SIDE CHANNEL ATTACK USING POWER ANALYSIS

K. RAHIMUNNISA
Assistant. Professor,
Karunya University,
Coimbatore.
Email: krahimunnisa@gmail.com

KAVYA T.S
Research Scholar,
Karunya University,
Coimbatore.
Email: kavyats86@gmail.com

ANOOP SURAJ A
Research Scholar,
Karunya University,
Coimbatore.
Email: anoopsuraj@ieee.org

Abstract

The ingenuity of the software involved in the theoretical basis of cryptographic algorithms is indeed phenomenally strong. But they can be broken only through the weakness of their implementations. This weakness of the theoretical base is manipulated to explore into the various dimensions and possibilities of Side Channel Attack (SCA). This in fact forms the core of the methodology employed in the study of SCA as described in this paper. This paper explores into the possibilities and uses of various Side Channel Attacks meticulously. SCA exploits the unintended information leakage from the implementation to extract the secret key. The information elicited via side channel attack may include power consumed by the system, timing parameters, electromagnetic information etc. The specific aim of this paper is directed towards eliciting side channel information through power analysis attack. The process also invariably makes use of a powerful Boolean technique, which utilises the Boolean form of extracted variables. This Boolean reasoning technique will try for all possible combinations and find out a true assignment, if there exist any. This system can be extended to attack cryptographic software implementations. SCA substantially reduces the complexity of performing cryptanalysis and defines it in a new light.

Keywords: **Cryptanalysis, Side Channel Attack, Power Analysis Attacks.**

I. Introduction

Many electronic systems contain implementations of cryptographic algorithms in order to provide security. Most of the security systems are broken by exploiting the weakness in their implementations. So considering the security during the entire system design is important. Cryptographic primitives such as encryption and hashing algorithms form the basis of most of the security systems. A cryptographic system may be treated as a mathematical function that performs a given mapping of its input to its output. The function can be implemented as a hardware or software unit. The term Cryptanalysis refers to the process of breaking a cryptographic system without a brute force search. This can be implemented by analyzing the statistical properties of the outputs under the application of targeted inputs [1]. These attacks are said to be infeasible in practice since large amount of data is required for the implementation.

A powerful class of attack is the side channel attack. All side channel attacks can be viewed as consisting of two phases.

- An observation phase, wherein information is gathered from the target system.
- Analysis phase in which the collected information is used to infer the secret key.

In this work, side channel attack is defined as which utilizes side channel information such as operation timing [3], [4], power dissipation [5], [6], [7], electromagnetic radiation [8], [9] and behaviour in the presence of induced faults [10]. It is proved that small amount of leaked information is enough to break the secret key [11]. Many methods are proposed to counter the side channel attack [12], [13], [14]. All these techniques are used to minimize the presence of side channel attacks. But it is very difficult to eliminate them completely [15].

Hardware and software implementations are more or less equally vulnerable for side channel attacks. Data exposure will occur in software implementation through memory bus exposure, core dump files etc. Even in the most secure software implementation also the problem of data exposure exists [16]. Possibility of data exposure from software computation after the computation is over is a real threat to the security [17].

In this work side channel information is utilized to propose a frame work for side channel attack. The analysis phase of the attack is formulated as a Boolean search problem and the deduction phase is performed using state of art satisfiability (SAT) solvers. This approach will increase the scope of side channel attacks by allowing a wide range of internal variables to be exploited. We know that in a DES algorithm, 16 rounds are there. By knowing the input to the S-Box in a round will allow the attacker to calculate the key. So the secret keys and other easy targets are often protected from exposure. On chip key generation and storage may be one way to protect the data [18]. Most of the variable values

seem to be harmless. But if it is exposed, it may be sufficient to deduce the secret keys with the help of powerful analysis techniques such as SAT solvers.

The various ways of extracting side channel variables includes

Timing Analysis: This uses the time taken by the crypto system to execute cryptographic algorithm. Timing attacks are often overlooked in the design phase because they are so dependent on the implementation. Also timing attack is algorithm independent [3], [4].

Power Analysis: This type of attack analyzes the power consumption of the unit while it performs cryptographic operation. Power analysis can be classified into Simple power analysis, Differential power analysis and Higher order differential power analysis [5], [6].

Tempest Attack: Tempest can be termed as Tiny Electro Magnetic Particles Emitting Secret Things. This type of attack uses the electromagnetic radiations emitted from the system while performing a cryptographic operation [8].

