

Verifiable Encryption of Digital Signatures and Applications

GIUSEPPE ATENIESE

The Johns Hopkins University

This paper presents a new simple schemes for verifiable encryption of digital signatures. We make use of a trusted third party (TTP) but in an *optimistic* sense, that is, the TTP takes part in the protocol only if one user cheats or simply crashes. Our schemes can be used as primitives to build efficient fair exchange and certified e-mail protocols.

Categories and Subject Descriptors: H.4.0 [Information Systems Applications]: General

Additional Key Words and Phrases: Certified e-mail, contract signing, digital signatures, fair exchange, public-key cryptography, proof of knowledge

1. INTRODUCTION

Whenever a message is sent over the Internet, there is no assurance that it will be delivered to the intended recipient. Even if the message has been delivered, the recipient may claim otherwise. This may be unpleasant particularly in today's society where networked computers are increasingly being used to exchange items between distrusted parties.

In the real world, some form of simultaneity can be achieved thanks to the physical proximity of the parties involved with an exchange. For instance, two parties can sign a contract *simultaneously* by holding the contract itself: One party will continue to hold the contract until the other party has signed. Similarly, when we buy an item from a store, the merchant could hold the item until we have provided the right amount of cash.

Unfortunately, physical proximity cannot be exploited in the digital world and exchanging items over the Internet is considered a difficult problem, called the *fair exchange problem*.

There have been several approaches to solve the fair exchange problem which are based on different definitions of fairness. Fairness is interpreted as *equal computational effort* in Even et al. [1985]. That is, it is assumed

Parts of this paper appeared in the 6st ACM Conference on Computer and Communication Security [Ateniese 1999] and in RSA 2002 Conference [Ateniese and Nita-Rotaru 2002]. In particular, verifiable encryption schemes appeared in [Ateniese 1999] and the certified email protocol appeared in [Ateniese and Nita-Rotaru 2002] (coauthored with Cristina Nita-Rotaru).

Author's address: Department of Computer Science, The John Hopkins University, 3400 North Charles Street, Baltimore, MD 21218; e-mail: ateniese@cs.jhu.edu.

Permission to make digital/hard copy of all or part of this material without fee for personal or classroom use provided that the copies are not made or distributed for profit or commercial advantage, the ACM copyright/server notice, the title of the publication, and its date appear, and notice is given that copying is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission and/or a fee.

© 2004 ACM 1094-9224/04/0200-0001 \$5.00

that two parties, Alice and Bob, have equal computational power and they exchange their items bit-by-bit and by taking turns. This approach does not require the intervention of a trusted third party but it involves many rounds of interactions.

A probabilistic approach is adopted in Ben-Or et al. [1990], that is, the probability of successfully completing the protocol is gradually increased after every round of interaction.

Asokan et al. [1997] introduce the *optimistic* approach. It relies on the existence of a trusted third party which is invoked only in case of an exception. As long as the two parties follow the exchange protocol, there is no need for the trusted party's intervention, but if one deviates from the protocol then the trusted party can easily restore fairness. This approach results in particularly efficient fair exchange protocols for generic items [Asokan et al. 1997, 1998a].

Asokan et al. [1998b] and Bao et al. [1998] show how to build fair exchange protocols by means of *verifiable encryption* of digital signatures (i.e., a way of encrypting a signature under a designated public key and subsequently prove that the resulting ciphertext indeed contains such a signature). Camenisch and Damgård [2000] generalize the schemes in Asokan et al. [1998b] so to achieve more efficient schemes that can be proved secure without relying on random oracles.

Solutions to the fair exchange problem can be employed to solve a related problem: The certified e-mail problem. In this case, an email message is to be exchanged with a receipt, that is, an incontestable proof that the email message was delivered to the intended recipient. In the real world, a letter can be sent through the postal service along with a request for a receipt. However, if the content of the letter is to be kept confidential from the postal service provider then the receipt can only state that an "envelope" was delivered and nothing can be certified about the content of the envelope. In the digital world, instead, a receipt can be built so that it can be shown as evidence that the actual message (and not just an "envelope") was delivered to the intended recipient and this can be done while assuring that the message is known only to the two parties involved, the sender and the recipient.

Several protocols provide valid solutions to the certified-email problem (see for instance [Asokan et al. 1998b; Ateniese et al. 2001; Bahreman and Tygar 1994; Bao et al. 1998; Deng et al. 1996; Micali 1997; Riordan and Schneier 1998; Zhou and Gollmann 1996]). They can be categorized based on the role of a trusted third party (TTP), which is responsible for restoring the protocol fairness. In particular, the protocols in Asokan et al. [1998b] and Micali [1997] provide very innovative solutions by allowing the trusted party to be offline and be invoked only in exceptional circumstances.

In this paper, we emphasize that the certified e-mail problem is, in several aspects, different from other similar problems, such as fair exchange or contract signing. Our intent is to relax some assumptions so to find valid solutions with improved efficiency. We also introduce new protocols for verifiable encryption of digital signatures that can be employed as primitives when designing a large class of protocols such as certified email, fair exchange, and contract signing.

2. CRYPTOGRAPHIC TOOLS

In this section, we present signature schemes that allow a prover to convince a verifier of the equality of discrete logarithms (even when working in different groups). In short, the problem is, given g_1^x, g_2^x , and a message m , generating a signature on m and, at the same time, showing that $\text{Dlog}_{g_1} g_1^x = \text{Dlog}_{g_2} g_2^x$ without revealing any information other than g_1^x and g_2^x .

We will denote an instance of this signature technique by $EQ_DLOG(m; g_1^x, g_2^x; g_1, g_2)$.

We make use of so-called “proof-of-knowledge” systems that allow demonstrating knowledge of a secret such that no useful information is revealed in the process. Namely, we define Schnorr-like signature schemes [Schnoor 1991] in order to show knowledge of relations among secrets. Substantially, these are signature schemes based on proofs of knowledge performed noninteractively making use of an ideal hash function $\mathcal{H}(\cdot)$ (à la Fiat and Shamir [1987]).

Let G_q denote the unique subgroup of $\mathbb{Z}_p^*$ of order q . The parameters p, q are primes such that q divides $p - 1$, for instance $p = 2q + 1$.

Let $g, h \in G_q$ be publicly known bases. The prover selects a secret $x \bmod q$ and computes $y_1 = g^x$ and $y_2 = h^x$. The prover must convince the verifier that:

$$\text{Dlog}_g y_1 = \text{Dlog}_h y_2.$$

The protocol, described by Chaum and Pedersen [1992], is as follows:

- (1) The prover randomly chooses $t \in \mathbb{Z}_q$ and sends $(a, b) = (g^t, h^t)$ to the verifier.
- (2) The verifier chooses a random challenge $c \in \mathbb{Z}_q$ and sends it to the prover.
- (3) The prover, then, sends $s = t - cx \bmod q$ to the verifier.
- (4) The verifier accepts the proof if:

$$a = g^s y_1^c \quad \text{and} \quad b = h^s y_2^c.$$

To turn the protocol above into a signature on an arbitrary message m , the signer can compute the pair (c, s) as:

$$c = \mathcal{H}(m \| y_1 \| y_2 \| g \| h \| g^t \| h^t), \quad s = t - cx,$$

where $\mathcal{H}(\cdot)$ is a suitable hash function. To verify the signature (c, s) on m , it is sufficient to check whether $c' = c$, where

$$c' = \mathcal{H}(m \| y_1 \| y_2 \| g \| h \| g^s y_1^c \| h^s y_2^c).$$

This signature scheme works properly also into an accurately chosen subgroup of $\mathbb{Z}_n^*$, where n is an RSA-like composite. In particular, in this paper, we work into the subgroup of all quadratic residues modulo n , denoted by Q_n . Explicitly, $Q_n \subset \mathbb{Z}_n^*$ is the set of elements $a \in \mathbb{Z}_n^*$ such that there exists an $x \in \mathbb{Z}_n^*$ with $x^2 \equiv a \bmod n$. We select n as product of two *safe* primes p and q , that is, such that $p = 2p' + 1$ and $q = 2q' + 1$ with p', q' primes. Thus, notice that Q_n is a cyclic group of order $p'q'$.

