A Novel Countermeasure to Resist Side Channel Attacks on FPGA Implementations

Y. Zafar and D. Har
Dept. of Information & Communication, Gwangju Institute of Science & Technology,
1 Oryong–dong, Buk–gu, Gwangju 500–712, Rep. of Korea
e-mail: yzafar@gist.ac.kr , hardon@gist.ac.kr

Abstract—**Side Channel Attacks (SCAs) have proven to be very effective in extracting information from algorithmically secure systems. Since the earliest reports of attacks exploiting side channels such as power consumption, timing behavior and electromagnetic radiation etc., the countermeasures to resist such attacks have also been proposed. A variety of countermeasures resisting SCAs have been presented that continue to fade away as resistant attack techniques are developed. Field Programmable Gate Arrays (FPGAs) were originally thought to be resistant to such attacks because of some inherent characteristics. Later, they were also found to leak information over the side channels. In this article a novel countermeasure is presented that hardens an FPGA based system with cipher embodiment, against SCAs. The proposed methodology of embedding single inverter ring oscillators within the synchronous cores helps improve immunity against electromagnetic, fault and glitch attacks, while the introduction of frequency hopping by randomly varying frequency driving the cipher hardens the system against power and timing attacks. The incorporated countermeasure enhances the immunity of FPGA based implementation against multiple types of SCAs without adversely affecting cost or performance.**

Keywords-Side Channel Attacks; countermeasures; FPGAs; frequency hopping

I. INTRODUCTION

The reconfigurable nature of FPGAs offers major advantages for cryptographic applications because of ever-changing standards and the high Application Specific Integrated Circuit (ASIC) costs. However, like other Complementary metal–oxide–semiconductor (CMOS) based custom ASICs, commercially available FPGAs based on similar technology also leak information over the side channel, requiring incorporation of countermeasures to resist SCAs [1–2]. When incorporating these countermeasures the biggest challenge is to keep design cost from escalating and performance from degrading [3]. A complex countermeasure that increases the design real-estate considerably is therefore detested and the one enhancing immunity of the system against multiple kinds of SCAs is venerated.

SCAs against ASIC implementations and their countermeasures have extensively been reported in literature. Introducing SCA countermeasures to FPGA implementations is a relatively recent trend [3–5]. A globally asynchronous locally synchronous ASIC with Advanced Encryption Standard (AES) cipher reported by Gurkaynak et al. [6], combines operation reordering and unpredictable latencies with asynchronous clock domains and self varying clock cycles to counter SCAs. However, it contains local clock generators with delay control and other asynchronous elements customized at switch level, incompatible with FPGA implementation. In this article an FPGA based multi-clock system with embedded single inverter ring oscillators (SIROs), embodying 128-bit AES cipher is presented that employs frequency hopping to enhance immunity against SCAs. FPGA implementation of SIROs driving micropipelined and synchronous architectures has already been reported [7–8].

The proposed design consists of different synchronous units triggered by their local SIRO based clock sources that reside within them. A special circuit called the hopper, with its local SIRO based clock source randomly selects the frequency that encrypts each new 128-bit data frame. We refer to this random frequency selection, as frequency hopping.

The remaining sections of this paper are organized as follows. Cryptanalysis and the side channel attacks are reviewed in Section II. Vulnerability of FPGAs against SCAs is also discussed in the same section. Possible countermeasures against the SCAs are summarized in section III. FPGA compliance of SIRO–based designs is high-lighted in Section IV. Section V describes the counter mode of Rijndael AES contained within the system. An overview of the experimental setup and system architecture is presented in Section VI. Frequency hopping incorporated through Hopper circuit, thwarting SCAs is discussed in Section VII and the conclusion is drawn in Section VIII by attributing enhanced SCA immunity to the proposed system.

II. CRYPTANALYSIS AND THE SCAS

Cryptanalysis (from the Greek *kryptós*, "hidden", and *analýein*, "to loosen" or "to untie") is the study of methods for obtaining the meaning of encrypted information, without access to the secret information which is normally required

to do so [9]. The first known recorded explanation of cryptanalysis was given by 9th-century Arab polymath, Abu-Yusuf Al-Kindi where he presented the method of frequency analysis to decipher encrypted messages [10].

