Project Proposal

**WebRTC Video Chat Android Application.**

05/01/2016

Dominic Rathbone

Student number: 12843140

CI360 – Mobile Application Development

School of Computing, Engineering and Mathematics

# 1) Elevator pitch

My android application will allow two users to message as well as voice and video chat directly with each other without the need for a intermediary server. To do so, my application will use WebRTC, a new technology aimed to provide peer-to-peer connectivity to the web. By avoiding the need for an intermediary server, it provides a service with greater anonymity and security which a lot of people find important in an age where online privacy has become such a concern.

# 2) Rationale

On a shallow level, my android application aims to provide a free service to users who want to to message as well as video and audio chat with each other over WiFi.

However, it is also being created to provide a method of communication and data exchange that avoids the inherent data privacy and security issues associated with client-server architectures due to the trust being placed on the providers running these applications and services to securely store your data. These problems are growing larger in the domain of mobile applications as many deploy a server based infrastructure to take the stress of data processing away from client devices and this is shown by the current OWASP (Open Web Application Security Project) top 10 mobile security concerns with the first on list is “Weak server side controls” [1].

These issues have become more apparent in the mainstream media in recent years with consumer's application data not only being suscesptible to these illegal and unethical attacks [2] but also becoming suscesptible to legal, intentional access with bills such as the “Investigatory Powers Bill” being drafted by the UK government which aims to force mobile and internet service providers to be able to decrypt user's data on request [3].

In this case, my application also serves users who are becoming more aware of these issues and want to avoid the possibility of third parties inappropriately using the data being stored on their servers.

# 3) Application

At the moment, the largest messaging platforms such as WhatsApp and Facebook Messenger both provide mobile applications for their services. From a normal user's perspective, it is hard to criticise these two applications as they have a great user interface and range of functionality, this being shown by their user bases of 900 million [4] and 700 million [5] respectively. However, they are liable to criticism from a technical perspective.

Although WhatsApp has implemented VoIP functionality through a library called PJSIP, it is not as advanced as WebRTC in terms of protocolling and technologies used [6]. On top of this, the majority of it's functionality is based on a client-server architecture which, again, is vunerable to the issues previously described with WhatsApp even being a target of a proposed ban by the government for their use of strong data encryption protocols [7].

Opposite to this, Facebook Messenger has implemented WebRTC for it's mobile video and voice calling early on in 2015. Although this is a good example of how WebRTC can be applied, Facebook Messenger has a security flaw in it's implementation in that it uses SDES (Session Description Protocol Security Descriptions) encryption instead of DTLS (Datagram Transport Layer Security) which is the standard encryption scheme for WebRTC. The problem with this is “the encryption keys are sent via the signaling servers and can be used to retroactively decrypt traffic.”[8]. This would mean that for example, this service would be liable to access by the government in the case that the “Investigatory Powers Bill” is passed even though it is using technology which should avoid this.

One application that provides an alternative to these applications is Sicher which aims to provide a free instant messaging service that is encrypted end-to-end. From their FAQ, it states it uses “point-to-point encryption, based on asymmetric cryptography. It means that only the recipient who owns the private key can decrypt the message. RSA cryptosystem is used with 2048 bit keys. Additionally all data exchange between mobile apps and Sicher servers is protected using SSL.”[9] to achieve this. Although, it does exchange this encrypted data through an external server, it attempts to reduce the risk of this by immediately deleting this data off their servers as soon as the recepient recieves it [10]. However, the level of encryption that it bases it's product on could suffer if the “Investigatory Powers Bill” was passed and they were forced to downgrade. Another criticism of mine is that the user interface is relatively immature and it has only been designed for use with a mobile phone as opposed to larger devices such as tablets.

A criticism of all of these applications is that they lack transparency in terms of their source code due to the fact that they are proprietary. By using an application that is open source, the level of transparency almost guarantees that there is very little room for inappropriate usage of the data that goes through it e.g for marketing. One messaging application like this is ChatSecure. An application that aims to provides symmetric OTR (off the record) encryption, which is a method of encryption specific to real time messaging, this works by using a new key for each and every message sent by an application.

