Designated Confirmer Signatures

David Chaum

DigiCash bv
Kruislaan 419
1098 VA Amsterdam
Netherlands

Abstract

This paper introduces a new kind of signature authentication and gives practical protocols that implement it. The technique can be used in ways that approach the functionality of known techniques, such as ordinary digital signatures and zero-knowledge proofs. But more importantly, it opens up a whole space of possibilities in between them.

The technique works in essence by allowing the signer to prove to the signature's recipient that designated parties can confirm the signature without the signer. But the signer is protected, since unless sufficient designated parties cooperate in confirmation, the signature is no more convincing than any other number.

1 Introduction

A zero-knowledge proof [GMR89], although convincing to the recipient, does not allow the recipient to convince anyone else. A self-authenticating digital signature [DH76], on the other extreme, not only allows the recipient to convince anyone simply by providing a copy of the signature, but also allows anyone so convinced to convince others without limitation.

Undeniable signatures occupy a particular position, somewhere in between these extremes, protecting both the interests of the signer in ensuring that the signatures are not subsequently misused by the recipient as well as those of the recipient in being able to convince others later. The recipient of an undeniable signature is convinced that anyone holding it can challenge its signer and that the signer cannot answer falsely. The reason this works is that the signer is always able to convince anyone that a valid signature is valid and that an invalid signature is invalid. Thus the recipient is at least sure that the signer cannot falsely deny a valid signature.

For the recipient, undeniable signatures do have the advantage over zero-knowledge that the recipient has something that can later, under certain circumstances, be used to convince others. But for many practical applications this protection is too weak. It relies on the signer cooperating in subsequent confirmations of the signature. If the signer should become unavailable, such as might be expected in case of default on the agreement represented by the signature, or should refuse to cooperate, then the recipient cannot make use of the signature.

The basic designated-confirmer protocol introduced here solves this weakness of undeniable signatures. It involves three parties. The recipient of the signature, Rita, is the party who needs no public key. The signer, Simon, and the confirmer, Colin, each have a public key accepted by Rita. The signing protocol consists only of interaction between Simon and Rita. It leaves Rita convinced that Simon has given her a designated-confirmer signature, for the agreed message, using Simon's private key and Colin's public key. The result is that Rita is convinced that Simon's signature on the message can be confirmed by Colin. Any subsequent confirmation protocol by Colin might, depending on how much he reveals, be zero-knowledge, designated-confirmer, or self-authenticating.

The paper first fully considers a basic system. Section 2 introduces the central concepts of the basic system. A signing protocol and two different kinds of confirming protocols are presented in the following three sections. Then Section 6 sketches some generalizations and constructions that cover a space that spans self-authenticating signatures and zero-knowledge proofs.

2 Basic System

A simple example construction approach for the basic designated-confirmer protocol is as follows. Simon gives Rita a self-authenticating digital signature on the agreed message signed with his own private key—except that the signature is incomplete in the sense that it "hinges" on the validity of a certain undeniable signature. This undeniable signature is created by Simon as if it were signed by Colin and it validly corresponds to Colin's public key. (Simon is able to create such a signature of Colin, but only on random messages, because after he chooses the signature he is free to choose any value for the message to be signed.) Simon then proves to Rita that the undeniable signature is valid.

Rita cannot prove anything about the transcript of her interaction with Simon, unless she gets help. But Colin, by virtue of his private key, can always help Rita by proving to anyone that the undeniable signature is valid, thereby convincing them of the validity of Simon's original incomplete signature. Such a proof by Colin can, of course, take a variety of forms.

The tricky part of the above construction approach is a way to make self-authenticating signatures that hinge on undeniable signatures. This has two aspects. If, on the one hand, the undeniable signature is not valid and can be chosen freely, then the self-authenticating signature should be worthless in the sense that anyone could easily have created it. If, on the other hand, the undeniable signature is valid, and someone is convinced of its validity, then they should consequently be convinced of the validity of the self-authenticating signature.

