ON THE SECURITY OF THE DIGITAL SIGNATURE ALGORITHM

IAN F. BLAKE AND THEODOULOS GAREFALAKIS

ABSTRACT. We present a key-recovery attack against the Digital Signature Algorithm (DSA). Our method is based on the work of Coppersmith [7], and is similar in nature to the attacks of Boneh et al ([5], [9]) which use lattice reduction techniques to determine upper bounds bounds on the size of an RSA decryption exponent under which it will be revealed by the attack. This work similarly determines provable upper bounds on the sizes of the two key parameters in the DSA for which the system can be broken. Specifically if about half of the total number of bits in the secret and ephemeral keys, assuming contiguous unknown bits in each key, are known, the system can be shown to be insecure. The same technique shows that if about half of the total number of bits in two ephemeral keys are known, again assumed contiguous unknown bits in each key, but with no knowledge of the secret key, the system can be shown to be insecure.

1. INTRODUCTION

The use of lattices and their reduction techniques is now a well established tool for attacking a variety of cryptosystems with many significant successes. The techniques generally rely on the so-called LLL ([14]) reduction method, outlined in the following section, which produces a short vector, that is a vector of relatively small Euclidean norm, in the lattice. The production of such a vector is known to be a hard problem although the LLL algorithm works well in practice for surprisingly large parameters. The ingenuity in applying this technique to a particular problem, such as showing the weakness of a cryptographic system, often lies in translating the equations governing the system, to a short-vector lattice problem.

The next section introduces the notion of lattices and their reduction and a discussion of the main properties of interest of the bounds on the sizes of the vectors in the reduced lattice. As an example of the use of this LLL algorithm, the determination of small solutions of univariate and bivariate modular polynomial equations, and the use of these results in showing certain weaknesses of the RSA system is discussed as the techniques are of interest in the attack on the DSA considered here in section 5. Section 4 considers the DSA algorithm as well as two previous attacks of Howgrave-Graham and Smart [13] and Nguyen and Shparlinski [18]. The attack on the DSA given here is somewhat different than that work and rather complements the work of Boneh et al ([3], [4], [5]) for RSA decryption exponents. The results show that if the sum of the unknown bits in the secret and ephemeral keys, assumed contiguous in each key, is less than half of the total number number of bits, then the system can be broken. The same result applies to the case

Date: July 27, 2001.

Key words and phrases. Cryptography, digital signatures.

where the secret key is completely unknown and the total number of unknown bits in two ephemeral keys, assumed contiguous in each key, is less than approximately half of the total number of bits. It is unlikely that such a large number of bits, especially in a secret/ephemeral key combination, would be leaked. Nonetheless, the results are of interest and show, along with the results of the cited work, that great care should be taken in ensuring that the choice of secret and ephemeral keys leaks no information. We note that our proof is based on the unproven assumption that the second shortest vector of the reduced lattice is sufficiently short. This assumption was checked in practice and found true in all cases tried.

A few observations on this technique are given in the final section, along with other comments. Our approach requires careful estimates of certain parameters and sufficient background material is given in the next two sections to yield these.

2. Lattices and LLL reduction

Let $\mathbb{R}^n$ denote the n-dimensional real Euclidean space with an inner product $< x, y >= \sum_{i=1}^n x_i y_i,\ x, y \in \mathbb{R}^n$ and norm $|x|^2 =< x, x >$. A lattice $\mathcal{L}$ will be a discrete subgroup of this space, consisting of the $\mathbb{Z}$-linear combinations of a set of basis vectors $B = \{w_1, \cdots, w_n\}$ i.e.

$$\mathcal{L}(B) = \{\sum_{i=1}^{n} \alpha_i w_i,\ \alpha_i \in \mathbb{Z}\}.$$

We assume all lattices considered span $\mathbb{R}^n$. In particular, any subgroup of $\mathbb{Z}^n$ is a lattice (an integer lattice) and this will be the case of interest in subsequent sections. There are an infinite number of bases for a lattice, related by unimodular matrices, and all have the same Gram determinant whose (i, j) element is $< w_i, w_j >$. The positive square root of the Gram determinant is sometimes referred to as $\text{vol}(\mathcal{L})$ or, equivalently, $\det(\mathcal{L})$. Of course if B is the matrix whose ith column is the basis vector w_i then B is nonsingular and the Gram matrix of the basis is $B^T B$.

