The Open Mathematics Journal, **2013, 6, 1-8**

A Comparative Analysis of Hybrid Encryption Schemes Based on Elliptic Curves

V. Gayoso Martínez, L. Hernández Encinas* and A. Martín Muñoz

Department of Information Processing and Coding, Information Security Institute (ISI), Spanish National Research Council (CSIC), Madrid, Spain

Abstract: Elliptic Curve Cryptography (ECC) can be used as a tool for encrypting data, creating digital signatures, and performing key exchanges. Regarding the encryption capability, the first schemes that appeared were either versions of already existing public-key algorithms (Massey-Omura and ElGamal) or new schemes specified ad hoc (Menezes-Vanstone). However, all the initial elliptic curve encryption schemes had similar problems, and thus were conveniently discarded by the academic community. The encryption schemes currently used are known as hybrid cryptosystems, as they use both symmetric and asymmetric techniques. Among those hybrid cryptosystems based on ECC, the best known ones are the Elliptic Curve Integrated Encryption Scheme (ECIES), the Provably Secure Elliptic Curve encryption scheme (PSEC), and the Advanced Cryptographic Engine (ACE).

In this work, we present an extensive review of the basic concepts of elliptic curves, the initial ECC encryption algorithms, and the current ECC hybrid cryptosystems. After that, we provide a comprehensive comparison of ECIES, PSEC, and ACE, highlighting the main differences between them. Finally, we conclude that, with the available data, it can be stated that ECIES is the best ECC encryption scheme from a performance and ease of implementation point of view.

Keywords: Elliptic curves, encryption, hybrid cryptosystem, public-key cryptography, ECIES, PSEC, ACE, Java Card.

1. INTRODUCTION

In 1987, Neal Koblitz proposed the use of elliptic curves over finite fields in order to implement some cryptosystems that were previosly specified for the multiplicative group of a finite field [1]. In the sections referring to the equivalent of the ElGamal algorithm, Koblitz detailed the procedure and computations to be performed with the points of an elliptic curve, including examples about how to choose those points. Additionally, Koblitz described how the Diffie-Hellman key exchange procedure could be implemented with elliptic curves. This scheme received the name ECDH (Elliptic Curve Diffie-Hellman). In an independent research [2], Victor Miller prepared a similar proposal in relation to the general model described by Diffie and Hellman, though he did not include comparisons with other existing implementations.

1.1. Definition of an Elliptic Curve

Given the field $\mathbb{F}$ and the affine plane $A^2(\mathbb{F}) = \mathbb{F}^2$ defined over $\mathbb{F}$, the corresponding projective plane is represented as the set

$$P^2(\mathbb{F}) = \{(X,Y,Z) \in \mathbb{F}^3 \mid (X,Y,Z) \neq (0,0,0)\}$$

together with an equivalence relation defined so that two points of the projective plane, (X,Y,Z) and (X',Y',Z'), are equivalent if and only if there exists a value $\lambda \neq 0$ so that $(X',Y',Z') = (\lambda X, \lambda Y, \lambda Z)$ [3]. The equivalence class of the point (X,Y,Z) is represented as $[X : Y : Z]$.

A plane curve defined over a field $\mathbb{F}$ can be expressed in the affine plane $A^2(\mathbb{F})$ by means of the equation $f(x,y) = 0$ using non homogeneous coordinates, or alternatively in the projective plane $P^2(\mathbb{F})$ through the equation $F(X,Y,Z) = 0$ expressed in homogeneous coordinates [4], where a polynomial is considered to be homogeneous if all its monomials have the same degree.

The plane curve has rational points when the coordinates of those points belong to the field $\mathbb{F}$ (not necessarily $\mathbb{Q}$) [5]. The existence of rational points on a curve depends on the genus g of the curve, which is a concept derived from the Riemann theorem [6]. The genus allows to classify the plane curves based on the degree of the polynomial that defines the curve and the singularities it has through the expression

$$g = \frac{(n-1)(n-2)}{2} - \sum_{P_i} \frac{m_{P_i}(m_{P_i}-1)}{2},$$

where n is the degree of the polynomial and m_{P_i} represents the multiplicity of each singular point P_i [7].

*Address correspondence to this author at the Department of Information Processing and Coding, Information Security Institute (ISI), Spanish National Research Council (CSIC), Madrid, Spain;
Tel: +34915618806; Ext: 458; Fax: +34914117651;
E-mails: victor.gayoso,luis,agustin@iec.csic.es

1874-1177/13

Depending on the genus of a curve, it is possible to determine the existence of rational points (and, in particular, the existence of points where the coordinates are integer values) [8]:

• A curve of genus $g = 0$ has either no rational points or an infinite number of them. In particular, it can have no points where the coordinates are integer values, it can have a finite number of them or even infinite points of that type.

