuring of the security specification for signature processing for the YAML format and a two-tier trusted mechanism.

3.2.1 Signature processing

The most obvious solution to data security in transit would be protection using encryption and digital signature. The easy relocation of the indirect referencing used for the signed object without invalidating the signature can be exploited by an adversary to gain unauthorized access to protected resources.

In Haleem et al. [51], we proposed a specification for signature processing as shown in Fig. 8. To eliminate the trapdoor for a possible rewriting attack, we add additional data to the signature specification. This data, called as "Ylocation," lists the location of the referenced elements (its ID, its immediate predecessor, siblings and its depth [31] in the document

Fig. 8 Signature specification


Signature:
Beginning of signed information
SignedInfo:
Id:
CanonicalizationMethod:
Alg:
Algorithm used for canonicalization
SignatureMethod:
Alg:
Details regarding the algorithm used for signature
Reference:
URI:
URI of the elements to be signed
Transforms:
Ordered list of transformations done before signing
TransAlg:
DigestMethod:
DigAlg:
DigVal:
SignatureValue:
The actual signature value
KeyInfo:
Details regarding the key being used

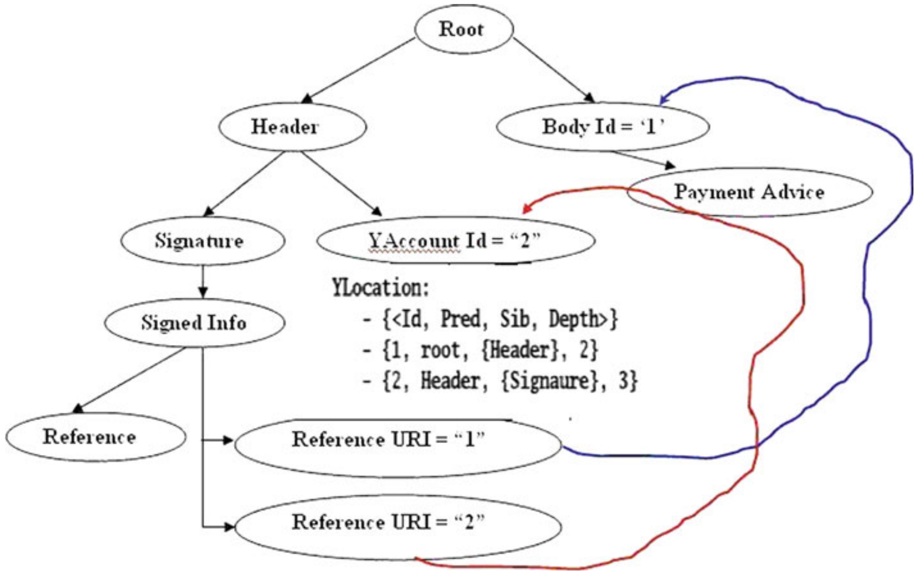


Fig. 9 Addition of location of signed objects in the signature specification

hierarchy) as shown in Fig. 9. "Ylocation" is appended after the existing signature specification structure, as a sibling of SignedInfo.

The proposed ID value assigned to each object should be unique. Each time an ID is assigned, it is recorded in the YASchema as a list. This prevents the assignment of duplicate IDs. For some reasons, if the sender and receiver are in difficulty to agree with the requirement of unique ID for "Body" elements, still they can agree with the depth information. The location information attached to the Signature specification clearly points toward the physical location of the signed object. The assignment of the unique ID and depth information to each signed object and its listing in the "Ylocation" part are added to protect the document from rewriting attacks [50].

This additional information cannot be made mandatory always with the signature specification due to the compatibility issues with the existing applications. Hence, we propose a two-tier mechanism to check the attacks.

3.2.2 The two-tier trusted mechanism to check the rewriting attacks

A two-tier approach is needed because (i) the addition of YLocation to the signature specification cannot be introduced to all situations in the existing applications, (ii) to strengthen the overall trust level of the mechanism so that security holes in one level can be exposed in the other level, and (iii) to suit the methodology to the constraints of the wireless mobile environment. The choice of making use of any/all of the proposed mechanisms is left to the user, depending upon the importance of the message and the situation of the surroundings. The simplest approach in the stack is designated as tier 0.

In tier 0, we utilize the inherent properties of the YASchema to check the attacks. In addition, we map the most common policy assertions in the WS Policy specification with the YASchema constructs. In tier 1, we combine the SOAP Account information and depth of the elements together. The unique ID is also added to this.

Tier 0

Simplicity and less vulnerability to attacks are the main objectives of this tier. In this level, instead of employing a separate policy file, we map the inherent properties of YASchema with a set of policy assertions as stipulated in WS Policy [46], WS PolicyAssertion [55], and WS SecurityPolicy [56]. Three basic assertions identified in Bhargavan et al. [48] are included in the proposal. These are (i) message predicate, (ii) integrity, and (iii) signature assertions. A message predicate is an assertion to list the mandatory parts of the message being transmitted. An integrity assertion lists the parts of the message to be signed. A confidentiality assertion lists the parts of the message to be encrypted. A combination of these preconditions prevents, in the form of a declarative XML format, the injection of bogus entries and alteration of message contents.

The features of YASchema that are useful in checking the attacks and their mappings with the base assertions can be summarized as (i) the elements that are required are marked (this helps in checking whether any of the required elements are absent after receiving the message; this is a message predicate assertion), (ii) the order of occurrence of the elements in the message is listed (this helps in checking whether any of the bogus elements are injected to the message structure by the intruder), and (iii) data type of the elements are specified. In addition to this, list of elements to be signed (integrity assertion), list of elements to be encrypted (confidentiality assertion), and age assertion [46] (which specifies the acceptable time period before messages are declared "stale" and discarded) are also added to the YASchema. A description about the mapping of policy assertions with YASchema constructs is as shown in Fig. 10.

YASchema and metafile can be exchanged between senders and receivers in the following ways: (i) sender and receiver agree upon a predefined schema and metafile, which is exchanged in advance, and (ii) YASchema and metafile are sent along with the message. The receiver first checks the message structure with the information available in the metafile.

```
type:B2
  # ****  Age Assertion (age:) ****
  :[tp: B2, rd: T, sz: 1, no:25, age:]
  # ****  Mandatory Assertion (rd:) ****
    invoice:[tg:_1_, cl:B2, tp:1, rd:T]
    date:[tg:_2_, cl:B2, tp:01, rd:T]
  # ****  Integrity Assertion (sd:) ****
    billto:[tg:_3_, cl:B2, tp:B2, rd:T, no:3, sd:T]
      given:[tg:_31_, cl:B2, tp:01, rd:T]
      family:[tg:_32_, cl:B2, tp:01, rd:T]
      address:[tg:_33_, cl:B2, tp:B2, rd:T, no:4]
        lines:[tg:_331_, cl:B2, tp:01, rd:T]
        city:[tg:_332_, cl:B2, tp:01, rd:T]
        state:[tg:_333_, cl:B2, tp:01, rd:T]
        postal:[tg:_334_, cl:B2, tp:01, rd:T]
    tax:[tg:_5_, tp:03, rd:T]
    total:[tg:_6_, tp:03, rd:T]
  # ****  All Policies Puttogether ****
policy:
  mand [_ALL_]
  sid [_3_]
```

Fig. 10 Mapping of base assertions with YASchema constructs

The methods for document transfer are (i) with a common metafile and YASchema agreed upon by the sender and receiver, and (ii) with the metafile and YASchema being part of the message transferred.

