REVISTA INVESTIGACION OPERACIONAL VOL. 36, NO. 3, 280-291, 2015

NEW SOFTWARES/NOVEDADES DE SOFTWARE

ECIES PROTOCOL WITH ISSUER AUTHENTICATION

Renier Tejera Trujillo[1] y Sacha Pelaiz Barranco[2]
Complejo de Investigaciones Tecnológicas Integradas (CITI). 114, No. 11901, e/ Ciclovía & Rotonda, ISPJAE, Marianao, La Habana, Cuba.

ABSTRACT
A hybrid encryption scheme is a cryptographic algorithm that can be built from two different phases: a key encapsulation mechanism (KEM) based on public key encryption, and a data encapsulation mechanism (DEM) based on a private key encryption scheme. The main advantage of hybrid cryptographic schemes is that they offer solutions to the problem of symmetric key exchange and the problem of efficiency typical of asymmetric encryption schemes. An example of a hybrid cryptographic algorithm is the Elliptic Curve Integrated Encryption Scheme (ECIES). ECIES is incorporated as international standard, however this protocol does not provide issuer authentication which is a drawback in many practical applications. This paper presents a secure, efficient and authenticated variant of ECIES protocol. The modification is based on the substitution of the mechanism for generating shared secret values included in the ECIES, for a one-pass authenticated key agreement protocol. This protocol is included in a cryptographic file-protection system which is used today in many practical applications and scenarios.

KEY WORDS: communication protocol, hybrid encryption scheme, key exchange protocol.

MSC: 68Q99

RESUMEN

Un esquema de cifrado híbrido es un algoritmo criptográfico que puede ser construido a partir dos fases bien definidas: un mecanismo de encapsulamiento de llave (KEM por sus siglas en inglés) basado en un esquema de cifrado de llave pública, y un mecanismo de encapsulamiento de datos (DEM por sus siglas en inglés) basado en un esquema de cifrado de llave privada. La ventaja principal de los esquemas criptográficos híbridos radica en que ofrecen solución al problema del intercambio de llaves simétricas y al problema de eficiencia propio de los esquemas asimétricos de cifrado. Un ejemplo de algoritmo criptográfico híbrido es el Esquema Integrado de Cifrado sobre Curvas Elípticas (ECIES por sus siglas en inglés). El ECIES se ha constituido como estándar internacional, sin embargo, este protocolo no ofrece autenticidad del emisor lo que representa un inconveniente en la práctica en muchas aplicaciones.
En este trabajo se propone una variante segura, eficiente y autenticada del protocolo ECIES. La modificación realizada se basa en la sustitución del mecanismo de generación de la llave secreta compartida incluido en el ECIES, por un protocolo autenticado de acuerdo de llaves de un solo paso. Esta variante del protocolo ECIES se integra a un sistema de protección criptográfica de ficheros que está siendo usado en varias aplicaciones prácticas.

1. INTRODUCTION

Security protocols aim to allow two or more entities to establish a secure communication over a hostile network. The goal of security protocols is to allow the exchange of sensitive data over insecure networks. Without doubt cryptography plays a very important role in the design of security protocols. The appearance of asymmetric cryptography [11] as a solution to the problem of key exchange leads its inclusion as a fundamental component in the development of cryptographic protocols designed to ensure one or more

[1] rtejera@udio.cujae.edu.cu
[2] spelaiz@udio.cujae.edu.cu

security objectives. The key exchange is a process in which two honest entities share a secret key that will be used to communicate via an insecure communication channel. The key exchange protocols are divided in two classes: key-agreement protocols, in which both parties contribute with information from which the secret key is generated and key-transport protocols in which a secure secret key is transmitted from one entity to another. The one-way communication models are a popular alternative used today (e.g. email). In these models, the issuer sends a single message with all necessary information to establish a communication with the recipient. These models requires the existence of a one-pass key exchange protocol, so it is important to have alternatives to wrap the secret key from an issuer entity A to recipient entity B.
Asymmetric encryption provides secure communication without previous agreement using the static public keys of the entities involved. However, its proven inferiority in computational performance compared to symmetric algorithms has leaded to the design of hybrid cryptosystems. Hybrid cryptosystems complements public key and private key cryptosystems taking both advantages for best results. A widely accepted example of hybrid cryptographic algorithm is Diffie-Hellman Integrated Encryption Scheme (DHIES) [1]. The term integrated is due to the use of asymmetric cryptography to wrap a secret key used by a symmetric encryption algorithm. The recent interest in Elliptic Curve Cryptosystems [17, 21] leads to the appearance of Elliptic Curve Integrated Encryption Scheme (ECIES) as a DHIES adaptation of the groups formed by the points of an elliptic curve defined over a finite field with an operation of point's addition acting as internal law.
When using ECIES to key wrapping (the message to encrypt is a symmetric key) this can be seen as a one-pass key exchange protocol that guarantees implied authentication of the recipient entity. However it is very desirable to have protocols where both the issuer and recipient are authenticated. One possible solution may be to apply a digital signature algorithm (e.g. ECDSA) to the encrypted message. However, this introduces the disadvantage that it is necessary to implement a digital signature scheme and also increasing processing and some transmission requirements. Even assuming the use of digital signatures could not guarantee some security objectives because an adversary can replace the transmitted digital signature by his own.

