# INF-2301 Computer Communication and security
# INF-2301

Alexander Saaby Einshøj

October 9, 2015

# 1 Introduction

In this assignment, a file-sharing solution is implemented and analyzed.

# 2 Technical Background

The assignment is written in C# using Visual Studio 2013.

## 2.1 RSA cryptography

RSA is an asymmetric cryptosystem used for secure data transmission. Being asymmetric means that two different keys are used encryption and decryption, with the encryption(public)key available for everyone.

The public key is based on two large prime numbers, which are private. To decode any message encrypted with RSA, one have to be in possession of either the public key itself, or the two prime numbers it is based on.

The RSA algorithm involves three steps: key generation, encryption and decryption [1]

## 2.2 AES cryptography

AES (Advanced Encryption Standard) is a symmetric cryptosystem for secure data transmission. A symmetric cryptosystem implies that only one key exists for both encryption and decryption of plaintext. AES has a fixed block size of 128 bytes, and a key size of either 128, 192 or 256 bytes. The key size decides how many transformation rounds are used to convert the plaintext to ciphertext.

## 2.3 C.I.A

C.I.A(Confidentiality, Integrity, Availability) are the primary goals of information security.

### 2.3.1 Confidentiality

The definition of confidentiality is as follows: "Avoidance of unauthorized disclosure of information or resources".[2]

The main goal of confidentiality is that only those who need to know the information being transmitted, get to know. The concept originates from the military, where a lot of information is handled by this concept.

To ensure confidentiality, one need to take in account the following concepts; Encryption, Access Control, Authentication and Authorization.

### 2.3.2 Integrity

Integrity is "ensuring that information hasn't been modified in an unauthorized way.".There are two main divisions of integrity - data integrity and origin integrity. Data integrity is maintaining the contents of data, whereas origin integrity is maintaining the source of that data.[2]

There exists several tools for ensuring integrity in addition to confidentiality, e.g. Backups, which is useful if data has been altered in an unintended way.

### 2.3.3 Availability

By availability we mean "Ensure that information/systems/...are accessible by those who are authorized in a timely manner."[2]

Some information is only useful if we can retrieve it when needed, e.g security code for bank login.

Availability is an important part of the C.I.A triad, because an attacker might not care about a systems integrity og confidentiality by directly attacking the availability.

## 2.4 A.A.A

Assurance, Authenticity and Anonymity are classified as secondary security goals, following C.I.A being the primary goals.

### 2.4.1 Assurance

Assurance in computer systems refers to how people and systems trust each other. It is not only important that the system can trust a given user, but also the other way around. It's vital that a user who buys something in a web store, can be assured of the system keeping its promises - or policies of not delivering their credit card information to a third party.

### 2.4.2 Authenticity

Authenticity is the ability to determine that statements, policies, and permissions issued by persons or systems are genuine.[3]

Nonrepudiation - a property that authentic statements cannot be denied - is typically achieved by using digital signatures, which is basically like a formal contract between two parts. One might say that a digital signature is better than a blue ink signature, because if a document signed digitally is edited, it automatically becomes invalid.

### 2.4.3 Anonymity

Anonymity is an important part of the A.A.A triad, perhaps the most important one. As of today, most people have no problem with throwing their names and personal information at anything, be it a social networking site, a forum, or any web site that requires registration in some way. We expose our e-mails, names, addresses and so forth. If these web sites take excessive measures to keep each individuals information secret, anonymity is not that big a problem, but the fact is they usually don't - or they can't. The NSA(National Security Agency) in USA have insight in everyone who uses Google Chrome, Facebook, AT&T and other big companies. This is a problem, not only because individuals anonymity is involuntarily compromised, but it happens without anyones consent or a lot of persons knowledge at all.

Aggregation, Mixing, Proxies and Pseudonyms are tools to prevent the compromise of individuals anonymity.

# 3 Design

This solution requires a client and a server, and it is specified by the server which file to send, where it is and what format is has. The server creates two objects, one for AES cryptography, and one for RSA cryptography, followed by the AES encryption object. The path with the plaintext is specified before entering the main-loop of the server side solution, where everything that happens while connected, described.

