Performance Analysis of Timing Attack on Elliptic Curve Cryptosystem

Mr. Praful V. Barekar
Master of Technology Scholar
Department of CSE
G. H. Raisoni College Of Engineering,
Nagpur

Prof. K. N. Hande
Assistant Professor
Department of CSE
G. H. Raisoni College Of Engineering,
Nagpur

1. Abstract

Cryptosystems often take slightly different amounts of running time depending on the input and the used key. This timing information, extracted from the decryption process, can be used to derive information about the secret key. This new class of attacks on implementations of cryptosystems is named Timing Attacks. Timing attacks attempt to exploit the variations in computational time for private key operations to guess the private key. This type of attack is primitive in the sense that no specialized equipment is needed. An attacker can break a key by simply measuring the computational time required by the user inputs and recording those user inputs. This paper is aimed to analyse the performance of Timing Attack on Elliptic Curve Cryptosystem. The main advantage of Elliptic Curve Cryptography is smaller key size, it is mostly used for public key infrastructure

Keywords: Cryptosystem, Timing Attack, Running Time, Elliptic Curve Cryptography, Public key Infrastructure.

2. Introduction

Timing Attacks were first introduced in a paper by Kocher [4]. Kocher describes the general idea of Timing Attack and shortly reports on some results for the RSAREF implementation of the RSA cryptosystem. He also claims that the same idea can be used for discrete logarithm based cryptosystems like Diffie-Hellman, DSS, and other systems. Later Kocher's timing attack on RSA was modified and practically examined, e.g. by Dhem et.al. on a RSA smart card implementation [6], and by Brumley and Boneh on the RSA implementation of the OpenSSL library [1]. We do however not know about any successful practical results for timing attacks when applied to elliptic curve cryptosystems.

Elliptic curve (EC) cryptosystems have gained large support in recent years after several standard documents on public key cryptography included EC cryptosystems and EC signature schemes [7]. Their main advantage in comparison to RSA is significantly smaller key sizes for similar security levels. The Wireless Transport Layer Security (WTLS) specification for securing wireless applications therefore also explicitly supports elliptic curves cryptography for wireless applications [8]. In this paper, we report on experimental results for timing attacks when applied to elliptic curve cryptosystems. We use a software implementation of the basic scalar point multiplication algorithm for points defined over finite prime fields GF(p) as described in [7]. We start with a short explanation of the necessary facts about elliptic curves, timing attack mechanism, Finally we describe practical results from our simulation of timing attacks.

3. Basics of Elliptic Curve

This section explains some necessary background information on elliptic curves (for a more detailed description, see [7]). Let GF(p) be the finite prime field with p elements, where p is prime. We define the group of points E(GF(p)) on an **elliptic curve** (a, b) ∈ GF(p)2 as the set of solutions (x, y) ∈ GF(p)2 to the equation

$$y^2 \equiv x^3 + a\,x + b\ (mod\ p)\ (1)$$

Together with a point at infinity O = (∞,∞). These points form an abelian (additive) group where the group operation is defined by the following formulas:

- O is the zero element.
- The negative point of (x, y) is the point (x, -y).
- For two non-zero points P1 = (x1, y1) and P2 = (x2, y2) with P1≠ P2, we determine the sum P3 = (x3, y3) = P1 + P2 as

$$x_3 \equiv -x_1 - x_2 - l^2\,(mod\ p) \qquad (2)$$
$$y_3 \equiv -y_1 + l(x_1 - x_3)(mod\ p) \quad (3)$$

Where $l \equiv (3x_1^{\,2} + a) / (2\,y_1)\ (mod\ p)$ if P1 = P2, and $l \equiv (y2 - y1) / (x2 - x1)\ (mod\ p)$ otherwise. For an integer k > 0 and a point P, we define scalar multiplication k · P for the point P as adding P (k -1) times to itself. There exist various algorithms for computing scalar multiplication (see [3]). In this paper, we consider only the following basic binary left-to-right fast multiplication algorithm:

Algorithm 1: (Scalar Point Multiplication)

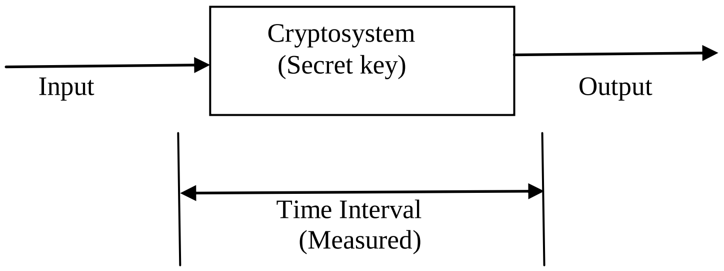
```
Input:  Point P, integer k > 0 with binary
          representation k = (1 kw-2... k0)2.
Output: Point k · P
1. Let Q = P
2. for j = w - 2 downto 0 do
       Q = 2 · Q
        if kj = 1 then Q = Q + P
   done
3. return Q
```