Approach: This paper proposes an authenticated variant of ECIES. The modification consists in the inclusion of the Elliptic Curve variant of the scheme MQV (ECMQV) to ensure implicitly issuer authentication during the process of generating the shared secret key.

Related works: MQV protocol was formalized by Law, Menezes, Qu, Solinas and Vanstone [19] and proposes one-pass, two-pass and three-pass variants, the latter with key confirmation. A paper presented by Kaliski [7] described the unknown key-share attack (UKSA) for this protocol and proposes a set of measures to avoid it, showing that the three-pass variant of MQV is resistant to these attacks. Hugo Krawczyk [18] presented a set of vulnerabilities for MQV relating to the security model Canetti-Krawczyk [8] and also formalized a variant known as HMQV (hashed MQV). However these vulnerabilities either are well known and avoidable or depend on the ability of an adversary to know secret information of the entities involved, which in practice is unlikely. Menezes [20] criticized the formal analysis performed by Krawczyk and raised the need to include validation of ephemeral public keys and static entities in case of HMQV. It also showed that the one-pass variant of HMQV is vulnerable to UKSA.

Two years later was published a symmetric key wrapping proposal [13] which include one-pass HMQV (HOMQV) to achieve issuer authentication. This selection was based on efficiency, security and flexibility of the HOMQV protocol. It is known that the HMQV protocol is more efficient than MQV [18], in case where is not necessary to verify that the ephemeral keys belong to the prime order group G, employed by the protocol (G-Test in accordance with the notation used by Krawczyk) and similar behavior for the remaining cases. Following the observation by Menezes of the importance of including the G-Test, Krawczyk [18] recognized that this validation is required for one-pass variant HOMQV protocol. From these results it can be assumed that for the present case (a one-way communication model) the one-pass variants of MQV and HOMQV have the same efficiency in terms of computational performance. The added HOMQV calculate a hash value that can be considered negligible with respect to operations on group elements.

2. DOMAIN PARAMETERS AND KEY PAIR GENERATION

This section briefly describes the elliptic curve domain parameters that are common to both entities involved in the protocol (i.e., the domain parameters), and the key pairs of each entity. Also it describes the schemes Elliptic Curve Diffie-Hellman (ECDH) and ECMQV.

Elliptic curve domain parameters

The domain parameters for cryptographic schemes using elliptic curves defined over prime finite fields F_q are a sextuple $\{q, a, b, P, n, h\}$ conformed as:

1. q: a prime power.
2. a, b: two field elements in F_q which define the equation $y^2 = x^3 + ax + b$ of the elliptic curve over F_q.
3. $P = (x_P, y_P)$: a finite point of prime order in $E(F_q)$.
4. n: the order of the point P.
5. h: the cofactor $h = \#E(F_q)/n$.

The selection of adequate domain parameters is very important in order to avoid many attacks could solve the Elliptic Curve Discrete Logarithm Problem ECDLP. For more information see [14].

Key pair generation

Given a valid set of domain parameters $\{q, a, b, P, n, h\}$, an entity A's private key is an integer d_A selected at random from the interval $[1, n-1]$. A's public key is the elliptic curve point $Q_A = d_A P$. The key pair of entity A is (Q_A, d_A). Similarly the key pair of entity B is (Q_B, d_B).

Elliptic Curve Diffie-Hellman (ECDH)

Diffie-Hellman (DH, ECDH if using elliptic curves) is perhaps the simplest protocol of key agreement. In ECDH scheme 2 entities agree a secret key shared over an insecure communication channel in the following way:

Set $\{q, a, b, P, n, h\}$ the domain elliptic curve parameters shared by A and B:

1. A selects a random integer $k \epsilon [1, n-1]$.
2. A computes the elliptic curve point kP.
3. A sends kP to B.
4. B selects a random integer $k' \epsilon [1, n-1]$.
5. B computes the elliptic curve point $k'P$.
6. B sends $k'P$ to A.
7. A can compute $K_A = k(k'P) = kk'P$ and B can compute $K_B = k'(kP) = k'kP$.

