



Private Robust & Efficient Computation of Aggregate Statistics

N. Cheng, A. Mitrokotsa

Cybersecurity & Applied Cryptography Chair
University of St. Gallen, Switzerland

Our Team



Katerina Mitrokovska
Full Professor in Cyber Security



Eriane Breu
Administration



Novak Kaluđerović
Postdoctoral Researcher



Uddipana Dowerah (External)
Postdoctoral Researcher



Frank Hartmann
Cryptography Engineer



Johannes Ernst
Doctoral Researcher



Nan Cheng
Doctoral Researcher



Jenit Tomy
Doctoral Researcher



Yu Liujun
Doctoral Researcher



Healthcare Data



Physical Activity
Data



Environmental
Data

Data Driven World



Aggregation Statistics

Health data



Diagnosis



Research
analytics



Medical
Monitoring

Governmental data



Education



Taxes



Immigration



Income

Financial data



Economic
growth



Market Prediction

Environmental data



Smart
metering



Environmental
Monitoring

Personal Data



Website
preferences



Lifestyle
tracking

What are they?

- ▶ Addition
- ▶ Max, Min
- ▶ KRE
- ▶ Average
- ▶ ...



Repair

Attack



Goal: Provable Security ← Well-Studied Assumptions
(e.g., hardness of factoring)

Main requirements



Privacy



Robustness



Efficiency

Main requirements



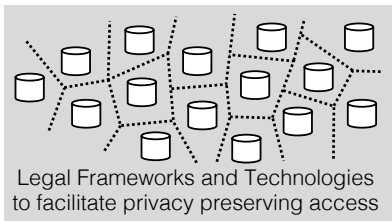
Privacy



Robustness



Efficiency



Data Silos

- Privacy Laws
- Competition

- ▶ EU Data Governance ACT (DGA) (effective from 2023)
facilitate the reuse of protected public-sector data

Aggregation Statistics: Examples



x_1



x_2

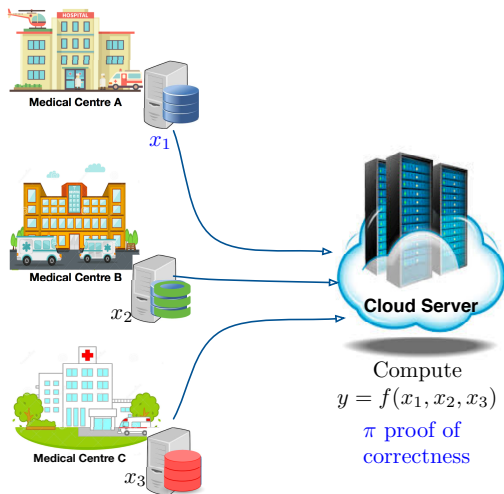
⋮



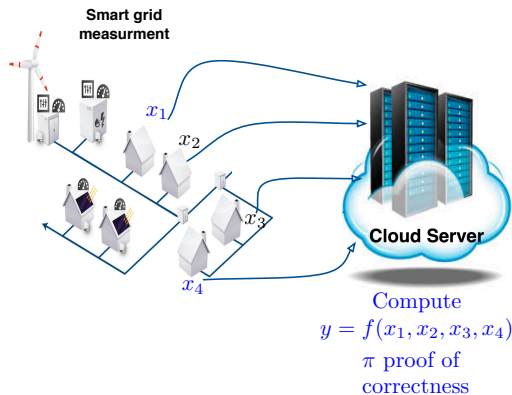
x_n



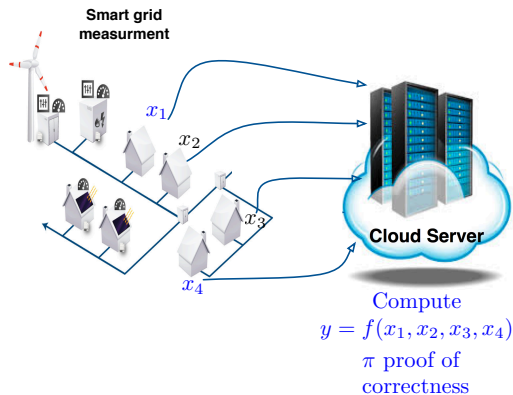
Aggregation Statistics: Examples



Aggregation Statistics: Examples



Aggregation Statistics: Examples



Privacy

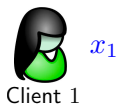


Robustness



Efficiency

Should we have a single server?