In the first method, the structure at the receiver is checked against the metafile and YASchema constructs. Anomalies of the document structure, especially in the modified bogus header part, is the indication that the message had been intercepted. In case the conformance with the metafile is successful, then the received document is checked as per the assertions and the specifications mentioned in the YASchema. As there will not be any specifications for the added bogus header entry in the YASchema, such documents are declared as tampered.

In the second method, the digest values of the metafile and the YASchema are attached with the message by the sender. These digest values are calculated again by the receiver and compared against the digest values already received. In any case, if the metafile or YASchema is modified by the attacker, it will be reflected in the digest values. Thus, at this stage itself, the attack can be detected. If there is a match, then the received structure is verified against the metafile and YASchema for anomalies.

This scheme has the following advantages : (i) YASchema is reused as a measure to check the rewriting attack and hence the payload needed for policy files gets reduced, (ii) several inherent features of YASchema can be derived as policy assertions (hence YASchema acts both as a schema and policy specification), and (iii) both metafile and YASchema are checked against the possible attacks using a two-tier mechanism, which is more comprehensive.

Even though many instances of XML rewriting attacks can be checked by these measures, mistakes in the preparation of policies due to lack of proper understanding of threats and security mechanisms can still lead to rewriting attacks. For example, if some of the message parts are not included in the message predicate, either a new bogus header can be inserted or the message can be routed to another user unintended by the sender [34]. We propose tier 1 scheme to enhance the trust level.

Tier 1

To enhance the trust level, we propose an account structure to be bundled with the data in transit. The structure that is called “YAccount” consists of the following information (i) number of child elements of the root, (ii) number of header elements, (iii) number of references for the signing element, (iv) predecessor of the signed object, (v) sibling relationship of the signed object, and (vi) depth of each signed element [31]. In addition to this, we propose a unique ID to be assigned to each signed object. This ID is also part of the YAccount. Inclusion of ID and depth eliminates the security holes mentioned in Gajek et al. [50].

The functionality of our approach is illustrated in Figs. 11 and 12. YAccount is calculated by the sender before sending the message and is included in the message to be transmitted in the signed format. M denotes the portion of the data that is to be signed. The signed part, YAccount calculated and signed, and the rest of the data are concatenated and sent to the receiver (Fig. 11). The receiver re-calculates the YAccount and compares the calculated YAccount with the account information received from the sender (Fig. 12).

3.3 Security analysis

We first consider a classical attack discussed in related papers. Then we discuss a particular attack that breaks the in-line approach as demonstrated in Gajek et al. [50]. Both the attacks must be detectable by our approach. Since the methodology adopted for YLocation and YAccount are identical, only the steps with YAccount are described here for brevity.

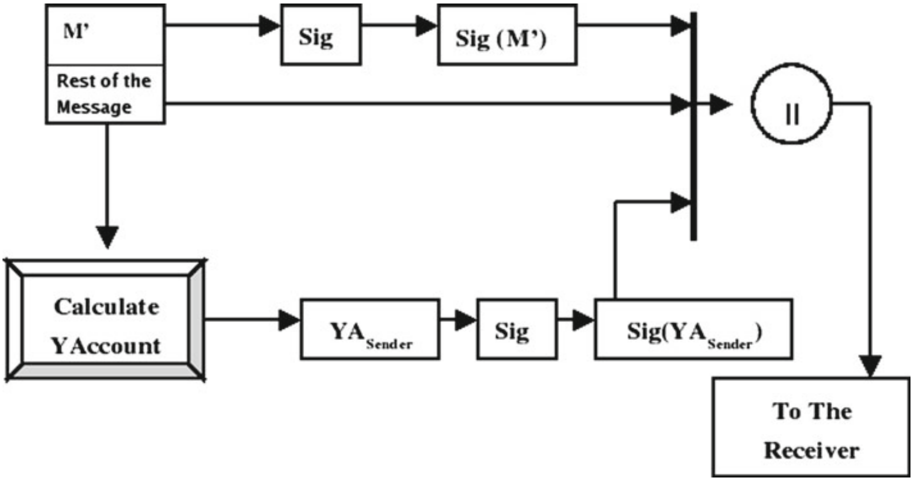


Fig. 11 YAccount before rewriting attack

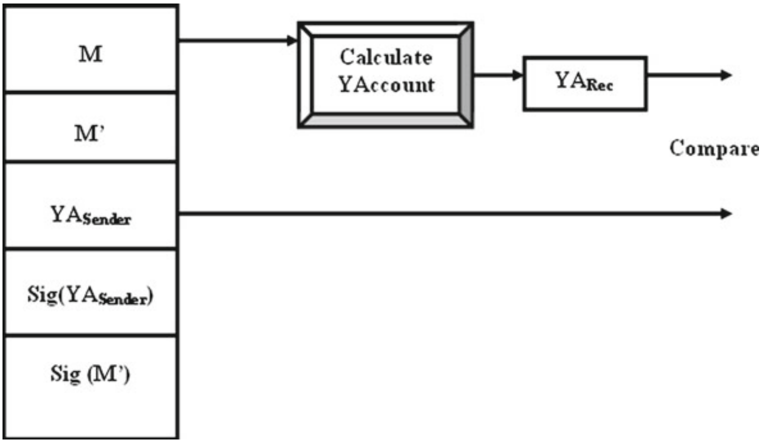
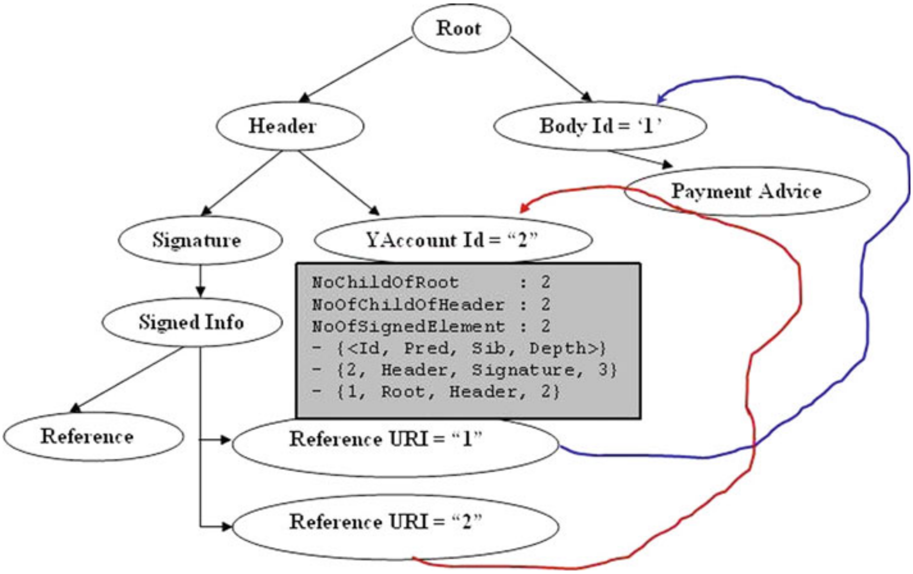


Fig. 12 YAccount after rewriting attack

3.3.1 Insertion of bogus entry

This attack is a type of modification of the message structure, by inserting bogus entries. The scenario is as follows: A customer, Alice, issues a payment voucher for 1,000 Euros to be transferred from her account to the supplier's (Bob's) account (Fig. 2). Some malicious attacker intercepts this message and modifies it asking to transfer 5,000 Euros instead of 1,000 Euros (Fig. 3). The elements related to the payment amount is not explicitly shown in the figures.