For a cryptographic system to remain secure, the secret keys used in performing the required security services must not be revealed. In cryptanalysis, typically the protected keys are revealed by analyzing the cryptographic algorithm. The task is time consuming and cumbersome. Another technique used in cryptanalysis is based on the fact that cryptographic algorithms are always implemented in software or hardware on physical devices which interact with and are influenced by their environments. These physical interactions can be monitored by adversaries to extract protected information. This type of information is called side-channel information, and the attacks exploiting side-channel information are called side-channel attacks. The whole idea of SCA attack is to look at the way cryptographic algorithm is implemented, rather than at the algorithm itself [11].

The first official information related to SCA attack dates back to the year 1965, when P. Wright reported in [12] that MI5, the British intelligence agency, was trying to break a cipher used by the Egyptian Embassy in London. However, a major contribution in this field was made by Paul Kocher in 1996, when he successfully launched an SCA called the timing attack against secure implementations. In his report, Kocher had measured the amount of time required for private key operations, to experimentally reveal secret information [13]. Another kind of SCA called the fault attack associated with incorporating faults in the operation of cryptosystems was first reported in 1997 by Boneh et al. [14]. Hardware processing done at abnormally high or low frequencies or under extreme temperature conditions may induce faults that an attacker can observe to evaluate hidden information.

SCAs observing power consumption were soon discovered to be far more effective than the ones observing other side channels. In 1999, Paul Kocher introduced Simple Power Analysis (SPA) and Differential Power Analysis (DPA) [15]. Power analysis attacks exploit the fact that power consumption of a CMOS circuit is data dependent based on the switching activity governed by (1).

$$P_D = C_L (V_{DD})^2 P_{0\rightarrow1} f \quad (1)$$

where P_D is the dynamic power consumed by a single CMOS gate; C_L is the gate load capacitance; V_{DD} is the supply voltage; $P_{0\rightarrow1}$ is the probability of a $0\rightarrow1$ output transition making the measurement data dependent and f is the clock frequency [16]. For DPA, the attacker measures the power consumption of the device while it processes a large set of cryptographic operations. As opposed to SPA, where the information is extracted directly from the power measurements, DPA uses statistical techniques to compare measured values to a set of estimated values.

In 2002 Agrawal et al., reported some of the first successful attacks based on the analysis of electromagnetic emissions [17].

Over the past decade perceptions regarding FPGAs have changed. Instead of being viewed as devices used only for prototyping, they are now seen as end-products containing low-cost, reconfigurable hardware. The status of Hardware Descriptive Languages (HDLs) has also changed from tools used for hardware simulation to synthesizable mediums. This transition has resulted in the migration of ASIC-based designs to the FPGA platforms, especially in the areas like cryptography that require frequent upgrade of end-products.

FPGAs have also been looked upon favorably for cryptographic applications due to certain inherent features, believed to provide practical security against SCAs. These features include the intrinsic parallel computing capability of FPGAs that leads to algorithmic noise complicating measurements related to a specific event and the ability of different bits in the observation area to contribute differently to the overall power consumption. This variation is due to the pre-laid out structure of interconnects between computational elements that results in different effective capacitances [18]. However, this notion of inherent FPGA immunity was negated in 2003 by Örs et al., when they successfully mounted a power analysis attack against an FPGA based elliptic curve cryptographic processor [19]. Attacks exploiting the electromagnetic leakage of FPGAs were first reported by Carlier in 2004 [20].

III. SCA COUNTERMEASURES

Countermeasures typically enhance the SCA immunity of a cryptosystems by either abating or adulterating leakages. Noise addition, data randomization, duplication & Boolean masking and implementation using dynamic & differential logic styles are some of the techniques involved in incorporating countermeasures [5]. The purpose is it to resist leakage measurements or to complicate the comparison of estimated vs. the measured values during statistical analysis based attacks.

Introduction of countermeasures to enhance SCA immunity in hardware often deals with transistor-level changes of the logic that are not easily applicable to FPGAs without manufacturers' support [3]. We therefore, observe a general trend in evolution of SCAs and related countermeasures targeting ASICs in the beginning and later being modified for FPGA compliance.

Different methodologies have been proposed for each category, and new methodologies continue to spring up as countermeasures cede to new kinds of attacks, immune to the outdated versions.

