MOLES: Malicious Off-Chip Leakage Enabled by Side-Channels

Lang Lin*, Wayne Burleson*, Christof Paar* §

*Department of Electrical and Computer Engineering, University of Massachusetts Amherst, USA

§Horst Görtz Institute for IT Security Ruhr University Bochum, Germany

ABSTRACT

Economic incentives have driven the semiconductor industry to separate design from fabrication in recent years. This trend leads to potential vulnerabilities from untrusted circuit foundries to covertly implant malicious hardware Trojans into a genuine design. Hardware Trojans provide back doors for on-chip manipulation, or leak secret information off-chip once the compromised IC is deployed in the field. This paper explores the design space of hardware Trojans and proposes a novel technique, "Malicious Off-chip Leakage Enabled by Side-channels" (MOLES), which employs power side-channels to convey secret information off-chip. An experimental MOLES circuit is designed with fewer than 50 gates and is embedded into an Advanced Encryption Standard (AES) cryptographic circuit in a predictive 45nm CMOS technology model. Engineered by a spread-spectrum technique, the MOLES technique is capable of leaking multi-bit information below the noise power level of the host IC to evade evaluators' detections. In addition, a generalized methodology for a class of MOLES circuits and design verification by statistical correlation analysis are presented. The goal of this work is to demonstrate the potential threats of MOLES on embedded system security. Nevertheless, MOLES could be constructively used for hardware authentication, fingerprinting and IP protection.

1. INTRODUCTION

Hardware security modules (HSMs) are special-purpose crypto-processors that execute cryptographic algorithms and store secret keys for embedded systems or general-purpose computers. They provide both logical and physical protections against unauthorized key tampering and various malicious attacks. However, the general trend of separating design from fabrication in the semiconductor industry causes the vulnerabilities of HSMs to untrusted integrated circuit (IC) foundries [1]. Malicious foundries can covertly embed hardware Trojans into the HSMs during the IC fabrication process to leak secret information or even completely destroy a crypto module [2].

Hardware Trojans can be categorized into a functional class and a parametric class, both of which have a large design space of size, structure and distribution methods [3]. Recently, the first exploration of the design space of hardware Trojans in [4] proposes a malicious core embedded into a general-purpose microprocessor to implement two Trojan mechanisms. These Trojans are difficult to detect by traditional Automatic Test Pattern Generation (ATPG) tests and IC layout inspections [5], because they only introduce less than 0.01% additional gate count. Some recently proposed function tests and fault analysis methods have detected the unusual behaviors of similar functional Trojans [6, 7]. However, it is still very challenging to detect more advanced Trojans with rare trigger patterns [8].

Side-channels are the inherent physical properties of a running IC, including timing, power consumption, electromagnetic radiation and even sound wave. Attacks based on side-channel information can impair the cryptographic routines of various embedded cryptosystems [9, 10, 11, 12]. General countermeasures against side-channel attacks include using selectively re-sized transistors [13], non-standard gate libraries [14] and multi-core architectures [15]. In the context of hardware Trojans, side-channel analysis is also proposed for positive use of Trojan detections. A key work in [16] uses the side-channel information as a physical fingerprint to distinguish the compromised ICs from the genuine ones. Other Trojan detection schemes based on power side-channels and path delay profiles are described in [17, 18].

Using side-channels as building blocks to implement hardware Trojans is a novel and promising concept recently proposed by us [19] and others [20]. Unlike other Trojan mechanisms, information leakage conveyed by side-channels is not direct digital information, but composed of analog signals that must be interpreted through advanced off-chip signal analyses. In this paper, we elaborate this concept through the design of "Malicious Off-chip Leakage Enabled by Side-channels" (MOLES). MOLES employs lightweight custom circuits to create additional side-channels that can completely compromise HSMs by leaking secret cryptographic keys. Inspired by spread-spectrum techniques in digital communications, we engineer MOLES to communicate with potential attackers below the noise power level of the compromised IC. Also, we generalize the design methodology of MOLES circuits and propose a verification process to determine the required efforts for multi-bit key extractions under different noise levels. By adopting the lightweight implementation and hidden communication techniques, we expect that no single Trojan detection scheme existing in the literature can detect MOLES. This paper is organized as follows: Section 2 introduces the basic concepts and detection theories. Section 3 generalizes the design method and verification process of MOLES implementations. Section 4 demonstrates the effectiveness of an experimental MOLES, and Section 5 draws conclusions.

