An effective symmetric key recovery scheme for secure onboard satellite applications

A.Ruhan Bevi [1], S. Malarvizhi [2]

[1] Assistant Professor, Department of Electronics and Communication Engineering, SRM University, Kattankulathur, 603203, India

[2] Professor and Head, Department of Electronics and Communication Engineering, SRM University, Kattankulathur, 603203, India

Abstract

Reconfigurable FPGAs promise high density, performance and good schedulable options with their wide range of SRAM target devices that can be hosted for any applications. The SRAM FPGAs are widely used due to their vast resources and ease to design, but critically suffers due to radiation errors called single event upsets (SEU). These errors are prominent when the FPGA device is employed beyond an altitude. Therefore, it is very essential to prevent or to correct the SEUs, particularly for applications in satellites.

The corruption of a cryptographic key by a SEU is considered to be a serious issue, because a single bit change in the key may damage the entire block of data and lead to severe increase in retransmission rates. The cryptographic keys are protected from SEUs by introducing an extra key recovery module (KRM). The proposed design is tailored for implementing a secured key for a Tiny Encryption algorithm (TEA) module resistant to SEU based on the stringent constraints of power, energy and cost. This paper proposes the FPGA implementation of TEA algorithm in hardware for onboard satellite applications.

Keywords: *Reconfigurable FPGA, Tiny Encryption algorithm, security, satellites*

1. Introduction

Security has gained importance recently, as there are increasing threats for the sensitive data to be transmitted. Security services for the satellite and space application presents challenges to the electronic field as they are operating on sensitive data which is either received or transmitted to the ground station. It is mandatory to ensure that the data should not be intercept by any unauthorized entries. So, a variety of cryptographic algorithms are used in the satellites for performing encryption and decryption depending on their applications. Hence, encrypting the sensitive data and commands is a good solution at a cost of designing and employing of complex cryptographic algorithms. There is a real need for maintaining the throughput and power dissipation in satellite applications to enable low power operations.

The growing technological advancements pose new threats [3] to the satellite communication and widely presented in the past literature. Threats in the dimensions of SEU attack due to cosmic radiation are a major cause of unconditional damage of satellites. The percentage of radiation penetration and the effect of the soft errors in satellite and aerospace applications make onboard data processing extremely difficult. Security architecture for earth observation satellites [15], [18] with AES algorithm and parity based fault detection were used to mitigate SEUs but consumes more power as the AES algorithm is highly complex consisting [10] of many mathematical iterations.

SEUs can be defined at the surface level as a class of errors originated due to the radiation which could change the state of a single bit data momentarily. The effect of such SEUs increases with the increase in the altitudes [9] and it is a major source of soft errors caused by external radiation. Though the count of SEU errors produced is less, the accumulated effect of such SEU error may propagate as multiple faults. As a case, SEU which complements a transistor from cut off to active state may produce a faulty output at that point, which therein continues as a cascade of errors throughout the path of the circuit, there by, multiplying the faults. In this way, SEU magnifies the errors on the onboard data. Several methods like prevention, detection, recovery methods for SEUs are used to reduce the error rates. The Recent Soft Error Rate (SER) testing [4] of SRAM-based FPGAs from Actel shows a significant and growing risk of functional failure due to the corruption of configuration data, especially when the system has higher densities.

The use of Selective Triple Modular Redundancy (STMR) [5] is a good alternate for TMR solution [6] as STMR triplicates only the gates that are proved to be SEU sensitive. This method attempts to reduce the costs associated with logic redundancy by applying them only to the most critical components of a design.

Implementing crypto algorithms in a dedicated hardware device improves the speed and performance [8], [16] of the system by multiple times than the software approach. This is because the FPGA architectures are void of the opcode fetch, decode and execute time of the processor employed, as it is concerned only with interconnect and propagation delays of the signals. The proposed TEA algorithm which is implemented in FPGA can be used as a h ardware accelerator support to the onboard system to raise the performance grade of the satellites. The use of key distribution and authentication in satellite systems is very essential and discussed in [17] but the security required for the data communication was not addressed. This paper uses an additional circuitry to secure the cryptographic keys of the TEA algorithm against SEU attacks.

