Secret Exponent Information Leakage for Timing Analyses

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

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

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

{lbatina, cjansen }@safenet-inc.com

Abstract. Side-channel attacks are a real threat to implementations of cryptographic algorithms. In this paper we focus on the secret exponent information leakage in modular exponentiation algorithms (like used in RSA cryptosystems), caused by obtaining information on the number of square and multiply operations performed. Using a random exponent model in an information theoretic approach, the exponent leakage is shown to be too small to exploit in practice, e.g. 6.06 bits for a 1024 bit exponent. An extension of the model to observation of multiple parts of the exponent is also given.

1 Introduction

Today many analysis methods exist which are used to attack implementations in hardware or software of cryptographic algorithms of both types i.e. public-key and secret-key cryptography. Rather than attacking the algorithms themselves, specific properties of the implementation such as timing behaviour ([1], [4]) and power consumption ([5]) 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 this paper we focus on asymmetric i.e. public-key algorithms, in particular on modular exponentiation which is used in RSA cryptosystem ([6]). The first question we ask ourselves is how much information we obtain (in the Shannon sense) when observing the total number of operations, i.e. modular multiplications and squarings, in a certain part of the secret exponent.

The simplest instance of this problem is to consider a left-to-right square-and-multiply algorithm for exponentiation in which we cannot distinguish between multiplication and squarings. In a first order approximation it is assumed that the secret exponents are chosen at random from the set of odd numbers

$3, 5, ..., 2^n - 1$. For this random exponent model the exact probability distribution is derived, and the entropy $H(K)$ of the number of operations K is determined. The results is showing that this information is far from exploitable for the practical bit-lengths in nowadays applications.

This observation can be furthermore extended in observing the situation in which the attacker is able to attack some specific part(s) of the exponent. The relation between exponent information leakage and the number of exponent parts is also given.

The remainder of this paper is organized as follows. In Section 2 we use the combinatorial approach to develop a simple model for Hamming weight leakage. In Section 3 we conclude that Hamming weight information is not really useful information i.e. if only info is the number of squarings and multiplications, the adversary cannot do much with it. Further refinement of possible attack is considered in Section 4. The general relation between information leakage and exponent length is proven to be logarithmic. In particular, the information leakage is 6.06 bits for a 1024 bits exponent, which is far too little to be practically exploitable in an attack. Section 5 concludes the paper.

2 A Random Exponent Model

Let us first assume that the secret exponent is an arbitrary odd number. We want to compute the number of possible exponents of length l up to $n\,(l \leq n)$ which require exactly k operations (squares or multiplies) to be computed. We will compute this in a number of steps.

First, we determine the number of (different) exponents of length exactly n (so with MSB equal to 1), which require exactly k operations: Least significant bit (LSB) is 1, because the exponent, say e, is odd, and therefore we have to count all different sequences of 0's and 1's for the remaining $n-1$ bits. Let us first calculate the number of operations for computing M^e by use of the standard square-and-multiply algorithm ([3]):

Algorithm 1. Square-and-Multiply Algorithm (from MSB)
INPUT: M and $e = \sum_{i=1}^{n-1} e_i 2^i = (e_0, e_1, ..., e_{n-1})$
OUTPUT: $Res = M^e$
1. $Res \leftarrow M$
2. For *n-2* to 0 do:
   2.1 $Res \leftarrow Res^2 \bmod N$
   2.2 if $e_i = 1$ then $Res \leftarrow Res \cdot M \bmod N$
3. Return Res