Acoustic Cryptanalysis: This type of attack utilizes the sound which may be audible during the time of computation. The sound produced by different keys can be observed and make use by the attacker.

Differential Fault Analysis: This type of attack relates to the ability to investigate ciphers and extracts key by generating faults in a system. Satisfiability is the problem of determining, whether the variable of a given Boolean formula can be assigned in such a way as to evaluate to true. Boolean formula is expressed in conjuctive normal form (CNF) which consists of literals and clauses. Boolean function is written only using AND, OR and NOT functions. A literal is either a variable or negation of that variable. A clause is a disjunction of literals. The function of SAT solver is to find the satisfying assignment to any Boolean formula, if there exist any. SAT problem may be classified as either 3-SAT (clauses are limited to at most three literals), 2-SAT (clauses are limited to at most two literals) and horn SAT (clauses are limited to at most one positive literal). Even for many EDA applications we can use SAT solvers which motivated advances in SAT solving techniques. SAT software tools are freely available [19], [20]. Many efficient heuristic methods exist to solve real life SAT problems.

This work gives some ample evidence of software and hardware leakage. There is no general frame work to transform this type of leakage into actual attack on security system. Usually implementation of a cryptographic system will take basic measures to protect keys and other directly related variables from leakage. So in such cases powerful reasoning methods like SAT solvers are required. In security point of view, knowledge of the internal variables that can be used to launch side channel attack can be included in the design guidelines to ensure more security. Earlier a model is proposed for DES as a SAT formulation to study the properties of DES algorithm and for traditional cryptanalysis [21]. But this is based on the knowledge of plain text and cipher text. But the results showed the inability of SAT to solve traditional cryptanalysis. This paper purely deals with the attempt to apply Boolean analysis technique to side channel attack.

II. Motivation

In this section we motivate the prevalence of leakage of intermediate variables while communicating and software computations.

A. Leakage of software variable values

The architecture of a typical hardware software system implementing a cryptographic algorithm is shown in the Fig. 1. The system will map a plain text to a cipher text depending upon the given secret key. The architecture is divided into hardware and software part. Operating system and the application programs that are layered on the top of system library will form the software section. Operating system interacts with the hardware subsystem using machine instructions. The hardware section consists of the processor, I/O devices and the memory system made up of on chip cache memory and system memory [2]. The processor has an on chip memory which stores the cryptographic key, thereby preventing the exposure of the key bits through the operations which causes it to be transmitted outside the processors secure perimeter [22]. The complexities involved in implementing such a hardware software system open a path for intentional or unintentional leakage of data values at various interfaces, which includes application – library, application – OS, library – OS and OS – hardware.

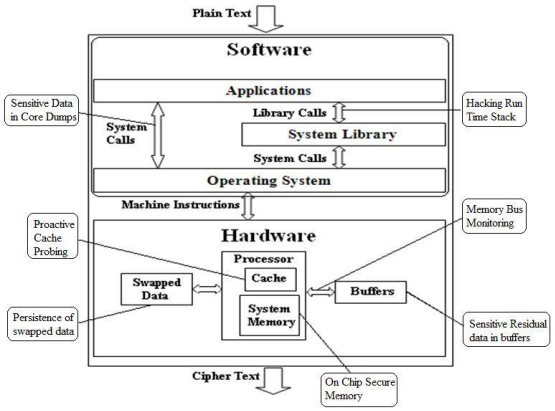

Fig.1. Leakage of software variable values

Unintentional leakage: This happens during normal operation due to bugs, improper policies etc. Jim

Chow [17] showed the existence of program data in system buffers in main memory long after the program terminated. Swapping of program data to disks greatly increases the probability of data exposure. Data will be present in the disk and there are many ways to extract that data [23]. Core dumps occur when an application program crashes after performing an illegal operation. This information can expose program data [22]. System crash report generated by a widely used OS will reveal sensitive information [24].

Intentional leakage: This is precipitated by attacks which are crafted to exploit latent system vulnerabilities. Examples of such malicious hardware or software vulnerabilities include hacking the run-time stack [25], proactively probing the cache using a Trojan process [26], monitoring the memory traffic on the system bus [27], *etc*. Also, there exist tools for dynamically examining the contents of program memory as the program is being executed [28]. Most of the software systems are implemented with the aim of maintaining security of data. But these systems also show security breaches [29].

B. Power Analysis Attack

Power Analysis attack utilises the power consumption of a device while performing a cryptographic operation. To measure a circuit's power, a resistor is inserted in series with the power or ground input. The voltage difference across the resistor divided by the resistance yields the current. Well equipped electronics labs have equipment that can digitally sample voltages at extra ordinary higher rates. Here the power traces of a smart card while performing encryption operation is analysed to proceed with the simple power analysis [5].

Simple Power Analysis (SPA) is a technique that involves directly interpreting power consumption measurements collected during cryptographic operations. SPA can yield information about a devices operation as well as key materials.

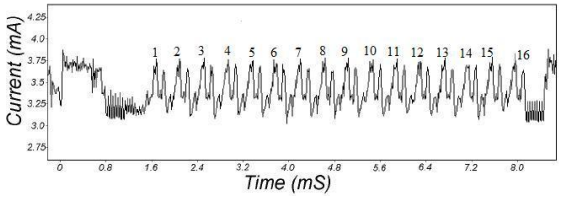

Fig.2. SPA traces showing an entire DES operation.

Fig. 2 shows the SPA trace from a typical smart card as it performs a DES operation. All the 16 rounds are clearly visible in the graph [5]. A trace refers to a set of power consumption measurements taken across a cryptographic operation. Fig.3 shows magnified view of the trace shown above, showing the second and third rounds of encryption and decryption operation.

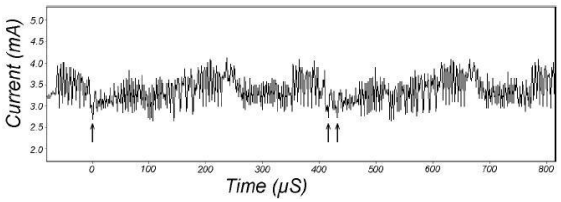

Fig.3. SPA trace showing DES round 2 and 3.

Many details of the DES algorithm operation is now visible [5]. As an example, the first and second 28 bit DES key registers are rotated once in round 2 and twice in round 3. This difference is indicated in the diagram with an up arrow key.

SPA can reveal the sequence of instructions being executed. So it can be used to break cryptographic implementations in which the execution path depends on the data being processed.

Attack on an 8052 microcontroller is discussed here. In the first stage it is useful to perform simple operations on the microcontroller and play around with the various possibilities for the measurement setup. The aim is to optimize the measurement setup of the microcontroller board. The next step is to choose the right value for resistor, R_m between global supply and the supply pin of the controller across which the current profile is taken. The value of R_m lies between 1ohms and 20 ohms. It is better to use a differential probe with high input resistance and very low input capacitance [30].

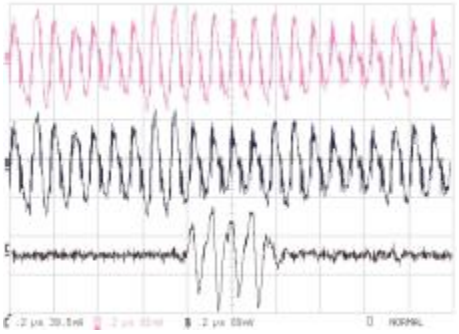

Fig.4. MOV 00 versus MOV FF instructions.

Fig.4 shows the current traces of the microcontroller while performing some simple MOV instructions. First and Second traces shows the profile of MOV 00 and MOV FF commands while the third trace is the zoomed difference of the two MOV commands [30].

III. SAT Frame Work for Enabling SCA

In this section, details of the proposed SAT based cryptanalysis framework are given. This technique is general and can be applied to any algorithm. The functionality of the cryptographic algorithm being targeted has to be represented as an equivalent Boolean Formula in conjuctive normal form (CNF).

Finally SAT solver is used to solve the resulting formula.

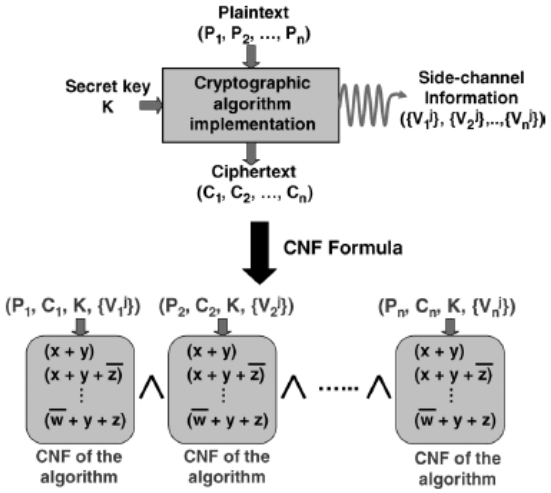

Fig.5. SAT formulation for side channel cryptanalysis

The implementation of a cryptographic algorithm having a side channel that leaks values of intermediate values is shown in Fig.5. The plain text P_i is mapped to cipher text C_i, for the secret key K_i. $\{V_i^j\}$ represents the value of k intermediate variables leaked when the implementation transforms P_i to C_i. Let $\Psi(P,C,K,V^1,V^2,...V^m)$ represents literals corresponding to plain text, cipher text, secret key and all the m internal variables respectively. Given n known plain text/cipher text pairs (P1,C1), (P2,C2),...(Pn, Cn), and intermediate variable values leaked by the side channel for each pair, $\{V_1^j, V_2^j, ... V_n^j\}$ for the same secret key K, we can generate the formula $\Psi(P_1,C_1,K_1,\{V_1^j\},\{V_1^j\}^c) \square \Psi(P_2,C_2,K_2,\{V_2^j\},\{V_2^j\}^c) \square .....\Psi(P_n,C_n,K_n,\{V_n^j\},\{V_n^j\}^c)$. $\{V_n^j\}^c$ is the set of intermediate variables other than V_n^j which remains unassigned [2]. The concatenated formula is input to a SAT solver which terminates with either satisfiable or unsatisfiable output.

IV. Experimental Results

In this section we are presenting the details of our experiments and the results. We have conducted our experiments on 8085 microprocessor trainer kit. Fig. 6 shows the current trace of microprocessor while performing addition of two numbers.

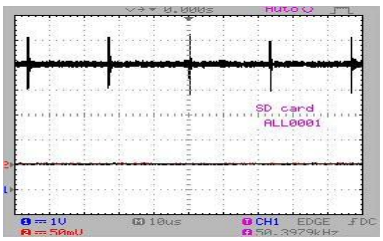

Fig.6. Output waveform while adding two numbers

A small value resistor is connected between the global supply and the controller supply pin across which the current profile is measured. This trace is shown in Fig. 6. Probe with a high input resistance and low input capacitance is use for measurement. Further statistical analysis is required to identify the instruction wise operation which can be utilised for power analysis attacks.

V. Conclusion

A workable space is attributed to the innovative performance of the side channel attack in the realm of today's challenging software technique. The lurking dangers of the above discussed software side channels in their specific locations are also demonstrated through the thrust given to the secret keys in such software. Also an automated SAT based framework for exploiting the vulnerabilities of the software side channel is explained. We have analysed the current traces of microprocessor while performing various basic operations, to differentiate the instruction wise performance at higher magnification. But the vital part of analysis rests on analysing the power traces of smart cards while performing encryption or decryption operations. The phenomenal part of the research area is focussed on the modification of SAT solvers to enable them to find the secret keys. Though the principle behind our research work has begun to come alive, the structural edifice is still an ongoing process awaiting may be many surprises and challenges.

References

[1]. E. Biham and A. Shamir, "Differential cryptanalysis of the full 16- round DES," in *Proc. Crypto '92*, Aug. 1992, pp. 487– 496.

[2]. Nachiketh R. Potlapally, Anand Raghunathan, Srivaths Ravi, Niraj K. Jha and Ruby B. Lee, "Aiding Side-Channel Attacks on Cryptographic Software With Satisfiability-Based Analysis", in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, April 2007, pp. 465-470.

[3]. P. Kocher, "Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems," in *Proc. Crypto '96*, Aug. 1996, pp. 104–113.

[4]. David Brumley and Dan Boneh, "Remote Timing Attacks are Practical", Proceedings of the 12th conference on USENIX Security Symposium - Volume 12 Washington, DC, 2003, pp. 1-14.

[5]. P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Proc.Crypto '99*, Aug. 1999, pp. 388–397.

[6]. T. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smartcard security under the threat of power analysis attacks," *IEEE Trans. Computers*, vol. 51, May 2002, pp. 541–552.

[7]. Thomas S. Messerges, Ezzy A. Dabbish and Robert H. Sloan, " Investigations of power analysis attacks on smartcards"**,** in *Proc. of the USENIX Workshop on Smartcard Technology on USENIX Workshop on Smartcard Technology*, 1999, pp. 17.

[8]. Dakshi Agrawal, Bruce Archambeault, Josyula R. Rao and Pankaj Rohatgi, "The EM Side-Channel(s)", in *4th International Workshop on Cryptographic Hardware and Embedded Systems*, 2002, pp. 29-45,.

[9]. W. van Eck, "Electromagnetic radiation from video display units: An eavesdropping risk," *Computers and Security*, vol. 4, 1985, pp. 269–288,.

[10]. D. Boneh, R. A. Demillo, and R. J. Lipton, "On the importance of checking cryptographic protocols for faults," in *Proc. Eurocrypt '97*, May 1997, pp. 37–51.

[11]. J. Kelsey, B. Schneier, D. Wagner, and C. Hall, "Side channel cryptanalysis of product ciphers," *J. Computer Security*, vol. 8, no. 2, 2000, pp. 141–158.

[12]. L. Benini, A. Macii, E. Macii, E. Omerbegovic, M. Poncino, andF. Pro, "A novel architecture for power maskable arithmetic units," in *Proc. Great Lakes Symp. VLSI*, Apr. 2003, pp. 136–140.

[13]. K. Tiri and I. Verbauwhede, "Securing encryption algorithms against DPA at the logic level: Next generation smart card technology," in *Proc. Cryptographic Hardware and Embedded Systems*, 2003, pp. 125–136.

[14]. Tiri K and Verbauwhede. I, "A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation", in *Proc. of Design, Automation and Test in Europe Conference and Exhibition*, Volume:1, Feb. 2004, pp. 246- 251.

[15]. J. S. Coron, D. Naccache, and P. Kocher, "Statistics and information leakage," *ACM Trans. Embedded Comput. Systems*, vol. 3, Aug. 2004, pp. 492–508.

[16]. J. Whittaker and H. H. Thompson, "Security bugs exposed: A systematic approach to uncovering software vulnerabilities," *Software Testing and Quality Engineering*, Mar. 2003, pp. 28–32,.

[17]. J. Chow, B. Pfaff, T. Garfinkel, K. Christopher, and M. Rosenblum, "Understanding data lifetime via whole system simulation," in *Proc. USENIX Security Symp.*, Aug. 2004, pp. 321–336,.

[18]. Safenet Inc., *Better Hardware Security Modules Through Better Design*. http://www.safenetinc.com/library/8/HSM Design principles.pdf, 2002.

[19]. L. Zhang and S. Malik, "The quest for efficient Boolean satisfiability solvers," in *Proc. Int. Conf. Computer-Aided Verif.*, July 2002, pp. 17–36,.

[20]. N. Een and N. Sorenson, "An extensible SAT solver," in *Proc. Int. Conf. Theory & Appl. Satisfiability Testing*, May 2003.

[21]. F. Massacci and L. Marraro, "Logical cryptanalysis as a sat problem," *J. Automated Reasoning*, vol. 24, Feb. 2000, pp. 165–203.

[22]. J. Viega, *Protecting Sensitive Data in Memory*. http://www.106.ibm.com/ developerworks/security /library/,2001.

[23]. S. Garfinkel and A. Shelat, "Remembrance of data passed," *IEEE Security and Privacy*, vol. 1, Feb. 2003, pp. 17–27.

[24]. P. Broadwell, M. Harren, and N. Sastry, "Scrash: A system for generating secure crash information," in *Proc. USENIX Security Symp.*, Aug. 2003.

[25]. V. Paretsky, *The Role of Hardware in Exposing Security Breaches*. http://www.ddj.com/print, 2005.

[26]. C. Percival, *Cache Missing for Fun and Profit*. http://www.daemonology.net/ papers/htt.pdf, 2005.

[27]. R. J. Anderson and M. G. Kuhn, "Tamper resistance - A cautionary note," in *Proc. USENIX Wkshp. Electronic Commerce*, Nov. 1996, pp. 1–11.

[28]. D. Farmer and W. Venema, *The Coroner's Toolkit*http://www.porcupine.org/ forensic cs /tct.html, 2005.

[29]. J. Whittaker, "Why secure applications are difficult to write," *IEEE Security and Privacy*, vol. 1, Mar. 2003, pp. 81–83.

[30]. Manfred Aigner and Elisabeth Oswald, Power Analysis Tutorial: available online: http://www.iaik.tugraz.at/content/research/implementation_attacks/introduction_to_impa/dpa_tutorial.pdf.