Towards a Flexible, Opensource BOard for Side-channel analysis (FOBOS)

Rajesh Velegalati and Jens-Peter Kaps
Cryptographic Engineering Research Group (CERG)
ECE Department, George Mason University
Fairfax, VA, USA
Email: rvelegal, jkaps@gmu.edu

***Abstract*—Side-channel analysis (SCA) attacks pose a growing threat to implementations of cryptographic algorithms implemented in software as well as in hardware. Current standard side-channel evaluation boards with Field Programmable Gate Arrays (FPGAs), that allow for exploring the vulnerability of cryptographic implementations on FPGAs, are expensive and available only for a few FPGA devices. Furthermore, a complete open source software package that includes drivers that run test cases on the board, control the measurement equipment, and contain several side-channel analysis techniques is not readily available. Each user has to assemble their own setup based on software packages from multiple sources, written in multiple languages and write parts themselves. Additionally, this complexity and cost makes it very difficult, if not impossible, to educate students on side-channel analysis through hands-on laboratory exercises. We introduced FOBOS, an open-source framework for conducting side-channel attacks on FPGAs, at the work in progress session of COSADE 2012, and it was met with a lot of interest from universities and research groups. We expect to release the first version this Summer. It will feature support for multiple FPGA devices and include all necessary software to run differential power analysis attacks, which are the most prominent kind of side-channel attacks. Furthermore, FOBOS integrates with the low cost OpenADC board to form a complete low-cost SCA solution for less than $200, which will be ideal for educational use. The components of FOBOS are build in a modular fashion so that it can easily be adapted for new FPGA boards, oscilloscopes, and attack techniques. Our next steps are integrating support for fault analysis, including circuitry to cause power and clock faults, and adding new targets, such as ASICs and smart cards.**

***Keywords*-Side-Channel Analysis (SCA), Differential Power Analysis (DPA), SCA Board, SCA Package**

I. Introduction and Motivation

Even though the cryptographic algorithms are designed to withstand rigorous cryptanalytic attacks, an adversary can obtain the secret information by observing the so called side-channel leakage from the cryptographic device. These side-channels can be power consumption [1],[2], execution time [3], or electromagnetic radiation [4],[5] of the device.The side-channels leak sensitive information whenever the device performs an operation using the secret data. Attacks which make use of such inherent physical leakage are called side-channel analysis (SCA) attacks. Generally, all hardware implementations of cryptographic algorithms are assumed to be vulnerable to side-channel cryptanalysis, if there are no special precautions in the implementation.

Hardware implementations of cryptographic algorithms target either Application Specific Integrated Circuits (ASICs) or Field Programmable Gate Arrays (FPGAs). Recent architectural advances of FPGAs are making them an alternative choice for low power applications where Application Specific Integrated Circuits (ASICs) are primarily used. A hallmark of FPGAs is the ability to implement parallelized architectures efficiently, and they also posses excellent resistance against invasive attacks since the underlying platform is regular and does not reveal information on the actual design content. Because of these features, FPGAs have become an attractive hardware platform for cryptographic implementations. While a few FPGA boards designed for SCA exist, many research groups from academia and industry use their own hardware harness, their own software for data acquisition and data analysis and sometimes their own FPGA boards or generic FPGA boards. This increases the complexity and effort needed to obtain a working SCA setup. Another, but costly option is the use of commercial SCA workstations.

Due to the importance of the topic of side-channel attacks, they became part of the curriculum of cryptography courses in many universities. However, only very few have associated laboratory exercises and hands-on examples due to the cost and complexity of current SCA setups.

To our knowledge no complete software package exists that contains everything needed for evaluating the side-channel attack resistance of FPGA implementations from data acquisition to analysis (see Sect:II). In this paper, we are presenting an initial framework for efficient side-channel evaluation of cryptographic implementations on hardware and software. Such an environment should be flexible, open-source and low cost and beneficial to both research and educational communities.

