Private Stream Aggregation with Labels in the Standard Model

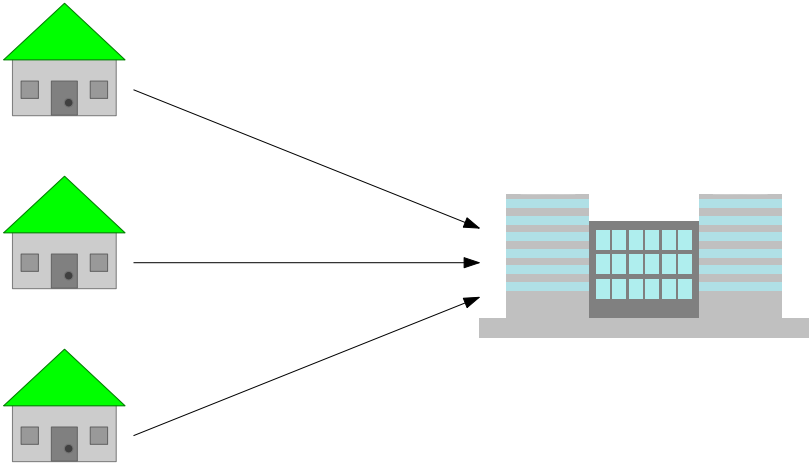
Johannes Ernst [1] Alexander Koch [2]

[1]University of St. Gallen

[2]Karlsruhe Institute of Technology

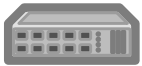
2021-07-12

Smart Meter Aggregation

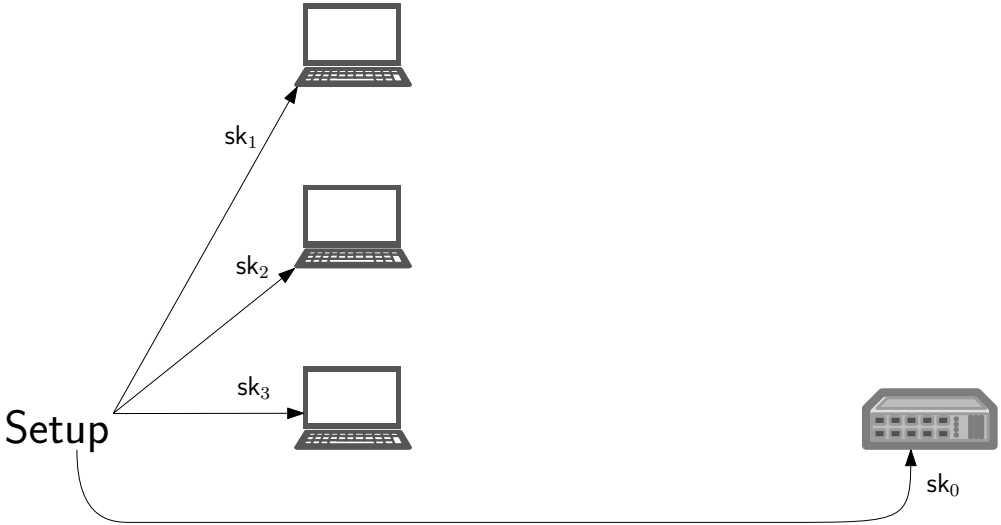


Private Stream Aggregation (PSA) [Shi+11]

