Security Considerations for Conducted and Radiated Emissions

Jason F. Reid*, Bouchra Senadji**, and Tee Tang**

*Information Security Institute, **School of Engineering Systems, Queensland University of Technology, Brisbane, QLD 4000, Australia

ABSTRACT : Useful information may be leaked by digital electronic circuits in their conducted and radiated electromagnetic emissions. The leakage signals can be captured and analysed using side-channel analysis techniques. This paper describes a current research project to explore the application of statistical analysis to measured electromagnetic emissions from a smart card system. Experimental results showed that logic state switching transient currents contain information related to the hamming weights of processed data. A differential side channel analysis (DSCA) technique was applied and it demonstrated that good correlation could be obtained when a correct sub-key was encountered.

Index Terms : electromagnetic emissions, switching transients, data security, side channel analysis,

I. INTRODUCTION

The field and discipline of electromagnetic interference/electromagnetic compatibility (EMI/EMC) is concerned with ensuring that a device 'performs' adequately in its intended environment, by neither interfering with the correct operation of other electronic devices, nor being interfered with by those other devices or itself. The usual notion of 'performance' focuses on meeting regulatory standards and avoiding functional failures. In this paper, we seek to raise awareness of another important dimension of 'performance' that also deserves consideration from an EMC perspective. This is the dimension of emissions security, also known in defence and military circles as TEMPEST. With the increasing adoption in the broader community of smart cards, RFID tags and wirelessly networked electronic devices, emissions security is no longer an esoteric concern that can be ignored. Each of these types of devices relies on cryptography to authenticate and secure communications over unprotected channels, for example, the electromagnetic spectrum or the Internet. Moreover, cryptography assumes that secret *keys* can be kept secret both when they are being stored and when they are being used in the confines of an integrated circuit, processor or embedded system. Unfortunately, this assumption is not well founded because semiconductor based processor logic leaks information about its internal state (including any cryptographic key it may be using) via the *side channels* of conducted and radiated electromagnetic emissions.

In this paper we explain the basic principles and techniques of *Side Channel Analysis*, in the course of reporting on a current research project that is exploring the application of EMC-based signal collection, processing and statistical analysis techniques to unearth useful information from conducted and radiated electrical noise.

II. CMOS AND POWER CONSUMPTION

The CMOS logic gate is the basic building block of most integrated circuits and processors. Although there is only a negligible current leakage when a CMOS logic gate maintains the same logical state across a clock transition, it dissipates a comparatively large amount of power when its logical state switches from 0 to 1 or 1 to 0. As a consequence of this behaviour, it is possible to coarsely model the instantaneous power consumption of a CMOS circuit as the sum of the power dissipated by each transitioning gate plus a Gaussian noise component. This is of great practical interest from a security perspective because processor instructions that directly manipulate data, (e.g. MOV, XOR) leak information about the data in the amount of power that the instruction consumes. For example, a MOV instruction with an operand of all zero bits consumes less power than one where all bits of the operand are set to ones. In fact, it is possible to 'read' the hamming weight of the operand (the number of bits in the byte that are set to one) directly from the trace of the power consumption at the time the instruction is executed. Eight different operand hamming weights can be clearly seen in Figure 1 as eight distinct peaks of increasing amplitude. Consider that the operand may be a cryptographic key that is assumed to be securely protected by the processor. By monitoring the device's power consumption, an attacker can discover important information about the key, possibly enough to violate the assumption of secrecy.

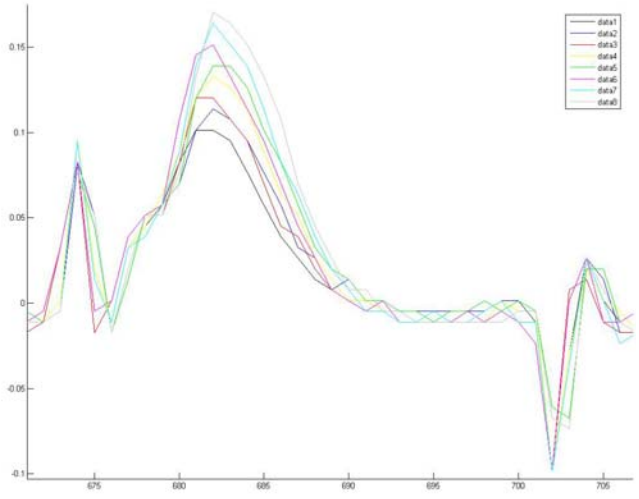


Figure 1: Power consumption traces of MOV instruction with operand hamming weights from 1 to 8

In addition to conducted emissions, a circuit also produces radiated emissions whose features are correlated, both directly and more subtly, to the instantaneous power consumption and therefore, to the data that the processor is manipulating. This is a simple consequence of the observation that a changing current flowing through a wire creates an electromagnetic field and the nature of the current can be inferred by monitoring the field. Since an integrated circuit is largely comprised of wires and current loops, each contributes its part to the complex radiated emission field. Useful and detailed information about the internal state of the device can be inferred by monitoring its electromagnetic emissions. Moreover, coupling effects due to parasitic capacitance between wires in an integrated circuit can result in the logical data they carry being modulated on strong carriers such as the higher order harmonics of the clock frequency [1]. The information can be recovered by tuning a sensitive wide band demodulator to these carrier frequencies. Our research project is exploring the nature of this electromagnetic information leakage and techniques for its capture and interpretation.

In constrained devices such as smart cards and RFID tags, effective electromagnetic shielding remains elusive so radiated side channel leakage is a serious issue. This is compounded by the fact that electromagnetic emissions can be monitored at a distance, leading to plausible, real-world attack scenarios. The electromagnetic side channel appears to be more difficult to sanitize than the conducted side channel whose information content can be reduced via filtering.

III. MEASUREMENT SETUP

In order to measure the power consumption, a readily available smart card programmer [2] was modified to include a current sensing resistor in the ground return of the onboard card holder. With a smart card inserted, the signals across the resistor were captured with a digital oscilloscope operating at a sampling rate of 200 MS/s. The digitised signals were transferred to a PC for further analysis. A typical experimental setup is shown in Figure 2a.

A number of algorithms were programmed into the smart card and current waveforms associated with each algorithm were captured. Figure 2b shows a typical waveform recorded.

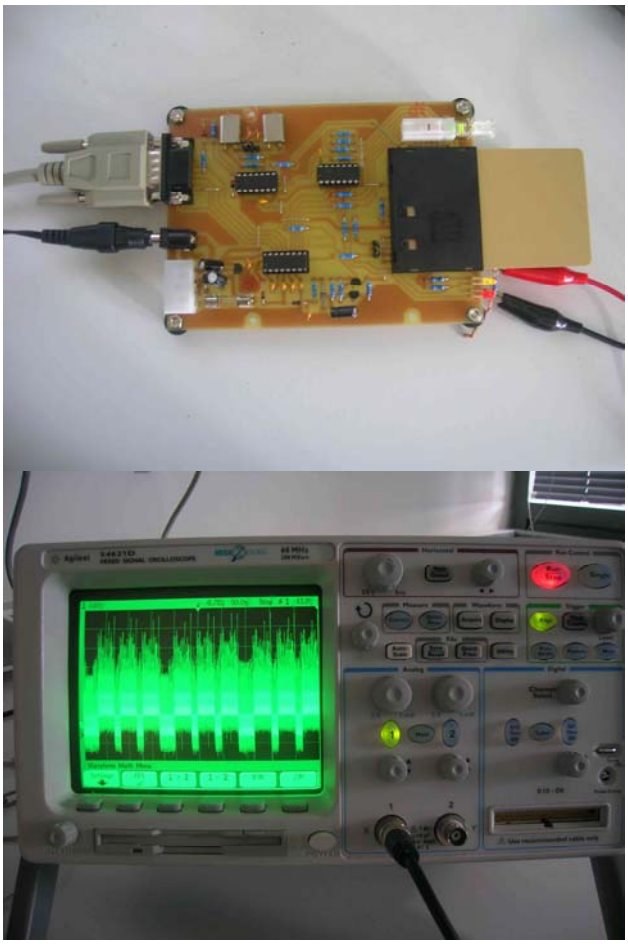


Figure 2a: Experimental Setup showing PIC16F84 smart card, modified reader and oscilloscope with test power trace.

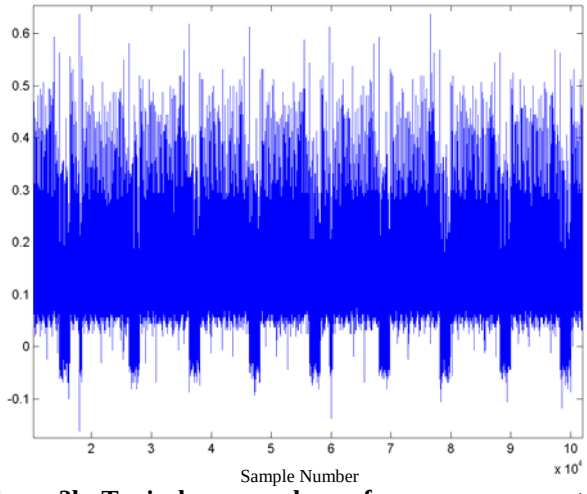


Figure 2b: Typical measured waveform across current sensing resistor

IV. DIFFERENTIAL SIDE CHANNEL ANALYSIS

In Section II we described information leakages that could be directly inferred by monitoring a single execution of a vulnerable instruction. By way of example, we showed that the hamming weight of the operand of a MOV instruction could be read from its corresponding power trace. This is known as *simple side channel analysis*, i.e., when useful information is inferred from the analysis of a single trace of a side channel. While knowledge of the hamming weight reduces the number of possible values an unknown key can take, it does not disclose the actual value since the positions of the set bits remains unknown. The technique of differential side channel analysis (DSCA) can address this problem. Differential side channel analysis, first introduced by Kocher et. al., [3] is a very powerful analysis method that can reveal actual bit values and their position, based on analysis of side channel traces gathered from multiple executions of an algorithm using the same secret key but different input data.

In our project, we are exploring a different technique to the more widely reported 'difference of means' approach introduced in [3]. We are drawing on ideas outlined in Mangard's recent thesis [4], to attack an implementation of the DES algorithm [5] implemented on a PIC16F84 based smart card. While we make no claims as to the novelty of the approach, to the best of the authors' knowledge, this is the first time that the approach has been reported in the context of attacking the DES algorithm.

We now provide an informal description of the general approach. In the algorithm being attacked, there must be an instruction that combines a few bits of the key with known input data to produce some intermediate result. This instruction should exhibit a data dependent side channel leakage. In our case, we exploit the fact that the hamming weight of the result of an XOR operation leaks in the power consumption trace. Furthermore, in the final iteration of the DES round function, the round sub key is XORed with a data block (see Figure 3) whose value can be directly calculated from the output (cipher text), so the attack preconditions are met.

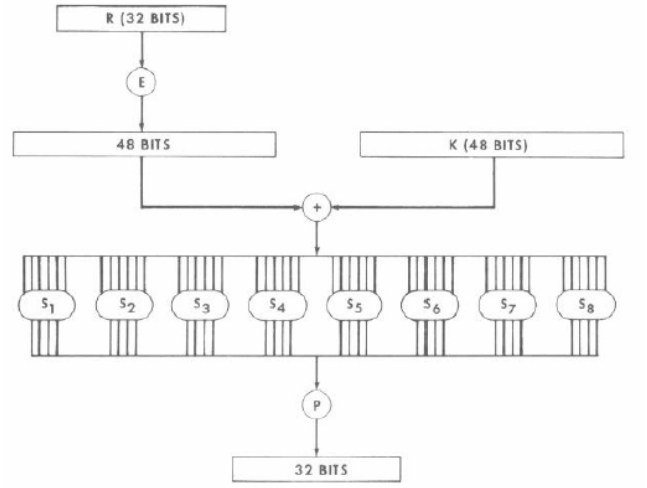


Figure 3: The DES round function showing the targeted XOR (+) of round key (K) and data. Source[5].

We record discrete time/power traces for the algorithm as it processes approximately three hundred different plaintext inputs with the same key. The corresponding cipher text outputs are also recorded. This produces a matrix where each row holds the time series of discrete power sample values for the algorithm as it processes a single cipher text, and each column holds the instantaneous power consumption at time *t*. The traces must be carefully aligned so that power consumption data points in each column correspond to the same algorithmic 'event' across each of the 300 cipher texts. An algorithmic event is represented by a column or vector of 300 discrete power values. One of these 'event vectors' (actually a few due to the high sampling rate, but for simplicity we will refer to one) will correspond to the targeted, vulnerable XOR operation. This concept is illustrated in Figure 4.

For each cipher text we reverse the final few steps of the algorithm to calculate the data block (R32) that was XORed with the 48 bits of the final round sub key. Due to the structure of the DES algorithm, we do not need to know the correct key to be able to do this. Since we are attacking the sub key 8 bits at a time, we calculate the hamming weight of the result of the XOR combination of the relevant 8 bits of the known data block with each of the 256 possible values of the sub key byte. We populate a matrix of 300 rows, each corresponding to a cipher text, and 256 columns, each corresponding to a possible value of the sub key. The data values in a column contain the calculated hamming weights of the intermediate XOR result for a single key guess across each of the 300 cipher texts. A column can be thought of as a

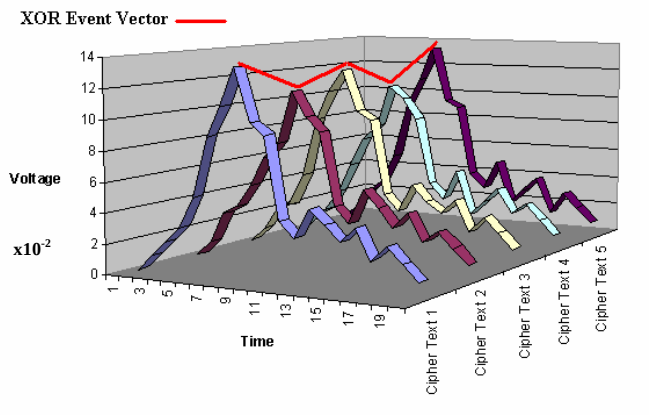
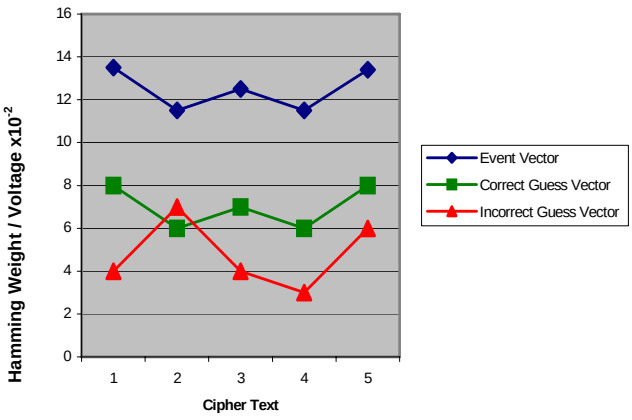


Figure 4: Instantaneous Power consumption at time *t* across multiple traces represents an 'event vector'.

vector of hypothetical power consumption values for one 'guess' of the key. We take each of these 256 'guess' vectors in turn and calculate its correlation with each of the *t* event vectors described in the previous paragraph. As we noted, one of the event vectors will correspond to the targeted XOR event. When the guess vector for the correct key is processed against the relevant XOR event vector, there will be a high degree of correlation (see Figure 6) because the power consumption of the XOR instruction (represented by the event vector) is a function of the hamming weight of the XOR result (represented by the guess vector). More simply, the XOR event vector and *correct* guess vector comprise pairs of data points which each 'swing in the same direction'. Incorrect guess vectors will not swing in the same direction across all pairs and will therefore not correlate to the same degree. This concept is illustrated in Figure 5. The high correlation reveals the correct sub key byte. We recalculate the guess vector matrix for the next sub key byte and repeat the process.

Note that we do not need to know where in a trace of *t* power values the XOR event occurs. The knowledge that somewhere there is a calculation of an intermediate result that directly combines key bits and known input bits via an instruction that leaks information about its operands is enough.



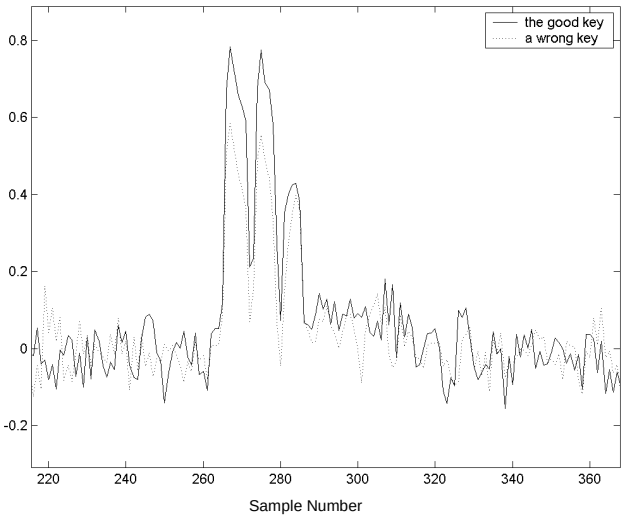


Figure 6: Correlation of correct and incorrect guess vectors. The 0.8 peak signifies the correct sub key.

V. CONCLUSIONS AND FUTURE WORK

Conducted and radiated electromagnetic emissions from a digital circuit contain useful information which can be captured and analysed. By using simple power measurement and analysis techniques, it has been demonstrated in this paper that encryption keys could be readily detected. This work will be extended in the future to retrieve information from captured radiated electromagnetic fields.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the contributions of Benjamin David, Jason Kong and Vania Sidharta who are undergraduate student members of the project research team. We also acknowledge the support of the Information Security Institute – QUT and the School of Engineering Systems.

REFERENCES

1. 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*, volume 2523 of *Lecture Notes In Computer Science*, pp.29-45, Springer, 2002.
2. Jaycar Electronics, *Engineering Catalogue*, p.25, 2005.
3. Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential Power Analysis. In *Advances in Cryptology—CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science*, pp.388–397, Springer, 1999.
4. Stefan Mangard. *Securing Implementations of Block Ciphers against Side-Channel Attacks*. PhD Thesis, Institute for Applied Information Processing and Communications (IAIK), Graz University of Technology, June 2004.

5. National Institute of Standards and Technology (US), Data Encryption Standard (DES), *Federal Information Processing Standards*, FIPS-PUB 46-3, 1999.

vectors