IV. FPGA RESIDENT SIRO

Embedded SIRO based clock sources have already been reported to drive co-existing synchronous systems in FPGAs [8]. The implementation of a ring oscillator in FPGA is simple and consists of a single inverting stage comprising of a 2–input NAND gate, the output of which is fed back to one of its inputs while a logic low on the other gate input is used to pause the oscillator. Fig. 1(a) shows a simple SIRO circuit. The 2-input NAND based SIRO implementation in FPGA is metaphorical, for the FPGA–centric circuit uses a logic element (LE) in the combinatorial mode to implement the functionality of gate, as shown in Fig. 1(b). Fast interconnects (direct / local) establish the feedback ring. Simplicity is necessary in case of FPGA based SIRO, for it to be technology independent as well as power efficient [7–8]. A change in environmental conditions affects it in the same way as it affects the other components of the co-existing system. Therefore, the SIRO based clock source and the circuit driven by it adapt to physical conditions in a similar manner [8].

In this article, an FPGA based multi-clock system with cipher embodiment is proposed. It consists of different synchronous units triggered by their local SIRO based clock sources that reside within them. A special circuit called the hopper, with its local SIRO randomly selects the frequency that drives the cipher. The system is hardened against electromagnetic, fault and glitch attacks, due to presence of multiple SIROs that are adaptive, unsynchronized and physically invisible (on any external pin) outside the FPGA. On the other hand, timing and power attacks are thwarted by the proposed frequency hopping scheme.

V. ADVANCED ENCRYPTION STANDARD

A block cipher developed by two Belgian cryptographers, Joan Daemen and Vincent Rijmen, submitted under the name 'Rijndael' was formally adopted as Advanced Encryption Standard (AES) by National Institute of Standards and Technology (NIST), USA on 6th of December, 2001 [21]. AES is a block cipher where several different transformations, such as 'Sub Bytes' (substitution of bytes by alternate values from S-Box), 'Shift Rows', 'Mix Columns' and 'Add Round Key' are iterated in so called rounds. Additional modes have also been proposed for AES to enhance its ciphering capabilities. In 2001, Dworkin recommended some modes for AES including the counter mode (CTR) guaranteeing change in ciphered value even when plain data and encryption key remain unchanged [22]. In counter mode AES (AES–CTR) the cipher key actually encrypts an initialization vector (IV) instead of plain data, as shown in Fig. 2. The resulting ciphered IV is the mask that is bitwise XORed with plain data in order to encrypt it. The ciphered data is XORed with the mask once again on the decryption side to regenerate plain data. In the presented model, a 128-bit IV initialized by Linear Feedback Shift Register (LFSR), consisting of 64-bit nonce for identification and 64-bit counter field is used. A unique counter value per frame ensures distinct ciphered data even if the two plain data frames are identical.

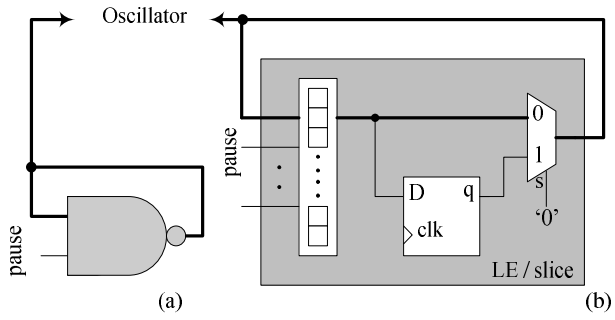

Figure 1. Single inverter ring oscillator (SIRO): (a) 2-input NAND based gate level design, (b) Logic Element based FPGA implementation.

VI. SYSTEM ARCHITECTURE

Xilinx's XC3S1000 low-cost FPGA was used to implement the system. A dedicated FPGA board with three serial ports was designed. Interestingly, despite being used for the implementation of a multi-clock system, the board does not house any oscillator.