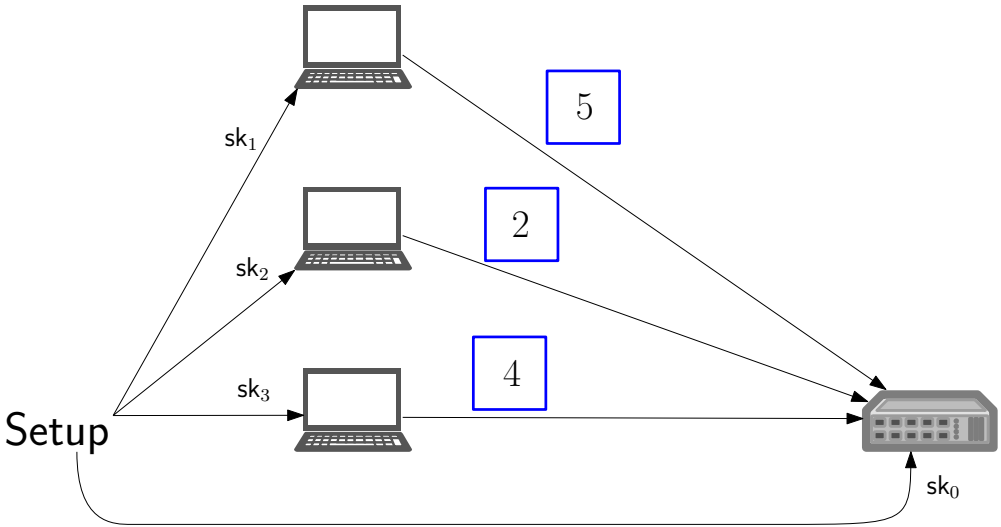
Setup



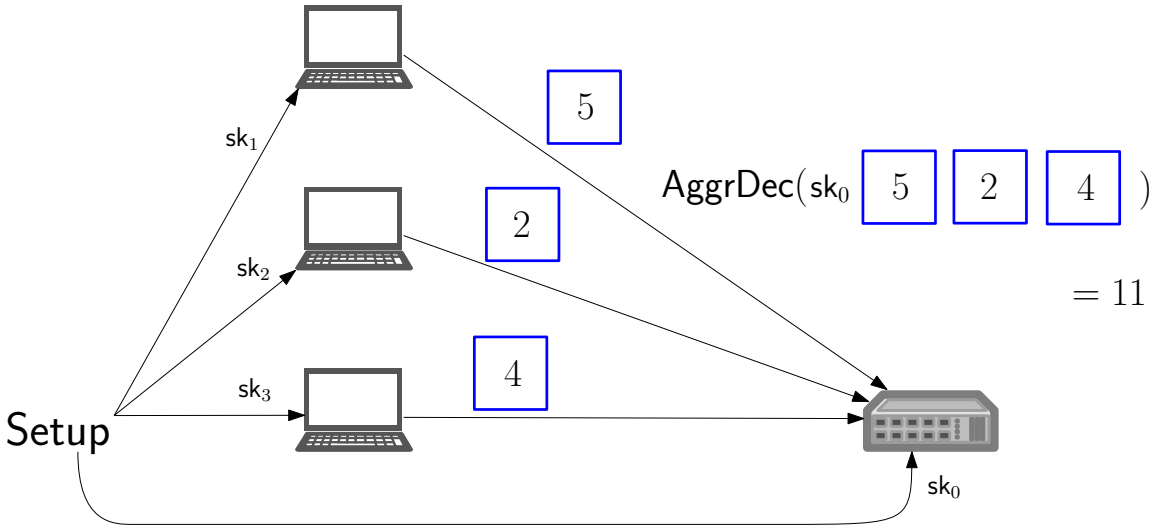
Private Stream Aggregation (PSA) [Shi+11]



