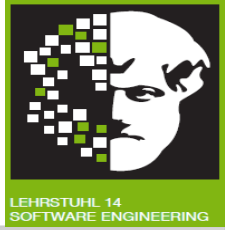


Homomorphe Kryptografie

Niklas A. Zbick

Homomorphe Kryptografie

- „Heiliger Gral“ der Kryptografie
- aktuell bedeutendes Forschungsgebiet
- ermöglicht u.a. sicheres Cloud-Computing
- noch sehr frisch, daher noch nicht im Einsatz

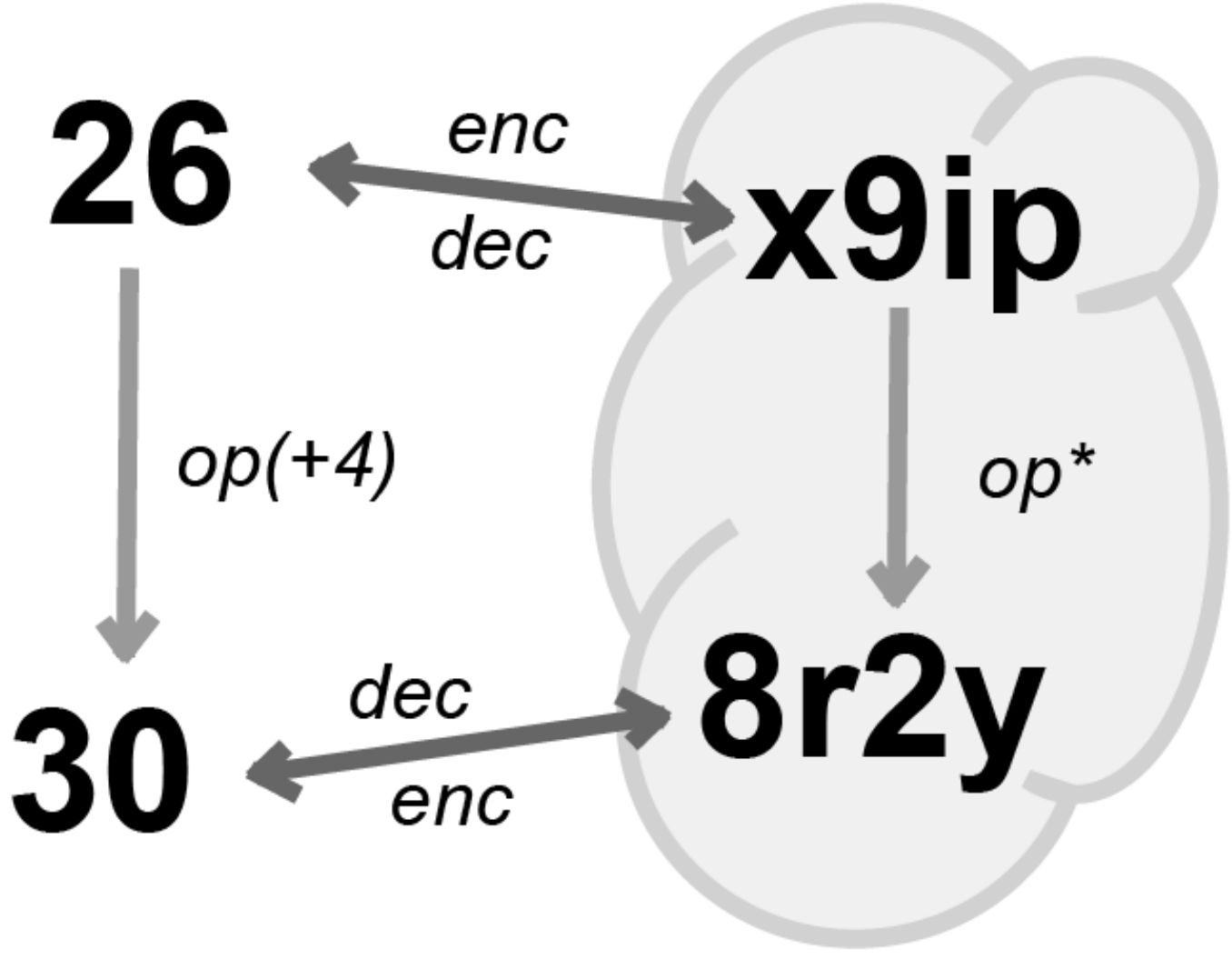
Gliederung

- Was ist Homomorphe Kryptografie?
- Gentrys Verfahren
- Wofür kann sie eingesetzt werden?
- Entstehungsgeschichte
- Aktueller Stand
- Fazit

Was ist Homomorphe Kryptografie? (1)

- Homomorphismus:
  - Strukturerhaltende Abbildung
  - $f(a+b) = f(a) + f(b)$ oder $f(a+b) = f(a) * (b)$
- Homomorphe Verschlüsselung:
  - $op(m) = dec(op^*(c))$
  - mit $enc(m) = c$ und $dec(c) = m$

Was ist Homomorphe Kryptografie? (2)



Homomorphe Kryptografie: Beispiel

- Reintext: $m = 7$
- Verschlüsseln: $enc(x) = x + 3$
- Entschlüsseln: $dec(x) = x - 3$
- Operation: $op(x) = x - 2$

$$\Rightarrow \quad op(m) = 7 - 2 = 5$$

$$= (10 - 2) - 3 = dec(\, op^*(c)\,)$$

Homomorphe Kryptografie: Weiteres Beispiel

- 2 Reintexte: m1 = 3, m2 = 4
- Verschlüsseln: *enc(x) = x * 2*
- Entschlüsseln: *dec(x) = x / 2*
- Operation: *op(x,y) = x + y*

=> *op(m1,m2) = 3 + 4 = 7*

= (6 + 8) / 2 = dec(op(c1,c2))*

- „additiv“ homomorph

Homomorphe Kryptografie: Und noch ein Beispiel

- RSA-Algorithmus:

$$\psi = \pi^e \bmod N$$

$$\Rightarrow \quad \prod_i \psi_i = (\prod_i \pi_i)^e \bmod N$$

- „multiplikativ“ homomorph

(nahezu) voll homomorphe Kryptografie

- Voll homomorphe Kryptografie
  - funktioniert für alle Operationen
  - funktioniert beliebig oft

- nahezu vollständig homomorph
  - (engl.: somewhat homomorphic)
  - beliebige Art von Operationen
  - aber: nicht beliebig oft

