A mobile gateway for medical auscultation

- Enhanced Client Certificate Authentication and MTLS Security

Jon Leijon
Abstract

Attempting to re-engineer a telemedicine application, this report sets out to make a cheap yet robust solution that is portable and easy to use. It will be a Inera compliant system using Android that authenticates in a secure way with a .NET server behind F5 full proxy.

The report begins with a background to telemedicine, and a technical specification that might fulfill the demands of privacy laws regarding health care records. The focus of the report moves on to inspects the choices for authentication. The results consists of formulating a possible solution for authentication by self signed client certificate between Android and a Windows server.

En mobil gateway för medicinsk auskultation

Sammanfattning

Arbetet i artikeln består i att omarbeta en existerande applikation för telemedicin till en lättanvänd och säker ”mobil gateway”. Vi ska skapa ett system som använder Android på klient sidan och sedan undersöker vi hur man kan synkronisera klient sidan att autentisera med en server skapad i .NET som använder F5 full proxy.

Examensarbetsrapporten ger en bakgrund till telemedicin och en teknisk specifikation av vad mjukvaran måste förhålla sig till, i form av lagar och existerande system. Sedan presenteras möjliga lösningar för autentikation. Resultatet blir ett förslag till hur den mobila gatewayen kan skapas och hur autentikation kan ske mellan mobilens Androidsystem och landstingets Windowsserver med bruk av self signed klientcertifikat.
Acknowledgements

My thanks to my tutor Johan Skönevik, without whom this thesis would not have been possible. I would also like to thank my mentor Marie Nordström for the constructive talks and encouragement.
# Contents

1 Introduction 1  
  1.1 FjärrKonsulten 1.0 2  
  1.2 Developing FjärrKonsulten Further 2  
  1.3 Necessary Upgrades 3  
  1.4 Use Case 3  
  1.5 System Architecture 4  
  1.6 Problem Statement 4  

2 Background 5  
  2.1 Telemedicine 5  
  2.2 Västerbottens Läns Landsting 5  
  2.3 Law for Using Personal Information in Public Records 6  
  2.4 E-Health Specifications from Inera 6  
  2.5 RIVTA Documents 7  

3 Theory 9  
  3.1 Method 9  
  3.2 The Android Platform 9  
  3.3 Device Management 10  
  3.4 VPN & APN 11  
  3.5 Connection Specifications for Hyper Text Transfer Protocol Secure (HTTPS) 13  
  3.6 Digital Certificates 13  
  3.7 Client Certification 14  
  3.8 Tokens 14  

4 Technical Solution 17  
  4.1 System Specification 17  
  4.2 Server Side 18  
  4.3 Client Side 19
1 Introduction

Figure 1: The workflow in FjärrKonsultan from a physical point of view.

The purpose of this paper is to conceive a security solution that can be used to authenticate mobile device users towards a server. This is useful when handling medical data, since Swedish law strictly prohibits any security flaws.

The front-end or, as it is referred in this paper, client side of this security solution is implemented in Android. The users of this client side application are health care professionals, which means that an authentication method designed for professionals is preferred over a private one.

As seen in Figure 1, the system consists of the stethoscope, phone, cellular provider, server and computer as it appears in 1.

The scope of the work is making the client side, and configuring the connection between the client side and the server side. There are two options for authentication, either configure the system to use the native health care authentication Säker IT för Hälso- och Sjukvården (SITHS) and reconstruct it for mobile, or make a custom solution for authentication.

In addition to making a choice for authentication, some of the specifications for other areas of the system is introduced and examined. They are related to our main problem in that they all deal with the security and integrity of our system. Some of the questions that these specifications answer:

- How can we assure a safe Android environment for the application to work in?
What is a safe way to use Bluetooth?

Is HTTPS the safest connection, or are Virtual Port Network (VPN) or Access Point Name (APN) safer choices?

Are VPN and APN practical?

1.1 FjärrKonsulten 1.0

The application we are re-engineering is called FjärrKonsulten. FjärrKonsulten has been in existence since 2004, and went through a remake in 2015 to be compatible with Windows 7 computers.

A practical example of the usage of FjärrKonsulten is how it has been utilized by pediatric specialists when a baby is born with a heart murmur. That murmur usually disappears within a week, and the baby is fine. With FjärrKonsulten, the family can return to their home and then have a distance consultation with the doctor when a week has passed, and the baby can usually be cleared of heart decease. Instead of traveling many hours by car to get to see the doctor, they can go to a local medical center. The auscultation happens in real-time, so the doctor gets the inputstream while the procedure is happening, and he reacts in real-time over a telephone link.

1.2 Developing FjärrKonsulten Further

We can cite the original template for the application[1], the usage of FjärrKonsulten should facilitate the following.

