Securing Space based Cyber-Physical System through a Reconfigurable Framework

Arshad Riazuddin and Shoab A Khan

***Abstract*— Cyber-physical systems (CPS) built on a Rich operating systems with varied applications running on it are the norm today. In order to provide a rich experience, the applications running on the CPS would want access to the myriad of hardware interfaces which are architected into them. Security of the embedded cyber-physical system is compromised by this limitless access requested by the applications to various hardware interfaces on the device as well as user data. Recognizing this problem, diverse solutions have been proposed on different topologies, by using either a standards based approach or not. In this paper, we present a technique that proposes HW/SW architecture for securing an embedded cyber physical system using the concepts of RED-BLACK separation. While the implementation of RED-BLACK separation is introduced in military communication devices, this concept has not yet taken hold in other high-end embedded cyber physical systems. The proposed architecture can be easily adopted across a broad spectrum of platforms linked to CPS, but not limited to, such as communication, space systems, medical devices, energy conservation etc. The results show that the proposed framework was successfully validated on a working system and achieved on the fly configurability as desired by the proposed architecture.**

***Index Terms*— RED-BLACK Separation, Cyber-Physical System (CPS), Software Defined Radio (SDR), Mobile.**

I. Introduction

"CYBER-Physical Systems (CPS) are integrations of computation with physical processes. Embedded computers and networks monitor and control the physical processes, usually with feedback loops where physical processes affect computations and vice versa." [1]. The applications of CPS encompass: space systems, energy conservation, smart grid energy metering, high confidence medical devices, autonomous networked vehicles, smart structures, communication, telemedicine etc. [1]. A prime example of cyber physical systems are space vehicles and systems which work autonomously, but still present challenges associated with communication, orbit determination and control, and payload management [2]. Distributed decision making about available bandwidth and control of the network will benefit cognitive radio in communication systems [1].

Manuscript received February 20, 2019; revised March 14, 2019.

Arshad Riazuddin is with Center for Advanced Research in Engineering (CASE), Islamabad, Pakistan (e-mail: Riazuddin.arshad@gmail.com).

Shoab A. Khan is with Computer Engineering Department, College of EME, National University of Sciences and Technology, Rawalpindi, Pakistan (e-mail: shoab@carepvtltd.com).

These CPS devices are connected to various other entities through multiple OTA interfaces, sharing information in a network centric environment. In order to provide networked services, there are many applications running on the embedded device which interface through the various HMI and OTA interfaces and have wide access to these components. The transmissions on OTA interfaces are usually protected by security measures as defined by their respective standards, and contain both computer level security (COMPSEC) and transmission level security (TRANSEC). The GSM cellular transmissions are protected by using encryption algorithm as well as frequency hopping on transmissions between the cell phone and base transceiver station (BTS). The CDMA network protects over the air transmission using encryption and direct sequence spread spectrum (DSSS) technology. Bluetooth technology also employs its own encryption algorithm and frequency hopping spread spectrum (FHSS) technology for protecting over the air transmissions. While Wi-Fi employs OFDM and direct sequence spread spectrum (DSSS) technology for protecting over the air transmissions. GPS technology also employs encryption spread spectrum (DSSS) technology for protecting over the air transmissions. This unhindered access to the interfaces, makes them vulnerable to manipulation from unauthorized entities.

Security for cyber-physical systems is becoming an issue, as demonstrated by the Stuxnet worm attack [3]. CPS devices use standard telecommunication networks to communicate over the long haul, and reports of various security weaknesses present in telecommunication architectures and other embedded systems are well published. The encryption cipher, which is used in GSM communication to secure wireless connectivity, can be intercepted and decrypted using a publicly available algorithm to discover the encryption key [4]. Smartphone applications can perform malicious actions on mobile handsets due to the security weaknesses present in them [5]. Using spoofed base station [6], Global System for Mobile (GSM) calls can be intercepted and outbound calls can be rerouted. The authors in [7] developed a testbed to perform Cyber vulnerability analysis. The testbed, which realistically represented a communication infrastructure using a mixture of advanced networking tools and logging analysis techniques, was used to perform system level testing to reveal attack characteristics. The subsystems included in this testbed were 2G cellular with GPRS packet data capability, supporting GSM/GPRS base station connectivity, and multiple IMS infrastructure for application server connectivity. Civilian GPS receivers are trivial to spoof, and this has been documented by several researchers. This can be done by using a GPS signal

simulator, and attaching a power amplifier and antenna to it [8]. The authors in [9] developed a software defined receiver snooper to mount attacks on GPS, and concluded that "nothing short of cryptographic authentication can guard against a sophisticated spoofing attack." Irza Et al. in [10] propose a many core processor system using the RED-BLACK separation concept for embedded systems. Keeping in view the diverse nature of embedded devices, the trusted computing group (TCG) in [11] highlights the need for protection from such vulnerabilities. Applications running on embedded devices can be secured using hardware virtualization extensions [12].

