Evaluating elliptic curve based KEMs in light of pairings

David Galindo, Sebastià Martín and Jorge L. Villar

Dep. Matemàtica Aplicada IV. Universitat Politècnica de Catalunya
Campus Nord, c/Jordi Girona, 1-3, 08034 Barcelona
e-mail: {dgalindo,sebasm,jvillar}@mat.upc.es

Abstract

Several efforts to put forward a set of cryptographic primitives for public key encryption, suitable to be standardized, have been taken recently. In two of them (in first place the NESSIE project, already finished, and in second place ISO/IEC 18033), the methodology by Victor Shoup for hybrid encryption, known as *Key Encapsulation Method-Data Encapsulation Mechanism* (KEM-DEM), has been accepted. In this work we re-evaluate the elliptic curve based KEMs studied to become standards, which are called ACE-KEM, ECIES-KEM and PSEC-KEM. We analyse both their security properties and performance when pairing curves are used. It turns out that these KEMs present a very tight security reduction to the CDH problem over pairing curves; moreover, one can even relate their security to the DL problem in certain pairing curves with a small security loss. It is also shown that ECIES-KEM arises as the best option among these KEMs when pairing curves are used. This is remarkable, since NESSIE refused ECIES-KEM over a general curve to be proposed as a standard. It is concluded that for medium security level applications, which is likely the case for many embedded systems (e.g. smart cards), ECIES-KEM should be considered the best candidate.

Keywords: public-key cryptography, key encapsulation mechanisms, pairings, standards, smart cards.

1 Introduction

A key encapsulation mechanism (KEM) is a probabilistic algorithm that produces a random symmetric key and an asymmetric encryption of that key. When properly combined with a symmetric encryption scheme it gives a secure encryption of arbitrary long messages (cf. [CS]). As far as we know, there are three elliptic curve based KEMs that have been considered for standardization (in particular in ISO/IEC 18033 [Sho04] and NESSIE [Nes03]), namely, ACE-KEM, ECIES-KEM and PSEC-KEM. Their security rely on different problems related to the discrete logarithm on elliptic curves (DL). PSEC-KEM and ECIES-KEM use the Random Oracle (RO) heuristic [BR93] in their security proofs, while ACE-KEM is proven secure in the standard model but based on a decisional assumption. They were first proposed as KEMs in [Sho01], the ISO standard draft for public key encryption by Victor Shoup, while in its original form they were submitted by IBM, Certicom and NTT corporations, respectively.

The hardness of these DL problems closely depends on the elliptic curve used. Indeed, special families of elliptic curves with easy point-counting, such as supersingular curves or anomalous curves, turned out to be insecure (as shown, among others works, in [MTV93] and [Sma99]). How these results must be interpreted is a quite debatated question. The conservative approach is to avoid special families of curves: maybe future developments may show inherent weaknesses in particular curves. The proposal is then to generate curves at random. A more efficient approach is to build curves with known order using complex multiplication techniques [LZ94], but then some randomness is losen in the way. Finally, the most appealing approach from a practical point of view is that any curve which has not been proven insecure can be used.

In [Jou00] a special family of curves, namely, elliptic curves with a non-trivial bilinear map (which will be hereafter referred as *pairing curves*), were found a positive application in cryptography, designing a one-round tripartite Diffie-Hellmann protocol. A breakthrough in this constructive direction was made in [BF01], presenting the most complete and practical identity-based encryption scheme to the date. Since then, pairings have been found a lot of applications in cryptography, mainly in the identity based framework, but also for designing special signature schemes, non-interactive protocols or new paradigms going beyond traditional public key infrastructure (see [DBS04] for a comprehensive account).

But in [Jou00] was also pointed out that in such curves the Decisional Diffie-Hellman (DDH) problem becomes easy. This could be interpreted as an inherent weakness of these curves, and therefore one should avoid them following the conservative approach. However, in [JN03] pairing elliptic curves were presented for which the DL is believed to be hard, and the Computational Diffie-Hellman problem is equivalent to the DL. Currently, pairing curves are being given more and more confidence by the cryptographic community.

Our contribution. We revisit the security proof of the elliptic curve based KEMs when they are performed over pairing curves. As a result, we show that *all these* KEMs *can be proven secure in the* RO *heuristic with respect to the* Computational Diffie-Hellman *assumption in a pairing curve*, and with a *very tight reduction*, improving then the concrete security claimed over a random curve. This enables to use as smaller key sizes in the schemes as possible, and therefore make these KEMs suitable to be implemented in constrained memory devices. It is worthwhile to note that, although the schemes are implemented over a pairing curve and we use efficient pairing computations to obtain the concrete security, *no pairing computation* is involved in a real implementation. The crucial point is that DDH problem is solvable in these groups.

Since ECIES-KEM has the best perfomance, we conclude ECIES-KEM *is preferable* among the others if pairing curves are used. This is noticeable, since when using a randomly generated curve an opposite result is obtained. In fact, ECIES-KEM has not been accepted to be proposed for standardization in NESSIE, while ACE-KEM and PSEC-KEM have been positively evaluated. We argue that for a medium level of security, which is likely the case *for smart cards applications,* ECIES-KEM *over* *pairing curves should be considered the best option*, in terms of efficiency and security, among the elliptic curve based KEMs.