2.1 Equality of Discrete Logarithms in a Group of Unknown Order

The signature scheme above works properly even when the signer is working over a cyclic subgroup of $\mathbb{Z}_n^*$, $G = \langle g \rangle$, whose order $\#G = p'q'$ is unknown, but its bit-length ℓ_G (i.e., the integer ℓ_G s.t. $2^{\ell_G-1} \leq \#G < 2^{\ell_G}$) is publicly known.

We, now, show how the signer can generate a signature on a message $m \in \{0, 1\}^*$ working with elements in G and, at the same time, showing knowledge of the discrete logarithm with respect to bases g and h satisfying $y_1 = g^x$ and $y_2 = h^x$. We make use of a hash function $\mathcal{H} : \{0, 1\}^* \rightarrow \{0, 1\}^k$, which maps a binary string of arbitrary length to a k -bit hash value. We also assume a security parameter $\epsilon > 1$. The signer computes a pair $(c, s) \in \{0, 1\}^k \times \pm\{0, 1\}^{\epsilon(\ell_G+k)+1}$ such that $c = \mathcal{H}(m \| y_1 \| y_2 \| g \| h \| g^s y_1^c \| h^s y_2^c)$.¹ This shows that the discrete logarithms of $y_1 = g^x$, with regard to base g , and $y_2 = h^x$, with respect to base h , are equal.

The signer, in possession of the secret x , is able to compute the signature (c, s) , provided that $x = \text{D log}_g y_1 = \text{D log}_h y_2$, by choosing a random $t \in \pm\{0, 1\}^{\epsilon(\ell_G+k)}$ and then computing c and s as

$$c = \mathcal{H}(m \| y_1 \| y_2 \| g \| h \| g^t \| h^t), \quad s = t - cx \quad (\text{in } \mathbb{Z}).$$

2.2 Equality of Discrete Logarithms from Different Groups

Suppose now that g and h have different orders, q_1 and q_2 , respectively. Thus, given two elements $y_1 = g^x$ and $y_2 = h^x$ of *different* groups $G_1 = \langle g \rangle$, $G_2 = \langle h \rangle$, the verifier can only conclude that the signer knows a value x such that $x \bmod q_1 = \text{D log}_g y_1$ and $x \bmod q_2 = \text{D log}_h y_2$. However, it is possible to prove that a secret x lies in a specific interval, more precisely given g^x with $-2^\ell < x < 2^\ell$ for an integer ℓ , it is possible to prove that x lies in the extended interval $]-2^{\epsilon(\ell+k)}, 2^{\epsilon(\ell+k)}[$. Hence, we can build a signature scheme for showing that $\text{D log}_g y_1 = \text{D log}_h y_2$ in $\mathbb{Z}$ by combining the scheme for showing knowledge of a value x with $x \bmod q_1 = \text{D log}_g y_1$ and $x \bmod q_2 = \text{D log}_h y_2$, and the scheme for showing that $-2^{\epsilon(\ell+k)} < x < 2^{\epsilon(\ell+k)}$. Clearly, this can be done only if the length ℓ can be chosen such that $2^{\epsilon(\ell+k)+1} < \min\{q_1, q_2\}$, where q_1, q_2 are the orders of g and h , respectively. Notice that this proof only works if one of G_1 or G_2 is an auxiliary group that is not fixed.

This idea is formalized by Camenisch and Michels [1999b]. They present a concrete protocol for proving equality of discrete logarithms from different groups based on a technique developed by Fujisaki and Okamoto [1997].

To provide a viable example of how it is possible to show that x lies in the extended interval $]-2^{\epsilon(\ell+k)}, 2^{\epsilon(\ell+k)}[$, we present a signature scheme derived from a protocol due to Chan et al. [1998], and Camenisch and Michels [1998a]. The scheme can be extended to the more general interval $]X - 2^{\epsilon(\ell+k)}, X + 2^{\epsilon(\ell+k)}[$, for a given integer X , as shown in Camenisch and Michels [1998a]. The signature on a message $m \in \{0, 1\}^*$, is the pair $(c, s) \in \{0, 1\}^k \times \pm\{0, 1\}^{\epsilon(\ell+k)+1}$ such that $c = \mathcal{H}(m \| y \| g \| g^s y^c)$. This shows knowledge of the discrete logarithm of $y = g^x$ with respect to base g and that this logarithm lies in $[-2^{\epsilon(\ell+k)}, 2^{\epsilon(\ell+k)}]$.

¹The (abused) notation $s \in \pm\{0, 1\}^{\epsilon(\ell_G+k)+1}$ denotes $|s| < 2^{\epsilon(\ell_G+k)+1}$.

To produce (c, s) , the signer in possession of the secret $x = D \log_g y \in]-2^\ell, 2^\ell[$ chooses a random $t \in \pm\{0, 1\}^{\epsilon(\ell+k)}$ and then computes c and s as

$$c = \mathcal{H}(m \| y \| g \| g^t), \quad s = t - cx \quad (\text{in } \mathbb{Z}).$$

The underlying interactive protocol is proved to be a proof of knowledge (statistical honest-verifier zero knowledge), under the strong RSA assumption, in Camenisch and Michels [1998b].

3. EFFICIENT VERIFIABLE ENCRYPTIONS

Suppose that Alice and Bob have agreed on a common message m . Alice generates her signature on m , $S_A(m)$, and sends it to Bob “encrypted” under the public key of a TTP, $\mathcal{T}$.

The problem, now, is that Alice must prove to Bob that the signature is valid, even though it is encrypted, and that $\mathcal{T}$ is able to extract $S_A(m)$ from the encryption that she has sent.

In general, given an instance s of a digital signature scheme on an arbitrary message, we make a $\mathcal{T}$ -verifiable encryption $c(s)$ of s if such an encryption can be verified to contain s in a way that ensures that $\mathcal{T}$ is allowed to recover s from $c(s)$ while no useful information is revealed to others about s itself.

In most of our protocols, the encryption function $c(\cdot)$ will be implemented via the ElGamal encryption scheme. That is, given a secret key x and a corresponding public key g^x , a message m is encrypted by generating a random r and computing $K_1 = mg^{xr}$, $K_2 = g^r$. To get m from K_1 , it is sufficient to compute $m = K_1 / (K_2)^x$.

3.1 RSA Signatures

Let $n = pq$, with $p = 2p' + 1$ and $q = 2q' + 1$, where p', q' are primes. Let (e, n) be Alice’s public key with prime $e > 2$ and d the corresponding secret key, that is, $ed \equiv 1 \pmod{2p'q'}$. To sign a message m , it is sufficient to compute $C = \mathcal{H}(m)^d \pmod{n}$ where $\mathcal{H}(\cdot)$ is a hash function defined as $\mathcal{H} : \{0, 1\}^* \rightarrow \mathbb{Z}_n$ [Rivest et al. 1978].² The signature is accepted only if $C^e \pmod{n}$ matches $\mathcal{H}(m)$. In order to make efficient the verifiable encryption of RSA signatures, we will make use of an *initialization phase* by which the user and the TTP $\mathcal{T}$ agree on common parameters.

Let Q_n be the subgroup of squares in $\mathbb{Z}_n^*$. During the initialization phase, Alice sends (e, n) to $\mathcal{T}$ (along with a certificate $CERT_A$). $\mathcal{T}$ verifies that (e, n) is the public key of Alice and randomly selects a $\bar{g} \in Q_n$. Then, $\mathcal{T}$ sets $g = \bar{g}^2 \pmod{n}$, signs and sends back $(g \pmod{n}, y = g^x \pmod{n})$, where x is a secret random element.³ The details are shown in Figure 1, where all the operations are taken

²For the sake of simplicity, we employ the hash-and-sign paradigm but, in practice, $\mathcal{H}(\cdot)$ is rather a redundancy function as defined in PKCS#1, ISO/IEC 9796 and so on (see Menezes et al. 1996a, p. 442).