2. BASICS OF MOLES

The threat model of an untrusted semiconductor manufacturing and supply chain involves two parties. The attacker is the malicious party who designs and covertly embeds hardware Trojans into the host IC to leak secret information. More importantly, only the attacker knows how to extract the information leakage from the compromised IC deployed in the field. The evaluator is the party who endeavors to detect hardware Trojans through standard IC tests and security evaluations.

Throughout the rest of this paper, we will play the role of attackers to design hardware Trojans through MOLES techniques.

To covertly hide in a compromised IC and evade most traditional IC function tests and hardware Trojan detection schemes, MOLES should be able to fulfill several design goals. First, the implementation of MOLES should use the minimum gate count to evade the IC mask inspections. Modern ICs contain large blocks of unused circuitry, which may be left from previous versions of the design or used for temporary testing purposes. These useless blocks may be ignored by standard IC inspections due to a high testing cost. MOLES could thus be distributed on such unused areas across the die, as long as it is small enough (e.g., implemented with 0.01% additional gate count) to fit in. Second, to survive both the time-domain and frequency-domain IC function tests, MOLES should not disturb the original functionality and I/O behaviors of the host IC and should synchronize with the IC global clock. Finally, MOLES techniques employ side-channels to convey information leakage. However, they should also incorporate a mechanism to evade the evaluator's detection scheme of side-channel analyses. Although both attackers and evaluators can access to the side-channel information, we will demonstrate later that MOLES can be designed with spread-spectrum techniques to enable only attackers' exploitability on side-channel leakage.

Note that the concept of MOLES to intentionally induce side-channels can also be used constructively to enhance hardware security. For instance, IC manufacturing variability differentiating each chip can be harnessed as a physical uncloneable function (PUF) for chip identification [21, 22]. In similar ways, MOLES can act as a unique chip identification circuit because only the MOLES designer can exploit the chip identification information conveyed by side-channel leakage. If designed with fewer responsibilities to evade various detection schemes, MOLES could be more robust and secure than variability-based PUFs.

2.1 MOLES Implementation

A generic MOLES can be implemented by different side-channels, such as power consumption, electromagnetic radiation and path delay. In this work, we specifically engineer a MOLES circuit to consume data-dependent power as a power side-channel to leak multi-bit secret keys. A critical feature of MOLES is the signal-to-noise ratio (SNR), defined as the power level of side-channel leakage to that of the host IC. An effective MOLES requires a low SNR to evade evaluators' detection, but a high enough SNR for the attacker to extract the secret key bits through long observation time. To meet such ends, we use spread-spectrum techniques to distribute the power of side-channel leakage to multiple clock cycles. The SNR for each clock cycle is sufficiently low to evade evaluators' detection, while the attacker can still exploit the side-channel information by averaging over a large number of clock cycles. To implement such a technique, we modulate each key bit with a long pseudo-random number (PN) sequence by an XOR operation. As shown in Figure 1, a binary pseudo-random number generator (PRNG) can generate a PN sequence $r_{K0}(t)$. The multi-bit key bus is covertly hardwired to the XOR gates by the attacker (only the key bit K_0 is shown in the figure). The output node of each XOR gate, with no connection to any I/O pin, is connected to a capacitive load that leaks a small amount of power $P_{0->1}(t)$ when a 0→1 logic transition occurs. The size of the load capacitor is an adjustable design parameter for MOLES, which determines the amount of side-channel information leakage and thus the SNR.

For a spread-spectrum system in communications, SNR is also affected by the "process gain" (the ratio of the spread bandwidth to the unspread bandwidth). In the context of MOLES, the process gain is the ratio of the PN sequence duration period to the 0→1 transition period of the key. A larger process gain can help overcome the low SNR for demodulating the side-channel information. For the attacker exploiting side-channel leakage by measuring the power consumption of a running compromised IC, the PN sequence duration period is equal to the observation (measurement) time. The transition period of the key depends on the crypto algorithm performed by the compromised IC. Since this work demonstrates a MOLES circuit embedded in a symmetric-key crypto-processor, the secret key is fixed with constant transition period. Consequently, the process gain is linearly proportional to the attacker's observation time. This suggests that the attacker can measure numerous power traces for achieving a large SNR to extract the secret keys.

