Humbold–Universität zu Berlin - Institut für Informatik

Proseminar
Moderne Verfahren der Kryptographie

Schlüsselmanagement und Angriffe auf Protokolle

1. Trusted Third Parties
   - Motivation
   - öffentliches Telefonbuch“ – Public-Key-Signaturverfahren

2. Protokolle zum Schlüsselmanagement – symmetrische Verfahren
   - GSM – Mobilfunknetz
   - Breitmaulfrosch – Protokoll
   - Verbesserung: Otway-Rees-Protokoll
   - Needham – Schroeder – Protokoll

3. Allgemeine Angriffe auf Protokolle
   - Impersonation
   - Replay – Attacken
   - Chess Grandmaster Problem (Man-in-The-Middle-Attack)

4. Beispielangriff: Simmons-Attacke auf das TMN-Protokoll

1. Trusted Third Parties
   1.1. Motivation

   Um mit einem Partner sicher zu kommunizieren, ist ein Schlüssel notwendig. Wird immer mit den gleichen Partnern kommuniziert, so tauscht man seine Schlüssel aus und das sichere kommunikative Tagesgeschäft kann durchgeführt werden. Wechseln nun aber die Kommunikationspartner häufig, so ist man gezwungen, zuerst einen Schlüssel zu besorgen. Das führt in offenen großen Systemen zu organisatorischen Problemen. Teilnehmer A kann von Teilnehmer B keinen Schlüssel erhalten, da A und B sich nicht kennen. Auch ein öffentlicher Schlüssel eines Public-Key-Verfahrens muß nicht zu 100% zu dem entsprechenden Inhaber gehören, solange man diesen Schlüssel per E-Mail erhalten hat.
   Diese Probleme werden durch die Einführung einer vertrauenswürdigen „dritten Instanz“ gelöst. Diese „Trusted Third Party“ unterstützt symmetrische sowie asymmetrische Systeme.

   1.2. „öffentliches Telefonbuch“ – Public-Key-Signaturverfahren

   Mit dem globalen Einsatz eines Public-Key-Verfahrens wird die Implementierung eines „öffentlichen Telefonbuches“ für die öffentlichen Schlüssel vereinfacht. In diesem Verzeichnis gibt es jeweils eine Zuordnung von Teilnehmern zu ihren öffentlichen Schlüsseln. Diese Verknüpfung wird nun mit dem privaten Schlüssel der Certificate Authority signiert, wobei ein Zertifikat entsteht. Das Zertifikat eines Teilnehmers kann nun mittels des öffentlichen Schlüssels der Trusted Third Party überprüft werden.
   Damit kommt auf die TTL für einen Teilnehmer die Identitätsprüfung sowie die Signierung des öffentlichen Schlüssels zu. Für dieses Verfahren der existiert der Industriestandard X.509. Anwendungen welche solch ein zertifikatbasierendes Schlüsselmanagement nutzen sind zum Beispiel SSL sowie S/MIME.

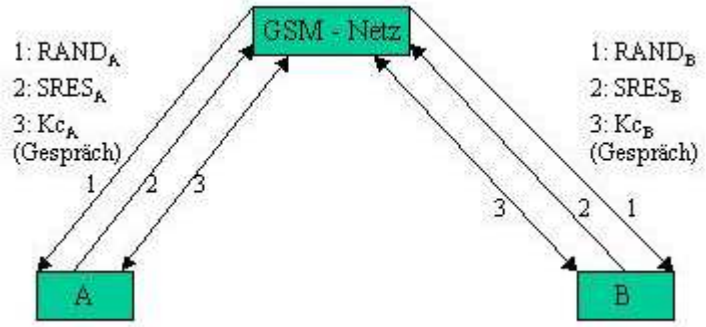
2. Protokolle zum Schlüsselmanagement – symmetrische Verfahren
   2.1. GSM – Mobilfunknetz

   Der GSM – Standard wird in Deutschland sowie in weiteren Ländern zur Mobilfunkkommunikation genutzt. Die Entwicklung war notwendig, da grenzüberschreitende Telefonie notwendig wurde. Zu schützen ist hierbei die Strecke vom Mobiltelefon zum Funkmast, also die Übertragung im Medium Luft. Es soll die Berechtigung zur Telefonie genauso gesichert sein wie die abhörsichere Übertragung der Signale. Dazu kommen der A3-Algorithmus (Authentifikation des Endgerätes), der A5-Algorithmus (Gesprächsverschlüsselung) sowie der A8-Algorithmus (Lieferung des Schlüssels) zur Anwendung.

