GSS-Übungsblatt 3

Chamier, Eickhoff, Gäde, Hölzen, Jarsembinski · SoSe 2016

1. Zentrale Begriffe der Kryptographie

1.2. Schlüsselaustausch

Bei assymetrischer Verschlüsselung müssen nur für jeden Kommunikationspartner ein öffentlicher und ein privater Schlüssel erstellt werden. Bei n Personen also 2n Schlüssel. Für eine symmetrische Verschlüsselung benötigt jedes Kommunkationspaar einen eigenen Schlüssel. Bei n Personen sind dies $\frac{n\cdot(n-1)}{2}$ Schlüssel.

1.3. Hybride Kryptosysteme

- Wenn Alice eine sehr lange Nachricht senden möchte, dann ist die Verschlüsselung mit dem relativ langen öffentlichen Schlüssel (4096-Bit) sinnvoll. Ist die Nachricht jedoch sehr lang, oder will Alice sehr viele Nachrichten verschicken, ist ein hybrides Kryptosystem effizienter, da sonst sehr häufig mit dem langen öffentlichen Schlüssel verschlüsselt werden müsste.
- Um ein hybrides Kryptosystem zu benutzen, schickt Alice Bob einen mit Bobs öffentlichen Schlüssel verschlüsselten symmetrischen Schlüssel. Mit diesem symmetrischen Schlüsel werden dann alle nachfolgenden Nachrichten verschlüsselt. Dies ist deutlich effizienter, da der symmetrische Schlüssel deutlicher kleiner (256-Bit) ist.
- ?

2. Parkhaus

2.2. Sicherheitsanalyse

Das System ist angreifbar, da anscheinend der Barcode für den Pauschalpreis vom Kino immer identisch ist, und man diesen einfach auf die Karte drucken könnte. Auch der Barcode zur Bezahlbestätigung ist identisch , und je nachdem wie überprüft wird, ob die Karte in den letzten 10 Minuten bezahlt wurde, kann man auch diesen Barcode auf die Karte drucken. Außerdem können mit der selben Karte in einem 10 Minuten Fenster beliebig viele Autos das Parkhaus verlassen.

Angreifermodell:
Rolle:Außenstehender,Benutzer
Verbreitung:Tickets kaufen oder benutzte Tickets auflesen
Verhalten:aktiv, beobachtend (Karte kaufen oder auflesen und analysieren)
Rechenkapazität:unbeschränkt

2.3. Umsetzung mit kryptographischen Techniken

Um die Parktickets gegen Betrug zu schützen, führen wir eine asymmetrische Verschlüsselung der Barcodes ein. Dafür besitzt der Ticketautomat geheime Schlüssel für alle Händler mit Sonderangeboten. Die Einzelhändler nehmen den einzigartigen Barcode des Tickets (erster oder zweiter Block von rechts) und verschlüsseln diesen mit ihrem öffentlichen Schlüssel. Anschlließend kann der Ticketautomat mit dem zugehörigen privaten Schlüssel, den Code auf Echtheit prüfen. Diese Verschlüsselung basiert auf RSA.

3. Authentifizierungsprotokolle

3.2. Authentifikationssytem auf Basis indeterministischer symmetrischer Verschlüsselung

Die indeterministische symmetrische Verschlüsselung schützt vor Angriffen auf den Klartext von Nutzername und Passwort. Jedoch schützt das Verfahren nicht vor einem Maskeraden-Angriff. Ein Man-In-The-Middle kann das Verschlüsselte Benutzer-Passwort-Paar abfangen und dieses benutzen, um sich beim Server zu authentifizieren. Da die Zufallszahl nicht vorher an den Server übertragen wird, kann der Server diese auch nicht von jeder anderer Zahl unterscheiden.

3.3. Challenge-Response-Authentifizierug

Bei der Challenge-Response-Authentifizierung wird beim Verbindungsaufbau zwischen User und Server gegenseitig die Echtheit durch das Stellen und die Berechnung von sogenannten Challenges und Responses gewährleistet. Die Verbindung wird initiiert mit einer Challenge des Users, auf die der Server mit der korrekten Response und einer neuen Challenge für den User antwortet. Der User berechnet die Response für die zweite Challenge und schickt diese mit einer dritten Challenge zurück zum Server. Dieser antwortet mit einer letzten Response und ist diese korrekt, wird die Verbindung aufgebaut.
Dieses System ist angreifbar, indem man folgendes tut:

- User schickt Challenge1 an Server1
- Server1 schickt Challenge2 und Response1 an User
- User schickt Challenge2 an Server2
- Server2 schickt Challenge3 und Response2 an User
- User schickt Response2 und Challenge3 an Server1
- Server1 schickt Response3 an User
- Verbindungsaufbau