• A curve of genus $g = 1$ can have no rational points, a finite number of them or even infinite rational points, but it can only have a finite quantity of points with integer coordinates.

• A curve of genus $g \geq 2$ can only have a finite number of rational points.

A point of a curve is singular if and only if the partial derivatives of the elliptic curve expression are cancelled at that point [9]. A singular point of a plane cubic curve is called a node if the point has two distinct tangents and a cusp if the point has a double tangent [10]. A curve is singular if it has at least one singular point, while it is regular when it contains no singular points [11]. Figs. (**1**) and (**2**) present two examples of singular points taking the form of a node and a cusp, respectively.

Based on the previous definitions, it can be stated that an elliptic curve E over the field $\mathbb{F}$ is a regular projective curve of genus 1 with at least one rational point [9, 12]. Every elliptic curve admits a canonical equation called the Weierstrass form. That equation in homogeneous coordinates is

$$Y^2Z + a_1XYZ + a_3YZ^2 = X^3 + a_2X^2Z + a_4XZ^2 + a_6Z^3,$$

with $a_1, a_2, a_3, a_4, a_6 \in \mathbb{F}$ and $\Delta \neq 0$, where Δ is the discriminant of E and can be computed in the following way [13]:

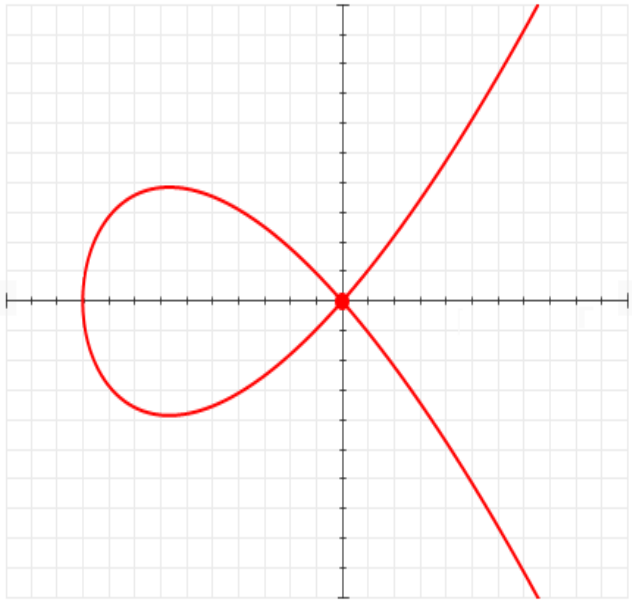

Fig. (1). Curve $y^2 = x^3$ with node in $(0,0)$.

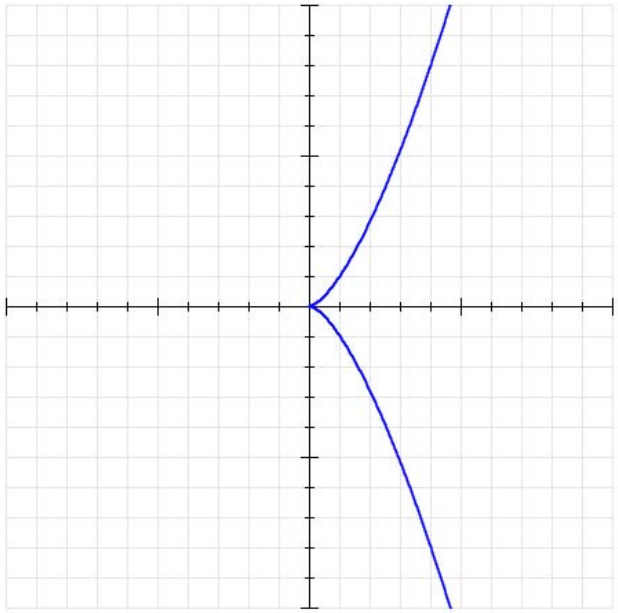

Fig. (2). Curve $y^2 + xy - x^3 = 0$ with cusp in $(0,0)$.

$$\Delta = -d_2^2d_8 - 8d_4^3 - 27d_6^2 + 9d_2d_4d_6,$$
$$d_2 = a_1^2 + 4a_2,$$
$$d_4 = 2a_4 + a_1a_3,$$
$$d_6 = a_3^2 + 4a_6,$$
$$d_8 = a_1^2a_6 + 4a_2a_6 - a_1a_3a_4 + a_2a_3^2 - a_4^2.$$

The Weierstrass equation is usually expressed in non homogeneous form, where the relationship between both equations is given by $f(x,y) = F(x,y,1)$ and $F(X,Y,Z) = f(X/Z, Y/Z)\cdot Z^3$, which produces the following affine equation:

$$E: y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6. \quad (1)$$

The homogeneous Weierstrass equation defines a projective plane curve with one point in the infinity, namely $\mathcal{O} = [0:1:0]$. In principle that curve does not have to be elliptic, as it could have singular points. Due to that fact, the condition $\Delta \neq 0$ assures that the curve is regular, which is equivalent to stating that every root of the equation must be necessarily simple [11, 14].

