INFORMATICA, 2005, Vol. 16, No. 1, 121–130

© 2005 *Institute of Mathematics and Informatics, Vilnius*

An Improvement of the Ateniese's Verifiable Encryption Protocol

Constantin POPESCU [*]

Department of Mathematics, University of Oradea
Str. Armatei Romane 5, Oradea, Romania
e-mail: cpopescu@uoradea.ro

Received: November 2003

Abstract. Verifiable encryption is a primitive that can be used to build extremely efficient fair exchange protocols where the items exchanged represent digital signatures. Such protocols may be used to digitally sign contracts on the Internet. This paper presents an efficient protocol for verifiable encryption of digital signatures that improves the security and efficiency of the verifiable encryption scheme of Ateniese. Our protocol can be applied to group signatures, key escrow and publicly verifiable secret and signature sharing to prove the fairness.

Key words: cryptographic protocols, verifiable encryption, security, digital signatures.

1. Introduction

Exchanging messages over the Internet is becoming a major business opportunity. Electronic commerce usually involves two distrusted parties exchanging one message for another, for instance an electronic check for an electronic ticket. Specialized applications may include contract signing, electronic purchase and certified electronic mail delivery. In simultaneous contract signing, Alice and Bob have agreed on a contract but neither wishes to sign unless the other signs as well. Face to face, this is easily solved: both simultaneously sign the contract. Unfortunately, simultaneity cannot be met in the discrete world. Therefore, it seems fruitful that many researchers have focused their attention on the fair exchange of digital signatures.

There have been several approaches to solve the fair exchange problem depending on the definition of fairness on which they are based. In (Even *et al.*, 1985), fairness is interpreted as equal computational effort. That is, both Alice and Bob generate a signature of the contract and then they communicate by taking turns and sending bit by bit their signatures to each other. Recently, papers (Asokan *et al.*, 2000) and (Bao *et al.*, 1998) have presented protocols for optimistically exchanged commonly used digital signature schemes. Both show that it is possible to build fair exchange protocols by means of what

[*] The author is presently at "Centre for Quantifiable Quality of Service in Communication Systems" (Q2S), NTNU, Trondheim, Norway. The centre is appointed Centre of Excellence by The Research Council of Norway. It is financed by the Research Council, NTNU and UNINETT, and supported by Telenor.

the authors in (Asokan *et al.*, 2000) have called verifiable encryption of digital signatures. This means to encrypt a signature under a designated public key and subsequently prove that the resulting cipher text indeed contains such a signature. The authors in (Camenisch and Damgaard, 2000), (Camenisch and Shoup, 2003) show how to generalize the scheme in (Asokan *et al.*, 2000) achieving more efficient schemes that can be proved secure without relying on random oracles. The fair exchange protocol using verifiable encryption was proposed by Ateniese (Ateniese, 2000) and Bao, Deng and Mao (Bao *et al.*, 1998) independently. These protocols apply ad-hoc techniques to create the fairness via a specific encryption scheme that confirms to a given signature. Unfortunately, the scheme proposed in (Ateniese, 2000) lack any formal security analysis.

In this paper we improve the security and efficiency of the verifiable encryption protocol of Ateniese (Ateniese, 2000), thus providing a valid primitive that is of interest in designing fair exchange protocols (Asokan *et al.*, 1997). The security of our verifiable encryption protocol follows from the security of the underlying signature scheme: a modification of the Cramer–Shoup's digital signature scheme (Cramer and Shoup, 2000). We prove that the underlying signature scheme is secure against an adaptively chosen message attack. Also, we show that our protocol can be applied to construct group signatures, key escrow and publicly verifiable secret and signature sharing.

The remainder of this paper is organized as follows. In the next section, we review some cryptographic tools necessary in the subsequent design of our verifiable encryption protocol. Then, we present our verifiable encryption protocol in Section 3. Furthermore, we discuss some applications of our protocol in Section 4. Finally, Section 5 concludes the work of this paper.