There are several important problems on lattices that are purported to be difficult and perhaps the following two are of the most interest in applications. For a given lattice $\mathcal{L}$ and basis B, the *The shortest vector problem (SVP)* is to determine the lattice vector with minimum non-zero length. The *closest vector problem (CVP)* is, given an arbitrary vector $v \in \mathbb{R}^n$, determine the lattice vector closest to v. The article of Cai [6] discusses the complexity of these problems. Ajtai [1] has shown that the SVP is NP-hard under randomized reductions. It is also known that CVP is NP-hard, even to determining it to within any constant factor ([2], [10], [16]). The article of Cai [6] discusses the complexity of these problems in greater detail.

The aim of basis reduction algorithms is to derive a new set of basis vectors that achieve minimization according to some criteria. In the case of the LLL algorithm [14] a 'short' vector appears as the first output basis vector and the columns are made as mutually orthogonal as possible. However, as noted, there is no efficient algorithm to find the shortest non-zero vector of a lattice.

If $\mathcal{L}$ is a lattice of full rank with a basis matrix $B_{\mathcal{L}}$ then if B is the basis matrix after reduction by the LLL algorithm, with columns (basis vectors) b_i, and if λ_1 is the length of the shortest non-zero vector of $\mathcal{L}$ then the LLL algorithm acting on $B_{\mathcal{L}}$ produces a basis $\{b_1, \cdots, b_n\}$ with the following properties:

i) $b_1, \cdots, b_n$ is a basis of $\mathcal{L}$.
ii) $|b_1| \leq 2^{(n-1)/2}\lambda_1$
iii) $|b_1| \leq 2^{(n-1)/2}(\det(\mathcal{L}))^{1/n}$
iv) $|b_2| \leq 2^{n/2}(\det(\mathcal{L}))^{1/(n-1)}$
v) $\det(\mathcal{L}) \leq \prod_i |b_i| \leq 2^{n(n-1)/2}\det(\mathcal{L})$

Thus the algorithm will produce a short vector in the lattice. The algorithm often performs better in practice than the above constants might indicate. It has been extensively investigated and many variants of it now exist.

The LLL algorithm provides a partial solution to SVP. It runs in polynomial time and approximates the shortest vector in the lattice to within a factor of $2^{n/2}$. Schnorr [19] improved this constant to $(1+\epsilon)^n$.

3. LLL and zeros of modular polynomial equations

It is noted in ([7]) that while solving a polynomial over the integers is an easy problem, finding modular roots of either univariate or bivariate polynomials tends to be more difficult. The results of Coppersmith [7] and Howgrave-Graham [12] are used below for the applications of interest here. Although it will turn out that the bivariate polynomial derived for our system is linear in each variable, it is nonetheless convenient to use the results available for more general bivariate polynomials. Their work strengthened the earlier work of Håstad [11] and Vallée et al [20]. The discussion in [17] summarizes the situation nicely. Observe that finding a root of the equation

$$x^e - c \equiv 0 \pmod{n}$$

for some positive encryption exponent e is thought to be as difficult as factoring the modulus n, and indeed the ability to solve the equation for $e = 2$ is provably equivalent to factoring n where n is known to be the product of two primes, in the sense noted in [15]. However the theorem of Coppersmith below shows that small roots of such equations can be determined more efficiently by using LLL reduction:

Theorem 1 (Coppersmith). *Let P be a monic integer polynomial in one variable of degree d modulo an integer n of unknown factorization. Then one can find all integer roots of the equation $P(x_0) \equiv 0 \pmod{n}$, $|x_0| \leq n^{1/d}$ in time polynomial in $(\log(n), 2^d)$.*

The technique is to observe that a suitably small solution to the modular polynomial equation is also a solution to the integer equation. However, the solution to the integer equation is, by construction of the lattice problem, equivalent to a short vector in the integer lattice. The technique for achieving this is encapsulated in the following lemma, where for $a(x) = \sum_i a_i x^i \in \mathbb{Z}[x]$, $||a(x)|| = (\sum_i a_i^2)^{1/2}$:

Lemma 1. *Let $a(x) \in \mathbb{Z}[x]$ be a polynomial of degree d and let X be a positive integer. Let $||a(xX)|| < n^h/\sqrt{d}$. Then if $a(x_0) \equiv 0 \pmod{n^h}$ with $|x_0| < X$, then $a(x_0) = 0$ holds also over the integers.*

The lemma is shown by a certain polynomial linearization technique that produces the appropriate lattice in which a small vector corresponds to a solution of the integer polynomial equation. The application of this lemma to RSA decoding is immediate [7].

