CS CAPSTONE TECHNOLOGY REVIEW

NOVEMBER 21, 2017

SMART HOME INTERCOM SYSTEM

PREPARED FOR

OREGON STATE EECS

D. KEVIN MCGRATH

PREPARED BY

GROUP 25

LAZAR SHARIPOFF

Abstract

The technology for the Smart Home Intercom system are reviewed and explained in this document. This will include a detailed description of three different tools for each of the three technology pieces, a compare and contrast of each of the tools in each tech piece, and an explanation of why we chose a certain tool in a given technology.

CONTENTS

1 VIDEO DISPLAY/CAMERA SOFTWARE

1.1 Overview

The Smart Home Intercom system is required to have a video call feature that can stream live video feed from one node to another. This section covers the possible choices of video streaming software and libraries that are available for the Raspberry Pi 3, and will go over the pros and cons of each of the options.

1.2 Criteria

This tech was chosen because the system needs to reliably stream video feed that is clear enough to be able to distinguish who the user is talking to. If the software that is used to stream the video is poorly optimized, or uses a video codec that consumes a large bandwidth the video quality may become grainy or even drop the connection between the nodes so that the video call is cut short.

1.3 Potential Choices

1.3.1 RPi-Cam-Web-Interface

RPi-Cam-Web-Interface (RPiCWI) is an open source web interface video streaming software that has a wide variety of applications like, streaming, surveillance, dvr recording, and time lapse photography. The interface is able to be opened in any browser, and can be extended using macro scripts. It encodes the streamed video in H.264 and is able to change the video quality with setting from 0-100, however the setting is not the same a tradition jpeg quality values as the RPi camera compression quality is not linear. [1]

1.3.2 Stream-m

Stream-m is an open source working prototype streaming software that can stream live video using the HTML5 video tag and Google's WebM, or standard H.264. WebM can be streamed right in the browser using a simple Http interface, support for video playback of the stream requires a an extra API. If the video is instead in H.264 format then an RMTP server is required, however video playback over RMTP is not currently supported with this software. [2]

1.3.3 VLC

VLC is an open source cross-platform multimedia player and framework that can play most media files and supports various streaming protocols. It is able to encode video in multiple formats like H.264, H.265, and MPEG-4. VLC also allows video playback from most types of encoded files, whether they are video, audio, or text files without the need for external codec packs making this software extremely versatile and easy to use.[3]

1.4 Discussion

RPiCWI is a good choice for streaming video because it is lightweight and its ability to be opened in any web browser to playback a stream is great, compared to Stream-m where only WebM playback is supported. However, the UI of RPiCWI seems pretty clunky and cluttered, and doesn't look like it can be customized easily to tailor for our need. Stream-m is extremely lightweight and allows an easy way to stream with HTML5 and WebM, but it's in prototype stage and is not currently being supported by the developers. VLC is a great choice for streaming video because of how powerful it is with streaming and video playback without the need for external codecs. Since VLC being as powerful as it is could also be a drawback, because it could mean that it has lots of built in codecs we don't need making it "heavier" than the other streaming tools.

1.5 Conclusion

We chose VLC because of it's power and customize-ability. It is able so playback multiple types of video, and is also able to stream multiple types of encoded video allows us to have the freedom to see what video encoding can work best for us without needing to install external codecs or change streaming tools makes it extremely versatile. It might be ”heavier” than the other streaming tools but given its power and the ability to customize the interface how we need to far outweighs the cost.

2 Video Display/Camera Hardware

2.1 Overview

The Smart Home Intercom system will have an interactive UI, and will be able to view the seconds nodes video feed when making a video call. This means that the display must have a clear image quality, and large enough that it isn't frustrating for the user to interact with because the video/font is too small to see. This section covers the possible displays available for the RPi.

2.2 Criteria

The reason that his tech was chosen is because the display is one of the main components of the system. The display is a main staple of the entire system and is more important than other tech like the camera, because the user will interact with the display every time they are using the system. If the display is too small, or to low of resolution then it will be hard for the user to read text or even see the video stream.

2.3 Potential Choices

2.3.1 2.4” HAT Primary Display for the Raspberry Pi

This is a small 2.4” TFT resistive touch screen for the RPi. It has a pixel resolution of 240x320, 65K RGB true to life colors, a typical frame rate of 17 fps, controllable backlight, and requires no external power cable as it is powered by the RPi. The display mounts directly to the RPi with 4 mounting screws, and only cost about $35. [4]

2.3.2 Raspberry Pi 7 Touchscreen Display

This is a 7” capacitive touch screen with a pixel resolution of 800x480 and features support for an on screen keyboard built into the raspbian OS without the need for physical keyboard. Power is supplied through an adapter board that also converts the signal from the display to the serial port on the RPi. This display can be mounted directly to the RPi or put into a range of cases, and costs about $60. [5]