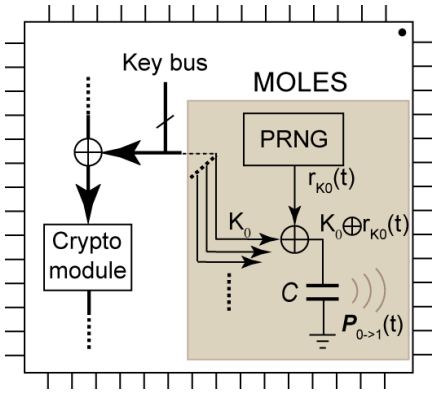

Figure 1: MOLES circuit embedded in a crypto-processor (circuit size not to scale)

In order to leak multiple key bits, we use the code division method, which is fundamental to many code division multiple access (CDMA) communication systems [23]. Essentially, we modulate each key bit with a different PN sequence so that the modulated power side-channels are orthogonal to each other. Then, multi-bit keys can be leaked simultaneously and each key bit can be extracted from the power traces by statistical correlation demodulation approaches, such as differential power analysis (DPA) [9]. Since the attacker chooses the PN sequences, no evaluator can demodulate the side-channel leakage without knowing the PN sequence corresponding to each key bit.

2.2 Detection Theory

The detection of side-channel information leakage and the extraction of multi-bit keys are based on the correlation demodulation theory. Assume that the attacker is able to measure the total power S(t) of a compromised IC with N clock cycles. N is also known as the number of power traces in DPA. Let us first consider the simple case of leaking one key bit. The attacker can make a key guess and classify the power traces into two groups (group 0 and group 1) based on the XOR result X_i of the key guess and the PN sequence. Then the attacker performs a traditional DPA to examine the correlation between the power traces and X_i. To render a better result, each power trace is integrated over the entire clock cycle before statistical analyses. Then the overall differential power is calculated by summing the integrated power values for all power traces in group 1 and subtracting the sum of those in group 0. Consequently, the correct

key guess will have a positive differential power value, while the wrong key guess will have a negative one.

In the exemplary MOLES shown in Figure 1, we assume that the secret key is the eight key bits K_0 to K_7 involved in an AES substitution box. The PRNG, implemented with a linear feedback shift register (LFSR), generates PN sequences that are synchronized with the clock of the AES (e.g., the clock period is T_c). The key bits K_0 to K_7 are XORed with the corresponding PN sequences of $r_{K0}(t)$ to $r_{K7}(t)$. The PN sequences are correlated by the following time shifts: $r_{K0}(nT_c) = r_{K1}((n-1)T_c) = \ldots = r_{K7}((n-7)T_c)$. The autocorrelation property of the LFSR guarantees the orthogonality of the eight PN sequences. Assume that the attacker measures the transient power consumption of the compromised AES with $N \cdot T_c$ time duration. The attacker-induced power component that leaks side-channel information of one key bit K_i is:

$$P_{K_i}(t) = \sum_{n=2}^{N} [K_i \oplus (r_{K_i}(n) - r_{K_i}(n-1))] \cdot P_{0\to 1}(t - nT_c), i = 0..7 \tag{1}$$

The total power of the compromised IC can be modeled as the sum of $P_{MOLES}(t)$ and $P_{noise}(t)$, thus the SNR is the ratio of $P_{MOLES}(t)$ to $P_{noise}(t)$:

$$\begin{aligned} S(t) &= P_{MOLES}(t) + P_{noise}(t) \\ &= \left[\sum_{i=0}^{7} P_{K_i}(t) + P_{PRNG}(t) \right] + \left[P_{AES}(t) + P_{AWGN}(t) \right] \end{aligned} \tag{2}$$

In the above equation, the power of the MOLES circuit is composed of the power of the LFSR-based PRNG and the eight XOR gates leaking side-channel information. Only the attacker can exploit the side-channel leakage by knowing the initial state and the structure of the LFSR. An evaluator even having the skill to perform side-channel analysis will take the side-channel information as noise power without knowing the implementation details of MOLES. Alternatively, the evaluator can make further efforts to conjecture the structure of LFSR (in order to detect the MOLES) by side-channel cryptanalysis. However, the attacker can react by using a non-linear LFSR (NLFSR) to defeat the evaluator's linear cryptanalysis. The design complexity of PRNG is another design space of MOLES, which is determined by the Trojan detection scheme of evaluators.

Both the power of AES crypto core and on-chip noise contribute to the "noise power" for MOLES. The on-chip noise power comes from power grid fluctuations, process variations, and thermal noise, which can be modeled as additive white Gaussian noise (AWGN). The noise power level significantly impacts the SNR and thus the detection of side-channel information leakage. The noise power profile could be much more complex than Gaussian noise if we also consider the power consumption of non-crypto components on the host IC. To deal with power side-channel analyses under different noise power profiles, advanced methods such as template attacks [24] can be applied, which are beyond the scope of this work.

