THE SECURITY OF PSEC-KEM VERSUS ECIES-KEM

David Galindo[2], Sebastià Martín[1] and Jorge L. Villar[1]

[1] Dep. Matemàtica Aplicada IV. Universitat Politècnica de Catalunya

Campus Nord, c/Jordi Girona, 1-3, 08034 Barcelona

e-mail: {sebasm,jvillar}@ma4.upc.edu

[2] Institute for Computing and Information Sciences. Radboud University Nijmegen

P.O.Box 9010 6500 GL, Nijmegen, The Netherlands

e-mail: d.galindo@cs.ru.nl

The KEM-DEM methodology for designing hybrid encryption is being used in several projects aimed at evaluating or eventually standardizing cryptographic primitives. In this work we revisit one of the elliptic curve based KEMs studied to become standards, namely PSEC-KEM. Its security is based on different assumptions related to the elliptic curve discrete logarithm problem. First of all, we point out that PSEC-KEM has a non-tight *security reduction to the Computational Diffie-Hellman (ECDH) problem. This obvious fact has been surprisingly ignored in the literature, or even contradicted. This remark has a direct consequence: the security of PSEC-KEM with the current 160 key bits length for elliptic curve cryptography is not guaranteed using only the reduction to ECDH. Fortunately, we show that previous security proofs for PSEC-KEM can be straightforward modified to obtain a tight reduction to the so-called Gap-ECDH problem. This seems to be the first time that such a reduction has been released into the public domain. Finally, we raise some doubts on the widespread opinion that ECIES-KEM offers less security guarantees than PSEC-KEM.*

Keywords: standardization, public-key cryptography, provable security, key encapsulation mechanisms, efficient reductions.

INTRODUCTION

A key encapsulation mechanism (KEM) is a probabilistic algorithm that produces a random symmetric key and an asymmetric encryption of that key. Using this random key in a suitable encryption scheme (referred to as a data encapsulation mechanism-DEM), a secure hybrid encryption of arbitrary long messages is obtained. The problem of designing secure DEMs in the standard model is efficiently solved using well-known cryptographic techniques (cf. [CS03]). Therefore, designing secure hybrid encryption schemes within the KEM-DEM methodology is reduced to designing secure KEMs.

Three elliptic curve based KEMs have been under scrutiny for eventual standardization (for instance in [Sho04, NES03a, CRY03]), namely, ACE-KEM, ECIES-KEM and PSEC-KEM. Their security relies on different problems related to the elliptic curve discrete logarithm (ECDL). PSEC-KEM use the Random Oracle (RO) heuristic [BR93] in its security proof; while ACE-KEM and ECIES-KEM can be proven secure either in the RO model or the standard model. These schemes were first proposed as KEMs in [Sho01], the ISO standard draft for public key encryption by Victor Shoup, while in their original form they were submitted by IBM, Certicom and NTT corporations, respectively.

From a practical point of view, it is agreed that using a 160 bit key size in these KEMs suffices to meet the current demanded security level. Thus, this parameter does not allow to difference between them. Therefore, comparisons take into account the underlying hard problem used in their security proofs, their computational cost or their resistance against physical attacks (cf. [NES03a]). Regarding PSEC-KEM and ECIES-KEM, the literature mostly considers PSEC-KEM *more secure* than ECIES-KEM. The main reason argued is that, in the RO model, the latter reduces to a stronger hardness assumption than the former.

Our contribution. In the first place, we find out that *there is no evidence* in the literature supporting that PSEC-KEM with a 160 bits key size results in a secure implementation for the current security level. This is because its security reduction to the ECDH is not efficient enough. In fact, we show that using the existing security reduction, a key size providing security guarantees should be of roughly 280 bits. Surprisingly, one even finds in the literature claims stating that PSEC-KEM has a tight security reduction to ECDH (for instance in [Shi01, Men01]). Our next goal, then, is to find some reasonable security argument for using a 160 bit key length.

In second place, we show that the security proofs for PSEC-KEM can be straightforward modified to obtain a tight reduction to an easier but still conjectured hard problem, namely, the Gap-ECDH problem. The new reduction does not appear in the previous literature and enables the use of the short key size with security guarantees.

