Dual-hiding side-channel-attack resistant FPGA-based asynchronous-logic AES : design, countermeasures and evaluation

Chong, Kwen-Siong; Ng, Jun-Sheng; Chen, Juncheng; Lwin, Ne Kyaw Zwa; Kyaw, Nay Aung;
Ho, Weng-Geng; Chang, Joseph; Gwee, Bah-Hwee

2021

Chong, K., Ng, J., Chen, J., Lwin, N. K. Z., Kyaw, N. A., Ho, W., Chang, J. & Gwee, B.
(2021). Dual-hiding side-channel-attack resistant FPGA-based asynchronous-logic AES :
design, countermeasures and evaluation. IEEE Journal On Emerging and Selected Topics in
Circuits and Systems, 11(2), 343-356. <https://dx.doi.org/10.1109/JETCAS.2021.3077887>

<https://hdl.handle.net/10356/151198>

<https://doi.org/10.1109/JETCAS.2021.3077887>

© 2021 IEEE. Personal use of this material is permitted. Permission from IEEE must be
obtained for all other uses, in any current or future media, including
reprinting/republishing this material for advertising or promotional purposes, creating new
collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted
component of this work in other works. The published version is available at:
<https://doi.org/10.1109/JETCAS.2021.3077887>

Downloaded on 26 Apr 2023 18:15:21 SGT

Dual-Hiding Side-Channel-Attack Resistant FPGA-Based Asynchronous-Logic AES: Design, Countermeasures and Evaluation

Kwen-Siong Chong, *Senior Member, IEEE*, Jun-Sheng Ng, Juncheng Chen, Ne Kyaw Zwa Lwin, Nay Aung Kyaw, Weng-Geng Ho, Joseph Chang, *Senior Member, IEEE* and Bah-Hwee Gwee, *Senior Member, IEEE*

Abstract—We present a side-channel-attack (SCA) resistant asynchronous-logic (async-logic) Advanced Encryption Standard (AES) accelerator with dual-hiding SCA countermeasures, i.e. the amplitude moderation (vertical dimension) and the time moderation (horizontal dimension). There are five contributions in this paper. First, we propose an async-logic design flow with relative timing to simplify the AES realization in Field-Programmable-Gate-Array (FPGA). Second, we optimize completion detection circuits therein to achieve a low power/overhead solution. Third, we propose a randomized delay-line control and a data-propagation control to amplify the dual-hiding SCA countermeasures for our async-logic AES accelerator. Fourth, we validate the async-logic design flow based on two commercially-available Sakura-X and Arty-A7 FPGA boards. Fifth, we comprehensively evaluate 74 SCA attacking models for our async-logic AES accelerator on these two boards, and compare the results against a benchmarking AES based on synchronous-logic (sync-logic). We show that our async-logic AES accelerator is unbreakable within 1 million electromagnetic (EM) traces where the sync-logic counterpart is breakable within $< 30K$ EM traces. To our best knowledge, our async-logic AES accelerator is the first async-logic AES design evaluated comprehensively at the first/last round, at various attacking locations (i.e. before/after Substitute-Box), and with various Hamming weight/distance, bit model, and zero-model of SCAs.

Index Terms— Advanced Encryption Standard (AES), Asynchronous circuits, Cryptography, Data security, Design methodology, Encryption, Field-Programmable-Gate-Array (FPGA), Side-channel-attack (SCA)

I. INTRODUCTION

INTERNET-of-Things (IoT) systems increasingly demand security features, including adopting encryption such as Advanced Encryption Standard (AES) [1] so that encrypted data can be shared among various parties via internet. The encrypted data would be considered secured even they are unfortunately intercepted by adversaries who may not know the key of AES. However, the hardware implementation of AES may lose its security if the hardware is physically attacked. Side-channel-attack (SCA) [2] is one form of the hardware attacks, by

correlating the encrypted data with the leakage within side-channel information [3] such as power or electromagnetic (EM) information [4]. By constructing a suitable attacking hypothesis model [5] using the encrypted data, SCA on an AES hardware could be surprisingly efficient to reveal the key of AES.

There are many techniques to mitigate SCA. In general, the broad concepts of SCA countermeasures are to apply ‘Masking’ and/or ‘Hiding’ [6] techniques. The former masking countermeasures aim to mask the relationship between the encrypted data and the side-channel leakage. The associated implementations include various masked logic [7] and algorithmic masked transformations [8]. Conversely, the latter hiding countermeasures aim to hide the side-channel leakage due to the circuits therein, not only vertically by controlling the side-channel leakage’s amplitude but also horizontally by adjusting the side-channel leakage’s position (in time). Fig. 1 summarizes the classification on hiding in SCA and their associated implementations; the classification on masking is excluded in this paper. The vertical hiding includes randomizing the leakage, i.e. making the leakage patterns look differently for difficult detection, and uniformizing the leakage, i.e. making the leakage patterns look-alike for difficult analysis. The horizontal hiding is essentially the leakage desynchronization, i.e. making the leakage patterns happen at different times. The vertical and horizontal implementations will be further summarized in Section II later. From Fig. 1, we remark that an asynchronous-logic (async-logic) implementation [9], by executing an encryption hardware asynchronously, could feature both vertical and horizontal hiding (termed as dual-hiding in this paper); the possible classifications/implementation are shown in bold in Fig. 1.

In this paper, we focus on async-logic to enable dual-hiding for an AES accelerator. The realization of async-logic AES accelerator is not new, having several pioneered works reported [10], [11], [12], [13]. However, most reported works only presented their async-logic designs or their async-logic library cells, but did not fully evaluate their SCA resistance against many different attacking hypothesis models. Some other works only analysed the Substitution-Box (S-Box) of AES [14]. Such analyses, unfortunately, were too simplistic. In fact, an async-logic AES accelerator might not naturally mitigate SCA where some works [15] had reported that a reset operation could leak the key in an async-logic S-Box. The reason why the reported async-logic design might not work as expected is similar to that

Kwen-Siong Chong, Jun-Sheng Ng, Juncheng Chen, Ne Kyaw Zwa Lwin, Nay Aung Kyaw, Weng-Geng Ho, Bah-Hwee Gwee and Joseph Chang are with Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798. (e-mails: KSChong@ntu.edu.sg, E190080@e.ntu.edu.sg, JUNCHENG002@e.ntu.edu.sg, KZLNe@ntu.edu.sg, aknay@ntu.edu.sg, WGHo@ntu.edu.sg, ejschang@ntu.edu.sg and ebhgwee@ntu.edu.sg)

Fig. 1: The classification and the associated implementations for SCA hiding countermeasures; dual-hiding async-logic is considered

in a synchronous-logic (sync-logic) design where flip-flops tend to leak the key either at the positive/negative clock edges. In the reported async-logic designs, some internal signals still have to be synchronized before next operation. As a result, the circuits that synchronize the operation could be very leaky.

In this paper, we resurrect async-logic as an emerging and yet practical technology in hardware security, and demonstrate its security applications comprehensively. Our motivations of using async-logic are to allow a fine-grain timing control easily, to enable dual-hiding SCA countermeasures effectively, and to validate the high SCA resistance by a proper use of async-logic. We describe a 128-bit dual-hiding async-logic AES accelerator based on a Xilinx Field-Programmable-Gate-Array (FPGA) platform. We research into an FPGA-based realization mainly for fast prototyping for our targeted IoT system and yet with high SCA resistance. There are five contributions in this paper.

- First, we propose a dual-rail async-logic design flow with relative timing to simplify the AES realization in FPGA; our FPGA-based async-logic AES accelerator is more design-efficient than the reported async-logic designs.
- Second, we optimize completion detection (CD) circuits for FPGA async-logic realization, achieving a power-efficient solution (i.e. $1.3\times$ to $1.64\times$ better).
- Third, we leverage on the async-logic operation to insert a randomized delay-line control and a data-propagation control to amplify the dual-hiding SCA countermeasures for AES, making the SCA evaluations difficult.
- Fourth, we validate our dual-hiding async-logic AES accelerator in two FPGA platforms – one is based on low-noise Sakura-X board [16] appropriate for ‘best-case’ SCA evaluations, and the other one based on the relatively noisier commercial Arty-A7 board [17] appropriate for ‘typical-case’ real-time applications.
- Fifth, to our best knowledge, our async-logic AES accelerator is the first async-logic AES prototype comprehensively evaluated with 74 SCA attacking hypothesis models (under the first-round/last-round attacks with a combination of Hamming distance/weight, bit model,

and zero-model under the correlation/differential EM attacks). The evaluation shows that, even with the best effort to re-align 1 million EM traces, our async-logic AES accelerator is still SCA-resistant. Both from the Sakura-X and Arty-A7 boards, only up to 3 (out of 16) sub-keys can be identified in any attacking hypothesis models. If all attacking hypothesis models are collectively used, only up to 8 (out of 16) sub-keys can be identified. We further show that the effort to evaluate the Arty-A7 board (which is noisier) is more difficult as its dual-hiding countermeasure features are more significant. For completeness, we benchmark our async-logic AES accelerator against a sync-logic AES counterpart which is easily attacked (with all 16 sub-keys broken @ 30K EM traces).