³By definition $g \in Q_n$ and with overwhelming probability the order of g is $p'q'$ (nevertheless paranoids may test whether $\gcd(g \pm 1, n) = 1$).

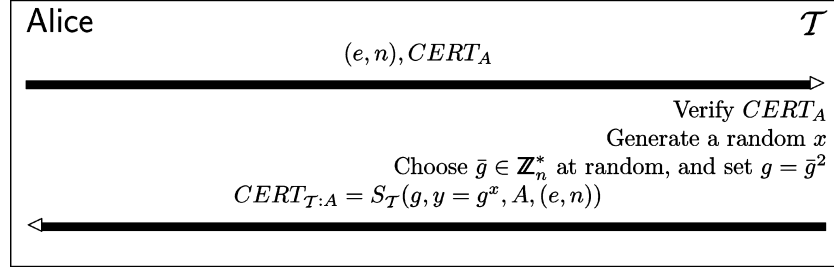


Fig. 1. Initialization phase.

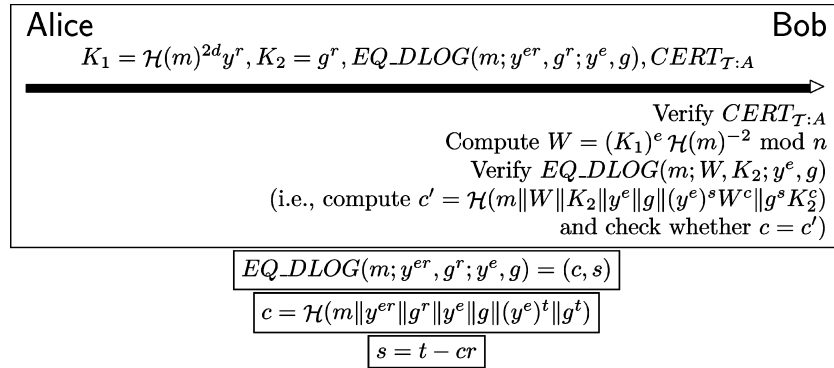


Fig. 2. Verifiable encryption of an RSA signature.

modulo n and “ A ” is a string of data identifying Alice. This phase is done only once and $\mathcal{T}$ does not need to know the factors of n .⁴

Given the message m , Alice computes $\mathcal{H}(m)^2$ and then signs it by computing $\mathcal{H}(m)^{2d} \bmod n$. Then, Alice encrypts the RSA signature via the ElGamal encryption scheme with the public key $y = g^x$, that is, by selecting a random r and computing $K_1 = \mathcal{H}(m)^{2d} y^r$ and $K_2 = g^r$. To prove that she has correctly generated a $\mathcal{T}$ -verifiable encryption, Alice releases an evidence showing that $D \log_{y^e}(y^{er}) = D \log_g(g^r)$ (via $EQ_DLOG(\cdot)$). The resulting message has four components as shown in Figure 2.

After receiving the message from Alice, Bob must verify that it is a $\mathcal{T}$ -verifiable encryption of Alice’s signature on m . Hence, he computes $\mathcal{H}(m)^2$ and then verifies $EQ_DLOG(m; K_1^e / \mathcal{H}(m)^2, K_2; y^e, g)$, where the bases $y = g^x$ and g are taken from $CERT_{T:A}$. Since n is a product of *safe* primes, g , g^x , and g^{xe} generate the same group, Q_n , with overwhelming probability. If Bob needs to know that he is working with quadratic residues modulo n (as required by a proof of soundness), he can just square all the parameters (even the bases) before performing any modular operations with them.

⁴However, we assume that Alice provides a proof of n being a product of safe primes during either the public-key certification process (executed with a Certification Authority) or the initialization phase (with $\mathcal{T}$ itself) (see Camenisch and Michels 1999a).

The hash on the message is squared by Alice in order to achieve semantic security⁵ (notice that d is odd since e is prime). If the message space is the same as the group generated by g , that is, $Q_n = \langle g \rangle$, then it is well-known that DDH⁶ implies the semantic security of the ElGamal encryption scheme modulo a composite. Notice that the mapping into Q_n does not affect the recovery algorithm: In case of dispute, $\mathcal{T}$ may get $\mathcal{H}(m)^d$ from $\alpha = \mathcal{H}(m)^{2d}$ by simply using the Euclidean algorithm. Namely, since $\alpha^e = \mathcal{H}(m)^2$ and $\gcd(e, 2) = 1$, there exist two integers a_1 and a_2 such that $\mathcal{H}(m)^d = \alpha^{a_1} \mathcal{H}(m)^{a_2}$.

Embedding our scheme in a fair exchange protocol needs more careful thoughts which also apply to other schemes in this paper. First of all, $\mathcal{T}$ may avoid to store secret values per capita. In our example, $\mathcal{T}$ has to store x for Alice. This can be avoided by simply inserting a symmetric encryption of x into $CERT_{\mathcal{T}:A}$ during the initialization phase. Thus, $\mathcal{T}$ needs to store only the symmetric encryption key.

Second, Alice should sign the message sent to Bob. This message should include the verifiable encryption of Alice's signature and a *label* describing what Alice expects from Bob. This would guarantee the correctness of the recovery phase performed by $\mathcal{T}$ in case of dispute.

3.2 Gennaro–Halevi–Rabin Signatures

Let n be the product of two safe primes $p = 2p' + 1$ and $q = 2q' + 1$. Alice's certified public-key is (n, s) , where s is randomly chosen in $\mathbb{Z}_n^*$. In order to sign a message m , Alice computes $e = \mathcal{H}(m)$ and $\sigma = s^{1/e} \bmod n$. To verify the signature, Bob computes $e = \mathcal{H}(m)$ and checks whether $\sigma^e = s \bmod n$. The Gennaro–Halevi–Rabin signature scheme [Gennaro et al. 1999] has been proved to be resistant against adaptive chosen message attack (i.e., where the attacker can dynamically ask the signer to sign any message, using him as an oracle), in the random oracle model [Bellare and Rogaway 1993], under the strong RSA assumption.⁷ Gennaro et al. [1999] provide other constructions eliminating the need for the random oracle model.

To make a $\mathcal{T}$ -verifiable encryption of the signature σ , Alice performs the initialization phase described earlier for the RSA scheme. Thus, Alice gets the certificate $CERT_{\mathcal{T}:A} = S_{\mathcal{T}}(g, y = g^x, A, (n, s))$ from $\mathcal{T}$, with g of order $p'q'$, generator of Q_n . The resulting protocol is shown in Figure 3 (all the operations are made modulo n).

As described in Gennaro et al. [1999], the output length of the hash function $\mathcal{H}(\cdot)$ should be $|n|$ -bit long.

⁵Intuitively, a cryptosystem is semantically secure if, a passive attacker, who knows that one of just two possible messages has been encrypted, cannot yield any information about which of the two was actually encrypted by simply analyzing the ciphertext.

⁶Roughly said, given $G = \langle g \rangle$, the *Decisional Diffie-Hellman assumption* (DDH) states that no efficient algorithm can distinguish between the two distributions $\langle g^a, g^b, g^{ab} \rangle$ and $\langle g^a, g^b, g^c \rangle$, where, a, b, c are random elements in $[1, |G|]$. The DDH is believed to be intractable in Q_n (see Boneh 1998).

⁷The strong RSA assumption states that given a random $z \in \mathbb{Z}_n^*$ with n an RSA modulus, finding a pair $(u, e) \in \mathbb{Z}_n^* \times \mathbb{Z}$ such that $e > 1$ and $u^e = z$ is hard to solve.

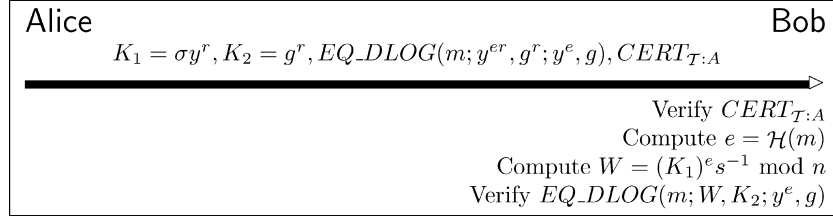


Fig. 3. Verifiable encryption of a Gennaro–Halevi–Rabin signature.