2. Preliminaries

We assume that each communication party has the ability to generate and verify digital signatures. In this section we present signature schemes allowing a prover to convince a verifier of the equality of discrete logarithms (Ateniese, 2000). The problem is, given g_1^x, g_2^x and a message m, generating a signature on a message m and, at the same time, showing that $D\log_{g_1} g_1^x = D\log_{g_2} g_2^x$ without revealing any useful information about x itself. We will denote an instance of this signature technique by $EDLOG(m; g_1^x, g_2^x; g_1, g_2)$. 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 (Schnorr, 1991) in order to show knowledge of relations among secrets. Substantially, these are signature schemes based on proofs of knowledge performed non-interactively making use of an ideal hash function H (à la Fiat–Shamir (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

$$D\log_g y_1 = D\log_h y_2.$$

The protocol, described by Chaum and Pedersen (Chaum and Pedersen, 1992), is run 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 sends $s = t - cx(\bmod q)$ to the verifier.

The verifier accepts the proof if $a = g^s y_1^c$ and $b = h^s y_2^c$. To turn the protocol above into a signature on an arbitrary message m, the signer can compute the pair (c, s) as

$$c = H\big(m \parallel y_1 \parallel y_2 \parallel g \parallel h \parallel g^t \parallel h^t\big), \quad s = t - cx,$$

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

$$c' = H\big(m \parallel y_1 \parallel y_2 \parallel g \parallel h \parallel g^s y_1^c \parallel h^s y_2^c\big).$$

In this paper we work into the subgroup of all quadratic residues modulo n, denoted by $QR(n)$. We select n as product of two safe primes p and q, i.e., such that $p = 2p' + 1$ and $q = 2q' + 1$, with $p\prime, q\prime$ primes. Thus, notice that $QR(n)$ is a cyclic group of order $p'q'$. The symbol $\parallel$ denotes the concatenation of two binary strings (or of the binary representation of group elements and integers).

Actually, the same signature scheme works properly even when the signer is working over a cyclic subgroup of $\mathbb{Z}_n^*$, $G =< g >$, whose order $\#G = p'q'$ is unknown but its bit-length l_G is publicly known. We make use of a hash function H: $\{0,1\}^* \rightarrow \{0,1\}^k$, which maps a binary string of arbitrary length to a k-bit hash value. In the next definition we show the knowledge and equality of the two discrete logarithms.

DEFINITION 1. Let $\epsilon > 1$ be a security parameter. A pair $(c, s) \in \{0,1\}^k \times \{-2^{\epsilon(l_G+k)+1}, \ldots, 2^{\epsilon(l_G+k)+1}\}$ satisfying $c = H(m \parallel y_1 \parallel y_2 \parallel g \parallel h \parallel y_1^c g^s \parallel y_2^c h^s)$ is a signature of a message $m \in \{0,1\}^*$ with respect to y_1 and y_2 and is denoted $EDLOG(m; g^x, h^x; g, h)$.

A signature (c, s) of a message $m \in \{0,1\}^*$ can be computed as follows. An entity knowing the secret key x, is able to compute the signature (c, s), provided that $x = D\log_g y_1 = D\log_h y_2$, by choosing a random $t \in \pm\{0,1\}^{\epsilon(l_G+k)}$ and then computing c and s as

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

A way of proving the security of the signature scheme above is via the oracle replay technique formalized in (Pointcheval and Stern, 1996) by Pointcheval and Stern.