The extraction of multi-bit keys can follow a bit-by-bit fashion. Since the SNR is very low in every clock cycle, the attack must observe enough power traces to exploit the side-channel information leakage. When extracting a single bit key from the power traces, the power side-channels of all other key bits become the noise power. However, this does not significantly affect the key extraction. The attacker can start by making a key guess K_0* of key bit K_0. Based on the key guess and the known PN sequence, the attacker can predict $X_0 = K_0^* \oplus [r_{K0}(nT_c) - r_{K0}((n-1)T_c)]$ for a certain time point nT_c. The power traces are then grouped based on whether X_0 is logic 1 or logic 0. After grouping N power traces, let us assume that m_0 power traces are in group 1 which is associated with the predicted logic 1, so N- m_0 power traces are in group 0. Then the differential power (DP) for the N power traces can be calculated by the mean of m_0 group 1 power traces minus the mean of N-m_0 group 0 power traces:

$$DP(N) = \frac{1}{m_0} \sum_{n \in grp1} \frac{1}{T_c} \int_{(n-1)T_c}^{nT_c} S(t)dt - \frac{1}{N - m_0} \sum_{n \in grp0} \frac{1}{T_c} \int_{(n-1)T_c}^{nT_c} S(t)dt \tag{3}$$

Since the PN sequences are pseudo-random binaries, we can assume that $m_0 \approx 0.5N$ when N is large enough. To derive Equation (3), the four power terms in Equation (2) have to be determined respectively. First of all, the correlation of $P_{PRNG}(t)$ and $P_{AWGN}(t)$ to the key bits is near zero, which will not contribute to the result. Secondly, the term $P_{AES}(t)$ could have some correlation with the key bits since the 8-bit keys are processed in the non-linear AES substitution box. However, HSM designers should have minimized this correlation by side-channel-resistant methods to avoid direct power analysis attacks on the AES crypto core. Besides, since the attacker groups the power traces by the PN sequences, the inherent power side-channel of AES is weak during the MOLES detection process.

Finally, for the first term in Equation (2), let us consider the single term $P_{K0}(t)$ and other 7 terms separately. For $P_{K0}(t)$, if the key is correctly guessed (i.e., $K_0 = K_0$*), exactly m_0 power traces in group 1 consume the power $P_{0\to1}(t)$, while the power traces in group 0 do not. As a result, DP(N) for a correctly guessed K_0 becomes:

$$\begin{aligned} &\frac{1}{m_0} \sum_{n \in grp1} \frac{1}{T_c} \int_{(n-1)T_c}^{nT_c} 1 \cdot P_{0\to1}(t)dt - \frac{1}{N - m_0} \sum_{n \in grp0} \frac{1}{T_c} \int_{(n-1)T_c}^{nT_c} 0 \cdot P_{0\to1}(t)dt \\ &= \frac{1}{T_c} \int_0^{T_c} P_{0\to1}(t)dt = \text{constant} > 0 \end{aligned} \tag{4}$$

On the other hand, if the key is wrongly guessed, the grouping is exactly opposite. As a result, DP(N) for a wrongly guessed K_0 becomes:

$$\begin{aligned} &\frac{1}{N - m_0} \sum_{n \in grp0} \frac{1}{T_c} \int_{(n-1)T_c}^{nT_c} 1 \cdot P_{0\to1}(t)dt - \frac{1}{m_0} \sum_{n \in grp1} \frac{1}{T_c} \int_{(n-1)T_c}^{nT_c} 0 \cdot P_{0\to1}(t)dt \\ &= -\frac{1}{T_c} \int_0^{T_c} P_{0\to1}(t)dt = -\text{constant} < 0 \end{aligned} \tag{5}$$

No matter how K_0 is guessed, each term in $\sum_{i=1}^{7} P_{K_i}(t)$ gives a result to DP(N) by:

$$\begin{aligned} &\frac{1}{M} \sum_{n \in grp1} \frac{1}{T_c} \int_{(n-1)T_c}^{nT_c} [K_i \oplus (r_{K_i}(n) - r_{K_i}(n-1))] P_{0\to1}(t)dt - \\ &\frac{1}{N - M} \sum_{n \in grp0} \frac{1}{T_c} \int_{(n-1)T_c}^{nT_c} [K_i \oplus (r_{K_i}(n) - r_{K_i}(n-1))] P_{0\to1}(t)dt \end{aligned} \tag{6}$$

Since the grouping for K_0 is non-correlated with $K_i^* \oplus [r_{Ki}(nT_c)-r_{Ki}((n-1)T_c)]$, there will be m_i power traces assigned to group 1 instead of m_0. Thus, the resulting DP(N) becomes:

$$\frac{m_i}{m_0}\cdot\frac{1}{T_c}\int_0^{T_c} P_{0\to1}(t)dt - \frac{N-m_i}{N-m_0}\cdot\frac{1}{T_c}\int_0^{T_c} P_{0\to1}(t)dt$$
$$= \frac{N(m_i-m_0)}{m_0(N-m_0)}\cdot\text{constant} = \frac{m_i/m_0-1}{1-m_0/N}\cdot\text{constant} \quad (7)$$

When N is large enough, the above result is close to 0 because both m_0 and m_i will be sufficiently close to 0.5 N.

