ISSN: 2320-8090

*Available online at http://www.journalijcst.com*

*International Journal of Current Science and Technology*
*Vol.5, Issue, 6, pp. 431-437, June, 2017*

IJCST

## RESEARCH ARTICLE

# SECURITY STRATEGIES IN EMBEDDED SYSTEMS

**Okereke G. E and Oluoha Onyekwere U***

Department of Computer Science, Faculty of Physical Sciences, University of Nigeria, Nsukka, Nigeria.

**ARTICLE INFO**

***Article History:***
Received 20th March, 2017
Received in revised form 5th April, 2017
Accepted 14th May, 2017
Published online 28th June, 2017

***Key words:***
Embedded systems, security treats, treat countermeasures, security strategies.

**ABSTRACT**

Embedded systems are a unique set of computing devices, with unique characteristics, which in turn gives rise to unique and interesting vulnerabilities. These vulnerabilities have become even more evident with the ever increasing rise in the use of embedded systems and the advent of the Internet of Things (IoT). The ever increasing cases of information security breaches necessitates the pressing need for greater and more comprehensive security measures for embedded systems. Unlike in general Information Security, where security is considered as the implementation of a hardware cryptographic algorithms and/or security protocols, embedded systems present a totally different set of metric which must be considered throughout the design process (from design conception to end of life and disposal). In other to ensure a more reliable and resilient security solution, these new metrics must be considered side-by-side other important metrics including cost, performance, and power.
This paper closely peruses the unique characteristics of embedded systems, establishing major challenges encountered in embedded system architecture design, and surveys solution techniques in current literature and emerging research proposed to address these unique challenges.



## INTRODUCTION

There is no agreed definition of embedded systems, due mainly to the fluidity and speed at which its' technology evolves (Tammy Noergaard, 2013). Mouaaz & Ahmed (2012) described an embedded system as a special-purpose computer system designed to perform small number of dedicated functions for specific application(s). Every embedded system is made up of software and hardware elements which interact with each other. Based on this, an abstraction could be derived to represent the general architecture of embedded systems.

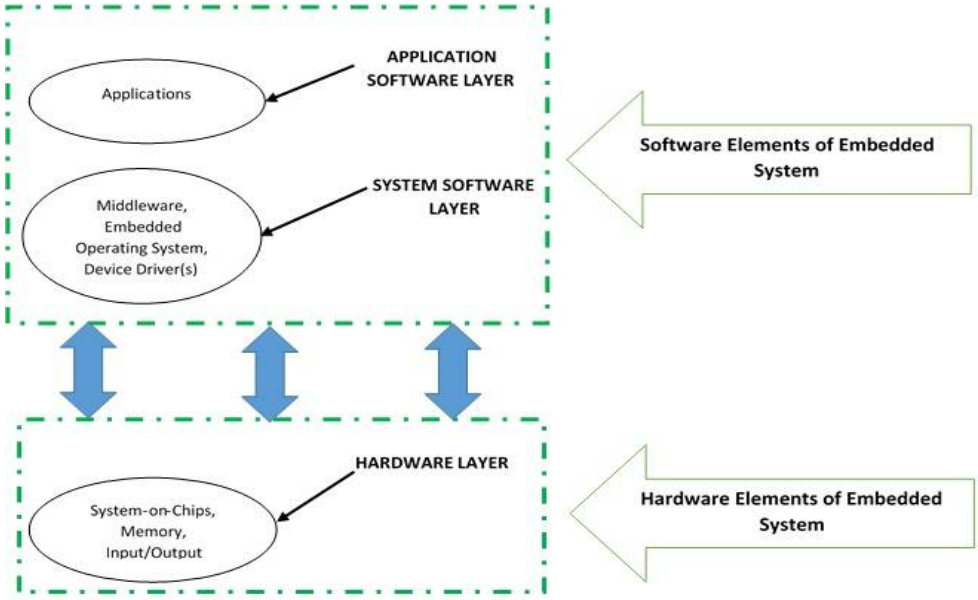


**Fig 1** Architecture of embedded systems

***Corresponding author: Okereke G. E***
Department of Computer Science, Faculty of Physical Sciences, University of Nigeria, Nsukka, Nigeria

A general abstraction of the architecture of embedded systems consisting of the application software layer, system software layer and the hardware layer, where the application software

layer and system software layer make up the software elements, while the hardware layer consists of the hardware elements.