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 $G_1 =< g >, G_2 =< h >$, the verifier can only conclude that the signer knows a value x such that $x \bmod q_1 = D\log_g y_1$ and $x \bmod q_2 = D\log_h y_2$. However, it is possible to prove that a secret x lies in a specific interval, more precisely given g^x with $-2^l < x < 2^l$ for an integer l, it is possible to prove that x lies in the extended interval $[-2^{\epsilon(l+k)}, 2^{\epsilon(l+k)}]$. Hence, we might build a signature scheme for showing that $D\log_g y_1 = D\log_h y_2$ in $\mathbb{Z}$ by combining the scheme for showing knowledge of a value x with $x\ mod\ q_1 = D\log_g y_1$ and $x \bmod q_2 = D\log_h y_2$ and the scheme for showing that $-2^{\epsilon(l+k)} < x < 2^{\epsilon(l+k)}$. Clearly, this can be done only if the length l can be chosen such that $2^{\epsilon(l+k)+1} < \min\{q_1, q_2\}$, where q_1, q_2 are the orders of g and h, respectively. This idea is formalized in (Camenisch and Michels, 1999). Camenisch and Michels proposed in (Camenisch and Michels, 1999) a concrete protocol for proving equality of discrete logarithms from different groups. Their protocol is mostly based on a technique developed by Fujisaki and Okamoto (Fujisaki and Okamoto, 1998). To provide a viable example of how it is possible to show that x lies in the extended interval $[-2^{\epsilon(l+k)}, 2^{\epsilon(l+k)}]$ we present a signature scheme derived from a protocol due to Chan, Frankel and Tsiounis (Chan *et al.*, 1998) and Camenisch and Michels (Camenisch and Michels, 1998) (see Definition 2). The scheme can trivially be extended to the more general interval $[X - 2^{\epsilon(l+k)}, X + 2^{\epsilon(l+k)}]$ for a given integer X.

DEFINITION 2. The signature on a message $m \in \{0,1\}^*$ is a pair $(c, s) \in \{0,1\}^k \times \{-2^{\epsilon(l_G+k)+1}, ..., 2^{\epsilon(l_G+k)+1}\}$ such that $c = H(m \parallel y \parallel g \parallel y^c g^s)$. This shows knowledge of the discrete logarithm of $y = g^x$ with respect to base g and that this logarithm lies in $[-2^{\epsilon(l+k)}, 2^{\epsilon(l+k)}]$.

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

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

The underlying interactive protocol is proved to be a proof of knowledge under the strong RSA assumption.

The Strong RSA assumption was independently introduced by Baric and Pfitzmann (Baric and Pfitzmann, 1997) and by Fujisaki and Okamoto (Fujisaki and Okamoto, 1997). It strengthens the widely accepted RSA assumption that finding e^{th}-roots modulo n, where e is the public and thus fixed exponent, is hard to the assumption that finding an e^{th}-root modulo n for any $e > 1$ is hard.

DEFINITION 3 (Strong RSA Problem). Let $n = pq$ be an RSA-like modulus and let G be a cyclic subgroup of $\mathbb{Z}_n^*$ of order l_g. Given n and $z \in G$, the Strong RSA Problem consists of finding $w \in G$ and $v \in \mathbb{Z}_{>1}$ satisfying $z \equiv w^v (\bmod n)$.

Assumption 1 (Strong RSA Assumption). *There exists a probabilistic polynomial time algorithm K which on input 1^{l_g} outputs a pair (n, z) such that for all probabilistic polynomial-time algorithms P, the probability that P can solve the Strong RSA Problem is negligible.*

3. The Verifiable Encryption Protocol

We note that our improvement refer to the Ateniese's protocol (Ateniese, 2000) which is based on the signature scheme of Cramer and Shoup (Cramer and Shoup, 2000).

Let Alice and Bob be two users willing to exchange digital signatures on a message m. We make use of a trusted third party, i.e., the trusted third party takes part in the protocol only if one user cheats or simply crashes. Let T be a trusted third party and let $P_U(m)$ denote the encryption of the message m with U's public key, whereas $S_U(m)$ denotes the signature generated by U on a message m. Alice generates a signature $S_A(m)$ and sends it encrypted to Bob by computing $C(S_A(m)) = P_T(S_A(m))$. The problem is that Alice must prove to Bob that the signature is valid and that T is able to get $S_A(m)$ from $C(S_A(m))$.

In our protocol, P_U is the ElGamal encryption scheme and S_U is a modification of the Cramer–Shoup's digital signature scheme (Cramer and Shoup, 2000). That is, given a secret key x and a corresponding public key $y = g^x (\bmod n)$, a message m is encrypted by generating a random r'' and computing $c_1 = mg^{xr''} (\bmod n)$, $c_2 = g^{r''} (\bmod n)$. To get m from c_1, it is sufficient to compute $m = c_1/(c_2)^x$.