As soon as the server receives a request from a client(containing the RSA public key), the newly generated AES key is encrypted with the public RSA key, before it then encrypts the file content to be sent to the server, with the AES key. When this is done, both the encrypted key and file is transmitted to

the connected client. The client connects to the server with its IP-address, then generates an RSA key, which it thereafter sends to the server. The response from the server includes the encrypted key and the encrypted file. The client then decrypts the received AES key with its RSA key, then the file content with the AES key, before it stores the decrypted file locally.

# 4 Implementation

This solution was implemented in C#

# 5 Results and Discussion

The created solution is implemented as simplistic as possible. The fact that C# is an object-oriented language, should have been utilized way more, especially if the solution had been more complicated. One could argue that it should have been, even in this solution, as there is a lot of similar code throughout the entire solution, e.g for writing and reading data off the networkstream.

Seeing this in the perspective that the solution satisfies only the most elementary requirements for file sharing, it is reasonable to believe it doesn't suffice for fulfilling the required elements in C.I.A and A.A.A. In the current solution, everything that's transmitted, is also encrypted, and even though it might seem secure to most people, it's not. In light of confidentiality, where only those who need to know, gets to know, this won't be true for this solution. Anyone can connect to the server and request the given file, and in that matter, confidentiality might not be as important as it could have been. If the solution required transmission of important or secret information, confidentiality would be a lot more vital.

Integrity tells us to maintain both the source of the data and the data itself. In this case, the data could easily be edited by an unwanted third party, due to the lack of security in our solution. An easy way to alter the data from the server till it reaches the client would be a "man-in-the-middle" attack. This includes a third party interrupting the "conversation" between server and client by e.g. grabbing the data sent from the server before it reaches the client, altering it, then proceeding to send this edited data to the client without anyone knowing. Our solution is definitely weak to this,

as we don't verify that the client has gotten the correct data at any point. On the other hand we don't allow for any input by the client either, so a potential attack wouldn't easily be able to alter the original data at server side. Although messing with the data itself would be quite a difficult task, it is possible, by compromising the server machine, and physically altering it.

Availability, which let's data be available to the client when it should be, is not an important aspect of this solution, at least not considered to be. The data to be transmitted is in this solution meant to always be available, as long as the server is up and running, meaning it doesn't matter who can get it, or if they can get it at a specific point in time. The problem arises if the data isn't available when the server IS running, and when it should be reachable. This could generally be a problem when hosting a web site, and the server gets DDOS(Distributed Denial Of Service)'ed. In this aspect of security, the source is the element most prone to attacks of any sort, because if the source is compromised, the content is unreachable by everyone.

A.A.A, the secondary triad of information security is also widely used for solving critical security issues. Assurance is generally quite important to keep in mind, as it is required for two communicating parts to trust each other. A person would usually not tell another person a secret unless they felt some sort of assurance that they wouldn't spread their secret to anyone else. This concept is just as important when you interact with a computer system. Not only can a system not be trustworthy by itself, but data might be unwantedly grabbed by someone who shouldn't have knowledge about it.

Authenticity allows the user to be certain that they hand out sensible information to credible systems, usually by using a digital signature. This is important when managing your bank account online, shopping or anything else where it is critical that your personal information is not mis-handled. In light of the fundamentals of this solution, authenticity is not important, but if the data to be transmitted was of any importance, authenticity would be a vital element to consider.

Anonymity is important when a client provides the provider with sensitive information that it wants to be kept between those two parts, or perhaps a trusted third party too. This concept is not a significant aspect of this solution, because the content is supposed to be available to anyone who wants it, and the server doesn't keep track of who makes a request, nor does a client provide any important information, aside from the IP-address. The IP-address could potentially be tracked by the server in a new solution, to have a clue of who requests the data, although it doesn't seem like a necessary

addition to the application.

# 6 References

**[1]https://en.wikipedia.org/wiki/RSA_%28cryptosystem%29 (Visited 07.October 2015)**

**[2]http://www.cs.arizona.edu/ collberg/Teaching/466-566/2012/Handouts/Handou 1.pdf (Visited 07.October 2015)**

**[3]Goodrich, Michael and Tamassia, Roberto (2014): Introduction to Computer Security. 1st edition. USA: Pearson Education Limited**