Investigation of DPA Resistance of Block RAMs in Cryptographic Implementations on FPGAs

Shaunak Shah
Corsec Security, Inc
Fairfax, VA, USA
Email: sshah@corsec.com

Rajesh Velegalati, Jens-Peter Kaps, David Hwang
ECE Department, George Mason University
Fairfax, VA, USA
Email: rvelegal, jkaps@gmu.edu, david.hwang@sloan.mit.edu

Abstract—**Security at low cost is an important factor for cryptographic hardware implementations. Unfortunately, the security of cryptographic implementations is threatened by Side Channel Analysis (SCA). SCA attempts to discover the secret key of a device by exploiting implementation characteristics and bypassing the algorithm's mathematical security. Differential Power Analysis (DPA) is a type of SCA, which exploits the device's power consumption characteristics. Several countermeasures to DPA have been proposed, however, all of them increase security at the cost of increased area which in-turn leads to increased power consumption and reduced throughput. FPGAs are popular due to their reconfigurability, lower development cost, off-the-shelf availability and shorter time to market. Block RAMs (BRAM) are large memories in FPGAs that are commonly used as ROM, FIFO, Look-up tables, etc. In this paper we explore the DPA resistance of BRAMs in Xilinx FPGAs and verify if their usage can improve the security. The results of our Advanced Encryption Standard (AES) implementations show that using BRAMs alone can improve the security over a look-up table (LUT) only design 9 times. Applying Separated Dynamic Differential Logic (SDDL) for FPGAs, a countermeasure against DPA, to this design doubles the security again leading to an 18 fold increase over the unprotected LUT design.**

Index Terms—**Cryptography, Differential Power Analysis, Block RAMs, Side Channel Analysis, SDDL, Xilinx FPGA.**

I. Introduction

FPGAs are preferred over custom design chips due to their programmability, off-the-shelf availability and lower development cost. In addition to combinational logic slices, many FPGAs also contain additional resources such as multipliers, memory blocks, etc. In Xilinx FPGAs these memory blocks are known as Block RAMs (BRAMs) and consist of fast static SRAM cells. Each BRAM in Spartan 3 devices can store up to 18,432 (18K) bits of data.

It is common practice to tradeoff slice area with BRAMs. For example in 2003, Chodowiec and Gaj [1] demonstrated a compact 32-bit datapath architecture for AES-128 utilizing only 222 slices and 3 BRAMs. Following them in 2004 Rouvroy et al. [2] used a similar concept and achieved better results utilizing 163 slices and 3 BRAMs. Chaves et al. [3] demonstrates an efficient use of BRAMs in a reconfigurable memory based co-processor. In [4] Chang et al. explain the use of BRAMs to save on area and implements an AES with 8 bit datapath using only 130 slices and 4 BRAMs. Drimer et al. [5] shows an efficient method to maximize the use of BRAMs and other elements while minimizing the use of LUTs, implementing a 32-bit AES T-Box design using only 93 slices with 2 BRAMs and 4 DSP blocks. These implementations clearly demonstrate that careful usage of BRAMs leads to drastic reduction of slice area consumption.

In 1998, Kocher et al. introduced a powerful technique for cryptanalysis called Differential Power Analysis (DPA) [6]. DPA attempts to recover the secret key used in a device, by correlating the device's instantaneous power consumption with the data being processed. Hiding techniques, a countermeasure against DPA, focus on equalizing the power consumption of each operation, thus hiding the correlation between data and power consumption. Examples of hiding are Simple Dynamic Differential Logic (SDDL) [7], Wave Dynamic Differential Logic (WDDL) [7] [8], to name a few.

Tiri et al. [7] compared area utilization and critical path delay of three algorithms namely Kasumi, Data Encryption Standard (DES) and Advanced Encryption Standard (AES) for an unprotected (single ended) design and a WDDL implementation. The results show an increase in area consumption (in terms of gates) for the WDDL implementations compared to single ended designs by a factor ranging from 3.2 to 3.6 with a small increase in critical path delay. Yu et al. implemented their prototype design on FPGAs in [9] and their results show an increase in slice count for WDDL and DWDDL implementations compared to single ended designs by a factor of 5.8 and 11.6 respectively. Thus gain in security is achieved at the cost of drastic increase in area consumption and critical path delay. A first step to lower the area overhead is the development of SDDL for FPGAs [10] and partial SDDL [11] which lead to an area increase of factor 3.1 and 2.3 respectively for an AES design.

