High-speed Side-channel-protected Encryption and Authentication in Hardware

Nele Mentens[1], Vojtěch Miškovský[2], Martin Novotný[3] and Jo Vliegen[†4]

[1] ES&S and imec-COSIC/ESAT, KU Leuven, Belgium, nele.mentens@kuleuven.be
[2] Faculty of Information Technology, CTU Prague, Czech Republic, miskovoj@fit.cvut.cz
[3] Faculty of Information Technology, CTU Prague, Czech Republic, novotnym@fit.cvut.cz
[4] ES&S and imec-COSIC/ESAT, KU Leuven, Belgium, jo.vliegen@kuleuven.be

Abstract. This paper describes two FPGA implementations for the encryption and authentication of data, based on the AES algorithm running in Galois/Counter mode (AES-GCM). Both architectures are protected against side-channel analysis attacks through the use of a threshold implementation (TI). The first architecture is fully unrolled and optimized for throughput. The second architecture uses a round-based structure, fits on a relatively small FPGA board, and is evaluated for side-channel attack resistance. We perform a Test Vector Leakage Assessment (TVLA), which shows no first-order leakage in the power consumption of the FPGA. To the best of our knowledge, our work is (1) the first to describe a throughput-optimized FPGA architecture of AES-GCM, protected against first-order side-channel information leakage, and (2) the first to evaluate the side-channel attack resistance of a TI-protected AES-GCM implementation.

Keywords: AES, Galois/Counter Mode (GCM), FPGA, Threshold Implementation (TI), Test Vector Leakage Assessment (TVLA)

1 Introduction

High-end applications like secure video streaming or videoconferencing require high-throughput and/or low-latency encryption and authentication. In order to achieve these strong requirements, the use of hardware implementation platforms is often unavoidable. Field-Programmable Gate Arrays (FPGAs) are increasingly used for high-speed applications, because FPGA designs have a shorter time to market than Application-Specific Integrated Circuit (ASIC) designs and can be achieved with a lower non-recurring engineering (NRE) cost. Further, FPGAs follow the latest technology nodes and contain dedicated application-specific components, which makes their performance less and less inferior to the performance of ASICs.

In terms of algorithms for encryption and authentication, a popular approach is to use a block cipher in an authenticated mode of operation. The most commonly used authenticated mode of operation is the Galois/Counter Mode (GCM). GCM was designed by McGrew and Viega and is standardized as SP800-38D by NIST [NIS07]. It uses established cryptographic primitives: the counter mode of operation (CTR) [LWR00] for data confidentiality and a Carter–Wegman message authentication code [NIS07] for data authentication.

Combining the Advanced Encryption Standard (AES) [DR02] with GCM leads to the well-understood and solid authenticated encryption scheme AES-GCM. The security

AES-GCM provides has been rigorously proven [MV04]. AES-GCM can encrypt and authenticate data in a single pass, making AES-GCM suitable for low-latency and high-throughput applications. In this paper, we present two FPGA implementations of AES-GCM. Besides AES-GCM, there are other authenticated encryption schemes; notably, the ongoing CAESAR competition recently assigned seven finalists out of tens of proposals for authenticated ciphers [cae].

When AES-GCM is implemented on an electronic device like an FPGA, a common way to extract secret information from the device is to use side-channel analysis. Side-channel attacks are a cost-effective method to extract secrets, such as passwords or cryptographic keys, from embedded implementations [Koc96]. A particularly effective strand of side-channel attacks is Differential Power Analysis (DPA) [KJJ99]. DPA is a technique that measures the instantaneous power consumption of an embedded device while the cryptographic implementation is being executed to extract the secret key. DPA does not require expensive equipment and the attack scales economically; hence, it is customary to design implementations with built-in side-channel attack countermeasures to mitigate this threat.

Side-channel attack countermeasures can be categorized into hiding and masking methods. The goal of hiding countermeasures is to break the correlation between the processed data and the power consumption, while the goal of masking countermeasures is to break the correlation between the processed data and the executed algorithm. In this paper, we concentrate on masking countermeasures, as introduced in [GP99, CJRR99]. Masking works by splitting secrets into several shares in a way that an attacker must recover the value of all shares to reconstruct the secret. In practice, masked implementations greatly increase the effort for an attacker to recover secrets, both in number of traces and computational effort. Usually, a masking scheme is designed for a specific algorithm and a specific implementation platform. There are masking proposals tailored towards software [HOM06, RP10, KHL11] or hardware [WOL02, CB08]. In hardware, a popular variant of Boolean masking is a threshold implementation (TI) [NRS08].

This paper starts from the AES-GCM architecture that is introduced in [VRM17], which consists of a threshold implementation with three shares, aiming at first-order DPA protection. Both the AES block cipher and the authentication components of AES-GCM are fully masked. The Galois Field multiplier used in the authentication tag generation is Boolean masked. Inside the AES round, the approach of Moradi et al. [MPL+11] is followed to construct the shared S-box. The architecture is fully unrolled and pipelined to achieve a maximal throughput.

Besides the implementation of the throughput-optimized architecture in [VRM17], we also introduce a round-based architecture using TI countermeasures that fits on the Spartan-6 FPGA of a Sakura-G board [GIS14], specifically designed for power analysis. We perform a Test Vector Leakage Assessment (TVLA) and conclude that the implementation shows no first-order side-channel leakage in the power consumption of the FPGA.

