Zero-Knowledge Undeniable Signatures

(extended abstract)

David Chaum

Centre for Mathematics and Computer Science
Kruislaan 413 1098 SJ Amsterdam

SUMMARY: Undeniable signature protocols were introduced at Crypto '89 [CA]. The present article contains new undeniable signature protocols, and these are the first that are zero-knowledge.

Introduction & Motivation

Digital signatures [DH] are easily verified as authentic by anyone using the corresponding public key. This "self-authenticating" property is quite suitable for some uses, such as broadcast of announcements and public-key certificates. But it is unsuitable for many other applications. Self-authentication makes signatures that are somewhat commercially or personally sensitive, for instance, much more valuable to the industrial spy or extortionist.

Thus, self-authentication is too much authentication for many applications. On the other hand, the remaining previously known authentication schemes offer too little authentication. A judge or arbiter cannot use them to resolve disputes as is possible with self authentication. With zero-knowledge "identification" techniques, for example, a judge would not be convinced of anything by a transcript of the interaction, because by definition anyone could generate indistinguishable transcripts. Also with conventional "identify-friend-or-foe" protocols, or any other system where both parties have all relevant secret keys, the cryptography cannot stop either party from producing valid transcripts.

In short, cooperation of the signer should be necessary to convince another party that a particular signature is valid—but a signer, falsely accused of having signed a particular message, should be able to prove his innocence.

Undeniable Signatures

The relatively new technique called "undeniable signatures" [CA] achieves these objectives. An undeniable signature, like a digital signature, is a number issued by a signer that depends on the signer's public key as well as on the message signed. Unlike a digital signature, however, an undeniable signature cannot be verified without cooperation of the signer.

The validity or invalidity of an undeniable signature can be ascertained by conducting a protocol with the signer, assuming the signer participates. If a "confirmation" protocol is used, the cooperating signer gives exponentially-high certainty to the verifier that the signature does correspond to the message and the signer's public key. If instead a "disavowal" protocol is conducted, the signer gives exponentially-high certainty that the signature does not correspond to the message and the signer's public key. In both protocols a cheating signer, even with infinite computing power, has only an exponentially small chance of success and an overwhelming probability of being detected.

Applications

Undeniable signatures are preferable to digital signatures for many upcoming applications.

Consider, for example, the signature a software supplier may issue on its software, allowing customers to check that the software is genuine and unmodified. With undeniable signatures, only paying customers are able to verify the signature, and they are ensured that the supplier remains accountable for the software.

All manner of inter-organizational messages, such as so called EDI, are a natural candidate for signatures that provide for dispute resolution. But self-authentication would greatly increase the illicit salability of such information.

Also for personal transactions, non-repudiation may be an essential component of security for the service provider; but the customer would like to ensure that, for instance, the signatures do not later end up in the newspaper.

Outline

First the underlying cryptography and form of a signature will be presented, which are the same as in [CA]. Then the new confirmation protocol will be described in detail and its security argued. Next the new disavowal protocol is presented followed by sketches of proofs for its properties. Finally some more recent results are discussed.

CRYPTOGRAPHIC SETTING AND SIGNATURES

Consider using the group of known prime order p. All values transmitted between the participants are elements of this group, the multiplicatively denoted group operation is easily computed by all participants, and taking the discrete log in the group is assumed to be computationally infeasible.

One potentially suitable representation is the multiplicative group of the field GF(2^n), where $p = 2^n-1$ is prime. A second is the group of squares modulo prime q, where $q = 2p+1$. (Notice that such choices rule out the Pohlig-Hellman attack on the discrete log [PH].) An attractive variation on the second approach represents group elements by the integers 1 to p; the group operation is the same, except that all results are normalized by taking the additive inverse exactly when this yields a smaller least positive representative.

A suitable group of prime order p and a primitive element g are initially established and made public for use by a set of signers. Consider a particular signer S having a private key x and a corresponding public key g^x. A message m ($\neq$1) is signed by S to form a signature, denoted z, which should be equal to m^x.

Computing the private key from the public key, assuming only random messages are signed, is the the discrete log problem; forging signatures on random messages is at least as hard as breaking Diffie-Hellman key exchange.