Several features of BRAMs indicate that their use might lead to implementations that are more resistant to DPA attacks than implementations using other resources such as LUTs or Distributed RAMs. One such feature is that BRAMs are glitch free. Glitches are unexpected output transitions due to hazards, resulting from combinational logic gates delays and routing delays. Therefore, glitches are data dependent and influence the dynamic power consumption. This results in information leakage which can be exploited by DPA. Implementations of the AES 8x8 S-Box (table with 256 8-bit entries) using LUTs occupy nearly 64 slices, similarly an 8x32 T-Box (table with 256 32-bit entries) occupies about 256 slices. Both implementations exhibit glitches. On the other hand, two of these S-Boxes or even T-Boxes can fit in one single BRAM. Therefore, it is common practice to use BRAM implementations in order to conserve slice area. This area reduction leads to less utilization of corresponding routing resources, which eventually helps avoiding glitches. Furthermore, BRAM cells do not have any combinational path from address to the output, hence they don't propagate glitches. In addition the output ports are latched with a self-timed circuit providing glitch-free read operations.

The second feature is that BRAMs consist of fast static SRAM cells. Konur et al. [12] explain in detail the structure of an SRAM cell, its operation and leakage power consumption during memory read and write operations. After preforming various experiments, they concluded that power consumed by SRAM cells during memory read operations remains almost the same, irrespective of reading '0' or '1'. Therefore, using a BRAM as a ROM should provide higher security against power analysis attacks than using combinational logic.

II. Attack Methodology

A. Experiment Designs

We implemented several designs varying the utilizations of LUTs, Distributed RAMs and BRAMs. Our designs are divided into two groups, small scale implementations (Test Design) and real world implementations (AES-128 cipher). The advantage of small scale implementations is that they are very easy to control, manipulate and analyze. Our Test Design is similar to designs used by Yu et al. [9] and Velegalati et al. [10] for WDDL and SDDL countermeasures respectively. The Test Design incorporates essential components of the block cipher AES. The real world implementations are of the AES-128 bit cipher [13] with a standard S-Box design and a T-box design. Analysis on larger-scale/real-world implementations allows us to relate and confirm whether the use of BRAMs leads to more DPA resistant implementation.

B. Measurement Setup and DPA Attack

The experiments were performed using a Xilinx Spartan 3E starter kit with a XC3S500eFG320-4 FPGA. An external power supply was used to power the FPGA core. Power consumption was measured using Tektronics CT-1 current probe and an Agilent DSO6054A oscilloscope, which has a bandwidth of 500MHz and can record samples up to 4GSa/sec. We applied an input clock frequency between 100KHz-500KHz.

We performed DPA attacks on all the designs mentioned in following sections based on their corresponding hamming distance model. We have used Pearson product-moment correlation coefficient, commonly known as Pearson's correlation to correlate instantaneous power consumption with hamming distance model [14].

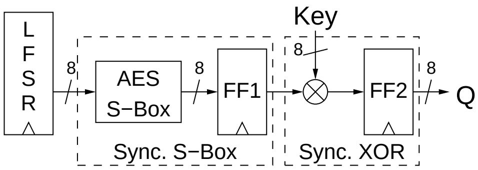


Fig. 1. Block Diagram of Test Design

III. Small Scale Design (Test Design)

The Test Design circuit consists of a synchronous (Sync.) S-Box whose input is connected to an 8-bit LFSR and output is XORed with an 8-bit Key. The result is stored in register FF2. The block diagram of this circuit is shown in Fig. 1. A Sync. S-Box is an S-Box followed by a register. It can be implemented using look-up tables and a register, Distributed RAMs or a BRAM. The later two options absorb the register. A Sync. XOR is an XOR gate followed by a register. In order to implement a Sync. XOR using BRAMs, the logic gate is replaced by a precomputed look-up table. We attack the design at the output of the LFSR. The hamming distance equation for this attack on single ended implementations is shown in Eq. (1) and for SDDL implementations in Eq. (2).

