2010 IEEE Annual Symposium on VLSI

Differential Power Analysis of CAST-128

K.H. Boey, Y. Lu, M. O'Neill, R. Woods
Institute of Electronics, Communications and Information Technology (ECIT),
Queen's University Belfast,
Belfast, Northern Ireland
{kboey01, ylu8, r.woods}@qub.ac.uk, m.oneill@ecit.qub.ac.uk

Abstract—Power analysis is used to reveal the secret key of security devices by monitoring the power consumption of certain cryptographic algorithm operations through a statistical analysis approach known as Differential Power Analysis (DPA). Whilst this has been applied extensively to attacks on FPGA devices, there has been little research into attacks on ASIC devices. Although standard DPAs are essentially independent of the block cipher that they target, some are less susceptible than others due to algorithm's structure, and therefore more difficult to attack such as the CAST-128. In this paper, we outline the first reported power analysis attack of CAST-128 as it falls into the category just outlined and it is the only algorithm that has not been practically broken either on FPGA or ASIC; it is also a common block cipher used in Canada. The paper outlines an approach that reveals all 128 bits of the secret key within 300,500 power traces, highlighting insights on attacking the registers rather than the Sbox. Finally, the effect of applying the Hamming weight power model on different widths of the target register under attack in ASIC device is evaluated.

I. INTRODUCTION

Today, most of financial transactions and personal information are transferred on the internet. Data encryption is crucial in this process to secure data from unauthorized people. In order to find solution to protect data encryption system and avoid adversary to reveal the encrypted data, a lot of research on how to break the encryption system have been carried out [1-9].

Power analysis is the most effective low-cost non-invasive and passive attack that can be performed on cryptographic devices. Power analysis attacks can be categorized as simple power analysis (SPA) attacks and differential power analysis (DPA) attacks [1]. Simple power analysis is carried out by visually inspecting the power trace that is captured when a cryptographic device executes a security algorithm to reveal the secret key. Differential power analysis uses statistical processing to analyze a large number of power traces in order to uncover the key.

Differential Power Analysis (DPA) attacks were first introduced by Kocher *et al.* [1] in 1999 and are a more effective approach in comparison to SPA. In DPA, the adversary does not need to know when a particular operation is actually computed by the cryptographic device. Differential power analysis employs a number of different analysis approaches such as differential mean analysis [1], correlation power analysis [2], template attacks [3] and frequency based analysis [4].

To date, DPA attacks have been carried out on a range of encryption algorithms like DES [5], SHACAL-2 [6], ARIA [7], and AES [8]. Although all block cipher are susceptible to DPA attack [9], some are less susceptible than others due to the algorithm's structure, and are therefore more difficult to attack and easier to protect from attack. To the authors' knowledge, this is the first paper to present a DPA attack against an ASIC hardware implementation of the CAST-128 encryption algorithm. This is important as the CAST-128 encryption algorithm has been widely used as the default cipher in some versions of the Gnu Privacy Guard (GPG) and Pretty Good Privacy (PGP) email enciphering tools [10], but study in breaking the algorithm is not widely covered by researcher.

CAST-128 [11] is a 64-bit block cipher developed by Carlisle Adams and Stafford Tavares in 1996, which was approved by the Communication Security Establishment (CSE) for use by the Government of Canada [12] and adopted as an ISO/IEC block cipher standard. Although CAST-128 is patented, it has been made available on a royalty free basis for both commercial and non-commercial use. It has similar structure as DES but with stronger round function. Each round involves the input word being combined (with addition, subtraction, or XOR) with a sub-key, and then rotated based on the value of another sub-key. Finally, the input word is split into four bytes, each one is passed through one of the Sboxes, and the outputs are combined, again using a combination of addition, subtraction, and XOR. It is believed that attackers cannot target the SBox structure in a similar manner to attacks on DES implementations. In order to implement protection

978-0-7695-4076-4/10 $26.00 © 2010 Crown Copyright
DOI 10.1109/ISVLSI.2010.14

IEEE computer society

particularly against the CAST-128 encryption algorithm, it is important to know how to attack it.

In this research, the DPA attack is performed on an ASIC realization of the Side-channel Attack Standard Evaluation Board (SASEBO) [13]. This evaluation board includes an ASIC cryptographic device comprising seven cryptographic algorithms, namely AES, DES, MISTY1, Camellia, SEED, RSA and CAST-128. The ASIC cryptographic device has been fabricated using TSMC 130-nm CMOS technology. The CAST-128 encryption algorithm has been selected to apply DPA attack over the other six cryptographic algorithms because it is the only one that has not been practically broken to date.

