RUHR
UNIVERSITÄT
BOCHUM

RUB

Lehrstuhl für Kryptologie und IT-Sicherheit
Prof. Dr. Alexander May
Mathias Herrmann, Enrico Thomae

Hausübungen zur Vorlesung

Kryptanalyse

WS 2010/2011

Blatt 2 / 20. Oktober 2010 / Abgabe bis spätestens 27. Oktober 2010, 10 Uhr (vor der Übung)

AUFGABE 1 (5 Punkte)**:**
Alice schickt eine Einladung m zu ihrer Geburtstagsparty an Bob und Berta. Dabei verschlüsselt Alice m mit den öffentlichen RSA-Schlüsseln (N, e_1) und (N, e_2) von Bob und Berta, wobei e_1 und e_2 teilerfremd sind.
Eve ist nicht zur Party eingeladen. Zeigen Sie, dass Eve trotzdem aus den Chiffretexten die Einladung m effizient berechnen kann.

AUFGABE 2 (5 Punkte)**:**
Alice hat aus ihren Fehlern gelernt und verschickt keine Einladungen mehr an Empfänger mit gleichen RSA-Moduln. Zu ihrer nächsten Feier will sie Bob, Berta und Birte einladen. Diese besitzen die paarweise teilerfremden RSA-Moduln N_1, N_2 und N_3 und verwenden alle den öffentlichen Exponenten $e = 3$. Die von Alice verschickte Einladung soll ein gültiger Klartext für alle Moduln sein, d.h. $m < \min\{N_1, N_2, N_3\}$.
Wiederum wurde die arme Eve nicht zur Party eingeladen. Zeigen Sie, dass Eve auch in diesem Fall m effizient berechnen kann.

AUFGABE 3 (5 Punkte)**:**
Sei $N = pq$ ein RSA-Modul mit $p < q$. Zeigen Sie durch eine Meet-in-the-Middle Attacke auf den Parameter p, dass man die Faktorisierung von N in Zeit und Platz $\tilde{\mathcal{O}}(N^{\frac{1}{4}})$ berechnen kann.
Hinweis: Verwenden Sie analog zur Vorgehensweise in Abschnitt 3.1.2 des Skripts eine Polynomdarstellung.

AUFGABE 4 (5 Punkte)**:**
Meet-in-the-Middle Angriff auf d mit bekannter Approximation
Implementieren Sie eine Variante des im Skript beschriebenen MITM Angriffs auf den geheimen RSA Schlüssel d. In unserem Szenario ist bereits eine Approximation $\tilde{d}$ von d gegeben. Die Approximation stimmt bitweise mit d überein, außer in den Bits mit Indices 224 bis 255. Dort sind die Bits in der Approximation alle gleich 0. (Das LSB hat Index 0.)
Der öffentliche Schlüssel, sowie die Approximation von d findet sich auf der Webseite zum Download.

Tips für Sage:

- Sollte ein Programmfragment nicht tun was Sie erwarten, dann lohnt es sich die Typen der einzelnen Objekte im Auge zu behalten (`type(a)`).
- Zu fast allen Befehlen und Funktionen erhalten Sie eine Beschreibung und Hilfe durch das Anhängen eines Fragezeichen, also z.B. `type?`. Vorsicht: Keine Klammern!
- Um in der Restklassengruppe $\mathbb{Z}/n\mathbb{Z}$ zu rechnen, gibt es in Sage die Funktion `Mod(m,n)`.
- Für die sortierte Liste mit zwei Komponenten können Sie ein dictionary verwenden. Die Benutzung funktioniert in etwa so:

```
#Erzeugt neues Dictionary
d=dict()

#Legt im Dictionary unter dem Schlüsselwert key
#den Wert value ab
d[key] = value

#Prüft ob der Schlüssel key im Dictionary vorhanden
d.has_key(key)

#Gibt den Wert zum Schlüssel key aus
d[key]
```