In summary, these are our contributions:

- We present the first high-throughput and TI-protected implementation of AES-GCM on an FPGA.
- We are the first to perform a TVLA evaluation on the full AES-GCM algorithm implemented on an FPGA.

The structure of the paper is as follows. Section 2 gives background information on the AES-GCM scheme, the threshold countermeasure, the side-channel vulnerabilities of AES-GCM, and the TVLA approach. In Sect. 3, related work is presented. Sections 4 and 5 describe the two AES-GCM architectures. In Sect. 6, the results are presented in terms of speed and occupied FPGA resources. Finally, conclusions and future work are described in Sect. 7.

2 Background

2.1 AES-GCM

Figure 1 gives a conceptual visualization of GCM. As explained in Sect. 1, GCM uses the CTR mode for data confidentiality. The counter is initialized with an initialization vector (IV) and is incremented providing $Y0$ to Yn. For every block of plaintext, the value of the counter is encrypted through the block cipher to generate a keystream. The XOR of the keystream with the plaintext generates the ciphertext.

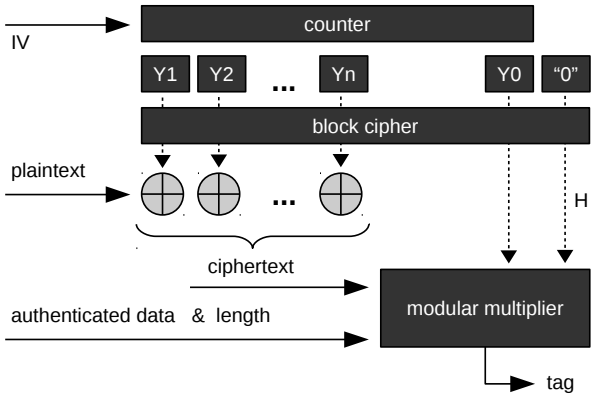

Figure 1: Conceptual visualisation of GCM

For data authentication, the main operation is a modular multiplication. The first operand of this multiplication is the data authentication key H. The value H is constant and is obtained by encrypting an all-zeros input. The second operand of the multiplication is the XOR of the previous multiplication product with an input value that first comes from the authenticated data; these are data that are not encrypted but only protected by the MAC. After each block of authenticated data is processed, each block of ciphertext is XORed with the previous multiplication product and multiplied with H. Finally, the result is XORed with the previous multiplication product and multiplied with a 128-bit value, representing the length of the authenticated data and the plaintext. The resulting product of all modular multiplications is XOR-ed with the encrypted value of $Y0$, which generates the MAC/TAG. To increase the reader's understanding, Figure 2 gives an overview of the consecutive inputs and outputs of the modular multiplier in GCM.

2.2 Threshold Implementation

Masking is delicate to implement, both in software and in hardware. One of the practical difficulties when implementing masking in hardware is that the glitching behavior in ASICs as well as FPGAs may weaken the masked hardware implementation, as shown by Mangard et al. [MPG05, MPO05]. Thus, a plain Boolean masked implementation in hardware must consider the glitching behavior to minimize leakage. Alternatively, Threshold Implementation (TI) is a masking implementation technique that provides security guarantees even when the underlying hardware glitches [NRS08].

2.3 Side-channel Vulnerabilities of AES-GCM

In AES-GCM, both the encryption and the tag generation datapaths must be protected against side-channel attacks. It is straightforward to see that an AES-GCM implementation invoking an unprotected AES as underlying block cipher would result in a design that

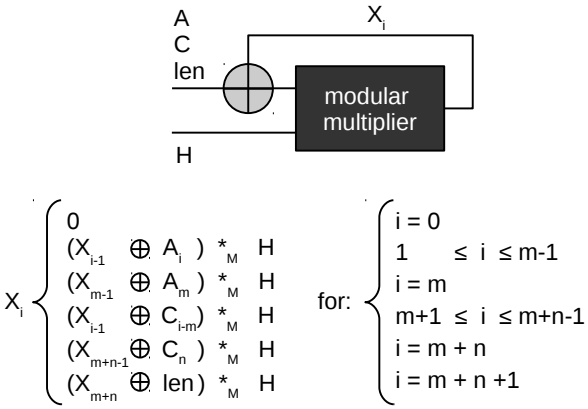

Figure 2: Detailed overview of the inputs and outputs of the modular multiplier in GCM, where $m-1$ is the number of fully filled authenticated data blocks and $n-1$ is the number of fully filled ciphertext blocks. The m^{th} and n^{th} blocks contain the padded data.

is vulnerable against DPA. This DPA attack against the block cipher in CTR mode can recover the secret key, resulting in a loss of confidentiality and authenticity. Note that it is reasonably easy to mount DPA attacks on the CTR mode even if the initial counter value is unknown [Jaf07].

DPA attacks against the authentication component of AES-GCM are also possible. Textbook DPA does not apply here if the multiplication in the 128-bit field is performed in a single cycle, because the attacker cannot apply a divide-and-conquer strategy. However, other slightly more sophisticated attacks still apply [BFG14, BCF+15]. The outcome of these attacks is the authentication key H. Note that once an adversary learns the authentication key H, he can forge new messages after seeing some ciphertexts, by exploiting the fact that the CTR mode is malleable. Hence, it is important to protect the authentication component of AES-GCM against side-channel attacks as well.