II. Previous Work

A. SCA - Hardware Platforms

The Side Channel Analysis Board (SCAB) introduced in [6], was one of the early efforts in developing evaluation platforms for conducting SCA attacks on implementations of cryptographic algorithms. This board housed an FPGA on which the cryptographic algorithms can be implemented along with an unrestricted access to power and clock pins to perform the following SCA attacks: Differential Power Analysis (DPA)

TABLE I
SASEBO BOARDS WITH FPGAS AS DEVICE UNDER TEST (DUT)

Board	Control FPGA	DUT FPGA	DUT Techn.	**Wires Control–DUT**	Host Data Communication	Status
SASEBO	Virtex-2 Pro	Virtex-2 Pro	130 nm	54	RS232	Discontinued
SASEBO-G	Virtex-2 Pro	Virtex-2 Pro	130 nm	53	RS232, FT245RL (USB)	
SASEBO-GII	Spartan-3A	Virtex-5	65 nm	46	FT2232D (USB)	Discontinued
SASEBO-B	Stratix-2	Stratix-2	90 nm	53	RS232, FT245RL (USB)	
SAKURA-X	Spartan-6	Kintex-7	28 nm	78	USB	Under Development

and fault analysis. Information about the board design and the status of the project is currently not available.

The Side-channel Attack Standard Evaluation Board (SASEBO) [7],[8],[9] was developed by the Research Center for Information (RCIS) of National Institute of Advanced Industrial Science and Technology (AIST) and Tohoku University as a common platform for evaluating side-channel attacks. These boards were developed with the intent of performing side-channel attacks on various hardware platforms like FPGAs, ASICs and Smart cards. SASEBO boards are designed with two SoCs, a cryptographic FPGA (or ASIC/Smart card) where the algorithm can be implemented and a control FPGA which directs the data flow between the software and the cryptographic FPGA. The data acquisition software which comes with SASEBO is written in C#. It does not provide support for different brands of oscilloscopes. Hence the user is required to tweak the code to provide support for his/her own oscilloscope. Only four different types of SASEBO boards with FPGAs as Devices Under Test (DUT) (shown in Table I) exist of which 2 are discontinued. Morita Tech [10] recently announced SAKURA as a successor to SASEBO project.

B. SCA - Data Acquisition & Analysis Platforms

The DPA Contest [11] organized jointly by VLSI research group of Telecom ParisTech university and AIST, is an online-based contest with the aim of having a fair confrontation between different attack methodologies. Currently three editions of this contest were introduced of which the first two deal primarily with attacking DES (v1) and AES (v2) using different techniques where as the goal of the third edition is to compare acquisition platforms and techniques. This contest provides a wealth of information regarding DPA statistical techniques, although all the data acquisition is obtained from SASEBO GII only.

The OpenSCA Toolbox [12] is an open source project which consists of set of Matlab codes and objects to perform DPA attacks. Using this toolbox one can conduct not only first order power analysis attacks but also the higher order and template attacks. The toolbox also comes with several examples, demonstrating the attacks. Currently the supported statistical testing procedures are Difference-of-Means, Correlation Power Analysis and Baysian analysis. All codes are written in Matlab and does not include data acquisition. In short, we can perform only data analysis using OpenSCA.

The DPA Workstation™ [13] is a state-of-the art proprietary SCA testing platform by *Cryptography Research, Inc.* It can perform data acquisition, processing and analysis and also has the ability to launch both power and EM attacks on multiple hardware platforms. The Inspector [14], a state-of-the art proprietary SCA and Fault Injection testing platform by *Riscure*, can perform not only SCA attacks (power & EM) but also fault injection attacks such as voltage, clock and fault injection through an optical (laser) source. The major drawback is that these tool are not freely available and licensing is very costly, thus not usable for educational purposes. Also, collaborations between research groups are difficult as they might not all have access to the DPA Workstation™or Riscure's Inspector.

The IAmeter [15], currently being developed at Virgina Tech, provides a modular set of scripts for data acquisition and a database to store the data collected and the acquisition settings. The IAmeter is portable i.e., can be used independent of the hardware platform or data acquisition equipment..