The bivariate case has a similar approach. For the bivariate polynomial $a(x,y) = \sum_{i,j} a_{i,j} x^i y^j \in \mathbb{Z}[x,y]$, we have the result:

Lemma 2 (Coppersmith, Howgrave-Graham). *Suppose $a(x,y) \in \mathbb{Z}[x,y]$ is a polynomial which is a sum of at most r monomials. Suppose that $a(x_0,y_0) \equiv 0 \pmod{n^h}$ for some integer n of unknown factorization, where $|x_0| < X$, $|y_0| < Y$ and $\|a(xX,yY)\| < n^h/\sqrt{r}$. Then $a(x_0,y_0) = 0$ holds over the integers.*

Boneh and Durfee [4] apply this result to the case of RSA decryption, briefly described here. The attack on the digital signature algorithm of the next section is similar to the approach of this work. Let the RSA parameters be $n = pq$, for two odd primes p, q and let the encryption and decryption exponents respectively be e, d where $ed \equiv 1 \pmod{\phi(n)/2}$ where it is assumed that $\gcd(p-1,q-1) = 2$. Since $\phi(n) = n - (p+q) + 1$ there exists an integer k such that

$$ed + k(\frac{n+1}{2} - \frac{p+q}{2}) = 1.$$

Since e, n are public and p, q, d are private, if we let $x = k$, $y = -(p+q)/2$ and $A = (n+1)/2$ the equation reduces to

$$x(A+y) \equiv 1 \pmod{e}.$$

The exponent e is typically on the order of n^α for α close to 1 while $|y|$ is usually on the order of $n^{0.5} \sim e^{0.5}$. If for a given value of $\delta < 0.5$ all the small solutions to the above equation, with $|y| < e^{0.5}$, $|x| < e^\delta$ can be found, then for decryption exponents $d < n^\delta$ the RSA system is shown to be insecure. Currently for $\delta < 1 - 1/\sqrt{2} \approx .292$ such a solution can efficiently be found using the lattice reduction techniques noted.

4. The Digital Signature Algorithm

In this section, we briefly describe DSA. A detailed presentation of the algorithm can be found in [15].

DSA bases its security on the presumed intractability of the discrete logarithm problem in the multiplicative group of finite fields, and in prime order subgroups. In the initialization phase, the following quantities are chosen:

- a prime p of size between 512 and 1024 bits in increments of 64;
- a prime q of size 160 bits, such that $q|p-1$;
- a hash function h mapping messages to the subgroup of order q
- a secret integer a in the subgroup of order q.

The parameters specify the finite field $\mathbb{F}_p$, and its unique subgroup G of order q. Assume a generator of this group is g, $G =< g >$. Since 1996 a prime p of at least 768 bits has been recommended [15].

To sign a message m, Alice performs the the following steps:

1. Choose $k \in \{1,...,q\}$ uniformly at random.
2. Compute $r = (g^k \mod p) \mod q$.
3. Compute $s = k^{-1}(h(m) + a \cdot r) \pmod{q}$.
4. Send (r,s) as the digital signature of the message m.

In this procedure the key a is referred to as the secret key, intended to be chosen only once, and k is the ephemeral key, often referred to as a nonce, chosen differently for each message.

The assumption here is that the only way to break this signing algorithm – and be able to forge signatures – is to recover either the secret key a, or the ephemeral key k. Notice that if one could find the discrete logarithm of r to retrieve k then it is a simple matter to find a from s and break the system. Indeed, if this is possible, then anyone could sign documents impersonating Alice. However, the parameters of the system were chosen in such a way that computing discrete logarithms in $\mathbb{F}_p^*$ and in G is computationally infeasible and the system is considered secure.

The authors are aware of two previous attacks on the DSA using lattice techniques ([13], [18]). Howgrave-Graham and Smart [13] consider the case where for some number of different signatures, a small fraction of the bits of the (distinct) parameters k are revealed. Under certain conditions they give an attack (using a CVP algorithm due to Babai, based on the LLL algorithm) that reveals the secret key a. The work considers the numbers of bits of k revealed in each signature and the number of signatures and uses certain heuristic assumptions to analyze the performance of this algorithm.