Both these properties can be accomplished with self-authenticating signature schemes that rely on one-way functions. One example type of signature is where the output of the one-way function is used to determine what would otherwise be the challenge of a zero-knowledge protocol [FS87]. Such a signature scheme is modified slightly so that the definition of the one-way function includes the undeniable signature in a suitable

way. The output of the new one-way function could, for instance, be defined as the output of the original function bitwise exclusive-OR'ed with the undeniable signature. [1]

Thus complete freedom of choice of what should be an undeniable signature allows complete freedom of choice of the output of the new one-way function, but limited choice of the undeniable signature means constraints on the output of the new one-way function.

3 Signing protocol

The purpose of this protocol is for Simon to sign a message and to convince Rita that the signature is in fact valid. For simplicity, Simon will use an RSA signature scheme with public key modulus n and exponent 3. Colin's public key will be $h := g^z$ where z is Colin's private key. This public key and all the computations in the protocols (unless otherwise noted) are in a group of prime order where discrete logs are assumed hard. The signing protocol is shown in figure 1. It consists of the following steps:

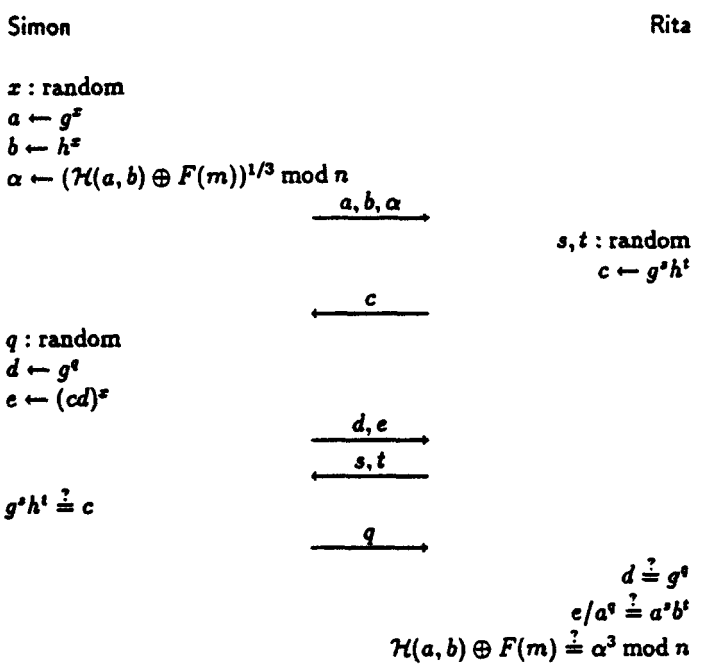

Figure 1: Signing a message

[1] Whatever *combining function* is used to achieve the combining, such as group operations or DES with fixed keys, it should be feasible to invert outputs of the new function to images of the original one-way function, given the ability to freely choose the undeniable signature inputs. If only the undeniable signature itself were included, Colin could forge Simon's signatures because he can control it completely; if the message is also included, Colin cannot forge the signatures.

1. Simon chooses a random x and computes $a = g^x$ and $b = h^x$. He computes the RSA-signature on $\mathcal{H}(a, b) \oplus F(m)$ where F is a suitable hash function and $\mathcal{H}$ is the combining function which destroys the multiplicative structure but which is easilly invertable. (An example could be a substitution-permutation network where the substitutions are DES encryptions with publicly known keys.) Finally he sends a, b and α to Rita.

