Tailored Signatures with DOSETA

D. Crocker
Brandenburg InternetWorking
bbiw.net

April 26, 2011

Abstract

Trust begins with a verifiably correct identifier, coupled with claims about the meaning of the identifier's presence. To date, Internet-scale authentication services have achieved limited deployment and less use, often with far more restricted actual semantics than are assumed by users. Domain Security Tagging (DOSETA) creates convenient specification, development, deployment and use of trust-related identifiers, by generalizing upon the core mechanisms of DKIM. Hence the effort needed for development and use of new authentication services is minimized. DOSETA is based on domain names as (organizational) identifiers and self-certifying keys stored in the Domain Name System. This paper summarizes the scope and details of DOSETA and describes an initial application for signing MIME objects.

Introduction

What does a signature mean? In the paper world, it might mean authorization for a charge on a credit card, or acknowledgment that a letter has been received, or the sale of a house. The language surrounding the signature defines its meaning. In the digital world, existing signature mechanisms typically are not as flexible. The meaning is built into the specification for a particular signature and the effort to create a new type of signature is typically quite high. Consequently, there is a very small range of digital signatures performed on the Internet today.

What if it were easy to define a new type of signature with new semantics? This is not an issue of basic algorithms, but of defining the semantics and the packaging, along with a small matter of a certificate authority, to start the trust hierarchy, and of deployment and use effort. DOmain SEcurity TAgging [DOSETA] provides this flexibility and ease. It is based on the core mechanisms from [DKIM], extracted into a library of protocol components that minimize the incremental effort to develop a purpose-built data signature mechanism.[1] This protocol design library is used by a signature protocol

[1] These core aspects of DOSETA were the essential contributions developed for DKIM's predecessor, DomainKeys, by Mark Delany, then of Yahoo! DKIM was an evolution of DomainKeys. DOSETA is merely stealing these earlier innovations for re-purposing to other signing activities.

designer to provide a high point of specification departure, primarily limited to definition of semantics and mapping from a template to the specifics of the environment for the new signature.

The core DOSETA services include:

1. A standardized mechanism for access to a signature's public key, using existing infrastructure.
2. A packaging method for associating key-related information with the data being signed, in a manner that can be invisible to a non-participating receiver of the data.
3. A basic set of cryptography algorithms, but this is extensible by registration.
4. A basic set of algorithms for data canonicalizations, to withstand small changes to the data when it is in transit, but this is extensible by registration.

This core is enhanced with a "template" for performing object-oriented authentication on data that conform to a classic header/content model. The template supports asserting a list of signature semantic "claims" through an extensible registry. Hence, a signature can assert multiple meanings, such as validation of the purported author and validation of the content.

An object-oriented approach is distinguished from a channel-oriented approach, such as SSL/TLS. The philosophical difference essentially means that a channel-oriented scheme protects the path and does not care what bits pass over it. An object-oriented scheme protects a package of data and does not care what path the data travel.

DOSETA is based on some simplifying assumptions:

1. Signatures are by organizations, not individuals. Hence, the identity and naming mechanism is relatively coarse-grained, specifically in the form of a domain name. (It is possible to use domain names to refer to individuals, but this has not typically proved practical at scale.)
2. Signature keys are self-certifying. Because a domain name is the signature identifier, a public key that is associated with the signature is stored under that name in the DNS. The premise is that the owner of the domain name controls what is put into the DNS under that name. Self-certifying keys have significant appeal, but they also have limitations for use. Some signatures really do need to be vetted by an outside trust authority. DOSETA does not (currently) satisfy such a requirement, when asserted.

   To the extent that higher-valued signature assurances are needed, adding in the use of DNSSEC can be helpful to reduce a concern that an independent agent might have modified the DNS records under the name.

Key Storage

DOSETA re-uses the DKIM/DomainKeys key storage mechanism. This employs a DNS TXT resource record, containing public key parameters to be used when validating a signature. A key query is made to the domain name:

<*selector*>**`._domainkey.`**<*domain*>

where:

domain: is the identifier used to do the signing.

selector: is an administrative qualifier, which supports use of multiple keys for the same identifier, such as to permit multiple individuals being able to sign, or to permit rolling over to a new key in a graceful manner. The full string is used to do a retrieval, but the string that specifies the signing "identifier" is only the base <*domain*> string.

The constant string "`_domainkey`" is used to signal that the sub-tree provides attribute information to the parent domain, in this case the parameters for a public key.

The key storage mechanism re-uses the DKIM/DomainKeys name format on the theory that there is no added security in defining a different scheme and name tree, such as using a different "underscore" constant string, and that there is considerable administrative benefit in avoiding the effort to create and maintain a new set of keys. However, it is a small matter for any new protocol designer to create a new naming tree, by specifying a different constant. (Populating and maintain a new tree of keys will be less easy.)

