

e-ISSN: 2582-2993

**Volume 09 Issue 03 Sep-
Dec 2026**

***Corresponding Author:**
Muskan Tahura,
Assistant Professor,
Department of Electronics
and Communication
Engineering, KCT
Engineering College,
Kalaburagi, Karnataka,
India

Submission Date: Apr 21,
2026

Copyright Received Date:
Apr 27, 2026

Security for Physical Component in Embedded System

Afiya Firdouse¹, Asharani², Muskan Tahura^{3*}

^{1,2}*Student, Department of Electronics and Communication
Engineering, KCT Engineering College, Kalaburagi,
Karnataka, India*

³*Assistant Professor, Department of Electronics and
Communication Engineering, KCT Engineering College,
Kalaburagi, Karnataka, India*

ABSTRACT

Embedded systems, for which technological developments appear to occur with a degree of "news almost every day," are becoming more and more important in our daily lives. There is a computer system everywhere, whether it is found in smart household appliances and smartphones or in industrial control systems and automobile electronics. None the less, as people become accustomed to the present, they are calling for stronger security measures to guard against changing online risks. The process of creating secure embedded systems has changed significantly during the last ten years. Since embedded devices were primarily used in isolation, they did not communicate with other devices and were therefore not vulnerable to outside threats. This paradigm was the norm until the emergence of the Internet of Things (IoT), but it has since undergone a radical shift.

Keywords:- Embedded systems, classification, review, survey, power management techniques

INTRODUCTION

Security issues pop up all over embedded systems and to be honest, they're not your typical tech headaches. These devices just don't have the firepower of standard computers. They are on narrow budgets in terms of memory, power and processing and thus expanding software defenses isn't easy. Manufacturers are going on the extra mile by directly adhering specific security chip and functionality to the board.

In that manner, these systems will be able to deal with more modern types of attack when they do so with limited muscle. Individuals are not comfortable in posting their bank account details as well as other sensitive information using systems and avenues they do not believe in and that is holding back e-commerce. Nevertheless, cell phone payment will continue to emerge, and will eventually be embedded into subsequent cell phones to make people actually use such services, the companies must make them feel safe - security features matter.

Many problems are associated with embedded systems. The market is changing so fast that companies must deliver products on time. However, this is not easy. There are many companies there. These companies want to be the ones to get their products on the market. They do not want other companies to come up with the product and be better than them.

The whole thing is really complicated. Companies like these are always trying to stay of other companies. Apart from the above, companies will need to protect their designs and concepts from the public. Embedded system security is one of the issues. The main thing is to fix the software. That is not all that needs to be done. There are things to consider here.

We need to take into account how these embedded systems are designed and how the embedded system shares data. Companies have to make sure that their embedded system security is strong. We have to think about how to keep the data safe in the embedded system.

Embedded devices must deal with information that people must not see. This makes embedded devices very easy for attackers to get into. When people make embedded system, they have to think about a lot of things. They have to think about more than just how big the embedded devices are, how much power they use.

Problem Statement

Embedded systems do not have resources to use strong encryption. This makes embedded systems easy to hack when you have them in your hands. It also makes it easy to attack them in other ways like from far away using the internet. Embedded systems are not secure. They do not have the security they need to keep data safe so people can steal information like secret ideas and get in without permission. We really need a security system that is simple, secure and works well and it has to be based on the hardware of the embedded systems. Embedded systems have to do things because they do not have a lot of resources to work with.

Objectives 1 Secure Data-at-Rest: Making sure the device's sensitive data is encrypted.

- **Secure Data-in-Transit:** Using hardware-accelerated encryption (like AES) to safeguard communication routes.
- **Secure Debug Interfaces:** Preventing unwanted access and reverse

engineering by locking off debug ports (such as JTAG).

- **Establishment of Root of Trust (RoT):** Ensuring that only authorized firmware runs by providing a reliable, unchangeable foundation for the system, usually through secure boot procedures.