Nguyen and Shparlinski [18] consider a similar scenario where it assumed that if for a polynomially bounded number of messages, about $\log_2^{1/2}(q)$ of the least significant bits of the ephemeral keys are known, then under certain conditions one can in polynomial time recover the signer's secret key a. The results are provable.

Our approach here is in a sense the inverse. We determine how large the keys a and k can be in order for them to be revealed by this attack by considering a single signature. In this respect, as has been noted, it is more similar in spirit to the results of [4] for the RSA decryption exponents. In essence we will show that if ℓ_1 of the most significant bits of the secret key are known and ℓ_2 of the most significant bits of the ephemeral key are known, not necessarily zero, $\ell_1+\ell_2 < \log_2 q$, the system is insecure. Additionally it will be argued, by using exactly the same procedure, that if no bits of the secret key are known but approximately half of the most significant bits of two ephemeral keys are known, the system is insecure. It is also noted that it is sufficient for the unknown bits to be contiguous for the above statements to hold concerning the numbers of known bits required for the attack to succeed.

The results also imply, for example, that even if a much larger subgroup than one of size 160 bits (as for DSA) is chosen, if $|a| < q^{1/2}$ and $|k| < q^{1/2}$, then while computing a as the discrete logarithm of the public key may still be infeasible (say, for $\log_2(q) \approx 600$), the method shows that a and k can be computed easily using LLL, and the system would be insecure.

5. The attack

In this section, we describe our attack against DSA. Our approach is not to recover the secret key by solving the related discrete logarithm problem directly. Instead, we take advantage of the form of the equation in step (3) of the signing procedure.

We note that step (3) alone does not reveal any information about a or k. However, as we will show, the equation in step (3) together with the assumption that a and k are of *relatively* small size, is enough to break the system.

By rearranging terms in Equation (3), we have

$$(1) \qquad k + \left(-\frac{r}{s}\right) \cdot a + \left(-\frac{h(m)}{s}\right) = 0 \pmod{q}.$$

That is, the pair (a, k) satisfies a modular equation of the form

$$(2) \qquad f(x, y) \equiv 0 \pmod{q},$$

where, in our case,

$$f(x,y) = y + Ax + B,$$

with

$$A = -\frac{r}{s} \quad \text{and} \quad B = -\frac{h(m)}{s}.$$

It is assumed the solution we are looking for is small, i.e., $|a| < X$, and $|k| < Y$, for some bounds X, Y that we specify later.

Given a modular polynomial equation such as Equation (2), which is known (or assumed) to have a small solution, Lemma 2 gives us an idea as to when this small modular solution is a solution to the integer equation. Since our Equation (2) has three monomials of degree at most one, define the polynomials

$$g_{0,0}(x,y) = q \ , \ g_{0,1}(x,y) = f(x,y) = y + Ax + B \text{ and } g_{1,0}(x,y) = qx$$

leading to the basis matrix of

$$\begin{bmatrix} q & 0 & 0 \\ 0 & qX & 0 \\ B & AX & Y \end{bmatrix}.$$

The matrix has determinant q^2XY. Combining the estimate of the length of a short reduced vector from the properties of an LLL reduced basis and the estimate in the Lemma 2, guarantees a solution of the integer equation if

$$2^{w/2}\det(L)^{1/w} \leq \frac{q^t}{\sqrt{w}}$$

or, as w, the dimension of the matrix, is 3 and t the power of the modulus, is 1, then

$$2^{3/2}(q^2XY)^{1/3} \leq \frac{q}{\sqrt{3}}.$$

If we let now $X = q^\alpha$ and $Y = q^\beta$, substituting these values and manipulating the equations gives

$$\alpha + \beta < 1 - \frac{4.5}{\log_2(q)}(1 + \frac{\log_2(3)}{3}) = 1 - \frac{6.877}{\log_2(q)}. \tag{3}$$

If condition (3) is satisfied, then the shortest vector of the reduced basis is guarantied to yield a polynomial $H_1(x,y)$ with the desired root over the integers. However, in order to actually obtain this solution, we need one more 'small' equation. For this, we use the second shortest vector. If the bound

$$|b_2| < \frac{q}{\sqrt{3}}$$

holds for the size of the second shortest vector, we obtain a second polynomial $H_2(x,y)$. It is important to note that $H_1(x,y)$ and $H_2(x,y)$ are linear in x and y, and are *linearly independent.* Thus, solving the linear system we would provably obtain the values.