In our scheme, first the tier 0 mechanism checks the payment advice. If a common metafile and YASchema are agreed upon by the sender and receiver, the structure at the receivers side is checked against the metafile and YASchema constructs. Anomalies of the document structure, especially in the modified bogus header part, is the indication that the message has been modified. If verification with the metafile is successful, then the received document is



```
NoChildOfRoot      : 2
NoOfChildOfHeader  : 2
NoOfSignedElement  : 2
- {<Id, Pred, Sib, Depth>}
- {2, Header, Signature, 3}
- {1, Root, Header, 2}
```

Fig. 13 Payment advice—YAccount before the attack

checked as per the assertions and specifications mentioned in the YASchema. There will not be any specifications for the added bogus header entry in the YASchema, and this document will be declared as tampered.

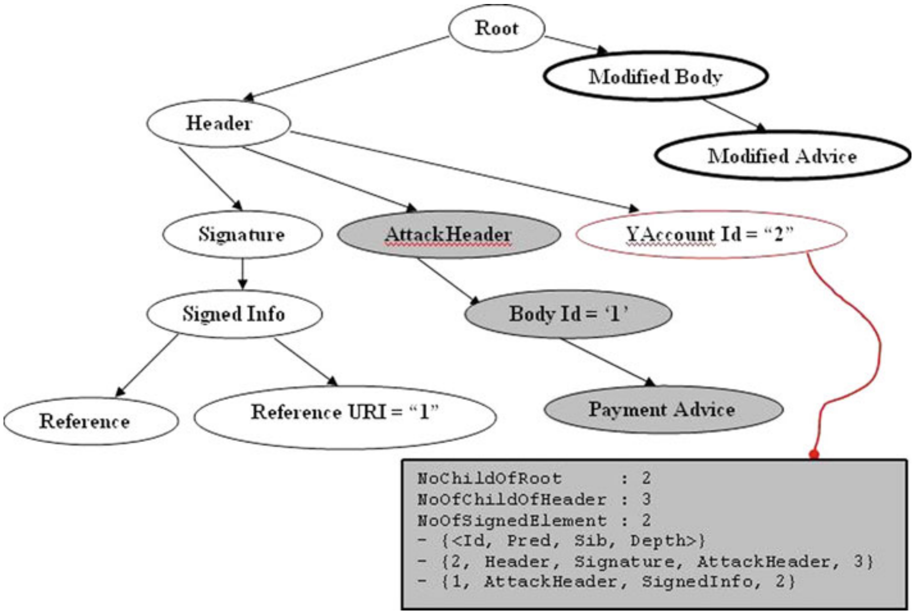
The metafile and YASchema can also be part of the message being transferred. In such cases, the digest values of the metafile and YASchema are also attached with the message by the sender. These digest values are calculated again by the receiver and compared against the digest values received. In this attack, if the metafile or YASchema is modified by the attacker, it is reflected in the digest values. Hence, at this stage itself, the attack can be detected. If there is a match, the received structure is verified against the metafile and YASchema for anomalies.

After successfully completing the tier 0 test, tier 1 test is carried out. The sender prepares the YAccount for the payment advice (Fig. 13). The existing structure of the payment advice is reflected in the YAccount information. The information is sent to the receiver. The attacker intercepts the message, modifies it, and sends it to the receiver in the next hop. The receiver now has to detect the attack on the message.

When the message is received, the receiver validates the YAccount information. The YAccount is calculated again (Fig. 14). The number of children of the root and the number of signed elements remain the same. But the number of header elements increases to 3 after the attack due to the addition of the bogus entries. Dissimilarities are also evident in the predecessor, sibling, and depth information of the signed elements. Due to the clear mismatch in the resulting scenario, the receiver discards the message. In addition, if an attacker changes the YAccount information according to the modified message contents, then this message will be invalidated by the receiver while validating the signature of the signed YAccount by the originator.

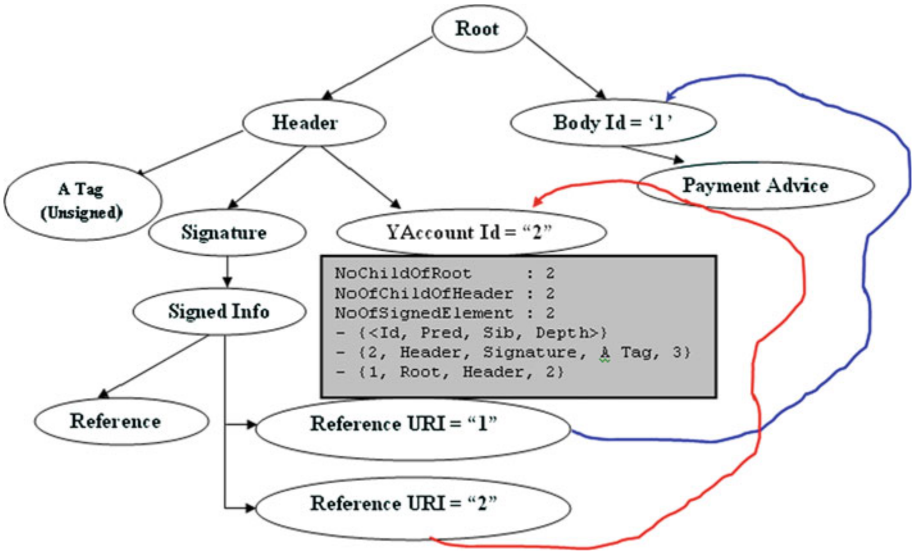
3.3.2 Header vulnerability exploitation

There is a possibility of at least one unsigned element to exist within the Header. An attacker can take advantage of this situation, as the YAccount structure is not taking care of the



```
NoChildOfRoot      : 2
NoOfChildOfHeader  : 3
NoOfSignedElement  : 2
- {<Id, Pred, Sib, Depth>}
- {2, Header, Signature, AttackHeader, 3}
- {1, AttackHeader, SignedInfo, 2}
```

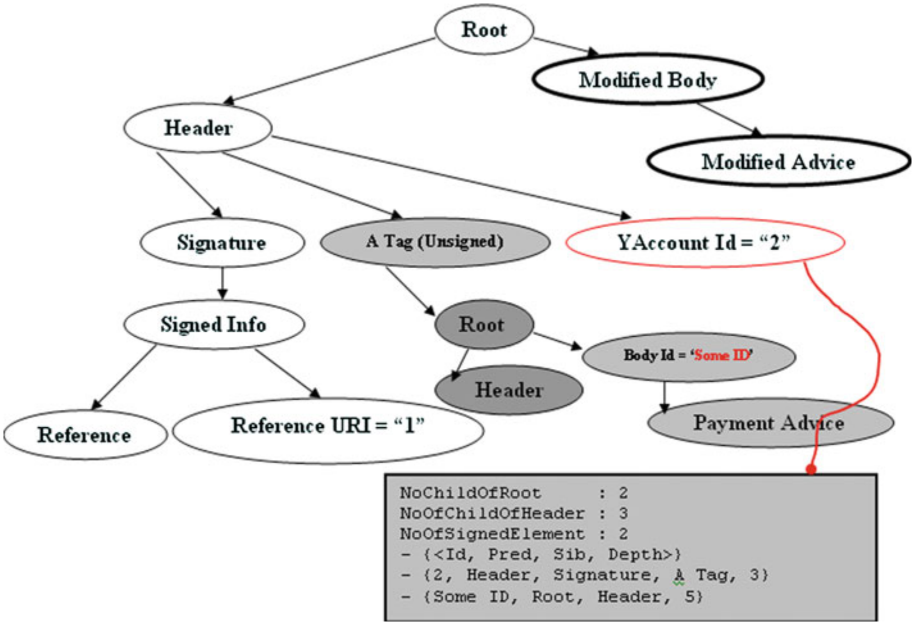
Fig. 14 Payment advice—YAccount after the attack



```
NoChildOfRoot      : 2
NoOfChildOfHeader  : 2
NoOfSignedElement  : 2
- {<Id, Pred, Sib, Depth>}
- {2, Header, Signature, A Tag, 3}
- {1, Root, Header, 2}
```

