**1. Programmieraufgabe**

|  |  |  |
| --- | --- | --- |
| **Teilaufgabe** | **Implementiert (J/N)** | **Kommentar** |
| Secret Key generieren | **J** |  |
| mit Secret Key Datei/ String verschlüsseln | **J** |  |
| Key Pair für asynchrone Verschlüsselung generieren | **J** |  |
| Secret Key verschlüsseln | **J** |  |
| verschlüsselten Sekret Key und verschlüsselte Datei/ String an Empfänger | **J** |  |
| Empfänger entschlüsselt Secret Key | **J** |  |
| mit Secret Key wird Datei/ String entschlüsselt | **J** |  |
| CIA-Konform! | **J** | Siehe unten. |
| Github, mit Dokumentation | **J** | Siehe generierte **javadoc**.  Verzeichnis: /doc |
| Diffie-Hellman-Agreement! | **J** |  |

**CIA-Konform:**

**Confidentiality:**

* Sicherer Schlüsselaustausch via DH mit 512 Byte Länge.
* Sichere symmetrische AES-128 Verschlüsselung.

**Integrity:**

* Berechnung und Übertragung eines SHA256 Message Digest ermöglicht einen einfachen Integritätscheck der Nachricht.
* Eine vollständige Integritätsprüfung der Daten sowie des Absender, kann nur durch den Einsatz einer Certificate Authority (+ RSA-Keys & Signaturen) ermöglicht werden.

**Availability:**

* Multithreaded Server stellt eine hohe Verfügbarkeit aus Clientsicht dar. (Jeder Verbindungsaufbau wird in einem eigenen Thread behandelt)
* TCP basierte Übertragung stellt Retransmission bei Packet-Loss sowie Laststeuerungsmechanismen sicher.
* AES-128 ist mit den meisten Java-Implementierungen kompatibel (AES-256 nicht).