By commutative $K_A = K_B$ so both entities have shared the same secret key. The ECDH scheme is vulnerable to man-in-the-middle attacks due to the absence of entities authentication. Various versions have been proposed to solve this problem such as the Menezes-Qu-Vanstone (MQV) scheme, ECMQV when using elliptic curves.

ECMQV

Set $\{q, a, b, P, n, h\}$ the domain elliptic curve parameters shared by A and B and set (Q_A, d_A) and (Q_B, d_B) their static key pairs and also set (kP, k) and $(k'P, k')$ their ephemeral key pairs. A do the following:

1. $l = \lceil \log_2 n \rceil / 2$.
2. A computes $u = (x_{kP}) \, mod \, 2^l + 2^l$.
3. A computes $s = (k + u d_A) \, mod \, n$.
4. A computes $v = (x_{k'P}) \, mod \, 2^l + 2^l$.
5. A computes $Z = s(k'P + vQ_B)$.
6. If $Z = O$ generate again A and B ephemeral keys kP and $k'P$.
7. return Z.

B can computes the same value of Z using previous algorithm just changing (d_A, k, Q_B, k') by (d_B, k', Q_A, k). If the values u, v, s computed by entity A are denoted by u_A, v_A, s_A and u_B, v_B, s_B the values computed by entity B we can see that $u_A = v_B$ and $v_A = u_B$. So:

$$Z = s_A(k'P + v_A Q_B)$$
$$Z = s_A(k' + v_A d_B)P$$
$$Z = s_A(k' + u_B d_B)P$$
$$Z = s_A s_B P$$

The values s_A and s_B acts as implicit digital signatures of A and B. These values can be saw as digital signatures because the unique entity that can compute s_A is A and the same for B. The implicit sense is due because B indirectly verify the validity of s_A computing $(kP + v_B Q_A) = s_A P$.

3. ELLIPTIC CURVE INTEGRATED ENCRYPTION SCHEMA, ECIES

The ECIES [15, 3, 4, 9] was proposed by Bellare and Rogaway like a variant of ElGamal asymmetric encryption system [12]. In ECIES a shared secret value is used to derive two symmetric keys (k_1, k_2). k_1 is used to encrypt the input message using a symmetric encryption algorithm and k_2 is used to authenticate the encrypted message. ECIES uses the following cryptographic primitives:

$KDF(S, l_k)$: Key derivation function constructed from a hash function. Receives as input a sequence S and returns a l_k bits key.

> $Enc_k(M)$: Symmetric encryption algorithm. Receives as input a message M and a secret key k and produces as output an encrypted message c ($c = Enc_k(M)$). The inverse (decryption) is denoted by $Dec_k(c)$ and it holds that $M = Dec_k(Enc_k(M))$.
>
> $MAC_k(M)$: Message authentication code. It is constructed from a hash function. Receives as input a message M and a secret key k and returns an authentication code.

During ECIES execution recipient B has a static public-private keys pair (Q_B, d_B) with $Q_B = d_B * P$ (P generator of prime order). To encrypt a message M and send it to B, A selects a random secret value $r_A \in_{\mathbb{R}} [1, n-1]$ (n order of P) and calculates the ephemeral public key $R_A = r_A * P$. Then computes the Diffie-Hellman value $Z = r_A * Q_B$ and from it derives the key $k = k_1||k_2$ using a key derivation function $k = KDF(R_A||Z, l_k)$. Then obtains the cipher text $c = Enc_{k_1}(M)$ using a symmetric algorithm with a secret key k_1, and an authentication tag from the encrypted secret key $t = MAC_{K_2}(c)$. A sends to B the message (R_A, c, t) who recovers the Diffie-Hellman value $Z = R_A * d_B$, derives the private key $k = k_1||k_2$, verifies the authentication tag t and decrypts the cipher text c to obtain the original message M ($M = Dec_{k_1}(c)$). Note that as additional element the R_A point can be compressed [14] leading to a reduction in the length of the information exchanged.
The ECIES security has been extensively analyzed. Abdalla, Bellare and Rogaway [1] made three variants of computational and decision Diffie-Hellman problems (CDHP and DDHP), valid on elliptic curves points groups (ECCDHP and ECDDHP), whose intractability from the computational viewpoint is sufficient to demonstrate their security. Cramer and Soup [10] demonstrated the ECIES security in random oracle model [16] under the assumption that the ECCDHP is intractable even when known an efficient algorithm to solve the ECDDHP. Other safety tests have been carried out from the assumption that the KDF and MAC functions are safe. The problem is that the safety tests are valid only under a precise set of constraints. For this reason it is necessary to pay attention principally to two factors that can directly threaten the security of the protocol.
Benign malleability: It has been standardized that a KDF function applies only to the x coordinate of the curve point Z and not to the entire point. This option is more efficient but causes the scheme suffers from a condition called benign malleability [22]. The problem is that an adversary is able from a cipher text c to produce a different valid cipher text c' for the same secret key k_1. For example $c = (R_A, c, t)$ and $c' = (-R_A, c, t)$ would be valid cipher texts from the same session key. This is because the elliptic curve points: $R_A = (x_A, y_A)$ and $-R_A = (x_A, -y_A)$ have the same x-coordinate. The problem of benign malleability causes

the protocol is not secure against attack formal definition of adaptive chosen cipher text [14]. During the protocol description is presented a KDF function variant applying both secret value Z as ephemeral public key R_A. This variant is designed to avoid the problem of benign malleability.
Variable symmetric key length: Symmetric keys k_i derived by the KDF function must be of fixed length (l_k). The key derivation function computes the value selected by taking the first L bits of a pseudo-random sequence produced from the input. This means that for any $0 < l'_k < l_k$, $KDF(R_A||Z, l'_k)$ is equal to the first l'_k bits to $KDF(R_A||Z, l_k)$.
A current trend is to apply the methodology KEM / DEM [2] to the public key encryption schemes being divided into key encapsulation mechanism (KEM) and data encapsulation mechanism (DEM). This procedure allows the analysis of two well defined states, which facilitates the study of the security of the protocol. It can be seen that the algorithm ECIES (and DHIES in general) is one example of paradigm KEM/DEM where R_A is generated during encapsulation mechanism keys and (c, t) is the result of the data encapsulation mechanism. From this observation and applying the methodology KEM/DEM, ECIES can be defined as:

ECIES-KEM/DEM

Input: Issuer static private key Q_B and message M.
Output: Cipher Text C.

1. $(R_A, Z) \leftarrow \text{ECIES/KEM}(Q_B, l)$.
2. $(c, t) \leftarrow \text{ECIES/DEM}(Z, M)$.
3. $C \leftarrow (R_A||c||t)$.

Due to the possibility of modifying the flow bypass secret session key in ECIES protocol to include issuer implicit authentication, which is the main objective of this work, it is important to analyze the KEM phase.

ECIES/KEM

The key encapsulation mechanism for ECIES (ECIES / KEM) is a probabilistic algorithm that takes as input the recipient B static public key (Q_B) and uses it for the derivation of the session secret key by using a KDF function. The algorithm works as follows:

1. A generate a random value $r_A \in_{\mathbb{R}} [1, n-1]$.
2. A compute $R_A = r_A * P$.
3. A compute $Z = r_A * Q_{B.}$
4. A obtain $k = KDF(R_A||Z, l_k)$.

The output is the key wrapping pair (k, R_A). In other way, reverse deterministic algorithm receives as input the static key pair (d_A, R_A). Uses the same KDF function and operates in the following manner:

1. B validate the ephemeral public key (R_A).
2. B compute $Z = R_A * d_B$.
3. B obtain $k = KDF(R_A||Z, l_k)$.

During the encapsulation phase the issuer executes the one-pass key agreement protocol Diffie-Hellman over elliptic curves (ECDH) [11]. As mentioned before this protocol does not guarantee the issuer implicit authentication (A to B) so it is proposed to use instead the protocol ECMQV to meet this security objective.

4. AUTENTICATED KEY AGREEMENT PROTOCOL OVER ELLIPTIC CURVE, ECMQV

The authenticated key agreement protocol ECMQV is a variant of MQV protocol over Elliptic Curve. It has been standardized [15, 4, 9] and is recognized as one of the most efficient extensions for key agreement Diffie-Hellman protocol [11]. In 2005, the NSA (National Security Agency) selected it as key exchange mechanism for the protection of "... critical or classified national security information..." [23]. The fundamental contribution made by ECMQV is to ensure issuer authentication by including in the process of generating the shared secret key Z the static public keys of both involved entities A and B (Q_A, Q_B) . This

