Design of Cryptographic Core for Protecting Low Cost IoT Devices

Dennis Agyemanh Nana Gookyi and Kwangki Ryoo

Abstract The security challenge of the Internet-of-Things (IoT) has to do with the use of low cost and low power devices in the communication network. This problem has given rise to the field of lightweight cryptography where less computational intensive algorithms are implemented on constrained devices. This paper provides the integration of lightweight encryption and authentication algorithms in a single crypto core. The crypto core implements a unified 128-bit key architecture of PRESENT encryption algorithm a new lightweight encryption algorithm. The core also implements a unified architecture of four lightweight authentication algorithms which come from the Hopper-Blum (HB) and Hopper-Blum-Munilla-Penado (HB-MP) family: HB, HB+, HB-MP, and HB-MP+. The hardware architectures share resources such as register, logic gates, and common modules. The core is designed using Verilog HDL, simulated with Modelsim and synthesized with Xilinx Design Suite 14.3. The core synthesized to 1130 slices at 189 MHz using Spartan6 FPGA device.

Keywords IoT · Lightweight cryptography · Encryption · Authentication
Crypto core · FPGA

1 Introduction

Lightweight encryption has gained a lot of attention in recent years. Many encryption algorithms are been proposed, tested and analyzed frequently. A popular lightweight encryption algorithm that has been accepted as an ISO/IEC standard is the PRESENT Algorithm [1]. PRESENT is a 128-bit key size, 64-bit block size, and a 31 round

D. A. N. Gookyi · K. Ryoo (✉)
Department of Information and Communication Engineering, Hanbat National University,
125 Dongseodaero, Yuseong-Gu, Daejeon 34158, South Korea
e-mail: kkryoo@hanbat.ac.kr

D. A. N. Gookyi
e-mail: dennisgookyi@gmail.com

J. J. Park et al. (eds.), *Advanced Multimedia and Ubiquitous Engineering*, Lecture Notes in Electrical Engineering 518,
https://doi.org/10.1007/978-981-13-1328-8_53

cipher. PRESENT requires 310 us to encrypt a block of data at 100 kHz. A newer lightweight encryption algorithm [2] was designed to reduce the encryption time and provide high throughput. This algorithm is a 128-bit key size, 64-bit block size, and an 8 round cipher. It requires 80 us to encrypt a block of data at 100 kHz. Both algorithms use the same 4-bit input ultra-lightweight substitution box (SBox). The permutation box (PBox) of PRESENT is simple bit permutation while that of the new algorithm involves the use of a key dependent one stage omega permutation network. This paper implements the hardware unification of these two lightweight encryption algorithms. Resources such as registers, logic gates, key generation algorithm, SBox, and PBox are shared to reduce hardware area.

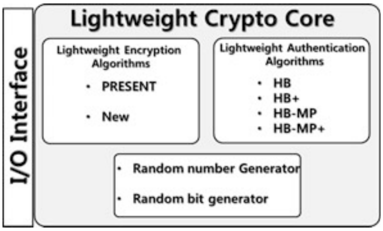
Lightweight authentication is an ever developing research area. These authentication algorithms use components such as pseudorandom number generators (PRNG) and cyclic redundancy check (CRC). The earliest form of lightweight authentication was proposed by Hopper and Blum which will go on to be known as HB [3] authentication protocol. Many variants of the HB protocol have been proposed in recent year with no real hardware implementation to analyze. This paper provides unified hardware architecture for four HB family lightweight authentication protocols: HB, HB+ [4], HB-MP [5] and HB-MP+ [6]. The proposed hardware architecture share resources such as random number and random bit generation modules, logic gates, a dot product module and a key generation module.

The unified crypto as core shown in Fig. 1 that incorporates lightweight encryption and authentication is implemented to give users the chance to choose between a high throughput but moderate security encryption and a low throughput but high-security encryption. It also provides users with authentication without the use of hash functions. The rest of this paper is organized as follows: Sect. 2 describes a summary of the algorithms in the crypto core, Sect. 3 describes the hardware implementation, Sect. 4 shows simulation and synthesis results and the conclusion and future work are covered in Sect. 5.

