IoT Security Project Requirements

Sean Caster, Vincenzo Piscitello, Adam Barton, Ryan Howerton, and Terri Hewitt

Senior Design

Fall 2018

Abstract

The goal of this project will be to implement a secured, federated Internet of Things (IoT) network with low minimum resource requirements to enable the connection of a wide variety of low-end devices which lack a general purpose operating system. Our objective will be to investigate the amount of security offered and resources required by various cryptosystems in order to provide organizations interested in securing their low-end device networks with a quantified estimate of the costs and benefits. In addition, we hope to accomplish graceful degradation on our network, meaning that the elimination or temporary unavailability of a given node will have minimal negative impact on the inter-connectivity of other nodes. This will encourage the securing of many resource limited devices which play a critical role in meeting a wide variety of human needs.

April 19, 2019

Contents

1 Revisions

Section Number	Original	New
Whole Document	The terms "ad-hoc" and "mesh network"	Changed to the more accurate terms "polymorphic" and "federated network"
4.1	Did not exist in original document.	Added bulleted requirements for quick reference.
4.6	subsubsections that describe the supported technologies	Removed the subsubsections due to being too specific for requirements
4.6	Bluetooth requirements	Removed Bluetooth requirements due to our decision to lower its priority
4.7	Named "System Security"	Renamed "Research Eligible Cryptosystem Requirements" and modularized the requirements such that the research phases were part of the research instead being of within the scope of the code-freeze.
5.4	The usability of the web application will be evaluated by a select group of beta testers outside of the development team. With minimal to no assistance from the team, beta testers should be able to figure out how to create an account, log in, check the status of the network, and find where to add a new node to the network. A minimum of two of the beta testers should be non-Computer Science majors. Any functionality issues reported during testing must be resolved if deemed necessary by either the development team or the client.	Removed.
5.3	Performance requirements listed tests which would be used to evaluate the cost of a cryptosystem on an entire network instead of individual devices	Rewrote the performance requirements for testing the cryptographic primitives and hash functions to reflect changes made to the structure of our tests. Tests were only able to be run on individual devices.
5.6	As a base level verification that our protocols are working, we'll simply ensure that a node given normal conditions is capable of initiating the key exchange phase with the server, authenticating itself during that exchange, and finally sending and receiving encrypted messages before ensuring that the decryption process produces the original encrypted messages. As a more advanced verification, our group will perform various holistic verifications of system security against different active attacks, such as partial DoS, MitM attempts, and attempts to clone a node.	[Section Rewritten. See 4.6]
7	Old Ghant Chart referenced Web Application and showed work starting in September.	Updated to more closely resemble actual timeline.

2 Introduction

2.1 System purpose

As it stands, a wide variety of mission critical tasks for many utility grids (water, power, etc.) are accomplished by low power embedded devices. By in large information security appears to be an afterthought in many internet connected devices thus creating significant opportunities for malicious persons to compromise these systems. Given that interconnected devices have become ubiquitous with modern life, the effects of a compromised system can be far reaching. The outcomes can range from the invasion of industrial security to the loss of life. Our project takes a look at the state of Internet of Things (IoT) security as we implement our own network. To solve these issues we will implement a federated infrastructure of which we will use to study the trade-offs of security and power consumption. Our project, Brypt, aims to provide a solution for architects of these networks to have the best of both worlds. In minimizing the cost of power while maximizing the security of IoT networks will demonstrate numerous benefits.

In addition to offering an implementation of what our investigations suggest to be an excellent general purpose Internet of Things cryptosystem, we will generate useful data about the costs of resource asymmetric cryptosystems that can be applied to the design of more special purpose IoT networks by organizations looking to secure their particular resource distribution optimally. These findings will be published in a final report alongside the implemented network to help in the potential design of such future projects.

2.2 System scope

The ultimate goal of this project will be to produce the implementation and verification of the secure networking paradigm succinctly named Brypt. Brypt combines the main cornerstones of the software, bridging and encrypting. The implementation will provide a central server, binaries for node endpoints, and a security protocol working on top of the application layer. In completing the work on this project our team aims to provide a novel solution to security in interconnected networks of embedded and general purpose devices. Some of the benefits Brypt will provide include the increased viability of secure cryptographic primitives in resource constrained environments across several of different communication technologies.

2.3 System overview

2.3.1 System context

The primary context of our system will be provided through a cloud based application served through our central server. Within our web application three primary interfaces will be accessible by the user. The first interface will be the base information page for Brypt; this page will contain information about the system, it's requirements, and downloads. The second interface will be the node management screen; these pages will provide provide node network authorization. The final user interface element of our system is the dashboard page which will contain information pertaining to the status of the user's connected clusters and aggregation of the nodal data. The only requirement of the user will be access to a browser capable device. Outside of our application interface, users will need to interact with our devices and/or binaries for their system of choice.

2.3.2 System functions

Our system capabilities will be provided through our central server, networking implementation, and security protocol. The complete system will need to operate over several communication standards, variable weather conditions, and battery constrained environments. An active portal to a full internet connection will be required for communication between the federated network and the central server.

2.3.3 User characteristics

There is a single user class within our system; this permissions class will act as the manager and maintainer of their organization's clusters. There can be multiple users with access to an association's clusters, although there may be one owner. An individual wanting to use our system and application should be familiar with technology, but may come from all different walks of life. We can expect users to have varying levels education and disability, so accessibility should be built into the user interface.

2.4 Definitions

- Control Server: The central server for a network.
- Coordinator: A node on a user's network with the added responsibility of routing the aggregate messages of the cluster and/or acting as a gateway for devices without direct internet access.
- Node: An endpoint in the network.

3 Specific Requirements

3.1 Summary

1. A central server hosted remotely.
2. Central server is RESTful oriented with basic CRUD (Create, Read, Update, Delete) commands to manipulate a database.
3. A website hosted remotely to provide basic project information.
4. A node network operating over at least one communication technology (e.g. WiFi, LoRa, BLE).
5. A network has one root coordinator that manages connections with its peers.
6. The network can respond to node information and sensor query requests. The responses will be made up of real and arbitrary values.
7. The network nodes have end-to-end encryption/decryption from themselves to the client facilitated by the codebase developed. (Need cryptoprimitive library and memory available, but don't need to have a specific phase up and running)
8. A user interface supported by a minimum of one operating system.
9. The user interface can query the central server to get network data in order to authenticate the user, find the network, and get basic registered node information.
10. The user interface can connect to the root coordinator.
11. The user interface can send requests to the root coordinator.
12. The user interface can display connected peers of the root coordinator.
13. The user interface can cycle network reading requests and parse them into a chart.

3.2 Functional Requirements

The central server acting as a network archive will operate on a remote host. The remote server will perform verification and authorization of the network components. The node architecture will be structured based off a multi-hop federated network. A new node will perform a series of acknowledgements to be configured into the topology. During the configuration process the node will generate a data structure to store neighbor information. The neighbor table will be used to communicate with the connected peers it has access to. Utilizing the information of its cluster, the node will be able to perform its base operation and listen for failed peers. The system's end-to-end security protocol will have the ability to encrypt and decrypt messages on each endpoint.

3.3 Usability Requirements

The system should have an interactive application which permits the user to connect to the access point, send network requests, and monitor aggregate sensor readings from multiple endpoints. The website portion of the project will support desktop versions of the Google Chrome Web browser. The website will serve to provide basic information about the project including, but not limited to problem statement, team members, and links to the source code. The desktop application should support at a minimum one operating system. Authenticated users should have access to the application and use it to monitor the state of the networked devices. The state of the network refers to connected devices and transmitted messages sent and received by the client. The system should tolerate invalid login credentials and provide feedback on access success or failure.

3.4 System Interface

There must be an implemented interface that allows for creation of user accounts. The implementation of the user account will give the users options to monitor their network. This interface should consist of a "dashboard" allowing for users to view the status of their networked devices.

3.5 System Modes and States

Each networked device will be in a connected or disconnected state in terms of the network and user interface. The network will only be aware of devices that have gone through the the connection handshake for the current life-cycle. A life-cycle in terms of the network is the period of time that a root coordinator has been started to the time it shuts down. The user interface will display connected nodes at the time it connects as a peer to the root coordinator while disconnected/powered off devices are hidden.

3.6 Physical Characteristics

This network will consist of devices using multiple communication types. Devices will be networked in a federated topology that may be designated by the user with the possibility of self-configuration addable at later date. Clusters should be capable of communicating over, at a minimum, WiFi. Each cluster will be connected with other clusters through network coordinators to provide inter-communication. Examples of communication types are LoRa, WiFi (802.11), and BLE. The client will have higher computational resources than many of the networked-devices. Networked devices will have varying computational resources as would be implemented in the real world.

3.7 Research Eligible Cryptosystem Requirements

While the deliverable will simply be the network codebase itself, capable of facilitating a wide variety of possible cryptosystems, overarching requirements for cryptosystems we'd like to investigate during our post-code-freeze research are outlined below.

