Key Update Countermeasure for Correlation-Based Side-Channel Attacks

Yutian Gui, Suyash Mohan Tamore, Ali Shuja Siddiqui & Fareena Saqib

Journal of Hardware and Systems Security

ISSN 2509-3428

J Hardw Syst Secur
DOI 10.1007/s41635-020-00094-x

Springer

Journal of Hardware and Systems Security
https://doi.org/10.1007/s41635-020-00094-x

Key Update Countermeasure for Correlation-Based Side-Channel Attacks

Yutian Gui[1] · **Suyash Mohan Tamore**[1] · **Ali Shuja Siddiqui**[1] · **Fareena Saqib**[1]

Received: 14 January 2020 / Accepted: 13 April 2020

Abstract
Side-channel analysis is a non-invasive form of attack that reveals the secret key of the cryptographic circuit by analyzing the leaked physical information. The traditional brute-force and cryptanalysis attacks target the weakness in the encryption algorithm, whereas side-channel attacks use statistical models such as differential analysis and correlation analysis on the leaked information gained from the cryptographic device during the run-time. As a non-invasive and passive attack, the side-channel attack brings a lot of difficulties for detection and defense. In this work, we propose a key update scheme as a countermeasure for power and electromagnetic analysis-based attacks on the cryptographic device. The proposed countermeasure utilizes a secure coprocessor to provide secure key generation and storage in a trusted environment. The experimental results show that the proposed key update scheme can mitigate side-channel attacks significantly.

Keywords Hardware security · Side-channel attack · Correlation power analysis · Electromagnetic analysis · Trusted Platform Module

1 Introduction

The side-channel attacks can steal the secret key used in the encryption engine [1]. During execution, the leakage of physical information (a.k.a. side-channel) is inevitable and can be utilized to reveal the information based on the fundamental principle that there is a correlation between the side-channel leakage and the internal state of the processing device, which is related to the secret information. In contrast to invasive attacks which require direct access to the internal components in the chip, the side-channel attack exploits external leaked information, such as power consumption, electromagnetic radiation, and time delay. Besides, the side-channel attack is passive which has become a critical threat to the security of cryptographic chips and devices.

To mitigate the risk of side-channel attacks, countermeasures such as message hiding [2] and masking technique [3, 4] are presented in the literature. The objective of such techniques is to increase the time required to reveal the secret key thereby protect cryptographic implementations from different side-channel attacks.

In this work, we propose a key update scheme with the integration of a secure coprocessor on the hardware-based implementation of the Advanced Encryption Standard (AES) to increase the resilience to different side-channel attacks.

Contributions This paper makes the following contributions:

1. We have applied the correlation power analysis (CPA) attack and the correlation electromagnetic analysis (CEMA) attack on hardware-based AES-128 and revealed the secret key successfully to show the effectiveness of side-channel attacks.
2. We present a flexible key update scheme and prove that the proposed scheme makes the design resilient to side-channel attacks by experiments.

✉ Yutian Gui
ygui@uncc.edu

Suyash Mohan Tamore
stamore@uncc.edu

Ali Shuja Siddiqui
asiddiq6@uncc.edu

Fareena Saqib
fsaqib@uncc.edu

[1] The University of North Carolina at Charlotte, Charlotte, NC 28223, USA

Published online: 30 May 2020

3. We integrate the Trusted Platform Module (TPM) chip with the FPGA fabric to generate and store secret keys in a secure environment for protecting the keys used for encryption.

Paper organization The paper is organized as follows. Section 2 introduces related work, and Section 3 describes the attack model. Section 4 explains the proposed countermeasure. The experimental setup and the result are presented in Sections 5 and 6. The security analysis is furnished in Sections 7 and 8 discusses limitations. Finally, conclusions are drawn in Section 9.

2 Related Works

2.1 AES Encryption

AES is a US National Institute of Standards and Technology (NIST) standard for symmetric encryption [5]. AES supports three key lengths (128, 192, and 256) to meet different requirements of security strength, using 10, 12, and 14 rounds of transformations respectively. AES is widely used in many areas, such as communication and data storage. In this paper, AES-128 is used as the target for exploring the resilience to side-channel attacks and the effectiveness of proposed countermeasures.

Figure 1 shows the encryption process of AES-128. Firstly, the key is expanded to a unique pseudo-random key for each round. There are four operations in the AES algorithm that adds to confusion and diffusion to make the algorithm resilient to linear and differential cryptanalysis:

- SubBytes: Byte-by-byte substitution with a 16x16 lookup table name S-box.
- ShiftRows: A cyclic shifting in each row by a certain offset.
- MixColumns: A mix-up of the bytes in each column separately.
- AddRoundkey: Add the round key to the output of the previous step.

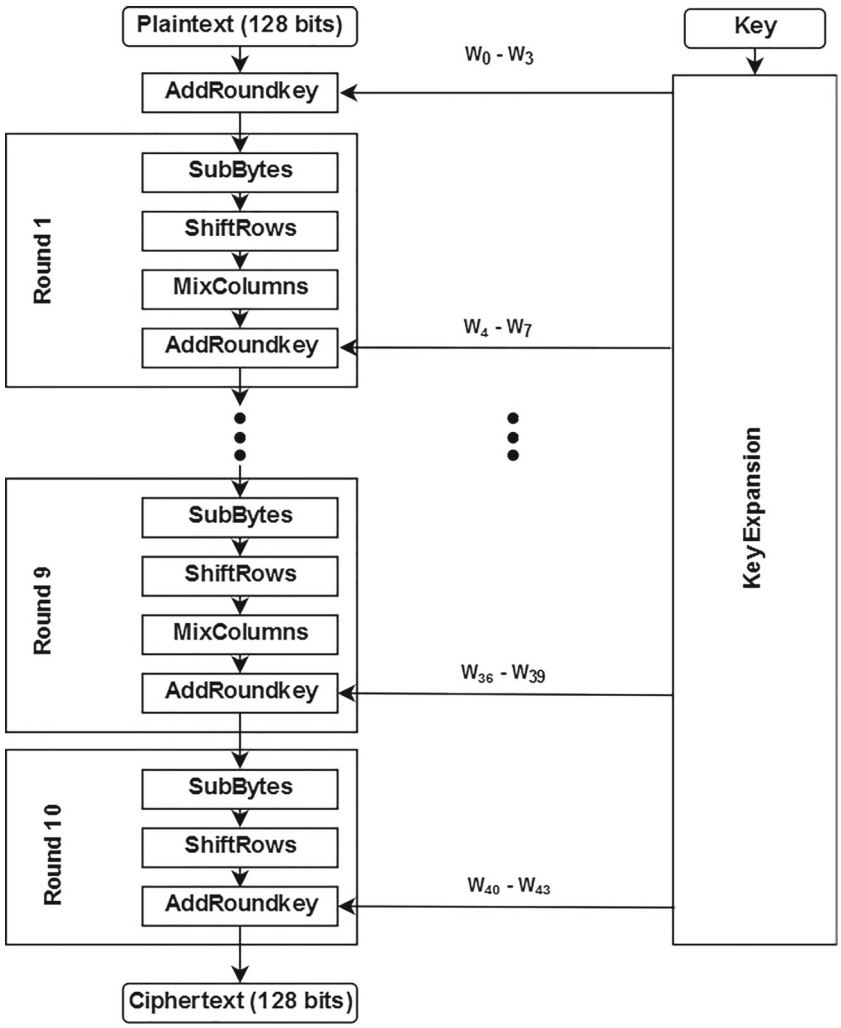

Fig. 1 AES-128 encryption process

In AES encryption, each round performs all four steps except the last round which only performs SubBytes, ShiftRows, and AddRoundKey operations.