So, the number of operations, which are squares (S) and multiplies (M) are determined with bits on positions from $n-2$ to 0. If this number is denoted as k, and w_H is its binary Hamming weight, we conclude:

$$k = n - 1 + w_H(e) - 1 = n - 2 + w_H(e). \tag{1}$$

We can now count how many exponents exist with some fixed number of operations k, which have bit-length exactly n. Here, LSB is also not taken into account (being fixed as 1). So, the number of different exponents is $\binom{n-2}{k-n}$ where $k = n, n+1, ..., 2n-2$.
Secondly, we want to determine the number of exponents of length up to n with exactly k operations involved in exponentiation. Let us denote this number with $G(n,k)$. We get the following:

$$G(n,k) = \sum_{i \geq \frac{k+2}{2}}^{n} \binom{i-2}{k-i} \tag{2}$$

To illustrate this formula let us find all exponents of length up to $n = 7$, for which $k = 8$. The possible lengths are $n = 5$: 11111; $n = 6$: 100111, 101011, 101101, 110011, 110101, 111001; and $n = 7$: 1000011, 1000101, 1001001, 1010001, 1100001. According to formula (2), we can easily check: $G(7,5) = \sum_{i=5}^{7} \binom{i-2}{8-i} = \binom{3}{3} + \binom{4}{2} + \binom{5}{1} = 1 + 6 + 5 = 12$.

Let us now rewrite the formula (2) in a different way to make calculations easier. We consider first the following case: $k = 2m$. In this case $G(n,k)$ becomes:

$$G(n,k) = \sum_{i=m+1}^{n} \binom{i-2}{2m-i} = \binom{m-1}{m-1} + \binom{m}{m-2} + \ldots + \binom{n-2}{2m-n}. \tag{3}$$

We will use the following theorem.

Theorem 1.

$$F_{n+1} = \sum_{j=0}^{\lfloor \frac{n}{2} \rfloor} \binom{n-j}{j} \qquad \forall n \geq 0. \tag{4}$$

Proof: Follows directly as sums of diagonals in Pascal's triangle ([7]).

It is easy to see that this equality can be applied on the equation (2) if and only if $n = k$ and in that case we get $G(n,n) = F_{n-1}$. This holds trivially for $n > k$. So, we conclude: $G(n,k) = F_{n-1}$, for all n, $n \geq k$, where k is fixed.
We still have to consider those cases for which $k > n$. Consider again the formula (4) (for k even):

$$\begin{aligned}
G(n,k) &= \sum_{i=\frac{k+2}{2}}^{n} \binom{i-2}{k-i} = \sum_{i=\frac{k+2}{2}}^{n} \left[\binom{i-3}{k-i} + \binom{i-3}{k-i-1} \right] \\
&= \sum_{i=\frac{k}{2}}^{n-1} \binom{i-2}{k-i-1} + \sum_{i=\frac{k}{2}}^{n-1} \binom{i-2}{k-i-2} \\
&= \sum_{i=\frac{k-1+2}{2}}^{n-1} \binom{i-2}{k-1-i} + \sum_{i=\frac{k-2+2}{2}}^{n-1} \binom{i-2}{k-2-i} \\
&= G(n-1,k-1) + G(n-1,k-2)
\end{aligned} \tag{5}$$

We obtain the recurrence, which can be further expanded:

$$\begin{aligned} G(n,k) &= G(n-1,k-1) + G(n-1,k-2) \\ &= G(n-2,k-4) + 2G(n-2,k-3) + G(n-2,k-2) \\ &= \ldots = \sum_{i=0}^{\frac{k}{2}-1} \binom{\frac{k}{2}-1}{i} G(n-\frac{k}{2}+1, 2+i) \end{aligned} \tag{6}$$

From here follows:

Theorem 2. *The numbers $G(n,k)$, for k even, satisfy the following equation:*

$$G(n,k) = G(n-1,k-2) + G(n-1,k-1) = \sum_{i=0}^{\frac{k}{2}-1} \binom{\frac{k}{2}-1}{i} G(n-\frac{k}{2}+1, 2+i). \tag{7}$$

Proof: By induction.

Using (7) we can easily compute $G(n,k)$ for arbitrary n and k. The numbers $G(n,k)$ are listed in Table 1, for $n \leq 12$ and $k \leq 20$.