      2.1.1. Ablauf Protokollkommunikation

      Der A3-Algorithmus nutzt ein „Challenge and Response“ – Verfahren. Das Mobilfunksystem erzeugt eine Zufallszahl RAND und sendet diese an das Endgerät. Dieses erzeugt mittels A3 und seinem persönlichen Schlüssels KI das Ergebnis SRES (signed response), welche zurück an das Mobilfunksystem gesendet wird. Im Mobilfunksystem ist der individuelle Schlüssel des Endgerätes in einer Datenbank, wodurch das Ergebnis SRES auch berechnet werden konnte. Der positive Vergleich beider SRES – Werte berechtigt zur Fortführung der Kommunikation.
      Mit dem A8-Algorithmus, dem individuellen Schlüssel sowie der Zufallszahl wird nun im Endgerät sowie im Mobilfunksystem ein

Sitzungsschlüssel KC errechnet. Die Sprachdaten des Gespräches werden nun digitalisiert um sie auf dem Luftweg mittels dem A5-Algorithmus und dem Sitzungsschlüssel KC zu verschlüsseln.



2.1.2. Kommunikation im GSM – „Ausland“

Im Ausland ist der individuelle Schlüssel des Endgerätes unbekannt. Weiterhin können der A3 sowie der A8-Algorithmus in einer anderen Form implementiert sein. Um die Authentifikation zu sichern, werden dem „ausländischen“ Mobilfunknetz Tripel der Form (RAND, SRES, KC) übermittelt, womit für das Endgerät das Protokoll eingehalten wird.

2.1.3. Replay – Attacken (nicht möglich)

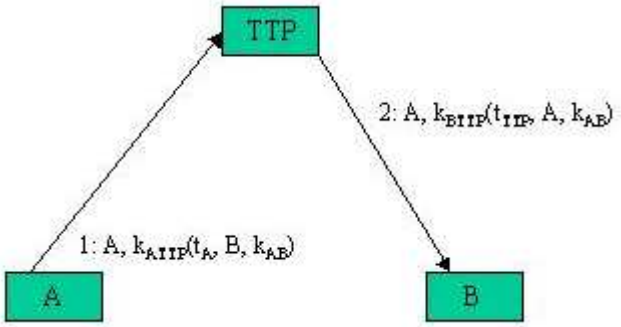
Replay – Attacken sind in der GSM – Implementation nicht möglich, da die gesendeten Nachrichten „frisch“ sind. Der gleiche Inhalt wird in einem anderen Gespräch auch anders verschlüsselt, da der Schlüssel aus einer Zufallszahl entsteht. Ein Abfangen von Nachrichten und das spätere „blinde Versenden“ würde zu keinem Erfolg führen.

2.2. Breitmaulfrosch – Protokoll

Das Breitmaulfrosch – Protokoll ist ein einfaches Verfahren, um zwischen zwei Parteien mittels einer Trusted Third Party (TTP) einen authentischen Sitzungsschlüssel zu vereinbaren.

2.2.1. Ablauf

Teilnehmer A sendet ein Kryptogramm an die TTP, welches mit dem gemeinsamen Schlüssel K_{ATTP} verschlüsselt wurde. In diesem Kryptogramm ist ein Zeitstempel t_A, der Name des gewünschten Kommunikationspartner B sowie der für die Kommunikation mit B gewünschte Sitzungsschlüssel K_{AB} enthalten. Dabei teilt A der TTP seinen Namen mit, da die TTP aus ihrer Datenbank den gemeinsamen Schlüssel K_{ATTP} finden muss. Nun versendet die TTP ein Kryptogramm an B, welches einen neuen Zeitstempel t_s, den Namen des Teilnehmers A sowie den gewünschten Sitzungsschlüssel K_{AB} enthält. Dieses Kryptogramm wird mittels dem Schlüssel K_{BTTP} verschlüsselt. Nun besitzen A und B den gemeinsamen Schlüssel K_{AB}. Vor dem Beginn der Kommunikation müssen Sie sich noch gegenseitig authentifizieren.

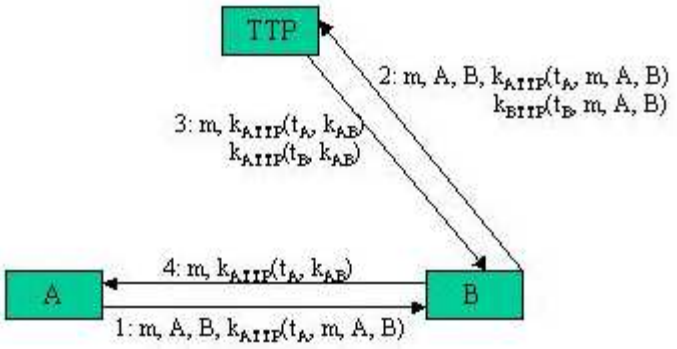


2.3. Verbesserung: Otway-Rees-Protokoll

Das Breitmaulfrosch – Protokoll ist einfach und sicher. Mittels der Trusted Third Party wird aber nur ein gemeinsamer Schlüssel ausgehandelt, wonach die beiden Teilnehmer sich gegenseitig authentifizieren müssen. Das Otway-Rees-Protokoll löst dieses Problem.