⋮



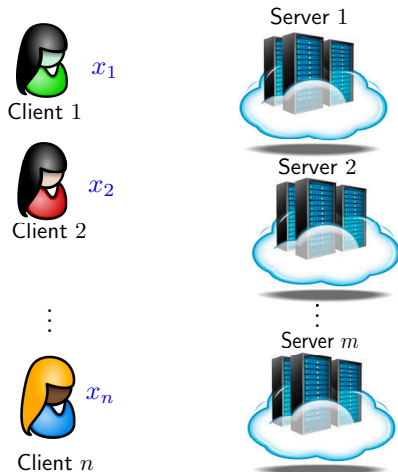
Should we have a single server?



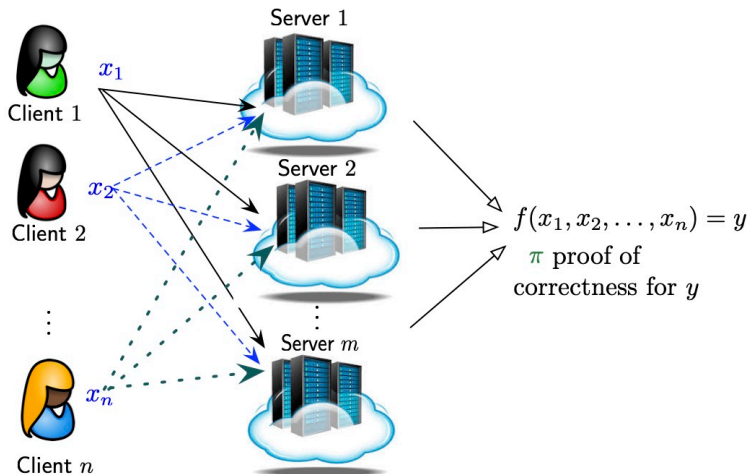
⋮



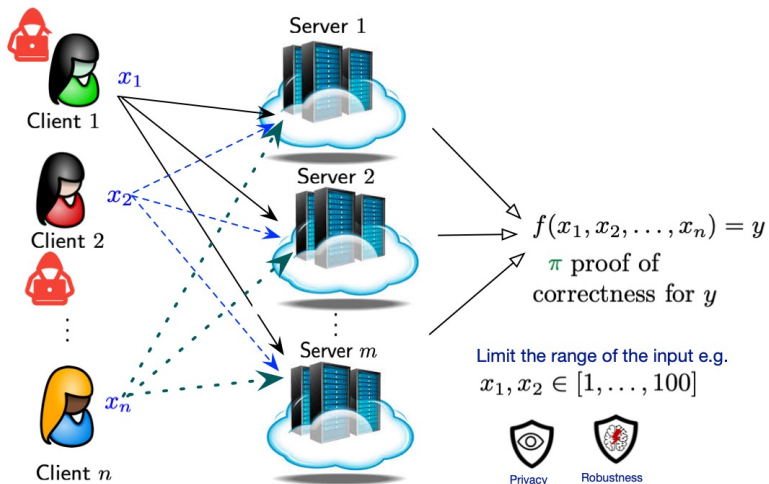
Should we have a single server?



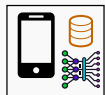
Should we have a single server?



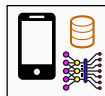
What about Malicious Clients?