Table 1. The behaviour of numbers $G(n,k)$.

k / n	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
2	1	0	0	...															0
3	1	1	1	0	...														0
4	1	1	2	2	1	0	...												0
5	1	1	2	3	4	3	1	0	...										0
6	1	1	2	3	5	7	7	4	1	0	...								0
7	1	1	2	3	5	8	12	14	11	5	1	0	...						0
8	1	1	2	3	5	8	13	20	26	25	16	6	1	0	...				0
9	1	1	2	3	5	8	13	21	33	46	51	41	22	7	1	0	0	0	0
10	1	1	2	3	5	8	13	21	34	54	79	97	92	63	29	8	1	0	0
11	1	1	2	3	5	8	13	21	34	55	88	133	176	189	155	92	37	9	1
12	1	1	2	3	5	8	13	21	34	55	89	143	221	309	365	244	247	46	1

The previous conclusions are easily verified. On the main diagonal we notice Fibonacci numbers. Also, the numbers under this diagonal are the same within each column. Namely, as we proved $G(n,k)$ does not depend on k, for all $k \leq n$ and this number is equal to $(n-1)^{\text{st}}$ Fibonacci number. We also note that the recurrence (5) holds.
We are interested in $G(10,14)$ i.e. how many exponents exist in the interval $[1, 2^{10}]$ which involve 14 operations while performing exponentiation. It is easy to check: $G(10,14) = \sum_{i=0}^{6} \binom{6}{i} G(4,2+i) = G(4,2) + 6G(4,3) + 15G(4,4) + 20G(4,5) + 15G(4,6) = 1 + 6 \cdot 1 + 15 \cdot 2 + 20 \cdot 2 + 15 \cdot 1 = 92$

Now let us assume that exponents are selected at random from the set of odd numbers $\{3, 5, \ldots, 2^n - 1\}$ with equal probability $(2^{n-1} - 1)^{-1}$. Let K denote the associated random variable denoting the number of operations in an exponentiation with the randomly selected exponent. The probability distribution of K is given by:

$$Pr[K = k] = (2^{n-1} - 1)^{-1} G(n, k). \tag{8}$$

As can be seen, even for very high values of n there is always a non-zero probability of very small number of operations.

3 Entropy of operations and exponent leakage

In the light of side-channel cryptanalysis, the previous consideration is tightly related to the following question: If the attacker possesses the exact information on number of operations k performed while exponentiation has been done, what is the probability for him to successfully guess the secret exponent e? We assume, as expected, that he has a detailed knowledge on algorithm of exponentiation and the relevant bit-lengths. So, we are interested in the expected amount of information that k gives about e. Formulated in an information theoretic sense: get an expression for the mutual information $I(K; E)$ between the random variables K and E, where E denotes the random exponent. This notion of given-away information of the key is introduced in [2].
For E and K random variables, we have:

$$\begin{aligned} I(K; E) &= H(K) - H(K|E) \\ &= H(E) - H(E|K). \end{aligned} \tag{9}$$

Here $H(K)$, $H(E)$ denote the entropies of K and E respectively, and $H(K|E)$ and $H(E|K)$ are conditional entropies of K with given E and vice versa. $H(E|K)$ is usually interpreted as the uncertainty about E given K.
Note that for any given exponent, the number of operations is exactly determined by (1). Hence, the uncertainty in K, given E is zero, resulting in:

$$\begin{aligned} &I(K; E) = H(K), \text{ and} \\ &H(E|K) = H(E) - H(K). \end{aligned} \tag{10}$$

Equations (10) clearly show that the information leakage is exactly equal to the entropy $H(K)$ of the number of operations. Having the exact probability distribution of this random variable from the previous section, this exponent leakage can precisely be determined.

