Vorlesung Sicherheit

Dennis Hofheinz

ITI, KIT

15.05.2014

Überblick

Überblick

Erinnerung

- Asymmetrische (Public-Key-)Verschlüsselung:

$$\texttt{Alice}_{sk} \xleftarrow{C:=\mathrm{Enc}(pk,M)} \texttt{Bob}_{pk}$$

- RSA:

$$\mathrm{Enc}(pk, M) = M^e \bmod N \qquad \mathrm{Dec}(sk, C) = C^d \bmod N$$

- Homomorphie von RSA

$$\mathrm{Enc}(pk, M) \cdot \mathrm{Enc}(pk, M') = \mathrm{Enc}(pk, M \cdot M')$$

- Homomorphie kann problematisch sein $\rightarrow$ Auktionen

Überblick

Lösungsvorschläge?

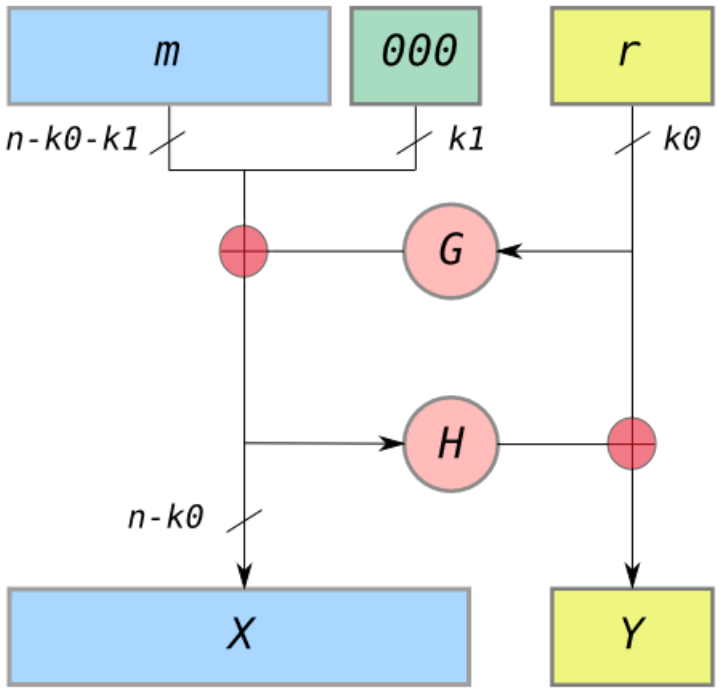
- **Diskussion:** Wie könnte man RSA „reparieren“?

RSA-Padding

- Erster Ansatz (PKCS#1 1.5, 1993): (Randomized) Padding
  - $\mathrm{Enc}(pk, M) = (\mathrm{pad}(M, R))^e \bmod N$ (mit Zufall R)
  - Dec erhält und überprüft $\mathrm{pad}(M, R)$, und extrahiert M
  - Beispiel: $\mathrm{pad}(M, R) = M||0^\ell||R$ (mit $M, R \ll N$)
- PKCS#1 1.5 problematisch
  - Kein Sicherheitsbeweis, 1998 sogar Problem gefunden
  - Homomorphe Eigenschaft der RSA-Funktion erlaubt subtile Änderungen an Nachricht
- PKCS#1 2.0 nutzt „RSA-OAEP“: besseres Padding

RSA-OAEP

- pad-Funktion von RSA-OAEP (G, H Hashfunktionen):



Quelle: Wikipedia

Sicherheit von RSA-OAEP

- Heuristisch so sicher wie „RSA-Funktion invertieren“
  - Sicher heißt hier: semantisch sicher selbst gegen aktive Angriffe
- Bester bekannter Angriff: N faktorisieren
  - $P, Q \rightsquigarrow \varphi(N) = (P-1)(Q-1) \rightsquigarrow d = e^{-1} \bmod \varphi(N)$
  - Bester bekannter Faktorisierungsalgorithmus: Zahlkörpersieb
  - Nach heutigem Stand 2048 als Bitlänge von N sicher
- Offene Forschungsfrage: N faktorisieren *notwendig*, um RSA(-OAEP) zu brechen?

