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

Dipl.-Inf. Ali Fessi

Übungen zur Vorlesung „Netzsicherheit“
Übungsblatt 4, WS08/09

Abgabe: 10. Nov. 2008
Besprechungstermin: 17. Dez. 2008

Aufgabe 1: Authentisierungsprotokolle; das Needham-Schröder-Protokoll

a) In der Vorlesung wurde ein Replay-Angriff auf das Needham-Schröder-Protokoll (symmetrische Variante) beschrieben. Welche Voraussetzung muss bzgl. des „Session Keys“ $K_{A,B}$ erfüllt werden, damit dieser Angriff möglich wird.
b) Dieser Angriff wurde bei dem Kerberos-Protkoll behoben. Erweiten Sie das Needham-Schröder-Protokoll, so dass der Replay-Angriff verhindert wird.

Aufgabe 2: Authentisierungsprotokolle; Kerberos

a) Schreiben Sie das Kerberos-Protokoll so um, dass es ohne Zeitstempeln und daher auch ohne synchronisierte Uhren, auskommt.
b) Begründen Sie, warum das Kerberos Protokoll, die Eigenschaft der "`Forward Secrecy'" nicht erfüllt.
c) Erweitern Sie das Kerberos-Protokoll, so dass für die Kommunikation zwischen *Alice* und dem Service *S1* die „Forward Secrecy“-Eigenschaft erfüllt wird.
d) In weit werden die Möglichkeiten für einen Wörterbuch-Angriff bei Kerberos V5 im Vergleich zu V4 eingeschränkt?

Aufgabe 3: Vertrauensmodelle in Public-Key-Infrastrukturen (PKI)

a) Abbildung 1 stellt eine PKI dar, die in einem Baum strukturiert ist[1]. Sei dabei *N* die Anzahl der Knoten in diesem Baum und *d* die Höhe des Baums (d.h. der längste Pfad zwischen einem Blatt und dem Wurzelknoten). Weiterhin gilt, dass jede CA an Hand des Namen eines Benutzers feststellen kann, ob der jeweilige Zertifikat in ihrem Unterbaum liegt.
Angenommen die Zertifikate von Alice und Bob sind in dieser PKI jeweils in einem Blatt des Baumes eingetragen. Schätzen Sie den Aufwand in Abhängigkeit von *d* im „Best case“ und „Worst Case“ ab, um einen Zertifikatskette zwischen Alice und Bob zu bilden.

b) Abbildung 2 stellt eine PKI in einem Graphen im so genannten „User-Centric“ Model dar, sowie es bei *Pretty Good Privacy (PGP)* der Fall ist. Der Einfachheit halber gehen wir davon aus, dass der Graph ungerichtet ist, d.h. wenn z.B. der öffentliche Schlüssel von Alice durch „Alice’ Friend“ signiert ist, dann ist der öffentliche Schlüssel von „Alice’s Friend“ auch durch Alice signiert. Die Zertifikate werden hier in einem zentralen PGP Server gespeichert, der auf Anfrage von einem Benutzer Alice eine Zertifikatskette von Alice zu einem anderen Benutzer Bob liefern kann.

[1] Ein Baum wird in der Graphentheorie als ”ein zusammenhängender Graph, der keine Zyklen enthält“ definiert.

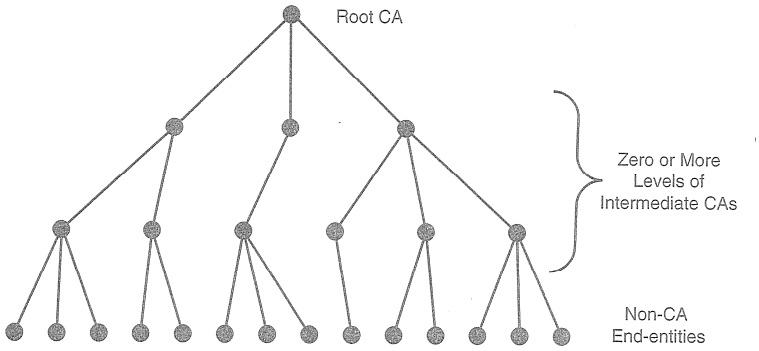

Abbildung 1: PKI mit einer strikten Hierarchie

Sei *V* die Menge der Knoten („Vertices“) und *E* die Menge der Kanten („Edges“) in dem Graphen *G* = (*V*, *E*).

