1/13/2015

Samuel Schmidt, Dominik Scholz

TGM Vienna

Verschlüsselung

5AHITT – SYT IndInf – 2014/2015

Index

[Task assignment 2](#_Toc409356763)

[Estimated and actual time taken 3](#_Toc409356764)

[Design 3](#_Toc409356765)

[Work protocol 4](#_Toc409356766)

[Testing 5](#_Toc409356767)

[Sources 5](#_Toc409356768)

# Task assignment

**Kommunikation [12Pkt]**  
Programmieren Sie eine Kommunikationsschnittstelle zwischen zwei Programmen (Sockets; Übertragung von Strings). Implementieren Sie dabei eine unsichere (plainText) und eine sichere (secure-connection) Übertragung.

Bei der secure-connection sollen Sie eine hybride Übertragung nachbilden. D.h. generieren Sie auf einer Seite einen privaten sowie einen öffentlichen Schlüssel, die zur Sessionkey Generierung verwendet werden. Übertragen Sie den öffentlichen Schlüssel auf die andere Seite, wo ein gemeinsamer Schlüssel für eine synchrone Verschlüsselung erzeugt wird. Der gemeinsame Schlüssel wird mit dem öffentlichen Schlüssel verschlüsselt und übertragen. Die andere Seite kann mit Hilfe des privaten Schlüssels die Nachricht entschlüsseln und erhält den gemeinsamen Schlüssel.  
  
**Sniffer [4Pkt]**  
Schreiben Sie ein Sniffer-Programm (Bsp. mithilfe der jpcap-Library [http://jpcap.sourceforge.net](http://jpcap.sourceforge.net/) oder jNetPcap-Library <http://jnetpcap.com/>), welches die plainText-Übertragung abfangen und in einer Datei speichern kann. Versuchen Sie mit diesem Sniffer ebenfalls die secure-connection anzuzeigen.  
  
**Info**  
Gruppengröße: 2 Mitglieder  
Punkte: 16

* Erzeugen von Schlüsseln: 4 Punkte
* Verschlüsselte Übertragung: 4 Punkte
* Entschlüsseln der Nachricht: 4 Punkte
* Sniffer: 4 Punkte

# Estimated and actual time taken

|  |  |  |  |
| --- | --- | --- | --- |
| *Task* | *Description* | *Estimate* | *Recorded* |
| Documentation | I²C and Accelerometer basic theory research with supplied pdfs as well as usual stuff (work protocol, testing…)  (gg) | 1,5h | 1,5h |
| UART | Right configuration needed  (reused code) | 30m | 15m |
| ADXL | Power modes, value parsing  (took some time to figure out the formula for correct conversion) | 1h | 3h |
| I²C | Configuration and R/W with our Tiva-Board  (Tivaware Peripheral Library proved to be a valuable guide in this adventure) | 7h | 5,5h |
| Sum | | 10h | 10,25h |

# Design

# Work protocol

As our assignment combines asymmetric and symmetric encryption its communication flow is very similar to SSL.

To recap, secure SSL sessions are established using the following technique:

1. The user's Web browser contacts the server using a secure URL.
2. The IIS server sends the browser its public key and server certificate.
3. The client and server negotiate the level of encryption to use for the secure communications.
4. The client browser encrypts a session key with the server's public key and sends the encrypted data back to the server.
5. The IIS Server decrypts the message sent by the client using its private key, and the session is established.
6. Both the client and the server use the session key to encrypt and decrypt transmitted data.

Besides the obvious security advantage of using this scheme, there is also an advantage as the asymmetric encryption is slower and only used for initializing the communication while the faster symmetric encryption is used for the rest of the conversation.

# Testing

# Sources

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| [0] | Title | Author/Institution | Link | Last used |
| [1] |  |  |  | 9.1.2014 |
| [2] |  |  |  | 12.1.2014 |
| [3] |  |  |  | 13.1.2014 |
| [4] |  |  |  | 14.1.2014 |