Research Journal of Information Technology

ISSN 1815-7432

Academic
Journals Inc.

www.academicjournals.com

OPEN ACCESS Research Journal of Information Technology

ISSN 1815-7432
DOI: 10.3923/rjit.2016.17.28

Research Article
Lightweight PRINCE Algorithm IP Core for Securing GSM Messaging using FPGA

[1,2]Yasir Amer Abbas, [1]Razali Jidin, [3]Norziana Jamil and [4]Muhammad Reza Z'aba

[1]College of Engineering, Universiti Tenaga Nasional, Selangor, Malaysia
[2]College of Engineering, Diyala University, Baquba, Diyala, Iraq
[3]Center of Information and Network Security, Universiti Tenaga Nasional, Selangor, Malaysia
[4]MIMOS Berhad, Technology Park Malaysia, Kuala Lumpur, Malaysia

Abstract

Monitoring and managing data from a remote asset to optimize maintenance and operation schedules using wireless communication have received more attention recently than before. Meanwhile, the rapid development of global system for mobile communication (GSM) systems makes communicating parties more vulnerable than ever to security attacks. The weaknesses in GSM security, such as flaws in implementation and cryptography algorithms, still need additional improvements and investigation to enhance the system performance in terms of security, cost and power consumption. In this study, a new security system design for securing GSM messaging with a lightweight PRINCE algorithm Intellectual Property (IP) Core using Field Programmable Gate Arrays (FPGA) is proposed. An energy and cost-efficient implementation of PRINCE algorithm implemented in an environment of a microprocessor system using XILINX FPGA board is developed. A complete microprocessor system is designed consisting of MicroBlaze processor, memory, serial communication and a PRINCE IP Core that can be contained in a single XILINX VIRTEX chip. The system can cipher the data using PRINCE algorithm on a VIRTEX-403 FPGA evaluation board and using GSM modems to communicate over a cellular network. Results show that the proposed design achieves a high speed of 31.765 MHz with a throughput of 2.032 Gbps at a low power consumption of 0.165 W and an efficiency of 2.126 Mbps per slice.

Key words: Lightweight cryptography, PRINCE, low power, FPGA, VHDL, IP Core

Received: January 02, 2016 Accepted: February 25, 2016 Published: March 15, 2016

Citation: Yasir Amer Abbas, Razali Jidin, Norziana Jamil and Muhammad Reza Z'aba, 2016. Lightweight PRINCE algorithm IP Core for securing GSM messaging using FPGA. Res. J. Inform. Technol., 8: 17-28.

Corresponding Author: Yasir Amer Abbas, College of Engineering, Universiti Tenaga Nasional, Selangor, Malaysia

Competing Interest: The authors have declared that no competing interest exists.

Data Availability: All relevant data are within the paper and its supporting information files.

INTRODUCTION

Today's industrial facilities such as electric power stations, oil refineries, chemical factories and manufacturing facilities are large and complex. The operators of plants must continuously monitor and control several sections of their facilities to operate them properly. Modern networking technology has made remote command and control of plants possible. These remote command and control networks are commonly known as Supervisory Control and Data Acquisition (SCADA) networks. In general, the SCADA system includes a main station (master station) and a number of substations (slave stations) with geographically distributed Remote Terminal Units (RTUs). Two types of communications exist for SCADA systems, namely, wired (directly wired, power line carrier and fiber optic) and wireless (microwave and cellular) (Gaushell and Block, 1993; Thomas *et al*., 2004; Igure *et al*., 2006). The SCADA-based wired communication system is difficult and costly to implement when natural barriers such as seas or mountains are found between the stations (Hong and Lee, 2010; Parikh *et al*., 2010; Hong *et al*., 2010). For instance, power stations in an island are geographically separated from those in the main land (Khan *et al*., 2015). Hence, establishing a wired communication system between these stations is not feasible and is expensive.

In this study, the wireless communications system based on GSM is cheaper and easier to install than the wired communications system in building a low-cost system for monitoring remote assets. The GSM communication system is widely used in monitoring remote assets for varies applications, such as home security systems (El-Medany and El-Sabry, 2008; Ahmad *et al*., 2011), monitoring and managing databases (Peijiang and Xuehua, 2008), temperature detection (Li *et al*., 2010) and public transportation management system (Al-Rousan *et al*., 2004; Baldini *et al*., 2010). However, the openness of wireless communications based on GSM makes the communicating parties more susceptible to failures and weakness that can be maliciously exploited.

Although, GSM attempted to prevent interception by using several techniques such as frequency hopping, the real-time interception of the exchanged information is completely practical. However, some commercial equipment are capable of simultaneously intercepting several collocated subscribers (Gonzalez-Castano *et al*., 2002). Although GSM was intended to be a secure wireless system and considered user authentication and over-the-air encryption, it is completely vulnerable to several attacks with each of the attacks aiming at a part of the network. The most important security flaws of the GSM are Subscriber Identification Module (SIM) card cloning, flaws in cryptographic algorithms and short range of protection and other issues (Toorani and Beheshti, 2008; Cleveland, 2006).

Several researchers focused on GSM encryption algorithms such as A5/3, A5/4, A5/1 and RC4 algorithms implemented in Field Programmable Gate Arrays (FPGA) to reduce the process time (Vrentzos *et al*., 2006; Ahmad, 2009; Gupta *et al*., 2013) but the security issue is not deeply investigated. Other published studies proposed the enhancement of the security of GSM. The AES-128 algorithm was used to secure voice communication through the GSM network based on FPGA (Ozkan *et al*., 2011). However, the authors used the Matlab software to implement the interface. On the one hand, a large number of Configurable Logic Blocks (CLB) are used for the cipher system. On the other hand, the mod operation is used as a ciphering technique in FPGA modules for securing voice transmitted through mobile Bluetooth (Firdaus and Yaakob, 2009). The security system used a simple mod operation for encryption and hence it can easily be recovered by a third party.

In our previous studies in (Abbas *et al*., 2014a, b) the PRINCE algorithm was implemented in FPGA to encrypt and decrypt the test data, resulting in improved processing time as well as lowered power consumption. Afterward, an energy and cost-efficient FPGA design used to cipher data transmission in a GSM modem is implemented using a test 8 bit-block cipher algorithm to provide a more secure wireless communication.

In the current study, a new approach for a real-time cipher system is proposed to improve the security of remote assets through a wireless communication system using the FPGA VIRTEX-403 board based on PRINCE algorithm. The communication system is more secure by using PRINCE algorithm to encrypt and decrypt data. The proposed system consists of two Intellectual Property (IP) Cores used to encrypt and decrypt the data in the GSM modem by using the PRINCE algorithm. The performance (e.g., power consumption and maximum frequency and slices) of the FPGA system designed for IP Cores is evaluated.

MATERIALS AND METHODS

Embedded system design using FPGA

Xilinx embedded system design: The Xilinx Embedded Development Kit (EDK) is used to design a complete embedded system for implementation in a Xilinx FPGA board. The EDK is a component of the Integrated Software

Environment (ISE) design suite for embedded system editions. The ISE is a Xilinx development system product required to implement Very High Definition Language (VHDL) designs into Xilinx programmable logic devices. The EDK includes three main system tools. First is the Xilinx Platform Studio (XPS) system tool, which is used to develop the embedded processor hardware. Second is the Software Development Kit (SDK) which is based on the Eclipse open-source framework. The SDK is used to develop the embedded software application. The SDK is also available as a standalone program. The last tool is the embedded processing IP cores, which include processors and peripherals (Xilinx Inc., 2013). Figure 1 shows the embedded design process flow.