There has indeed been much intensive security study, research breakthroughs and documentation in diverse Information Technology areas including networking and cryptography. However, securing embedded systems have remained a major albatross due to the unique characteristics presented by embedded systems. This problem has been magnified by the increasingly cheaper and ubiquitous Internetand with the advent of the Internet of Things (IoT) (Jayavardhana *et al*, 2013), which transforms Internet communications to a Machine-to-Machine (M2M) basis (Surapon & Panwit, 2015), resulting to every connected object becoming uniquely identifiable (with its status and location becoming public knowledge), eventually leading to a seamless connection between the digital and physical world.

1. Processor: The processing power of embedded systems are typically limited, resulting in a reduced capacity to run typical treat countermeasures such as virus scanners and antivirus systems, intrusion detection system, etc.
2. Power Considerations: A major limitation in embedded systems is restricted power availability. They often operate on small batteries and additional power draining tasks directly results to shorter lifespan. Thus, embedded systems can only commit a limited fraction of its power to provide system security.
3. Physical Issues: Embedded systems are often built to be exposed, used under strenuous conditions and often away from the direct control of the owner/operator (such as in public locations, battle field, etc).
4. Unmanned and remote operation: embedded system are installed in inaccessible locations which are highly dynamic and configurable, thus updates and patches deployment present major challenges.

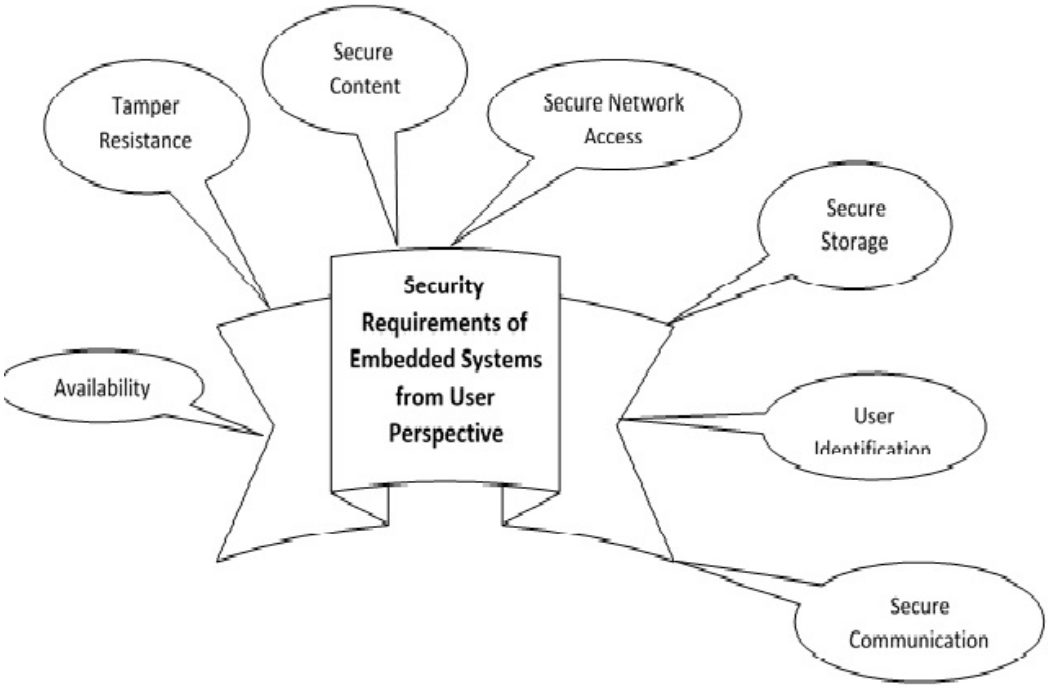


**Fig 2** Embedded Systems Security Requirements: End-User Perspective

Embedded systems have become pervasive, helping to operate everything from home appliances, cars, aircrafts, healthcare equipment and very sensitive mission critical systems. Indeed such technologies are one of the fastest growing sectors in IT, with most electronic devices in the market having a form of embedded systems components (Michael, 2007).

Embedded systems usually access, store and communicate security sensitive data. It is therefore imperative that embedded systems be carefully designed with security as a major factor to consider in other to ensure that they don't end up becoming embedded liabilities.