CONFIRMATION PROTOCOL

A verifier V receiving z, which is claimed to be the signature of signer S on message m and thus equal to m^x, can establish the signature's validity using the confirmation protocol of Figure 1.

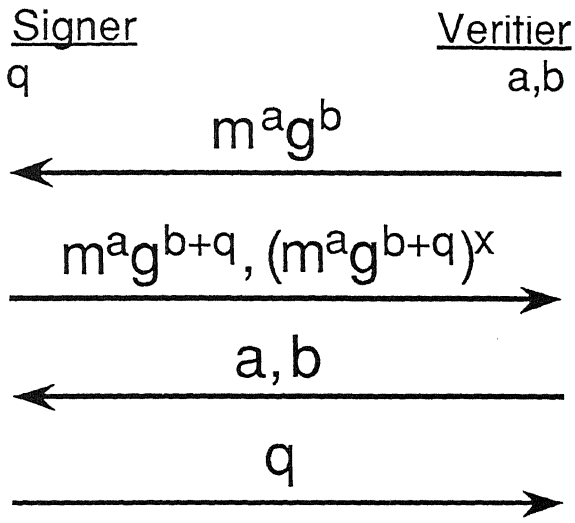

Fig. 1. Confirmation protocol

Each party should initially choose secret random group elements uniformly: S chooses q and V chooses a and b. The first message is formed by V as shown by the first arrow of Figure 1. The second message arrow shows the response of S as a pair of group elements. Next V sends a and b in message 3 so that S can reconstruct the first message. Only once this reconstruction is successful does S send message 4 to reveal q. Finally, by substituting z for m^x, V can reconstruct message 2 and ensure that it was formed properly.

Security of Confirmation

There are two essential properties:

Theorem 1: The protocol of Figure 1 is zero-knowledge [GMR].

Proof: If V sends a message 3 that should result in a message 4 being sent, V can form the message 2 determined by any random message 4. Any V not sending such a valid message 3 does not receive message 4, but can simulate the message 2 pair as g^y and g^{xy}, by choosing y as a random group element.

Theorem 2: Even with infinite computing power S cannot with probability exceeding p^{-1} provide a valid response for an invalid signature.

Proof: Essentially the same argument as that of [CA] suffices.

DISAVOWAL PROTOCOL

An alleged signer may wish to convince a verifier that a particular message z is not a valid signature corresponding to the signer's public key g^x and message m, i.e. that $z \neq m^x$. To do this, the alleged signer cooperates in an instance of a disavowal protocol. The signer can cheat with probability $1/(k+1)$, where k is a mutually agreed constant and order k operations must be performed by the signer. In practice k might be 1023, for instance, and the protocol could be conducted 2 times for a chance of cheating that is less than one in a million or 10 times to give a chance of only 2^{-100}.

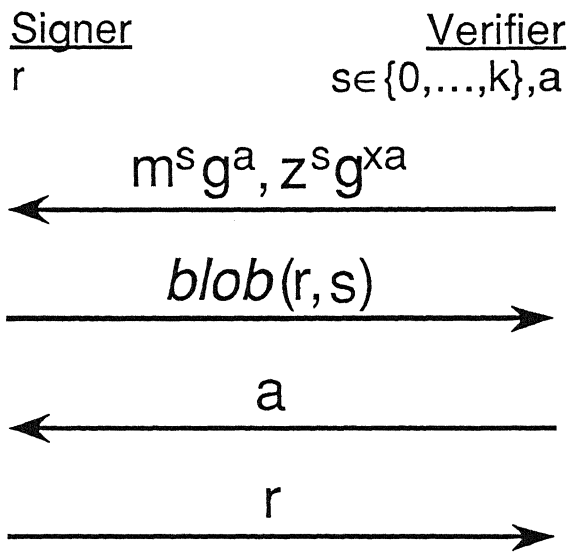

Fig. 2. Disavowal protocol

Consider a single use of the protocol of Figure 2. Initially V chooses an integer s uniformly between 0 and k and chooses a independently and uniformly over the group elements. The first arrow shows how the pair of values sent by V should be formed. Now S can determine the value of s by trial and error. An efficient approach for this raises the first component of the message to the x power and forms a quotient with the second component. The $k+1$ trial quotients can then be computed each by a single multiply from the quotient of the valid signature with z. (Since these quotients are independent of a they can be used for multiple instances of the protocol.) If no s is found, S uses a random value.