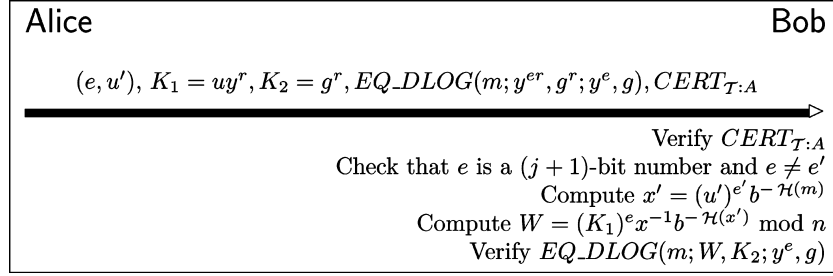


Fig. 4. Verifiable encryption of a Cramer–Shoup signature.

To achieve semantic security, it is sufficient to square σ (notice that, since $(\sigma^2)^{\mathcal{H}(m)} = s^2$ and $\gcd(\mathcal{H}(m), 2) = 1$, it is easy to recover σ) or, alternatively, to select a priori the parameter s as a quadratic residue.

3.3 Cramer–Shoup Signatures

Let j and z be two security parameters such that $j+1 < z$. Let $n = pq$, where p and q are z -bit safe primes, that is, $p = 2p' + 1$ and $q = 2q' + 1$ with both p' and q' . Alice's public-key is (n, b, x, e') , where b, x are randomly chosen in the subgroup of quadratic residues modulo n , Q_n , and e' is a random $(j+1)$ -bit prime. To generate a Cramer–Shoup signature [Cramer and Shoup 1999] on a message m , Alice randomly selects a $(j+1)$ -bit prime $e \neq e'$ and $u' \in Q_n$. Then, she computes $x' = (u')^{e'} b^{-\mathcal{H}(m)}$ and, finally, $u = (xb^{\mathcal{H}(x')})^{1/e}$.⁸ The resulting signature on m is (e, u, u') . To verify it, Bob checks that e is a $(j+1)$ -bit number different from e' and computes $x' = (u')^{e'} b^{-\mathcal{H}(m)}$. Then, Bob checks whether $x = u^e b^{-\mathcal{H}(x')}$. The Cramer–Shoup signature is quite efficient and provably secure (in the standard model) against the adaptive chosen message attack under the strong RSA assumption.

To make a $\mathcal{T}$ -verifiable encryption of the signature (e, u, u') , Alice performs an initialization phase (the one described for the RSA scheme). Thus, Alice gets the certificate $CERT_{T:A} = S_T(g, y = g^x, A, (n, b, x, e'))$ from $\mathcal{T}$, with g of order $p'q'$. Then, Alice releases (e, u') along with the verifiable encryption of u . The details are shown in Figure 4.

In the matter of semantic security, it should be noted that $u \in Q_n$, hence $K_1 = uy^r$ is already a quadratic residue.

⁸ $\mathcal{H}(\cdot)$ is a hash function $\mathcal{H} : \{0, 1\}^* \rightarrow \{0, 1\}^j$.

3.4 Guillou–Quisquater Signatures

A TTP generates common parameters v , $n = pq$ and, for each user, a secret key B and an ID-based public-key J such that $B^v J \equiv 1 \pmod n$. Only the TTP knows the order of $\mathbb{Z}_n^*$. The Guillou–Quisquater signature [Guillou and Quisquater 1988] on a message m is computed as follows: randomly choose $r \in \mathbb{Z}_n^*$ and compute $T = r^v \pmod n$, $d = \mathcal{H}(m \| T)$, and $D = r B^d \pmod n$, where $\mathcal{H}(\cdot)$ is a suitable hash function. The resulting signature is the pair (d, D) . The TTP, which has generated the system parameters, may cease to exist after the initialization phase.

To verify the signature, it is sufficient to check whether $d = \mathcal{H}(m \| D^v J^d)$, in fact notice that $D^v J^d \equiv (r B^d)^v J^d \equiv r^v (B^v J)^d \equiv r^v \pmod n$.

To make a verifiable encryption of a Guillou–Quisquater signature (d, D) , we slightly modify the idea proposed in Bao et al. [1998]. Basically, the modulus n is generated as product of two safe primes, that is, $n = pq$ with $p = 2p' + 1$ and $q = 2q' + 1$, and an element $g \in Q_n$ of order $p'q'$ is selected. The public-key of $\mathcal{T}$ is $y = g^x \pmod n$. To generate a $\mathcal{T}$ -verifiable encryption of (d, D) , Alice sends the ElGamal encryption of D , that is, $K_1 = D y^r$, $K_2 = g^r$, along with $V = D^v$, d itself and a signature showing knowledge of two equal discrete logarithms, $EQ_DLOG(m; y^{vr}, g^r; y^v, g)$ (notice that Alice does not know the order of g). The signature is verified by checking the correctness of $EQ_DLOG(m; K_1^v/V, K_2; y^v, g)$ and testing whether $d = \mathcal{H}(m \| V J^d)$.

3.5 Discrete Logarithms

In this section, we show a simple method of verifiably encrypting discrete logarithms, which can be used to make verifiable encryptions of digital signatures like DSA, ElGamal, and Schnorr.

The problem we want to solve is the following: Alice and Bob agree on a common value α^x and Alice wants to generate a $\mathcal{T}$ -verifiable encryption of x , given α^x . Clearly, we are assuming that finding the discrete logarithm x from α^x is hard.

Suppose that $\mathcal{T}$ could select a group G of order n in which computing the discrete logarithm is an easy task, that is, given an element $g \in G$ and $g^x \pmod n$, getting x is trivial. Therefore, Alice could make a verifiable encryption of x by just sending α^x and $g^x \pmod n$, and then proving that $D \log_\alpha \alpha^x = D \log_g g^x$. Obviously, only $\mathcal{T}$ should be able to compute discrete logarithms with respect to base g , that is, g , and a description of the group G should be publicly available and constitute a trap-door function of some sort. Two possible suitable trap-door functions are those defined in the Naccache–Stern and the Okamoto–Uchiyama public-key cryptosystems [Naccache and Stern 1998; Okamoto and Uchiyama 1998].

For the sake of simplicity, we focus on the Naccache–Stern cryptosystem but all the results can easily be adapted to work with the Okamoto–Uchiyama scheme. In the Naccache–Stern cryptosystem, an RSA modulus $n = pq$ is generated along with a small integer B . Then, σ is computed as a square-free odd B -smooth integer such that it divides $\phi(n)$ and is prime to $\phi(n)/\sigma$ (a suggested size is $\sigma > 2^{160}$). Let g be an element whose multiplicative order modulo n is

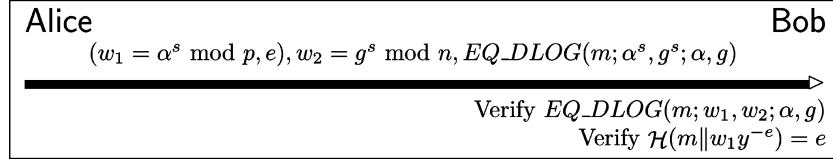


Fig. 5. Verifiable encryption of a Schnorr signature.

a large multiple of σ . A message $m < \sigma$ is encrypted by computing $g^m \bmod n$. Decryption is performed using the prime factors of σ , getting m by chinese remaindering (see Naccache and Stern 1998 for details). The resulting scheme is quite efficient, indeed the optimized version costs a couple of RSA operations with a similar modulus. The security of the scheme relies on the *higher residuosity problem* which is widely believed to be infeasible.