***Characteristics and Vulnerabilities of Embedded Systems***

Generally speaking, embedded system vulnerabilities are not very different from those in traditional computer systems. These vulnerabilities include weak communication channels, weak password and authentication mechanisms, weak data storage methods, etc. But this vulnerabilities are made more complex to handle due to the unique characteristics of embedded systems (Lee & Seshia, 2011). Some of the major characteristics of embedded systems include;

5. Network connectivity: It has become common place for embedded systems to be connected (wired or wireless). Such connections facilitates updates deployment, remote control, data collection.

These characteristics as enumerated above result in major constraints on both communication and computing capacity of the embedded system, giving rise to a unique set of vulnerabilities, which can be maliciously exploited by an attacker to breach security (Pradeep & Sridhar, 2014). These vulnerabilities include (Sri & Tilman, 2008);

1. Power Exhaustion Attacks: Power resources in embedded systems could easily be drained by targeting its limited battery power. This could be carried out in various ways, including an increase in computational load, sleep cycles reduction, an increase in the use of sensors or increase in communication with various peripherals, etc (Lyes *et al*, 2008).
2. Tampering: embedded systems proximity to a potential attacker could give rise to physical intrusions such as snooping attacks on system bus or power analysis attacks. They are also susceptible to attacks resulting in

confusing and incorrect operation of sensors/peripherals, such as when the calibration of a sensor is tampered with.

3. Network Intrusion Attacks: Networked embedded systems, often comprise of a complex network of components and are vulnerable to the same type of remote exploits that are common for workstations and servers (eg malware attacks and buffer overflow attacks). An attack on any of its network of Components could result in an adverse complex cascade of events in the network.
4. Privacy Issues: Data in an embedded system (such as cryptographic keys or electronic currency on smart cards) is vulnerable to unauthorized access and information theft.
5. Authenticity Issues: Embedded systems are susceptible to malicious introduction of forged information or incorrect data (such as overwriting data in an electricity meter or introducing a wrong video feed in security cameras).
6. Vulnerabilities associated with thermal virus/cooling system failure: Embedded systems operate optimally within specified environmental conditions and are often exposed to potential attacks that cause environmental damage or over heat their system.
7. Reprogramming of systems for other purposes (stealing): A great number of embedded systems are general-purpose computing systems, which are dedicated for a particular use. They are therefore susceptible to unauthorized reprogramming for varied uses such as when a gaming consoles is reprogrammed to run Linux.

A vast range of solutions and approaches have been proposed to mitigate such known vulnerabilities.

***Physical Attacks and Countermeasures***

Algorithms and protocols that have been theoretically proven to be very secure have had their security features defeated by physical attacks (Choden *et al*, 2014). The very nature of embedded systems leaves them very vulnerable to physical attacks such as reverse engineering, counterfeiting attacks (such as cloning and overproduction), micro-probing and eavesdropping. Once under an electron microscope, circuitry of integrated circuits can become exposed using a range of chemicals, and thus become vulnerable to micro-probing. When sensitive information passing between electronic devices become intercepted by an unauthorized recipient, such recipients are said to have eavesdropped.

Device authentication can be effectively used to counter counterfeiting attacks. This is implemented by assigning every legitimately manufactured product a unique ID and getting it registered in a designated security database for authentication and activation (Leest &Tuyls, 2013).

Hardware Trojan attacks, which are described simply as the malicious modification of a design in an untrusted design house/fabrication facility are also a unique set of physical attacks on embedded systems (Narasimhan *et al*, 2012). They are difficult to detect since they very easily bypass common software-implemented defenses. Approaches such as logic testing and side-channel analysis based methods could be used effectively in countering such attacks (Narasimhan *et al*, 2011).

***Logical/Software Attacks and Countermeasures***

Malicious software attacks seem to increase with an increase in the amount of codes used to build a software (CERT, 2005). Majority of software attacks today involve code injection attacks; where malicious codes are injected remotely into the network.