The developed FPGA–based cipher system is plugged into an existing audio communication device as shown in Fig. 3. On the transmitter side, the device receives data from a sensor. After pre-amplification and filtering, the data is digitized by analog-to-digital converter (ADC). The digital data passes through a codec where the bit rate is reduced and the 18-byte frames consisting of two header bytes and 16 data bytes are serially transmitted to a modem at 57.6kbps with 1 stop bit and no parity. While receiving, the modem sends serial data with similar specifications to the codec. The codec after decompressing pushes the digital data to a digital-to-analog converter (DAC) that generates an analog signal which after amplification is sent to the transducer. The cipher system communicates with the codec and modem of the audio communication device serially. It does not require any external oscillator to communicate with these devices as its serial ports are designed to have the same

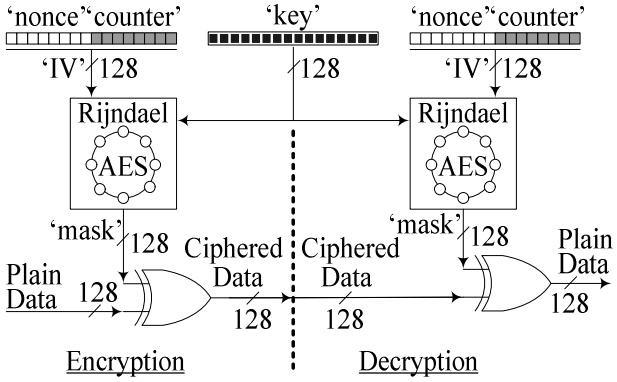

Figure 2. Block diagram of 128-bit AES–CTR

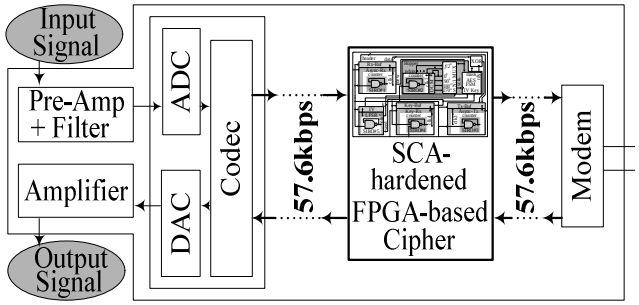

Figure 3. SCA hardened cipher system plugged into pre-existing audio communication device

specifications (57.6kbps, 1 stop bit with no parity) as required by the host communication device.

The transmitter section of the presented multi-clock system as shown in Fig. 4, consists of five units. These units are: the receiver, the cipher, the linear feedback shift register (LFSR), the key receiver and the transmitter. Each unit is independently synchronous, driven by its local SIRO-based clock source that resides within it. The five clock sources local to the five units are independent of each other, none affecting the other or its unit.

Data enters the FPGA-based system serially, in the form of 18-byte frames. Each frame consists of two header bytes used for synchronization and 128-bit (16-byte) data. The data is received by an asynchronous receiver 'Async-Rx' designed to communicate with the host device at its port settings, as explained in the last paragraph. The start bit is a 'Space', while a single stop bit is represented by 'Mark State'. 'Async-Rx' after receiving the data, stores it in a 128-bit buffer 'Rx-buf'. 'Async-Rx' and 'Rx-Buf' constitute the first unit of the system called the receiver unit. The receiver unit has an embedded oscillator 'SIRO#1', the output of which is divided by a counter to generate clock source 'clk1', which is further divided to generate the desired baud rate.

This unit is also responsible for generating the 'new' pulse for two 'clk1' cycles that triggers the cipher unit to prepare 'mask' for XORing required by AES operating in the counter mode. At the rising edge of 'new', the least significant bits of LFSR select the frequency to be used for

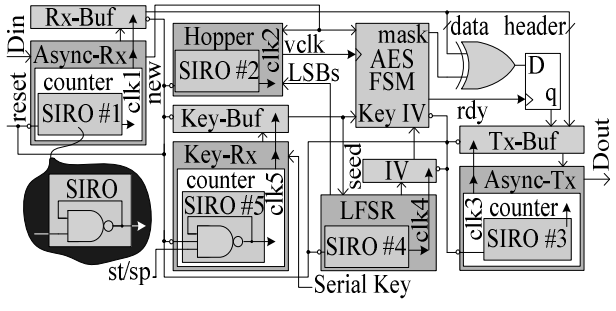

Figure 4. Frequency–hopping based SCA hardened Cipher system

ciphering process. So while the 16-bytes of plain data are buffered into the 'Rx-Buf', cipher unit calculates the 'mask'.

The most important unit of all is the cipher unit. It contains three subsections. The first one is a finite state machine called AES FSM, responsible for the execution of various steps of AES algorithm. This FSM controlling the steps of encryption and key scheduling is driven by an oscillating signal 'vclk' provided by the other subsection of the unit called the hopper circuit. This circuit responsible for the realization of frequency hopping will be discussed at length in Section VII of the article. A 128-bit XOR, performing the operation between 'mask' and the plain data constitutes the third subsection of the cipher unit.

