Side-Channel Entropy for Modular Exponentiation Algorithms

Lejla Batina[1,2] and Cees Jansen[2]

[1]K.U. Leuven ESAT/COSIC,
Kasteelpark Arenberg 10
B-3001 Leuven-Heverlee, Belgium

[2]SafeNet BV
Boxtelseweg 26a, 5261 NE
Vught, The Netherlands

{lbatina, cjansen }@safenet-inc.com

Abstract. In our previous work we used the combinatorial approach to develop a simple model for Hamming weight leakage. In this paper we go one step further by considering exponents to be random but co-prime to some even composite number. This is based on the assumption that a private exponent is a unit in $Z^*_{\varphi(N)}$, chosen uniformly at random. The probability of such an exponent depends on the choice of the modulus N, i.e. what type of primes are p and q. We consider safe primes, strong primes and the situation where $p-1$ and $q-1$ are products of small prime factors. The result is given by an upper bound on the information leakage in this refined model. It is shown that the worst case information leakage amounts to only 3.6 bits for 1024 bit moduli.

1 Introduction

Today many analysis methods exist which are used to attack implementations in hardware and software of cryptographic algorithms. Rather than attacking the algorithms themselves, specific properties of the implementation such as timing behaviour [3, 8] and power consumption [9], during execution of the algorithm are exploited to gain information about the secret key. These, so called side channel attacks can be very effective in breaking the cryptosystem, necessitating careful implementation and consideration of countermeasures.

In a first order approximation in a previous paper [1] it was assumed that the secret exponents are chosen at random from the set of odd numbers. For this random exponent model (REM) the exact probability distribution was derived, and the entropy $H(K)$ of the number of operations K determined. In this paper the validity of this model is shown to hold when the exponent is relatively prime to the Euler (Totient) ϕ function [7] of the modulus. We focus on RSA keys, i.e. the triplets (d, p, q), where pq is the secret factorization of the modulus N, (so $N = pq$) and d is the private RSA exponent. Our conclusions will be shown to be

consistent with those of [14]. Finally, when considering the use of strong primes we conclude that the entropy does not significantly differ from the one derived in the REM as introduced in [1]. Moreover, we prove that all options for primes i.e. from safe to random primes, are compliant with the developed model. The results are showing that this information is far from exploitable for the practical bit-lengths in nowadays applications.

As related previous work we mention that of Waldvogel and Massey [14] in which they considered the entropy of Diffie-Hellman keys.

The remainder of this paper is organized as follows. In Section 2 we review shortly our previous work. In Section 3 the relevance of the random exponent model is discussed for real-life RSA applications. An lower bound on the exponent entropy is derived, showing that the REM provides a good approximation for the exponent leakage of the more realistic model. Section 4 concludes the paper.

2 First Approximation: The Random Exponent Model and its generalization

In our previous work we focused on modular exponentiation which is used in the RSA cryptosystem [12]. Modular exponentiation i.e. the calculation of M^e mod N is usually performed by use of the standard square-and-multiply algorithm [6]:

Algorithm 1 Square-and-Multiply Algorithm (from MSB)

Require: M and $e = \sum_{i=0}^{n-1} e_i 2^i = (e_0, e_1, ..., e_{n-1})$
Ensure: $Res = M^e$
  $Res \leftarrow M$
  for *n-2* to 0 **do**
    $Res \leftarrow Res^2$ mod N
    if $e_i = 1$ then $Res \leftarrow Res \cdot M$ mod N
  end for
  Return Res

The question is how much information can be obtained when observing the total number of operations, i.e. modular multiplications and squarings, in a certain part of the secret exponent. Here, a computationally unrestrained enemy is considered as we discuss the side-channel security in the sense of Shannon Information Theory.

Assumptions for the REM:

1. The secret exponent d is an arbitrary odd number modulo N.
2. Timings for the operations of multiplying and squaring of two numbers are the same. This assumption is realistic considering the required constant-time implementations as the necessary condition for side-channel security. "Good" hardware implementations nowadays meet this condition.