First, we describe a modification of the Cramer–Shoup's digital signature scheme (Cramer and Shoup, 2000). Let $\varepsilon > 1$ be a security parameter and let $l_p, l_{\lambda_1} > l_{\lambda_2}, l_{\gamma_1} > l_{\gamma_2}$ denote lengths. Define the integral ranges $\Lambda = [2^{l_{\lambda_1}} - 2^{l_{\lambda_2}}, 2^{l_{\lambda_1}} + 2^{l_{\lambda_2}}]$ and $\Gamma = [2^{l_{\gamma_1}} - 2^{l_{\gamma_2}}, 2^{l_{\gamma_1}} + 2^{l_{\gamma_2}}]$ such that for all $(x, e) \in \Lambda \times \Gamma$, we have $0 < x + 2^{2l_p} < e$. Finally, let H: $\{0,1\}^* \to \Lambda$ be a collision-resistant hash function.

To generate her public and secret keys, Alice runs the following algorithm:

1. Select random secret l_p-bit primes p', q' such that both $p = 2p' + 1$ and $q = 2q' + 1$ are also prime. Set the modulus $n = pq$.
2. Chose two random elements $a, a_0 \in QR(n)$.
3. The public key consists of the tuple (n, a, a_0, H).
4. The corresponding secret key consists of (p', q').

To sign a message $m \in \{0,1\}^*$ Alice uses the following algorithm:

1. Choose a prime $e \in \Gamma$ that was not used before.
2. Choose a random integer $r \in \Lambda$.
3. Compute $B = H(m \parallel e \parallel r)$ and $u = (a^B a_0)^{1/e} (\bmod n)$.
4. Output the signature (u, e, r).

Checking whether a tuple (u, e, r) is a valid signature on a message $m \in \{0,1\}^*$ with respect to the public key n can be done as follow:

1. Check whether $(u, e, r) \in \mathbb{Z}_n^* \times \Gamma \times \Lambda$.
2. Compute $B' = H(m \parallel e \parallel r)$.
3. Check whether $u^e \equiv a^{B'} a_0 (\bmod n)$.

If $(u, e, r) \in \mathbb{Z}_n^* \times \Gamma \times \Lambda$ and $u^e \equiv a^{B'} a_0 (\bmod n)$ are valid then the tuple (u, e, r) is a valid signature on a message $m \in \{0,1\}^*$.

In order to make efficient the verifiable encryption of a modification of the Cramer–Shoup's digital signature scheme, we will make use of an initialization phase (Ateniese, 2000) by which the user and the trusted third party T agree on common parameters.

The initialization phase is as follows:

1. Alice sends (n, a, a_0, H) to T, along with a certificate $Cert_A$ (Alice's certificate).
2. T verifies that (n, a, a_0, H) is the public key of Alice and randomly selects a $g \in QR(n)$, such that $y = g^x(\bmod n)$, where x is a secret random element. The trusted third party T signs and sends back $Cert_{TA} = S_T(g, y = g^x, ID_A, (n, a, a_0, H))$, where ID_A is an identity information of a user Alice.
3. Alice gets the certificate $Cert_{TA} = S_T(g, y = g^x, ID_A, (n, a, a_0, H))$ from T, with g of order $p'q'$.

The protocol for verifiable encryption is as follows:

1. Alice encrypts the signature (u, e, r) using the ElGamal encryption scheme with public key $y = g^x(\bmod n)$ as follows. Selects a random r'' and computes $c_1 = uy^{r''}(\bmod n)$ and $c_2 = g^{r''}(\bmod n)$ and show that $D\log_{y^e} y^{er''} = D\log_g g^{r''}$ via $EDLOG(m; y^{er''}, g^{r''}; y^e, g)$. Then Alice sends e, r, c_1, c_2 and $Cert_{TA}$ to Bob.
2. Bob verifies $Cert_{TA}$ and checks that $e \in \Gamma$ and $r \in \Lambda$.
3. Bob computes $B = H(m \parallel e \parallel r)$ and $W = (c_1)^e a_0^{-1} a^{-B}(\bmod n)$.
4. Bob verifies that $D\log_{y^e} W = D\log_g c_2$ via $EDLOG(m; W, c_2; y^e, g)$ (see Definition 1) and it ends the protocol if these are correct.