Fig. 15 Payment advice with an unsigned tag in the header—YAccount before the attack

unsigned elements. Such a scenario is shown in Fig. 15. The YAccount information calculated is also shown in Fig. 15. An attacker can move the body part (which contains the actual payment to be made) including its parent and sibling elements beneath "A Tag" and can add a bogus element at the original position (Fig. 16).



```
NoChildOfRoot      : 2
NoOfChildOfHeader : 3
NoOfSignedElement : 2
- {<Id, Pred, Sib, Depth>}
- {2, Header, Signature, A Tag, 3}
- {Some ID, Root, Header, 5}
```

Fig. 16 Payment advice with an unsigned tag in the header—YAccount after the attack (possibility I)

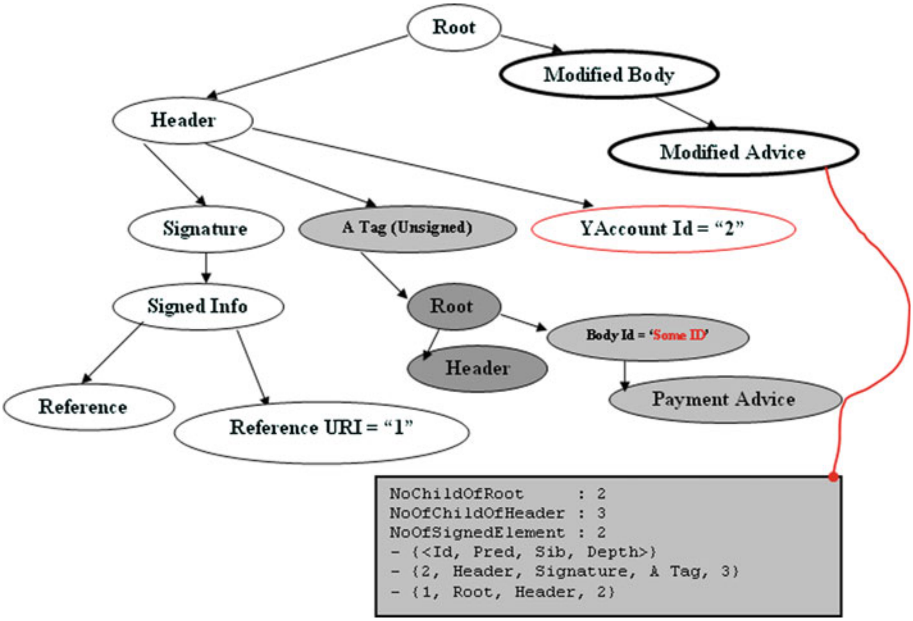
As described in the earlier case, verification with a shared metafile and YASchema commonly agreed upon by the sender and receiver can report the anomalies in the received document, if any.

Tracing the attack in tier 1 is as follows. From Fig. 15, it can be observed that the moved element preserves its relationship with the predecessor and siblings. Hence, the newly calculated YAccount information provides the same information as before for (i) number of children under root as 2, (ii) number of children of the Header as 3, and (iii) number of signed elements as 2. But the list of ID, predecessor, siblings, and depth information are of importance here. Even if the attacker directs the reference to the forged element so that he can be safe from a signature anomaly (see Fig. 16), then also, the ID and depth information will be different. The ID value of the forged element will be different, as the ID is to be unique. If this restriction cannot be forced due to any reason, the depth information will be different as the structure is altered.

However, there is a second possibility. The attacker can still use the forged element for the YAccount calculation (Fig. 17). In this case, the entries in the YAccount received and sent by the sender will be the same. But the anomaly will be traced in the signature comparison step. This addresses the hole discussed in Gajek et al. [50].

4 Performance evaluation

Our primary focus is on verbosity reduction of the document and thereby the overall conservation of energy. The five categories of data listed in Table 1 are used for the evaluation purpose. For each category of messages, two versions of the same message in each category are serialized (YAML and XML serializations). YAML serializations are evaluated in three formats—plain format and after size reduction phase I (YAML Phase I), and after size



```
NoChildOfRoot      : 2
NoOfChildOfHeader  : 3
NoOfSignedElement  : 2
- (<Id, Pred, Sib, Depth>)
- (2, Header, Signature, A Tag, 3)
- (1, Root, Header, 2)
```

Fig. 17 Payment advice with an unsigned tag in the header—YAccount after the attack (possibility II)

reduction phase II (TYAML). XML is selected as the benchmark format for performance comparison.

4.1 Verbosity

We first compare the verbosity of YAML datasets with that of existing alternative non-binary serializations. Then, we compare TYAML with YAML.

4.1.1 YAML versus existing alternative non-binary serializations

The message sizes of three YAML formats are compared against ONX, JSON and SOX formats (Figs. 18, 19, 20, respectively). These three formats are chosen as representatives of the alternative non-binary serializations. Only *Small*, *Medium*, and *Composite* formats are chosen for comparison. The following points are observed: (i) for *Small* category, all formats exhibit more or less similar performance except for the SOX format that outperforms YAML, and (ii) when the complexity of the dataset increases, YAML takes the minimum number of bytes than the other serializations. Hence, the biggest advantage for YAML is in the *composite* category. This result justifies the choice of YAML for the proposed work.

4.1.2 TYAML versus YAML

We observe that YAML is occupying the least number of bytes for data representation among the existing non-binary standards. From Fig. 21, we observe that TYAML performs better than YAML in all message categories. This demonstrates the effect of the verbosity reduction measures applied to YAML. It is evident that TYAML performs better than all other non-binary formats.