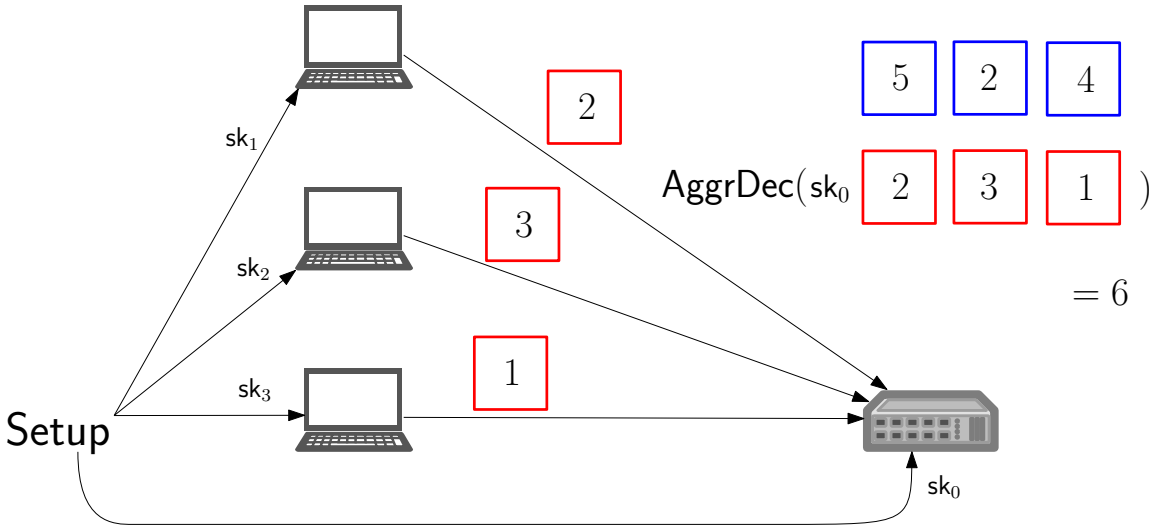
Private Stream Aggregation (PSA) [Shi+11]



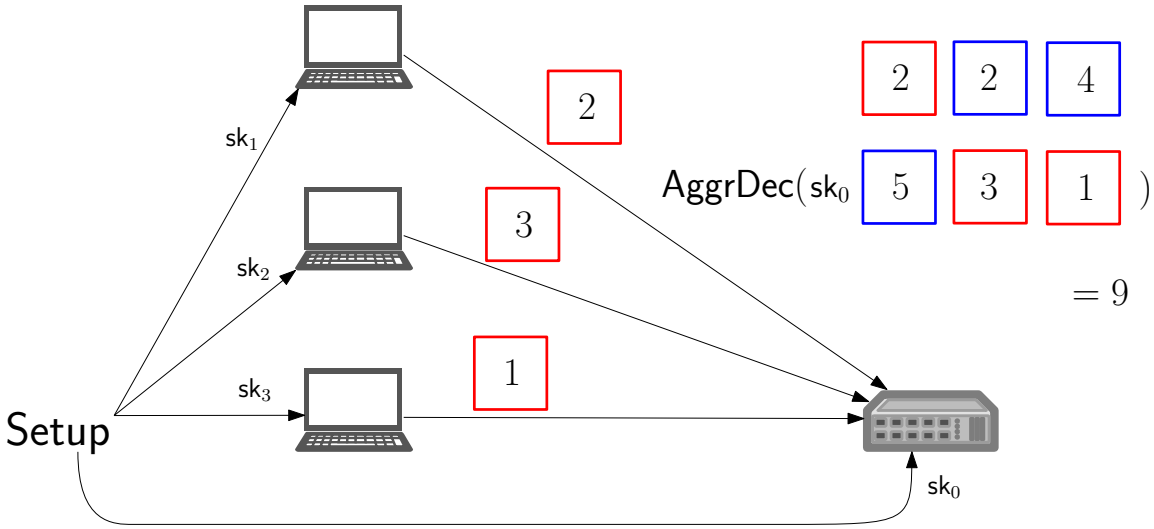
Private Stream Aggregation (PSA) [Shi+11]



Private Stream Aggregation (PSA) [Shi+11]



Private Stream Aggregation (PSA) [Shi+11]



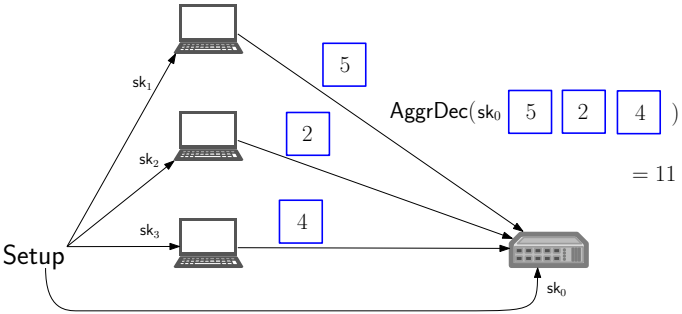
Private Stream Aggregation

PSA

$\mathsf{Setup}(1^\lambda, 1^n) \to (\mathsf{pp}, \mathsf{sk}_0, \ldots, \mathsf{sk}_n)$

