Innovations in Mobile Multimedia Communications and Applications:

New Technologies

Ismail Khalil
Johannes Kepler University Linz, Austria

Edgar Weippl
SBA Research, Austria

Information Science
REFERENCE

ISBN:978-1-60960-563-
(Hardcopy)

978-1-60960-564-
(ebook)

Table of Contents

Chapter 2
Optimizing Resource Consumption for Secure Messaging in Resource Constrained Networks

P. P. Abdul Haleem
National Institute of Technology, India

M. P. Sebastian
Indian Institute of Management, India

ABSTRACT

Conservation of resources such as bandwidth, energy and memory are of a concern in Resource Constrained Networks (RCNs). Wireless mobile devices, especially low cost devices are stifled by the limited resources such as battery power, screen size, input, memory and processors. The low cost wireless mobile devices penetrating the developing world market demand for a cost effective messaging format that fits within the constrained wireless environment. Reduction of verbosity is considered to be one of the most effective steps in controlling the resource consumption in RCNs. This chapter presents a method for optimizing resource consumption by the use of a new messaging format with less verbosity. The proposed format is based on YAML Ain't Markup Language (YAML), which is further enhanced with message level security specifications.

INTRODUCTION

The abundant growth of Internet services and the increased number of wireless mobile users resulted in the penetration of wireless mobile devices into the realm of wired networks. This penetration raised many issues related to wireless mobile networks and devices. Wireless mobile devices (with limited resources like battery power, memory, processing power, input and screen) together with wireless networks (with unreliable channels, low bandwidth, increased latency,

DOI: 10.4018/978-1-60960-563-6.ch002

increased rate of retransmission of lost packets, and weak security features) impose a number of limitations. However, users expect to access much wider range of applications in wireless mobile devices than in the conventional devices.

A major constraint in the wireless mobile devices is power. Even with the latest processors and memory chips, the rechargeable battery attached to the mobile devices still tends to fall behind the expectations. The RF part of cellular engine, which is responsible for transmission and reception of messages, is the biggest consumer of energy in a mobile device (Michael, 2005). Security protocols also consume power – the number of packets transmitted or received and the size of the keys are two important factors to be taken care to conserve energy. Wireless networks are generally less reliable, less secure and with more message latency.

Verbosity reduction of messages is a major concern in the constrained wireless mobile environment (Kangasharju, Lindholm, & Tarkoma, 2008). Reducing the verbosity of messages may have an impact cutting across several layers of the wireless networking protocols. XML is the default protocol for messaging ("SOAP-Tutorial", 2007). However XML faces many problems when used in the wireless mobile environment (Kangasharju, 2005). The number of bytes required for data representation is huge in XML. Due to this verbosity, XML buffers need to be flushed more often at the time of input and output, leading to lesser throughput. Also, larger messages are vulnerable to retransmissions. The highly textual nature of XML makes the string parsing compulsory for further processing. XML documents are structured and this adherence of the document to the accompanying structure is to be verified by the parsers. XML has been one of the easy targets for hackers, due to its long term use and universality.

Distributed computing and systems have changed significantly with the introduction of wireless communication and mobile nodes. These differences are not only quantitative (in that the links and nodes have inferior performance), but also qualitative (in that the characteristics also differ from fixed links and stationary computers). In tune with the widespread use and popularity of wireless mobile devices, complex operations in distributed and collaboration technologies allow people to move across organizational boundaries and to collaborate among/in organizations and communities 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 in this context are (i) support for virtual web communities, (ii) support for web recommender systems that are adaptive to the ubiquitous devices, and (iii) adaptive content generation and delivery. Present day popular applications like Web services and agent communications languages (ACLs) also make use of a data format.

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 upon 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 (Tamine-Lechani, Boughanem, & Daoud, 2009). 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 the data in Collaborative Working Environments (CWEs).

Reduction of message size results in the reduction of number of transmissions, which in turn reduces the data transmission cost and storage

space (Lai, Chen, & Chen, 2008). It is interesting to note that, in the wireless world, reducing the message size is of primary importance than the processing efficiency gains through an alternative format (Kangasharju et al., 2008). Hence it is clear that the verbosity of the data format is the major factor which directly influences all the constraints, including the energy consumption, of wireless mobile devices and networks.

Coupled with this, the application of a lightweight cryptographic system suitable to the wireless mobile environment can reduce the power consumption considerably. ECC is already in use in the wireless mobile security arena due to its dramatic decrease in the key size and running time without compromising the security. A 160-bit ECC key offers the same level of security as a 1024-bit RSA key (Chou, 2003). The energy analysis on various cryptographic algorithms and key exchange protocols reveals considerable savings with ECC without compromising in security (Potlapally, Ravi, Raghunathan, & Jha, 2003).

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. At the same time, low-cost devices are steadily increasing their market share, especially in developing countries. The projected shipments of ultra-low cost devices is estimated to be 24.3 million in the Asia Pacific region, as opposed to 0.6 million in the North America region (Gokran, 2007).

Thus, a messaging standard that performs well within the constraints of wireless mobile environments is a need. This chapter presents a messaging standard that avoids some of the limitations of XML (pertaining to the wireless mobile environment), retaining the good aspects. The suggested format has the following features: (i) simple and flexible, both for the user and the application programmer, (ii) editable and easily readable, (iii) less verbose without using compression/ binary encoding (i.e., without affecting the readability), (iv) facility for a schema definition, (v) consumes less bandwidth and needs short transmission time, (v) support for secured transmission of messages in transit, and (vi) support for ECC compatibility.

RELATED WORKS

Data Formatting Schemes

Design of alternative messaging formats to XML has received attention among the researchers and many proposals are available in the literature. The standard compression techniques, alternative formats for XML, caching (Devaram & Andresen, 2003) and binary encoding are the notable works in this area. It is worth mentioning that these methods propose good alternatives for the wired world. For instance, standard compression techniques such as Millau (Girardot & Sundaresan, 2000), Gzip (Cai, Ghandeharizadeh, Girardot, Schmidt, & Song, 2002), XMill (Cai et al., 2002) yield better performance with larger messages with a high redundancy rate. With SOAP, this is a working solution, as HTTP supports generic compression tools. But, it may not perform well with large sets of short messages, typical in the wireless mobile scenario. Compression routines will have to employ different compression techniques on different parts of the XML document (for elements and attributes). Also, a compression/decompression layer will be an additional overhead to the resource scarce wireless mobile devices (Schmelzer, 2004).