Fig. 18 Verbosity—YAML versus alternative non-binary formats—*small* category

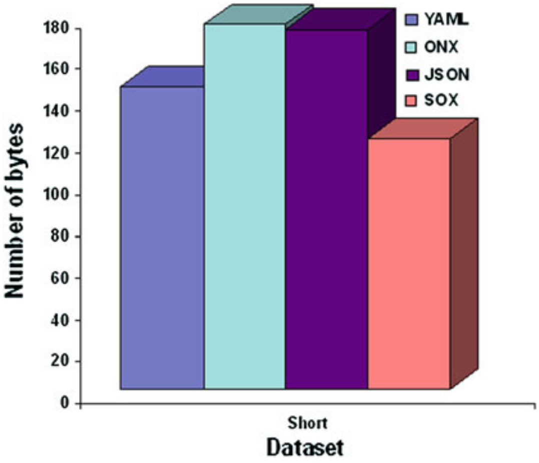


Fig. 19 Verbosity—YAML versus alternative non-binary formats—*medium* category

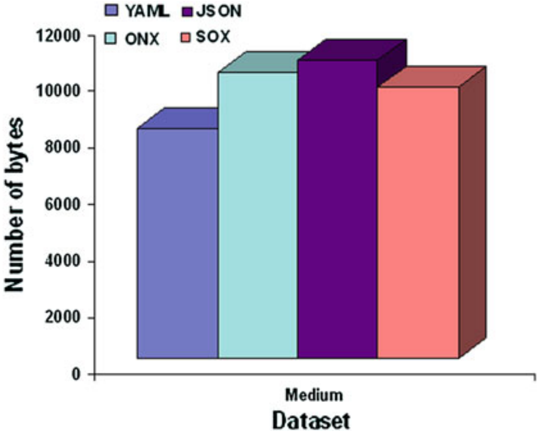
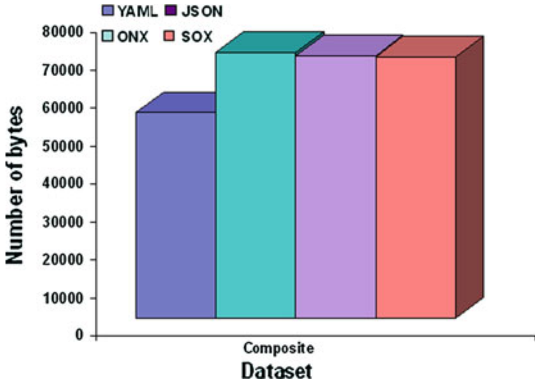


Fig. 20 Verbosity—YAML versus alternative non-binary formats—*composite* category



4.2 Content density

Content density (CD) is a measure to assess the amount of structure in an XML document. The computation of this metric is explained in White et al. [57].

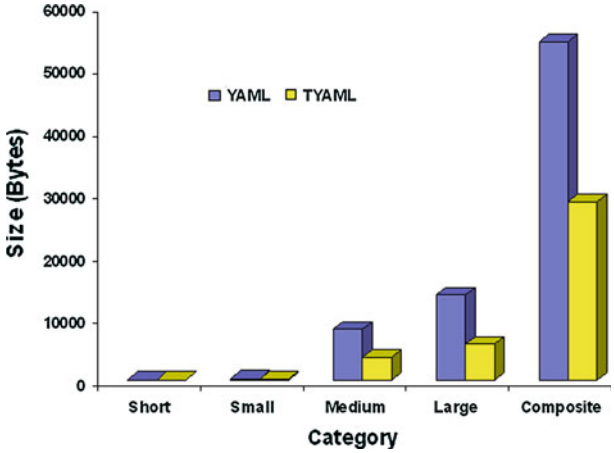


Fig. 21 Verbosity—TYAML versus YAML

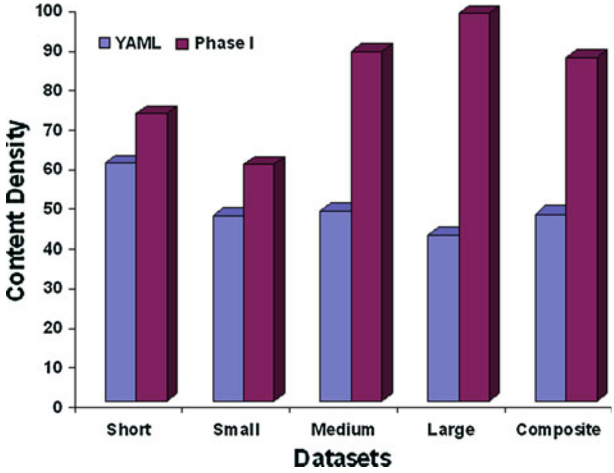


Fig. 22 Content density—YAML versus YAML phase I

4.2.1 YAML versus YAML phase I

CDs of YAML and YAML Phase I are compared in Fig. 22. All categories of datasets report higher CD values in YAML Phase I. It is an indication about the verbosity reduction achieved after Phase I thinning of YAML.

4.2.2 TYAML versus XML

A comparison of CDs of TYAML and XML is shown in Fig. 23. As in the previous case, all categories of datasets report higher CD values in TYAML. Hence, it can be concluded that our verbosity reduction measures have helped in increasing the value of CD of the TYAML format considerably.

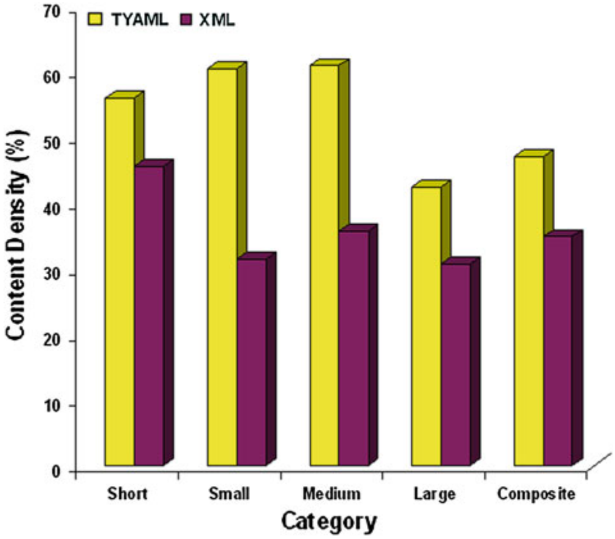


Fig. 23 Content density—TYAML versus XML

4.3 Energy consumption

The energy-conservation efforts should consider the cost of transmitting, receiving, and even discarding packets [58]. The states in which the energy is consumed are (i) transmit (when the node is transmitting packets), (ii) receive (when the node is receiving packets), (iii) idle (when the node is waiting for any packet transfers), and (iv) sleep (when the node can neither receive nor transmit). Among these, only (i) and (ii) are the cases of useful energy consumption.

The linear relationship between energy consumed and packet size is represented by the equation

$$\text{Cost} = m \times \text{size} + b \tag{1}$$

where "m" is the incremental cost to the packet size and "b" is the fixed cost associated with channel acquisition [58]. We make use of the energy model discussed in Allard et al. [59] to calculate the energy spent at each node due to a flow. The actual energy consumption can also be due to factors such as overhearing (reception of packets intended for other stations), unsuccessful (colliding) transmissions and reception of collisions other than successful transmission and successful reception. Since our objective is to find the impact of verbosity over energy consumption, we take only the energy consumption for transmission of data packets in a flow. We ignore the energy needed for acknowledgments (for simplifying the calculation). Accordingly, the base formula in Allard et al. [59] is modified for our calculations. The energy consumption in transmitting packets at node N due to another node M (depending upon whether the node N belongs to a flow or not, and where in the flow the node M is situated) in the network is as shown in Eq. 2.

$$E_{N/M} = T_{M>0}(T_{M=N} E_{T_{\text{pck}}}) \tag{2}$$

where $E_{N/M}$ = energy spent at node N due to node M, $E_{T_{\text{pck}}}$ = energy spent for transmission of one data packet, and $T_P = 1$ if P is true, 0 otherwise.

