Aufgabe 11

Asymmetrische Kryptografie

Im Gegensatz zu symmetrischen Verfahren, verwenden asymmetrische Verfahren 2 unterschiedliche Schlüssel zum Ver- bzw. Entschlüsseln:

- einen Public-Key (öffentlicher Schlüssel – für jedermann zugänglich) und
- einen Private-Key (secret key – geheim)

In dieser Aufgabe geht es neben ein paar einführenden Begriffen um den Diffie-Hellman-Schlüsseltausch und um das RSA-Verfahren.

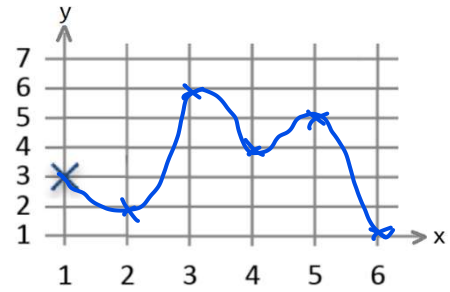
Aufgaben

a) Suchen Sie aus dem **Herdt-Buch „Netzwerke-Sicherheit“ das Kapitel 11** folgende Informationen:

- Was ist eine Einwegfunktion? (S. 127)
- Was ist eine Trapdoor-Einwegfunktion? (S.128)
- Bei kryptografischen Verfahren werden häufig sogenannte Restwert-Operationen (Modulo-Operationen) verwendet. Was versteht man unter „Modul“? (S128)
- Wie wird der Modulo-Operator in der Programmiersprache C bzw. Java geschrieben?
- Um zu sehen, warum es sich bei der diskreten Exponentialfunktion um eine Einwegfunktion handelt, schreiben Sie die Wertetabelle der diskreten Exponentialfunktion $y = 3^x \bmod 7$ auf.

x	1	2	3	4	5	6
y	3	2	6	4	5	1

Zeichnen Sie die Werte in die Grafik ein:



Daraus sollte erkennbar sein, dass es zu dieser diskreten Exponentialfunktion keine einfache Umkehrfunktion gibt, welche zu einem bestimmten y-Wert den x-Wert bestimmen kann.

b) Diffie-Hellman-Schlüsseltausch

- Auf welcher Einweg-Funktion basiert der DH-Schlüsseltausch?

- Lesen Sie sich das **Kapitel 11.3 aus dem Herdt-Buch „Netzwerke-Sicherheit“** (S. 131) durch.

- Obwohl in der Praxis viel größere Zahlen verwendet werden, probieren wir den angegebenen Ablauf des DH-Schlüsseltauschs mit kleinen Zahlen aus.

  Alice und Bob einigen sich beide auf eine Primzahl p und eine Zahl g:
  zB p = 7 und g = 3
  Diese Zahlen sind nicht geheim (public) und können auch über das Internet geschickt werden.
  Alice wählt eine natürliche Zahl a mit a<p aus und behält sie für sich (geheim).
  zB a=2
  Bob wählt eine natürliche Zahl b mit b<p aus und behält sie für sich (geheim).
  zB b=5

  Alice berechnet die Zahl $\alpha = g^a \bmod p$ = _____
  Bob berechnet die Zahl $\beta = g^b \bmod p$ = _____

  Diese Zahlen α und β werden nun jeweils zum anderen Teilnehmer geschickt.

  Alice bekommt β und berechnet daraus den Key K1 $K1 = \beta^a \bmod p$ = _____
  Bob bekommt α und berechnet daraus den Key K2: $K2 = \alpha^b \bmod p$ = _____

  Wenn Sie richtig gerechnet haben, sollten K1 = K2 = K (gemeinsamer Schlüssel) sein.
  Begründung:
  $K1 = \beta^a \bmod p = (g^b)^a \bmod p = g^{b*a} \bmod p$
  $K2 = \alpha^b \bmod p = (g^a)^b \bmod p = g^{a*b} \bmod p$
  } ident

  Jemand der die Verbindung abhört (Mallory), kann zwar α und β mitlesen, aber daraus nicht a oder b berechnen.
  Dazu müsste Mallory die Umkehrung der diskreten Exponentialfunktion lösen, d.h. den diskreten Logarithmus lösen, was nach heutigem Stand (wenn die Zahlen ausreichend groß sind) nicht mit vertretbarem Aufwand möglich ist.
  Ohne a bzw b kann der Key K auch nicht berechnet werden.