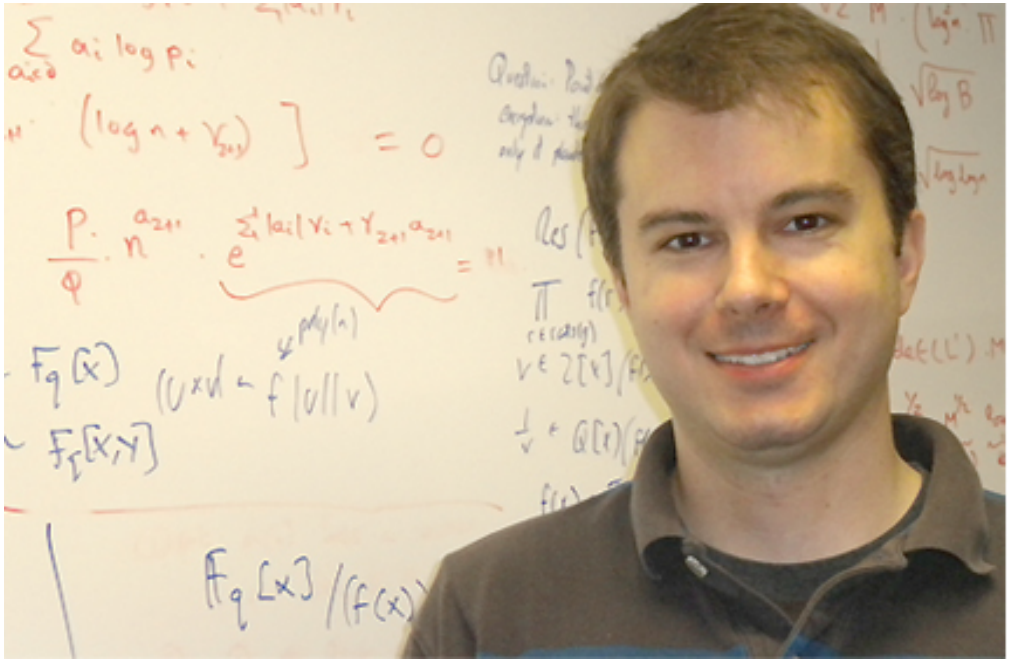
DAS VERFAHREN VON GENTRY

Craig Gentry

a.espncdn.com/combiner/i?img=/i/
headshots/mlb/players/full/30267.png

http://en.wikipedia.org/wiki/Craig_Gentry

Craig Gentry



http://www.forbes.com/forbes/2009/0713/breakthroughs-privacy-super-secret-encryption.html