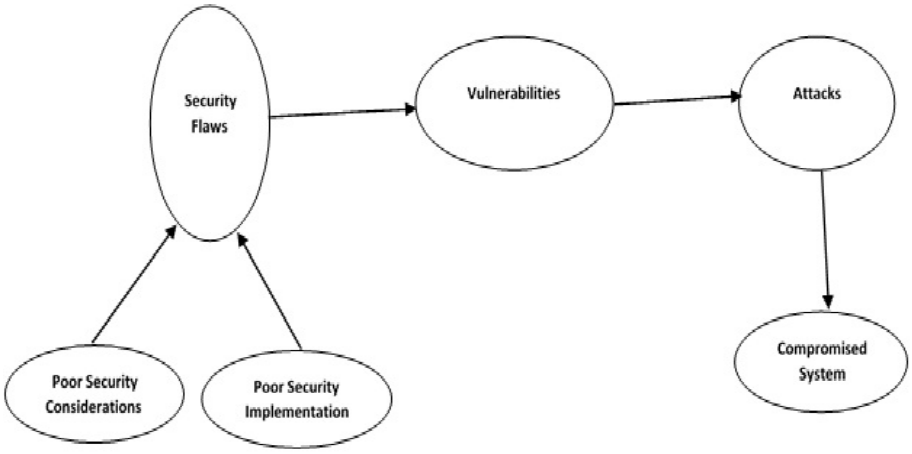


**Fig 3** Progression Of Security Risks In Embedded Systems

***Attacks on Embedded Systems and Countermeasures***

Embedded systems security treats are categorized by the attack means and attack objectives. While attack objectives can include reducing availability, overcoming integrity or preventing privacy, the attack means can be based on side channels, logic attacks (such as cryptographic or software based) or physical attacks (including eavesdropping, reverse engineering and micro-probing). This paper shall restrict itself to exploring security treats based on the means to lunch attacks.

Such malicious codes may seeks to demolish the code integrity of a software program (Milenkovic *et al*, 2005). They do so by changing instructions dynamically with the intent of taking over a programs execution. Another good example of logical attacks are cryptographic attacks, where weaknesses in cryptographic protocols are exploited to carry out attacks (Raghunathan *et al*, 2004). Other software attacks include integer errors, heap-based buffer overflows, and stack-based buffer overflows (Sri &Tilman, 2008).

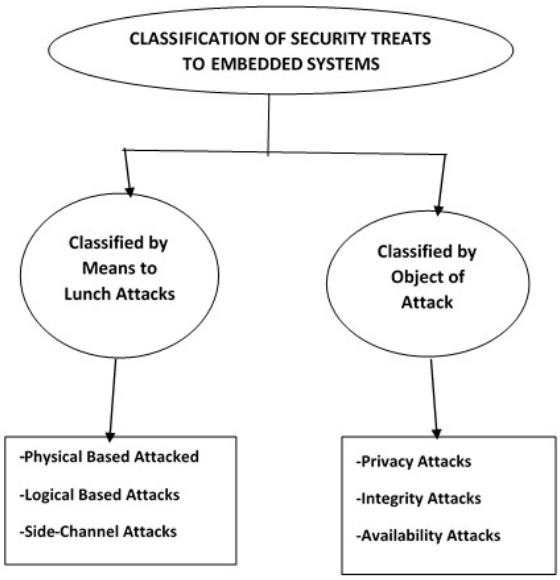


**Fig 4** Classification Of Security Treats To Embedded Systems

Various counter measures against software attacks are prevalent. These include:

1. operating system based measures: Recent operating systems divide their memory processing into code and data processing. The data segment is made non-executable and the code segment is made READ only. This makes it harder for an intruder to inject code into an executing software.
2. Safe programming language library functions: new string functions that are less vulnerable to exploitation are proposed in language libraries such as C and Java. For example, ensuring that strings are always NULL terminated.
3. Behavior anomaly detection: a profile of all permitted software behavior is compared to actual software behavior. A single deviation from permitted profile is expected to trigger an event as a potential security attack. The profile is set up by deciding a threshold value for a number of allowed errors. When this threshold is reached, the anomaly is reported to the system, which terminates the program or declines a given system call.
4. Static code analysis: code analyzers analyze given software, without actually running those software programs. Analysis can either be performed on the object code or source codes. The resultant quality of the analysis depends on whether selected program statements are analyzed or the entire program code. Such analysis can expose vulnerabilities.
5. Architecture based measures: program function return addresses are seen to be most widely attacked targets of buffer overflows. Hardware or architectural assisted counter measures to protect function return addresses exits. Such as hardware assisted runtime monitoring (Mao & Wolf, 2007) and randomized instruction emulation.