The CAST-128 algorithm, ASIC evaluation environment and the power analysis technique employed in this research are reviewed in section II. A description of the DPA attack strategy is provided in section III and results are provided in section IV.

II. Background

A. *CAST-128 Algorithm*

The CAST-128 is a Feistel cipher and is similar in structure to the well known Data Encryption Standard (DES) [14] algorithm (shown in Fig. 1), but differs from DES in term of round key and Sbox size. DES's round key is a permutation of secret key while CAST-128's round key is generated using multiple layers of non-linear element (Sbox). ASIC CAST-128 encryption algorithm required 8KB for eight different Sboxes, where each of them has 8-bit input (2^8=256 addresses) and 32-bit output (4 Bytes) that requires 256 x 4 Bytes. DES encryption algorithm only required 256 Bytes for eight 6-bit input and 4-bit output. The detail of the CAST-128 and DES encryption algorithm implementation on the ASIC chip are is discussed in [16].

CAST-128 consists of three types of round functions. These three types of round functions are used iteratively as shown in Algorithm 1. For a secret key length of more than 80 bits, the algorithm uses 16 rounds. For key lengths of between 40 and 80 bits, the algorithm uses 12 rounds.

ALGORITHM 1: CAST-128

Function Type 1:	$I = ((Km_i + D) <<< Kr_i)$
	$f = ((S_1[I_a] \wedge S_2[I_b]) - S_3[I_c]) + S_4[I_d]$
Function Type 2:	$I = ((\mathrm{Km_i} \wedge D) <<< \mathrm{Kr_i})$
	$f = ((S_1[I_a] - S_2[I_b]) + S_3[I_c]) \wedge S_4[I_d]$
Function Type 3:	$I = ((Km_i - D) <<< Kr_i)$
	$f = ((S_1[I_a] + S_2[I_b]) \wedge S_3[I_c]) - S_4[I_d]$

Rounds 1, 4, 7, 10, 13, and 16 use *f* function Type 1.
Rounds 2, 5, 8, 11, and 14 use *f* function Type 2.
Rounds 3, 6, 9, 12, and 15 use *f* function Type 3.
where,
Km is used as a "masking" key;
Kr is used as a "rotation" key;
D is the data input to the *f* function;
I_a, I_b, I_c, I_d are the most significant byte down to the least significant byte of *I*, respectively;
S_1, S_2, S_3, S_4 are SBox 1, 2, 3 and 4;
i is the round number;
"+" and "-" are addition and subtraction modulo 2^{32};
"^" is bitwise exclusive OR;
"<<<" is a circular left shift operation.