$\mathsf{Enc}(\mathsf{pp}, \mathsf{sk}_i, \ell, x_i) \to c_{i,\ell}$

$\mathsf{AggrDec}(\mathsf{pp}, \mathsf{sk}_0, \ell, c_{1,\ell}, \ldots, c_{n,\ell})$
$\to$ sum: $\sum_{i \in [n]} x_{i,\ell}$



Aggregator Obliviousness

- Indistinguishability based game
- Encryption queries
- Corruption queries
- Challenge queries with $\{x_i^0, x_i^1\}_{i \in U}$

Aggregator Obliviousness

- Indistinguishability based game
- Encryption queries
- Corruption queries
- Challenge queries with $\{x_i^0, x_i^1\}_{i \in U}$

Conditions for $\mathcal{A}$:

- If $\mathcal{A}$ has corrupted the aggregator:

$$\sum x_i^0 = \sum x_i^1$$

- *encrypt-once*: only get one message per user and label

Contribution

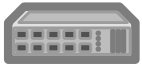
- PSA with labels in standard model based on key-homomorphic PRF
  - similar to [Val17], using proof techniques of [ABG19]
- Implementation with lattice based PRF in ROM
- Performance evaluation and comparison
- Description of decentralized setup and fault tolerance

Key-Homomorphic Pseudorandom Functions

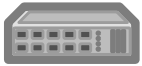
- Pseudorandom Function:
  - $\mathsf{PRF}_k \colon \mathcal{Z} \to \mathcal{Y}$
  - $k \xleftarrow{\$} \mathcal{K} \colon \mathsf{PRF}_k \overset{c}{\approx} \mathsf{RF}$
- Key-Homomorphic Pseudorandom Function:
  - $(\mathcal{K}, +), (\mathcal{Y}, +)$ are groups
  - $\forall z, k_1, k_2 \colon \mathsf{PRF}_{k_1}(z) + \mathsf{PRF}_{k_2}(z) = \mathsf{PRF}_{k_1 + k_2}(z)$

PSA Scheme with Key-Homomorphic PRF

Setup



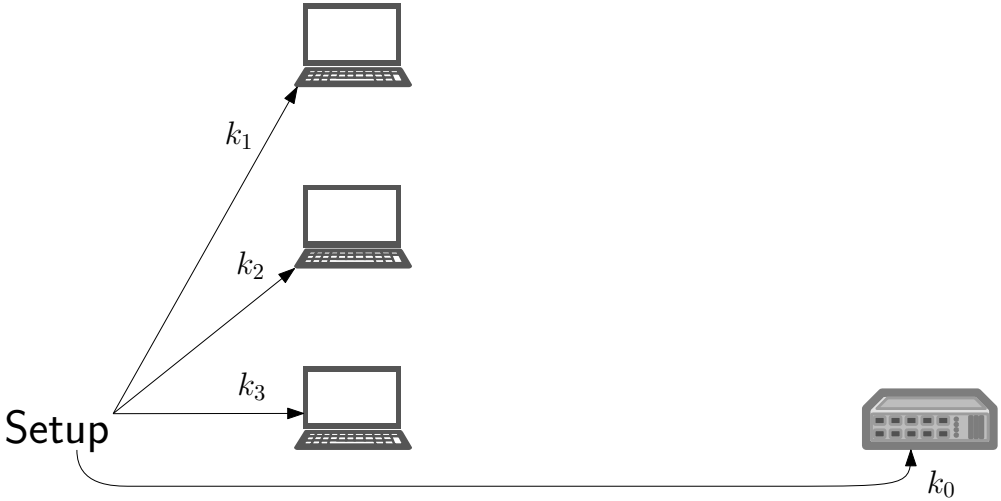
PSA Scheme with Key-Homomorphic PRF



Setup

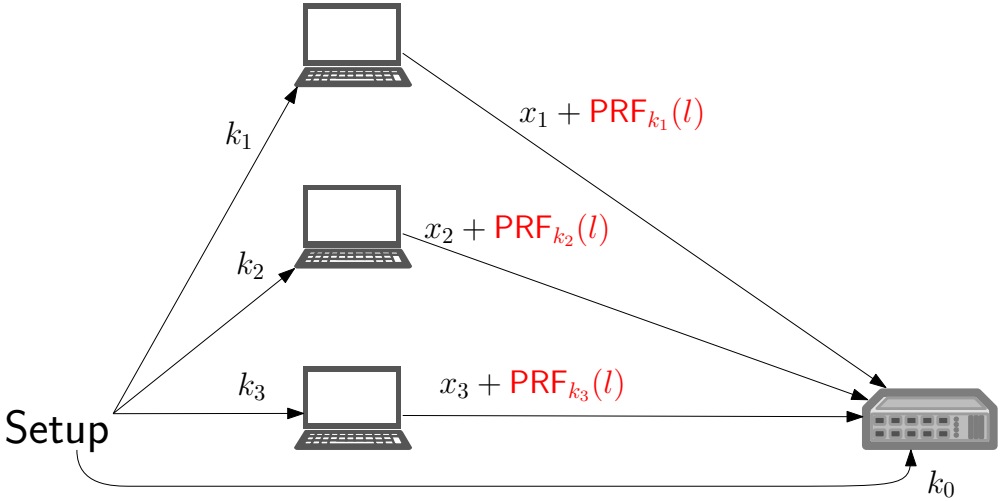
$$k_0 = k_1 + k_2 + k_3 \Rightarrow \mathsf{PRF}_{k_0}(l) = \mathsf{PRF}_{k_1}(l) + \mathsf{PRF}_{k_2}(l) + \mathsf{PRF}_{k_3}(l)$$

PSA Scheme with Key-Homomorphic PRF



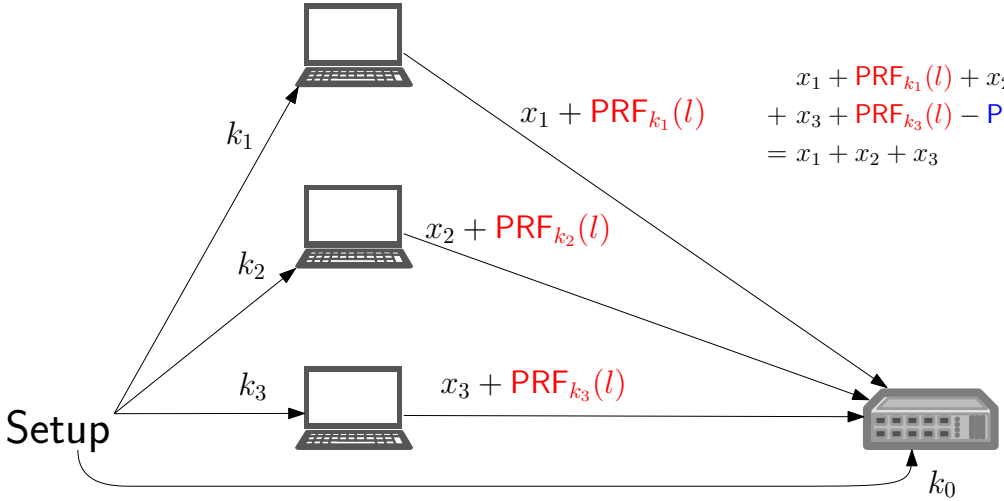
$$k_0 = k_1 + k_2 + k_3 \Rightarrow \mathsf{PRF}_{k_0}(l) = \mathsf{PRF}_{k_1}(l) + \mathsf{PRF}_{k_2}(l) + \mathsf{PRF}_{k_3}(l)$$

PSA Scheme with Key-Homomorphic PRF



$$k_0 = k_1 + k_2 + k_3 \Rightarrow \mathsf{PRF}_{k_0}(l) = \mathsf{PRF}_{k_1}(l) + \mathsf{PRF}_{k_2}(l) + \mathsf{PRF}_{k_3}(l)$$

