STRATH NAME

DOCTORAL THESIS

Thesis Title

Author:
John SMITH
Examiner:
Dr Man Page

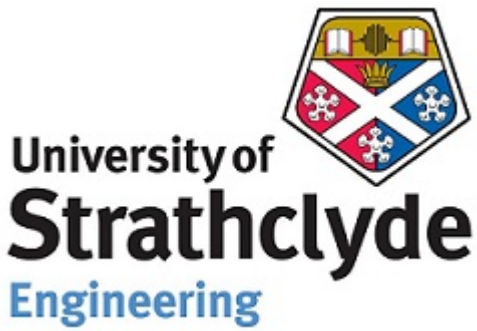
Supervisor:
Dr. James SMITH
Phd Student:
Greg

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

in the

Research Group Name
Department or School Name

March 10, 2016



Declaration of Authorship

I, John SMITH, declare that this thesis titled, "Thesis Title" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

STRATH NAME

Abstract

Faculty Name
Department or School Name

Doctor of Philosophy

Thesis Title

by John SMITH

The Thesis Abstract is written here (and usually kept to just this page). The page is kept centered vertically so can expand into the blank space above the title too…

Contents

Chapter 1

Introduction

The Internet of Things or IoT is the concept of a huge network of physical objects connected and communicating to themselves and to the world wide web. Devices can include domestic appliances, buildings, cars. As it becomes a rapidly growing concept with over 50 million devices expected to be connected to the web by 2020, (need ref) the security of the transmissions of these devices is becoming a more and more pressing issue. IoT's main benefits are the remote control of devices and appliances, for the device to have the ability to send information about it's state, such as a vending machine reporting that it has run out of a certain item, and to allow the machines to be more automated and to work with other machines, like a home hub device that can turn on the lights and central heating when an occupant is arriving home, with the lights and heating not being connected to each other but to the central hub.

However IoT will be ultimately be useless if it is unsecure. IoT is an emerging field but there have already been some high profile security disasters. Ranging from relatively less serious problems such as some "hackers" been able to glean important wfif information from your internet connected lights**[1]** to very concerning, potentially fatal security breaches like someone gaining unauthorised access to your car and assuming control. There have been three examples of this with a Jeep Cherokee, Toyota Prius and Tesla. The hackers were able to control the accelerator, door locks and brakes, among other things. This highlights a very real problem that will only become more important. Too often security is an afterthought but it really needs to be built into products from the offset.

Within the last three years there have been three high profile security breaches on commercial cars, one on a Cheroke Jeep **jeephack** a Toyota Prius **priushack** and a tesla **teslahack**

With that in mind the subject of this report is the secure transmission of a users private home temperature data. If they have a system that monitors the temperature of all the rooms, that data can be used to figure out when they are likely to be home or not. So, using an Arduino Due as the base station that talks to the temperature sensors throughout the house, it takes the sensor data signs then encrypts it and sends it using an Ethernet Shield to a remote server.

Chapter 2

Background

This chapter will briefly explain the cryptologys used to make the application secure and what kind of attacks this can protect against.

2.1 Cryptography

2.1.1 TweetNaCl

2.1.2 Public Key

2.1.3 Signature

2.1.4 Authenticated Encryption

2.2 Technollogies used

php/java/C++/SQL

Chapter 3

Design

3.1 IoT Platform

The basic concept of this platform is an Arduino Due that takes the current temperature of the room from a DS1820 temperature sensor. Then that data is signed and encrypted with TweetNaCL before being transmitted, using an Ethernet Shield, across to an SQL server. A web application takes the SQL data decrypts, checks the signature is valid then displays on a website.

graphic here pls

Why was the Due chosen, 32 bit?

DS1820 is a lost cost temperature sensor that is very accurate, 12 bits of precision? and is also low power. It can scavenge power from the data with the arduino and thus does not need it's own power source.

For the prototype, an Ethernet Shield was used as it is much cheaper than a WiFi shield but ultimately completes the same job. The shield is a simple way to connect arduinos to the internet. The shield used was the second revision, R2 and has a w500 ethernet controller.

What are some of the options for the base station, Due/MSP430? And for the internet connection Ethernet?WiFi shield?

3.2 Server Side

For the prototype, an Apache server, SQL server and Tomcat server was set up using XAMPP on a desktop. A Java web app was created as that is the language the writer has the most experience in and there are Java implementations of the TweetNaCl library, among other variations, available. The SQL table is a simple table that holds a key, timestamp and the signed and encrypted temperature sensor. (example?). The web app upon being accessed decrypts and checks the signature of each entry, using the keys that it has stored, in the table before converting the raw hex temperature data into more readable integers and displaying in a simple HTML table that can be accessed by the user. When the Arduino has data to send it will make a POST request to a PHP file on the Apache server which takes the data given to it and places it in the SQL server. (security flaw!)

3.2.1 XAMPP

explain what XAMPP, Apache, SQL and Tomcat are. (maybe should be in background) Xampp is a

3.2.2 Java web app

The Java Web app details what the server is do when it gets various types of request, be it get or post...(more on requests?). In this type of application you can dynamically printout all the HTML that will be used to make up the page. The usual HTML, head, body tags are printed at the top and the titles in the table are printed as well. (How it gets the keys?!). The web app uses JDBC to create a driver(idk man) to get the connection to the SQL database. Then using Java language it builds up a SQL query to take out all the values from the database and executes that. This puts all the table entries into a result set and the app cycles through that results set getting the relevant information out. The signed and encrypted hex is encoded as string and some leading zeros are lost in the conversion from byte array to string in the Arduino so these are added now before the string is converted back into a byte array. There is a try catch around the crypto_box_open and crypto_sign_open method so the server doesn't crash if one result set has been broken. Following this is the conversion from hex into integer for the user to read (how does it do it?) and finally the values are written to the browser along with the ending html tags.

3.2.3 PHP

PHP is a server-side scripting language

There are two files in the server, connect.php and add.php. The Arduino makes a post request to the add.php which effectively just calls it and the first thing the add file does is call connect which has the server details and makes creates a connection. Following that there is a SQL query that inserts the values sent in the post request into the appropriate table entries then close the connection.

3.3 NaCl

The keypair, crypto_sign, crypto_box and equivalent opens were used. These are simple to use, abstracted methods that make this library easy to use. For the encryption the method needs the message to be encrypted which needs to have the first 32 bytes be zero, an empty array that needs to be at least the size of the message with the leading zeros, the length of the message, the nonce, arduino public key and the servers private key. This will reveal the temperature data with the signature. To remove the signature, the crypto_sign_open method needs the server secret signature key and the signed cipher array.

Chapter 4

Strength Of Security

4.1 SubSection

4.2 Other SubSection

Chapter 5

Results

5.1 Power Consumption!

5.2 Other SubSection

Chapter 6

Critical Evaluation

6.1 SubSection

6.2 Other SubSection

Chapter 7

Conclusion

7.1 SubSection

7.2 Other SubSection

Appendix A

Appendix Title Here

Write your Appendix content here.

Bibliography

[1] Leslie Lamport, *LaTeX: a document preparation system*, Addison Wesley, Massachusetts, 2nd edition, 1994.