***Side Channel Attacks and Countermeasures***

When performing side channel attacks, system properties such as power consumption, electromagnetic emissions and processing interval are observed while operations (such as cryptographic operations) are performed. Information gathered can then be maliciously put to use in successfully perpetrating attacks, without leaving any traces [Mangard *et al* (2007) and YongB (2005)]. Examples of side-channel attacks include electromagnetic analysis attack, timing analysis attacks, power analysis attacks and fault injection attacks (Koeune & Standaert, 2006). In cryptographic attacks, the intruder exploits the weakness in cryptographic protocols used (such as breaching a system by obtaining the right password). Solutions to counter such attacks include using proof-carrying code and installation of runtime monitors for detecting security policy violations. Countermeasures to such attacks include:

1. Masked code execution: Noise can be inserted while running codes to confuse an intruder. Substitution boxes (SBOXes) are also used in cryptology to max execution (Rostovtsev & Shemyakina, 2005).
2. Window method: This is applied in public key crypto systems to avoid power analysis attacks. A modular exponentiation is performed by dividing the said exponent into various window sizes and carrying out the exponentiation by randomly selecting a window (Nedjah, 2007).

**Table 1** Common Attacks And Countermeasures In Embedded Systems

| | Attacks on Embedded Systems | Countermeasures |
|---|---|---|
| 1 | Physical Attacks: | -Device Authentication |
| | -Reverse Engineering | -Logic Testing |
| | -Counterfeiting | -Side-channel Analysis |
| | -Micro probing | |
| | -Eavesdropping | |
| | -Hardware Trojan | |
| 2 | Logical Attacks | -Safe Programming Language |
| | -Code Injection | -Behavior Anomaly Detection |
| | -Cryptographic Attacks | -Static Code Analysis |
| | -Integer Errors | |
| | -Stack-based Overflows | |
| 3 | Side-Channel Attacks | -Runtime Monitors |
| | -Electromagnetic Analysis | -Masked Code Injection |
| | -Timing Analysis | -Signal Suppression Circuits |
| | -Power Analysis | |
| | -Fault Injection | |

3. Inclusion of dummy instructions for random delays: These random delays act to confuse the intruder when attempting to place an attack. However, such random delay countermeasures should be carried out extensively to prevent a successful attack (Aciiçmez *et al*, 2007).
4. Frequent modification of cryptographic algorithms can also be leveraged on to counter side channel attacks. This is necessary in ensuring that power analysis attacks are prevented.
5. Signal suppression circuits are used to reduce the Signal-to-Noise Ratio (SNR), in other to prevent an attacker from deciphering the power profile. Also, software level current balancing can be implemented by source modification and by introducing nops to keep the current constant (Muresan *et al*, 2005).
6. Processor design can be non-deterministic, so that independent instructions are isolated and out-of-order execution performed in a random form by the processor. Also, clock signal could be randomized so that the processor can confuse an attacker in a power analysis attack.
7. Special instructions can be designed to ensure their power signature is difficult to analyses. It could also

be designed to ensure its power consumption is data independent (Tillich & Großschädl, 2007).

***Observations and Findings***

The main objective of a security solution should be to ensure that availability, confidentiality, authentication and data integrity are not compromised during the lifetime of the embedded system (Lyes *et al*, 2008). In other to successfully achieve this aim, security concerns must be carefully considered throughout the products lifetime (from design, manufacture, shipment and sales, operation and to disposal/end of life). Including security planning in the entire life cycle of the embedded device is very critical (Pradeep & Sridhar, 2014).Further, a combination of strategies such as treat modelling, penetration tests and static analysis should be employed.

Light weight cryptography and other innovative security solutions are required for embedded systems. These innovative security solutions should have less computational requirements, be smaller in size and have a lower energy intake.

Furthermore, it is observed that embedded systems are heterogeneous, and may include integral parts such as hardware, software, optics, mechanical components, etc. all these must be carefully considered while designing a security strategy.

In addition, it is observed that manufactures avoid using costly security solutions, having cheap security solutions would make embedded systems more secure, since manufactures would be more willing to adopt the security solutions (Pradeep & Sridhar, 2014). Achieving this would need the combined effort of major Original Equipment Manufacturers (OEMs), Original Design Manufacturers (ODMs) and network carriers.

