Limits of a Conjecture on a Leakage-Resilient Cryptosystem

David Galindo*

CNRS/LORIA, Équipe Cassis, Bât. A, 54506 Vandoeuvre-lès-Nancy Cedex France

Srinivas Vivek*

University of Luxembourg, FSTC, 6 rue Richard Coudenhove-Kalergi, L-1359 Luxembourg

Abstract

Recently it was conjectured that an ElGamal-based public-key encryption scheme with stateful decryption resists lunch-time chosen ciphertext and leakage attacks in the only computation leaks information model. We give a non-trivial upper bound on the amount of leakage tolerated by this conjecture. More precisely, we prove that the conjecture does not hold if more than a $\left(\frac{3}{8} + o(1)\right)$ fraction of the bits are leaked at every decryption step, by showing a lunch-time attack that recovers the full secret key. The attack uses a new variant of the Hidden Number Problem, that we call Hidden Shares - Hidden Number Problem, which is of independent interest.

Keywords: leakage-resilient cryptography, ElGamal, hidden number problem, lattice-based attacks

1. Introduction

Leakage-resilient cryptography [1, 2] is a recent research line that aims at building countermeasures and/or defences against side-channel attacks while providing security reductions in a provable security manner. The methodology is as follows: an abstract model specifying the leakage data is chosen and the goal is to exhibit a reduction from a hardness assumption to a hypothetical side-channel adversary. The theory of leakage-resilient cryptography has witnessed a tremendous activity despite its short life. However, meeting the strongest security levels (resiliency against continual leakage attacks [3]) under the weakest assumptions (memory leakage [3]/auxiliary input [4]) is still out of reach from a practical point of view. To our knowledge existing schemes are inefficient when compared to their counterparts in the non-leakage setting; moreover, current leakage-resilient constructions are conceptually far more complex than those a practitioner currently finds in its cryptographic tool-box.

In this sense it is worth to mention the work by Kiltz and Pietrzak [5]. They propose BEG a pairing-based analogue of the ElGamal encryption scheme [6] with stateful decryption which is leakage-resilient against lunch-time chosen ciphertext and leakage attacks (CCLA1). The latter means that the classical distinguishing adversary against ElGamal is given access to decryption and leakage oracles only before the challenge ciphertext is given. The basic idea is to set the ElGamal secret key to be a group element (in contrast to an integer), and then multiplicatively share it. While splitting the secret key before decryption is a well-known technique, the novelty of this work is to propose to split a *group element* rather than an *integer*. It is shown that if the secret key length is κ, then the BEG scheme is secure against leakage of at most $\lambda \ll \frac{\kappa}{2}$ bits at every decryption step. More precisely, $\lambda < \frac{\kappa}{2} - \omega(\log \kappa)$ to make it infeasible to guess the remaining bits of the secret key by a brute force attack. Their proof uses the so-called Generic Bilinear Group Model [7].

The authors of [5] discuss the limitations of getting a security proof for a similar leakage-resilient property of ElGamal with stateful decryption over *arbitrary groups*. Nevertheless they conjecture that, for certain

*Corresponding author. Tel: +33 3 54 95 86 51.

Email addresses: david.galindo-chacon@loria.fr (David Galindo), srinivasvivek.venkatesh@uni.lu (Srinivas Vivek)

arbitrary groups where the Decisional Diffie-Hellman problem is hard, ElGamal with stateful decryption, that they call $\mathsf{EG}^\star$, might be resistant against side-channel attacks that abide by the Continual Leakage Split-State model.

Our Contribution In this work we impose a limit on the conjecture from [5], by proving that if a minimum of $\left(\frac{3}{8}+o\left(1\right)\right)\kappa$ bits are leaked at every invocation of the secret key, a CCLA1 attack exists against ElGamal with stateful decryption scheme $\mathsf{EG}^\star$ that recovers the secret key. Interestingly, our limit to the conjecture applies to any arbitrary instantiation of the underlying group. To achieve this result we define a new problem, called Hidden Shares - Hidden Number Problem, which is a close but new variant of the Hidden Number Problem [8].

Open Problem As far as we know, no (stateful) ElGamal-based public-key encryption scheme with constant public-key size exists in the literature offering (provable) resistance against continual leakage attacks (in the standard model). Finding such a scheme remains a challenging open question.