On the other hand, using [Mau94] there are elliptic curves where DL can be reduced to CDH. Then, it is possible to give an exact security result *relating the* IND-CCA *security of these* KEMs *to the* DL *problem.* The good news is that they are closely related, due to small security losses in the reduction. From a theoretical point of view, this gives more confidence on the security of these KEMs over pairing curves. In particular, we show that for the current security level (that is, 2^{80}), breaking ECIES-KEM *is equivalent* to solving the DL problem on a pairing curve with a prime order subgroup with a 2^{121} security of the DL. We point out that such a concrete estimation with respect to the DL is rarely found in the literature. Finally, taking into account the state of the art in pairings, we give some examples of pairing curves where the schemes can be implemented.

2 Security properties of existing elliptic curve based KEMs

We first summarize some notation. If p is a positive integer, then $|p|$ denotes the length of its binary representation. If A is a non-empty set, then $x, y \leftarrow A$ denotes that x, y have been uniformly and independently chosen in A. On the other hand, if $\mathcal{A}$ is a probabilistic polynomial time (PPT) algorithm, then $x \leftarrow \mathcal{A}$ denotes that x is the output of $\mathcal{A}$. $Hash$ and KDF denote a hash function and a key derivation function, respectively (cf. [CS]).

IND-CCA security of a KEM. A KEM consists of three algorithms:

- A *key generation* algorithm $\mathcal{K}$, a probabilistic algorithm which takes as input a security parameter 1^ℓ and outputs a public/secret-key pair $(\mathsf{pk}, \mathsf{sk})$.
- An *encapsulation* algorithm $\mathcal{E}$, a probabilistic algorithm taking as inputs a security parameter 1^ℓ and a public key pk and returning an encapsulated key-pair (K, C), with $K \in \{0,1\}^{p(\ell)}$, $C \in \{0,1\}^{q(\ell)}$, for some polynomials $p, q \in \mathbb{Z}[\ell]$.
- A *decapsulation* algorithm $\mathcal{D}$, a deterministic algorithm that, on inputs a security parameter 1^ℓ, an encapsulation C and a secret key sk; outputs a key K.

It is required to be sound, that is, for almost all $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{K}(1^\ell)$, and almost all $(K, C) \leftarrow \mathcal{E}(1^\ell, \mathsf{pk})$ we have that $K = \mathcal{D}(1^\ell, C, \mathsf{sk})$.

Here follows the description of the attack game used to define the IND-CCA security of a KEM:

1. The adversary queries a *key generation oracle,* which computes $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{K}(1^\ell)$ and returns pk.

2. The adversary makes a sequence of calls to a *decryption oracle*, submitting encapsulations C of its choice, for which the decryption oracle responds with $\mathcal{D}(1^\ell, C, \mathsf{sk})$.

3. The adversary queries an *encryption oracle*, which computes:

$$(K_0, C^*) \leftarrow \mathcal{E}(1^\ell, \mathsf{pk}); \quad K_1 \leftarrow \{0,1\}^{p(\ell)}; \quad b \leftarrow \{0,1\}$$

and returns the pair (K_b, C^*).

4. The adversary issues new calls to the decryption oracle, subject only to the restriction that a submitted ciphertext $C \neq C^*$.
5. The adversary outputs $b' \in \{0, 1\}$.

For a PPT adversary $\mathcal{A}^{\mathrm{KEM}}$ we define

$$\mathsf{Adv}_{\mathrm{KEM},\mathcal{A}}(\ell) := \left|\mathsf{Pr}\left[\mathcal{A}^{\mathrm{KEM}}(1^\ell) = 1 \,|\, b = 0\right] - \mathsf{Pr}\left[\mathcal{A}^{\mathrm{KEM}}(1^\ell) = 1 \,|\, b = 1\right]\right|.$$

We say that a KEM is IND-CCA secure if for all PPT adversaries $\mathcal{A}$ the function $\mathsf{Adv}_{\mathrm{KEM},\mathcal{A}}(\ell)$ grows negligibly in ℓ.

Elliptic curve discrete logarithm problems. Let $E_{a,b}(\mathbb{F}_q)$ denote the group of points of the elliptic curve

$$E_{a,b} \;:\; y^2 = x^3 + ax + b$$

over the prime finite field $\mathbb{F}_q$, $q > 3$. Let $G_p = \langle P \rangle$ be a cyclic group of prime order p, where $P \in E_{a,b}(\mathbb{F}_q)$. Then:

– The *discrete logarithm* (DL) is the problem of finding u when given (P, uP).
– The *computational Diffie-Hellman* problem (CDH) is the problem of finding uvP when given (P, uP, vP).
– The *decisional Diffie-Hellman* problem (DDH) is the problem of distinguishing (P, uP, vP, uvP) from (P, uP, vP, wP).
– The *gap Diffie-Hellman* problem (gap-CDH) is the problem of finding uvP when given (P, uP, vP) and an oracle $\mathcal{O}$ that correctly solves the decisional Diffie-Hellman problem.

It is assumed that $u, v, w \leftarrow \mathbb{F}_q$. Notice that all three KEMs are intended to be performed on random elliptic curves, so all these problems are assumed to be intractable. All of them are well established, except for the gap-DH problem, which was formally introduced in [OP01]. It is an open problem to establish the relations between them. In fact, we know little more than the obvious reductions, which are DDH infeasible $\Rightarrow$ CDH infeasible $\Rightarrow$ DL infeasible. Thus, the better way known to attack these problems in a general elliptic curve is to solve DL. The fastest method for solving DL on a random elliptic curve is the Pollar ρ method [Pol78], which runs in exponential time $\sqrt{\pi q/2}$ for a group with q elements. It is unknown whether there exist groups for which the CDH problem is substantially easier than the DL problem, while the DDH problem appears to be easier than the CDH problem in general. We refer the reader interested in the state of the art to [MW00].