The Joint Tactical Radio Software Program (JTRS) was launched by the US government with the objective of standardizing the architecture of Software Defined Radio (SDR) [13]. The architecture was named Software Communication Architecture (SCA) [13].The SCA was developed for both military and commercial purposes, but the main aim was for military application. A security supplement to the SCA [14] was published by the US government with the purpose of securing the SDR by introducing a new set of interfaces and specifications. The SCA divides the radio into two parts, red (confidential data) and black (un-confidential or encrypted data), with both functional and electrical isolation between them. The CSS is responsible for the encryption and authentication of data between the red and black interfaces.

In this paper we present a RED-BLACK separation framework for securing digital content stored and processed in embedded devices. The framework introduces a reconfigurable external interface which can be used to prevent side channel attacks by reconfiguring the interface to behave differently at a variable rate.

This paper is organized as follows. Section I introduces the RED-BLACK separation and need to extend it to other embedded devices. In Section II, various techniques that have been used to implement security in embedded devices are presented. Section III describes the proposed framework. Section IV presents the implementation of the framework in a software defined radio (SDR) application along with results of the implementation. Section V presents a comparison and analysis with other relevant solutions. Section 6 concludes this paper.

II. RELATED WORK

Security vulnerabilities of cyber physical systems are well recognized and various alternatives for securing embedded devices from these unwanted intrusions have been proposed.

The article [15] describes a design flow and verification process which the National Security Agency (NSA) and Xilinx have developed for satisfying the NSA requirements for a high-grade cryptographic. The flow implements a *fence*, where no routing or logic may be present, using the partial reconfiguration design flow and Xilinx toolset. Using a new design constraint called NOBOUNDARYCROSS, the place and route tool can be instructed to keep the routing within a designated area. The place and route tools were modified so that the IO cells of the FPGA can also be kept in their respective isolation areas. A special tool was developed by Xilinx for the NSA, Isolation verification tool (IVT), for isolation verification. In this flow, the final placed and routed design and the separate NCD file for each isolated region are merged. The isolation regions are then verified using IVT, as all the functionality in each isolated region is known by the tool. The Virtex-4 FPGA device was thoroughly analyzed from a security perspective by performing various tests by the NSA. In addition, to meet the high security requirements of the US government a security monitor was developed and implemented within the Virtex-4 fabric. These tools and devices are not publicly available and can only be used by specialized agencies of the US government.

The authors in [16] present a reconfigurable crypto subsystem (CSS) which is in line with the Secure Software Communication Architecture (SSCA) proposed for SDR with in the overall Joint Tactical Radio software program (JTRS). The proposed CSS architecture consists of a communication block (CommB) and a Control Block (ControlB). The CommB is implemented using 32-bit MicroBlaze processor from Xilinx with a hardware accelerator (DRCA) to implement parts of various cryptographic algorithms. A second MicroBlaze processor is required for the control block (ControlB). Several peripherals, such as a true random generator for key generation, UART to fill data from Red side, and a secure ICAP block for dynamic re-configurability.

In a similar manner ARM has proposed the TrustZone® technology for security on smart devices [17]. TrustZone combines the hardware architecture of the ARM processors and AMBA bus with Trusted Execution Environment (TEE) software to form a system wide methodology. This methodology creates additional modes of operations in addition to normal mode: secure domain, and monitor mode. A secure area is created within the processor of the smart device, and TEE safeguards the secure storage and processing of sensitive data through hardware isolation.

Implementation of a Single chip cryptographic (SCC) systems is explored in [18] by Fitzgerald Et al. in a secure Altera FPGA. The paper describes the implementation of a RED-BLACK separated reference implementation in which one NIOS-II processor is used for RED processing, and another NIOS-II processor is used for BLACK processing. The RED and BLACK cores are connected through two AES-GCM hardware implementations. The AES-GCM mode has been used so that data can be authenticated as well as encrypted. The unencrypted data from the RED processor is sent over a serial routing interface (SRI) to the AES core where it is encrypted and forwarded to the BLACK processor over another SRI; while encrypted data from the BLACK processor is sent over a SRI to the second AES core for decryption. The decrypted data is sent to the RED processor over a SRI for processing or off chip communication. The implementation also includes a fence of unused logic area around the secured partition area for isolation purposes.

This paper [19] describes the architecture of a reconfigurable

multi-core crypto-processor for multi-channel communication systems (MCCP). The proposed design supports CTR, CBC-MAC, CCM and GCM block cipher modes which are used in communication systems. Low level primitives used in cryptography are implemented in hardware while higher level cipher modes are implemented in software. The number of cores implemented in a MCCP can be varied. The MCCP when used in a SDR communication system is embedded in a design consisting of a main controller and a communication controller, and is used as a red/black boundary. The main focus of the paper is on re-configurability of the crypto-processor and not on the framework itself.