The verifiable encryption of x , given α^x , is performed by computing $g^x \bmod n$ and showing that $D \log_{\alpha} \alpha^x = D \log_g g^x$ via $EQ_DLOG(m; \alpha^x, g^x; \alpha, g)$, for an arbitrary message m . Although we still use the same notation (i.e., $EQ_DLOG(\cdot)$) as in the rest of the paper, proving equality of discrete logarithms from different groups requires an additional step by which the verifier generates an element of secret order that is subsequently used in the protocol. Without this element, the proof of equality may not work properly (see Camenisch and Michels 1999b).

3.6 Schnorr and Other Discrete-log-type Signatures

Let α be a generator of the unique cyclic group of order a prime q in $\mathbb{Z}_p^*$, where p is some large prime number such that q divides $p - 1$. Alice (the signer) selects the private key $1 \leq a \leq q$, computes $y = \alpha^a \bmod p$ and publishes (p, q, α, y) . To generate a Schnorr signature on a message m , Alice selects a random secret integer k , with $1 \leq k \leq q - 1$, and computes $r = \alpha^k \bmod p$, $e = \mathcal{H}(m || r)$ ⁹, and $s = ae + k \bmod q$. Alice's signature on m is the pair (s, e) . Bob verifies the signature by checking whether $e = e'$, where $e' = \mathcal{H}(m || \alpha^s y^{-e})$.

Notice that α is raised to the power of s by Bob. This suggests the following technique to make a $\mathcal{T}$ -verifiable encryption of the signature (s, e) : Alice sends $\{\alpha^s, e\}$ to Bob, proves knowledge of $D \log_{\alpha} \alpha^s$ and, finally, makes a verifiable encryption of s . The details are shown in Figure 5, where (g, n) denotes the public-key of $\mathcal{T}$ generated according to the Naccache–Stern cryptosystem (in particular, we set $\sigma > q$).

Showing, via $EQ_DLOG(\cdot)$, that the discrete logarithm of w_1 equals that of w_2 , implicitly proves that the signer (Alice) knows $D \log_{\alpha} w_1$ and $D \log_g w_2$, respectively. The exponent s must satisfy some range constraints as described in Section 2; this might affect the choice of the random element k during the signature process.

The technique above can be also generalized to work with several discrete-log-based signatures schemes, such as Poupard and Stern [1998], ElGamal and DSA.

⁹ $\mathcal{H}(\cdot)$ is a hash function $\mathcal{H} : \{0, 1\}^* \rightarrow \mathbb{Z}_q$.

3.7 Security Analysis

A verifiable encryption of a public-key signature scheme is a protocol between a prover and a verifier that convinces with high probability the verifier that an encrypted value is indeed a signature on a known message.

More formally, let the triple (G, E, D) denote a semantically secure crypto system, where k is the security parameter and $(E, D) \leftarrow G(1^k)$ is a pair such that $D(E(m)) = m$ for all messages $m \in \{0, 1\}^k$. Let $(SK, PK) \leftarrow G_2(1^{k_2})$ a pair of secret and public key, respectively. Let $s \in S(m)$ denote a signature on a given message m under (SK, PK) . A verification algorithm $V(PK, s, m)$ will return 1 if s is indeed a signature on m under PK , an arbitrary value otherwise. Inspired by the definition in Camenisch and Damgard [2000], we have the following definition.

Definition 1. A verifiable encryption scheme for a signature s on a message m consists of a two-party protocol (P, V) and an extraction algorithm R that extracts s from the output generated by V . Let O_P denote the output generated by V , when interacting with P on input $(E(s), m, k)$. The following properties must hold:

- (1) If P and V are honest then $O_P \neq \perp$ for all $(E, D) \leftarrow G(1^k)$ and for all possible values of s and m .
- (2) For all polynomial time $\bar{P}$, all positive polynomials $p(\cdot)$, all sufficiently large k , and all $(E, D) \leftarrow G(1^k)$ we have: $\Pr[V(PK, R(D, O_P), m) \neq 1 \text{ and } O_P \neq \perp] < \frac{1}{p(k)}$.
- (3) Signature Hiding: Given oracle access to P , V does not gain a non-negligible advantage into computing any signature $s' \in S(m)$.
- (4) Unforgeability: Given oracle access to P , which can be queried to obtain several valid verifiably encrypted signatures under PK , it is impossible to generate a valid verifiably encrypted signature under PK on a new message without further queries to the oracle.

In this paper, we have analyzed the verifiable encryption of RSA-based signatures as well as discrete-log-based ones. For all the RSA-based signatures, we have solved the problem of verifiably encrypting a signature by releasing a verifiable encryption of a e th root modulo n , where n is the product of two safe primes. Whereas in the discrete-log setting, we release a verifiable encryption of a discrete logarithm.

We concentrate on the interactive protocol underlying the verifiable encryption we used for the RSA signature scheme.

Let $M = H(m)^2$, $y = g^x$, and $h = y^e$. The interactive protocol between Alice (prover) and Bob (verifier) is run as follows:

- (1) Alice chooses at random r, t and sends Bob:

$$\{K_1 = M^d y^r, K_2 = g^r, a = h^t, b = g^t\}.$$

- (2) Bob chooses a random challenge $c \in \{0, 1\}^{160}$ and sends it to Alice.
- (3) Alice replies with $s = t + cr \pmod{p'q'}$.

(4) Bob computes $W = K_1^e/M$ and checks whether

$$h^s = aW^c \quad \text{and} \quad g^s = bK_2^c.$$

Suppose that Alice can successfully answer two distinct challenges. That is, given two challenges c_1, c_2 from Bob, Alice answers s_1, s_2 . As noted in Chaum and Pedersen [1992], we then have

$$h^{s_1-s_2} = W^{c_1-c_2} \quad \text{and} \quad g^{s_1-s_2} = K_2^{c_1-c_2}.$$

Thus, if $c_1 - c_2 < p'q'$, a value r exists such that

$$r = \text{Dlog}_h W = \text{Dlog}_g K_2 = (s_1 - s_2)/(c_1 - c_2) \pmod{p'q'}.$$

Consequently, $W = h^r$, $K_2 = g^r$, and $K_1 = (Mh^r)^d$. This proves that (K_1, K_2) is indeed an encryption of $H(m)^{2d}$.

Notice that we do not have to prove that the protocol above is a proof of knowledge, nevertheless it is interesting to analyze this possibility. We must rely on the strong RSA assumption, in fact the knowledge extractor does not know the group order $(p'q')$ and thus cannot compute $(s_1 - s_2)/(c_1 - c_2)$ (modulo $p'q'$). However, this computation can be carried out in $\mathbb{Z}$ since it is easy to see that, under the strong RSA assumption, $(c_1 - c_2)$ divides $(s_1 - s_2)$ (in $\mathbb{Z}$). Furthermore, since Alice (the prover) knows the factorization of n (excluding the protocol for the Guillou–Quisquater signatures), we should slightly modify the protocol as follows:

- During the initialization phase the TTP $\mathcal{T}$ selects a composite $\hat{n}$ and $\hat{g} \in \mathbb{Z}_{\hat{n}}^*$, and sends
 $CERT_{\mathcal{T}:A} = S_{\mathcal{T}}(g, y = g^x, A, (e, n), \hat{g}, \hat{n})$ to Alice.
- Given $K_1 = M^d y^r$, $K_2 = g^r$, $a = h^t$, $b = g^t$ and $K_3 = \hat{g}^r \pmod{\hat{n}}$, Alice now convinces Bob that:

$$\text{Dlog}_y(K_1^e/M) = \text{Dlog}_g K_2 = \text{Dlog}_{\hat{g}} K_3.$$

Therefore, we can apply the proof techniques in Camenisch and Michels [1998b] and Fujisaki and Okamoto [1997].

4. A CERTIFIED E-MAIL PROTOCOL

A fair exchange of digital signatures can be built via verifiable encryption schemes as shown in Bao et al. [1998] and Asokan et al. [1998b]. The three-pass scheme in Bao et al. [1998] is simple but has the drawback that one of the participants can decide when the two signatures will be exchanged after the protocol has started. In particular, Alice may receive Bob's signature after a certain amount of time, which can be chosen by Bob so to make his signature useless for Alice. A better model for fair exchange is presented in [Asokan et al. 1998a, 1998b].