2.4 Test Vector Leakage Assessment (TVLA)

To verify the resistance against side-channel analysis, the side-channel leakage needs to be evaluated. Leakage assessment using Welch's t-test was proposed by Goodwill et al. in [GGJR+11] and extended by Schneider and Moradi in [SM15]. The proposed method uses side-channel traces collected during encryption/decryption performed on data from two different data sets. Usually, one set of fixed data and one set of random data is used (fixed vs. random test). The t-test provides us with the probability that samples of both data sets were drawn from the same population. For each sample point of traces, a so-called t-value is evaluated. If the t-values do not lie between -4.5 and 4.5 for each sample point, the device fails the test.

3 Related Work

In 2005, Yang et al. present a throughput-optimized ASIC implementation of AES-GCM in a 0.18 μm standard cell library, achieving a throughput of 34 Gbit/s, running at a frequency of 271 MHz [YMK05]. In 2006, Hodjat et al. concentrate on the optimization of the critical path of the AES cipher through pipelining nin order to maximize the throughput [HV06]. This results in architectures achieving 30 to 70 Gbit/s in a 0.18 μm standard cell library. In [BJM+14, PSJ+16], the authors go one step further by adding pipelining registers

in the AES S-boxes. In 2007, Lemsitzer et al. present a pipelined implementation of AES-GCM, optimized for FPGAs [LWFB07]. Their architecture supports key lengths of 128, 192 and 256 bits. It reaches a throughput of 15.3 Gbit/s on a Virtex-4 FPGA. In the same year, Zhou et al. reach a throughput of 20 Gbit/s by pipelining the AES round and balancing the critical path of the modular multiplier [ZMH07]. Improved architectures are presented by Zhou et al. in 2009, resulting in a throughput of 31 Gbit/s and 39 Gbit/s on Virtex-4 and Virtex-5 FPGAs, respectively [ZMH09]. The optimizations are applied to the multiplier architecture, resulting in a shorter critical path. Finally, we mention two FPGA implementations of AES-GCM in a real-world scenario. The first one is presented by Vliegen et al. [VKSM16] in 2016; it describes the integration of AES-GCM in an FPGA-based system for the real-time communication of medical video streams. The second one is reported by Martinasek et al. [MHS+18] in 2018. They achieve a throughput of 200 Gbit/s for an implementation of the complete IPsec protocol and present a practical realization based on a commercial 200 Gbit/s network card.

Whereas the aforementioned work concentrates on increasing the throughput, another line of research aims at side-channel security through threshold implementations. In 2011, Moradi et al. present a very compact threshold implementation of AES [MPL+11]. An improved architecture, resulting in a more compact, threshold-protected AES implementation, is presented by Bilgin et al. in [BGN+14]. All these designs are serialized and do not target high throughput. The work of Diehl et al. [DAF+18] implements and assesses TI-protected architectures of a number of candidates of the CAESAR competition; a TVLA assessment of the full AES-GCM algorithm with TI protection is not included.

In summary, related work covers high-throughput AES and AES-GCM implementations on the one hand as well as threshold-protected AES implementations on FPGAs and ASICs on the other hand. Our work is the first to implement and evaluate a combination of all these notions. The first part of our work concentrates on the implementation of the full AES-GCM algorithm with both high throughput and side-channel protection as a goal; the resulting architecture is implemented on a high-end Xilinx Virtex-7 device. The second part of our work reduces the occupied resources of the architecture such that it fits on the Spartan-6 FPGA of a Sakura-G board; TVLA is used to assess the side-channel leakage of this implementation.

4 The Proposed Throughput-optimized TI-protected Architecture

This section describes the first architecture presented in this paper, namely the throughput-optimized and TI-protected architecture. A top-level representation of the implementation is shown in Fig. 3. From left to right there is a counter, an AES threshold component, and a threshold-protected modular multiplier (MALU GMS). The components are glued together through registers, XOR gates and a multiplexer. This section discusses the three components and describes how the architecture works.