Aggregation Statistics in Federated Learning

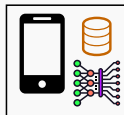


Client 1



Client 2

⋮

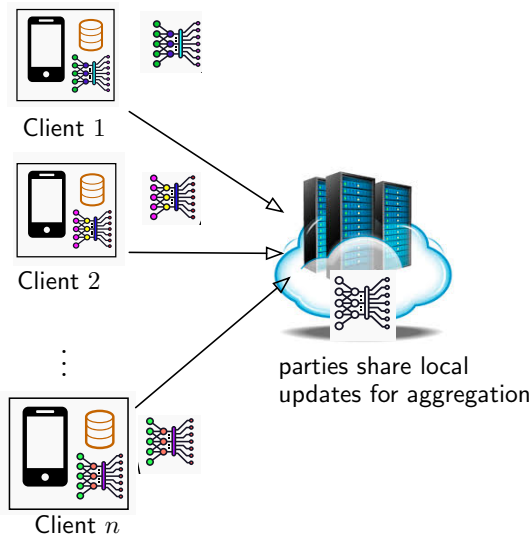


Client n

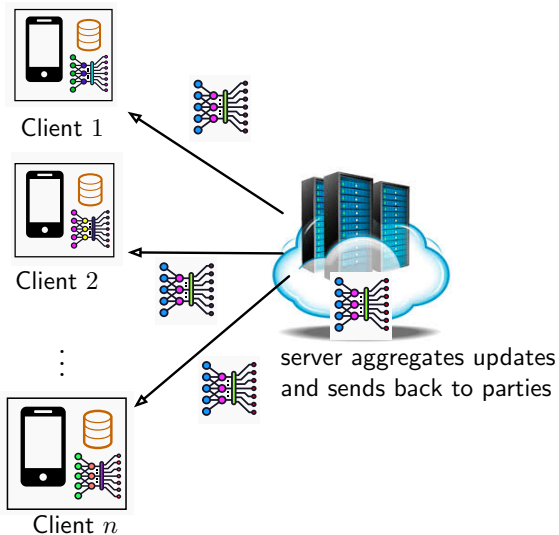


each party makes an update
using its local dataset

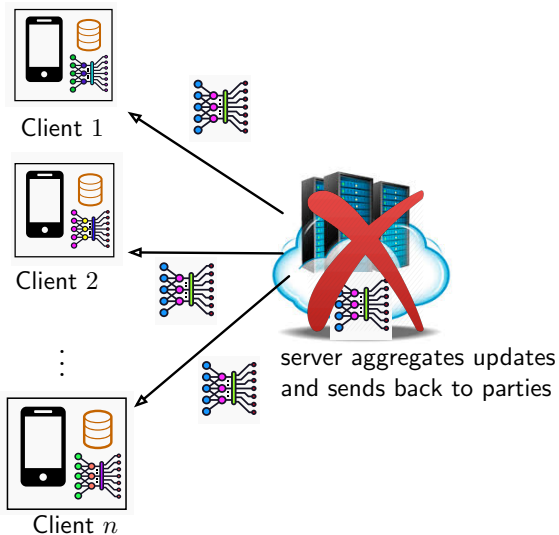
Aggregation Statistics in Federated Learning



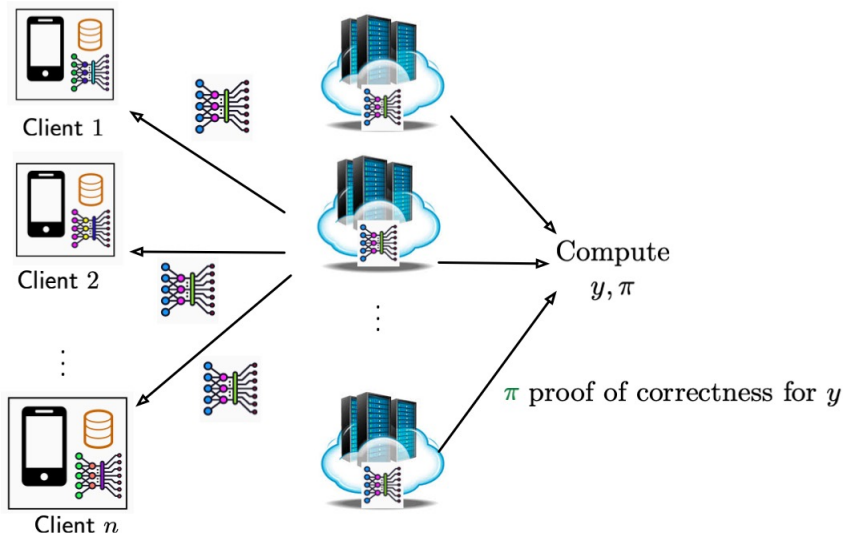
Aggregation Statistics in Federated Learning



Aggregation Statistics in Federated Learning



Aggregation Statistics in Federated Learning



Main Objectives in Secure Aggregation

- ▶ **Decentralise** the computation of aggregation statistics
- ▶ **Robust** computation of aggregation statistics (considering malicious clients/servers)
- ▶ High **efficiency** (communication & computation)

Main Objectives in Secure Aggregation

- ▶ **Decentralise** the computation of aggregation statistics
 - ▶ **Robust** computation of aggregation statistics (considering malicious clients/servers)
 - ▶ High **efficiency** (communication & computation)
-
- ▶ **Understand Trade-offs between objectives we care about!**



Privacy



Robustness



Efficiency

Can we solve this?