3. Each user chooses her/his private key independently and uniformly in the set of all odd numbers $\{3, 5, ..., 2^n - 1\}$. In particular, for all $d \in \{3, 5, ..., 2^n - 1\}$, $P(X_A = d) = P(X_B = d) = \frac{1}{2^{n-1}-1}$ where $P(X_A = d)$ gives the probability that Alice chooses the number d as her private exponent.

The number of exponents of length up to n bits with exactly k operations involved in exponentiation was denoted with $G(n, k)$. The probability distribution of K is given by: $P\{K = k\} = (2^{n-1} - 1)^{-1} G(n, k)$, where K denotes the random variable associated with the number of operations in an exponentiation with the randomly selected exponent. The probability that an attacker is able to successfully guess the secret exponent d is of interest. He possesses the exact information on number of operations k performed while exponentiation has been done and he has a detailed knowledge of the algorithm of exponentiation and the relevant bit-lengths.

We derived the expression for the mutual information $I(K; E)$ between the random variables K and E, where E denotes the variable associated to a random exponent. This notion of given-away information of the key was already introduced in [4].

With the aforementioned three assumptions we obtained the following lower bound on the conditional exponent uncertainty:

$$H(E|K) = H(E) - H(K) \geq n - 1 - \log_2(2n - 3) \text{ bits.} \tag{1}$$

From (1) the logarithmic relation between information leakage and maximum exponent length can be observed.

Figure 1 shows the exponent entropy loss upper bounds.

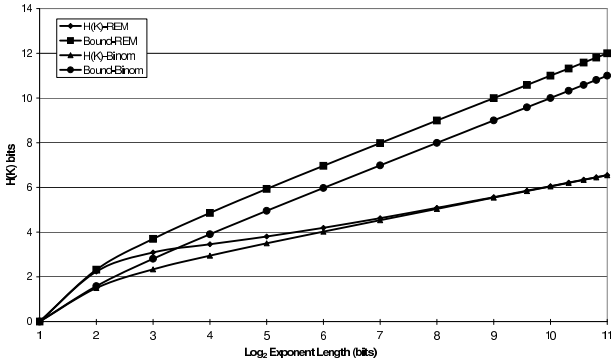

Fig. 1. Exponent entropy loss for two different models (Random Exponent Model (REM) and Binomial Model (BM).

The previous results are generalized by considering the possibility that the attacker is observing the random exponent in two or more parts. The following

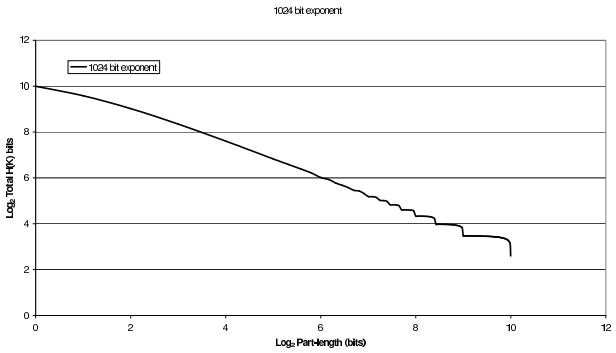

Fig. 2. Random Exponent Model - Observation in Parts

result was obtained in [1]:

$$H(K^t) = h_B(\ell - 1) + \Big(\Big\lfloor \frac{n}{\ell} \Big\rfloor - 1\Big) h_B(\ell) + h_B\Big(n - \Big\lfloor \frac{n}{\ell} \Big\rfloor \ell - 1\Big) \quad \text{bits.} \tag{2}$$

Here $H(K^t)$ denotes the total amount of information obtained by observing the operations from all parts. Figure 2 shows $H(K^t)$ as a function of the part-length for a 1024 exponent on a log-log scale.

3 How relevant is the model

In this section we will look into the more realistic model in which exponents are taken to be random, but co-prime to a random even composite number.

