A Reconfigurable Arduino Crypto FPGA Shield

Ryan Bornhorst
Gomathy Venkata Krishnan
Meiqi Zhao
Dustin Schnelle

Version 1.1 1/24/18

Table of Content

Project Description

Our team is going to build a reconfigurable Arduino crypto FPGA shield where the Arduino and FPGA shield are computational partners. Our device will use the Arduino platform to configure an FPGA shield that's function is to implement cryptographic algorithms. The FPGA and the software library it uses will contain the implementations of commonly used cryptographic algorithms, like what is supplied in the Sodium library. The device will implement the cryptographic algorithms to guarantee high-assurance. The Arduino will use the FPGA shield as a computational partner. The devices together will insure high-assurance by focusing on formal assurance to provide rigorous confirmation that the generated implementations are correct and secure using programs such as Cryptol and SAW.

Requirements

- The hardware/software cryptography that the FPGA is using will demonstrate “formal assurance” to guarantee that the algorithms used are correct and secure.
- The FPGA shield will utilize cryptographic algorithms to create a hardware crypto on the board using software synthesis tools.
- The FPGA shield can be configured dynamically using an API.
- The system will demonstrate the high performance of four different implementations of commonly used cryptographic algorithms.

Deliverables

A reconfigurable Arduino with a crypto FPGA shield that would securely test 4 different crypto algorithms and efficiently display the security level is the main goal. The Arduino will use a custom API which will provide a user friendly interface for consumers. A custom board design for this prototype will be designed to ensure authenticity and security.

Objective

- Gain expertise in arduino programming
- Gain expertise in FPGA programming especially for a specific FPGA shield
- Design a custom Arduino FPGA board
- Learn to use cryptographic libraries to implement high-assurance algorithms
- Learn to create APIs
- The arduino crypto FPGA shield should be dynamically reconfigurable

Needs

Our project The Reconfigurable Arduino Crypto FPGA Shield focuses on remedying the situation of less security in embedded system through the development of a high-assurance crypto FPGA shield for Arduino. We will set up the development environment generating and uploading code on the FPGA, and be able to interact with the FPGA shield from Arduino. We will develop a variant of Sodium, or a wrapper cryptographic library, for Arduino that automatically leverages an FPGA shield.

Background

Inexpensive open hardware platforms, such as the popular Arduino, have had a huge impact on the embedded systems industry. Especially in the world of the Internet-of-Things (IoT). Arduino's open-source microcontroller platform has become popular to both hobbyist and educators because of its easy-to-use hardware and software. The low cost of the Arduino platform also helps. As Arduino has become more popular in academics, the amount of documentation for the Arduino platform has greatly increased. There are already two course at the University of California that teach Arduino, its programming environment, and interfacing principles for an IoT specialization.

With the massive growth of the IoT the world is changing, allowing for the innovation of new designs and home products. This growth of devices depending on connecting to the internet mean that we need to focus on the security as first class component when communicating sensitive data. It's becoming increasingly necessary to ensure the security of embedded systems connected to the network, so adding cryptography to the functionality is important.

On solution to the problem is using a FPGA's shield to perform the encryption and add to the functionality of the Arduino. The current Arduino FPGA shields on the market however don't guarantee high-assurance of the software, firmware, and hardware. High-assurance is need for the embedded system to provide objective evidence of the system's correctness and security. Usually this evidence comes from a hand-written test bench, but we are looking to obtaining this evidence from formal assurance or what is known as formal methods.

Constraints

- Should be affordable
- Easy to deploy
- Should be secure “use efficient cryptographic algorithms ”
- Should produce results proving system’s correctness

Marketing Requirements

- The shield should design should be compact
- The system should be plug and play
- The system should be low cost while maintaining high functionality
- The system will consume low power
- There will be an online forum that supports the consumers and gives them constant updates about the device and features.
- The system will cater to the requirements of the cybersecurity market and will have features that will be constantly updated.
- Future software updates will remain compatible with the hardware and the firmware used on the system.
- Hardware add-ons will be compatible with all the models.
- Guarantee security over a network.
- Meets health and safety concerns.
- The system will be durable within its intended operating environment.

Engineering Requirements

Performance

- The system should guarantee “formal assurance.”

Functionality

- Arduino and FPGA should be able to interface with each other.
- The system should have a reset (software or hardware) incase it hangs.
- The system should work with at least three different Arduino development boards.
- The system API will be able to use multiple crypto algorithms.
- The FPGA should be chosen in a way that it will be dynamically reconfigurable.

Economic

- Keep cost low while implementing a high performance FPGA.
- The total cost and manufacturing cost shouldn't exceed $50.

Energy

- The system will operate for at least 5V or 3.3V logic.
- The system will function using the embedded system power input.
- The system will operate indefinitely while plugged in.

Health and Safety

- The system will be FCC approved.
- The system should not emit too much heat or radiations.

Legal

- The system should be 100% open source.

Maintainability

- Future software updates will remain compatible with the hardware and the firmware used on the system.

Manufacturability

- The product should be able to easily interface with the arduino headers.
- The custom board should be manufactured with readily available parts.

Operational

- The system should be able to operate in the temperature range of 0^0C to 75^{o}C.
- The system must be able to withstand vibrations of up to 60 Hz with a peak magnitude of 1mm for a period of 1 minute.
- The system should withstand a drop from a height of 6 feet and still operate.
- The system should not emit harmful or dangerous radiations.