Concrete security. The efficiency a the reduction is the relationship between an *attacker* who breaks the cryptosystem with probability at least ϵ in time t, doing less than q_D calls to a decryption oracle, and less than q_K calls to an oracle for hash or a *KDF* function; and the implied (t', ϵ') *solver* against the corresponding trusted cryptographic assumption. Such an attacker is referred as a $(t, \epsilon, q_D, q_{\mathcal{O}_i})$ attacker for short. Then, the security reduction is *tight* if $\frac{t'}{\epsilon'} \approx \frac{t}{\epsilon}$, and *not tight* if $\frac{t'}{\epsilon'} > q_D \frac{t}{\epsilon}$. It is also stated that a scheme is *very tight* if $\epsilon \approx \epsilon'$ and t' is equal to t plus a linear

quantity in the number of oracle calls. The tighter is the reduction, the smaller is the gap between the computational efforts needed to break the scheme and to solve the underlying problem. This has a great impact in the efficiency of the scheme, since a tight security reduction allows to use smaller security parameters.
To be consistent with the time units commonly used in the literature, we use the sentence *a problem $\mathcal{P}$ has a 2^t security level* to say that, an attacker against $\mathcal{P}$, running in time less than 2^t 3-DES encryptions (cf. [LV01]), has a “negligible” success probability.

Known results about elliptic curve based KEMs. The first step of the key generation algorithm in the three schemes studied is to build a suitable curve E, together with a point P that generates a secure cyclic subgroup G_p of E with prime order p. Moreover, p is of size ℓ, where ℓ is the security parameter. So we will assume that the key generation algorithm takes the group parameters (E, P, p) as input.

We summarize now the security properties of the KEMs discussed, as well as their performance and the evaluation presented in the NESSIE project. A schematic description of these algorithms can be found in appendix A.

In table 2 we summarize the exact security results known for the KEMs we are interested in, along with the reference where these results come from. In these expressions, q_K denotes the number of queries made to the KDF oracle, L_{G_p} is the time needed to check a Diffie-Hellman triple in G_p and SR_q is the time needed to compute a square root modulo q. We point out that in the ECIES-KEM security reduction claimed in [Den02] lacks the time to compute a square root in $\mathbb{F}_q$, which is needed in order to obtain the two points in $E(\mathbb{F}_q)$ that have a given x-coordinate.

Scheme	Assumption	Reduction	Random Oracle	Reference
ACE-KEM	DDH	very tight	No	[CS]
	Gap-CDH	$\epsilon' \approx \epsilon$ $t' \approx t + q_K(2L_{G_p} + SR_q)$	Yes	[Sho01] [Den02]
	CDH	Not tight	Yes	[Sho00]
ECIES-KEM	Gap-CDH	$\epsilon' \approx \epsilon$ $t' \approx t + q_K(2L_{G_p} + SR_q)$	Yes	[Den02]
PSEC-KEM	CDH	$\epsilon' \approx \frac{1}{q_D+q_K}\epsilon$ $t' \approx t$	Yes	[Sho01]

Table 1: IND-CCA KEMs concrete security over a random curve

As we can see, ACE-KEM offers several possible concrete security estimates, depending on which problem its security is based. In the case of the NESSIE evaluation the emphasis is put on the DDH problem, since the claimed security is achieved in the standard model. On the other hand, ECIES-KEM presents a very tight reduction to the gap-CDH problem, while PSEC-KEM has a not tight reduction to the CDH problem. Both schemes are analysed with the RO heuristic. In table 2 we have the parameters length in bytes for a 2^{80} IND-CCA security bound in each scheme. To compute them, it is assumed that DDH, Gap-CDH and CDH problems have comparable security to the DL problem in a random curve. Although this is widely believed, we emphasize that these *are extra assumptions*. Both ACE-KEM and ECIES-KEM use a group G_p with a $p \approx 2^{160}$ cardinality, while PSEC needs $p \approx 2^{280}$. This important difference arises from the not tight reduction in the security proof of PSEC-KEM. We notice that NESSIE parameter length estimation for PSEC is not exact (it is stated that a 160-bit prime is enough), and we argue it in appendix B.

Scheme	Operations in Enc	Operations in Dec	(K, C) length 16-Byte Keys	Public key length	Secret key length
ACE-KEM	5	3	76	80	80
ECIES-KEM	**2**	**1**	**36**	**20**	**20**
PSEC-KEM	2	2	67	35	35

Table 2: Performance features over random curves (lengths using a point compression technique)

In terms of performance, ECIES is clearly the best option. Not only presents the smallest computation time, but also the smallest parameter length. However, since its security is based on a quite new assumption, NESSIE refused to propose standardizing ECIES-KEM, while accepted ACE-KEM and PSEC-KEM, since these schemes base its security in well studied assumptions. In the next section we argue that, if pairing curves are used, this conclusion is no longer valid. Moreover, we provide evidences that in this case ECIES-KEM arises as the best candidate.