***Research Trends In Embedded System Security***

In other to overcome challenges emanating from observations/findings above, substantial research and development in the area of embedded systems security has been accomplished over the past decade, with innovative approaches being investigated and proposed in recent years to combat the ever increasing treats to embedded systems. For instance, the potentials of using Physical Layer Security (PHYSEC) for securing key management for embedded systems has been investigated recently (Christian *et al*, 2016). This novel research presented and demonstrated the first ever implementation of an ultra-low power implementation of CRKG (Channel Reciprocity based Key Generation) and CRRD (Channel Reciprocity based Relay Detection) schemes on an 8-bit processor, thus showing the application of PHYSEC for use on resource-constrained devices and platforms. This innovative implementation has proved to be a potential lightweight security solution for embedded devices.
Current research have also explored the use of Direct Sequence Spread Spectrum (DSSS) technology for implementing security in embedded systems. This proposed hybrid architecture uses microprocessor technology to generate short message bursts, utilizing hybrid transmission mediums, thus allowing for efficient and accurate transmission of sensitive data with low Bit Error Rates (BER). This novel implementation would prove very useful in technologies such as emerging autonomous hybrid vehicles (Trevor &Yazdani, 2011). In addition, most security mechanisms for securing communications over the Internet are too heavy for use in embedded systems. A synthesis of different innovative ways to secure communications in embedded systems in a meshed network has been proposed to overcome this challenge (Christine & Jessye, 2014).

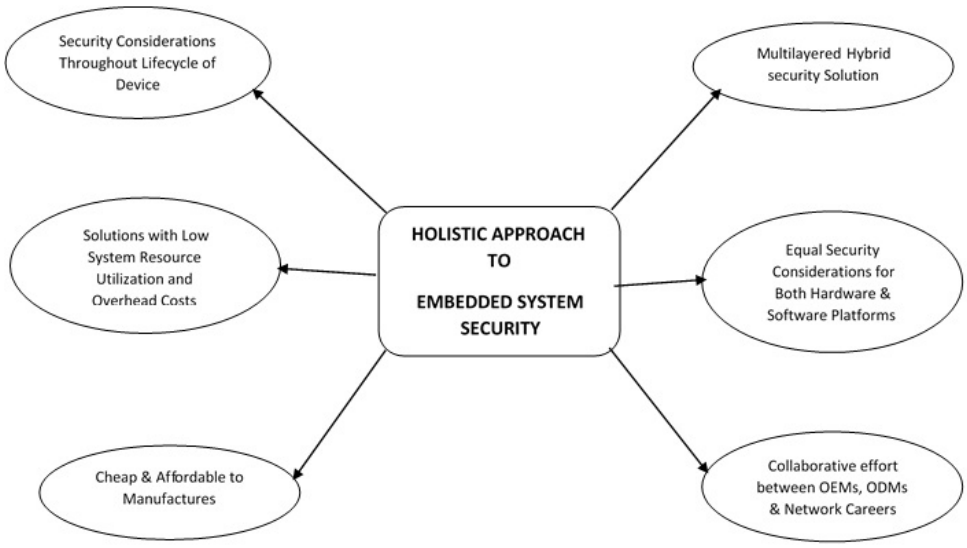


**Fig 5** A Holistic Approach To Security in Embedded System

Another observation is the largely overlooked disposal/end-of-life phase of an embedded system. It is essential to put this into consideration in other to prevent inadvertent release of sensitive data, software or other sensitive information. For instance, secret keys in volatile memory could be extracted long after the device is decommissioned and disconnected from power (Tiri & Verbauwhede, 2004).

In this research for instance, the IPsec protocol suits has been compressed and adapted to the 6LoWPAN, thereby offering security tools for Internet-of-Things devices at the application level. Furthermore, resent research work has resulted in the design of a compact and highly reliable Physical Unclonable Function (PUF) architecture based on Resistive Random Access Memory (RRAM). This innovative PUF cell per bit architecture could be used to securely generate keys without

Error Correction Codes (ECC), thus resulting in a very significant increase in security levels of the PUF. This architecture if implemented in embedded systems, can result in a reliable and highly secure device, with an operational lifetime of up to 10years, under military conditions (Ayush *et al*, 2016). In addition, Choden *et al* (2014) presents a multilevel authentication protocol based on SoP (System of PUFs) that can overcome current vulnerabilities, ensuring breach recognition and recovery. This approach minimizes resource allocation and totally eliminates the need for expensive error correction and power intensive Hash functions. Recent research in rugged and resilient vehicular devices has resulted in a novel design framework for development of high-confidence vehicular control systems that can be securely used in combat environments (Miroslav *et al*, 2013). This proposed framework will ensure that embedded systems will continue functioning securely, in other to guarantee that the vehicle will remain in control and function, while faced with a variety of attacks on its sensors, actuators, communications and computing resources. This research work is set to revolutionize our modern-day battle fields.