AES FSM performs the Rijndael encryption and key scheduling in parallel and the round keys are generated at runtime. Different steps of the two concurrent processes are distributed over the states of FSM, in such a manner that the round key to be added during a round is generated prior to its addition. NOPs (no operations) are inserted along the two concurrent processes to synchronize the sequences and schedule the use of a single, 4-byte S-box implemented using limited memory resources of FPGA, over the two processes.

Insertion of 'new' initiates the FSM that pulls the 'rdy' signal low and latches the 'key' provided by the key receiver unit and the 'IV' provided by the LFSR unit. This 'rdy' signal does not go high till all the 10 rounds of AES are complete and any change in cipher's inputs 'IV' or 'key', is ignored till 'rdy' is low. Once all the rounds are complete, the 'rdy' signal goes high to inform the unit that the 'mask' is ready to be XORed with plain data. FSM maintains the state keeping 'rdy' signal high, till a positive edge is encountered on 'new' and the FSM is re-initialized.

As AES FSM calculates the 'mask', 'Rx-Buf' is filled with serially received data. Once the 'Rx-Buf is full, it is bitwise XORed with 'mask' and sent to the transmitter unit, driven by 'SIRO#3' generated 'clk3'. The ciphered data is loaded into the 128-bit transmit buffer 'Tx-Buf' of the transmitter section, from where the asynchronous transmitter 'Async-Tx' fetches a byte at a time to transmit it out of the cipher system to the modem, serially, at the system specifications of 57.6kbps, 8 data bits, 1 stop bit and no parity. Baud rate for transmission is generated in this unit by its local clock source 'clk3'.

Another unit is the 128-bit LFSR driven by its locally embedded 'SIRO#4'. Its responsibilities include initialization of IV and providing support to the hopper circuit.

The fifth unit is the key receiver. Its main responsibility is to serially receive encryption 'key' and store it. For this purpose its structure resembles that of the receiver unit. It consists of an asynchronous receiver 'Key-Rx' operating at the same specifications as the other serial ports and a 128-bit buffer called 'Key-Buf' that latches the 'key' once it is fed to the system from outside. This unit is clocked by its local

'SIRO#5' based clock source 'clk5'. However, 'SIRO#5' is different from other ring oscillators of the system as it consists of a three input NAND gate instead of a two input NAND. The additional input is the st/sp signal that stops 'SIRO#5' from oscillating once the 128-bit 'key' is completely buffered into 'Key-Buf'. This 'key' is used by the cipher unit for the encryption. In addition it is also used to initiate the LFSR by acting as its seed.

VII. HOPPER CIRCUIT AND FREQUENCY HOPPING

Fig. 5 shows the hopper circuit of the cipher unit. This circuit is responsible for the realization of frequency hopping presented in this article that hardens the FPGA-based cipher against SCAs. Hopper circuit embodies 'SIRO#2', the oscillating output of which is divided by an n-bit counter to generate n sub-frequencies. Four of these sub-frequency signals are fed to the Digital Clock Managers (DCMs) of the FPGA device. Each of the DCMs creates additional four frequency signals with phase shifts of 0°, 90°, 180° and 270° w.r.t. the input signal. One of these N number of frequencies, at the rising edge of signal 'new' generated by the receiver unit, is selected as 'vclk', by an N:1 multiplexer. The select lines of the multiplexer are connected to the k least significant bits (LSBs) of LFSR, where $N = 2^k$. As discussed in the previous section, 'vclk' is the signal that drives AES FSM of the cipher unit and positive transition on 'new' forces a new frequency to ride this signal. It is to say that while a 16-byte frame of plain data is being buffered in, 'IV' is encrypted by 'key' to generate a 'mask' for XORing, as a randomly selected

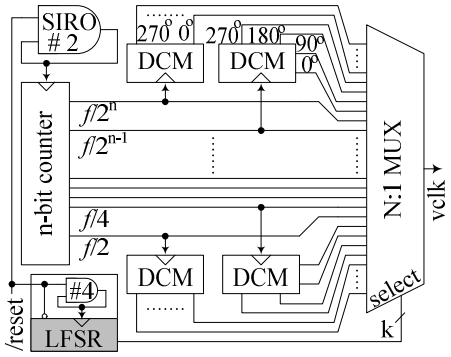

Figure 5. Hopper circuit

frequency drives the AES FSM. The FSM creates a new 'mask' with incremented IV as a new randomly selected frequency drives it for the next data frame. One can observe 'vclk' driving AES FSM, hopping to a different frequency after every 'new' pulse in Fig. 6(a). We refer to this cipher frequency randomization as the process of incorporating frequency hopping in communication security, a characteristic generally associated with transmission security.