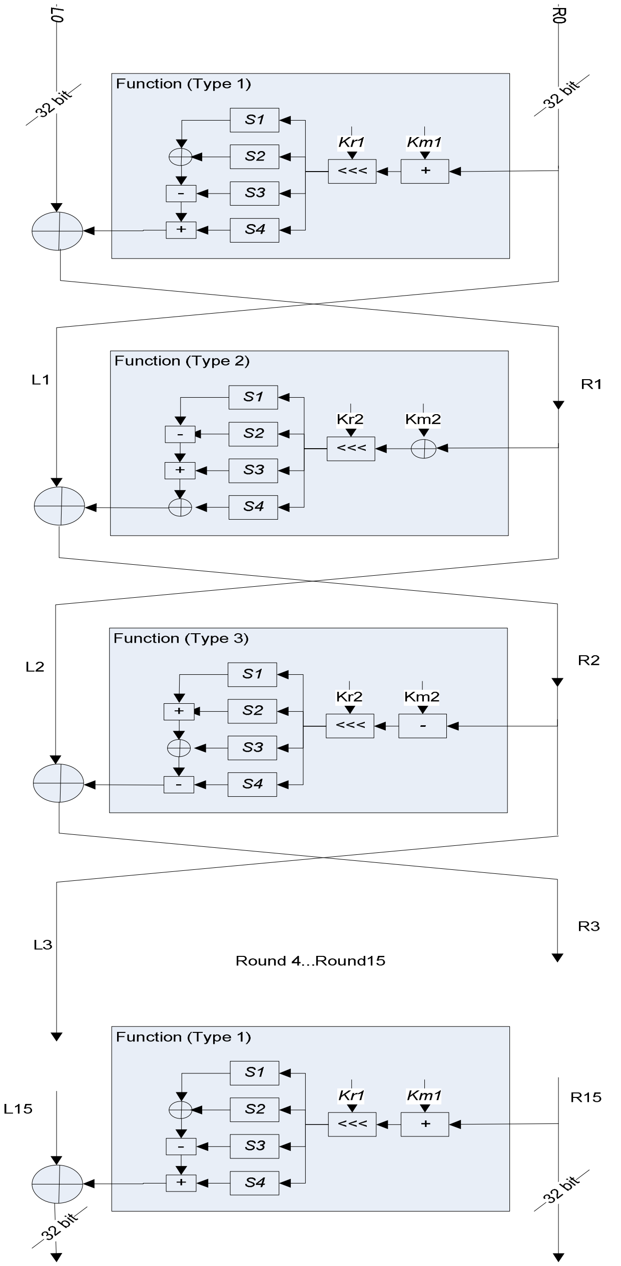

Fig. 1 CAST-128 Algorithm structure

CAST-128 has eight 32-bit S-boxes which are based on bent function [15], where a bent function is a boolean function of *n* variables that has nonlinearity equal to 2^{n-1} to $2^{(n/2)-1}$. These S-boxes have an 8-bit input and a 32-bit output. Four S-boxes are used in the round function, namely S_1, S_2, S_3 and S_4. The rest of the S-boxes are used in the key schedule. The CAST-128 round keys are formed from the 128-bit secret key, $x_0x_1x_2x_3x_4x_5x_6x_7x_8x_9x_Ax_Bx_Cx_Dx_Ex_F$. Round keys 1 to 4, $K_1 \ldots K_4$, are generated using equations (1) to (8). The remaining 28 rounds keys are generated by iterating through equations (1) to (12) seven times. Each round key is 32 bits in length. Round keys 1 to 16 are assigned as the masking keys, Km_i and round keys 17 to 32 are used as Kr_i. Kr_i only use the least significant 5 bits of the round key.

$$z_0z_1z_2z_3 = x_0x_1x_2x_3 \wedge S_5[x_D] \wedge S_6[x_F] \wedge S_7[x_C] \wedge S_8[x_E] \wedge S_7[x_8] \tag{1}$$
$$z_4z_5z_6z_7 = x_8x_9x_Ax_B \wedge S_5[z_0] \wedge S_6[z_2] \wedge S_7[z_1] \wedge S_8[z_3] \wedge S_8[x_A] \tag{2}$$
$$z_8z_9z_Az_B = x_Cx_Dx_Ex_F \wedge S_5[z_7] \wedge S_6[z_6] \wedge S_7[z_5] \wedge S_8[z_4] \wedge S_5[x_9] \tag{3}$$
$$z_Cz_Dz_Ez_F = x_4x_5x_6x_7 \wedge S_5[z_A] \wedge S_6[z_9] \wedge S_7[z_B] \wedge S_8[z_8] \wedge S_6[x_B] \tag{4}$$

$$K_i = S_5[z_8] \wedge S_6[z_9] \wedge S_7[z_7] \wedge S_8[z_6] \wedge S_5[z_2] \tag{5}$$
$$K_{i+2} = S_5[z_A] \wedge S_6[z_B] \wedge S_7[z_5] \wedge S_8[z_4] \wedge S_6[z_6] \tag{6}$$
$$K_{i+3} = S_5[z_C] \wedge S_6[z_D] \wedge S_7[z_3] \wedge S_8[z_2] \wedge S_7[z_9] \tag{7}$$
$$K_{i+4} = S_5[z_E] \wedge S_6[z_F] \wedge S_7[z_1] \wedge S_8[z_0] \wedge S_8[z_C] \tag{8}$$

$$x_0x_1x_2x_3 = z_8z_9z_Az_B \wedge S_5[z_5] \wedge S_6[z_7] \wedge S_7[z_4] \wedge S_8[z_6] \wedge S_7[z_0] \tag{9}$$
$$x_4x_5x_6x_7 = z_0z_1z_2z_3 \wedge S_5[x_0] \wedge S_6[x_2] \wedge S_7[x_1] \wedge S_8[x_3] \wedge S_8[z_2] \tag{10}$$
$$x_8x_9x_Ax_B = z_4z_5z_6z_7 \wedge S_5[x_7] \wedge S_6[x_6] \wedge S_7[x_5] \wedge S_8[x_4] \wedge S_5[z_1] \tag{11}$$
$$x_Cx_Dx_Ex_F = z_Cz_Dz_Ez_F \wedge S_5[x_A] \wedge S_6[x_9] \wedge S_7[x_B] \wedge S_8[x_8] \wedge S_6[z_3] \tag{12}$$