Relevanz von RSA(-OAEP)

- Nachteil von RSA(-OAEP): rechenaufwändig
  - Naiver Algorithmus für Exponentiation modulo ℓ-Bit-Zahl benötigt $\mathbf{\Theta}(\ell^3)$ Bitoperationen, schlecht parallelisierbar
  - Es existieren (asymptotisch) bessere Algorithmen
  - **Aber:** Für realistische ℓ ist naiver Algorithmus am schnellsten
- Gründe, warum RSA(-OAEP) trotzdem benutzt wird:
  - Einfach zu implementieren
    - Einfache Arithmetik
    - Ver- und Entschlüsselung sehr ähnlich
  - Mit Optimierungen ($e = 3$ bei Verschlüsselung, CRS nutzen bei Entschlüsselung) teilweise konkurrenzfähig
- RSA als „Trapdoor One-Way Permutation“ interessant

Überblick

ElGamal (1985)

- Szenario: zyklische Gruppe $\mathbb{G} = \langle g \rangle$
- $pk = (\mathbb{G}, g, g^x)$, $sk = (\mathbb{G}, g, x)$ (mit x zufällig)
- $\mathrm{Enc}(pk, M) = (g^y, g^{xy} \cdot M)$ (mit y zufällig)
- $\mathrm{Dec}(sk, (Y, Z)) = Z/Y^x \qquad (= (g^{xy} \cdot M)/(g^y)^x = M)$
- Beobachtung: Verschlüsselung probabilistisch
- **Aber:** ElGamal wie RSA homomorph

$$\begin{aligned}\mathrm{Enc}(pk, M) \cdot \mathrm{Enc}(pk, M') &= (g^y, g^{xy} \cdot M) \cdot (g^{y'}, g^{xy'} \cdot M') \\ &= (g^{y+y'}, g^{x(y+y')} \cdot M \cdot M') = \mathrm{Enc}(pk, M \cdot M')\end{aligned}$$

Mehr über ElGamal

- ElGamal unter naheliegender Annahme semantisch sicher
  (allerdings *nicht* gegen aktive Angriffe)
- Nicht-homomorphe Varianten von ElGamal existieren
- Kandidaten für geeignete Gruppen $\mathbb{G}$:
  - (Echte) Untergruppen von $\mathbb{Z}_P^*$ (mit P prim)
  - Allgemeiner: Untergruppen von $\mathbb{F}_q^*$ (mit q Primpotenz)
  - Effizienter: (Untergruppen von) elliptischer Kurve $\mathbf{E}(\mathbb{F}_q)$
- Realistische Gruppengröße:
  - $|\mathbb{G}| \approx 2^{2048}$ (für $\mathbb{G} \subset \mathbb{Z}_P^*, \mathbb{F}_q^*$)
  - $|\mathbb{G}| \approx 2^{200}$ (für $\mathbb{G} \subseteq \mathbf{E}(\mathbb{F}_q)$)

Überblick

Demonstration

- **Demonstration:** Geschwindigkeitsvergleich RSA/elliptische Kurven

Überblick

Zusammenfassung asymmetrische Verschlüsselung

- Public-Key-Verschlüsselung löst Schlüsselverteilungsproblem
- RSA wichtig, aber ohne Padding problematisch $\rightsquigarrow$ RSA-OAEP
- ElGamal in kleineren Gruppen möglich (Effizienz)
- Beide Verfahren (ungepadded) homomorph (Vorteil/Nachteil)

Aktuelle Forschung

- Mehr/andere Funktionalität, zum Beispiel:
  - Identitätsbasierte Verschlüsselung (löst Zertifizierungsproblem)
  - Vollhomomorphe Verschlüsselung (Berechnungen delegieren[1])
  - Funktionale Verschlüsselung (viele sk_f, $\mathrm{Dec}(sk_f, C) = f(M)$)
  - Broadcast-Verschlüsselung (Beispielanwendung: Pay-TV)
- Andere Probleme, alternative mathematische Strukturen
  - Public-Key-Verschlüsselung so sicher wie Faktorisierung
  - Gitterbasierte Verschlüsselung

[1] momentan noch um Größenordnungen zu ineffizient