We have designed a TEA algorithm for satellite security applications to handle the sensitive data to be transmitted from or to the satellite accompanied by a key recovery mechanism as in [2]. The proposed design protects the keys of the TEA encryption algorithm from the SEU attacks and the symmetric key is recovered during the application itself. The real time operations are ensured thereby sustaining the reliability of the data transfer.

The paper is organized as follows. The section 2 focuses on SEU and its attacks. The TEA algorithm and the recovery protocol are addressed in section 3. The section 4 deals with the implementation of the hardware modules and conclusions are drawn in section 5.

2. SINGLE EVENT UPSETS ATTACKS ON FPGA

Electronic faults or failures in boards are generally classified as hard and soft fails. The hard fails of a circuit is often termed as permanent fails which are repairable. The causes of such a f ailure may be a loose connection, open or a short circuit etc which contributes to the permanent failure of the device. A soft fail on the other hand is a temporary failure and it may or may not be destructive. It can be classified as a t ransient fault produced by the environmental conditions like pressure, temperature, alpha particles etc or a intermittent fault produced by the variations of non environmental conditions like parasitic value changes, noise etc.

SEU are a class of soft errors which is defined in [7] as " Radiation induced errors in micro electronic circuits caused when charged particle usually from the radiation belts or from the cosmic rays loose energy by ionizing the medium through which they pass, leaving behind a wake of electron hole pairs" ... NASA Thesaurus.

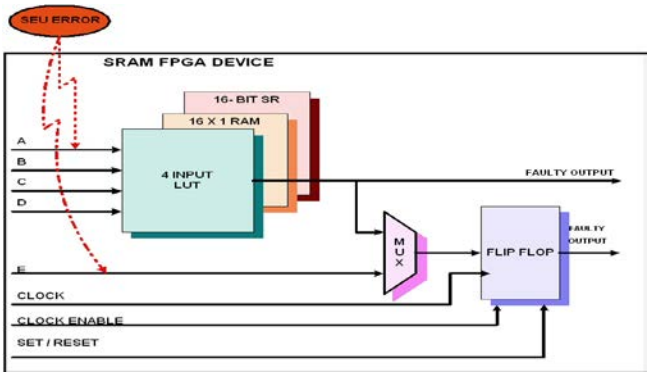


Figure 1: SRAM FPGA device affected by SEU error

There are many definitions that exist in the literature for the SEU and [4] states SEU as an electronic circuit, that bears no permanent hardware fault, may witness unexplained events resulting in single bit changes spontaneously in the system, and there is no way to repeat such failures. This SEU is produced due to one of the following reasons.

- When a n ucleus of an unstable isotope looses energy and moves from higher to lower state. This loss is accompanied by the emission of a highly kinetic alpha particle which bombards with a silicon circuit to produce a SEU.
- Due to the bombardment of the neutrons from the cosmic rays. The various cascaded layers of the cosmic rays tend to have a r eaction with the silicon devices.
- The P type semiconductor Boron has a strong affinity to react to the radiation which produces SEU.

We define SEU are the class of soft errors with the following properties.

a) Their occurrence is random, so they are termed as random errors.
b) They are temporary or non permanent which may not exist for more than a period of time.
c) They are caused by the radiation induced charged particle, striking a silicon device.
d) They are produced on the circuits that are employed beyond an altitude above sea level.
e) Percentage of SEU errors increase with the increase in the altitude and negligible at the ground level.
f) Guaranteed single bit changes or flips produced in the silicon circuits. The functional failure arises due to the corruption of a single or multiple bit on

a SRAM FPGA board due to the damage of the memory, programmable interconnect or a s elect line of a multiplexer etc as shown in figure 1.

g) SEU injection increases twice as the technology is scaled down.

The basic approach to deal with SEU is either to prevent it or to recover from its effect. The prevention methods usually consist of manufacturing FPGAs tolerant to such errors. The major FPGA vendors like Xilinx and Altera manufacture radiation tolerant [9] and hardened boards for aerospace and space applications. These boards are very expensive and tough for modifications. Curing SEU, is really a complicated task of detecting the fault and thereby recover the fault by correcting it. Assessment of the fault through testing methods requires skill and advanced testing tools to measure the impact of the error in the functionality of the circuit. Both prevention and curing schemes work on the common objective, to secure FPGAs employed for space and aerospace community against the SEU attacks.