To sum up, we will get DP (N) > 0 if the key bit K_0 is correctly guessed. Otherwise, we will get a resulting DP (N) less or equal to zero. Similarly, other key bits can be extracted as long as the power traces are grouped by the correct key guess. The increase of key bits will not affect the extraction of a single key bit, due to the orthogonality of the PN sequences to modulate each key bit.

3. EXPERIMENTAL MOLES DESIGN

MOLES circuits can be implemented and optimized through a custom IC design flow. Then they can be incorporated into the host IC design in circuit simulation tools to generate the power traces of the compromised IC. Next, a design verification process is required to analyze the simulated power traces of the compromised IC under appropriate noise power models. After confirming the effectiveness of key extraction, the attacker can finally implant MOLES circuits into the genuine IC. In a real side-channel analysis, the attacker actually performs a similar verification process through the measurement of real transient power traces and off-chip statistical analyses for information extractions. A generalized design methodology is described below.

At the first design stage, several design spaces of MOLES should be determined, such as the size of load capacitance, the type of PRNG and the number of key bits to leak. Then the transistor-level netlist of custom MOLES circuit is implemented with realistic device models. After embedding MOLES into a low-level netlist of the host IC, the power traces of the compromised IC are simulated by HSPICE to make a power profile. Pragmatically, the designer should be ready to simulate at least 10000s power traces, so that the SNR is low enough to evade a simple side-channel analysis detection by the evaluator.

The verification process to extract the secret key can be performed by signal processing tools, such as MATLAB. To address the on-chip noise impacts on MOLES circuits, different noise power profiles can be also generated. The verification process can adjust the power level of both the compromised IC power profile and noise power profile to achieve a realistic on-chip SNR. The two power profiles are added up as a final simulated measurement power (SMP) profile. This SMP profile is then executed by a statistical correlation demodulation engine to verify a key extraction. If the key cannot be extracted, the designer should simulate more power traces at the simulation stage to increase the process gain. If the key still cannot be extracted, the MOLES circuits should be modified at the earliest design stage to convey more side-channel leakage. The entire design and verification process follows a heuristic approach.

In this work, we implement an experimental MOLES and an AES crypto core with the 45nm predictive technology model (PTM) [25]. We synchronize the clock of MOLES and AES to be 100 MHz. We simulate the power traces of the compromised IC with 10 sampling points in each clock cycle. Since a typical SPICE simulation with accurate power models for a large circuit can take days, we only simulate 20,000 power traces as a starting point.

As shown in Figure 2, the PRNG in MOLES is implemented by an LFSR of degree 20, with a maximum length primitive polynomial of $x^{20}+x^{13}+x^9+x^5+1$. Eight XOR gates are employed to leak 8-bit keys. By applying the ring generator architecture [26] for the LFSR, the design only contains 49 equivalent gates.

We initially set the load capacitance of each XOR gate as 0.1 pF to leak as much side-channel information as possible. To validate the consistence of the ideal side-channel leakage model and the power with the realistic device model, we simulate the power traces of the MOLES circuit alone, as shown in Figure 3. For simplicity, we only use two XOR gates to leak two key bits. The corresponding bit sequences of the two XORs (X0 and X1) are shown on top of the 15 power traces. We can see that the power traces during $5T_c$ to $7T_c$, $8T_c$ to $10T_c$ and $11T_c$ to $13T_c$ are almost consistent with the side-channel leakage model: a small amount of power is consumed when a 0→1 logic transition occurs. Note that the inconsistent power is caused by the uncorrelated power consumed by the registers of the LFSR circuit.