The paper is organized as follows. Section II briefly reviews the SCA countermeasures and async-logic design concept. Section III describes our proposed dual-hiding async-logic AES accelerator in FPGA. Section IV presents the experimental results and comparison. Finally, the paper is concluded in Section V.

II. REVIEW: SCA HIDING COUNTERMEASURES AND ASYNC-LOGIC DESIGN CONCEPT

We will first categorize the SCA hiding countermeasures and then review the async-logic design concept. This section will provide a preamble of why async-logic is potentially advantageous to counteract SCA.

A. SCA Hiding Countermeasures

As depicted in Fig. 1 earlier, let us first look at the vertical hiding where the randomization aims to confuse adversaries by producing different leakage patterns (e.g. power/EM) from encryption to encryption. Some examples include randomly multiplexing two different S-Boxes [18], adjusting process-voltage-temperature (PVT) parameters (e.g. via Dynamic Voltage Frequency Scaling – DVFS) [19], injecting noises [20], controlling the flow of data [21], etc. On the other hand, the

uniformization aims to obscure adversaries by producing similar leakage patterns from encryption to encryption. This can be achieved at the system, module and logic levels. Some examples at the system level include to regulate power/voltage via LDOs/inductive voltage controllers/current equalizers [22], [23], [24], [25] and/or shielding [26]. Some examples at the module level include introducing a power-balanced S-box [27] or replicating AES sub-system, S-Box or other compensators [28], [29] to flatten the power/EM emission. Some examples at the circuit level include adopting various switching-balanced logic styles, including WDDL [30], SABL [31], SDMLp [32], DyCML [33], DDCVLS [34], 3-phase logic [35], and Time Enclosed Logic (TEL) [36], [37].

Now, look at the horizontal hiding which essentially desynchronizes the leakage patterns so that the alignment of the leakage patterns is difficult. A simplistic way is to leverage on the sync-logic approach by randomly shifting the clock, inserting dummy operations, or adjusting the clock delay variations via DVFS. An esoteric way is to use the async-logic approach, either based on the bundled data [38] or self-detection [39], in the sense the delay variation is not periodically clock-sampled but arbitrarily self-timed. To achieve a higher degree of desynchronization, the self-detection approach is preferred as it is innately a fine-grained data-driven PVT-aware adjustment.

Conceptually, DVFS [19], in a coarse-grain manner, embodies both vertical and horizontal hiding features to counteract SCA. The async-logic implementations [10], [11] similarly embody both vertical and horizontal hiding features, but the reported async-logic designs are somewhat not effective to counteract SCA. In this paper, we will present how our proposed async-logic AES accelerator can counteract SCA in FPGA, and explain why we can achieve the effective fine-grain vertical and horizontal hiding.

B. Dual-Rail Async-Logic Design Concept

Fig. 2 depicts a basic dual-rail pipeline architecture [40]. For simplicity, we only depict two pipeline stages, $Stage_i$ and $Stage_{i+1}$, where the data are stored in the dual-rail latches, $Latch_i$ and $Latch_{i+1}$, and detected by the CD circuits, CD_i and CD_{i+1} . The latches and CD circuits are primarily constructed by CMuller gates [39]. At $Stage_i$, there are N -bit dual-rail inputs, M -bit dual-rail outputs, and handshake left and right acknowledge signals $Lack_i$ and $Rack_i$. For ease of understanding, the handshake signals are represented with dotted arrows, and are used to control the latches in the dual-rail data encoding [39] where two wires are used to represent 1-bit information. This dual-rail encoding enables two operation phases, i.e. evaluation phase where two wires are at the opposite logic state, and reset phase where two wires are both '0's (or called NULL data). When $Rack_i$ and $Rack_{i+1}$ are '1' for the evaluation phase, only the valid data can be transferred into $Latch_i$ and $Latch_{i+1}$. When the $Rack_i$ and $Rack_{i+1}$ are '0' for the reset phase, only the NULL data can be transferred into the $Latch_i$ and $Latch_{i+1}$. To generate a logical function, the dual-rail macrocell, which is sandwiched between the latches, is used. The dual-rail macrocell can be any dual-rail circuits [15].

Fig. 2 also labels the number within round/oval shapes to explain the async-logic operation sequences; some sequences (e.g. Sequences 3a and 3b) are performed concurrently. Initially, all the dual-rail signals are NULL, and the handshake signals are '1'. When the inputs become valid (see Sequence 1), $Latch_i$ will store the valid inputs (see Sequence 2). CD_i , consisting of NOR gates and completion tree (CT), will detect the N -bit inputs to be all valid, then $Lack_i$ will be '0' (see Sequence 3a), indicating that all N -bit inputs have already been stored. The dual-rail macrocell will concurrently perform its operation, generating the M -bit outputs (see Sequence 3b) stored at $Latch_{i+1}$ (see Sequence 4b) in $Stage_{i+1}$. CD_{i+1} will detect the outputs to be all valid, $Lack_{i+1}$ ($Rack_i$) will be '0' (see Sequence 5b), indicating that all M -bit outputs have already been stored. When the inputs become NULL (see Sequences 4a-5a), $Latch_i$ will reset the NULL inputs (see Sequence 6a), causing $Lack_i$ to be '1' again (see Sequence 7a) and resetting the dual-rail macrocell (see Sequence 7b). $Rack_{i+1}$ triggering to '0' (see Sequences 5c-7c) will reset $Latch_{i+1}$ (see Sequence 8b) which will further trigger CD_{i+1} to see $Lack_{i+1}$ ($Rack_i$) to '1' (see Sequence 9b). A new operation will be ready when the inputs become valid again.

Fig. 2: Dual-rail async pipeline structure

III. PROPOSED ASYNC-LOGIC AES ACCELERATOR

We now present our proposed dual-hiding async-logic AES accelerator by highlighting four sub-sections, i.e. *A* - Architecture with relative timing, *B* - Design flow, *C* - Circuit optimization, and *D* - Dual-hiding SCA countermeasures. The sub-sections *A* and *B* collectively form one of our main contributions - dual-rail async-logic design flow with relative timing. The sub-sections *C* and *D* delineate our contributions on low power/area implementation, and high SCA resistance.

A. Architecture with Relative Timing

Fig. 3 depicts the architecture of our dual-hiding async-logic AES accelerator. Conceptually, it is globally synchronous and locally asynchronous (GSLA) where the primary inputs and outputs are synchronized by a global clock Clk but the internal signals are operated asynchronously. For simplicity, Fig. 3 only

shows in bold the main primary inputs (i.e. the start signal *KDrdy*, the plaintext *DataIn*, the clock *Clk*, and the key *KeyIn*) and the primary output (i.e. ciphertext *DataOut*); other inputs/outputs are not shown. The primary inputs and outputs are clock-synchronized both for ease of simple integration with other clock-based modules (e.g. a microcontroller) and for a simple SCA evaluation.

To achieve GSLA, our async-logic AES accelerator comprises two main modules, the sync-logic interface and the async-logic AES core. The sync-logic interface is essentially a simple state-machine to control the flip-flops to transfer, multiplex and latch data. On top of that, it has a random seed generator to generate a 32-bit select signal *Sel* for controlling the delay variables which are used to enable horizontal hiding – see later in Section III.D.

The async-logic AES core (see the big dotted box in Fig. 3) is a dual-rail async-logic design. We remark that the async-logic AES core is not Quasi-Delay-Insensitive (QDI) [9] – different from some reported QDI-based AES designs [12]. QDI is one of the async-logic approaches [12] of implementing async-logic circuits. The advantages of QDI include timing-assumption-free synchronization and high robustness toward process-voltage-temperature (PVT) variations. When an async-logic circuit is designed to be QDI, it is able to operate correctly, independent of any gate and wire delays. To achieve the independent delay attribute, QDI circuits need to abide by a strict data signaling protocol to transfer/synchronize data; the timing or data synchronization issues encountered in sync-logic circuits are no longer critical in QDI circuits. The strict data signaling protocol requires all the data signal transitions to be detected/acknowledged. Due to the strict data signaling protocol in QDI [12] for detection/acknowledgment, QDI circuits incur higher power/area overheads. We do not adopt QDI for two reasons. First, we would like to simplify the interface parts for GSLA. Second, we want to relax the timing assumptions so that

we not only optimize the FPGA circuit implementations but also make dual-hiding SCA countermeasures easier to be realized. From Fig. 3, the shaded boxes are the circuits realized in the dual-rail encoding and the unshaded boxes the circuits realized in standard single-rail logic. The dotted arrows represent the handshake signals.