Next S sends message 2 containing a blob [BCC] committing to the value of s, but hiding s until the randomly selected r is revealed. (An attractive example is

multivalued-blobs based on the discrete log problem that protect the verifier unconditionally, as described in [BCC] §§6.6 and 6.2.2.) Upon receiving the blob as message 2, V can send a. And before finally providing r as the final message, S checks that a can be used to reconstruct the first message.

SECURITY OF DISAVOWAL

Again two things are proved:

Theorem 3: The protocol of Figure 2 is zero-knowledge.

Proof: An interaction in which V sends the correct a, which V can always recognize, is trivially simulated. Any V not supplying an acceptable a only receives a blob, and so the type of zero-knowledge depends on the type of blob.

Theorem 4: Even with infinite computing power S cannot with probability exceeding $1/(k+1)$ provide a valid response for a valid signature.

Proof: if $z = m^x$, a hides s perfectly in the first message. Since the value committed to by the blob cannot be changed, S's best strategy is to guess s.

RECENT WORK

One new result is "convertible" undeniable signatures [BCDP]. These allow the signer to make a single value public that turns all of his undeniable signatures into self-authenticating digital signatures. The signer does not lose the exclusive ability to make signatures and can even selectively convert individual signatures.

The author is aware of some work in preparation:

A signer can "distribute" his undeniable signature signing and/or disavowal abilities among a set of trustees in such a way that a majority of the trustees are necessary and sufficient to perform these functions.

By confirming signatures on random messages in advance, a signer can later simply send, such as by electronic mail, undeniable signatures that the recipient can confirm without further interaction.

The confirmation and disavowal protocols remain zero-knowledge even if multiple instances are conducted in parallel, because of the initial commitment made by the verifier. Another consequence of such "verifier commit" protocols is that it can be made infeasible for covertly cooperating verifiers to be convinced by choosing their single challenge based on coin-flips.

Blobs formed from undeniable signatures can be used to show that the signer can satisfy an agreed predicate. These proofs require only a few messages because blob opening is a parallelizeable confirmation protocol. Such proofs are

"undeniable" in the sense that anyone who trusts the randomness of the challenges can later conduct either the confirmation or disavowal protocol with the singer and be convinced whether or not the proof transcript is valid.

CONCLUSION

Undeniable signatures that are Zero-Knowledge can be achieved. They are essentially as efficient in confirmation, and nearly so in disavowal, as other known undeniable signature schemes.

ACKNOWLEDGEMENTS

It is a pleasure to thank the following people for contributing to this paper in one way or another through discussions with the author: Charles Bennett, Jurjen Bos, Joan Boyar, Gilles Brassard, Ivan Damgård, Eugène van Heyst, Tatsuaki Okamoto, and Torben Pedersen.

REFERENCES

[BCC] Brassard, G., D. Chaum, and C. Crépeau, "Minimum disclosure proofs of knowledge," *Journal of Computer and System Sciences*, vol. 37, 1988, pp. 156–189.

[BCDP] Boyar, J., D. Chaum, I. Damgård, and T. Pedersen, "Convertible undeniable signatures," to be presented at CRYPTO '90.

[CA] Chaum, D. and H. van Antwerpen, "Undeniable signatures," Advances in Cryptology—CRYPTO '89, Springer-Verlag, 1990, pp. 212–216.

[DH] Diffie, W. and M.E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, Vol. IT-22, 1976, pp. 644–654.

[GMR] Goldwasser, S., S. Micali, and C. Rackoff, "The knowledge complexity of interactive proof-systems," Proceedings, 17th Annual ACM Symposium on the Theory of Computing, May 1985," pp. 291–304.

[PH] Pohlig, S. and M.E. Hellman, "An improved algorithm for computing logarithms over GF(p) and its cryptographic significance," *IEEE Transactions on Information Theory*, vol. IT-24, 1978, pp. 106–110.