2 Cryptographic Core Algorithms

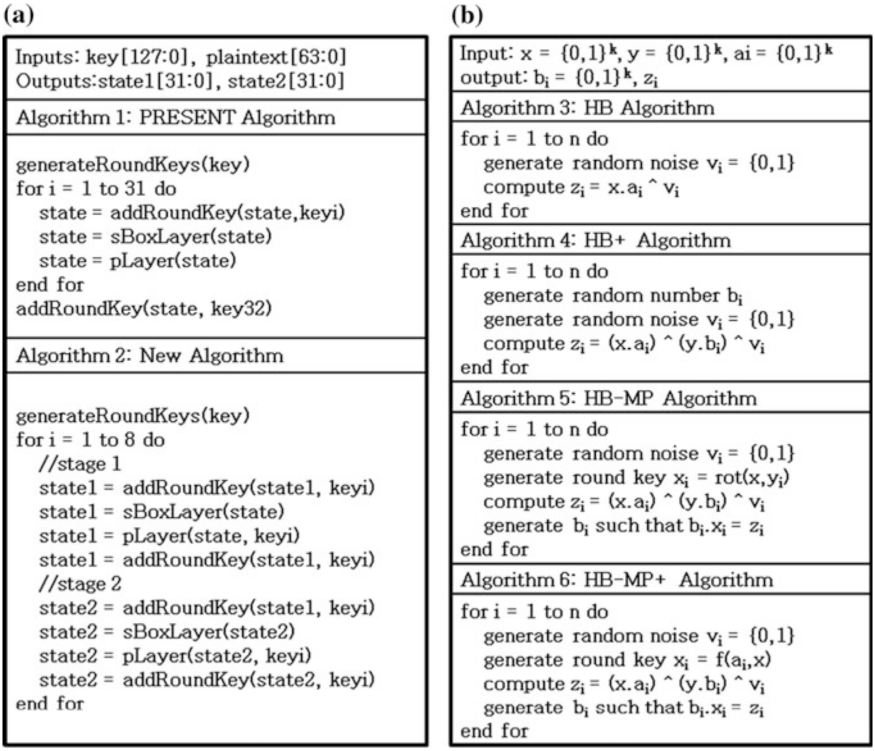
The cryptographic core is shown in Fig. 1. It consists of two lightweight encryption algorithms and four lightweight authentication algorithms which are described in this section.

Fig. 1 Lightweight cryptographic core



The lightweight encryption algorithms consist of the PRESENT algorithm and a new algorithm. The flow of the two algorithms is shown in Fig. 2a. PRESENT algorithm is an 80/128 bit key Substitution-Permutation (SP) cipher which uses 31 rounds to encrypt a block of 64-bit data. The new encryption algorithm is 128 bit key Feistel cipher which uses 8 rounds to encrypt a block of 64-bit data. The algorithms both use the same 4-bit input/output SBox. PRESENT encryption uses simple permutation as its PBox while the new encryption uses a key dependent one stage omega permutation network as its PBox. Both algorithms consist of operations such as generateRoundKeys (for the generation of round key), addRoundKey (for XORing round key and data), sBoxLayer (for passing data through an SBox) and pBoxLayer (for passing data through a PBox).

The crypto core also consists of HB, HB+, HB-MP and HB-MP+ authentication algorithms. The algorithms flow is shown in Fig. 2b. All algorithms consist of input keys, generation of random numbers and noise bits, dot products, and XOR units.



(a)

```
Inputs: key[127:0], plaintext[63:0]
Outputs:state1[31:0], state2[31:0]

Algorithm 1: PRESENT Algorithm

generateRoundKeys(key)
for i = 1 to 31 do
    state = addRoundKey(state,keyi)
    state = sBoxLayer(state)
    state = pLayer(state)
end for
addRoundKey(state, key32)

Algorithm 2: New Algorithm

generateRoundKeys(key)
for i = 1 to 8 do
    //stage 1
    state1 = addRoundKey(state1, keyi)
    state1 = sBoxLayer(state)
    state1 = pLayer(state, keyi)
    state1 = addRoundKey(state1, keyi)
    //stage 2
    state2 = addRoundKey(state1, keyi)
    state2 = sBoxLayer(state2)
    state2 = pLayer(state2, keyi)
    state2 = addRoundKey(state2, keyi)
end for
```

(b)

```
Input: $x = \{0,1\}^k$, $y = \{0,1\}^k$, $ai = \{0,1\}^k$
output: $b_i = \{0,1\}^k$, $z_i$

Algorithm 3: HB Algorithm
for i = 1 to n do
    generate random noise $v_i = \{0,1\}$
    compute $z_i = x.a_i$ ^ $v_i$
end for

Algorithm 4: HB+ Algorithm
for i = 1 to n do
    generate random number $b_i$
    generate random noise $v_i = \{0,1\}$
    compute $z_i = (x.a_i)$ ^ $(y.b_i)$ ^ $v_i$
end for

Algorithm 5: HB-MP Algorithm
for i = 1 to n do
    generate random noise $v_i = \{0,1\}$
    generate round key $x_i = rot(x,y_i)$
    compute $z_i = (x.a_i)$ ^ $(y.b_i)$ ^ $v_i$
    generate $b_i$ such that $b_i.x_i = z_i$
end for

Algorithm 6: HB-MP+ Algorithm
for i = 1 to n do
    generate random noise $v_i = \{0,1\}$
    generate round key $x_i = f(a_i,x)$
    compute $z_i = (x.a_i)$ ^ $(y.b_i)$ ^ $v_i$
    generate $b_i$ such that $b_i.x_i = z_i$
end for
```

Fig. 2 Lightweight algorithms **a** encryption algorithms **b** authentication algorithms

3 Proposed Hardware Architecture

Figure 3 shows the proposed hardware architecture datapath for the unified lightweight ciphers. The input consists of the *plaintext* and *key* while the output is the *ciphertext* which concatenates two 32 bit registers *dreg_msb* and *dreg_lsb*. Multiplexers are used to route the data based on the selection signal *protocol*. When the *protocol* signal is asserted, PRESENT encryption is activated and when the *protocol* signal is de-asserted, the new encryption is activated. The key generation algorithm is also shown in the same Figure where the new algorithm consists of only left rotation by 25 bits while the PRESENT algorithm consists of rotation, SBOX and XORing with the *counter* value. Two 32 bit input SBOX and PBOX are used.

Figure 4 shows the proposed hardware architecture datapath for the unified lightweight authentication algorithms. The inputs consist of *ran_num_in*, *key1*, and *key2* while the output is the *auth_out*. The architecture consists of random bit and a random number generation unit which uses linear feedback shift registers (LFSR). The dot product unit computes the dot product of the key and the random number using AND (&) and XOR (^) gates in the equation: ^ (*key* [63:0] & *random* [63:0]). The key generation unit computes round keys for HB-MP and HB-MP+. The *protocol* signal is a 2 bit signal (*protocol* [1:0]) that activates HB (*protocol* [1:0] = 00), HB+ (*protocol* [1:0] = 01), HB-MP (*protocol* [1:0] = 10) and HB-MP+ (*protocol* [1:0] = 11).