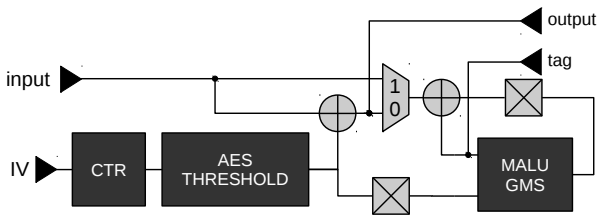

Figure 3: The proposed top-level architecture

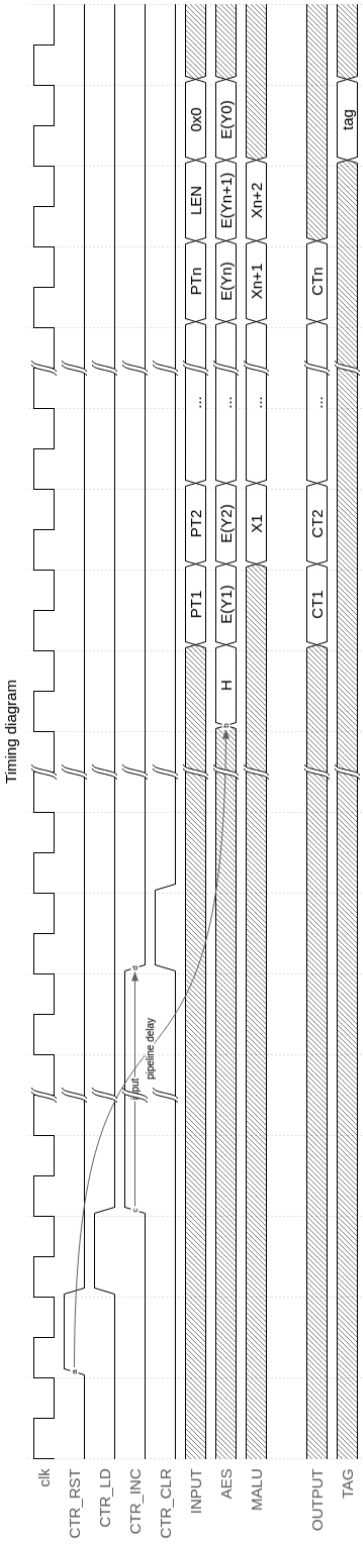

Figure 4: Sequence of operations in our AES-GCM implementation

Additionally, it is pointed out that for this implementation, padding is not implemented in hardware, but is expected to be done externally. This implies that the length field, required in the final steps of GCM, is to be provided through the input interface.

Figure 4 shows a timing diagram that is referred to in the description of the components and the top-level functionality. The notation we use for the processed values is explained in Sect. 2.1.

4.1 Components

4.1.1 Counter

The counter is initialized with 128 '0' bits for the calculation of H (CTR_RST in Fig. 4). Subsequently, the counter loads the 96-bit initialization vector (IV). For the sake of simplicity, the case where the length of the IV differs from 96 bits is ignored. Upon loading the IV, it is padded with $0x00000002$, which loads $Y1$ into the counter (CTR_LD in Fig. 4). Next, the counter is incremented every clock cycle for each additional plaintext block (CTR_INC in Fig. 4). Finally, the counter is loaded with the IV padded with $0x00000001$, which loads $Y0$ (CTR_CLR in Fig. 4). This comes down to a counter which has two load signals and a reset signal.

The value of $Y0$ is used only after a complete authenticated encryption is done. Because the first value that is used is $Y1$ (which is the incremented version of $Y0$), loading and incrementing the counter takes two clock cycles. This additional clock cycle is not necessary when the second load signal is applied to load $Y1$.

4.1.2 AES Threshold

In order to obtain a threshold implementation with three shares, the linear operations in the AES round, namely AddRoundKey, ShiftRows and MixColumns are applied to each of the three shares in parallel. The non-linear operation, i.e. SubBytes, consists of sixteen 8-bit S-box lookups in parallel. A shared implementation of the S-box is achieved by following the approach of Moradi et al. [MPL+11].

Depending on the key width, AES runs a number of cycles. This implementation focuses on a 128-bit key, which results in 10 rounds. These 10 rounds are fully unrolled. The straightforward way of adding pipelining to the architecture would be to insert a register in between each round. However, the threshold implementation of the S-box requires remasking pipelining registers anyway, resulting in a five-stage pipeline, as explained in [MPL+11]. Therefore, we do not introduce additional pipelining registers in between the rounds. This leads to a pipeline with 50 stages for the AES Threshold component. Further, every round needs a round key, which is derived from the original 128-bit key. The key schedule component is therefore also unrolled 10 times, with registers at the same positions inside the S-boxes. Once the pipeline is filled, a new 128-bit encryption is calculated every clock cycle. Because the key schedule is also pipelined, it applies the correct round keys to all the data in the pipeline.

4.1.3 MALU GMS

MALU GMS is the threshold-protected Galois Field multiplier, where MALU stands for Modular Arithmetic Logic Unit, as introduced by Sakiyama et al. in [SBM+06], and GMS refers to the Generalized Masking Scheme proposed by Reparaz et al. in [RBG+15]. Whereas the architecture described in [SBM+06] can be configured to different datapath depths, our throughput-optimized AES-GCM version uses the full depth in order to optimize for speed. Figure 5 shows the single MALU cell (right) and the fully unrolled and fully combinational 128-bit wide datapath of the MALU (top left).

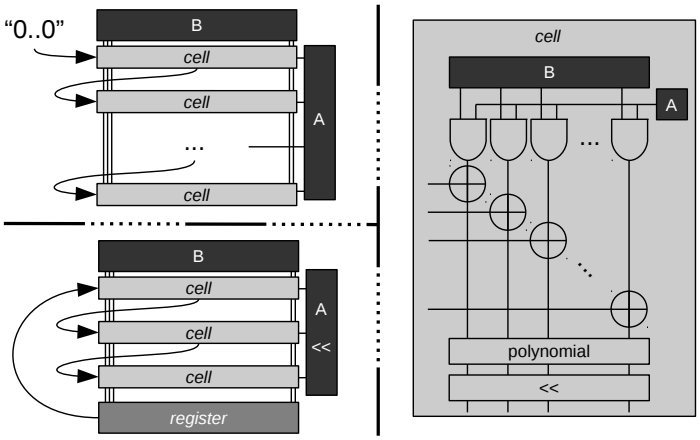

Figure 5: MALU architecture presented in [SBM+06], with one MALU cell (right), a fully unrolled 128-bit wide MALU (top left), and a 128-bit wide MALU with a datapath depth of 3 (bottom left).

