ArticSoft

S/MIME – great idea, pity about the implementation

Summary

If correctly implemented, the S/MIME standard seems an attractive proposition for providing simple signature and encryption 'envelope' functions for e-mail and the attachments going with it. However, despite the interoperability challenges of EEMA and others over the last four years it remains a challenge to get one e-mail provider working successfully with another.

Because S/MIME was developed to provide an 'envelope' around the mail, its content protection stops once the mail has been unpacked. Protection is not bonded into the text and the files, something that is essential for later audit verification or when text and files must be sent to multiple recipients and their agreement captured.

Alternative methods that focus on information as objects have significantly more functionality to offer. Low cost practical implementations that make existing technologies easy to implement are needed before more confusing standardization is carried out.

Introduction

S/MIME, the secure version of MIME, started off around 1995, originating with RSA as a means of implementing their (then patent controlled) algorithm and the PKCS series of standards.

The second version http://www.imc.org/rfc2311 dates from 1998 but had a number of serious restrictions, one of which was a limitation to 40 bit DES (perhaps as a result of US attempts to prevent the export of strong cryptographic products).

The third version of the IETF standard http://www.imc.org/rfc3369 is dated August 2002 and says, "This syntax is used to digitally sign, digest, authenticate, or encrypt arbitrary message content."

Battle of the Standards

Many of the difficulties implementers have faced with the S/MIME standard have been caused by aiming at a constantly moving target. Far from the 'standard' being stable for several years so that product manufacturers could have time to gain experience there have been changes to the encryption algorithms being used.

Just as importantly, and not immediately clear from the IETF documents, the standard places reliance upon more than one other standard for it to function. Key amongst these is the format of a public key certificate as expressed in a standard called X.509. This was developed by the PKIX working group of IETF, and the following quotation from their charter may indicate the spread of interdependent standards and their sources:

"PKIX has produced several informational and standards track documents in support of the original and revised scope of the WG. The first of these standards, RFC 2459, profiled X.509 version 3 certificates and version 2 CRLs for use in the Internet. Profiles for the use of Attribute Certificates (RFC XXXX [pending]), LDAP v2 for certificate and CRL storage (RFC 2587), the Internet X.509 Public Key Infrastructure Qualified Certificates Profile (RFC 3039), and the Internet X.509 Public Key Infrastructure Certificate Policy and certification Practices Framework (RFC 2527 - Informational) are in line with the initial scope.

The Certificate Management Protocol (CMP) (RFC 2510), the Online Certificate Status Protocol (OCSP) (RFC 2560), Certificate Management Request Format (CRMF) (RFC 2511), Time-Stamp Protocol (RFC 3161), Certificate Management Messages over CMS (RFC 2797), Internet X.509 Public Key Infrastructure Time Stamp Protocols (RFC 3161), and the use of FTP and HTTP for transport of PKI operations (RFC 2585) are representative of the expanded scope of PKIX, as these are new protocols developed in the working group, not profiles of ITU PKI standards."

Given that there are more available standards for implementing with 'than you can shake a stick at' and almost as many implementers with their own (or their employer's proprietary) ideas about how they should be implemented the result has been a predictable disaster with suppliers concentrating on establishing market share. At the same time, the (then freeware) provider PGP Inc. also had it's own adaptation called PGP/MIME, which would have become the de facto standard if a number of manufacturers had not perceived that they were missing out on a major market opportunity.

The Internet Mail Consortium remarked in their paper on S/MIME and Open PGP that, "S/MIME v3 and OpenPGP are both protocols for adding authentication and privacy to messages. However, they differ in many ways, and are not designed to be interoperable. … Of course, having two protocols that do the same thing is much worse than having one. At some point soon, IMC would like to get clear guidance from its members about a single protocol that it should pursue. Until then, it *(IMC)* will work with the many companies and individuals who are writing and implementing each of the protocols to help guide them towards standards status."

These rather cold statements do not sit well with advertising claims that the use and implementation of the S/MIME standard is already mature and that any product you purchase will immediately interoperate seamlessly with all products from major manufacturers and service providers.

For the last three years EEMA https://www.eema.org/home1.asp the European E-Mail Association has run an e-mail PKI challenge to see how many vendors are able to achieve e-mail or PKI interoperability. Early this year (2002) the UK National Security Agency (CESG) published a report on the results of their interoperability trials http://www.cesg.gov.uk/technology/pki/cloud-cover/Final%20Report%20v1-2.pdf.

This is the brightest piece of reading to date, showing that ten providers, Baltimore, Conclusive, Entrust, Novell, Reflex Magnetics, RSA Security, Guardeonic Solutions, Tumbleweed and Utimaco had achieved a reasonable level of technical interoperability in tightly defined conditions and whilst their engineering support was on hand to resolve difficulties. Sadly we do not see the names of Microsoft, Sun or Lotus in the list.

So interoperability works then?

To an extent. Even the CESG paper is careful to point out that, "CESG strongly recommends that vendors look closely at their use of directories to share information with other products, and aspects of key management." All the solutions require considerable specification and definition of 'security policies' that themselves do not interoperate before the user sees a consistent behavior between one product and another.

There still remains unsolved problems such as 'what does revocation mean and how do you realistically implement it outside the boundary of a single enterprise' and 'how long should you wait for a response from an external authority before making what kind of decision?' or 'what does time stamping mean if you connect to the service over the Internet?' and 'everyone talks about non-repudiation but where is the official definition for what it is and precisely what it means?'

In the meantime users are left horrified and confused. To most reasonable human beings, the text of the white paper so far is little more than meaningless techno-babble.

What does interoperability mean for the ordinary person?

To 'the man on the Clapham omnibus' there is an expectation that things will work in a particular way – consistent with his normal experience or otherwise self-evident from its behavior.

If he has a physical document not only can he sign it, but so can others. Perhaps he can lock it up so that people can only see it if he gives permission (either by unlocking it or by lending them the key or giving them a copy, depending on how careful he wants to be). If a physical document is altered he would want to know because he might not agree. If he does agree he might want other people to also show that they agree.

So interoperability for the user means looking at functions that he understands and trying to match them to the capabilities of technology.

Now as we know, anything is technically possible (given that you don't mind the cost, the timescale and changing your requirements to fit what is delivered). So it is theoretically possible to say that all the user interoperability requirements can be met by S/MIME, but no expert worth his salt would ever recommend it.

What the standard does is baldly stated by the IETF group, "This syntax is used to digitally sign, digest, authenticate, or encrypt arbitrary message content." There is no thought about multiple signing, having an original and then subsequent alterations together with an audit history, and maintaining all these together.

What routes are there going forwards?

At the moment there are no standards addressing the handling of objects such as text or files where they are to be considered as real objects – in the computer programming sense of having inheritance, having lifecycle, being capable of audit. Current standards address themselves only to the application of a security layer, usually as a temporary envelope that is (seamlessly) discarded once it has been checked rather than persisting.

Some suppliers, such as ArticSoft with their ContentAssurity product have gone some way to providing these features without altering the content they are protecting. Most others only consider a message to be transitory text being exchanged between two places. Others are focused upon one or two specific document types (CAD drawings, Word documents) without considering associated text and other documents that may be relevant to the process taking place.

We are some time away from seeing formal standards in this area. This is mainly because technical standards are still focused upon how technical mechanisms are supposed to function. They leave it up to the implementer to find the best way (if there is one) of applying the standard to a business requirement. Since technical standards writers are not usually business people it may be some time before that world aligns itself with normal commercial or personal requirements.

At the moment we need to spend more time implementing business based solutions that can be made to operate quickly and without massive costs, and gain practical experience before setting new standards.