When using the RSA public key cryptosystem, one encounters the Euler Totient function $\phi(N)$, where N is the modulus, usually a composite integer, that is the product of two primes. The public encryption exponent e and the corresponding secret decryption exponent d are related by the fact that they are each others multiplicative inverse in the residue ring where the operation is modular multiplication (mod $\phi(N)$), i.e.: $e \cdot d \equiv 1 \bmod \phi(N)$. Clearly, e and d must be units in the ring, i.e. $\mathrm{GCD}(e, \phi(N)) = 1$. The number N_e of exponents satisfying this requirement is given by $N_e = \phi(\phi(N))$. This number of exponents is of interest in problems where the total diversity of public-secret key-pairs is relevant, e.g. in considering the exponent information leakage due to side channel information. In the sequel we will look at the minimum and maximum values that this number N_e can take on for various moduli N.

Given an integer N in its general form:

$$N = \prod_i {p_i}^{e_i}, p_i \neq p_j$$

and all $e_i > 0$, where p_i are prime factors of N. Then, by definition

$$\phi(N) = \prod_i (p_i - 1) {p_i}^{e_i - 1} = N \prod_i (1 - \frac{1}{p_i}) \tag{3}$$

From the formulas above it is evident that the highest value that ϕ can attain is $N-1$, if and only if N is a prime. Moreover can it be seen from (3) that the multiplicities e_i do not matter, and hence the least value of ϕ occurs when N is composite and equal to the product of many different small primes.

In the standard RSA setting the modulus N is the product of two distinct primes p and q, and therefore $\phi(N) = (p-1)(q-1) = N-(p+q)+1$, which always contains a factor of 4. If the objective is to have as many exponents as possible, i.e. to get the highest value of $\phi(\phi(N))$, then clearly the best option is that $(p-1) = 2r$ and $(q-1) = 2s$, with r and s again primes.

3.1 Sophie Germain primes

For $(p-1) = 2r$ and $(q-1) = 2s$, with r and s again primes one gets $\phi(\phi(N)) = 2(r-1)(s-1)$. Prime numbers r with the property that $2r+1$ is again prime, are known as safe primes [11] or Sophie Germain primes.

Suppose one uses safe primes, then we have $\phi(N) = 4rs$ and $\phi(\phi(N)) = 2(r-1)(s-1)$, so we have

$$\frac{\phi(\phi(N))}{\phi(N)} = \frac{1}{2}\left(\frac{1}{s} - \frac{1}{r} + \frac{1}{rs}\right), \tag{4}$$

showing that the fraction of real RSA exponents is about half that of the $\phi(N)$.

Also, $\phi(\phi(N)) = 2(\frac{p-3}{2})(\frac{q-3}{2}) = \frac{1}{2}(N-3(p+q)+9)$. As the minimum value of $p+q$ is obtained for $p = q = \sqrt{N}$ we have the upper bound $\phi(\phi(N)) = \frac{1}{2}(N-6\sqrt{N}+9)$. Now if we assume that indeed $p \simeq q \simeq \sqrt{N}$, we loose no more than approximately 1 bit in the entropy of the secret exponent and this is comparable with our previous results when taking all odd exponents. Under this assumption we arrive at

$$\frac{\phi(\phi(N))}{\phi(N)} = \frac{1}{2}\left(\frac{\sqrt{N}-3}{\sqrt{N}-1}\right)^2.$$

3.2 Strong primes

Let us consider strong primes, i.e. $p = 2k_p r + 1$ and $q = 2k_q s + 1$, where r and s are large prime factors of $p-1$ and $q-1$ resp. and k_p and k_q are arbitrary small integers [11]. We now have $\phi(N) = 4k_p k_q rs$ and consequently $\phi(\phi(N)) = (r-1)(s-1)\phi(4k_p k_q)$, so one can write

$$\frac{\phi(\phi(N))}{\phi(N)} = \frac{\phi(4k_p k_q)}{4k_p k_q}\left(\frac{1}{s} - \frac{1}{r} + \frac{1}{rs}\right) \tag{5}$$

It now depends on the factorization of k_p and k_q with how many bits the entropy of the exponent is decreased. The smallest decrease occurs if k_p and k_q are powers of 2, the reduction being $\frac{1}{2}$, or 1 bit in entropy. The biggest decrease occurs if k_p and k_q are composite, containing many small distinct prime factors.

