

Using WebDAV for Improved Certificate Revocation and Publication

David W. Chadwick and Sean Anthony

Computing Laboratory, University of Kent, UK

Abstract. There are several problems associated with the current ways that certificates are published and revoked. This paper discusses these problems, and then proposes a solution based on the use of WebDAV, an enhancement to the HTTP protocol. The proposed solution provides instant certificate revocation, minimizes the processing costs of the certificate issuer and relying party, and eases the administrative burden of publishing certificates and certificate revocation lists (CRLs).

Keywords: revocation, CRLs, LDAP, HTTP, WebDAV.

1 Introduction

The most common and standardized way of publishing X.509 certificates is in corporate LDAP servers. Several technical problems with the use of LDAP directory servers have been widely documented [1], including: the use of binary encodings, the lack of a distributed directory capability, and the inability to search for certificates containing specific fields. Other problems are less technical in nature and instead are operational, but they are nevertheless just as inhibiting to successful deployments. For example, most corporate firewalls do not allow the LDAP protocol to pass through them, therefore certificates or certificate revocation lists (CRLs) cannot be easily accessed by external organizations. Finally, we have the complexity of installing and managing LDAP servers, for example, loading the correct schema and setting the correct access rights, which although not insoluble, nevertheless cause frustration and inconvenience especially to small scale deployments with a lack of specialist staff. For these reasons we wanted to find an alternative mechanism to LDAP for publishing X.509 certificates and CRLs that would not suffer from these problems. We wanted a generic solution that would support both public key certificates (PKCs) and attribute certificates (ACs), and that most (ideally all) organizations would already be familiar with. We chose to use Apache servers and the HTTP protocol, since these are ubiquitous. But HTTP on its own is insufficient, since it does not provide a number of useful features, such as the ability to search for specific content. For this reason we investigated (and subsequently implemented) the WebDAV extensions to HTTP [2].

The most common way of revoking public key certificates is through the use of CRLs. However these suffer from a number of well documented operational problems such as the potential for denial of service attacks from lack of availability, or the

consumption of too many resources due to their increasingly large size. We will address the whole issue of certificate revocation in the next section, and describe why we have adopted an alternative approach for revocation based on WebDAV.

The rest of this paper is structured as follows. Section 2 re-appraises the whole issue of certificate revocation and proposes a different approach to addressing this issue. Section 3 describes the WebDAV extensions to HTTP and how they can be used for X.509 certificate and CRL storage and retrieval. Section 4 describes our implementation of WebDAV in our PERMIS authorization infrastructure, in order to store X.509 attribute certificates used for authorization. This mechanism can similarly be used by PKIs to store public key certificates and CRLs. Section 5 discusses our approach, compares it to other work, and concludes.

2 Reappraising Revocation

There are several different approaches that have been taken to the complex issue of revocation of certificates, and of informing remote relying parties when revocation has taken place. A relying party is any Internet based service that consumes the issued certificate (whether it be a PKC or an AC). The primary objective of revocation is to remove a certificate (and all its copies, if any) from circulation as quickly as possible, so that relying parties are no longer able to use it. If this is not possible, a secondary objective is to inform the relying parties that an existing certificate in circulation has been revoked and should not be used or trusted. The latter can be achieved by requiring either the relying parties to periodically check with the certificate issuer, or the certificate issuer to periodically notify the relying parties. Of these, requiring the relying parties to periodically check with the certificate issuer is preferable for two reasons. Firstly, it places the onus on the relying parties rather than on the issuer, since it is the relying parties who are taking the risk of using revoked certificates. Secondly, an issuer may not know who all the relying parties are, so will have difficulty contacting them all, but the relying parties will always know who the certificate issuer is.

The simplest approach to certificate revocation, that used by X.509 proxy public key certificates [6], the Virtual Organisation Membership Service's (VOMS) X.509 attribute certificates [8] and SAML attribute assertions [7] (which are to a first approximation simply XML encoding of attribute certificates), is to never revoke a certificate, and instead to issue short lived certificates that will expire after a short period of time. The certificates are thus effectively and automatically removed from circulation after this fixed period expires. The assumption is that it is unlikely that the certificates will ever need to be revoked immediately after they have been issued and before they have expired due to abuse, therefore the risk to the relying parties is small. Risk (or more precisely risk exposure) is the probability of occurrence multiplied by the loss if the event occurs. Because short lived certificates are only valid for a short period of time, the probability of occurrence is small. Of course, the loss or amount of damage that can be done in a short period of time can be huge, so short lived certificates are not always the best solution where the resulting loss can be high. Consequently SAML attribute assertions have the optional feature of containing a "one time use" field which means that the consuming service can only use the attribute assertion once to grant access, and then it should never be used again. This is