REMARK 1. The signer must generate a random prime e from the interval Γ with each signature. These prime need not be chosen from the uniform distribution primes. The only requirement is that the probability of generating two equal primes should be negligible (for more details, see (Cramer and Shoup, 2000)). In order to efficient generate these prime we use the Miller–Rabin test (Rabin, 1980) to test for primality. Suppose we choose random numbers from the interval Γ until we have found a number that passes a number of trial divisions and a single Miller–Rabin test. We will make a number of Miller–Rabin tests that reject some composite numbers that pass the trial division test. Once we have found a number that passes a single Miller–Rabin test, we have to perform a number of additional Miller–Rabin tests to reduce the error probability sufficiently.

The security of our verifiable encryption protocol follows from the security of the underlying signature scheme. Next, we prove that the modification of the Cramer–Shoup's digital signature scheme (Cramer and Shoup, 2000) is secure against an adaptively chosen message attack. Being similar to (Gennaro *et al.*, 1999) we require that:

- for every H a collision-resistant hash function, all primes $e \in \Gamma$ and every two messages m_1 and m_2 the distribution $H(m_1 \parallel e \parallel r)$ and $H(m_2 \parallel e \parallel r)$ induced by the random choice of r are statistically close;
- the Strong RSA Assumption holds in a world where there exists an oracle that on input a message m, a prime $e \in \Gamma$ and an $B \in \Lambda$ outputs an $r \in \Lambda$ such that $B = H(m \parallel e \parallel r)$.

Theorem 1. *The signature scheme presented above is secure against adaptively chosen messages attack under the Strong RSA Assumption and the assumption that H is a collision-resistant hash function satisfying the above conditions.*

Proof. This is derived from the proof in (Ateniese *et al.*, 2000). Assume that the attacker A queries signature for K messages and then outputs a signature (u', e', r') on the message m'. We now show that if we take control over the hash function, then we can use this attacker to break the Strong RSA Assumption, i.e., we are given a z and an n and must find an w and v such that $w^v \equiv z(\bmod n)$.

Let $((u_1, e_1, r_1), m_1), ..., ((u_K, e_K, r_K), m_K)$ denote the signature-message pairs that are constructed during the interaction with A. In order for A to be successful its output $((u', e', r'), m')$ must satisfy $(u', e', r') \neq (u_i, e_i, r_i)$ for $1 \leqslant i \leqslant K$. Depending of whether $e_i \nmid e'$ for some i, there are two games to calculate a pair $(w, v) \in \mathbb{Z}_n^* \times \mathbb{Z}_{>1}$ satisfying $w^v \equiv z \ (\bmod n)$ from which we randomly chose one each time then play with the attacker. As mentioned before, we are assuming that there is an oracle that input a message m, a prime $e \in \Gamma$ and a $B \in \Lambda$ outputs an $r \in \Lambda$ such that $B = H(m \parallel e \parallel r)$. The adversary is allowed to query this oracle as well. The first game goes as follows:

1. Select $x_1, ..., x_K \in \Lambda$ and $e_1, ..., e_K \in \Gamma$.
2. Set $a = z^{\prod_{1 \leqslant l \leqslant K} e_l} \bmod n$.
3. Choose a random $r \in \{0, 1\}^{2l_p}$ and set $a_0 = a^r \bmod n$.
4. For all $1 \leqslant i \leqslant K$, compute $u_i = z^{(x_i + r) \prod_{1 \leqslant l \leqslant K; l \neq i} e_l} \bmod n$.
5. Start A, feed it the (u_i, e_i, r_i), where we get r_i from the oracle, and eventually obtain $(B'; [u' = (a^{B'} a_0)^{1/e'} \bmod n, e', r'])$ with $B', r' \in \Lambda$ and $e' \in \Gamma$.
6. If $\gcd(e', e_j) \neq 1$ for all $1 \leqslant j \leqslant K$ output fail and stop. Otherwise, let $\widetilde{e} = (B' + r) \prod_{1 \leqslant l \leqslant K} e_l$. Since $\gcd(e', e_j) = 1$ for all $1 \leqslant j \leqslant K$, we have $\gcd(e', \widetilde{e}) = \gcd(e', (B' + r))$. Hence, by the extended Euclidean algorithm, there exist $\alpha, \beta \in \mathbb{Z}$ such that $\alpha e' + \beta \widetilde{e} = \gcd(e', (B' + r))$. Therefore, letting $w = z^{\alpha} (u')^{\beta} \bmod n$ and $v = e' / \gcd(e', (B' + r)) > 1$ since $e' > (B' + r)$ we have $w^v \equiv z \ (\bmod n)$.