Finally, current research work in IP infrastructure in embedded systems has resulted in major innovations in recent times. The Infrastructure IP Security (IIPS) has been proposed for use in embedded systems, which offers functional flexibility, scalability and high security, while incurring an ultra-low hardware overhead (Xinmu *et al*, 2015). This innovative implementation of IIPS provides strong protection for embedded systems against scan-based attacks, counterfeiting attacks and hardware Trojan attacks.

## CONCLUSION

This paper has discussed the most important issues in embedded system security and shown that it is a daunting task to address these issues in embedded systems due largely to their unique set of characteristics, which makes the implementation of general security solutions difficult (if not impossible).Security is constantly evolving and threats constantly change over time. Today, over 90% of all computing devices are classified as embedded systems. This underscores the great need for a comprehensive security approach for embedded systems. However, due to their size, need for mobility and low production cost, they have very limited capacity, thereby making them more vulnerable to attacks.

Finally, the eternal struggle of optimizing the trade-off between resources used and security requirements has made securing embedded systems a very difficult task. A holistic approach is therefore required to tackle the security challenges in embedded devices, beginning from the systems development phase to it disposal/end of life phase.

## References


1. Aciiçmez O, Koç ÇK, Seifert J-P (2007) “On the power of simple branch prediction analysis”. In: ASIACCS ’07: proceedings of the 2nd ACM symposium on information, computer and communications security. ACM, New York, pp 312-320.
2. AyushShrivastava, Pai-Yu Chen, Yu Cao, Shimeng Yu, & ChaitaliChakrabarti (2016) “Design of a reliable RRAM-based PUF for compact hardware security primitives”. 2016 IEEE International Symposium on Circuits and Systems (ISCAS). DOI:10.1109/ISCAS.2016.7539050.
3. CERT Coordination Center (2005) CERT/CC vulnerabilities statistics 1988-2005. CERT Coordination Center.
4. ChodenKonigsmark S. T., Leslie K. Hwang, Deming Chen, Martin D. F. Wong (2014) “System-of-PUFs: Multilevel Security for Embedded Systems”. 2014 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS). DOI: 10.1145/2656075.2656099
5. Christian T. Zenger, Mario Pietersz, ChristofPaar (2016) “Preventing Relay Attacks and Providing Perfect Forward Secrecy using PHYSEC on 8-bit μC”. 2016 IEEE International Conference on Communications Workshops (ICC). DOI: 10.1109/ICCW.2016.7503773.
6. Christine Hennebert and Jessye Dos Santos (2014) “Security Protocols and Privacy Issues into 6LoWPAN Stack: A Synthesis”. IEEE INTERNET OF THINGS JOURNAL, VOL. 1, NO. 5, OCTOBER 2014.
7. JayavardhanaGubbia, RajkumarBuyyab, SlavenMarusica and MarimuthuPalaniswami (2013) “Internet of Things (IoT): A vision, architectural elements, and future directions”. Future Generation Computer Systems. Volume 29, Issue 7, September 2013, Pages 1645-1660.
8. Koeune F, Standaert F-X (2006) “A tutorial on physical security and side-channel attacks”. In: Foundations of security analysis and design III: FOSAD 2004/2005, pp 78–108
9. Lee E.A and Seshia S.A. (2011) “Introduction to Embeded Systems – A Cyber-Physical System Approach”. Lee & Seshia, 2011. ISBN: 0557708575, 9780557708574.
10. Leest V. V. D. and Tuyls P. (2013) “Anti-counterfeiting with hardware intrinsic security,” in Proc. Conf. Des., Autom. *Test Eur.*, 2013, pp. 1137–1142.
11. LyesKhelladi, YacineChallal, AbdelmadjidBouabdallah, NadjibBadache (2008) “On Security Issues in Embedded Systems: Challenges and Solutions”. International Journal of Information and Computer Security, Inderscience, 2008, 2 (2), pp.140-174. <hal-00389976>
12. Mangard S., Oswald E., and Popp T. (2007) “Power Analysis Attacks: Revealing the Secrets of Smart Cards”. New York, NY, USA: Springer 2007.-488, San Diego, CA, June 2007
13. Michael Georg Grasser (2007) "Security improvement in embedded systems via an efficient hardware bound checking architecture", International Journal of Web Information Systems, (2007) Vol. 3 Iss: 1/2, pp.153 - 172. DOI: http://dx.doi.org/10.1108/ 17440080710829270
14. Milenkovic M, Milenkovic A and Jovanov E (2005) “Hardware support for code integrity in embedded processors”. In: CASES ’05: proceedings of the 2005 international conference on compilers, architectures and synthesis for embedded systems. ACM, New York, pp 55-65.
15. MiroslavPajic, Nicola Bezzo, James Weimer, Oleg Sokolsky, Nathan Michael, George J. Pappas, Paulo Tabuada and Insup Lee (2013) “Demo Abstract: Synthesis of Platform-aware Attack-Resilient Vehicular Systems”. 2013 ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS). DOI: 10.1109/ICCPS.2013.6604030.