The async-logic AES core first accepts *DataIn* and *KeyIn* and converts them into dual-rail signals via the single-rail to dual-rail converters (S2Ds). The arrival times of the generated dual-rail signals are randomly controlled by the variable delays. Then, the pre-Add Round operation is performed, followed by 10 rounds of repetitive AES operation conducted in Stages 1-3. Each round of AES operation has Look-up-Table (LUT) S-Box, ShiftRow, MixColumn and Add Round operations, save the last round which does not have the MixColumn operation [1]. The Round Constant (Rcon) State Machine generates the Rcon signals for the Key Expansion.

There are three dual-rail macrocells. The first macrocell is for the pre-Add Round operation. The second macrocell is for S-Box, ShiftRow, MixColumn, Add Round, Key Expansion and Multiplex operations. The third macrocell is for Rcon operation. The second and third macrocells are sandwiched by the latches/CD circuits based on the dual-rail pipelines (refer back to Fig. 2). The second and third macrocells need to form async-logic rings for cyclic data propagation [39]. We do this by augmenting latches and CD circuits (e.g. Stages 2-3).

The async-logic AES accelerator is designed to abide by the following timing assumptions. When *DataIn* and *KeyIn* are ready, *KDrdy* is triggered at the positive clock edge, our async-logic AES accelerator will start computing 10 rounds of AES operation asynchronously, and at the 11th clock cycle, *DataOut* will be stored at the flip-flops. Fig. 4 depicts the timing diagram. The 10 clock cycles are set to be the same as that in its sync-logic counterpart. The 10-clock latency depends on how fast the async-logic AES core can generate the internal signal *IntData*;

Fig. 3: Architecture of the proposed async-logic dual-hiding AES accelerator

see at Stage 2 in Fig. 3. Referring to Fig. 4, *IntData* will alternate between the NULL data (black regions) and valid data (as numbered 1-10) for 10 rounds. At the 10th round, *IntData* will remain valid until the 11th clock cycle. The total delay of *IntData* could vary according to various randomized delays, but must be less than the 10-clock duration. The clock counter could be set to other numbers (e.g. < 10), making *DataOut* be available at a different clock cycle, provided that *IntData* has already been computed for 10 rounds of AES operation.

Fig. 4: The timing diagram for the async-logic AES accelerator

We use a relative timing approach [41] to determine when to stop the async-logic operation. This is achieved by detecting the Rcon State Machine for controlling 10 rounds of the AES operation. The Rcon State Machine will generate the hex number sequence 01, 02, 04, 08, 10, 20, 40, 80, 1B and 36 [6]. Each hex number follows the dual-rail encoding having valid and NULL data alternately. When H'36 is generated, the number sequence will stop. Fig. 5 depicts the timing control circuit. There are two circuit paths, one is the CD for allowing the async-logic ring operation to be propagated, and the other one the Stop Detection for stopping the async-logic ring operation. The Stop Detection in part detects in the round constant number of H'36. For the first 9 rounds of the AES operation, only delay path *D1* will be activated. For the last round, both delay paths *D1* and *D2* will be activated but *D2* is relatively faster than *D1* so that the ring operation will be halted, meaning the handshake signaling sequence will be disrupted. This delay assumption can be easily satisfied by using the relative timing approach [41] where the CD has at least an extra 2-input NOR gate delay and 1 2-input CMuller gate delay. We place the CD circuits in *D1* and Stop Detection circuits in *D2* close to one another to minimize the wire delays, and use the Xilinx Vivado timing analyzer tool to check the overall delay paths *D1* and *D2*. For safety reasons, we design the CD and Stop Detection circuits so that *D1* has at least 25% longer delay than *D2*. Once the async-logic operation stops, the generated output will temperately be latched. The ciphertext output will be stored in the flip-flops at the 11th clock, producing *DataOut*.

B. FPGA Design Flow

Fig. 6 depicts the FPGA design flow for realizing our async-logic AES accelerator. There are three phases – the cell preparation phase, design phase, and FPGA realization phase. The cell preparation phase is to build FPGA-synthesizable dual-

rail and CMuller library cells for async-logic operations. We first define generic single-rail functions (e.g. AND, OR, etc.) and then construct their corresponding generic dual-logic functions based on dual-rail logic implementations. For the specific dual-rail logic implementation, we adopt the wave-dynamic-differential logic (WDDL) [30]. We finally create dual-rail FPGA primitive cells to represent WDDL dual-rail cells. Although it is well-reported that WDDL suffers from high SCA leakage [42], we still use WDDL because WDDL is simple and we could leverage on async-logic operations to mitigate the leakage – see later.

Fig. 5: The timing control circuit having two delay paths, *D1* and *D2*. *D2* must be faster than *D1* for proper operation.

Fig. 6: FPGA design flow for the dual-hiding async-logic AES accelerator

In the design phase, it has 4 main steps, including the pipeline architecture, standard synthesis, S2D conversion, and CD setting. We earlier depicted in Fig. 3 our adopted pipeline architecture. We use a structural approach to arrange various latches to form async-logic rings, and the associated macrocells are sandwiched between latches. The arrangement of the latches will determine how the data will flow, including a possibility to include more pipeline stages. In fact, async-logic operation is beneficial to adjust data flow dynamically, and is flexible

enough to shift some functions to other pipeline stages without causing much synchronization disruption. In Fig. 3, our aim is to keep the architecture to have minimum possible hardware resources – we do not consider to use additional pipeline stages as another means to inject noise to counteract SCA.

The three macrocells are first coded in Verilog, and then synthesized using the standard synthesis flow. Thereafter, their netlists are converted into dual-rail netlists. The handshake circuits and variable delays are constructed structurally using the FPGA primitive cells.

The CD setting determines how the CD circuits generate the handshake signals to acknowledge various pipeline stages. This setting is coupled with the delay-lines to enable horizontal hiding. We will present how we optimize the DC circuits in Section III.C, and achieve the SCA hiding in Section III.D.

In the FPGA realization phase, it is to configure the FPGA setting correctly so that the Xilinx Vivado tool can perform a successful synthesis on async-logic circuits. Fig. 7 depicts the constraint and the synthesis options. Particularly, the “set_multicycle_path” option is used to let the tool have a timing check within 10 clock cycles duration. Otherwise, some warning messages happen as the tool fails to analyze an async-logic design. Further, we restrict the tool not to flatten the architecture (and circuit optimization) because we do not want our dual-rail circuits and CD circuits to be further optimized, possibly creating any unwanted timing violations.

```

create_clock -period 41.666 -name lbus_clkn [get_ports lbus_clkn]
...
set_multicycle_path -setup -from [get_pins AES/U1/Kvld_reg/C]
    -to [get_pins AES/U1/dat_reg*/D] 10
set_multicycle_path -hold -from [get_pins AES/U1/Kvld_reg/C]
    -to [get_pins AES/U1/dat_reg*/D] 9
...
synth_design -top CHIP_SASEBO_GIII_AES -part xc7k160tfg676-1
    -flatten_hierarchy none

```

} Constraint file
} Synthesis file

Fig. 7: Constraint and synthesis option

C. Circuit Optimization

We optimize our async-logic AES accelerator based on the Xilinx FPGA chips. As CMuller gates are critical cells for async-logic circuits, we adopt a latch-based synthesizable approach based on the behavioral coding. Figs. 8 (a) and (b)

depict a 2-input CMuller gate and a resettable 2-input CMuller gate respectively. In Fig. 8, the upper diagrams show the Verilog coding and the lower diagrams show the corresponding FPGA implementations using the 2-input LUT cell, 3-input LUT cell and/or the primitive latch, termed as LDCE. For the 2-input CMuller gate, when the inputs A and B are both ‘1’, the output Q is ‘1’. When the inputs A and B are both ‘0’, the output Q is ‘0’. When A and B are at the opposite states, the output Q keeps the previous state. For the resettable 2-input CMuller gate, when NR is ‘0’, the output Q is reset to ‘0’. When NR is ‘1’, the resettable CMuller gate is functionally the same as the 2-input CMuller gate. The resettable 2-input CMuller gate is used for initialization (e.g. in the latches).

Fig. 8: Verilog coding and its corresponding FPGA implementation for (a) 2-input CMuller gate, (b) Resettable 2-input CMuller gate

Because both the key and data have 128 bits that need to be completion-detected, the implementation of the CD plays an important role. We optimize the CD circuits based on different configurations. Figs. 9 (a) to (c) depict three different CD implementations. The CD circuit in Fig. 10 (a) is based on the state-holding approach [39], consisting of a set-logic detecting all ‘1’s and a reset-logic detecting all ‘0’s. Both the set-logic and reset-logic will finally be detected by a 2-input CMuller gate. The CDs in Figs. 9 (b) and (c) are largely based on 4-input CMuller gates and 2-input CMuller gates respectively. The 4-input CMuller is similarly coded as the 2-input CMuller as depicted in Fig. 8 (a). For simplicity, we term the CD circuits in Figs. 9 (a) – (c) as CD-A, CD-B and CD-C respectively.