2.3.1. Ablauf

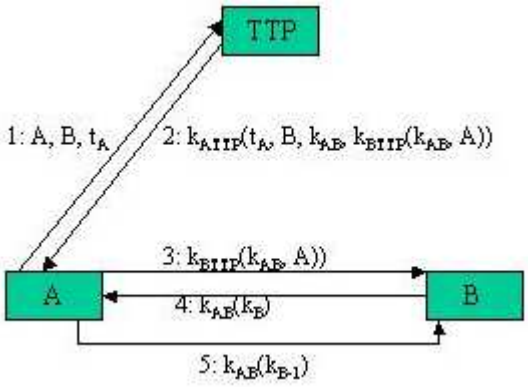
Teilnehmer A sendet zuerst ein Kryptogramm an B, in welchem der Einmalwert t_A, der Name A sowie der Name B enthalten ist. Das Kryptogramm wird mit dem Schlüssel K_{ATTP} verschlüsselt. Teilnehmer B baut nun auch ein mit K_{BTTP} verschlüsseltes Kryptogramm auf, welches den Einmalwert t_B, den Namen A sowie den Namen B enthält. Selbiges Kryptogramm versendet B zumsammen mit dem von A erhaltenen Kryptogramm an die TTP. Diese entschlüsselt beide Kryptogramme womit sie die jeweiligen Einmalwerte t_A sowie t_B erhält. Sie kann nun einen Schlüssel K_{AB} generieren, welcher in die zwei entstehenden Kryptogramme zusammen mit den Werten t_A sowie t_B gelegt wird. Das Paket für Teilnehmer A wird mit K_{ATTP} und das Paket für Teilnehmer B mit K_{BTTP} verschlüsselt. Diese beiden Pakete werden an B versendet, welcher aber nur das für ihn bestimmte Paket mit K_{BTTP} öffnen kann. Ist in diesem der Einmalwert t_B enthalten, so akzeptiert B den Schlüssel K_{AB} und versendet das andere Kryptogramm an A. Dieses konnte er nicht öffenen, da es mit K_{ATTP} verschlüsselt ist. Teilnehmer A entschlüsselt dieses und wird den Sitzungsschlüssel K_{AB} akzeptieren, wenn der Einmalwert t_A enthalten ist. Eine weitere gegenseitige Authentifizierung von A und B ist nicht notwendig, da nur A und B die Nachricht an die TTP gültig verschlüsseln konnten.



2.4. Needham – Schroeder – Protokoll

2.4.1. Ablauf

Nur A tritt in Kontakt mit der TTP, wobei zuerst die Namen von Teilnehmer A und Teilnehmer B sowie ein Einmalwert t_A übermittelt werden. Die TTP berechnet nun ein K_{ATTP} – verschlüsseltes Kryptogramm für A, welches den Einmalwert t_A, den Namen von B, den Sitzungsschlüssel K_{AB} sowie ein Kryptogramm für B enthält. Das Kryptogramm für B ist mittels K_{BTTP} verschlüsselt und enthält den Namen von A sowie den Sitzungsschlüssel K_{AB}. Teilnehmer A überprüft hier den Zeitstempel und leitet im Erfolgsfall das Kryptogramm für Teilnehmer B weiter. Teilnehmer B entschlüsselt das Kryptogramm mittels K_{BTTP} und erhält damit den Sitzungsschlüssel K_{AB} sowie den Namen von Teilnehmer A. Um seine Verbindung mit Teilnehmer A zu verifizieren, initiiert B ein modifiziertes „Challange and Response" – Verfahren. Hierbei übersendet er einen mittels K_{AB} verschlüsselten Einmalwert an t_B. Als Ergebnis wird der mittels K_{AB} verschlüsselte Wert t_B-1 erwartet.