2.3.3 Raspberry Pi 10.1inch HDMI LCD Capacitive Touch Screen

This is a large 10.1” LCD capacitive touch screen with a pixel resolution of 1024x600, and features multiple input video sources like: HDMI, VGA, and AV. This screen also features an on board multi language OSD menu that can adjust brightness, contrast, and even alter power settings. The RPi can mount directly to the back of the screen, which is powered by an external 12V/1A power source. This screen comes with a case and a bracket that goes on the bottom to hold the screen at a 45 degree angle while sitting on a table, and is a bit on the high end with a price of about $117. [6]

2.4 Discussion

The 2.4" screen is a small and very affordable screen, but is too small for the functionality the system is supposed to have. Using the screen would be too difficult to use without some sort of stylus, and video calls would be extremely small to look at. Compared to the 2.4" screen the 10.1" screen is a monster. The large screen would make using the system much easier to use, video calls would be easily visible, and the on board brightness controls are great addition. However, the price of the 10.1" screen is a bit too steep compared to the 7" screen, which offers support for a built in virtual key board, a good size screen that will allow for the same functionality as the larger screen but with a more affordable price.

2.5 Conclusion

We chose the 7" screen because it offers the best usability for our project. The screen size is large enough that it wont be a pain to use it without a stylus, and will make video calls clear enough for our purposes. Another reason we chose this screen is because of the price, the 10.1" screen would be great to use but the cost is much too high vs its added benefits, making the 7" screen more a more cost effective choice.

3 ENCRYPTION

3.1 Overview

The Smart Home Intercom system requires that the network be encrypted so that the streamed video/audio is not able to be accessed from an outside source and easily viewed. This section covers three different encryption libraries that are supported on the RPi.

3.2 Criteria

The reason this tech was chosen is because the the system is required to have an encrypted network making video and audio calls between two nodes secure and unable to be viewed from an outside source. There are plethora of encryption libraries available, and since the system will running on a mesh network, the encryption library has to be efficient so it doesn't clog up the network, or cause video quality to drop.

3.3 Potential Choices

3.3.1 NaCl (Salt)

NaCl is an open source encryption library that supports multiple languages like Python, C, and C++. It encrypts data using AES and claims to have record speeds for encrypting bulk amounts of data with the best secret-key cipher available. The library also supports automatic CPU-specific tuning, meaning that it supports multiple implementations of the same function giving different CPU's the ability to automatically detect the best option for it. [7]

3.3.2 Salsa20

Salsa20 is an open source stream cipher that is much faster than traditional AES encryption, making it ideal for applications that are more reliant on speed than encryption confidence. It written in C++ and supports various operating systems like linux and windows. Salsa20 also provides support for perfect forwarded secrecy which means that past sessions are protected against future compromises of secret keys or passwords.[8]

3.3.3 OpenSSL

OpenSSL is an open source implementation of Transport Layer Security (TLS) and Secure Sockets Layer (SSL) protocols, and is also a general purpose encryption library written in C with support for most versions of Unix and Unix-like operating systems. Wrappers for OpenSSL allow it to be used in various programming languages. However, it does suffer from vulnerabilities like plaintext recovery attacks and timing attacks on RSA keys. [9]

3.4 Discussion

OpenSSL is one of the most widely used encryption libraries, which means that is supported almost everywhere, and has tons of documentation for implementation and troubleshooting. However, in contrast to Salsa20 and NaCl it is vulnerable to many attacks. NaCl is another very popular encryption library because of its speed and security, and its ability to optimize itself based on its CPU is a very good feature, but compared to Salsa20 the encryption method is still slower using AES. Salsa20 is the fastest encryption library of the three choices even with NaCl being able to optimize itself, and is can even be implemented in faster forms using Salsa20/12 or Salsa20/8 making it one of the fastest 256bit stream ciphers available. However, this increase in encryption speed for Salsa20 comes at the cost of reduced security in the encryption.

3.5 Conclusion

We chose Salsa20 because of its extremely fast encryption method compared to the other two choices, because the system is a mesh network and data transfer will degrade over multiple nodes. If the encryption is slow it could either make the entire network work at crawl, or even prevent any connections between nodes because of timeout.

4 References

[1] https://elinux.org/RPi-Cam-Web-Interface

[2] https://github.com/vbence/stream-m

[3] https://www.videolan.org/vlc

[4] https://www.element14.com/community/docs/DOC-77883?ICID=rpiaccsy-access- products

[5] https://www.element14.com/community/docs/DOC-78156?ICID=rpiaccsy-access- products

[6] https://www.amazon.com/Waveshare-10-1inch-HDMI-LCD-Capacitive/dp/B01CU7VX5Q

[7] https://nacl.cr.yp.to/index.html

[8] https://www.cryptopp.com/wiki/Salsa20

[9] https://www.openssl.org/