In [20] the author proposes the architecture for a Red-Black crypto sub-system block for implementation in a ASIC/FPGA, which reduces the dependency on software to insure the proper use of the bypass block in the crypto subsystem. The proposed architecture has a crypto access controller which lies in the red world, and acts as a bridge between the red and black world. The access controller is connected to a bus fabric on chip through which multiple peripherals can move data using a central DMA controller. A bypass controller is implemented so that the data can be bypassed from the crypto controller and passed to the black world in an unencrypted manner. The decision on whether to bypass the crypto controller or not is provided by the higher level entity in the overall system. The solution presented lacks any option for configurability and is a hardwired solution.

The solutions proposed in [16], [18] and our solution are different implementations of the CSS architecture proposed in [14], but our solution presents a novel approach to security and configurability by introducing a reconfigurable 'on the fly' cryptographic interface (CI). The focus in [16], [18] is also on the implementation of cryptographic algorithms, while our main focus is on a reconfigurable framework, and we are showing it by implementing cryptographic algorithms proposed in [21], [22]. The solution proposed in [15] is a highly restrictive cryptographic implementation using specialized tools and not open for researchers, while our solution does not rely on any such specialized infrastructure. While the solution proposed in [20] lacks any option for configurability and is a hardwired solution.

III. PROPOSED HW/SW ARCHITECTURE

Embedded devices used in cyber physical systems today, based on their network centric usage, have become vulnerable to security lapses. Such devices in high-end applications may have multiple radios, processors, DSP and dedicated processing elements implemented in them to support the myriad needs of these devices: Wi-Fi, Bluetooth, 3G/LTE, complex signal processing algorithms, ISP, graphics codecs, and multiple displays. The problem identified by us has been addressed in different ways by the authors in [12], [16], [18] and the advantages/disadvantages of these methods have been discussed in section 2. Our solution proposes a generic framework based on RED-BLACK separation and providing protection from side channel attacks which can be implemented to secure such devices. The concept of Software Defined Radio (SDR) has been developed around the use of multiple processing components, whether it is general purpose processors (GPP), digital signal processor (DSP), dedicated ASIC or programmable FPGA to meet the processing requirements of the waveform [23]. The same concept of multiple processing elements can be seen in today's cell phones, and multimedia devices. This leads us to partition a cyber-physical embedded device into 3 distinctive architectural areas: network interface (RF), general purpose and modem areas as in a SDR, and thus a RED-BLACK separation can be implemented between the modem side and the general purpose processing elements. In a similar manner, a space system is a cyber-physical embedded device which can be partitioned into a network interface (microwave), compute (cyber) engine (BLACK) and a physical interface (RED).The proposed framework through its configurability and 'on the fly' interface configurability can be integrated into many kinds of devices.

Following the terminology of [14] we call our interface module a Crypto sub system, CSS, block. The CSS block functional architecture as proposed in [14] is shown in Fig. 1.

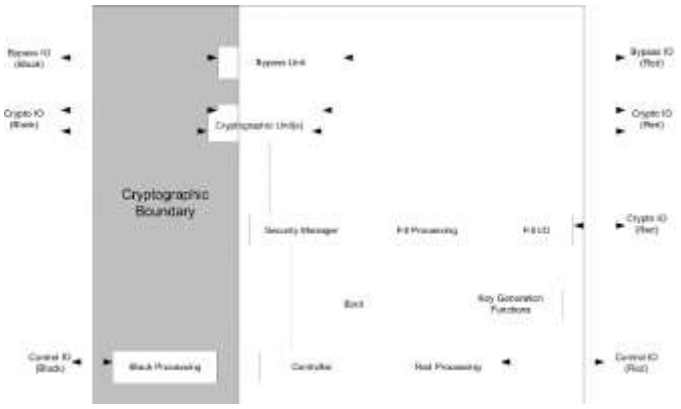

Fig. 1. Cryptographic System Boundary

The primary function of the CSS block is to provide secure communication channel between Red side and Black side. The functionality enclosed within the CSS block as proposed in [14] is the following:

1) Keys and algorithm management
2) Security policy enforcement
3) Bypass unit
4) Message authentication
5) Integrity checking
6) Encryption/Decryption of data

A. Framework

In our proposed framework we will call the Red-Black path as forward path, and the Black-Red path as reverse path. The overall framework consists of

1) Cryptographic Interface (CI)
2) Framework Logic (FL)
3) User Processing Elements (UPE)

B. Cryptographic Interface

Separate bidirectional cryptographic interface (CI) will be used to transport the data and control information from the forward and reverse paths through the CSS block. The cryptographic interface is a serial interface in which packets coming into the interface block are made up of header and data blocks. The header packet contains routing information and travels as clear data (plaintext) in both forward and reverse directions; while the data packets are either encrypted or in clear mode depending on the header packet contents. The CI block in the CSS module decodes the header packet and routes the data packet as per the header requirements. The frame structure is shown in TABLE I .

