Availability and Structure of Security in Embedded Systems

Dr. Afsaneh Minaie, Utah Valley University

Afsaneh Minaie is a professor of Computer Engineering at Utah Valley University. She received her B.S., M.S., and Ph.D. all in Electrical Engineering from University of Oklahoma. Her research interests include gender issues in the academic sciences and engineering fields, Embedded Systems Design, Mobile Computing, Wireless Sensor Networks, Nanotechnology, Data Mining and Databases.

Mr. Jacob Haven Gosling,

Jacob Gosling is a computer engineer working in the private sector. He received his bachelor’s in computer engineering from Utah Valley University. His interests focus on embedded systems, especially embedded security.

Dr. Reza Sanati-Mehrizy, Utah Valley University

Reza Sanati-Mehrizy is a professor of Computer Science Department at Utah Valley University, Orem, Utah. He received his M.S. and Ph.D. in Computer Science from the University of Oklahoma, Norman, Oklahoma. His research focuses on diverse areas such as: Database Design, Data Structures, Artificial Intelligence, Robotics, Computer Aided Manufacturing, Data Mining, Data Warehousing, and Machine Learning.
Availability and Structure of Security in Embedded Systems

Abstract:

With a growing number of cases of information security-breaches, there is a great need for efforts towards securing electronic systems. Embedded systems are ubiquitous systems that are used to capture, store, manipulate, and access sensitive data. The security of these systems poses several unique and interesting security challenges.

This paper focuses on the spectrum of security: the history, the current utilizations, and the future. Throughout the paper, items that are of particular importance to embedded systems are highlighted. This paper also addresses the lack of integration of embedded security educational activities in different undergraduate computing and engineering programs.

Introduction:

An embedded system is an application-specific computer system which is built into a larger mechanical or electrical system, often with real-time computing constraints. An embedded system thus refers to a system that is controlled by a computer that resides within the system\textsuperscript{1}. Ninety-eight percent of all microprocessors manufactured are used as components in embedded systems\textsuperscript{2}. Embedded computing systems are continuously adapted in a wide range of applications such as automotive/transportation, government/military, medical equipment, telecommunications, avionics/aeronautics, aerospace electronics, office automation, data-communication, industrial automation, and consumer electronics\textsuperscript{3}. Embedded systems are responsible for an enormous number of safety and security-critical applications. These systems also manage critical information.

Embedded systems have fascinating security issues with challenging design problems. Embedded systems continue to provide the core for a wide range of applications, from RFIDs to satellites. Efforts to incorporate security into many embedded systems have only recently been initiated. The resource and power dependency of embedded systems continue to be a challenge for state-of-art security practices. There are many unique challenges to building security into embedded devices. Security is an important issue because of the roles of embedded systems in many mission and safety-critical systems. Attacks on cyber systems have proved to cause physical damages.

The internet is an amazing and information-rich utility. Recently, there has come into the spotlight a new benefit and corresponding problem. The Internet of Things (IOT), or simply the need to have all devices connected in the name of convenience. Such items include “smart” thermostats, security systems, refrigerators, light bulbs, the list goes on and on. Almost daily a new item is being introduced to take an aspect of one’s life and integrate it to a smartphone.

This may not seem like a problem but it is not the functionality of the device that is the issue, it is the implementation. These devices have at least one thing in common-- they are small. It is expected that all new devices to be small and have more functionality than their previous
versions. This has only compounded problems with these IOT devices. With the need to keep up with the expectations of their customers (reducing the size and increasing functionality) many manufacturers of these devices have not taken security into consideration or it has not been addressed seriously.

With the decreased size of most devices, there is a side effect that is often forgotten. With a smaller device comes less resources. Using smaller microcontrollers and components means less memory and slower processors. While this wouldn’t be an issue if these devices were themselves not connected to the internet, it is the act of connecting these devices to the internet where the real issues become apparent. Devices with less memory and slower processors are not as good at utilizing traditional security methods. Often, after a device is engineered, security is only an afterthought. This is particularly a problem because security hasn’t been considered and there may not even be any room left on the system memory to run security.