The previous game is only successful if A returns a new signature with $\gcd(e', e_j) = 1$ for all $1 \leqslant j \leqslant K$. We now present a game that solves the Strong RSA Problem in the other case, that is, when $\gcd(e', e_j) \neq 1$ for some $1 \leqslant j \leqslant K$. Note that $\gcd(e', e_j) \neq 1$ means $\gcd(e', e_j) = e_j$ since e_j is prime. The second game goes as follows:

1. Select $x_1, ..., x_K \in \Lambda$ and $e_1, ..., e_K \in \Gamma$.
2. Choose a random $j \in \{1, ..., K\}$ and set $a = z^{\prod_{1 \leqslant l \leqslant K; l \neq j} e_l} \bmod n$.
3. Choose a random $r \in \{0, 1\}^{2l_p}$ and set $u_j = a^r \bmod n$ and $a_0 = u_j^{e_j} / a^{x_j} \bmod n$.
4. For all $1 \leqslant i \leqslant K$, $i \neq j$, compute $u_i = z^{(x_i + e_j r - x_j) \prod_{1 \leqslant l \leqslant K; l \neq i, j} e_l} \bmod n$.
5. Start A, feed it the (u_i, e_i, r_i), where we get r_i from the oracle, and eventually obtain $(B'; [u' = (a^{B'} a_0)^{1/e'} \bmod n, e', r'])$ with $B', r' \in \Lambda$ and $e' \in \Gamma$.
6. If $\gcd(e', e_j) \neq e_j$ output fail and stop. Otherwise, we have $e' = te_j$ for some t and can define $b = (u')^t / u_j \bmod n$ if $B' \geqslant x_j$ and $b = u_j / (u')^t \bmod n$ otherwise. Hence $b \equiv (a^{|B' - x_j|})^{1/e_j} \equiv (z^{|\widetilde{e}|})^{1/e_j} \ (\bmod n)$ with

$\widetilde{e} = (B' - x_j)\prod_{1\leqslant l\leqslant K; l\neq j} e_l$. Since $\gcd(e_j, \prod_{1\leqslant l\leqslant K; l\neq j} e_l) = 1$ it follows that $\gcd(e_j, |\widetilde{e}|) = \gcd(e_j, |B' - x_j|)$. Hence, by the extended Euclidean algorithm, there exist $\alpha, \beta \in \mathbb{Z}$ such that $\alpha e_j + \beta\,|\widetilde{e}| = \gcd(e_j, |B' - x_j|)$. Therefore, letting $w = z^{\alpha} b^{\beta} \mod n$ and $v = e_j / \gcd(e_j, |B' - x_j|) > 1$ since $e_j > |B' - x_j|$, we have $w^v \equiv z \pmod n$.

Consequently, by playing randomly one of game 1 or game 2 with A one can solve the Strong RSA Problem. Since the latter is assumed to be infeasible, it follows that no such attacker can exist.

We compare the modification of Cramer–Shoup's signature scheme, presented above, to the one by Cramer and Shoup (Cramer and Shoup, 2000). The public key size in our signature scheme is smaller than its counterpart in Cramer and Shoup. The latter consists of a tuple (n, h, x, e'), where n is a modulus, h and x are elements of $QR(n)$ and e' is a prime. In contrast, our signature scheme's public key is a tuple (n, a, a_0, H), where n is a modulus, a and a_0 are elements of $QR(n)$ and H is a hash function which is, incidentally, also needed in a Cramer and Shoup public key. Thus, the size difference is due to the prime e' in the latter. A Cramer–Shoup signature is a tuple (y, y', e) where e is a small prime, $y' \in QR(n)$ and $y \in \mathbb{Z}_n^*$, i.e., both are n-bit integers. This is about the same as for our signature scheme. The cost of signing in (Cramer and Shoup, 2000) amounts to generating a prime, computing its inverse and three exponentiations. Hence, the cost of signing is higher in (Cramer and Shoup, 2000) than in our signature scheme.