Packaging of Parameters

Digital signature mechanisms usually impose their presence on the receiver of data. [OpenPGP] has specialized, in-line packaging. [S/MIME] uses MIME Multipart/Secure packaging. For recipients of the data who do not participate in the security mechanism, this largely renders the data unusable.[2]

In contrast, the DOSETA scheme puts the signature information into a separate header field, out of the way of software (and users) not prepared to process it. Within this header field, parameters use a simple attribute/value textual tagging format.

Cryptographic Routines

DOSETA re-uses the set of cryptography algorithms used for DKIM. These are defined as extensible sets, so that the effort of adding new algorithms is primarily the work of defining new registry entries.

---

[2] Note also that OpenPGP and S/MIME are typically tied to confidentiality content encryption, as well as signing. DOSETA can be enhanced to support confidentiality but it currently only has the task of authentication.

Data Canonicalization

In transit, some services subject data to transformation, such as reducing a string of linear white space to a single string, or mapping newline to a particular character (or character pair.) Changes like these often are benign. They do not change the "meaning" of the data and it makes sense to define the signature in a way that is robust against the changes. DOSETA re-uses the two canonicalization schemes currently in DKIM. However, an additional scheme is being contemplated, to provide robustness against some additional transformations that appear to be common. Note, however, that the more robust a canonicalizations algorithm, the more opportunity there is for a bad actor to find a way to exploit the signature insensitivity.

Signature Template

DOSETA defines a generic signing protocol template, for data that has a header and separate content, such as email and MIME. A variety of other data formats appear to be friendly candidates for this model, such as JSON and XML.

When conforming to the template, a new signature designer merely needs to define:

D-Signature association:	How is the signature data linked to the cover header and the content?
Semantics signaling:	How does the consuming application detect that the signature is present? Although this will normally be accomplished by detecting the signature in a standardized header field that holds the signature attributes, other approaches might make sense in some situations.
Semantics:	The meaning(s) of a signature. A registry supports definition of multiple "claims" that can be listed and asserted by a signature.
Header/Content mapping:	How are the actual header and content data for a particular signing service mapped from the generic DOSETA template?

Claims Registry

As described earlier, a signature can have different or multiple meanings. The DOSETA signature template defines a registry for signature semantics, so that one or more can be asserted at the time of signing. The initial entries for the registry are:

handled:	The signer had a role in processing the object. (This claim is approximately equivalent to the semantics of DKIM.)
validauth:	Purported author of object is valid

validdata: All of the content is valid.

validfields: The listed portions of the object are valid.

MIME Authentication

As an initial demonstration of DOSETA's flexibility and utility, there is a definition of authentication for signed MIME bodies [MIMEAUTH].

To follow the template described above:

D-Signature association:	The **`Content-Authentication:`** field is defined to hold the parameters
Semantics signaling	The presence of a **`Content-Authentication:`** signals the presence of a MIMEAUTH signature.
Semantics:	The meaning of a MIMEAUTH signature is asserted by listing one or more claims from the DOSETA Claims Registry.
Header/content mapping:	Specified MIME fields map to the DOSETA template's header and the MIME Body maps to the DOSETA templates content.

DOSETA's signature template re-uses an interesting feature from DKIM, namely the selective inclusion of header fields to be covered by the signature. The reason that not all fields are automatically included refers back to DKIM's email context: In transit, some fields are added (and therefore can not be part of the signature) and some fields are subject to particularly violent transformations that would break the signature.

In addition to permitting selective inclusion of MIME header fields, this mechanism permits selective inclusion of fields that are part of the container holding the data object. That is, the signature can also cover parts of the "parent" object, such as an email message header or an HTTP header. Hence, this mechanism can be useful for signing a web page.

Status

The DOSETA and MIMEAUTH specifications are quite new and are still going through early reviews. Early returns have been encouraging.

The primary open source implementation of DKIM is [OpenDKIM]. There are plans to enhance the library so that it also support DOSETA and MIMEAUTH.

References

[DKIM] Allman, E., Callas, J., Delany, M., Libbey, M., Fenton, J., and M. Thomas, "DomainKeys Identified Mail (DKIM) Signatures", RFC 4871, May 2007.

[DOSETA] Crocker, D. and Kucherawy, M., “DomainKeys Security Tagging (DOSETA)”, Work in Progress, <http://datatracker.ietf.org/doc/draft-crocker-doseta-base/>, March 2011.

[MIMEAUTH] Crocker, D. and Kucherawy, M., “MIME Content Authentication using DOSETA (MIMEAUTH)”, Work in Progress, <http://datatracker.ietf.org/doc/draft-crocker-doseta-mimeauth/>, February 2011.

[OpenDKIM] The OpenDKIM Project, <http://opendkim.org/>

[OpenPGP] Callas, J., Donnerhacke, L., Finney, H., and R. Thayer, "OpenPGP Message Format", RFC 4880, November 2007.

[S/MIME] Ramsdell, B. (ed), “S/MIME Version 3 Message Specification”, RFC 2633, June 1999