$$P_{est.} = \mathrm{HD}(lfsr_{(i-1)}, \mathrm{SBOX}^{-1}(k_{guess} \oplus Q_i)) \qquad (1)$$

$$P_{est.} = \mathrm{HD}(0x00, \mathrm{SBOX}^{-1}(k_{guess} \oplus Q_i)) \qquad (2)$$

A. Basic Test Design Circuits

We implemented three versions of the Test Design: (1.) S-Box and XOR in LUTs, (2.) explores the use of Distributed RAMs, and (3.) explores the use of BRAMs for (a) S-Box, (b) XOR , or (c) both. The post place-and-route results of these seven implementations are summarized in the top half of Table I.

The best results for minimum area are achieved by design (3a) and (3c). Both designs implement the S-Box in a BRAM, thus resulting in slice area reduction by over a factor of 5 compared to LUT implementations. Implementing the XOR in BRAM does not lead to any significant reduction in slice count because 8 XORs consume only 4 slices. Design (3a) and (3c) are also the fastest designs. The critical path of a LUT based S-Box implementation consist of multiple LUTs and corresponding connections, leading to a slower design.

The security of a design against DPA attacks is determined by the number of measurements required to recover the key, also known as *Measurements To Disclosure* (MTD). We count one encryption as one measurement independent of the number of samples the oscilloscope takes during one encryption. The MTDs shown in Table I indicate the maximum values obained from several independent experiments.

It is clearly visible that S-Box in BRAM implementations (3a) and (3c) have about 26 times higher MTD compared to S-Box in LUTs, hence they provide an increased resistance against DPA. It is an important point to note that implementing only XOR in BRAM does not increase the security of the design.

TABLE I
Implementation Results of Basic Test Design

Design	Slices	FFs	4 input LUTs	BRAMs	Minimum Delay	MTD
1. S-Box and XOR in **LUTs**	85	24	161	0	7.737 ns	> 456
2a. Only S-Box in **Distributed RAMs**	95	23	191	0	7.727 ns	> 256
2b. Only XOR in **Distributed RAMs**	117	32	219	0	9.115 ns	> 256
2c. S-Box and XOR in **Distributed RAMs**	128	32	247	0	6.695 ns	> 256
3a. Only S-Box in **BRAM**	16	16	29	1	5.710 ns	$> 13,000$
3b. Only XOR in **BRAM**	81	16	157	1	8.350 ns	> 256
3c. S-Box and XOR in **BRAMs**	12	8	25	2	5.569 ns	$> 13,000$
4a. Duplicate of Design (3a.)	26	32	41	2	5.710 ns	$> 6,000$
4b. Triplicate of Design (3a.)	35	48	53	3	5.710 ns	$> 3,000$
4c. Quadruplicate of Design (3a.)	44	64	65	4	5.710 ns	> 500
4d. Design (3a.) with dummy circuit	101	32	194	1	6.839 ns	$> 14,000$
5a. **SDDL** of S-Box and XOR in **LUTs** (1.)	283	64	502	0	18.160 ns	$> 10,000$
5b. **SDDL** of only S-Box in **BRAM** (3a.)	24	64	32	2	19.120 ns	$> 25,000$

B. Duplicate Test Design Circuits

The results from basic design implementations show that designs (3a) and (3c) have best resistance against DPA, occupy least slice area and have minimum critical path delay. Hence, an argument can be made that the low area consumption which leads to lower dynamic power consumption might also lead to a higher DPA resistance. Therefore, in order to verify this claim, we increase the area consumption through replication of the design (3a). Design (4a), (4b) and (4c) are duplication, triplication and quadruplication of design (3a) respectively, as shown in Table I. A slight variation is observed in the slice area because the control logic of design (3a) is not replicated. From the implementation results shown in Table I, we observe drastic reduction in MTD due to increase in related logic. However, replicating design (3a) creates an unfair scenario. It increases the signal strength along with data dependent power consumption by applying the same data to multiple inputs.