So, the efficiency of our verifiable encryption protocol is better than that of protocol in (Ateniese, 2000). For instance, compared to the scheme in (Ateniese, 2000), our verifiable encryption protocol is about two times smaller when choosing the same modulus (1200-bit composite modulus n, 768-bit prime modulus p, 160-bit prime modulus q and 128-bit hash function H) for both schemes.

4. Applications

The presented verifiable encryption protocol has numerous applications, including fair exchange protocols, group signatures, key escrow, publicly verifiable secret and signature sharing. For instance, in a fair exchange protocol, two parties Alice and Bob want to exchange some valuable digital data (signatures on a contract, e-cash), but in a fair way: either each party obtains the other's data or neither party does. One way to do this is by employing a trusted party T, but for the sake of efficiency with T only involved in crisis situations. One approach to this problem is to have both parties verifiably encrypt to each other their data under T's public key and only then reveal their data to each other. If one party backs out unexpectedly, the other can go to T to obtain the required data.

In a group signature scheme (Ateniese *et al.*, 2000), (Ateniese and de Medeiros, 2003), (Popescu, 2000) when a user joined a group, whose membership is controlled by the group manager, the user may sign messages on behalf of the group, without revealing his individual identity. However, under appropriate circumstances, the identity of the individual who actually signed a particular message may be revealed, using an entity, called the anonymity revocation manager. Verifiable encryption protocol may be used in the following way as a component in such a system. When a group member signs a message, he encrypts enough information under the public key of the anonymity revocation manager, so that later, if the identity of the signer needs to be revealed, this information can be decrypted. To prove that this information correctly identifies the signer, he makes a Pedersen commitment to this information, proves that the committed value identifies the user, encrypts the opening of the commitment and proves that the cipher text decrypts to an opening of the commitment.

Although one can implement group signatures without it, by using verifiable encryption one can build a more modular system, in which the group manager and anonymity revocation manager are separate entities with independently generated public keys.

5. Conclusions

In this paper we proposed a secure and efficient verifiable encryption protocol to improve the Ateniese's scheme. Our protocol may be used as a building block for designing efficient fair exchange of digital signatures. Furthermore, our verifiable encryption protocol can be applied to group signatures, key escrow and publicly verifiable secret and signature sharing.

Acknowledgements

The author would like to thank prof. Svein J. Knapskog for valuable comments and discussions.

References

Asokan, N., M. Schunter, M. Waidner (1997). Optimistic protocols for fair exchange. In *Proceedings of 4th ACM Conference on Computer and Communication Security*. Zurich, Switzerland. pp. 7–17.

Asokan, N., V. Shoup, M. Waidner (2000). Optimistic fair exchange of digital signatures. *IEEE Journal on Selected Areas in Communications*, **18**(4), 593–610.

Ateniese, G. (2000). Efficient verifiable encryption (and fair exchange) of digital signatures. In *Proceedings of 6th ACM Conference on Computer and Communications Security*. Kent Ridge Digital Labs, Singapore. pp. 138–146.

Ateniese, G., J. Camenisch, M. Joye, G. Tsudik (2000). A practical and provably secure coalition-resistant group signature scheme. In *Proceedings of Crypto 2000*. Santa Barbara, USA. pp. 255–270.

Ateniese, G., B. de Medeiros (2003). Efficient group signatures without trapdoors. In *Proceedings of AsiaCrypt 2003*. Taipei, Taiwan. To appear.

Bao, F., R. Deng, W. Mao (1998). Efficient and practical fair exchange protocols with off-line TTP. In *Proceedings of IEEE Symposium on Security and Privacy*. Oakland, SUA. pp. 77–85.

Baric, N., B. Pfitzmann (1997). Collision-free accumulators and fail-stop signature schemes without trees. In *Proceedings of Eurocrypt'97*. Konstanz, Germany. pp. 480–494.