Certified e-mail protocols are strictly related to fair exchange protocols. Indeed, in a certified e-mail scheme the intended recipient gets the mail content

if and only if the mail originator receives an irrefutable *evidence* that the message has been, or will soon be, received by the recipient. Thus, the *evidence* may be a proof-of-delivery (the message was delivered to the recipient) or a proof-of-submission to a TTP (post office), which will forward any message to the intended recipient in a short time. Notice that, a certified e-mail protocol is not a simultaneous exchange of items but rather a *asymmetric* exchange since the message has to be sent first to allow the recipient to compute a receipt based on the message received.

In the rest of the paper, we show how to use our verifiable encryption schemes to build an efficient certified e-mail protocol.

4.1 Definitions

We assume that each communication party has the ability to generate and verify digital signatures. We say that the TTP $\mathcal{T}$ is *visible* if it is obvious from the final outcome that $\mathcal{T}$ participated during the protocol. A protocol is *invasive* [Asokan et al. 1998a; Micali 1997] if one can tell that it was run by just looking at the outcome. Typically, if the TTP $\mathcal{T}$ is visible then the protocol that employs $\mathcal{T}$ is invasive (notice that the reverse is not true).

A certified e-mail protocol should minimally provide:

- *Fairness*: No party should be able to interrupt or corrupt the protocol to force an outcome to his/her advantage. The protocol should terminate with either party having obtained the desired information, or with neither one acquiring anything useful.
- *Monotonicity*: Each exchange of information during the protocol should add validity to the final outcome. That is, the protocol should not require any messages, certificates, or signatures to be revoked to guarantee a proper termination of the protocol. (This is important because if revocation is needed to ensure fairness then the verification of the validity of the protocol outcome becomes a bottleneck as it would require TTP's active participation.)
- *TTP invisibility*: It makes it impossible to determine whether the TTP has been involved or not. (If the TTP were visible it would imply that the information added by the TTP to any protocol outcome cannot be removed and must be signed by the TTP itself. Verifying such a signature from the TTP would be then part of the receipt verification process. This would be too impractical for several reasons. For instance, the TTP is trusted during the exchange but may not be so after a certain period of time. Still the validity of the receipt is bound to the validity of the signature generated by the TTP. The invisibility property provides some peace of mind: Once the exchange has took place, the TTP may cease its activity, disappear, modify its policies, or change public keys without affecting the validity of the protocol outcome.)
- *Timeliness*: It guarantees that both parties will achieve their desired items in the exchange within finite time.

Occasionally, it is desirable to keep the content of confidential e-mails secret from trusted parties acting as intermediaries. Thus, an optional feature is

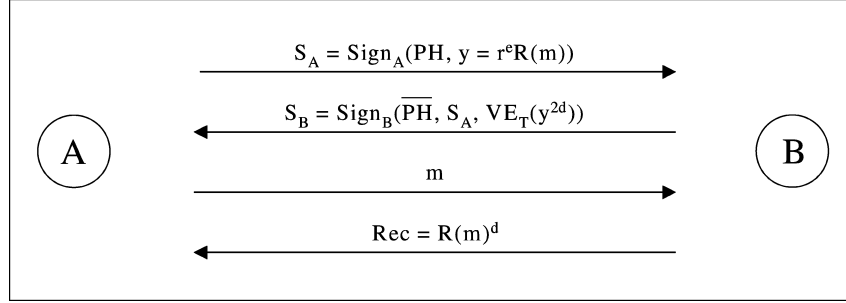


Fig. 6. Off-line certified e-mail protocol via verifiable encryption of RSA signatures.

—*Confidentiality*: In case the exchange is deemed confidential, the protocol should not need to disclose the message contents to any other party excepting the sender and the recipient.

4.2 The Protocol

A certified e-mail protocol using verifiable encryption is shown in Figure 6. We will assume that the communication is carried over private and authenticated channels.

The protocol provides fairness; specifically, it ensures that the sender receives the receipt if and only if the recipient will have the message in his mailbox within a finite period of time. The protocol is designed so that the TTP is invoked only in case of dispute. As long as both *A* and *B* follow the protocol steps, there is no need to involve the trusted entity in the protocol. This represents an improvement over the approach employed by online protocols, where a trusted entity is needed for each transaction.

Moreover, the protocol is designed to make sure that *A* cannot misbehave. Only *B* is allowed to cheat by not sending the message in the last step. Since the sender initiates the exchange process, it appears natural to desire that the recipient of the message be relieved by any burden caused by malicious senders.

User *B* receives the certificate $CERT_{TTP:B}$ by engaging in an initialization phase with the TTP. Similarly, *B*'s public key is (e, n) with e' and n product of safe primes, and $QR(n)$ is the subgroup of squares in which we operate. We make use of two protocol headers, PH and $\overline{PH}$, which contain relevant information such as the identities of the parties involved (*A*, *B*, and TTP), the cryptographic algorithms employed, timestamps and transaction IDs to prevent replay attacks, and other pertinent information about the protocol.

The protocol consists of the following steps (all the operations are taken modulo n):

- Step 1*. The sender *A* selects a random r , computes $y = r^e R(m)$, and signs it including a protocol header PH . Such a signature, denoted by S_A , is sent to *B*.
- Step 2*. The recipient *B* squares y and computes $(y^2)^d = r^2 R(m)^{2d}$. It then computes the verifiable encryption of y^{2d} , $VE_T(y^{2d})$, and sends the result to *A*. However, *B* has to sign the result in order to include a protocol header $\overline{PH}$

and the sender's signature S_A . More importantly, B 's signature (S_B) makes it possible to neutralize *malleability* attacks against the ElGamal encryption and also preserves B 's *protocol view* at that specific point in time.

- Step 3.* After receiving the message from B , A verifies the signature and that the encryption contains the correct receipt. If that is the case, A sends m to B .
- Step 4.* The recipient B reads the message m and sends the receipt $Rec = R(m)^d$ to A .

If B does not send the receipt in Step 4, then A contacts the trusted entity and both run the following protocol:

- Step 1.* A sends B 's signature, S_B , to the TTP along with r and m .
- Step 2.* The TTP verifies first the signatures S_A and S_B (S_A is contained in S_B). Then, it recovers y^{2d} from the verifiable encryption and computes y^d via the Euclidean algorithm. Finally, the TTP checks whether the value $s = y^d/r$ is indeed a valid signature of the message m under B 's public key (i.e, it checks whether $s^e = R(m)$). If so, it sends s to A and forwards m to B .

The TTP has to forward the message m to B to nullify any attempts of the sender A to successfully retrieve a receipt without revealing the message m to B . Specifically, A may not have sent the message m in Step 3 above. The protocol fairness is built around the assumption that the sender A can verify that the verifiable encryption indeed contains a valid receipt. Only the TTP can recover the receipt from the verifiable encryption.

In our protocol, the recipient B does not have to be always online. If B is offline, then TTP may have at least three possible behaviors:

- (1) TTP acts as a SMTP-like server: TTP will keep state and retry until B is online.
- (2) TTP acts as a SMTP-like server and B has his own online mail server that collects his emails. This is the most likely case: TTP will forward any messages to B 's mail server.
- (3) TTP acts as a POP-like server: TTP will keep state until B retrieves his messages.

The receipt, in our protocol, is a proof that the message was delivered to either B or TTP (which will forward it to B within a finite period of time). Indeed, the receipt states only that B has the ability, or potential, to read the message if he wants to, but B may decide to stay offline, or may not want to read the message, or may automatically delete it (e.g., in case of *spam*). This implies that the receipt obtained by the sender should be interpreted as a proof-of-submission, which is weaker than a proof-of-delivery provided by other certified email schemes, such as the one in Asokan et al. [1998b].