3 Security analysis over pairing curves

Let $E_{a,b}(\mathbb{F}_q)$ be the group of points of an elliptic curve over the prime finite field $\mathbb{F}_q$. Let $G_p = \langle P \rangle$ be a cyclic subgroup of $E_{a,b}(\mathbb{F}_q)$ with p elements, where p is a large prime. Let $\mathbb{G}$ be a cyclic group with p elements. We say that E is a *pairing curve over* $\mathbb{F}_q$ *with respect to* G_p if there exists a map $e : G_p \times G_p \to \mathbb{G}$ with the following properties:

1. Bilinear: that is, $e(aP, bQ) = e(P, Q)^{ab}$, for all $P, Q \in G_p$ and all $a, b \in \mathbb{Z}$.
2. Non-degenerate: The map does not send all pairs in $G_p \times G_p$ to the identity in $\mathbb{G}$.
3. Computable: There is an efficient algorithm to compute $e(P, Q)$ for any $P, Q \in G_p$.

We call them pairing curves because the usual way to implement the map e is using the Weil or Tate pairings [Men93]. In this case, the group $\mathbb{G}$ is the multiplicative group of a certain finite extension $\mathbb{F}_{q^k}$. The number k is called the *embedding degree* and is the smallest positive integer such that $p|(q^k - 1)$.

With such a map, the DDH problem is solvable in G_p. The non-degeneracy property of the Weil and Tate pairings implies that $e(P, P)$ is p-th root of unity, and then (P, aP, bP, cP) is a valid Diffie-Hellman quadruple if and only if $e(aP, bP) = e(P, cP)$.

It turns out then that the gap-CDH and CDH problems are polynomial time equivalent in G_p, since there exists a polynomial time algorithm replacing the DDH oracle solver. We use this fact positively to tightly relate the security of ACE-KEM, ECIES-KEM and PSEC-KEM to the CDH problem in these curves.

Another consequence of the map e is that solving DL problem in G_p can be transformed into solving the DL problem over the finite field $\mathbb{F}_{q^k}$, which can be computed using an index calculus algorithm running in subexponential time. This have been applied to attack the DL problem over supersingular curves in [MTV93, FR94]. We should have to take this into account when computing secure key sizes for each scheme.

Revisiting concrete security with respect to CDH. We already know that in pairing curves gap-CDH and CDH problems are equivalent. According to the results summarized in table 1, this implies that ACE-KEM and PSEC-KEM are straightforward secure with respect to the CDH problem. Indeed, they present a very tight reduction to the CDH, and the concrete security estimation is obtained by replacing L_{G_p} by doubling the time needed to compute the map e, which will be denoted by T_e.

In the case of the PSEC-KEM security proof in [Sho01], the solver of the CDH problem takes profit of a (t, q_D, q_K, ϵ) adversary against the IND-CCA security of PSEC-KEM to generate a list of $q_D + q_K$ elements containing the solution abP to the instance (P, aP, bP) with probability roughly ϵ. Since in a random curve the DDH problem is assumed to be intractable, we were forced to output an element of the list chosen uniformly at random, so the probability was decreased by a factor $q_D + q_K$. The reduction was then not tight. Since in a pairing curve DDH is efficiently solvable, we can find the correct value abP testing the entries on the list, obtaining thus a solver of CDH with probability roughly ϵ within time $t + 2(q_K + q_D)T_e$. Therefore, PSEC-KEM presents a very tight security reduction, allowing the use of shorter ciphertexts for the same level of security in a random curve, as we shall see in the next section. In table 3 these concrete security results are summarized.

Scheme	Assumption	Reduction
ACE-KEM	CDH	$\epsilon' \approx \epsilon$ $t' \approx t + q_K(4T_e + SR_q)$
ECIES-KEM	CDH	$\epsilon' \approx \epsilon$ $t' \approx t + q_K(4T_e + SR_q)$
PSEC-KEM	CDH	$\epsilon' \approx \epsilon$ $t' \approx t + 2(q_K + q_D)T_e$

Table 3: Security results over a pairing curve

Hardness of the CDH problem over pairing curves. When working with pairing curves we are restricting the set from which the elliptic curves are drawn, and then “some randomness” is lost with respect to the original key generation algorithm in these KEMs. Therefore, we obtain a new CDH problem, which can be called *CDH-pairing* (CDHP) problem, and that could be easier to solve than the CDH problem. Must we trust the hardness of the CDHP problem? We will ask positively to this question from two points of view. On the one hand, we take into account the current status of pairing curves in cryptography research. As the survey [DBS04] shows, they are being intensively applied by the cryptographic community to design new appealing protocols. The new assumptions arising in these protocols are at least stronger than the assumption that CDHP problem is hard. As a consequence, the trustness on these new assumptions implies trustness on the CDHP assumption.

On the other hand, using a technique due to Maurer [Mau94], it is possible to generate certain pairing elliptic curves with a cyclic group G_p for which CDH and DL problems are equivalent. The basic idea is to transport computing DL in G_p to computing DL in an auxiliary group whose number of points has a suitable smoothness bound B. In the latter, the computation of DL can be carried out with a generic algorithm on subgroups of small size. The running time of this reduction is $\mathcal{O}(B\cdot(\log(p)^2)$ group operations in G_p and field operations in $\mathbb{F}_p$ and $\mathcal{O}(\log(p)^3)$ calls to the CDH solver for G_p [MW00]. Since no attacks (different from Pollard ρ method) against the DL over pairing curves with a suitable embedding degree have been found, this theoretical equivalence gives a good taste about the hardness of the CDHP problem.