B. SASEBO-R

The Side-channel Attack Standard Evaluation Board (SASEBO) [13] was developed by the Research Center for Information Security at the National Institute of Advanced Industrial Science and Technology (AIST) together with the Graduate School of Information Science, Tohoku University, Japan. The SASEBO family of circuit boards contains an ASIC cryptographic device, a controller and a USB interface to interact with the computer. The SASEBO circuit board with an ASIC cryptographic device is called SASEBO-R. The ASIC cryptographic device supports seven cryptographic algorithms, namely AES, DES, MISTY1, Camellia, SEED, CAST-128 and RSA. In the implementation of the CAST-128 cryptographic algorithm on the exported SASEBO-R board, the upper 72 bits of the key are fixed to 0x000102030405060708 (in hexadecimal) and the lower 56 bits are available for modification. The detail architecture for the CAST-128 circuit, included in SASEBO-R, is available in ISCAS2007 [16].

C. Correlation Power Analysis

Correlation power analysis (CPA) was first proposed by Brier *et al.* [2]. This type of power analysis technique requires a power model to attack a cryptographic device. The adversary needs to build a hypothetical model of the cryptographic device under attack. This hypothetical model, *P* is used to predict intermediate values within the cryptographic device for all secret key combinations. Pearson's correlation function [17] is then used to calculate the correlation coefficient between the power trace, *T,* and the hypothesis, as stated in (13). In a successful attack, only the correct key hypothesis will lead to a high correlation coefficient.

CPA allows the adversary to choose a power model other than the typical Hamming weight model. Both techniques are used in this attack. The normalization coefficient in (13) provides attackers with a better visualization of the correct key guess. CPA is one of the most widely used power analysis techniques and it has been applied to successfully attack a range of cryptographic algorithms [5-8].

$$C(T,P) = \frac{E(T \cdot P) - E(T) \cdot E(P)}{\sqrt{Var(T) \cdot Var(P)}} \tag{13}$$

where,
T is the measured power traces,
P is the hypothetical power model.

The Hamming weight (HW) model is the most commonly used to model the side-channel leakages in power analysis. It considers the number of bit transitions in a device as an image of its leakage. The HW model can be used to mimic the data value in a register at a particular time. The HW of a data block is the number of logic '1' bits. For a 4 bit data block with the logic value of 1110b, the Hamming weight, HW(1110b) = 3.

Hamming distance (HD) power model is also a popular power model used in power analysis. HD requires knowledge of both the previous and the current value of the target intermediate result. Therefore, it is typically necessary that the target intermediate value can always be predicted. In this experiment, power analysis attacks were applied to round 1 where the previous value of the registers are not predictable. Therefore, Hamming distance model is less applicable than Hamming weight power model.

III. Practical CPA Attack of CAST-128

In this research, CPA has been used to attack the CAST-128 encryption algorithm on a SASEBO-R board. The attack environment employed in this work is illustrated in Fig. 2. An Agilent mixed-signal oscilloscope (MSO6104A) is used to capture power traces from the device under attack with a 1 GHz sampling frequency. Random sequences of plaintext, X, are issued and sent to both the hardware equipment controller and a CAST-128 software tool. The controller generates and sends encryption commands to the cryptographic device under attack, together with the plaintext sequence. The oscilloscope is then triggered by the device under attack using a START signal from the device and transfers the resulting power traces to the computer, where they are stored as a database of plaintext values and their corresponding power traces, denoted as P(X).

The CAST-128 software tool generates a matrix of power model values related to the plaintext values and the round key hypothesis using HW, denoted as *H(X, Key).*The matrix of the power model has the Km_i round key hypothesis size of 8-bit and the X plaintext size of 32-bit. For elaboration, round key 1 is used in this discussion. Round key 1, Km_1 also denoted as $K_{1,}$ is added to the plaintext using 32 bit modulos addition in *Function Type 1*. A circular left shift using rotation key, $Kr_{1,}$ is applied to the result of the addition to produce *I*. *Sbox1*, *2*, *3* and *4* are applied to *I*, in *f* function. The rotation key is not used as part of the power model because it can be analyzed easily using a direct search, which is only 32 tries.