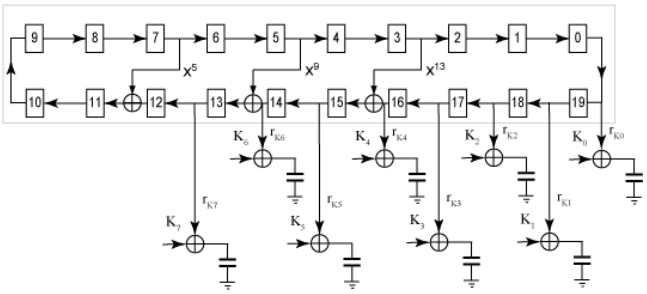

Figure 2: Diagram of an experimental MOLES

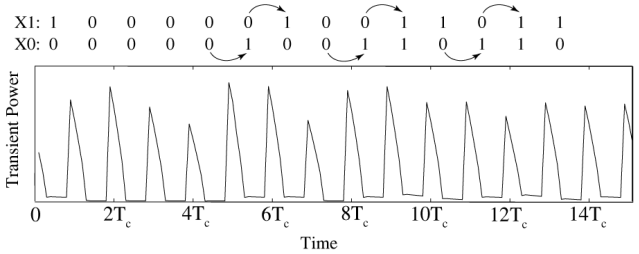

Figure 3: Consistence of the side-channel leakage model and the simulated power traces

In the verification process, we perform two verification phases to demonstrate the key extraction and study the SNR impacts on the required number of power traces (RPT) :

1) Verification of key independency: the key extraction should be independent on key value. In this phase, we configure the MOLES to leak only 2 key bits and simulate 4 sets of power profiles by setting the secret key as all possible values (i.e., 00, 01, 10 and 11). Then we try to extract the secret key for all four cases. For each case, we first set SNR = -10 dB to examine the RPT to extract both the two key bits. Then we modify the SMP profile by linearly (in dB) decreasing the SNR to determine the relation between SNR and RPT.

2) Verification of design scalability: the key extraction should be effective with the increase of key bits (at least 8-bit key for an

AES substitution box). In this phase, we configure the MOLES to leak 8 key bits by setting an arbitrary key of 01010110. Then we study the RPT with SNR variations in the same way as before.

4. RESULTS

The results of the verification phase 1 are shown in Table 1. For different keys with a given SNR, the RPT varies slightly due to the uncertainty of noise power. Besides, Figure 4 illustrates the near inverse-linear relation between the log-scale SNR and the RPT. As a showcase, Figure 5 plots the differential power curves for key=00 at SNR = -20dB. With the increase of power traces, the differential power curves of the correct key guesses (represented by solid lines) gradually stand out from those of the wrong key guesses (represented by dash lines). We highlight the point 4204 (recognized as the RPT), when both the curves of K_0=0 and K_1=0 stand out above zero to indicate the correct key guess 00. Although the extraction of K_1 is slightly earlier than that of K_0, we count the RPT by the larger number of power traces.

Table 1: SNR vs. RPT for different keys

SNR(dB)	*Key = 00*	*Key = 01*	*Key = 10*	*Key = 11*
-10	497	303	760	801
-20	4204	3162	4893	5223
-30	8850	9751	10123	8533
-40	14432	17910	18920	16882

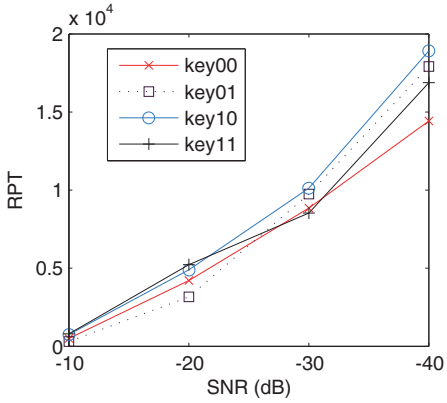

Figure 4: RPT as a function of SNR

The result of the verification phase 2 for the case of SNR=-20dB is shown in Figure 6. To extract the 8-bit key bit-by-bit, we plot 16 differential power curves. The eight correct-key-guess curves are represented by solid lines. The RPT is highlighted as 13000, where the correct key 01010110 is accurately extracted. Compared with the phase 1, the increase of key bits leads to an increasing RPT. The reason is that additional key bits can compromise the SNR for extracting a single key bit. Besides, we also verify the case of SNR = -10dB that results an RPT of 3200. For the case of SNR= -30dB, our initial 20000 power traces are not enough to extract all key bits. This indicates that the interpretation of side-channel leakage conveying a large key size (e.g., 256-bit key) under extremely low SNR requires signification computation efforts during the off-chip side-channel analyses.

5. CONCLUSIONS

In this work, we demonstrate a novel class of hardware Trojans, the MOLES, which can intentionally leak secret information through side-channels. We formulate the mechanism and detection methods of MOLES in theory. To emphasize the threats on embedded system security, we expose a wide design space of MOLES circuits and provide a verification process for multi-bit key extractions. Simulations of a compromised crypto core demonstrate the effectiveness of MOLES to leak secret keys under different noise power levels. By minimizing the gate count and applying spread-spectrum techniques, MOLES is very promising to evade most detection strategies, such as optical inspections, advanced function tests and physical fingerprinting analyses.