The embedded system development is divided into two main parts: Hardware and software. The hardware development can use the XPS to design the processor and other peripherals connected to the system buses. This hardware can be transferred to the software design part, the SDK, as Microprocessor Hardware Specification (MHS) files. The SDK can create different applications for the hardware design using C and C++.

Microprocessor with GSM communication: The communication between the FPGA board and GSM modem is established through a universal asynchronous receiver/transmitter (UART) interface and AT commands are sent to control the GSM modem through the UART. The UART in the FPGA consists of three parts: Transmitting, receiving and baud rate generator. The transmitting part of the UART is responsible of handling received instructions by outputting the data according to the UART protocol. The receiving part of the UART monitors the line, a line that generates a falling edge means that data is transmitted in the line. The baud rate is a simple frequency divider, the sending and receiving of the UART must have the same baud rate. The GSM modem and the FPGA board use serial communication to exchange data between them (Li *et al*., 2010). Kuo *et al*. (2005) implemented the IP core for AES on FPGA to encrypt and decrypt the data with different IP cores. However, their results reveal a high maximum frequency and low number of slices but the obtained throughput and efficiency are low. The work in (Ozkan *et al*., 2011) has used the AES algorithm to be implemented on FPGA to perform the encryption of the speech after converting it to digital bit stream. About 4162 slices were used for encryption and about 2722 slices were used for decryption which is considered too high. On the other hand, the GSM devices are used by Misal *et al*. (2014) to send and receive the Short Message Service (SMS) from the control component without securing SMS. The system can lock or unlock the door using the SMS system that is based on the GPRS and GSM services but the security of SMS transmission was very low. The models designed in FPGA based UART using GSM design (Gaikwad, 2013) still need more investigations to further improve the security of text messages (SMS).

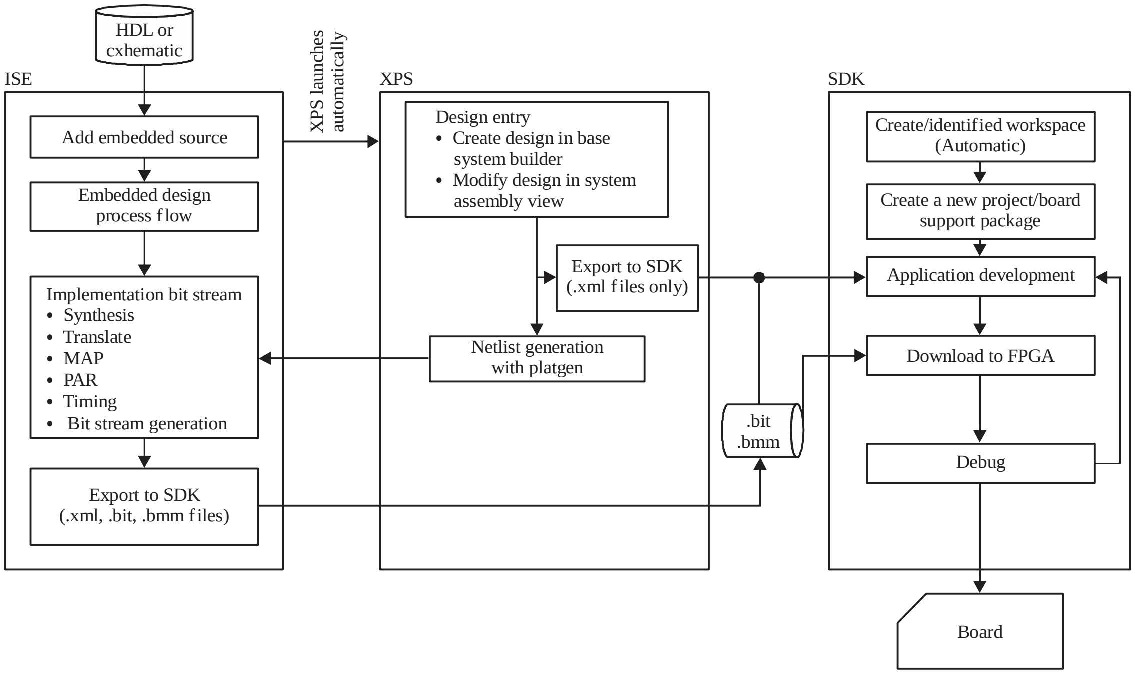

HDL or
cxhematic
ISE
Add embedded source
Embedded design process flow
Implementation bit stream
• Synthesis
• Translate
• MAP
• PAR
• Timing
• Bit stream generation
Export to SDK (.xml, .bit, .bmm files)
XPS launches automatically
XPS
Design entry
• Create design in base system builder
• Modify design in system assembly view
Export to SDK (.xml files only)
Netlist generation with platgen
.bit
.bmm
SDK
Create/identified workspace (Automatic)
Create a new project/board support package
Application development
Download to FPGA
Debug
Board

Fig. 1: Embedded design process flow (Xilinx Inc., 2013)

Serial communication is the process of sequentially sending data one bit at a time. The serial port on a PC can be set to full-duplex mode, which means that it can send and receive data at the same time. Any information or data is encapsulated by a start bit and a stop bit to form a data frame. Both the receiver and the transmitter must agree on the number of data bits and the baud rate before data exchange starts.

The soft-core MicroBlaze processor on the Virtex-403 FPGA board is used to process the received data. The basic MicroBlaze architecture consists of 32-bit general-purpose registers, an Arithmetic Logic Unit (ALU), a shift unit and two levels of interrupts. The majority of MicroBlaze instruction sets are achievable within a single-cycle execution. Figure 2 shows the FPGA model design communicating with GSM through an RS-232 UART.

AT commands: AT commands are used to control the operation of the GSM modem. Specific AT commands must be used to receive or send SMS through the GSM modem. Some of the commands used to operate the modem are shown in Table 1. The GSM modem will reply accordingly to the host for every command that it receives. An error message will be issued if the command sent to the GSM modem is not correct or not within the time limit.

A sample code for the communication module using the AT commands to test transmission in the form of SMS over the GSM wireless network is given in Fig. 3. Basically, the code uses three registers to send and receive SMS between the FPGA and the GSM modem. The code is written in C using AT commands in the program to communicate the encryption and decryption of the SMS through these three registers. The registers are for the encryption IP Core, decryption IP Core and UART.

PRINCE algorithm: PRINCE is a 64 bit Substitution-Permutation Network (SPN) lightweight block cipher supporting a 128 bit key (Borghoff *et al*., 2012). The cipher has 12 rounds at its core and each round function consists of the addition of a round-dependent constant and a fixed key, 4×4 = 16 parallel S-boxes and a linear diffusion. The first half of the 128 bit secret key is used as the pre and post-whitening keys, whereas the second half of the key is used directly in the round functions. To perform decryption, the key is first XORed with a fixed value and the same circuit can be re-used for encryption. The overhead of performing decryption is therefore minimized.

PRINCE is the first lightweight block cipher to be optimized with respect to latency (Doroz *et al*., 2014). Previous proposals focus mainly on having a small footprint in hardware. Encryption using PRINCE can be performed in just one clock cycle if an unrolled implementation is deployed.