In order to cross-verify these results we created design (4d). In this design, we increased the area consumption of design (3a) by adding a LUT based S-Box whose inputs are connected to an LFSR with different feedback coefficients. Hence this LFSR produces data which is independent from the original design. The results of this design in Table I show that increasing the area, and with it the total dynamic power consumption, leads to an only marginally increased DPA resistance compared to design (3a). This confirms that the resistance to DPA does not depend on the total amount of power consumption but on the power consumed by related logic.

C. SDDL for FPGAs on Test Designs

We implemented a countermeasure proposed in [10] against DPA called Separated Dynamic and Differential Logic for FPGAs (SDDL for FPGAs) on the test designs (1) and (3a) as (5a) and (5b). SDDL for FPGAs eliminates the correlation between the data being processed and the power consumption of the circuit by ensuring a constant power consumption per clock cycle. This is achieved through duplication of the original circuit into direct and complementary parts. During one half of the clock cycle the inputs to the circuit and the outputs of LUTs and memory elements in the circuit are pre-charged to logic ’0’. During the next half of the clock cycle the original computations takes place. Thus guaranteeing constant switching activity per clock cycle. We followed the design flow described in [10] [11] to implement pre-charge, duplicate and complement of the logic circuits in the test designs with the exception of BRAMs.

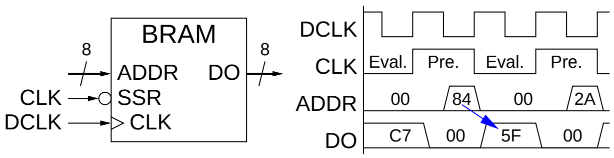


Fig. 2. Pre-charged Block RAM for SDDL on FPGAs

The pre-charge circuit for BRAMs is implemented using the circuit shown in Fig 2 which is based on [15]. The outputs of the BRAMs are synchronously cleared to logic ’0’ by connecting the clock of the circuit to the set/reset (SSR) inputs if the BRAM. We operate BRAMs at twice the clock frequency compared to the rest of the circuit in order to compensate for the clock cycle delay indicated by the arrow in the wave form of Fig. 2. The contents of the duplicated BRAM is computed using Eq. (3).

$$\mathrm{BRAM}_{\mathrm{duplicate}}(addr) = \overline{\mathrm{BRAM}_{\mathrm{original}}\left(\overline{addr}\right)} \qquad (3)$$

The last rows of Table I shows the results for designs (5a) and (5b). The results for the design (5a.) are from [10]. We observe that implementing SDDL on design (1) increases the MTD by 22 times. This is less than the improvement achieved by implementing the S-Box in BRAM (3a). Also the slice area for (5a) is more than 3 times larger than (1) due to circuit duplication and the addition of pre-charge logic. However, SDDL can also be applied to the design (3a) which approximately doubles its MTD. The number of BRAMs doubles but the number of slices is less than doubled. Compared to design (5a), (5b.) improves the MTDs by a factor

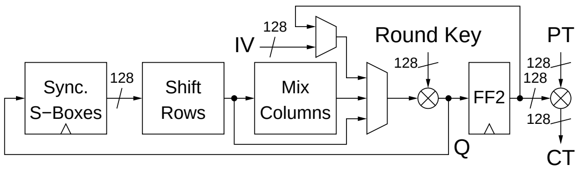


Fig. 3. Block Diagram of AES-128 test Circuit

2.5 and needs less than $1/10^{\text{th}}$ the slice area.

IV. AES 128-bit Implementation

Advanced Encryption Standard (AES) [13] is an iterative block cipher that applies a round function several times. The round function consists of four different transformations: SubBytes, ShiftRows, MixColumns and AddRoundKey. Each round uses an intermediate key called "round key" which is derived from the original key through key scheduling.

We have implemented the AES cipher with 128-bit key length and 128-bit wide datapath. It is an encryption only design with on-the-fly key scheduling and it requires 11 clock cycles for one encryption. The SubBytes function is realized through 16 Sync. S-Boxes. The cipher is implemented in Output Feedback (OFB) mode and can therefore generate new outputs without requiring new plaintext. This enables us to easily collect multiple power samples for DPA. The block diagram for this design is shown in Fig. 3.