4 Efficiency analysis over pairing curves

Computing the security parameter. Let us assume that the IND-CCA security of any of these KEMs is (t, q_D, q_K, ϵ)-broken by some adversary $\mathcal{A}$. Since this adversary can be run repeatedly (with the same input and indepedent internal coin tosses), the expected time to distinguish a real encapsulation from random with advantage roughly 1 is t/ϵ. Thus, the security parameter of the scheme is $n_{\text{KEM}} = \log(t/\epsilon) = n + m$, where $n = \log t$ and $m = \log(1/\epsilon)$.

Usually, $q_D \leq 2^{30}$ (that is, up to one billion decryption queries are allowed), and $q_K \leq t = 2^{60}$. We also consider that evaluating a KDF function is a unit operation (that is, takes the same time as a 3-DES encryption). Using Miller's algorithm, computing a pairing in $E_{a,b}(\mathbb{F}_q)$ with embedding degree k can be done in $\mathcal{O}(k \log q)$ multiplications in $\mathbb{F}_q$ (cf. [Men93]), while computing a square root modulo q takes at most $\mathcal{O}(\log^2 q)$ multiplications in $\mathbb{F}_q$ (cf. [Coh93]). Assuming that a multiplication in $\mathbb{F}_q$ takes 10 times longer than one hash query, and that k is small, we obtain

$$t'_{\text{ECIES}} \approx t + 2^{60} \cdot 10 \cdot (4 \log q + \log^2 q) \approx 2^n + 2^{63} \cdot \log^2 q$$

for ACE and ECIES-KEM, and

$$t'_{\text{PSEC}} \approx t + 2^{61} \cdot 10 \cdot \log q \approx 2^n + 2^{64} \cdot \log q$$

for PSEC-KEM. In the following, we compute the exact security only for $t = 2^{80}$ for ECIES-KEM and PSEC-KEM, since the result for ACE-KEM is equal to the former. Setting $e = 0$ and $n = 80$, we obtain $n_{\text{ECIES}} = 80$ (respectively $n_{\text{PSEC}} = 80$), that is, a 2^{80} security level in each scheme. Let us compute the minimal parameter length to obtain this security level. An advantage roughly 1 in the IND-CCA game implies that the solver computes CDH succesfully with probability roughly 1 in time t'_{ECIES} (respectively t'_{PSEC}). Assuming that $|q| \approx 200$, then

$$t'_{\text{ECIES}} \approx 2^{80} + 2^{63} \cdot 2^{15} \quad \text{and} \quad t'_{\text{PSEC}} \approx 2^{80} + 2^{64} \cdot 2^{8}.$$

Both reductions are pretty meaningful and then, to get a 2^{80} security level on any of these KEMs it is enough a group G_p with at least a 2^{80} security of the CDH problem. If we make the *additional assumption* that the CDH and DL problems have comparable security, then we need a group G_p with $|p| \approx 160$. In the case of PSEC this is a great improvement, compared to a length of roughly 280 bits needed over a random curve.

On the other hand, using a technique due to Maurer [Mau94], one can build certain pairing elliptic curves with a cyclic group G_p for which CDH and DL problems are equivalent. As it was claimed in the previous section, the running time of this reduction is $\mathcal{O}(B \cdot (\log(p)^2)$ group operations in G_p and field operations in $\mathbb{F}_p$ and $\mathcal{O}(\log(p)^3)$ calls to the CDH solver for G_p [MW00]. Since in our case the computation of CDH instances is by far the most expensive operation, the reduction to the DL problem can be carried out with a 2^{22} factor decrease in security for all three schemes, therefore with a total time of $2^{80} \cdot 2^{22}$. Due to this somewhat small factor, the security of the scheme and the DL problem are tightly related. This allows to conclude that all three KEMs achieve provable security in the RO model, with the 2^{80} IND-CCA bound, in a group G_p with a 2^{102} security of the DL problem, provided that the DL to CDH reduction of [Mau94] holds for this group.

Curves	Related Problem	Assumptions	Minimal security level
Pairing curves	CDH	RO	2^{80}
Maurer paring cirves	DL	RO	2^{102}

Table 4: Discrete log KEMs for the 2^{80} security bound

Performance. It is the turn now to study the performance of each scheme over pairing curves. Since all three security reductions are very tight, we have seen that a 2^{80} IND-CCA security is achieved under a 2^{80} security level for the CDHP problem. Assuming that CDH and DL problems have comparable security in a pairing curve, and that the embedding degree is large enough to keep the DL infeasibility in $\mathbb{F}_{q^k}$ (in which case, the best attack known is to use the Pollard ρ method in G_p), a pairing curve $E_{a,b}(\mathbb{F}_q)$ with a group G_p with $|p| \approx 160$ is needed. However, as explained in the next section, the state of the art in pairing curves doesn't enable to claim that $|p| \approx |q|$, but $|p| \leq |q| \leq 2\,|p|$. The performance comparison among the three KEMs will be state then in bit units and in terms of the size of q. The results are presented in table 5.