1.1. Known Hidden Number Problems

The Hidden Number Problem (**HNP**) was originally introduced by Boneh and Venkatesan [8] to demonstrate the hardness of computing the most significant bits of the secret key in the Diffie-Hellman key exchange mechanism. In its most generic form it can be described as follows. Let $f_\alpha : \mathcal{D} \to \mathcal{V}$, $\alpha \in \mathcal{A}$ be a family of maps between algebraic domains $\mathcal{D}$ and $\mathcal{V}$. The map is parametrized by α, that takes values from some set $\mathcal{A}$.

Definition 1. [Generic HNP [9]] Suppose some partial information about $f_\alpha\left(t\right) \in \mathcal{V}$ is given for several values of t, chosen uniform randomly from a subset $\mathcal{T} \subseteq \mathcal{D}$, find α.

Typically, $\mathcal{D}$ and $\mathcal{V}$ are finite fields $\mathbb{F}_p$. An instance of the Generic HNP problem is the Modular Inversion Hidden Number Problem (MIHNP) [10], also called $\mathbb{F}_p$-Inverse-HNP [9]. In MIHNP, we have $\mathcal{D} = \mathcal{T} = \mathbb{F}_p \backslash \{-\alpha\}$, $\mathcal{V} = \mathcal{A} = \mathbb{F}_p$, where

$$f_\alpha\left(t\right) = \mathrm{MSB}_{k,p}\left(\frac{1}{\alpha+t}\right),$$

where $\mathrm{MSB}_{k,p}(z)$ means the (integer representing the) k most significant bits of $z \pmod p$, and the elements of $\mathbb{F}_p$ are identified with integers of fixed bit-length $\lfloor \log_2 p \rfloor + 1$. In other words, we are given $n+1$ pairs $\left(t_i, \mathrm{MSB}_{k,p}\left(1/\left(\alpha+t_i\right)\right)\right)$, for $i = 0, \ldots, n$, where $t_i \in \mathbb{F}_p \backslash \{-\alpha\}$ are chosen uniform randomly and independently, and the goal is to find a polynomial-time (in $\log p$) algorithm to recover $\alpha \in \mathbb{F}_p$ completely. The hardness of variants of this problem has been used to construct efficient algebraic PRNGs and MACs [10]. A close variant of MIHNP is one where

$$f_{\alpha,\beta}\left(t\right) = \mathrm{MSB}_{k,p}\left(\frac{\beta}{\alpha+t}\right).$$

Another problem related to the HNP was addressed in [11] and it is called the "HNP with hidden multipliers" (HM-HNP). In [9], the same problem is referred to as $\mathbb{F}_p$-Approx-HNP.

Definition 2. [HM-HNP [11, 12, 13]] Given n pairs $\left(\mathrm{MSB}_{k,p}\left(t_i\right), \mathrm{MSB}_{k,p}\left(\alpha t_i\right)\right)$, where $\alpha \in \mathbb{F}_p$ and $t_i \xleftarrow{\$} \mathbb{F}_p$, find α.

Definition 3. [HS-HNP] Given n pairs $\left(\mathrm{MSB}_{k,p}\left(t_i\right), \mathrm{MSB}_{k,p}\left(\frac{\alpha}{t_i}\right)\right)$, where $\alpha \in \mathbb{F}_p$ and $t_i \xleftarrow{\$} \mathbb{F}_p \backslash \{0\}$, find α.

To our knowledge the HS-HNP has not been explicitly addressed before. Note that unlike the Generic HNP, in HS-HNP (also HM-HNP) only a partial information about the values t_i is given. The name "hidden shares" follows from the fact that t_i and $\frac{\alpha}{t_i} \pmod p$ are two multiplicative shares of $\alpha \in \mathbb{F}_p$.

We stress that in spite of similarities in the techniques used to solve various variants of HNP, there are significant differences in the technical details. FoThese differences eventually reflect in the amount of partial information (value of k) required to solve the problem with a reasonable success probability. For comparison, the values of k required for HNP, MIHNP, and HM-HNP are $\sqrt{\log p}+\log\log p$, $\frac{1}{3}\log p$, and $\frac{4}{5}\log p$, respectively [8, 10, 11]. As we shall see in Section 3, HS-HNP requires the value of k to be at least $\frac{3}{4}\log p$.