2.2 Side-Channel Attack

The side-channel attack (SCA) targets physical leaked information. The device during execution leaks physical information, which can be used for revealing the internal operation processed on the device thereby stealing the secret non-invasively, including power consumption [6, 7], electromagnetic radiation [7, 8], time delay [1, 9], and temperature [10].

Power consumption side-channel information can be used for attacking the encryption engine during computing extensive operations that produce power transients for each encryption round. The power consumption of a device depends on the processed data and the executed instructions. The power analysis collects the real-time power consumption of encryption and builds different mathematical models to find the correlation between the variation of power and hardware operations on the victim device thereby reveal the secret key. There are several power analysis attacks, including:

- Simple power analysis (SPA): A power analysis method which extracts the secret key by looking at the variation of power consumption directly. SPA is inefficient when the noise is huge or on the hardware-based implementation because of its parallel nature [6].
- Differential power analysis (DPA): An advanced power analysis which uses statistical analysis to reduce the noise and find the correlation between the power consumption and the key information even the power variation is very small [6].
- Correlation power analysis (CPA): A more efficient attack which uses hamming weight to build the model and Pearson coefficient to evaluate the correlation between the hypothetical model and the actual power consumption. The guessed key with the highest coefficient can be considered as most likely the correct value of the secret key. The efficiency of the CPA attack is higher than DPA [11].

Another form of side-channel attack is electromagnetic analysis (EMA). Electromagnetic radiation is caused by the internal processing and activity of cryptographic device hardware. Constant changing states of different elements inside the hardware, for example, logic gates, flip-flops, and registers, cause changes in the EM radiation pattern. By collecting the fluctuation of EM radiation and analyzing it, the secret key can be also extracted similar to power analysis. In [8], Gandolfi et al. present an EM side-channel attack on a smart card chip implementing DES encryption. Bu et al. [12] modify the EM attack to decrease the number of total traces required to extract key information using a pre-processing technique which reduces noise levels.

Countermeasures for side-channel attacks are classified as hiding technique and masking technique that eliminate or reduce the correlation of side-channel and the bit switches.

The hiding technique aims at reducing the signal-to-noise ratio (SNR) for leaked information [13] which can be realized by randomization or equalization. Madlener et al. [14] apply shuffling techniques to GF(2^n) and propose a randomized multiplication scheme by rescheduling algorithm. In [15], Huss et al. propose a new countermeasure based on the nature of reconfiguration of FPGA for SCA mitigation. The physical architecture of the implementation is reconfigurable so that the dynamic power variation is randomized along with the change of data path. For equalization techniques, the core idea is to achieve equal power consumption at each moment for SCA mitigation. By using dual-rail pre-charge logic [16], the glitch will be remove; therefore, the dynamic power consumption is equalized to a constant value. This technique is also applied in sense amplifier-based logic (SABL) [17] and wave dynamic differential logic (WDDL) [18].

Applying key update scheme on cryptographic implementation has also been explored by existing works. Instead of relying on one, [19] demonstrates a security model that switches the secret key randomly to increase the leakage resiliency. However, the proposed framework is implemented on the software level and the strength of resilience is not provided. In [20], Medwed et al. propose a re-keying scheme that uses a key derivation function to generate random session keys for every block of the message to prevent the DPA attack. To meet the demand for multi-party communication, this scheme is modified in [21] which presents a multi-party key generation process for all the nodes in the network. In [22], Xi et al. replace the arithmetic key update function with a strong physically unclonable function (PUF) to provide the resilience to power analysis.

In contrast to hiding techniques, masking techniques are applied on the algorithmic level and randomize the intermediate values of the computation or by adding dummy instructions to avoid dependencies between the internal operation and the leaked side-channel information without changing the original functionality of the design [3, 4]. In [23], Nikova et al. propose a threshold implementation (TI) countermeasure against first-order DPA based on secret-sharing.

2.3 Trusted Platform Module

Trusted Platform Module (TPM) is a security coprocessor chip specified by the Trusted Computing Group (TCG) [24]

and standardized by the International Organization (ISO) for enhancing the security of hardware devices. A TPM chip contains a built-in true random number generator (TRNG), a tamper-resistant non-volatile memory (NVM), and several functionalities to realize root of trust (RoT).

TPM supports various encryption standards, such as AES, RSA encryption, and hash function to provide RoT [25] and authentication for hardware devices and communication [26]. Compared with TPM 1.2, TPM 2.0 enables greater crypto-agility by supporting more and newer cryptographic algorithms. Beyond that, TPM 2.0 has a three-level hierarchy architecture and allows multiple keys and algorithms per hierarchy. Keys used for encryption and authentication are derived from the primary keys and can be stored in the tamper-resistant persistent memory on the TPM chip. For generating elliptic curve Diffie-Hellman (ECDH) session keys, TPM can use NISTP256 and BNP256 curves to generate public-private key pairs. The public key of the communicating node is multiplied with a node's own private key to generate a symmetric AES key.

3 Attack Model

In this work, we demonstrate the power analysis and electromagnetic analysis-based side-channel attacks on AES-128 engine implemented on FPGA.

3.1 Correlation Power Analysis

1. The power traces are normalized using pre-amplifier and collected by oscilloscope during the execution of the processing encryption.
2. Make the key prediction. The original key is divided into 16 subkeys. For each subkey, guess every possible value.
3. Predict the power consumption using the Hamming weight leakage model to extract dynamic power consumption which reflects the data moving and operation. Hamming weight model presented in [27] states the correlation between data processed by the CMOS device and the electricity consumed at the same time. The mathematical equation for the hamming weight model is defined as:

$$E = aH(x) + b \tag{1}$$

where E is the electricity consumption and $H(x)$ is the Hamming weight of the data.

4. Evaluate the correlation between the modeled power and the actual power trace by using the Pearson correlation coefficient ρ which is defined as:

$$\begin{aligned}\rho(A, B) &= \frac{cov(A, B))}{\sigma_A \sigma_B} \\ &= \frac{E[(A - \mu_A)(B - \mu_B)]}{\sqrt{E[(A - \mu_A)^2]}\sqrt{E[(B - \mu_B)^2]}}\end{aligned} \tag{2}$$

where A and B are variables, cov denotes the covariance, σ denotes the standard deviation, μ is the mean value, and E is the expectation value. In this work, two variables are the hypothetical value and the actual power trace, so the Pearson correlation coefficient is applied in this way:

$$C(h, t) = \frac{\sum_{d=1}^{D}[(h_d - \bar{h})(t_d - \bar{t})]}{\sqrt{\sum_{d=1}^{D}(h_d - \bar{h})^2 \sum_{d=1}^{D}(t_d - \bar{t})^2}} \tag{3}$$

where h is the hypothetical value of the subkey, t is the power trace, and D is the total number of collected power traces.