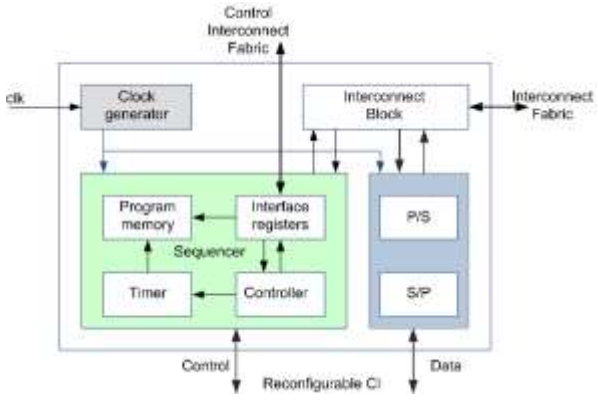

Fig. 2. Reconfigurable Cryptographic interface

The CI is a reconfigurable block which can support both custom and proprietary interfaces. The CI block consists of a programmable sequencer along with data path elements. The program memory of the sequencer can be loaded with transmit and receive microcode for serial buses as needed.

Through analysis of the various interfaces and to keep the sequencer as simple as possible we have discovered that if we support the following four OPCODES we can implement most of the interfaces. The OPCODES are shown in TABLE II

TABLE I

FRAME STRUCTURE

Byte Number	Byte Meaning	Byte Description	Bit Description
0	Control	This is the control byte which describes the purpose of the data and its validity. The bits are explained in the next column	0 – If set the data in the control byte and other bytes is valid 1 – Bypass the data, do not perform encryption or decryption 2-Perform encryption on data 3-Perform decryption of data 4 - Authenticate data 5-7 Reserved for future use
1	Reserved for future use	Reserved for future use	Reserved for future use
2	Destination code	The data can be intended for UPE modules inside the framework or for other modules connected to forward or reverse side.	One byte destination code identifying which UPE the data is intended for.
4-3	Data size	Number of data bits which are going to start from next byte.	Two byte data size which identifies the amount of data which will be contained in this frame.
N-5	Data	Actual data	Data value.

TABLE II

INSTRUCTIONS OF THE SEQUENCER

OPCODE	OPCODE Mnemonic	Description
No operation	NOP	No operation
Sample flag	SAMPF	Wait for a input flag to be equal to a particular value, before going to next instruction
Compare flag	CMPF	Compare the flag, before going to next instruction
Jump unconditionally	JUMP	Jump unconditionally to the address specified in the control word
Jump conditionally	JUMPC	Jump conditionally to the address specified in the control word

The control word of the sequencer is shown in TABLE III.

TABLE III

Bits	Field	Description
31:29	OPCODE	OPCODE
28:21	Jump Address	Jump Address used in Jump instruction
20:0	Flags	Wait or conditional flags which can be used to move to the next instruction in the sequence

The program memory contains the microcode for the various serial buses we will support at different memory locations. The controller will generate the control signals to the program memory for accessing the data at particular memory locations. There are many instances where the sequencer has to wait for some particular timed events in order to start a new cycle. For example in certain serial buses like I2S, ASP etc. a frame signal is generated periodically to which the entire data transmission/reception is aligned. The timer which is being provided in the sequencer can be used to generate such a frame alignment signal, and the sequencer will also align the data transmission/reception with the frame signal.

The interconnect fabric is a 32-bit proprietary split transaction bus. The CI is the master and initiates read or write transfers. A write transaction is started by the assertion of a write request signal along with the placement of destination address and write data on the respective address and write data buses. The slave responds with a ready signal back to the master indicating that it has accepted the transfer and executed it. A read transaction is started by the assertion of a read request signal along with the placement of destination address on the address bus. The slave responds with a ready signal back to the master indicating that it has accepted the transfer and executed it, along with the requested data on the read data bus. An interrupt signal is also present on the bus through which the slave can indicate to the master to perform a necessary read/write transaction.

The presence of a CI block in the framework enables the implementation of a generic solution: whether this framework is implemented in a FPGA, or used in a system for interfacing to off the shelf processors/DSP. The latency to start a transaction once it is activated is 1 clock cycle.

C. Framework Logic

The framework logic (FL) module sits between the CI and various processing elements inside the CSS block. The FL block sits in both the forward and reverse data paths and implements control logic and buffers inside it to store the data from both forward and reverse paths.