We proceed now to show that indeed the second shortest vector is short enough. It is well-known (see [4]), that the size of the second shortest vector in the reduced basis is bounded by

$$|b_2|^2 \leq |b_2^*|^2 + \frac{1}{4}|b_1|^2. \tag{4}$$

We need to give an upper bound for $|b_2^*|$. As shown in [4],

$$|b_2^*|^2 \leq 4\frac{\det(L)}{|b_1|},$$

which is equivalent to

$$|b_2^*|^2 \leq 4\frac{q^{2+\alpha+\beta}}{|b_1|}.$$

In order for the second basis vector to also meet the bound of Lemma 2 we need $|b_2| < q/\sqrt{3}$ and in order for this to be true, from the above estimate, we need

$$|b_1| \geq 16q^{\alpha+\beta},$$

which is sufficient for applying Lemma 2, allowing the same small solution to the first equation to satisfy this relation also. With two linearly independent solutions, the solution can be obtained.

The previous discussion is captured by the following proposition for the specific case of DSA.

Proposition 1. *Let α, β be real numbers in the range $[0,1]$. Let L be the 3-dimensional lattice defined above. If the shortest vector of the LLL-reduced basis has length at least $16q^{\alpha+\beta}$, then the attack described here will break the DSA, provided the secret key a satisfies $|a| < q^{\alpha}$, and the secret exponent k satisfies $|k| < q^{\beta}$, with $\alpha+\beta \leq 1-6.877/(\log_2(q))$. For $\log_2(q) \approx 160$ the bound for $\alpha+\beta$ becomes ≈ 0.957.*

It is interesting to note that the larger q gets, the closer to 1 the sum $\alpha + \beta$ is allowed to be. It is not clear how much this bound can be improved with the approach used here. The results indicate that some care must be taken in choosing the secret and ephemeral keys in the DSA, a and k. As far as the authors are aware, this is the first attack of this nature on DSA.

6. Extensions

The above attack tacitly assumed the unknown bits were the least significant for both keys. It is immediately clear that if the unknown bits are contiguous anywhere in the keys, then the key x can be written $x' + \beta$ and the key y can be written as $y' + \delta$ where x' and y' represent the unknown bits and the constants the known bits. Multiplying these quantities appropriately and further adjusting the constants of the equation, yields quantities x'' and y'' representing the unknown quantities where the unknown bits are in the least significant positions and all other bits zero. The original equation

$$y + Ax + B \equiv 0 \pmod{q}$$

is then easily transformed through multiplies and additions to

$$y'' + A'x'' + B' \equiv 0 \pmod{q}$$

and the same technique of finding small solutions applies.

Similarly if two signatures are available, the two equations can be used to eliminate the secret key and the resulting equation relating the two ephemeral keys is of the same form. Thus earlier statements about secret and ephemeral keys also apply to two ephemeral keys.

7. Comments

The attack described here was programmed in NTL on a 500MHz PC and the secret key a was obtained in a few seconds for parameters of the size of the DSA parameters. Even for much larger size parameters, for instance, $\log_2(q) = 500$, it took no longer than 5 seconds to recover the secret key.

It should be reiterated that the attacks of Howgrave-Graham and Smart [13] and Nguyen and Shparlinski [18] are much more likely to be effective in practice, since many fewer bits of the ephemeral keys need to known. The tradeoff is that we only need 2 signatures, instead of polynomially many, and that the number of bits that are leaked in each key need not be individually bounded. Instead, the total number needs to be about half the bits.

It is not clear how the attack described can be improved if additional signatures with known bits were available. It would be of interest if this method could be extended to a higher dimensional lattice problem as the number of signatures increased, but at this point it is not clear how this might be achieved.

Acknowledgment: The authors would like to thank Professor Igor Shparlinski for his comments on the original version of this paper that significantly expanded the applicability of the work. Specifically he noted that the technique in the original version of the work applies to the case when the most significant bits of the keys are known but not necessarily zero. He also noted that by eliminating the secret key, given two signatures, the same results apply for two ephemeral keys. The authors would also like to thank the two reviewers for their perceptive and useful comments that significantly improved the work.