2.4.2. Nachteil

Teilnehmer B hat in dieser Version des Protokolls keine Möglichkeit das von der TTP erhaltene Kryptogramm auf „Frische" zu überprüfen, da B keinen direkten Kontakt zur TTP aufnimmt und somit keinen Zeitstempel übermittelt. Dies wird zum Problem, wenn der Schlüssel K_{AB} durch einen Angreifer C berechnet oder anderweitig erlangt wird. So kann der Angreifer ein Kryptogramm $K_{BTTP}(A, K_{AB})$ abfangen und später noch einmal an B senden. Teilnehmer B kann die „Frische" der Nachricht nicht überprüfen. Das anschließende modifizierte „Challenge and Response" – Verfahren wird von C abgefangen. Angreifer C antwortet mittels K_{AB} auf diese Challenge.
Noch effektiver lässt sich ein Angriff durch die Ermittlung von K_{ATTP} durchführen. Der Angreifer erhält „auf Vorrat" ihn interessierende Schlüssel K_{AX} von der Trusted Third Party.

3. Allgemeine Angriffe auf Protokolle

3.1. Impersonation

Ein Angreifer C gibt sich als Teilnehmer A aus. Dieser Angriff wird verhindert, indem durch eine dritte vertrauenswürdige Instanz (Trusted Third Party) jedem Kommunikationsteilnehmer eine geheime Information eindeutig zugeordnet wird. Hierbei ist einmal ein symmetrischer Schlüssel oder auch ein öffentlicher Schlüssel eines asymmetrischen Verfahrens möglich.

3.2. Replay – Attacken

Hierbei wird eine gültige Nachricht durch den Angreifer C abgefangen und zu einem späteren Zeitpunkt nochmals versendet.
Als einfaches Beispiel kann der Transfer eines Geldbetrages von einer Bank X zu einer Bank Y aufgeführt werden. Angreifer C zahlt DM 100,- auf der Bank X ein um diese auf sein Konto bei der Bank Y überweisen zu lassen. Würde die kryptographische Nachricht aus den Informationen Bank X, Bank Y, Kontonummer sowie Betrag (X, Y, K.-Nr., 100,- DM) bestehen, dann könnte der Angreifer diese Nachricht abfangen und mehrfach an seine Bank Y versenden. Damit würde die einmal getätigte Einzahlung der 100,- DM mehrfach ausgeführt werden.
Diesen Replay – Attacken wird mit Zeitstempeln oder Einmalwerten entgegengewirkt. Zur Realisierung der Zeitstempel ist eine synchrone Uhrzeit auf den Kommunikationsseiten notwendig.

3.3. Chess Grandmaster Problem (Man-in-The-Middle-Attack)

Der Angreifer C führt gleichzeitig zwei Protokollkommunikationen mit Teilnehmer A und Teilnehmer B durch. Die Antworten von A leitet er an B weiter, wobei dieser glaubt diese Antworten original von C zu erhalten.
Am Beispiel eines Schachspiels zwischen dem Angreifer (Anfänger) C und den beiden Großmeistern A sowie B wird diese Methode verdeutlicht. Das Spiel zwischen A und C fängt zuerst an, wobei A den ersten Zug führt. Nun beginnt das Spiel zischen C und B. Angreifer C beginnt hier mit dem Anfangszug von A aus dem Spiel A-C. Im Spiel C-B wird B darauf angemessen reagieren. Genau diesen Zug führt C im Spiel A-C. Angreifer C ist nun ein Übermittler der Reaktionen von A und B aufeinander, wobei A und B jeweils im Glauben sind, gegen den Anfänger C zu spielen

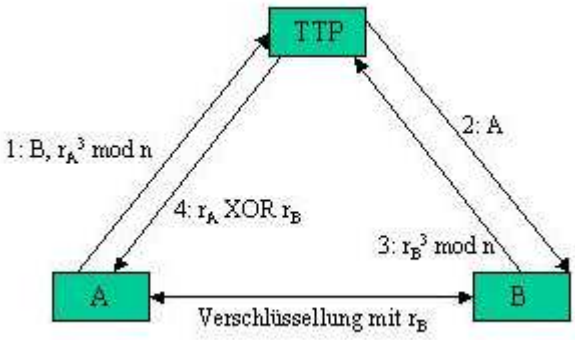
3.3.1. Möglichkeit – No-Key-Protokoll

Diese Methode kann für das No-Key-Protokoll eingesetzt werden. Bei diesem wird ein Geheimnis zwischen zwei Partnern kommuniziert, ohne das vorher ein gemeinsamer Schlüssel getauscht wurde. Dazu verschließt Teilnehmer A das Geheimnis mit seinem Schlüssel und versendet es an Teilnehmer B. Dieser verschließt das Geheimnis mit seinem Schlüssel und schickt es an A zurück. Dieser entfernt sein Schloss um das Geheimnis an Teilnehmer B zu senden, welcher nun auch sein Schloss entfernt womit er das Geheimnis sieht. Der Angreifer C könnte sich in die Mitte dieses Weges stellen und das Paket von Teilnehmer A an Teilnehmer B abfangen, sein eigenes Schloss anbringen umd das Paket zurück zu A zu senden. Dieser würde nun sein Schloss öffenen um das Paket an B zu senden. Auf diesem Weg wird es wieder vom Angreifer C abgefangen, welcher sein Schloss öffnet und das Geheimnis einsieht. Dieser Angriff ist möglich, da A bei dem zurückgeschickten Paket nicht erkennen kann, ob das Schloss auch zum Teilnehmer B gehört.