To achieve encryption and reduce the area requirement, the designers select an optimal S-box and minimize the number of operations in the linear diffusion.

Figure 4 shows the structure of the PRINCE cipher. The components of this cipher are highlighted as follows:

- **Key schedule:** The secret key is 128 bits divided into two halves: k_0 and k_1. The k_0 is used directly as the pre-whitening key. For the post-whitening operation, another key denoted as $k_0' = (k_0>>>1)\oplus(k_0>>63)$ is used, which is a slight modification of k_0

Table 1: AT commands used to operate the GSM modem

AT command	Meaning
AT+CMGR	Read messages
AT+CMGS	Send message
AT+CMGF	Set the message type: Text mode or PDU
AT+CMGD	Delete message

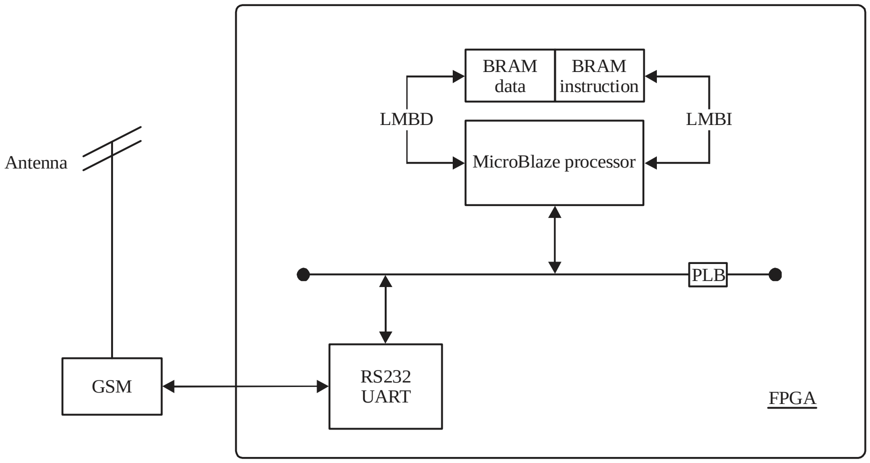


Fig. 2: MicroBlaze processor communicating with GSM

- Pre and post-whitening refer to the addition of key content before and after a core cipher operation. In the case of PRINCE, the core operation is referred to as PRINCE-Core. The k_1 key is used directly in the key addition phase of the round functions R and R^{-1}
- **Round functions R and R^{-1}:** The round functions consist of the following: an XOR with a fixed key k_1, an XOR with a round-dependent constant RC, an S-box layer S (and its inverse S^{-1}) and a linear diffusion M (and its inverse M^{-1})
- **Round-dependent constant:** The constants are defined from the XOR operation $RC_i \oplus RC_{11-i} = \alpha$ for $0 \leq i \leq 1$, with α = coac29b7c97c50dd (in hexadecimal)
- **S-box layer S (and its inverse S^{-1}):** The S-box layer uses a mapping of 4-4 bit, as defined in Table 2. The PRINCE algorithm has 16 active S-boxes in 4 consecutive rounds
- **Linear diffusion layer M (and its inverse M^{-1}):** The linear layer XORs three input bits to produce a single output bit. Each output bit uses different input bits. It is designed to maximize diffusion
- **Middle involution:** This component is composed of the functions SR^{-1}, M' and SR. The functions SR and SR^{-1} are a shift rows operation, whereas M' is a linear diffusion

The middle involution can be seen as a connector for the forward and inverse round functions. This connector affects the encryption such that key k is equal to the decryption with key ($k \oplus \alpha$).

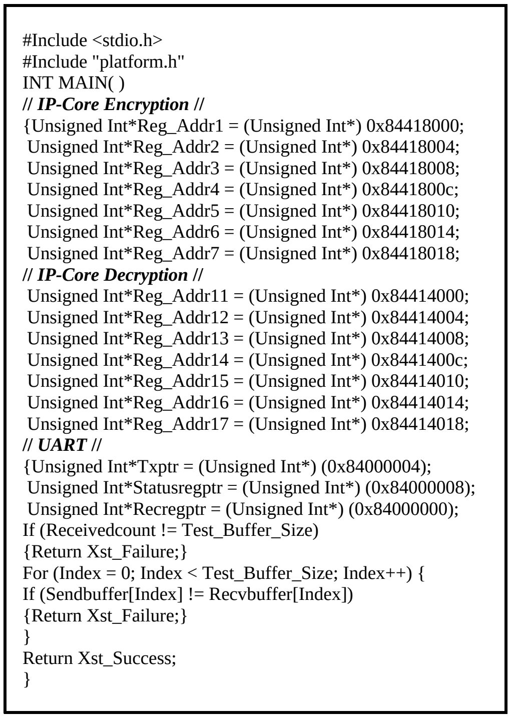
Table 2: S-box and S-box inverse layer of PRINCE

X	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S (X)	B	F	3	2	A	C	9	1	6	7	8	0	E	5	D	4
X	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S^{-1} (X)	B	7	3	2	F	D	8	9	A	6	4	0	5	E	C	1

Source: Borghoff *et al.* (2012)

Implementation of the PRINCE algorithm using VHDL: The main goal of using the proposed PRINCE block cipher algorithm has been described. Therefore, this cipher can be realized for different future broad applications to meet real-time requirements. The IP Core hardware model design has to be completed first; the PRINCE algorithm is designed and implemented on the FPGA. Different stages are involved in System-on-Chips (SoC) design. The stages are VHDL code synthesize, mapping, placing and routing, simulation, IP Core creation and real-time execution. Figure 5 shows the top-level design for the data flow overview.

Figure 6 shows the interface of the block cipher in our design. The design involves three ports: Two ports for the



```
#Include <stdio.h>
#Include "platform.h"
INT MAIN( )
// IP-Core Encryption //
{Unsigned Int*Reg_Addr1 = (Unsigned Int*) 0x84418000;
 Unsigned Int*Reg_Addr2 = (Unsigned Int*) 0x84418004;
 Unsigned Int*Reg_Addr3 = (Unsigned Int*) 0x84418008;
 Unsigned Int*Reg_Addr4 = (Unsigned Int*) 0x8441800c;
 Unsigned Int*Reg_Addr5 = (Unsigned Int*) 0x84418010;
 Unsigned Int*Reg_Addr6 = (Unsigned Int*) 0x84418014;
 Unsigned Int*Reg_Addr7 = (Unsigned Int*) 0x84418018;
// IP-Core Decryption //
 Unsigned Int*Reg_Addr11 = (Unsigned Int*) 0x84414000;
 Unsigned Int*Reg_Addr12 = (Unsigned Int*) 0x84414004;
 Unsigned Int*Reg_Addr13 = (Unsigned Int*) 0x84414008;
 Unsigned Int*Reg_Addr14 = (Unsigned Int*) 0x8441400c;
 Unsigned Int*Reg_Addr15 = (Unsigned Int*) 0x84414010;
 Unsigned Int*Reg_Addr16 = (Unsigned Int*) 0x84414014;
 Unsigned Int*Reg_Addr17 = (Unsigned Int*) 0x84414018;
// UART //
{Unsigned Int*Txptr = (Unsigned Int*) (0x84000004);
 Unsigned Int*Statusregptr = (Unsigned Int*) (0x84000008);
 Unsigned Int*Recregptr = (Unsigned Int*) (0x84000000);
If (Receivedcount != Test_Buffer_Size)
{Return Xst_Failure;}
For (Index = 0; Index < Test_Buffer_Size; Index++) {
If (Sendbuffer[Index] != Recvbuffer[Index])
{Return Xst_Failure;}
}
Return Xst_Success;
}
```