2. The Conjecture

We denote the field of prime order p by $\mathbb{F}_p$. The field $\mathbb{F}_p$ is identified with $\mathbb{Z}_p$, and we use them interchangeably. By $\mathrm{MSB}_{k,p}(z)$, we mean the (integer representing the) k most significant bits of $z \pmod p$, where the elements of $\mathbb{Z}_p$ are represented by integers of fixed bit-length $m = \lfloor \log p \rfloor + 1$. For instance, $\mathrm{MSB}_{2,7}(3) = 1$. The notation "log" always refers to logarithm to the base 2. Finally, we denote by $c, d \xleftarrow{\$} \mathbb{Z}_p$ the sampling of values c and d uniform randomly and independently from the set $\mathbb{Z}_p$.

2.1. KEM and the leakage model

Formally, KEM consists of three algorithms KG, Enc and Dec. The key generation algorithm KG on input a security parameter κ produces a pair of public and secret keys (pk, sk). The encapsulation algorithm Enc, taking only pk as input, outputs an encryption C of a key K. The decapsulation algorithm Dec on inputs sk and C outputs K. The goal of an adversary is to distinguish the encryption of a given key from that of a random key. An adversary may be a Chosen Plaintext Attack- (CPA-) adversary if it has no access to a decryption oracle. If it does have access to such an oracle then it is called a Chosen Ciphertext Attack (CCA) adversary. CCA adversaries can be further classified into CCA1- or CCA2-adversaries. A CCA1-adversary cannot query the decapsulation oracle after obtaining the challenge ciphertext.

In a KEM *with stateful decapsulation*, $\mathsf{KEM} = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec1}, \mathsf{Dec2})$, the decapsulation algorithm Dec is split into two parts Dec1 and Dec2 executed consecutively. Each such algorithm uses different parts of the memory that therefore leak independent side-channel data, thus obeying to the *Only Computation Leaks/Split-State* model [1, 14]. The secret key of KEM now consists of two parts $sk_i = (\sigma_i, \sigma'_i)$, each part residing on a different portion of the memory. Dec1 can access σ_i, while Dec2 has access only to σ'_i. The decapsulation procedures may update the secret key sk_i to $sk_{i+1} = \left(\sigma_{i+1}, \sigma'_{i+1}\right)$ after each access to it. We refer to one execution of the decapsulation query as a *round*.

The leakage in each round is modelled as the output of two adversarially chosen efficiently computable functions $f_i(\cdot)$ and $g_i(\cdot)$ whose output length is bounded by λ bits each; λ is the *leakage parameter*. The function $f_i(\sigma_{i-1}, r_i)$ models the leakage produced when computing with σ_{i-1} and the internal randomness r_i used by Dec1. The function $g_i(\sigma'_{i-1}, w_i, r'_i)$ models the leakage wrt. σ'_{i-1}, the internal randomness r'_i used by Dec2, and the information w_i shared between Dec1 and Dec2. The KEM is said to be (κ, λ) secure under the Chosen Ciphertext with Leakage Attacks 1 (CCLA1) if the scheme remains secure even when an adversary can obtain λ bits of leakage from each of the two functions f_i and g_i, at every decapsulation query.

Stateful ElGamal KEM. Let the output of $\mathsf{Gen}(\kappa, \lambda)$ be a cyclic group $\mathbb{G} = \langle g \rangle$ of prime order p, generated by g. Let $\mathsf{EG}^\star = (\mathsf{KG}_{\mathsf{EG}^\star}, \mathsf{Enc}_{\mathsf{EG}^\star}, \mathsf{Dec1}_{\mathsf{EG}^\star}, \mathsf{Dec2}_{\mathsf{EG}^\star})$ be the stateful KEM defined in Section 3.1 in [15] as follows:

1. $\mathsf{KG}_{\mathsf{EG}^\star}(\kappa, \lambda)$: Compute $(\mathbb{G}, g, p) \leftarrow \mathsf{Gen}(\kappa, \lambda)$. Choose random $x \xleftarrow{\$} \mathbb{Z}_p$ and $\sigma_0 \xleftarrow{\$} \mathbb{Z}_p^*$. Set $h = g^x$ and $\sigma'_0 = x \cdot \sigma_0^{-1} \pmod p$. The public key is $pk = (\mathbb{G}, g, p, h)$, and the secret key is $sk = (\sigma_0, \sigma'_0)$.
2. $\mathsf{Enc}_{\mathsf{EG}^\star}()$: Choose random $l \xleftarrow{\$} \mathbb{Z}_p$. The ciphertext is $C = g^l$, and the key is $K = h^l$.
3. $\mathsf{Dec1}_{\mathsf{EG}^\star}(\sigma_{i-1}, C)$: Choose random $r_i \xleftarrow{\$} \mathbb{Z}_p^*$. Set $\sigma_i \leftarrow \sigma_{i-1} \cdot r_i \pmod p$, and $K'_i = C^{\sigma_i}$. Return (r_i, K'_i).
4. $\mathsf{Dec2}_{\mathsf{EG}^\star}\left(\sigma'_{i-1}, (r_i, K'_i)\right)$: Set $\sigma'_i \leftarrow \sigma'_{i-1} \cdot r_i^{-1} \pmod p$, and $K = K_i'^{\sigma'_i}$. Return K as the shared secret key.

Claim [5, Conjecture 1] $\mathsf{EG}^\star$ is CCLA1 secure if $p - 1$ has a large prime factor.

The above statement is incomplete if the leakage parameter λ (i.e., the amount of leakage from each of $\mathsf{Dec1}_{\mathsf{EG}^\star}$ and $\mathsf{Dec2}_{\mathsf{EG}^\star}$) is not specified. For instance, the conjecture could hold for $\lambda \ll \frac{\log p}{2}$. More precisely, $\lambda < \frac{\log p}{2} - \omega(\log \log p)$ to make it infeasible to guess the remaining bits by a brute force attack. Since leakage is modelled by functions $f_i(\sigma_{i-1}, r_i)$ and $g_i\left(\sigma'_{i-1}, (r_i, K'_i)\right)$, it is easy see that if $\lambda > \log p/2$, then one can completely recover r_i in the i^{th} round and also some bits of the initial state (σ_0, σ'_0). In at most $2\lceil \log p \rceil$ rounds, the secret key x can be fully recovered. This is the trivial attack.

Relationship to the HS-HNP. In the following we argue that it is possible to obtain 2λ most significant bits of each of the two shares of the secret key in $\mathsf{EG}^\star$. This is a consequence of the fact that λ bits of σ_i

(respectively, σ'_i) can be leaked from each of $f_i(\sigma_{i-1}, r_i)$ and $f_{i+1}(\sigma_i, r_{i+1})$ (respectively, $g_i\left(\sigma'_{i-1}, (r_i, K'_i)\right)$ and $g_{i+1}\left(\sigma_i, \left(r_{i+1}, K'_{i+1}\right)\right)$), where $i \geq 1$. After $2n+2$ rounds of execution of $(\mathsf{Dec1}_{\mathsf{EG}^\star}, \mathsf{Dec2}_{\mathsf{EG}^\star})$, an adversary will be able to obtain $n+1$ pairs $\left(\mathrm{MSB}_{2\lambda,p}\left(\sigma_{2i+1}\right), \mathrm{MSB}_{2\lambda,p}\left(\sigma'_{2i+1}\right)\right)$ of the secret key, where $i = 0, \ldots, n$.

The above observation leads us to an instance of HS-HNP (Definition 3) with $\alpha = x$, $t_i = \sigma_{2i+1}$ and $\frac{\alpha}{t_i} \equiv \sigma'_{2i+1} \pmod p$, where $i = 0, \ldots, n$. Note that t_i is uniform random and independent in $\mathbb{Z}_p^*$. Hence investigating HS-HNP is a natural approach to resolve the above conjecture. This is the topic of the next section.

Let us notice that in [15] it is wrongly stated that breaking the CCLA1 security of the stateful ElGamal KEM is related to the Hidden Multipliers - HNP (Definition 2). That is, using our notation, the HM-HNP boils down to an adversary that is getting leakage on t_i and αt_i, which corresponds to $t_i = \sigma_{2i+1}$ and $\alpha t_i \equiv \sigma'_{2i+1} t_i^2 \pmod p$, where $i = 0, \ldots, n$. Apparently, the value αt_i is never computed in the ElGamal KEM nor can possibly be computed by leakage functions.