We attack the design for one byte of the key at a time at the output of the Sync. S-Boxes, after the last round of encryption. The equation for calculating the Hamming Distance is shown in Eq. (4) where Q_{11} is the output of the last round which XORed with the plaintext PT to produce the ciphertext CT. Because of the OFB mode, the output of the last round Q_{11} is the same as the input to the next round Q_1'. The last round does not use MixColumns.

$$\begin{aligned} P_{est.} &= \text{HD}(\text{SBOX}(CT \oplus PT), \text{SBOX}(k_{guess} \oplus Q_1')) \\ P_{est.} &= \text{HD}(\text{SBOX}(Q_{11}), \text{SBOX}(k_{guess} \oplus Q_{11})) \end{aligned} \tag{4}$$

$$P_{est.} = \text{HD}(0x00, \text{SBOX}(k_{guess} \oplus Q_{11})) \tag{5}$$

From the basic Test Design circuits results we observe that LUT based implementations and Distributed RAM based implementations have similar results. The reason is that the distributed RAMs and LUTs are sharing the same chip resources. An AES design using LUTs only does not fit on our test chip, however, a design using Distributed RAMs does. Therefore, we compare implementations using Distributed RAMs, BRAMs, and SDDL.

A. Standard S-Box Implementation using Distributed RAMs

In design (6a.), the S-Box is implemented using distributed RAMs. The design uses 20 S-Boxes each of size 2^8x8 (2K bits), 16 S-Boxes performing the SubBytes operation on 16 bytes of data simultaneously and 4 S-Boxes performing the SubBytes operation in the key scheduling section.

B. Standard S-Box Implementation using Block RAMs

In design (6b), the Sync. S-Box is implemented using BRAM. We implemented 20 S-Boxes using 10 partially filled BRAMs in dual port mode. This reduces slice area by approximately $20 \cdot (64 \text{ slices}) = 1280$ slices.

C. T-box Implementation using Block RAMs

This design was proposed by the authors of AES in [16] for software implementations on 32-bit micro-processors. In 2001, Fischer demonstrated its hardware implementation in [17]. The T-box design computes one complete AES round just by using look-up tables followed by a large XOR network. The SubBytes operation and MixColumns operation are reformulated and implemented as 8x32 look-up tables. The T-box operation with the T-box equations is explained in [16] and [18]. In design (6c.), the 20 T-Boxes are implemented using 10 completely filled BRAMs.