C. Drawbacks of Current SCA Evaluation Platforms

An efficient SCA evaluation platform should fulfill the following criteria:

- Flexibility: Able to support multiple hardware platforms/technologies/vendors.
- Open Source: Community support will allow for rapid development and adoption of the latest devices and technologies.
- Reproducibility: Results published in research should be reproducible to obtain a fair side-channel analysis of cryptographic algorithms.
- Broad-Spectrum Acceptance: Should be accepted by both educational (low-cost) and research/industry (state-of-the-art) communities.

We have shown in Sect. II-A and Sect. II-B that a complete (acquisition to analysis), free and open source solution is not available. Therefore, research groups and industry who do not want to invest in proprietary SCA testing platforms employ home grown scripts, programs and platforms. Their main disadvantages are that they are mostly written in an ad-hoc fashion and therefore difficult to maintain and extend. These scripts and platforms are also proprietary and hence, their results are not reproducible by other research groups. Hence there is a need for a flexible and complete open-source framework for SCA that allows fair and comprehensive evaluation of implementations on hardware platforms with reproducible results.

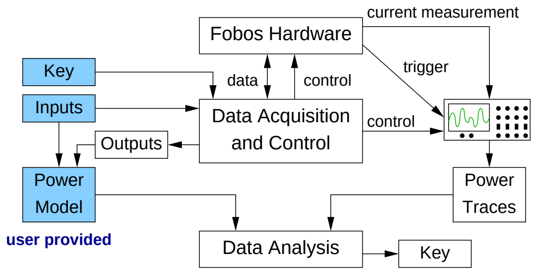

Fig. 1. Components of FOBOS

III. Our Approach

We call our framework for efficient side-channel evaluation of hardware platforms - FOBOS. This abbreviation stands for Flexible Open-sources BOard for Side-channel analysis. FOBOS, loosely named after the Greek god Phobos ($\phi \acute{o} \beta o \varsigma$) who personifies fear and can pierce shields. FOBOS is designed to be an inexpensive side-channel analysis setup that includes a complete software package with programs for DUT control, data acquisition and data analysis. In order to evaluate side-channel leakage of hardware platforms, FOBOS uses off-the shelf FPGA boards as control and DUT which are less expensive than the traditional setup. Furthermore, we integrated support for the low cost data acquisition board OpenADC [16], eliminating the need of a costly digital oscilloscope for several analysis scenarios. Thus, it enables universities to add active side-channel analysis laboratory exercises to their cryptography classes. FOBOS is designed in a modular fashion to allow for a multitude of DUTs while maintaining the remainder of the setup, hence making FOBOS flexible. The FOBOS software package, documentation, and hardware components will be released as open-source for quick adaptation of newer technologies. Designers of cryptographic implementations and countermeasures against DPA and DEMA on FPGAs can test their design techniques on FPGAs from various vendors and with different technologies. As the hardware and software are open source, the results are reproducible by researchers from different groups.

Figure 1 shows various components of FOBOS. It consists of the *FOBOS Hardware* as well as software for *Data Acquisition and Control* and *Data Analysis*. The FOBOS Hardware consists of two FPGA boards that are connected to each other. It is also possible to use the SASEBO GII board instead. The user has to provide the hardware description of the cipher under investigation, the key, a set of inputs and a power model. The Data Acquisition and Control module configures and controls the FOBOS Hardware and the Oscilloscope. It takes the user provided key and inputs and sends them to the FOBOS Hardware which in turn encrypts the inputs with the key and returns the outputs (i.e. ciphertext). As soon as the FOBOS Hardware starts with the encryption, it sends a trigger signal to the oscilloscope to start data acquisition. The Data Analysis module uses the user supplied power model, which can be based on inputs and/or outputs, and the power traces collected by the oscilloscope to recover the key.