For example, suppose again that $p \simeq q \simeq \sqrt{N}$, and that both k_p and k_q are at most 16 bit composite integers with values $k_p = 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13$ and $k_q = 17 \cdot 19 \cdot 23$. In this case $\phi(4k_pk_q) = 2 \cdot 2 \cdot 4 \cdot 6 \cdot 10 \cdot 12 \cdot 16 \cdot 18 \cdot 22$ and the reduction factor $\frac{\phi(4k_pk_q)}{4k_pk_q}$ becomes 0.1636, resulting in a decrease of the exponent entropy of at most 2.6 bits. The way strong primes are generated (see [2]), the values of k_p and k_q are usually less than 16 bits for 512 bit primes. Consequently, the entropy loss is also less than 2.6 bits.

Strong primes were introduced in the context of so-called "non-weak RSA keys. A simple method for finding strong, random large primes was given by John Gordon already in 1984 [2]. The reasoning behind it was that with strong primes one counters factoring, as well as cycling attacks. There exist arguments for both, using strong or just random primes. Rivest and Silverman [13] argue that it is unnecessary to use strong primes in the RSA cryptosystem. Namely, for the most recent factoring method based on elliptic curves (ECM) due to H. Lenstra [10], strong primes offer no extra protection. On the other hand, Joye et al. [5] state that, since strong primes can only yield a safer solution and it is not much more difficult to generate them, it is better to use strong instead random primes. Some standards still recommend use of strong primes for RSA, for example ISO/IEC9594-8.

3.3 Random primes

Similar reasoning as in the previous subsection shows that a lower bound to $\phi(\phi(N))$ is obtained by assuming $\phi(N)$ to be equal to $2^2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \ldots a_{max}$. In this case the fraction of exponents $\frac{\phi(\phi(N))}{\phi(N)}$ is given by equation (3) when taking a_{max} for the largest p_i. This bound is realistic in that it sometimes refers to primes of the form $(a_j \cdot \ldots \cdot a_k + 1)$ that exist but sometimes they are the product of two or more smaller primes. Figure 3 shows this bound for the decrease of the exponent entropy as a function of bit lengths. As can be seen the extra condition imposed on exponents results in a minor decrease of the entropy, viz. at most 3.6 bits for 1024 bit RSA.

Let us consider an example. Take $\phi(N) = 2^2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 = 60060$, so $\phi(\phi(N)) = 2^2 \cdot 4 \cdot 6 \cdot 10 \cdot 12 = 2^8 \cdot 3^2 \cdot 5 = 11520$. One candidate for N is $(2 \cdot 5 \cdot 13 + 1)(2 \cdot 3 \cdot 7 \cdot 11 + 1) = 131 \cdot 463 = 60653$. The same holds for $N = 23 \cdot 2731 = 62813$. So a 16 bit modulus results in an exponent entropy of somewhat less than 14 bits.

The conclusion from the above is that the diversity (or entropy) of the secret exponents is a little bit less than assumed in our previous model. One should consider odd numbers d, $3 \leq d \leq \phi(N) - 1$, that are relatively prime to $\phi(N)$ and there exactly $\phi(\phi(N))$ of them. Hence, the lowest entropy is achieved for $2^2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \ldots a_{max}$. The values of a_{max} can be determined for 512, 1024, 2048 bit moduli and consequently $2^2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \ldots a_{max}$ computed. The first 132 prime numbers were used to calculate the bound depicted in Figure 3.

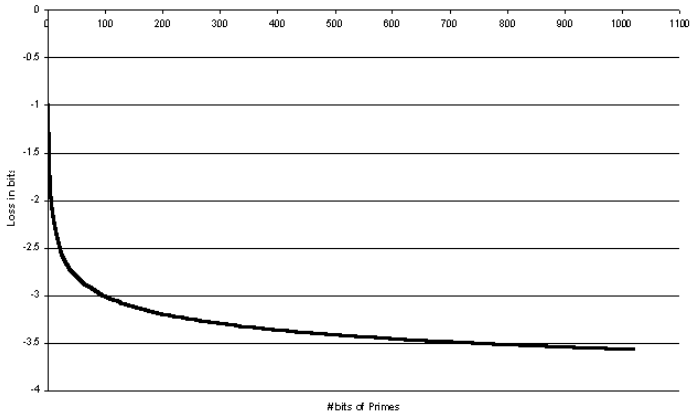

Fig. 3. Entropy loss due to decomposition of $\phi(N)$ into smallest primes.