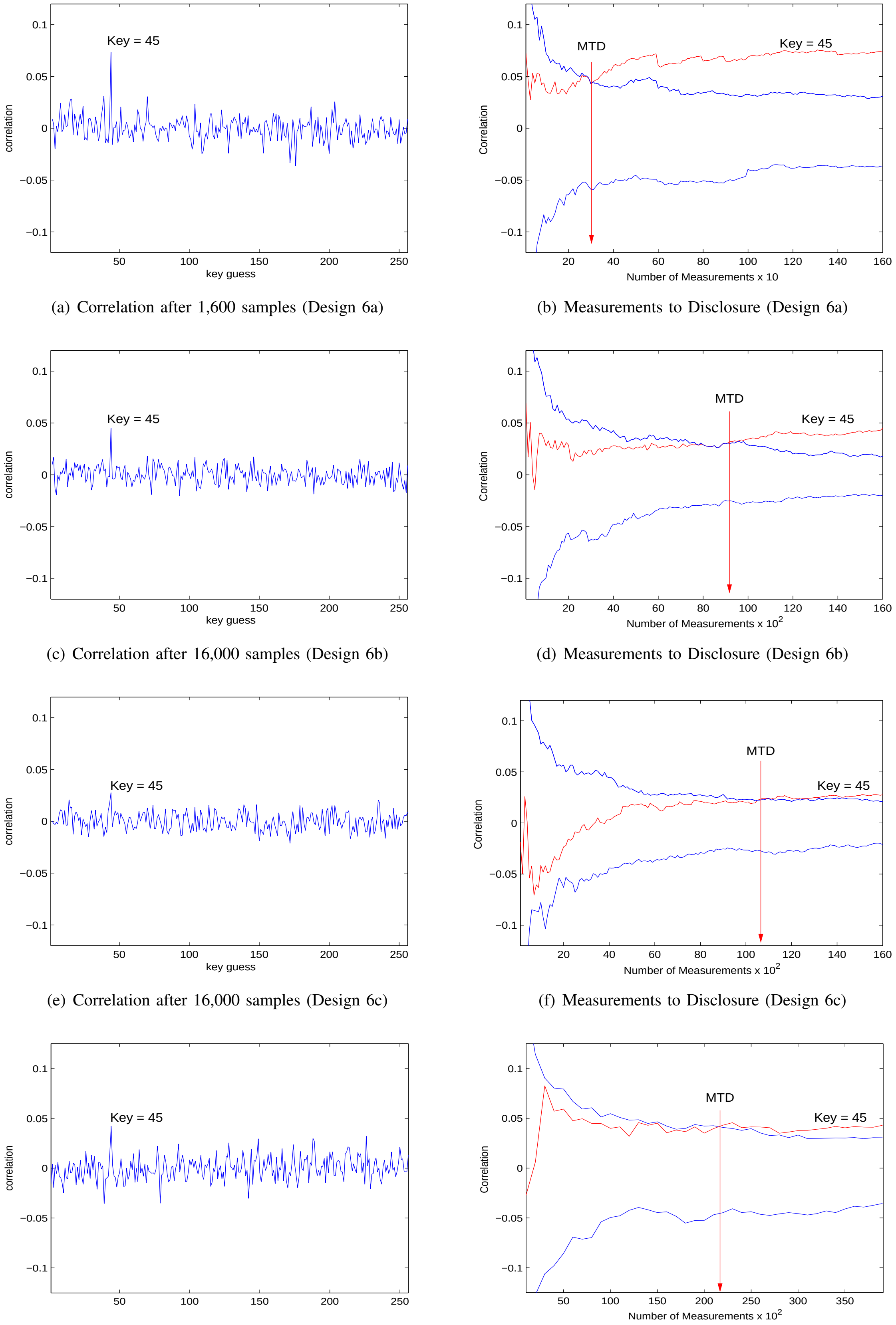
D. SDDL for FPGAs on T-box Implementation

Table II shows clearly that AES T-box implementation (6c) is more secure compared to standard AES-128 implementations (6a)(6b). Hence we implemented SDDL for FPGAs countermeasure on the T-box implementation as design (6d). Design (6d.) consumes approximately 3.2 times the slice area and twice the number of BRAMs compared to design (6c.). We use Eq.(5) to calculate the Hamming Distance for the SDDL design.

V. Results and Conclusions

The post place-and-route implementation results for all 4 AES designs are shown in Table II and the corresponding correlation and MTD plots are shown in Fig. 4. The results are similar to the corresponding Test Design implementations (Table I) which confirms that our Test Designs are an appropriate approximation of a large scale design. The unprotected/single ended (SE) AES implementations which are utilizing BRAMs (6b)(6c) have an approx. 9 times higher MTD than the Distributed RAM based design (6a). At the same time, their slice consumption is 4 times lower at the cost of 10 BRAMs. Applying SDDL to design (6c) doubled the MTD at the expense of 20 BRAMs and a 3 times higher slice count. Therefore (6d) has an 18 fold increase in security over (6a) and uses fewer slices.

Table III shows a comparison of our results with other published secure implementations on FPGAs. R.P. McEvoy et al. [20] achieves a similar increase in security with iWDDL over their SE design compared to our results at the cost of more than 4 times increase in area. iWDDL is deeply pipelined, hence the delay is smaller than of their SE design. Kaps et al. [11] AES design has an 8-bit datapath and a low MTD. However their security increase for Partial SDDL as well as SDDL for FPGAs is similar to that of the designs presented in this paper. Their area increase is 2.3 to 3.1 times. Nassar et al. [19] WDDL implementation of AES is 4.5 times bigger than their SE design. The delay doubles, which is typical for WDDL and SDDL implementations. Their BCDL



(a) Correlation after 1,600 samples (Design 6a)