Definition 1: The **Elliptic Curve Discrete Logarithm Problem** (ECDLP) is defined as the problem computing the integer k only with knowledge of the elliptic curve E and the two points P and $Q = K\cdot P \in E(GF(p))$ without prior knowledge of k. ECDLP is generally considered to be a very difficult problem if the field characteristic p is sufficiently large (at least 160 bit), and the order of the group of points on the elliptic curve is prime (or "almost prime"). Up to now, there exists no algorithm that can solve ECDLP for such curves in reasonable time. Therefore point multiplication was chosen as the "trapdoor one-way function" used in elliptic curve public key cryptosystems (ECC) and elliptic curve signature schemes (ECDSA). For more details on applications of elliptic curves for public key cryptography, we refer to [7]. Timing attack use the following scenario: Assume that the unknown secret key k is stored securely and cannot be accessed directly. We can compute **k · Pi** for arbitrarily many random points Pi. Moreover we are able to determine the running time Ti of each of these computations (in our simulation, we uses Algorithm 1 for fast multiplication). We might also be able to determine running times for a single point addition or point doubling for arbitrarily chosen points.

4. Timing Attack

Recently, a new class of cryptanalysis aimed at a cryptosystem's implementation-specific weaknesses has attracted great interest. This kind of cryptanalysis exploits the leak of information such as timing, power consumption, and electromagnetic radiation from system operations to facilitate attacks on the cryptosystem. Since the information used by the attack is not in the "main channel", the input or output, we call these types of attacks "side-channel" attacks. In this paper, we will focus on timing attacks.

Let's think the cryptosystem as a black box with input and output which constitute the "main channel" of the system. We can measure the time it takes for the system to give an output after given an input. The time required for different inputs may vary, forming a timing distribution. If this timing distribution is related to the secret (key bits) in the system, we may have a way to reveal the secret key.



5. The Attacker's Task

The attacker has the ability to observe a sequence of elliptic curve operations, thus, the attacker's aim is to calculate and exploit the probabilities of certain sequences of bits given an observed sequence of elliptic curve operations. Using the information of such conditional probabilities, the key-space that has to be searched to find the correct ephemeral key, can be significantly reduced. This is because certain combinations of patterns in the power trace and certain combination of digits are less likely than the others (or even not possible at all). The attacker's task can be stated in a more formal way. Let X be a random variable that denotes a sequence of elliptic-curve operations and $|X|$ the length of X (i.e. the number of elliptic-curve operations in this sequence). For example, X="DDD" (i.e. the realization of the random variable X consists of three consecutive elliptic-curve point-double operations) thus $|X| = 3$, or X="DAD" (i.e. the realization of the random variable X consists of an elliptic-curve point-double operation, an elliptic-curve point-addition operation and an elliptic-curve point-double operation) thus $|X| = 3$. Let Y be a random variable that denotes a sequence of digits in the digit representation of k and $|Y|$ the length of Y (i.e. the number of digits). For example Y = "000" (i.e. the realization of the random variable Y consists of three consecutive zeros) thus $|Y| = 3$, or Y = "01" (i.e. the realization of the random variable Y consists of a zero and a one digit) thus $|Y| = 2$. Then the attacker's goal is to calculate and exploit the conditional probability.

6. Mathematical Model

Let us denote a set of inputs (plaintexts) to the system by $S_M = \{M_1,M_2,M_3,\ldots\ldots M_n\}$ All the possible keys compose the key set denoted by $S_K = \{K_1,K_2,K_3,\ldots\ldots\ldots\ldots K_d\}$ where d is the number of possible keys. If the cryptosystem implementation we want to attack is vulnerable to timing attacks, the timing distribution of the input will be dependent on the key used in the system. Thus for key K_i , we will have a timing distribution donated by) $P_i(t)= F(S_M , K_i)$ which is different from that of other keys.

For the system we want to attack, we measure the timing information for a set of input values from the set S_M, and form a timing distribution P(t). The attack to the system will be reduced to a usual detection problem which tries to detect K_i knowing P_i (t) and P(t). We can apply, at least in theory, regular detection solutions to solve the problem. For example, the detection problem has a general form of the solution: if T (P(t), K_i) > Threshold (K_i , S_M) , K_i is detected. As long as we find the proper transform function T() and the threshold functions, we break the system.