These problems are the reason for millions of devices connected to the internet with little to no security. The possible problems that can occur range from recreational hacking, spying, personal damage, all the way to national security. An example of an issue that occurred recently was the Denial of Service that happened earlier this year. Hacker groups from around the world developed a virus that spread amongst embedded devices and waited for a command. When the command was sent, it resulted in massive internet failures around the United States and Europe. It seems hard to conceive but refrigerators and thermostats were responsible for bringing down large parts of the internet for an extended period.

This paper will briefly consider the history of security, the different techniques used today, and the possibilities for the future. The goal will be to show where security has come from, where embedded security is, the possibilities of the future, and if the embedded security concepts are integrated in computing and engineering programs.

**History:**

The idea of security, specifically encryption, has been around almost as long as mankind. There are many early examples of encrypted messages being sent amongst military leaders in Egypt. That could be considered the most basic example of encryption, simply taking a message and coding with a key that the sender and the receiver share. This method works if the sender and the receiver manage to keep the key secret. This method was by far the most popular seen throughout history. A more recent example of this is the Enigma machine used during World War II by the Germans to encrypt and decrypt their messages. While this was a much more complex version of using an encryption and decryption method, the fundamentals are the same.

With the advent of computers and the need to connect them, an issue was soon discovered that made the traditional methods of encryption no longer viable. Often when devices are communicating over the internet the two users have never met and as such have never exchanged security keys. This was the driving force in the advent of Public Key Exchanges. The basic idea behind the Public Key Exchange is one person posts their public-key online, then anyone who wants to send a message to that person and does not want anyone else to decode it will encrypt their message with the public-key. Then the only person that can decode this information is the
person who has the corresponding private key, which ideally is the person who posted the public side of the key. This method provides a way by which one can be sure that their message is going to the owner of the key and that only they can decode it.

**Encryption Algorithms:**

Many new implementations have been added to this but the fundamentals are the same. Something that has changed are the types of cryptographic algorithms. In the beginning, aka the 70’s, a cryptographic method was developed called the Data Encryption Standard (DES). This encryption algorithm consisted of a 54-bit key using the hardware that was available then. This was more than enough encryption to protect files and communication. Technology quickly advanced and far surpassed the capabilities of the systems that were encrypted with DES. This method is no longer considered an acceptable method of encryption. This is because it would be trivial for a mediocre computer by today’s standards to break this encryption.

With the loss of integrity in the DES algorithm, a variant was conceived that used a three 56-bit numbered key. This method is considered reasonably secure by today’s standard, but it is a pretty resource demanding process to utilize it. While these are only two of the early encryption algorithms, there were a lot that were tried and used in personal and enterprise situations. However, they have been proven not secure or simply not used due to popularity. There are newer algorithms and they will be discussed in the next section.

**Communication Means:**

With the growth, computational ability, and the advent of the personal computer, there became a need to communicate amongst computers. The public key exchange was discussed early to give an example of security on a large scale. This section focuses on the method of communication rather than the algorithms and security schemes. In the beginning, the popular communication method was a protocol known as RS232, or serial communication. This method was not very fast and only worked on a one-on-one basis. As there were more computers available and the need to connect them all grew a new protocol was developed. This means was called Ethernet. When a computer is plugged into a router with a wire it is using the Ethernet protocol. Ethernet devices have a globally unique ID called the Medium Access Control (MAC) address. They also have a local address called the IP address and various TCP ports. The router knows where and to whom to send information with the help of these addresses. The fact that the MAC address is globally unique provides a minor level of security in that there should only be one device with that address.

Ethernet was the predominate method of communication between devices for a few decades. Even today if you’re doing anything that needs speed or security you’re going to opt for Ethernet. Due to the need for convenience there developed a new method. This is the Wireless method, often thought of as Wi-Fi. The wireless protocol IEEE 802.11 standard communication protocol for Wi-Fi, was designed to be as secure, as reliable, and as fast as Ethernet. It literally was designed to be broadcast over the air, but to behave as a wire. Due to this, wireless communication was designed from the bottom up to obtain a level of security that a single wire presents.
Today’s Security Methods:

Today there are many new advancements in the security, encryption, and means of communication. This section will specifically look at the newest forms of encryption, the current techniques for implementing security, and embedded devices specifically.