2. Rita chooses a random s and t and forms $c = g^s h^t$ which she sends to Simon.

3. Simon chooses a random q and forms $d = g^q$. He multiplies c by d before raising it to the power x to get e. He sends d and e to Rita. This is related to the

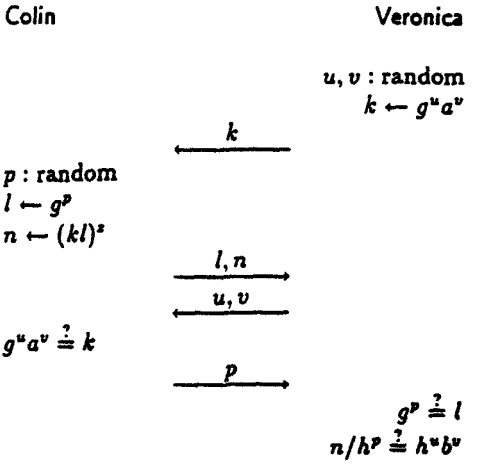

Figure 2: Confirmation protocol

confirmation protocol of [Cha91]. Simon will only reveal q when Rita shows that c was correctly formed.

4. Rita sends s and t to Simon who verifies that c was indeed formed correctly.

5. Simon sends q to Rita.

6. Rita verifies that q is correct, and checks that $a^s b^t = e/a^q$. This convinces Rita that b is equal to a^x, but does not leave her with a way to prove it to anyone else.

4 Confirmation protocol

This section shows how Colin can confirm the signature Simon created. This protocol leaves the verifier Veronica convinced that the signature is correct, but like figure 1 does not allow her to convince anyone else. It is shown in figure 2.

The principle behind this protocol is similar to that of figure 1. Colin convinces Veronica that b is equal to a^x, without giving her any transferable proof.

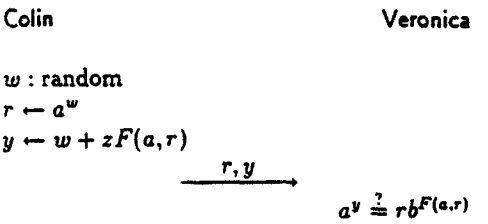

Figure 3: Conversion protocol

5 Conversion protocol

There is also a way in which Colin can convert the designated-confirmer signature into a self-authenticating digital signature. This is shown in figure 3. Here Colin forms a non-interactive proof that someone knows how to express b as a power of a.

The basic idea of the conversion is that only someone who knows how to express b as a power of a can form a pair (r, y) such that $a^y = r \cdot b^{F(a,r)}$ where F is a suitable oneway function. It is interesting that h doesn't appear here, so the public key h of Colin is no longer associated with the now self-authenticating signature.

6 Generalizations

The basic signature scheme can be generalized by including multiple confirmers. More than one confirmer's public key could be combined in the undeniable signature (such as by taking the product of public keys), so that the cooperation of all the confirmers would be needed for any confirmation. The more confirmers required, the harder it would be to get confirmation, and, in some intuitive sense, the closer the signature scheme approaches a zero-knowledge protocol. And if Simon's key is included, then the result is minimum disclosure [BCC88].

Multiple designated confirmer signatures could give the effect that selected subsets of a set of participants could be required. (More efficient ways to achieve threshold functions are being studied.) Another extreme case would be if a single message were signed separately for each participant's public key. This approaches the effect of self-authenticating signatures.

7 Summary and Conclusions

The designated confirmer signatures have practical protocols and offer a rich structure of intriguing possibilities for signature authentication.

Plenty of work remains, however. A more rigorous treatment of the subject, more efficient constructions, and constructions based on other assumptions would all be of interest. Also, efforts to develop actual uses are ultimately needed.

8 Acknowledgments

To be supplied in final paper.

References

[BCC88] Gilles Brassard, David Chaum, and Claude Crépeau. Minimum disclosure proofs of knowledge. *Journal of Computer and System sciences*, 37:156–189, 1988.

[Cha91] David Chaum. Zero-knowledge undeniable signatures. In I.B. Damgård, editor, *Advances in Cryptology—EUROCRYPT '90*, pages 458–464, Springer-Verlag, 1991.

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, November 1976.

[FS87] Amos Fiat and Adi Shamir. How to prove yourself: practical solutions to identification and signature problems. In A.M. Odlyzko, editor, *Advances in Cryptology—CRYPTO '86*, pages 186–194, Springer-Verlag, 1987.

[GMR89] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof-systems. *SIAM Journal of Computation*, 18(1):186–208, 1989.