7. Timing Attack applied to Elliptic Curves

Timing attack is based on the following idea. Denote by k_j the j-th bit of the secret key k. Then we get the following equation for the total running time in Algorithm 1 with input point P_i :

$$T_i = e_i + \sum_{j=0}^{w-1} (D_{i,j} + k_j A_{i,j}). \qquad (4)$$

In this formula, $D_{i,j}$ denotes the time needed for a point doubling operation for bit j, and $A_{i,j}$ denotes the time for an addition operation for bit j, e_i is some "noise" (run time for looping, if-operation, and other external influences). Note that both the doubling time $D_{i,j}$ and the addition time $A_{i,j}$ depend on the chosen random point P_i (index i) and the iteration index j. Let 0 < r < w, and assume that we already know the "upper" bits k_{w-1} , ... , k_{r+1} of the bin- ary representation of the secret key k. Kocher's fundamental idea is the fact that we can determine the values of $D_{i,r}$ and $A_{i,r}$ with this information. For every sample point P_i we use the known bits of k to determine the value of the point Q at the beginning of iteration j = r in Algorithm 1. Once we know these points we can determine both $D_{i,r}$ and $A_{i,r}$ using the decryption device (we assume that we can determine the run time for a single point operation). The next step of timing attack is the usage of $D_{i,r}$ and $A_{i,r}$ to determine a (probable) value for the next bit k_r using statistical methods. Similar to the above method, we can assume that we also know all intermediate timing values $D_{i,j}$ and $A_{i,j}$ for all r ≤ j ≤ w-1 and all samples indeces i. Therefore we can compute T_i minus the "upper part" of the sum in (4)

8. Experimental Results

Table 1 describes some practical information applied to random scalars of different sizes. We show the length of prime number, the complete running time, and the average number of iterations for one bit of the scalar. It should be noted that all the point computations and field inversions were repeated several times, until the variance of the single timings for these operations was sufficiently small (and hopefully the timing error sufficiently small). Seeing the number of iterations (i.e. backtracks) especially for large scalars, we dare to conclude that several parameters of our implementation of the timing attack algorithm (e.g. the sample number, the definition of "sufficiently different") are not yet chosen optimally. Further examinations and determination of optimal parameters have to be done in future.

Length of Prime Number	Average number of iterations per bit	Total Running Time
2^{24}-1	3.54	11 min 13 sec
2^{24}	0.71	1 min 20 sec
2^{32}-1	1.16	18 min 37 sec
2^{32}	0.84	7 min 1 sec
2^{48}-1	1.10	27 min 15 sec
2^{48}	0.94	12 min 8 sec
2^{96}-1	2.33	53 min 32 sec
2^{96}	1.28	22 min 7 sec
2^{128}-1	4.06	58 min 8 sec
2^{128}	1.37	43 min 32 sec
2^{159}-1	7.38	59 min 26 sec
2^{159}	1.88	53 min 15 sec

9. Conclusion

The running time of the attack can be several hours; it always succeeded to determine the secret scalar. We are optimistic that further experiments can greatly improve the still large running time by searching for more optimized parameters. Therefore timing attacks should be considered as a serious threat for EC security system implementations in mobile applications and hence anticipated. Anticipating timing attacks is quite simple

10. References

[1] Ekambaram Kesavulu Reddy, “Elliptic Curve Cryptosystems and Side-channel Attacks”, Published in international Journals of Network Security, 2009.

[2] Dhem J.-F., Koeune F., Leroux P.-A., Mestre P., Quisquater J.-J., and Willem J.- L.s, “A practical implementation of the timing attack”, Proc. of the Third Working Conference on Smart Card Research and Advanced Applications (CARDIS 1998), LNCS 1820, 1998.

[3] Marc Joye, “Elliptic curves and side channel analysis”, published in ST Journal of System Research, 2003.

[4] Boneh D. and Brumley D. Remote timing attacks are practical, To appear in the 12th Usenix Security Symposium, 2003.

[5] Lu Z., Mah M., Neve M., and Peeters E., “Timing Attacks on Elliptic Curve Cryptosystems”, Project Presentation for course "CS588: Cryptology Principles and Applications, Fall 2001", University of Virginia, Department of Computer Science, available at

[6] Kocher P., Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and the Systems, In N. Koblitz, editor, Advances in Cryptology - CRYPTO'96, Lecture Notes in Computer Science vol. 1109, pp. 104 - 113, 1996.

[7] Wireless Application Protocol Forum, Wireless Transportation Layer Security, Version WAP-261-WTLS-20010406-a,21. April 2001,

[8] An Implementation Tutorial on “Elliptic Curve Cryptography”, By Anoop MS.