Camenisch, J., M. Michels (1998). A group signature scheme with improved efficiency. In *Proceedings of Asiacrypt'98*. Beijing, China. pp. 160–174.

Camenisch, J., M. Michels (1999). Separability and efficiency for generic group signature schemes. In *Proceedings of Crypto'99*. Santa Barbara, USA. pp. 106–121.

Camenisch, J., I. Damgaard (2000). Verifiable encryption and applications to group signatures and signature sharing. In *Proceedings of Crypto 2000*. Santa Barbara, USA. pp. 331–345.

Camenisch, J., V. Shoup (2003). Practical verifiable encryption and decryption of discrete logarithms. In *Proceedings of Crypto 2003*. Santa Barbara, USA. pp. 126–144.

Chan, A., Y. Franchel, Y. Tsiounis (1998). Easy come-easy go divisible cash. In *Proceedings of Eurocrypt'98*. Helsinki, Finland. pp. 561–575.

Chaum, D., T. Pedersen (1992). Wallet databases with observers. In *Proceedings of Crypto'92*. Santa Barbara, USA. pp. 89–105.

Cramer, R., V. Shoup (2000). Signature schemes based on the Strong RSA Assumption. *ACM Transactions on Information and System Security*, **3**(3), 161–185.

Even, S., O. Goldreich, A. Lempel (1985). A randomized protocol for signing contracts. *Communications of the ACM*, **28**(6), 637–647.

Fiat, A., A. Shamir (1987). How to prove yourself: practical solutions to identification and signature problems. In *Proceedings of Crypto'86*. Santa Barbara, USA. pp. 186–194.

Fujisaki, E., T. Okamoto (1997). Statistical zero knowledge protocols to prove modular polynomial relations. In *Proceedings of Crypto'97*. Santa Barbara, USA. pp. 16–30.

Fujisaki, E., T. Okamoto (1998). A practical and provably secure scheme for publicly verifiable secret sharing and its applications. In *Proceedings of Eurocrypt'98*. Helsinki, Finland. pp. 32–46.

Gennaro, R., S. Halevi, T. Rabin (1999). Secure hash-and-sign signatures without the random oracle. In *Proceedings of Eurocrypt'99*. Prague, Czech Republic. pp. 123–139.

Pointcheval, D., J. Stern (1996). Security proofs for signature schemes. In *Proceedings of Eurocrypt'96*. Zaragoza, Spain. pp. 387–398.

Popescu, C. (2000). Group signature schemes based on the difficulty of computation of approximate e-th roots. In *Proceedings of Protocols for Multimedia Systems (PROMS 2000)*. Crakow, Poland. pp. 325–331.

Rabin, M.O. (1980). Probabilistic algorithms for testing primality. *Journal of Number Theory*, **12**, 128–138.

Schnorr, C.P. (1991). Efficient signature generation for smart cards. *Journal of Cryptology*, **4**(3), 239–252.

C. Popescu received the MSc degree in computer science from the University of Timisoara, Timisoara, Romania, in 1992. In 1992 he became an assistant professor at the Department of Mathematics, University of Oradea, Oradea, Romania. Since 1998 he has been a lecturer at the Department of Mathematics, University of Oradea. In 2001 he has received the PhD degree in computer science (cryptography) at the Babes–Bolyai University, Cluj Napoca. Since 2003 he has been an associate professor at the Department of Mathematics, University of Oradea. Currently he is a postdoc at Centre for Quantifiable Quality of Service in Communication Systems (Q2S), Norwegian University of Science and Technology, Trondheim, Norway. The centre is appointed Centre of Excellence by the Research Council of Norway. His research interests include cryptography, network security, group signatures and security protocols.

Ateniese patikrinamo kriptavimo protokolo pagerinimas

Constantin POPESCU

Patikrinamas kriptavimas gali būti naudojamas labai efektyvių skaitmeninių parašų apsikeitimo protokolų sudarymui. Tokie protokolai gali būti naudojami kontraktų skaitmeniniam pasirašymui internete. Šiame straipsnyje pristatomas efektyvus skaitmeninių parašų patikrinamo kriptavimo protokolas, kuris pagerina Ateniese patikrinamo kriptavimo schemos saugumą ir efektyvumą.