Current Encryption Methods:

Since the days of DES and 3DES there have been some advancements in cryptography. Specifically, two new cryptographic methods will be discussed in this section: AES and RSA. These both are considered fully secure and good options for new development. These algorithms both come with the added benefit of being simple to implement.

**Advanced Encryption Standard (AES)**

The Advanced Encryption Standard Algorithm was another replacement for DES. AES was the result of a competition put on by the NIST (National Institute of Standards and Technology). Seeing the need for a new, stronger, faster encryption method NIST put together a competition with a reward for a new secure encryption method. The result was obviously AES; it is the standard used by the US Government today. AES is not just newer than its DES predecessor, it is also better suited for today’s computer architectures. This makes it so it runs faster, more seamlessly, and it is easy to implement. One requirement for the new algorithm to qualify for the competition was that it had to be easy to implement in hardware as well as in software. This makes it possible for the encryption to be done completely in its own silicon, as either a part of the main processor chip or as an off-chip option.

**RSA (Rivest-Shamir-Adleman)**

DES and AES are both what is known as a symmetric key algorithm. This means that for the encryption to work both sides must have a key that allows for the message to be decrypted. The process of exchanging this key is not always a simple process. Unless the two parties have physically met in person and exchanged the key, then some form of electronic transfer of this data must happen. To maintain the security of the key this exchange must be encrypted. This poses the issue of how to transfer information that is encrypted in a manner that the receiving party can decrypt it and no other party can. This is where RSA comes into the equation. RSA is simply the initials of the men who patented it in 1978. They were Ronald Rivest, Adi Shamir, and Leonard Aldleman.

RSA is a public key algorithm. It is unique in that it can be used to decode and encode information. This is done by one sending their public key to the person they desire to decode the information. Then they take the information and encrypt it with their private key. The message then can be sent to the recipient to be decoded. When the recipient is ready to send a message back they simply must encrypt their message using the public key and send it to the original sender. This person then can use their private key to decrypt the information. This provides a means to transfer data in an encrypted state without having to have sent the key by non-electronic transfer.
Current Security Implementations:

There are 3 main methods of security implementation that will be reviewed in this section: hardware implementation, software, and a hybrid of both. The benefits and disadvantages of each will be discussed.

Hardware Implementation:

This is an implementation of security modules completely in hardware. These are built into a silicon chip and designed at the gate level up. They could be done as a separate chip that is placed close to the processor, or be implemented into a custom processor chip. The main benefit of choosing this option is speed. With a hardware implementation, the amount of time that the chip could take to run the encryption and decryption would be negligible. The disadvantage to this method comes down to 2 factors, cost and changeability. Building a custom integrated circuit (IC) is not a cheap affair. It can easily run on the order of 100’s of thousands of dollars depending on the complexity and the desired size of the chip. If a custom IC is manufactured, then there is no changing it. So, if requirements change, then the whole development and manufacturing process must start all over again. This is both expensive and time consuming.

Software Implementation:

The next option is the most popular means of managing security-- software. It’s easy to see why software has been and continues to be the most popular method of securing devices. Software is changeable, is easy to implement, and is generally cheap. This makes it a great choice for use in an environment that can change overnight. However, the disadvantages of using software are not so obvious. To run software, it takes processing power. This means that any other software using the processor will be slowed down or even halted. This can be a very serious concern for many applications. To compound this issue, encryption algorithms are not easy things for processors to manage. Encryption takes a lot of processor power, so if you are running it on a system that is low on resources or where the data that is being received is time critical, you may want to consider another option.

Hardware and Software Implementation:

The hybrid of these two options meets somewhere in the middle, utilizing the benefits of hardware as well as software. There are certain parts of encryption that are especially hard on a processor, things like large multipliers. These can be realized in hardware. The processor would simply load the two numbers to be multiplied and wait until result is available. This will not only increase the speed of the multiplication but it will also take the strain off the processor so it can dedicate those resources to other programs. This combination of hardware and software provides a way to benefit from the speed of hardware, and it allows the processor to utilize this extra hardware for things other than just security. It can unload all its multiplication onto the module making the system overall much faster. This way the whole system is benefitted. Most importantly, the parts of encryption that are the most processor intense are generally going to be the same. This means that one should be able to maintain the newest security features without having to manufacture a new chip.
Another possibility that leans more towards changeability and a little less towards speed, is the concept of using an FPGA to run your security either partially or entirely. This provides a completely changeable system. It would be trivial to update the security methods as well as the software to deal with the constantly changing security threats.

