Nutzerauthentifizierung

Dozent: Prof. Dr. Michael Eichberg
Kontakt: michael.eichberg@dhbw.de
Version: 1.1.1

Folien: [HTML] https://delors.github.io/sec-nutzerauthentifizierung/folien.de.rst.html
[PDF] https://delors.github.io/sec-nutzerauthentifizierung/folien.de.rst.html.pdf
Fehler melden: https://github.com/Delors/delors.github.io/issues

1. Grundlagen

Challenge-Response Authentifizierung

🔎 Beobachtung

Die Verwendung einer (kryptographischen) Hashfunktion alleine ist nicht ausreichend zur sicheren *Benutzerauthentifizierung über eine nicht-sichere Verbindung*.

Eine einfache Replay-Attacke ist möglich.

Ein einfaches Protokoll basierend auf einer Hashfunktion f wäre:

Challenge-Response Protokoll ≙ 🏴 *Herausforderung- und Antwortprotokoll*

Alice	unsicherer Kanal	Bob
Gibt Benutzerkennung ID ein	→ sendet ID	sucht zu ID Schlüssel K in der Datenbank
		↓
Gibt Passwort K' ein	sendet r ←	wählt zufällige Zahl r
		↓
berechnet: $Res' = f(K', r)$	→ sendet Res'	$f(K, r)? = Res'$

? Frage

Wie bewerten Sie die Sicherheit (dieses Protokolls/Ansatzes)?

Zero-Knowledge Protokolle

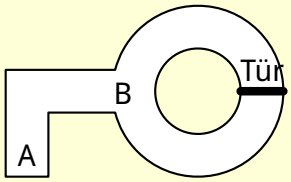
> Die Idee ist, dass man jemanden davon überzeugen möchte, dass man eine bestimmte Information hat, ohne diese Information zu offenbaren.

Beispiel

Der geheimnisvolle Geheimgang

Peggy möchte Victor überzeugen, dass Sie den Code zur Tür kennt, ohne ihn zu offenbaren.

- Peggy wählt einen der Wege zur Tür, während Victor an der Stelle A steht und darauf wartet, dass Sie bei der Tür ist.
- Sobald Peggy Bescheid gibt, dass Sie an der Tür angekommen ist, geht Victor zu Punkt B und sagt Peggy auf welchem Weg sie zurückkommen soll.
- Kommt sie auf dem falschen Weg zurück, dann kennt sie den Code der Tür (offensichtlich) nicht. Kommt sie auf dem richtigen Weg zurück, könnte es noch immer Zufall gewesen sein mit Wahrscheinlichkeit $\frac{1}{2}$.



Bei n Spielen ist die Wahrscheinlichkeit, dass Peggy immer zufällig den korrekt Weg gewählt hat $\frac{1}{2^n}$.

Viele Zero-Knowledge Protokolle basieren darauf, dass man im Prinzip ein Spiel spielt, das man auch zufällig gewinnen kann. Durch die Wiederholung des Spiels wird die Wahrscheinlichkeit jedoch für permanentes zufälliges Gewinnen sehr schnell sehr klein (exponentiell). Somit kann man für praktische Zwecke hinreichend sicher sein, dass der Beweisführende (im Beispiel Peggy) über das Wissen verfügt, das er vorgibt zu besitzen, wenn er immer gewinnt.

Nach 20 Runden ist die Wahreinschlichkeit nur noch $1/2^{20} = 1/1048576$.

Mit 128 Runden erreicht man ein Sicherheitsniveau, das vergleichbar ist mit anderen kryptographischen Verfahren (AES-128, SHA-256, ...).

Fiat-Shamir Protokoll

Voraussetzungen

- gegeben zwei zufällige Primzahlen p und q und $n = p \cdot q$
- Peggys geheimer Schlüssel s wird dann zufällig bestimmt; $s \in \mathbb{Z}_n^*$ und s coprimal zu n.
- Der öffentliche Schlüssel wird dann wie folgt berechnet: $v = s^{-2} \mod n$. Der öffentlichen Schlüssel besteht dann aus den zwei Zahlen v und n.

Protokoll

1. Peggy berechnet x unter Verwendung einer beliebigen Zufallszahlen $r \in \mathbb{Z}_n^*$:

   $x = r^2 \mod n$
2. Peggy sendet die Zahl an Victor.
3. Victor wählt zufällig ein Bit $b \in \{0, 1\}$
4. Peggy berechnet $y = r \cdot s^b \mod n$
5. Peggy sendet y an Victor.
6. Victor testet: $x \cdot v^{-b} \mod n \stackrel{?}{=} y^2 \mod n$

Übung: klassisches Fiat-Shamir-Protokoll

1.1. Beispiel: Fiat-Shamir-Protokoll mit kleinen Zahlen

Zwei Parteien, Peggy (die sich authentifizieren möchte) und Victor (der Prüfer), führen das Fiat-Shamir-Protokoll durch. Verwenden Sie die folgenden Werte:

- $p = 3, q = 7 \rightarrow n = p \cdot q = 21$
- Peggys geheimer Schlüssel ist $s = 2$
- Peggy wählt die Zufallszahl $r = 4$
- Victor wählt die Herausforderung $b = 1$

Beantworten Sie folgende Fragen:

1. Berechnen Sie den öffentlichen Schlüssel v.
2. Berechnen Sie den Wert x, den Peggy an Victor sendet.
3. Berechnen Sie die Antwort y, die Peggy an Victor sendet.
4. Führen Sie die Verifikation als Victor durch