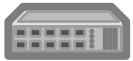
PSA Scheme with Key-Homomorphic PRF



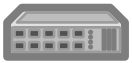
$$x_1 + \mathsf{PRF}_{k_1}(l) + x_2 + \mathsf{PRF}_{k_2}(l) + x_3 + \mathsf{PRF}_{k_3}(l) - \mathsf{PRF}_{k_0}(l) = x_1 + x_2 + x_3$$

$$k_0 = k_1 + k_2 + k_3 \Rightarrow \mathsf{PRF}_{k_0}(l) = \mathsf{PRF}_{k_1}(l) + \mathsf{PRF}_{k_2}(l) + \mathsf{PRF}_{k_3}(l)$$

Decentralizing the Setup (techniques from [Cho+20])

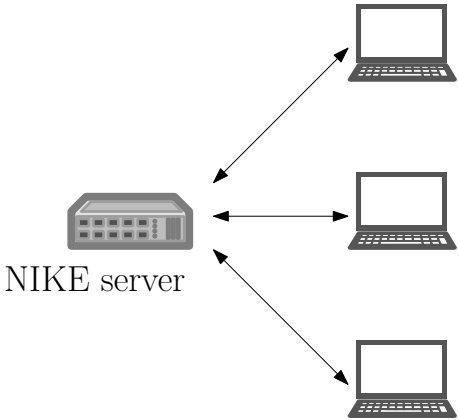


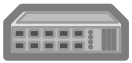
NIKE server



$$k_0 = k_1 + k_2 + k_3$$

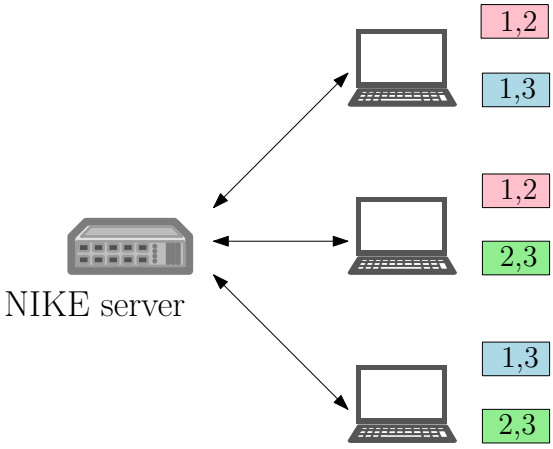
Decentralizing the Setup (techniques from [Cho+20])

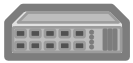




$$k_0 = k_1 + k_2 + k_3$$

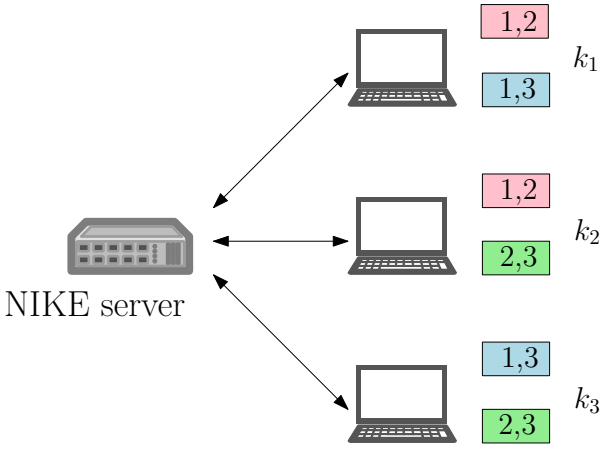
Decentralizing the Setup (techniques from [Cho+20])



