Arbitration in Tamper Proof Systems

If DES ≈ RSA Then What's the Difference Between True Signature and Arbitrated Signature Schemes ?

George I. Davida Brian J. Matt
Electrical Engineering and Computer Science Department
University of Wisconsin-Milwaukee
Milwaukee, WI 53201

Abstract

Given that tamperfree devices exist it is possible to construct true signature schemes that have the advantages of arbitrated signature schemes, protection against disavowing or forging messages, and lacking certain short commings. Other cryptographic protocols can also be improved. The contents of tamperfree devices cannot be examined as well as not modified.

1 Introduction

Digital signature systems not employing arbitrators are vulnerable to forging by back dating messages if secret keys are lost or stolen. One would like to avoid arbitrators, even blind arbitrators if possible. Other protocol schemes, [7] have similar difficulties. By employing "tamperfree" systems we will show that such difficulties can be avoided.

2 Digital Signatures

In true signature schemes the sender's signed messages are sent directly to the receiver. The receiver checks the validity and authenticity of the message upon receipt. The role of third parties is that of storing secret information until a dispute arises, ie *trusted third parties* or providing a *public directory*. Examples of such schemes include [9,10,11]. In arbitrated schemes "all signed messages are transmitted from S (the sender) to R (the receiver) via an arbitrator A[1], who serves as a witness."[1] For any message the arbitrator can determine the validity and authenticity of the message the sender provides and allows the receiver to be certain the message that he receives has been examined by the arbitrator.

To be an effective witness the arbitrator must:

1. Prevent a message sender from disavowing messages by leaking his secret key.
2. If the secret key is truly stolen, prevent someone from forging messages.

The system should ideally have a low operational overhead in terms of transmission costs, and computation. Both prior and especially concurrent costs should be minimized. Centralized arbitrators have to contend with message congestion and non-centralized arbitrated systems are more complex and expensive. Physical security and trustworthiness of the arbitrator are a concern especially if multiple arbitrators are necessary. This can be dealt with by employing blind arbitrators[8].

[1] It is not required that the message go directly to A. It might be sent to B first who then sends it to A.

C. Pomerance (Ed.): Advances in Cryptology - CRYPTO '87, LNCS 293, pp. 216-222, 1988.

It is issues such as these that we wish to address. First we describe a public key system based on conventional key systems due to Desmedt and Quisquater[5] then our signature system utilizing similar assumptions.

3 Tamperfree Public Key System

In "Public Key Systems Based on the Difficulty of Tampering"[5] Desmedt and Quisquater present public key cryptosystems based on conventional cryptosystems in a "tamperfree" environment. They state the following assumptions:

- Hard conventional cryptosystems exist
- It is feasible to make tamperfree devices.[5]

These "tamperfree" devices cannot be examined (to determine key values) or altered. As an example the following system was presented.

Let:	S	supersecret key
	sk	secret key (of the user)
	PK	public key (of the user)
	G	Generation algorithm for the public key
	E	Message encryption algorithm
	D	Message decryption algorithm
	E'&D'	Encryption and Decryption algorithm (1st system)
	E''&D''	Encryption and Decryption algorithm (2nd system)
	P	plain text
	C	cipher text

To generate a public key user A does the following:

$$G : E''_S(sk_A) \rightarrow PK_A$$

If user A wants to send a message to user B:

$$E : E'_{D''_S(PK_B)}(P) \rightarrow C$$

and user B decrypts by:

$$D : D'_{sk_B}(C) \rightarrow P$$

As a result of the tamperfree nature of the device user A can employ user B's public key PK_B without learning user B's secret key sk_B.

4 The Notarized System

We assume:

- The existence of a conventional cryptosystem either lacking weak keys[2] or with the existence of means of avoiding using the weak keys.
- The cryptosystems are secure, i.e. able to withstand cryptoanalytic methods.

[2] A term introduced by Davies pertaining to the weak keys of DES.[4]

- That an adequate public directory system exists.
- The existence of a cryptographic hashing function.