Finally, we review the security comparison between ECIES-KEM and PSEC-KEM at the light of these new considerations. It is generally agreed that PSEC-KEM offers better security guarantees than ECIES-KEM. The main reason argued to support this (see [Men01, Shi01, Den02, NES03b] is that the security of ECIES-KEM in the RO model is reduced to the Gap-ECDH problem, while PSEC-KEM is reduced to the potentially harder ECDH problem. However, the only practical reduction known for PSEC-KEM is the one we present, and that means using the tight reduction to the Gap-ECDH problem. On the other hand, security arguments in the standard model are known for ECIES-KEM, but not for PSEC-KEM. Our point of view is that we can not claim that one of the schemes is more secure than the other from a provable security point of view with the information currently available.

The rest of this work is organized as follows. In Section 2, the key ingredients about PSEC-KEM and ECIES-KEM are summarized as well as their main security properties found in the literature. In Section 3, a secure key size for PSEC-KEM is computed using the inefficient security reduction to the ECDH problem. Since the key size obtained is longer than desired, in Section 4 we show the reduction of PSEC-KEM to the Gap-ECDH problem, which provides security guarantees in the Random Oracle model for this KEM with a short key size. Finally, we end in Section 5 with some remarks about the security comparison between PSEC-KEM and ECIES-KEM.

PSEC AND ECIES SECURITY PROPERTIES AVAILABLE IN THE LITERATURE

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 from 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. [CS03]). Let us recall the definition of Key Encapsulation Mechanism (KEM).

Preliminaries

Definition 1 (Key Encapsulation Mechanism) *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 or a special symbol $\perp$ meaning there was a failure in the execution of the algorithm.*

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* $K = \mathcal{D}(1^\ell, C, \mathsf{sk})$.

Roughly speaking, a KEM is said to have *indistinguishability against chosen-ciphertext attacks* (IND-CCA) if an adversary with access to a decapsulation oracle can not distinguish between encapsulations of a fix and a randomly generated key (see [CS03] for details).

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 $\mathbb{F}_q$, $q > 3$. For finite fields with characteristic 2 or 3, the equation defining an elliptic curve takes different forms [Men93]. 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* (ECDL) is the problem of finding u when given (P, uP), where $u \leftarrow \mathbb{Z}_p$.
– The *computational Diffie-Hellman* problem (ECDH) is the problem of finding uvP when given (P, uP, vP), where $u, v \leftarrow \mathbb{Z}_p$.
– The *decisional Diffie-Hellman* problem (ECDDH) is the problem of distinguishing (P, uP, vP, uvP) from (P, uP, vP, wP), where $u, v, w \leftarrow \mathbb{Z}_p$.
– The *Gap Diffie-Hellman* problem (Gap-ECDH) is the problem of finding uvP when given (P, uP, vP) and an oracle $\mathcal{O}$ that correctly solves the decisional Diffie-Hellman problem, where $u, v, w \leftarrow \mathbb{Z}_p$.
– The *Oracle Diffie-Hellman* problem (ODH) is the problem of distinguishing $(P, uP, vP, H(uvP))$ from (P, uP, vP, K), where $H : \{0,1\}^* \to \{0,1\}^{hlen}$ is

a suitable hash function, $u, v \leftarrow \mathbb{Z}_p$, $K \leftarrow \{0,1\}^{hlen}$, and an oracle $\mathcal{H}_v$ which answers $H(zvP)$ when queried with zP is available. The query uP is not allowed.

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 [BSS99], so all these problems are assumed to be intractable. All of them are well established, except for the gap-ECDH problem, which was formally introduced in [OP01]. It is an open problem to establish all the relations between them. In fact, we rigorously know little more than the obvious reductions, which are ECDDH infeasible $\Rightarrow$ ECDH infeasible $\Rightarrow$ ECDL infeasible; Gap-ECDH infeasible $\Rightarrow$ ECDH infeasible and ODH infeasible $\Rightarrow$ ECDH infeasible. Thus, the best way known to attack these problems in a general elliptic curve is to solve ECDL. The fastest method for solving ECDL 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 ECDH problem is substantially easier than the ECDL problem, while the ECDDH problem appears to be easier than the ECDH problem in general. We refer the reader interested in the state of the art to [MW00].

Criterion 2 (Concrete security) *The efficiency of a 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 a hash or a KDF function; and the implied (t', ε')* solver *against the corresponding trusted cryptographic assumption. Such an attacker is referred as a $(t, \varepsilon, q_D, q_{\mathcal{O}_i})$ attacker for short. Following the usual terminology, the security reduction is* tight *if $\frac{t'}{\varepsilon'} \approx \frac{t}{\varepsilon}$, and* not tight *if $\frac{t'}{\varepsilon'} > q_D \frac{t}{\varepsilon}$. 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. The optimal tightness is achieved with* very tight *reductions.*

Criterion 3 (Security level) *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. The current demanded security level is 2^{80}.*

Previous security claims

A schematic description of these algorithms can be found in Table 1. The first step of the key generation algorithm in the 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, ℓ) as input.