My application would provide an alternative to traditional instant messaging applications such as these by completely basing it around the WebRTC (Web Real Time Communication) native API for Android. This API allows the an instance of the application to form a peer to peer connection with another instance to directly exchange data over. From the user's perspective, this connection gives them the ability to stream data such as text, video and audio to and from one another. By using this alternative architecture, it avoids the data passing through a third party server and avoids the security and privacy issues associated with third party client-server applications mentioned previously.

# 4) Scenario of use

A use case for my application would be someone (refered to as user A) who wants to communicate with their friend (refered to as user B) privately. They would open up the application and enter a temporary name to identify themselves by. From the application, they would then share this name with user B by email (amongst the various other forms of sharing my application would implement e.g text, facebook). After user B receives this email, user B would go to the application, click “join chat” and enter user A's name, sending a chat offer to User A. This would trigger a notification on user A's application with their offer, once user A has accepted the offer, they enter a new chat room and either of them could either trigger a video/voice chat or just send plain text messages to each other. After they are done and the users would disconnect from the room and it would be disposed.

# 5) Technical overview

When a user first

Give a technical overview of your application. This may change during development, but at this stage you should have a rough idea of how the application could be implemented. Include details such as persistence, communications, significant processing, inputs, outputs, web services etc. (max. 500 words incl. diagrams)

# 6) Technical challenges

Describe briefly the main technical challenges you foresee. (max. 200 words).

# 7) UI design draft

# 8) Work plan

Describe the work needed to implement the application, and a timeline showing the different phases including requirements analysis, design and development, demo preparation and write-up at the end of the project. (max. 400 words incl. diagrams)

# 9) Ethical considerations

Discuss any ethical considerations related to the project or the application you develop. With respect to the application, think about privacy, data protection, health and safety or any moral implications. With respect to the development project, think about the involvement of users in your requirements analysis and testing. (max. 200 words)

# 10) Resources

List the resources you need to carry out the project, including access to specific hardware (e.g. smartphone), software (e.g. development environment), data sets, online services, organisations, people, etc. (up to 200 words).

# References

[1] OWASP. (2014). *OWASP Mobile Security Project - Top Ten Mobile Risks.*Available: https://www.owasp.org/index.php/Mobile\_Top\_10\_2014-M1. Last accessed 06/01/2016.

[2] William Enck, Damien Octeau, Patrick McDaniel, and Swarat Chaudhuri. (N/A). *A Study of Android Application Security.* Available: http://www.cs.rice.edu/~sc40/pubs/enck-sec11.pdf. Last accessed 06/01/2016.

[3] Home Office. (2015). *Draft Investigatory Powers Bill.* Available: https://www.gov.uk/government/publications/draft-investigatory-powers-bill. Last accessed 06/01/2016.

[4] Leo Sun. (2015). *Facebook Inc.'s WhatsApp Hits 900 Million Users: What Now?.*Available: http://www.fool.com/investing/general/2015/09/11/facebook-incs-whatsapp-hits-900-million-users-what.aspx. Last accessed 06/01/2016.

[5] Alexei Oreskovic. (2015). *Facebook Messenger added 100 million users in the last three months.* Available: http://uk.businessinsider.com/facebook-messenger-has-700-million-users-2015-6?r=US&IR=T. Last accessed 06/01/2016.

[6] Philipp Hancke. (2015). *What’s up with WhatsApp and WebRTC?.* Available: https://webrtchacks.com/whats-up-with-whatsapp-and-webrtc/. Last accessed 06/01/2016.

[7] Rory Cellan-Jones. (2015). *Does the government really want to ban WhatsApp, iMessage and Skype?.* Available: http://www.bbc.co.uk/news/technology-33737813. Last accessed 06/01/2016.

[8] Philipp Hancke. (2015). *MESSENGER EXPOSED: Investigative Report.* Available: https://cdn.andyet.com/webrtc-reports/messenger-report.pdf. Last accessed 06/01/2016.

[9] N/A. (N/A). *Sicher FAQ.* Available: http://www.shape.ag/en/faq/sections/sicher.php#why-is-sicher-secure. Last accessed 06/01/2016.

[10] N/A. (N/A). *Sicher Privacy Policy.* Available: http://www.shape.ag/en/privacy/sicher/. Last accessed 06/01/2016.