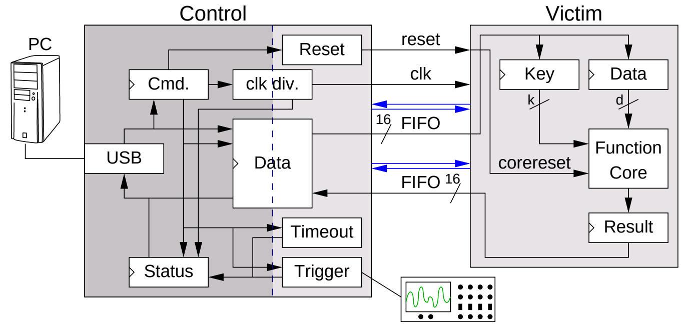

Fig. 2. Schematic Diagram of FOBOS Hardware

IV. Architecture of FOBOS

The following sections describe the functionality of various components of FOBOS.

A. FOBOS Hardware

A schematic diagram of the FOBOS hardware is shown in Fig. 2. It consists of two boards *Control Board & DUT Board* connected together by the so called bridge connector. The cryptographic algorithms whose security needs to be evaluated are to be implemented on the FPGA of the DUT board.

1) Control Board:: The control board used by FOBOS is either a Nexys2 or a Nexys3 board. Table II shows details of both boards. The control board contains several modules (see Fig. 2) and two clock domains. It uses the on-board 50 MHz oscillator as base clock for the USB communication to the PC. The second clock is generated through a clock divider circuit which uses the Digital Clock Managers (DCMs) to generate a clock in the range of 350 KHz $\sim$ 50 MHz from the 50 MHz oscillator on board depending upon the user's choice and the oscilloscope specification. This clock is used for communication with the DUT FPGA and also provided to the DUT FPGA board.

TABLE II
FOBOS FPGA Control Boards
both use USB2 for communication with the PC

Board	FPGA	Technology	Connector	Cost
Nexys 2	Spartan-3E	90 nm	Hirose FX2 (43)	\$149
Nexys 3	Spartan 6	45 nm	VHDC (40)	\$199

The control board receives commands from the PC and returns a status. This is facilitated through the 8-bit Command and Status registers. We use them to implement a simple protocol between PC and Control FPGA which is explained in Sect. IV-B2.

The Trigger module generates a reference point from which the oscilloscope should start measuring the power consumption of the DUT FPGA. Depending upon the user's requirement, this reference point can be set through a command to the beginning of the cryptographic operation or to specific clock cycle during the computation. This reference point is later used to perform signal alignment over several power traces.

A Timeout module makes sure that PC receives a status (of TIMEOUT) if an exception occurs during the communication with the DUT or if the DUT does not respond within a given time. This timeout value can be specified through a command. The timeout counter is automatically reset each time the DUT returns data.

The Reset module is used to send a reset signal to the function core implemented on the DUT FPGA. This is useful if for example a cryptographic operation takes 1,000 clock cycles to complete, however, the interesting event happens in the 30th clock cycle. The user can then reset the DUT automatically every 35 clock cycles and start a new encryption without having to wait for the encryption to complete.

2) DUT Board:: We are investigating several FPGA boards available on the market, which can be used as DUT boards for FOBOS. Table III shows some potential DUT boards. The column "V_{Core} Jumper" indicates whether the board contains a jumper on the core power line which allows for by-passing the on board core power supply and inserting a current sensor (resistor or current probe) to measure the power consumption of the DUT FPGA. So far, we have successfully used the Spartan 3E Starter Kit, Spartan 3E-1600 Developer Board, and the Altera DE1 board as FOBOS DUT boards. As the Altera DE1 does not have V_{Core} Jumper, we had to de-solder the voltage regulator for core voltage. On all boards we also removed several capacitors. Our preliminary investigation (shown in Table III) into the other boards have shown that it is possible to modify them in order to measure the current of the core supply. For each DUT board we plan on publishing instructions on how to modify it for DPA and the printed circuit board (PCB) layout of the bridge connector.