3. Hidden Shares - Hidden Number Problem

Let p be an m-bit integer. We have $m = \lfloor \log p \rfloor + 1$. Let $y_i = 2^{m-k} \cdot \mathrm{MSB}_{k,p}\left(t_i\right)$ and $b_i = 2^{m-k} \cdot \mathrm{MSB}_{k,p}\left(\frac{\alpha}{t_i}\right)$ in Definition 3. Let $t_i = y_i + \delta_i$, $\frac{\alpha}{t_i} \pmod p = b_i + \epsilon_i$, where $0 \leq \delta_i, \epsilon_i < 2^{m-k}$, where $i = 0, \ldots, n$. Note that the integers y_i and b_i are known, while the integers δ_i and ϵ_i are unknown. We have

$$\left(y_i + \delta_i\right)\left(b_i + \epsilon_i\right) \equiv \alpha \pmod p. \tag{1}$$

Since α is an unbounded variable, we will eliminate it from the $n+1$ equations represented by (1). We obtain

$$\left(y_i + \delta_i\right)\left(b_i + \epsilon_i\right) - \left(y_0 + \delta_0\right)\left(b_0 + \epsilon_0\right) \equiv 0 \pmod p.$$

On rearranging the terms, we obtain a set of n equations (for $i = 1, \ldots, n$) as

$$\begin{aligned}(-1)\,\delta_0\epsilon_0 + (1)\,\delta_i\epsilon_i + (-b_0)\,\delta_0 + (b_i)\,\delta_i + (-y_0)\,\epsilon_0 + (y_i)\,\epsilon_i & \\ + (y_i b_i - y_0 b_0) \quad &\equiv \quad 0 \pmod p.\end{aligned} \tag{2}$$

For the sake of clarity, let us denote the coefficients in the above $\left(i^{\text{th}}\right)$ relation by A_i, B_i, C_i, D_i, E_i, F_i, G_i, respectively in the order. Equation (2) can now be rewritten as

$$A_i\delta_0\epsilon_0 + B_i\delta_i\epsilon_i + C_i\delta_0 + D_i\delta_i + E_i\epsilon_0 + F_i\epsilon_i + G_i \equiv 0 \pmod p. \tag{3}$$

Note again that the only unknowns in the above equation are δ_0, δ_i, ϵ_0 and ϵ_i. Equation (3) can be rewritten over the integers as

$$A_i\delta_0\epsilon_0 + B_i\delta_i\epsilon_i + C_i\delta_0 + D_i\delta_i + E_i\epsilon_0 + F_i\epsilon_i + G_i + p \cdot \mu_i = 0, \tag{4}$$

where the μ_i are unknowns, and $i = 1, \ldots, n$. The quantities μ_i are of little interest compared to that of δ_i and ϵ_i. We shall now construct a lattice that captures the relations defined by (4). Since (4) contains non-linear terms like $\delta_0\epsilon_0$ and $\delta_i\epsilon_i$, we "linearize" the relation by treating the non-linear terms as a separate variable. It is also desirable to have the solution we are looking for correspond to a "short" vector in the lattice. Our construction is similar to the one in [10, Section 3.1].

3.1. Setting up the lattice

The lattice we construct has dimension $4n+4$ and it is represented by a $(4n+4) \times (4n+4)$ matrix M consisting of rational entries. The lattice is generated as the row span of the matrix M. The structure of M is as follows:

$$M = \begin{pmatrix} J & R \\ 0 & P \end{pmatrix}, \tag{5}$$

where J and P are diagonal matrices having dimensions $(3n+4) \times (3n+4)$ and $n \times n$, respectively. Matrix R has dimensions $(3n+4) \times n$. The rows of M correspond to the terms present in the n relations represented

Figure 1: The matrix M for the case $n = 2$. Let $\phi = 2^{k-m}$.

$$M = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & G_1 & G_2 \\
0 & \phi & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_1 & C_2 \\
0 & 0 & \phi & 0 & 0 & 0 & 0 & 0 & 0 & 0 & D_1 & 0 \\
0 & 0 & 0 & \phi & 0 & 0 & 0 & 0 & 0 & 0 & 0 & D_2 \\
0 & 0 & 0 & 0 & \phi & 0 & 0 & 0 & 0 & 0 & E_1 & E_2 \\
0 & 0 & 0 & 0 & 0 & \phi & 0 & 0 & 0 & 0 & F_1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \phi & 0 & 0 & 0 & 0 & F_2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \phi^2 & 0 & 0 & A_1 & A_2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \phi^2 & 0 & B_1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \phi^2 & 0 & B_2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & p & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & p
\end{pmatrix}$$

by (4). The first row is associated with the constant term, the next $n+1$ rows correspond to the variables δ_i, next $n+1$ rows with ϵ_i, while the further $n+1$ rows correspond to $\delta_i\epsilon_i$. The last n rows are associated with the terms μ_i. Each of the last n columns of M correspond to a relation in (4), while the first $3n+4$ columns are associated with the inverse of an upper bound on the size of the quantities 1, δ_i, ϵ_i and $\delta_i\epsilon_i$ (in the solution we are interested in). In what follows, we give a complete description of the matrix M.