Table 2 Simulation parameters

Parameter	Values
Channel type	Channel/WirelessChannel
Radio propagation model	Propagation/TwoRayGround
Network interface type	Phy/WirelessPhy
Link-/MAC-layer	IEEE 802.11
Interface queue	Queue/DropTail/PriQueue
Antenna model	Antenna/OmniAntenna
Max packet in IFQ	50
Topology	500 × 400
Transmission range (m)	250
Bandwidth	10 Mbits/s
Routing protocol	AODV

Table 3 Record sets—categories

Type	YAML	Phase I	TYAML	XML
Small	145	120	117	191
Medium	320	281	260	445
Composite	2,171	1,182	1,144	5,881

We use NS-2 simulations to measure the packets generated during the transmission. To simulate the scenario, a network configuration needs to be defined. The node configuration setting is according to Table 2. The configuration for a wireless node assigns the AODV routing algorithm as its ad hoc routing protocol, the link layer type as LL, and the Medium Access Control (MAC) protocol as IEEE 802.11. The queue between the MAC layer and the link layer is used. A "tcp" agent is attached to the source node, and a connection is established to a tcp "sink" agent attached to the destination node. A "ftp" traffic generator is attached to the "tcp" agent. The size of the data packets is set as 1.5 kilobytes. For the simulation, the typical transmission and reception costs for Lucent Silver card as specified in Feeney et al. [58] are used. The transmission power used is 1.3 W, and reception power is 0.9 W. The simulation time varies according to the size of the record sets (tabulated in Table 3). For example, in the case of YAML, the *Small* dataset size is 145 bytes. Simulation is run for 116 secs, which is the theoretical time to transfer 145 terabytes. This variation is applied to study the effect of the number of packets being generated and transmitted for different message categories.

It is observed from Fig. 24 that (i) the *Small* dataset reports major gains than the other two categories, (ii) TYAML displays gains in all categories, (iii) XML reports the maximum energy consumption in all categories, and (iv) TYAML demonstrates better performance than the plain YAML format. This indicates that the verbosity reduction results in the reduction of energy consumed. The reduction in the number of packets results in reduced network traffic and less depletion of the memory space.

4.4 Formatting overhead

In this section, we compare the sizes of the additional data structures needed for the proposed scheme.

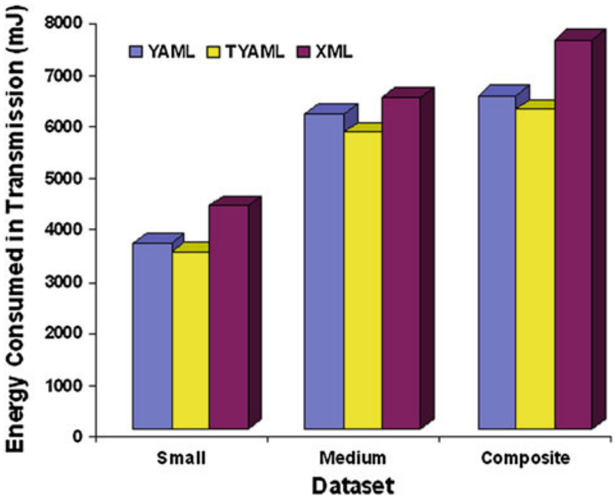


Fig. 24 Energy consumption during transmission of packets

Table 4 Dataset size versus metafile size (in bytes)

	Dataset	Metafile
Small	117	19
Small	215	68
Medium	3,727	91
Large	5,872	105
Composite	28,642	112

4.4.1 Size comparison between dataset and metafile

Metafile is an additional payload used for thinning and trust management. The sizes of datasets and the corresponding metafiles are as shown in Table 4. It can be seen that the metafile size is relatively very small compared to the dataset size. Thus, we conclude that the introduction of the metafile structure does not have a considerable impact on the overall payload to be transmitted.

4.4.2 Size comparison between WS policy, SOAP account, and YAccount

The WS Policy file, SOAP Account, and YAccount are compared in Fig. 25. It is observed that (i) WS Policy files are larger in size compared to SOAP Account and YAccount, (ii) for smaller categories, the size difference between SOAP Account and YAccount is negligible, and (iii) YAccount files are the shortest for all types of datasets.

4.4.3 Size comparison between WS policy and YASchema

Figure 26 compares the WS Policy files and YASchema. For *Short*, *Small* and *Medium* datasets, YASchema takes less number of bytes than WS Policy. But for the remaining categories, WS Policy file outperforms YASchema.

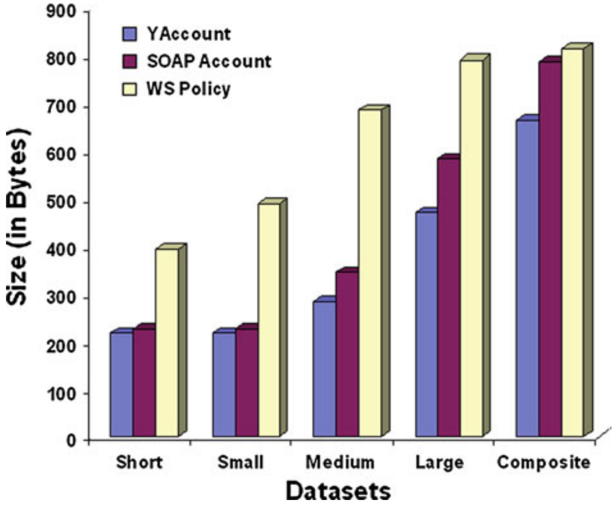


Fig. 25 Size comparison: WS policy, SOAP account, and YAccount

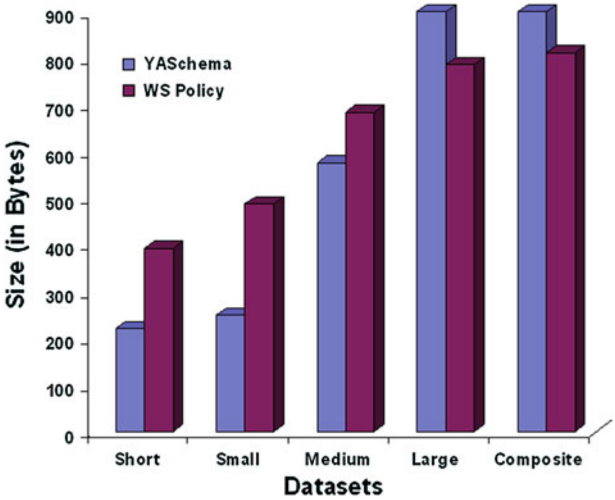


Fig. 26 Size comparison: WS policy and YASchema

YASchema has additional entries other than the policy assertions like data types and order of occurrences. In spite of that YASchema is performing better than WS Policy files in all the 3 categories. The drawback in the larger categories is an indication of the need for careful choice of the policy rules in YASchema.

4.4.4 Relative size comparison between WS policy, YASchema, and YAccount with respect to the dataset

The growth % of WS Policy, YASchema, and YAccount constructs with respect to the size of the datasets is as shown in Fig. 27. It is observed that (i) YAccount occupies the minimum number of bytes, (ii) the advantage of YAccount is more visible in the smaller categories of

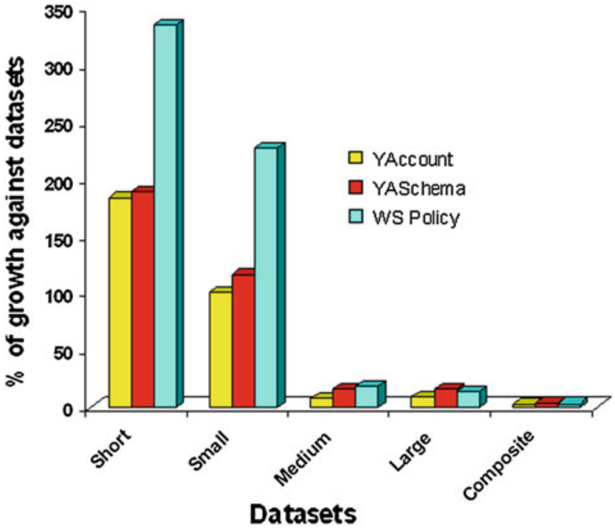


Fig. 27 Size comparison (relative to dataset size)—WS policy, YASchema, and YAccount

datasets, and (iii) YASchema, in spite of the additional constructs, performs better than the WS Policy file in smaller categories (*Short*, *Small*, and *Medium*).