4 Conclusions

We have extended the validity of the Random Exponent Model to the more realistic situation of RSA exponents which are co-prime to the $\phi(N)$. If one considers the total number $\phi(\phi(N))$ of such exponents, it was shown that for safe primes the available fraction of exponents is about one half of the total. For strong primes this fraction can be somewhat less, depending on the small factors in the $\phi(N)$. In this case the equivalent loss in entropy, assuming equally probable and independent exponents, is less than 3 bits for 1024 bit exponents.

For the case of random primes an obvious lower bound to the fraction $\frac{\phi(\phi(N))}{\phi(N)}$ results in a useful upper bound for the entropy loss. This bound indicates a loss of at most 3.6 bits for 1024 bit exponents.

The results of this paper indicate that the REM is a good way to model the information leakage due to side channel analysis.

References

1. L. Batina and C. Jansen. Secret exponent information leakage for timing analyses. In B. Macq and J.-J. Quisquater, editors, *Proceedings of the 23rd Symposium on Information Theory in the Benelux, Louvain-la-Neuve, Belgium*, pages 225–232. Werkgemeeschap voor Informatie-en-Communicatietheorie, 2002.
2. J. Gordon. Strong primes are easy to find. In T. Beth, N. Cot, and I. Ingemarsson, editors, *Advances in Cryptology: Proceedings of EUROCRYPT'84*, number 209 in Lecture Notes in Computer Science, pages 216–223. Springer-Verlag, 1984.
3. G. Hachez, F. Koeune, and J.-J. Quisquater. Timing attack: what can be achieved by a powerful adversary? In A. Barbé, E. C. van der Meulen, and P. Vanroose, editors, *Proceedings of the 20th symposium on Information Theory in the Benelux*, pages 63–70, May 1999.

4. C. J. A. Jansen. Key signature schemes. In *Proceedings of the Seventh Symposium on Information Theory in the Benelux*, pages 197–205, Noordwijkerhout, The Netherlands, 1986.
5. M. Joye, J. J. Quisquater, and T. Takagi. How to choose secret parameters for RSA and its extensions to elliptic curves. *Designs, Codes and Cryptography*, 3(23):297–316, 2001.
6. D. E. Knuth. *The Art of Computer Programming*, volume 2/Seminumerical Algorithms. Addison-Wesley, 1997.
7. N. Koblitz. *A Course in Number Theory and Cryptography*, volume 114 of *Graduate text in mathematics*. Springer-Verlag, Berlin, Germany, second edition, 1994.
8. P. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS and other systems. In N. Koblitz, editor, *Advances in Cryptology: Proceedings of CRYPTO'96*, number 1109 in Lecture Notes in Computer Science, pages 104–113. Springer-Verlag, 1996.
9. P. Kocher, J. Jaffe, and B. Jun. Differential power analysis. In M. Wiener, editor, *Advances in Cryptology: Proceedings of CRYPTO'99*, number 1666 in Lecture Notes in Computer Science, pages 388–397. Springer-Verlag, 1999.
10. H. W. Lenstra Jr. Factoring integers with elliptic curves. *Ann. of Mathematics*, 126:649–673, 1987.
11. A. Menezes, P. van Oorschot, and S. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1997.
12. R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.
13. R. L. Rivest and R. D. Silverman. Are 'strong' primes needed for RSA. 1999.
14. C. P. Waldvogel and J. L. Massey. The probability distribution of the Diffie-Hellman key. In *Advances in Cryptology: Proceedings of AUSCRYPT'92*, number 718 in Lecture Notes in Computer Science, pages 492–504. Springer-Verlag, 1993.