Fig. 9: Completion Detection (a) CD-A using the state holding approach, (b) CD-B using 4-input CMuller gates, and (c) CD-C using 2-input CMuller gates

Figs. 10 (a) – (c) depict the EM profile (in time-domain across the 10 rounds of operation) of the async-logic AES accelerators embodying CD-A, CD-B and CD-C respectively. As expected, the async-logic AES accelerator embodying CD-C has the largest EM amplitude variations, and those embodying CD-A and CD-B have similar EM amplitude variations. In terms of delay, the async-logic AES accelerator embodying CD-C has the longest delay, followed by that embodying CD-B, and that embodying CD-A has the shortest delay.

Table I further tabulates the normalized gate-count, power and delay of the AES accelerators embodying CD-A, CD-B and CD-C respectively; the results are normalized with respect to the AES accelerator embodying CD-C. From Table I, all AES accelerators have similar gate-count, and that embodying CD-C has the lowest number. In terms of average power dissipation (based on 10,000 AES operations), the AES accelerators embodying CD-A and CD-B is $1.3\times$ and $1.64\times$ more power than that embodying CD-C. In terms of delay (based on the minimum (min) and maximum (max) cases), the delays of the AES accelerators embodying CD-A and CD-B are respectively $0.91\times$ and $0.93\times$ of the AES accelerator embodying CD-C. In view of the higher EM/delay variations for amplitude/time moderation and power-efficiency, we will use the AES accelerator embodying CD-C for our SCA evaluation in Section IV later.

Fig. 10: The EM amplitude/time variations for the async-logic AES accelerators embodying (a) CD-A, (b) CD-B and (c) CD-C

TABLE I
THE NORMALIZED GATE-COUNT, POWER AND DELAY FOR THE ASYNC-LOGIC AES ACCELERATORS EMBODYING CD-A, CD-B AND CD-C

AES Accelerator	Normalized Gate-Count	Normalized AES Power	Normalized Delay (Min/Max)
AES with CD-A	$1.04\times$	$1.30\times$	$0.91\times/0.91\times$
AES with CD-B	$1.02\times$	$1.64\times$	$0.93\times/0.93\times$
AES with CD-C	$1.00\times$	$1.00\times$	$1.00\times/1.00\times$

D. Dual-Hiding SCA Countermeasures

Our async-logic AES accelerator features dual-hiding SCA countermeasures collectively from a number of techniques. From the vertical hiding, it features the noise injection and data-propagation control which collectively moderate the power/EM peaks. From the horizontal hiding, it features async-logic dual-rail operations and randomized data-driven delays which collectively introduce delay jitters.

The SCA countermeasures applied largely relate to the delay-line control, leveraging on async-logic operation to moderate the peaks and timing. The specific SCA countermeasure implementations are further depicted in Figs. 11 (a) – (d). Fig. 11 (a) depicts the 32-bit select (*Sel*) signal circuit to control randomized delays during the async-logic operation. For every first encryption (after reset), *Sel* is generated based on *DataIn* by converting 128-bit data down to 32-bit via XOR operations. This is to mitigate the risk that adversaries try getting the same *Sel*. For subsequent encryption operations, *Sel* is generated based on the previous encryption output.

Fig. 11 (b) depicts the delayed-shifted S2D conversion circuits by generating the dual-rail signals from *DataInR*, *KeyIn* and *InitConf*. *DataInR* is the registered input *DataIn* (from flip-flops). To reduce area/power overheads, we do not keep *KeyIn* in the flip-flops, so *KeyIn* is S2D-converted directly. *InitConf* is the default round constant/control signals which are not needed to be latched, so it is S2D-converted directly. The data dual-rail signals are first triggered by the stored *KDrdyR* signal (i.e. registered from *KDrdy* signal), and then are randomly delayed by the *Sel* signal. In our design, the 32-bit *Sel* signal is randomly applied to different delay-lines therein where each delay-line has up to 32 delay variations (with 5 bits control). Put simply, the data, key and round constant/control signals are randomly delayed. These delay variations moderate the occurrence of peaks at different times.

Fig. 11 (c) depicts another level of control, by controlling 4 reset signals *NRST* to turn on the latches. The reset signals are determined using the completion signal of the Pre-Add Round operation. In this case, the commencement of the latches is random, moderating the operations for the dual-rail macrocells.

Fig. 11 (d) depicts the backward delay controls in Stages 1 and 2, further moderating the timing for each round of AES operations. In Stage 1, we need to handshake both the key and data, so a CMuller gate is needed to combine the completion signals from the key and data. Thereafter, we insert four delay-lines controlled by *Sel*, each randomly generating delay jitters for 4 S-Boxes for async-logic operation. In Stage 2, we use a delay-line controlled just by an internal generated a signal *Cont* so the delay variation is not purely dependent on *SEL*.

The random delay is bounded, having maximum (max) and minimum (min) delays for one encryption. There are two parts causing the max and min delays. The first part is due to the delay-line control circuits. This first part is more critical as a similar (random) delay, large or small, applies to all 10 rounds of AES operation, having an accumulative effect. The second part is caused by the data-dependency of async-logic operation. This second part is less critical because the 10 rounds of AES operation have the average effect; some fast delays in some

rounds could be offset by some small delays in other rounds.

Fig. 11: Delay control: (a) Random seed generator for Sel , (b) the delayed-control S2D conversion circuit, (c) the delayed-control reset signal, and (d) back-forward delay control circuits in Stages 1 and 2.

Figs. 12 (a) and (b) respectively depict the overlapping EM and power traces using 10 encryption operations; one of the traces is bold for easy reference. From Figs. 13, their power and EM traces are largely moderated both in the amplitude and time domains. In other words, it would be increasingly difficult to realign the EM and power traces for effective SCA evaluations. The EM traces are cleaner attributed to a good quality EM probe used (pick-up coil: 1.6mm). The power traces are noisier, meaning that it would be even more difficult to attack the power side-channel (when compared to the EM side-channel). We will use the EM traces for SCA evaluation in Section IV.

IV. EXPERIMENTAL RESULTS

In this section, we first provide the FPGA implementation results for our async-logic AES accelerator followed by detailed SCA evaluations.

A. FPGA Prototyping

We evaluate our async-logic AES accelerator on two different FPGA boards, i.e. the Sakura-X board [16] embodying the Xilinx Kintax-7 chip, and the Arty-A7 board [17] embodying the Xilinx Artix-7 chip. We evaluate these two boards for different noise settings, enabling us to quantify their degrees of SCA resistance. Based on our evaluation, the Sakura-X board is quite clean (less noisy), but the Arty-A7 board is somewhat noisy. In this paper, we term the former as a clean board for the possible best-case scenario, and the latter as a noisy board for a common case scenario – practical scenario. Fig. 13 depicts the SCA platform used both for the clean and noisy boards.

Fig. 12: Overlapping traces of 10 encryptions: (a) EM, and (b) power

Table II tabulates the synthesis results of the async-logic AES accelerator based on the Sakura-X and Arty-A7 boards. For ease of result interpretation, we also include a standard sync-logic design [16] without any SCA countermeasure. The intention here is to help readers to interpolate the power and area overheads for the reported SCA countermeasures [7], [30], [31], [36], [37], [44], applied; these reported countermeasures suffer high overheads, e.g. $3\times$ to $10\times$ more overheads yet without guaranteeing good resistance to SCA. For a fair comparison, both the sync-logic and async-logic AES designs are synthesized based on the same operating conditions. From Table II, our async-logic AES designs, suffer $7.3\times$ - $7.5\times$ more gate counts, these penalties are expected as the FPGA design flow is not intentionally developed for async-logic designs. Area

optimization for async-logic circuits could be possible but such optimization is beyond the scope of this paper; we remark again that one of our main objectives is to evaluate how async-logic circuits are beneficial to counteract SCAs.

Fig. 13: SCA evaluation boards including Sakura-X and Arty-A7

TABLE II
THE SYNTHESIS RESULTS OF SYNC-LOGIC AND ASYNC-LOGIC AES
ACCELERATORS ON THE SAKURA-X AND ARTY-A7 BOARDS

AES Design	Sakura-X Board		Arty-A7 Board	
	Sync-logic	Async-logic	Sync-logic	Async-logic
Gate count	2,340	17,490	2,421	17,628
No. of Clocks	10	10	10	10
Max frequency [#]	29MHz	16MHz	28MHz	22MHz
Power dissipation [*]	30mW	115mW	NA	NA

^{*} Operated at 12MHz [#] Based on the same synthesis condition

All the sync-logic and async-logic AES accelerators take 10 clock cycles to complete one encryption. As we fix the clock @ 12MHz, both the sync-logic and async-logic AES accelerators have the same delay, but they can be clocked at a higher clock rate. The sync-logic AES accelerators can operate up to 28-29MHz, where the async-logic accelerators up to 16-22MHz. For the maximum speed, the delay penalty in the async-logic AES accelerators is largely due to the additional reset (NULL) operations. The maximum frequency of the async-logic AES accelerators is determined by the worst-case delay, i.e. based on the worst-case random delay. The latency delay is preserved if the operating clock frequency is slower than the maximum allowable frequency. It is worthwhile to remark that if a 12MHz clock is fixed, the async-logic AES accelerator can complete the operation within 8 clock cycles in the Sakura-X board and within 6 clock cycles in the Arty-A7 board. But the sync-logic AES accelerators still need to wait for 10 clock cycles. Put simply, for speed-uncritical applications, the async-logic AES accelerator is in fact advantageous for fast latency.