GSS-Übungsblatt 3

Chamier, Eickhoff, Gäde, Hölzen, Jarsembinski · SoSe 2016

5 RSA-Verfahren

3.2. Anwendung

Anhang A RSA-Decipher

Listing 1: RSA_Decipher.rb

```
def decipher ciphertext, d
  m = ciphertext.map { |c| c = c**d % 102709; c.chr }
  return m.join ''
end

p decipher [14979, 20999, 9653, 14027, 36157, 43875, 41820, 9653,
    36157, 12509, 64444, 64444,
26369, 65262, 9262, 33022, 20999, 7919, 20999, 14027, 36157, 7919,
    41820, 70510,
43875, 36157, 98809, 15171, 9262, 60740, 9653, 70510, 43875, 9653,
    36157, 26006,
98126, 9653, 36163, 9653, 70510, 36157, 16871, 41820, 8848, 98126,
    15973, 41820,
60740, 17460, 36157, 10331, 70510, 60740, 14027, 9653, 41820, 98809,
    9653, 14027,
36163, 15171, 43875, 9653, 9262, 9262, 9653, 82965, 36157, 64444, 8848,
    98126, 20999,
15973, 100066, 100066, 41820, 9653, 9262, 9653, 82965, 36157, 82040,
    33022, 41820,
70510, 35349, 15171, 16871, 36157, 26006, 33022, 35349, 9262, 9653,
    7919, 82965,
36157, 43875, 41820, 9653, 36157, 27583, 29673, 70510, 26369, 98699,
    64444, 41820,
8848, 98126, 9653, 14027, 98126, 9653, 41820, 15973, 36157, 57187,
    15171, 70510,
36157, 80112, 33022, 7919, 7919, 16871, 15171, 9653, 14027, 15973,
    9653, 14027,
70510, 36157, 20999, 70510, 43875, 36157, 43875, 33022, 100066, 20999,
    60740, 9653,
98126, 15171, 9653, 14027, 41820, 60740, 9653, 36157, 10331, 70510,
    60740, 14027,
41820, 98809, 98809, 9653, 82965, 36157, 27341, 20999, 60740, 33022,
    70510, 60740,
7919, 26369, 36157, 20999, 70510, 43875, 36157, 27341, 20999, 60740,
    14027, 41820,
98809, 98809, 7919, 29043, 15171, 70510, 15973, 14027, 15171, 9262,
    9262, 9653,
82965, 36157, 26006, 41820, 36163, 41820, 70510, 60740, 26369, 10331,
    15973, 15973,
33022, 8848, 29043, 36157, 20999, 70510, 43875, 36157, 80112, 15171,
    16871, 9653,
```

```
14027, 26369, 10331, 70510, 33022, 9262, 44783, 7919, 41820, 7919,
    82965, 36157,
100308, 41820, 15171, 36163, 9653, 15973, 14027, 41820, 7919, 8848,
    98126, 9653,
36157, 51682, 9653, 14027, 98809, 33022, 98126, 14027, 9653, 70510,
    82965, 36157,
12509, 14027, 20999, 70510, 43875, 9262, 33022, 60740, 9653, 70510,
    36157, 43875,
9653, 14027, 36157, 65262, 14027, 44783, 73347, 15973, 15171, 60740,
    14027, 33022,
73347, 98126, 41820, 9653, 82965, 36157, 43875, 33022, 7919, 36157,
    82040, 64444,
10331, 26369, 51682, 9653, 14027, 98809, 33022, 98126, 14027, 9653,
    70510, 82965,
36157, 10331, 20999, 15973, 98126, 9653, 70510, 15973, 41820, 98809,
    41820, 29043,
33022, 15973, 41820, 15171, 70510, 7919, 73347, 14027, 15171, 15973,
    15171, 29043,
15171, 9262, 9262, 9653, 36157, 20999, 70510, 43875, 36157, 70510,
    33022, 15973,
20999, 9653, 14027, 9262, 41820, 8848, 98126, 36157, 33022, 9262, 9262,
    9653, 36157,
33022, 70510, 43875, 9653, 14027, 9653, 70510, 36157, 54813, 70510,
    98126, 33022,
9262, 15973, 9653, 82965, 36157, 43875, 41820, 9653, 36157, 16871,
    41820, 14027,
36157, 41820, 70510, 36157, 43875, 9653, 14027, 36157, 29673, 9653,
    35349, 20999,
70510, 60740, 36157, 20999, 70510, 43875, 36157, 43875, 9653, 14027,
    36157, 51682,
15171, 14027, 9262, 9653, 7919, 20999, 70510, 60740, 36157, 35349,
    9653, 98126,
33022, 70510, 43875, 9653, 9262, 15973, 36157, 98126, 33022, 35349,
    9653, 70510,
36157, 17460, 26369, 98699], 4343
```