DPA Protected Implementation of OCB and COLM Authenticated Ciphers

Mohsen Jahanbani¹, Zeinolabedin Norozi¹, Nasour Bagheri²,³

¹ Department of Electrical Engineering, Imam Hossein University - Tehran- Iran, (e-mail: znorozi@ihu.ac.ir.)
² Department of Electrical Engineering, Shahid Rajaee Teacher Training University- Tehran- Iran
³ School of Computer Science, Institute for Research in Fundamental Sciences (IPM), Tehran, Iran

Corresponding author: Nasour Bagheri (e-mail: nbagheri@srttu.edu).

ABSTRACT Authenticated ciphers are designed to provide two security requirements simultaneously, i.e. confidentiality and integrity. The CAESAR competition was ended with introducing six authenticated ciphers for several applications as the winners. The OCB and COLM authenticated ciphers are two AES-based winners, respectively for high-speed and defense in-depth applications. Similar to the implementation of any other cryptographic algorithm, unprotected implementations of these ciphers could also be vulnerable to side-channel attacks, especially differential power analysis (DPA). In this work, first, the OCB and COLM ciphers are implemented on FPGA of SAKURA-G board. Then their vulnerability is shown with power leakage detection using t-test over the power traces. Also, the first-order protected version of these ciphers is presented using two masking scheme, i.e. threshold implementation (TI) and domain-oriented masking (DOM). To verify these countermeasures, the first and second-order t-test is conducted, to indicate the resistance of protected schemes to the first-order DPA attacks. Finally, the hardware implementation of two protected and unprotected versions of ciphers on FPGA are benchmarked based on the criteria of area, maximum frequency, and throughput. Additionally, the ratio of the increased area and decreased throughput to the unprotected ciphers have been compared with previous works.

INDEX TERMS Authenticated cipher, OCB, COLM, Threshold implementation, Domain-oriented masking.

I. INTRODUCTION

Authenticated encryption (AE) provides two security objectives, i.e. confidentiality and integrity of plaintext simultaneously. The most important reason to use these ciphers is to prevent message forgery. By adding authentication service to the message, the received ciphertext does not accept as it is and it will be verified to see whether it has been manipulated or forged. A large number of application and protocols need those two services simultaneously. Until recently, these two services were provided separately. Unfortunately, traditional methods to design an AE scheme have low efficiency because they are two passes and need two separate keys. Motivated by the CAESAR competition, several new AE schemes have been proposed in literature that are based on block ciphers, stream ciphers, permutations, or dedicated schemes (in January 2013, a competition began to design AE ciphers with the support of NIST called CAESAR competition with two objectives: widespread application and superiority over GCM). In the block ciphers-based AE schemes, one of the block ciphers off-the-shelf, e.g., AES, is used in a particular mode as a black box. For example, SP800-38C standard [1] suggests CCM (Counter with CBC-MAC) [2] with AES cipher. For high-performance application, SP800-38D standard [3] suggest the GCM (Galois/Counter mode). Currently, these two schemes are widely used, but unfortunately, none of these schemes are efficient and multi-purpose [4].

In 2019, the CAESAR competition ended with six winners in three use-cases. The cryptographic algorithms that are selected through the competition process and evaluated publicly are usually resistant to known statistical attacks, while yet they may be vulnerable in an unprotected hardware implementation against side-channel attacks. Hence, protecting the equipment that implementing the cryptographic algorithms against side-channel attacks is an important issue. These attacks use computation time, power consumption, or electromagnetic radiations to extract sensitive information such as the keys. One of the major kind of the side-channel attacks is the differential power analysis (DPA) attack [5] and its improved version, i.e., correlation power analysis (CPA) attack [6]. Several techniques have
been suggested to protect the hardware [7]–[15]. These techniques are classified into two groups of hiding and masking [8]. Masking schemes have received lots of attention in the literature. In these schemes, by randomization the intermediate values, power consumption becomes independent of the values being processed.

In this paper, we consider the security analysis of two CAESAR winners, i.e., OCB [16] and COLM [17] against differential power analysis and also provide the required countermeasures.

A. CONTRIBUTION

The OCB [16] and COLM [17] ciphers are the only winners of CAESAR competition that use AES cipher with full round as primitive. Although no security vulnerabilities have been reported for these ciphers, vulnerabilities to power analysis should be researched. As the input and output of AES in these ciphers are masked with ∆ values, the power analysis attack is not trivial and is a challenging task. Hence, as our primer aim, we will show the vulnerability of these selected ciphers against side-channel attacks, the key recovery is not considered in this work. This issue can be investigated using leakage detection methods such as the t-test. Therefore, in this research, the unprotected version of these ciphers are implemented on FPGA and t-test is conducted. In the proposed protected architectures, the nonlinear part, i.e., AES S-box is protected via TI masking scheme using hybrid 2/3 shares and DOM masking scheme using two shares, and all linear parts are repeated as many times as the number of shares. Then the power leakage is investigated using the first and second-order t-test. Finally, the protection costs are calculated and compared with others.