Remark 1. In our protocol, the sender A has to reveal the message m to the TTP in case of dispute. If message privacy has to be preserved, it is sufficient to substitute m with $\overline{PH} || P_B(m)$ in the protocol above, where $P_B(\cdot)$ represents the public-key encryption under B 's public key and $\overline{PH}$ is a protocol header. Notice

that the receipt assumes a new format:

$$Rec = R(\overline{\overline{PH}} || P_B(m))^d,$$

which has to be interpreted in a special way: it is considered a valid receipt of the message m only when accompanied by m and $\overline{\overline{PH}}$ such that

$$(Rec)^e = R(\overline{\overline{PH}} || P_B(m)).$$

The public-key encryption $P_B(\cdot)$ should be deterministic or, if randomized, the sender A must reveal the random parameters used to encrypt the message. The approach we have taken for the implementation of the protocol is to encrypt the message m as $E_k(MAC_l(m) || m)$, $P_B(k || l)$, where k, l are random secret values; $MAC_l(\cdot)$ is a MAC function, such as HMAC-SHA-1; $P_B(\cdot)$ is a deterministic public-key encryption algorithm, such as plain RSA; $E_k(\cdot)$ is a symmetric-key encryption algorithm, such as AES in CBC mode.

The new protocol header $\overline{\overline{PH}}$ has to be checked, either by B or the TTP, to contain the correct information relevant to the protocol. Moreover, it has to clearly state that the receipt Rec has to be interpreted in the special way described above. This inevitably makes the new protocol invasive.

Remark 2. The certified e-mail protocol presented above works for RSA signatures but can easily be extended to work for other schemes based on a similar setting such as Rabin and Guillou–Quisquater [Guillou and Quisquater 1988] signature schemes, or provably unforgeable signature schemes such as Cramer–Shoup [Cramer and Shoup 1999] and Gennaro–Halevi–Rabin [Gennaro et al. 1999]. To do so, it is enough to replace the verifiable encryption of RSA signatures with that of the desired signing algorithm.

4.3 Analysis

In this section, we present an analysis of our protocol. Our claim is follows.

CLAIM 1. *The protocol above is a certified e-mail protocol, which provides fairness, monotonicity, timeliness, and TTP invisibility. Moreover, the protocol optionally provides confidentiality of the message, that is, the arbitration can be performed without revealing the e-mail content to the trusted intermediary.*

Clearly our protocol provides TTP invisibility since the structure of the receipt does not indicate whether the TTP was involved or not in dispute resolutions. The protocol provides also monotonicity, since any signature (including the receipt) will not be revoked in order to guarantee a proper termination of the protocol. Confidentiality is achieved by encrypting the actual message content.

The protocol provides timeliness since the protocol resolution can be executed in a finite period of time and messages are sent through *resilient* channels [Asokan et al. 1998a], that is, messages will be delivered within a time lapse which may be arbitrarily long, yet finite.

Regarding fairness, there are two cases to consider. In the first case, the TTP is not invoked and both A and B follow the protocol steps. In the second case, the TTP is invoked by A to solve a dispute (B cannot invoke the TTP). Observe

that, in our protocol, the first two messages are used to collect evidences that A can use in case of dispute. Indeed, when the TTP is not invoked, the protocol messages that are relevant are only those in Steps 3 and 4, where a message m is sent in exchange of the corresponding receipt. Therefore, fairness is preserved in this case. If the TTP is invoked (by A) then B 's signature (S_B) will be sent to the TTP along with the message m and the blinding factor r . The TTP will compute y^d from the verifiable encryption and will check whether

$$(y^d/r)^e = R(m).$$

If that is the case, then A and B receive y^d/r and m , respectively. Hence, even in this case fairness is preserved since the sender will receive a signature on a message which is being forwarded to the recipient through a resilient channel.

It is interesting to notice that the recipient does not need to contact the TTP in case of dispute. This feature makes our protocols very attractive in real-world environments in which recipients would prefer to assume a passive role rather than being actively involved in dispute resolutions caused by malicious senders. More importantly, the recipient is *stateless* in the sense that it does not need to store state information on transactions in which he is involved.¹⁰ Indeed, the recipient may not store anything about the first two steps of the protocol and, in principle, the message embedded by the sender in the value y in Step 1 of the protocol could be different from the message sent in Step 3.

4.4 Comparisons

In this section we compare our protocol with the state of the art in the field.

Some of the offline protocols are not monotonic, for instance, the protocol in Asokan et al. [1998a] requires signatures to be revoked in order to guarantee fairness. Among monotonic and off-line protocols, we believe those in Micali [1997] and Asokan et al. [1998b] represent the state-of-the-art in the field. The work of Micali [Micali 1997] shows that it is possible to achieve a simple certified e-mail protocol with only three messages (one less than our protocol). However, it should be noticed that

- (1) the recipient of the message has to be actively involved in the dispute resolution and is forced to keep state;
- (2) a time limit has to be incorporated into the message by the sender to force the recipient to send the receipt within a specified period of time. This has to be done in order to guarantee fairness;
- (3) a *reliable* channel (as opposed to a resilient channel) is required between the recipient and the TTP.

A channel is reliable when it is always operational and operates without delays. It is very difficult to build a reliable channel in some network environments, such as wireless networks. This fact may limit the applicability of the protocol in Micali [1997]. Furthermore, for each message received, the recipient

¹⁰What we state here is that the protocol does not require B and TTP to store state information. It is important to notice, however, that implementing a resilient channel with the current technology may require B and TTP to do so.

is forced to communicate with the trusted intermediary in case of dispute and this has to happen before the time limit expires.

The work in Asokan et al. [1998b] presents a fair-exchange protocol, which is provably secure in the random oracle model. The authors specialize their protocol to work as an offline certified e-mail scheme that, similarly to our protocol, requires only resilient channels. Their protocol is based on *verifiable escrow schemes*, which essentially are verifiable encryptions where each encryption comes with an attached *condition* that specifies a decryption policy. The scheme in Asokan et al. [1998b] has some drawbacks, more specifically:

- (1) It is expensive in terms of communication complexity, performance, and amount of data transmitted. This is mainly due to the *cut-and-choose* interactive proof technique employed to achieve a verifiable escrow.
- (2) The recipient has to keep state and both the sender and the recipient have to be actively involved in dispute resolutions (but notice that one of the participants may successfully complete the dispute protocol when the other is unavailable).
- (3) The TTP needs to keep state.

The main advantage of our scheme is that it allows the recipient to be *stateless*. Indeed, the recipient is not involved in dispute resolutions and is not even supposed to contact the TTP in any case.

In other schemes, a user would be forced to keep track of all messages received, store state information, and engage in protocol resolutions with the TTP. This may be very unappealing for those receiving hundreds of messages (including *junk* emails) per day, or for those operating in environments where servers storing state information may frequently *crash*. In some cases (such as in Micali [1997]), operations have to be made before a time limit expires which may make it even impossible to guarantee fairness in some environments.

Stateless-recipient protocols are particularly useful when employed by users equipped with mobile devices, such as cellular phones or wireless PDAs, or any other device that cannot reliably store state information.

Remark (on timely completion). Some contract signing protocols, such as the one in Asokan et al. [1998b], provide the feature that either party is able to definitively finish a transaction on its own (i.e., by interacting with the TTP only). We believe that this feature of *independent timely completion* is important in contract signing or fair exchange but not in certified email (and our protocol does not provide it). The reason is that, in a certified email protocol, the sender *A* originates the transactions so, implicitly, *A* has always the ability to complete them whenever it is more convenient. For example, *A* may have a message for *B* but may decide to send it later (e.g., after a month) because, for some reason, receiving *B*'s receipt after a certain amount of time would give her some advantage.

If *A* gains any advantage from receiving *B*'s receipts at certain times then employing a certified email protocol with the independent timely completion

property would not help: A may just delay the transaction and have the receipt from B when it is more convenient to her.

5. CONCLUSION

This paper presented simple and particularly efficient verifiable encryption protocols for digital signatures. These protocols may be used as building blocks in the design of efficient fair exchange of digital signatures and certified email protocols.

ACKNOWLEDGMENTS

We are grateful to Cristina Nita-Rotaru for her comments on this paper and for her contribution to the certified email protocol [Ateniese and Nita-Rotaru 2002]. Cristina's feedbacks were enlightening and her work on the system design and implementation of the certified email protocol has been invaluable.