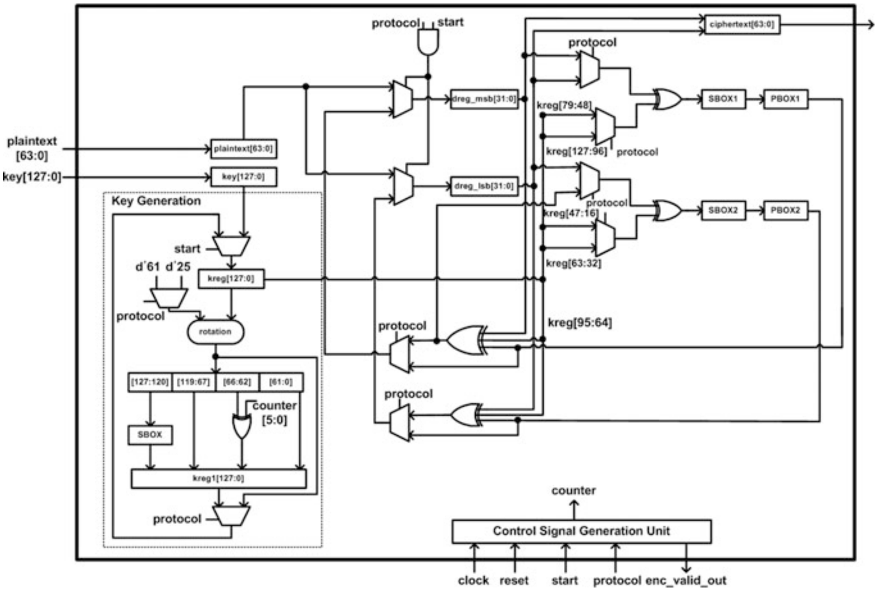


Fig. 3 A unified encryption hardware architecture

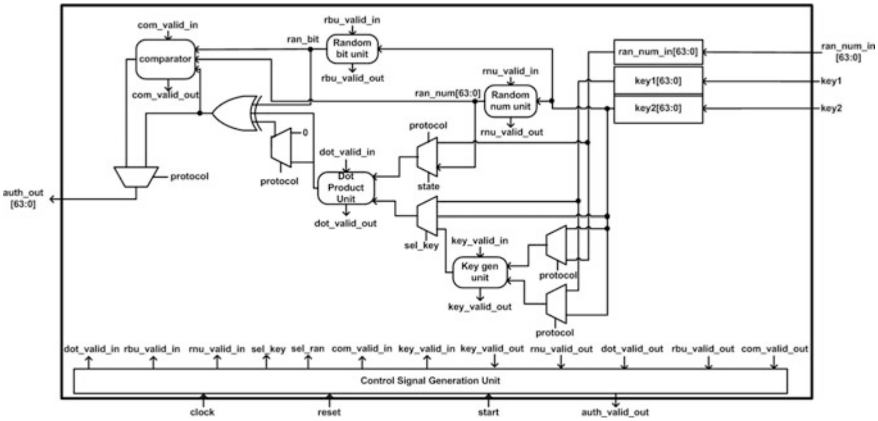


Fig. 4 A unified authentication hardware architecture

4 Results and Discussion

The hardware architecture of the proposed crypto core was designed using Verilog HDL and was verified using FPGA. Xilinx Spartan6 was used for the purposes of synthesis and Mentor Graphics ModelSim SE-64 10.1c was used for the purposes of simulation. The synthesis results are tabulated in Table 1. From the results, the proposed crypto core saves up to 443 slices as compared to implementing the algorithms individually.

Table 1 Hardware synthesis results

Algorithms	FPGA device	Area (slices)	Max Freq. (MHz)
PRESENT [7]	Spartan3 XC3S400	202 (2.5%)	254
NEW [2]	Virtex6 XC6VLX760	196 (0.17%)	337
HB [8]	Spartan6 XC6SLX100	77 (0.06%)	311
HB+ [8]	Spartan6 XC6SLX100	302 (0.24%)	223
HB-MP [8]	Spartan6 XC6SLX100	430 (0.34%)	157
HB-MP+ [8]	Spartan6 XC6SLX100	366 (0.3%)	160
Proposed Crypto Core	Spartan6 XC6SLX100	1130 (0.9%)	189
Area saving		443	–

5 Conclusion

In this paper, we propose the hardware architecture of an integrated crypto core that combines two lightweight encryption algorithms and four lightweight authentication algorithms. The core synthesized to 1130 slices at 189 MHz maximum clock frequency on Spartan6 FPGA device. The core saves up to 443 slices as compared to implementing the algorithms individually. In future works, we will be looking at adding a key sharing algorithm to the core and implementing it on a System-on-Chip (SoC) platform.

Acknowledgements This research was supported by the MSI (Ministry of Science, ICT and Future Planning), Korea, under the Global IT Talent support program (IITP-2017-0-01681) and Human Resource Development Project for Brain scouting program (IITP-2016-0-00352) supervised by the IITP (Institute for Information and Communication Technology Promotion).

References

1. Bogdanov A, Paar C, Poschmann A (2017) PRESENT: an ultra-lightweight block cipher. LNCS, vol 4727, pp 450–466. Springer, Berlin
2. Gookyi DAN, Park S, Ryoo K (2017) The efficient hardware design of a new lightweight block cipher. Int J Control Autom 1(1):431–440
3. Hopper NJ, Blum M (2001) Secure human identification protocols. In: Advances in cryptology —ASIACRYPT 2001, LNCS, vol 2248, pp 52–56. Springer, Heidelberg
4. Juels A, Weis SA (2005) Authenticating pervasive devices with human protocols. In: LNCS, vol 3621, pp 293–308. Springer, Berlin
5. Munilla J, Peinado A (2009) A further step in the HB-family of lightweight authentication protocols. Comput Netw 51(9):2262–2267
6. Leng X, Mayes K, Markantonakis K (2008) HB-MP+ Protocol: an improvement on the HB-MP protocol. In: IEEE international conference on RFID. IEEE Press, pp 118–124
7. Sbeiti M, Silbermann M, Poschmann A, Paar C (2009) Design space exploration of PRESENT implementation for FPGAs. In: 5th southern conference on programmable logic. IEEE Press, pp 141–154
8. Gookyi DAN, Ryoo K (2017) Hardware design of HB type lightweight authentication protocols for IoT devices. In: International conference on innovation convergence technology. INCA, Korea, pp. 59–60