B. ORGANIZATION

The rest of the paper is organized as follows: Section II is associated with the literature review, including masking schemes, description of OCB and COLM ciphers, power leakage assessment, and related works. In Section III, the protected version of OCB and COLM schemes are presented and implemented with an 8-bit architecture in hardware. The experimental results of power leakage on protected and unprotected OCB and COLM and the cost of protecting them are reported in Section IV. The conclusion and suggestions for further research are presented in Section V.

II. BACKGROUND INFORMATION

In this section, the two masking schemes, leakage assessment, OCB and COLM ciphers specifications, and related works to protect winners of CAESAR competition against side-channel attacks are described.

A. MASKING SCHEMES

A masking scheme describes a secret sharing scheme to break sensitive values and defines the secure operation on shares as multi-party computation (MPC). Boolean masking uses XOR operator for operation between shares in a way that the sensitive value of x ∈ GF(2^m) is broken down to d+1 shares (x_1,...,x_d) such that x = ∑_{i=1}^{d+1} x_i. Masking schemes can be implemented in software or hardware. Due to the existence of physical faults such as glitches in hardware, the assumption of independence, which is required for secure masking, is removed. Therefore, the efforts began to design glitches-resistant schemes.

Threshold Implementation (TI): One of the glitch resistant approaches is TI [9], which is algorithmically simple and resistant to first-order DPA attacks. If the number of shares is s, the lower bound of the number of input and output shares is a function of required security order d and the algebraic degree of non-linear function t, which is calculated based on Equation (1) [10]:

\[
\text{s}_{in} \geq td + 1, \quad \text{s}_{out} \geq \left(\frac{s_{in}}{t}\right)
\]

A TI scheme is secure if the three properties correctness, non-completeness, and uniformity made are satisfied as follows:

Correctness: suppose TI representation of the function \( \tilde{y} = F(\tilde{x}) \) is intended, which \( \tilde{x} = (x^1, ..., x^s) \) is the shared representation of input x with s shares such that x = ∑_{i=1}^{s} x_i, then for output shares \( \tilde{y} = (y^1, ..., y^t) \) we have y = ∑_{i=1}^{t} y_i to ensure correctness property. For this purpose, we can use component functions \( f_{(1, 0)}(\tilde{x}) = y^i \) to calculate F. Although, it is not easy to find these component functions when function F is non-linear.

Non-completeness: The function F is d^{th}-order non-complete if each combination of d component functions \( f^i \) is independent of at least one input share \( x^j \), where \( d \) is a security order.

Uniformity: the security of a Boolean masking scheme is based on the uniform distribution of the shares and masks. If the input of a TI function is shared uniformly, the output should be a uniform sharing as well, because the output of a nonlinear function such as S-box is used as the input of the next function (for example the next round of cipher). Hence, given all possible input shares \( X = \{\tilde{x} | \bigoplus_{i=1}^{s} \tilde{x} = x\} \), all output shares set \( \{f^1, ..., f^n | \tilde{x} = x\} \) should be selected uniformly from set \( Y = \{\tilde{y} | \bigoplus_{i=1}^{t} \tilde{y} = y\} \) as all possible shares of y = F(x).

Providing uniformity for nonlinear functions, especially functions with the high algebraic degree is a challenge ahead of this structure. One solution is the re-masking using uniform fresh random in case of the non-uniform output of component functions [8], [10]. This value is produced by a pseudo-random number generators (PRNG). For example, if the TI version of multiplication \( z = xy \in GF(2^m) \) is desirable then products \( x_iy_i \), for \( i, j \in \{1, 2, ..., d + 1\} \), are calculated, i.e. \( (d + 1)^2 \) components are computed. To calculate the output, the number of output shares should be compressed securely. The first-order TI for this multiplication with \( t = 1, d = 1, s_{in} = s_{out} = 3, r = 2 \) is described as Equation (2) [8]:

\[
\begin{align*}
z_1 &= x_1y_1 \oplus x_1y_2 \oplus x_2y_1 \oplus r_1 \oplus r_2 \\
\end{align*}
\]

\[
\begin{align*}
z_2 &= x_2y_1 \oplus x_1y_3 \oplus x_3y_2 \oplus r_1 \\
z_3 &= x_3y_2 \oplus x_1y_3 \oplus x_3y_1 \oplus r_2
\end{align*}
\]
Note that three shares have been used to achieve the first-order security and uniformity was obtained by adding two fresh random \( r_1 \) and \( r_2 \).