Each device is "tamperfree" hence the device registers cannot be examined externally. Each device contains:

1. Two key registers
2. Message counter
3. Clock
4. Encryption/decryption device
5. Cryptographic hashing function (possibly employing the encryption/decryption device)
6. An initialization flag register.

The flag register, supersecret and device secret key registers, message counter and clock employ erasable, non-volatile memory.

A device resides in one of four states, never used (pre startup), active, non-active (used and power failed) and permanently disabled (post self-destruct). Upon startup the super secret key S is set. This is the key that all devices must share for communication to be possible. The device specific secret key skd is set. The device generates a device specific public key PKd that is installed in a public directory. It is by employing this key that the device notarizes/authenticates messages. The clock is set and the message counter is initialized, probably to zero. The initialization flag is set. Once the flag is set the initialization flag itself, the message counter, and the secret keys cannot be updated. Also modification of the clock value is now restricted.

The device is now in the active state and available for cryptographic work. Keys entered during the active state are all stored in erasable, volatile memory. Should the device suffer a power failure it moves in to the non-active state.

To restore a device to the active state the clock must be reset. The time can be reset only to a value later than the time at power failure. The difference cannot be larger than some finite value or the device self destructs. The reset can only be performed by authorized personnel, this is assured by requiring that secret data be entered by the authorized person as part of the time reset process.

While in the active state each device can perform three functions, user public key generation, message encryption, and message decryption.

Let:

S	supersecret key
sku	secret key (of the user)
PKu	public key (of the user)
skd	secret key (of the device)
PKd	public key (of the device)
G	Generation algorithm for the public key
E	Message encryption algorithm
D	Message decryption algorithm
$E\&D$	Encryption and Decryption algorithm
P	plain text
C	cipher text
CHF	Cryptographic Hashing Function

To generate a user public key the user A enters her secret key sku_A and selects the key generation function:

$$G : E_S(sku_A) \rightarrow PKu_A \tag{1}$$

This algorithm is also employed, using a skd_α instead of a sku_A to generate the public key of device α:

$$G : E_S(skd_\alpha) \rightarrow PKd_\alpha \tag{2}$$

For user A to encrypt a message using device α and send it to user B user A's secret key and the public key of user B are entered into the device. The message counter mc and time stamp ts are concatenated to the message and a cryptographic hash ch is made of the result. The cryptographic hashing function hashes using the user's secret key sku_A and then the device's secret key skd_α:

$$CHF_{skd_\alpha}(CHF_{sku_A}(P//mc//ts)) \rightarrow ch \tag{3}$$

This cryptographic hash provides both user A's signature and device α's notarization.

The crptographic hash is concatenated with the the message, message count and time stamp and encrypted using user B's public key.

$$E : E_{D_S(PKu_B)}(ch//P//mc//ts) \rightarrow C \tag{4}$$

The text $ch//P//mc//ts$ can be retained by user A as proof that he generated the message at the specified time.

When user B decrypts the message he enters his user secret key sku_B plus the public key of user A and device α. First the cipher text is decrypted:

$$D : D_{sku_B}(C) \rightarrow ch//P//mc//ts \tag{5}$$

Second the cryptographic hash of $P//mc//ts$ is computed and compared with ch.

$$CHF_{D_S(PKd_\alpha)}(CHF_{D_S(PKu_A)}(P//mc//ts)) \rightarrow ch' \tag{6}$$

If ch' does not match ch the message is rejected. See figure 1.[3]

5 Remarks

The scheme does not need a *Trusted third party* as do true signature schemes employing conventional cryptosystems. No separate arbitrator is employed since the devices serve as witnesses. Unlike public key based true signature schemes, the user can *properly* disavow messages by promptly informing a judge or referee of the loss. This is since the device is "tamperfree", messages cannot be forged since no one has access to a device's secret key skd and the supersecret key S, and devices cannot be coerced into back dating messages.

