# Paillier

Erfunden von und benannt nach Pascal Paillier im Jahr 1999

Ist ein additives homomorphes Verschlüsselungsverfahren

Anwendungen

* E-Voting

Nachdem alle Wahlberechtigten ihre Stimmen verschlüsselt übermittelt haben, wird das Ergebnis aus diesen berechnet und dann entschlüsselt. Für die Errechnung des Ergebnisses wird der Private-Key nicht benötigt.

* Zero-Knowledge Beweis

Ist eine Methode, mit der eine Partei einer anderen Partei beweisen kann, dass sie über Wissen verfügen ohne Informationen über dieses Wissen weiter zu geben.

Beispiel: 2 Bälle mit unterschiedlichen Farben.

## Paillier KeyGen

Man wählt zwei ausreichend große Primzahlen p und q. Ausreichend groß entspricht 1024 Bit, also Zahlen mit je 309 Ziffern.

Danach berechnet man n. n ist das Produkt von p und q. Danach berechnet man l als das kleinste gemeinsame Vielfache von (p-1) und (q-1). Vereinfacht kann man auch l = (p-1)(q-1) setzen.

Man wählt dann noch ein zufälliges g aus dem Restklassenring modulo n², oder vereinfacht setzt man g auf n+1.

Außerdem benötigt man noch das multiplikative Inverse m zu l modulo n.

Daraus ergibt sich der Public Key (n,g) und der Private Key (l,m)

## Paillier Verschlüsselung

Zur Berechnung des Ciphertextes mit dem Public Key (n,g) braucht man noch eine Zufallszahl r, für die gilt: ggT(r,n) = 1 und 0 < r < n.

Dann ergibt sich der Ciphertext c einer Nachricht m mit der Formel:

## Paillier Entschlüsselung

Man berechnet die Nachricht m aus dem Ciphertext c mit Hilfe des Private Keys (l,m) (und des Public Keys (n,g) natürlich) durch die Formel

## Paillier Teilhomorphie

Die Verschlüsselung ist additiv homomorph

Eine Multiplikation von zwei verschlüsselten Werten entspricht der Addition der unverschlüsselten Werte.

Da der ist, ist ein gültiger Ciphertext von .

# Übersicht über Teilhomomorphe Algorithmen

|  |  |
| --- | --- |
| **Algorithmus** | **Homomorphie Eigenschaft** |
| RSA | multiplikativ homomorph |
| Padded RSA (OAEP) | nicht homomorph |
| Goldwasser Micali | additiv homomorph |
| Paillier | additiv homomorph |

# Vollhomomorphe Verschlüsselungen

Eine Funktion behält die Ringstruktur von bei. Also ist immer noch ein Ring, wenn vollhomomorph ist.

Das heißt bei einer Vollhomomorphen Verschlüsselung können Additionen und Multiplikationen durchgeführt werden.

Gentry‘s Algorithmus (2009) war der erste vollhomomorphe Verschlüsselungsalgorithmus, mit Verwendung von Zahlengittern. Seine Laufzeitkomplexität ist

## Aktuelle Laufzeiten

|  |  |
| --- | --- |
| Energieverbrauch Profilklassifikation | < 1 Sekunde |
| Verschiedene medizinische Diagnosen | < 2 Minuten |
| Gen-basierte Diagnosen | < 10 Minuten |
| Lauflängenkodierung (bei 48 Kernen)  Zur Bild/Videokompression. | Ca 30 Minuten |

Komplexität basiert auf dem Sicherheitslevel der Daten, der Komplexität der Daten und der Optimierung.

## Probleme

Die Wahl geeigneter Parameter ist schwer. Außerdem sind bisherige Implementierungen nicht alltagstauglich, zum einen wegen der Geschwindigkeit der Algorithmen. Zum anderen verrauschen die Daten. Bei Gentry zum Beispiel muss nach spätestens 30 Operationen eine Bereinigung durchgeführt werden, damit die Daten korrekt und lesbar bleiben.

Auch in der Sicherheit gibt es einen Angriffspunkt. Die Verschlüsselungen sind mit Chosen-Ciphertext-Attacken angreifbar.

# Hybrid-Homomorph

Bei einem hybrid-homomorphen Verschlüsselungsalgorithmus, werden zwei verschiedene teilhomomorphe Verschlüsselungsalgorithmen verwendet. Zum einen eine additive homomorphe Verschlüsselung und eine multiplikativ homomorphe Verschlüsselung. In dem von uns gewählten Verfahren wurde Goldwasser-Micali als additive Variante gewählt und RSA als multiplikative Variante.

## Ablauf

1. Erstellen der Keys für G. Micali
2. Verschlüsseln mittels G. Micali
3. Erstellen der Keys für RSA
4. Verschlüsseln mittels RSA
5. Durchführen der Operationen
6. Entschlüsseln mittels RSA
7. Entschlüsseln mittels G. Micali

Aufpassen muss man hier nur, dass wenn zuerst mit Goldwasser-Micali verschlüsselt wird und dann mit RSA. Dann muss bei der Entschlüsselung zuerst mit RSA entschlüsselt und dann mit Goldwasser-Micali entschlüsselt werden.

# Vor- und Nachteile

Nachteile:

* Sehr hohe Laufzeiten
* Große Rechnerkapazitäten werden benötigt
* Bei der Anwendung, ist die gültige Rechtsprechung, abhängig vom Standort der Server

Vorteile:

* Gewinn an Datenschutz
* Mobilität
* Outsourcing