solves the main difficulty of the Diffie-Hellman protocol preventing susceptible to attacks such as man-in-the-middle attacks. In this paper the interest is the one-pass variant (oECMQV, one-pass ECMQV) by the analyzed communication model.
During the oECMQV protocol description is used the following notation: l ($l = \lceil \log_2 n \rceil$) denotes the bit length of n. If Q is a point of an elliptic curve then $\overline{Q}$ is defined as $\overline{Q} = \overline{x} \bmod 2^{l/2} + 2^{l/2}$ where $\overline{x}$ is selected by the field elements representation $\mathbb{F}_q$ of the x-coordinate of the elliptic curve point Q. Note that $\overline{Q} \bmod n \neq \theta$.
oECMQV: let the entities A y B whit his static public-private key pairs (Q_A, d_A), (Q_B, d_B). To exchange a secret key with the recipient B, the issuer A generates a secret random value $r_A \in_{\mathbb{R}} [1, n-1]$, then computes the value $R_A = r_A * P$ and sends it to B. A computes $s_A = (r_A + \overline{R_A} d_A) \bmod n$ checking that $Z_A \neq \theta$. Meanwhile the receiver B performs a validation of the ephemeral key R_A (if invalid execution aborts), then computes $s_B = (1 + \overline{Q_B}) d_B \bmod n$ and $Z_B = h s_B * (R_A + \overline{R_A} * Q_A)$ with $Z_B \neq \theta$. The shared secret value is $Z_A = Z_B$. From this value is derived the shared secret key k using a KDF function. This is necessary because the shared secret value can have weak bits (bits of information on Z that can be predicted with a non-negligible advantage).

Security Notes

The security of oECMQV protocol has not been demonstrated in a distributed computing model only heuristics suggest that owns mutual digest authentication [5]. A fundamental aspect inside ECMQV protocol (for all variants) is the ephemeral public key validation. This value is generated for each execution of the protocol and must be validated as a mechanism to prevent Small Subgroup Attacks and Curve Invalid Attacks. Although this observation may be noted in the analyzed case is not necessary to validate $n * R_A = \theta$. This validation is intended to verify that the ephemeral public key (R_A) does not belong to a subgroup of G (prevents attack by subgroup). For the specific case where the cofactor h = 1 there exists a subgroup minor G' of G generated by P so just check that (R_A) is a finite element of G ($R_A \neq \theta$). Generally the condition to be met $h \leq 4$ the small subgroup attack is not effective since an attacker has very few options to select a subgroup in G which leads to very few may know bits of the private key attacked entity. An additional element adopted by the ECMQV protocol to avoid this check is to ensure that the shared secret value Z is an element of the finite group G generated by P. This is accomplished by including the cofactor in the calculation of the points Z_A,Z_B. The condition $Z_A = Z_B \neq \theta$ guarantees a finite order element in G.

The oECMQV ensures the commitment of r_A secret value does not affect the security of the protocol as the shared secret key k depends on the static private key d_A of the issuer entity A (this property will be denoted by r_A-Security). A result of this observation is that it can be possible to pre-calculate the values $(r_A, r_A * P)$. Some security attributes for a key agreement protocol are described in [6] and are accepted by the cryptographic community:

Perfect forward secrecy (PFS): in the case of one-pass protocols this property cannot be ensured in general because the recipient does not intervene with additional secret information in the process of key generation. This causes that if recipient's static private key is exposed the attacker can decrypt all the information transmitted. The oECMQV protocol only guarantees the issuer forward secrecy (SFS) as an attacker obtain the issuer's static private key cannot know the ephemeral secret keys because their calculation involving a secret value r_A that change for each session.

Known Key: This attribute is not guaranteed since the recipient no contributes with secret information during the execution of the protocol. In this way an attacker from the knowledge of a prior session key can use it to change the information transmitted during the session and forward it to the recipient.

Unknown key share (UKS): The information provided by Kaliski [7] exposed the vulnerability of the protocol against these attacks. However this is avoided including the use of a KDF function in the derivation process of the session secret key value k from Z and also the identities of both involved parties $(\widehat{A}, \widehat{B})$.

Performance Notes

The oECMQV protocol guarantees the attributes: little overhead in communication (related to the number of bits transmitted) and minimum number of messages [6]. Concerning computer performance it can be noted that the predominant operation in oECMQV is the scalar multiplication. In the issuer side is needed to compute $r_A * P,\ hs_A(1+\overline{Q_B}) * Q_B$ and in the recipient side is needed to compute $hs_B * (R_A + \overline{R_A} * Q_A)$. It may be noted that the values $\overline{Q_B},\ \overline{R_A}$ only use half of the bits of the x- coordinate of the elliptic curve points Q_B, R_A respectively. This increases the efficiency of the protocol because for the calculation of $\overline{R_A} * Q_A$ it is only consumed half of the execution time compared to an operation to take all the bits of the integer.
If it is taken as a comparison parameter for scalar multiplication operation the length of the scalar integer involved it can be said that the operation $r_A * P$ represents 1l scalar multiplication (where l is the length in bits of the integer r_A). In this way the operation $\overline{R_A} * Q_A$ represents 0,5l scalar multiplications. From this notation can be stated that during the execution of the oECMQV protocol the issuer requires 2l scalar multiplications while the recipient performs 1,5l scalar multiplication in the process of calculating the agreed value. During validating the ephemeral public key R_A the scalar multiplication can be removed.