[3] If it is required that user B use device β to decrypt the message then user A encrypts by:

$$E : E_{D_S(PKd_\beta)}(E_{D_S(PKu_B)}(ch//P//mc//ts)) \rightarrow C \tag{7}$$

and user B decrypts by:

$$D : D_{sku_B}(D_{skd_\beta}(C)) \rightarrow ch//P//mc//ts \tag{8}$$

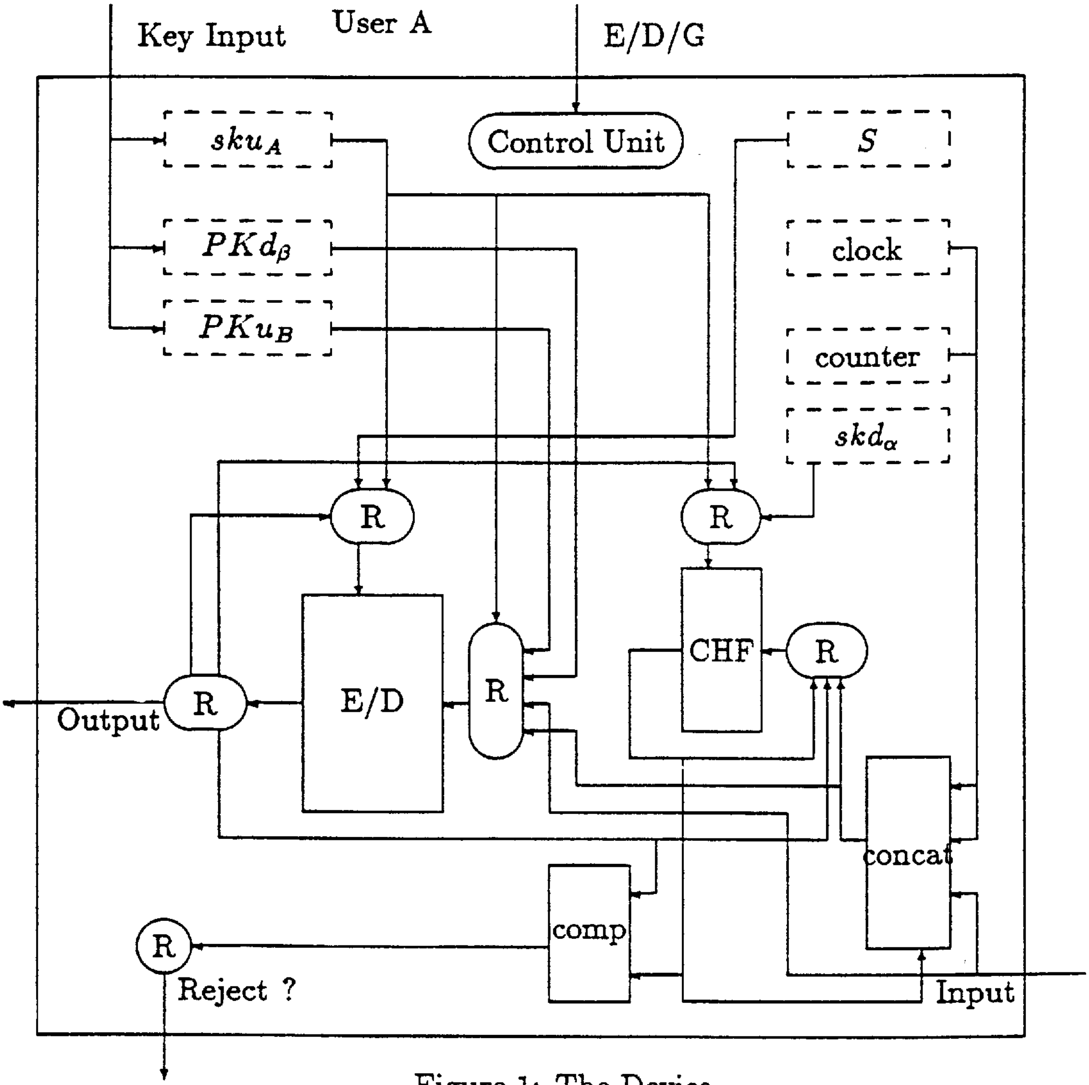

Figure 1: The Device

6 Selectively Breakable Cryptosystems

In some situations, relatively unsophisticated users engage in transactions where unaided key generation is difficult for one or both users and mutual suspicion is high. The following system was proposed:[7]

- There exists unordered pairs (A, B) of users
- There exists a *Trusted Third Party*
- The *Trusted Third Party* generates for each user pair the following:
  - A pair (R_a, R_b) of keys called the *Retrieval Keys*.
  - A pair (M_a, M_b) of keys called the *Message Keys*.