The binary encoding technique was employed to reduce the size of the messages. Wireless Application Protocol Binary XML (WBXML), also called Binary XML ("WAP Binary XML Content Format", 1999), is the oldest known format of this kind. The lack of support to the XML namespaces is considered to be a problem in utilizing WBXML as an alternative to XML (Sandoz & Pericas-Geertsen, 2005). The Xebu ("Xebu Se-

rialization Format", 2007) format builds up its tokenization during a stream of messages, instead of considering each XML document in isolation, as in most of the other formats. But this format remains neutral to human language as WBXML. Self descriptive nature and easy readability properties of XML are hampered due to the introduction of the above enhancements. Also the user always has to depend on a conversion process to convert to binary format and revert back to the original format – ease of preparation and readability are affected.

Several alternative serialization formats were proposed to reduce the verbosity of XML which include YAML, JavaScript Object Notation (JSON) (Crockford, 2006), Open Node Syntax (ONX) (Open Node Syntax, 2005) and Simple Outline XML (Simple Outline XML: SOX, 2002). These formats suggest methods for conserving memory. They also tend to maintain the agility of XML. Among these standards, we have chosen YAML as the base format due to the following reasons (Clark, 2002; YAML, 2007) (i) YAML's versatile block indent syntax allows formatting of structured data in a visually uncluttered way, making it exceptionally human readable, (ii) YAML packs the information with less number of bytes among majority of the existing standards, (iii) YAML is a superset of JSON syntax, (iv) YAML offers extensible data types beyond primitives (i.e., beyond strings, floats, ints and bools) which can include class-type declarations or Unicode types, (v) YAML is resistant to delimiter collision, (vi) it is easier to embed a YAML document or other type of documents within a YAML document due to YAML's insensitivity to quotes and braces in scalar values, (vii) YAML handles indents as small as a single space, and this may offer better compression than markup languages, (viii) YAML offers a simple relational scheme that allows repeats of identical data to be referenced from two or more points in the tree rather than entered redundantly at those points, and (ix) YAML's lack of an associated command language (seen as a relative security benefit as parsers at least should be safe to apply to tainted data without fear of a latent command-injection security hole).

Message Level Security

In addition to reducing the verbosity of the messages, secure transmission of the messages is also to be guaranteed. Security need to be ensured for the messages in transit either at transport level or at message level. The transport level security is based on the Secure Sockets Layer (SSL) or the Transport Layer Security (TLS) that runs beneath the HTTP. Message level security (Bertino & Martino, 2006) includes all the security benefits of SSL, with additional features. It is an application layer service and facilitates the protection of data between applications. Its main features include (i) does not use the heavy SSL, (ii) data chunks are protected, and (iii) intended to work with intermediaries. It can be more effective and has the added flexibility that the message can be sent over any transport protocol.

The advantages of message level security over transport layer security are: (i) provision for end-to-end security as opposed to the point-to-point security(End-to-end security is ideal when multiple intermediary nodes exist between the two endpoints), (ii) support for message security while in transit and at rest (as opposed to only while in transit on the wire in transport layer security), and (iii) support for element-wise signing and encryption. Hence, it is evident that message level security specifications need to be included along with the format specification to make it secure.

As stated earlier, resource conservation is of primary concern in RCNs. Security algorithms and protocols generally consume the major chunk of the limited available resources in RCNs. The main sources of power drain during a secure wireless session are the number of packets transmitted/ received and the size of messages required for establishing a session.

Implementation of ECC based security techniques are discussed in several papers (J.-H. Han, Kim, Jun, Chung, & Seo, 2002) (Edoh, 2004). Application of ECC cuts down the computational and communication costs without compromising the complexity of the process.

ECC is used in the design of the elliptical curve variations of Integrated Encryption Scheme (IES) and Signcryption. Elliptic Curve Integrated Encryption Scheme (ECIES) is a public-key encryption scheme based on ECC which provides semantic security against chosen-plaintext and chosen-ciphertext attacks (Certicom-Research, 2000). Signcryption is a method to reduce the cost of the signature-then-encryption method, by combining the functions of digital signature and public key encryption in a logically single step (Zheng, 1997a). Signcryption costs on the average 50% less in computation time and 91% less in message expansion than signature-then-encryption with RSA (Zheng, 1997b). Also, Signcryption technique provides faster and better security than conventional methods. Many papers are available making use of the effectiveness of Signcryption, in the wireless world (Peng & Li, 2005; Zhang et al., 2005; Park et al., 2006). Implementation of ECC based techniques are discussed in several papers (J.-H. Han et al., 2002; Edoh, 2004 for instance). Application of ECC cuts down the computational and communication costs without compromising the complexity of the process. Thus, it can be seen that IES and Signcryption with ECC appear to be better mechanisms in the constrained wireless mobile environment.

Generic specifications for encryption, signature and Signcryption are formulated in (Haleem & Sebastian, 2008). These specifications are derived from the W3C specifications for XML encryption ("XML Encryption Syntax and Processing", 2002) and signature ("XML-Signature Syntax and Processing", 2008). The generic specifications are to be made compatible with ECC based algorithms.

A digital signature is used for representing the document's identity. Any change in the document alters the message digest. Hence it is necessary to have a uniform standard for the messages and their digests that are being signed as there can be similar YAML messages, varying only in white spaces or line breaks. Also, there are possibilities of serializing the elements within a node in different orders. Even though reordering preserves the semantics, the signature derived will be distinct for these semantically equivalent messages. These effects are to be nullified in order to ensure that the message digest being created is not affected by the textual variations. This processing is popularly known as canonicalization, and is effectively implemented for XML files. W3C published the specifications for two types of canonicalization algorithms: Inclusive XML Canonicalization (Canonical XML, 2001) and Exclusive XML Canonicalization (Exclusive XML Canonicalization, 2002). The inclusive canonicalization process facilitates a case to case conversion of elements and seems to be the simple compared to the exclusive canonicalization process.