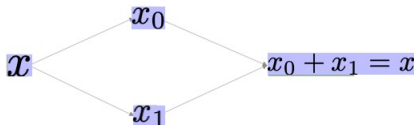
- ▶ FHE based solutions?
 - ▶ Can be **expensive** for general functions!
 - ▶ May require key management for multiple parties!



- ▶ Can we do better?
 - ▶ Need a solution that scales well and provides strong privacy and efficiency guarantees!
 - ▶ Secret Sharing & Function Secret Sharing!

What is Secret Sharing (SS)?

- ▶ Share a value $x \in G$ among multiple parties
- ▶ Parties have no information on x

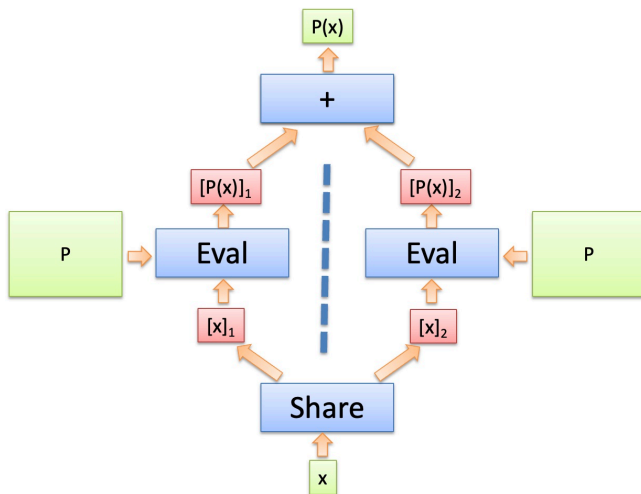


- ▶ x can be re-constructed from x_i 's
- ▶ Denoted with $[x]$

Sharing types

- ▶ Boolean SS: $x \in \{0,1\}^n$, for each $j \in [m]$, party P_j holds $\{[x_i]_j\}_{i \in [n]}$ that $\oplus_{i=1}^n [x_i]_j = x_i$
- ▶ Arithmetic SS: $x \in \{0,1\}^n$, for each $j \in [m]$, party P_j holds $[x]_j$ that $\sum_{j=1}^m [x]_j = x \bmod \mathbb{Z}_{2^n}$

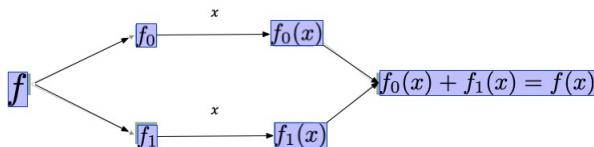
Computations using Secret Sharing



- Widely-used, e.g., biometric authentication, PPML.

What is Function Secret Sharing ¹ (FSS)?

- ▶ Share a function $f : \{0,1\}^n \rightarrow G$ among multiple parties
- ▶ Parties have no information on f
- ▶ f_i can be evaluated efficiently at any x

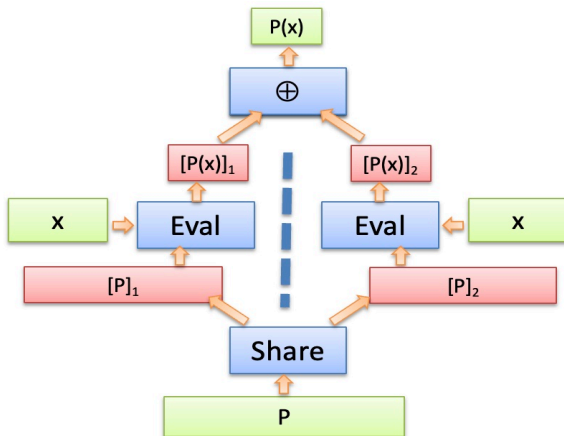


- ▶ $f(x)$ can be re-constructed from $f_i(x)$'s

¹Boyle, Elette, Niv Gilboa, and Yuval Ishai. Function secret sharing. EUROCRYPT 2015

