Praktikum
Entwicklung von JavaCard-Applets

P. H. Schmitt, B. Beckert

WS 2000/2001

Web-Seite zum Praktikum

`i12www.ira.uka.de/~beckert/Lehre/Praktikum-JavaCard/`

Szenarium

Es gibt eine *Service-Provider*, bestehend aus dem Host-Rechner mit Bildschirm, Tastatur und iButton[1]-Lesegerät, der einen Service zur Verfügung stellt. Als Beispiel für den Service denke man sich die Bereitstellung eines Internetzugangs (Internet-Cafe).

Außerdem gibt es mehrere Benutzer, von denen jeder einen Java-iButton hat. Die Benutzer können mit Hilfe ihres Buttons den Service in Anspruch nehmen.

Jeder Benutzer hat beim Provider ein Konto, das nach einem gewissen Zeitraum abgerechnet wird. Wie die Abrechnung erfolgt, spielt hier keine Rolle. Die anfallenden Kosten sind eine Funktion der Zeitdauer, für die der Service in Anspruch genommen wurde. Wir nehmen an, die Kostenfunktion sei nicht-monoton (es gebe beispielsweise einen Spezialtarif für Langzeitnutzer). Daher haben beide, der Nutzer und der Provider, ein Interesse, dem jeweils anderen die Nutzungsdauer nachweisen zu können.

Der Provider stellt den Service bereit, nachdem der Button in den Leser gesteckt und eine PIN eingegeben wurde. Eine Sitzung wird vom Benutzer dadurch beendet, daß er den Button aus dem Lesegerät zieht. Der Provider kann die Sitzung dadurch beenden, daß er den Service nicht mehr zur Verfügung stellt (dann muß der Benutzer den Button aus dem Leser ziehen, um sicher sein zu können, daß sein Konto nicht mehr belastet wird). Die Sitzung dauert also pder Definition so lange, wie der Button nach Eingabe der PIN im Lesegerät steckt.

Sicherheitskonzept

Das auf dem Button und dem Provider-Rechner implementierte Protokoll soll sicherstellen, daß:

- Der Provider dem Benutzer die Nutzungsdauer nachweisen kann.
- Der Benutzer (mit Hilfe des Buttons) dem Provider die Nutzungsdauer nachweisen kann.

[1] Für dieses Seminar verwenden wir statt einer SmartCard mit Java den iButton mit Java, auf dem ebenfalls JavaCard läuft.

- Ein Dritter (beispielsweise ein Dieb), der den Button besitzt aber die PIN nicht weiß, den Service nicht zu lasten des zum Button gehörenden Kontos in Anspruch nehmen kann; das gleiche soll gelten, wenn der Dritte die PIN weiß aber den Button nicht besitzt.

- Der Provider, wenn er unberechtigt einen Button hat, aber nicht weiß, zu welchem Konto er gehört, und er die PIN nicht kennt, das zum Button gehörende Konto nicht belasten kann. Ebenso kann er kein Konto belasten, zu dem er den Button nicht besitzt, selbst wenn er die PIN weiß.

Man beachte, daß Provider und Nutzer/Button dem jeweils anderen nur nachweisen können, daß die Nutzungsdauer *mindestens* einen bestimmten Wert hat; die Nicht-Benutzung kann nicht nachgewiesen werden.

Die PINs sollen auf dem Provider-System so verschlüsselt sein, daß man durch ein Eindringen in dieses System die PINs nicht bekommt.

Neben den Sitzungen bei denen der Service in Anspruch genommen wird, gibt es eine besondere initiale Sitzung, bei der das Konto eingerichtet wird, die Public keys ausgetauscht werden, und die PIN vereinbart wird. Die Anmeldung und eine Inanspruchnahme des Services kann nicht in derselben Sitzung erfolgen. Das Protokoll für die initiale Sitzung soll ebenfalls im Rahmen des Praktikums spezifiziert und implementiert werden.

Folgende Annahmen können gemacht werden:

- Es ist davon auszugehen, daß der Vorgang, bei dem die Software auf den Button gespielt wird, sicher ist; dabei gibt es keinen Angriff.

- Die Verbindung zwischen Provider und Button ist – zumindest bei initialen Sitzungen – sicher; die übertragenen Daten können nicht abgehört oder verfälscht werden.

- Die Frage, wie der Provider bei oder vor der initialen Sitzung die Bonität und Identität des Benutzers überprüft, spielt hier keine Rolle.

Grob-Entwurf des Verfahrens

Die Kommunikation zwischen Provider-System und Button beruht auf einem asymmetrischen Kryptographie-Verfahren (Public-key-Verfahren).[2]

Die beteiligten haben (nach der initialen Sitzung) folgendes Wissen:

Benutzer: PIN.

Karte: PIN, Kontonummer, eigenen Public und Private key, Public key des Providers.

Provider: eigenen Public und Private key, eine Tabelle in der für jedes Konto enhalten ist: Kontonummer, Public key des Buttons.

[2] Im Rahmen des Praktikums ist es nicht notwendig, die Daten tatsächlich zu verschlüsseln. Die Funktionen zur Ver- und Entschlüsselung mit einem Public-key-Verfahren können also durch Dummy-Funktionen ersetzt werden.

Protokoll während einer Sitzung: In regelmäßigen Zeitabständen bestätigen sich Button und Provider gegenseitig, daß eine bestimmte Zeit vergangen ist; diese Bestätigung authentisieren sie durch elektronische Unterschrift, was vom jeweils anderen überprüft wird. Außerdem wir die Bestätigung mit dem Public key des jeweils anderen verschlüsselt, um die Übertragung sicher zu machen.

Der Button überprüft die vergangene Zeit mit Hilfe der eingebauten Echtzeituhr.

Struktur der Dokumentation

Die Spezifikation der Protokolle und der Software sowie eine Beschreibung des Implementierung sollen zu einem Dokument zusammengefaßt werden. Dieses Dokument soll in etwa die folgende Struktur haben:

- Einleitung: Aufgabenstellung, gemachte Erfahrungen, aufgetretene Probleme, etc.
- Grobentwurf: Spezifikation der Protokolle (einschließlich der initialen Sitzung).
- Feinentwurf: Spezifikation der APDUs, Beschreibung der einzelnen Klassen.
- Implementierung

Sowohl zum Grob- als auch zum Feinentwurf sollen mit einem UML-CASE-Tool Klassendiagramme und Interaktionsdiagramme erstellt werden.