**Embedded devices:**

Embedded devices provide a unique set of challenges when it comes to security. These devices are found in all sorts of environments with all sorts of connections. There are a few characteristics that are generally true amongst them all: size, resources, and the need to be connected to the internet. Following is a review of each of these— the reason why they have these traits and why it makes it hard as far as security is concerned.

**Size:**

Embedded devices are traditionally very small. This is because they are meant to fit into a small place. Many embedded devices are replacing traditional devices or being added to them. People expect these devices to fit a certain form factor. Due to this, these devices must meet these size specifications. To illustrate this point, look to the idea of a smart thermostat. People expect a thermostat to fit a certain size on their wall. If you stray too far from what people expect then they won’t purchase it. So, a smart thermostat must fit into the size of a traditional thermostat. This limits the amount and the size of hardware that can be contained within the device.

**Resources:**

Closely related to size is amount of resources found within the device, specifically the amount of memory, the speed and number of processors, and the means of connection. This lack of resources can take a task that is seemingly trivial to a desktop pc and turn it into something that can take seconds for an embedded device. As was addressed previously, encryption is a resource heavy task. This creates a variety of issues with both security and performance and the balancing of the two.

**Need to be connected:**

There is a trend that is quickly moving towards a standard in the world of consumer goods. All electronic devices need to be interconnected and accessible on smart phones, anywhere in the world. This trend has led to the advent of every new device having a connection to the internet and a corresponding mobile app. This provides a great deal of convenience but opens up the consumer to many different types of cyber-attacks.

In the introduction, the importance of embedded security was illustrated with the example of DOS attacks crippling large parts of the internet. There are other threats that having these devices connected to our personal networks can cause. Some of these include privacy issues, security issues, and financial issues. When one wears a smartwatch, or has a smart thermostat, this person is inviting a device that’s sole job is to watch your habits into their life. The smart watch monitors your steps and location and the thermostat monitors your comings and goings. If
criminals could access this information they could determine when you are home and when you’re not home. They could track your exact location. This provides them with an opportunity to attack your house or you. While it’s important to view the big picture of these little devices being able to cripple the internet, it’s important to note that they could be instrumental in exposing you or your family to increased risk.

Now that encryption, its history, its importance, and some methods that are currently being used have been addressed, let’s review some methods that are currently being developed and used specifically for embedded devices.

**Security by Design:**

Security by design is a fascinating idea to implement a security measure, not just in software or as an external design, but as the idea of implementing security at the gate level in a microcontroller. The fundamental idea is that within this processor there will be certain paths that are considered secure paths-- they will only ever be accessed by internal signals of the process. Then there will be unsecure paths. These will be paths that connect to the outside world and will be unsecured. The data on these unsecure paths will be evaluated and if it’s within the bounds of normal operating, then the data will be processed. At no point will any outside data have the ability to manipulate any of the internal processes of the microcontroller.

This method is believed to be a possible solution to the current problem of devices being taken over and manipulated to do things they were never intended to do. Adding these extra paths and gates will increase the size of the microcontroller but not enough that it would make it impossible to utilize them in these applications. Another issue with building a completely new chip design with a completely new data path is that there is a good chance that a custom integrated development environment will be needed to compile and load data onto the device. There may also need to be changes in the methodologies used in designing and writing code for these special devices.

**Security Modules:**

The easiest way to understand what a security module is to think of a foreman at a factory. The foreman’s job is to watch over his crew of men and verify that they are doing the things that they should be doing. He watches to make sure everyone is being safe. He monitors whether everyone is on time regularly. His job is also to make sure that everyone is doing their share of work and being productive. This is essentially the same thing as the security module, only instead of productivity that is being monitored, it’s monitoring the known behaviors and looking for anomalies. The security module looks for behaviors that are out of the ordinary; things like longer than usual function calls, functions using more processor power than usual, etc. The security module tracks how much time tasks take and throws a warning flag if these times are violated.