By assuming a uniform probability distribution on K and observing that K takes on values $k \in \{2, 3, ..., 2n - 2\}$, we can easily bound $H(K)$ from above:

$$H(K) \leq \log_2(2n - 3) \text{ bits}. \tag{11}$$

Approximating $H(E) \approx n - 1$ we obtain the following lower bound on the conditional exponent uncertainty:

$$H(E|K) = H(E) - H(K) \geq n - 1 - \log_2(2n - 3) \text{ bits}. \tag{12}$$

From (11) and (12) the logarithmic relation between information leakage and maximum exponent length can be observed. Applying (12) for $n = 1024$ bits, we get $H(E|K) \geq 1012$, so the average amount of information that K gives about E is not more than 12 bits. However, from Table 2 it can be seen that the exact exponent leakage in this case amounts to 6.06 bits.

Table 2. Data for the entropy and upper bound for two models.

		Ran. exp. model		Binomial Model	
N	$Log(N)$	$H(K)$	Bound	H(K)	Bound
2	1	0	0	0	0
4	2	2.24	2.32	1.50	1.58
8	3	3.09	3.70	2.33	2.80
16	4	3.46	4.86	2.95	3.90
32	5	3.18	5.93	3.50	4.95
64	6	4.11	6.97	4.02	5.98
128	7	4.63	7.98	4.54	6.99
256	8	5.09	8.99	5.04	7.99
512	9	5.57	10.00	5.54	9.00
1024	10	6.06	11.00	6.05	10.00
2048	11	6.54	12.00	6.55	11.00

A slightly different model is one in which the most significant bit of the exponent is always taken to be 1. The justification of this model comes from the fact that in practice RSA key generation appliances may force the secret exponent to be of maximum length. For this model the length n is fixed and the Hamming weight of the exponent is a binomially distributed random variable. Consequently, the number of operations is a binomially distributed random variable, K, which takes on values $k \in \{n-2, \ldots, 2n-2\}$. Use of this distribution, yields the following upper bound on the entropy of K:

$$H(K) \leq \log_2(n+1) \text{ bits.} \tag{13}$$

The two bounds for $H(K)$ differ by ≈ 1 bit. For long exponent lengths, the contribution of short exponent tends to become negligible and the two models give identical results for exponent leakage. Figure 1 shows the exponent entropy loss for both models and their corresponding upper bounds.

4 Observation in parts

In this section we generalize the results from the previous section by considering the possibility that the attacker is observing the random exponent in two or more parts. The assumption here is that the number of operations (total of squares and multiplies when performing an exponentiation) in each part is known to the

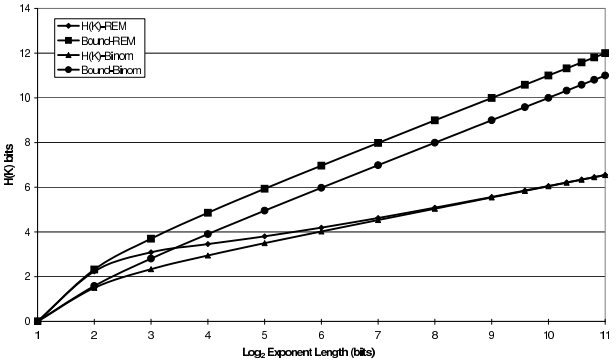

Fig. 1. Exponent entropy loss for two different models (Random Exponent Model (REM) and binomial model

adversary. Obviously, if each part comprises only one bit of the exponent, then he knows the value of that bit in each part exactly from the corresponding one or two operations observed. If, on the other hand, the number of parts is less, with more bits in each part, then the uncertainty for the attacker will be higher, as there is no longer a one-to-one correspondence between the values of the bits and the number of operations for each part. We consider the fixed exponent length model in which the number of operations has a binomial distribution. It is assumed that the exponent of length n bits is partitioned into parts of equal length l bits, with the exception of the most significant part, which contains the remaining $n - \lfloor \frac{n}{l} \rfloor l$ bits.