http://www.forbes.com/sites/andygreenberg/2011/04/06/darpa-will-spend-20-million-to-search-for-cryptos-holy-grail/

Idee des Verfahrens

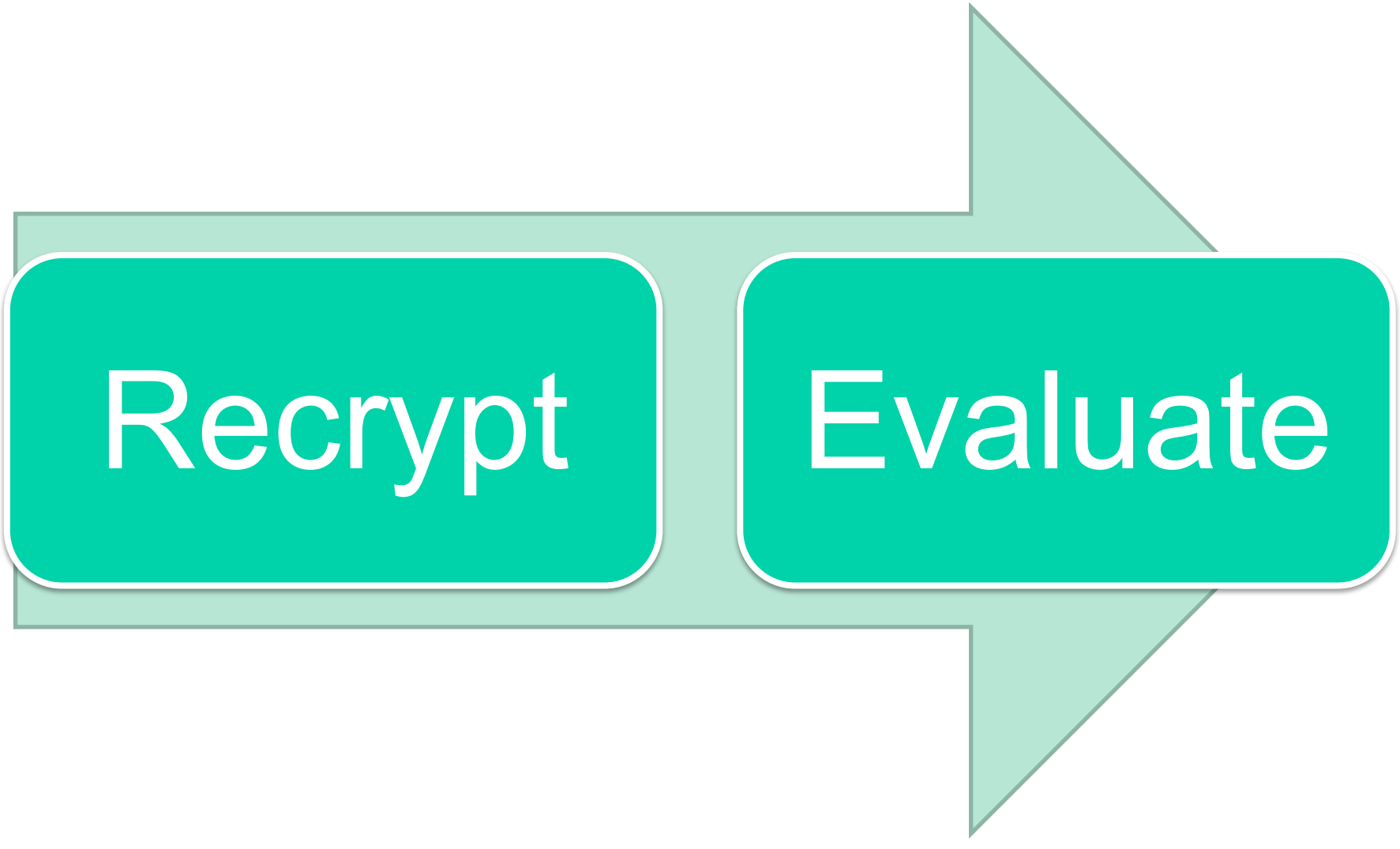
- Benutze nahezu voll homomorphes Verfahren
- Optimiere die Geheimtexte vor jeder Operation
  - Annahme: Fehlertoleranzschranke *N*
  - *m1*, *m2* haben Fehler $n > \sqrt{N}$
  - Operation ist ein Multiplkation *m1*m2*