Let $P[i', j']$ denote the entry in the i'^{th} row and the j'^{th} column of the matrix P. The matrix P has p on all of its main diagonal, i.e. $P[i, i] = p$ for $1 \le i \le n$. The i^{th} row of P corresponds to the term μ_i in (4). The diagonal matrix J has $J[1, 1] = 1$ (for the constant term), $J[i', i'] = 2^{k-m}$ for $2 \le i' \le 2n+3$ (for terms δ_i and ϵ_i), and $J[i', i'] = 2^{2(k-m)}$ for $2n+4 \le i' \le 3n+4$ (for terms $\delta_i\epsilon_i$). As we shall later see, these entries of J are required to bound the norm of the vector corresponding to our solution. Each of the n relations of (4) is described in matrix R (excluding terms $p \cdot \mu_i$, which are described by matrix P). The entry $R[i', j']$ is the coefficient of the term corresponding to row i' in the j'^{th} relation. Hence the columns of matrices R and P together completely describe the system of equations (4).

As an illustration, the matrix M is described for the case $n = 2$ in Figure 1, where $\phi = 2^{k-m}$. The terms corresponding to the 12 rows of M are (from top to bottom) 1, δ_0, δ_1, δ_2, ϵ_0, ϵ_1, ϵ_2, $\delta_0\epsilon_0$, $\delta_1\epsilon_1$, $\delta_2\epsilon_2$, μ_1, and μ_2, respectively.

Let $\epsilon_i = e_i$, $\delta_i = d_i$ $(0 \le i \le n)$ and $\mu_i = u_i$ $(1 \le i \le n)$ be a solution to the system of equations (4), where $0 \le e_i, d_i < 2^{m-k}$. Note that such a solution exists from the way the system was constructed. Let v be a (row) vector, of length $4n+4$, defined as $v = \langle 1, d_0, \ldots, d_n, e_0, \ldots, e_n, d_0e_0, \ldots, d_ne_n, u_1, \ldots, u_n \rangle$. Since e_i, d_i and u_i satisfy the system (4), it is easy to see that

$$v \cdot M = \left\langle 1, \frac{d_0}{2^{m-k}}, \ldots, \frac{d_n}{2^{m-k}}, \frac{e_0}{2^{m-k}}, \ldots, \frac{e_n}{2^{m-k}}, \frac{d_0e_0}{2^{2(m-k)}}, \ldots, \frac{d_ne_n}{2^{2(m-k)}}, 0, \ldots, 0 \right\rangle. \tag{6}$$

Note that the vector $v \cdot M$ has a leading 1, and n trailing zeros. Its length is $4n+4$ and its Euclidean norm $\|v \cdot M\|_2$ satisfies

$$\|v \cdot M\|_2 < \sqrt{3n+4}. \tag{7}$$

If we are able to find the vector $v \cdot M$, then we can readily recover the values d_i, e_i, and hence solve the system of (4). Since the vector $v \cdot M$ has a bounded length, we can try to choose a value for k such that the resulting lattice has a sufficiently large determinant, and hence is unlikely to have many vectors shorter than $\|v \cdot M\|_2$. Then we can run a lattice reduction algorithm, say LLL [16], to obtain reduced basis vectors. We then hope that by exploiting the structure of the vector $v \cdot M$, we can obtain it as a simple combination of a few short basis vectors. Since it appears hard to rigorously bound the probability of failure, we had to content ourselves with the heuristic arguments, as it is common with these techniques.

We now give an estimate for the suitable values of k. The Gaussian heuristic gives an estimate of the expected number of lattice points in a sphere of given volume. Consider a full rank lattice L' in $\mathbb{R}^{n'}$whose determinant is $\det(L')$. Let $V_{n'}(r')$ denote the volume of an n'-ball $S_{n'}(r')$ of radius r', centered at the origin.