Scheme	Operations in Enc	Operations in Dec	(K, C) length 16-Byte Keys	Public key length	Secret key length
ACE-KEM	5	3	$128 + 3\,|q|$	$4\,|q|$	$4\,|q|$
ECIES-KEM	**2**	**1**	**128 +** $\mathbf{|q|}$	$\mathbf{|q|}$	$\mathbf{|q|}$
PSEC-KEM	2	2	$128 + 2|q|$	$|q|$	$|q|$

Table 5: Performance features over pairing curves with 2^{80} security (lengths using a point compression technique)

We present in the next section some pairing curves where $|p| \approx |q| \approx 160$. With these values, the performance features of ACE-KEM and ECIES-KEM are equivalent to those of table 2, while in PSEC-KEM (K, C) length is reduced from 67 to 56 bytes, and public/secret keys are reduced from 35 to 20 bytes, thus obtaining a great improvement. From these values, we easily see that ECIES-KEM presents in every feature the best performance. Since all three KEMs base its security in the same problem, that is, CDHP problem, we conclude that ECIES-KEM should be considered the best option among these KEMs over pairing curves.

5 Examples of pairing curves

In this section we propose some curves in which the schemes can be performed, and we also discuss why they are suitable. Our aim is to give some examples of pairing curves $E_{a,b}(\mathbb{F}_q)$ to perform the schemes and where the CDH problem is assumed to be hard. We start describing the conditions that a candidate curve must hold. In the first place, we want pairing curves with a small embedding degree k, in order to obtain an efficient pairing computation. However, we cannot use embedding degrees as small as possible: we must take into account that the field $\mathbb{F}_{q^k}$ has to be large enough to fit into the demmanded security level. In our case, we are looking for a 2^{80} security level of the DL problem, which corresponds to $1024 \leq |q^k| \leq 1464$ using the estimates by Lenstra and Verheul [LV01] and the parameters used nowadays.

Unfortunately, curves with a small embedding degree are extremely rare, as shown in [BK98]. An exception are supersingular elliptic curves [Men93], which have $k \leq 6$. But, inasmuch as we are looking for small security parameters, only supersingular elliptic curves with $k = 6$ can fit into our purposes. However, it is not easy to generate such curves over prime finite fields, and the popular constructions use the field $\mathbb{F}_{3^m}$ (cf. [Gal01]).

Following [Gal04], an algorithm for generating curves with arbitrary k and with a large prime factor p of any form is proposed in [CP01]. Although it solves the embedding degree problem, it has the drawback that produces curves with $q > p^2$. For instance, this means that for $k = 10$ and $|p| \approx 160$, the algorithm returns a curve $E_{a,b}(\mathbb{F}_q)$ with $|q| > 320$. It is an active area of research to obtain pairing curves in which $|q| \approx |p|$ and $k \geq 7$. First steps in this direction have been taken, for instance, in [DEM02, BW03, SB04]. From [SB04] we take three curves with $k = 6$, and from the indications in section 4.1 in [BW03] we derive two curves with $k \geq 7$, which can be used to implement the schemes. These curves are presented in table 6.

6 Conclusions

In this paper we have studied the performance and security properties in pairing curves of the elliptic curve KEMs proposed to be standardized. First of all, we have summarized the previous properties claimed in the recent literature, and we have fixed some inexact results. Our main contribution has been to show that, on the one hand, despite their different behaviour from a security point of view in a general

$$E_{a,b}(\mathbb{F}_q) \ : \ y^2 = x^3 + ax + b; \quad |G_p| = p$$

k	6
q	730996464809526906653170358426443036650700061957 **(160 bits)**
a	-3
b	259872266527491431103791444700778440496305560566
p	730996464809526906653171213409755627912276816323 **(160 bits)**
k	6
q	801819385093403524905014779542892948310645897957 **(160 bits)**
a	-3
b	237567233982590907166836683655522398804119025399
p	801819385093403524905015674986573529844218487823 **(160 bits)**
k	6
q	4691249309589066676602717919800805068538803592363589996389 **(192 bits)**
a	-3
b	3112017650516467785865101962029621022731658738965186527433
p	2345624654794533338301358959942345572918215737398529094837 **(192 bits)**
k	12
q	920232877090278825268750317426886859921955755544079858267 71/85608987307 **(233 bits)**
a	9202328770902788252687503174268868433066055296961210513155893123/268268
b	16615350225787512515295967795006976 1
p	913438543748756510266439474266015799682269184 01 **(157 bits)**
k	10
q	2135990600736570192904215403867777226265004384865396904585 2/744353055146817624352242647863971020 81 **(320 bits)**
a	70368760954882
b	292300571380664269334019416279395865565081894912 0
p	2451995203788982715713779282071262924274547507211534336 1 **(185 bits)**

Table 6: Curves to implement KEMs equivalent to CDH

curve, they present a very tight security reduction to the CDH problem over pairing curves (CDHP); and, on the other hand, to suggest that ECIES-KEM should be considered the best option among these KEMs in environments using pairing curves. This is quite suprising, since ECIES-KEM in a random curve was refused to become a standard.

We have discussed the hardness of the CDHP problem; which even though it may be easier than the standard CDH problem, it is harder than the usual problems considered in provably secure schemes using pairings. Taking into account the state of the art in pairing curves generation, some concrete curves to perform the schemes for the current demmanded security levels have been presented. A major breakthrough in the efficient implementation of these KEMs would be to find methods to generate pairing curves with embedding degree at least 7 and $|q| \approx |p|$. In this case, using ECIES-KEM over pairings curves should be suitable not only for pairings cryptographic environments, but also for medium level security settings with constrained computing and memory capabilities. This is likely the case for many embedded systems, like smart cards.