**Domain-oriented masking (DOM):**
Another masking scheme that provides the same level of security as TI, while it requires less area and randomness, is domain-oriented masking (DOM). DOM can also be scaled easily to arbitrary protection orders for any circuit. DOM has based on the distribution of shares in \( d+1 \) domain such that shares in each domain are independent of the other domains. For cross-domain computations, fresh random is added to make these terms independent. Also, to prevent glitch propagating, the registers are added between domains. For example, the first-order secure AND gate is computed in three stages of calculation, resharing, and integration as shown in Fig. 1.

**FIGURE 1. First-order DOM-secure AND gate [14]**
Compared to TI, the number of shares has reduced from \( td + 1 \) to \( d + 1 \) for a \( d \)-th order security and the number of fresh random bits has reduced from \( (d + 1)^2 \) to \( d(d + 1)/2 \), with the cost of extra clocks.

**B. POWER LEAKAGE EVALUATION**
Differential power analysis (DPA) was used to analyze the difference between the measured powers and guessed power based on the power model. However, the proper selection of the power model is a difficult and time-consuming task and requires knowing underlying architecture. In recent years, a power leakage evaluation method has been proposed in [18] and [19] with the further description that used the Welch test to distinguish between two distributions. Unlike DPA, this test is independent of the basic leakage model, but it does not lead to key recovery. After collecting enough number of traces, traces are divided into two sets based on an intermediate value. In order to conduct the first-order t-test using the first-order statistical moment, t-statistics is calculated as Equation (3).

\[
t = \frac{(\mu_0 - \mu_1)}{\sqrt{s_0^2/n_0 + s_1^2/n_1}}
\]  
(3)

Where \( \mu_0 \) and \( \mu_1 \) are the mean of two distributions \( Q_0 \) and \( Q_1 \), \( s_0 \) and \( s_1 \) are the standard deviations, and \( n_0 \) and \( n_1 \) are the number of distribution samples. To use the t-test, the null hypothesis indicates that the samples are selected from one distribution and cannot be distinguished. The threshold \(|t| > 4.5\) is determined to reject the null hypothesis with the probability of 99.999%. In this case, the samples have two different distributions, and therefore, the power consumption is statistically related to intermediate values that are being processed. The non-specific t-test is a kind of test that investigates leakage detection without real attack. In this type of test, a set has constant input D (e.g., the plaintext, AD, or Npub) and another set has random input. The constant input value D or random value is selected randomly. According to [19], the next inputs are calculated as follows:

\[
i_{n+1} = f(\text{INPUT}, \text{out}_i, \text{random})
\]
(4)

Where \( \text{random} \) is equal to tossing the coin to select the input. For the first-order protected scheme, the variables are shown with two shares, and the constant input (INPUT) is broken down into shares: INPUT\(^1\), INPUT\(^2\). Then the next inputs are calculated as follows [19]:

\[
i_{n+1} = f(\text{INPUT}, \text{out}, \text{random})
\]  
\[
= \begin{cases} 
(\text{INPUT} \oplus r^1, r^1) & \text{if } \text{randombit} \text{ is } 0 \\
(r^1, r^2) & \text{if } \text{randombit} \text{ is } 1 
\end{cases}
\]  
(5)

Where \( r \) is the mask random value.

The higher-order t-test is used to evaluate the leakage of masked schemes and is conducted on preprocessed traces. For this purpose, the variances of the preprocessed traces are required to be estimated. For the second-order t-test, at first, the traces are converted to mean-free squared, i.e., \( Y = (X - \mu)^2 \), then the variance \( Y \) is estimated as follows [19]:

\[
S^2 = \frac{1}{n} \sum (X - \mu)^2 - \frac{1}{n} \sum (X - \mu)^2
\]  
(6)

**C. RELATED WORKS**
So far, scanty studies have been conducted to protect CAESAR competition schemes against side-channel attacks. These research studies are tabulated in Tab. (1). In all of these studies, the vulnerability of ciphers have been shown using DPA attack or t-test. The reviewed schemes show that most researches use TI masking as the protecting method.

<table>
<thead>
<tr>
<th>Reference</th>
<th>Cipher</th>
<th>Status in competition</th>
<th>Implementation platform</th>
<th>Attack method</th>
<th>Protecting method</th>
</tr>
</thead>
<tbody>
<tr>
<td>[20]</td>
<td>ACORN</td>
<td>Winner</td>
<td>Software (microprocessor)</td>
<td>DPA</td>
<td>dedicated masking</td>
</tr>
<tr>
<td>[21]</td>
<td>ASCON</td>
<td>Winner</td>
<td>FPGA</td>
<td>t-test</td>
<td>TI</td>
</tr>
<tr>
<td>[22]</td>
<td>Keyak</td>
<td>Third round candidate</td>
<td>FPGA</td>
<td>DPA</td>
<td>TI</td>
</tr>
<tr>
<td>[23]</td>
<td>ASCON</td>
<td>Winner</td>
<td>FPGA</td>
<td>-</td>
<td>TI</td>
</tr>
<tr>
<td>[24]</td>
<td>CLOC</td>
<td>Third round candidates</td>
<td>FPGA</td>
<td>t-test</td>
<td>TI</td>
</tr>
<tr>
<td>[25]</td>
<td>COLM</td>
<td>Winner</td>
<td>FPGA</td>
<td>CPA</td>
<td>DOM</td>
</tr>
</tbody>
</table>
D. OCB and COLM AUTHENTICATED CIPHER

