Abstract

The XML is rapidly becoming one of the most important data formats of the electronic data exchange, process intercommunication in ways not possible before. An example of such data are the electronic business documents round via Internet between partners of B2B business model. There is no doubt that the security of exchanged data is one of the most relevant task to be done. In this article the authors will present some of XML security aspects, point to electronic business data exchange.

XML technology

XML (eXtensible Markup Language) is relatively young, but very fast and still developing technology. The first time XML was presented at the SGML 96 Conference in Boston (November 1996). The 1st XML Conference was held by
the Graphic Communications Association (San Diego, March 97). In October 97 the 14C came with a note on “14C Data Formats on XML, SGML, HTML, and RDF”[2]. Until now the XML specification is backed by SoftQuad, Adobe, IBM, HP, Microsoft, Netscape, Lockheed Martin, NCSA, Novell, Sun, Boston University, Oxford University, and the Universities of Illinois and Waterloo [4].

The XML is a meta-markup language that provides a format for describing structured data. XML has much in common with data description languages, such as DDS and SQL, but there is a major conceptual difference. An XML document is completely self-contained and includes both data and data description. Other languages only describe data (field names, data types, and sometimes-valid field values). The data itself are sorted in relational database independent of the data description. With XML, a programmer should be able to extract the meaning of the data from a document without additional information.

On the surface, XML looks like HTML (Hyper Text Markup Language). Both are derived from the Standard Generalised Markup Language (SGML)[5]. The tools that generate HTML can often be reused to generate XML[6,7,8]. Both HTML and XML use marks <, > and & to create element and attribute structures. The main difference is that XML hasn’t got a limitation of pre-set library of tags. When the document is marked up in XML, it is possible to choose the tag name that is the best way describes the contents of the element. This facilitates more precise declarations of content and more meaningful search results across multiple platforms. It is flexible enough to handle an incredibly wide variety of information, and also allows for such information to be self-describing, so that it may be manipulated by software that has not been previously exposed to a description of the underlying meaning behind the data (see fig. 1). With its powerful expressiveness and flexibility, the XML promises to give the EDI’s functionality without excessive costs increase [9, 10].
Looking at so many advantages of the XML technology, more and more companies decides to use it to transmit their business (read strategic) data across the Web, and this way the security of documents becomes increasingly relevant.

When you need to make a secure way of data exchanging via Internet, you ought to put a point to four things [1]:

- confidentiality, means, no one else can access or copy the data,
- integrity, the data isn’t altered as it goes from the sender to the receiver,
- authentication, the document actually came from the purported sender,
- non-repudiability, the sender of the data cannot deny that they sent it, and they cannot deny the contents of the data.

In most of the authors known e-business solutions, including their own one applied in some of polish companies [11, 12], the security of data exchanging is implemented with using the Secure Sockets Layer (SSL) protocol. This protocol provides the first two functions and a part of third, but it is not enough for secure the four point.
Secure “pipe” with SSL

Secure Sockets Layer was introduced by Netscape in March 1995 in the Netscape 1.1 browser as the method for encrypting data between web browsers and web servers [2]. SSL creates a secure channel to pass some sensitive data (e.g. credit card details) through Internet. SSL uses digital certificates to provide the authentication of web servers and browsers. Digital certificates store the public key of a person or entity, and bind this key to information about that person, and to the issuer of the certificate. A public key is used to encrypt data and the corresponding private key can decrypt this data. SSL uses the asymmetrical encryption algorithm RSA (the RSA algorithm is named after Ron Rivest, Adi Shamir and Len Adleman, who invented it in 1977 [14]) and creates a secure “pipe” between the browser and the web server, and until now it became the most popular standard for establishing the secure Web based data exchange of all data formats, including the XML data too.

Although SSL is the unquestionably leader of web communication secure, but in nova days it is not enough. It is difficult to prove that an SSL session occurred, more over, to prove that a business document (e.g. an order) has been received using SSL at all [2]. This way is possible to imagine a situation when Partner A executes some big electronic order, produces some goods, sends them and pays for transport, and then it turned out that Partner B doesn’t know anything about this order. Of course it is rather impossible, but it could be a simple human mistake, but who pay for it? In conventional kind of ordering way (the paper one) there is always a person on the other side, a document is signed, and we may call somebody to account. With SSL our employee could not prove his innocence, that he has an order as it is, he do not modifies it or create it himself. In other words as important as protecting the confidentiality of business communications is ensuring their long-term authenticity (who is the sender?), data integrity (have the data been
modified in transit?), and support for non-repudiation (can the sender deny sending them?);