The control logic present in this block is responsible for decoding the frame structure shown in TABLE I. The decoder looks at bit 0 of the frame structure to see if the data packet received is valid or not. For a valid packet the different fields of the packet are decoded and loaded in the respective control logic. The received data is loaded into the buffers implemented in the framework logic, and full flag is generated when the received data count is equal to the “data size” field in the received packet. The destination code allows the FL block to generate a request to the required UPE block. Once the request is granted the data is read the from the dual port buffer by the target UPE.

The latency through the framework logic is 3 clock cycles.

D. User Processing Elements

The generic nature of the proposed framework allows the user to implement multiple processing elements in the CSS block. These processing elements can be related to key management, encryption, decryption, authentication, and integrity checking. The processing elements only need to confirm to the FL input/output signal requirements in order to function correctly.

The framework can be used in implementations where the UPE blocks need to establish multipoint communication channels between the forward and reverse paths or vice versa. The framework has catered for this by providing crossbar architecture to link blocks in such a case.

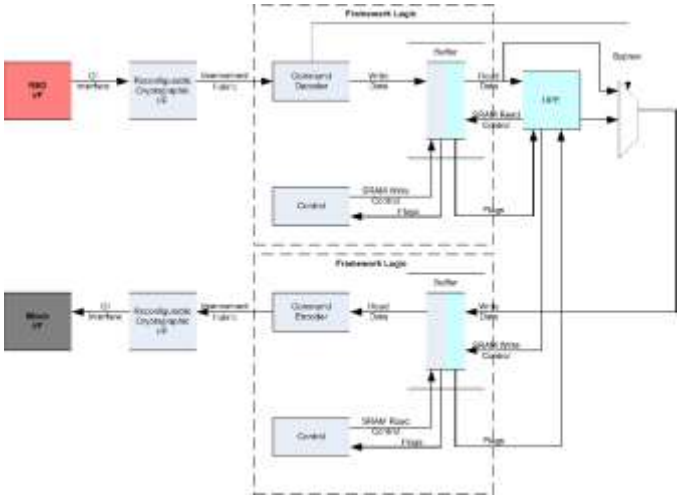

Fig. 3. Framework hookup to User Programmable Interfaces

The SRAM read/write bus, consist of the typical Xilinx block RAM signals such as *enable*, *write enable*, *address*, *write data*, and *read data*. In addition to these signals a *request* and *grant* signal has been added so that the correct UPE elements can be connected together. Additional flag signals such as *full* and *empty* have also been incorporated for flow control purposes.

IV. HW/SW Architecture Adoption in an SDR Implementation

The validation of the proposed methodology, the framework has been implemented in a software defined radio (SDR) solution for supporting several narrow and wide band networking waveforms. The SDR with these waveform can be used in the design of any cyber physical system. The SDR consists of a modem side PCB and a general purpose (GPP) side PCB which are connected through a crypto PCB which implements the RED-BLACK separation within the radio. The GPP side runs the radio's software as well as interfaces to the HMI devices such as keypad, display, microphone, as well as audio CODEC and Ethernet for data connectivity. The GPP PCB contains an ARM processor, and DSP to handle all this functionality. The modem PCB contains an ARM processor, DSP and FPGA to handle the coding/decoding, framing/de-framing, modulation/demodulation of the message that is to be transmitted or received according to the waveform running on it. A microcontroller, FPGA, non-volatile memory, and various functionality is present on the board to make the SDR a tamper resistant product. The microcontroller and FPGA communicate through the CI interface. The CI interface is reconfigurable and supports both standard and proprietary interfaces. The re-configurability feature allows the CI to function as a SPI, I2C, audio serial port (ASP), I2S, UART or any other proprietary interface. The modem board interfaces to the RF functionality of the SDR. The crypto, GPP and modem boards are custom developed PCB to implement a SDR solution. A binary configuration file is created from the Xilinx tools, and stored in the system. The microcontroller authenticates the binary file before loading it into the FPGA, providing an added integrity check on the overall system.

The crypto PCB contains a Xilinx SPARTAN3A DSP XCSD3400A FPGA, a microcontroller to make the SDR tamper proof, and non-volatile memory for key storage. The FPGA interfaces to the GPP board on one side and the modem board on the other side, as shown in Fig. . The GPP board sends plaintext to the crypto board which is then encrypted or passed through to the modem side for processing. In a similar manner the crypto FPGA receives encrypted or plaintext data from modem board which is decrypted or passed through as plaintext to the GPP side. The CI connects the GPP and modem boards with a frame structure as defined by the framework. The data which is not encrypted is always authenticated when being sent from RED-BLACK or BLACK-RED side.

