

**FH Oberösterreich / Campus Hagenberg**

SFS – Projekt

.NET Crypto API Overview

|  |  |
| --- | --- |
| **Autoren:** | Martin Haslehner  Philipp Haider |
| **Abgabetermin:** | 29.1.2016 |
| **Semester:** | 1. Semester |
| **Lehrender:** | Mag. Dr. Dietrich Birngruber |

**Inhaltsverzeichnis**

[1 Was wurde gemacht? 1](#_Toc441697772)

[2 Lizenz-Server 1](#_Toc441697773)

# Was wurde gemacht?

Im Rahmen dieser Projektarbeit wurde die .NET Crypto API und den damit verbundenen Komponenten studiert. Es gibt in diesem Bereich zwei sehr ähnliche APIs. Eine davon ist die seit Windows NT 4.0 verfügbare Crypto API, kurz CAPI. Seit Windows Vista existiert aber mit der Cryptography API: Next Generation, kurz CNG, bereits ein Nachfolger. Obwohl noch beide Varianten verwendbar sind, wurde in dieser Projektarbeit hauptsächlich mit der neuen Variante gearbeitet.

Praktisches Wissen im Umgang mit der CNG wurde in Form von Code-Beispielen angeeignet. Diese Code-Beispiele umfassen verschiedenste Problemlösungen im Bereich der Kryptographie:

* Symmetrische Verschlüsselung: Advanced Encryption Standard (AES)
* Asymmetrische Verschlüsselung: Rivest Shamir Adelman (RSA)
* Schlüsselaustausch: (Elliptische Kurven) Diffie-Hellman
* Hashing: SHA-2
* Signatur: (Elliptische Kurven) Digitally Signature Algorithm (DSA), RSA

Die Code-Beispiele sind im beigefügten Projekt als ausführbare F#-Skripte implementiert. Darüber hinaus wurde mit einem Floating-License-Server ein größeres Beispiel realisiert.

# Lizenz-Server

Bei dieser Aufgabenstellung kam ein iteratives Vorgehensmodell zur Anwendung. Ausgangsbasis bildete die einfachste mögliche Umsetzung. In jeder Iteration wurde versucht nur eine Bedrohung zu identifizieren und diese umgehend zu beheben.

Damit entwickelte sich in mehreren Schritten eine immer robustere und sicherere Lösung. In dieser Aufgabenstellung war hauptsächlich die Verwendung von digitalen Signaturen bzw. Public-Private-Key-Verfahren notwendig.

Abbildung 1 zeigt die einfachste Implementierung und Abbildung 2 die Implementierung nach der Anwendung mehrerer Mitigations.