Authentication A new node must be able to authenticate its messages to a server in any cryptosystem we implement with a given key or key pair. In systems which perform aggregate authentication, a fallback, non-aggregate, authentication technique should be available. This will help (A) determine which in a group of messages is malicious/inauthentic given failed verification and (B) avoid high-cost aggregate verifications when better-equipped base stations are unavailable.

Privacy Encryption technique should not allow for compromise of privacy, even to a Byzantine adversary. This means, not only should our system use AES-CTR correctly (i.e. increment nonces and pad correctly), it should also be resilient to things such as state-induced Key Reinstallation attacks and Man-in-the-Middle attacks (this will also depend upon our Key exchange and authentication requirements being met). Cryptosystem as a whole should maintain privacy, as opposed to only encryption functions used. This means we'll use Encrypt-then-Authenticate to prevent Authentication techniques from leaking data over multiple key uses.

DOS attacks:
System will degrade gracefully in event of Denial of Service (malicious, or natural) meaning that if a node loses its normal bridge to a coordinator node, it will broadcast to search for other nodes able to serve as a new bridge. This program state should be under careful scrutiny, since adaptive response to possibly malicious activity opens a large attack surface to MitM attacks

MitM (Man in the Middle) attacks:
System should be able to use a single (possibly unique) fingerprint for each node in order to prevent MitM during Key Agreement. Once Key Agreement has been accomplished, system should continue to authenticate communication so as to prevent any control message from altering its behavior without a signature.

Rogue Node Attacks:
Cryptosystem must resist imitation of legitimate nodes by avoiding reuse of keystreams (i.e. any key, nonce pair in an AES-CTR encryption must be used only once). System will not compromise privacy to Rogue Nodes under any circumstance, and given a DoS-oriented Rogue Node, will provide administrator alert mechanism for nodes

not reporting as expected to client.

3.8 Information Management

Information about the nodes, users, and network clusters should be stored in a database accessible to the application. Any personally identifiable information stored in the database for each user should be kept to the bare minimum required to identify and contact that user.

Not all documents will have values associated with each field. Information stored about the separate network clusters may vary from one cluster to the other. Different network groups may have fields unique to a specific group. The database must be capable of handling some diversity in the information stored for objects in the same collection.

3.9 Policies and regulations

As outlined above, the network will be communicating on various radio frequencies. We will be staying away from those frequencies that are highly regulated, such as the ISM (industrial, scientific, and medical) bands. Radio frequencies have been divided between industries by various organizations around the world, such as the U.S. Federal Communications Commission and the International Telecommunication Union. These organizations place rules on, require licensing for the use of, and carefully coordinate specific radio frequencies, so that there is no interference between signals on these bands.

We will be employing technologies that communicate on frequencies outside and between regulated or restricted ranges (such as Bluetooth and Wi-Fi).

4 Verification

4.1 Functional Requirements

Each aspect of the project will be tested individually and as a system. The central routing authority will be tested to ensure that it reliably queried. Our networking implementation will be tested to ensure proper topology is handled with the changing state as well as the transmission of messages. The security protocol tests will ensure encryption and decryption is working properly for production environments. Testing along the way as well as system tests will verify that this functionality works properly.

4.2 Usability Requirements

On completion of the application, a brief 1-2 day period of system testing testing should be performed to ensure the cloud based portion of the system works as expected with less than a 5% rate of failure.

4.3 Performance Requirements

Tests should be written to assist in a cursory evaluation of power draw (measured in Watts) of each cryptographic primitive and hash function planned to be used in the network. The tests should be performed over a minimum period of three days. Results from the tests should be recorded in a spreadsheet listing the date the test was performed, the voltage and current draw for each type of device which will be connected to the network (ex. If three M0 Featherboards are to be used in the network, only one M0 Featherboard should be used as a representative during testing), and the calculated wattage for each primitive and hash function run on the device.

4.4 System Modes and States

Verification of system modes and states will be implied through the functional network. Tests for adding, connecting, and removing devices to stress the functionality of connection will be done to ensure that system modes and states are working as intended.

4.5 Physical Characteristics

Each device will be tested by authenticating, connecting, operating on the network, and removing from the network. Standard operation of device lifecycles will be stressed to ensure that the network can queue and handle requests. Devices of differing communication types will all be tested in the same way to verify that all communication types succeed in connecting.

4.6 System Security

The verification that the network has what's needed to provide a platform for our security research will be very simple. One only need verify that the messaging system is capable of storing 1-2KB of cryptographic information (keys, key-components, nonces, etc.) on each of our low-end devices, logical control over which coordinators the feathers are using, client-side software capable of storing the same information, and a server which performs user-authentication. As a matter of good data-stewardship, the server will employ the bcrypt password hasing function to all user passwords, so as to protect them in the case of a server breach.