1.2. Group Structure

Given an elliptic curve E defined over a field $\mathbb{F}$ by means of equation (1), and given the elliptic curve points $P = (x_P, y_P)$, $Q = (x_Q, y_Q)$, and $R = (x_R, y_R)$, the operation $+$ is defined as follows [12, 15, 16]:

1. $\forall P \in E$, $P + \mathcal{O} = \mathcal{O} + P = P$.
2. Given a point P, there exists $-P = (x_P, -y_P - a_1x_P - a_3)$, so that $P + (-P) = \mathcal{O}$. It is

important to notice that P and $-P$ are the only points of the curve whose first coordinate is x_P.

3. Given two points P and Q such that $P \neq \pm Q$, then $R = P + Q$, with

$$x_R = \lambda^2 + a_1\lambda - a_2 - x_P - x_Q,$$
$$y_R = \lambda(x_P - x_R) - y_P - a_1 x_R - a_3,$$
$$\lambda = \frac{y_Q - y_P}{x_Q - x_P}.$$

4. Given a point P, $R = P + P = 2P$ has the following coordinates:

$$x_R = \lambda^2 + a_1\lambda - a_2 - x_P - x_Q, y_R = \lambda(x_P - x_R) - y_P - a_1 x_R - a_3, \lambda$$
$$= \frac{3x_P^2 + 2a_2 x_P + a_4 - a_1 y_P}{2y_P + a_1 x_P + a_3}.$$

The Mordell-Weil theorem [17, 18] states that the set of elliptic curve points, together with the sum operation, form an abelian group, where the sum operation has the following properties:

• Associativity: $\forall P, Q, R \in E, (P + Q) + R = P + (Q + R)$.

• Identity element: $\forall P \in E, P + \mathcal{O} = \mathcal{O} + P = P$.

• Inverse element: given a point $P = (x, y)$, there exists only one point P' such that $P + P' = \mathcal{O}$, where $P' = -P$.

• Commutativity: $\forall P, Q \in E, P + Q = Q + P$.

This properties imply that all the rational points of an elliptic curve defined over $\mathbb{Q}$ (or an extension of $\mathbb{Q}$) can be obtained from a finite number of points. In the case of finite fields, the number of generators needed is exactly 1 or 2 [19].

1.3. Elliptic Curves Over Finite Fields

The order of a finite field $\mathbb{F}$ is the number of elements of that field. If the order of a finite field is q, then $q = p^m$, where p is a prime number called the characteristic of the field, and m is a positive integer [14].

In general, elliptic curve cryptosystems use two types of finite fields $\mathbb{F}_q$ with $q = p^m$ elements: $\mathbb{F}_p$ (prime finite fields) and $\mathbb{F}_{2^m}$ (binary finite fields).

In prime finite fields, the elements of $\mathbb{F}_p$ are $\{0, 1, 2, \ldots, p-1\}$, and the operations are performed modulo p [20].

In comparison, in finite fields of the type $\mathbb{F}_{2^m}$ the elements are represented as bit strings of length m. If $f(x)$ is an irreducible polynomial of degree m with coeficientes in $\mathbb{F}_2$, then the field $\mathbb{F}_{2^m}$ can be interpreted as the set of polynomials with coefficients in $\mathbb{F}_2$ of degree less than the degree of $f(x)$ [20]:

$$\mathbb{F}_{2^m} = \mathbb{F}_2[x] / (f(x)).$$

In practice, the Weierstrass equation is not used, and the following simplified equations with affine coordinates are used depending on the characteristic of the finite field $\mathbb{F}$ where the elliptic curve is defined:

• If the finite field is a prime field, i.e. $\mathbb{F} = \mathbb{F}_p$, where $p > 3$ is a prime number, the equation defining the (non-supersingular) elliptic curve becomes:

$$y^2 = x^3 + ax + b. \quad (2)$$

• If the finite field is a binary field, i.e. $\mathbb{F} = \mathbb{F}_{2^m}$, where m is an integer number, then the equation of the (non-supersingular) elliptic curve is:

$$y^2 + xy = x^3 + ax^2 + b. \quad (3)$$

1.4. Order of an Elliptic Curve

The order of an elliptic curve E defined over a field $\mathbb{F}_q$ of characteristic p, denoted as $\#E(\mathbb{F}_q)$, is the number of points of $E(\mathbb{F}_q)$. If the base field is a finite field, the order of the curve is finite and is made up of the points of the curve that satisfy the curve equation plus the point in infinity.