5. ECIES-A/KEM

The key element of this proposal is to change the KEM phase of the protocol ECIES by adapting the protocol oECMQV as authenticated for the process of generating the session secret key. Thus is obtained a ECIES-A scheme (ECIES authenticated) that during the encapsulation process (ECIES-A/KEM) operates as follows:

1. A generate a random value $r_A \in_{\mathbb{R}} [1, n-1]$.
2. A compute $R_A = r_A * P$.
3. A compute $s_A = (r_A + \overline{R_A} d_A) \bmod n$.
4. A compute $Z = hs_A(1+\overline{Q_B}) * Q_B$ con $Z \neq \theta$.
5. A obtain $k = KDF(R_A || Z || \widehat{A} || \widehat{B}, l)$.

The output is the encapsulated key pair (k, R_A). The deterministic algorithm for the inverse process receives as input the key pair (d_A, R_A) and performs the following operations:

1. B Validate the ephemeral public key (R_A)
2. B compute $s_B = (1+\overline{Q_B}) d_B \bmod n$
3. B compute $Z = hs_B * (R_A + \overline{R_A} * Q_A)$
4. B compute $k = KDF(R_A || Z || \widehat{A} || \widehat{B}, l)$

The computational performance differences between the variants ECIES/KEM and ECIES-A/KEM is minimal. Table 1 [13] shows a comparison of KEM phases between ECIES, HOMQV and ECIES-A variants.

Table 1. Comparison of KEM Phases for asymmetric encryption schemes on Elliptic Curves.

KEM phase	$A \to B$	Implicit - Authentication	$r_A -$ Security	Forward Secrecy (issuer)	# Scalar Multiplications (A/B)
ECIES	R_A	Only B	No	No	$2l/1l$
ECIES-A	R_A	A,B	Si	Si	$2l/1{,}5l$
HOMQV	R_A	A,B	Si	Si	$2l/1{,}5l$

The last column shows the number of l scalar multiplications required to execute the protocol for the issuer (A) and the recipient (B). Considering the possibility of pre computed values $(r_A, r_A * P)$ decreases in 1 scalar multiplication issuer requirements for ECIES-A/KEM and HOMQV cases.

6. ECIES-A

To complete the proposed hybrid encryption scheme is necessary to include the original ECIES/DEM phase. In this way a hybrid encryption scheme with explicit authentication is possible (ECIES-A).
ECIES-A KEM/DEM

Input: Recipient static public key Q_B and message M.
Output: Cipher Text C.

1. $(R_A, Z) \leftarrow$ ECIES-A/KEM(Q_B, l).
2. $(c, t) \leftarrow$ ECIES/DEM(Z, M).
3. $C \leftarrow (R_A||c||t)$.

ECIES-A Advantage

$A^{(Q_A, d_A)}$

$r_A \in_{\mathbb{R}} [0, n-1]$.

$R_A = r_A * P$.

$l = \lceil \log_2 n \rceil / 2$.

$u_A = (\,\overline{R_A}\,) \mathrm{mod} 2^l + 2^l$.

$s_A = (r_A + u_A d_A) \mathrm{mod}\, n$.

$v_A = (\,\overline{Q_B}\,) \mathrm{mod} 2^l + 2^l$.

$Z_A = s_A(1 + v_A) * Q_B, Z \neq \theta$.

$k_1||k_2 = \mathrm{KDF}(R_A||Z_A||\hat{A}||\hat{B}, l_k)$.

$B^{(Q_B, d_B)}$

$$\xrightarrow{\hspace{8cm}}$$

$$A \rightarrow B: \left(R_A, c = \mathrm{Enc}_{k_1}(M), t = \mathrm{HMAC}_{k_2}(c)\right).$$

$l \leftarrow \lceil \log_2 n \rceil / 2$.

$u_B \leftarrow (\,\overline{Q_B}\,) \mathrm{mod} 2^l + 2^l$.

$s_B \leftarrow (1 + u_B) d_B \mathrm{mod}\, n$.