TABLE III
FOBOS FPGA DUTs

Board	**FPGA**	**Techn-ology**	**V_{Core} Jumper**	**Cost**
Spartan 3E Starter	Spartan-3E	90 nm	yes	$159
Spartan 3E-1600 Dvlp.	Spartan-3E	90 nm	yes	$225
Altera DE1	Cyclone-II	90 nm	no	$150
Cyclone III Starter	Cyclone-III	65 nm	yes	$199
Genesys Board	Virtex-5	65 nm	no	$449
Altera DE2-115	Cyclone-IV	60 nm	no	$299
Altys Board	Spartan-6	45 nm	no	$199
Altera DE4	Stratix-IV	40 nm	no	$2,995
Xilinx ML605	Virtex-6	40 nm	no	$1,795
Xilinx KC705	Virtex-7	28 nm	no	$1,695

3) FOBOS Control—DUT Protocol: The FOBOS Control-DUT Protocol uses a simple FIFO interface to transfer data to and from the control and DUT FPGAs. The functionality of the input and output ports of the protocol is described in [17], [18]. All data and key to and from the FPGA is broken into segments. The first 2 bytes (16-bit) of each segment is a command word, which decides the nature of the segment and the number of bytes being sent. The format of the 16-bit command words is shown in Fig 3. A '0' value in the LSB and a '0' value in the MSB of the command word indicates that a key is being sent. Similarly a '1' value in the LSB indicates that data is send. The bit in position '1' indicates with a '0' that more segments are following the current one, a '1' indicates that the current segment is the last. This protocol does not require the control board to know what the block size of the cryptographic function is. We will provide a VHDL description of a wrapper that translates our FIFO based protocol with in-band signaling to separate buses for key and data. The widths of theses buses, indicated by 'k' and 'd' in Fig 2, can be defined by the user.

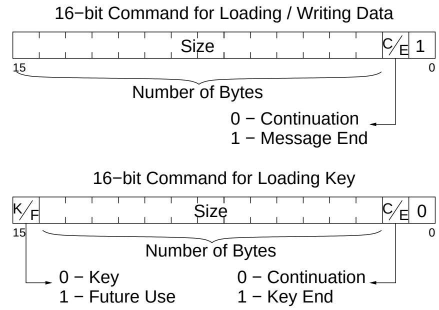

Fig. 3. FOBOS Protocol

B. FOBOS Software

1) FOBOS Software Control Flow:: The FOBOS control flow is shown in Fig. 4. The control script parses the configuration files and initializes the FOBOS environment. It performs a simple tool check to verify whether the necessary library files essential for data transfer and oscilloscope control are installed.The control script then assigns the hardware and oscilloscope attribute values as specified by the user in the configuration files. The FOBOS hardware then performs a built-in self test to check whether all the attributes are set accordingly and issues an appropriate status message to the control script. If the control receives an error code it exits the program displaying a proper error message. On receiving a success code, the control script instructs the oscilloscope to digitize its analog inputs which in turn waits for the trigger signal from the control board to start capturing data. The plaintext and the key are then transferred to the FOBOS hardware through USB and the control script waits until it receives data from the oscilloscope. Once the oscilloscope data is captured, the control script writes the outputs from the FOBOS hardware to a file.

FOBOS has support for two data capturing modes, called *Single Capture* and *Multi Capture* Single Capture mode, as shown in Fig. 5a), assumes that a power trace contains a single encryption whereas in Multi Capture mode, as shown in Fig. 5b), it contains multiple encryptions. Once all data has been captured the control is transferred to data analysis module.

2) FOBOS PC—Control Communication Protocol:: FOBOS uses the command & status registers to control the PC—Control communication. The command register is used (shown in Fig. 2) to pass the option values to the modules inside the control FPGA and to signal the control board that PC is ready to transmit the data. The status register (shown in Fig. 2) on the other hand, is used for signaling the PC that the control FPGA is ready to transmit the data obtained from DUT FPGA or to report errors.

