Authenticated Encryption for Low-Power Reconfigurable Wireless Devices

Samant Khajuria and Birger Andersen

Aalborg University, Denmark, and Copenhagen University College of Engineering, Lautrupvang 15, 2750 Ballerup, Denmark; e-mail: skh@es.aau.dk, bia@ihk.dk

Abstract

With the rapid growth of new wireless communication standards, a solution that is capable of providing a seamless shift between existing wireless protocols and high flexibility as well as capability is crucial. Technology based on reconfigurable devices offers this flexibility. In order to avail this enabling technology, these radios have to propose cryptographic services such as confidentiality, integrity and authentication. Therefore, integration of security services to these low-power devices is very challenging and crucial as they have limited resources and computational capabilities.

In this paper, we present a crypto solution for reconfigurable devices. The solution is a single pass Authenticated Encryption (AE) scheme that is designed for protecting both message confidentiality and its authenticity. This makes AE very attractive for low-cost low-power hardware implementation. For test and performance evaluation the design has been implemented in Xilinx Spartan-3 sxc3s700an FPGA. Additionally, this paper analyzes different hardware architectures and explores area/delay tradeoffs in the implementation.

Keywords: authenticated encryption, confidentiality, message authentication, FPGA, wireless devices.

Journal of Cyber Security and Mobility, Vol. 1, 189–203.

1 Introduction

Over the past decade, wireless devices have become an indispensable part of our life. With time like every other technological device, the features and capabilities of the wireless devices are also evolved. Nowadays, devices like mobile phones are able to do lot more in addition to their traditional roles of voice communication. This has motivated new application domains for wireless networks. For example, wireless sensor networks (WSNs) are used in various applications, including environmental monitoring, military systems, health care, etc. Vehicular ad hoc networks (VANETs) promise road safety, while disruption-tolerant networks (DTNs) bring low-cost best-effort connectivity to challenged environments with little or no infrastructure [1]. Furthermore, the concept of Internet of Things (IoT) has picked up surge of interest with enormous applications in home and industry. Due to the advancement in the field, these networks offer a world of truly ubiquitous computing. With these additional abilities of the radios that are applicable across a wide range of areas within the wireless infrastructure, these radios have to implement cryptographic services such as confidentiality, integrity and authentication.

Typically, devices are equipped with an antenna from where they receive the data and-then-they process and transmit. Since the devices are compact and wireless, they are highly energy constraint. Data processing and wireless communication count for the greatest part of the energy consumed by a device. Especially in case of sensors, the need to operate for longer period of time demands for better and careful management of power resources. On top of this security is very challenging and crucial as devices have limited resources and computational capabilities. In order to provide data confidentiality and other cryptographic services, there is a need for lightweight schemes that can promise similar security as compared to traditional cryptographic schemes.

Future visions of wireless devices are foreseen as the devices connecting to a wide range of different networks or devices. This can be achieved by changing the characteristics of the devices by making software changes. By doing this the devices can adapt to the user preferences and the operating environment and support multiple standards without requiring separate radios for each standard. The possibility of dynamically adapting according to the environment is through the re-configuration of device's components. More specifically, the re-configurability is the ability of adjusting operational parameters for the transmission on-the-fly without any modifications

on the hardware components. Unlike implementing these functional blocks on inflexible Application Specific Integrated Circuits (ASICs) in the past, the technologies such as Field programmable Gate Arrays (FPGAs) are used to build radio functional blocks. FPGAs have reconfigurable capability and deliver flexibility of programmable architectures with power efficiency and performance. The reprogrammable nature of FPGAs makes them ideal for wireless devices, so any upgrades or changes in the operational parameters can be easily uploaded to the device without any hardware reconfigurations. FPGAs also allow the feature of partially reconfiguring the devices, the model is known as shared resource model. As compared to dedicated resource model, shared resources are capable of supporting ex., multiple waveforms across a single set of processing resources; this allows for much more efficient usage of the resources. Partial reconfiguration allows the replacement of one or multiple functional blocks with a different implementation while other portions are either being used by other applications or going unused. Without partial reconfiguration, it would be necessary to reconfigure entire FPGA. Using partially reconfigurable platform FGPAs for wireless devices will substantially decrease the component count of the devices and reduce power consumption while still providing the necessary functionality.