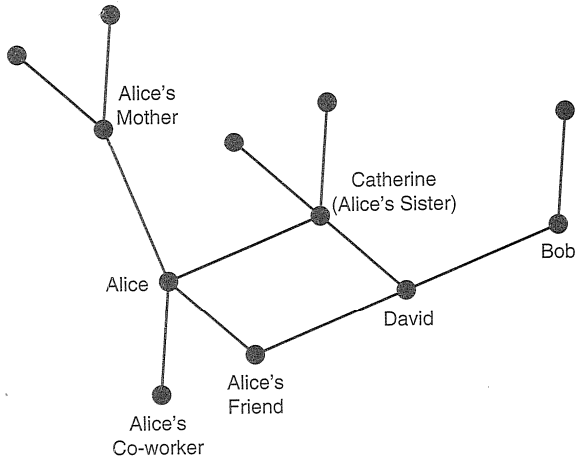

Abbildung 2: PKI mit PGP

Folgender Algorithmus berechnet den kürzesten Pfad von Alice nach Bob. Dabei ist *U* (”unvisited“) die Menge der noch zu bearbeitenden Knoten (am Anfang gilt also *U* = *V*). Wenn der Algorithmus terminiert, dann liefert der Wert *Distanz*(*Alice*) die Länge des Pfades zwischen Alice und Bob.

$\forall v \in V$
    $Distanz(v) := \infty; Vorgaenger(v) := null;$
$Distanz(Alice) := 0; Vorgaenger(Alice) := Alice; U := V;$

$while(U \neq \oslash)$
{
    $Waehle\ u \in U\ so\ dass\ Distanz(u)\ minimal;$
    $U := U - u;$
    $if(u = Bob)$
        STOP;
    $\forall (u, v) \in E$ mit $v \in U$
        if $Distanz(u) + 1 \prec Distanz(v)$
        {
            $Distanz(v) := Distanz(u) + 1;$
            $Vorgaenger(v) := u;$
        }
}

Schätzen Sie den Aufwand ab, um die Zertifikatskette zwischen 2 Benutzern Alice und Bob zu bilden, in Abhängigkeit von der Anzahl der Knoten *N* und der Anzahl der Kanten *M*.

c) Abbildung 3 stellt eine dritte Möglichkeit dar, wie eine PKI strukturiert sein kann: der Graph besteht aus mehreren Bäumen, wobei die Wurzelknoten dieser Bäume untereinander beliebig mit einander verbunden sein können (aber es sind auch nicht alle unbedingt mit einander verbunden).

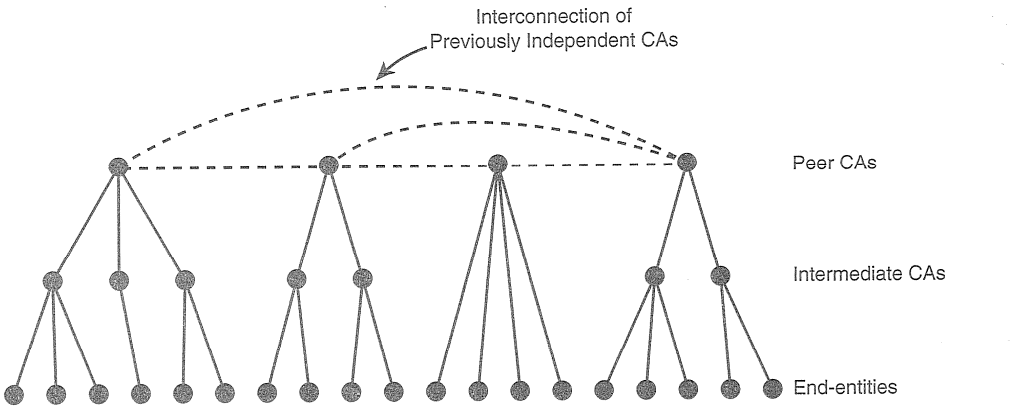

Abbildung 3: Teilweise hierarchische PKI

Sei *n* bei dieser Struktur die Anzahl der obersten CAs, *m* die Anzahl der Kanten, welche die obersten CAs miteinander verbinden und d_{max} die maximale Höhe eines Baumes. Schätzen Sie hier auch den Aufwand im „Worst Case“ ab, um eine „Chain of Trust“ zwischen Alice und Bob zu bilden, in Abhängigkeit von d_{max}, *n* und *m*.