  Nur Bob und Alice sind nun im Besitz des geheimen Schlüssels K.

c) RSA-Verfahren

- Woher kommt die Abkürzung RSA?

- Auf welcher Einweg-Funktion basiert das RSA-Verfahren?

- Zur Verschlüsselung mittels RSA-Verfahren müssen sich die Teilnehmer jeweils ein Schlüsselpaar (private und public key) erzeugen.
  Obwohl in der Praxis auch hier wieder große Zahlen verwendet werden müssen, probieren wir die Schlüsselpaar-Erzeugung wieder anhand kleiner Zahlen aus.

  Schlüsselerzeugung:
  - Mit einem geeigneten Verfahren erzeugt man zwei zufällige unterschiedliche Primzahlen p und q und berechnet ihr Produkt n = p * q (dies ist auch die nicht umkehrbare Funktion). n heißt RSA-Modul.
    zB p = 23 und q = 47
    RSA-Modul n = p * q = 1081
  - Mit den beiden Primzahlen kann man den im Folgenden benötigten Wert der Eulerschen-Funktion φ(n) berechnen: φ(n) = (p-1)*(q-1) = 1012
  - Man wählt eine Zahl e mit 1<e< φ(n), die zu φ(n) teilerfremd ist, d.h. der größte gemeinsame Teiler ggT(e, φ(n))=1. Dieser sogenannte Verschlüsselungsexponent e bildet zusammen mit dem RSA-Modul den öffentlichen Schlüssel.
    zB e = 3
  - Das RSA-Schlüsselpaar wird komplettiert durch den privaten Schlüssel d.
    d heißt auch Entschlüsselungsexponent und muss so gewählt werden, dass 1<d< φ(n) und insbesondere e*d = 1 mod φ(n) gilt.

    zB d=675 erfüllt diese Bedingung. Test: e*d mod φ(n) = 3 * 675 mod ____ = ____

  Dadurch ergeben sich:
  - Privater Schlüssel (n,d)
  - Öffentlicher Schlüssel (n,e)

- Ver- und Entschlüsselung:
  Ein Klartext m mit 0≤m<n wird durch den RSA-Algorithmus folgendermaßen verschlüsselt:
  $m^e \bmod n = c$
  Klartexte m≥n müssen in Blöcke aufgespalten werden, so dass jeder Block kleiner oder gleich n ist. Diese Blöcke können dan einzeln verschlüsselt werden. Auf diese Weise ist es also möglich, Klartexte in beliebiger Länge zu verschlüsseln.

  Die Entschlüsselung erfolgt nach demselben Verfahren wie die Verschlüsselung nur dass statt dem Klartext m der Chiffretext c und statt e der private Schlüssel d eingesetzt wird:
  $c^d \bmod n = (m^e)^d \bmod n = m$

- Test der Ver- und Entschlüsselung:
  Testen wir wieder die Ver- und Entschlüsselung anhand eines Klartextes:
  Angenommen Alice möchte Bob den Text „HTL“ mittels RSA-Verschlüsselung verschlüsselt schicken, so dass nur Bob den Text wieder entschlüsseln kann.
  Dazu erzeugt sich Bob ein Schlüsselpaar (wir nehmen gleich das Schlüsselpaar vom vorigen Beispiel) und schickt Alice seinen öffentlichen Schlüssel.

  Mit Bob's öffentlichen Schlüssel wird Alice nun die Nachricht verschlüsseln:
  Klartext: H T L -> als Kodierung nehmen wir die ASCII-Tabelle (A=65, B=66, C=67…)

Klartext	H	T	L
m	72	84	76

  Wir werden jeden Buchstaben einzeln verschlüsseln:
  $c = m^e \bmod n$

m	72	84	76
c	303	316	90

  Dieser verschlüsselte Chiffretext c wird nun an Bob gesendet.
  Bob kann nun mit seinem privaten Schlüssel (und auch nur Bob, den niemand anders hat den privaten Schlüssel von Bob) diesen Text wieder entschlüsseln.
  $m = c^d \bmod n$

c	303	316	90
m	72	84	76

  Zur Berechnung können Sie zB folgende Seite verwenden: www.wolframalpha.com