In this paper, we present a crypto solution for reconfigurable wireless devices. Section 2 summarizes the security issues and two main security objectives for wireless devices. Section 3 provides a brief overview of single pass authenticated encryption scheme. Section 4 details the architecture and overall design of the implementation, while the results are presented in Section 5. Finally conclusions are drawn in Section 6.

2 Security Objectives

In order to communicate between two or more devices or to enjoy the flexibility of reconfigurable radios to upgrade or adapt to user preferences many security countermeasures needs to be taken into account. Reconfiguring the radios has many benefits; however the ability to reconfigure radio functionalities with software may lead to many security problems such as unauthorized use of application and network services, unauthorized modification of software and manipulation of devices. For example, malicious software can be uploaded into the device that changes its radio frequency so that the device will no longer function within the regulated constraints. This could lead to the Denial of Service (DoS) attacks. Additionally, transmission of unencrypted

data over insecure channel could compromise the confidentiality and integrity of the data.

The above mentioned security issues often concerns with two main security objectives: confidentiality and authenticity of the data. The objective of confidentiality is to keep the contents of the information secure and no one but the sender and authorized receivers are able to read the data. Authentication of message data verifies the origin and improper or unauthorized modification of data. In the past, confidentiality of the data was the main issue considered. This was mainly because no other security objectives such as authentication or integrity prevented to have access to the information. Only message encryption can protect data from eavesdroppers. However encryption of messages provides some sort of authentication but as compared to present authentication techniques it is weak and cannot be relied upon. In addition to confidentiality, authentication services have been implemented but as add on feature to provide extra information security. Encryption algorithms are used to ensure confidentiality while Message Authentication Codes (MAC) can be used to provide authentication. In past few years, techniques have been invented which can combine encryption and authentication into a single algorithm [2–4]. Combining these two security features and performing single pass operation we expect this will provide the following advantages for hardware implementation:

- The rapid growth of portable low-cost devices with limited area has opened a vast scope for compact circuit design opportunities. Implementation of a single algorithm instead of two separate algorithms definitely has less area requirements. Reduction in area requirements on chip is directly proportional to the reduction in cost.
- Small and compact designs tend to consume less power as compared to bulky designs. This is an attractive feature for low-power devices like Cellular phone, PDAs, smartcards and especially wireless sensor devices.
- Even though separate keys are used for encryption and authentication for better security of the system, both the keys are usually derived from the same master key. This will have a slight advantage with regards to the key storage issues over separate algorithms.
- Most of the new designs target performance goals like throughput and throughput-area trade-off. In many cases, combined schemes are based on block ciphers, and designers have tried to be efficient with the number of block cipher calls required for getting both confidentiality and

authentication from the algorithm. Based on the mode of the operations some of these combined schemes can run in parallel and achieve much higher speed than older techniques.

3 Authenticated Encryption

The cryptographic schemes that provide both confidentiality and authentication are called authenticated encryption schemes. The scheme is designed in such a way that the sender produces the ciphertext as well as an authentication tag which is verified by the receiver.