16. MouaazNahas and Ahmed M. Nahhas (2012) "Ways for Implementing Highly-Predictable Embedded Systems Using Time Triggered Co-Operative (TTC) Architectures". In Embedded Systems – Theory And Design Methodology. InTech 2012. ISBN 978-953-51-0167-3.
17. Muresan R., Vahedi H., Zhanrong Y. and Gregori S. (2005) "Power-smart system-on-chip architecture for embedded cryptosystems". In: CODES+ISSS '05: proceedings of the 3rd IEEE/ACM/IFIP international conference on hardware/software codesign and system synthesis. ACM, New York, pp 184-189.
18. Narasimhan S., Yueh W., Wang X., Mukhopadhyay S., and S. Bhunia (2012) "Improving IC security against Trojan attacks through integration of security monitors," IEEE Des. Test Comput. Special Issue Smart Silicon, vol. 29, no. 5, pp. 37-46, Oct. 2012.
19. Narasimhan S., Wang X., Du D., Chakraborty R.S., and S. Bhunia (2011) "TeSR: A robust temporal self-referencing approach for hardware trojan detection," in Proc. IEEE Symp. Hardware Oriented Trust Security, 2011, pp. 71-74.
20. Nedjah N, de MacedoMourelle L and da Silva RM (2007) "Efficient hardware for modular exponentiation using the sliding-window method". In: ITNG '07: proceedings of the international conference on information technology. IEEE Computer Society, Washington, pp 17-24.
21. Pradeep E. and Sridhar C.H (2014) "Analysis and Security Implementation in Embedded Systems". *International Journal of Innovative Science, Engineering and Technology.* Volume 1, Issue 10, December, 2014. www.ijiset.com.
22. Ravi S, Raghunathan A, Kocher P, Hattangady S (2004) "Security in embedded systems: design challenges". *Trans Embed Comput Syst* 3(3):461-491
23. Rostovtsev A. and Shemyakina O. (2005) "AES side channel attack protection using random isomorphisms". Cryptology ePrint Archive, Report 2005/087.
24. Sri Parameswaran and Tilman Wolf (2008) "Embedded systems security-an overview". *Des Autom Embed Syst* (2008) 12: 173-183. DOI 10.1007/s10617-008-9027-x.
25. SuraponKraijak & PanwitTuwanut (2015) "A Survey On Iot Architectures, Protocols, Applications, Security, Privacy, Real-World Implementation And Future Trends" 11th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM 2015). DOI: 10.1049/cp.2015.0714
26. Tammy Noergaard (2013) "Embedded Systems Architecture: A Comprehensive Guide for Engineers and Programmers". 2nd Edition. Elsevier Inc 2013. ISBN: 978-0-12-382196-6.
27. Tillich S, Großschädl J (2007) "Power-analysis resistant AES implementation with instruction set extensions". In: Paillier P, Verbauwhede I (eds) Proceedings of the 9th international workshop on cryptographic hardware and embedded systems (CHES 2007), Wienna, Austria, September 10-13. Lecture notes in computer science, vol. 4727. Springer, Berlin, pp 303-319.
28. TiriK. And Verbauwhede I. (2004) "A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation," in Proc. Conf. *Des., Autom. Test Eur.*, 2004, p. 10246.
29. Trevor Holden and J.Yazdani (2011) "Hybrid security for hybrid vehicles exploring smart grid technology, powerline and wireless communication". 2011 2nd IEEE PES International Conference and Exhibition on Innovative Smart Grid Technologies. DOI: 10.1109/ISGTEurope.2011.6162746.
30. Xinmu Wang, Yu Zheng, Abhishek Basak, and SwarupBhunia (2015) "IIPS: Infrastructure IP for Secure SoC Design". IEEE Transactions On Computers, VOL. 64, NO. 8, AUGUST 2015.
31. YongB in Zhou DF (2005) "Side-channel attacks: ten years after its publication and the impacts on cryptographic module security testing". *Cryptology* ePrint Archive, 2005/388

f f f f f f f