Based on the 10,000 encryptions for the power measurements on the Sakura-X board, the async-logic AES accelerator is 3.8× more power than its sync-logic counterpart. For the Arty-A7 board, we cannot measure the power of both the sync-logic and async-logic AES accelerators independently as there are other interface modules running simultaneously. As a result, we do not include the power reading for the Arty-A7 board. We reasonably expect the power overhead could be within 4× based on the present async-logic AES design architecture. Further power optimization for the async-logic AES accelerator could

be possible but such optimization is beyond the scope of this paper.

B. SCA Evaluations

Table III tabulates our SCA evaluation information and settings. We use up to 1 million (1M) EM traces for each AES design operated at 12MHz; it takes up to 2 days for 1M EM trace measurements for each board. The oscilloscope used is a Lecroy 2.5GS/s oscilloscope, and the EM probe is the MT-545 EM probe. The SCA evaluation software tool is based on the SCA platform in [45]. We consider the attacks at both the first round (FR) and last round (LR) operations, and the attacking locations include both before and after the 8-bit S-Box. We consider the Correlation EM Analysis (CEMA) attack, Differential EM Analysis (DEMA) attack, and Zero-Model attack. Within the CEMA, we consider Hamming Distance (HD) and Hamming Weight (HW) for 8-bit bus, and the Bit Model for 8 different individual bits. For DEMA, we also consider the 8 different individual bits. For each AES design, 74 attacks are evaluated.

TABLE III
THE SCA EVALUATION INFORMATION/SETTINGS

No	Item	Remark
1	Maximum Traces	Up to 1 million EM traces
2	Frequency	12MHz
3	Oscilloscope	2.5GS/s (Lecroy)
4	EM probe	MT-545 (pick-up coil: 1.6mm)
5	SCA software	From async2secure.com [45]
6	First Round (FR)	Locations at both before and after S-Box
7	Last Round (LR)	Locations at both before and after S-Box
8	CEMA @ HD	8-bit bus
9	CEMA @ HW	8-bit bus
10	CEMA @ Bit	Bit 0 to bit 7 (8 cases)
11	DEMA	Bit 0 to bit 7 (8 cases)
12	Zero-Model	When the input of S-Box is zero

We first perform SCA using the original EM traces (i.e. without any re-alignment). Table IV tabulates the reading “Value 1@Value 2” where the Value 1 is the number of sub-keys broken within Value 2 which is the number of EM traces used. In Table IV, for simplicity, we group the FR and LR attack together based on the highest number of sub-keys broken in any of the attacking hypothesis models in either FR or LR attacks. For the Sakura-X board, the sync-logic AES accelerator is vulnerable where CEMA@HD requires only 1.4K EM traces to break 16 (all) sub-keys (at the LR attack). CEMA@HW, CEMA@Bit and DEMA are able to break some (but not all) keys within 1M traces. It appears that Zero-Model is insufficient to break any sub-keys – this is largely attributed to the look-up-table (LUT) S-Box implementation [46] where the switching activities therein when input = 0 is not much different from that using other input combinations. We remark that the SCA attacks the weakest point – once CEMA@HD hypothesis model is able to attack all sub-keys, the entire AES accelerator is considered vulnerable. Other SCA attacking models would be used as reference only. For the async-logic AES accelerator in the Sakura-X board, only one sub-key (out of 16) can be broken with CEMA@Bit.

In the Arty-A7 board, the sync-logic AES accelerator is more SCA resistant, but all the sub-keys can still be broken by using

CEMA@HD with 30K traces (vis-à-vis 1.4K traces or $21\times$ more resistance). The 30K EM traces to break all sub-keys are still considered very ‘poor’ in terms of SCA resistance. Whereas, for the async-logic AES accelerator in the Arty-A7 board using 1M EM traces, 1 sub-key is broken with CEMA@Bit and DEMA@Bit.

TABLE IV
THE SCA EVALUATION ON BOTH SYNC-LOGIC AND ASYNC-LOGIC AES ACCELERATORS (USING ORIGINAL EM TRACES)

No	Model	Sakura-X (Clean)		Arty-A7 (Noisy)	
		Sync-logic	Async-logic	Sync-logic	Async-logic
1	CEMA @ HD	16 @ 1.4K	0 @ 1M	16 @ 30K	0 @ 1M
2	CEMA @ HW	1 @ 1M	0 @ 1M	4 @ 1M	0 @ 1M
3	CEMA @ Bit	2 @ 1M	1 @ 1M	1 @ 1M	1 @ 1M
4	DEMA @ Bit	1 @ 1M	0 @ 1M	1 @ 1M	1 @ 1M
5	Zero Model	0 @ 1M	0 @ 1M	0 @ 1M	0 @ 1M

We further re-align the EM traces, based on our best effort, to attack the async-logic AES accelerators in the Sakura-X and Arty-A7 boards. Figs. 14 (a) and (b) respectively depict the 50 original traces and 50 re-aligned traces within the LR operation for the Sakura-X board. Figs. 15 (a) and (b) depict the same for the Arty-A7 board.

Fig. 14: 50 EM traces on Sakura-X (a) original, and (b) re-aligned

Fig. 15: 50 EM traces on Arty-A7 (a) original, and (b) re-aligned

From Fig. 14 (b), the trace re-alignment in the Sakura-X board is relatively simple as the traces are quite clean. From Fig. 15 (b), the trace re-alignment in the Arty-A7 board is quite difficult as the traces very noisy. We have to try many attempts to find a good reference for the re-alignment. Put simply, the effort spent to re-align the noisy traces is much more difficult than that for the clean traces.

Table V further tabulates the SCA results with the EM trace re-alignments for various attacking hypothesis models from the async-logic AES accelerators embodied in the Sakura-X and Arty-A7 boards. We now separate the FR attack and the LR attack. The attacking locations are before the S-Box and after the S-Box. Based on the attacking results, we make the following comment. First, the async-logic AES accelerators in both boards are reasonably robust in the sense that none of the standalone attacking models is able to break all the sub-keys – a maximum of three sub-keys are broken. Second, because of the nature of the async-logic operation, the CEMA@HD and CEMA@HW, which could be using an 8-bit data for modelling and be effective in sync-logic AES accelerators, are no longer effective. The bit models using CEMA and DEMA could complement the attacks, and in fact we observe that the bit models are more effective to attack the async-logic AES accelerators. Third, it is possible to combine various attacking

TABLE V
THE SCA EVALUATION ON ASYNC-LOGIC AES ACCELERATORS (USING RE-ALIGNED EM TRACES)

No	Round	First Round (FR)				Last Round (LR)			
		Before S-Box		After S-Box		Before S-Box		After S-Box	
		Attack Location		Attack Location		Attack Location		Attack Location	
		Board		Board		Board		Board	
1	CEMA @ HD*	N/A	N/A	0	0	1	1	N/A	N/A
2	CEMA @ HW	0	0	0	0	3	3	0	0
3	CEMA @ Bit0	NA	NA	0	0	1	1	NA	NA
4	CEMA @ Bit1	NA	NA	0	0	0	0	NA	NA
5	CEMA @ Bit2	NA	NA	0	0	3	0	NA	NA
6	CEMA @ Bit3	NA	NA	0	0	0	1	NA	NA
7	CEMA @ Bit4	NA	NA	1	0	3	3	NA	NA
8	CEMA @ Bit5	NA	NA	0	0	0	1	NA	NA
9	CEMA @ Bit6	NA	NA	1	0	2	1	NA	NA
10	CEMA @ Bit7	NA	NA	1	0	0	0	NA	NA
11	DEMA @ Bit0	NA	NA	0	0	1	1	NA	NA
12	DEMA @ Bit1	NA	NA	0	0	0	1	NA	NA
13	DEMA @ Bit2	NA	NA	0	0	3	0	NA	NA
14	DEMA @ Bit3	NA	NA	0	0	0	2	NA	NA
15	DEMA @ Bit4	NA	NA	0	0	1	1	NA	NA
16	DEMA @ Bit5	NA	NA	0	0	1	0	NA	NA
17	DEMA @ Bit6	NA	NA	0	0	0	1	NA	NA
18	DEMA @ Bit7	NA	NA	0	0	0	0	NA	NA
19	Zero Model	NA	NA	0	0	0	0	NA	NA
Total sub-keys [#]		0	0	3	0	8	7	0	0

* The Hamming Distance (HD) model is either $HD(S\text{-}Box(PT), PT)$ at the FR where PT is plaintext or $HD(Inverse\ S\text{-}Box(CT), CT)$ at the LR where CT is a ciphertext

[#] Total sub-keys are the combined number of sub-keys; some broken sub-keys are the same among different models

hypothesis models to collectively attack the async-logic AES accelerators. Nonetheless, at the FR attack, a maximum of 3 sub-keys are broken, whereas at the LR attack, a maximum of 8 sub-keys are broken. We attribute that the trace re-alignment helps better in the LR attack than in the FR attack. Fourth, for the FR attack, the pre-round operation (i.e. at the location before the S-Box) is leakier. However, the 1M EM traces are obviously insufficient to break all sub-key, and the difficulty is further compounded by the re-alignment of traces. Fifth, for the LR attack, at the attacking location before the S-Box operation (i.e. using inverse S-Box modelling) is leakier. Although the async-logic operation has somewhat delay-dependent, its completion detection still in part re-synchronizes some delays for each round of operation. This could explain the LR attack would be riskier than the FR attack. However, the SCA efficacy depends on how well the trace re-alignment can be performed.