$v_B \leftarrow (\,\overline{R_A}\,) \mathrm{mod} 2^l + 2^l$.

$Z_B \leftarrow s_B * (R_A + v_B * Q_A)$.

$k_1||k_2 \leftarrow \mathrm{KDF}(R_A||Z_B||\hat{A}||\hat{B}, l_k)$.

$t' \leftarrow \mathrm{HMAC}_{k_2}(c)$.

$[t == t']\; M \leftarrow \mathrm{Dec}_{k_1}(c)$.

Figure 2 ECIES-A Protocol

The main advantage of ECIES-A is that provides implicit authentication of both entities involved (issuer and recipient). This eliminates the difficulties associated with the schemes that do not provide issuer authentication. The modification is based on the use of a standardized one-pass key agreement protocol ECMQV. This protocol has been widely discussed by the international cryptographic community and is recognized as one of the most efficient and safe variants of the Diffie-Hellman protocol. A modification of the protocol ECIES based on cryptographic primitives included in international standards facilitates its analysis and implementation into practical solutions.

A very attractive aspect of this variant is given by its computational efficiency. The ECIES-A maintains the same number of bits and messages exchanged during a communication that the original variant ECIES. It is resistant to lack of shared key attacks, invalid curve attacks and small subgroup attacks, without this representing an increase in terms of computational cost and presents a very similar performance to the ECIES (Table 1).

ECIES-A includes a key agreement protocol which guarantees r_A-security making possible to pre-calculate the (r_A, R_A) values so a scalar multiplication can be removed (from the issuer) in protocol execution. This aspect allows you to implement a solution based on ECIES-A with better computational performance in the issuer side than the original ECIES.
In detail the protocol ECIES-A is presented in Figure 1.

7. IMPLEMENTATION

This section presents some experimental results in terms of computational time. All testing was done on a personal computer with Intel Core 2 Duo microprocessor, 2.66 GHz and 2 Gbytes of RAM.

Modular arithmetic

Despite there are many C++ library for cryptography such as Crypto++, LiDIA and MIRACL, we developed our own C++ library named BigInt for modular arithmetic. The performance of BigInt in terms of computational time is good enough for our purposes. As example in tables 2 and 3 we present some experimental results for modular exponentiation and modular inverse operations and the comparison with MIRACL v5.4.1library. The time is measured in milliseconds and we used the tool Intel Parallel Studio 11.

Table 2. Modular exponentiation performance.

Bit length	MIRACL	BigInt
160	0.3	0.3
256	0.8	0.6
384	2.2	2.2
512	6.6	4.7

Table 3. Modular inversion performance.

Bit length	MIRACL	BigInt
512	0.1	0.4
1024	0.2	1.1
2048	0.7	3.2

Elliptic Curves arithmetic

For elliptic curve arithmetic we developed a C++ library EllipticCurve. We used elliptic curves in Weierstrass form defined over prime finite fields. For elliptic curve points representation we used standard projective coordinates [14] and for scalar multiplication which is the main operation in elliptic curve arithmetic in terms of computational time we implemented a NAF binary method [14]. The NAF method is not the best method for scalar multiplication but is good enough for our purposes. As example table 4 presents the performance in milliseconds of scalar multiplication operation and the comparison with MIRACL library.

Table 4. Scalar multiplication performance.

Bit length	MIRACL	BigInt
160	3	17
256	9	55
384	29	148
512	66	309

ECIES-A

The ECIES-A protocol was also fully implemented in C++ language and was integrated in a cryptographic file-protection system also developed by the authors of this paper. This system are used with very good results in many practical applications and is projected its generalization in many others applications and scenarios. As example table 5 presents the performance in seconds of the ECIES-A protocol.

Table 5. ECIES-A performance.

Bit length	Sender A	Receipt B
160	0.066	0.065
256	0.181	0.180
384	0.499	0.493
512	1.004	1.002

8. CONCLUSIONS

The ECIES-A protocol is an authenticated hybrid cryptosystem option. Its use as a key wrapping protocol ensures secure communication without prior agreements. The security of ECIES-A is supported in the deep analysis of the cryptographic standards ECIES and ECMQV, especially its one-pass variant (oECMQV). The ECIES-A is a secure and efficient cryptographic scheme becoming a very attractive option for implements in models of one-way communication.