The authenticated encryption scheme consists of three algorithms: a key generation algorithm, an encryption algorithm and a decryption algorithm. The encryption algorithm takes a key, a plaintext and an initialization vector and it returns a ciphertext. Given the ciphertext and the secret key, the decryption algorithm returns plaintext when the ciphertext is authentic and invalid when the ciphertext is not authentic. The scheme is secure if it is both un-forgeable and secure encryption scheme [5]. When an attacker is not able to successfully produce a ciphertext C, a nonce N, and a tag σ (three parameters which maintain the integrity of the message) even if the attacker convinces the receiver to will believe that the sender was the originator, then the scheme is *un-forgeable*. The term *secure* is related towards confidentiality of the scheme, where confidentiality means, that an attacker cannot understand the contents of the message M, even after knowing the ciphertext C and the nonce N. One way to achieve this is to make the encryption scheme indistinguishable from a random permutation; this is a standard definition that is used in many security proofs such as the security proofs of the modes of operation for block ciphers.

The goal of authenticated encryption is to provide privacy and integrity. Two possible notations are used for the authenticity of AE, INT-PTXT (Integrity of the plaintexts) – $M = D_K(C)$ was never encrypted by the sender, it is computationally infeasible to produce a ciphertext decrypting to a message that is never encrypted by the sender and INT-CTXT (Integrity of the ciphertexts) – C was never transmitted by the sender, it is computationally infeasible to produce a ciphertext not previously produced by a sender. Privacy goals for encryption schemes consists of indistinguishability (advantage of a reasonable adversary determining what message was sent, M or M') and non-malleability (advantage of a reasonable adversary being able to change the message to be meaningful), each of which are considered under either chosen-plaintext or chosen-ciphertext attack. This

leads to two indistinguishability notations of security IND-CPA (indistinguishability under a chosen plaintext attack), IND-CCA (indistinguishability under a chosen ciphertext attack) and two non-malleability security notations, namely NM-CPA (non-malleability under a chosen plaintext attacks), NM-CCA (non-malleability under chosen ciphertext attack).

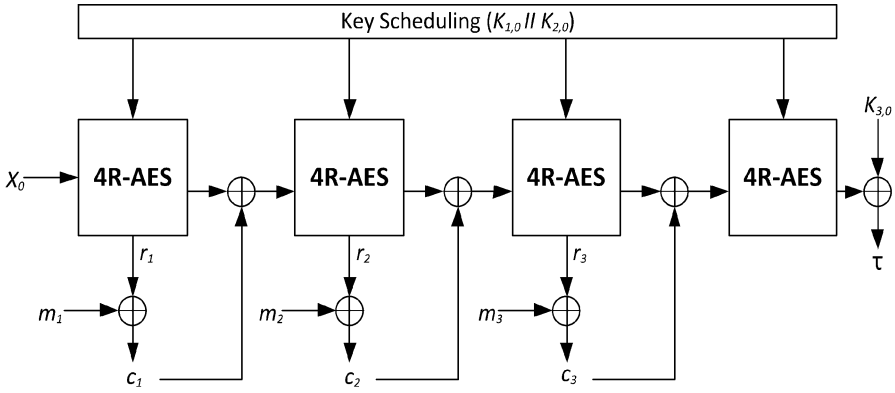
3.1 ASC-1: An Authenticated Encryption Stream Cipher

The idea behind single pass Authenticated Encryption is to achieve faster encryption and message authentication by performing both the encryption and message authentication in a single pass as opposed to the traditional approach which requires two passes, i.e., one for encryption and other for authentication. In the past several single pass provable secure AE schemes have been proposed, for example, IACBC and IAPM [2]. Other provably secure AE schemes that use a block cipher as a building block were also presented in [3, 4]. In this section we describe a single pass authenticated encryption scheme ASC-1 [6]. The design of ASC-1 authenticated encryption scheme uses a four round Advanced Encryption Standard (AES) as a building block. The scheme uses single cryptographic primitive to achieve both message secrecy and authenticity. It is also shown that ASC-1 is secure if one cannot tell apart the case when the scheme uses random round keys from the case when the round keys are derived by a key scheduling algorithm.

As shown in Figure 1, ASC-1 is a single pass AE scheme that uses four round AES with 128-bit key as an underlying block cipher. ASC-1 is divided into two steps – Initial phase generation, Encryption in CFB (Cipher feedback)-like mode and authentication of the data. At the decryption side, same steps are repeated and the computed tag is matched with the received tag for verification.