Another means by which one could monitor a microcontroller is to watch things like I/O lines, temperature and current consumption. It would be fairly easy to get a base line reading on what the average temperature of a chip is. Then, program the security module to monitor the
temperature of the chip and if it goes outside of the acceptable range, then set a flag and warn the system that it may be compromised. The device could monitor communication lines. If they become significantly busier than the base line for an extended period of time, this could indicate a problem as well. Finally, an increase in current consumption would indicate an increase in microcontroller activity. So, if this activity happens out of the ordinary it could also be an indicator of a problem. This could be done with every aspect of the microcontroller that is measurable.

A security module is a very ingenious idea-- the module itself does not add any extra work to the processor and only interrupts the system if there is a very high chance that there is a problem. The fact that you could choose for a variety of symptoms to monitor is also very appealing as a solution. This method seems like the easiest to integrate and easiest to maintain method of increasing security in embedded systems.

Future of Embedded Security:

Now that the history and the current security realm has been discussed, let’s look to the future to see what possibilities lie ahead for security, specifically embedded security.

Quantum Security:

While the name makes it sound like something out of a sci-fi movie, the concept and possibility of quantum security are very real. There are quite a few different applications being proposed for quantum computing and security. It is only a matter of time before the quantum realm is an everyday experience. Quantum computing, while it offers a universe of possibilities for both computing and security based off its unique and amazing characteristics, it also poses a problem for everyone with a traditional system. The problem is that if someone were to have and use a real quantum computer, there would be no way to keep them from cracking all of our traditional security methods. Overnight all the methods that are currently considered to be secure would become superfluous. So, the advent of quantum computing both has its exciting possibilities and terrifying consequences.

The fundamental principle that is utilized in the basic form of quantum cryptology and computing is that of qubits. Qubits are quantum bits. Like in traditional computer’s bits, qubits can be either a 1, 0, or both. The state of being both a 1 and a 0 is known as superposition. This is what makes the theory of quantum computing seem so fast. Theoretically it could process up to $2^n$ calculations at once, n being the number of qubits. This possibility is what scares everyone who is running a traditional computer with traditional security. With this type of computation power, it would be trivial for one of these computers to break trusted encryption methods.

The other fascinating thing that happens with qubit is that if they have ever interacted with another qubit they retain a type of unexplainable connection. This connection provides a way of knowing what both qubits are doing with only having one of them. The other could theoretically be anywhere else in the universe. It is known that when you have a pair of qubits connected, one will be spinning in one direction and the other will be spinning in the other. If you change one of
them to spin in the opposite direction, say with a laser, you change not only the one effected by the laser but the other qubit as well. This phenomenon is known as entanglement. The number of advancements that could be made by utilizing this event are countless\textsuperscript{11}.

As far as security is concerned, having two qubits connected to each other and then giving one to someone else, enables you to have a communication method that is un-hackable. You would simply need to have a system that can both read the spin of the qubit and manipulate the spin of the qubit. Then you could simply send binary values back and forth and have a computer interpret the messages. This is the ultimate in private communications. Luckily however, these methods have for the most part remained in the lab and in theory. While there have been some minor advancements, it turns out that manipulating these particles has proven difficult.

\textbf{Elliptic Curve Cryptography:}

A less fantastical method of security that is in its early stages and has the potential to change the way that we do public key algorithms, is called Elliptic Curve Cryptography (ECC)\textsuperscript{7}. One characteristic of ECC is that it is very easy to generate for the person encrypting, but is very hard for anyone trying to decode the information. This makes it a real candidate for embedded devices. While it may be very simple for a computer to utilize ECC, it is very complicated and detailed in its combination of math functions. Simply put, ECC generates keys through the properties of the elliptical curve equation. This is seen in figure 1. The ECC method can create keys that are 164 bits in length that are as secure as traditional systems with keys 1024 bits in length. It can do this with a lot less computing power. It is safe to say that it is a very secure method to encrypt data. It is utilized in many cryptographic applications to secure data, including bitcoin\textsuperscript{8, 9, 10}.

\[ y^2 = x^3 + ax + b, \]

