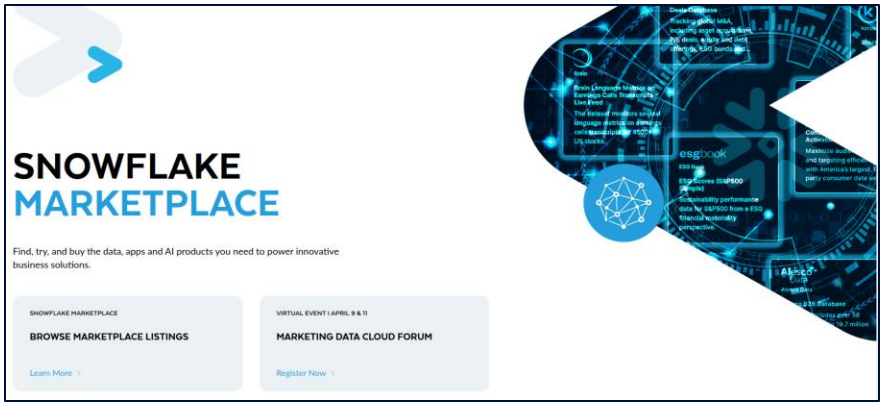
PraaS: Verifiable Proofs of Property as-a-Service with Intel SGX

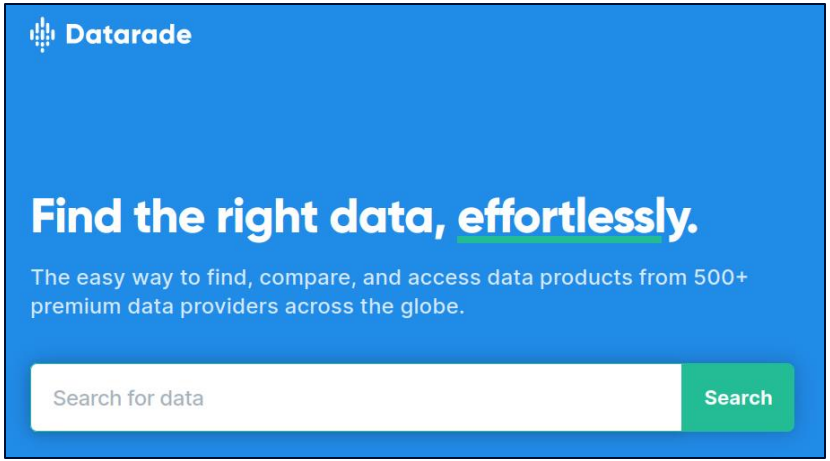
Istemi Ekin Akkus, Ivica Rimac, Ruichuan Chen