Initial phase generation – Initial phase consists of an initialization vector X_0 and three keys $K_{1,0}$, $K_{2,0}$, $K_{3,0}$. To calculate these values ASC-1 uses 56-bit of the counter and applies 128-bit AES block cipher to $0^{70}\|00\|Cntr$, $0^{70}\|01\|Cntr$, $0^{70}\|10\|Cntr$, $l(M)\|00000011\|Cntr$, using Master key K_M, where $l(M)$ is the 64-bit representation of the bit length of the Message M.

Encryption Process – Before initializing encryption process, Keys $K_{1,0}$ and $K_{2,0}$ are concatenated together and AES-256 key scheduling algorithm is applied to derive 14 round keys. Keys K_2, K_3, K_4 and K_5 are used as round keys in the first round and Keys K_7, K_8, K_9 and K_{10} are used in the second round. Keys K_{11} and K_{12} are used as whitening keys in the first and second rounds of 4R-AES transformation respectively. In AES key scheduling round



$$X_0 = E_K(0^{70}\|00\|Cntr), \quad K_{1,0} = E_K(0^{70}\|01\|Cntr), \quad K_{2,0} = E_K(0^{70}\|10\|Cntr),$$

$$K_{3,0} = E_K(l(M)\|0^6\|11\|Cntr)$$

Figure 1 The encryption algorithm of ASC-1. The message consists of three blocks. The ciphertext consists of the counter value, three ciphertext block and authentication tag.

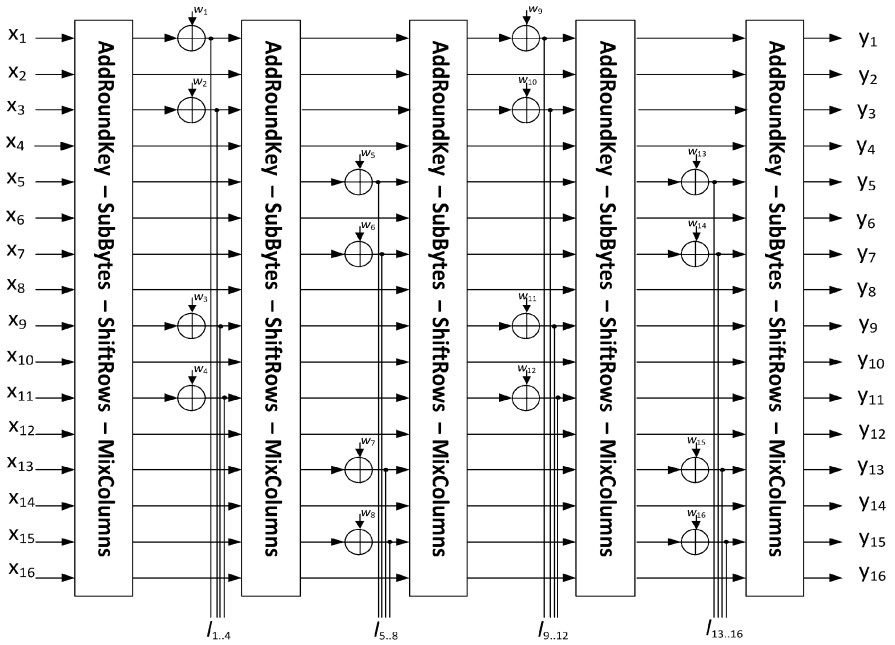


Figure 2 The 4R-AES transformation.

keys can either be generated on-the-fly or they can be stored in the internal memory. On the other hand, in ASC-1, because of using K_1 and K_{11} for key whitening, it is only possible to store the keys in the memory during the key setup phase, and then read them from this memory whenever they are required by the encryption/decryption unit.

ASC-1 Encryption Block consists of four round AES as shown in Figure 2. To initialize the encryption module, a 128 bit initialization vector is provided as an input to the ASC-1 encryption algorithm. ASC-1 performs a number of transformations to the input data to give a 128-bit leak $l_1, l_2, \ldots, l_{16}$ and output state $y_1, y_2, \ldots, y_{16}$. ASC-1 stream cipher performs four discrete transformations: *AddRoundKey*, *SubBytes*, *ShiftRows* and *MixColumns*. Four bytes are leaked at the end of every round and positions of the leaks depend on the number of the round (even or odd). Finally, a whitening key byte is added before each extracted byte. The AES-256 key scheduling algorithm is again applied to $K_{13} \| K_{14}$ to derive 14 keys that are used by the third and the fourth 4R-AES transformation, and the process is repeated as long as we need new keys.

4 Proposed ASC-1 Architecture

The high-level architectural organization of the ASC-1 encryption core is presented in Figure 3. The system is divided into five logical blocks. The initial input interface is responsible for feeding data to the key logic and the processing core. Key logic handles all the key scheduling operations and processing core block performs all the main encryption process. SBox block is a ROM that is used for the SubBytes transformation by key logic and core block. Finally the control unit is used for the synchronization and communication with the external logic. Let us further look into the functionality of each logic block in detail.

Initial input interface – For initial phase generation, i.e., initialization vector X_0 and three keys $K_{1,0}$, $K_{2,0}$, $K_{3,0}$, a new counter/nonce is loaded. The initial input interface concatenates the values of the counter with the predefined values stored in the local registers. The processing core unit is then notified that an initial state is available for processing.

Key Logic – In the above mentioned scheme, every encryption round requires a new round key. Once the new key is loaded, the key logic block starts generating round keys based on a single external key. Three possible approaches can be used to generate round keys – Online approach, Offline or stored-key approach and use of an external source ex., key generator or an

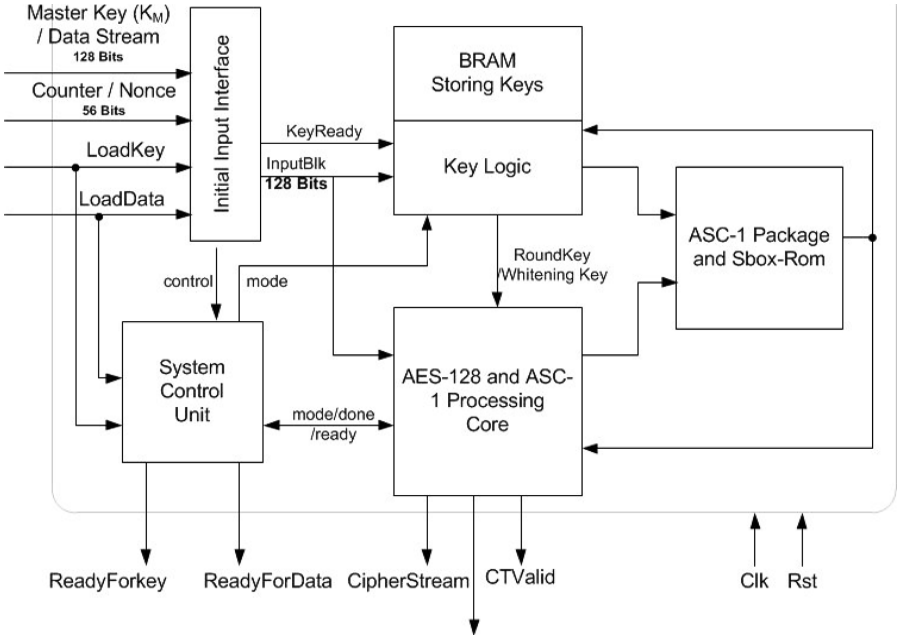


Figure 3 Block diagram of ASC-1.

external processor. Our design is based on "offline" or "stored-key" approach, where all the round keys are calculated upon the reception of the initial cipher key before the start of encryption and stores them in a local memory. The memory is accessed at every encryption round in order to provide the necessary round key. Opting for stored-key approach has many advantages in our design as compared to "online" or external source approach ex., for initial phase generation same key (K_M) is used to encrypt initialization vector (IV), two initial keys ($K_{1,0}$, $K_{2,0}$) for key scheduling for ASC-1 encryption and key ($K_{3,0}$) for authentication of data. The round keys derived from the Master key (K_M) is stored in the memory and during the encryption process right round key is accessed from the memory to perform encryption operation. In case encryption of stream data, 14 round keys are derived by loading 256-bit key, i.e., $K_{1,0} \| K_{2,0}$ to the key logic unit for key expansion. The key logic block performs two main functions. The key expansion process and read/write round keys to the memory block. The first one is performed whenever a new cipher key is inserted to the block and second one is to fetch round keys from the local memory for encryption process.

AES and ASC-1 processing core – The processing core block consists of AES-128 and ASC-1 encryption process. AES encryption core is used only for the generation of Initialization vector and keys used in ASC-1. Once the IV and the keys are encrypted using AES-128, keys are fed into the key logic block for the calculation of round keys and IV is used to initiate ASC-1 scheme. The underlying block used in ASC-1 is AES, so same transformations are applied to the block but in different order. AddRoundKey transformation is the first block and after MixColumns, KeyWhitening is applied to the specific bytes before extracting from intermediate rounds. Four round AES ASC-1, operates in a Cipher Feedback (CFB) mode which means that the processing of each plaintext block has to be completed before the processing of the next one starts. Therefore, implementation presented here is sequential. However from Figure 3, parts of implementation could be implemented in parallel architecture.

Systems control unit – The unit is implemented as a finite state machine to supervise the core between AES and ASC-1, generate address for accessing the round keys from the block and handle communication between blocks. The unit generates the signal to notify the external source that a new plaintext may be loaded as soon as core is ready.

Authentication Tag (τ) – Finally the authentication tag is calculated once n numbers of block are encrypted (the maximum number of messages and maximum length to be encrypted is 2^{48}),

4.1 Frame Delay

The end goal of ASC-1 authenticated encryption scheme is to achieve both message secrecy and authenticity in a single cryptographic primitive with the focus to achieve high throughput and minimal overhead for wireless devices. Based on the design of ASC-1, two different approaches are proposed – Key setups during transmission or parallel key setup with the encryption core.

As shown in Figure 4, when the frames passes through the core, only payload have to be encrypted and rest remains in the plaintext. However to initiate the encryption of the payload requires Initial phase generation i.e. calculating initialization vector and keys for the encryption based on Counter ($Cntr$) and Master Key (K_M). The process is repeated for every frame, where Counter values are varied but Master Key remains same for the session.

In the first approach, the key setup is triggered at the start of the transmission. The unencrypted prefix (header) of the frame is validated and passed though the bypass unit and waits for the encrypted and authenticated pay-

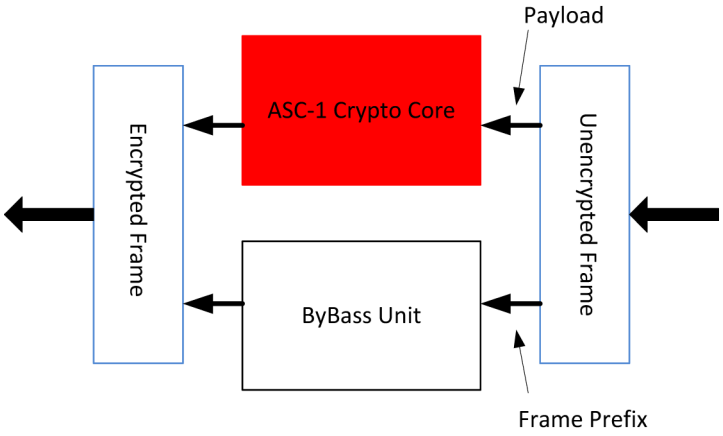


Figure 4 Crypto architecture.

load. Once the encryption is done, the whole frame is packed and sent to the transmitter. During the transmission of the frame, key setup for next frame is performed and stored in the logic. Based on previous sections, two hardware architectures were investigated: basic iterative and parallel architecture. Depending on the area constraints and acceptable delay for the specific applications, either of the architecture for initial phase can be chosen. Based on our results, iterative architecture has a latency of 248 ns, whereas parallel architecture takes about half the time but three times in area. However in either of the architectures this approach may cause some minor end-to-end delays.

To overcome these delays, keys can also be computed in parallel with the encryption core. In this approach, initial phase is generated before the start of the transmission and keys are stored in the internal logic. For subsequent frames new keys are generated in parallel with the encryption core processing last block of the previous frame. This approach may not cause any delays but it comes with the cost of high area consumption.

5 Implementation Results

The results of hardware implementation of "ASC-1: An Authenticated Encryption Stream Cipher" are tabulated in this section. ASC-1 is implemented in VHDL and the target device is Xilinx Spartan-3 sxc3s700an FPGA. The software used for this design is Xilinx ISE-12.4. This is used for writing,

Table 1 Performance of AES-128 encryption in parallel and iterative architecture.

AES-128 Encryption		
Performance	Iterative	Parallel
Number of Slices	1736	15550
Number of Clock Cycles	62	30
Latency (ns)	248	120
Throughput (Gbps)	0.516	up to 32

debugging and optimizing, and all the simulations are carried out in ISim simulator.

5.1 ASC-1 Performance

In an ASC-1 scheme, the underlying block cipher, i.e. AES, is used only in the forward encryption direction for both ASC-1 encryption and decryption. This characteristic make ASC-1 an attractive candidate for hardware where area is limited. Each round in the scheme consists of four basic transformations, i.e., SubBytes, ShiftRows, MixColumns and AddRoundKey. The S-Box byte substitution function can be implemented either by using combinational logic or using a 256×8 bit look-up table, using ROM (Read Only Memory). Use of ROM is the most optimal implementation in terms of area/performance – in an FPGA. To access ROM, inputs used as addresses and output is acquired at the data out bus. A state matrix consists of 16 bytes and for each byte substitution 16 ROMs have to be used. FPGA used in this implementation Xilinx Spartan-3AN provides fast on-chip memories, called BlockRAMs. BlockRAMs can be configured as dual port ROMs. This reduces the amount of ROMs in half, i.e. 8. This whole process requires only one clock cycle. Other three transformations during the encryption/decryption process are basic operations and takes minimal resources.

Table 1 presents the detail implementation results for the AES-128 encryption system. AES encryption is used during the initial phase i.e., for the calculation of IV and keys used for encryption and authentication of data. Same key is used to encrypt all the initial values in ECB non-feedback mode. With encryption in non-feedback mode, processing of data blocks can be performed independently from other blocks and all the blocks can be encrypted in parallel. Following table shows the throughput, latency and area used for parallel and iterative hardware architectures. The system is set to 250 MHz with a clock cycle of 4 ns.

Table 2 Performance of ASC-1 encryption core iterative architecture.

Performance	Iterative
Number of Slices	1796
Number of Clock Cycles	41
Latency (ns)	164
Throughput (Gbps)	0.780

A huge trade-off between area and performance of the system can be clearly seen. The number of slices used in a parallel architecture is almost nine times as much as in an iterative architecture. However, on the other side, the throughput of the Iterative architecture is much lower than parallel architecture.

Table 2 provides the results of ASC-1 encryption core; the core consists of 4-Round AES and operates in CFB mode to compute an authentication tag over the encrypted message. In feedback modes it is not possible to encrypt next block of data until encryption of previous block is completed. As a result, data blocks must be encrypted sequentially, with no capability of parallel processing.

As compared to AES iterative architecture, data is processed only four times instead of ten times and initial and final rounds are not included. The order of bit transformations inside each round is also different as compared to AES; AddRoundKey transformation is performed at the start of each round unlike AES.

6 Conclusion

In this paper, we presented a single pass authenticated encryption scheme: ASC-1 for wireless reconfigurable chips with the focus to achieve high throughput and low overhead. The goal of this scheme is to address two main security objectives, i.e., Confidentiality and Authenticity. This is achieved by performing both the encryption and message authentication in a single pass as opposed to the traditional approaches, which requires two passes. Additionally, we have designed and implemented ASC-1 authenticated encryption scheme on FPGAs. The crypto module, i.e., ASC-1 is placed on the re-configurable chip is responsible for the confidentiality and integrity of the data flow passing through it from both the sides. We have also explored any possible frame delay due to the initial key setup with every frame. Based on the available resources, two different approaches are proposed.

After analyzing the performance parameters, we conclude that ASC-1 is suitable for low-cost low-power reconfigurable wireless devices with negligible or no delays. The resulting implementation consumes moderate number of slices on FPGA and achieves throughput in the range of 0.8 Gbps. Comparing with traditional two pass approaches, the presented design demonstrates high throughput and small area to performance ratio.

References

[1] D. Ma and G. Tsudik. Security and privacy in emerging networks. IEEE Wireless Communications, 17(5), 12–21, October 2010.

[2] C. Jutla. Encryption modes with almost free message integrity. In Advances in Cryptology EUROCRYPT 2001, Lecture Notes in Computer Science, Vol. 2045, pp. 529–544. Springer Verlag, Berlin, 2001.

[3] V.D. Gligor and P. Donescu. Fast encryption and authentication: XCBC encryption and XECB authentication modes. In Proceedings of Fast Software Encryption 2001, M. Matsui (Ed.), Lecture Notes in Computer Science, Vol. 2355. Springer Verlag, Berline, 2001.

[4] P. Rogaway, M. Bellare, J. Black, and T. Krovetz. OCB: A block-cipher mode of operation for efficient authenticated encryption. In Proceedings of 8th CCS. ACM, New York, 2001.

[5] M. Bellare and C. Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In Advances in Cryptology – ASIACRYPT 2000, Vol. 1976, pp. 531–545. Springer Verlag, Berlin, 2000.

[6] G. Jakimoski and S. Khajuria. ASC-1: An authenticated encryption stream cipher. In Selected Areas in Cryptography 2011, Lecture Notes in Computer Science, Vol. 7118, pp. 356–372. Springer Verlag, Berlin, 2011.

Biographies

Samant Khajuria is a PhD student at the Center for Tele Infra Structure (CTIF) Copenhagen at Aalborg University (Denmark). He received his Bachelor in Electronics and Communication in 2006 from PES Institute of Technology – Bangalore (INDIA) and Masters Degree in Communication networks (specializing in security) in 2008 from Aalborg University Copenhagen. He started as a research assistant at the Center for Wireless Systems and Applications (CWSA), before starting his PhD. Major research areas include Cryptography, Cognitive Radio, Computer Networks, FPGAs.

Birger Andersen is a Professor at Copenhagen University College of Engineering, Denmark, and Director of Center for Wireless Systems and Applications (CWSA) related. He received his M.Sc. in Computer Science in 1988 and his Ph.D. in Computer Science in 1992, both from University of Copenhagen. He was an assistant professor at University of Copenhagen, a visiting professor at Universität Kaiserslautern, Germany, and an associate professor at Aalborg University. Later he joined the IT Department of Copenhagen Business School, Denmark, and finally Copenhagen University College of Engineering. He is currently involved in research in wireless systems with a focus on security.