Fig. 3: Code for FPGA and GSM communication

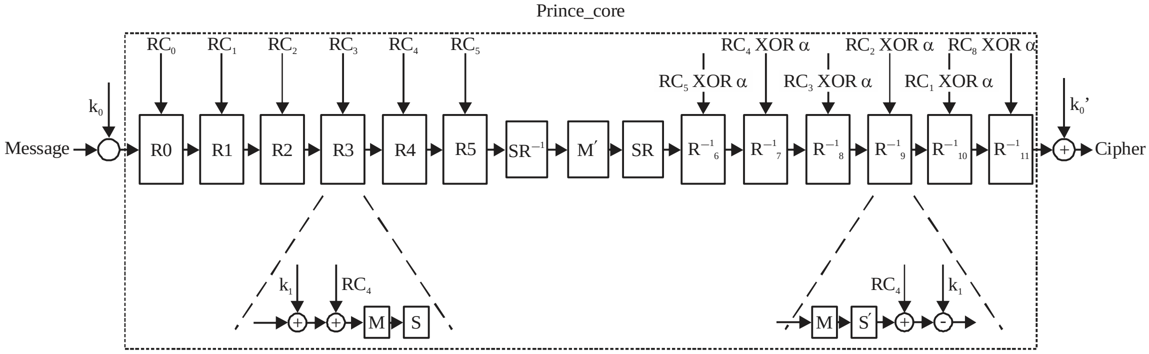


Fig. 4: PRINCE core encryption (Borghoff *et al.*, 2012)

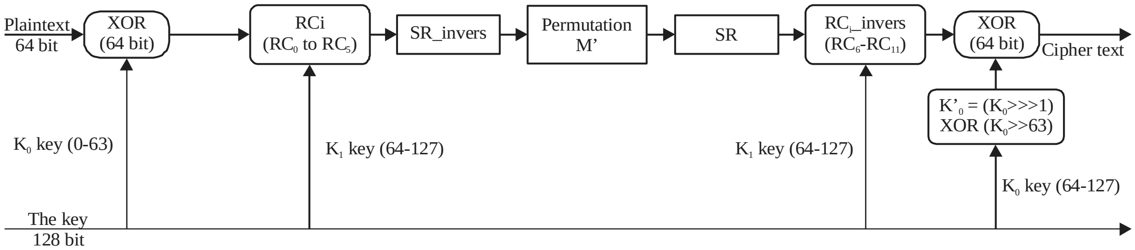


Fig. 5: Data flow of the PRINCE encryption unit

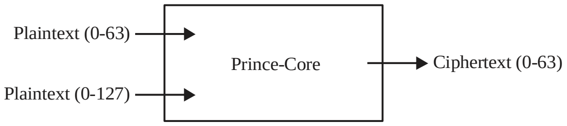


Fig. 6: Top module interface of the PRINCE

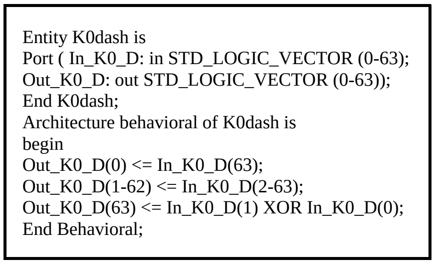
input of the 64 bit plaintext and the 128 bit key and an interface port representing the ciphertext 64 bit output. The shows the top-level design for the data flow overview.

Figure 6 shows the interface of the block cipher in our design. The design involves three ports: Two ports for the input of the 64 bit plaintext and the 128 bit key and an interface port representing the ciphertext 64 bit output. The total numbers of the pins used for input and output is 256 pins. Therefore, large FPGA boards Virtex-403 are selected because they have more input and output pins as well as a large number of logic resources.

The main goal of our design is to achieve low-latency and low-cost hardware implementation. During our model design, the lowest possible gate is investigated to obtain a low-latency hardware component of the PRINCE algorithm. PRINCE Core consisted of many components such as the following: XOR 64-bit, S-box Layer, M layer, M' layer, RC constant and Key schedule. The most expensive operation is the S-box layer. In this design, block RAM is not used; instead, all the components were built using a look-up-table (LUT) only. Therefore, the number of LUT is 32 slices for the Virtex-403 resources.

The design of the key schedule component with 63 right-shift operations typically requires long execution duration and large area. Our approach for this key schedule is to create a custom circuit in the form of a simple wiring route, with the route depending on the position of the input and output bit locations before and after the shift operation. This routing circuit requires only one slice of FPGA area of Virtex-403. Figure 7 shows the VHDL code for the simple wiring route of the key schedule entity or shift operation.

The concurrent hardware design makes the delay time short. This technique allows the block cipher model to encrypt the input data in all hardware components within one clock cycle. The low-area modification in our hardware architecture resulted in good performance in terms of maximum frequency, throughput and occupied slices.



```
Entity K0dash is
Port ( In_K0_D: in STD_LOGIC_VECTOR (0-63);
Out_K0_D: out STD_LOGIC_VECTOR (0-63));
End K0dash;
Architecture behavioral of K0dash is
begin
Out_K0_D(0) <= In_K0_D(63);
Out_K0_D(1-62) <= In_K0_D(2-63);
Out_K0_D(63) <= In_K0_D(1) XOR In_K0_D(0);
End Behavioral;
```

Fig. 7: Key schedule to find K0'

The PRINCE decryption unit is almost similar to the encryption design. The data flow of decryption is shown in Fig. 8. The decryption unit requires the last round key in the encryption unit to be the input key for the first round of decryption. These decryption keys are generated by the XOR operation: Key (K$_1$) with α or RC$_{11}$ to obtain the new key (K$_1$) fed into the decryption unit.

Design of PRINCE IP Core with GSM modem: The PRINCE IP Core and the microprocessor are designed and tested using ISE and XPS, respectively. The MicroBlaze is a soft-core processor that provides flexibility to the designer. The processor system is built with the selected components that reduce the hardware area. The components of the embedded system are selected as follows: MicroBlaze as a software processor, BRAM as a processor memory and RS-232/UART as the peripherals interface. The BRAM is divided into two parts: The first is for the transfer instruction, whereas the second is for the transfer data using Local Memory Bus Instruction (LMBI) and Local Memory Bus Data (LMBD), respectively. In addition, the RS-232/UART is used to connect the FPGA to the outside devices.

The proposed IP Core is connected to the microprocessor using the XPS to create a real-time embedded system. The

IP Core will connect to the other peripherals and MicroBlaze using Processor Local Bus (PLB). The IP Core is designed to be a slave processor in this model. The secure IP Core that has been created is responsible for the encryption and decryption of the data. Every message will be encrypted by the IP Core before being sent to the GSM. Similarly, the received message will be decrypted by the IP Core before being passed to the GSM. Basically, the IP Core will secure the communication system between the electrical substations. Figure 9 shows the structural design of the two secure IP Cores connected to MicroBlaze, GSM and other peripherals.