The MALU GMS multiplier uses three shares for each input and three shares for the output. If the inputs x and y to the multiplier are shared into x_1, x_2, x_3 and y_1, y_2, y_3, respectively, the shared multiplier first computes a series of cross-products $x_i \cdot y_j$ and then combines them to compress the number of output shares, as defined in [RBG+15]. This leads to the MALU GMS multiplier architecture shown in Fig. 6, where the three shares of the first input (x), coming from the XOR gate in Fig. 3, are denoted by 0', 1' and 2'. The three shares of the second input (y), coming from the register in Fig. 3, are denoted by 0, 1 and 2.

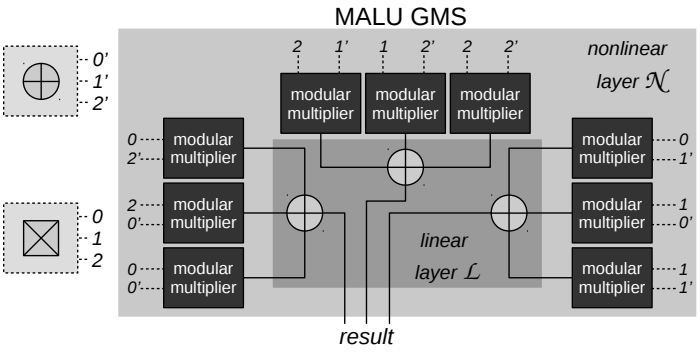

Figure 6: Architecture of the MALU GMS introduced in [RBG+15].

The fully parallel, pipelined AES architecture outputs 128 bits every clock cycle. Therefore, the MALU GMS needs to compute the multiplication result in one cycle as well, in order to be capable of processing a new 128-bit value in each clock cycle.

We note that the multiplications that involve the authenticated data (as shown in Fig. 1) do not need three shares for both inputs, since the authenticated data are not considered to be secret. Therefore, three unprotected Galois Field multipliers could be used in parallel to process the three shares of the other input. Nevertheless, since we re-use the multiplier for calculations in which both operands are secret, both operands are represented in three shares, where the sharing of the authenticated data consists of adding two all-zero masks.

4.2 Top-level Functionality

The underlying CTR mode for the encryption in GCM uses the block cipher as a stream cipher, which generates a block of key bits rather than a single symbol. Because we use a pipelined implementation of the block cipher, the plaintext needs to be fed after the pipeline is filled, i.e. when the first block of key bits is ready. Moreover, one additional clock cycle has to be waited for before applying the plaintext, because of the calculation of H.

The authentication code protects every block of authenticated data and every block of ciphertext. Because the CTR mode behaves as a stream cipher, the ciphertext is generated by XORing the plaintext with the keystream. When decrypting, the ciphertext is again XORed with the keystream, resulting in the plaintext.

When using the implementation for encryption, the multiplexer in Fig. 3 is set to '1' for every block of authenticated data and for the length fields. For every block of plaintext, the multiplexer is set to '1'. When using the implementation for decryption, the multiplexer in Fig. 3 is set to '0' for every block. Hence, for a dedicated decryption implementation, the multiplexer can be removed.

The latency for the computation of the first block of ciphertext is $50 + 1 + m$ clock cycles, where 50 is the pipeline depth of the AES Threshold block, one clock cycle is needed for the calculation of H, and m is the number of blocks of authenticated data. For the computation of the tag, we need to take into account an additional latency of $n + 1 + 1$ clock cycles, where n is the number of plaintext blocks, one clock cycle is needed for the MAC update with the length field, and one clock cycle is needed for the encryption of $Y0$.

5 The Proposed Iterative TI-protected Architecture

In order to be able to evaluate the complete TI-protected AES-GCM architecture on the Sakura-G side-channel evaluation board, we reduce the occupied FPGA resources. The following changes are made to the architecture described in Sect. 4:

- The AES Threshold block in Fig. 3 is changed from a fully unrolled architecture with a pipeline depth of 50 stages to an iterative round-based architecture. Five cycles are needed for the computation of one round, since the pipeline depth of the S-box is five. The latency of one AES computation is 50 clock cycles (ten rounds at five cycles per round). Because of the feedback loop in the architecture, a new plaintext can only be applied every 50 clock cycles.
- Since the AES Threshold block only generates a new output every 50 cycles, the MALU GMS does not need to complete the multiplication in one cycle; it can take up to 50 cycles for one multiplication. Therefore, we change the MALU from a fully unrolled architecture (as shown in Fig. 5 on the top left) to an architecture with a depth of three (as shown in Fig. 5 on the bottom left). Now it takes $\lceil 128/3 \rceil = 43$ cycles for one modular multiplication to complete.

6 Results

6.1 Implementation Results

The presented architecture is synthesized and implemented using Xilinx' Vivado Design Suite (v2014.4) on a Xilinx Virtex-7 XC7VX485T FPGA. The results of the relevant related work are presented together with our results in Table 1.