To reveal the whole 32-bit of round key 1, the matrix of power model is first applied to Sbox4 [I_d], (I_d= ((Km_i <<<Kr_i

$+ X <<<Kr\ )\&\ 0xFF)$, with the key hypothesis of least significant 8-bit of permutated round key 1, $Km_1 <<<Kr_1$, the X plaintext size of 32-bit and 32-bit of Sbox4 output. The X plaintext value used in the power model are from 0 to 255 for every 8-bit. Lastly, Pearson's correlation coefficient function is performed on the power traces, P(X), and the power model, H(X, Key). The correct 8 bit key hypothesis can be distinguished from the highest correlation between P and H.

$$H(X, key) = (S_4[((Km_i + X)<<<Kr_i)\&\ 0xFF)])$$

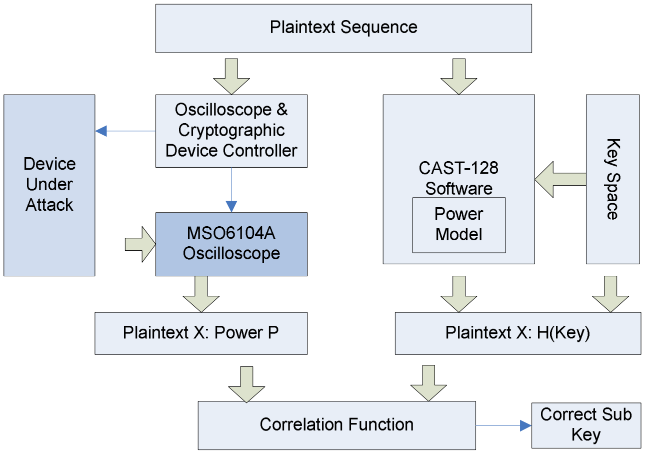

Fig. 2 Attack environment for device under Attack

The same approach is applied to the other 3 Sboxes in order to obtain the rest of 24-bit of permutated round key, $Km_1 <<<Kr_1$. Kr_1 is needed to reveal the round key, Km_1. The 5-bit rotation key, $Kr_{1,}$ can be analyzed easily using direct search. Alternatively, the rotation key can be obtained by analysing another set of power traces with one bit value different in the plaintext. By assuming the value in the plaintext is due to the one bit value different in the Km_1, denoted as Km_1*. Once the two permutated round key are obtained from the CPA of both sets of power traces, the Kr_1 can be revealed by taking the different of $Km_1<<<Kr_1$ and $Km_1*<<<Kr_1$ as shown in equation (13a)

$$\begin{aligned} I &= (Km_1 + (X+1)) <<< Kr_1 \\ &= (Km_1 +1) <<< Kr_1 + X <<< Kr_1 \\ &= Km_1* <<< Kr_1 + X <<< Kr_1 \\ & where,\ Km_i* \text{ is } Km_1 +1 \end{aligned}$$

$$(Km_1*<<<Kr_1) - (Km_1<<<Kr_1) = 1<<<Kr_1 \tag{13a}$$

Another more effective way to reveal the most significant byte of permutated round key is to attack on the register instead of Sbox1. This is because register switches value at clock edge. A register needs a setup and hold time so it will be easier to notice value change by observing the power traces where more sampling points are showing this feature. Once the lower 3 bytes of the permutated round key are revealed by attacking Sbox S_2, S_3 and S_4, the most significant byte of the permutated round key can be revealed from the register. The power model values related to the plaintext and the round key hypothesis for this attack is shown below, denoted as H(X, Key).

$$\begin{aligned} H(X, key) &= HW(f(X, key)) \\ &= HW(S_1[I_a] \wedge S_2[I_b]) - S_3[I_c]) + S_4[I_d]) \end{aligned}$$

The target intermediate value in CAST-128 used in the CPA attack is the result obtained from round 1 after $f(R_0)$ is *XOR*ed with L_0 as shown equations (14) and (15) and also demonstrated in Fig. 3. The result from round 1 will then be stored in registers, L_1 and R_1, where the target register is R_1. The operating time for 1 round (1 clock cycle) is 41.66ns (24MHz), and the sampling rate is 1GSa/s, hence there are approximately 42 sampling points captured in 1 clock cycle.

$$L_i = R_{i-1}; \qquad (i:= 1\ldots16) \tag{14}$$

$$R_i = L_{i-1}\wedge f(R_{i-1}, Km_i, Kr_i) \qquad (i:= 1\ldots16) \tag{15}$$

If CPA is only applied to round 1, the adversary will be unable to reveal the entire secret key because the round key used in round 1, K_1, was derived from z_2, z_6, z_7, z_8 and z_9 as shown in (5) These intermediate value are derived from x_0, x_1, x_2, x_3, x_8, x_9, x_A, x_B, x_C, x_D, x_E and x_F as shown in (1), (2) and (3). In order to reveal the entire secret key, the adversary will need to attack another round which uses a round key that is derived from x_4, x_5, x_6 and x_7. Therefore, the adversary will need to attack either round 3 or round 4 as shown in equation (7) and (8). Round 3 or round 4 can be attacked by applying a similar approach as discussed above for round 1 on registers R_3 or R_4, as shown in equations (16) and (17).

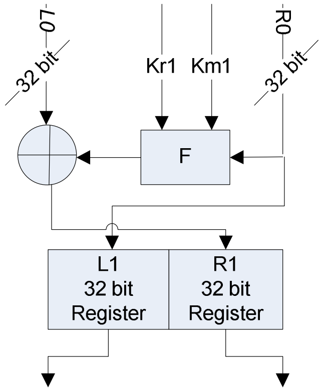

Figure 3 CPA performed on target register, R_1

$$R_3 = L_2\wedge f(R_2, Km_3, Kr_3) \tag{16}$$

$$R_4 = L_3\wedge f(R_3, Km_4, Kr_4) \tag{17}$$

An alternative approach using a chosen-plaintext attack is also evaluated to attack round 1. In this context, chosen-plaintext attack can reduce algorithmic noise contributed by the 'XOR' of L_0 for equation (15) when i = 1. Assuming $L_{0,}$ the most significant 32 bits of the plaintext input, is assigned a value of all zeros, equation (15) can then be simplified, resulting in equation (18). This can significantly reduce the number of power traces required in the attack, but it limits the practical employment of the attack. The chosen-plaintext approach can only be applied to round 1, as L_2 in equation (16) cannot be simplified to zeros when attacking either R_3 or R_4.

$$R_1 = f(R_0, Km_1, Kr_1) \tag{18}$$

IV. CPA Results for CAST-128

The CPA attack is applied on the least significant 8 bits of the 128-bit secret key, termed the sub-key. After the first round of CAST-128 is completed, the result, R_1, is stored in a register. By correlating the Hamming weight power model for the 32-bit register in which R_1 is stored to the actual power traces, the correct round key can be revealed.

In order to attack the data stored in the register for the first round, power traces from the SASEBO-R had to be captured. The arithmetic mean of all the power traces needed to reveal the correct round key captured from the SASEBO-R ASIC board is shown in Fig. 4. Post-processing is applied on the round key to obtain sub-key. A straightforward way would require search over 37-bit round key (5-bit Kr_i and 32-bit Km_i) to match the sub-key to the round key. If the round key is generated on the fly, the adversary can map the round key to the secret key by performing a CPA attack on the Sbox of the F Function used by key scheduler. This will reduce the post-processing effort.

All of the CAST rounds executed on the ASIC chip can be seen in Fig 4. From the correlation graph shown in Fig. 5, 300,500 power traces (From Fig. 5: 601 x 500 = 300,500) are needed to reveal the correct sub-key, 95. However, by using the chosen-plaintext attack for round 1, only 75,000 power traces (From Fig. 6: 150 x 500 = 75,000) are needed to reveal the correct sub-key, as illustrated in Fig. 6.