In developing and testing the IP Core, the following steps are adopted. First, the ISE tools are used to develop the PRINCE encryption and decryption entities. Second, simulation tools have been used to test both entities and subsequently XPS has been used to instantiate a "New slave-type" IP Core, which is then added to the encryption and decryption entities (ISE) developed earlier. Third, the IP Core with the attached slave interface has been tested to verify its syntax, synthesized into a bit stream and then finally downloaded to an FPGA chip. Figure 10 shows the RTL schematic for PRINCE IP-Core with ISE.

Figure 11 shows XPS with IP Cores of encryption and decryption of PRINCE, MicroBlaze, memory and UART that are interconnected through the system bus (processor local bus). Except for MicroBlaze, each IP Core has its own addresses of a 32 bit system memory map. The XPS enables the complete microprocessor system to be synthesized and implemented to produce a bit stream that can be downloaded to the FPGA. This on-chip microprocessor design is then transferred to another tool called SDK to prepare for the test suite. The test suite is written in C. A PRINCE Lightweight Crypto-Algorithm was used to design hardware IP Core for increasing the security in communication devices. The concurrent design model with a low design area produces high performance in terms of throughput and frequency with low-power consumption. The embedded system proposed in this study can be implemented in any portable communication device because of its low-power consumption.

The steps required within the SDK consist of hardware and software phases. First, the hardware design is downloaded to the FPGA platform. Second, a test program written in C or C++ is prepared. The function of a simple C program is to assign the value of the 64-bit plaintext and the

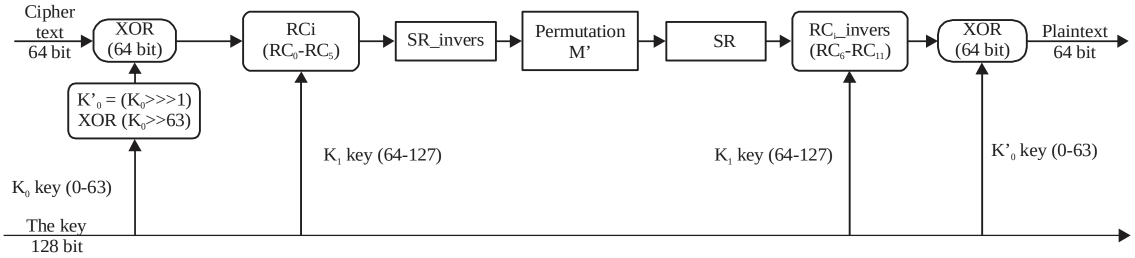


Fig. 8: Data flow of the PRINCE decryption unit

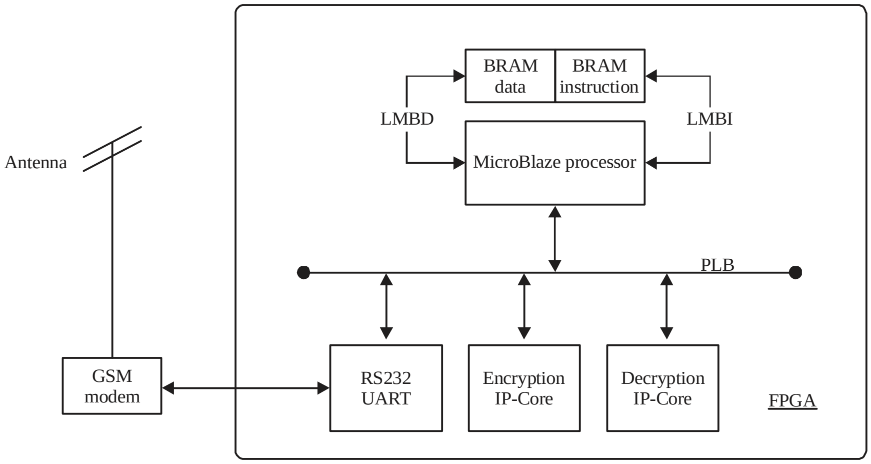


Fig. 9: PRINCE algorithm SoC communicating with GSM modem

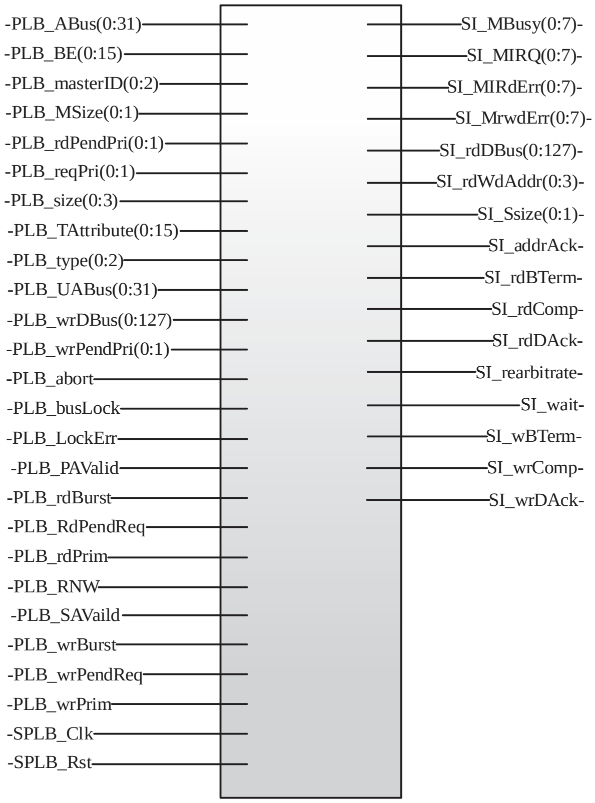


Fig. 10: RTL schematic for PRINCE IP-Core

128 bit key, therefore six 32 bit registers have been used. The result obtained from the cipher operation of the IP Core requires another two 32 bit registers as ciphertext. In addition, the real-time execution using this kind of hardware and software set-up has been performed for this design.

RESULTS AND DISCUSSION

Simulation analysis performed for both of the encryption and decryption functions have been coded in VHDL targeted for the XILINX Virtex-ML403 XC4VFX12 (Package FF668 with speed grade-10) FPGA. The Xilinx ISE V14.5 WebPack and ModelSimXE P.58f were used as synthesis and simulation tools. 64-bit hexadecimal plaintexts (0x0000000000000000 and 0xFFFFFFFFFFFFFFFF) and 128 bit key (0x000…..0000, 0xFF…. 000, 0x000…. FFF) in addition to other texts are fed into the hardware model for the simulation test vector. The ciphertext 64 bit output is produced in one clock cycle. For the encryption function, Fig. 12 shows the test vector for some inputs and outputs in a simulation graph. The decryption function is shown in the Fig. 13.

The hardware IP Core design of the encryption and decryption model has low-power consumption because it has a small area; the number of occupied slices used with Virtex-403 is 956. The total power for the proposed designed is 0.16 W. The securing IP Core for electrical substation wireless messaging was proposed to overcome the weakness of GSM communication with low-power consumption architecture. The system design proposed yields more secure and high-throughput data communications. Figure 14 shows the Virtex-403 board connected through UART to the GSM modem; the secure SMS data is sent to a mobile phone.