\textit{Figure 1: Elliptical Curve form}\textsuperscript{14}

\textbf{Embedded Security in Education:}

Embedded systems security is an important issue because of the roles of embedded systems in many mission and safety-critical systems. Within the software and computer engineering fields, the understanding of security is essential to the successful design and construction of modern embedded systems. In order to successfully design and construct modern systems, it is essential that computer engineers and software engineers to understand security very well. Embedded security concerns are multi-dimensional and requires that computing professional understand topics from computing theory, software, and hardware. Even though the embedded systems technology is growing rapidly, the educational activities in the area of security of these systems have not been substantial. There are very few resources specific to embedded systems security which are designed for undergraduate education. The following are examples of how different universities are integrating embedded systems security concepts into their computing and engineering programs:

\textbf{Milwaukee School of Engineering:}
The approach which is taken at the Milwaukee School of Engineering in order to prepare their software engineering students for understanding security is to offer three course sequence in software security. Each of these three courses targets different aspects of security. Following are the three courses that they offer:

- “Introduction to Network Security which offers their students an understanding of the nature of network security.
- Secure Software Development that focuses on the design and construction of software systems in a manner in which security is built into the product from the beginning of development.
- Information Security offers students an understanding of the techniques used to ensure that data and other system information is protected using the most appropriate techniques.”

Both the students and faculty experiences have been very positive.

**Rochester Institute of Technology:**

At Rochester Institute of Technology, they have developed a multidisciplinary curriculum involving three departments which are computer engineering, software engineering, and computer science. Their effort focuses on integrating security into software design and implementations, hardware design and implementations, hardware-software co-design. They use cryptographic applications as the motivating security focus. They have developed a cluster of applied cryptography courses, and created a laboratory with state-of-art field programmable gate array (FPGA) hardware boards and development stations tailored for the study of efficient software, hardware, and combined hardware-software implementations. They offer the following three courses on security:

- Cryptography I- Introduction to Cryptography
- Cryptography II – Advanced Cryptography Algorithms
- Hardware and Software Design for Cryptographic Applications

They have introduced their students to cross-disciplinary areas by successfully adding performance considerations to an existing cryptography course, developed and taught a hardware and software design course. They are working on developing a new course on software security.

**Modules for Embedded Systems Security Project:**

This project has been funded in part by National Science Foundation to improve embedded systems security education for undergraduate students in the computing and engineering programs. They have developed the following eight online modules on embedded systems security:

1. General Introduction to Computer Security
2. Introduction to Cryptography
These eight modules were offered in the following two software engineering courses:

- Embedded Systems Programming
- Software Project in Computer Networks

Offering of these modules in the above courses was successful. They are in the process of refining these modules and adding more modules.

**Conclusion:**

Security is an important requirement in emerging embedded systems, especially considering the connectivity that most of these systems provide to system networks, private networks, or the internet. This paper focused on the spectrum of security: the history, the current utilizations, and the future. Throughout the paper, items that are of particular importance to embedded systems was highlighted.

There are many forms of cryptography, security, and means of implementing them. Methods are still in use that were originally implemented in the 1970’s. Others from that era have long been outdated. One thing is for sure in this changing field, it is important to make sure that every device has a high enough level of security to protect it and to protect others from it if it were to be compromised. There are many methods out there currently to do this. However, many of these are not viable solutions for embedded devices due to resources. A few of them show a lot of promise, namely the security module technique. The future holds a lot of exciting possibilities in the world of security and in the world of computing. The technologies that could be made from the principles utilized in quantum computing alone is mind boggling. The most exciting method of security that the future holds is ECC. This is simply because of how much easier it is on microcontrollers to utilize it and how much stronger the encryption is when compared to other resources. This seems to be the future of embedded security and it can’t come soon enough.

Within the software and computer engineering fields, the understanding of security is essential to the successful construction of modern embedded systems. In order to successfully design and construct modern embedded systems, it is essential that computer and software engineers to understand security very well. The only hope for improving security across the world of embedded systems is to educate the developers. However, the integration of embedded systems security concepts in undergraduate curriculums has not started in many universities. In order to have a secure society, it is crucial to integrate embedded systems security concepts into undergraduate computing and engineering curricula.
References: