

A DPA attack on RSA in CRT mode

Marc Witteman
Riscure, The Netherlands

1 Introduction

RSA is the dominant public key cryptographic algorithm, and used in an increasing number of smart card applications. Modern smart cards include dedicated cryptographic processors to speed up processing time and can generally perform 1024 bit RSA operations in less than a second. Typical applications are in the payment and identification area, where public key cryptography can provide strong authentication combined with flexible key management. Transaction duration is generally an important aspect as this easily becomes a bottleneck for successful deployment.

A popular implementation variant of RSA, referred to as 'Chinese Remainder Theorem' (CRT) [1], can speed up the execution almost four times. Although the CRT variant is more complex than a straightforward binary exponentiation, it runs faster as it works with smaller numbers.

This paper describes a side channel attack on RSA CRT implementations that does not require any knowledge of the input message. Knowledge of the public key and side channel measurements of known signatures of variable messages are sufficient to recover the primes and the private key. As the method does not require knowledge of the input message, it is particularly effective on digital signature schemes. The attack method has been validated on smart cards and unless specific countermeasures in the cryptographic implementation are present, it is successful in retrieving both primes and the private key within one day of acquisition and analysis work.

Section 2 provides background knowledge on CRT, side channel attacks and related work. Section 3 explains the attack theory, while section 4 provides some additional information about Riscure.

2 Background information

This section provides background information on CRT, side channel analysis and related work.

CRT

A CRT (Chinese Remainder Theorem) implementation performs its execution in three steps: reduction, exponentiation and recombination. For this, it needs a number of pre-computed parameters. This RSA variant is specified in detail in section 14.5.2 of [1].

A brief recap of the formulas for the pre-computation and the execution:

Pre-computation

$$d_p = d \bmod (p-1) \quad // d \text{ is the private exponent, } p \text{ is one of the primes}$$

$$d_q = d \bmod (q-1) \quad // q \text{ is the other prime}$$

$$k = p^{-1} \bmod q$$

Reduction

$$m_p = m \bmod p \quad // m \text{ is the message to sign}$$

$$m_q = m \bmod q$$

Exponentiation

$$s_p = m_p^{d_p} \bmod p$$

$$s_q = m_q^{d_q} \bmod q$$

Recombination

$$s = ((s_q - s_p) * K) \bmod q + s_p \quad // s = m^d \bmod n \text{ is the signature}$$

Side channel analysis

Secure devices deploying a carefully designed communication protocol do not disclose any secret data contained in the device directly. Such a device can still leak data via other forms of physical interaction, known as side channels. For example, time duration can reveal parts of a secret, as the time taken by the chip to perform an operation may depend upon branches in its programming controlled by secret data. Further, the device's power consumption and electromagnetic field can be measured and subsequently related to the operations on a secret.

In practice, power analysis is one of the most fruitful side channels. Broadly speaking, the power consumed by a chip in any fraction of a second depends on

the total number of bits changed during that time frame. A detailed trace of the power consumed during an operation can be made using a high frequency oscilloscope in combination with special interface equipment. This trace can then be examined for patterns corresponding to the operation performed.

Occasionally, a few traces will already contain sufficient information to extract the secrets from a chip – this is called Simple Power Analysis (SPA). Otherwise, a technique known as Differential Power Analysis (DPA) [2] can be applied: several thousands of traces are measured, each one stored along with the data that was given to the chip for its transaction. Various statistical analyses are then performed, essentially focusing on a known or estimated bit of data at a time and determining where it is processed: the traces are divided into 2 groups depending on the value the chosen bit must have had, an average is taken over each group and the two averages are compared. The difference shows significant peaks at points at which the power consumed depends on the chosen bit. By careful application of such techniques, the power consumption can be analyzed to reveal a secret key. An improvement of this approach is found by computing correlation traces rather than differential traces: this method computes correlation between samples and a property of multiple bits of data.

The attack described in this paper makes use of correlation traces with as a property the Hamming-weight of the processed data bytes to increase the signal-to-noise ratio and accelerate the analysis. Further, it is a known cipher text attack and it does not pose any specific requirements on the CRT implementation.

Related work

In the past, several attacks on CRT-based exponentiation algorithms have been presented. One of the first – and simplest – methods is the one described by Novak [3]. This SPA attack focuses on the first part of the recombination, where s_p is subtracted from s_q . Since the result of the subtraction can be negative, an addition of q to the least positive residue is required. By using an adaptive-chosen cleartext sequence an attacker can observe over a side channel trace whether the conversion is performed. This sequence ultimately leads to the recovery of primes p and q .