A cursory view on Table V alludes to that both the Sakura-X and Arty-A7 boards have similar SCA results where about 7 or 8 sub-keys are collectively are broken. However, we remark that the effort to attack the noisy board is a lot more and difficult. Figs. 16 (a) and (b) respectively depict the correlation result vs the number of EM traces for the Sakura-X and Arty-A7 boards.

Fig. 16: The correlation result vs the number of EM traces (for the 11th sub-key = A7) based on (a) Sakura-X, and (b) Arty-A7

In Figs. 16 (a) and (b), we depict the same broken sub-keys (i.e. sub-key = A7 at 11th sub-key) using CEMA@HD. The correction result for the Sakura-X board is 0.022, whereas that for the Arty-A7 board becomes 0.004, 5.5 $\times$ smaller. For the Sakura-X board, after 50K EM traces, the correlation result of the correct sub-key is higher than that of the other sub-key, and is about 2 $\times$ times higher from 250K EM traces onwards. Conversely, for the Arty-A7 board, the correlation result of the correct sub-key is higher with at least 880K EM traces, and only remains marginally higher up to 1M traces. In other words, the confidence level of detecting the correct sub-key in the Arty-A7 board is much lower than that in the Sakura-X board. The low confidence level of detecting the correct sub-key would also make the “black box” SCA difficult, meaning that the async-logic AES accelerator is more secured.

Table VI lists several reported dual-rail and compensated crypto designs and our proposed AES accelerator (Sakura board), all having hiding countermeasures. A direct SCA resistance comparison among these designs remains contentious due to their different implementation platforms (ASIC vs FPGA where their noises are different), different algorithms (AES, DES, PRESENT, SERPENT) and their SCA analyses (HW, HD and others). For simplicity, we extract their respective measurement-to-disclosure (MTD) improvement, power cost and area cost when each of them is compared against their corresponding unprotected counterparts (as reported in their papers if available). In terms of SCA resistance between the ASIC and FPGA implementations, the ASIC implementations could offer more fine-grained controls, e.g. by using dedicated metal routing [26], timing/capacitance balancing [36], and even customized cells with transistor sizing, to mitigate SCA. Of various ASIC AES designs, the reported design [26] with metal routing greatly limits the EM emanation, hence achieving the best MTD improvement. The FPGA implementations are unfortunately bounded by the dedicated FPGA architecture and its associated synthesis/routing rules, hence possibly limiting SCA resistance. Although the comparison in Table VI is qualitative, our proposed async-logic AES accelerator does feature the highest MTD improvement in FPGA, up to 714 $\times$ more SCA resistant compared to its unprotected counterpart. We further remark that our proposed async-logic AES accelerator is the first async-logic design comprehensively SCA evaluated; the reported async-logic AES designs (i.e. no. 2 [12] and no. 10

TABLE VI
THE COMPARISON ON DUAL-RAIL AND COMPENSATED CIPHER DESIGNS

No	Design by	Platform	Algorithm	Countermeasure	Evaluation Model	MTD Improvement*	Power Cost [†]	Area Cost [†]
1	Bellizia 2018 [36]	ASIC	SERPENT	TEL (iDDPL)	HW only	NA	4.4 $\times$	3.5 $\times$
2	Bouesse 2005 [12]	ASIC	Full AES	Async-logic dual-logic	Not disclosed	NA	N/A	N/A
3	Shan 2020 [29]	ASIC	Full AES	ML HD Compensator	HD only	446 $\times$	1.38 $\times$	1.38 $\times$
4	Hwang 2006 [47]	ASIC	Full AES	Dual-rail	HD only	120 $\times$	2.7 $\times$	2.1 $\times$
5	Singh 2020 [25]	ASIC	Full AES-128	Integrated All-Digital LDO	HD & HW	>125 $\times$	1.32 $\times$	1.37 $\times$
6	Das 2021 [26]	ASIC	Full AES-256	CDSA & lower metal route	HD only	>83k $\times$	1.5 $\times$	1.37 $\times$
7	He 2011 [42]	FPGA	AES S-Box	Precharge-absorbed DPL	Not disclosed	250 $\times$	N/A	1.3 $\times$
8	Soares 2011 [44]	FPGA	Full DES	GALS pipeline (pipe-8)	Not disclosed	8.3 $\times$	N/A	13.5 $\times$
9	Bellizia 2020 [37]	FPGA	PRESENT-80	SC-DDPL	HW only	35 $\times$ - 176 $\times$	2.8 $\times$	10.0 $\times$
10	Kotipali 2014 [10]	FPGA	Full AES	Async-logic NCL	Not disclosed	NA	N/A	N/A
11	This work	FPGA	Full AES	Dual-Hiding async-logic	HD, HW, Bit & Zero Models	714 $\times$	3.8 $\times$	7.4 $\times$

* Measurement-to-disclosure (MTD) improvement – the ratio of the minimum traces between the protected cipher over the baseline (unprotected cipher). For the protected cipher, the minimum trace could be the maximum traces used during the evaluation.

[†] Power cost and area cost are the overheads extracted in their respective papers where their respective baseline designs may not be the same.

[10]) do not have detailed SCA evaluations.

In summary, our dual-hiding async-logic AES accelerator is advantageous to counteract against SCA, and would be more advantageous in the presence of noise which is the typical scenario in a real-life application. Other hiding/masking countermeasures could also be applied to make the async-logic design even more SCA resistant. We further remark that our dual-hiding async-logic approach is also applicable to the ASIC implementation. Such ASIC async-logic implementation could be achieved by building the async-logic cells, and similarly by imposing the timing assumptions for AES operation. Other SCA countermeasure techniques such as metal routing and capacitance balancing could be applied. In fact, the ASIC async-logic implementation could be even easier to achieve more randomized delays (via the delay-line control), hence hindering the synchronization effort during the SCA evaluation. In short, async-logic would feature an emerging and yet practical technology to mitigate hardware security issues.

V. CONCLUSIONS

We have presented the dual-hiding async-logic AES accelerator to counteract SCA. Leveraging on the innate self-timed operation in async-logic circuits, we have proposed the randomized delay-line control and async-logic data-propagation control to achieve the dual-hiding SCA features, i.e. amplitude moderation in the vertical dimension, and time moderation in the horizontal dimension. We have validated our async-logic design flow and the completion detection optimization in Xilinx FPGA platform, and have prototyped the design in two FPGA boards, i.e. Sakura-X board and Arty-7 board. We have comprehensively evaluated our async-logic AES accelerator in both boards and benchmarked the async-logic AES accelerator against its sync-logic counterpart. We have shown that our async-logic AES accelerator is highly SCA resistant, unbreakable with 1 million EM traces. The proposed async-logic AES accelerator occupied ~17K gates, dissipated 115mW@12MHz, 1.2V, and its outputs were synchronized after 10 clock cycles.

ACKNOWLEDGMENT

This research project is supported by the National Research Foundation, Singapore under its National Cybersecurity R&D (NCR) Research Programme in Assuring Hardware Security by Design in Systems on Chip, SOCure (NRF2018NCR-NCR002-001), and NTUtive GAP fund (NGF-2017-03-013).