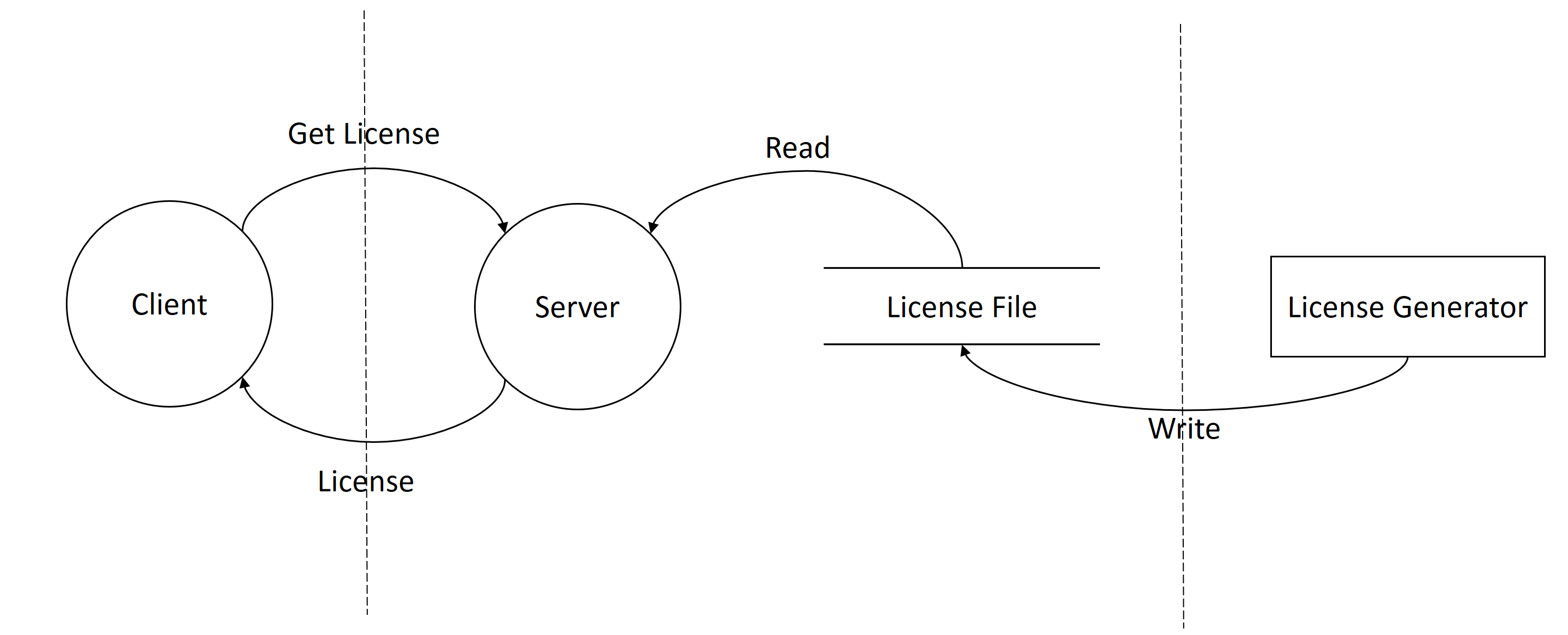


Abbildung 1: Erste (unsichere) Implementierung

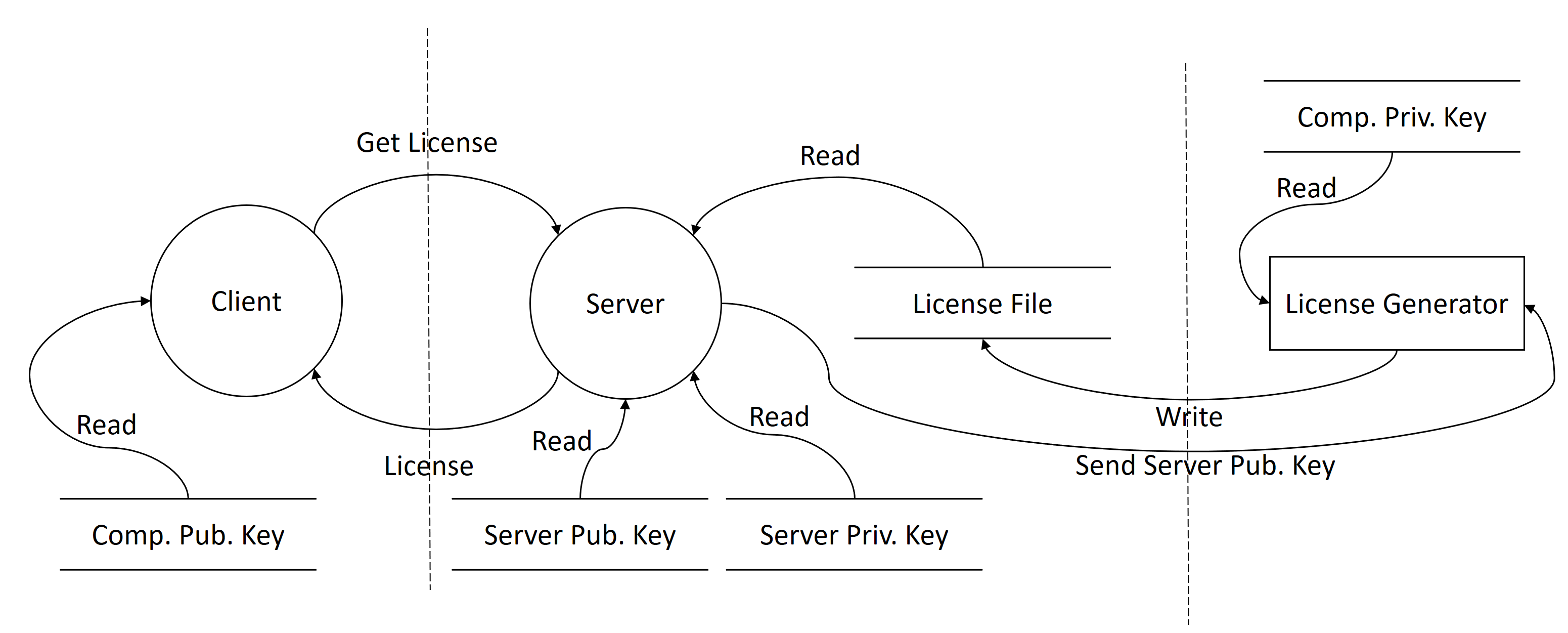


Abbildung 2: Implementierung im Endausbau

Der Endausbau lässt sich wie folgt zusammenfassen:

1. Der Lizenz-Server generiert ein Schlüssel-Paar, sodass der private Schlüssel im CNG Schlüsselspeicher liegt und nicht exportierbar ist.
2. Beim Kauf von Lizenzen muss der Kunde den Lizenzserver installieren und den generierten öffentlichen Schlüssel dem Hersteller mitteilen.
3. Der Hersteller generiert für jede Lizenz eine zufällige (sichere) Zufallszahl, ergänzt diese um den öffentlichen Schlüssel des Servers und signiert das mit seinem privaten Firmenschlüssel. Dieser Schlüssel verlässt den Hersteller nie. Die generierten Lizenzen werden in Form einer Datei dem Server bereitgestellt.
4. Der Server liest die Lizenzdatei und stellt einen Web-Service zur Verfügung, mit dem Klienten Lizenzen anfordern und freigeben können.
5. Bevor der Lizenz-Server eine Lizenz an einen Klienten sendet, signiert er die Lizenz mit seinem privaten Schlüssel. Damit ist sichergestellt, dass nur dieser Server diese Lizenz-Datei ausliefern darf.
6. Der Client fordert eine Lizenz an. Ist keine Lizenz verfügbar, gibt der Server den HTTP-Status-Code 403 zurück. Andernfalls das von ihm signierte Lizenz. Damit eine Lizenz vom Client akzeptiert wird, prüft dieser ob die Signatur des Servers, mit dem öffentlichen Schlüssel der in der Lizenz enthalten ist, übereinstimmt. Anschließend überprüft er ob der öffentliche Schlüssel des Herstellers mit der Signatur der Lizenz übereinstimmt. Damit ist sichergestellt, dass nur der Hersteller Lizenzen ausstellen kann und diese nur von genau einem Server gültig sind.

Eine wichtige Erkenntnis von diesem Fallbeispiel und der einhergehenden Threat-Analyse ist: Wenn der Kunde die ausführbaren Dateien und Bibliotheken auf seinem System hat, kann er diese dekompilieren, egal ob managed oder unmanaged Code. Sobald er sie dekompiliert hat, kann er alle Sicherheitsmaßnahmen umgehen. Dafür gibt es viele prominente Opfer.