References

[1] M. Ajtai, The shortest vector problem is NP-hard for randomized reductions, Proc. 30th ACM Symp. on Theory of Computing, pp.10-19, 1998.

[2] S. Arora, L. Babai, J. Stern and Z. Sweedyk, The hardness of approximate optima in lattices, codes and systems of linear equations, 34th FOCS, pp. 724-733, 1993.

[3] D. Boneh, An attack on RSA given a small fraction of the private key bits, *Asiacrypt '98, Lecture Notes in Computer Science*, vol. 1514, pp. 25-34, 1998.

[4] D. Boneh and G. Durfee, Cryptanalysis of RSA with private key less than $N^{0.292}$, *Eurocrypt '99, Lecture Notes in Computer Science,* vol. 1592, pp. 1-11, 1999.

[5] D. Boneh and G. Durfee. Cryptanalysis of RSA with private key d less than $N^{0.292}$. *IEEE Transactions on Information Theory*, 46, 2000.

[6] Jin-Yi Cai, The complexity of some lattice reduction problems, *ANTS IV, Lecture Notes in Computer Science*, Vol. 1838, Wieb Bosma ed., pp. 1-32, 2000.

[7] D. Coppersmith, Small solutions to polynomial equations, and low exponent RSA vulnerabilities, *J. Cryptology*, vol. 10, pp. 233-260, 1997.

[8] D. Coppersmith, Finding a small root of a univariate modular equation, *Proceedings of Eurocrypt 1996, Lecture Notes in Computer Science* vol. 1070, pp. 155-166, 1996.

[9] G. Durfee and P.Q. Nguyen. Cryptanalysis of the RSA schemes with short secret exponent, Asiacrypt 2000, *Lecture Notes in Computer Science*, Vol.1976, pp.14-29, 2000. In *LNCS.*

[10] P. van Emde Boas, Another NP-complete partition problem and the complexity of computing short vectors in a lattice, Tech. Report 81-04, Dept. Mathematics, University of Amsterdam, 1980.

[11] J. Håstad, Solving simultaneous equations of low degree, *SIAM J. Computing*, vol. 17, pp. 336-341, 1988.

[12] N.A. Howgrave-Graham, Finding small roots of univariate modular equations revisited, in *Cryptography and Coding, Lecture Notes in Computer Science*, Vol. 1355, pp. 131-142, 1997.

[13] N.A. Howgrave-Graham and N.P. Smart, Lattice attacks on digital signature schemes, HPL Technical Report, 1999-90, available at `http://www.hpl.hp.com/techreports/1999`, to appear *Designs, Codes and Cryptography.*

[14] A.K. Lenstra, H.W. Lenstra and L. Lovász, Factoring polynomials with rational coefficients, *Mathematische Annalen*, vol. 261, pp. 515-534, 1982.

[15] A.J. Menezes, P.C. Van Oorschot, and S.A. Vanstone. *Handbook of Applied Cryptography.* The CRC Press series on discrete mathematics and its applications. CRC Press, 2000 N.W. Corporate Blvd., Boca Raton, FL 33431-9868, USA, 1997.

[16] D. Micciancio, The shortest vector in a lattice is hard to approximate to within some constant, Elect. Coll. on Comp. Complexity, Report no. 16, 1998.

[17] P. Nguyen and J. Stern, Lattice reduction in cryptology: an update, *ANTS IV, Lecture Notes in Computer Science*, Vol. 1838, Wieb Bosma ed., pp. 85-112, 2000.

[18] P. Nguyen and I. Shparlinski, The insecurity of the digital signature algorithm with partially known nonces, preprint `http://www.di.ens.fr/ nguyen/pub.html`.

[19] C.P. Schnorr, A hierarchy of polynomial time lattice basis reduction algorithms, *Theoretical Computer Science*, vol. 53, pp. 201-224, 1987.

[20] B. Vallée, M. Girault and P. Toffin, How to guess l-th roots modulo n by reducing lattice bases, *Proc. AAECC-6*, *Lecture Notes in Computer Science*, vol. 357, pp. 427-442, 1988.

Department of Electrical and Computer Engineering, University of Toronto, Toronto, M5S 3G4, Canada

E-mail address: `ifblake@comm.toronto.edu`

Department of Mathematics, Royal Holloway College, Information Security Group, Egham, Surrey TW20 0EX, UK

E-mail address: `theo.garefalakis@rhul.ac.uk`