Some of the important properties of the winners of CAESAR competition are shown in Tab. 2. Amongst these six winners, only OCB [16] and COLM [17] ciphers use full round AES algorithm as the underlying block cipher. The number of AES and multiplication in GF(2^{128}) in OCB ciphers is 3 + 4N_{AD} + 2N_M and 4 + 4N_{AD} + 4N_M, respectively. Also, the number of AES and multiplication in GF(2^{128}) in COLM is 4 + 4N_{AD} + 2N_M and 21 + 4N_{AD} + 2N_M, respectively (N_{AD} and N_M are the number of AD and plaintext blocks, respectively). In both ciphers, the inputs and output of AES are masked with the variable Δ. In this study, we intend to show whether these ciphers are resistant against side-channel attacks.

TABLE 2. The features of CAESAR competition winners

<table>
<thead>
<tr>
<th>Cipher</th>
<th>Primitive</th>
<th>Inv-free</th>
<th>Intermediate Tags</th>
<th>Parallel</th>
<th>Nonce-miscrypt</th>
<th>Security</th>
<th>Resistant</th>
</tr>
</thead>
<tbody>
<tr>
<td>OCB [16]</td>
<td>AES</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>AEGIS [26]</td>
<td>AES</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>COLM [17]</td>
<td>AES</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Deoxy [16]</td>
<td>AES</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>ACORN [27]</td>
<td>LFSR</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>ASCON [28]</td>
<td>SPN</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

1) OCB AUTHENTICATED CIPHER

Offset Codebook (OCB) authenticated cipher is a block cipher mode. Its initial version of OCB (OCB1) [29] was presented in 2001. Its current version (OCB3) was presented in FSE 2011 and accepted as RFC 7253 [16]. Also, OCB3 has been selected as one of the winners of CAESAR competition for high-performance applications. This cipher uses a tweakable block cipher structure that makes the power analysis attack difficult compared to block ciphers. This cipher supports key lengths of 128, 192, and 256. The plaintext is processed in 128-bit blocks and produces a 128-bit ciphertext and tags with lengths of 64, 96, and 128 bits. Fig. 3 shows the structure of OCB3. In the top section of the figure, the calculation of masking values (Δ) is shown. The nonce N is 96 bits which use 10-bit-padding to make the 128-bit block. For each encryption, the new values for the nonce are generated. If the nonce is used with the same key, the confidentiality and authenticity of the scheme will be endangered. Δ values are calculated using Fig. 2.

According to Fig. 2, if N is chosen as counter and increased by 1 for every new message, the Top and Ktop values change for every 64 messages. The ciphertext is calculated as C[i] = \text{E}_K\{M[i] \oplus Z[i] \oplus Z[i]\} for each 128-bit block of M[i] plaintext. AD is processed in the bottom part of Fig. 3 and used to calculate the final tag.

2) COLM AUTHENTICATED CIPHER

COLM authenticated cipher [17] is a block cipher mode that uses the Encrypt-Linear mix-Encrypt structure. COLM use 128-bit key and message, 64-bit tag. The security level (as log2 of the number of AES calls) for confidentiality and integrity is 64-bit. COLM refers to COLM with the intermediate tag with period τ. COLM0 is recommended for defense in depth scenario. COLM has linear mixing functions ρ and ρ^{-1} that calculated as Equation (7):

\[
\rho = y = x \oplus 3 \cdot st;
\rho^{-1} = x = y \oplus 3 \cdot st
\]

Where x, st \in \{0,1\}^{128} and y, st' \in \{0,1\}^{128}. Δ values are calculated using Fig. 4.

\[
\Delta_i = 3 \cdot 2^i \cdot L
\]

\[
\Delta_i = 3 \cdot 2^i \cdot L
\]

\[
\Delta_i = 7 \cdot 3^{i-1} \cdot L
\]

\[
\Delta_i = 7 \cdot 3^{i-1} \cdot L
\]
The tagged ciphertext is computed from the padded plaintext and IV. Decryption is the same as encryption. The verification will be successful if we have $C[m + 1] = C[m + 1]$. COLM is depicted in Fig. 5. The stages of COLM include generation of subkey $L$, IV, tagged ciphertext, decryption, and verification.