The keys have the following properties

$$M_a : T \rightarrow C_a(T) \tag{9}$$
$$M_b : T \rightarrow C_b(T) \tag{10}$$
$$M_b : C_a(T) \rightarrow T \tag{11}$$
$$M_a : C_b(T) \rightarrow T \tag{12}$$

where T is the plain text message and C is the ciphertext.

1. M_a cannot be derived from M_b nor can M_b be derived from M_a.[4]
2. It is computationally infeasible to derive M_b or R_b from M_a and R_a and vice versa.
3. It is feasible to compute M_a and M_b from R_a and R_b.

Naturally M_a and M_b should be a good cryptosystem. In fact they should have the properties of a public key cryptosystem[6] with the addition that it should be computationally infeasible to derive either M_a or M_b from the other.

Both parties receive their message keys from the *Trusted Third Party* in such a manner that they cannot disavow the reception. The third party stores the retrieval keys in separate secure areas (physically secure and or cryptographically secure)[5] guarantees the proper delivery of the message keys. To a cryptoanalyst not in collusion with either of the parties the system appears as a conventional cryptosystem. To both A and B it appears as a public key cryptosystem with each holding the "public key".

One problem with the above scheme is that A or B could have their key stolen or claim it was stolen. Resulting as in the case of digital signatures in forged or disavowed messages. Consider a situation where the *Trusted Third party* and both A and B have devices similar to those in section number 4. Each device shares the same supersecret key and the device public key for user A's device and B's device is known to the *Trusted Third Party*. The *Trusted Third Party* merely generates a secret key and sends it to both user A and B. Both users use this key as a secret user key for communications with the other party. This communication is through each user's device and the messages are stamped as in section 4. This solution does not require any additional hardware for user's A and B.

[4] Actually this follows from condition two.

[5] If the *Trusted Third Party* cannot be trusted so far a key sharing scheme can be employed. [2,12,3]

7 Acknowledgement

The authors wish to Professor Yvo Desmedt for several helpful suggestions.

References

[1] Selim G. Akl. Digital signatures: a tutorial survey. *IEEE Computer*, 16(2):15–24, 1983.

[2] G. R. Blakley. Safeguarding cryptographic keys. In *Proc. AFIPS 1979 NCC*, pages 313–317, 1979.

[3] George I. Davida, Richard A. DeMillo, and Richard J. Lipton. Protecting shared cryptographic keys. In *Proc. 1980 Symp. on Security and Privacy*, pages 100–102, IEEE, April 1980.

[4] Donald W. Davies. Some regular properties of the 'data encryption standard'. In *Advances in Cryptology: Proceedings of Crypto82*, pages 89–96, 1983. Actually presented at Crypto-81.

[5] Yvo Desmedt and Jean-Jacques Quisquater. Public key system based on the difficulty of tampering (is there a diffference between des and rsa?). In *CRYPTO'86 Abstracts and Papers*, pages 15-1 - 15-3, 1986.

[6] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Trans. on Inform. Theory*, 22(6):644–654, 1976.

[7] Brian J. Matt. *Selectively Breakable Cryptosystems and Personal Keys*. Master's thesis, University of Wisconsin Milwaukee, May 1981.

[8] Henk Meijer and Selim Akl. Digital signature schemes. *Cryptologia*, 6(4):329–338, October 1982.

[9] R. C. Merkle. Protocols for public key cryptosystems. In *Proc. 1980 Symp. on Security and Privacy*, pages 122–134, IEEE, April 1980.

[10] M. Rabin. Digitalized signatures. In R. A. DeMillo et al., editors, *Foundations of Secure Computation*, pages 155–166, 1978.

[11] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptograms. *Communications of the ACM*, 21(2), Feburary 1978.

[12] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.