Table 1 shows that the throughput of our throughput-optimized threshold-protected architecture is in the same order of magnitude as the throughput of previously published

Table 1: Comparison of AES-GCM implementation results on FPGA

	FPGA (Xilinx)	Slices	Slices [%]	BRAM	f_{max} [MHz]	Throughput [Gbit/s]	Side-channel protection
[LWFB07][1]	Virtex-4 FX100	27'800	66	0	120	15.3	NA
[ZMH07][2]	Virtex-4 LX40	16'378	88	0	161	20.608	NA
[ZMH09][3]	Virtex-5 LX85	4'628	36	0	324	41.47	NA
this[4]	Virtex-7 X485T	38'241	50	0	119	15.24	1^{st}-order
this[5]	Virtex-7 X485T	3'433	4.5	0	278	0.188	1^{st}-order

[1] results for the 128-bit version with the highest throughput
[2] results for the 128-bit speed-efficient encryption
[3] results for the 128-bit highest-throughput encryption
[4] results for our 128-bit throughput-optimized TI-protected encryption
[5] results for our 128-bit iterative TI-protected encryption

unprotected architectures. This is thanks to the comparable clock frequency that can be achieved for our protected architecture on a high-end Virtex-7 device. In terms of configurable resources, the insertion of TI protection introduces a large overhead. Note that the slices inside a Virtex-4 FPGA contain 4-to-1-bit Lookup Tables (LUTs), while from Virtex-5 onwards, the slices consist of 6-to-2-bit LUTs. Table 1 shows that the absolute number of slices between the most recent previous work and this work differs in an order of magnitude. Nevertheless, the position of the Virtex-7 X485T device in the Virtex-7 family is similar to the position of the Virtex-5 LX85 device in the Virtex-5 family. With this in mind, it is clear that with the increasing capabilities of FPGAs, developing side-channel robust implementations is becoming relatively affordable.

The throughput of the proposed architecture is 15.24 Gbit/s. If we assume the most strict requirements on the incoming random numbers, all random masks need to be refreshed every clock cycle. For a single AES round (including key expansion) this is 960 bits per clock cycle for the remasking inside the S-boxes. Further, the plaintext and the key need to be shared, which results in 512 random bits per clock cycle. In total, this comes down to 1472 random bits per clock cycle at 8.4 ns per clock cycle. As a result, if we want to achieve a throughput of 15.24 Gbit/s with the most strict requirement of fresh masks in every clock cycle for first-order DPA protection, the pseudorandom number generator needs to provide random masks at a throughput of 175.24 Gbit/s. This is a throughput that is impossible or, assuming many pseudorandom number generators in parallel, highly impractical to achieve on a Virtex-7 FPGA. Therefore, we conclude that we either need to relax the requirements on the freshness of the masks or reduce the throughput of the AES-GCM architecture, with the additional benefit of the possibility to make the architecture smaller in terms of FPGA resources and to adapt the inputs and outputs to a more commonly used 32-bit or 64-bit interface.

The implementation results of our iterative TI-protected implementation show a decrease in the number of slices of a factor 10 compared to our throughput-optimized architecture. The throughput goes down drastically due to the feedback loop in the AES Threshold block and in the MALU GMS. The iterative architecture is suitable to be implemented on the Spartan-6 FPGA of a Sakura-G board.

6.2 Evaluation Results

The side-channel analysis resistance of the implementation is evaluated on a Sakura-G board. We apply fresh randomness in each cycle for the remasking registers inside the AES Threshold block and for dividing the input values into shares. In our lab measurement setting, the random bits are generated through an on-chip Linear Feedback Shift Register (LFSR). The measurement is done at a clock frequency of 15 MHz. The power traces are collected by a PicoScope 6404D using a sampling period 6.4 ns. A total number of 1,000,000 traces are collected for the evaluation.

The side-channel leakage is evaluated using a first order, non-specific, fixed vs. random Welch's t-test described in [SM15]. For the fixed data set, all input data (IV, key, authentication data, plaintext) are fixed. For the random data set, the authentication data and the plaintext are random, and the IV and the key are fixed. All shared inputs are reshared before each encryption for both data sets.

The results of the t-test evaluation are shown in Fig. 7. As the t-values for all samples lie within an interval $(-4.5, 4.5)$, we can conclude that there is no first-order leakage discovered by the evaluation.

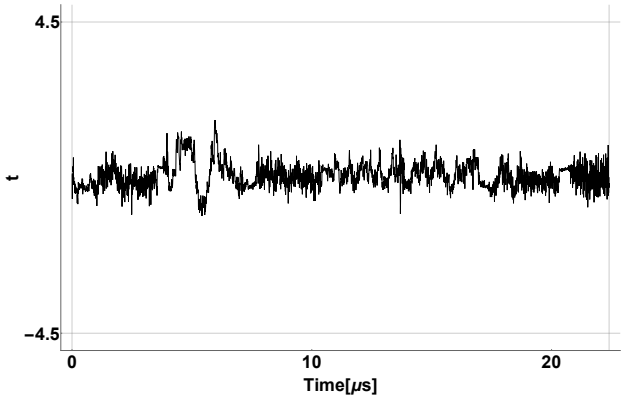

Figure 7: T-values for the whole duration of the AES-GCM computation.

7 Conclusion

In this paper, we investigate the maximum throughput we can achieve for a threshold-protected AES-GCM implementation on a Virtex-7 FPGA. We implement a fully parallel, fully unrolled and pipelined architecture of AES in combination with a Galois Field multiplier, using three input shares and three output shares. The architecture results in a throughput of 15.24 Gbit/s at a clock frequency of 119 MHz. In order to evaluate the side-channel attack resistance of the implementation, we reduce the occupied FPGA resources of the architecture by applying an iterative round-based structure. This way, the architecture fits the Spartan-6 FPGA on a Sakura-G board. The Test Vector Leakage Assessment of the implementation shows no first-order power leakage. Our work is the first to introduce a throughput-optimized threshold-protected FPGA implementation of AES-GCM and the first to evaluate the full AES-GCM algorithm with threshold protection on an FPGA.