The guessed subkey with the highest coefficient is considered as most likely the correct subkey used in the encryption.

3.2 Correlation Electromagnetic Analysis

To verify the effectiveness of EMA on the FPGA-based implementation of AES-128, the correlation electromagnetic analysis (CEMA) is also performed. EM emissions are the direct cause of the energy consumption of the CMOS device and as described in [8], a correlation exists between EM signal peaks and the data under process. Hence, we can consider EM emission instead of power consumption in the equation given in [27] and use hamming weight to build EM leakage model. The CEMA attack is similar to the CPA attack with the following differences:

- Different from the passive probe used in CPA attack, the CEMA attack uses a specialized EM probe.
- The process of EM capture is non-contact and the EM radiation is more susceptible to the environmental noise, so the amplification factor in EM capture is higher than power collection.
- Because the trigger signal also produces EM radiation (the first peak and the second peak in Fig. 7), the encryption process starts from the third clock cycle.

4 Proposed Countermeasure

In this work, we propose a secure design based on the key update scheme integration with tamper-resistant secure coprocessor for a resilient AES implementation.

4.1 Key Update Scheme

Different side-channel attacks require varying sizes of sample traces to achieve a successful attack that results in different amounts of time to capture and analyze leaked information. This time period is called measurement To disclose (MTD) period which denotes the time from the start of the physical information collection process to the end of the successful attack.

We use the least needed power/electromagnetic traces (LNT) to quantify the least amount of time needed for a successful attack. In other words, the less the amount of collected traces used for revealing the key, the higher efficiency the performed attack has.

To mitigate the risk of side-channel attacks, the main target in this work is to increase the LNT. We propose a key update scheme to achieve this goal. The proposed key update scheme is to change the value of the key used in AES encryption before the last used key can be revealed by the side-channel attack and also preserve forward secrecy. This scheme is applied and implemented on both the sender side and the receiver side.

The following are the steps of the proposed scheme which is also shown in Fig. 2:

1. Determine the LNT for single key (LNTS) of the target hardware. This process is repeated several times and the average value will be used as LNTS to remove the random variation.
2. Generate a list of random secret keys for encryption/decryption on TPM and share the key list with the receiver.
3. Set the update period (UP) which is less than LNTS and share it with the receiver.
4. Start the encryption process with the first key and change the key following the order of the key list when the value of the counter reaches the value of UP on both sender side and receiver side. The counter ECT and the counter DCT are used for recording the number of encryption processes have been completed with the current key on the sender side and the receiver side, respectively.

4.2 Secure Key Generation and Storage

The key update scheme requires multiple keys, so the security of key generation and key storage is critical. The keys can be generated on the fly using embedded structures such as a strong PUF [22], or can be stored in non-volatile memory or the secure memory on processor. The memory on processor is vulnerable to readout using the test structures such as scan-chain [28] and JTAG [29] used for testing the hardware. Even with the secure design components, the keys may be vulnerable to side-channel analysis techniques that reveal key location or behavior during execution [10].

In this work, we integrate the TPM chip with the FPGA fabric which provides secure key generation and storage

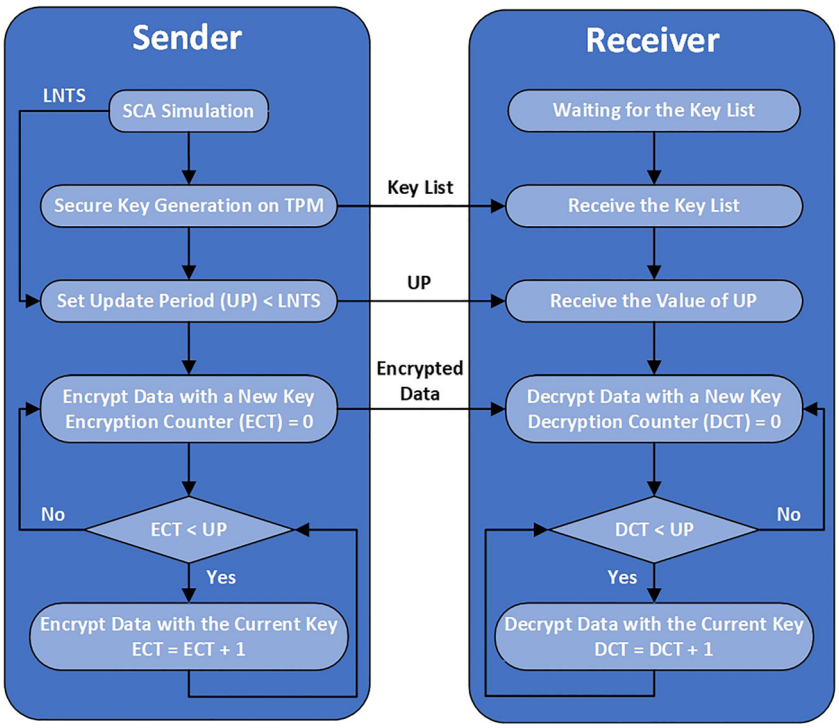

Fig. 2 Proposed key update scheme

for the key update scheme. TPM supports encryption and authentication, also has a tamper-resistant non-volatile memory for key storage. The integration can be done at different levels of abstraction that is hardware bare metal or supported to core operating system functions. We demonstrate the integration on the Microblaze-based system, consisting of a PL-based Serial Peripheral Interface (SPI) core that is connected to the Microblaze with the help of the AXI interconnect. A memory interface generator (MIG) is used to connect the onboard DDR RAM. Pins from the SPI core are connected to the TPM's SPI interface.

To provide the support for the TPM chip, we set up the Microblaze core with Petalinux and interface with TPM interface specification (TIS). The TIS is provided by the device file, on which the wrappers are defined using the TPM software stack (TSS) [30, 31]. The TSS is cross-compiled and set up on the target system.

Figure 3 shows the integration of FPGA design fabric with the encryption (AES) engine and the controller over the SPI interface with TPM for the secure key generation and storage. The controller coordinates the co-work between the FPGA fabric and the TPM throughout the encryption process, including communication with the TPM and key update. All the keys in the key list are generated by the built-in TRNG and stored in the tamper-resistant NVM on the TPM chip. All the data communications between the FPGA fabric and the TPM chip, including instruction transmission and key exchange, are realized through the SPI interface.

5 Experimental Setup

5.1 Power Analysis

To capture and collect power traces of the AES encryption engine process on hardware, the following devices are used in this work:

- Sakura-X experimental board [32] which has two onboard FPGA chips: a Kintex-7 chip where the AES-128 implemented and a Spartan-6 chip for controlling and triggering
- LNA-1050 low noise amplifier [33]
- DSA 70404C oscilloscope [34]
- E3612A DC power supply [35]
- Agilent N2862B Passive Probe [36]

The experimental setup of power capture is shown in Fig. 4. The AES-128 encryption engine is implemented on the Kintex-7 FPGA chip and the trigger is implemented on the Spartan-6 FPGA chip. The power consumption is captured and recorded by the oscilloscope as shown in Fig. 5.

5.2 Electromagnetic Attack

For collecting EM radiation, we used CW505 Planar H-field EM probe [37]. The setup of EM capture is shown in Fig. 6.