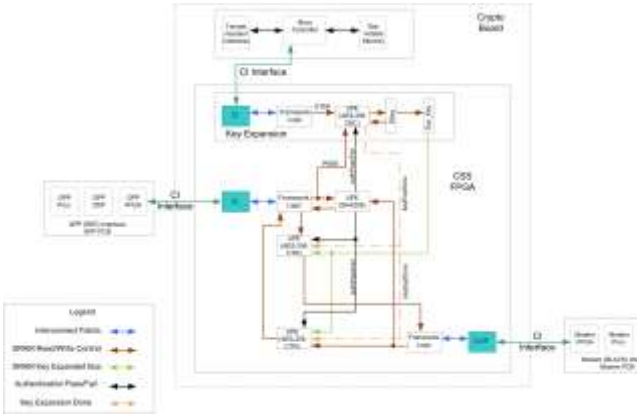

Fig. 4. CSS FPGA interfaced to RED and BLACK interfaces

In this particular solution five UPE are implemented in the FPGA for encryption, decryption, authentication and key generation. For encryption and decryption purposes a steam cipher, AES 256-bit in CTR mode, is implemented in one of the UPE modules. For authentication purposes the SHA-256 algorithm is implemented in another UPE module. While another two UPE implements the key expansion process.

The key expansion process consists of a two-step process. The transaction key (TEK) itself is stored in BLACK format in non-volatile storage, and is transferred to the FPGA by the microcontroller through CI. The key encryption key (KEK) which is required to decrypt the TEK is transferred from the GPP board to the FPGA in plaintext. The plaintext data coming from GPP board is authenticated before being forwarded to the UPE module responsible for key decryption. The key decryption UPE implements the 256-bit AES in CBC mode. Once the TEK is decrypted it is stored in internal FPGA memory. The key expansion UPE module then expands the

TEK for use by the AES modules for encryption and decryption.

The SHA-256 algorithm was implemented as proposed in [21]. The AES algorithm was implemented as proposed in [22].

TABLE V

FPGA IMPLEMENTATION RESULTS

IP	LUT	FF	RAM
SHA-256 UPE	1352	1706	0
AES-CTR-256 UPE	1436	1102	2
AES-CTR-256 UPE	2482	1310	3
AES-CBC-256 UPE	1436	1102	2
FL	101	160	2
CI	1152	502	1

The implementation of the FL and CI blocks is symmetric in the GPP, modem and Crypto FPGA, therefore the same amount of area is consumed in all the FPGA.

Using the reconfigurable nature of the CI, the following interfaces were qualified and tested to be working reliably with maximum throughput of 12.5MB/s while other proprietary interfaces were also developed, verified and used.

(1) SPI
(2) I2C
(3) ASP
(4) I2S
(5) UART

The proposed framework is architected to be used in a myriad of applications from internet of things (IoT), to handheld devices, to SDR applications. Some of these applications require performance while others are sensitive to power. The implementation results show that with a minimal increase in latency the proposed framework can scale for increasing performance, while power remains fairly constant.

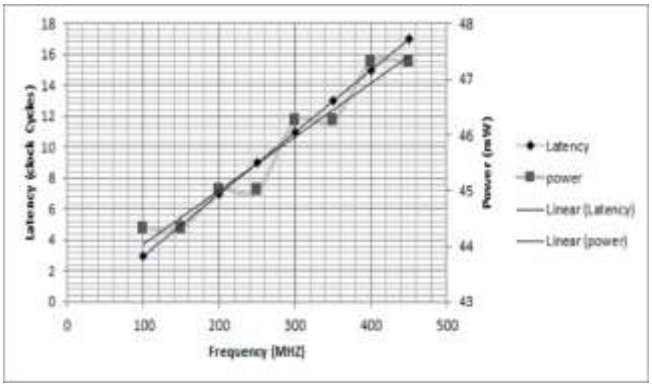

Fig. 5. CSS FPGA interfaced to RED and BLACK interfaces

V. ANALYSIS AND COMPARISON

Results of a reconfigurable framework designed for RED-BLACK separation in accordance with the requirements of the Security Software Communication Architecture Specification have been presented in this paper.

TABLE IV

SOLUTION COMPARISON

Proposed Solution	Mclean et al. **[15]**	Grand et al, **[16]**	Fitzgerald et al. **[18]**
On the fly reconfigurable cryptographic interface to avert side channel attacks	No such feature	No such feature	No such feature
Generic Framework compliant to CSS	Not compliant to CSS	Framework Compliant to CSS but not generic	No Framework. CSS implemented using FPGA properties
Compatibility shown with other Cryptographic implementation	No such feature	Own implementations of Cryptographic implementation	Own implementations of Cryptographic implementation
Not restricted to FPGA solutions	Highly restrictive FPGA implementation with specialized tools not available off the shelf	Restricted to FPGA implementation due to usage of Xilinx MicroBlaze Processors	Restricted to FPGA implementation due to usage of FPGA security features and Altera NIOS processors