Computation using 2-Party FSS ²³

- ▶ Reverse roles of function/program and input
- ▶ Share size can grow with program size



²Boyle, Elette, Niv Gilboa, and Yuval Ishai. Function secret sharing. EUROCRYPT 2015

³Boyle, Elette, Niv Gilboa, and Yuval Ishai. Secure computation with preprocessing via function secret sharing. TCC 2019

Constructions and applications

Very efficient constructions for *simple* classes from OWF

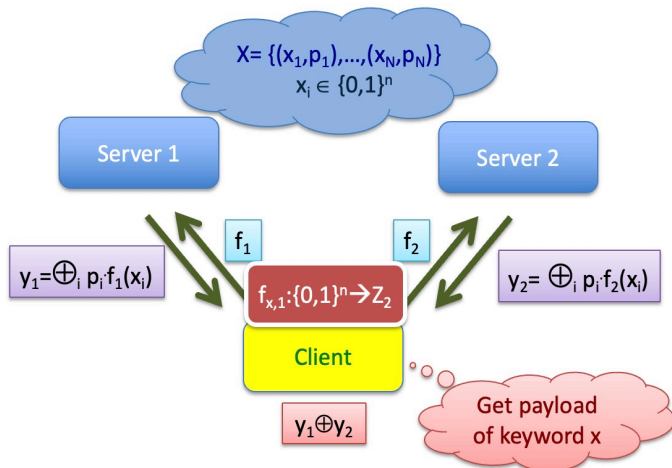
- ▶ Distributed Point function
 - ▶ $f_{\alpha,\beta}(\alpha) = \beta$
 - ▶ $f_{\alpha,\beta}(x) = 0$ for $x \neq \alpha$
- ▶ Distributed Comparison function
 - ▶ $f_{\alpha,\beta}(x) = \beta$ for $x < \alpha$
 - ▶ $f_{\alpha,\beta}(x) = 0$ for $x \geq \alpha$
- ▶ Interval Containment function, etc.

Applications

- ▶ Reading (e.g., PIR)
- ▶ Writing (e.g., private storage)
- ▶ General secure computation.

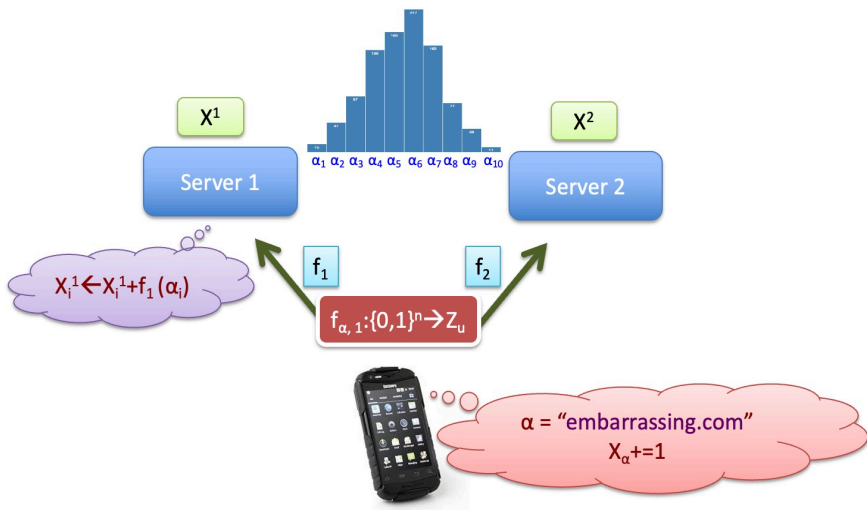
Applications: Private information Retrieval

- ▶ Keyword search with payloads



Applications: Secure aggregation

- ▶ Client does not need to know which items are being tracked
- ▶ Server's work proportional to number of items being tracked



Related Work

Ref.	Techniques	Aggregation	Pro	Con
Prio ⁴	SNIPS	Sum, variance; Or/and; Max/min; Frq Count	Non-interactive Validation Lightweight proof	Heavy encoding Limited efficiency
Prio+ ⁵	SNIPS + Boolean SS	-	Reduced overhead	Limited efficiency
Poplar ⁶	IDPF+ Malicious Sketching	t -heavy hitters	Efficiency	Result must be revealed
Doplar ⁷	V-IDPF	-	One-round reduction	-

- ▶ Prio and Prio+ assume malicious clients and **semi-honest** servers.
- ▶ Poplar and Doplar assume both malicious clients and **malicious** servers.