The Hasse theorem [21] provides the following expression related to the order of the curve, where t is the trace of the curve [14]:

$$\#E(\mathbb{F}_q) = q + 1 - t, \quad |t| \leq 2\sqrt{q}.$$

1.5. Elliptic Curve Parameters

For elliptic curves defined over $\mathbb{F}_p$, where $p > 3$ is a prime number, the set of parameters that must be used in relation to a specific elliptic curve is

$$\{\mathcal{P}\} = (p, a, b, G, n, h),$$

where:

• p is the prime number that specifies the finite field $\mathbb{F}_p$.

• a and b are the elements of $\mathbb{F}_p$ that define the curve $E(\mathbb{F}_p)$ given by the equation $y^2 = x^3 + ax + b$.

• $G = (x_G, y_G)$ is the generator of a cyclic subgroup of the curve.

• n is the prime number that indicates the order of G.

• h is the elliptic curve cofactor, computed as $\#E(\mathbb{F}_p) / n$.

In comparison, if the curve is defined over $\mathbb{F}_{2^m}$, the set of parameters that must be managed is

$$\{\mathcal{P}\} = (m, f(x), a, b, G, n, h),$$

where:

• m is the positive integer number that specifies the finite field $\mathbb{F}_{2^m}$.

• $f(x)$ is an irreducible polynomial of degree m.

• a and b are the elements of $\mathbb{F}_{2^m}$ that define the curve $E(\mathbb{F}_{2^m})$ given by the expression $y^2 + xy = x^3 + ax^2 + b$.

• G, n and h have the same meaning as in the case of prime elliptic curves.

2. ELLIPTIC CURVE CRYPTOSYSTEMS

2.1. Early Cryptosystems

The first encryption schemes based on elliptic curves were the equivalent versions of the Massey-Omura [22] and ElGamal [23] cryptosystems, both presented by Koblitz in 1985 (and published in 1987) [1], and the Menezes-Vanstone cryptosystem [24].

Algorithm **1** describes the Massey-Omura protocol by which user **U** sends the message m to user **V**, where E is an elliptic curve defined over the finite field $\mathbb{F}_q$ of q elements, and there exists a publicly known relationship between the plaintexts and some points of the curve, so for any message m the point $P_m \in E$ is known by all the parties.

Algorithm 1. Massey-Omura Cryptosystem with Elliptic Curves

1. **U** must generate a random number c, with $0 < c < \#E$, where c and the order of the curve, $\#E$, are relative primes. **U** must send the computed elliptic curve point, cP_m, to **V**.
2. **V** must generate a random number d (where $0 < d < \#E$ and the values d and $\#E$ are coprimes). After that, **V** must send the elliptic curve point dcP_m to **U**.
3. After receiving the information from the other user, **U** must transmit to **V** the curve point $c'dcP_m = dP_m$, where $c'c \equiv 1 (\mathrm{mod}\ \#E)$.
4. Finally, user **V** must obtain the curve point P_m associated to the message m by computing $d'dP_m$, where $d'd \equiv 1 (\mathrm{mod}\ \#E)$.

Similarly, given a curve E, a generator G, and a publicly available correspondence of plaintexts and curve points, Algorithm **2** presents the steps that must be performed so user **U** can send the message m to **V** using the ElGamal encryption scheme adapted for elliptic curves.

One of the main disadvantages of the Massey-Omura and ElGamal versions adapted for elliptic curves is that plaintexts and encrypted messages must be represented as points of an elliptic curve E. This disadvantage, when using elliptic curves whose order $\#E$ is a high value, produces a limitation which is more theoretical than practical. However, the requirement to build tables stating the relationship between every possible message and its related elliptic curve point limits the usefulness of these cryptosystems to closed environments (enterprises, small groups, etc.) where all possible messages are previously established.

Algorithm 2. ElGamal Cryptosystem with Elliptic Curves

1. **V** must choose randomly the value v, making publicly available the key $V = vG$.
2. Taking into account the message m and its associated curve point P_m, user **U** must generate a random value k and send the pair of points $(kG, P_m + kV)$ to **V**.
3. After receiving the pair of points, **V** can recover the curve point associated to the message by multiplying the point kG by the value v, thus obtaining the point $v(kG) = k(vG) = kV$. After that, **V** must substract the generated point from $P_m + kV$.