In 2003 Foque, Martinet and Poupard [4] presented an extension of Novak's attack, where the message is only required to be known, not necessarily chosen. This enables the attack in cases where padding schemes make Novak's original attack

unfeasible. However, this attack only works when the difference in length of the prime factors is small (e.g. 10 bits).

Several DPA attacks have also been proposed, the first publicly available being the one presented by Boer, Lemke and Wicke at CHES 2002 [5]. The technique they used has been named “Modular Reduction on Equidistant Data”. By choosing equidistant input data to the RSA operation, it is possible to attack the modular reduction step and compromise the exponent one byte at a time. The drawback of this method is that the attacker is required to have the possibility to choose messages.

Another attack on the modular reduction step was proposed by Jaffe in 2006 [6]. This attack does not require the attacker to choose the message as it is a known text attack, but it requires the modulus operation to be implemented as a long division. If this is the case, the attack easily allows the retrieval of both primes, by means of approximation of their value starting either from the most significant or from the least significant byte.

After developing this attack and writing this paper, the author discovered that along with several other attacks on public key algorithms, Amiel et al. [7] provide a compact description of a similar attack technique on RSA-CRT to the one provided in this paper. At the time of writing, the publication of Amiel et al. [7] is not freely available on the internet.

3 Attack theory

The side channel analysis method targets the prime p in the recombination step. Consider the following rewritten recombination formula:

$$s = x * p + s_p$$

where $x = ((s_q - s_p) * K) \bmod q$ represents an intermediate internal value.

Further, note that the bit length of signature s equals the bit length of n , but the bit length of s_p equals the length of prime p , generally equal to half the length of n . This implies that the most significant half of s is almost equal to the most significant half of $x * p$. We can therefore use $x * p$ as an approximation of s and neglect the relatively small term s_p :

$$s \approx x * p \quad // \ x * p \text{ is an approximation of } s$$

We could now estimate $x \approx s / p$ if we knew p (signature s is assumed to be public). Since x is an intermediate value processed in the secure device, we should be able to observe correlation between x and traces measured during repeated signing operations. This implies that if we do not know p we can use correlation traces to test hypothetical values of p . In other words: if we try all hypothetical values p' for p and correlate the corresponding values of $x' = s / p'$ with side channel traces, only the correct value of p' demonstrates significant correlation.

A practical problem that occurs here is that the entropy of p is far too large to test all hypothetical values. We therefore use a divide-and-conquer approach with three steps: partitioning, repetition and brute forcing. The following sections elaborate these steps.

Partitioning

The complexity of the problem is reduced by breaking p into smaller parts. We use a binary representation for the variables, and write p_i to indicate byte i of the representation of p , with p_0 the most significant byte. Note that x'_0 depends mostly on p'_0 , and we can therefore define a function y using the most significant byte of hypothesis p' :

$$y(p'_0) = s / p'_0$$

This definition implies the most significant byte of $y(p'_0)$ is similar to the most significant byte of $x' = s / p'$:

$$y_0(p'_0) \approx x'_0$$

In fact, it may be a little larger since we neglect subsequent bytes of p' in the denominator. Next, we correlate computed values of $y_0(p'_0)$ for each hypothesis p'_0 . For values of p'_0 close to the real p_0 , we know x'_0 is close to x_0 and thus that $y_0(p'_0)$ should show correlation.

Note that several values near p'_0 may result in significant correlation for $y_0(p'_0)$ with the sample traces. For instance, if s_0s_1 (concatenation of the most significant two bytes of s) are $0x2378$, then $0xDE \leq p'_0 \leq 0xE3$ will all lead to the same value $y_0(p'_0) = 0x28$. Especially in a noisy environment it is very well possible that the p'_0 corresponding to the best correlation value does not represent the actual p_0 , but a value within a short range around p_0 . In the next section we demonstrate that this complication can actually be exploited to speed up the attack.

Repetition

After fixing the p'_0 with the highest correlation, we repeat the attack to find subsequent bytes of p' . We extend the definition of y , by using the concatenation of p'_0 and p'_1 :

$$y(p'_0p'_1) = s / p'_0p'_1$$

We now test for correlation with $y_1(p'_0p'_1)$ as this value strongly depends on p'_1 , whereas $y_0(p'_0p'_1)$ hardly depends on p'_1 . For example, if $s = 0x2378A52E$, $p'_0 = 0xE1$ and p'_1 can take any value, then $y(p'_0p'_1)$ ranges between $0x282E$ and $0x285B$.