To test the resilience to EM attack, we collected 30,000 EM traces with the same set of plaintexts and keys used in power collection for performing CEMA attack. Figure 7 shows an example of EM trace of the whole AES encryption process.

5.3 TPM Configuration

The process of key generation and storage is performed in a trusted environment on the TPM. All the keys used in the key update scheme are generated by the TRNG on the TPM and stored in the non-volatile memory.

Figure 8 shows the experimental setup of TPM integration on the Sakura-X board. TPM supports several interfaces such as Low Pin Count (LPC), SPI, and I2C. The TPM chip used in this work is Infineon OPTIGATM TPM 2.0 SLB9670 which is encapsulated in Iridium 9670 Evaluation Boards [38] and connected with the Sakura-X board via the SPI port.

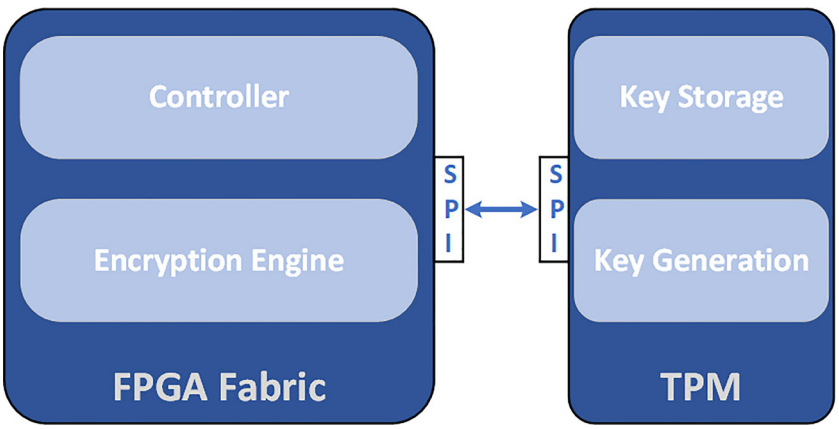

Fig. 3 Integration of FPGA fabric and TPM

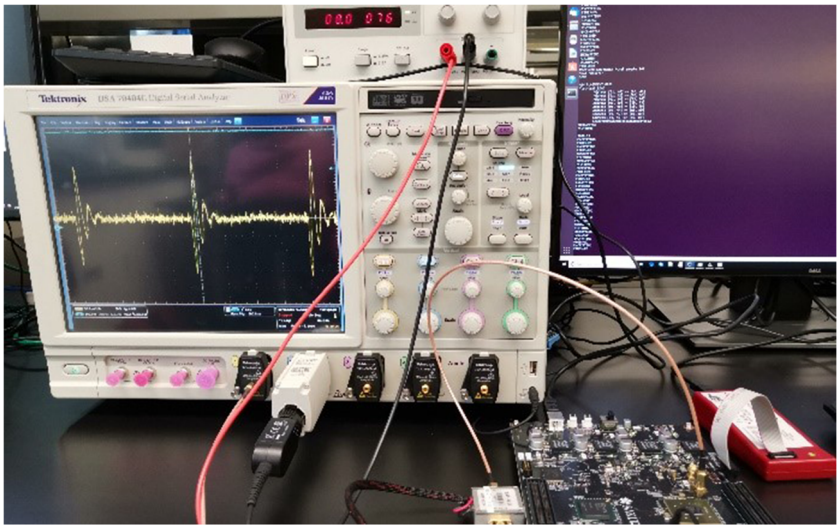

Fig. 4 Experimental setup of power capture

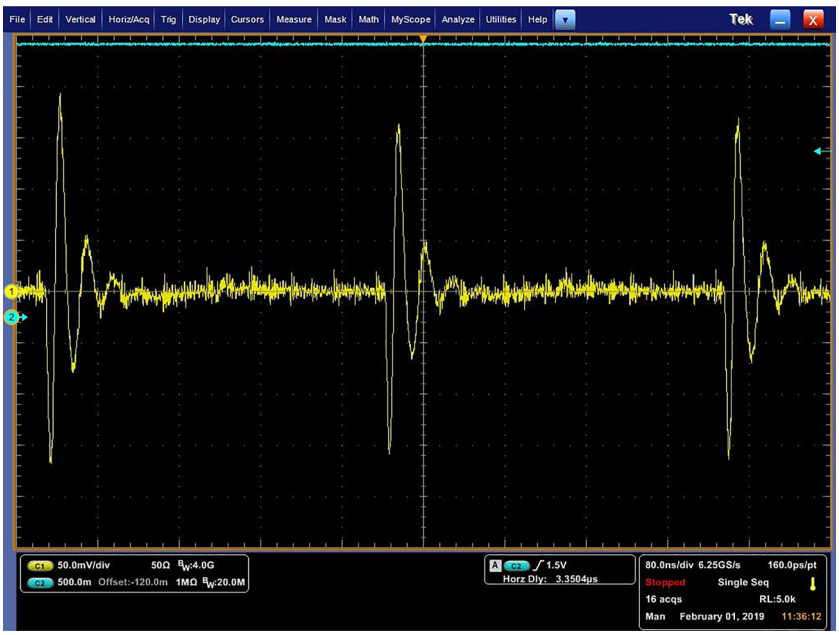

Fig. 5 Power trace of first three rounds in encryption

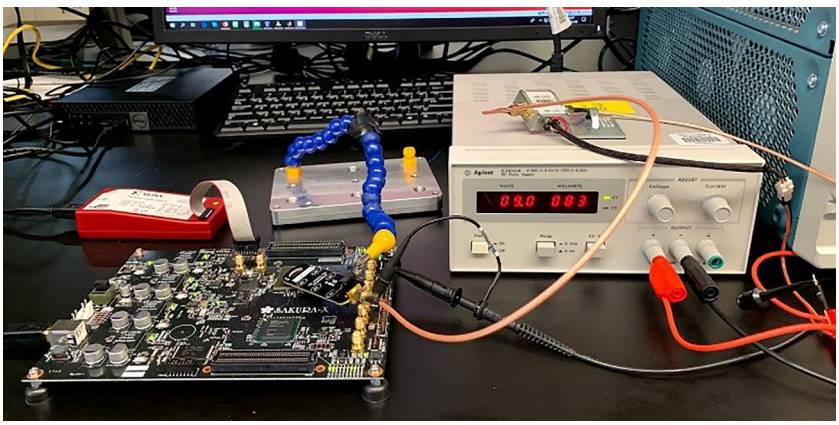

Fig. 6 Experimental setup of EM capture

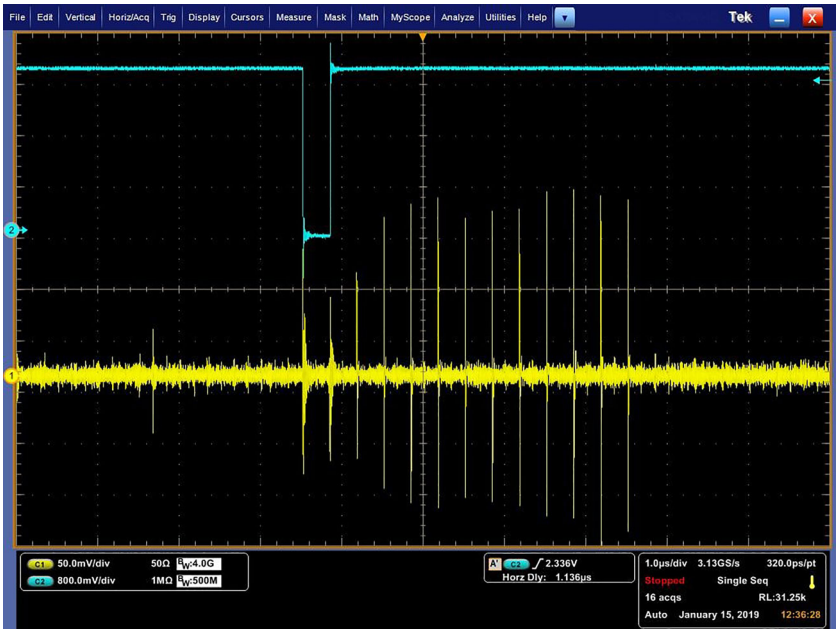

Fig. 7 EM trace of AES encryption. The yellow line is EM radiation of the chip and the blue line is the trigger signal

The high quality of random bit-sequences generated by the TRNG on the TPM chip has been proved by the NIST test in the previous work [39]. A software driver was written to provide the support for the integration.

6 Experimental Results

6.1 Power Analysis

We first applied the CPA attack on the first subkey of the AES key with the collected power traces. We used four random keys and ran the AES-128 encryption with the same set of 30,000 random plaintexts for each key.

Figure 9 shows the result of CPA attack on the first subkey used in the AES encryption with four different keys. The first subkey of the 1^{st} key 1D 22 BF 01 AC 77 D9 21 EA 34 15 F5 36 89 10 A2 is revealed correctly with around 7000 power traces, and the first subkeys of the 2^{nd} key (F0 1E D2 3C B4 5A 96 78 09 AF 81 EB 27 CD 1F A9), the 3^{rd} key (97 45 C3 73 1D AD 77 B1 17 B5 76 F4 5B 4C 1E E0), and the 4^{th} key (2B 7E 15 16 28 AE D2 A6 AB F7 15 88 09 CF 4F 3C) can also be revealed with around 5000, 6000, and 5000 power traces, respectively.