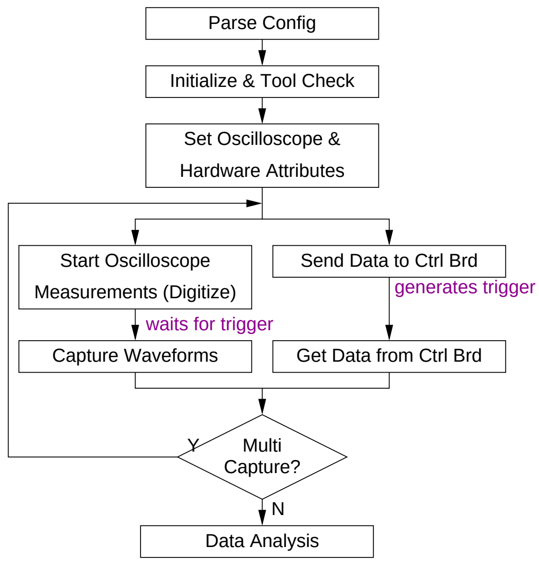

Fig. 4. FOBOS Control Flow

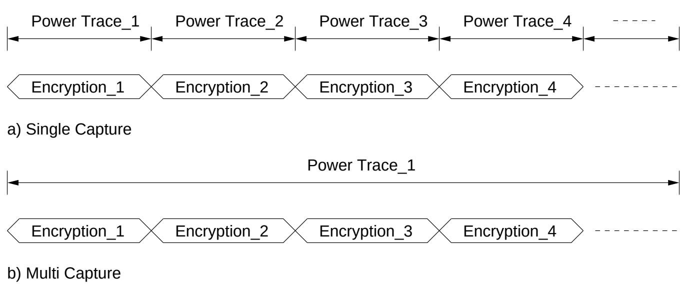

Fig. 5. Capture Modes

3) FOBOS Data Acquisition Module:: The data acquisition module configures the oscilloscope and retrieves its data. Its behavior is determined by a configuration file which uses a generic, oscilloscope brand independent description. A special, oscilloscope dependent sub-module translates the configuration file to commands which are oscilloscope specific. The sub-module of our prototype uses the Virtual Instrument Software Architecture (VISA) library which is a standard for configuring and programming instruments using a variety of interfaces. Presently, the FOBOS prototype supports communication for oscilloscopes from Agilent Technologies. In future we plan to provide support for oscilloscopes from other manufacturers.

FOBOS also supports data acquisition using OpenADC [16]. OpenADC is an low cost open source data acquisition hardware which can digitize signals at 105MS/s using an 10-bit ADC. It also has several features like low noise amplifier with adjustable gain, adjustable phase shift and an external clock input for acquisition and target synchronization. We configured the FOBOS control board to control OpenADC and to capture and send its acquired data to the PC.

4) FOBOS Data Analysis Module:: The Data Analysis module consists of two sub-modules: A raw data processing module, and a DPA attack module. The raw data processing module transforms the raw data obtained from the oscilloscope

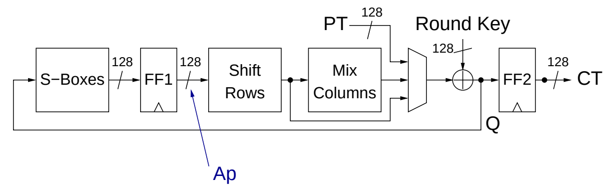

Fig. 6. Block Diagram of the AES Core

to the actual voltage values using the acquisition information returned by the oscilloscope.

The DPA attack module contains a library of the state-of-the art side-channel distinguishers. The user has to generate a hypothetical power model and can choose to test his/her own power model with one or all distinguishers to (try) obtain the secret information. Presently, the FOBOS prototype only supports CPA and Mutual Information analysis as side-channel distinguishers.

V. CPA Attack on AES using FOBOS

This section describes a Correlation Power Analysis (CPA) attack of an implementation of the Advanced Encryption Standard (AES) [19] using FOBOS. AES is an symmetric-key cipher.It applies four different transformations, SubBytes, ShiftRows, MixColumns, and AddRoundKey, per round and iterates through several such rounds depending upon the key size. For each round, an intermediate key called "round key" is derived from the original key through a reversible key scheduling function. We have implemented a basic iterative architecture of AES with 128-bit key length and a 128-bit wide datapath requiring 11 clock cycles for one encryption. Key scheduling is done on-the-fly and the SubBytes function is realized through look-up-tables. The block diagram for this design is shown in Fig. 6.

