DomainKeys Identified Mail Demonstrates Good Reasons to Re-invent the Wheel

Stephen Farrell

Distributed Systems Group,
Department of Computer Science,
Trinity College, Dublin 2, Ireland
stephen.farrell@cs.tcd.ie
https://www.cs.tcd.ie/Stephen.Farrell/

Abstract. DomainKeys Identified Mail is an anti-spam proposal that involves mail servers digitally signing outbound email and verifying signatures on inbound email. The scheme makes no use of existing public key infrastructure or email security standards. This paper provides an outline of the scheme and discusses some reasons why re-use of existing standards is inappropriate in this context.

1 Introduction

Domain Keys Identified Mail (DKIM) [1] is an anti-spam approach that involves digitally signed email. The most basic rationale for DKIM is that it allows for better whitelist management since the digital signatures allow a verifier to more reliably detect that a message has originated from some mail domain. Even if it did nothing else, DKIM might be justified on this basis - that it is a real improvement over whitelists based on mail server IP addresses.

Typically however DKIM signature checking would form a part of a broader set of anti-spam measures, so a valid signature does not directly result in delivery of the message, but may rather be used to "turn down" the level of subsequent checking for that message, thus saving resources and allowing those released resources to be dedicated to checking unsigned email. In this way it is hoped that DKIM will result is more reliable delivery of genuine messages as well as better detection of certain types of spam.

DKIM also involves a second and separate mechanism allowing a domain to express a policy about its outbound email. In particular this policy allows a domain to state that it actually sends no email at all, which would be appropriate for some banking server domains. This mechanism, when combined with the signature mechanism, is aimed at reducing the ability of bad actors to create emails that appear to originate from domains where such strict policies actually apply. The basic idea here is to make some current phishing techniques somewhat less attractive, though recognizing that DKIM cannot "solve" phishing, or, more generally, spam.

From the above, we can see that DKIM will require some way to distribute public keys for signature verification – basically a public key infrastructure (PKI) or equivalent. There are at least three standard ways to do this, using the X.509 based PKIX

A.S. Atzeni and A. Lioy (Eds.): EuroPKI 2006, LNCS 4043, pp. 145 – 153, 2006.

approach [2], the OpenPGP based approach [3] or the XKMS approach [4]. In fact DKIM uses none of these. Similarly, DKIM must use some signature format, and again there are some standards in this area, primarily S/MIME [5] and XML Signature [6]. And once more, DKIM doesn't make use of these.

In the remainder of the paper we give a brief outline of DKIM, then examine why DKIM doesn't use a PKI, followed by consideration of why DKIM doesn't use one of the standard signature formats and lastly we offer some tentative conclusions.

2 DKIM Outlined

As stated DKIM consists of two parts – the first is the basic signature scheme [7] and the second describes the Sender Signing Policy (SSP) [8]. There is also a threat analysis document [9] that provides some additional background in terms of the threats that DKIM is intended to counter and also in terms of the new threats which come into play when a system like DKIM has been deployed. Since all of these documents are currently in draft form, we won't consider them in too much detail – detail that is still subject to change – but will rather take a somewhat abstract view of DKIM.

DKIM signatures are carried in a mail header field (DKIM-signature), placed there by a mail server, often called a Message Transfer Agent (MTA), and mostly not by a Mail User Agent (MUA). Similarly, the general intent is that DKIM signature verification is carried out by an MTA and not by a MUA. DKIM is therefore primarily a server-server protocol unlike more traditional email security protocols. While there have been suggestions that DKIM-enabled MUAs might be useful, the current IETF activity is not addressing this so we will therefore ignore DKIM-enabled MUAs in the remainder of the paper.

A DKIM signature can cover the body of the message as well as a number of mail header fields, in particular the "From:" header will often be signed. The DKIM-signature header field indicates which other parts of the message were signed, as well as the signing algorithm and other signature parameters.

The general model is that the public key to verify the signature is stored in the DNS entry of the signing/originating domain (which can sometimes differ!). A verifier therefore has to do a new DNS lookup to retrieve that key as part of signature verification. However, MTAs commonly do such lookups at the moment, e.g. to verify that a sender is not on a DNS based blacklist. Once the public key is retrieved then the signature can be checked and the message passed for further processing.

One important aspect of DKIM signature processing is that badly signed messages are to be treated as if they were unsigned. Practically, this is necessary because so many MTAs actually modify[1] messages in transit that it will be quite common for signatures not to verify for totally innocuous reasons.[2] So, in contrast to many signature applications, signature verification failure doesn't necessarily lead to a message processing failure.

---

[1] At the moment how multiple signatures might be placed on a message, e.g. by mail list agents is not well defined. So we omit consideration of such issues for the present.

[2] Hopefully, DKIM will start a move to only make signature-friendly changes to messages, but that's for the future.