

Digital Signature Schemes

by

Henk Meijer and Selim Akl
Department of Computing and Information Science
Queen's University
Kingston, Ontario, Canada

May 1982

(Extended summary of paper presented
at CRYPTO 81, Santa Barbara,
California, August 1981)

This work was supported by a scholarship from the Ontario Graduate Scholarship Program and by Grants A3336 and G0381 from the Natural Sciences and Engineering Research Council of Canada.

1. Introduction

In order to fully exploit the advantages of electronic mail, communication systems are needed that allow the authentication and the validation of the messages exchanged. Both purposes can be achieved by the use of a digital signature, i.e. a message dependent quantity that can be computed only by the sender of the message based on some private information. Signature schemes can be divided into two categories: "true" and "arbitrated". True signatures can be produced and checked by the sender and the receiver. A judge can be called in to settle possible disputes. In an arbitrated signature scheme, on the other hand, all communications involve a so-called arbitrator, who authenticates and validates the signed messages. The security of most arbitrated signature schemes depends very heavily on the trustworthiness of the arbitrators who have access to the contents of the messages. This paper introduces four new digital signature schemes that decrease this dependency. The schemes involve one or more arbitrators who validate and authenticate messages and signatures. In order to hide the contents of the messages, data sent via these arbitrators is enciphered under a key known only to the sender (S) and the receiver (R). Nonetheless, the arbitrator (A) receives enough information to prevent the sender and receiver from giving an incorrect interpretation to a signed message.

In the following, let

k_{xy} = key shared by x and y

k_x^p = public key of x

k_x^s = secret key of x

E_k = enciphering function with key k

D_k = deciphering function with key k.

\oplus = bit-wise exclusive OR operation.

2. One arbitrator-secret key cryptosystem

S constructs $M = \langle S\text{-id, R-id, seq.nr, data} \rangle$ and splits M up into blocks M_1, M_2, \dots, M_t of m bits each. Now $\langle V, M_1, \dots, M_t, W \rangle$ where V is a random m-bit vector and $W = V \oplus M_1 \oplus \dots \oplus M_t$, is enciphered using a feedback encryption scheme as follows

$$C_0 = E_{k_{SR}}(V)$$

$$C_1 = E_{k_{SR}}(M_1 \oplus C_0)$$

$$C_2 = E_{k_{SR}}(M_2 \oplus C_1)$$

.

.

.

$$C_t = E_{k_{SR}}(M_t \oplus C_{t-1})$$

$$C_{t+1} = E_{k_{SR}}(W \oplus C_t).$$

The random vector V is used to disguise repeated messages and bit-sequences that often occur at the beginning of messages. For ease of presentation, we will use the notation

$$C = E_{k_{SR}} (\langle V, S\text{-id}, R\text{-id}, \text{seq.nr}, \text{data}, W \rangle)$$

i.e. $C = \langle C_0, \dots, C_{t+1} \rangle$.

Using the same notation C' is defined as follows

$$C' = E_{k_{SA}} (\langle V', S\text{-id}, R\text{-id}, \text{seq.nr}, C, W' \rangle)$$

where, as before, V' is a random m -bit vector and W' is the sum of the previous blocks.

S sends C' to A who decipheres it and applies to the resulting message the necessary authentication and validation tests. Now A drops W' , adds the signature C'_{t+1} , a "message true" or "message false" flag, vectors V'' and W'' , defined as usual, and sends C'' to R where

$$C'' = E_{k_{RA}} (\langle V'', V', S\text{-id}, R\text{-id}, \text{seq.nr}, C, C'_{t+1}, \text{flag}, W'' \rangle).$$

3. One arbitrator-public key cryptosystem

This scheme is based on the Rivest-Shamir-Adleman public key cryptosystem. Let a and b be the large prime numbers used in that system. The message $M = \langle M_1, \dots, M_t \rangle$ can be enciphered by

$$C_0 = E_k (V), C_1 = E_k (M_1 + C_0), \dots, C_t = E_k (M_t + C_{t-1}),$$

$$C_{t+1} = E_k (W + C_t), \text{ where } V \text{ is a random number less than } ab,$$

$$W = V + M_1 + \dots + M_t \text{ and addition is performed modulo } ab. \text{ Using}$$

the notation introduced in section 2 we can write $C = E_k (\langle V, M_1, \dots, M_t, W \rangle)$.

Using this feedback encryption process, C' , C'' and C''' can be constructed as follows:

$$C' = E_{k_R^p} (D_{k_S^s} (\langle V, S\text{-id}, R\text{-id}, \text{seq.nr}, \text{data}, W \rangle))$$

$$C'' = E_{k_A^p} (D_{k_S^s} (\langle V', S\text{-id}, R\text{-id}, \text{seq.nr}, C', W' \rangle))$$

$$C''' = E_{k_R^p} (D_{k_A^s} (\langle V'', S\text{-id}, R\text{-id}, \text{seq.nr}, \text{time}, \text{date}, C', \text{flag}, W'' \rangle))$$

S sends C'' to A who, after deciphering and adding the correct time and date of receipt and the appropriate flag, constructs C''' and sends it to R.

4. (n, i) arbitrator - secret key cryptosystem

To further decrease the dependency on the reliability of A, a scheme could involve a set of, say, n arbitrators. An (n, i) arbitrator scheme is defined as a scheme where S chooses a set of n arbitrators of which R has to use i to authenticate and validate the signed message. Increasing the value of n will make it harder for S to bribe enough arbitrators to deny a signed message. Similarly, forging the sender's signature becomes more difficult with an increasing value of i.

5. (n, i) arbitrator - public key cryptosystem

This scheme, which will require a simpler key management procedure than the one just described is a combination of the signature schemes of sections 3 and 4.

6. Conclusion

The four digital signature schemes presented in this paper do not hinge upon the integrity of the arbitrators to the same extent as previous schemes. A typical cost versus security tradeoff will be the deciding factor when choosing one of the four systems for a particular application.

For a list of references and a more detailed treatment of the concepts presented here, the reader is referred to [1].

7. Reference

- [1] H. Meijer and S. Akl, Digital Signature Schemes for Computer Communication Networks, Proceedings of the Seventh IEEE Data Communications Symposium, Mexico City, October 1981, pp. 37-41.