Let $h_B(m)$ denote the entropy (in bits) of a binomially distributed random variable, representing the number of operations in an exponent part of length m bits. Let $H(K^t)$ denote the total amount of information obtained by observing the operations from all parts. Then if $n - \lfloor \frac{n}{l} \rfloor l > 0$, the following relation holds:

$$H(K^t) = h_B(l-1) + \Big(\Big\lfloor \frac{n}{l} \Big\rfloor - 1\Big) h_B(l) + h_B(l)\Big(n - \Big\lfloor \frac{n}{l} \Big\rfloor l - 1\Big) \quad \text{bits.} \qquad (14)$$

This can be seen from the independence of the parts and the fact that the least and most significant bits are always 1. Figure 2 shows $H(K^t)$ as a function of the part-length for a 1024 exponent on a log-log scale. The steps in the curve are caused by the sudden change in the number of parts $\lfloor \frac{n}{l} \rfloor$.

A more realistic model in which exponents are taken to be random, but co-prime to a random even composite number should be further developed. Also according to [8] short RSA exponent can be successfully attacked, so in practice not all bit lengths are possible.

5 Conclusion

We have considered secret exponent information leakage for modular exponentiation algorithms in an information theoretic setting. To this end, expressions

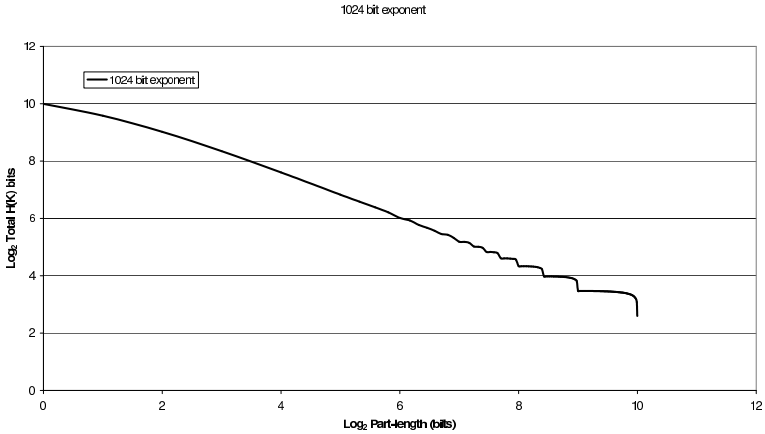

Fig. 2. Random Exponent Model - Observation in Parts

for the number of exponents resulting in a given number of square and multiply operations have been derived. These expressions straightforwardly lead to the probability distribution of the number of operations in the random exponent model. This random exponent model has subsequently been used to obtain data on the exponent leakage, showing that the amount of information is too small to exploit in practice. A further refinement of the model, considering the observation of multiple parts of the exponent, has also been given.

References

1. G. Hachez, F. Koeune, and J.-J. Quisquater. Timing attack: what can be achieved by a powerful adversary? *Proceedings of the 20th symposium on Information Theory in the Benelux*, pages 63–70, May 1999.
2. C. J. A. Jansen. Key signature schemes. *Proceedings of the Seventh Symposium on Information Theory in the Benelux, Noordwijkerhout, The Netherlands*, pages 197–205, 1986.
3. D. E. Knuth. *The Art of Computer Programming*, volume 2/Seminumerical Algorithms. Addison-Wesley, 1997.
4. P. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS and other systems. *Lecture Notes in Computer Science, Springer-Verlag*, pages 104–113, 1996. Advances in Cryptology-CRYPTO 96.
5. P. Kocher, J. Jaffe, and B. Jun. Differential power analysis. *Lecture Notes in Computer Science, Springer-Verlag*, pages 388–397, 1999. Advances in Cryptology-CRYPTO 99.
6. 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.
7. K. H. Rosen. *Handbook of Discrete and Combinatorial Mathematics*. CRC Press, 2000.
8. M. J. Wiener. Cryptanalysis of short rsa secret exponents. *IEEE Transactions on Information Theory*, 36(3):553–558, May 1990.