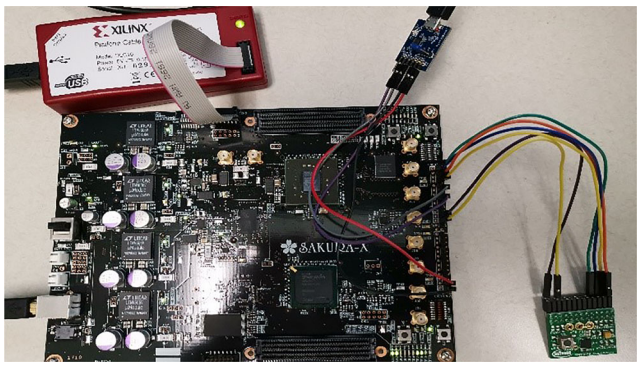

Fig. 8 TPM configuration on the Sakura-X board

6.2 Electromagnetic Analysis

We also applied CEMA on the first subkey of the AES key with the collected EM traces to explore the efficiency of the EM-based side-channel attack. The EM traces collected for the CEMA attack used the same set of plaintexts and the 4^{th} key used in the CPA attack (2B 7E 15 16 28 AE D2 A6 AB F7 15 88 09 CF 4F 3C). The result is shown in Fig. 10.

For the 4^{th} key, the LNTS of EMA attack is around 15,000 which is much higher than LNTS of CPA (5000). There are two reasons: (1) The EM probe used in this experiment does not have enough sensitivity. (2) The level of environmental noise is too high.

However, if the environmental noise can be isolated and the probe is sensitive enough, the CEMA attack can reach the same level of efficiency as CPA [7]. The EM-based side-channel attack is more threatening because it is a non-contact attack comparing with power analysis attacks.

6.3 Applying Key Update

To mitigate the risk of the side-channel attack, we applied the proposed key update scheme. Based on the result of CPA attack on AES-128 encryption with different keys, the lowest LNTS is around 5000 and the highest LNTS is

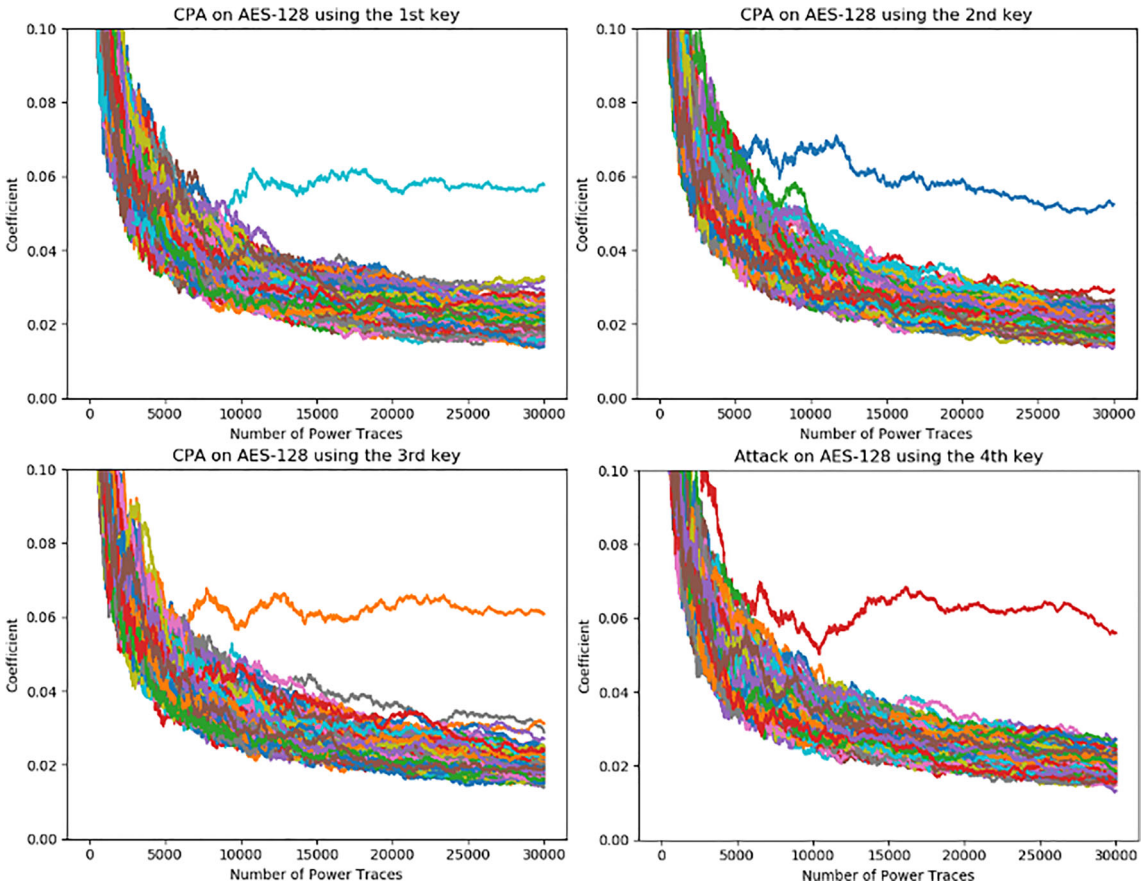

Fig. 9 The result of CPA attack on the first subkey used in the AES encryption with four different keys

around 7000 which is much less than the CEMA attack on EM traces collected with the same key and the same set of plaintexts. (Under ideal conditions, the CEMA attack can reach to the same level of efficiency as CPA [7].) If the proposed key update scheme can mitigate the CPA attack, it must also be efficient on CEMA mitigation. To remove the influence of random deviation, we set the value of the update period (UP) to 3000 traces (40% less than the lowest LNTS). The sender begins the encryption process with the first random key, then changes the key to the next one after every 3000 full encryption processes following the loop order:

$$1^{st}key \rightarrow 2^{nd}key \rightarrow 3^{rd}key \rightarrow 4^{th}key \rightarrow 1^{st}key \cdots\cdots$$