Reliability and Availability

- The product will have frequent software updates to ensure that it works at the expected output efficiency.
- The product will be operational 99% of the time.
- The product will always be supported.
- Future hardware or firmware updates will be communicated to the consumers months in advance.

Social and Culture

- The product should contribute a library on the Arduino website.
- Publish announcements on related blogs and forums to gain visibility for the library.

Usability

- User of the system should be able to learn 80% of its functionality within 2 hours.
- The system API should be well documented.
- The API would be user friendly and can be subject to refactoring to enhance functionality at any time.
- Documentation for the hardware, software and firmware will be provided.

Marketing Requirements	**Engineering Requirements**	**Justification**
5,6,9	The system should *ensure high assurance* for correctness and security.	This is based on having a high security platform.
2,8	*Arduino and FPGA should be able to interface* with each other.	This is based on making all of the hardware plug and play.
7,8	The system should have *a reset* (software or hardware) incase it hangs.	This is based on making the system easier to debug.
2,8	The system should work with at *least three different Arduino* development boards.	This is based on wanting the system to be functional across different platforms.
6,7,9	The system API will be able to use *multiple crypto algorithms*.	This is based on ensuring that the system is able to use the appropriate security that is needed.
2,3,8	The FPGA should be chosen in a way that it will be *dynamically reconfigurable*.	This is based on allowing the system API to use multiple crypto algorithms depending on user choice.

3,6	Keep *cost low* while implementing a *high performance FPGA*.	This is based on making the product marketable to a variety of consumers.
3	The total cost and *manufacturing cost* shouldn't exceed $50.	This is based on keeping the entire system at a low cost to the consumer.
4	The system will *operate* for at least 5V or 3.3V logic.	This is based on allowing the system to be compatible with different hardware and maintaining low power consumption.
2,4	The system will function using the embedded system *power input*.	This is based on the system being plug and play.
2	The system will *operate indefinitely* while plugged in.	This is based on the system being plug and play.
10	The system will be *FCC approved*.	This is based on ensuring the system meets government regulations.
10	The system should *not emit* too much *heat or radiations*.	This is based on ensuring the system meets safety regulations.
5,6,7	The system should be *100% open source*.	This is based on allowing users to configure and modify the device how they want.
6,7,8	*Future software updates* will remain *compatible* with the hardware and the firmware used on the system.	This is based on guaranteeing that the system will always be operational as well as allowing future modifications.
1,2,8	The product should be able to *easily interface* with the arduino headers.	This is based on making the system plug and play.
3,8	The custom board should be *manufactured* with readily available parts.	This is based on enhancing product lifetime and maintainability.

2,8,11	The system should be able to operate in the *temperature range* of 0^0C to 75^oC.	This is based on ensuring the system is operational within its intended environment.
11	The system must be able to *withstand vibrations* of up to 60 Hz with a peak magnitude of 1mm for a period of 1 minute.	This is based on ensuring the system is operational within its intended environment.
11	The system should *withstand a drop* from a height of 6 feet and still operate.	This is based on ensuring the system is operational within its intended environment.
10	The system should *not emit* harmful or dangerous radiations.	This is based on ensuring that the system meets safety regulations.
6,7	The product will have frequent *software updates* to ensure that it works at the expected output efficiency.	This is based on enhancing product lifetime and maintainability.
2	The product will be *operational* 99% of the time.	This is based on making the system plug and play.
5,6,7,8	The product will always be *supported*.	This is based on enhancing product lifetime and maintainability.
5,6,7,8	Future hardware or firmware updates will be *communicated* to the consumers months in advance.	This is based on enhancing product lifetime and maintainability.
5,7	The product should *contribute* a library on the Arduino website.	This is based on ensuring the system is open source.
5	*Publish announcements* on related blogs and forums to gain visibility for the library.	This is based on gaining visibility for the product.
2,5	User of the system *should be able to learn* 80% of its *functionality* within 2 hours.	This is based on making the system plug and play.

5,7	The system API should be *well documented*.	This is based on allowing other developers to modify the API to fit their needs as well as make it easier to debug the software for future updates.
5,7	The API would be *user friendly* and can be subject to refactoring to enhance functionality at any time.	This is based on keeping the software open sourced and easy to use.
2,5,7	*Documentation* for the hardware, software and firmware will be provided.	This is based on making the system plug and play as well as easy to maintain.

Marketing Requirements

1. The shield design should be compact
2. The system should be plug and play
3. The system should be low cost while maintaining high functionality
4. The system will consume low power
5. There will be an online forum that supports the consumers and gives them constant updates about the device and features.
6. The system will cater to the requirements of the cybersecurity market and will have features that will be constantly updated.
7. Future software updates will remain compatible with the hardware and the firmware used on the system.
8. Hardware add-ons will be compatible with all the models.
9. Guarantee security over a network.
10. Meets health and safety concerns.
11. The system will be durable within its intended operating environment.

Bibliography

El-Abd, Mohammed. (2017). A Review of Embedded Systems Education in the Arduino Age: Lessons Learned and Future Directions. (p. 79-89). Kuwait City: American University of Kuwait. Retrieved from http://online-journals.org/index.php/i-jep/article/view/6845/4454

Joe and Dan. (2017). A Reconfigurable Arduino Crypto Shield. Portland: Galois
A Reconfigurable Arduino Crypto Shield.pdf