Enhanced Certificate Revocation System

by

Silvio Micali

Laboratory for Computer Science
MIT

ROUGH DRAFT

ABSTRACT.
We apply off-line digital signatures to provide a novel approach to certificate revocation. Our approach dismisses with traditional CRLs and yields public-key infrastructures that are several-hundred times cheaper to run than traditional ones.

More generally, our technology also yields effective methods to lengthen the validity of a digital signature.

1 Certificates and Public-Key Certificates

In many a setting, it is necessary to certify certain data, as well as to revoke already issued certificates. For instance, in a Public-Key Infrastructure (PKI) it is necessary to certify users' public keys.

In a digital signature scheme, each user U chooses a signing key SK_U and a matching verification key, PK_U. User U uses SK_U to compute easily his digital signature of a message m, $SIG_U(m)$, while anyone knowing that PK_U is U's public key can verify that $SIG_U(m)$ is U's signature of m. Finding $SIG_U(m)$ without knowing SK_U is practically impossible. On the other hand, knowledge of PK_U does not give any practical advantage in computing SK_U. For this reason, it is in U's interest to keep SK_U secret (so that only he can digitally sign for U) and to make PK_U as public as possible (so that everyone dealing with U can verify U's digital signatures), At the same time, in a world with millions of users, it is essential in the smooth flow of business and communications that PK_U be *really* the legitimate key of user U. To this end, users' public keys are certified. At the same time it is also necessary to revoke some of the already issued certificates.

Certification and Revocation of Public keys. Typically, certificates for users' public keys are produced and revoked by certifying authorities called CAs.[1] A CA can be considered to be a trusted agent having an already certified (or universally known) public key. To certify that PK_U is U's public key, a CA typically digitally signs PK_U together with (e.g., concatenating it with) U's name, a certificate serial number, the current date (i.e., the certification date), and the expiration date.[2] The CA's signature of PK_U is then inserted in a Directory and/or given to U himself.

Upon receiving the (alleged) digital signature of user U of a message M, $SIG_U(M)$, a recipient R needs to obtain a certificate for PK_U. (Indeed, $SIG_U(M)$ may be a correct digital signature of M with respect to some public key PK_U, but R has no guarantee that PK_U is indeed U's public key.) Recipient R may obtain this certificate from the Directory, from his own memory (if he has previously cashed it), or from U himself. Having done this, R verifies (1) the correctness of the CA's certificate for PK_U with respect to the CA's public key, and (2) the correctness of $SIG_U(M)$ with respect to PK_U. (If the CA's public key is not universally known, or cashed with R, then a certificate for this key too must be obtained.)

Certificate retrieval is thus quite doable (though not necessarily cheap). Unfortunately, this is not the only retrieval that R needs to do. Indeed, it is crucially important that R makes sure that the certificate for PK_U has not been revoked. This check, of course, may not be needed after the certificate's expiration date, but is needed during the certificate's alleged life-time. A user's certificate can be revoked for a variety of reasons, including key compromise and the fact that the user is no longer associated with a particular CA.

To enable a recipient to establish whether a given certificate has been revoked, each CA periodically issues and gives the Directory a Certificate Revocation List (CRL for short), in general containing an indication of all the (not yet expired) certificates originally issued by him. A CRL typically

[1] A complete public-key infrastructure may involve other authorities (e.g., PCAs) who may also provide similar services (e.g., they may certify the public keys of their CAs). For simplicity sake, however, we shall ignore the complete picture: in fact, the present inventions can be easily applied to the full picture anyway.

[2] Before so certifying U's public key, it is necessary to perform additional steps, such as properly identifing user U. The present inventions, however, do not depend on these additional steps.

consists of the issuer's digital signature of (1) a CRL *header* comprising the issuer name (as well as the type of her signature algorithm), the current date, the date of the last update, and the date of the next update, together with (2) a complete list of the revoked certificates (whose date has not yet expired), each with its serial number and revocation date. Since it is expected that a CA revokes many of her certificates, a CRL is expected to be quite long.

After performing some checks on the CA's CRL (e.g., checking the CA's digital signature, checking that the CRL has arrived at the expected time, that a certificate declared revoked in the previous CRL of that CA —and not yet expired— still is revoked in the current CRL, etc.), the Directory stores it under its CA name.

When a user queries it about the revocation of a certificate issued by a given CA, the Directory responds by sending to the user the latest CRL of that CA. The user can then check the CRL signature, the CRL dates (so as to receive a reasonable assurance that he is dealing with the latest one), and whether or not the certificate of interest to him belongs to it.

While CRLs are quite effective in helping users establishing which certificates are no longer deemed valid, they are also extremely expensive, because they tend to be very long and need to be transmitted very often.

CRLs Costs

The National Institute of Standards and Technology has tasked the MITRE corporation a study of the organization and costs of a PKI for the Federal Government [1]. This study estimates that CRLs constitute by far the largest entry in the Federal PKI's cost list. According to MITRE's estimates/assumptions, in the Federal PKI there are about three million users, each CA serves 30,000 users, 10% of the certificates are revoked[3], CRLs are sent-out bi-weekly, and, finally, the recipient of a digital signature requests certificate information 20% of the time.[4] The study envisages that each revoked certificate is specified in a CRL by means of about 9 bytes: 20 bits of serial number and 48 bits of revocation date. Thus, in the Federal PKI, each CRL is expected to comprise thousands of certificate serial numbers and

[3] 5% because of key compromise and 5% because of change in affiliation with the organization connected to a given CA.

[4] The remaining 80% of the time he will be dealing with public keys in his cache.

their revocation dates; its header, however, has a fixed length, consisting of just 51 bytes.

At 2 cents per kilobyte, the impact of CRL transmission on the estimated yearly costs of running the Federal PKI is quite stunning: if each federal employee verifies 100 digital signatures per day on average, then the total PKI yearly costs are $ 10,848 Millions, of which 10,237 Millions are due to CRL transmission. If each employee is assumed to verify just 5 digital signatures a day on average, then the total PKI yearly costs are $ 732 Millions, of which 563 Millions are due to CRL transmission.

The MITRE study thus suggestes that any effort should be made to find alternative and cheaper CRL designs. This is indeed our goal.

2 The New Certification/Revocation System

To avoid the dramatic CRL costs, we put forward a novel *Certification/Revocation System*, where requesting users no longer receive the latest list of revoked certificates (of a given CA), but an individual and very succint piece of information about every single certificate they are interested in.

The new system replaces CRLs with novel information structure called *Certification/Revocation Status*, *CRS* for short. Unlike CRLs, each CRS is a short and individualized piece of information for a given certificate. We envisage CRS to be issued with the same frequency that was deemed appropriate for the CRLs. In a CRS update, a CAs sends twenty times more bits and she would send for comparable CRL update. However, CRS allow the Directory to answer users' queries much more succintly than before.

2.1 CRS Usage

Let us now describe the prefered embodiment of the new certification/revocation system. For simplicity of presentation (and because their very low cost allows us to do so), we shall envisage here that CRS are updated daily.

CA operations.

- *(Making a Certificate.)* A CA produces the certificate of a user's public key by digitally signing together traditional quantities (e.g., the user's public key, the user name, the certificate' serial number, the type of signature algorithm of the issuer, the certification date, and the expiration date) plus two new quantities: a 100-bit value Y —for "YES"— and a 100-bit value N —for "NO". These values are, at least with very high probability, unique to the certificate.

The CA generates Y by selecting a secret 100-bit values, Y_0, and then evaluating on it a given one-way function F 365 times (i.e., as many days in a year).[5] Thus, $Y = Y_{365} = F^{365}(Y_0)$. The CA generates N by selecting a secret value N_0 and then evaluating F on it once; that is, $N = F(N_0)$.

The CA may select Y_0 and N_0 at at random (in which case she must separately store them) or pseudo-randomly (e.g., she computes them from a secret master key —which she keeps in storage— and other inputs such as the string YES —respectively, NO,— the certificate serial number, and the issue date) in which case she can recompute them when needed rather storing them at all times.

- (Updating the CRS.) Daily, a CA sends the Directory the following information:

  (a) An authenticated and updated dated "list" of all serial numbers corresponding to issued and not-yet-expired certificates.

  For simplicity, let this information consist of the CA's digital signature of a 2^{20}-bit string S together with the current date (note that S comprises as many bits as there are serial numbers). The nth bit of S is 1 if serial number n corresponds to an issued and not-yet-expired certificate.

  (b) The new certificates made that day; and

  (c) For each not-yet-expired certificate made by her, she sends a 100-bit value computed as follows. Assume that the current day is the ith day in some given system of reference (i.e., the ith day of the year, or the ith day after the start date of the certificate, and so on). Then, if the certificate is still valid, the CA sends the value Y_{365-i} $(= F^{365-i}(Y_0)$, which she may easily compute by evaluating

[5] Rather than just a one-way function, a CA C may use several one-way functions, or a one-way hash function H. For instance, she may choose $Y_1 = H(Y_0, C, 1, \mathit{date}, \mathit{serial\ number})$, $Y_2 = H(Y_1, C, 2, \mathit{date}, \mathit{serial\ number})$, and so on.

F $365 - i$ times on input Y_0). If the certificate has been revoked that very day, she sends the value N_0.

For each revoked certificate the CA preferably also sends her direct digital signature that the certificate has been revoked, including additional infornation, such as the revocation date, reasons for revocation, etc., and no longer needs to send other (c)-type information about such a certificate.

Note 1: The value $Y = Y_{365}$ is the public-key of a second digital signature scheme, whose secret key is Y_0. This second scheme is capable of signing a limited number of messages; (namely, the integers between 1 and 365), but it is very fast, since there are one-way functions F that are extremely easy to evaluate. Indeed, the CA is the signer of an off-line/on-line signature scheme in the sense of Even, Goldreich, and Micali [2]: in an off-line step, she uses a first (traditional) signature scheme to sign the public key Y within the certificate, and then, in an on-line step, she uses the second signature scheme to sign a value in the interval [1,365] in a most fast way.

The further signature of integer i, Y_{365-i}, indicates that a certificate is valid up to date i. Of course, if a certificate is valid up to date i, it is also valid up to any date between o and i. Indeed, if $j < i$, the signature Y_{365-j} was released before Y_{365-i}. Illegally extending the validity of a certificate is very hard and requires signing a message never signed before by the legitimate signer.

Updating a given CRS is very efficient, and at most 120 bits about it need to be sent to the Directory for each issued and not-yet-expired certificate: the certificate serial number (i.e., 20 bits) and a 100-bit (YES/NO) value.

Many other CRS designs are also possible; in particular, based on other types of off-line signing. They are not always as efficient, though. For instance, the CA can sign the certificate's serial number together with the new date i and YES (if the certificate continues to be valid), or together with the new date i, NO, and the revocation date (if the certificate has ceased to be valid.)

In the above system, the amount of information sent by a CA to the Directory at each update is roughly twenty times as long as a CRL. Indeed, in a CRL update, the CA sends, on average, 9 bytes (72 bits) for 10% of the certificates. For a CRS update, the CA sends 120 bits for each certificate in

Step (c), and just 1 bit per certificate in Step (a). The transmission of Step (b) is the same for both systems.

Directory Operations.

- *(Response to CRS Update.)* For every CA, the Directory preferably stores all not-yet-expired certificates issued by her, organized by serial number, and for each of them it also stores its latest YES-value, if the certificate is still valid, and the 100-bit value $F^{-1}(N)$ otherwise.

  The Directory checks that each newly received certificate is well-formed and properly signed. (In particular, it checks that the certification/issue date coincides with the current day.)

  The Directory checks that the latest list of not-yet-expired certificates of every CA is fine. (In particular, it checks that its date coincides with the current one, that the list is complete, and that no certificate declared invalid in the previous list is declared valid now.)

  For every certificate, the Directory, upon receiving its latest 100-bit value V, performs the following check. Assume that the current day is i, then the Directory checks that either $F^i(V) = Y_{365}$ or $F(V) = N$.

- *(Response to Users' Inquiries.)* Assume, for simplicity, that signature recipients receive the certificates of their signature senders from the senders themselves. Thus, users make Directory queries just for determining the validity status of a certificate already known to them.

  When a user U inquires about the status of a given certificate (e.g., by specifying its CA and its serial number), the Directory retrieves and sends to U the latest 100-bit value relative to that certificate.[6]

  Should U inquire about a serial number that does not correspond to any not-yet-expired certificate issued by the CA, then the Directory sends U a proof that no such certificate exists (using the information received from tha CA in step (a) of a CRS update.)

[6]If the certificate has been revoked, then the Directory may send the CA's direct digital signature of this fact as an alternative to sending $F^{-1}(N)$; else, it may send him this richer piece of information upon further demand of the user.

Note 2: The Directory is not much trusted, because it cannot "make valid" a revoked certificate. Indeed, if the current date is i, and the certificate has been revoked at date $j < i$, the Directory has only received from the CA the 100-bit values $Y_{365-(j-1)}, \ldots, Y_{365-1}$. Thus, to make the certificate appear valid, it should be able to compute Y_{365-i} $(= F^{-(i-(j-1))}(Y_{365-(j-1)}) = F^{365-i}(Y_0))$, and thus invert F at least once on input $Y_{365-(j-1)}$, which he cannot do because F is a one-way function and because (unlike the CA) it does not know Y_0.

Similarly, the Directory cannot "revoke" a valid certificate. Indeed, in order to convince U that the certificate has been revoked it should be able to compute $F^{-1}(N)$, which again it cannot do.

For these reasons we do not recommend that the CA signs the YES- and NO-values of every CRS update. Indeed, Y and N are signed within the certificate, and only the CA may easily invert F on them a few times.