It has been shown that the proposed architecture though is designed for cyber physical embedded device but is flexible enough to be used in a myriad of devices such as cyber physical devices. The solutions proposed in [16], [18] and our solution are different implementations of the CSS architecture proposed in [14], but our solution presents a novel approach to security and configurability by introducing a reconfigurable ‘on the fly’ cryptographic interface (CI). The implementation in [16] uses CORBA for message passing, and these messages are stored in a FIFO before being parsed. Two instances of 32-bit MicroBlaze CPU are used in [16] to implement the communication and control blocks. The communication block implements the communication channels between the RED and BLACK side, as well as the cryptographic algorithms. The controller block implements the security management and key algorithms. In our implementation, the crypto FPGA is responsible for the communication between the RED and BLACK PCB, while also handling all the cryptographic algorithms. The message received through the cryptographic interface is used to set up the communication path between the RED and BLACK side, and this communication channel is

reconfigurable. Our solution provides 'on the fly' configurability of off-chip interfaces, to prevent side channel attacks, a reconfigurable framework with very little overhead, and implementation results on a working SDR solution using custom developed PCB. The authors of [16] have not shared if their solution has been practically implemented. The overall solution presented in [20] is a hardwired solution with no configurability for a crypto subsystem (CSS). The crypto subsystem lies between the RED and BLACK side, and interfaces to the RED side through a CSS access controller. The access controller is connected to a bus fabric on the chip through which multiple peripherals can move data using a central DMA controller. No implementation results of the framework or crypto algorithms are reported. The implementation results of a single chip SCC in a secure Altera FPGA are shared by the authors in [18]. The implementation uses 2 NIOS-II processors, one for RED side and one for BLACK side, and AES-GCM core for encryption and authentication purposes. The paper presents results about the AES-GCM implementation while the other resources are built-in features of Altera Secure FPGA. There is no presentation of a framework or 'on the fly' configurability which is what our solution is providing.

VI. CONCLUSION

The results of a reconfigurable framework designed for RED-BLACK separation has been presented in this paper, and the proposed framework fulfills the requirements of the Security Software Communication Architecture Specification, and supports the RED-RED, RED-BLACK, and BLACK-BLACK transfers as outlined. The CI between the RED-BLACK sides can be reconfigured 'on the fly' at selected intervals to guard against side channel attacks. This concept is similar to frequency hopping in communication to guard against eavesdropping. The proposed framework can be used to secure the network interface side and overcome the problem highlighted in [8], [9] for a space based system. Practical implementation results of the framework in a software defined radio (SDR) implementing networking waveforms with custom developed PCB(s) has been presented. The focus of our work is to provide a reconfigurable framework and not on providing any unique architectural implementations of encryption (AES) [16], [18], or authentication (SHA) as done in [16] algorithms. We therefore use already provided architectures for AES and SHA in [22], [21] respectively to prove the viability of our solution. The reconfigurable nature of the CI in our proposed architecture removes any dependency on using runtime re-configurability FPGA devices. The 'on the fly' configurability of the CI interface also has an impact on the software side of the system in which it is being used. If the hardware communication channel between the two interfaces is changed, this leads to a change in the drivers of the operating system and the application which is calling it. In our current implementation, this is being handled in a very controlled manner. Whenever the CI interface is about to switch, all communication is stopped and the application is notified of this impending change which leads to a system reboot. Future research on dynamic switching of interfaces and drivers can be carried out to make this a seamless effort.

REFERENCES

[1] E. A. Lee, "Cyber Physical Systems: Design Challenges," in *11th IEEE Symposium on Object Oriented Real-Time Distributed Computing (ISORC)*, 2008.

[2] A. T. Klesh, J. W. Cutler and E. M. Atkins, "Cyber-Physical Challenges for Space Systems," in *2012 IEEE/ACM Third International Conference on Cyber-Physical Systems*, Beijing,China, 2012.

[3] S. Karnouskos , "Stuxnet worm impact on industrial cyber-physical system security," in *Proceedings IEEE 37th Annual Conference Industrial Electronics*, Melbourne, Australia, 2011.

[4] R. McMillan, "New 'Kraken' GSM-cracking software is released - GSM eavesdropping for the masses comes to Black Hat," *PC World Business Center*, 22 July 2010.

[5] Y. Zhou and X. Jang, "Dissecting Android Malware: Characterization and Evolution," in *2012 IEEE Symposium on Security and Privacy (SP)*, San Francisco, 2012.

[6] Y. Song, K. Zhou and X. Chen, "Fake BTS attacks of GSM System on Software Radio Platform," *Journal of Network*, vol. 7, no. 2, pp. 275-281, 2012.

[7] B. V. Leeuwen, V. Urias, C. Glatter and A. Interrante-Grant, "Testbed for Cellular Telecommunications Cyber Vulnerability Analysis," in *2013 IEEE Military Communications Conference, MILCOM-2013*, San Diego, 2013.

[8] J. S. Warner and R. G. Johnston, "A simple demonstration that the Global Positioning System (GPS) Is Vulnerable to Spoofing," *Journal of Security Administration*, 2003.

[9] T. E. Humphreys, B. M. Ledvina, M. L. Psiaki, B. W. O'Hanlon and P. M. Kintner, Jr, "Assessing the Spoofing Threat:Development of a Portable GPS Civilian Spoofer," in *ION GNSS Conference*, Savanna, GA, USA, 2008.

[10] J. Irza, M. Doerr and M. Solka, "A Third Generation Many-Core Processor for Secure Embedded Computing Systems," in *2012 IEEE Conference on High Performance Extreme Computing (HPEC)*, Waltham, 2012.

[11] TCG Published, "TCG Specification - TPM 2.0 Mobile Reference Architecture," TCG, 2014.

[12] Z. Zha, M. Li, W. Zang, M. Yu and S. Chen, "AppGuard: A hardware Virtulization Based Approach on Protecting User Applications from Untrusted Commodity Operating System," in *2015 International Conference on Computing, Networking, and Communications (ICNC)*, Garden Grove, 2015.

[13] JTRS Standards, "Software Communications Architecture Specification," JTRS Standards, San Diego, 2006.

[14] Joint Tactical Radio System (JTRS) Joint Program Office, "Security Supplement to the Software Communication Architecture Specification V1.1," Joint Tactical Radio System (JTRS) Joint Program Office, 2001.

[15] M. Mclean and J. Moore, "http://mil-embedded.com/pdfs/NSA.Mar07.pdf," March 2007. [Online]. Available: http://mil-embedded.com.

[16] M. Grand, L. Bossuet, L. B. Gal, D. Dallet and G. Gogniat, "A reconfigurable Crypto Sub System for the Software Communication Architecture," in *2009 International Military Communications Conference (MILCOM 2009)*, Boston, 2009.

[17] ARM Limited, "ARM Security Technology - Building a Secure System using TrustZone Technology," ARM Limited, 2009.

[18] A. Fitzgerald, M. Lukowiak, M. Kurdziel, C. Mackey, K. Smith, B. Boorman, D. Harris and W. Skiba, "FPGA-Based, Multi-Processor HW-SW System for Single-Chip Crypto Applications," in *2010 Military Communications Conference (MILCOM 2010)*, San Jose, CA, 2010.

[19] M. Grand, L. Bossuet, G. Gogniat, B. Le Gal, J.-P. Delahaye and D. Dallet, "A Reconfigurable Multi-core Cryptoprocessor for Multi-channel Communication Systems," in *2011 IEEE International Parallel & Distributed Processing Symposium*, 2011.

[20] N. Ali, "Novel Architecture for Software defined Radio," in *2011 IEEE International Conference on Microwaves, Communications, Antennas and Electronic Systems (COMCAS)*, Tel Aviv, 2011.

[21] L. Dadda, M. Macchetti and J. Owen, "The Design of a High Speed ASIC unit for the hash function SHA-256 (384, 512)," in *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition Designers' Forum (DATE 04)*, Paris, 2004.

[22] S. M. Farhan, S. A. Khan and H. Jamal, "An 8-bit systolic AES architecture for moderate data rate applications," *Microprocessors and Microsystems,* vol. 33, no. 3, pp. 221-231 , 2009.

[23] J. Mitolla III, "Software Radio Architecture: A mathematical perspective," *IEEE Journal on Selected Areas in Communications,* vol. 17, no. 4, pp. 514-538, 2002.

Mr. Arshad Riazuddin obtained his MS and BS degrees in electrical engineering from University of Texas at Arlington, Arlington, Texas in 1988 and 1986 respectively. He is currently pursuing his Ph.D. in Computer Engineering from Center of Advanced Research in Engineering (CASE), Islamabad.

He has over 20 years of industrial experience. His areas of expertise are ASIC design, computer architecture and embedded systems.

Dr. Shoab A. Khan did his PhD in Electrical and Computer Engineering from Georgia Institute of Technology; Atlanta, GA. Dr. Khan's areas of specialization are Digital Signal Processing, Digital Design and Communication System.

He is a Professor of Computer Engineering at College of EME, National University of Science and Technology (NUST). He is an inventor of 5 awarded US patents and has 260+ international publications. His book on Digital Design is published by John Wiley & Sons and is being followed in national and international universities. Dr. Shoab Ahmed Khan has more than 22 years of industrial experience in companies in USA and Pakistan. He has been awarded Tamgh-e-Imtiaz (Civil), National Education Award 2001 and NCR National Excellence Award in Engineering Education. He is the Chairman of Pakistan Association of Software Houses (P@SHA) and is a member of Board of Governance of many entities in the Ministry of IT and Commerce. He has also served as member of National Computing Council and National Curriculum Review Committee.