IT-Sicherheit
Prof. Dr. Michael Zohner
Wintersemester 2021/2022

Hochschule Fulda
University of Applied Sciences

Übung 5: Digitale Signaturen und Hashfunktionen v1.4*

Aufgabe 1 Gegeben seien die folgenden DSA Werte:

- Primzahlen $p = 47$ und $q = 23$ sowie Generator $g = 25$
- Der geheime Schlüssel $K_D = x = 13$
- Zufallszahl $k = 9$
- Nachricht $M = 18$
- Die Zwischenwerte: $k^{-1} = 18$ und $s^{-1} = 7$.

Führen Sie die DSA Algorithmen zur Schlüsselerzeugung sowie zum Signieren und zum Verifizieren durch und geben Sie dabei die fehlenden Werte an:

A) Den Verifikationsschlüssel y

B) Die Signatur der Nachricht (r, s)

C) Den Verifikationswert v, welcher mit r verglichen wird

Aufgabe 2 Gegeben seien die folgenden Funktionen f wobei alle Werte der Funktionen bekannt sind, außer anderweitig angegeben:

1. $f(M) = a \cdot M + b \mod p$
   für zufällige und bekannte Zahlen a und b sowie einer Primzahl p, die alle Bitlänge von 2048 haben.
2. $f(M) = g^M \mod p$
   für eine bekannte Basis g und bekannte Primzahl p, die beide eine Bitlänge von 2048 haben.
3. $f(M) = (g^M \mod N) \mod 2^{128}$
   für eine bekannte Basis g, einen bekannten 4096-bit Modulus $N = p \cdot q$ und **un**bekannten Primzahlen p, q.

Welche der Eigenschaften für kryptographische Hashfunktionen (Einwegeigenschaft, schwache- oder starke Kollisionsresistenz) sind jeweils für die Funktionen erfüllt?

Aufgabe 3 Weisen Sie die Korrektheit von DSA nach, indem Sie zeigen, dass für korrekt generierte Signaturen gilt: $v = r$.

Aufgabe 4 - Bonuspunkteaufgabe (TaskID 5A) Die Spielstation 3 führt nur Software aus, die von der Hersteller*in mittels DSA signiert wurde. Auf diese Weise soll verhindert werden, dass Nutzer*innen wahlfreie Software auf dem Gerät ausführen. Aufgrund einer unzureichenden Autorisierung haben Sie die Möglichkeit, sich für kurze Zeit gültige Signaturen zu wahlfreien Nachrichten vom Signaturserver berechnen zu lassen. Sie finden außerdem heraus, dass der Zufallswert k zum Signieren einer Nachricht M wie folgt berechnet wird:

$$k = g^{\text{SHA2}-40(M+z_1)+z_2} \mod q$$

wobei g und q konstante Werte sind, die Werte z_1 und z_2 zufällig gewählt wurden aber für alle Signaturen gleich sind und die Funktion $\text{SHA2} - 40$ die letzten 40-bit des $\text{SHA2} - 256$ Algorithmus ausgibt.
Nutzen Sie Ihren Zugang zum Signaturserver um den geheimen Signaturschlüssel x zu rekonstruieren. Laden Sie den Signaturschlüssel x auf dem Moodle Element "Abgabe Übung 5: DSA Signatur" hoch. Nutzen Sie die Task-ID "5A" unter Eingabe einer Nachricht M um die Werte p, q, g, z_1 sowie eine DSA Signatur (r_M, s_M) zur Nachricht M aus der Übungsumgebung zu erhalten.

Hinweise:

- Der Wert $z_1 + M$ muss für die Eingabe in die Hashfunktion SHA2 als Byte Array dargestellt werden. Der Wert des Byte Arrays soll dabei gleich dem Dezimalwert von $z_1 + M$ sein. Achten Sie dabei auch auf die Position des höchstwertigen Bytes!
  Beispiel: Die Dezimalzahl 42435 ist im Hexadezimalformat $0xA5C3$ und würde als Byte Array dargestellt werden als $arr = \{0xC3, 0xA5\}$.

- Sie können den privaten Schüssel x bei bekanntem k berechnen als:

$$x = r^{-1} \cdot ((k \cdot s_1) - M_1) \mod q \tag{1}$$

Um den Wert k zu ermitteln, wenn dieser für zwei unterschiedliche Nachrichten M_1 und M_2 und Signaturen s_1 und s_2 verwendet wurde, berechnen Sie:

$$\begin{aligned} s_d &= (s_1 - s_2) \mod q \\ M_d &= (M_1 - M_2) \mod q \\ k &= (M_d \cdot s_d^{-1}) \mod q \end{aligned} \tag{2}$$

wobei s_d^{-1} und r^{-1} das modulare Inverse von s_d bzw. r modulo q sind (Berechnung via erweitertem Euklidschen Algorithmus).

- Sie können den Signaturschlüssel x als Dezimalzahl mit der Aufgaben-ID "5AV" verifizieren (Antwort als OK/NOK).
- Zur Verarbeitung beliebig großer Zahlen gibt es die Klassen BigInteger in Java bzw. Integer in C++ (via Crypto++).
- Die Abgabefrist ist der 06/Dezember, 23:59 Uhr. Ein erfolgreiches Bestehen wird via Moodle mitgeteilt.

***Versionshistorie** v1.0: Initialversion. v1.1: Hinweis zur Eingabe des Wertes $z_1 + M$ in die Hashfunktion und Aufgabe 3 neu hinzugefügt. v1.2: Aufgabe 1 upgedated und Aufgabe 3 als Hinweis für Bonuspunkteaufgabe aufgelöst. v1.3: Text geupdated - in Aufgabe 4 wird der Wert p statt dem Wert y von der Übungsumgebung ausgegeben. v1.4: Hinweise zur Berechnung von k in Aufgabe 4 hinzugefügt.