The performance of the FPGA system designed for PRINCE IP Cores is also evaluated. The goals of these proposed approaches have been implemented for optimizing the area and power consumption or maximizing the throughput and comparing it with some related works. The proposed PRINCE IP core improves the security of remote assets in a GSM wireless communication system which is in fact not considered in previous research works (Peijiang and Xuehua, 2008; Li *et al*., 2010; Ahmad *et al*., 2011). The proposed design has decreased the slices number to 956 for encryption or decryption compare with the work in (Misal *et al*., 2014) which has a slices number of 4162 and 2722 for the encryption and decryption respectively. Moreover, The proposed PRINCE IP core achieves a throughput of 2.032 Gbps

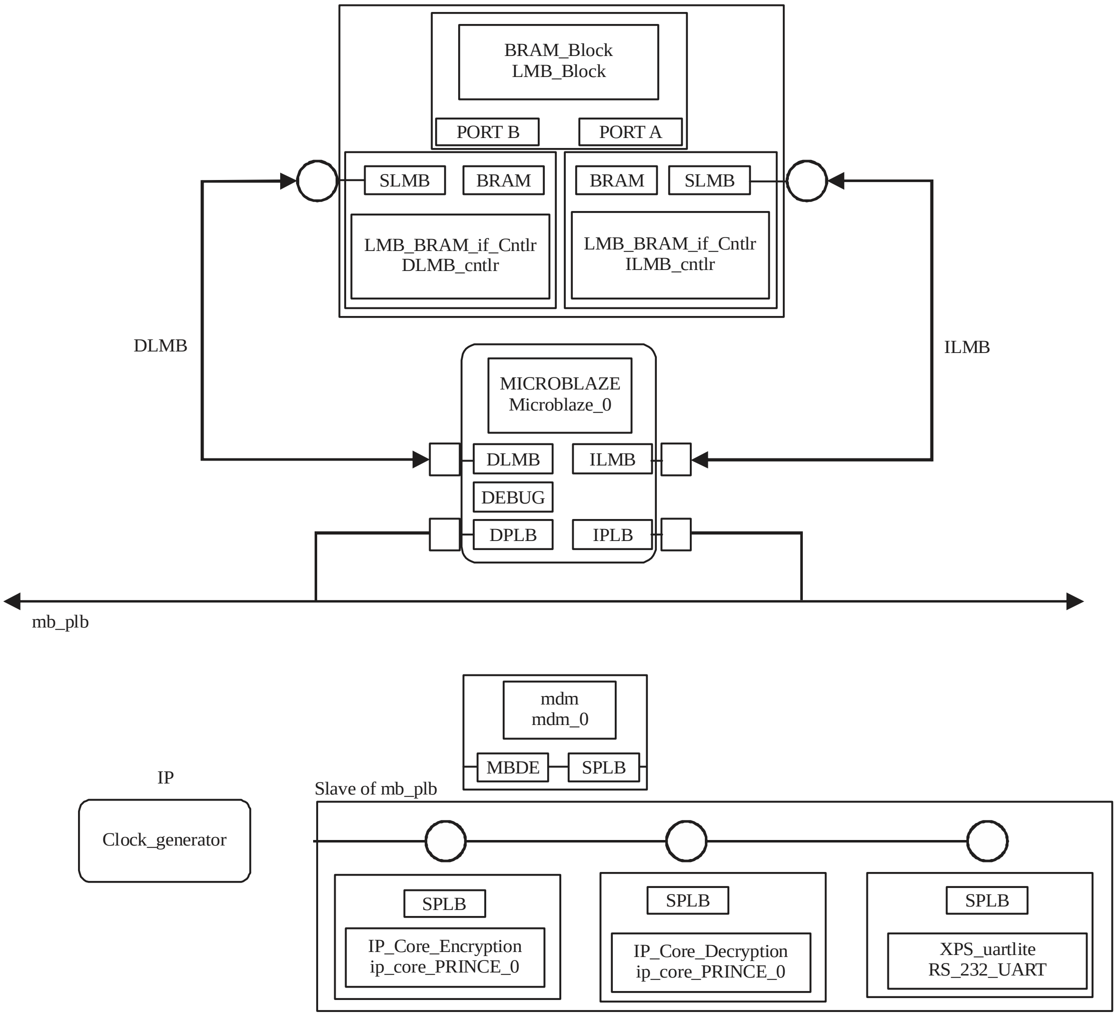
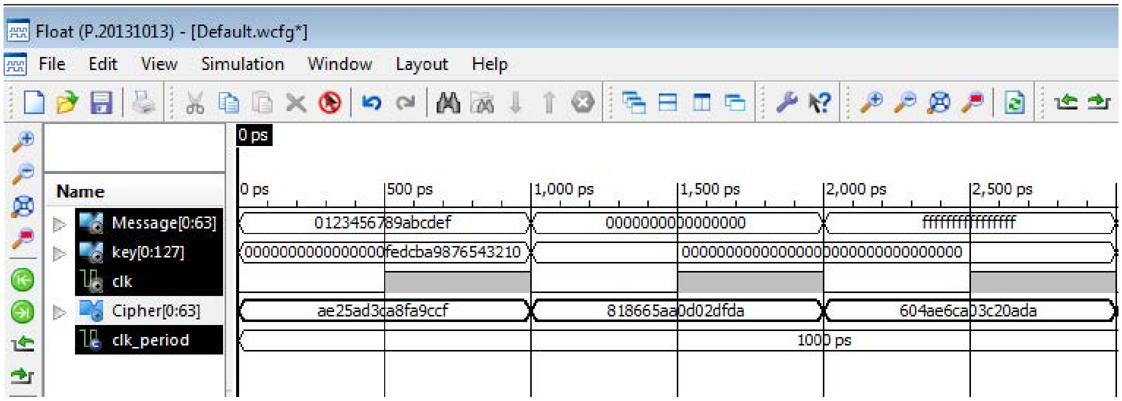


Fig. 11: MicroBalze with IP-Core



```
Message[0:63]   0123456789abcdef                   0000000000000000   ffffffffffffffff
key[0:127]      0000000000000000fedcba9876543210   00000000000000000000000000000000
clk
Cipher[0:63]    ae25ad3ca8fa9ccf                   818665aa0d02dfda   604ae6ca03c20ada
clk_period      1000 ps
```

Fig. 12: Encryption for the PRINCE Lightweight Algorithm

and efficiency of 2.126 Mbps per slices wears the throughput of 795 Mbps and efficiency of 0.94 Mbps per slices are obtained in (Kuo *et al*., 2005).