The Menezes-Vanstone cryptosystem for elliptic curves was designed precisely to overcome this limitation, as instead of matching each message with a point of the curve E, it represents the plaintexts as ordered pairs of $\mathbb{F}^* \times \mathbb{F}^*$, where $\mathbb{F}^* = \mathbb{F} \setminus \{0\}$ and those pairs do not necessarily have to represent the coordinates of an elliptic curve point. Using this cryptosystem, it is possible to split any plaintext in blocks, where each block could be easily encoded as an ordered pair. The disadvantage of this procedure is that, instead of transforming each clear message into a single point of the curve (where the binary representation of every point of the curve has the same length), the size of the encrypted message depends directly on the length of the plaintext.

Algorithm **3** presents the steps that are necessary in order to complete the encryption and decryption procedures of a message represented as the element $x = (x_1, x_2) \in \mathbb{F}_q^* \times \mathbb{F}_q^*$ using the Menezes-Vanstone cryptosystem adapted for elliptic curves. In the procedure, the order of the generator of a cyclic subgroup of E, defined as G, is n, while v and $V = vG$ represent the private and public keys of user **V**, respectively.

In the Menezes-Vanstone cryptosystem, the expansion factor (i.e. the ratio between the size of the cryptogram and the length of the clear message) is 2, since a plaintext $x = (x_1, x_2)$, consisting of two elements of the finite field $\mathbb{F}^*$, produces the cryptogram (Y_0, y_1, y_2), where $y_1, y_2 \in \mathbb{F}$ and Y_0 is an elliptic curve point with two coordinates that belong to the finite field, so in summary the total number of finite elements to be transmitted is 4. When using point compression in order to decrease the length of the information to be transmitted (i.e. only the first coordinate of the point is sent along with an additional byte that includes the necessary data for recovering the whole point), the expansion factor is reduced to approximately 1.5.

Algorithm 3. Menezes-Vanstone Cryptosystem for Elliptic Curves

1. **U** must choose a random value $k \in \{1,\ldots,n-1\}$ and compute $Y_0 = kG \in E$, $y_1 = c_1 x_1 (\text{mod } p)$, and $y_2 = c_2 x_2 (\text{mod } p)$, where $(c_1, c_2) = kV$.
2. **U** must send the cryptogram $\mathcal{C} = (Y_0, y_1, y_2)$ to **V**.
3. After receiving the cryptogram, **V** will recover the pair $x = (x_1, x_2)$ by means of the operations $x_1 = y_1 c_1^{-1} (\text{mod } q)$ and $x_2 = y_2 c_2^{-1} (\text{mod } q)$, where $vY_0 = (c_1, c_2)$.

In comparison, the expansion factor of the variants of the Massey-Omura and ElGamal cryptosystems is 2, as the clear message is considered to be a point of the curve and, in each one of those cryptosystems, it is necessary to transmit 2 elliptic curve points. In practice, a high value for the expansion factor implies that the encryption of the information generates cryptograms much larger than those produced when using a symmetric key algorithm like AES.

Another disadvantage resulting from the design of the Menezes-Vanstone cryptosystem is that it is necessary to perform operations with the points of the elliptic curve in each encryption process. As depending on the plaintext length it is necessary to divide the plaintext in multiple segments and perform asymmetric encryption operations with each of those segments, when the number of segments increases the performance of the Menezes-Vanstone scheme degrades much faster than in the case of using a symmetric encryption algorithm.

In addition to the previously mentioned practical disadvantages, Klaus Kiefer showed in 1998 that, under certain conditions, this cryptosystem is insecure [25]. Kiefer also demonstrated that, contrary to the terms of its specification, the Menezes-Vanstone cryptosystem cannot be considered a probabilistic encryption algorithm.

2.2. Hybrid Cryptosystems

Due to the reasons mentioned in the previous section, over the years the academic community abandoned the study of the three initial cryptosystems based on elliptic curves. As an illustrative example, while in the first edition of the work by Douglas Stinson [26] both the ElGamal and the Menezes-Vanstone cryptosystems for elliptic curves were included, in the second and third editions these schemes were replaced by ECIES, the Elliptic Curve Integrated Encryption Scheme. Even in one of the latest books about this subject, co-authored by Alfred Menezes and Scott Vanstone [14], the Menezes-Vanstone scheme is not included.

However, the discovery of the limitations of these early cryptosystems did not imply the abandonment of the search for a practical and secure elliptic curve cryptosystem, as it only caused a change of direction, occupying now the spotlight the hybrid encryption schemes, which bring the best characteristics of both symmetric and asymmetric cryptography. The most important hybrid schemes that use elliptic curves are ECIES, PSEC (Provably Secure Elliptic Curve encryption scheme) [27, 28], and ACE (Advanced Cryptographic Engine) [28, 29].

Of the three schemes, ECIES is available in a greater number of standards (ANSI X9.63 [30], IEEE 1363rd [31], ISO/IEC 18033-2 [32], and SECG SEC 1 [33]). PSEC can be found in ISO/IEC 18033-2 [32], IETF RFC 4051 [34], and the set of algorithms selected for the NESSIE (New European Schemes for Signatures, Integrity and Encryption) project [35, 36], while ACE is available in ISO/IEC 18033-2 [32], and the final selection of NESSIE [35, 36].

Table 1. Encryption Process in ECIES, PSEC, and ACE

ECIES	PSEC	ACE
$u \in [1, n-1]$	$r \in \{0,1\}^l$	$u \in [1, n-1]$
$U = uG$	$K = \text{KDF}(0_{32} \| \bar{r})$	$U = uG$
$P = uV$	$K = t \| k_1 \| k_2$	$P = uW$
$P = (x_P, y_P)$	$u = t (\text{mod } n)$	$P' = uZ$
$K = \text{KDF}(\bar{U} \| \overline{x_P})$	$U = uG$	$P = (x_{P'}, y_{P'})$
$K = k_1 \| k_2$	$P = uV$	$\alpha = \text{HASH}(\bar{U} \| \bar{P})$
$\mathsf{c} = \text{ENC}_{k_1}(m)$	$s = r \oplus \text{KDF}(1_{32} \| \bar{U} \| \bar{P})$	$u' = \alpha u (\text{mod } n)$
$tag = \text{MAC}_{k_2}(\mathsf{c})$	$c = \text{ENC}_{k_1}(\mathsf{m})$	$Q = uX + u'Y$
$\mathcal{C} = (U, c, \text{tag})$	$\text{tag} = \text{MAC}_{k_2}(c)$	$K = \text{KDF}(\bar{U} \| \overline{x_{P'}})$
	$\mathcal{C} = (U, \mathsf{n}, s, \text{tag})$	$K = k_1 \| k_2$
		$\mathcal{C} = \text{ENC}_{k_1}(\mathsf{m})$
		$\text{tag} = \text{MAC}_{k_2}(\mathsf{n})$
		$\mathcal{C} = (U, P, Q, c, \text{tag})$

Table **1** includes a comparative summary of the encryption process of ECIES, PSEC, and ACE, using for all of them an equivalent notation in order to highlight the similarities between them.

The meaning of the functions and elements included in Table **1** is the following:

• KDF (Key Derivation Function): Mechanism that produces a set of keys from keying material and some optional parameters.

• HASH: Digest function.

• ENC (Encryption): Symmetric encryption algorithm.

• MAC (Message Authentication Code): Function used to authenticate a message.

• G: generator of the cyclic subgroup of elliptic curve points used in the procedure.

• n: order of the point G.

• u: temporary private key of the user who sends the cryptogram, U.

• U: temporary public key of **U**, where $U = uG$.

• v: permanent private key of the user who receives the cryptogram, V.

• V: permanent public key of **V**. In ECIES and PSEC, $V = vG$, whilst in ACE the public key is composed of four points of the elliptic curve, so $V = (W, X, Y, Z)$.

• r: random binary string whose length l is fixed.

• 0_{32}: 32-bit string representing the integer value 0.

• 1_{32}: 32-bit string that represents the integer value 1.

• t: $(\lambda + 128)$-bit string that must be interpreted as an integer value, where λ is a security parameter whose value is the length in bits of the working finite field (i.e., $\lceil \log_2 p \rceil$ or m, depending on the type of finite field).

3. COMPARISON OF ECIES, PSEC, AND ACE

After reviewing the three schemes, the main differences that can be identified are the following:

• In PSEC and ECIES, the receiver's public key is a point on the elliptic curve, $V = v \cdot G$, where v is the receiver's private key, and G is the generator of the group of points of the cyclic subgroup used in the computations. In contrast, in ACE the public key consists of four elliptic curve points, that is, $V = (W, X, Y, Z)$.

• PSEC uses twice a key derivation function in order to obtain a pair of MAC and symmetric encryption keys, whilst ACE and ECIES use such a function only once.

• ECIES and ACE use the first coordinate of a point of the curve generated during the calculations (instead of both coordinates) as an input parameter to the key derivation function previously mentioned, while PSEC requires to use both coordinates.

• PSEC is the only scheme that uses the XOR function during the key generation process (regardless of its usage as a symmetric encryption function).

• Cryptograms in ECIES consist of three elements (the sender's ephemeral public key, the encrypted message, and a MAC code), while cryptograms in PSEC include one additional element, a binary string, and in ACE the cryptograms include two additional elliptic curve points.

Table 2. Number of Group Operations in ECIES, PSEC, and ACE

	ECIES	PSEC	ACE
Group exponentiations	2	2	5
Group multiplications	0	0	1
Random numbers	1	1	1
Hash calls	1	2	2
Symmetric cipher calls	1	1	1
MAC calls	1	0	0

Table 3. Estimated Performance of ECIES, PSEC, and ACE

	ECIES	PSEC	ACE
Encyption cycles	2500K	2500K	6250K
Encryption time	5 ms	5 ms	12.5 ms
Decryption cycles	1250K	2500K	3750K
Decryption time	2.5 ms	5 ms	7.5 ms

Table 4. Comparison of Usual Parameter Lengths (in Bytes) for ECIES, PSEC, and ACE

	ECIES	PSEC	ACE
Private key	20	20	80
Public key	20	20	80
Cryptogram	36	52	76

After presenting the encryption procedure for the three schemes, we will present an evaluation in terms of performance, security and adaptability to the hardware platforms where they can be implemented.

3.1. Performance

Regarding the performance aspects, Table **2** presents a comparison about the number of operations that are needed in order to encrypt a message with the three cryptosystems. The data contained in that table have been extracted from the Nessie final report [36].

On the other hand, Table **3** presents the estimated performance of the three encryption schemes using a Pentium III PC with a clock frequency of 500 MHz [36].

Another aspect that must be taken into account in order to evaluate the efficiency of the schemes is the expansion factor. Table **4** shows the private and public key lengths in bytes, as well as the cryptogram length equally in bytes, where the length of the finite field elements, the plaintext and the hash function output are 20, 16, and 20 bytes, respectively. In this comparison, the elliptic curve points use the compression feature, so instead of the two coordinates only one is sent along with an addition byte that can be used to obtain the proper value of the second coordinate.

3.2. Security

The security level of the three schemes is a topic on which there is no agreement in the academic community. The final NESSIE report selected PSEC and ACE, leaving out of the selection ECIES. This was due, among other factors, to the bening malleability problems that affect ECIES, and to the fact that ECIES is an unauthenticated KEM (Key Encapsulation Mechanism) scheme (i.e., it does not check if the element $P = uV$ used for the calculation is correct based on other parameters of the cryptogram), while PSEC and ACE are authenticated schemes that do not have bening malleability problems.

However, there are several solutions to prevent benign malleability [28]. In addition, the usefulness of the authenticated KEM models in comparison to the unauthenticated models is unclear, as in the DEM (Data Encapsulation Mechanism) phase a MAC algorithm is used to authenticate the message data [37]. Finally, some authors believe that the analysis included in NESSIE is incomplete and that, with the currently known data, it cannot be stated that PSEC is safer than ECIES [38]. NESSIE documents were delivered in 2004 and, due to the completion of the European Union project to which they were related, they have not been reviewed since then.

3.3. Adaptability

The last topic that must be considered when comparing ECIES, PSEC, and ACE is related to the capabilities available on the devices that will implement those encryption schemes. While certainly in PCs there are no limitations in this regard, since any cryptographic function can be programmed using for instance C++ or Java Standard Edition, in devices with limited resources such as smart cards there are significant differences.

In Java Card, for example, there are no methods for performing point additions or scalar multiplications. However, in Java Card it is possible to use the Diffie-Hellman function with elliptic curves, which can be considered equivalent to the scalar product with the particularity that the Java Card API defines, as a result of this function, the first coordinate of the point of the curve representing the multiplication or the output of a hash function that takes as input the value of that coordinate. Given that Java Card does not have functions for adding elliptic curve points or computing modulo operations, ECIES is the only encryption scheme of the three considered that can be implemented efficiently in Java Card.

4. SUMMARY

Throughout this contribution, we have presented the early elliptic curve cryptosystems (Massey-Omura, ElGamal, and Menezes-Vanstone) and their evolution, the hybrid cryptosystems (ECIES, PSEC, and ACE). Hybrid cryptosystems present the advantage of using the best techniques of both assymetric and symmetric cryptography.

After evaluating the three criteria (performance, security and functionality available on the PC and Java Card platforms), in view of their features it can be stated that the best asymmetric encryption scheme based on elliptic curves is ECIES. Not only ECIES provides a good overall performance, but it can be easily implemented in PC and Java Card.

CONFLICT OF INTEREST

None declared.

ACKNOWLEDGEMENTS

This work has been partially supported by Ministerio de Ciencia e Innovación (Spain) under the grant TIN 2011-22668.

REFERENCES

[1] Koblitz N. Elliptic curve cryptosystems. Math Comput 1987; 48: 203-9.

[2] Miller V. Use of Elliptic curves in cryptography. Lect Notes Comput Sci 1986; 218: 417-26.

[3] Milne JS. Elliptic curves. Charleston (South Carolina): Book Surge 2006.

[4] Washington LC. Elliptic curves. Number theory and cryptography. 2nd ed. Boca Raton (Florida): Chapman & Hall/CRC 2008.

[5] Silverman JH, Tate J. Rational points on elliptic curves. New York: Springer-Verlag 1992.

[6] Riemann B. Theorie der Abel'schen functionen. J für Mathematik 1857; 54: 115-55.

[7] Griffiths PA. Introduction to algebraic curves. Providence Rhode Island: American Mathematical Society 1989.

[8] Connell I. Elliptic Curve Handbook. Preprint 1999.

[9] Koblitz N. Algebraic aspects of cryptography. Berlin: Springer-Verlag 1998.

[10] Erickson M, Vazzana A. Introduction to number theory. Boca Raton (Florida): Chapman & Hall/CRC 2008.

[11] Enge A. Elliptic curves and their applications to cryptography: An introduction. Boston: Kluwer Academic Publishers 1999.

[12] Silverman JH. The arithmetic of elliptic curves. 2nd ed. New York: Springer-Verlag 2009.

[13] Menezes AJ. Elliptic curve public key cryptosystems. Boston: Kluwer Academic Publishers 1993.

[14] Hankerson D, Menezes A, Vanstone S. Guide to elliptic curve cryptography. New York: Springer-Verlag 2004.

[15] Cohen H. Handbook of elliptic and hyperelliptic curve cryptography. Boca Raton Florida: Chapman & Hall/CRC 2006.

[16] Koblitz N. A course in number theory and cryptography. 2nd ed. Berlin: Springer-Verlag 1994.

[17] Mordell LJ. On the rational solutions of the indeterminate equations of the third and fourth degrees. Proc Cambridge Philosophical Soc 1922; 21: 179-92.

[18] Weil A. L'arithmétique sur les courbes algébriques. Acta Math 1929; 52: 281-315.

[19] Riesel H. Prime numbers and computer methods for factorization. 2nd ed. Boston: Birkhäuser 1994.

[20] Bach E, Shallit J. Algorithmic number theory. Cambridge (Massachusetts): MIT Press 1996.

[21] Katz NM, Mazur B. Arithmetic moduli of elliptic curves. Princeton (New Jersey): Princeton University Press 1985.

[22] Massey LJ, Omura JK. OMNET Associates. Method and Apparatus for Maintaining the Privacy of Digital Messages Conveyed by Public Transmission. Filling date: 1982-09-14. Issue date: 1986-01-28. United States patent 4.567.600.

[23] ElGamal T. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. IEEE Trans Inform Theory 1985; 31: 469-72.

[24] Menezes A, Vanstone S. Elliptic curve cryptosystems and their implementation. J Cryptol 1993; 6: 4: 209-24.

[25] Kieffer K. A Weakness of the Menezes-Vanstone cryptosystem. Lect Notes Comput Sci 1998; 1361: 201-6.

[26] Stinson D. Cryptography: Theory and Practice. 3rd ed. Florida: Boca Raton Chapman & Hall/CRC: 2006.

[27] NTT Corporation. PSEC-KEM Specification, v. 2.2, 2008.

[28] Shoup V. A Proposal for an ISO Standard for Public Key Encryption, v. 2.1, preprint 2001.

[29] Cramer R, Shoup V. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. SIAM J Computing 2003; 33, 1: 167-226

[30] ANSI X9.63. Public Key Cryptography for the Financial Services Industry: Key Agreement and Key Transport Using Elliptic Curve Cryptography, American National Standards Institute 2001.

[31] IEEE Std 1363a. Standard Specifications for Public Key Cryptography - Amendment 1: Additional Techniques. Institute of Electrical and Electronics Engineers 2004.

[32] ISO/IEC 18033-2. Information Technology -- Security Techniques -- Encryption Algorithms -- Part 2: Asymmetric Ciphers, International Organization for Standardization 2006.

[33] SECG SEC 1. Elliptic Curve Cryptography, v. 2.0. Standards for Efficient Cryptography Group 2009.

[34] Eastlake D. Additional XML Security Uniform Resource Identifiers (URIs), IETF RFC 4051, 2005.

[35] NESSIE Consortium. Portfolio of Recommended Cryptographic Primitives, v. 1.0, 2003.

[36] NESSIE Consortium. Final Report of European Project Number IST-1999-12324 Named New European Schemes for Signatures, Integrity, and Encryption, v. 0.15, 2004.

[37] Dent AW. ACE-KEM and the general KEM-DEM structure. NES/DOC/RHU/WP5/023/3. NESSIE Project 2002.

[38] Galindo D, Martín S, Villar JL. The security of PSEC-KEM versus ECIES-KEM, In: Proceedings of the 26th Symposium on Information Theory in the BeNeLux 2005; pp. 17-27.

Received: December 24, 2012 Revised: February 25, 2013 Accepted: March 02, 2013