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. $r \leftarrow \mathbb{Z}_p^*$	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 x-coord. of rW
5. Output $(\mathsf{pk}, \mathsf{sk})$	4. $K = KDF(C\|\|x)$	4. $K = KDF(C\|\|x)$
	5. Output (K, C)	Output K

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

Figure 1: Description of ECIES-KEM and PSEC-KEM

A so-called *key derivation function KDF* has been used in these KEMs. This function can be considered as a hash function for our purposes. In Table 1 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 is the time needed to check a Diffie-Hellman triple in G, 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], the authors do not take into account the time to compute

Scheme	Assumption	Reduction	Random Oracle	Reference
ECIES-KEM	gap-ECDH	$\varepsilon' \approx 2\varepsilon$ $t' \approx t + q_K(2L_G + SR_q)$	Yes	[CS03, Den02]
	ODH	$\varepsilon' \approx 2\varepsilon$ $t' \approx t$	No	[ABR01]
PSEC-KEM	ECDH	$\varepsilon' \approx \frac{2}{q_D+q_K}\varepsilon$ $t' \approx t$	Yes	[Sho01]

Table 1: IND-CCA KEMs exact security

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.

Since the ODH problem is a non-traditional Diffie-Hellman related problem, it has been paid more attention to the security of ECIES-KEM in the RO model. As we can see, ECIES-KEM presents a very tight reduction to the Gap-ECDH problem; while PSEC-KEM has a non tight reduction to the ECDH problem, due to the factor $q_D + q_K$ dividing ε. However, this feature in the reduction for PSEC has been mostly ignored by the literature, and even one can find claims stating that this reduction is tight (see [Men01, Shi01]). In this way, in Table 2 we find lengths for some parameters related to these KEMs that have been proposed for the 2^{80} IND-CCA security level in each scheme. To compute them, the usual way is to set that in a random curve Gap-ECDH or ECDH problems have complexity similar to the ECDL problem. The table assumes that the field size is similar to the group size $p \sim q$ which is usually assumed in the applications. However, we point out that in the proposals for a security parameter for PSEC-KEM in the literature has not been taken into account its security reduction. Indeed, we show in the next section that a larger parameter is needed if the only security argument available is the reduction to ECDH.

KEM	Problem	Exponentiations in Enc/Dec	(K, C) length 16-Byte Keys	Security parameter
ECIES	Gap-ECDH	**2/1**	**36**	160
PSEC	ECDH	2/2	56	160

Table 2: Claimed performance features over random curves (byte lengths using a point compression technique)

COMPUTING THE SECURITY PARAMETER FOR PSEC-KEM UNDER ECDH PROBLEM

Let us assume that the IND-CCA security of PSEC-KEM is $(t, q_D, q_K, \varepsilon)$-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 $n_{\mathrm{PSEC}} = \log(t/\varepsilon) = n + m$, where $n = \log t$ and $m = \log(1/\varepsilon)$.

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. Setting $m = 0$ and $n = 80$, we obtain $n_{\mathrm{PSEC}} = 80$, that is, a 2^{80} security level in each scheme. Using the reduction in Table 1, we obtain an algorithm $\mathcal{B}$ that solves ECDH with

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

From this expression, an advantage roughly 1 in the IND-CCA game implies that the solver computes ECDH successfully with probability roughly 2^{-60} in time $t' = 2^{80}$. However, an algorithm solving ECDH with probability roughly 1 is needed to find the parameter length. The reason is that in practice the computational hardness of these problems is estimated for algorithms effectively giving a solution. Running the algorithm $\mathcal{B}$ with independent internal coin tosses 2^{60} times and returning the most frequent answer, ECDH is solved with probability roughly 1. The computational effort needed to do this is $2^{60} \cdot 2^{80} = 2^{140}$. Then PSEC-KEM reduction to ECDH is only meaningful for subgroups in which the best way known to solve ECDH requires at least 2^{140} basic operations. Assuming that ECDH and ECDL 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 runs in exponential time $\sqrt{p}$.

Obviously, we are interested in using PSEC-KEM with a shorter security parameter, namely, with a 160 bits security parameter. To do this, a new security argument is needed, this time with a tight reduction.