![Illustration of COLM authenticated cipher](image)

### III. PROTECTED ARCHITECTURES

A review of previous studies in Section II-C showed that so far, no protection scheme has been proposed for OCB and COLM. Additionally, most of CAESAR candidates are protected with TI or DOM masking scheme. Therefore, in this section, a hardware architecture based on TI and DOM masking scheme is proposed to protect them against the first order DPA attacks. As mentioned in Section II-D, OCB and COLM ciphers have AES as primitive, several functions and multiplication in $GF(2^{128})$. The finite field multiplications used in these ciphers are with constant 2 that are linear operations [31]. The AES cipher includes linear parts and a non-linear S-box part. To protect the linear parts of the cipher, these parts are repeated as many times as the number of the shares, but for non-linear parts (Section II-A), the complicated methods are needed. Therefore, in the next section, the protected AES based on TI and DOM masking schemes will be presented.

#### A. Protected AES

So far, several architectures have been proposed to protect AES against DPA attacks [8]–[14], [32]. These architectures are based on TI or DOM masking scheme. Protecting S-box as the non-linear part of AES is the most complicated and sensitive part. The algebraic degree of AES S-box is seven. Thus, its direct sharing needs at least 8 shares, which is not feasible. As AES S-box is equivalent to inversion in $GF(2^8)$, its hardware implementation usually is done using combinatorial logic based on Canright architecture [33]. In this case, the inversion in $GF(2^8)$ is represented as the operation in $GF(2^4)$ with three multiplications, a square-scaler and an inversion in $GF(2^4)$. In $GF(2^4)$ with a normal basis, each multiplication needs three $GF(2^2)$ multiplier, the square-scaler consist of wiring and XOR and the inversion require three $GF(2^2)$ multiplier.

1) TI PROTECTED S-BOX ARCHITECTURE

$GF(2^8)$ multiplier as non-linear part of S-box do not satisfy uniformity feature due to not being a permutation. Therefore, fresh random bits are needed during computation. The total number of the required fresh random bits leads to an increase in the number of I/O or the hardware area. Therefore, it would be better to distribute this requirement along with several clocks. The protected S-boxes need a larger area and fresh random bits for implementation. Additionally, the long data path leads to an increase in glitch. Therefore, to implement protected AES, we use an 8-bit protected S-box is implemented based on [10] and [24], that used TI approach with hybrid 2-/3 shares, as shown in Fig. 6. The protected S-box has five pipeline stages that separated by a dotted line, as shown in Fig. 6. This architecture requires 16 bit fresh random bits to re-share from 2 to 3 share and two fresh random bits per each $GF(2^2)$ multiplier. In total, 40 fresh random bits are required for each S-box per clock.

![First-order protected AES S-box using TI masking scheme](image)

2) DOM PROTECTED S-BOX ARCHITECTURE

The DOM-protected S-box is presented in [14] with two shares. Like TI, this scheme uses the tower field method to turn the operation in lower subfields $GF(2^2)$. The main difference of two schemes is in the implementation of $GF(2^8)$ multipliers. These multipliers are implemented with the generalization of the first-order secure AND described in Section II-A. Protected design, as shown in Fig. 7, has five stages of the pipeline to increase throughput. In the first stage, the linear mapping gets 8-bit inputs shares $A_x$ and $B_x$ and combines them in their domain. Two registers are added after linear transformation to avoid glitch propagation. Also, $GF(2^4)$ multipliers receive their inputs from the linear mapping. In the second and third stages, the glitch can occur due to the combination of square-scaler outputs and multiplication gates. The inputs of stage 4 are independent and so do not require any register. In the last stage, linear mapping produces the S-box outputs that are stored in a state or key registers or fed into the next S-box. For three $GF(2^2)$ multiplier, 12 bits fresh random are needed. Six bits fresh
random are required for a $GF(2^4)$ inverter. Thus, every S-box running requires 18 fresh random bits.

3) PROTECTED AES ARCHITECTURE
The protected AES with 8-bit architecture implemented using a TI/DOM protected S-box is shown in Fig. 8. Using the mask $m_k$ and $m_d$, the key and message were broken down into two shares of $k_1 = m_k$, $k_2 = m_k \oplus k$, $d_1 = m_d$ and $d_2 = m_d \oplus d$. Therefore, the shares are saved in two data register and two key registers. Additionally, two 32 bits mixcolumn have been implemented. In the 10th round of AES, the AES output is obtained when done signal becomes 1. Every block encryption of DOM-protected AES needs 246 clocks whereas TI-protected need 205 clocks.