The receiver also updates the key following the same order shared by the sender for data decryption. We applied the same CPA attack on the collected power traces with the same set of plaintexts using key update scheme to verify the effectiveness. The 1^{st} key was used for encryption in three time periods in this experiment (1^{st}-3000^{th}, 12001^{st}-15000^{th}, 24001^{st}-27000^{th}), it means that totally 9000 power traces using the 1^{st} key are collected for CPA attack. Similarly, the 2^{nd}, 3^{rd}, and 4^{th} keys are used for encryption at regular intervals with 9000 (3001^{st}-6000^{th}, 15001^{st}-18000^{th}, 27001^{st}-30000^{th}), 6000 (6001^{st}-9000^{th}, 18001^{st}-21000^{th}), and 6000 (9001^{st}-12000^{th}, 21001^{st}-24000^{th}) random plaintexts, respectively.

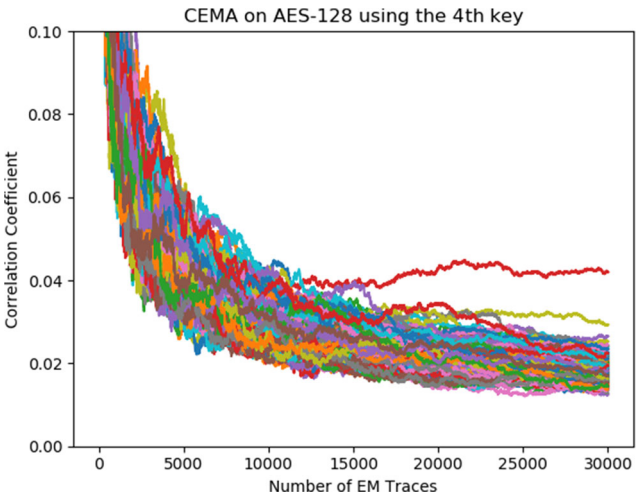

Fig. 10 The result of the CEMA attack on the first subkey of the 4^{th} key used in the AES encryption

The result is shown in Fig. 11. After applying the proposed key update scheme, none of the subkeys is revealed even with up to 30,000 power traces totally, and up to 9000 traces for a single key (the 1^{st} and the 2^{nd} key). In contrast, the first subkeys of the 1^{st} key and the 2^{nd} key can be extracted correctly with around 7000 power traces and 5000 power traces without applying the proposed key update scheme. This means that, even with a deterministic update order, the proposed key update scheme is still secure to mitigate correlation-based attacks because the accumulative correlation model built with previous keys is disturbed continuously every time the new key is applied.

6.4 Key Generation on TPM

Figure 12 shows the process of key generation on TPM. In this experiment, 8 random keys were generated by the TRNG which can be used for the key update scheme in the encryption process. The average time to generate 8 random keys is 0.014 s in 100 runs. The size of the key list and the length of each key are controllable for meeting different security needs.

7 Security Analysis

The encryption engine during execution is vulnerable to side-channel attacks and has been shown in Section 6. The vulnerability of cryptographic devices roots in the high correlation between the leaked information and the static implementation of the encryption engine. In this work, we propose a key update scheme which is resilient to side-channel attacks.

One advantage of the proposed scheme is that the strength of security is completely controllable by changing the length of the key list, modifying the update order, or adjusting the update period. A longer key list (more random keys) or a higher update frequency (reduce the update period) can enhance the resilience to SCA attacks further, but also leads to higher overhead. To ensure the security of the proposed scheme, in this work, the sharing process of key list is expected to be performed in a trusted environment before the data communication process.

In [20, 21], Medwed et al. propose a re-keying scheme that generates random keys using a key derivation function. However, the key derivation function is implemented on the same fabric with the encryption engine which brings an extra area overhead and risk of tampering attack. The state-of-the-art FPGA devices natively support key rolling to encrypt the bitstream which allows the user to break up the bitstream into multiple AES encryption messages, each encrypted with its own unique rolling key which are derived from the initial key [40]. However, the on-chip AES logic cannot be used for any purpose other than bitstream encryption/decryption and the initial key is stored in the RAM or eFUSE which is still readable by laser stimulation techniques [10]. Moreover, the size of bitstream and the time delay are greatly increased along with activating the key rolling scheme. By comparison, the proposed key update scheme in this work is a general solution scheme and all the keys are generated based on a primary key which is never visible outside of the TPM [41]. In [42], a shifter is used for producing randomness for the key rotation scheme. However, the shifting-based random number generator can only produce pseudo-random numbers. In contrast, we use the built-in TRNG on the TPM to produce true random numbers and the quality has been proved by the NIST test [39]. The use of true random numbers can enhance the strength of key update scheme. To protect the process of key generation, [22] uses a strong PUF to generate keys based on the subthreshold current array proposed in [43].

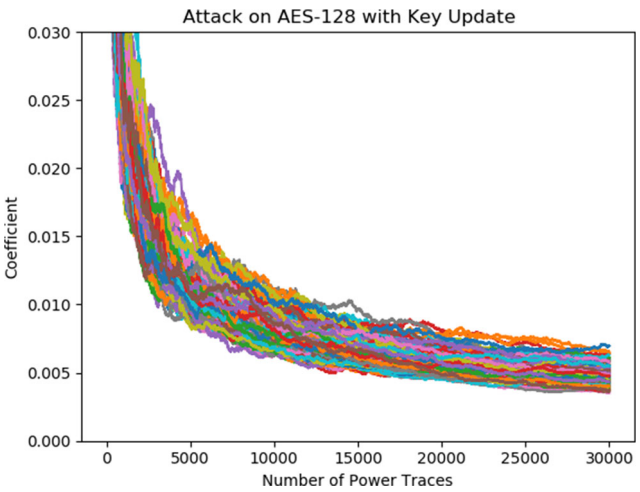

Fig. 11 The result of CPA attack on the first subkey used in the encryption after applying the key update scheme

```
$ ./tpm_keygen.sh
Generating Primary Key
ObjectAttribute: 0x00030072

CreatePrimary Succeed ! Handle: 0x800000ff

Generating Symmertic keys
77D809A16E13C11613F6A2F3F57D3ADD
01662F482BE0BE86C5E142B3541B5FB9
72F0C957178C96ECA600CDB04596F110
2610E9B66AE20A7F4EA7549BA5316F96
9AB4C746E8E898FE992B23BE68B672E5
1EC5BB56BEBF2B6514CF9F88D394BD90
6EC57619896BB1C4F8A9594864049DEB
B7B33CEC5BE58D46680FADEC6413AF40
```

Fig. 12 The process of key generation on TPM

The proposed PUF shows good performance but it is only resistant to simple power analysis.