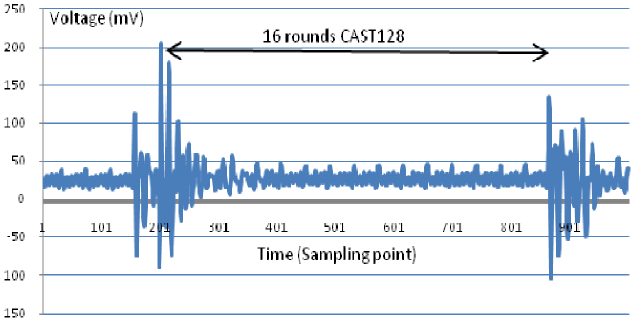

Fig. 4 Arithmetic mean of all the power traces

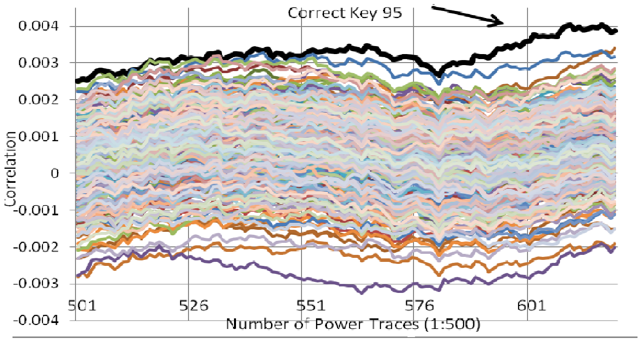

Fig. 5 Correlation value of each key candidate

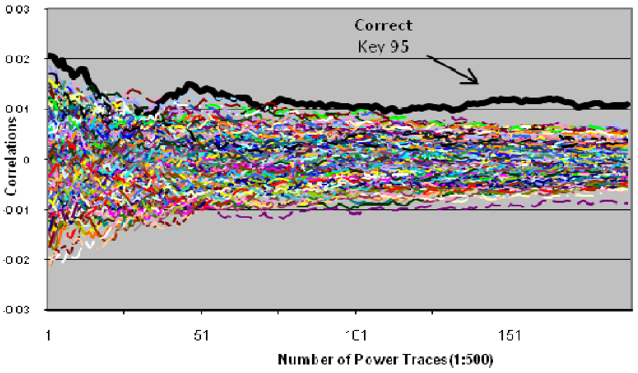

Fig. 6 Correlation value of each key candidate with chosen plaintext attack

V. Analysis and Discussions

By analysing the correlation values of the hypothetical key candidates within clock cycles 1 and 2, the true peak for the least significant 8 bits of the target sub-key 0x000102030405060708090A0B0C0D0E5F (in hex.) was revealed. In Fig 7, the correct sub-key shows a peak around sampling points 42 to 44 which corresponds to the fact that the target register under attack is at the end of round 1.

The CAST-128 encryption algorithm uses 16 rounds and intermediate data are stored in registers for each round before the encryption algorithm proceeds to the next round. By attacking the register, it is easier to correlate the value to the hardware implementation as compared to attack on the Sbox. This is because Sbox are mainly combinational logic which will change whenever the input value change regardless of clock edge. Besides that, noise will cause the logic value to change.

The effect of predicting only a fraction of the bits in a target register has already been analyzed in numerous publications [18]. Most of the research has been conducted on FPGA devices and micro-controller i.e. software, however there has not been as much research on ASIC devices, which is important as most of the implementations in commercial domains will eventually end up in this technology. Beside that, it is easier to reveal the secret key from FPGA device as compared to ASIC chip. This is because in the ASIC design, the logic gates used to implement CAST-128 are more optimized. The logic gate used in ASIC design are mainly primitive cells (AND, OR, D-flip flop and others) and the cell placement are closer to each other which help to reduce the dynamic power consumption. FPGAs are built from one basic "logic-cell", duplicated hundreds or thousands of time. A logic-cell is basically a small lookup table (LUT) , D-flip flop and a 2-to-1 mux (to bypass the flip flop if desired). These basic "logic cell" were configured using the software design tool as basic logic gates which consume higher power as compared to primitive cells in the ASIC design.

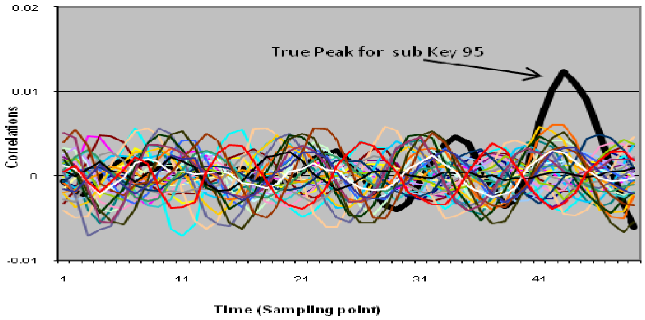

Fig. 7 Correlation value of key candidate within clock cycle 1 to 2

In this research, the HW power model for different widths of the target register has also been analysed under the chosen-plaintext attack. The attack for widths of 8, 16 and 24 bits were evaluated with the result shown in Table 1. Using CPA and a HW over 8 bits of the register, 160,000 power traces is not sufficient to reveal the sub-key. When 16 bits of the register are targeted, approximately 110,000 traces are needed to reveal the correct sub-key while for 24 bits of the register, about 93,500 power traces are needed. Therefore, the number of power traces required reduces by applying the HW model on more register bits.