LITERATURE SURVEY

Scholarly research [1] and government agencies have long acknowledged the need of security in inserted situations, such as IoT[2].

Ongoing assaults on Internet framework abused weaknesses in IoT gadgets to dispatch appropriated disavowal of-administration assaults, which features the proceeded with the requirement for novel security arrangements in this space[3].

Albeit universally useful working frameworks contain an assortment of security components, inserted OS forms are restricted, and checking for installed working frameworks is obliged. A few procedures are utilized to give working framework security at run-time. Ordinary components incorporate a believed figure base and a reference screen [4].

The product instruments authorize a security strategy and admittance to figure objects, individually. Dynamic data stream security can be functioned to working frameworks to keep information from incoming paths from being utilized as guidelines or bounce objectives [5].

Information driven methodology adds security data to capacity areas and registers to follow safety stages [6]. A later approach evaluates the usage of designs in the processor's programmed counters and series

in accordance with instructions using a neural structure [7]. These boundaries reveal peculiar working framework behavior. To identify shocking changes, the Fingerprint Evident Processor uses bit coins to designate information values. These characteristics are used to identify changes to OS information[8].

Embedded Specific Advancement (2020-2024)

Five key areas are examined in a systematic 2024 review of microcontroller units (MCUs) and constrained platforms: hardware Roots of Trust (RoT) with secure boot, post quantum cryptography (PQC) implementations, Physically Unclonable Functions (PUFs) for authentication, side channel attack mitigations, and Trusted Execution Environments (TEEs).

As these features are integrated into devices like Cortex-M and RISC-V platforms, it observes the shift from niche to baseline requirements. Composing primitives, such as PUF-derived keys flowing into secure boots within a TEE, are recommended by practitioners [9].

Embedded Systems and their Security Issues

Security must be taken into consideration concurrently with the existing restrictions (consumption, size, etc.) if it is a new constraint to be considered in the early phases of building an embedded system. Only by making trade-offs between certain performance factors and the degree of security demanded by the application can design decisions be made. Prior to selecting the optimal compromise between security and performance, a thorough assessment is required because security always has a price. [10]

METHODOLOGY

An embedded system is neither a desktop computer nor a microchip card. This is why it is necessary to propose some security solutions which take account of the specifics of these systems.

Security in dedicated software systems and secured hardware systems is currently a field of interest. It therefore makes sense to take inspiration from these systems when we are thinking about security applications in embedded systems.



Fig. 1: Threats of embedded system.

The principle of deep security (ICTER project) The French National Research Agency's (Agence Nationale de la Recherche) ICTER [ICTER 06] project examines the security possibilities of reconfigurable hardware platforms for embedded systems. Reconfigurable hardware platforms, like FPGA2 circuits, strike a balance between the performance of application-specific integrated circuits (ASIC) (thanks, for instance, to a parallel implementation of algorithms) and the flexibility of microprocessor-based software platforms.

The people taking part in the ICTER project think it is an idea to begin by looking at security from two angles, which are

software and hardware. When it comes to hardware they think we should look at the systems and the physical things that can protect us from attacks. The ICTER project participants believe that this will help us understand how to defend ourselves against all kinds of attacks, on the hardware side of things.

A good example of this is integrated circuits, like Field Programmable Gate Arrays or FPGAs. These have a lot of bits packed into a matrix and they can do lots of things from logic to arithmetic, memory and input/output. All these things are connected by a network of programmable connections, and their settings are stored in SRAM, Flash memory or anti-free components.

Having some basic security is not the same as what the ICTER project's looking for. They want security to be a part of every step of the design process for both hardware and software. When a designer just adds security in one place without thinking about how it affects parts it can create weaknesses that an attacker can use. This can happen in ways using either software or hardware tricks. Field Programmable Gate Arrays or FPGAs can be vulnerable to these kinds of attacks if security is not built in from the start.