⁴Corrigan-Gibbs, H., Boneh, D. Prio: Private, robust, and scalable computation of aggregate statistics. (NSDI 2017)

⁵Addanki, Surya, et al. Prio+: Privacy preserving aggregate statistics via boolean shares. (SCN 2022)

⁶Boneh, Dan, et al. Lightweight techniques for private heavy hitters. (S&P21)

⁷Davis, Hannah, et al. "Verifiable Distributed Aggregation Functions." (PETS 2023)

Incremental Distributed Point Function (IDPF)

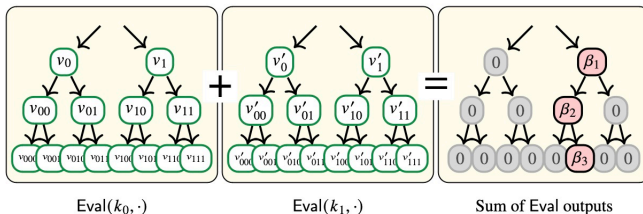
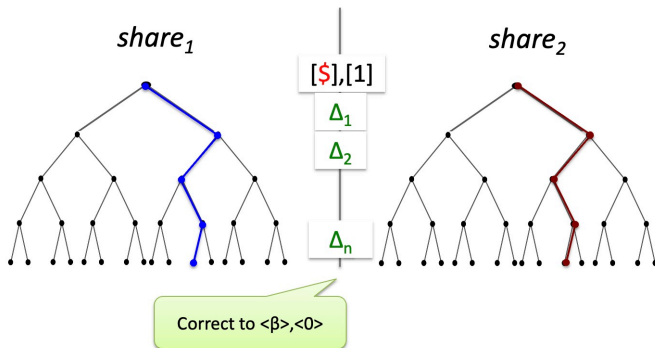
Definition. Given a random binary string $\alpha \in \{0,1\}^n$ and values $\beta_1 \in \mathbb{G}_1, \dots, \beta_n \in \mathbb{G}_n$, it outputs a key pair (k_0, k_1) .

$$(k_0, k_1) \leftarrow \text{IDPF.KeyGen}(\alpha, \beta_1, \dots, \beta_n)$$

ensuring that $\forall j \in \{1, \dots, n\}, x \in \{0, 1\}^j$,

$$\text{IDPF.Eval}^f(k_0, 0, x) + \text{IDPF.Eval}^f(k_1, 1, x) = \begin{cases} \beta_j & \text{if } x = \text{Prefix}(\alpha, j) \\ 0 & \text{otherwise.} \end{cases}$$

DPF vs IDPF



Main Project Objectives

Task 1: Compute aggregation statistics against malicious clients

- ▶ Avoid (data or even model) **poisoning** attacks of the final computation
- ▶ Traditional approach use zero knowledge (range) proofs
- ▶ Use Function Secret Sharing (FSS) which provides high efficiency guarantees.

Main Project Objectives

Task 2: Compute aggregation statistics against malicious servers

- ▶ Servers may bias the final result and provide false aggregation computations
- ▶ Control malicious servers behaviour using secure multi party computation techniques (e.g., MAC verification of SPDZ)
- ▶ Combine in an efficient way function secret sharing and MAC approach from SPDZ framework

Main Project Objectives

Task 3: Computing aggregation statistics with high efficiency guarantees

- ▶ Consider that the servers perform the input verification instead of the clients before the actual aggregation;
- ▶ Recent advances of FSS based computation gates with optimised communication rounds and volume (e.g., DPF, DCP, ICF) suitable for conditional checks.

A Preliminary Result

Input Boolean SS from potential malicious clients

- ▶ Flexible for different sources
- ▶ Easy to validate client inputs.

Computing secure aggregation (e.g., k^{th} ranking element) using IDPF/VIDPF

- ▶ sub-linear communication volume cost.
- ▶ communication rounds depends only on input length
- ▶ communication rounds independent with the number of submissions
- ▶ can be easily extended to a malicious server setting using SPDZ

Thank you for your Attention!

Contact: <https://cybersecurity.unisg.ch/people>