We are aware that the silicon implementation of MOLES techniques faces several critical issues, such as the process variation impacts on side-channels and the computation overhead of detection in low signal-to-noise ratio. However, these issues can be addressed through design optimizations or advanced signal processing methods, which will direct our future research.

ACKNOWLEDGMENTS

We thank Kevin Fu, the members of SPQR and the anonymous reviewers for laboratory assistance and constructive suggestions. This work was supported in part by the NSF Grant CNS-0627529.

6. REFERENCES

[1] High Performance Microchip Supply, annul report by Defense Science Board, http://www.acq.osd.mil/dsb/, 2005.

[2] S. Adee: The hunt for the kill switch. In: IEEE Spectrum, Vol. 45, Issue 5, pp. 34-39, 2008.

[3] X. Wang , M. Tehranipoor , J. Plusquellic: Detecting Malicious Inclusions in Secure Hardware: Challenges and Solutions. In: 1st IEEE International Workshop on Hardware-Oriented Security and Trust (HOST), pp. 15-19, 2008.

[4] S. T. King, J. Tucek, A. Cozzie, C. Grier, W. Jiang, Y. Zhou: Designing and implementing malicious hardware. In: Proceedings of the 1st USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET), pp. 1-8, 2008.

[5] J. M. Soden, R. E. Anderson, C. L. Henderson: IC Failure Analysis: Magic, Mystery, and Science. In: IEEE Design & Test of Computers, Vol. 14, pp. 59-69, 1997.

[6] M. Banga, M. S. Hsiao: A Region Based Approach for the Identification of Hardware Trojans. In: IEEE HOST, pp. 40-47, 2008.

[7] R. Chakraborty, S. Paul, S. Bhunia: On-Demand Transparency for Improving Hardware Trojan Detectability. In: IEEE HOST, pp. 48-50, 2008.

[8] Y. Jin, N. Kupp, Y. Makris: Experiences in hardware Trojan design and implementation. In: IEEE HOST, pp. 50-57, 2009.

[9] P. Kocher, J. Jaffe, B. Jun: Differential Power Analysis. In: Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO), LNCS 1666, pp. 388-397, 1999.

[10] P. Kocher: Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In: CRYPTO, LNCS 1109, pp. 104-113, 1996.

[11] A. Shamir, E. Tromer: Acoustic cryptanalysis. http://people.csail.mit.edu/tromer/acoustic/

[12] M. Hutter, S. Mangard, M. Feldhofer: Power and EM attacks on passive 13.56 MHz RFID devices, In: Workshop on Cryptographic Hardware and Embedded Systems (CHES), LNCS 4727, pp. 320-333, 2007.

[13] L. Lin, W. Burleson: Analysis and mitigation of process variation impacts on power-attack tolerance, In: Proceedings of ACM/IEEE Design Automation Conference (DAC), pp. 238-243, 2009.

[14] K. Tiri, I. Verbauwhede: A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation. In: Design, Automation and Test in Europe (DATE), pp. 246-251, 2004.

[15] J. Ambrose, S. Parameswaran, A. Ignjatovic: MUTE-AES: a multiprocessor architecture to prevent power analysis based side channel attack of the AES algorithm. In: ACM/IEEE International Conference on Computer-Aided Design (ICCAD), pp. 678-684, 2008.

[16] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, B. Sunar: Trojan Detection using IC Fingerprinting. In: IEEE Symposium on Security and Privacy, pp. 296-310, 2007.

[17] R. M. Rad, X. Wang, M. Tehranipoor, J. Plusquellic: Power supply signal calibration techniques for improving detection resolution to hardware Trojans. In: ACM/IEEE ICCAD, pp. 632-639, 2008.

[18] Y. Jin, Y. Makris: Hardware Trojan Detection Using Path Delay Fingerprint. In: IEEE HOST, pp. 51-57, 2008.

[19] L. Lin, M. Kasper, T. Güneysu, C. Paar, W. Burleson: Trojan side-channels: lightweight hardware Trojans through side-channel engineering. In: CHES, LNCS 5747, pp. 382-395, 2009.

[20] Y. Alkabani, F. Koushanfar: Extended Abstract: Designer's Hardware Trojan Horse, In: IEEE HOST, pp. 82-83, 2008.

[21] G. Suh, S. Devadas: Physical Unclonable Functions for Device Authentication and Secret Key Generation, In: ACM/IEEE DAC, pp. 9-14, 2007.

[22] D. Holcomb, W. Burleson, K. Fu: Initial SRAM state as a fingerprint and source of true random numbers for RFID tags, In: Proceedings of the Conference on RFID Security, 2007.