FIGURE 7. First-order protected AES S-box using DOM masking scheme [14]

B. PROTECTED OCB ARCHITECTURE
OCB cipher has several stages and function include AES cipher, stretch, trunc, and multiplication in $GF(2^{128})$ field. The AD and plaintext/ciphertext are processed with AES, and its output enters the next AES after passing through stretch, mul, and trunc functions. Then the ciphertext/plaintext and tag are generated. As each protected AES module consume a high area of the circuit, one module of protected AES has been implemented and is used serially. Fig. 9 shows the first-order protected 8-bit hardware architecture for OCB using TI/DOM masking scheme. The only non-linear part of OCB is AES S-box that is protected with three shares in TI and two shares in DOM. Thus we use the same architecture for TI and DOM protected OCB with different protected-AES. Each linear part has been repeated twice. The linear function Mul and stretch functions are used to calculate $\Delta$ and $\delta$. Mul function only requires multiplication by constants 2 in $GF(2^{128})$. The register $\Delta_k$ is used to save these outputs. The register checksum is used to keep the sum of inputs and register $\text{Auth}_k$ is used to save AD processing results, and finally, the results obtained from both of them are used to calculate the tag.

C. PROTECTED COLM ARCHITECTURE
As mentioned in Section II-D2, COLM has three stages: generation of L, IV, and tagged ciphertext. AES processes the AD, and the plaintext/ciphertext and output enters the second AES after passing through the function $\rho$. Finally, ciphertext/plaintext and the tag are generated. COLM include AES, linear function $\rho$, $\Delta$ calculations and several linear multiplications in $GF(2^{128})$. Similar to OCB, the same architecture proposed for TI and DOM-protected COLM with two different protected AES and each linear unit has been repeated twice. Fig. 10 shows the proposed first-order protected 8-bit hardware architecture for TI/DOM-protected COLM.

Generating $\Delta$ requires multiplication by constants 2, 3, 7 and 49 in $GF(2^{128})$. Filed multiplication has the heavy computation, but multiplication by 2 (doubling) on this field can be reduced to a shift and a few XORs. Other multiplication can be calculated as $\text{mul3}(x) = \text{mul2}(x) \oplus x$; $\text{mul7}(x) = \text{mul3}(\text{mul2}(x)) \oplus x$; $\text{mul49}(x) = \text{mul7}(\text{mul7}(x))$. Two 128-bit registers $\Delta_l$ and $\Delta_h$ are used to store the output.
The outputs of function $\rho$ are stored in two 128-bit registers of $R_W$ and $R_Y$, respectively. Because the function $\rho$ and $\Delta$ should be computed simultaneously, $mul2(x)$ is implemented two times. The 128-bit registers $R_T$ are responsible for keeping the sum of inputs to calculate the tag.
IV. Practical Results

In this section, the practical result for power leakage assessment on unprotected and protected of selected authenticated ciphers is presented. Then the implemented ciphers are benchmarked, and costs of protection are compared. For these analyses, some equipment is needed that described as bellow.

---

**Figure 9.** The proposed 8-bit hardware architecture for TI/DOM protected OCB

**Figure 10.** The 8-bit hardware architecture for protected TI/DOM COLM. Adapted from [25]
A. EXPERIMENTAL SETUP

Practical experiments were done on SAKURA-G board [34] (Fig. 11). This board contains the main FPGA, a Xilinx Spartan 6 XC6SLX75 which used to implement the ciphers and a control FPGA, a Xilinx Spartan 6 XC6SLX9 which controls the communications between the main FPGA and a PC. In all experiments, the power measurement was made in target FPGA on $V_{dd}$ path. This measurement was made using digital oscilloscope Infinium Keysight DS090604A [35] with a sampling rate of 20 Gs/s and bandwidth of 6 GHz. The output of the AC amplifier embedded on the SAKURA-G is measured. In order to implement selected ciphers on FPGA, the proposed architecture of Fig. 9 and 10 were coded in VHDL language at RTL level and synthesized using Xilinx ISE V14.7 software. The correctness of the circuit functionality is checked with Mentor Graphic Modelsim 10.1c tools. The main FPGA has been clocked in different frequencies of 0.5 MHz to 10 MHz. The frequency change was done using frequency dividers inside control FPGA. Additionally, the random values and initial inputs sharing are generated by control FPGA and is sent to main FPGA. The outputs also are in a shared form with equal to the number of the input shares. For t-test, the selection of the constant or random inputs also is done in control FPGA.

B. POWER LEAKAGE ASSESSMENT

In this subsection, the amount of power leakage of the unprotected and protected version of OCB and COLM ciphers is done by first and second-order t-test.

1) POWER LEAKAGE ASSESSMENT ON THE UNPROTECTED OCB AND COLM

In order to assess the power leakage and run the t-test, the unprotected OCB and COLM are implemented on FPGA based on the architecture of Fig. 9 and 10 (only the black section on top of the figures with unprotected AES). In the total number of 18,000 traces were collected based on the method described in the Section II-B using a non-specific t-test, the t values were calculated based on Equation (5) and using ttest2 command in MATLAB software as shown in Fig. 12. The results for both ciphers show $|t| > 4.5$ that indicate a considerable leakage, which is expected for unprotected schemes.

![FIGURE 12. Result of t-test using 18,000 traces](image)

2) POWER LEAKAGE ASSESSMENT ON THE PROTECTED OCB AND COLM

This study focuses on the first-order DPA resistant selected authenticated ciphers, thus the first and second-order t-test is done. To validate the countermeasures taken to protect OCB and COLM ciphers, the TI-protected version with hybrid 2-/3 shares and DOM-protected with 2 shares were implemented on FPGA based on the proposed architecture of Fig. 9 and 10. The results of the implementation of non-specific first-order t-test on 18,000 collected traces in Fig. 13 show that TI and DOM-protected OCB and COLM ciphers passes the t-test, and can be resistant against the first-order DPA attacks. But as expected, the results of the second-order t-test in Fig. 14 show that the scheme is not resistant against the second-order DPA.
C. PROTECTION COST OF OCB AND COLM CIPHERS

To benchmark the unprotected and protected versions of OCB and COLM ciphers, our proposed architectures are described in VHDL code at RTL level, synthesized, and implemented on SAKURA-G board FPGA. The results of the hardware implementation of the proposed architectures for unprotected and protected in term of area (slice and LUT), maximum frequency, throughput, and efficiency are compared in Tab. 3, and Fig. 15.

The unprotected OCB has about, 11% lower area (LUT), 5% higher maximum frequency, 23% higher throughput and 31% higher efficiency than COLM. Also, comparison of protected versions shows that OCB has better performance in all of the criteria. DOM-protected OCB has the lowest area and TI-protected OCB has the highest throughput while TI-protected OCB has the highest efficiency. Therefore, if the area is important DOM-protected OCB is selected and if throughput or efficiency is important, TI-protected OCB is selected.

<table>
<thead>
<tr>
<th>Ciphers</th>
<th>Design</th>
<th>Area (LUT)</th>
<th>Area (Slice)</th>
<th>$F_{\text{max}}$ (MHz)</th>
<th>$T_p$ (Mbps)</th>
<th>Efficiency (Mbps/KLUT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>COLM OCB</td>
<td>Unprotected</td>
<td>2296</td>
<td>991</td>
<td>149.6</td>
<td>38.94</td>
<td>16.95</td>
</tr>
<tr>
<td>COLM OCB</td>
<td>TI-protected</td>
<td>2035</td>
<td>924</td>
<td>158.2</td>
<td>50.6</td>
<td>24.86</td>
</tr>
<tr>
<td>COLM OCB</td>
<td>DOM-protected</td>
<td>5176</td>
<td>2379</td>
<td>87.3</td>
<td>26.5</td>
<td>5.1</td>
</tr>
<tr>
<td>COLM [25] OCB</td>
<td>TI-protected</td>
<td>4435</td>
<td>2145</td>
<td>95.1</td>
<td>33.9</td>
<td>7.6</td>
</tr>
<tr>
<td>COLM [25] OCB</td>
<td>DOM-protected</td>
<td>4894</td>
<td>2038</td>
<td>80.2</td>
<td>20.8</td>
<td>4.25</td>
</tr>
<tr>
<td>COLM [25] OCB</td>
<td>DOM-protected</td>
<td>4130</td>
<td>1830</td>
<td>86.4</td>
<td>30.7</td>
<td>7.43</td>
</tr>
</tbody>
</table>
Comparison of results on Fig. 15 shows that TI-protected COLM has the highest increased area by a factor 2.25 and DOM-protected COLM has the lowest increase area by a factor 2.03. In term of throughput, TI-protected COLM has the lowest decrease by a factor 1.47, and DOM-protected COLM has highest decreased by a factor 1.87.

To the best of our knowledge, so far, no protected architecture has been proposed for OCB and COLM ciphers except [25] (previous research by authors which presented a DOM-protected COLM). Therefore, we can compare the increased area ratio and the decreased throughput ratio of this research with other protected AE ciphers as shown in Tab. 4. Although decreased throughput ratio of proposed protected ciphers is not lowest, the increased area ratio is better than the others (2.03 for DOM-protected OCB).

TABLE 4. Comparison Cost of AE protection with others (‘IR’ and ‘DR’ means increase ratio and decrease ratio, respectively).

<table>
<thead>
<tr>
<th>Ref.</th>
<th>Ciphers</th>
<th>Primitive</th>
<th>Protection method</th>
<th>Protected vs Unprotected</th>
</tr>
</thead>
<tbody>
<tr>
<td>[21]</td>
<td>ASCON</td>
<td>Sponge</td>
<td>TI</td>
<td>2.15 1.24</td>
</tr>
<tr>
<td></td>
<td>ACORN</td>
<td>-</td>
<td>TI</td>
<td>2.34 1.32</td>
</tr>
<tr>
<td>[22]</td>
<td>ACORN</td>
<td>Sponge</td>
<td>TI</td>
<td>3.1 1.1</td>
</tr>
<tr>
<td></td>
<td>CLOC</td>
<td>AES</td>
<td>TI</td>
<td>5 1.6</td>
</tr>
<tr>
<td>[23]</td>
<td>SILC</td>
<td>AES</td>
<td>TI</td>
<td>2.5 1.6</td>
</tr>
<tr>
<td></td>
<td>JAMBU</td>
<td>AES</td>
<td>TI</td>
<td>2.7 1.4</td>
</tr>
<tr>
<td></td>
<td>Ketje Jr</td>
<td>Sponge</td>
<td>TI</td>
<td>3.9 1.6</td>
</tr>
<tr>
<td>[24]</td>
<td>COLM</td>
<td>AES</td>
<td>DOM</td>
<td>2.13 1.87</td>
</tr>
<tr>
<td>[25]</td>
<td>OCB</td>
<td>AES</td>
<td>TI</td>
<td>2.18 1.49</td>
</tr>
<tr>
<td></td>
<td>COLM</td>
<td>AES</td>
<td>TI</td>
<td>2.25 1.47</td>
</tr>
<tr>
<td></td>
<td>OCB</td>
<td>AES</td>
<td>DOM</td>
<td>2.03 1.65</td>
</tr>
</tbody>
</table>

VI. CONCLUSION AND FUTURE RESEARCH

In this research, the resistance of two authenticated ciphers OCB and COLM, as two winners of CAESAR competition, were investigated against DPA attacks. Despite the existence of inherent masking in the algorithm, we demonstrate that these two ciphers are vulnerable to DPA attacks and do not pass the t-test. The proposed protected versions of these ciphers using TI masking scheme based on hybrid 2-/3 shares and DOM masking scheme with 2 shares are resistant against the first-order power analysis attacks, considering the results of the first and second-order t-test. The unprotected OCB cipher has about 11% lower area and 23% higher throughput compared to COLM cipher. In term of protection cost, DOM-protected OCB has the lowest area cost by a factor 2.03. Also, TI-protected OCB has the highest throughput by a factor 1.47. While TI-protected OCB has the highest efficiency. The increased area ratio for DOM-protected OCB is equal to 2.03 that is better than others. If the resistant against higher-order attacks is required, it is necessary to use a higher number of shares, with more costs.

The future research includes the investigation of the vulnerability of the four remained winners of CAESAR competition, as well as representation of the protected version against DPA attacks. Moreover, instead of t-test, the DPA attack can be used that leads to key recovery. Additionally, introducing the protected versions of CAESAR winners against higher-order attacks is proposed as future work.

REFERENCES


This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2942781, IEEE Access


MOHSEN JAHANBANI received the BS in electrical engineering in 2004, and MS degree in telecommunication in the field of cryptography from Imam Hossein University, Tehran, Iran in 2009. He is now a Ph.D candidate at the Electrical Engineering Department of Imam Hossein University. His research interest includes hardware implementation, lightweight implementation, side-channel attacks and countermeasures. He has published a few technical papers and participated in journals of information security and cryptography.

ZEINLABEDIN NOROZI received his BS and MS degrees in applied mathematics in 2004 and 2007 from University of Tehran, respectively. He has Ph.D. degree in applied mathematics - cryptography in 2012 from Kharazmi University. In 2007, he joined the Electrical Engineering Department as an assistant professor at Imam Hossein University, Tehran, Iran. His research interest includes symmetric cryptography, with an emphasis on block cipher and authenticated encryption, steganography algorithms and lightweight implementation and side-channel attacks. He has published several technical papers and participated in many scientific conferences in information security and cryptography. His contact email is: znoroz@ihu.ac.ir.
NASOUR BAGHERI received the M.S. and Ph.D. degrees in Electrical Engineering from Iran University of Science and Technology (IUST), Tehran, Iran, in 2002 and 2010 respectively. He is currently an associate professor at the electrical engineering department, Shahid Rajaee Teacher Training University, Tehran, Iran. He is the author of over 100 articles in information security and cryptology. His research interests include cryptology, more precisely, designing and analysis of symmetric schemes such as lightweight ciphers, e.g., block ciphers, hash functions and authenticated encryption schemes, cryptographic protocols for constrained environment, such as RFID tags and IoT edge devices and hardware security, e.g., the security of symmetric schemes against side-channel attacks such as fault injection and power analysis. A record of his publication is available at google scholar: https://scholar.google.com/citations?user=32llx44AAAAJ&hl=en