Figure 15 shows the real-time test of PRINCE IP Core with the test suite program executing on MicroBlaze processor and with the test result of the encryption displayed on a

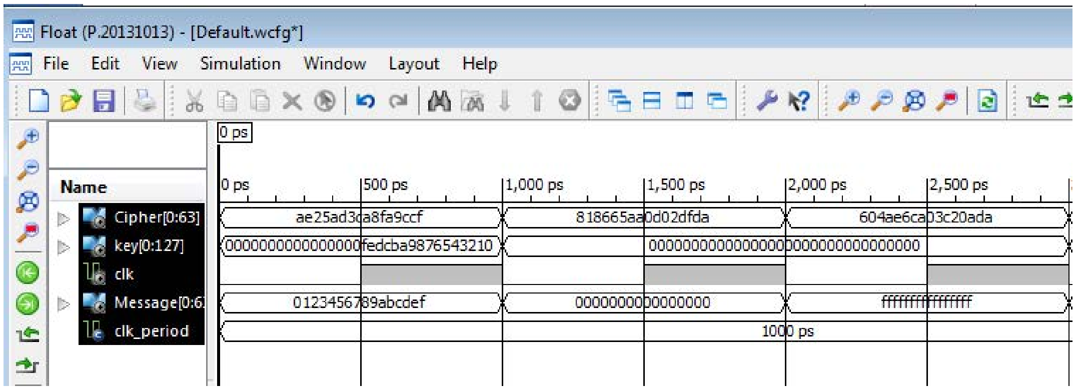


Fig. 13: Decryption for the PRINCE lightweight algorithm

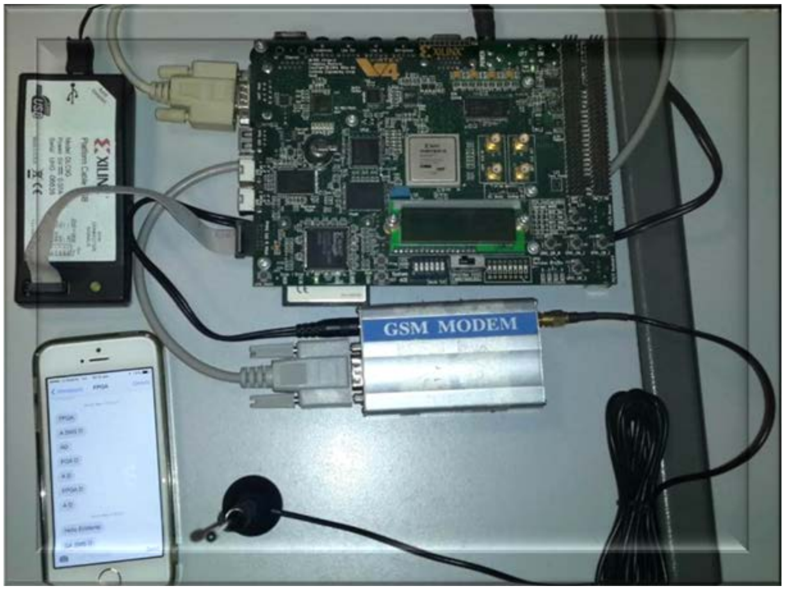
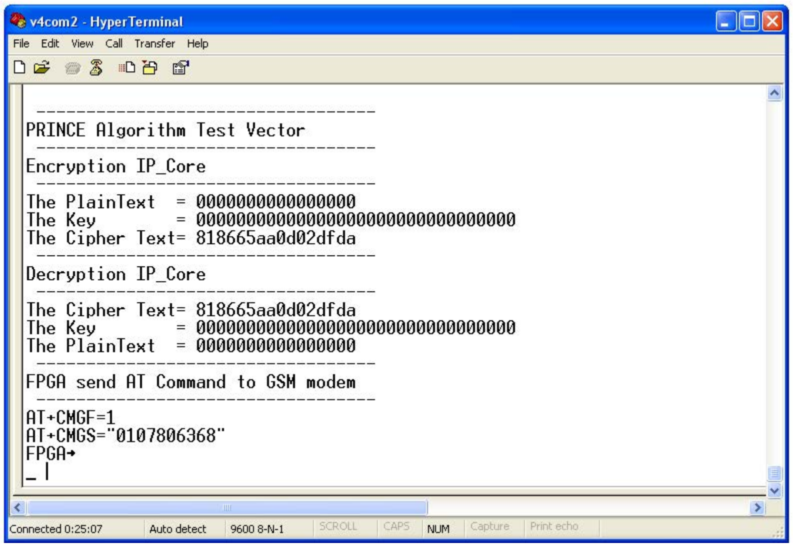


Fig. 14: FPGA board with GSM and mobile phone



```
 -----------------------------------
PRINCE Algorithm Test Vector
 -----------------------------------
Encryption IP_Core
 -----------------------------------
The PlainText  = 0000000000000000
The Key        = 00000000000000000000000000000000
The Cipher Text= 818665aa0d02dfda
 -----------------------------------
Decryption IP_Core
 -----------------------------------
The Cipher Text= 818665aa0d02dfda
The Key        = 00000000000000000000000000000000
The PlainText  = 0000000000000000
 -----------------------------------
FPGA send AT Command to GSM modem
 -----------------------------------
AT+CMGF=1
AT+CMGS="0107806368"
FPGA→
_ |
```

Fig. 15: Real-time Hardware and Software execution

Table 3: Comparison the proposed PRINCE IP core with existing FPGA Implementations

Algorithm	FPGA Boards	Block size	Clock cycle	Max freq. (MHz)	Thr/put (Mbps)	Total slices	Efficiency (Mbps per Slice)
Proposed design IP Core (Encrypt or decrypt)	Virtex-403	64	1	31.76	2032	956	2.126
AES IP Core encrypt (Kuo *et al*., 2005)	Spartan -2 XCS200-5	128	12	74.6	795.72	842	0.945
AES IP Core encrypt (Kuo *et al*., 2005)	Spartan -2 XCS200-5	128	17	67.9	511.09	854	0.519
AES IP Core decrypt (Kuo *et al*., 2005)	Spartan -2 XCS200-5	128	12	65.9	702.58	1068	0.658
AES IP Core decrypt (Kuo *et al*., 2005)	Spartan -2 XCS200-5	128	17	61.9	466.16	1150	0.405
IP Core encrypt (Misal *et al*., 2014)	-	128	1	95.09	-	4162	-
IP Core decrypt (Misal *et al*., 2014)	-	128	1	96.25	-	2772	-

hyper-terminal. The hyper-terminal has been used in this case as display or human interface and executes on a PC connected to the on-chip UART through the serial port. The hyper-terminal displays test vectors with ciphertext generated by the PRINCE IP Core as device on test. The encryption data is sent from the FPGA using the GSM modem to the mobile phone. PRINCE IP Core architecture has been designed to encrypt or decrypt the input data within one clock cycle while maintaining low latency and high speed. The investigations of PRINCE IP Core with Virtex-403 FPGA board indicates a high speed of 31.76 MHz, throughput of 2.032 Gbps, low-power consumption 0.165 W and efficiency of 2.126 Mbps per slice. Table 3 presents a comparison of the PRINCE IP Core implemented on FPGA and some related works.

CONCLUSION

This study proposes a PRINCE light-weight crypto algorithm in the form of hardware circuits called IP core. The hardware architecture was designed to encrypt and decrypt the data within one clock cycle while maintaining high throughput, low power and high efficiency as compared with those in previously cited works. This low-power design is intended to be implemented in a GSM or any other portable communication device. The development of this model is not costly and it can overcome the weakness of communications in electrical substations. Our module successfully transmitted and received the encryption data from the FPGA board to mobile phone through a GSM modem.

The results showed that the design of IP Core inside the microprocessor has a low-power consumption of 0.165 W and 956 slices were used for every IP Core. Moreover, the maximum frequency is 31.76 MHz, throughput is 2.3 Gbps and efficiency is 2.1 Mbps per slice for the encryption or decryption module. This proposed hardware was implemented on Virtex-4 ML403 FPGA board. The number of slices for the whole system design consists of MicroBlaze processor, BRAM, encryption IP Core, decryption IP Core, UART and all interfaces are 4,017 slices.

ACKNOWLEDGMENT

This study is supported by Ministry of Education, Malaysia under Fundamental and Prototype Research Grant Scheme Phase 1/2013.