REFERENCES

- [1] W. Stallings, *Cryptography And Network Security Principles And Practice*, Fifth ed., v139, Pearson Education, 2011.
- [2] P. Kocher, J. Jaffe, B. Jun and P. Rohatgi, "Introduction to differential power analysis," *J. Cryptogr. Eng.*, v1, n1, pp. 5-27, 2011.
- [3] A. A. Pammu *et al.*, "A high throughput and secure authentication-encryption AES-CCM algorithm on asynchronous multicore processor," *IEEE Trans. Inf. Forensics Secur.*, v14, n4, pp. 1023 - 1036, 2018.
- [4] E. T.-Sánchez, J. Castro and A. J. Acosta, "A Methodology for Optimized Design of Secure Differential Logic Gates for DPA Resistant Circuits," *IEEE J. Emer. and Sel. Topics in Circ. and Syst.*, v4, n2, pp. 203-215, Apr. 2014.
- [5] A. A. Pammu, K.-S. Chong, Y. Wang and B.-H. Gwee, "A highly efficient side channel attack with profiling through relevance-learning on physical leakage information," *IEEE Trans. Dependable Secure Comput.*, v16, n3, pp. 376-387, 2019.
- [6] S. Mangard, O. Elisabeth and T. Popp, *Power Analysis Attacks: Revealing the Secret of Smart Cards*, Springer, 2007.
- [7] T. Popp and S. Mangard, "Masked dual-rail pre-charge logic: DPA-resistance without routing constraints," in *Proc. CHES*, 2005, pp. 172-186.
- [8] Y. Wang and Y. Ha, "A Performance and Area Efficient ASIP for Higher-Order DPA-Resistant AES," *IEEE J. Emer. and Sel. Topics in Circ. and Syst.*, v4, n2, pp. 190-202, Apr. 2014.
- [9] K.-L. Chang, J. S. Chang, B.-H. Gwee and K.-S. Chong, "Synchronous-Logic and Asynchronous-Logic 8051 Microcontroller Cores for Realizing the Internet of Things: A Comparative Study on Dynamic Voltage Scaling and Variation Effects," *IEEE J. Emer. and Sel. Topics in Circ. and Syst.*, v3, n1, pp. 23-34, Feb. 2013.
- [10] S. Kotipalli, Y.-B. Kim and M. Choi, "Asynchronous advanced encryption standard hardware with random noise injection for improved side-channel attack resistance," *J. Electrical and Computer Engineering*, v2014, pp. 1-13, 2014.
- [11] D. Shang, *et al.*, "High-security asynchronous circuit implementation of AES," *IEE Proc. - Computers and Digital Techniques*, v153, n2, pp. 71-77, 2006.
- [12] G. F. Bouesse, M. Renaudin, S. Dumont and F. Germain, "DPA on quasi delay insensitive asynchronous circuits: formalization and improvement," in *Design, Automation and Test in Europe*, 2005, pp. 424-429.
- [13] K.-S. Chong *et al.*, "Side-Channel-Attack Resistant Dual-Rail Asynchronous-Logic AES Accelerator Based on Standard Library Cells," *2019 Asian Hardware Oriented Security and Trust Symposium (AsianHOST)*, Xi'an, China, 2019, pp. 1-7.
- [14] J. Wu, Y. Shi and M. Choi, "Measurement and evaluation of power analysis attacks on asynchronous S-Box," *IEEE Trans. Instr. and Meas.*, v61, n10, pp. 2765-2775, 2012.
- [15] K.-S. Chong, *et al.*, "Counteracting differential power analysis: Hiding encrypted data from circuit cells," in *Proc. IEEE Int. Conf. Electron Devices and Solid-State Circuits*, 2015, pp. 297-300.
- [16] Research Center for Information Security, "Power Analysis Attacks on SASEBO," National Institute of Advanced Industrial Science and Technology, 2010.
- [17] Digilent, "Arty A7 Reference Manual," 2020.
- [18] H. Kaul *et al.*, "A 4900×m2 839Mbps side-channel attack resistant AES-128 in 14nm CMOS with heterogeneous Sboxes, linear masked MixColumns and dual-rail key addition," in *Sym. on VLSI Circuits*, Kyoto, 2019, pp. C234-C235.
- [19] A. Singh, *et al.*, "Improved power/EM side-channel attack resistance of 128-Bit AES engines with random fast voltage dithering," *IEEE J. Solid-State Circuits*, v54, n2, pp. 569 - 583, 2019.
- [20] D. Das, S. Maity, S. B. Nasir, S. Ghosh, A. Raychowdhury and S. Sen, "ASNI: Attenuated Signature Noise Injection for Low-Overhead Power Side-Channel Attack Immunity," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 10, pp. 3300-3311, Oct. 2018.
- [21] S. N. Dhanuskodi and D. Holcomb, "Enabling microarchitectural randomization in serialized AES implementations to mitigate side channel susceptibility," in *Proc. IEEE Comp. Society Annual Symp. VLSI*, 2019, pp. 314-319.
- [22] M. Kar, *et al.*, "Reducing power side-channel information leakage of AES engines using fully integrated inductive voltage regulator," *IEEE J. Solid-State Circuits*, v53, n8, pp. 2399 - 2414, 2018.
- [23] C. Tokunaga and D. Blaauw, "Securing Encryption Systems With a Switched Capacitor Current Equalizer," *IEEE J. Solid-State Circuits*, v45, n1, pp. 23 - 31, 2010.
- [24] G. B. Ratanpal, R. D. Williams and T. N. Blalock, "An on-chip signal suppression countermeasure to power analysis attacks," *IEEE Trans. Dependable Secure Comput.*, v1, n3, pp. 179-189, 2004.

- [25] A. Singh et. al., "Enhanced Power and Electromagnetic SCA Resistance of Encryption Engines via a Security-Aware Integrated All-Digital LDO," *IEEE J. Solid-State Circuits*, v55, n2, pp. 478–493, Feb. 2020.
- [26] D. Das, et. al., "EM and Power SCA-Resilient AES-256 Through >350× Current-Domain Signature Attenuation and Local Lower Metal Routing," *IEEE J. Solid-State Circuits*, v56, n1, pp. 136–150, Jan. 2021.
- [27] M. Bucci, L. Giancane, R. Luzzi and A. Trifiletti, "A dynamic and differential CMOS lookup table with data independent power consumption for cryptographic applications on chip cards," *IEEE Trans. Dependable Secure Comput.*, v4, n4, pp. 245–251, 2007.
- [28] X. Li, C. Yang, J. Ma, Y. Liu and S. Yin, "Energy-Efficient Side-Channel Attack Countermeasure With Awareness and Hybrid Configuration Based on It," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 12, pp. 3355–3368, Dec. 2017.
- [29] W. Shan, S. Zhang, J. Xu, M. Lu, L. Shi and J. Yang, "Machine Learning Assisted Side-Channel-Attack Countermeasure and Its Application on a 28-nm AES Circuit," in *IEEE Journal of Solid-State Circuits*, v55, n3, pp. 794–804, March 2020.
- [30] K. Tiri and I. Verbauwhede, "A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation," *Proc. Design, Automation and Test in Europe Conference and Exhibition*, 2004, pp. 246–251.
- [31] K. Tiri, M. Akmal and I. Verbauwhede, "A dynamic and differential CMOS logic with signal independent power consumption to withstand differential power analysis on smart cards," in *Proc. European Solid-State Circuits Conf.*, 2005, pp. 403–406.
- [32] L. N. Ramakrishnan, et. al., "SDMLp: On the use of complementary Pass transistor Logic for design of DPA resistant circuits," in *IEEE Int. Symp. on Hardware-Oriented Security and Trust*, 2012, pp. 31–36.
- [33] M. W. Allam and M. I. Elmasry, "Dynamic current mode logic (DyCML): a new low-power high-performance logic style," *IEEE J. Solid-State Circuits*, v36, n3, pp. 550–558, 2001.
- [34] E. Tena-Sánchez, J. Castro and A. J. Acosta, "Low-power differential logic gates for DPA resistant circuits," in *17th Euromicro Conf. on Digital System Design*, 2014, pp. 671–674.
- [35] M. Bucci, L. Giancane, R. Luzzi and A. Trifiletti, "Three-phase dual-rail pre-charge logic," in *CHES 2006: Cryptographic Hardware and Embedded Systems*, 2006, pp. 232–241.
- [36] D. Bellizia, G. Scotti and A. Trifiletti, "TEL logic style as a countermeasure against side-channel attacks: secure cells library in 65nm CMOS and experimental results," *IEEE Trans. Circuits Syst. I, Reg. Papers*, v65, n11, pp. 3874–3884, 2018.
- [37] D. Bellizia, S. Bongiovanni, M. Olivieri and G. Scotti, "SC-DDPL: A Novel Standard-Cell Based Approach for Counteracting Power Analysis Attacks in the Presence of Unbalanced Routing," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, v67, n7, pp. 2317–2330.
- [38] K.-S. Chong, B.-H. Gwee and J. S. Chang, "Energy-efficient synchronous-logic and asynchronous-logic FFT/IFFT processors," *IEEE J. Solid-State Circuits*, v42, n9, pp. 2034–2045, 2007.
- [39] J. Sparso and F. Steve, *Principles of Asynchronous Circuit Design - A Systems Perspective*. Springer, 2001.
- [40] T. Lin, K.-S. Chong, J. S. Chang, and B.-H. Gwee, "An ultra-low power asynchronous-logic in-situ self-adaptive VDD system for wireless sensor networks," *IEEE J. Solid-State Circuits*, v48, n2, pp. 573–586, Feb. 2013.
- [41] K. S. Steven, R. Ginosar and S. Rotem, "Relative timing [asynchronous design]," *IEEE Trans. Very Large Scale Integration (VLSI) Systems*, v11, n1, pp. 129–140, 2003.
- [42] W. He, E. d. I. Torre and T. Riesgo, "A precharge-absorbed DPL logic for reducing early propagation effects on FPGA implementations," in *Proc. Int. Conf. on Reconfigurable Computing and FPGAs*, 2011, pp. 217–222.
- [43] A. Gornik, A. Moradi, J. Oehm and C. Paar, "A hardware-based countermeasure to reduce side-channel leakage: design, implementation, and evaluation," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, v34, n8, pp. 1308–1319, 2015.