From the literature survey, it can be concluded that most of the existing message formatting schemes including XML are verbose in nature and are not suitable for efficient use in RCNs. Hence, there is a need for developing a lightweight and compact messaging format for RCNs. It is observed that YAML is the most promising existing data format which can serve as a base format for developing the lightweight compact message format.

It is learnt that message level security is better than transport level security for messaging. To implement message level security for the new format, specifications must be developed for encryption, signature and Signcryption of the messages. A canonicalization process also must be developed to minimize the effect of textual variations on the message digest. However it is observed that provision of security to messages is a heavyweight process. So it is necessary to minimize the security overheads without compromising the security. ECC allows phenomenal

reduction in key sizes for the same level of security and hence there is a need for developing specifications for ECC compatiblity with the new format.

THE PROPOSED METHOD

In order to reduce the consumption of resources such as memory, bandwidth and battery power, we mainly concentrate on reducing the verbosity of messages. Care is taken to see that reducing the verbosity is achieved without compromising the proved features of XML such as self descriptive nature, readability, ease of preparation, schema awareness and security.

Our work uses YAML as the base format, a data serialization language devised by Brian Ingerson, Clark Evans, and Oren Ben-Kiki ("YAML", 2005). It has a human-readable format that takes features from languages such as XML, C, Python, Perl, as well as the format for electronic mail as specified by RFC 2822 ("YAML", 2005). YAML is optimized for data serialization, configuration settings, log files, Internet messaging and filtering ("YAML", 2005).

The advantages of YAML are summarized as follows: (i) easily readable and editable by any standard word processor, (ii) interacts well with scripting languages, (iii) uses host languages' native data structures, (iv) has a consistent information model, (v) enables stream-based processing, (vi) as expressive and extensible as XML, and (vii) its serialization reduces the verbosity of the message considerably, that too without affecting its readability ("YAML", 2005).

Our proposal consists of two parts: (i) lightweight format, and (ii) security specifications to achieve message level security. The proposed format based on YAML is designated as Thinned YAML (TYAML).

The TYAML Format

The steps in developing TYAML are (i) addition of schema awareness, (ii) verbosity reduction Phase I, and (iii) verbosity reduction Phase II.

Addition of Schema Awareness to TYAML: - Removing clutter from contents is an important step in verbosity reduction. This helps in conserving memory, when there are a group of messages with a common structure to be transmitted. This process results in the creation of a schema definition for TYAML – we designate it as YASchema.

There are three kinds of nodes in YAML - scalar, sequence, and mapping. Sequence means an ordered series of entries (Ben-Kiki, Evans, & Dot, 2008); mapping means an unordered association of unique keys to values (Ben-Kiki et al., 2008); and scalar means any datum with opaque structure presentable as a series of Unicode characters (Ben-Kiki et al., 2008). Combining these primitives generate a directed graph structure. Apart from this, there can be mappings of sequences and sequences of mappings.

The entire document is scanned one node at a time. The message structure is to be evaluated to decide about the type of schema to be constructed. The type can be of any of the three kinds - scalar, sequence and mapping. Mappings of sequences and sequences of mappings are considered as special cases. Primitive details like name of the element, its data type and an ID value are added to the schema for every member in a node. Once this process is completed, we get the YASchema description for the message to be transferred.

Advantages of YASchema: - YASchema offers the following improvements (i) squeezing techniques are applied to define schema definitions in flow styles, (ii) elements of schema and their properties are designed in such a way that they require only minimum bytes (this is done without hampering the readability), (iii) facility to define ID codes for the elements of YAML are given (this arrangement significantly reduces the size of the original message), and (iv) the process of

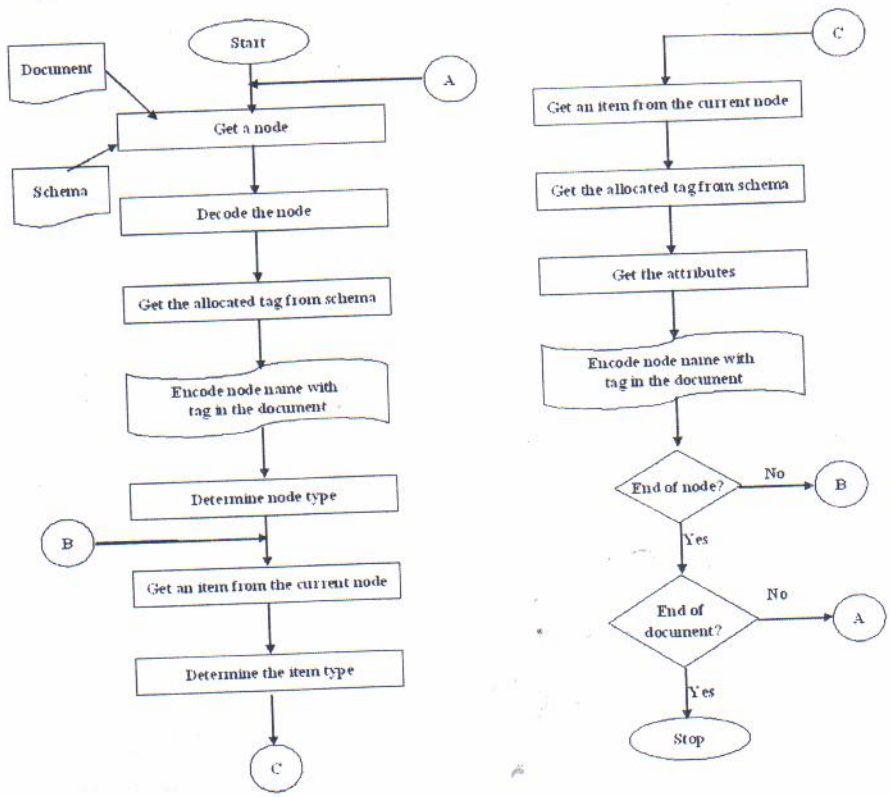
reusing the same schema across multiple sessions are envisioned (a special directive is added in the YAML file to specify the schema definition to be referred).

Verbosity Reduction Phase I: - In this process, the built-in features of plain YAML is used in cutting the verbosity of the message. It can be seen that YAML's natural approach of representing information provides high degree of readability but also boosts the memory requirements needed for representing the information. A method of representation known as "Flow style" method in YAML (Ben-Kiki, Evans, & Dot, 2008) helps one to squeeze the size considerably. There are no significant gains in converting scalars using this method, as the effort made in this regard may not produce good results. In certain cases, reorganization of the contents to this style yields better performance. For e.g., a complex message consisting of a customer invoice containing many product details can be subjected to this stage for better conservation of message size. Reorganization of elements into flow style is done in this stage.

