kpsp: Zwischenbericht symmetrische und asymmetrische Kryptographie

Marianne Schoch, Pascal Schwarz

21. April 2013

1 Abstract

Dieser Teil wird im finalen Bericht enthalten sein.

2 Einleitung / Idee

Die Software erlaubt die Erzeugung von verschlüsselten und signierten Nachrichten sowie deren Entschlüsselung. Die Nachrichten werden in Dateien gespeichert resp. aus Dateien gelesen.

Die Nachricht selbst wird mit einem symmetrischen Verfahren unter Verwendung eines zufälligen Schlüssels verschlüsselt. Dieser Schlüssel wird dem Empfänger der Nachricht ebenfalls übermittelt. Dazu wird der public Key des asymmetrischen Verschlüsselungsverfahrens verwendet.

Um die Integrität der Nachricht sicherstellen zu können wird des Weiteren eine Signatur, wiederum mit Hilfe des asymmetrischen Verschlüsselungsverfahrens und einer zusätzlichen Hashfunktion, der Nachricht beigefügt.

Zusätzlich ermöglichen wir die Erzeugung von Schlüsselpaaren für das asymmetrische Verschlüsselungsverfahren.

3 Theorie

3.1 Nachrichtenformat

3.1.1 Überblick

Die zu übermittelnde Datei besteht aus drei Teilen, die in den nachfolgenden Abschnitten beschrieben werden. Die Teile werden dabei in der folgenden Art markiert:

```
1 -----BEGIN <Abschnittname> [Option]-----
2 <Inhalt des Abschnittes, Base64 kodiert>
3 -----END <Abschnittname>-----
```

3.1.2 Teil KEYCRYPTED

Enthält den zufälligen „Sitzungsschlüssel“, der für das symmetrische Verschlüsselungsverfahren verwendet wird. Als Option wird das verwendete asymmetrische Verschlüsselungsverfahren angegeben. Da bei RSA die Länge des Plaintexts bekannt sein muss, wird diese (in Bytes) ebenfalls angegeben. Die Abschnittsmarkierung kann dann beispielsweise folgendermassen aussehen:

```
1 -----BEGIN KEYCRYPTED RSA 8----
2 <Base64 kodiertes Resultat der RSA-Verschluesselung des
      Sitzungsschluessels>
3 -----END KEYCRYPTED-----
```

Die Anwendung von RSA auf den Sitzungsschlüssel wird in Gruppen von 4 Bytes vorgenommen. Dabei werden die Bytes konkatieniert und als Zahl intepretiert.

Für Keys, deren Länge nicht ohne Rest durch vier teilbar sind (dies ist beispielsweise bei DES der Fall) wird der Input um Nullen ergänzt. Damit diese Nullen beim Entschlüsseln ignoriert werden können, wird die Längenangabe benötigt.

3.1.3 Teil MSGCRYPTED

Für die Verschlüsselung der eigentlichen Nachricht kommt ein symmetrisches Verfahren zum Einsatz. Falls es sich dabei um eine Blockchiffre handelt, wird neben dem Namen des Algorithmus ebenfalls angegeben, in welchem Modus die Blöcke verkettet werden. Falls ein Initialisierungsvektor benötigt wird, wird dieser zufällig erzeugt und in diesem Teil der Nachricht, mit einem „,“ vom Ciphertext separiert, abgelegt.

Kommt AES mit CBC als Modus zum Einsatz, sieht der Abschnitt folgendermassen aus:

```
1 -----BEGIN MSGCRYPTED AES256 CBC-----
2 <Base64 kodierter IV>,<Base64 kodiertes Resultat der Verschluesselung>
3 -----END MSGCRYPTED-----
```

3.1.4 Teil SIGNATURE

Die Signatur wird erzeugt, indem die verschlüsselten Inhalte der Teile KEYCRYPTED und MSGCRYPTED konkateniert werden. Nach der Anwendung eines Hash-Verfahrens wird beispielsweise RSA für die Erstellung der Signatur verwendet. Die Optionen für diesen Teil der Datei enthalten das verwendete Hashverfahren (woraus die Länge des Hashes abgeleitet werden kann) als auch das für die Signatur verwendete Kryptosystem. Ein Beispiel mit SHA256 und RSA sähe demnach so aus:

```
1 -----BEGIN SIGNATURE SHA256 RSA-----
2 <Base64 kodiertes Resultat der RSA-Signierung von
      SHA256(KEYCRYPTED|MSGCRYPTED)>
3 -----END SIGNATURE-----
```

3.2 Schlüsselformat

3.2.1 RSA

RSA Schlüssel bestehen aus Exponent (e oder d) und dem Produkt der beiden Primzahlen (n). Diese werden in einer Datei mit dem folgenden Format gespeichert:

```
1 -----BEGIN RSA PUBLIC KEY-----
2 <Base64 kodierte Binaerdarstellung von e>,<Base64 kodierte
      Binaerdarstellung von n>
3 -----END RSA PUBLIC KEY-----
```

3.3 Verwendung

3.3.1 Dateien

plain Datei mit zu verschlüsselndem Inhalt oder Resultat der Entschlüsselung

crypt Datei mit Resultat der Verschlüsselung

rsapriv Datei mit eigenem privatem RSA-Schlüssel

rsapub Datei mit eigenem öffentlichem RSA-Schlüssel

rsapubrecv Datei mit öffentlichem RSA-Schlüssel des Empfängers

3.3.2 Schlüsselerzeugung

Erzeugung eines Schlüsselpaares:

```
1 ./hskeygenerator
```

3.3.3 Ver- und Entschlüsselung

Verschlüsselung:

```
1 ./hsencrypt <asymm. kryptosystem> <hashverfahren> <symm. kryptosystem>
      <modus> <empfaenger public key> <verschluesselte datei>
```

Entschlüsselung:

```
1 ./hsdecrypt <sender publickey> <verschluesselte datei>
```

4 Implementation

Dieser Teil wird im finalen Bericht enthalten sein.

5 Testfälle

Dieser Teil wird im finalen Bericht enthalten sein.