PSEC-KEM UNDER THE GAP-ECDH ASSUMPTION

In the case of the PSEC-KEM security proof in [Sho01], the solver of the ECDH problem makes use of a $(t, q_D, q_K, \varepsilon)$ adversary against the IND-CCA security of PSEC-KEM to generate a list of $q_D + q_K$ elements containing the solution uvP to the instance (P, uP, vP) with probability roughly ε. Since in a random curve the ECDDH 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 is then not tight. However, if we are given access to a DDH oracle solver, then we can find the correct value uvP by testing the entries on the list, obtaining thus a solver of the Gap-ECDH problem with probability roughly ε within time $t + (q_K + q_D)L_G$, where L_G is the time needed to check a Diffie-Hellman tuple. Therefore, PSEC-KEM has a tight security reduction to the Gap-ECDH problem.

It is possible to give another security reduction, this time using the DDH oracle inside the reduction. It suffices with applying Theorem 3 in [Den03], but giving all parties DDH oracle access. In fact, the theorem by Dent gives a security reduction for a generalization of PSEC-KEM using an asymmetric encryption scheme with suitable properties.

Let us compute its security parameter for the 2^{80} security level under the Gap-ECDH problem. We can assume that evaluating the ECDDH oracle takes a time unit. Then, as done in the previous section, we set $n_{\text{PSEC}} = 80$ and $m = 0$, obtaining $t' \approx 2^{80} + 2^{60}$ and $\varepsilon' \approx 1/2$, obtaining a solver for the Gap-ECDH problem with expected time 2^{80}. Assuming that Gap-ECDH and ECDL problems have similar complexity, a 160 bits key size is enough to guarantee PSEC-KEM security, since the best attack known against ECDL is Pollard ϱ method.

IS PSEC-KEM MORE SECURE THAN ECIES-KEM?

There is a widespread opinion that PSEC-KEM is more secure than ECIES-KEM. As already explained in the introduction, it is argued that ECDH assumption is more reliable than Gap-ECDH assumption, and then PSEC-KEM offers more security than ECIES-KEM. Indeed, this argument was crucial in the decision in the NESSIE project to not include ECIES-KEM in its portfolio of recommended cryptographic primitives [NES03b]. Several sources from where the preference for PSEC-KEM can be traced are [Men01, Shi01, Den02].

However, the only argument for secure efficient implementations of PSEC-KEM is the reduction to the Gap-ECDH problem, which we have presented in this work. We think it is not longer possible to maintain that the assumptions supporting the security of PSEC-KEM are more reliable than the assumption of ECIES-KEM, since the former uses one of the assumptions of the latter to argue its security in efficient implementations. On the other hand, very little attention has attracted the fact that ECIES-KEM is a derivative of the scheme DHIES [ABR01], which has been proven secure in the standard model under a non-standard discrete logarithm related assumption. As far as we know, there is no security argument for PSEC-KEM in the standard model.

For these reasons, we think that it can not be set up a strict security comparison between these schemes with the information currently available. We hope that the concerns raised in this work will help to better fix the security properties of these schemes.

REFERENCES

[ABR01] M. Abdalla, M. Bellare and P. Rogaway. The Oracle Diffie-Hellman assumptions and an analysis of DHIES. In *CT-RSA 2001*, vol. 2020 of *Lecture Notes in Computer Science*, pp. 143–158, 2001.

[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.

[BSS99] I.F. Blake, G. Seroussi and N. Smart. *Elliptic Curves in Cryptography*, vol. 265 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, 1999.

[CRY03] CRYPTREC. Evaluation of cryptographic techniques, 2003. Information-Technology Promotion Agency, Japan. `http://www.ipa.go.jp/security/enc/CRYPTREC/index-e.html`.

[CS03] Ronald Cramer and Victor Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal of Computing*, 33:167–226, 2003.

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

[Den03] A.W. Dent. A designer's guide to KEMs. In *IMA Int. Conf. 2003*, vol. 2898 of *Lecture Notes in Computer Science*, pp. 133–151, 2003.

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

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

[Men01] A. Menezes. Evaluation of security level of cryptography: The revised version of PSEC-2 (PSEC-KEM). Technical report, CRYPTREC, 2001. `http://www.shiba.tao.go.jp/kenkyu/CRYPTREC/fy15/cryptrec20030424_outrep.html`.

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

[NES03a] NESSIE. NESSIE security report. version 2.0, 2003. `http://www.cryptonessie.org/`.

[NES03b] NESSIE. Portfolio of recommended cryptographic primitives. NESSIE consortium., 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.

[Shi01] R. Shipsey. PSEC-KEM. Technical Report `NES/DOC/RHU/WP5/001/a`, NESSIE, 2001.

[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. Available at `http://www.shoup.net/iso`.