Verbosity Reduction Phase II: - This stage uses YASchema for reducing the message size. All primitive details regarding each and every element in the message are defined in the schema, in the schema creation stage itself. In addition to this, special ID codes are provided for each element.

The message is scanned and the elements are encoded with the ID codes. The ID codes could be referred from the accompanying schema, or if the sender likes to reuse a schema that had been sent earlier, the schema to be referred could be specified in the message itself. This makes the decoding easy, making the message schema aware and is a useful value addition to YAML. The process is as shown in Figure 1.

Figure 1. Verbosity Reduction Phase II



Incorporating Message Level Security

In addition to optimizing the message size, secure transfer of messages is a must. Exchanging the information without disclosing its contents is vital in the present-day networked world. Apart from ensuring the confidentiality of messages, authentication, data integrity and support for non-repudiation also must be ensured. Incorporating these mechanisms makes TYAML safe and secure.

There is a need for a uniform specification to describe and exchange the parameters and associated information. The security specification defines the formulation of document structure and the associated schema for a particular purpose. This is essential to maintain uniformity among the different platforms and application program interfaces.

Security protocols and algorithms play a limiting or negative role in process efficiency and energy efficiency. So, restricting the number of packets transmitted or received and using smaller keys are important to overcome the above limitations. Analysis on the energy consumption for the processing of various cryptographic algorithms underlines the need to have messages with reduced size and small key sizes (R. Potlapally Ravi & Niraj, 2006).

Accordingly in this chapter, the generic specifications in (Haleem & Sebastian, 2008) are refined to include ECC based algorithms (ECIES, ECDSA, and EC Signcryption), to make TYAML compatible with ECC based algorithms.

Canonicalization Process: - All the causes of variations that occur for XML messages will not be applicable to the sliced form of YAML, as there are subtle differences in the methodology adopted for the representation of information. YAML specification (Ben-Kiki et al., 2008) gives an insight into the need for canonicalization. Also, the methods to compare the equivalences of two YAML structures are discussed in the specification. YAML supports the need for scalar equality by insisting that every scalar tag must specify a mechanism to produce the canonical form of any formatted content.

The inclusive canonicalization process is formulated for TYAML. The canonicalization process ensures that conversions are done to make a unified format for messages to be signed. We have taken a subset of the canonicalization process (relevant to YAML) suggested by W3C (Canonical XML, 2001) for our purpose, as follows: (i) the document is represented in UTF-8 encoded format, (ii) line breaks are normalized to "#xA", (iii) default attributes (specified in the YASchema) are added to each element if no values are provided, (iv) white spaces are normalized, (v) name space declarations except superfluous in nature are preserved, (vi) attributes are included in ascending lexicographic order if the order of occurrence of the attributes are not specified in the accompanying YASchema, and (vi) facilities to include or exclude comments after canonicalization.

Specification for ECIES: - The skeleton structure of the specifications for ECIES is as shown in Figure 2. *EncrData* is the root element. The encrypted part of the document is initiated with the *EncrData* mapping. It is the core element where all encrypted data are enclosed. Elements specific to ECIES are *KDF*, *MAC* and *SharedInfo*. These elements house the key derivation function, message authentication code function and the shared info called *S1* and *S2*, respectively. Also, the *CipherValue* element encompasses the results v (the public key), c (the cipher key) and t (the message tag).

Specification for ECDSA: - The generic specification for digital signature is modified to include the elements needed for ECDSA operations. The minimal structure is as shown in Figure 3. The specification includes provisions to maintain information such as signature algorithm, digest method, signature value and key information.

Specification for EC Signcryption: - The skeleton structure of the specification and YASchema

for EC Signcryption is as shown in Figure 4. The specification has provision to include all the parameters needed to perform EC Signcryption - Signature Algorithm, Encryption Method, One-way Hash, Keyed Hash, Key Information and Signcryption Values.

PERFORMANCE EVALUATION

Messages differing in size and complexity are normally exchanged between the users. We classify the messages into five types: *Short* type represents the simple messaging format with only text, *Small* category consists of a single record with String, Float and DateTime types of data, *Medium* category contains the details of 25 customer records, and *Large* and *Composite* categories are constituted as invoice records with varying degrees of complexity. These categories are designed in such a way to provide consistent test data scheme.

YAML, XML and TYAML versions are serialized for each category of messages for performance evaluation.

VERBOSITY

Verbosity: TYAML vs. Other Non Binary Formats: - Verbosity of TYAML data sets are compared against ONX, JSON and SOX formats (Figures 5, 6 and 7, respectively). These three formats are chosen as representatives of the alternative non-binary serializations.

The observations from Figures 5, 6 and 7 are: (i) for *Small* category, all formats exhibit more or less similar performance level; the SOX format has a slight advantage over TYAML (ii) when the complexity of the data set increases, TYAML

Figure 2. (a) Specifications for ECIES and (b) Corresponding YASchema

```
EncrData:
  Type:
  EncrMethod:
    Algorithm:
  KDF:
    Algorithm:
  MAC:
    Algorithm:
  SharedInfo:
    S1:
    S2:
  KeyInfo:
    n:
    g:
      x:
      y:
    a:
    b:
    p:
    h:
  CipherData:
    CipherValue:
      vValue:
      cValue:
      tValue:
```

(a)

```
tp:B2
  EncrData:(tg:Ed, tp:B2, rd:T, nsl:, ns:)
   Type:(tg:_Tp_, tp:1, rd:T)
   EncrMethod:(tg:_Em_, tp:B2, rd:T)
     Algorithm:(tg:_Al_, tp:1, rd:T)
     KDF:(tg:, tp:B2, rd:T)
       Algorithm: (tg:_Al_, tp:1, rd:T)
     MAC:(tg:, tp:B2, rd:T)
       Algorithm: (tg:, tp:1, rd:T)
     SharedInfo:(tg:_Shi_, tp:B2, rd:T)
       S1: (tg:, tp:1, rd:T)
       S2: (tg:, tp:1, rd:T)
     KeyInfo:(tg:_Ki_, tp:B2, rd:T)
       n: (tg:, tp:1, rd:T)
       g: (tg:, tp:B2, rd:T)
         x: (tg:, tp:1, rd:T)
         y: (tg:, tp:1, rd:T)
       a: (tg:, tp:1, rd:T)
       b: (tg:, tp:1, rd:T)
       p: (tg:, tp:1, rd:T)
       h: (tg:, tp:1, rd:T)
     CipherData:(tg:_Cd_, tp:B2, rd:T)
       CipherValue:(tg:_CV_, tp:B2, rd:T)
         vValue: (tg:_vv_, tp:1, rd:T)
         cValue: (tg:_cv_, tp:1, rd:T)
         rValue: (tg:_rv_, tp:1, rd:T)
```

(b)

Figure 3. (a) Specifications for ECDSA and (b) Corresponding YASchema

```
Signature:
  SignedInfo:
    SignatureMethod:
      Algorithm :
    Reference:
      URI:
      DigestMethod:
        Algorithm:
      DigestValue:
  SignatureValue:
    rValue:
    sValue:
  KeyInfo:
    ECDSAKeyValue:
      n:
      g:
        x:
        y:
      a:
      b:
      p:
      h:
```

(a)

```
tp:B2
Signature:(tg:Sg, tp:B2, rd:T, nsl:, ns:)
  SignedInfo:(tg:_Si_, tp:B2, rd:T)
    SignatureMethod:(tg:_Sm_, tp:B2, rd:T)
      Algorithm : (tg:_Al_, tp:1, rd:T)
    Reference:(tg:_Rf_, tp:B2, rd:T)
      URI: (tg:_Ur_, tp:1, rd:T)
      Transforms:(tg:_Tr_, tp:B2, rd:F)
        TransAlg: (tg:_Ta_, tp:1, rd:F)
      DigestMethod:(tg:_Dm_, tp:B2, rd:F)
        DigAlg: (tg:_Da_, tp:1, rd:T)
      DigVal: (tg:_Dv_, tp:1, rd:T)
  SignatureValue:(tg:_Sl_, tp:B2, rd:T)
    rValue: (tg:_rV_, tp:1, rd:T)
    sValue: (tg:_sV_, tp:1, rd:T)
  KeyInfo:(tg:_Ki_, tp:B2, rd:T)
    ECDSAKeyValue: (tg:_dsaKv_, tp:B1, rd:T)
      n: (tg:, tp:1, rd:T)
      g: (tg:, tp:B2, rd:T)
        x: (tg:, tp:1, rd:T)
        y: (tg:, tp:1, rd:T)
      a: (tg:, tp:1, rd:T)
      b: (tg:, tp:1, rd:T)
      p: (tg:, tp:1, rd:T)
      h: (tg:, tp:1, rd:T)
```

(b)

Figure 4. (a) Specifications for EC Signcryption (b) Corresponding YASchema

```
ECSigncryption:
  Type:
  SignedInfo:
    SignatureMethod:
      SigAlgorithm:
    EncryptionMethod:
      EncAlgorithm:
    Reference:
      URI: ""
      OnewayHash: One way hash function
      KeyedHash: Keyed Hash Function
    KeyInfo:
      n:
      g:
        x:
        y:
      a:
      b:
      p:
      h:
  SigncryptionValue:
    CValue:
    RValue:
    SValue:
```

(a)

```
tp:B2
ECSigncryption:(tg:_EcYs_, tp:B2, rd:T)
  Type:(tg:_tp_, tp:1, rd:T)
  SignedInfo:(tg:_Si_, tp:B2, rd:T)
    SignatureMethod:(tg:_Sm_, tp:1, rd:T)
      SigAlgorithm:(tg:_SAg_, tp:1, rd:T)
    EncryptionMethod:(tg:_Em_, tp:1, rd:T)
      EncAlgorithm:(tg:_EAg_, tp:1, rd:T)
    Reference:(tg:_Rf_, tp:B2, rd:T)
      URI:(tg:_Ui_, tp:1, rd:T)
      OnewayHash:(tg:_Oh_, tp:1, rd:T)
      KeyedHash:(tg:_Kh_, tp:1, rd:T)
      KeyInfo:(tg:_Kf_, tp:B2, rd:T)
        n:(tg:, tp:1, rd:T)
        g:(tg:, tp:1, rd:T)
          x:(tg:, tp:1, rd:T)
          y:(tg:, tp:1, rd:T)
        a:(tg:, tp:1, rd:T)
        b:(tg:, tp:1, rd:T)
        p:(tg:, tp:1, rd:T)
        h:(tg:, tp:1, rd:T)
    SigncryptionValue:(tg:_Sv_, tp:B2, rd:T)
      CValue:(tg:_C_, tp:1, rd:T)
      RValue:(tg:_R_, tp:1, rd:T)
      SValue:(tg:_S_, tp:1, rd:T)
```

(b)

needs only less number of bytes than the other serializations. Hence TYAML has maximum advantage in the *composite* category.

Verbosity: TYAML vs. XML, YAML (XML in WFF):- The verbosity of TYAML is compared against plain YAML and XML. For this compari-

Figure 5. Verbosity: TYAML vs. Other Non Binary Formats (short category)

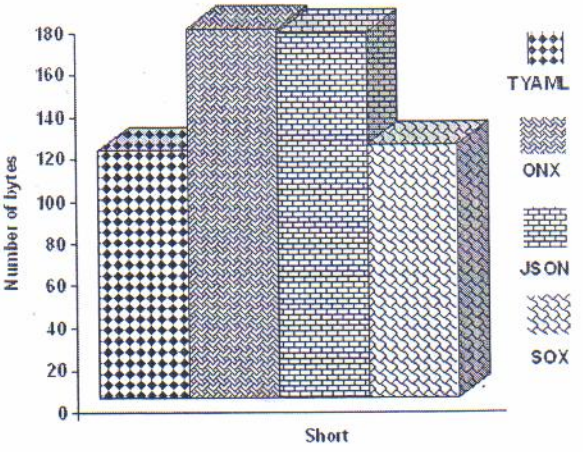


Figure 6. Verbosity: TYAML vs. Other Non Binary Formats (medium category)

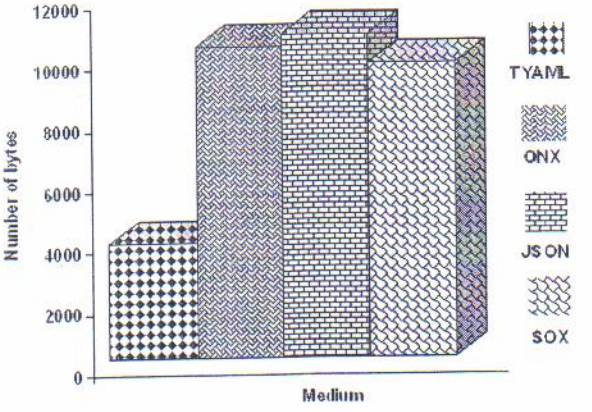
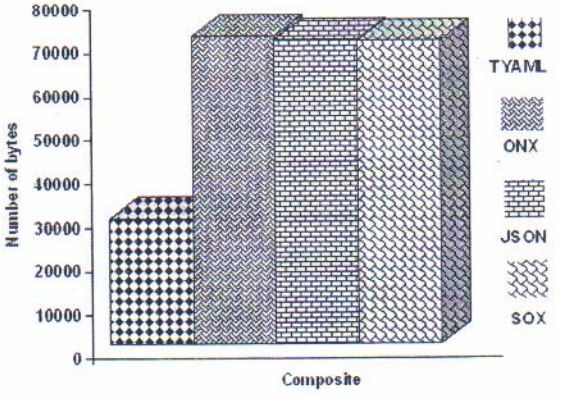


Figure 7. Verbosity: TYAML vs. Other Non Binary Formats (composite category)



son XML is serialized in the well formed form (WFF), as WFF is accepted universally by all the browsers. It can be observed from Figure 8 that TYAML is performing better than YAML and XML in all categories. The advantage of XML is more visible in the *medium*, *large* and *composite* categories.

Verbosity: TYAML vs. YAML (both without schema):- YAML and TYAML are compared for verbosity in different categories. It can be observed from Figure 9 that TYAML has a marginal advantage in *short* and *small* categories than YAML. In the other categories, TYAML has an advantage over YAML. This demonstrates the effectiveness of the thinning measures for the verbosity reduction.

Packets Needed: TYAML vs. XML:- Maximum Segment Size (MSS) in TCP/IP on an Ethernet network is taken as the reference for the calculation of the number of packets (1460 bytes). The number of packets needed for the transmission of data is crucial in the constrained wireless mobile environment because of the limitations such as packet loss, retransmission, energy consumption, etc. It can be observed from Figure 10 that TYAML data sets require minimum number of packets, especially in the larger categories (*large* and *composite*).

SECURITY SPECIFICATIONS

The performance of TYAML specifications are compared with that of XML. A key size of 283 bits is used for the EC algorithms. The parameters and other details of the process and processed values are represented as per the specifications in TYAML and XML, for comparison.

Figure 8. Verbosity: TYAML vs. XML (in WFF) and YAML

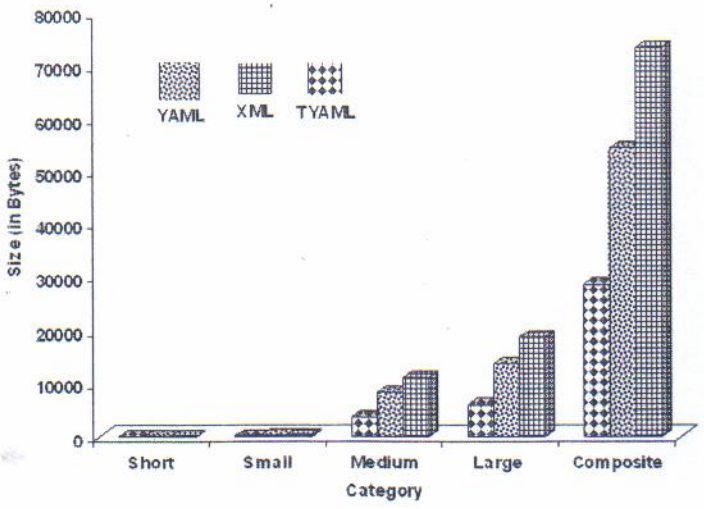


Figure 9. Verbosity: TYAML vs. YAML (both without schema)

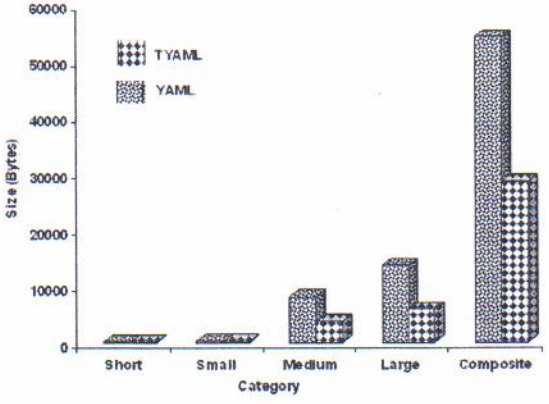
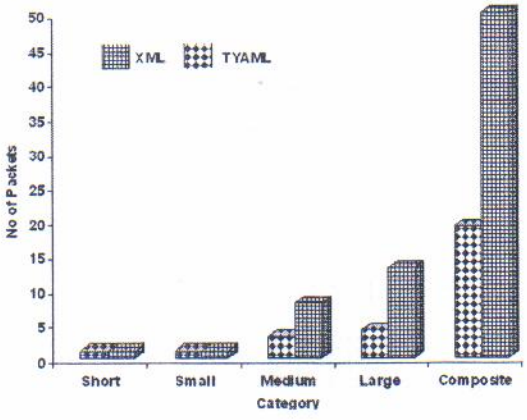


Figure 10. Packets Needed: TYAML vs. XML



Verbosity for ECIES Specification: TYAML vs. XML: - The verbosity of the XML specification and XML Schema for ECIES are compared with the corresponding TYAML specification and YASchema in Figure 11. It can be seen that TYAML/YASchema combination is performing better than the XML/XML Schema combination.

Verbosity for ECDSA Specification: TYAML vs. XML: - The verbosity of XML specification and XML Schema for ECDSA are compared with the corresponding TYAML specification and YASchema as shown in Figure 12. In this case also, TYAML/YASchema combination is performing better than the XML/XML Schema combination.

Verbosity for EC Signcryption Specification: TYAML vs. XML: - The specifications (XML and TYAML) and corresponding schemas (XML Schema and YASchema), after applying the EC version of Signcryption, are compared for verbosity in Figure 13. It can be seen that the TYAML/YASchema combination is performing better than the XML/XML Schema combination.

Figure 11. Verbosity for ECIES Specification: TYAML vs. XML

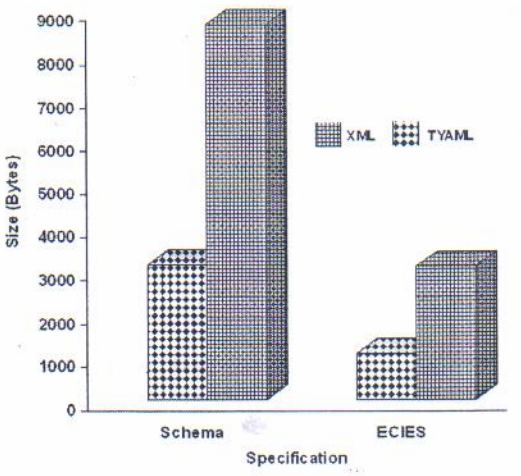
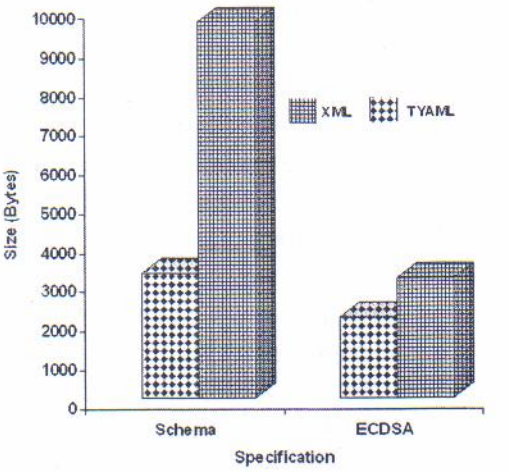


Figure 12. Verbosity for ECDSA Specification: TYAML vs. XML



The results of the verbosity comparisons for EC based specifications between XML and TYAML are as shown in Figure 14. The TYAML format reports gain in all categories.

CONCLUSION

The challenges posed by the networks and devices due to the increased penetration of wireless mobile devices calls for the need to have a messaging standard suitable for constrained wireless mobile environments. This chapter proposed a solution to overcome the limitations imposed by latency, bandwidth requirement and low battery backup in wireless networks.

XML, in its new incarnation as a messaging protocol, is the default format for the purpose. But the inherent limitations of the constrained wireless mobile environment make XML unsuitable in this environment. The suggested method is based on YAML, a user friendly lightweight

Figure 13. Verbosity for EC Signcryption Specification: TYAML vs. XML

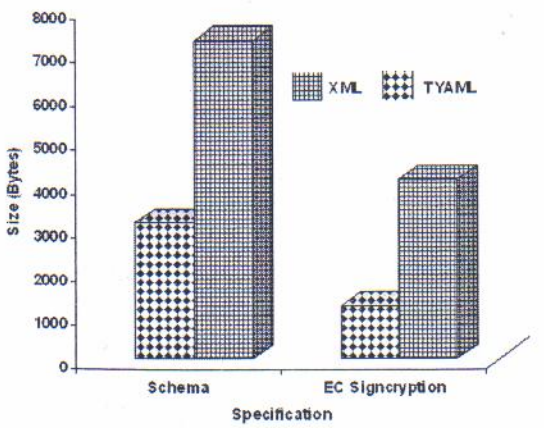
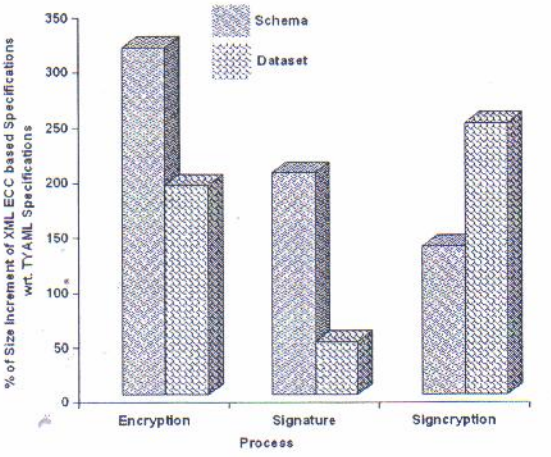


Figure 14. Verbosity of EC based Specifications: TYAML vs. XML



messaging format. It is observed that message size is reduced using YAML in its natural form itself. This is further enhanced by verbosity reduction measures. To optimize the results, we have proposed a schema definition also. Apart from the verbosity reduction, proposals for fast and energy conserving security processing are also included. Improvements are observed in the case of message size, packets needed, and in terms of security specifications.

Several enhancements are needed to furnish this as a messaging standard. 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). These are topics suggested for further research. The improvement over the message size can save battery life, transmission time and energy consumption.

REFERENCES

Ben-Kiki, O., Evans, C., & Dot, I. (2008). *YAML 1.2 Specification*, (Available at http://yaml.org/spec/1.2/).

Cai, Ghandeharizadeh, Girardot, Schmidt, & Song. (2002, September). *A Comparison of Alternative Encoding Mechanisms for Web Services*. In 13th International Conference on Database and Expert Systems Applications, Aix en Provence, France.

Canonical, X. M. L. (2001, March). Retrieved from http://www.w3.org/TR/xml-c14n

Certicom-Research. (2000). *Standards for Efficient Cryptography, sec 1: Elliptic Curve Cryptography (Tech. Rep.)*. Foster City, CA: Certicom Research.

Chou, W. (2003). *Elliptic Curve Cryptography and Its Applications to Mobile Devices (Tech. Rep.)*. College Park, MD: University of Maryland.

Clark, K. G. (2002, July). *Look Ma, No Tags*. Retrieved from http://www.xml.com/pub/a/2002/07/24/yaml.html.

Crockford, D. (2006, July). *The application/json Media Type for JavaScript Object Notation (JSON)*. Retrieved from http://tools.ietf.org/html/rfc4627

Devaram, & Andresen. (2003). *SOAP Optimization via Parameterized Client-Side Caching*. Retrieved from http://java.sun.com/ developer/technicalArticles/ xml/fastinfoset/

Edoh, K. D. (2004). *Elliptic Curve Cryptography: Java Implementation*. In INFOSECCD '04: Proceedings of the 1st Annual Conference On Information Security Curriculum Development (pp. 88–93). New York, NY: ACM.

Exclusive, X. M. L. Canonicalization. (2002, July). Retrieved from http://www.w3.org/ TR/xml-exc-c14n/).

Girardot, & Sundaresan. (2000). *Millau: An Encoding Format For Efficient Representation And Exchange Of XML Over The Web*. In Proceedings of the 9th International World Wide Web Conference On Computer Networks (p. 747-765). Amsterdam: North-Holland Publishing Co. Retrieved from citeseer.ist.psu.edu/article / widener01open.html

Gokran, P. (2007, November). *Making your Low-Cost Handsets Plans Succeed*. In Telecom Era (pp. 23–32).

Haleem, P. P. A., & Sebastian, M. P. (2008). An Alternative Approach For Slicing Down The Message Size And Enhancing The Security In Wireless Mobile Network. *The Mediterranean Journal of Computers and Networks*, 5, 148–149.

Han, J.-H., Kim, Y.-J., Jun, S.-I., Chung, K.-I., & Seo, C.-H. (2002). *Implementation Of ECC/ ECDSA Cryptography Algorithms Based On Java Card.* In ICDCSW '02: Proceedings of the 22nd International Conference On Distributed Computing Systems (pp. 272–278). Washington, DC: IEEE Computer Society.

Han, Y., Yang, X., Wei, P., Wang, Y., & Hu, Y. (2006). ECGSC. *Elliptic Curve Based Generalized Signcryption, LNCS4159*, 956–965.

Johnson, & Menezes. (1999, August). *The Elliptic Curve Digital Signature Algorithm (ECDSA) (Tech. Rep.).* Technical report CORR 99-34, Department of C&O, University of Waterloo, 1999.

Kangasharju, J. (2005). *Mobile XML Messaging (Tech. Rep.). Department of Computer Science.* Finland: University of Helsinki.

Kangasharju, J., Lindholm, T., & Tarkoma, S. (2008). XML Security with Binary XML for Mobile Web Services. *International Journal of Web Services Research, 5*, 1–19. doi:10.4018/ jwsr.2008070101

Lai, Y.-X., Chen, Y.-L., & Chen, H. (2008, November). PEJA: Progressive Energy-Efficient Join Processing For Sensor Networks. *Journal of Computer Science and Technology, 23*(6), 957–972. doi:10.1007/s11390-008-9191-2

Michael, M. P. (2005). *Research Seminar On Energy Awareness, University Of Helsinki.* Energy Awareness for Mobile Devices.

Open Node Syntax. (2005, June). Retrieved from http://www.seairth.com/ web/onx/onx.html

Park, N., Kim, H., Chung, K., Sohn, S., & Won, D. (2006). *Xml-Signcryption Based LBS Security Protocol Acceleration Methods In Mobile Distributed Computing.* In ICCSA 2006: International Conference On Computational Science And Its Applications (LNCS 3984, pp. 251–259). Glasgow, UK: Springer-Verlag

Peng, C., & Li, X. (2005, September). *Threshold Signcryption Scheme Based On Elliptic Curve Cryptosystem And Verifiable Secret Sharing.* In 2005 International Conference On Wireless Communications, Networking And Mobile Computing (Vol. 2, pp. 1182–1185).

Potlapally, N., Ravi, S., Raghunathan, A., & Jha, N. (2003, August). *Analyzing The Energy Consumption Of Security Protocols.* In Proceedings Of The 2003 International Symposium Of Low Power Electronics And Design, Seoul, Korea.

Sandoz, P., & Pericas-Geertsen, S. (2005). *Fast Infoset@Java.net.* In Xtech 2005: XML, The Web And BeyondRetrieved from http://idealliance.org/ proceedings/xtech05/ papers/04-01-01/

Schmelzer, R. (2004, November). *Will Binary XML Solve XML Performance Woes?* Retrieved from http://searchwebservices.techtarget.com / tip1,289483, sid,26 gci1027726,00.html

Simple Outline, X. M. L. SOX. (2002). Retrieved from http://www.langdale.com.au/ SOX/

Soap-tutorial. (2007). Retrieved from http://www. w3schools.com/ soap/default.asp

Tamine-Lechani, L., Boughanem, M., & Daoud, M. (2009, July). *Evaluation Of Contextual Information Retrieval Effectiveness: Overview Of Issues And Research.* Knowledge and Information Systems.

WAP Binary XML Content Format. (1999, June). Retrieved from http://www.w3.org/ TR/wbxml/

Xebu Serialization Format. (2007, September). Retrieved from https://hoslab.cs.helsinki.fi/ homepages/xebu/

XML Encryption Syntax and Processing. (2002, December). Retrieved from http://www.w3.org/ TR/xmlenc-core/

XML-Signature Syntax and Processing. (2008, June). Retrieved from http://www.w3.org/ TR/ xmldsig-core/

YAML. (2005)Retrieved from http://yaml.org/

Yaml. (2007, November). Retrieved from http:// en.wikipedia.org/ wiki/YAML)

Zhang, F., Mu, Y., & Susilo, W. (2005). *Reducing Security Overhead For Mobile Networks*. In AINA '05: Proceedings Of The 19th International Conference On Advanced Information Networking And Applications (pp. 398–403). Washington, DC: IEEE Computer Society.

Zheng, Y. (1997a). *Digital Signcryption or How to Achieve Cost (Signature & Encryption) << Cost(Signature) + Cost(Encryption)*. LNCS 1294, pp.165–179). Retrieved from citeseer.ist.psu.edu/ zheng97digital.html

Zheng, Y. (1997b). *Signcryption And Its Applications In Efficient Public Key Solutions*. In Proc. 1st International Information Security Workshop (pp. 291–312). Retrieved from citeseer.ist.psu.edu/ zheng97signcryption.html