Fig. 6 shows the operation of AES FSM under the influence of frequency hopping. The post-layout simulation results for AES FSM driven by 'vclk' are presented in Fig. 6(a). Signal 'new' pulls 'rdy' low to initiate the execution of Rijndael steps. The 'key' encrypts 'IV' to generate the 'mask', upon which 'rdy' goes high and stays high till a positive transition on 'new' is encountered. Each 'new' pulse selects a different frequency for 'vclk'. The serial data

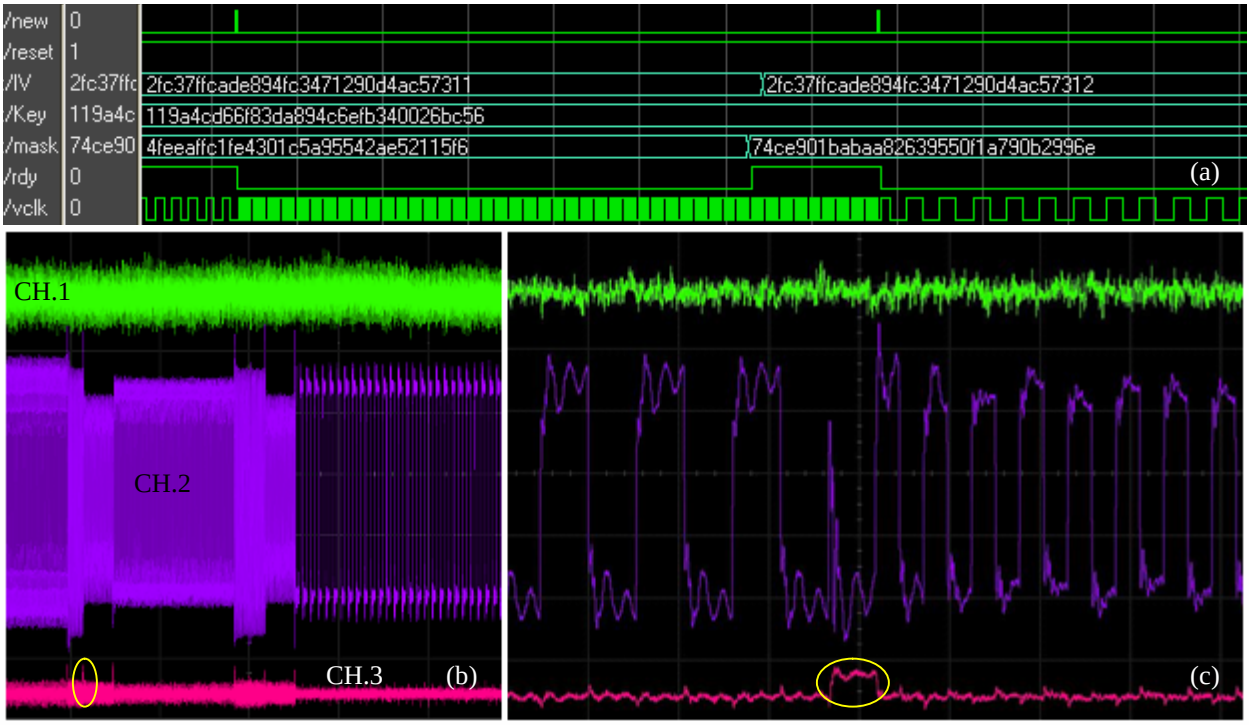

Figure 6. Frequency hopping based AES–CTR: (a) Post–layout simulation high-lighting frequency hopping [23], (b) Current measurements upon frequency hopping captured on oscilloscope, (c) zoomed view of oscilloscope capture

streams in and out of our AES–CTR based system at a constant rate of 57,600bps. If there is an error free transmission of data, then each frame consisting of 16 data bytes, 2 header bytes, one start and one stop bit per byte, streams in and out of the system in approximately 3.125 mSec. Whereas, the experimental results demonstrate that minimum amount of time with highest self-generated frequency selected to perform ciphering, creates the 'mask' in 2.052635 µSec. Therefore, any frequency can be randomly selected by hopper to drive AES FSM that does not interrupt the continuous flow of data. The point to be noted here is that the cipher itself executes a particular step of algorithm (like the Sub Byte) under observation for power measurements, at random intervals w.r.t. the initialization of encryption process because of the randomly selected frequency driving AES FSM for each frame.