Again we try all hypothetical values for p'_1 , but this is not sufficient. Since the value of p'_0 was only approximated in the previous step, we also have to test adjacent values for p'_0 . We select a range r , for which experimentally 16 seems to be a good value. Next we vary the concatenation of the two prime bytes for $r*256$ values, where the previously found value for p'_0 would be in the middle of range r .

This second hypothesis testing confirms the right value of p'_0 , because wrong values of p'_0 would not be able to predict $y_1(p'_0p'_1)$. It also finds a rough approximation for p'_1 . With the previous example: if p'_0 would be inaccurately set to $0xE0$ then $y(p'_0p'_1)$ would vary between $0x285B$ and $0x2889$, for which p'_1 is almost completely outside the correct range.

This step can be repeated to recover subsequent prime bytes by seeking correlation with bytes of $y_k(p'_0p'_1...p'_k)$ for increasing k . However, this approach faces a complication for the last byte of p .

Brute force

In the attack principle we chose to neglect s_p , since this value is relatively small and mostly affects the lower half of s . However, when we try to recover the least significant prime byte, this is tested by observing correlation with the least significant byte of x' .

Note that $x = (s - s_p) / p$, with s_p smaller than p . From this we can deduce $x' = s / p$ will for roughly half of the signatures be one higher than x . When testing correlation with the least significant byte of x' (to find the least significant byte of p) the correlation results will be deteriorated due to this small difference. This could be overcome by using more sample traces, but there is a more effective approach.

As we have the public key, we know modulus n . By definition we know $n = p * q$. Since p and q are primes we know that n has no other divisors than these primes. Rather than trying to find all bytes of p' we will do trial divisions of n / p' , for the last part of p' , so if we have one uncertain byte we could do 256 trial divisions. If the division $n / p' = q'$ leaves no remainder we know that we have found both p and q .

This last part of the attack is mathematically exact since there is no risk of errors due to approximations or noise. Therefore, this brute-force approach is preferable over experimental retrieval of the final bytes of p , as long as it is computationally feasible. In practice this means that the last two bytes would be brute-forced. With margin r for the previous byte this requires at most $r * 2^{16}$ trial divisions, which is a matter of a few seconds.

As soon as p and q are computed, the private exponent d is trivially computed from public exponent e and both primes by: $d = e^{-1} \text{ mod } \text{lcm}(p-1, q-1)$.

4 About Riscure

Riscure is a security test laboratory and security test tool vendor specialized in smart card and embedded device technology. Riscure performs thorough and innovative security evaluations for vendors and issuers of payment, pay-tv, telecom and identity applications. Further, Riscure developed the Inspector [8] side channel test platform which their clients use to evaluate the threats of side channel attacks on smart cards and embedded devices. Riscure continuously researches innovative analysis methods for encryption algorithms and new technology. Last year's release of Inspector includes the test method described in this paper.

For more information please contact Pascal van Gimst, Director Sales & Business Development at vangimst@riscure.com or +31 15 251 4090. More information about Riscure can be found at www.riscure.com.

5 References

- [1] A. Menezes, P. van Oorschot, and S. Vanstone, "Handbook of Applied Cryptography", CRC Press, 1996.
- [2] Paul Kocher, Joshua Jaffe and Benjamin Jun, "Differential Power Analysis", in Proceedings of Advances in Cryptology – CRYPTO'99, Springer – Verlag, 1999.
- [3] R. Novak, "SPA-Based Adaptive Chosen-Ciphertext Attack on RSA Implementation", 2002.
- [4] P.A. Fouque, G. Martinet, G. Poupard, "Attacking Unbalanced RSA-CRT Using SPA", 2003.
- [5] B. den Boer, K. Lemke, and G. Wicke, "A DPA Attack Against the Modular Reduction within a CRT Implementation of RSA", 2002.
- [6] J. Jaffe, "More Differential Power Analysis: Selected DPA Attacks", 2006.
- [7] F.Amiel, B.Feix and K.Villegas, "Power Analysis for Secret Recovering and Reverse Engineering of Public Key Algorithms", 2007
- [8] Inspector side channel test tool: <http://www.riscure.com/inspector.html>