    => neuer Fehler > *N*
- Bootstrapping

Die Funktionen

- *KeyGen*: Generiert das Schlüsselpaar sk und pk
- *Encrypt*: Verschlüsselt den Reintext mit pk
- *Decrypt*: Entschlüsselt den Geheimtext mit sk
- *Evaluate*: Wendet eine Operation an. Bekommt diese und die Operanden als Parameter übergeben
- *Recrypt*: Erzeugt neuen Geheimtext mit geringem Fehler

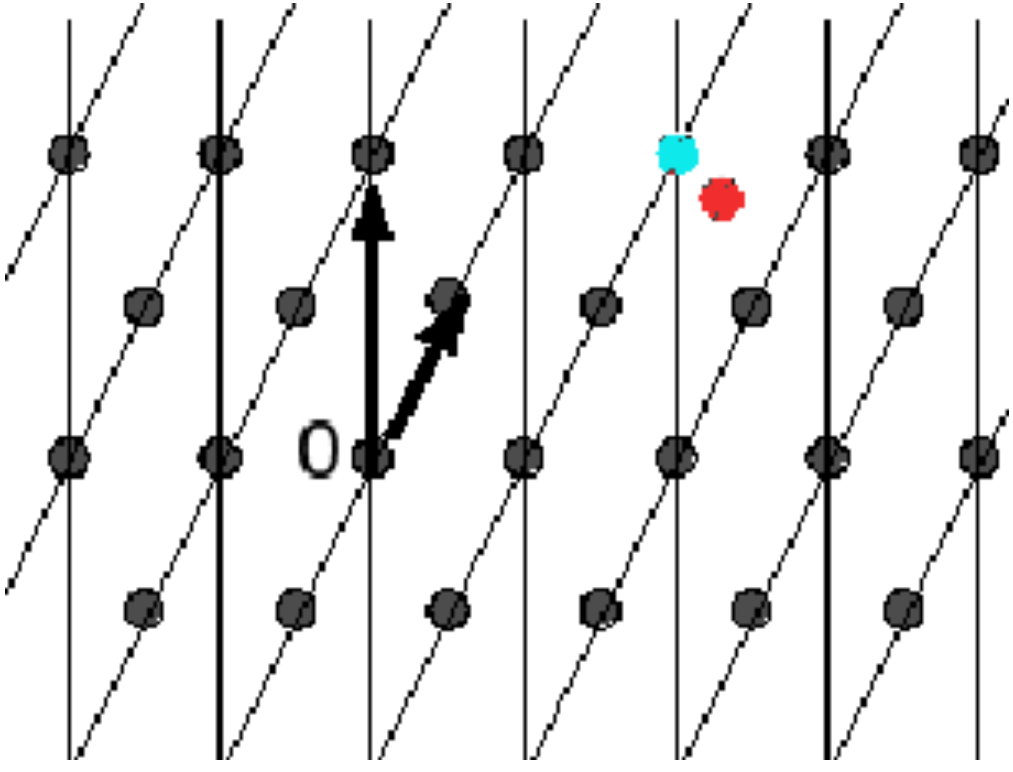
Bootstrapping



Mathematischer Hintergrund