It is basically like creating low-power systems, which embedded systems designers are familiar with. You are optimum at one point; you undermine the other; the benefits go by in both ways.

Hardware security solutions: embedded systems well, they're complicated and pretty varied. You've got microprocessors, internal communication systems (think buses or networks on chip), memory for instructions and data, control units, all sorts of I/O, reconfigurable or fixed hardware, and a bunch of peripherals for things like external communication. All these parts can help secure systems and data, but honestly, if someone's clever, they can turn them against their original purpose during an attack.

System level hardware security solutions: At this level we consider the system as a whole. It is at this level that it is possible to permanently analyze the internal and external activity of the system in order to detect any irregular operations. The external activity is measured by sensors (temperature, input power voltage, etc.) and the analysis compares measurements taken in current operation with the corresponding properties expected in normal operation. The system should include a controller to

automatically detect any suspicious external activity. Internal activity is inspected by monitors placed on the internal communications network(s). Data exchange between the various components of the internal architecture of the system is monitored and compared with the activity expected during normal operation. We need to have a thorough understanding of how the system behaves during regular operation, whether we are considering internal or external monitoring.

This is not always easy because normal operation can involve a wide range of behaviours. If the spectrum is too wide, there's a chance that undetected attacks could change how the system functions. If it is too narrow, there could be a chance of false alarms because of typical environmental changes (such a rise in temperature). The system should act deterministically in both situations, meaning that some computations shouldn't happen at random.[12]

Hardware security solutions at architecture level: A single module (microprocessor, hardware accelerator, memory, etc.) is taken into consideration at this stage. These modules' architecture should be adaptable, effective, and fault-tolerant without providing excessive amounts of information channels.

The effective application of algorithms guaranteeing data confidentiality, integrity, and non-repudiation services (asymmetric and symmetric encryption and hash algorithms) has been the subject of numerous studies. In practical terms, there can be a significant gap between an encryption algorithm's software and/or hardware implementation and its mathematical expression.

For instance, an international competition suggested the best software and hardware implementation solution was launched during the development of the AES symmetric encryption standard.

We can make it harder for people to get information from the sides when they try to attack our systems without actually touching them. For example, DPA attacks use how much power something is using to figure out the code that is used to encrypt things. We cannot make this information so clear by adding some noise to the power that is being used.

As we said before DPA attacks work by looking at how the code that is used to make the encrypted text is related to how much power the circuit is using when it is doing this. We can stop this from happening by adding a number to the math that is being done which gets rid of the connection between the two things. DPA attacks use this connection to get the encryption key. If we remove it the DPA attack will not work on the encryption key. The result or the intricacy of the computations are unaffected by this addition.

Hardware security solutions at the logical level-At this level, we consider logic gates (AND, OR, XOR, NOT, etc.). Concerning security, the essential characteristic is to construct gates which do not allow any information at all to escape through side channels.[13]

RESULTS AND DISCUSSION

Security hardware is really important for embedded systems. Internet of Things devices and critical infrastructure are at risk because they can be hurt easily. Security hardware can help Internet of Things devices and critical infrastructure. Using

security hardware has value. Designers have to think about the extra cost of security hardware and the potential problems of security hardware. Designers cannot just add security hardware. Hope that everything will be okay, for Internet of Things devices and critical infrastructure.

To work with the Internet of Things system you need to know a lot, about the components you are using. This includes the hardware and the software. You also need to know about the security requirements of your Internet of Things system and critical infrastructure.

You have to understand what the security hardware can do and how it can help your Internet of Things devices and critical infrastructure. The Internet of Things system needs to be secure, so you need to know about the security hardware. How it works with your Internet of Things devices and critical infrastructure.

Hardware security is not the answer to every problem. However, hardware security has some things about it. When people who design things think about hardware security they have to think about how power it uses if there are any mistakes in the hardware security and if it works with the things we use now. Hardware security is important so people should think about these things when they think about hardware security.