As an efficient countermeasure, masking is widely used for mitigating side-channel attacks. In [44], an order 1 perfectly masked algorithm is presented which masks original secret data with an additively masked value to reduce the correlation between the intermediate values and the input. To increase the efficiency and the security of masking, threshold implementation (TI) is proposed in [23] which combines the ideas of secret sharing, threshold cryptography, and multi-party computation protocols. The original secret data is divided into multiple shares using Boolean addition and processed independently, and cannot be revealed unless the number of leaked shares is higher than the preset threshold. However, the area overhead of threshold implementation is very high. For example, the area overhead after applying the countermeasure based on threshold implementation proposed in [45] is higher than 350%.

In this work, an independent TPM chip is used for key generation. All the keys are generated by the TRNG and stored in the tamper-resistant NVM on the TPM chip so that the risk of tampering attack is reduced significantly. The area overhead of the proposed design is incurred by the storage for multiple keys used in the key update scheme and depends on the length of each key and the key list. All the keys are stored on the TPM chip, so there is no extra area overhead incurred by key storage on the FPGA fabric. For AES-128, the size of each key is 16 bytes. The size of NVM on SLB9670 TPM2.0 chip is 6962 bytes [46] which is able to store up to 435 AES-128 keys. For time overhead, the result shows that the average time to generate one random key is less than 2 ms. Considering the enhancement of security brought by the proposed scheme and the TPM chip, the overhead is fairly small. In addition, TPM supports different sizes and types of keys (RSA, ECC, and AES). This feature makes the proposed scheme more flexible and practical in different scenarios to fulfill various users' needs.

8 Limitations

As shown in Fig. 3, the FPGA fabric communicates with the TPM chip via the SPI interface, including sending commands and key exchange. Currently, the SPI interface and the communication process are unprotected as shown in Fig. 8. As a result, the attacker can perform the eavesdropping attack on the communication process directly to steal the secret data.

To mitigate this risk, one practical solution is to integrate the TPM chip into the system on chip (SoC). A commercial form of this integration is Intel Platform Trust Technology (PTT) which implements TPM in system firmware. PTT supports full TPM 2.0 specification but uses the existing processor on the SoC. Benefit from the highly integrated nature of PTT, the difficulty of performing eavesdropping attacks on TPM is increased significantly. Integrating TPM into the main fabric can also reduce the risk of optical attack based on thermal laser stimulation (TLS) because the backside of the TPM chip is not directly exposed anymore after integration.

9 Conclusion

In this paper, a key update scheme is proposed as a countermeasure for side-channel attacks. The keys are updated during short intervals to reduce the correlation and dependence between the leaked information and the secret key. By calculating the least needed power/EM traces (LNT) of the target device and updating the key before any subkey can be revealed on each node synchronously, the risk of the power analysis attack and the EM analysis attack is mitigated as shown in the experiments.

To protect keys from tampering attacks, the keys are generated and stored securely on the TPM. The length of each key and the number of keys are controllable which means that the proposed scheme supports various encryption standards and the security strength is flexible to meet different demands.

Funding Information This research has been sponsored by the National Science Foundation under Grant Nos. 1814420, 1819694, and 1819687.

References

1. Kocher PC (1996) Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In: Advances in cryptology — CRYPTO '96, vol 1109, pp 104–113. https://doi.org/10.1007/3-540-68697-5_9
2. Fournaris AP, Koufopavlou O (2012) Protecting CRT RSA against fault and power side channel attacks. In: 2012 IEEE Computer Society Annual Symposium on VLSI, pp 159–164. https://doi.org/10.1109/isvlsi.2012.54
3. Prouff E, Rivain M (2013) Masking against side-channel attacks: a formal security proof. In: Advances in cryptology - EUROCRYPT 2013, vol 7881, pp 142–159. https://doi.org/10.1007/978-3-642-38348-9_9
4. Masoumi M, Habibi P, Jadidi M (2015) Efficient implementation of masked AES on side-channel attack standard evaluation board. In: 2015 International Conference on Information Society (i-Society), pp 151-156. https://doi.org/10.1109/i-society.2015.7366878
5. United States National Institute of Standards and Technology (NIST) (2001) Announcing the Advanced Encryption Standard (AES). In: Federal Information Processing Standards Publication 197