Future work consists of optimizing both architectures in terms of FPGA resources by building on recent research results showing that it is enough to have a two-share implementation. Further, it is interesting to investigate to which extent the applied random input bits can be reused, as opposed to the conservative approach of providing the implementation with fresh randomness every clock cycle.

References

[BCF+15] Sonia Belaïd, Jean-Sébastien Coron, Pierre-Alain Fouque, Benoît Gérard, Jean-Gabriel Kammerer, and Emmanuel Prouff. Improved side-channel analysis of finite-field multiplication. In Tim Güneysu and Helena Handschuh, editors, *Cryptographic Hardware and Embedded Systems - CHES 2015 - 17th*

International Workshop, volume 9293 of *Lecture Notes in Computer Science*, pages 395–415. Springer, 2015.

[BFG14] Sonia Belaïd, Pierre-Alain Fouque, and Benoît Gérard. Side-channel analysis of multiplications in GF(2128) - application to AES-GCM. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security*, volume 8874 of *Lecture Notes in Computer Science*, pages 306–325. Springer, 2014.

[BGN+14] Begül Bilgin, Benedikt Gierlichs, Svetla Nikova, Ventzislav Nikov, and Vincent Rijmen. A more efficient AES threshold implementation. In David Pointcheval and Damien Vergnaud, editors, *Progress in Cryptology – AFRICACRYPT 2014: 7th International Conference on Cryptology in Africa*, pages 267–284. Springer, 2014.

[BJM+14] Lejla Batina, Domagoj Jakobovic, Nele Mentens, Stjepan Picek, Antonio De La Piedra, and Dominik Sisejkovic. S-box pipelining using genetic algorithms for high-throughput aes implementations: How fast can we go? In *International Conference in Cryptology in India*, pages 322–337. Springer, 2014.

[cae] CAESAR: Competition for authenticated encryption: Security, applicability, and robustness.

[CB08] D. Canright and Lejla Batina. A very compact "perfectly masked" S-box for AES. In Steven M. Bellovin, Rosario Gennaro, Angelos D. Keromytis, and Moti Yung, editors, *Applied Cryptography and Network Security, 6th International Conference, ACNS*, volume 5037 of *Lecture Notes in Computer Science*, pages 446–459, 2008.

[CJRR99] Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards sound approaches to counteract power-analysis attacks. In Michael J. Wiener, editor, *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference*, volume 1666 of *Lecture Notes in Computer Science*, pages 398–412. Springer, 1999.

[DAF+18] William Diehl, Abubakr Abdulgadir, Farnoud Farahmand, Jens-Peter Kaps, and Kris Gaj. Comparison of cost of protection against differential power analysis of selected authenticated ciphers. In *International Symposium on Hardware Oriented Security and Trust (HOST)*, pages 147–152. IEEE, 2018.

[DR02] Joan Daemen and Vincent Rijmen. *The Design of Rijndael*. Springer-Verlag, 2002.

[GGJR+11] Benjamin Jun Gilbert Goodwill, Josh Jaffe, Pankaj Rohatgi, et al. A testing methodology for side-channel resistance validation. In *NIST non-invasive attack testing workshop*, volume 7, pages 115–136, 2011.

[GIS14] Hendra Guntur, Jun Ishii, and Akashi Satoh. Side-channel attack user reference architecture board SAKURA-G. In *Global Conference on Consumer Electronics (GCCE)*, pages 271–274. IEEE, 2014.

[GP99] Louis Goubin and Jacques Patarin. DES and differential power analysis (the "duplication" method). In Çetin Kaya Koç and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems, First International Workshop, CHES'99*, volume 1717 of *Lecture Notes in Computer Science*, pages 158–172. Springer, 1999.

[HOM06] Christoph Herbst, Elisabeth Oswald, and Stefan Mangard. An AES smart card implementation resistant to power analysis attacks. In Jianying Zhou, Moti Yung, and Feng Bao, editors, *Applied Cryptography and Network Security, 4th International Conference, ACNS*, volume 3989 of *Lecture Notes in Computer Science*, pages 239–252, 2006.

[HV06] Alireza Hodjat and Ingrid Verbauwhede. Area-throughput trade-offs for fully pipelined 30 to 70 gbits/s AES processors. *IEEE Transactions on Computers*, 55(4):366–372, 2006.

[Jaf07] Joshua Jaffe. A first-order DPA attack against AES in counter mode with unknown initial counter. In Pascal Paillier and Ingrid Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop*, volume 4727 of *Lecture Notes in Computer Science*, pages 1–13. Springer, 2007.

[KHL11] HeeSeok Kim, Seokhie Hong, and Jongin Lim. A fast and provably secure higher-order masking of AES S-box. In Bart Preneel and Tsuyoshi Takagi, editors, *Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop*, volume 6917 of *Lecture Notes in Computer Science*, pages 95–107. Springer, 2011.

[KJJ99] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Michael J. Wiener, editor, *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397. Springer, 1999.

[Koc96] Paul C. Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In *Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference*, pages 104–113, 1996.

[LWFB07] Stefan Lemsitzer, Johannes Wolkerstorfer, Norbert Felber, and Matthias Braendli. Multi-gigabit GCM-AES Architecture Optimized for FPGAs. In Pascal Paillier and Ingrid Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems - CHES 2007: 9th International Workshop*, pages 227–238. Springer, 2007.

[LWR00] Helger Lipmaa, David Wagner, and Phillip Rogaway. Comments to NIST concerning AES modes of operation: CTR-mode encryption, 2000.

[MHS+18] Zdenek Martinasek, Jan Hajny, David Smekal, Lukas Malina, Denis Matousek, Michal Kekely, and Nele Mentens. 200 Gbps hardware accelerated encryption system for FPGA network cards. In *Workshop on Attacks and Solutions in Hardware Security (ASHES)*, pages 11–17. ACM, 2018.

[MPG05] Stefan Mangard, Thomas Popp, and Berndt M. Gammel. Side-channel leakage of masked CMOS gates. In Alfred Menezes, editor, *Topics in Cryptology - CT-RSA 2005, The Cryptographers' Track at the RSA Conference 2005*, volume 3376 of *Lecture Notes in Computer Science*, pages 351–365. Springer, 2005.

[MPL+11] Amir Moradi, Axel Poschmann, San Ling, Christof Paar, and Huaxiong Wang. Pushing the limits: A very compact and a threshold implementation of AES. In Kenneth G. Paterson, editor, *Advances in Cryptology – EUROCRYPT 2011: 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 69–88. Springer, 2011.

[MPO05] Stefan Mangard, Norbert Pramstaller, and Elisabeth Oswald. Successfully attacking masked AES hardware implementations. In Josyula R. Rao and Berk Sunar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2005, 7th International Workshop*, volume 3659 of *Lecture Notes in Computer Science*, pages 157–171. Springer, 2005.

[MV04] David A. McGrew and John Viega. The security and performance of the galois/counter mode (GCM) of operation. In Anne Canteaut and Kapalee Viswanathan, editors, *Progress in Cryptology - INDOCRYPT, 5th International Conference on Cryptology in India*, volume 3348 of *Lecture Notes in Computer Science*, pages 343–355. Springer, 2004.

[NIS07] NIST. Recommendation for Block Cipher Modes of Operation: Galois Counter Mode (GCM) and GMAC (SP800-38D), 2007.

[NRS08] Svetla Nikova, Vincent Rijmen, and Martin Schläffer. Secure hardware implementation of non-linear functions in the presence of glitches. In Pil Joong Lee and Jung Hee Cheon, editors, *Information Security and Cryptology - ICISC, 11th International Conference*, volume 5461 of *Lecture Notes in Computer Science*, pages 218–234. Springer, 2008.

[PSJ+16] Stjepan Picek, Dominik Sisejkovic, Domagoj Jakobovic, Lejla Batina, Bohan Yang, Danilo Sijacic, and Nele Mentens. Extreme pipelining towards the best area-performance trade-off in hardware. In *International Conference on Cryptology in Africa*, pages 147–166. Springer, 2016.

[RBG+15] Oscar Reparaz, Begul Bilgin, Benedikt Gierlichs, Svetla Nikova, and Ingrid Verbauwhede. Consolidating Masking Schemes. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *Advances in Cryptology - CRYPTO 2015*, volume 9215 of *Lecture Notes in Computer Science*, pages 764–783. Springer, 2015.

[RP10] Matthieu Rivain and Emmanuel Prouff. Provably secure higher-order masking of AES. In Stefan Mangard and François-Xavier Standaert, editors, *Cryptographic Hardware and Embedded Systems, CHES 2010, 12th International Workshop*, volume 6225 of *Lecture Notes in Computer Science*, pages 413–427. Springer, 2010.

[SBM+06] Kazuo Sakiyama, Lejla Batina, Nele Mentens, Bart Preneel, and Ingrid Verbauwhede. Small-footprint ALU for public-key processors for pervasive security. In *Workshop on RFID Security 2006*, Lecture Notes in Computer Science, page 12. Springer, 2006.

[SM15] Tobias Schneider and Amir Moradi. Leakage assessment methodology. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 495–513. Springer, 2015.

[VKSM16] Jo Vliegen, Bob Koninckx, Dave Singelée, and Nele Mentens. Real-time encryption and authentication of medical video streams on FPGA. TRUDEVICE'16 conference report, 2016.

[VRM17] Jo Vliegen, Oscar Reparaz, and Nele Mentens. Maximizing the throughput of threshold-protected AES-GCM implementations on FPGA. In *Verification and Security Workshop (IVSW)*, pages 140–145. IEEE, 2017.

[WOL02] Johannes Wolkerstorfer, Elisabeth Oswald, and Mario Lamberger. An ASIC implementation of the AES sboxes. In Bart Preneel, editor, *Topics in Cryptology - CT-RSA 2002, The Cryptographer's Track at the RSA Conference*, volume 2271 of *Lecture Notes in Computer Science*, pages 67–78. Springer, 2002.

[YMK05] Bo Yang, Sambit Mishra, and Ramesh Karri. A high speed architecture for Galois/Counter Mode of operation (GCM), 2005.

[ZMH07] G. Zhou, H. Michalik, and L. Hinsenkamp. Efficient and high-throughput implementations of AES-GCM on FPGAs. In *2007 International Conference on Field-Programmable Technology*, pages 185–192, 2007.

[ZMH09] G. Zhou, H. Michalik, and L. Hinsenkamp. Improving throughput of AES-GCM with pipelined Karatsuba multipliers on FPGAs. In *Reconfigurable Computing: Architectures, Tools and Applications: 5th International Workshop, ARC*, pages 193–203, 2009.