5 Conclusions

The trade-off between data quality and resource usage is a major concern in RCNs. Measures to tackle the "inflation problem" of information is a right step toward conserving resource usage. This paper presented a schema-centric approach for trusted exchange of document, using a document exchange format with reduced verbosity.

The proposed document exchange scheme was based on YAML, with the introduction of a schema and thinning measures. The good features of this format included less verbosity (without making use of any compression or binary encoding), schema definition, use of YASchema as a policy file, a structure known as YAccount with novel measures, rewriting attack detection with a refined signature specification, and a two-tier mechanism to check the rewriting attacks. Performance evaluation confirmed that the proposed scheme performs better than the existing popular formats. The limitation of our scheme is the compatibility issues for the signature (considering the abundant use of the existing format).

Several enhancements are needed to furnish this as a complete messaging standard. Measures such as "schema hardening" are to be incorporated to make the format more resilient to rewriting attacks. TYAML/YASchema combination can be fully geared up to contain the attribute-based access control mechanism as envisaged in policy languages like eXtensible Access Control Markup Language (XACML). Also, more policy assertions can be included in the YASchema to fine tune it as a robust policy file. TYAML/YASchema combination can be refined to be used as a content language in Agent Communication Languages (ACL). These are the suggested topics for further research.

Acknowledgments We thank anonymous reviewers for their very useful comments and suggestions.

References

1. Li H (2000) XML and industrial standards for electronic commerce. Knowl Inform Syst 2(4):487–497
2. Hanslo WS, MacGregor KJ (2003) Using XML messaging for wireless middleware communication. In: Proceedings of the South African telecommunication network & applications conference, (SATNAC '03), Southern African telecommunication networks and applications conference, Southern Cape, South Africa, September 2003. North-Holland Publishing Co
3. Kangasharju J, Lindholm T, Tarkoma S (2007) XML messaging for mobile devices: from requirements to implementation. Int J Comput Telecommun Netw 51(16):4634–4654
4. Lee H-H, Lee W-S (2010) Consistent collective evaluation of multiple continuous queries for filtering heterogeneous data streams. Knowl Inform Syst 22(2):185–210
5. Ng W, Lam W-Y, Wood PT, Levene M (2005) XCQ: a queriable XML compression system. Knowl Inform Syst 4:421–452
6. Gupta M, Tu M, Khan L, Bastani F, Yen I-L (2005) A study of the model and algorithms for handling location-dependent continuous queries. Knowl Inform Syst 8:414–437
7. Michael MP (2005) Energy awareness for mobile devices. In: Research seminar on energy awareness. University of Helsinki
8. Potlapally N, Ravi S, Raghunathan A, Niraj K (2006) Heterogeneous grid computing for energy constrained mobile device. IEEE Trans Mobile Comput 5:128–143
9. Gu GQ, Lu J-Z (2008) Some issues of computer networks: architecture and key technologies. J Comput Sci Technol 21(5):708–722
10. Artail H, Shihab M, Safa H (2009) A distributed mobile database implementation on pocket PC mobile devices communicating over Bluetooth. J Netw Comput Appl 32(1):96–115
11. Orthogonal frequency-division multiplexing. Available at http://en.wikipedia.org/wiki/OFDM/
12. Lai Y-X, Chen Y-L, Chen H (2008) PEJA: progressive energy-efficient join processing for sensor networks. J Comput Sci Technol 6(6):957–972
13. Kangasharju J, Lindholm T, Tarkoma S (2008) XML security with binary XML for mobile web services. Int J Web Serv Res 5(3):1–19
14. Botia JA, Gomez-Skarmeta AF, Doung HH, Demeure I (2010) A context-sware data sharing service over MANet to enable spontaneous collaboration. In: Proceedings of the IEEE workshop on enabling technologies: infrastructure for collaborative enterprises (WETICE '08), 17th workshop on enabling technologies: infrastructure for collaborative enterprises. IEEE Computer Society, Washington, DC, USA, pp 159–164
15. Blaya JAB, Demeure I, Gianrossi P, Lopez PG, Navarro JAM, Meyer EM, Pelliccione P, Tastet-Cherel F (2009) POPEYE: providing collaborative services for ad hoc and spontaneous communities. Service oriented computing and applications, vol 3, no 1. Springer, London, pp 25–45
16. Leite B, Bezerra D, de Assis F, de Carvalho T (2010) Symbolic data analysis tools for recommendation systems. Knowl Inform Syst 26(3):385–418
17. Mrida D, Baldiris S, Fabregat R, Velez J, Huerva D (2007) A user model that incorporates characteristics of access devices in MAS-SHAAD. In: Proceedings of the TUMAS-A workshop (UM 2007), 11th international conference on user modeling, Corf, Greece, pp 51–55
18. Rosaci D, Sarné GML, Garruzzo S (2009) MUADDIB: a distributed recommender system supporting device adaptivity. ACM Trans Inform Syst 27:1–41
19. Mrida D, Fabregat R, Prat X, Huerva D, Velez J (2008) A dynamic content generator for adaptation in hypermedia systems. Lecture notes in computer science, vol 5149. Springer, Berlin, pp 320–323
20. Göksedef M, Şule G-O (2010) A dynamic content generator for adaptation in hypermedia systems. Lecture notes in computer science, vol 37, no 4. Pergamon Press Inc, pp 2911–2922
21. Rosaci D, Sarn GML (2006) MASHA: a multi-agent system handling user and device adaptivity of Web sites. User modeling and user-adapted interaction, vol 16, no 5. Springer, the Netherlands, pp 435–462
22. Malandrino D, Mazzoni F, Riboni D, Bettini C, Colajanni M, Scarano V (2009) MIMOSA: context-aware adaptation for ubiquitous web access. Person Ubiq Comput 14(4):301–320
23. Beach A, Gartrell M, Xing X, Han R, Lv Q, Mishra S, Seada K (2010) A context-sware data sharing service over MANet to enable spontaneous collaboration. In: Proceedings of the workshop on mobile computing systems & applications (WETICE '08), eleventh workshop on mobile computing systems & applications. Annapolis, Maryland ACM, New York, NY, USA, pp 60–65
24. Tamine-Lechani L, Boughanem M, Daoud M (2009) Evaluation of contextual information retrieval effectiveness: overview of issues and research. Knowl Inform Syst 24(1):1–34