- Extend qualitative care of elders to their home environment.
- Eliminate unnecessary trips to the hospital for exams that could take place in their home, or live-in facility, group home or local health center.
- Shorter intervals for checkups and better follow-through when the caregiver performs the exams in the patients home, and in some instances, the patient can perform its own auscultation.
- Give greater background of health parameters for doctor visits in a nursing facility.
- Allow for home care to the many elders whom wishes for home care, and also to children, who are known to recuperate better in a safe environment.
- Be implemented to improve the cost-benefits of the provider: it frees resources and traveling for health professionals.
- Create shorter waiting periods; direct consultation instead of referral and traveling to a specialist.
- Keep the patients in focus, the technology should connect the institutions of city, region and private actors.
• Improve follow-ups on different types of heart disease.

It is also desirable to give a base for an application that can include other types of sensors. Android has not previously been utilized as a mobile platform within the region in relation to telemedicine, but seems a natural step in the evolution of ubiquitous telemedicine. That is also a reason to do a strong validation of the platform before the project continues. If it is thoroughly investigated and proven to be safe to use with the Västerbottens Läns Landsting (VLL) servers, other telemedicine solution can follow.

A lot of investments are made for the infrastructure of a hospital. But the very centralized structure should not exclude the possibility of medicine outside the hospital, and the need for interconnectedness with the hospital. This is what FjärrKonsulten achieves.

1.3 Necessary Upgrades

To stay relevant to the needs of health care professionals, the software FjärrKonsulten needs to be upgraded. At the moment, the software only works with Windows on dedicated machines, connected to the same private network. This could be extended to work with an Android-application that connects to a server online. The proposed solution is to make an Android application that works offline to perform the recording together with the stethoscope. With a click from the user, the client side loads the recording directly onto the VLL server side. A server side part of the system will also be designed in this paper.

As such, there is no need to continue with live streaming of auscultations as was the praxis in FjärrKonsulten 1.0.

The application needs to connect to a Bluetooth stethoscope called Littmann 3200, produced by 3M. It has an API for Java and Android that transfers the digital audio.

To make an 'easy to use' application [2], it is positive to remove the bigger obstacles for an intermediate user, such as having to do extraneous typing on software keyboards. To save time, all components should be clickable, i.e. Dates should be clickable from a calendar, the password can be slide pattern or thumbprint, Social Security Number (SSN) can be input by scroll list.

1.4 Use Case

See appendix for a figure of the use case. It details how a nurse visits a patient, starts the app, presses record and does the auscultation. The software takes the information and bundles it in eXtensible Markup Language (XML) form. When the nurse presses send, the software creates a connection to the server and pushes the XML to the server. Figure 6 is a simple use case for the implementation of authentication, which in the use case is done through a certificate.
1.5 System Architecture

We are building the solution with VLL in mind, and the current configuration of their servers is that they use a full proxy, the F5 system [3]. In our implementation, the F5 means that we can use a server placed inside the VLL - firewall which we first assumed could not be done, because the VLL servers have heightened security standards. The F5 can achieve heightened security. This means that we can use the authentication services that are connected to the VLL, which are Active Directory (AD)-verification and SITHS-verification. This also means that we can load our web service onto the VLL secure server storage and run it as a typical server. It is sufficient with a DNS-server connection to use the basic functionality of the F5 system.

VLL IT department is currently evaluating Mobile Device Manager (MDM) systems, but nothing is yet decided [4]. Currently, the IT-department is not regulating the use of its mobile devices. The devices are mostly iPads, and the usage is mostly for email and sometimes keeping track of the patient locations. None of the devices are used with telemedicine, it is all used for ‘in house’ services, within the hospital. Once a MDM-system is in place, handling of keys and user authentication can be done with more ease. single sign on, which Inera states as an preferable authentication, is easier to implement with the MDM system as well.

1.6 Problem Statement

As described in the introduction, our goal is to create a secure authentication that conforms to good practices.

What is a easy & lawful method to authenticate a health care professional in a low risk scenario, using Android & .NET?
2 Background

Some background was presented in the previous chapter, this chapter continues to establish the condition of the surrounding systems as well as the condition of the problem.

The limitations to the solution are set by the infrastructure of telemedicine in the region, and by Swedish laws and recommendations regarding telemedical applications.

2.1 Telemedicine

Telemedicine is the term for all healthcare performed over a distance. This type of medicine has been afforded new possibilities due to the deployment of internet access.

Still, telemedicine is not a ubiquitous solution. Homes in many rural regions in Sweden are not connected to any network except for the telephone line [5]. Mobile internet access is lacking in several parts of Västerbotten.

There is a ‘public good’ in providing access to medical services outside of the hospital environment. Telephone consultations, video conferences or, as in this report, transfers of auscultation recordings. The proliferation of telemedicine differs from country to country, based on laws and practices, as well as depending on the development of internet access [6].

In the US, an E-Health framework is already in place, where all the specifications are clearly stated from the government [7]. Europe is only the third most evolved continent in E-Health according to WHO [6].

In Sweden, service for E-Health is coordinated by Inera AB. Each region has its own R & D and technical center for. To do their job, Inera creates some standards for encryption and security based on Swedish Law and best practices. They publish this on their RIVTA (Rules for Interoperability in HealthCare - Technical Instructions) website.

2.2 Västerbottens Läns Landsting

When VLL is referred to in this paper, it refers to the geographical and political area of Västerbotten, situated in the north of Sweden. The VLL is the political and fiscal entity that provides health care, culture, and public transport to the region.

Sweden is parted into 21 regions, and each Landsting has a political responsibility for the health care in the region. There is cooperation between the Landsting, some of the cooperation is provided by Inera.
2.3 Law for Using Personal Information in Public Records

Applicable laws for E-Health does not regulate platforms nor encryption methods, it just gives a framework of expected security standards [8].

From PUL(Law of Personal Information):

... Responsible for the Personal Information should adhere to appropriate technical and organizational measures to protect the personal information that is being treated.

These measures should establish a security level that is appropriate considering:

1. the technical possibilities available,
2. the cost of applying the measures,
3. the particular risks of the handling of personal information, and
4. how sensitive the personal information in question is.

It is out of the scope of this assignment, but it is something to keep in mind when designing a system in any healthcare setting.

2.4 E-Health Specifications from Inera

Some groundwork about the security concerns for the system has already been addressed by Inera in their E-klient standard for mobile E-klient [9].

The E-klient refers to a common standard for services to be shared throughout all regional health services [10]. This is an optional solution for each region, but it has the potential for great savings for regions when development can be shared and not reproduced in each region.

Even if the application never appears as a E-klient shared resources, we have the benefit of citing Inera as the authority on E-Health security.

One of Ineras recommendations for using mobile devices are the methods for authenticating the user. The strongest form of authentication is the SÎTHS-card. This, however, is difficult to implement on a mobile platform due to its probable lack of a smartcard-reader. ¹

A negative aspect of the SÎTHS-authentication is that dongles or separate card-readers may have a short lifespan when upgrading software and hardware. It is also more user-friendly to just keep track of the mobile device and not have to worry about peripherals. An alternative presented with favor is onetime-passwords in combination with a PIN-code for the mobile device. Another solution is to have a certificate in conjunction with a password or fingerprint.

The goal is to have a single sign on for all applications on the device. The end user should only keep track of one password. Inera evaluates the performance on different types of

¹Although there are a few tablets with integrated smart card readers on the market, no mobiles with such capability could be found.
They recommend a Mobile Enterprise Application Platform (MEAP) for producing applications if the goal is that more than three applications should be produced. These are some of the recommendations to keep in mind when developing the system in this project.

2.5 RIVTA Documents

RIVTA is 'Regelverk för interoperabilitet inom Vård och omsorg - Tekniska anvisningar'. A free translation to English could be 'Rules for Interoperability for Health Care and Nursing Care - Technical Instructions'. RIVTA make some recommendations for encryption protocols that we should honor when sending our messages from client side to server side.

RIVTA has ranked the security of cipher suites, a rank that is modern enough for deployment today. The document is revised each year, so the cipher suites get updated when security holes are detected. As long as the FjärrKonsulten project and the RIVTA project are maintained, these security recommendations will be a good starting point.

The proposed application will push its messages with one of the highest ranked cipher suites, in combination with some sort of Client Certificate or Token.

The recommendations are that any token or login-key should be valid for at most 24 hours, and should be inactive for a maximum of 30 minutes. The server side should not allow client-renegotiations (due to renegotiation-attacks), and not use Transfer Layer Socket (TLS) compression.
3 Theory

Android is a current and well-developed Operating System (OS), with many interesting security structures in place for a developer to use, the outlines of Android are drawn below. The safety of certificates and Elliptical Curve Diffie-Hellman (ECDH) algorithm are discussed and analyzed. The merits of using tokens are presented as well as the difficulties. A method is devised which will consist of experimental prototyping.

3.1 Method

The method which suites this problem is to do an experimental prototype[11]. It is a common method to do a throwaway design that collects the specification of a problem[12].

The important part in experimental prototyping is to clearly define the purpose and how to measure the achievement of the purpose[13].

In our case, the prototyping is a mix between high and low fidelity prototyping. We present different ideas of authentication(low fidelity) and choose one for further experimentation(high fidelity). The purpose is to get some info about connection and authentication. The evaluation of the experiment will be successful if we acquire knowledge of how the task can be done.

The purpose of the experiment is to do a throwaway prototype, one that can lead to the knowledge of how to construct the base of our telemedicine system. This will be done in the second step with more resources and higher precision. Although if the prototype experiment turns out to be a solid system, parts of the prototype might be used as a base for the second step.

3.2 The Android Platform

Android is a open source OS. The Android software stack in Figure 2 consists of a hardware level of drivers, that is connected to the native libraries. Previously the Dalvik machine, now the Android runtime, runs the Android framework that powers the application created by a developer.

Disadvantages of Android: Viruses are multifold within the Android Platform, in spite of the enhanced security that comes from using a permissions based architecture[14]. A possible explanation is that 83 % of all devices in the world uses Android[15].

Generally, there are three kinds of ways to invite viruses to a Android device. The first way is to install software that is not from the Google Play Store. The second is to 'root'
the device, which allows administrative access to the device. And the third is to grant permissions to programs that should not have been trusted.

A high number of Android devices are running on outdated OS’[16]. This is partly due to the lack of a centralized update system for android, which means the hardware provider usually has to provide the OS updates as well. This is a security concern as well since safety vulnerabilities become known in older versions and updated to newer versions.

**Advantages of Android**: A good reason to choose Android as a OS is that it is free, provides great capabilities, and the hardware it resides on is cheap in relation to its processing power. It is also very easy to program towards. There are many affordances in the standard Android libraries, so it is not as code heavy as for example the programming language C.

It has some ’System supervising’-features which allow a bit more security, mostly in the device management system. Android can natively connect to the cellular network, which affords mobility outside of wifi and broadband. Android can give plenty of processor power and many healthy and needed peripherals in an ’off-the-shelf’ device. If done right, Android can save a lot of money and shorten the step to utilizing telemedicine in the region. It is at least theoretically possible to use Android in a safe way [17], and many measures can be taken to relieve the pitfalls of Android security.

### 3.3 Device Management

Inera recommends the use of a MDM system for any region that will use at least three mobile applications in their regional E-Health strategy over the coming two years. The
main the point of using MDM is that it will allow an administrator or administration unit to control the mobile devices. It differs between platforms of course, but most MDMs will allow administrators to do tasks such as monitor devices, grant or revoke access for users, distribute software and secure communications. The administration options depends on the OS that is being administered, since MDM is a 3rd party service and not something provided by the OS manufacturer. It can only administer what has been offered up for remote administration by the OS manufacturer in the API.

For the client side FjärrKonsulent it would mean that the client certificates could be uploaded directly to the mobiles. Also, a single sign on could be used and secured in MDM-system. In the future, FjärrKonsulent could be intertwined with other healthcare-applications in a general mobile interface thanks to a MDM service.

3.4 VPN & APN

Virtual Private Network(VPN) and Access Point Network(APN) are similar in that they allow a device to connect to a service on a trusted level. APN can be used to connect a phone directly to the network of a specific host, for example, the VLL-network. A VPN is used to connect to a foreign network from an already existing network connection. The APN is usually not considered as a secure network solution, but if it can be directly linked from the network provider, it works as well as VPN[4].

The benefits are that once logged in, a variety of services could be available for the user inside the network instead of just what has been made available over the networking protocol. A downside is that it can be expensive to implement these services.
Elliptic Curve Diffie-Hellman Algorithm

Client Side
Private Number

Server Side
Private Number

Shared Number

The Random Private Number is raised to the power of the Shared Number:

Now the combined numbers are sent between Server and Client.

The shared master secret is made by raising the private number to the power of the received combined number.

Figure 3: ECDH illustrated with colored keys instead of numbers. The final key is used as a modulus to encrypt and decrypt the messages sent over the connection.
3.5 Connection Specifications for HTTPS

The most important thing for a secure connection is to configure a secure communications protocol. When a client contacts the server, the first thing to happen is the 'handshake'. This includes a client certificate after the server has accepted a cipher suite in accordance with RIVTA standards.

Nothing is encrypted until the end of this process, but the process doesn't state the key explicitly, thusly it cannot be decrypted by a third party. The keys exchanged are relative to the master shared secret, which is not known to any 3rd party. This is done by the ECDH algorithm[18], as described in figure 3.

The public key certificate is distributed freely to any party that tries to connect to the server. The private key to the certificate is a secret however, it is only used for signing before sending a message. This allows the receiver to know that the holder of the public certificate has the private key to the certificate as well. A 'Master Secret' is created in the ECDH algorithm between client and server. In the handshake, a personal number is added to the public key, sent to the other side and then combined with the personal number from that side. This leads to both sides of the conversation having the same key for decryption.

All this is defined in the TLS connection protocol, including how cipher suites get decided and how the client certificate is sent [19]. This makes it very useful to use HTTPS and not HTTP with Secure Socket Layer (SSL).

3.6 Digital Certificates

A digital certificate (or public key certificate) is a file comprised of user information, key information, and a digital signature of the verifying authority. This format of the certificate is called a x509 certificate [20].

An SSL or TLS certificate is used for encryption of data, typically in HTTPS communication. It is also used for authenticating the sender of the certificate, typically when the receiver compares the publisher of the certificate to a list of trusted Certificate Authority (CA).

The most common and visible certificates are the CA's that are used for secure connections in web-browsers, for example when using google mail or bank services.

CAs are not necessary when using certificates where both parties have prior knowledge of each other, as in the case of our client and server. The CA validate that this is a certificate that belongs to a certain company or person. Instead of CA-certificates, the known parties can create self signed certificate, which follows the same structure as the CA-certificate, but the original certificate is signed by someone who is not universally recognized as a proper CA.

The handshake has the option to include authentication of the client side. If a certificate is needed, it will be requested by the server in the 'server hello'- part of the handshake. All user messages are signed with the private key of the client certificate. To authenticate the user, the client certificate can be used to verify that all signed messages originate from someone who has the private key.
3.7 Client Certification

A client certificate is a certificate used for authentication. It is similar to a server certificate, it conforms to the X509v3-standard, but it has a different Object Identifier (OID) [21]. The client certificate is published and used for authentication a user, and never for encrypting a connection. A public key certificate is a certificate used for both authentication and secure connections.

The client certificate should guarantee that the device is an authorized device. However, keys are easily duplicated. It should be used with a key store with the intent of not being moveable.

To accomplish this the certificate is saved in the applications 'sandboxed' space on the device, where it will not be accessible to the other users and applications of the device.

Since the certificate is a file, it can be guarded by the system. This makes it safer than a regular password since a regular password is subject to social manipulation and human weakness.

Client certificates can be combined with other authentication to form 2FA. Client certificates can be connected to AD, a user handler for Windows systems. ¹This is used by VLL, and means that a client certified application could theoretically get safe access to patient records.

In an article by Parsons [19], he highlights that the client certificate is sent unencrypted in the TLS handshake. If a handshake is made without a client certificate, and later a certificate is requested by the server or offered up by the client, then it will be sent under the guise of encryption. This is true for the certificate itself, but as stated under the section digital certificates, the server should verify that the private key has been used to sign the previous part of the server/client communication. So unless the self signed is very easy to forge, releasing it publicly should not be a problem. Ie, the self signed certificate should be authorized by a certificate known to the server.

3.8 Tokens

It could be beneficial to have a token validator in the case that tokens are to be used. For example a middleware software like identityserver3 [22]; the client connects against the server, that in turn connects it to identityserver3, that applies an authentication layer on top of OAuth or open connector.

Tokens can be used without HTTPS, they can be used to pass on authentication to another party and tokens can be used to scale up a service because the connection can keep track of authentication, freeing up server space [23]. All this makes tokens an easy way to send something with validation.

Mobilt BankID

This is a verificator, it verifies that the user is who she or he claims to be. It consists of a certificate that is downloadable from the individuals bank, then uploaded to a mobile device that contains the application Mobilt BankID. This is usable to access government services

¹If the password provided with the certificate matches the AD password, authentication to servers could be direct from mobile.
and make payments [24]. This could be used by the server to request a token when the client wants to create a session.

To get a mobil BankID however, the individual health care professional needs to contact a bank, identify themselves, and then turn over the certificate to VLL. If the employer cannot control the certificate or token, it should not be deemed safe for usage.

**Mobil SÎTHS-ID**

SÎTHS is a key-card that is used within the health care in many regions in Sweden, one of which is VLL. The SÎTHS-card contains a chip that requires a smart-card - reader. A smart card reader can be bought as an addon for both iPhone and Android. When the reader is correctly setup, it can be used for authentication against the Swedish health care registry. This is a secure solution, it is already in use in hospitals and primary care health centers.
4 Technical Solution

This portion of the paper contains the results or, to put it differently, our method applied to our problem. From the information in the chapters on certificates as authentication, the experimental prototype is described and proven to work.

4.1 System Specification

![Diagram of VLL system]

**Figure 4:** This is, of course, a highly simplified explanation of the VLL-system. But basically, it consists of a publicly accessible part and a part that is only accessible from within the VLL-network. Red lines are unsecured connections, and the internal network has not been marked. The internal network can of course access both internal and external services with the appropriate authentication.

The structure of the Android application is that it should not need a network connection to perform its recording and saving tasks. This is important because if an auscultation is in progress, the medical health care professional might have traveled far to get there, or there might not be possible to delay the auscultation. The network status is not checked until the uploading of the data.

To be able to record the inputstream of the Bluetooth stethoscope, the device needs to have a pairing with the stethoscope. If the stethoscope is turned on and within range, the application connects on startup. After that, the pairing is checked again when the recording button is pushed. The streaming is captured by an `InputStreamListener`.
When the recordings are done and the send button is pushed, the audio recordings are collected in an XML-schema, that will be pushed to the server. The schema contains, besides audio, a timestamp, a sender, and a subject, i.e., the patient. Because of the strict laws regarding handling personal data [8], we try to circumvent the need to use any defining personal information. One measure is to make the client side only allowed to POST over the HTTPS connection. This makes sure that no information gets relayed back from the server, the client side only gets confirmation that the packets sent have been received through the TCP protocol.

Another security measure is that the user of the application gets a physical note encoding the caretakers SSN to a number symbolizing a caretaker. The number is converted within the secure servers and converted back into the SSN of the patient. This practice makes sure that no personal information is transferred at all over the HTTPS connection. It is just the symbolizing number, the auscultation audio, and a certificate signifying the caregiver.

A client certificate will be used to authenticate the user towards the server. It would be even safer with a SITHS-card, but with the other safety measures, such as obfuscating the SSN, the certificate with a passphrase is secure enough. For the network connection, a HTTPS connection is preferred over VPN and APN. It is unnecessary to implement a VPN solution on a problem that only requires HTTPS. Also with the F5 proxy, it is easy to implement HTTPS in this proof of concept.

To create the certificates, the OpenSSL project is used. It can provide self signed certificates for authentication by a simple command-line interface. The server certificate is created first and then used to create the client certificate, so that the client certificate authenticity can be assured. In a complete solution, this should be done and stored on an offline machine, dedicated to just creating keys and certificates.

The VLL-servers have compliancy to the Inera E-klient, and they use Windows software. Windows compliancy is part of the development strategies of VLL. For this reason, the server is limited to .NET applications. With ASP.NET we will develop a secure server side of the application. This can provide additional security from the pre-existing system in place at the VLL, such as the F5-proxy and the AD-registry.

## 4.2 Server Side

For the designer, setting up a server and understanding its implementation is difficult, even more so in a highly developed and advanced environment such as .NET.

With an IIS server running the web service, the TLS handshake protocol is implemented in the IIS server side. The client certificates are mapped to a certificate checking service or to an Active Directory [25]. The client certificate is presented to the web service when a request is being processed, but the validation is most easily mapped in the IIS server. In practice, this means that the authentication can be independent of the server side of our implementation. The web service can further check if the certificate has the correct credentials with additional filters. So the connection can be denied at server and service-level of the software stack. This way of implementing could be changed to map the validation of client certificate to the web service instead of validating in IIS.
Another solution is to use a self host-middleware to allow connections from the internet. In this implementation, we use netshell [26] to add a server certificate to the port, as well as to open it to outside traffic [27]. The rest is controlled with code from within the web service. This is a base for HTTPS where the connection is controlled by the service if it is setup correctly with filters and x509Validator.

When setting up the server, we define a x509Validator that will only validate certificates from our CA. It should also contain a restricted list, which checks for banned or out of commission certificates.

When the handshake is done and the communications are setup, the client posts to the server in the form of XML-files. These files are validated and then stored on the server.

4.3 Client Side

After the 'send' button in 'FjärrKonsulten' is pushed, an XML-file is constructed using the recorded audio, a patient Identifier and a timestamp. The 'send' Activity is called after those tasks are correctly carried out.

To make a TLS handshake with a self signed certificate, we need to setup a custom trust manager. We used a coding example from Chariot Solutions [28] to be able to test a trust manager for self signed certificate, and with a small modification, we can choose cipher suites manually. To create a further backward compatible application, we can use Google services to download additional cipher suites, that will comply with Ineras standards. This is because cipher suites in the API are only updated between Android versions, but it does not hinder us from accessing them through the Google services.

The activity loads the client certificate, then the acceptable server certificates. The key for the client certificate is also loaded. All this is saved in the applications storage space,
where it should only be accessible by our application. The connection is made with the
HttpsURLConnection[29].

To use the self signed certificate, we add an exact copy of it to our application and load
it into the trust manager for our connection. That way, the only trusted certificate is our
server certificate and it should not accept any other connections. This is vital so that the
authentication is mutual.

If internet access is unavailable, the file is saved, with the possibility to auto send it from
a background service when a network connection is detected. If the return code is ok, the
auscultation information is erased from the device and the user gets a green checkbox that
confirms the transfer.
5 Discussion

In this part the result is analyzed, then we look at the broader context of the problem area and where else this is useful, we take a look at the ethical quandaries of implementing the system and finally we look at what is left to develop this solution fully.

5.1 Result Analysis

The report started with the question:

What is an easy & lawful method to authenticate a health care professional in a low risk scenario, using Android & .NET?

Our result is in the experimental prototype. By making this solution for the Android platform in combination with a Windows server, we have proven that there is a solution worth pursuing VLL and FjärrKonsulten. The base of authentication that FjärrKonsulten will stand on, could be reused for several other mobile applications within the VLL-FOU.

The research questions state that the solution should comply with the law, which we mention in the section Law for Using Personal Information in Public Records and in the section E-Health Specifications from Inera. We have used the recommendations from Inera to build the application, so the result is that the system does comply with the law. Of course, the more advanced recommendations could have been followed as well, like utilizing the SITHS authentication. The reason we will not is because it is a relatively low-risk application, in comparison to other medical applications. Not much personal information can leak if the application gets hacked. The greater security concern is that the Landsting needs to remain trustworthy in the eyes of the public.

The theory that a client certificate would be safe and practical has been thoroughly proven by others[19]. That it is practical for FjärrKonsulten becomes apparent in the theory and result parts of this report. To make an upload to the server is practically as fast as an unverified connection through HTTPS. The server compares already existing hashes, so the only cost for the client is sending the certificate, which is usually between 2-5 kilobytes of data.

The experimental method has served its purpose since we have acquired knowledge about how the task can be done. It can be done with the system we have created. It will probably not serve as the basis for the alpha version of FjärrKonsulten 2.0, since mostly exploratory coding has been done.
5.2 Conclusion

The success of the work depends upon the legality of the solution, in regards to patient records being kept safe. This solution offers the latest in TLS-encryption. If it is paired with a MDM and an active development, it should be usable for at least a couple of years.

For someone else with a similar problem, this offers a solution. The contents of the application is not important, the sensors could easily be replaced for a pulse-monitor, a sonogram or a device that monitors blood sugar levels. The vital part of the client side application is the secure connection between the two trusted parties within the Landsting.

Unfortunately, there is no simple way to test the security of this solution. Most of the tests are intellectual, for example:

Could we inject code into the application? If the user of the phone installs a malicious application, able of sharing FjärrKonsulten sandbox, what could be exploited? Or in another scenario; an intent could be sent directly to the send-activity-class, and a connection could be made to the server. This could potentially be exploitable in an unsecured Android environment. The greatest liability for the system is the inability to control the users of the Android device. A MDM is essential for controlling the safety of the whole environment.

Perhaps a reading from an Android-tablet can be as valuable as any chart in a hospital some day, but for VLL there is still a long way to go with developing telemedicine. Ethically, one might consider the Swedish proverb "En levande landsbygd", translated "A living countryside". The possibility to treat patients at home should be good for people who have a long way to the hospital. On the other hand, it is cheaper not to have people come to the hospital. It might be misused so that it becomes harder to meet a doctor since it is more expensive than the fast telemedicine evaluation. Then telemedicine would become a risk for patient safety.

5.3 Future Work

A few fields for further developing the application as well as the authentication are:

- It is worth looking into making the client certification as 2FA, by combining it with a password or something else that the user knows. This is likely needed to give a user proper authorization to access medical records, or to request almost any feedback from the server.
- Password recovery has to be a possibility if 2FA is active.
- Connect client certs to AD and SITHS authentication. With this, mobile users can be connected to the 'Inner Services' of the VLL system, without the need for a smart card reader.
- Maintaining certificates should be automated, perhaps with a keychain and if the certificates can be delegated through a MDM, that would be optimal.
Mostly the combination of a MDM and certificates is interesting since this could become a great cost for a growing system where each certificate needs individual, physical installing.
References


A  Glossary
Glossary

Android  The OS developed by Google for mobile phones. 1, 3, 4, 7, 9, 10, 15, 17, 19, 21, 22
application  a software with a GUI. 1–3, 5–7, 9–11, 14, 17–22
auscultation  Auscultation is the monitoring of bodysounds. In our case it is performed with the use of a digital stethoscope. 2, 3, 5, 17
authentication  To validate the correctness of a purported identity.. 1, 3, 4, 6, 9, 13–15, 17, 18, 21, 22
authorization  To give credentials to a party. 22
Bluetooth  A connection standard. 17
certificate  A physical file that is designed to validate its owner. 9, 17
E-Health  Health care performed over the internet. 5–7, 10, 21
E-klient  A common standard for services to be shared throughout the health care regions of Sweden. If a region complies to the E-klient standard, it may use the services of Inera. 6, 18
F5  a full proxy layer that acts as a intermediate to a server, and can communicate between client and server in realtime. Used for enhanced security as well as message queueing and bundling of traffic. 3, 4
FjärrKonsulten  The system for auscultation. This includes the mobile application and the server software. The system might include database and other services in the future, after the prototype stage. 1–3, 7, 11, 19, 21, 22
health care professional  A doctor, nurse or other sufficiently trained medic. 4, 15, 21
HttpsURLConnection  A connected HttpsURLConnection allows access to the negotiated cipher suite, the server certificate chain, and the client certificate chain if any. 20
<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inera</td>
<td>Cofunded effort to create common technical solutions between health care regions in Sweden. 3–7, 10, 18, 19, 21</td>
</tr>
<tr>
<td>Landsting</td>
<td>The political and geographical political entity that is in charge of health care in a region. The 21 Landsting combined constitute the health care system of Sweden. 5, 21, 22</td>
</tr>
<tr>
<td>Mobilt BankID</td>
<td>Authentication service created by Nordea AB. 14, 15</td>
</tr>
<tr>
<td>.NET</td>
<td>.NET is the platform designed by Microsoft that is used for Windows programming. 3, 4, 18, 21</td>
</tr>
<tr>
<td>region</td>
<td>Refers to a political region of Sweden used for coordinating health care and public health care. Used as a translation of 'Landsting'. 5, 10</td>
</tr>
<tr>
<td>RIVTA</td>
<td>Rules for interoperability in health care and nursing care - technical instructions. A handbook developed by Inera. 5, 7, 13</td>
</tr>
<tr>
<td>verification</td>
<td>To ensure the validity of a document or statement. 4</td>
</tr>
<tr>
<td>Västerbottens Läns Landsting</td>
<td>The political and geographical political entity that is in charge of health care in the region of Västerbotten. 5, 7</td>
</tr>
<tr>
<td>Windows</td>
<td>Microsoft OS, mainly for PC’s and servers. 2, 3, 14, 18, 19, 21</td>
</tr>
<tr>
<td>x509 certificate</td>
<td>The standard that most certificates follow, with entries such as user, CA, date etc. It is the physical file format of the certificate. 13</td>
</tr>
</tbody>
</table>
# Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AD</td>
<td>Active Directory. 4, 14, 18, 22</td>
</tr>
<tr>
<td>APN</td>
<td>Access Point Name. 2, 11, 18</td>
</tr>
<tr>
<td>CA</td>
<td>Certificate Authority. 13, 19</td>
</tr>
<tr>
<td>ECDH</td>
<td>Elliptical Curve Diffie-Hellman. 9, 12, 13</td>
</tr>
<tr>
<td>HTTPS</td>
<td>Hyper Text Transfer Protocol Secure. 2, 7, 13, 14, 18, 19, 21</td>
</tr>
<tr>
<td>MDM</td>
<td>Mobile Device Manager. 4, 10, 11, 22, 23</td>
</tr>
<tr>
<td>MEAP</td>
<td>Mobile Enterprise Application Platform. 7</td>
</tr>
<tr>
<td>OS</td>
<td>Operating System. 9–11</td>
</tr>
<tr>
<td>SITHS</td>
<td>Säker IT för Häls o- och Sjukvården. 1, 4, 6, 15, 18, 21, 22</td>
</tr>
<tr>
<td>SSL</td>
<td>Secure Socket Layer. 13</td>
</tr>
<tr>
<td>SSN</td>
<td>Social Security Number. 3, 18</td>
</tr>
<tr>
<td>TLS</td>
<td>Transfer Layer Socket. 7, 13, 14, 18, 19, 22</td>
</tr>
<tr>
<td>VLL</td>
<td>Västerbottens Läns Landsting. 3–5, 11, 14, 15, 17, 18, 21, 22</td>
</tr>
<tr>
<td>VPN</td>
<td>Virtual Port Network. 2, 11, 18</td>
</tr>
<tr>
<td>XML</td>
<td>eXtensible Markup Language. 3, 18</td>
</tr>
</tbody>
</table>
B Use Case Figure

FjärrKonsulten Use Case

Setting: An auscultation done in a patients home or place of care

Actions

1. Turns on Phone.
2. Turns on Application FjärrKonsulten
3. Application checks for stethoscope pairing
4. Application waits for record or send to be pushed.
5. Recording is made by the nurse.
   (This step is a bit more precarious since the nurse needs to handle the phone, stethoscope and patient at the same time.)
6. Send is pushed by the nurse.
7. A passphrase or additional authentication might be requested by the application.
8. The application sets up the certificates and hashsignature.
9. The server is called and connection is set up.
9.1. The data from recordings are pushed to the server side

Figure 6: Above is a proposed use-case, it is a scenario in line with what has been specified in the chapters 'Necessary Upgrades' and 'Field of Use'.