Fig. 6 (b & c) present the oscilloscope screen captures during physical measurements used for analysis. Agilent Infiniium DS08104A Oscilloscope was used for these measurements. Channel no. 1 (CH.1) displays the current readings acquired using Agilent 1147A current probe with the main power supply of the FPGA board. Channels no. 2 and 3 (CH.2 & CH.3) display the status of signals 'vclk' and 'new'. In Fig. 6(b) frequency of 'vclk' is observed to change upon transitions on 'new'. However, this frequency change does not affect the current readings. Fig. 6(c) is the zoomed view of a segment of capture presented in Fig. 6(b). A transition on 'new' changes the frequency of signal 'vclk', significantly, but the current patterns used by the attacker, do not high-lighting this change, explicitly. Therefore, the activity of a particular section of the algorithm like the s-box under observation cannot easily be identified / distinguished on the time scale, especially when repeated measurements of the same instance are required to deduce the exact information, as in case of DPA. Monitored activity is distributed randomly on the time axis because of this frequency drift, and the physical current measurements do not convey enough information to identify this activity at specific time. This makes very complicated the synchronization of the DPA and SPA signatures, thwarting power analysis based on statistical techniques. The result is what may be described as "smearing the peaks of differential trace due to de-synchronization effect". The attacker is therefore, forced to collect more data, and the system thus exhibits itself as the one with enhanced immunity against power analysis attacks.

VIII. CONCLUSION

The proposed system, because of its architectural features inherently resists fault and glitch attacks. The clock signals do not present themselves externally. Therefore, the extraction of timing information or insertion of glitches is more cumbersome. Manipulation of frequencies for fault injection is also difficult because of the adaptive nature of SIROs [8]. Multiple and unsynchronized oscillating signals immunize the system against Electromagnetic attacks. Furthermore, random selection of oscillating frequency at run time, to act as a clock source driving AES FSM means unpredictable clock, that complicates the synchronization of the power signatures. When using differential power analysis, the exact time of a particular operation under observation varies randomly due to this characteristic referred to as frequency hopping in the article. It can therefore safely be concluded that the presented methodology of introducing frequency hopping to embedded SIRO driven FPGA implementations enhances their immunity against multiple types of SCAs without performance or cost trade-offs.

ACKNOWLEDGMENT

This work was supported in part by the Center for Distributed Sensor Network at GIST and in part by the Regional Innovation Program funded from Ministry of Knowledge Economy.

REFERENCES

[1] F.-X. Standaert, "Secure and efficient use of reconfigurable hardware devices in symmetric cryptography", Ph.D. Thesis, UCL Crypto Group, Universite' catholique de Louvain, Belgium, June 2004.

[2] E. Peeters, F.-X. Standaert, N. Donckers, and J.-J. Quisquater, "Improved higher-order side-channel attacks with FPGA experiments", CHES 2005, LNCS 3659, 2005, pp. 309-323.

[3] T. Wollinger, J. Guajardo, and C. Paar, "Cryptography on FPGAs: State of the art implementations and attacks", ACM Transactions on Embedded Computing Systems, Aug. 2004, vol. 3, no. 3, pp. 534–574.

[4] F. -X. Standaert, F. Mace, E. Peeters, and J. -J. Quisquater, "Updates on the security of FPGAs against power analysis attacks", ARC 2006, LNCS 3895, 2006, pp. 335-346.

[5] F.-X. Standaert, E. Peeters, G. Rouvroy, and J.-J. Quisquater, "An overview of power analysis attacks against Field Programmable Gate Arrays", Proceedings of the IEEE, vol. 94, no. 2, Feb. 2006, pp. 383-394.

[6] F. Gurkaynak, S. Oetiker, H. Kaeslin, N. Felber, and W. Fichtner, "Improving DPA security by using globally-asynchronous locally-synchronous systems", Proceedings of ESSCIRC 2005**,** Sep. 2005, pp.407- 410.