Many thanks to Victor Shoup, Jan Camenisch, and Marc Joye for their insightful comments on the conference paper on verifiable encryption [Ateniese 1999], and to Breno de Medeiros for his feedbacks on some parts of this paper.

REFERENCES

- ASOKAN, N., SCHUNTER, M., AND WAIDNER, M. 1997. Optimistic protocols for fair exchange. In *Fourth ACM Conference on Computer and Communication Security*. ACM Press, 8–17.
- ASOKAN, N., SHOUP, V., AND WAIDNER, M. 1998a. Asynchronous protocols for optimistic fair exchange. In *IEEE Symposium on Security and Privacy* (Oakland, CA).
- ASOKAN, N., SHOUP, V., AND WAIDNER, M. 1998b. Optimistic fair exchange of digital signatures. *IEEE Journal on Selected Areas in Communications* 18, 4, 593–610, 2000. Extended abstract in *Advances in Cryptology—EUROCRYPT'98*. Lecture Notes in Computer Science, vol. 1403. Springer-Verlag, Berlin, 591–606.
- ATENIESE, G. 1999. Efficient verifiable encryption (and fair exchange) of digital signatures. In *Sixth ACM Conference on Computer and Communications Security (ACM CCS'99)*. Also appeared as IBM Research Report.
- ATENIESE, G., DE MEDEIROS, B., AND GOODRICH, M. T. 2001. TRICERT: distributed certified e-mail schemes. In *ISOC 2001 Network and Distributed System Security Symposium (NDSS'01)* (San Diego, CA, USA).
- ATENIESE, G. AND NITA-ROTARU, C. 2002. Stateless-recipient certified e-mail system based on verifiable encryption. In *RSA 2002, McEnery Convention Center* (San Jose, CA, USA, Feb. 19–22).
- BAHREMAN A. AND TYGAR, J. D. 1994. Certified electronic mail. In *Proceedings of Symposium on Network and Distributed Systems Security* (Feb. 1994). I. Society, 3–19.
- BAO, F., DENG, R. H., AND MAO, W. 1998. Efficient and practical fair exchange protocols with off-line TTP. In *IEEE Symposium on Security and Privacy* (Oakland, CA).
- BELLARE, M. AND ROGAWAY, P. 1993. Random oracles are practical: A paradigm for designing efficient protocols. In *First ACM Conference on Computer and Communication Security*. ACM Press, 62–73.
- BEN-OR, M., GOLDBREICH, O., MICALI, S., AND RIVEST, R. 1990. A fair protocol for signing contracts. *IEEE Transactions on Information Theory* IT-36, 1, 40–46.
- BONEH, D. 1998. The decision Diffie-Hellman problem. In *Algorithmic Number Theory (ANTS-III)*. Lecture Notes in Computer Science, vol. 1423. Springer-Verlag, Berlin, 48–63.
- CAMENISCH, J. AND DAMGARD, I. B. 2000. Verifiable encryption, group encryption, and their applications to separable group signatures and signature sharing schemes. In *Advances in Cryptology—ASIACRYPT'00*. Lecture Notes in Computer Science, vol. 1976. Springer-Verlag, Berlin, 331–345.

- CAMENISCH, J. AND MICHELS, M. 1998a. A group signature scheme with improved efficiency. In *Advances in Cryptology—ASIACRYPT'98*. Lecture Notes in Computer Science, vol. 1514. Springer-Verlag, Berlin, 160–174.
- CAMENISCH, J. AND MICHELS, M. 1998b. A Group Signature Scheme Based on an RSA-Variant. Technical Report RS-98-27, BRICS, Aarhus. An earlier version appears in Camenisch and Michels [1998].
- CAMENISCH, J. AND MICHELS, M. 1999a. Proving in zero-knowledge that a number is the product of two safe primes. In *Advances in Cryptology—EUROCRYPT'99*. Lecture Notes in Computer Science. Springer-Verlag, Berlin.
- CAMENISCH, J. AND MICHELS, M. 1999b. Separability and efficiency for generic group signature schemes. In *Advances in Cryptology—Crypto'99*.
- CHAN, A., FRANKEL, Y., AND TSIOUNIS, Y. 1998. Easy come—easy go divisible cash. In *Advances in Cryptology—EUROCRYPT'98*. Lecture Notes in Computer Science, vol. 1403. Springer-Verlag, Berlin, 561–575. Updated and corrected version available as GTE Technical Report.
- CHAUM, D. AND PEDERSEN, T. 1992. Wallet databases with observers. In *Advances in Cryptology—Crypto'92*, 89–105.
- CRAMER, R. AND SHOUP, V. 1999. Signature schemes based on the strong RSA assumption. In *Sixth ACM Conference on Computer and Communication Security*. ACM Press.
- DENG, R. H., GONG, L., LAZAR, A., AND WANG, W. 1996. Practical protocols for certified electronic e-mail. *Journal of Networks and Systems Management* 4, 3, 279–297.
- EVEN, S., GOLDBREICH, O., AND LEMPEL, A. 1985. A randomized protocol for signing contracts. *Communications of the ACM* 28, 6, 637–647.
- FIAT, A. AND SHAMIR, A. 1987. How to prove yourself: practical solutions to identification and signature problems. In *Advances in Cryptology—CRYPTO'86*. Lecture Notes in Computer Science, vol. 263. Springer-Verlag, Berlin, 186–194.
- FUJISAKI, E. AND OKAMOTO, T. 1997. Statistical zero knowledge protocols to prove modular polynomial relations. In *Advances in Cryptology—CRYPTO '97*. Lecture Notes in Computer Science, vol. 1294. Springer-Verlag, Berlin, 16–30.
- GENNARO, R., HALEVI, S., AND RABIN, T. 1999. Secure signatures, without trees or random oracles. In *Advances in Cryptology—EUROCRYPT'99*. Lecture Notes in Computer Science, vol. 1592. Springer-Verlag, Berlin, 123–139.
- GUILLOU, L. C. AND QUISQUATER, J. J. 1988. A paradoxical identity-based signature scheme resulting from zero-knowledge. In *Advances in Cryptology—CRYPTO'88*. Lecture Notes in Computer Science, vol. 403. Springer-Verlag, Berlin, 216–231.
- MENEZES, A. J., VAN OORSCHOT, P. C., AND VANSTONE, S. A. 1996. *Handbook of Applied Cryptography*. CRC Press, Boca Raton, FL (ISBN 0-8493-8523-7).
- MICALI, S. 1997. Certified e-mail with invisible post offices. Presented at the 1997 RSA Security Conference.
- NACCACHE, D. AND STERN, J. 1998. A new public key cryptosystem based on higher residues. In *Fifth ACM Conference on Computer and Communications Security*. ACM Press, 59–66.
- OKAMOTO, T. AND UCHIYAMA, S. 1998. A new public-key cryptosystem as secure as factoring. In *Advances in Cryptology—EUROCRYPT'98*. Lecture Notes in Computer Science, vol. 1403. Springer-Verlag, Berlin, 308–318.
- POUPARD, G. AND STERN, J. 1998. Security analysis of a practical “on the fly” authentication and signature generation. In *Advances in Cryptology—EUROCRYPT'98*. Lecture Notes in Computer Science, vol. 1403. Springer-Verlag, Berlin, 422–436.
- RIORDAN, J. AND SCHNEIER, B. 1998. A certified e-mail protocol. In *Thirteenth Annual Computer Security Applications Conference* (Dec.), 100–106.
- RIVEST, R. L., SHAMIR, A., AND ADLEMAN, L. M. 1978. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM* 21, 2, 120–126.
- SCHNORR, C. P. 1991. Efficient signature generation by smart-cards. *Journal of Cryptology* 4, 3, 161–174.
- ZHOU, J. AND GOLLMANN, D. 1996. Certified electronic mail. In *Proceedings of Computer Security—ESORICS'96*. Springer-Verlag, Berlin, 55–61.

Received February 2002; revised February 2003, April 2003, August 2003; accepted December 2003