We attack our AES design during the first round at the output of the register FF1 indicated by Ap in Fig. 6. The equation for calculating the Hamming Distance (HD) between the current value at AP and the previous value is shown in (1). We use Pearson's Correlation to correlate the instantaneous power consumption with the HD model.

$$P_{est.} = \mathrm{HD}(\mathrm{SBOX}(CT_i), \mathrm{SBOX}(k_{guess} \oplus PT_{i+1})) \qquad (1)$$

Figure 7 shows a snippet of the hardware attributes specified in the FOBOS configuration file. FOBOS Control sends data from datain.txt and a key from keyin.txt, which are both in the format of ASCII coded Hexadecimal values, to the DUT. FOBOS Control sets the timeout to 30,000 clock cycles and the trigger to 4 clock cycles after processing starts. The DUT clock is set to run at 500 KHz and the result will be stored in hexadecimal values in the file outputs.txt

A snippet of oscilloscope attributes from osc_config.txt file is shown in Fig. 8. FOBOS control connects to the instrument specified by the VISA address from the RESOURCE attribute. The voltage ranges of the channels of the oscilloscope are specified in terms of vertical full-scale value in volts. The time range of the channels are specified in terms of horizontal full-scale value in seconds. This means 0.0125 Volts/div for channel-1, 2 Volts/div for channel-2, and 0.01 Sec/div. We

```
DATA_FILE = datain.txt
KEY_FILE  = keyin.txt
CLK_FREQ = 500 KHz
TIME_OUT = 30000
TRIGGER = 4
CAPTURE_MODE = multi
```

Fig. 7. Snippet of config.txt

```
RESOURCE  = GPIB0::7::INSTR
CHANNEL_RANGE1 = 0.1V
CHANNEL_RANGE2 = 16V
TIME_RANGE = 0.001
TRIGGER_SOURCE =  CHANNEL2
TRIGGER_MODE =  EDGE
TRIGGER_SLOPE = POSITIVE
```

Fig. 8. Snippet of osc_config.txt

also set the trigger source to be channel-2 and the condition on trigger to be positive edge.

FOBOS control sends the data from the oscilloscope i.e. the power traces, inputs, and outputs to the data analysis module. Algorithm 1 shows the pseudo-code for the data analysis at an abstract level. The first step involves processing the raw power trace using the preamble information to obtain the *measured_power_trace*. The module then calculates *est_power_traces* from the power model described in (1). The

Algorithm 1 Pseudo-code for Data Analysis

Require: Inputs, Outputs, Power Model, Power_trace, Trigger_trace
1: measured_power_trace = process_raw_trace(Power_trace, Trigger_trace, preamble);
2: est_power_traces = get_hyp_trace(Inputs, Outputs, Power Model);
3: **for** key_guess = 00 to FF **do**
4: corrcoef[key_guess]=pearsons(est_power_trace[key_guess], measured_power_trace);
5: **end for**
6: obtained_key = max or min(corrcoef[key_guess]);

CPA attack is conducted on a sub-byte of the key.Hence there are 256 different key guess values and correspondingly 9 different HD values i.e. $0 \rightarrow 8$. The data analysis module than calculates the Pearson's Correlation for all the key guesses by correlating the *est_power_traces* and *mes_power_trace*. The correct key sub-byte will be the extreme outliers in the set of all the correlation value. We repeat the entire process from Step-2 of the Algorithm 1 to recover the remaining key sub-bytes.

The data analysis module also plots two graphs, called the Correlation Plot, which shows how well each individual key guess correlates with the power trace, and the Measurements to Disclosure (MTD) plot, which shows the number of encryption required to disclose the sub-key byte. .

VI. Conclusion

Currently FOBOS is a prototype under development. We hope that our choice of making the complete design open-source, giving the user the option of using Matlab or Octave, and by enabling the use of Xilinx and Altera university program boards, will make a hands-on side-channel attack experience possible for a wider audience. FOBOS is designed to have the flexibility of extending the DUT to ASICs and Smart cards. ASIC DUT boards can be designed to have a socket into which an ASIC chip can be simply plugged-in (similar to SASEBO-R). In order to support evaluating smart cards, a board with a smart card reader along with power measurement circuitry can designed. Both boards should have a connector to easily and securely connect them to the FOBOS control board.