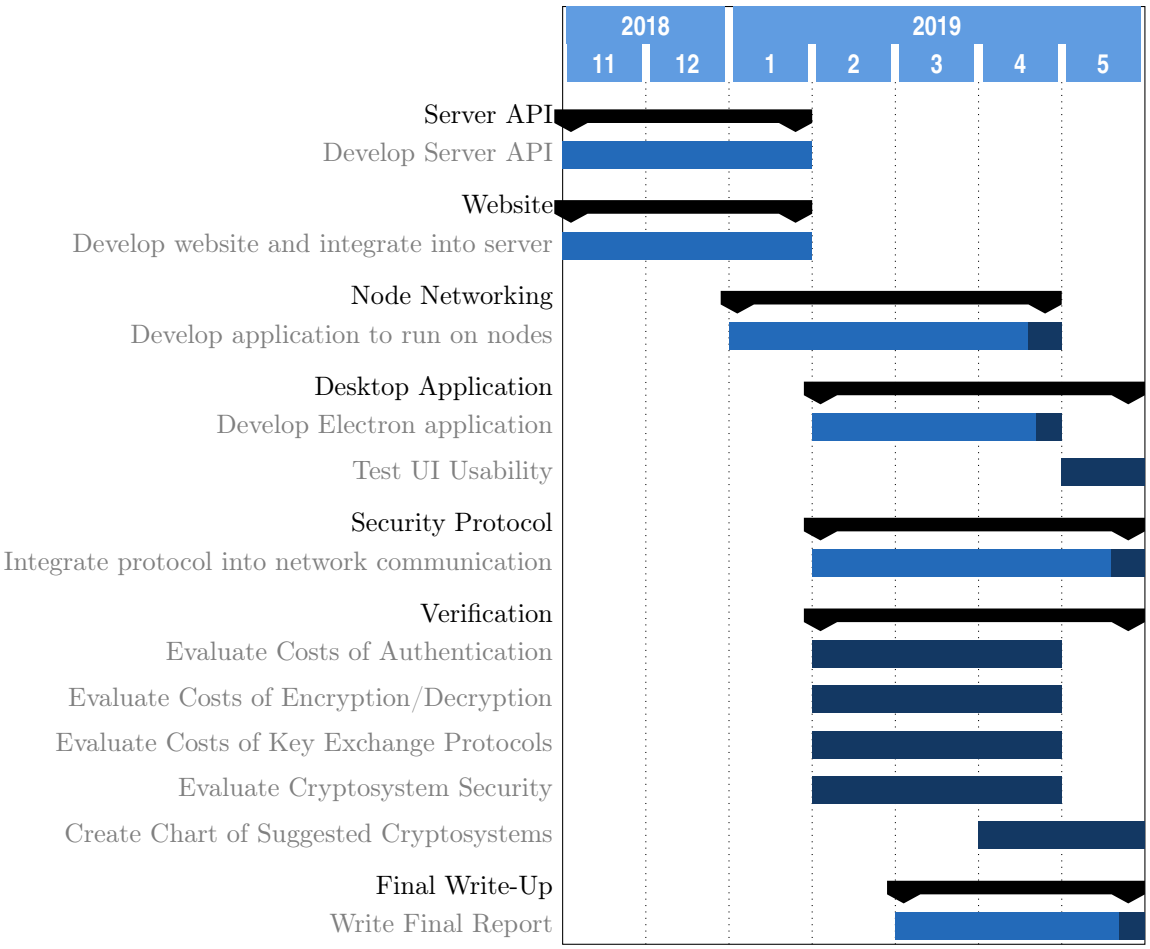
4.7 Information Management

The database(s) will be tested to ensure that information is correctly stored and retrieved by the server. Tests should be created to ensure that data cannot be modified during the process of storing it in the database or requesting it from the database.

4.8 Policies and Regulations

Each node will only be able to communicate on frequencies that are unregulated, as defined by their hardware. We will also test the reception capabilities of the nodes to ensure that there is no interference from outside sources, and that within their natural limits they are able to send and receive to other nodes on the network.

5 Project Timeline



6 Appendices

6.1 Acronyms and Abbreviations

- AES: Advanced Encryption Standard
- API: Application Program Interface
- BLE: Bluetooth Low Energy
- DDoS: Distributed Denial of Service
- DoS: Denial of Service
- FAAS: Fast Authentication with Aggregate Signatures
- MitM: Man in the Middle
- SQL: Structured Query Language
- UI: User Interface
- URL: Universal Resource Locator
- UUID: Universally Unique Identifier