08.07.2024

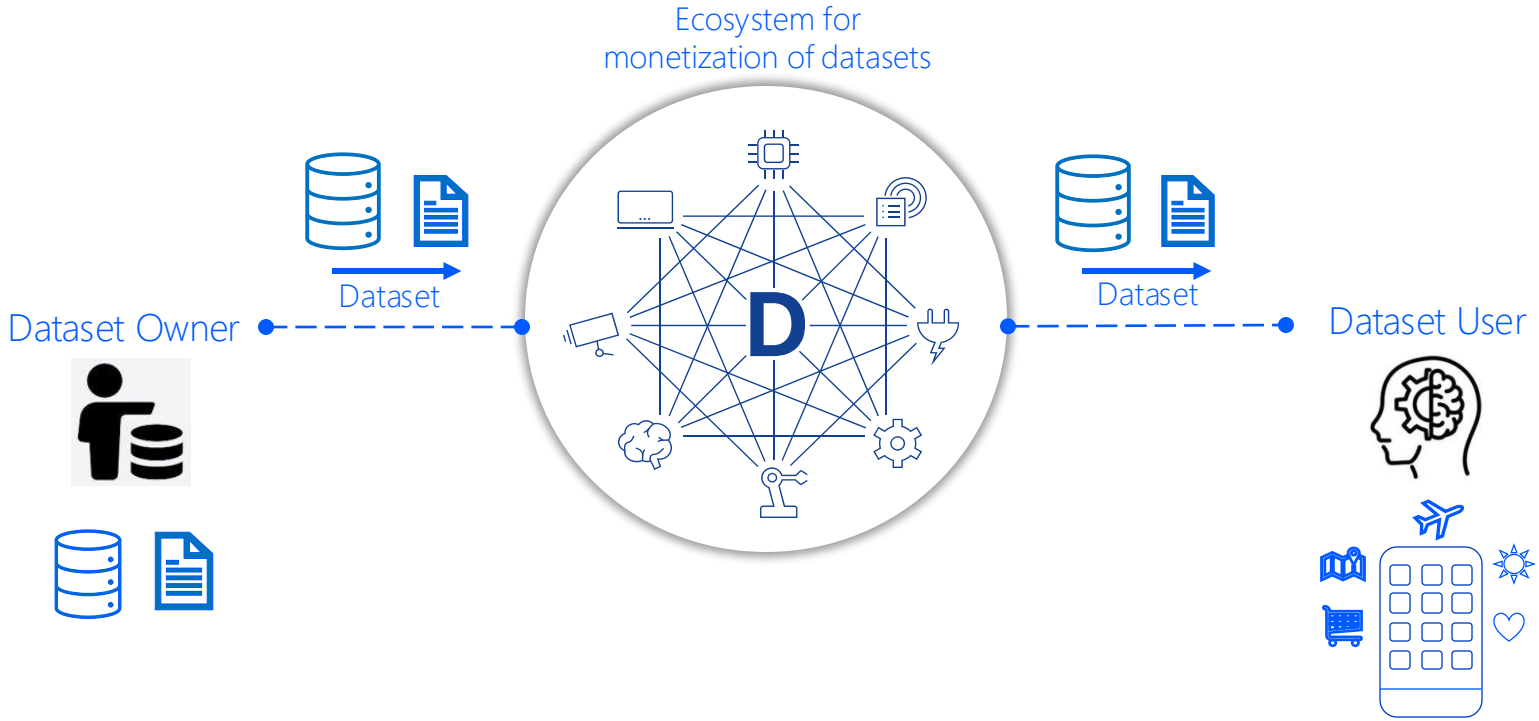
7th Workshop on System Software for Trusted Execution (SysTEX 2024)

The future is full of datasets

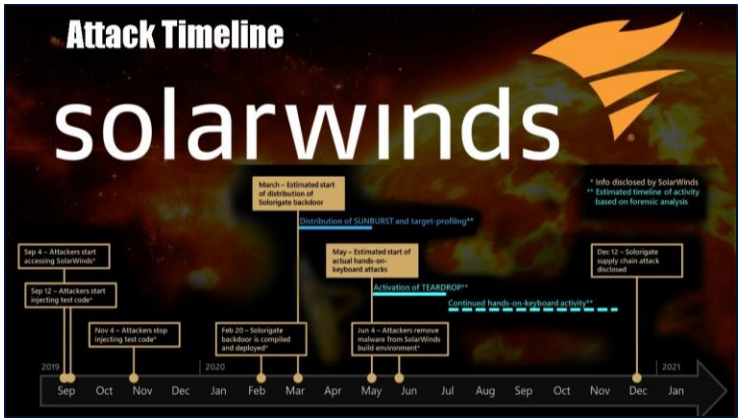




The future is an Industry 4.0 ecosystem



The future is full of software supply chain attacks



ENERGY

Ransomware attack forces shutdown of largest fuel pipeline in the U.S.

PUBLISHED SAT, MAY 8 2021•8:48 AM EDT | UPDATED SUN, MAY 9 2021•9:21 AM EDT

Emma Newburger
@EMMA_NEWBURGER

SHARE

Colonial Pipeline

NEWS

SentinelOne: More supply chain attacks are coming

At RSA Conference 2021, SentinelOne threat researcher Marco Figueroa discussed the implications of the SolarWinds attacks, which he called one of the biggest hacks ever.