- Benutzt Idealgitter
  (engl.: ideal lattices)

- Reduzierbar auf bekannte Gitterprobleme
  - Closest Vector Problem (CVP)
  - Bounded Distance Decoding Problem (BDDP)
  - alle sind NP-schwer



http://en.wikipedia.org/wiki/Lattice_problem

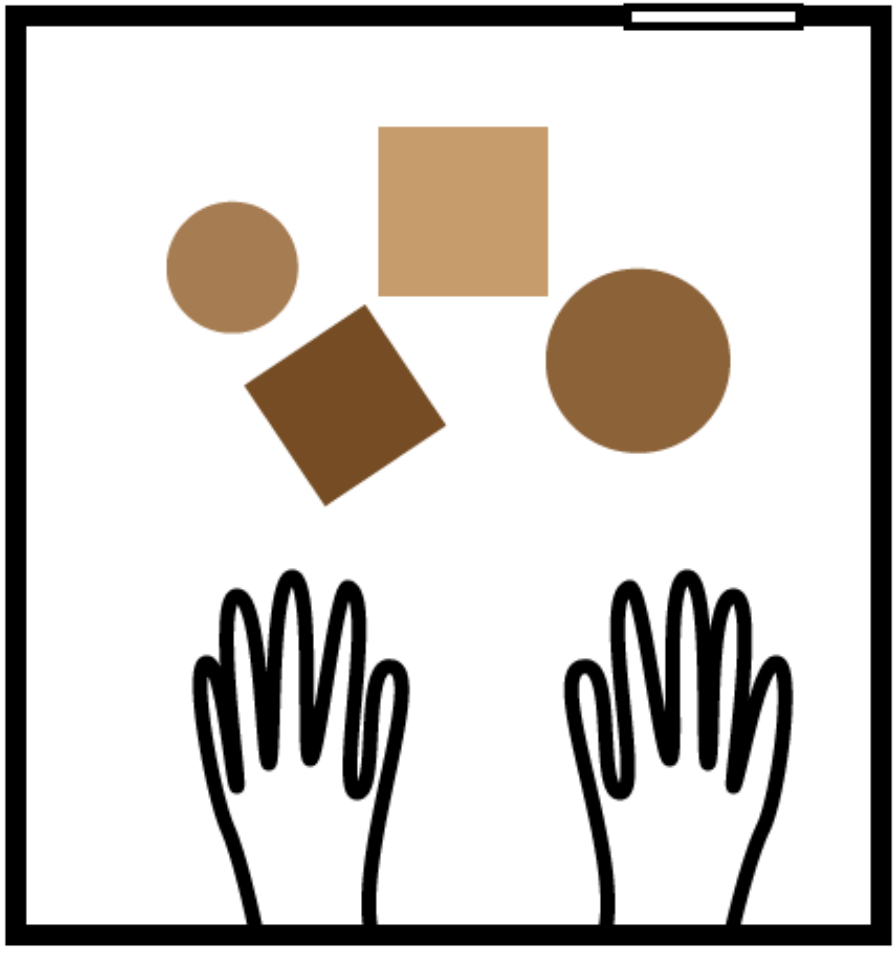
Analogie

- Analogie von Gentry erdacht

- Szenario:
  - Juwelierladen
  - Besitzerin: Alice
  - Misstrauen gegenüber Mitarbeitern
  - Arbeit mit wertvollen Materialen