References

[BF01] D. Boneh and M. Franklin. Identity-Based encryption from the Weil pairing. In *Advances in Cryptology – CRYPTO ' 01*, vol. 2139 of *Lecture Notes in Computer Science*, pp. 213–229, 2001.

[BK98] R. Balasubramanian and N. Koblitz. The improbability that an elliptic curve has subexponential discrete log problem under the Menezes-Okamoto-Vanstone algorithm. *Journal of Cryptology*, 11(2):141–145, 1998.

[BR93] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM CCS*, pp. 62–73. ACM Press, 1993.

[BW03] F. Brezing and A. Weng. Elliptic curves suitable for pairing based cryptography. Cryptology ePrint Archive, Report 2003/143, 2003. `http://eprint.iacr.org/`.

[Coh93] H. Cohen. *A Course in Computational Algebraic Number Theory*, vol. 138 of *Graduate Texts in Mathematics*. Springer, 1993.

[CP01] C. Cocks and R.G.E. Pinch. Identiy-based cryptosystems based on the Weil pairing, 2001. Unpublished manuscript.

[CS] Ronald Cramer and Victor Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. To appear at *SIAM Journal of Computing*. Available at `www.shoup.net`.

[DBS04] R. Dutta, R. Barua and P. Sarkar. Pairing-based cryptography : A survey. Cryptology ePrint Archive, Report 2004/064, 2004. `http://eprint.iacr.org/`.

[DEM02] R. Dupont, A. Enge and F. Morain. Building curves with arbitrary small mov degree over finite prime fields. Cryptology ePrint Archive, Report 2002/094, 2002. `http://eprint.iacr.org/`.

[Den02] A.W. Dent. ECIES-KEM vs. PSEC-KEM. Technical Report `NES/DOC/RHU/WP5/028/2`, NESSIE, 2002.

[FR94] G. Frey and H.G. Rück. A remark concerning m-divisibility and the discrete logarithm problem in the divisor class group of curves. *Mathematics of Computation*, 62:865–874, 1994.

[Gal01] S. Galbraith. Supersingular curves in cryptography. In *Advances in Cryptology – ASIACRYPT 2001*, vol. 2248 of *Lecture Notes in Computer Science*, pp. 495–513, 2001.

[Gal04] S. Galbraith. Pairings, 2004. Unpublished manuscript.

[JN03] A. Joux and K. Nguyen. Separating Decision Diffie-Hellman from Diffie-Hellman in cryptographic groups. *Journal of Cryptology*, 16(4):239–247, 2003.

[Jou00] A. Joux. A one round protocol for tripartite Diffie-Hellman. In *ANTS 2000*, vol. 1838 of *Lecture Notes in Computer Science*, pp. 385–394, 2000.

[LV01] A.K. Lenstra and E.R. Verheul. Selecting cryptographic key sizes. *Journal of Cryptology*, 14(4):255–293, 2001.

[LZ94] G.J. Lay and H.G. Zimmer. Constructing elliptic curves with given group order over large finite fields. In *ANTS '94*, vol. 877 of *Lecture Notes in Computer Science*, pp. 250–263, 1994.

[Mau94] U. Maurer. Towards the equivalence of breaking the Diffie-Hellman protocol and computing discrete logarithms. In *Advances in Cryptology — CRYPTO '94*, vol. 839 of *Lecture Notes in Computer Science*, pp. 271–281, 1994.

[Men93] A. Menezes. *Elliptic Curve Public Key Cryptosystems*, vol. 234 of *The Kluwer International Series in Engineering and Computer Science*. Kluwer Academic Publishers, 1993.

[MTV93] A.J. Menezes, T.Okamoto and S.A. Vanstone. Reducing elliptic curve logarithms to a finite field. *IEEE Transactions on Information Theory*, 39:1639–1646, 1993.

[MW00] U. Maurer and S. Wolf. The Diffie-Hellman protocol. *Designs, Codes, and Cryptography*, 19:147–171, 2000.

[Nes03] Nessie. NESSIE security report. version 2.0, 2003. `http://www.cryptonessie.org/`.

[OP01] T. Okamoto and D. Pointcheval. The gap-problems: a new class of problems for the security of cryptographic schemes. In *PKC 2001*, vol. 1992 of *Lecture Notes in Computer Science*, pp. 104–118, 2001.

[Pol78] J.M. Pollard. Monte carlo methods for index computation mod p. *Mathematics of Computation*, 32:918–924, 1978.

[SB04] M. Scott and P.S.L.M Barreto. Generating more MNT elliptic curves. Cryptology ePrint Archive, Report 2004/058, 2004. `http://eprint.iacr.org/`.

[Sho00] V. Shoup. Using hash functions as a hedge against chosen ciphertext attacks. In *Advances in Cryptology – EUROCRYPT ' 2000*, vol. 1807 of *Lecture Notes in Computer Science*, pp. 275–288, 2000.

[Sho01] V. Shoup. A proposal for an ISO standard for public key encryption. Technical Report 2.1, 2001.

[Sho04] V. Shoup. Draft ISO/IEC 18033-2: An emerging standard for public-key encryption. Technical report, ISO/IEC, 2004.

[Sma99] N. Smart. The discrete logarithm problem on elliptic curves of trace one. *Journal of Cryptology*, 12(3):193–196, 1999.