Developing hardware security is a process that takes time. This means it can take longer to get things done. It can cost more money. To get it right designers need to follow some rules. They have to make sure the design is secure. They have to test it a lot. They have to check that the design is good. Designers should always do these things when they are working on hardware

security. Hardware security is very important. Designers should always follow these steps when working on hardware security.

CONCLUSION

Installed frameworks usually do not have safety measures because they do not have enough resources to put security protocols in place. We have made a security solution that uses hardware to make applications and operating system code work properly. This solution can work in different situations and find attacks, including new threats that nobody has seen before.

Our project shows that this way of doing things is a step forward in keeping embedded systems safe in many different situations. Embedded systems can be used in a lot of ways, and our security solution is made to make them safer. The security solution that uses hardware is a way to make sure embedded systems are secure. We have gotten a lot of ideas from people and companies to fix this problem.

The truth is that we are just getting started. Now we need to make platforms more flexible without making them slower and we have to make sure that systems are safer, without spending too much money or using technology that does not work very well. We need to make platforms more flexible.

We need to make systems more secure, and we have to do all of this without making platforms slower or systems more expensive. In terms of tools, we need methods to design items automatically with end, to-end security. Designers need approaches that ensure safety. These are not available yet. We still need people to create design techniques for designers to use.

REFERENCES

1. Pouraghily, A., Wolf, T., & Tessier, R. (2017). Hardware support for embedded operating system security. *IEEE*.
2. Federal Trade Commission. (2015). *Internet of things: Privacy security in a connected world*.
3. Sanger, D. E., & Perlroth, N. (2016, October). A new era of internet attacks powered by everyday devices. *The New York Times*.
4. Jaeger, T. (Ed.). (2008). *Operating system security*. Morgan and Claypool.
5. Suh, G. E., Lee, J. W., Zhang, D., & Devadas, S. (2004). Secure program execution via dynamic information flow tracking. In *Proceedings of the 11th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-XI)* (pp. 85–96). Boston, MA.
6. Vachharajani, N., Bridges, M. J., Chang, J., Rangan, R., Ottoni, G., Blome, J. A., Reis, G. A., Vachharajani, M., & August, D. I. (2004). RIFLE: An architectural framework for user-centric information-flow security. In *Proceedings of the 37th International Symposium on Microarchitecture* (pp. 243–254). Portland, OR.
7. Waksman, A., & Sethumadhavan, S. (2010). Tamper evident microprocessors. In *Proceedings of the IEEE Symposium on Security and Privacy* (pp. 173–188). Oakland, CA.
8. Sekar, R., Bendre, M., Dhurjati, D., & Bollineni, P. (2001). A fast automaton-based method for detecting anomalous program behaviors. In *Proceedings of the IEEE Symposium on Security and Privacy* (pp. 144–155). Oakland, CA.
9. Arora, D., Ravi, S., Raghunathan, A., & Jha, N. K. (2005). Secure embedded processing through hardware-assisted runtime monitoring. In *Proceedings of*

- the Design, Automation and Test in Europe Conference and Exhibition (DATE'05)* (pp. 178–183). Munich, Germany.
10. Hu, S. K., Chandrikakutty, H., Goodman, Z., Tessier, R., & Wolf, T. (2016). Dynamic hardware monitors for network processor protection. *IEEE Transactions on Computers*, 65(3), 860–872.
 11. Thomas, T., Pouraghily, A., Hu, K., Tessier, R., & Wolf, T. (2015). Multi-task support for security-enabled embedded processors. In *Proceedings of the 26th IEEE International Conference on Application-Specific Systems, Architectures and Processors (ASAP)* (pp. 136–143). Toronto, ON, Canada.
 12. Institution of Engineering and Technology (IET). (2005). The celebrated Maroochy water attack. *Computing & Control Engineering Journal*, 16(6), 20–25.
 13. Dagon, D., Martin, T., & Staner, T. Mobile phones as computing devices: The viruses are coming! *IEEE Pervasive Computing*.