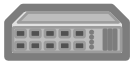


$$k_0 = k_1 + k_2 + k_3$$

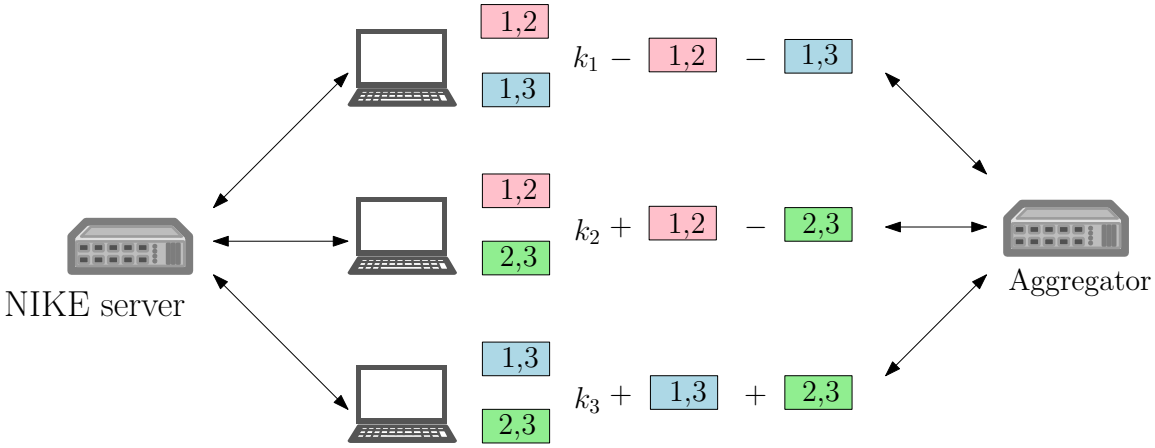
Decentralizing the Setup (techniques from [Cho+20])



$$k_0 = k_1 + k_2 + k_3$$



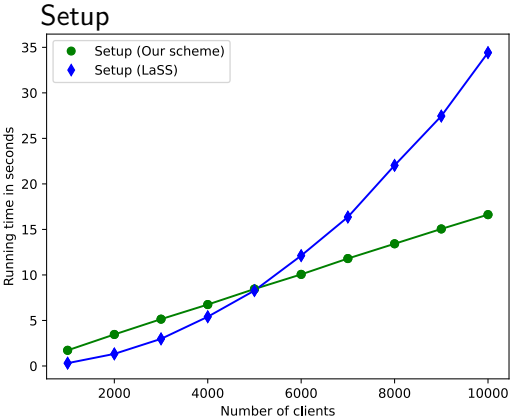
Decentralizing the Setup (techniques from [Cho+20])

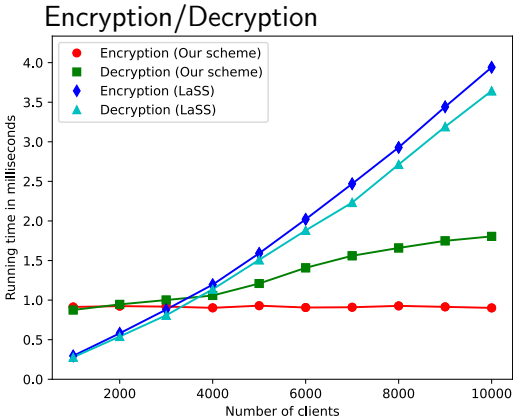


Implementation

- Written in Go (source code on https://github.com/johanernst/khPRF-PSA)
- PRF based on LWR, secure in ROM
  - $k \in \mathbb{Z}_q^{2096}$, $H \colon \mathcal{L} \to \mathbb{Z}_q^{2096}$, $q = 2^{128}$, $p = 2^{85}$
  - $\mathrm{PRF}_k(\ell) := \lfloor \langle H(\ell), k \rangle \rfloor_p \in \mathbb{Z}_p$ ([Bon+13])
  - For H we use 524 calls to SHA3-512
- Key-size: $\approx$ 268 KBit
- Message-size: 85 Bit
- 114 bit security for message space of size 2^{64} and 2^{20} users

Performance in Practice





LaSS: Waldner et al. [Wal+21]

Results and Limitations

- Properties of new scheme:
  - supports labels, secure under adaptive corruptions
  - secure in the standard model (when instantiated with PRF from [Bon+13] or [BP14])
  - small ciphertexts, fast encryption and decryption
- Limitations:
  - implementation (so far) only in ROM
  - encrypt-once restriction