3. TEA AND ITS RECOVERY MODULE DESIGN

3.1 TEA Implementation

Although, there are many complex encryption algorithms present in the open literature [1], [8], [10], [16] TEA algorithm is chosen because of its structure which is optimal and noted for their simplicity of description and implementation. The TEA stands for Tiny Encryption algorithm which is a symmetric keyed algorithm that operates on 64-bit blocks and uses a 128-bit key. The block ciphers used are light weighted and suitable to be used in portable wireless communication applications like wireless sensor nodes [11], [12], [13], [14] for ensuring security. Symmetric algorithms are faster (as both the communicating parties share a common key) and hard to break, when worked with lengthy keys. The key distribution of the symmetric algorithms is difficult as it is a tough task to manage and predistribute the keys from one end to the other end. Hardware implementations of symmetric crypto algorithms are less intensive in mathematical computations. The implementation of encryption algorithm in hardware is a complex process and time consuming than a software encryption but it is adopted due to the following advantages.

a) Guaranteed high speed and performance than a software encryption.
b) Tough to break the code, so the security level is increased.
c) Difficult for the attacker to understand the functionality of the silicon layout of the device.
d) Reduced latency and increased throughput is used to meet any real time constraints.
e) Reduced consumption of power and energy.

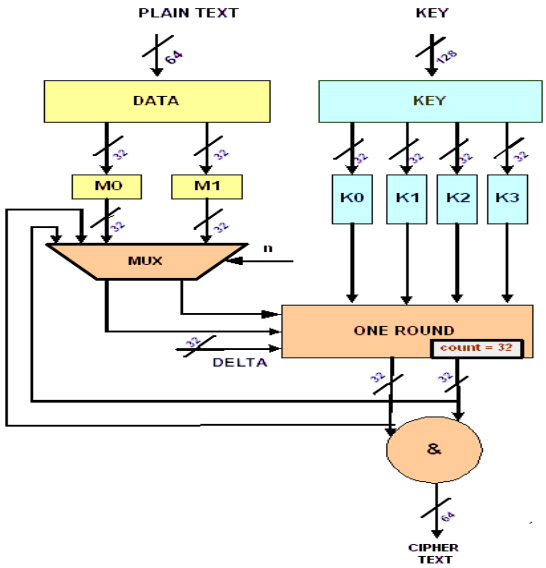


Figure 2: TEA implementation with 32 rounds

The TEA operates on 64 (block size) data bits and 128-bit key with 32 cycles. In this block cipher, 64-bit data block is separated into two halves and 128-bit key is divided into four subkeys. The round function is applied to one half of the data, using the subkeys and the output is (exclusive-or-ed) with the other half given in (3). Each cycle follows the same model and the two halve are then fed back as an input for the next round function. The constant delta = ∂ = ($\sqrt{5}$ −1) * 2^31 = $9E3779B9_h$ is derivative of the golden number ratio to distinguish the sub keys.

In the TEA encryption process, the 64 bits block cipher is divided into two halves say M0, M1 and the key is divided into four subkeys say K0, K1, K2 and K3 of size 32 bits each. These data and subkeys are then applied to the round function which performs 32 iterations. It uses big – endian approach for every round function. Each of the iteration executes a single round function using one half of the data and subkeys.

$$M0[i] = M0[i\text{-}1] \boxplus F(M1[i\text{-}1], K[0, 1], DELTA[i]) \quad (1)$$
$$M1[i] = M1[i\text{-}1] \boxplus F(M0[i\text{-}1], K[2, 3], DELTA[i]) \quad (2)$$

The round function, F, is defined by

$$F(M, K[j,k], DELTA[i]) = ((M << 4) \boxplus K[j]) \oplus (M \boxplus delta[i]) \oplus ((M >> 5) \boxplus K[k]) \quad (3)$$

TEA is an iteration cipher, whose ith round has the inputs M0(i-1) and M1(i-1) as in (1) and (2) which is derived from the previous round and the subkey K(i) derived from the 128 bit key K.

Each round is executed as shown in the equations above and the multiplexer in figure 2 is used to select the block cipher either from input or from the feedback to proceed with the further iterations. At the end of the 32nd iteration, it outputs the complete cipher text. The single TEA round function performs the simple mixed orthogonal algebraic functions such as right/left shifts, integer addition and exclusive or operations.

The TEA algorithm is implemented in VHDL and one round is completed for one clock cycle. Totally 32 clock cycles were required to complete all the rounds and to produce the output. The clock frequency is set to 25MHz, where the clock period is 40ns. The full encrypted output after 32 rounds is observed at 1.3 micro seconds approximately.

3.2 Key recovery module for TEA algorithm

A key recovery module (KRM) is designed at the cost of extra hardware to protect the cryptographic keys from the single bit changes occurring in satellites. The KRM designed for TEA consists of random number bank (RNB), which holds a set of 64 random numbers (R1,R2….R64) each of them are of 128 bits in length as in figure 3.

Each random number from the RNB is associated with a 128 bit symmetric key for TEA. So, there are 64 symmetric keys present in the RNB which can be used for operating TEA algorithm. The keys in the RNB cannot be directly accessed as they would be damaged by the SEU attack. The TEA algorithm is designed in such a way, that it will not work directly on the keys supplied by the communicating end but it would work on the key provided by the KRM.

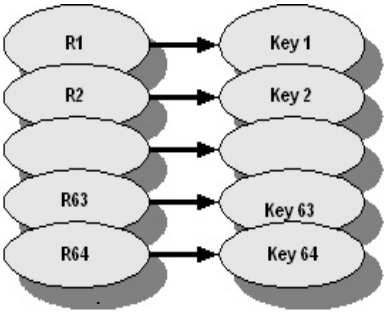
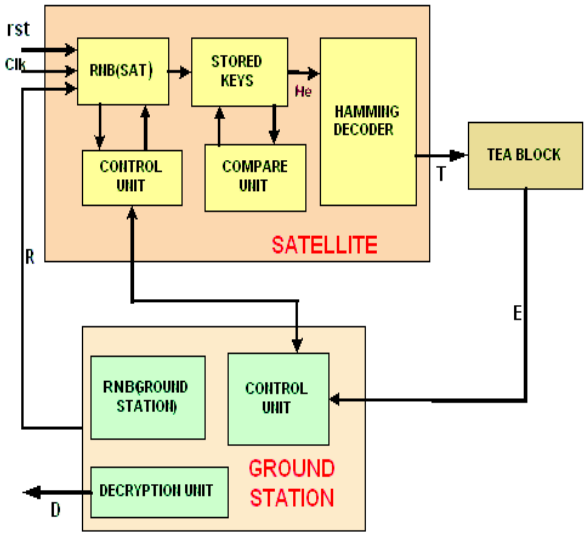


Figure 3: Random numbers associated with key

Considering the satellite applications, the ground station (party1) and the satellite (party2) have identical random number bank (RNB). The 128 bit key is hamming encoded along with the random number R to provide immunity against SEU. The hamming encoder is internally included as it is used to detect and correct single bit error and to detect double bit errors in random number and key. The number of parity bits added is 9 for every combination of random number Ri and key which is less redundant. Each random number in the RNB is unique and every R belonging to the RNB is associated with a key for TEA module.



T – Trusted key
E – Encrypted data
D – Decrypted data
H_e – Hamming encoded key
R – Random number

Figure 4: Hardware module generated for trusted key implementations.

When the ground station intends to communicate with the satellite, it initiates a random number Ri Є RNB. The random number Ri is sent to the satellite through a secured channel. The satellite receives the incoming Ri and it compares the incoming Ri with all the elements of its own RNB. It is to be noted, that RNB of ground station and satellite are same and has the same elements. i.e. {RNB (Sat)} = {RNB (GS)}. The compare unit in figure 4, performs bit wise xor operation of the incoming RiЄ RNB (GS) with all the elements of the RNB (Sat). The compare unit of the satellite decides whether the incoming Ri is one of the trusted element of RNB (sat). After a successful comparison, the stored Hamming encoded key H_e of the random number Ri is decoded using internal hamming decoder circuit as in figure 4.The decoded 128 bit key is now considered as a trusted symmetric key T for encrypting the sensitive data of the satellite using TEA algorithm. The control unit is responsible for incrementing the pointer in the RNB after every use of the key and also to zero the entries of

the keys which are already used. This increases the reliable usage of the key, as the past keys may be useful for the attacker to predict the future keys. The encrypted data E (cipher text) is now sent to the ground station. The decryption unit accepts the encrypted data E and by using the same symmetric trusted key T, it decrypts E and generates the plain text D.

Key features of the random numbers are as follows:

- The values of R are randomly distributed along the RNB.
- Every value of R does not contain a valid key. The length of the R is 128 bits and there are 2^{128} combinations of R are possible, but only 2^7 (64) combinations are valid.
- R itself is never used as a key.
- Any R value other than the 64 (2^7) combinations inside the RNB is considered to be unauthorized and invalid or considered as an attempt of the attacker. The length of the R is kept to be 128 bits to make R resistant against the brute force attacks.
- Once if the R belongs to RNB releases the associated key, then the key is zeroed and never used again. The reason is, the previously released keys should not clue the attacker to identify the future communication.
- The TEA algorithm will operate only on the elements of RNB where Ri € RNB .
- Both R and Keys are encoded in Hamming format as they are prone to SEUs.

3.3 Symmetric key sharing between Satellite and Ground station

It is clear from the section 3.2, that the random number generated from the ground station releases a symmetric trusted key in the satellite. Once when the ground station sends random number to the satellite, the satellite compares and verifies whether the incoming R is authorized. The authorized R is capable of releasing the associated key in the satellite. The satellite uses the trusted key to encrypt the sensitive data using TEA algorithm and the encrypted data is sent to the ground station.

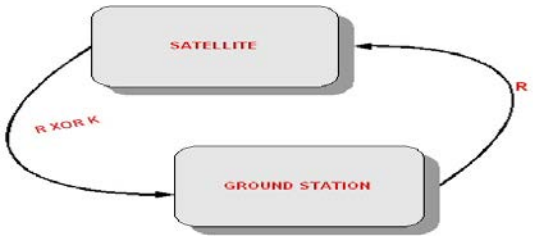


Figure 5: Signal transmission between satellite and ground stations

The hardware in the satellite performs xor operation of the trusted key T and the random number R and sends it to the ground station as in the figure 5. The ground station already knows R and now it receives R xor K. It performs a simple xor operation between R and (R xor K) and retrieves the key K. This is given by K = R xor (R x or K). This is supposed to happen before the encrypted data from the satellite reaches the ground station. Therefore, both the communicating parties (satellite and ground station) share a common symmetric key from the RNB.

4. IMPLEMENTATIONS OF KEY RECOVERY MODULE ON TEA ALGORITHM

We have considered a single bit flip of a cryptographic key of a TEA algorithm which leads to a selection of a corrupted key. Therefore the TEA module is fitted with a key protection and recovery protocol explained above, which operates using random number generated once. The entire TEA along with Key recovery module is coded in VHDL, simulated and synthesized and successfully implemented in Xilinx Virtex4 family. The figure 7,8,9 shows the schematics of TEA algorithm, Key recovery module and the TEA with KRM respectively. The table 1 shows the various performance factors like number of slices occupied, memory used, logic and routing delay of the circuit designed and the throughput of the hardware. The throughput in Kbps for the different implementations is measured as the number of bits delivered in one second. An important metric for embedded systems called energy efficiency [10] is calculated as the throughput per energy (Gbits/J) which is presented in table1.The figure 6 shows the complete delay analysis of TEA with and without KRM. The code size and the slices used for the TEA encryption process in this paper is less and the delay comparisons shows that the TEA algorithm with KRM pays only a small compensation of resources when compared to a TEA without KRM. This is clearly tolerable to the security offered to the keys against the SEU attack.

Table 1: Performance analysis of proposed system

(Target device: XC4VLX25FF668-12)

Parameters	TEA Implementation	Key Recovery Module (KRM)	TEA + Key Recovery Module (KRM)
No. of slices	233	975	1325
No of FlipFlops	139	376	625
Memory kbps	226956	252512	256736
Logic Delay	4.563 ns	15.63 ns	32.23 ns
Routing delay	2.669 ns	63.76 ns	89.98 ns
Area % of total amount	2%	5%	12%
Power (mW)	145.2	412.34	454.77
Throughput (Mbps)	266	198	184
Energy (Gbits/J)	1.8	0.48	0.40

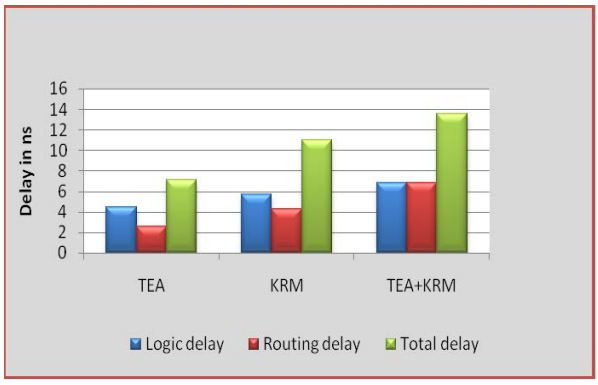


Figure 6: Delay comparisons between TEA with and without key recovery module

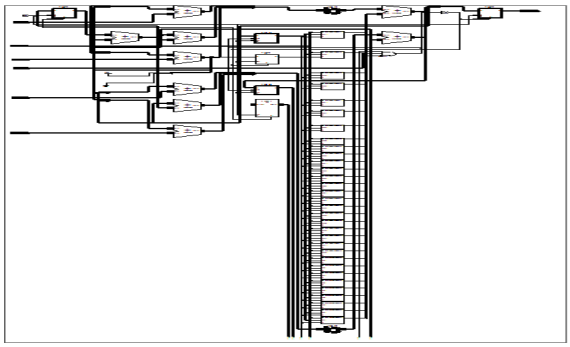


Figure 7: Schematic of TEA block

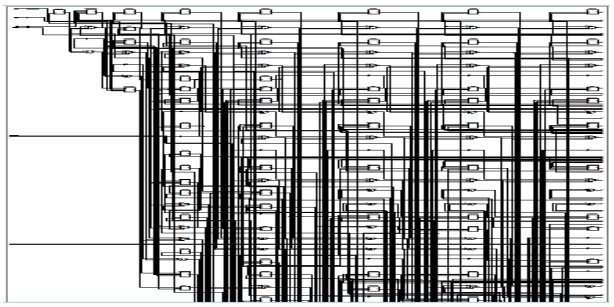


Figure 8: Schematic of Key Recovery Module

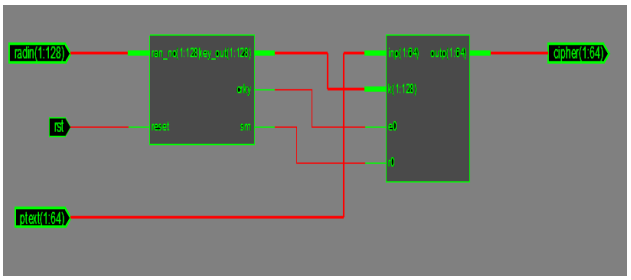


Figure 9: Schematic of Key recovery module generating a key for TEA encryption block

5. CONCLUSIONS

The results are investigated for the utilization of the resources both for with and without the usage of the key recovery scheme. It is very clear that the available resources of the target device can be used for the implementation of the key recovery module. The length of the random number generated is kept large so that it is very useful in resisting the brute force attacks. In future, the number of secret keys can be increased and the length of the random number can also be increased to provide good tolerance against brute force attacks.

The proposed scheme for key recovery is definitely a good alternative for a full TMR solution which is very complicated and robust. The overhead produced by the additional circuitry to the TEA module is clearly tolerable because 80% resources of high density FPGA families are unused. This area can be successfully utilized to overcome the existing security challenges caused by the single event upsets of today's and future satellites.

References

[1] Thomas Eisenbarth, Sandeep Kumar, Christof Paar, Axel Poschmann and L. Uhsadel, "A survey of lightweight cryptography implementations" in IEEE Design & Test of Computers, Special Issue on Secure ICs for Secure Embedded Computing ,vol 24, no 6, pp 522-533, November 2007.

[2] Marico Juliato ,Catherine Geboytes, "An Approach for recovering satellites and their cryptographic capabilities in the presence of SEUs and attacks", in *IEEE* Conference, August 2008.

[3] S.Bain,"The increasing threat to satellite communications", in Online Journal of Space Communication, November,2003.