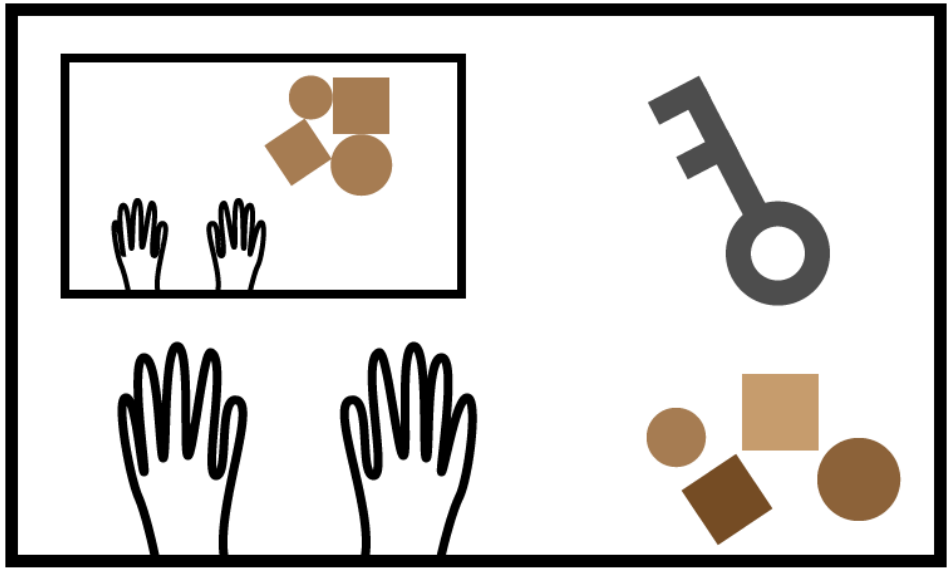
Analogie: Voll homomorphe Verschlüsselung

- Vergleichbar zu einer Handschuhbox
- verschlossen (außer für Alice)
- Mitarbeiter können darin arbeiten
- Mitarbeiter können Dinge hineinlegen
- Alice entnimmt fertiges Ergebnis



Analogie: Bootstrapping

- Handschuhe werden nach 1 Minute unbrauchbar
- Box 2 enthält Master Key für Box 1
- Mitarbeiter packt Box 1 in Box 2
- Öffnet Box 1
- Arbeiter weiter an Produkt
- Nach 1 Minute ...



ANWENDUNGEN

Anwendungen

etwas konkretere Beispiele

- Suche auf verschlüsselten Daten
- Spam-Filter auf verschlüsselten E-Mails
- Elektr. Patientenakten
- Online-Auktionen
- Elektr. Wahlen

ENTSTEHUNG & AKTUELLER STAND

Entstehung

- Geprägt durch Rivest et al. Ende der 1970er
- 2009: Gentrys Verfahren in der Theorie
- seit 2009: Implementierungen von Gentrys Verfahren
- Alternativen, basierend auf Learning With Errors-Problem
- Namen: Vaikuntanathan, Brakersky, Lauter, Halevi

Aktueller Stand

- Gentrys Implementierung: 2 Std. zur Key-Generierung
- Microsoft-Forscher Lauter et al.
  - effizient: 250 Millisekunden für Key-Generierung
  - aber: nur Addition und Multiplikation möglich
  - deckt trotzdem viele Anwendungen ab
- Voll homomorphe Verschlüsselung noch nicht marktfähig

Fazit

- Interessantes, mathematisch sehr komplexes Gebiet
- Im Fahrwasser des Cloud-Hypes
- rasante Entwicklung
- VHV wird in Zukunft einsatzfähig sein
- aber:
  - Wird es sich wirklich durchsetzen?
  - Alternativen: Heim-Server, Standard-Verschlüsselung, Policies
  - Gesetzlich eingeschränkt?


That's all Folks!