[7] Y. Zafar and M. M. Ahmad, "A novel FPGA compliant micropipeline", IEEE Transactions on Circuits & Systems II, Sep. 2005, 52, (9), pp. 611–615.

[8] Y. Zafar, "FPGA-compliant micropipeline based asynchronous systems", PhD thesis, M.A. Jinnah Univ., 2005. http://eprints.hec.gov.pk/510/1/383.html.htm

[9] M. J. Aqel, Z. A. Alqadi, and I. M. El Emary, "Analysis of stream cipher security algorithm", Journal of Information and Computing Science, 2007, vol. 2, no. 4, pp. 288-298.

[10] I. A. Al-Kadi , "The origins of cryptology: The Arab contributions", Cryptologia, vol. 16, no. 2, April 1992, pp. 97–126.

[11] Y. Zhou and D. Feng, "Side-channel attacks: Ten years after its publication and the impacts on cryptographic module security testing", NIST Physical Security Testing Workshop, Hawaii, USA, Sep. 2005. Cryptology ePrint Archive, Report 2005/388, 2005, http://eprint.iacr.org/

[12] P. Wright, "Spy Catcher: The candid autobiography of a senior intelligence officer", Viking Press (Penguin Group), 1987.

[13] P. C. Kocher, “Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems”, LNCS 1109, 1996, pp. 104-113.

[14] D. Boneh, R. A. Demillo, and R. J. Lipton, “On the importance of checking protocols for faults”, Advances in Cryptology–Eurocrypt '97, LNCS 1233, Springer, Berlin, 1997, pp. 37-51.

[15] P. Kocher, J. Jaffe, and B. Jun, “Differential Power Analysis”, CRYPTO 1999, LNCS 1666, Aug. 1999, pp. 388-397.

[16] J. M. Rabaey, “Digital integrated circuits”, Prentice Hall International, 1996.

[17] D. Agrawal, B. Archambeault, J. R. Rao, and P. Rohatgi, “The EM side-channel(s)”, CHES 2002, LNCS 2523, August 2002, pp. 29-45

[18] L. Shang, A. Kaviani and K. Bathala, “Dynamic power consumption in Virtex-II FPGA family”, ACM/SIGDA tenth international symposium on Field-programmable gate arrays (FPGA'2002), Feb. 2002, pp. 157-164.

[19] S. B. Örs, E. Oswald, and B. Preneel, “Power-analysis attacks on an FPGA—First experimental results”, CHES 2003. LNCS 2279, 2003, pp. 35–50.

[20] V. Carlier, H. Chabanne, E. Dottax, and H. Pelletier, “Electromagnetic side channels of an FPGA implementation of AES”, IACR E-print Archive 2004/145, 2004. http://eprint.iacr.org

[21] National Institute of Standards and Technology, “Advanced Encryption Standard”, FIPS PUB 197, 2001. http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[22] M. Dworkin, “Recommendation for block cipher modes of operation: Methods and techniques”, NIST Special Publication 800–38A, 2001.

[23] Y. Zafar and D. Har, “A novel countermeasure enhancing side channel immunity in FPGAs”, Proceedings of IARIA International Conference on Advances in Electronics and Micro-electronics (ENICS 2008), Sep.2008, pp. 132-137.

Yousaf Zafar received a B.S. degree in Electrical Engineering from the University of Wyoming, USA in 1993. His MS in Electronics Engineering from GIK Institute of Engineering Sciences & Technology, Pakistan led to his PhD in the same field from M.A. Jinnah University, Pakistan in 2005. He is currently working as a Postdoctoral Research Associate at the Department of Information and Communication, Gwangju Institute of Science and Technology, Rep. of Korea. His research interests include Reconfigurable Asynchronous Processing and hardware realization of complex mathematical algorithms related to communication and information security.

Dongsoo Har received the B.S. and M.S. degrees in Electronics Engineering from Seoul National University, Rep. of Korea in 1986 and 1988, respectively. He finished his PhD in Electrical Engineering at Polytechnic University at Brooklyn, NY, USA in 1997. He is currently an Associate Professor at the Department of Information and Communication, Gwangju Institute of Science and Technology, Rep. of Korea. His research interests include multimedia processing IP design and implementation, as well as design and implementation of Low-power embedded systems.