(b) Measurements to Disclosure (Design 6a)

(c) Correlation after 16,000 samples (Design 6b)

(d) Measurements to Disclosure (Design 6b)

(e) Correlation after 16,000 samples (Design 6c)

(f) Measurements to Disclosure (Design 6c)

(g) Correlation after 40,000 samples (Design 6d)

(h) Measurements to Disclosure (Design 6d)

Fig. 4. Correlation for AES Designs: (a),(b) for Design 6a, (c),(d) for Design 6b, (e),(f) for Design 6c, and (g),(h) for Design 6d

TABLE II
SUMMARY OF IMPLEMENTATION RESULTS FROM AES-128 DESIGNS

Design	Slices	FFs	4 input LUTs	BRAMs	Minimum Delay	MTD
6a. Standard AES-128, Distributed RAMs	1,727	422	3,374	0	15.876 ns	300-1,300
6b. Standard AES-128, BRAMs	412	262	787	10	13.875 ns	$> 9,500$
6c. T-box AES-128, BRAMs	370	262	705	10	13.461 ns	$> 11,500$
6d. SDDL of T-box AES-128, BRAMs (6c.)	1,236	518	1,410	20	26.922 ns	$> 23,000$

TABLE III
COMPARISON WITH OTHER PUBLISHED RESULTS

Authors	Technology	Implementations	Area	Minimum Delay	MTD	Security Gain
Nassar et al.	Stratix II	SE	1,078 ALMs + 40 Kb RAM	13.91 ns	8,000	1
[19]	AES-128	WDDL	4,885 ALMs	26.97 ns	$> 150,000$	20
		BCDL	1,841 ALMs + 160 Kb RAM	19.74 ns	$> 150,000$	20
R.P. McEvoy et al.	Spartan 3E	SE	761 Slices	19.50 ns	1,000	1
[20]	Whirlpool	iWDDL	3,300 Slices	7.70 ns	$> 20,000$	20
Kaps et al.	Spartan 3E	SE	393 Slices	—	500	1
[11]	AES-128	Partial DDL	928 Slices	—	$> 12,000$	24
		SDDL for FPGAs	1,222 Slices	—	$> 12,000$	24
This Paper	Spartan 3E	S-Box Distributed RAM	1,727 Slices	15.88 ns	300- 1,300	1
	AES-128	S-Box BRAM	412 Slices + 10 BRAMs	13.88 ns	$> 9,500$	7
		Tbox BRAM	370 Slices + 10 BRAMs	13.46 ns	$> 11,500$	9
		SDDL Tbox BRAM	1,236 Slices + 20 BRAMs	26.99 ns	$> 23,000$	18

design has the lowest area increase however, it uses 4 times more RAM and registers than their SE design. BCDL exploits a tradeoff of ALMs versus RAM. This is similar to our BRAM implementations which exploit the tradeoff of slice area versus BRAMs. However, our design (6d) uses even fewer slices than the SE design (6a). The MTD for both WDDL and BCDL designs is the highest reported on FPGAs so far. However, their increase in security over their SE design is similar to our designs.

In this paper we have shown that just converting an implementation to use BRAMs increases the level of security against DPA attacks. We expected this result from the fact that BRAMs are glitch free and their power consumption during read operations remains almost constant [12]. We have also shown that BRAMs can be used with the hiding countermeasure SDDL to further increase the security. The total security gain over our unprotected (SE) LUT only implementation is similar to that achieved by other research groups.

REFERENCES

[1] P. Chodowiec and K. Gaj, "Very compact FPGA implementation of the AES algorithm," in *Cryptographic Hardware and Embedded Systems - CHES 2003, Cologne, Germany*, ser. LNCS, vol. 2779. Springer, Sep. 2003, pp. 319–333.

[2] G. Rouvroy, F.-X. Standaert, J.-J. Quisquater, and J.-D. Legat, "Compact and efficient encryption/decryption module for FPGA implementation of the AES Rijndael very well suited for small embedded applications," in *International Conference on Information Technology: Coding and Computing, ITCC 2004*, vol. 2. IEEE, 2004, pp. 583–587.