To further improve the correlation coefficient result, an adversary could apply the HW model on 64 bits of the register under attack. The trade-off in applying the HW model on more register bits is an increase in the computation time needed to analyse the power traces. However, overall, the time taken to perform a full attack is reduced.

Table 1 Evaluation of attack for different widths of target register.

Number of register bits under attack	Power Trace Number
8	N/A
16	110000
24	93500
32	75000

VI. CONCLUSIONS

In this paper, the first CPA attack of the CAST-128 encryption algorithm is presented. The CAST-128 secret key can be successfully revealed in 300,500 power traces using the proposed attack strategy with CPA attack. A chosen-plaintext attack can be used to simplify the attack and reduce the number of power traces required. The number of power traces required to successfully reveal the secret key can also be reduced by increasing the width of the target register on which the HW power model is applied in an attack.

ACKNOWLEDGMENT

This research is funded under an EPSRC leadership fellowship grant (EP/G007586/1).

REFERENCES

[1] P. C. Kocher, Joshua Jaffe, and Benjamin Jun, "Differential Power Analysis," Advances in Cryptology- CRYPTO'99, California, USA, August 15-19, 1999, LNCS 1666, Springer, pp. 388-397, 1999.

[2] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in CHES 2004, LNCS 3156, 2004, pp. 16-29.

[3] S. Chari, J.R. Rao, P. Rohatgi, "Template Attacks," In: B.S. Kaliski Jr., C¸ .K. Ko¸c, C. Paar (eds.): CHES 2002, LNCS 2523, 2003, pp. 13–28.

[4] C. H. Gebotys, S. Ho, and C.C. Tiu, "EM Analysis of Rijndael and ECC on a Wireless Java-Based PDA," CHES 2005, LNCS vol. 3659, pp. 250-264, Springer-Verlag 2005.

[5] F.-X. Standaert, S. B. Ors, J.-J. Quisquater, and B. Preneel, "Power analysis attacks against FPGA implementations of the DES," in Field Programmable Logic and Application. Heidelberg, Germany: Springer-Verlag, 2004, vol. 3203, LNCS, pp. 84–94.

[6] Yingxi Lu, M. O'Neill, J. McCanny, "Differential Power Analysis of SHACAL-2 Hardware Implementation," ISCAS 2008, pp. 2933-2936

[7] C.K. Kim, M. Schlaffer, S.J. Moon, "Differential Side Channel Analysis Attacks on FPGA implementation of ARIA," ETRI Journal, Volume 30, Number 2, April 2008

[8] S. B. Ors, F. Gurkaynak, E. Oswald, and B. Preneel, "Power analysis attack on an ASIC AES implementation," ITCC 2004, Las Vegas, NV.

[9] E. S. Mangard, E. Oswald, and F.-X. Standaert, "One for All - All for One: Unifying Standard DPA Attacks," Cryptology ePrint Archive, Report 2009/449, Sept, 2009.

[10] "The International PGP Home Page," http://www.pgpi.org/

[11] C. Adams, "The CAST-128 Encryption Algorithm," RFC2144, May, 1997 (http://www.ietf.org/rfc/rfc2144.txt)

[12] CSE, "IT Security Program – Cryptographic Algorithms," http://www.cse-cst.gc.ca/services/crypto-services/crypto-algorithms-e.html

[13] "SASEBO Quick Start Guide," version 1.0, October, 2008, (http://www.rcis.aist.go.jp/special/SASEBO/)

[14] "FIPS 46-3: Data Encryption Standard," FIPS FUB 46.3, Oct 25, 1999.

[15] O. S. Rothaus, "On 'Bent' Functions," Journal of Combinatorial Theory, 20(A), 1976, pp 300-305

[16] T. Sugawara, N. Homma, T. Aoki, and A. Satoh, "A High-Performance ASIC Implementation of the 64-bit Block Cipher CAST-128," Proceedings of 2007 International Symposium on Circuits and System (ISCAS 2007), pp. 1859-1862, May 2007.

[17] J. L. Rodgers, W. A. Nicewander, "Thirteen ways to look at the correlation coefficient," The American Statistician, 42(1):59–66, Feb 1988.

[18] S. Mangard, "Hardware Countermeasures against DPA – A Statistical Analysis of Their Effectiveness," in the proceedings of CT-RSA 2004, LCNS, vol 2964, pp 222-235, San Francisco, CA, USA, February 2004.