The Directory does not even need to be trusted when it says that a given certificate "does not exist." Indeed, it provides U with a proof of this fact that is properly authenticated by the CA.

USER OPERATIONS.

If U has inquired about a certificate of CA with a given serial number and the Directory sends him a proof that no certificate with that serial number exists, U checks this proof.

Else, let i be the current date, and let V be the 100-bit value U receives from the Directory about the certificate he has inquired about, containing Y as its YES-value and N as its NO-value.

Then, U checks whether $F^i(V) = Y$ (in which case he concludes that the certificate is valid); if this is not the case, he checks whether $F(V) = N$ (in which case he concludes that the certificate has been revoked). If none of these two cases applies, he concludes that the Directory is purposely denying him service.

3 The Advantages of the New System

Our system enjoys three main advantages over the traditional CRL one:

1. It SAVES DRAMATICALLY on bit transmissions and costs.

   (Recall that there are few CRS/CRL updates. Indeed, they typically occur bi-weekly and are performed by the CAs which are few in numbers. By contrast, there are very many queries of users to the Directory about certificate validity.)

2. It always PROVIDEs A POSITIVE AND EXPLICIT STATEMENT about the validity status of each not-yet-expired certificate.

   (By contrast, CRLs provided only indirect evidence; that is, the absence of a given serial number from a CRL was taken to mean thatthe corresponding certificate was still valid. Positive and explicit statements are much clear and advantageous from a legal point of view —e.g., from the point of view of liability— preferable to "double negatives.")

3. It always allows a COMPLETE AND SATISFACTORY ANSWER to any possible query of a user to the Directory –and without trusting the latter in any special way.

   (By contrast, in a CRL-based system, if a user queries —by error, malice, or other reason— the Directory about a serial number S that does not belong to any not-yet-expired certificate issued by a given CA, the Directory cannot prove this to the user. Indeed, showing that the latest CRL of that CA does not comprise S is not such a proof. (It may actually be construed as proving that S's certificate is valid.) Even giving the user all not-yet-expired certificates issued by CA is not such a proof: the user may suspect that the Directory is purposely withholding the "right" certificate. Indeed, it is the CA to be basically trusted in the system, the Directory service is trusted to a much lesser extent.)

3.1 CRS Costs

Let us now illustrate more precisely the transmission and cost savings of the CRS approach. Assume, for concreteness, that a certificate, if not revoked, is valid for one year; that is, that the time interval between its issue date and its expiration date is one year. Since the savings of the CRS approach increase more than linearly with the total number of revoked certificates, and thus with the total number of certificates, let us assume that each user has only one certificate, and thus that each CA issues 30,000 certificates a year.[7]

Then, since 10% of the issued certificates are revoked before their expiration date, we expect that each CRL comprises 3,000 (=10% of 30,000) items. Therefore, disregarding the 51-byte header, the expected length of a CRL is some 27,000 (3,000 times 9) bytes; that is, some 214,000 bits.

Though in some occasions these CRLs will be "pushed" by the CAs directly to their users (like in the emergency following to a major compromise of the system), they are ordinarily distributed in two modes: (1) *bi-weekly* from each of the about 100 CAs to the Directory, and (2) *daily* from some Directory agent to a requesting user. Of the two costs, the second is absolutely greater. Even making the assumption that each user verifies only 5 digital signatures a day on average (and that 20% of the time he experiences a cache miss and queries the Directory), on average there will be 3 Million daily CRL transmissions due to Mode 2, versus less than 40 (=100 CAs times 2 days/5 working days) daily transmissions due to Mode 1.

Our certification/revocation system replaces each CRL with a CRS which is roughly twenty times as long. Thus, Mode-1 costs jump from 40 CRL-transmissions per day to the equivalent of 800 CRL-transmissions per day. (Assuming that CRL are updated bi-weekly while CRS daily, Mode-1 costs would jump from 40 CRL-transmissions per day to the equivalent of 2,000 CRL-transmissions per day.) However, each of the 3,000,000 Mode-2 costs will decrease from transmitting one CRL (i.e., 214,000 bits plus a digital signature) to transmitting just 120 bits. Therefore, *even assuming that CRS are updated daily,* CRS ARE 1,000 TIMES CHEAPER THAN CRLs.

References

[1] NIST. *Public-key Infrastructure Study.* Gaithersburgh, MD, April 1994.

[2] S. Even, O. Goldreich, and S. Micali. *On-Line/Off-line Digital Signing.* Proc. Crypto 89, Santa Barbara, CA, August 1989.

Protected by U.S. Patent No. 5,016,274.

[7]Presumably, instead, the typical user of a PKI will have multiple certificates, because he will use different signing keys for each of his different functions: private citizen, member of different organizations, sub-units, etc.