



# Formel

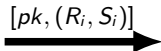
Inhalt...

## Algorithmus basierend auf ElGamal

### Alice

- ⇒ Erstellt Bloom Filter ihrer Daten
- ⇒ Verschlüsselt jede Stelle ihres Bloom Filters mittels ElGamal

$$(R_i, S_i) = (g^{r_i}, pk^{r_i} * g^{1-BF_1[i]})$$



### Bob

- ⇒ Erstellt Bloom Filter seiner Daten
- ⇒ Selektiert jene Stellen in seinem BF die den Eintrag null besitzen.

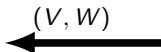
⇒ Multipliziert an diesen Stellen die Werte des Ciphertextes von Alice auf

⇒ Rerandomisiert die entstandenen Ergebnisse

⇒ Alice entschlüsselt mit sk Ciphertext von Bob

⇒ Bestimmt Anzahl der Einträge an denen beide Bloom Filter null sind

⇒ Berechnet die Set-Union der BF



$(V, W)$

$$V = (g^s * \prod_{i:BF_2[i]=0} R_i)$$

$$W = (pk^s * \prod_{i:BF_2[i]=0} S_i)$$

## Algorithmus basierend auf Paillier

### Alice

- ⇒ Erstellt Bloom Filter ihrer Daten
- ⇒ Invertiert jede Stelle des Bloomfilters.
- ⇒ Verschlüsselt jede Stelle ihres Bloomfilters mittels Paillier

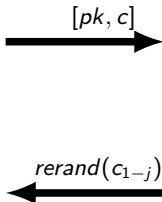
$$c = (g^m * r^n) \bmod n^2$$

- ⇒ Alice entschlüsselt mit sk Ciphertexte von Bob
- ⇒ Bestimmt Anzahl der Einträge an denen beide Bloom Filter null sind
- ⇒ Berechnet die Set-Union der BF

### Bob

- ⇒ Erstellt für jedes Element des Datensatzes einen Bloomfilter seiner Daten
- ⇒ Selektiert in jedem Blommfilter jene Stellen die den Eintrag Eins besitzen.
- ⇒ Addiert an diesen Stellen die Werte des Ciphertextes des Clients auf
- ⇒ Rerandomisiert die entstandenen Ergebnisse mit verschlüsselter Null

$$Rerandc_j = (c_j * encrypt_{paillier}(0))$$



Verhältnis	15/14	15/7.5	15/5	15/2
Runtime (sec)	221	247	211	222
Abweichung	0.01%	3.3%	8.8%	36.8%

Table 1: Hashf: 14, Array:3029660

Runtime (sec)	108	83	47	11
Abweichung	4%	6%	13%	51%
Array	1442696	1009887	577079	144270

Table 2: Verhältnis: 100/1, Anzahl Hashf.: 10