- [44] R. Soares, et. al., "A robust architectural approach for cryptographic algorithms using GALS pipelines," *IEEE Design & Test of Computers*, v28, n5, pp. 62–71, 2011.
- [45] Async2Secure, "SCA Pre-Qualification on AES," [Online]. Available: <http://async2secure.com/services/evaluation>.
- [46] A. Nadjia and A. Mohamed, "Efficient implementation of AES S-box in LUT-6 FPGAs," in *4th Int. Conf. on Electrical Engineering (ICEE)*, Boumerdes, 2015, pp. 1–4.
- [47] D. D. Hwang et al., "AES-Based Security Coprocessor IC in 0.18- μ m CMOS With Resistance to Differential Power Analysis Side-Channel Attacks," in *IEEE Journal of Solid-State Circuits*, v41, n4, pp. 781–792, April 2006.

Kwen-Siong Chong (S'03-M'09-SM'13) received the B.Eng., M.Phil. And Ph.D. degrees in electrical and electronic engineering from Nanyang Technological University (NTU), Singapore, in 2001, 2002, and 2007 respectively.

He is presently a Senior Research Scientist with Temasek Laboratories @ NTU, Singapore. He is/was principal investigator (PI), co-PI, and collaborator of several research projects, including the projects supported from National Science Foundation (Singapore), Defense Advanced Research Projects Agency (USA), Ministry of Education (Singapore), and Public Sector Research Funding (Singapore). He co-founded two start-ups. His research interests include hardware security, space-grade resilient circuits and systems, asynchronous VLSI designs, low-voltage low power VLSI circuits, and signal processing.

Dr. Chong was the Chair of IEEE Circuits and Systems (CAS) Society, Singapore Chapter, in 2017 and 2018. He served as an organizing committee for several conferences, including the ASP-DAC 2014, DSP-2015, DSP-2018, SOCC2019 and ISICAS2021. He has been a member of IEEE CAS Society VLSI Systems and Applications Technical Committee since 2009. He is an IEEE senior member.

Jun-Sheng Ng (S'21) received the B.Eng. degrees in electrical and electronic engineering from Nanyang Technological University (NTU), Singapore, in 2019.

He is presently a Ph.D. student in NTU, Singapore. He is a recipient of NTU Research Scholarship. His research interests include digital VLSI design and hardware security. His current research focuses on side-channel attack and its countermeasures. He is a student member of IEEE.

Juncheng Chen (S'19) received the B.Eng. degree in information engineering from Guangdong University of Technology in 2017. In 2019, he received the joint M.Sc. degree in Integrated Circuit Design from Technical University of Munich (TUM) and Nanyang Technological University (NTU).

He is currently a Ph.D. candidate working in a Hardware Security team in NTU. His research interests include the applications of Machine Learning and Digital Signal Processing in Side Channel Analysis

Ne Kyaw Zwa Lwin received the B.Eng. and M.Sc. degrees in electrical and electronic engineering from Nanyang Technological University (NTU), Singapore, in 2011 and 2014 respectively.

He is currently a Research Associate with School of Electrical and Electronic Engineering, NTU. His current research interests include hardware security, space-grade resilient circuits and systems, and digital VLSI design.

Nay Aung Kyaw received the B.Eng. and M.Sc. degrees in electrical and electronic engineering from Nanyang Technological University (NTU), Singapore, in 2011 and 2015 respectively.

He is currently a Research Associate with School of Electrical and Electronic Engineering, NTU. His current research interests include hardware security, mobile application development and big-data processing.

Weng-Geng Ho (S'10-M'16) received the B.Eng. (Hons.) and Ph.D degrees in electrical and electronic engineering from Nanyang Technological University (NTU), Singapore, in 2009 and 2016 respectively. He was a recipient of the NTU Graduate Research Scholarship.

Dr. Ho is currently a Research Fellow with the Centre for Integrated Circuits and Systems, NTU, Singapore. His current research interests include digital VLSI design, asynchronous-logic circuit

design, NoC-based multicore platform design and low power secured memory design.

Joseph S. Chang received the B.Eng. ECE degree from Monash University, and the Ph.D. degree from the Department of Otolaryngology, Faculty of Medicine, University of Melbourne, Australia. He is currently with Nanyang Technological University (NTU), Singapore, where he is a (full) professor and the former Director of Virtus (IC Design Center of Excellence), member of the NTU's Research Council, was previously the (inaugural) Associate Dean of Research and Graduate Studies at the

College of Engineering, and held several other senior academic/administrative positions. He is an adjunct professor at Texas A&M University, USA, and is serving as an international examiner/assessor for several universities, including highly-ranked universities.

His research is multi-disciplinary and publishes prolifically with over 200 science and engineering publications (including several ranked #1 publications in Google Scholar and some of the most downloaded papers in various fields), over 90 awarded and pending patents, and has licensed over 10 patents to industry, knowhow and technology to industry. He has been awarded numerous academic, defense and industrial grants exceeding \$20M, including from the grant agencies beyond national borders such as Defense Advanced Research Projects Agency (USA) and the E.U., from multinational corporations, and from Singapore's funding agencies.

He has founded six startups, has designed numerous related products adopted for industry and commercially, including obtaining sizable funding from venture capitalists (including Airbus Ventures) and multi \$M industry research and industry contracts.

Dr. Chang served as the Guest Editor for the Proceedings of the IEEE for two special issues and Corresponding Guest Editor for the IEEE J. Emerging and Selected Topics and for the IEEE Circuits and Systems Magazine. He was an Associate Editor of several IEEE publications, including the IEEE Transactions on Circuits and System-I, IEEE Transactions on Circuits and Systems-II, and IEEE Circuits and Systems Magazine, and was a Senior Editor of the IEEE Journal Emerging and Selected Topics, and the Editor of the Open Column of the IEEE Circuits and Systems Magazine. He recently founded the new Flexible Hybrid and Printed Electronics Special Interest Group (CASSFlexible), and was the Chair of the Analog Signal Processing Technical Committee, of the Life Sciences Systems and Applications Technical Committee and of the Biomedical and Life Science Circuits and Systems Technical Committee, all of the IEEE Circuits and Systems Society. He has chaired several international conferences, including the IEEE-National Institutes of Health (NIH) Life Sciences Systems and Applications Workshop, the IEEE-NIH CAS Medical and Environmental Workshops, and the International Symposium on Integrated Circuits and Systems. He was an IEEE Distinguished Lecturer for 2012-2013. He was the keynote/plenary speaker at several major conferences, including the IEEE ICCAS'13, IBBN'14, 2014 IEEE Asynchronous Circuits and Systems Conference, IEEE IECES'15, 2016

IEEE Midwest Symposium on Circuits and Systems, and at the IEEE NEWCAS'18. He is a recipient of numerous technical awards, including the 2018 IEEE Darlington Award, IEEE Outstanding AE Award, IEEE CAS Outstanding Prize, Temasek Lab's Best Publication Award, and several Best Paper Awards at major conferences.

Bah-Hwee Gwee received his BEng (Hons) degree from University of Aberdeen, UK, in 1990. He received his MEng and PhD degrees from Nanyang Technological University (NTU), Singapore, in 1992 and 1998 respectively. He been an Associate Professor in NTU since 2005. He has been working on several research projects including USA DARPA with research grant amounting to more than US\$10m. He has published more than 140 technical papers, 6 patents (3 granted in USA) and 2 Start-up Companies

in 2005 and 2020.

Dr Gwee was the Chairman of IEEE Circuits and Systems (CAS) Singapore Chapter in 2005, 2006, 2013 and 2016. He was the Chairman of CASS DSP Technical Committee (2019-2020). He was the General Co-Chair of IEEE DSP 2018 and IEEE SOCC 2019. He had presented keynotes in IEEE PAINÉ 2020 and IEEE APCCAS 2020. He has also served as Associate Editors of several journals, including IEEE CAS Magazine (2021-2022) IEEE T-CAS II (2010-2011, 2018-2019 and 2020-2021) and IEEE T-CAS I (2012-2013). He was awarded Temasek Laboratories @ NTU Best Publication Award in 2012, NTU Teaching Excellence Award in 2013 and Singapore Defence Technology Prize in 2016. He was an IEEE Distinguished Lecturer in 2009-2010 and in 2017-2018.

His current research interests include hardware security, side-channel attacks and countermeasures, machine learning image processing, asynchronous-logic circuit design, and camouflage IC design.