References

[1] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Advances in Cryptology - CRYPTO'99*, ser. Lecture Notes in Computer Science (LNCS), vol. 1666. Berlin: Springer Verlag, Aug 1999, pp. 388–397.

[2] S. Mangard, E. Oswald, and T. Popp, *Power Analysis Attacks, Revealing the Secrets of Smart Cards*. Springer, 2007.

[3] P. C. Kocher, "Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems," in *Advances in Cryptology - CRYPTO 96*, ser. Lecture Notes in Computer Science (LNCS), vol. 1109. Berlin: Springer-Verlag, 1996, pp. 104–113.

[4] K. Gandolfi, C. Mourtel, and F. Olivier, "Electromagnetic analysis: Concrete results," in *Cryptographic Hardware and Embedded Systems–CHES 2001*, ser. Lecture Notes in Computer Science (LNCS), c. K. Koç, D. Naccache, and C. Paar, Eds., vol. 2162. Springer-Verlag, 2001, pp. 251–261.

[5] J.-J. Quisquater and D. Samyde, "Electromagnetic analysis (EMA): Measures and counter-measures for smart cards," in *Smart Card Programming and Security, Proceedings of E-smart*, ser. Lecture Notes in Computer Science (LNCS), vol. 2140. Berlin: Springer-Verlag, 200–210 2001.

[6] J. Anderson and H. M. Heys, "Side channel analysis of cryptographic hardware using SCAB," in *The Seventeenth Newfoundland Electrical and Computer Engineering Conference, NECEC-2007*, Newfoundland, Canada, Nov 2007.

[7] *Side-channel Attack Standard Evaluation Board SASEBO-GII Specification*, Version 1.01 ed., Research Centre for Information Security (RCIS), National Institute of Advanced Industrial Science and Technology (AIST), Nov 2009.

[8] T. Katashita, A. Satoh, K. Kikuchi, H. Nakagawa, and M. Aoyagi, "Evaluation of DPA charactersitics of SASEBO for board level simulations," in *First International Workshop on constructive side channel analysis an secure design*. COSADE, 2010.

[9] A. Satoh, T. Katashita, and H. Sakane, "Secure implementation of cryptographic modules—development of a standard evaluation environment for side channel attacks," *Synthesiology*, vol. 3, no. 1, pp. 56–65, Jul 2010.

[10] M. Tech, "Sakura hardware project," http://www.morita-tech.co.jp/SAKURA/en/index.html.

[11] "DPA contest," http://www.dpacontest.org/home/.

[12] "OpenSCA," http://www.cs.bris.ac.uk/home/eoswald/opensca.html.

[13] "DPA Workstation," http://www.cryptography.com/technology/dpa-workstation.html.

[14] Inspector, "Riscure," http://www.riscure.com/tools/inspector.

[15] L. Judge, M. Cantrell, C. Kendir, and P. Schaumont, "A modular testing environment for implementation attacks," in *Workshop on Redefining and Integrating Security Engineering at ASE/IEEE International Conference on Cyber Security 12 (RISE)*, Dec 2012.

[16] "OpenADC," http://www.newae.com/tiki-index.php?page=OpenADC.

[17] *Hardware Interface of a Secure Hash Algorithm (SHA)*, v. 1.4 ed., Cryptographic Engineering Research Group, George Mason University, Jan 2010.

[18] K. Gaj, E. Homsirikamol, and M. Rogawski, "Fair and comprehensive methodology for comparing hardware performance of fourteen round two SHA-3 candidates using FPGA," in *Cryptographic Hardware and Embedded Systems, CHES 2010*, ser. LNCS, S. Mangard and F.-X. Standaert, Eds., vol. 6225. Springer Berlin / Heidelberg, 2010, pp. 264–278.

[19] *Advanced Encryption Standard (AES)*, National Institute of Standards and Technology (NIST), FIPS Publication 197, Nov 2001, http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.