Fig. 2. An example of secured channel implemented with SSL, a business document located in FTP directory is automatically downloaded by a partner’s system

One other thing it is necessary to consider when implementing SSL is this: All the encryption and decryption is performed internally, and SSL only decrypts a document sent as part of a current session. This means that you cannot cache SSL-encrypted documents. If some sensitive information has been received over an SSL connection and it is necessary to store it securely, it has to be explicitly saved as decrypted file on a local machine. The data received thought confidentiality SSL channel is all too frequently left unprotected on the server. If you were an aggressor much more attractive for you is to break into a back-end database containing thousands of credit card numbers, than sniff IP packets in transit in order to obtain a single user’s credit card number.

The globally-recognized method for satisfying the requirements for secure business transactions is to use digital certificates to enable the encryption and digital signing of the exchanged data. In that point the XML, in opposite to other technologies,
gives some mechanism to take control of authenticity, integrity and non-repudiability; there are XML Signature [15] and XML Encryption [16], which allows to provide the security in electronic data exchange.

**XML Signatures**

The XML Signature specification is proposed by the World Wide Web Consortium (14C) and Internet Engineering Task Force (IETF). They introduce a `<signature>` element which contains all the information needed to process a digital signature. Relaying on invention of public-key cryptography, XML signatures add authentication, data integrity, and support for non-repudiation to the data that they sign.

To create a digital signature for an electronic business file or a part of it, the data to be signed is transformed by an algorithm with the private key of the sender uses. That transformation determined by the sender's private key can only be untransformed if sender's public key will be used, this way a recipient of the transformed data can be confident of the origin of the data. This role is very useful when an author of signed information is going to deny the fact that he send it. With SSL the layer of security is very often transparent for a sender and a recipient. This way the recipient has got an information received from someone, but he cannot proof the sender that he is the author. This information could be send as well by him as well could be an unfair play of the recipient. This ability of XML technology names nonrepudiability and this is a difference from the SSL solutions. XML Signature is very scaleable and it is able to be customized to different solutions – as well the all contents of a file could be signed as well the file could be split into logical parts and each part could be signed with another sender's private key (by another person). Each digital signature could be referred to an XML element contained inside the `<signature>` element, or an external XML document being references by a URI. It is also possible to sign an external non-XML resource by a URI reference using.
The signed version of the document presented in Fig. 1 is shown in Fig. 3, there are predefined tags inside the `<signature>` element.

Fig. 3. An example of a business document signed with XML signature.

The XML business document processed by XML Signature is still opened for unauthorized access and the secret data may be read by third person (an
aggressor). To save the data before an unauthorized access there is implemented the XML Encryption specification.

**XML Encryption**

The XML Encryption specification is proposed by the World Wide Web Consortium (14C) and Internet Engineering Task Force (IETF). They introduce a `<EncryptedElement>` element which contains encrypted part of data and information about used algorithm of encryption. Relaying on invention of public-key cryptography or with owns algorithm using, XML encryption is a powerful tool for secure data interchange. An example of business document encrypted with XML encryption is sown at fig. 4.

![Diagram of XML Encryption](image)

**Fig. 4.** An example of business document encrypted with XML encryption

The electronic interchange of business data with XML encryption technology allows to secure documents whole the time. Not only during the transportation through the Web like it is implemented in SSL. The data are always secured in the computing system of sender, and they are save in the computing system of recipient, too. In SSL the data is encrypted only during a transport, and they are opened on attack after the transport. In XML the data is encrypted until it will be undone by some authorized person.
Conclusion

As XML becomes a vital component of the emerging electronic business infrastructure, we need trustable, secure XML messages to form the basis of business transactions. One key to enabling secure transactions is the concept of a digital signature, ensuring the integrity and authenticity of origin for business documents. XML Signature is an evolving standard for digital signatures that both addresses the special issues and requirements that XML presents for signing operations and uses XML syntax for capturing the result, simplifying its integration into XML applications. But what the XML technology brings new over the existing secure data exchange solutions, e.g. SSL? In the secure data sending across the Web, there are four things to be provide:

- confidentiality, no one else can access or copy the data,
- integrity, the data isn't altered as it goes from the sender to the receiver,
- authentication, the document actually came from the purported sender,
- nonrepudiability, the sender of the data cannot deny that they sent it, and they cannot deny the contents of the data.

SSL provides the first three functions; the XML technology provides this fourth.

Bibliography


http://www.vordel.com/news/articles/02-03-08.html