RECEIVED JANUARY, 2014
REVISED JANUARY, 2015

REFERENCES

[1] ABDALLA, M., BELLARE, M. and ROGAWAY, P. (2001): DHIES: An encryption scheme based on the Diffie-Hellman Problem. Full version of current paper, available from authors' web pages. www.di.ens.fr/~mabdalla/papers/dhes.pdf

[2] ABE, M., GENNARO, R. AND KUROSAWA, K. (2005): Tag-KEM/DEM: a New Framework for Hybrid Encryption and a New Analysis of Kurosawa-Desmedt KEM. In **Proc. Eurocrypt'2005**, 128-146, Springer-Verlag, Aarhus, Denmark.

[3] ANSI X9.62 (1999): Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algortihm (ECDSA). American National Standards Institute. Available from http://webstore.ansi.org/

[4] ANSI X9.63 (2001): Public Key Cryptography For The Financial Services Industry: Key Agreement and Key Transport Using Elliptic Curve Cryptography. American Nacional Standards Institute. Available from http://webstore.ansi.org/

[5] ANTIPA, A., BROWN, D., MENEZES, A**.** et al. (2003): Validation of Elliptic Curve Public Keys. In **Proceedings of the 6th International Workshop on Theory and Practice in Public Key Cryptography**, 211-223, Springer-Verlag, Miami, USA.

[6] BLAKE-WILSON, S., JOHNSON, D. and MENEZES**,** A. (1997): Key Agreement Protocols and their Security Analysis. In **Proc. of Sixth IMA International Conference on Cryptography and Coding**, 30-45, United Kingdom.

[7] BURTON S. and KALISKI, JR**.** (2001): An unknown key-share attack on the MQV key agreement protocol. **ACM Transactions on Information and System Security (TISSEC**), 4, 275-288.

[8] CANETTI, R. and KRAWCZYK, H**.** (2001): Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels. In **Advances in Cryptology-EUROCRYPT'2001**, LNCS 2045, 453-474, Innsbruck, Austria.

[9] CERTICOM (2002): SEC 1: Elliptic Curve Cryptography. Standards for efficient cryptography. Avalaible from http://www.secg.org/sec1-v2.pdf

[10] CRAMER, R. and SHOUP**,** V. (2003): Design and Analysis of Practical Public-Key Encryption Schemes Secure against Adaptive Chosen Ciphertext Attack. **SIAM Journal on Computing**, 33, 167-226.

[11] DIFFIE, W. and HELLMAN, M**.** (1976): New Directions in Cryptography. **IEEE Transactions on Information Theory,** 22, 644 - 654.

[12] ELGAMAL, T. (1985): A public key cryptosystem and a signature scheme based on discrete logarithms. **IEEE Transactions on Information Theory**, 31, 469-472.

[13] HALEVI, S. and KRAWCZYK, H. (2011): One-Pass HMQV and Asymmetric Key-Wrapping. In **Public Key Cryptography–PKC'2011**, 317-334, Springer Berlin/Heidelberg, Taormina, Italy.

[14] HANKERSON, D., MENEZES, A. and VANSTONE, S**.** (2004): **Guide to Elliptic Curve Cryptography**. Springer Verlag, New York.

[15] IEEE P1363A. (2004): Standard Specifications for Public-Key Cryptography-Amendment 1: Additional Techniques. Avalaible from http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1335427

[16] KATZ, J. and LINDELL, Y., (2014): **Introduction to Modern Cryptography.** Chapman and Hall/CRC Press, 2 edition. Florida, USA.

[17] KOBLITZ, N. (1987): Elliptic Curve Cryptosystems. **Mathematics of Computation**, 48, 203-209.

[18] KRAWCZYK, H**.**, (2005): HMQV: A High-Performance Secure Diffie-Hellman Protocol. In **Advances in Cryptology-CRYPTO '05, LNCS 3621**, 546-566, Santa Barbara, USA.

[19] LAW, L., MENEZES, A., SOLINAS, J. et al**.** (2003): An efficient protocol for authenticated key agreement. **Designs, Codes and Cryptography**, 28, 119-134.

[20] MENEZES, A. (2005): Another Look at HMQV. **Journal of Mathematical Cryptology**, 1, 47-64.

[21] MILLER, V.S**.** (1985): Use of elliptic curves in cryptography. In **Advances in Cryptology-Proceedings of CRYPTO'85, LNCS 218**, 417-426, Santa Barbara, USA.

[22] SHOUP, V. (2011): A Proposal for an ISO Standard for Public Key Encryption. Cryptology ePrint Archive, Report 2001/112. Avalaible from https://eprint.iacr.org/2001/112.pdf

[23] STASAK, J. (2004): **NSAs Elliptic Curve Licensing Agreement**. Presentation to the IETF´s Secutity Area Advisory Group, USA.