Technische Universität München
Informatik VIII
Netzarchitekturen und Netzdienste
Prof. Dr.-Ing. Georg Carle

Dipl.-Inf. Ali Fessi

Übungen zur Vorlesung „Netzsicherheit“
Übungsblatt 3, WS08/09

Abgabe: 26. Nov. 2008
Besprechungstermin: 03. Dez. 2008

Aufgabe 1: Kryptographische Hash-Funktionen – Grundlagen

a) Wie viele Bits soll eine ideale kryptographische Hash-Funktion als Ausgabewert haben, damit der Aufwand, eine Kollision zu finden, mit einem Brute-Force Angriff auf einer symmetrischen Blockschiffre mit 100 Bits Schlüssellänge vergleichbar wäre?
b) Welche Gefahr muss man beachten, wenn man MD5 für Digitale Signaturen verwendet möchte.

Aufgabe 2: Kryptographische Hash-Funktionen – Der MD5-Standard

Der Standard MD5 (Message Digest 5) ist im Request for Comments (RFC) Nr. 1321 definiert. Dies ist auf der Web Seite der Internet Engineering Task Force (IETF) unter: http://www.ietf.org/rfc/rfc1321.txt zu finden.

a) Ermitteln Sie aus dem Abschnitt „Executive Summary“ die wichtigen Eigenschaften von MD5, die schon aus der Vorlesung allgemein für kryptographische Hash-Funktionen bekannt sind.
b) Die Berechnung eines MD5-Wertes besteht aus 4 Runden mit jeweils 16 Rechenschritten. Für die Berechnung wird bei jeder Runde eine andere logische Funktion verwendet (in der Vorlesung mit *g* bezeichnet). Ermitteln Sie diese logischen Funktionen aus dem Abschnitt 3.4 von RFC 1321. Beachten Sie hierbei die Notation, die in Abschnitt 2 erläutert wird.
c) Für die Berechung bei jedem Schritt bei MD5 wird:
   - ein Wert aus einer Tabelle *T[j]* (der mit dem Ergebnis der logischen Funktion *g*, dem Inhalt des Registers *A* und dem Eingabetext $y_i[k]$ XOR-verknüpft wird)
   - und ein Wert *s* (für die zyklische Verschiebung CLS_s) benötigt.

Im Abschnitt Appendix 3 (A.3) von RFC 1321 wird eine Referenzimplementierung in der Programmiersprache *C* vorgestellt. Ermitteln Sie aus diesem Code welche Werte für *T[j]* und *s* verwendet werden, beispielweise für Runde 1, Schritt 1, und für Runde 3, Schritt 16.

Aufgabe 3: Kryptographische Hash-Funktionen – SHA-1

a) Bei SHA-1 werden an der Stelle von der Substitutionstabelle *T[j]* wie bei MD5 vier verschiedene Konstanten verwendet. Recherchieren Sie und ermitteln Sie diese Konstanten.
b) Die Berechnung eines Hash Wertes mit dem SHA-1-Algorithmus bei einem Textblock mit 512 Bits erfordert 80 Schritten. Nach wie vielen Schritten ist jedes Bit von der Eingabe in die Berechnung des Hashwertes mit eingeflossen?
c) Wie lässt sich bei SHA-1 der Wert W_{19} berechnen?

Aufgabe 4: Message Authentication Codes

Der Standard HMAC (HMAC: Keyed-Hashing for Message Authentication) ist im RFC 2104 definiert. Dies ist unter: http://www.ietf.org/rfc/rfc2104.txt definiert.

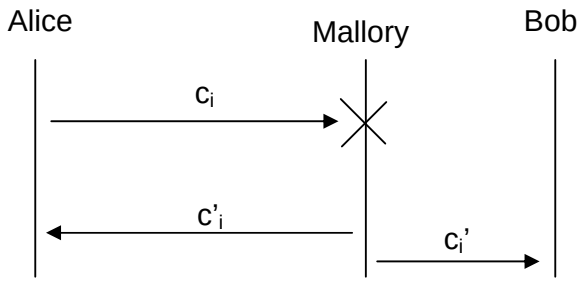
a) Ermitteln Sie aus dem Abschnitt „Abstract“, ob HMAC an einer bestimmten kryptographischen Hash-Funktion gebunden?
b) Ermitteln Sie aus dem Abschnitt „Definition of HMAC“ die Formel, mit welcher der HMAC-Wert einer Nachricht gerechnet wird. Erklären Sie diese Formel.
c) Geben Sie eine andere MAC-Funktion an, die Sie aus der Vorlesung kennen.
d) Kann man MAC-Funktionen für digitale Signaturen verwenden? Begründen Sie Ihre Antwort.

Aufgabe 5: Angriffe auf (Un)sichere Kanäle

Angenommen beim Entwurf eines Protokolls zur Absicherung der Kommunikation zwischen jeweils 2 Kommunikationspartner werden folgende Design-Entscheidungen getroffen:

- Für die Datenintegrität soll eine kryptographische Hash-Funktionen verwendet werden, nämlich SHA-1 und bei jeder Nachricht die ersten 96 Bits des Hash-Wertes anschließend an die Nachricht dran gehängt:
  (m , first-96-bits(H(m)))
- Für die Verschlüsselung soll ein gemeinsamer vorkonfigurierter 256-bit langer Schlüssel *K* verwendet werden.
- Für die Verschlüsselung nimmt man die Performance-Vorteile von AES im CTR-Mode in Anspruch und verwendet den Schlüssel *K* für die Berechnung der notwendigen „Key Streams“. Um zusätzlichen Aufwand für den Schlüsselmanagement einzusparen, wird der selbe Schlüssel *K* in beide Richtungen eines Kommunikationskanals verwendet.
- Für die Generierung der „Key Streams“ k_i werden jeweils die Sequenznummer *i* der jeweiligen Nachricht mit einem Counter *j* für jedes einzelne Block dieser Nachricht zusammen konkateniert (*i* || *j*) und gemeinsam mit *K* als Eingabe für die AES-Block-Chiffre-Funktion verwendet. Die Sequenznummer *i* wird mit Null initialisiert und bei jeder neuen Nachricht um Eines erhöht. *j* wird bei jeder Nachricht mit Null initialisiert und bei jedem Block innerhalb dieser Nachricht um eins erhöht.
  k_i = E(i||0, K) || E(i||1, K) || E(i||2, K) || ….

a) Angenommen „Alice“ und „Bob“ benutzen das oben beschriebene Protokoll für ihre Kommunikation. Ein Angreifer „Mallory“ fängt die verschlüsselten Nachrichten c_i von Alice ab und verhindert sie daran bei Bob anzukommen. Angenommen „Mallory“ kann zusätzlich bei manchen Nachrichten den Plain Text m_i raten. Begründen Sie, warum in diesem Fall:
   - „Mallory“ eigene Nachrichten an Alice c'_i zurück schicken kann, so dass wenn Alice diese Nachrichten zu m'_i entschlüsselt nicht merkt, dass die Nachrichten nicht von Bob kommen.
   - „Mallory“ sogar eigene Nachrichten c''_i an Bob verschicken kann, bei denen Bob nicht unterscheiden kann, ob sie von „Alice“ oder von „Mallory“ kommen.



b) Begründen Sie, warum der Entwurf dieses Protokolls einige Schwachstellen hat, und schlagen Sie Verbesserungen vor, um dieses Protokoll sicherer zu gestalten.