REFERENCES

Abbas, Y.A., R. Jidin, N. Jami and M.R. Z'aba, 2014a. Securing electrical substation's wireless messaging with a Lightweight Crypto-Algorithm IP core. Proceedings of the IEEE International Conference on Power and Energy, December 1-3, 2014, Kuching, pp: 159-163.

Abbas, Y.A., R. Jidin, N. Jamil, M.R. Z'aba, M.E. Rusli and B. Tariq, 2014b. Implementation of PRINCE algorithm in FPGA. Proceedings of the 6th International Conference on Information Technology and Multimedia, November 18-20, 2014, Putrajaya, pp: 1-4.

Ahmad, A.W., N. Jan, S. Iqbal and C. Lee, 2011. Implementation of ZigBee-GSM based home security monitoring and remote control system. Proceedings of 54th IEEE International Symposium on Circuits and Systems, August 7-10, 2011, Seoul, pp: 1-4.

Ahmad, M., 2009. Enhanced A5/1 cipher with improved linear complexity. Proceedings of the International Conference on Multimedia Signal Processing and Communication Technologies, March 14-16, 2009, Aligarh, pp: 265-267.

Al-Rousan, M., A.R. Al-Ali and K. Darwish, 2004. GSM-based mobile Tele-monitoring and management system for inter-cities public transportations. Proceedings of the IEEE International Conference on Industrial Technology, Volume 2, December 8-10, 2004, United Arab Emirates, pp: 859-862.

Baldini, G., I.N. Fovino, M. Masera, M. Luise and V. Pellegrini *et al*., 2010. An early warning system for detecting GSM-R wireless interference in the high-speed railway infrastructure. Int. J. Crit. Infrastruct. Protect., 3: 140-156.

Borghoff, J., A. Canteaut, T. Guneysu, E.B. Kavun and M. Knezevic *et al*., 2012. PRINCE-A Low-Latency Block Cipher for Pervasive Computing Applications. In: Advances in Cryptology-ASIACRYPT, Wang, X. and K. Sako (Eds.). Springer, New York, ISBN: 9783642349614, pp: 208-225.

Cleveland, F., 2006. Use of wireless data communiicatiions in power system operations. Proceedings of the Power Systems Conference and Exposition, October 29-November 1, 2006, Atlanta, GA., USA., pp: 631-640.

Doroz, Y., A. Shahverdi, T. Eisenbarth and B. Sunar, 2014. Toward Practical Homomorphic Evaluation of Block Ciphers Using Prince. In: Financial Cryptography and Data Security, Bohme, R., M. Brenner, T. Moore and M. Smith (Eds.). Springer, New York, ISBN: 9783662447741, pp: 208-220.

El-Medany, W.M. and M.R. El-Sabry, 2008. GSM-based remote sensing and control system using FPGA. Proceedings of the International Conference on Computer and Communication Engineering, May 13-15, 2008, Kuala Lumpur, pp: 1093-1097.

Firdaus, W. and H.J. Yaakob, 2009. Synchronization system for crypto initialization over GSM voice channel. Proceedings of the 9th WSEAS International Conference on Multimedia, Internet and Video Technologies, September 3-5, 2009, Budapest, Hungary, pp: 241-244.

Gaikwad, P.K., 2013. Development of FPGA and GSM based advanced digital locker system. Int. J. Comput. Sci. Mobile Applic., 1: 18-23.

Gaushell, D.J. and W.R. Block, 1993. SCADA communication techniques and standards. IEEE Comput. Applic. Power, 6: 45-50.

Gonzalez-Castano, F.J., J. Vales-Alonso, J.M. Pousada-Carballo, F.I. de Vicente and M.J. Fernandez-Iglesias, 2002. Real-time interception systems for the GSM protocol. IEEE Trans. Veh. Technol., 51: 904-914.

Gupta, S.S., A. Chattopadhyay, K. Sinha, S. Maitra and B.P. Sinha, 2013. High-performance hardware implementation for RC4 stream cipher. IEEE Trans. Comput., 62: 730-743.

Hong, S. and M. Lee, 2010. Challenges and direction toward secure communication in the SCADA system. Proceedings of the 8th Annual Communication Networks and Services Research Conference, May 11-14, 2010, Montreal, QC., Canada, pp: 381-386.

Hong, S., M. Lee and D.Y. Shin, 2010. Experiments for embedded protection device for secure SCADA communication. Proceedings of the Asia-Pacific Power and Energy Engineering Conference, March 28-31, 2010, Chengdu, pp: 1-4.

Igure, V.M., S.A. Laughter and R.D. Williams, 2006. Security issues in SCADA networks. Comput. Secur., 25: 498-506.

Khan, M.R.B., R. Jidin, J. Pasupuleti and S.A. Shaaya, 2015. Optimal combination of solar, wind, micro-hydro and diesel systems based on actual seasonal load profiles for a resort island in the South China Sea. Energy, 82: 80-97.

Kuo, F.H., S.T. Yen and C.C. Liu, 2005. A multi-FPGA rapid prototyping system with the reusable AES core. Inform. Technol. J., 4: 262-270.

Li, X., Q. Yuan, W. Wu, X. Peng and L. Hou, 2010. Implementation of GSM SMS remote control system based on FPGA. Proceedings of the 2nd International Conference on Information Science and Engineering, December 4-6, 2010, Hangzhou, China, pp: 2132-2135.

Misal, P., M. Karule, D. Birdawade, A. Deshmukh and M. Pathak, 2014. Door locking/unlocking system using SMS technology with GSM/GPRS services. Int. J. Electron. Commun. Comput. Eng., 5: 192-194.

Ozkan, M.A., B. Ors and G. Saldamli, 2011. Secure voice communication via GSM network. Proceedings of the 7th International Conference on Electrical and Electronics Engineering, December 1-4, 2011, Bursa, pp: II-288-II-292.

Parikh, P.P., M.G. Kanabar and T.S. Sidhu, 2010. Opportunities and challenges of wireless communication technologies for smart grid applications. Proceedings of the Power and Energy Society General Meeting, July 25-29, 2010, Minneapolis, MN., pp: 1-7.

Peijiang, C. and J. Xuehua, 2008. Design and implementation of remote monitoring system based on GSM. Proceedings of the Pacific-Asia Workshop on Computational Intelligence and Industrial Application, Volume 1, December 19-20, 2008, Wuhan, pp: 678-681.

Thomas, M.S., P. Kumar and V.K. Chandna, 2004. Design, development and commissioning of a Supervisory Control and Data Acquisition (SCADA) laboratory for research and training. IEEE Trans. Power Syst., 19: 1582-1588.

Toorani, M. and A. Beheshti, 2008. Solutions to the GSM security weaknesses. Proceedings of the 2nd International Conference on Next Generation Mobile Applications, Services and Technologies, September 16-19, 2008, Cardiff, pp: 576-581.

Vrentzos, E., G. Kostopoulos and O. Koufopavlou, 2006. Hardware implementation of the A5/3 & A5/4 GSM encryption algorithms. Proceedings of the World Automation Congress, July 24-26, 2006, Budapest, pp: 1-6.

Xilinx Inc., 2013. Embedded system tools reference manual: EDK. UG111 (v14.5), March 20, 2013. http://class.ece.iastate.edu/cpre488/resources/est_rm.pdf.