25. Kumar A, Shankar R, Momaya K, Gupte S (2010) The market for wireless electricity: the case of India. Ener Policy 38(3):1537–1547
26. Tella A, Oluyemisi DY (2010) The future of ICT in developing world: forecasts on sustainable solutions for global development. Ind J Lib Inform Sci 4(2):115–131
27. United Nations Development Programme (2010) Energy for a sustainable future. The secretary-generals advisory group on energy and climate change summary report and recommendations. UNDP, New York
28. Aker JC, Mbiti IM (2010) Mobile phones and economic development in Africa. J Econ Perspect 24(3):207–232
29. Snoli L (2010) The rise of the ultra-low-cost mobile phone. Available at http://www.itworld.com/personal-tech/130741/the-rise-ultra-low-cost-mobile-phone
30. Kounavis C (2010) Access to mobile networks available to over 90% of world population 143 countries offer 3G services. Available at http://www.wirelessresearch.eu/archives/349
31. Benameur A, Kadir FA, Fenet S (2008) XML rewriting attacks: existing solutions and their limitations. In: Proceedings of the international conference on applied computing (IADIS 2008). Algarve, Portugal, CoRR, IADIS Press. Available at http://arxiv.org/abs/0812.4181
32. Bartel M, Boyer J, Fox B, LaMacchia B, Simon E (2008) XML signature syntax and processing, 2nd edn. W3C recommendation. Available at http://www.w3.org/TR/xmldsig-core/
33. Sinha SK, Benameur A (2008) A formal solution to rewriting attacks on SOAP messages. In: Proceedings of the 2008 ACM workshop on secure web services (SWS '08) Alexandria, Virginia, USA, 2006, November. ACM, New York, pp 53–60
34. Rahaman MA, Rits M, Schaad A (2006) An inline approach for secure SOAP requests and early validation. In: Proceedings of the OWASP Europe 2006 conference (OWASP '06). Leuven, Belgium, pp 19–33
35. Eggenberger M, Prakash N, Matsumoto K, Thurmond D (2009) Policy based messaging framework systems. Lecture notes in computer science, vol 4749. Springer, Berlin, pp 497–505
36. Zhao G, Chadwick D (2005) Trust infrastructure for policy based messaging in open environments. In: Proceedings of the 14th IEEE international workshops on enabling technologies: infrastructure for collaborative enterprise, (WETICE '05). IEEE Computer Society, Washington, DC, pp 144–149
37. Chandramouli R, Subbalakshmi P (2001) Wireless LAN: issues and challenges. Available at http://www.ece.stevens-tech.edu/~mouli/WSTAarticle1.doc
38. Girardot M, Sundaresan N (2000) Millau: an encoding format for efficient representation and exchange of XML over the Web. Comput Netw 33(1–6):747–765
39. Cai M, Ghandeharizadeh S, Girardot M, Schmidt R, Song S (2004) Xebu: a binary format with schema-based optimizations for XML data. In: Proceedings of the 13th international conference on database and expert systems applications (DEXA 2002). LNCS vol 2453. Springer, Berlin, pp 269–289
40. Kangasharju J, Tarkoma S, Lindholm T (2005) Xebu: a binary format with schema-based optimizations for XML data. In: Proceedings of the international conference on web information systems engineering (WISE '05), sixth international conference on web information systems engineering. LNCS vol 3806. Springer, New York, pp 528–535
41. Martin B, Jano B (2007) WAP Binary XML content format. W3C note, June 1999. Available at http://www.w3.org/TR/wbxml/
42. Ben-Kiki O, Evans C, dt Net I (2010) YAML Aint́ markup language (YAML) version 1.2, 3rd edn. Available at http://www.yaml.org/spec/1.2/spec.html
43. Crockford D (2007) The application/json Media Type for JavaScript Object Notation (JSON). Available at http://tools.ietf.org/html/rfc4627
44. Jacobs S (2005) Open node syntax version 0.6.9. Available at http://www.seairth.com/web/onx/onx.html
45. Simple Outline XML: SOX (2002) WAP binary XML content format. Available at http://www.langdale.com.au/SOX/
46. Bajaj S, Box D, Chappell D, Curbera F, Daniels G, Hallam-Baker P, Hondo M, Kaler C, Langworthy D, Malhotra A, Nadalin A, Nagaratnam N, Nottingham M, Prafullchandra H, von Riegen C, Schlimmer J, Sharp C, Shewchuk J (2004) Web services policy framework (WS-Policy). Available at http://www.ibm.com/developerworks/library/specification/ws-polfram/
47. Nadalin A, Kaler C, Hallam-Baker P, Monzillo R (2004) Web services security: SOAP message security 1.0 (WS-Security 2004). Available at http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf
48. Bhargavan K, Fournet C, Gordon AD, O'Shea G (2005) An advisor for web services security policies. In: Proceedings of the 2005 workshop on secure web services (SWS '05). ACM, Fairfax, pp 1–9
49. Rahaman MA, Rits M, Schaad A (2007) SOAP-based secure conversation and collaboration. In: Proceedings of the 2007 IEEE international conference on web services (ICWS 2007). IEEE Computer Society, Salt Lake City, pp 471–480

50. Gajek S, Liao L, Schwenk J (2007) Breaking and fixing the inline approach. In: Proceedings of the 2007 ACM workshop on secure web services, (SWS '07). ACM, New York, pp 37–43
51. Abdul Haleem PP, Sebastian MP (2008) An alternative approach for slicing down the message size and enhancing the security in wireless mobile network. Mediterr J Comput Netw 5(4):148–159
52. Qin J, Taffet MD (2004) Vocabulary use in XML standards in the financial market domain. Knowl Inform Syst 6:269–289
53. Lanka S, Parikh P (2000) XML shredding. Midterm project. New York University, Courant
54. Unal O, Afsarmanesh H (2010) Semi-automated schema integration with SASMINT. Knowl Inform Syst 23(1):421–452
55. Box D, Hondo M, Kaler C, Maruyama H, Nadalin A, Nagaratnam N, Patrick P, von Riegen C, Shewchuk J (2003) Web services policy assertions language (WS-PolicyAssertions). Available at http://xml.coverpages.org/ws-policyassertionsV11.pdf
56. Della-Libera G, Gudgin M, Hallam-Baker P, Hondo M, Granqvist H, Kaler C, Maruyama H, McIntosh M, Nadalin A, Nagaratnam N, Philpott R, Prafullchandra H, Shewchuk J, Walter D, Zolfonoon R (2005) Web services security policy language (WS-SecurityPolicy). Available at http://specs.xmlsoap.org/ws/2005/07/securitypolicy/ws-securitypolicy.pdf
57. White G, Kangasharju J, Brutzman D, Williams S (2007) Efficient XML interchange measurements note. Available at http://www.w3.org/TR/exi-measurements/
58. Feeney LM, Nilsson M (2001) Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In: Proceedings of the annual joint conference of the IEEE computer and communications societies, (INFOCOM 2001), twentieth annual joint conference of the IEEE computer and communications societies, vol 3. Alaska, pp 1548–1557
59. Allard G, Minet P, Nguyen D-Q, Shrestha N (2006) Evaluation of the energy consumption in MANET. Lecture notes in computer science, vol 4104. Springer, Berlin, pp 170–183

Author Biographies

P. P. Abdul Haleem received his B.E. degree from Bangalore University, India, in 1994 and his masters and PhD degrees in Computer Science & Engineering from National Institute of Technology Calicut, India in 2005 and 2011, respectively. His research interests include Mobile Computing, XML Technologies and Semantic Web. He is currently Professor & Head in the Department of Computer Science & Engineering at MES College of Engineering, Kuttippuram, Kerala, India.

M. P. Sebastian is currently Professor and Area Chairman of Information Technology & Systems at Indian Institute of Management Kozhikode, India. He received his masters and PhD degrees in Computer Science & Engineering from the Indian Institute of Science, Bangalore. He started his career as a Scientist at Space Applications Centre (ISRO), Ahmedabad. Then he served at Bharat Electronics Ltd, Bangalore and National Institute of Technology Calicut. His areas of teaching/ research interest include Cloud Computing, Enterprise Resource Computing, Information Security Management, IT Strategy, Networks Management and Software Project Management, and have published many research papers. He is on the editorial board of many journals and is a reviewer of international journals including IEEE Transactions on Vehicular Technology, American Optics Letters, Wiley InterScience Journal on Wireless Communications and Mobile Computing, and IIMB Management Review.