6. Kocher PC, Jaffe J, Jun B (1999) Differential power analysis. In: Advances in cryptology — CRYPTO '99, vol 1666, pp 388–397. https://doi.org/10.1007/3-540-48405-1_25
7. Nomata Y, Matsubayashi M, Sawada K, Satoh A (2016) Comparison of side-channel attack on cryptographic cirucits between old and new technology FPGAs. In: 2016 IEEE 5th Global Conference on Consumer Electronics, pp 1–4. https://doi.org/10.1109/gcce.2016.7800555
8. Gandolfi K, Mourtel C, Olivier F (2001) Electromagnetic analysis: concrete results. In: Cryptographic hardware and embedded systems - CHES 2001, vol 2162, pp 251–261. https://doi.org/10.1007/3-540-44709-1_21
9. Ling Z, Luo J, Zhang Y, Yang M, Fu X, Yu W (2012) A novel network delay based side-channel attack: modeling and defense. In: 2012 Proceedings IEEE INFOCOM, pp 2390–2398. https://doi.org/10.1109/INFCOM.2012.6195628
10. Lohrke H, Tajik S, Krachenfels T, Boit C, Seifert JP (2018) Key extraction using thermal laser stimulation: a case study on Xilinx Ultrascale FPGAs. In: Cryptology ePrint Archive, Report 2018/717. https://eprint.iacr.org/2018/717
11. Brier E, Clavier C, Olivier F (2004) Correlation power analysis with a leakage model. In: Cryptographic hardware and embedded systems - CHES 2004, vol 3156, pp 16–29. https://doi.org/10.1007/978-3-540-28632-5_2
12. Bu A, Dai W, Lu M, Cai H, Shan W (2018) Correlation-based electromagnetic analysis attack using Haar wavelet reconstruction with low-pass filtering on an FPGA implementaion of AES. In: 2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/ 12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE), pp 1897–1900. https://doi.org/10.1109/TrustCom/BigDataSE.2018.00288
13. Mulder ED, Eisenbarth T, Schaumont P (2018) Identifying and eliminating side-channel leaks in programmable systems, vol 35, pp 74-89, https://doi.org/10.1109/mdat.2017.2766166
14. Madlener F, Sotttinger M, Huss SA (2009) Novel hardening techniques against differential power analysis for multiplication in GF(2^n). In: 2009 International Conference on Field-Programmable Technology, pp 328–334. https://doi.org/10.1109/fpt.2009.5377676
15. Huss SA, Sotttinger M (2017) A novel mutating runtime architecture for embedding multiple countermeasures against side-channel attacks. In: Hardware IP security and trust, pp 165-184. https://doi.org/10.1007/978-3-319-49025-0_8
16. Popp T, Mangard S (2005) Masked dual-rail pre-charge logic: DPA-resistance without routing constraints. In: Cryptographic hardware and embedded systems – CHES 2005, pp 172-186. https://doi.org/10.1007/11545262_13
17. Tiri K, Akmal M, Verbauwhede I (2002) A dynamic and differential CMOS logic with signal independent power consumption to withstand differential power analysis on smart cards. In: Proceedings of the 28th European Solid-State Circuits Conference, pp 403–406. https://ieeexplore.ieee.org/document/1471550
18. Hwang DD, Tiri K, Hodjat A, Lai BC, Yang S, Schaumont P, Verbauwhede I (2006) AES-based security coprocessor IC in 0.18-muhboxmCMOS with resistance to differential power analysis side-channel attacks, vol 41, pp 781-792, https://doi.org/10.1109/JSSC.2006.870913
19. Mankar P (2017) Key updating for leakage resiliency with application to Shannon security OTP and AES modes of operation. In:2017 International Conference on IoT and Application (ICIOT), pp 1–4. https://doi.org/10.1109/ICIOTA.2017.8073631
20. Medwed M, Standaert F, Großschädl J, Regazzoni F (2010) Fresh re-keying: security against side-channel and fault attacks for low-cost devices. In: Progress in cryptology - AFRICACRYPT 2010, pp 279–296. https://doi.org/10.1007/978-3-642-12678-9_17
21. Medwed M, Petit C, Regazzoni F, Renauld M, Standaert F (2011) Fresh re-keying II: securing multiple parties against side-channel and fault attacks. In: Smart card research and advanced applications, pp 115–132. https://doi.org/10.1007/978-3-642-27257-8_8
22. Xi X, Aysu A, Orshansky M (2018) Fresh re-keying with strong PUFs: a new approach to side-channel security. In: 2018 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), pp 118–125. https://doi.org/10.1109/hst.2018.8383899
23. Nikova S, Rechberger C, Rijmen V (2006) Threshold implementations against side-channel attacks and glitches. In: ICICS 2006: information and communications security, pp 69–83. https://doi.org/10.1007/11935308_38
24. Trusted Computing Group (2011) TPM structures. https://trustedcomputinggroup.org/wp-content/uploads/TPM-Main-Part-2-TPM-Structures_v1.2_rev116_01032011.pdf. Accessed 27 February 2019
25. Gui Y, Siddiqui AS, Saqib F (2018) Hardware based root of trust for electronic control units, vol 2018, pp 1-7, https://doi.org/10.1109/SECON.2018.8479266
26. Siddiqui AS, Gui Y, Lawrence D, Laval S, Plusquellic J, Manjrekar M, Chowdhury B, Saqib F (2018) Hardware assisted security architecture for smart grid, vol 2018, pp 2890-2895, https://doi.org/10.1109/IECON.2018.8591401
27. Messerges TS, Dabbish EA, Sloan RH (1999) Investigations of power analysis attacks on smartcards. In: Proceedings of USENIX Workshop on Smartcard Technology, pp 151–161
28. Jin Y (2015) Introduction to hardware security, vol 4, pp 763–784, https://doi.org/10.3390/electronics4040763
29. Rosenfeld K, Karri R (2010) Attacks and defenses for JTAG, vol 27, pp 36-47, https://doi.org/10.1109/MDT.2010.9
30. Intel (2018) The TPM2 software stack: introducing a major open source release. https://software.intel.com/en-us/blogs/2018/08/29/tpm2-software-stack-open-source Accessed 04 October 2019
31. IBM (2019) IBM's TPM 2.0 TSS. https://sourceforge.net/projects/ibmtpm20tss/ Accessed 04 Oct 2019
32. Satoh Laboratory (2014) SAKURA-X. http://satoh.cs.uec.ac.jp/SAKURA/hardware/SAKURA-X.html Accessed 11 March 2019
33. RF BAY INC (2006) LNA-1050. http://www.rfbayinc.com/upload/files/lna/lna-1050.pdf Accessed 11 March 2019
34. Tektronix (2013) DPO/DSA/MSO70000 Series Oscilloscopes. http://download.tek.com/document/55W-22447-9.pdf Accessed 11 March 2019
35. Keysight (formerly Agilent's Electronic Measurement) (2007) E3612A 30W Power Supply, 60V, 0.5A or 120V, 0.25A. https://www.keysight.com/en/pd-838247-pn-E3612A Accessed 11 March 2019
36. Keysight (2018) N2862B Passive Probe, 10:1, 150 MHz, 1.2 m. https://www.keysight.com/en/pd-1938439-pn-N2862B/
37. ChipWhisperer Wiki (2018) CW505 Planar H-Field Probe. https://wiki.newae.com/CW505_Planar_H-Field_Probe Accessed 11 March 2019
38. Infineon (2017) IRIDIUM SLB 9670 TPM2.0 LINUX. https://www.infineon.com/cms/en/product/evaluation-boards/iridium9670-tpm2.0-linux Accessed 16 March 2019
39. Suciu A, Carean T (2010) Benchmarking the true random number generator of TPM chips. In: CoRR. arXiv:1008.2223 Accessed 10 April 2019

40. Xilinx (2018) Using encryption and authentication to secure an UltraScale/UltraScale+ FPGA Bitstream. https://www.xilinx.com/support/documentation/application_notes/xapp1267-encryp-efuse-program.pdf Accessed 09 October 2019
41. Trusted Computing Group (2013) Endorsement Key (EK) and Platform Certificate Enrollment Specification Frequently Asked Questions. https://trustedcomputinggroup.org/wp-content/uploads/IWG-EK-CMC-enrollment-for-TPM-v1-2-FAQ-rev-April-3-2013.pdf
42. Wang A, Wang C, Zheng X, Tian W, Xu R, Zhang G (2017) Random key rotation: side-channel countermeasure of NTRU cryptosystem for resource-limited devices. Computers & Electrical Engineering 63:220–231. https://doi.org/10.1016/j.compeleceng.2017.05.007
43. Kalyanaraman M, Orshansky M (2013) Novel strong PUF based on nonlinearity of MOSFET subthreshold operation. In: 2013 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), pp 18–23. https://doi.org/10.1109/HST.2013.6581558
44. Blömer J, Guajardo J, Krummel V (2004) Provably secure masking of AES. In: SAC 2004: selected areas in cryptography, pp 69–83. https://doi.org/10.1007/978-3-540-30564-4_5
45. Moradi A, Poschmann A, Ling S, Paar C, Wang H (2011) Pushing the limits: a very compact and a threshold implementation of AES. In: Advances in Cryptology - EUROCRYPT 2011, pp 69–89. https://doi.org/10.1007/978-3-642-20465-4_6
46. Infineon (2018) OPTIGA™ TPM SLB 9670 TPM2.0 Data Sheet. https://www.infineon.com/dgdl/Infineon-data-sheet-SLB9670_2.0_Rev1.3-DS-v01_03-EN.pdf?fileId=5546d462689a790c016929ed3b5e4ffb Accessed 29 April 2019