A Description of KEMs

$(\mathsf{pk},\mathsf{sk}) \leftarrow \mathcal{K}(E,P,p,\ell)$	$(K,C) \leftarrow \mathcal{E}(\mathsf{pk})$	$K \leftarrow \mathcal{D}(C,\mathsf{sk})$
1. $w,x,y,z \leftarrow \mathbb{Z}_p$	1. $r \leftarrow \mathbb{Z}_p$	1. Parse C as (C_1,C_2,C_3)
2. $W := wP$, $X := xP$,	2. $C_1 := rP$	2. $\alpha := Hash(C_1 \Vert C_2)$
$Y := yP$, $Z := zP$	3. $C_2 := rW$	3. $t := x + y\alpha$
3. $\mathsf{pk} := (E,P,p,W,X,Y,Z,\ell)$	4. $Q := rZ$	4. If $C_2 \neq wC_1$,
4. $\mathsf{sk} := (w,x,y,z,\mathsf{pk})$	5. $\alpha := Hash(C_1 \Vert C_2)$	output $\perp$ and halt
5. Output $(\mathsf{pk},\mathsf{sk})$	6. $C_3 := rX + \alpha rY$	5. If $C_3 \neq tC_1$,
	7. $C := (C_1,C_2,C_3)$	output $\perp$ and halt
	8. $K := KDF(C_1 \Vert Q)$	6. $Q := zC_1$
	9. Output (K,C)	7. $K := KDF(C_1 \Vert Q)$
		8. Output K

Description of ACE-KEM

$(\mathsf{pk},\mathsf{sk}) \leftarrow \mathcal{K}(E,P,p,\ell)$	$(K,C) \leftarrow \mathcal{E}(\mathsf{pk})$	$K \leftarrow \mathcal{D}(C,\mathsf{sk})$
1. $s \leftarrow \mathbb{Z}_p$	1. $r \leftarrow \mathbb{Z}_p$	1. $Q := sC$
2. $W := sP$	2. $C := rP$	2. If $Q = \mathcal{O}$
3. $\mathsf{pk} := (E,P,p,W,\ell)$	3. Set x the	output $\perp$ and halt
4. $\mathsf{sk} := (s,\mathsf{pk})$	x-coordinate of rW	3. Set x
5. Output $(\mathsf{pk},\mathsf{sk})$	4. $K = KDF(C \Vert x)$	x-coordinate of rW
	5. Output (K,C)	4. $K = KDF(C \Vert x)$
		5. Output K

Description of ECIES-KEM

$(\mathsf{pk},\mathsf{sk}) \leftarrow \mathcal{K}(E,P,p,\ell)$	$(K,C) \leftarrow \mathcal{E}(\mathsf{pk})$	$K \leftarrow \mathcal{D}(C,\mathsf{sk})$
1. $s \leftarrow \mathbb{Z}_p$	1. $r \leftarrow \{0,1\}^\ell$	1. Parse C as (C_1,C_2)
2. $W := sP$	2. $H := KDF(0_{32} \Vert r)$	2. $Q := sC_1$
3. $\mathsf{pk} := (E,P,p,W,\ell)$	3. Parse H as $t \Vert K$	3. $r := C_2 \oplus KDF(1_{32} \Vert C_1 \Vert Q)$
4. $\mathsf{sk} := (s,\mathsf{pk})$	4. $\alpha := t \bmod p$	4. $H := KDF(0_{32} \Vert r)$
5. Output $(\mathsf{pk},\mathsf{sk})$	5. $Q := \alpha W$	5. Parse H as $t \Vert K$
	6. $C_1 := \alpha P$	6. $\alpha := t \bmod p$
	7. $C_2 := r \oplus KDF(1_{32} \Vert C_1 \Vert Q)$	7. If $C_1 \neq \alpha P$,
	8. $C := (C_1,C_2)$	output $\perp$ and halt
	9. Output (K,C)	8. Output K

Description of PSEC-KEM

B PSEC parameter length over a random curve

We use the notation introduced in section 4. Let us assume the IND-CCA security of PSEC-KEM is (t,q_D,q_K,ϵ)-broken by some adversary $\mathcal{A}$. Then the security parameter of the scheme is $n = \log(t/\epsilon) = n + m$, where $n = \log t$ and $m = \log(1/\epsilon)$, and $q_D \leq 2^{30}$, $q_K \leq 2^{60}$. The concrete security reduction for PSEC-KEM over a random curve is $t' \approx t$ and $\epsilon' \approx \frac{\epsilon}{q_D+q_K}$. Setting $m = 0$ and $n = 80$ (that is, a 2^{80} security level in the scheme), we obtain

$$t' \approx t = 2^{80} \quad \text{and} \quad \epsilon' \approx 1/2^{60} = 2^{-60}.$$

From the last expression, an advantage roughly 1 in the IND-CCA game implies that the solver computes CDH succesfully with probability roughly 2^{-60} in time $t' = 2^{80}$. However, an algorithm solving CDH with probability roughly 1 is needed to find the parameter length. Running this algorithm with independent internal coin tosses 2^{60} times and returning the most frequent answer, CDH is solved with probability roughly 1. The computational effort needed to do this is $2^{60} \cdot 2^{80} = 2^{140}$. Assuming that CDH and DL problems have equivalent hardness over a random elliptic curve, we conclude that PSEC-KEM needs a subgroup G_p with $|p| \approx 280$, since the best attack known is using the Pollard ρ method.