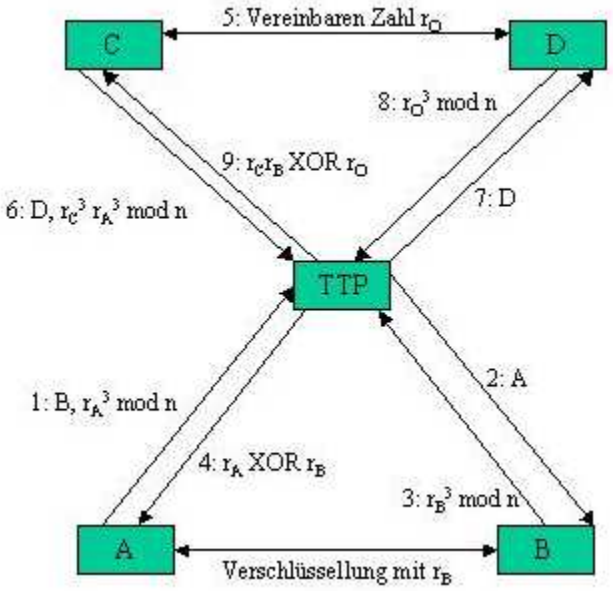
4. Beispielangriff: Simmons-Attacke auf das TMN-Protokoll

Das TMN-Protokoll ist ein Schlüsselmanagement-Protokoll in Mobilfunksystemen, welches 2 sichere Verfahren (RSA-Verfahren, One-Time-Pad) nutzt. Diese Verfahren

sind alleine betrachtet sicher aber in der hier auftretenden Kombination ist das Gesamtprodukt unsicher.
Vor dem Aufbau einer sicheren Kommunikation werden keine Schlüssel ausgetauscht. Dies findet im Moment der Notwendigkeit statt.
Möchte Teilnehmer A mit Teilnehmer B kommunizieren, so ermittelt A eine Zufallszahl r_A und sendet, unter Angabe des gewünschten Kommunikationspartners B, diese mit dem öffentlichen Schlüssel (RSA –Verfahren) der Trusted Third Party (Mobilfunksystem) an selbige. Diese informiert Teilnehmer B über den Kontaktwunsch, woraufhin B eine Zufallszahl r_B errechnet und diese mit dem öffentlichen Schlüssel der TTP an selbige sendet. Nun verschlüsselt die TTP r_B unter Verwendung des Schlüssels r_A mit dem One-Time-Pad – Verfahren und sendet das Ergebnis an Teilnehmer A welcher diese Verschlüsselung rückgängig macht. Nun besitzen A und B den gemeinsamen Schlüssel r_B.



Die beiden Angreifer C und D müssen den Verkehr zwischen A, der TTP und B abhören um den Wert aus Schritt Nr.3 (r_B^3 mod n) zu erlangen. Hier fügt C einen eigenen Wert r_C^3 an und es entsteht $r_C^3(r_B^3 \text{ mod } n) = (r_C^3 r_B^3 \text{ mod } n)$ welcher als neue Verbindungsanfrage mit D an die TTP gesendet wird. Diese teilt im nächsten Schritt D diesen Wunsch von C mit. Die Angreifer C und D haben sich vorher einen gemeinsamen Wert r_O ausgedacht, welchen D nun an RSA-verschlüsselt an die TTP sendet (r_O^3 mod n). Diese kodiert mit dem One-Time-Pad – Verfahren den Schlüssel r_O mit dem Faktor $r_C r_B$. Aus der Kenntnis von r_C und r_O kann C nun r_B ermitteln ($r_C r_B * r_C^{-1} \text{ mod } n = r_B$) um die für sicher gehaltene Kommunikation zwischen A und B zu belauschen.


C
D
5: Vereinbaren Zahl r_O
8: r_O^3 mod n
9: $r_C r_B$ XOR r_O
7: D
6: D, r_C^3 r_A^3 mod n
TTP
2: A
1: B, r_A^3 mod n
4: r_A XOR r_B
3: r_B^3 mod n
A
B
Verschlüsselung mit r_B