[4] Paulgraham, Michae lCaffrey,Jason Zimerman, Prasanna Sundararajan "Consequences and categories of SRAM FPGA Configuration SEUS" in Military and Aerospace Programmable logic devices international conference, Washington 2003.

[5] P. Samudrala, J. Ramos, and S. Katkoori , " Selective triple modular redundancy (STMR) based single-event upset (SEU) tolerant synthesis for FPGAs" in IEEE Transactions on Nuclear Science, 51:2957–2969, 2004.

[6] Andrea Manuzzato, Student Member, IEEE, Simone Gerardin, "Effectiveness of TMR-Based Techniques to MitigateAlpha-Induced SEU Accumulation in Commercial SRAM-Based FPGAs", in IEEE transactions on nuclear science, vol. 55, no. 4, August 2008.

[7] Fan Wang, Vishwani D.Agarwal , "Single Event Upset: An Embedded Tutorial" in 21st IEEE International conference on VLSI design , 2008.

[8] A.Hodjat and I.Verbauwhede, "High-throughput programmable crypto coprocessor," in IEEE Micro, vol. 24, no. 3, pp. 34–45, May/Jun. 2004.

[9] C.C. Yui, G.M. Swift, C.Carmichael, R. Koga and J.S.Geoge, "SEU Mitigation Testing of Xilinx Virtex II FPGAs", Proceedings of NSREC03, Monterey, California,USA, July 2003.

[10] Guy Gognia, Tilman Wolf, Wayne Burleson, Jean-Philippe Diguet, Lilian Bossuet, and Romain Vaslin, "Reconfigurable Hardware for High-Security/High-Performance Embedded Systems: The SAFES Perspective", in IEEE Transactions On Very Large Scale Integration (VLSI) Systems, vol. 16, No. 2, February 2008.

[11] P Israsena, "Securing ubiquitous and low-cost RFID using tiny encryption algorithm", 1st International Symposium on Wireless Pervasive Computing, 2006.

[12] P. Israsena, "Design and implementation of low power hardware encryption for low cost secure RFID using TEA", in Fifth International Conference on Information, Communications and Signal Processing, 2006.

[13]P.Israsena,"On XTEA-base Encryption/Authentication Core for Wireless Pervasive Communication" in International symposium on Communications and Information Technologies, ISCIT, 2006.

[14] Aaron Garett, John Hamilton and Gerry Dozier , "A comparison of genetic algorithm techniques for the cryptanalysis of TEA ,, in International jornal of Intelligent control systems, September, 2007.

[15]Roy-Chowdhury, J. Baras, M.Hadjitheodosiou, and S. Papademetriou, "Security issues in hybrid networks with a satellite component" in IEEE Wireless Communication", 12(6):50–61, 2005.

[16] Goodman and A. P. Chandrakasan, "An energy-efficient reconfigurable public-key cryptography processor," in IEEE Journal of Solid-State Circuits, vol. 36, no. 11, pp. 1808–1820, Nov. 2001.

[17] Papoutsis, G. Howells, A. Hopkins, and K. McDonald-Maier, "Key generation for secure inter-satellite communication", in Second NASA/ESA Conference on Adaptive Hardware and Systems (AHS 2007), pages 671–681. IEEE Computer Society, 2007.

[18] T. Vladimirova , R.Banu, and M. Sweeting, "On-board security services in small satellites"' in MAPLD Proceedings,2005.

First Author

A. Ruhan Bevi is a Assistant Professor in the Department of ECE and pursuing Ph.D in S.R.M university in the area of VLSI and Embedded systems. She has completed B.E in Electronics and C ommunication Engineering and Masters in Embedded Systems in Anna University, Chennai, India. The research interests include security architecture design, adaptive security using FPGA, Low power embedded design etc.

Second Author

Dr. S. Malarvizhi is a Professor and Head, in the School of ECE, SRM University performing teaching, research, design and consulting in the general area of Electronics and Communication Engineering. She completed Masters in the area of Applied Electronics and Ph.D in Anna University, Chennai, India. Having a total teaching experience of 20 years, she has published in all reputed journals and conferences. She is a member of IEEE and reviewer for many international conferences and j ournals. The research interests include VLSI signal processing, embedded security, low power VLSI design, MIMI OFDM implementations etc.