[3] R. Chaves, G. Kuzmanov, S. Vassiliadis, and L. Sousa, "Reconfigurable memory based AES co-processor," in *International Parallel and Distributed Processing Symposium, IPDPS 2006*, April 2006, p. 8.

[4] C.-J. Chang, C.-W. Huang, H.-Y. Tai, M.-Y. Lin, and T.-K. Hu, "8-bit AES FPGA implementation using block RAM," in *Conference of the IEEE Industrial Electronics Society, IECON 2007*, Nov. 2007, pp. 2654–2659.

[5] S. Drimer, T. Güneysu, and C. Paar, "DSPs, BRAMs and a pinch of logic: Extended recipes for AES on FPGAs," *ACM Trans. Reconfigurable Technol. Syst. (TRETS)*, vol. 3, no. 1, pp. 1–27, 2010.

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

[7] K. Tiri and I. Verbauwhede, "A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation," in *Design, Automation and Test in Europe (DATE'04)*. IEEE Computer Society, Feb 2004, pp. 246–251.

[8] D. Hwang, K. Tiri, A. Hodjat, B.-C. Lai, S. Yang, P. Schaumont, and I. Verbauwhede, "AES-based security coprocessor IC in 0.18-μm CMOS with resistance to differential power analysis side-channel attacks," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 4, pp. 781–792, Apr 2006.

[9] P. Yu and P. Schaumont, "Secure FPGA circuits using controlled placement and routing," in *CODES+ISSS '07: Hardware/software codesign and system synthesis*. New York, NY, USA: ACM, 2007, pp. 45–50.

[10] R. Velegalati and J.-P. Kaps, "DPA resistance for light-weight implementations of cryptographic algorithms on FPGAs," in *Field Programmable Logic and Applications, FPL 2009*, M. Daněk, J. Kadlec, and B. Nelson, Eds. IEEE, Aug 2009, pp. 385–390.

[11] J.-P. Kaps and R. Velegalati, "DPA resistant AES on FPGA using partial DDL," in *IEEE Symposium on Field-Programmable Custom Computing Machines – FCCM 2010*. IEEE, May 2010, pp. 273–280.

[12] E. Konur, Y. Özelçi, E. Arikan, and U. Ekşi, "Power analysis resistant SRAM," in *World Automation Congress (WAC)*, July 2006.

[13] *Advanced Encryption Standard (AES)*, National Institute of Standards and Technology (NIST), FIPS Publication 197, Nov 2001.

[14] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in *CHES 2004*, ser. LNCS, vol. 3156. Berlin / Heidelberg: Springer, Aug 2004, pp. 135–152.

[15] R. Velegalati and J.-P. Kaps, "Techniques to enable the use of block RAMs on FPGAs with dynamic and differential logic," in *International Conference on Electronics, Circuits, and Systems, ICECS 2010*. IEEE, Dec 2010, accepted, to be published.

[16] J. Daemen and V. Rijmen, "AES proposal: Rijndael," in *First Advanced Encryption Standard AES Conference*, Ventura, California, USA, 1999.

[17] V. Fischer and M. Drutarovsk, "Two methods of Rijndael implementation in reconfigurable hardware," in *Cryptographic Hardware and Embedded Systems – CHES 2001*, ser. LNCS, vol. 2162. Springer Berlin / Heidelberg, January 2001, pp. 77–92.

[18] K. Gaj and P. Chodowiec, *Cryptographic Engineering*. Springer, 2009, ch. FPGA and ASIC Implementations of AES, pp. 235–294.

[19] M. Nassar, S. Bhasin, J.-L. Danger, G. Duc, and S. Guilley, “BCDL: A high speed balanced DPL for FPGA with global precharge and no early evaluation,” in *Design, Automation and Test in Europe, DATE 2010*. IEEE, Mar 2010, pp. 849–854.

[20] R. P. McEvoy, C. C. Murphy, W. P. Marnane, and M. Tunstall, “Isolated WDDL: A hiding countermeasure for differential power analysis on FPGAs,” *ACM Trans. Reconfigurable Technol. Syst.*, vol. 2, no. 1, pp. 1–23, Mar 2009.