[23] J. Proakis: Digital communications, 4th edition, McGraw-Hill, 2000.

[24] S. Chari, J. R. Rao, P. Rohatgi: Template attacks, In: CHES, LNCS 2523, pp. 12-28, 2002.

[25] Predictive Technology Model (PTM), http://www.eas.asu.edu/~ptm/

[26] J. Rajski, J. Tyszer: Primitive polynomials over GF(2) of degree up to 660 with uniformly distributed coefficients. In: Journal of Electronic Testing: theory and applications, pp. 645-657, 2003.

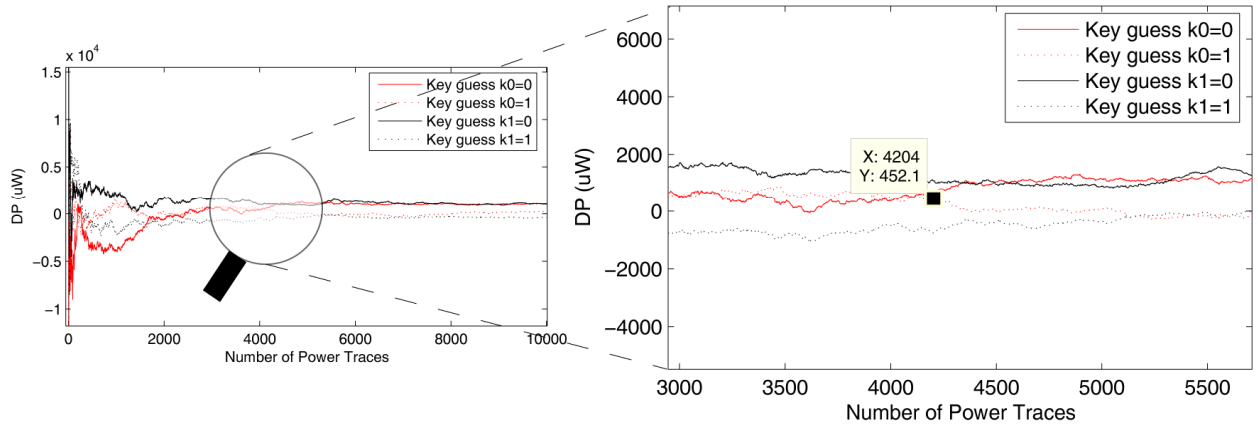

Figure 5: Differential power curves indicating the correct key 00 (SNR = -20dB)

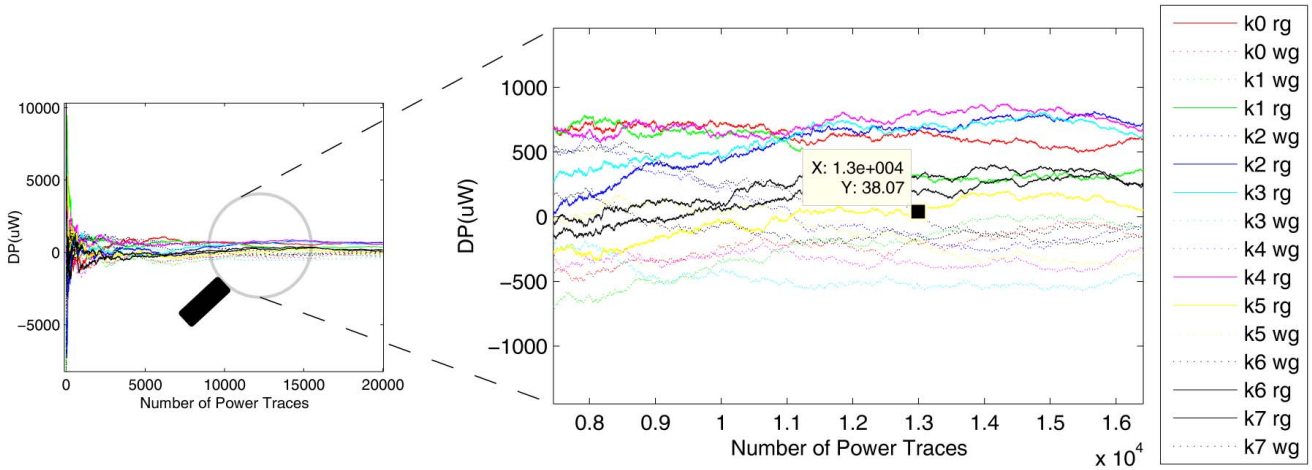

Figure 6: Differential power curves indicating the correct key 01010110 (SNR = -20dB)