Lemma 4. [Gaussian Heuristic [17, pp. 28]] *The "expected" number of lattice points of L' in $S_{n'}$ is $\frac{V_{n'}(r')}{\det(L')}$.*

An explicit formula for $V_{n'}(r')$ is

$$V_{n'}(r') = \frac{\pi^{\frac{n'}{2}}}{\Gamma(\frac{n'}{2}+1)} r'^{n'}, \tag{8}$$

where $\Gamma(\cdot)$ is the Gamma function. In our case, the determinant of the lattice M in (5), $\det(M)$, is

$$\det(M) = \frac{p^n}{2^{(m-k)(4n+4)}} \geq \frac{2^{(m-1)n}}{2^{(m-k)(4n+4)}}. \tag{9}$$

The above inequality follows from the fact that p is an m-bit integer. Note that the value of $\det(M)$ increases with the value of k. By (7), $\|v \cdot M\|_2 < \sqrt{3n+4}$. We would like to choose such a value for k so that the expected number of lattice points of M in $S_{4n+4}(\sqrt{3n+4})$ is at most one. By Lemma 4 and (9), it suffices if

$$\frac{2^{(m-1)n}}{2^{(m-k)(4n+4)}} \geq V_{4n+4}(\sqrt{3n+4}).$$

On rearranging the above inequality, we get

$$\frac{k}{m} \geq \frac{3+\frac{4}{n}}{4+\frac{4}{n}} + \left(\frac{n + \log_2\left(V_{4n+4}(\sqrt{3n+4})\right)}{m\,(4n+4)} \right). \tag{10}$$

From (8), $V_{4n+4}(\sqrt{3n+4}) = \frac{\pi^{2n+2}}{(2n+2)!}(3n+4)^{2n+2}$. Therefore, in (10),

$$\frac{n + \log_2\left(V_{4n+4}(\sqrt{3n+4})\right)}{m\,(4n+4)} = O\left(\frac{\log n}{m}\right).$$

If $1 \ll n \ll m$, we obtain $k = \left(\frac{3}{4} + o(1)\right) m$.

Finally we can conclude that there is an efficient (heuristic) method to solve the HS-HNP problem with $k = \left(\frac{3}{4} + o(1)\right) m$.

Remark 5. On applying the above method to attack the scheme $\mathsf{EG}^\star$ (c.f. Section 2.1), we obtain that the stateful KEM $\mathsf{EG}^\star$ is *not* CCLA1 secure if the leakage parameter $\lambda \geq \left(\frac{3}{8} + o(1)\right) \log p$. We would like to note that our attack does not exploit any information about the elements of the underlying group $\mathbb{G}$ in $\mathsf{EG}^\star$. Hence this attack will work for any instantiation of the group.

3.2. Implementation details

We have implemented the above method to solve HS-HNP in the "PARI/GP" computer algebra system [18]. The experiments were run on an Intel(R) Core i7-2600 CPU with 4 GB RAM, running cygwin (ix86/GMP-4.2.1 kernel) 32-bit version. The results are given in Table 1. For every (n, m) pair, 10 random m-bit primes p were chosen. For each p, 10 random hidden numbers α were chosen. The running time reported is averaged over 100 (p, α) pairs for each row of the table. For lattice reduction, we have used the routine `qflll` in PARI/GP.

In the experiments, we have observed that there is one (reduced) basis vector of very low norm, of order $\frac{1}{2^{m-k}}$. This is because the rows of the matrix M (Equation (5)) corresponding to the terms $\delta_i \epsilon_i$ $(0 \leq i \leq n)$ can add up to produce the vector $\left\langle \underbrace{0, \ldots, 0}_{2n+3}, \underbrace{2^{k-m}, \ldots, 2^{k-m}}_{n+1}, \underbrace{0, \ldots, 0}_{n} \right\rangle$.

We have also observed that there are many (reduced) basis vectors with norm of about $\sqrt{3n+4}$. In order to overcome these issues, we "randomize" the relations from Equation (2) by multiplying each relation by a random independent element of $\mathbb{Z}_p^*$. This will not alter the solution set and yet there will be very short vectors of the order $\frac{1}{2^{m-k}}$. But the experiments suggest that there will be only one (reduced) basis vector (second shortest) of norm about $\sqrt{3n+4}$, and we can get the required values of d_i and e_i from the corresponding entries of the basis vector. This heuristic has *not* failed even once for the 900 (p, α) pairs of Table 1, provided k is chosen according to Equation (10).

Acknowledgements. The first author has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement n° 258865.

Table 1: HS-HNP: implementation results

n	m (bits)	k (bits)	$\frac{k}{m}$	dimension(M)	time (sec)
2	256	216	0.844	12	0.031
2	512	429	0.838	12	0.087
2	1024	856	0.836	12	0.290
5	256	205	0.800	24	0.507
5	512	408	0.797	24	1.367
5	1024	813	0.794	24	4.468
10	256	200	0.781	44	4.144
10	512	398	0.777	44	10.911
10	1024	794	0.775	44	32.395

References

[1] S. Micali, L. Reyzin, Physically observable cryptography (extended abstract), in: M. Naor (Ed.), TCC, Vol. 2951 of LNCS, Springer, 2004, pp. 278–296.

[2] S. Dziembowski, K. Pietrzak, Leakage-resilient cryptography, in: FOCS, IEEE, 2008, pp. 293–302.

[3] Y. Dodis, K. Haralambiev, A. López-Alt, D. Wichs, Cryptography against continuous memory attacks, in: FOCS, IEEE Computer Society, 2010, pp. 511–520.

[4] Y. Dodis, S. Goldwasser, Y. T. Kalai, C. Peikert, V. Vaikuntanathan, Public-key encryption schemes with auxiliary inputs, in: D. Micciancio (Ed.), TCC, Vol. 5978 of LNCS, Springer, 2010, pp. 361–381.

[5] E. Kiltz, K. Pietrzak, Leakage resilient elgamal encryption, in: M. Abe (Ed.), ASIACRYPT, Vol. 6477 of LNCS, Springer, 2010, pp. 595–612.

[6] T. E. Gamal, A public key cryptosystem and a signature scheme based on discrete logarithms, IEEE Transactions on Information Theory 31 (4) (1985) 469–472.

[7] D. Boneh, X. Boyen, E.-J. Goh, Hierarchical identity based encryption with constant size ciphertext, in: R. Cramer (Ed.), EUROCRYPT, Vol. 3494 of LNCS, Springer, 2005, pp. 440–456.

[8] D. Boneh, R. Venkatesan, Hardness of computing the most significant bits of secret keys in diffie-hellman and related schemes, in: N. Koblitz (Ed.), CRYPTO, Vol. 1109 of LNCS, Springer, 1996, pp. 129–142.

[9] I. E. Shparlinski, Playing "hide-and-seek" with numbers: the hidden number problem, lattices, and exponential sums, in: Symposia in Applied Mathematics, Vol. 62, American Mathematical Society, 2005, pp. 153–177.

[10] D. Boneh, S. Halevi, N. Howgrave-Graham, The modular inversion hidden number problem, in: C. Boyd (Ed.), ASIACRYPT, Vol. 2248 of LNCS, Springer, 2001, pp. 36–51.

[11] N. Howgrave-Graham, P. Q. Nguyen, I. Shparlinski, Hidden number problem with hidden multipliers, timed-release crypto, and noisy exponentiation, Math. Comput. 72 (243) (2003) 1473–1485.

[12] N. Howgrave-Graham, N. P. Smart, Lattice attacks on digital signature schemes, Des. Codes Cryptography 23 (3) (2001) 283–290.

[13] P. Q. Nguyen, I. Shparlinski, The insecurity of the digital signature algorithm with partially known nonces, J. Cryptology 15 (3) (2002) 151–176.

[14] F.-H. Liu, A. Lysyanskaya, Tamper and leakage resilience in the split-state model, in: R. Safavi-Naini, R. Canetti (Eds.), CRYPTO, Vol. 7417 of LNCS, Springer, 2012, pp. 517–532.

[15] E. Kiltz, K. Pietrzak, Leakage resilient elgamal encryption, slides of ASIACRYPT 2010 presentation (2010).

[16] A. K. Lenstra, H. W. Lenstra, L. Lovász, Factoring polynomials with rational coefficients, Mathematische Annalen 261 (4) (1982) 515–534.

[17] P. Q. Nguyen, Hermite's constant and lattice algorithms, in: P. Q. Nguyen, B. Vallée (Eds.), The LLL Algorithm, Information Security and Cryptography, Springer Berlin Heidelberg, 2010, pp. 19–69.

[18] The PARI Group, Bordeaux, PARI/GP, version `2.3.4`, `http://pari.math.u-bordeaux.fr/` (2012).