By Arielle Waldman, News Writer

Published: 19 May 2021

[oss-security] backdoor in upstream xz/liblzma leading to ssh server compromise

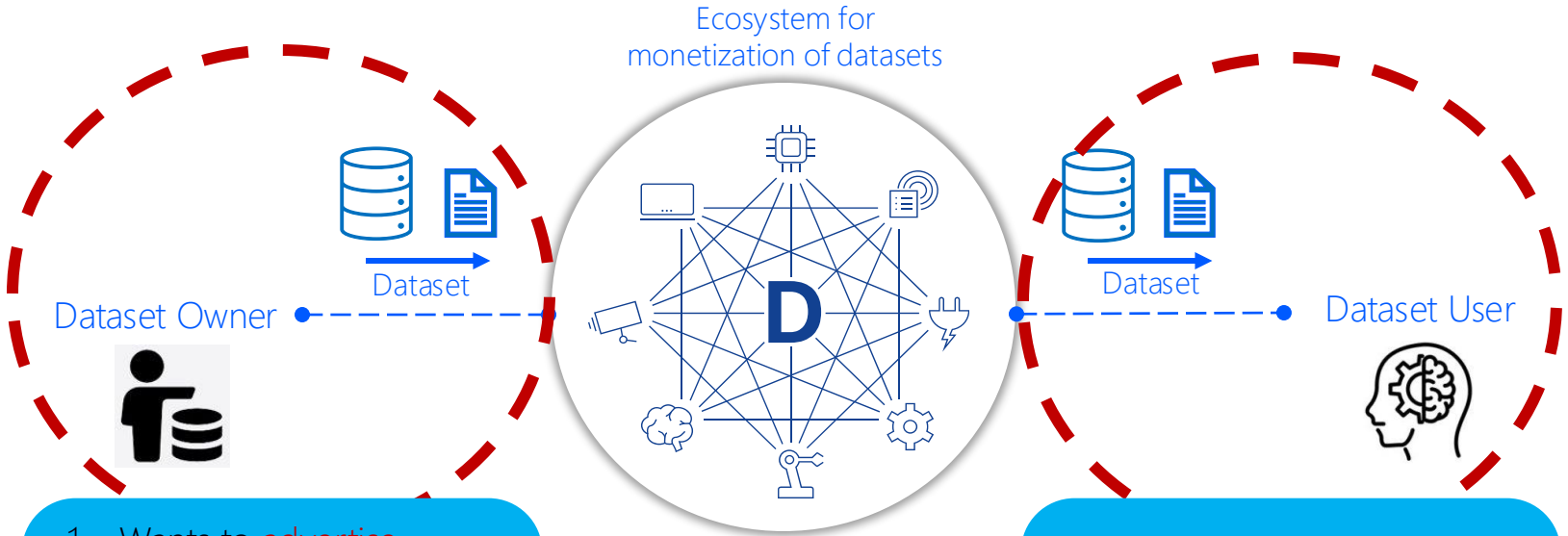
Thread information [Search the oss-security archive]

Andres Freund [this message]
` Alex Gaynor

CVE-2024-3094
Severity: 10.0

NOKIA BELL LABS

Motivation for an Industry 4.0 Ecosystem



1. Wants to advertise properties of their assets effectively
2. Wants to protect the confidentiality of their assets

Wants to check properties of the assets before buying

- Statistics, formatting, provenance, internal consistency, privacy, copyrighted material, …

Additional Requirements & Goal

Utility is application-dependent

- Customizability
- On-demand computation

Datasets are often large

- Scalability

Datasets can be streaming

- Low latency

Monetization is important

- Low cost

What is the minimum cost and trust to obtain the maximum performance?
- Without violating confidentiality of datasets and not breaking other requirements

Goal & Idea

Provide 3rd party verifiable proofs about datasets

with

high scalability, low latency, low cost and “acceptable trust”

while

preserving confidentiality of datasets

Trusted Execution Environments (TEEs) in public clouds

NOKIA BELL LABS

Agenda

- Motivation

- Background & Assumptions
  - SGX Remote Attestation
  - Threat Model and Assumptions

- PraaS Overview

- Evaluation

SGX Remote Attestation

Ensuring the intended enclave is running

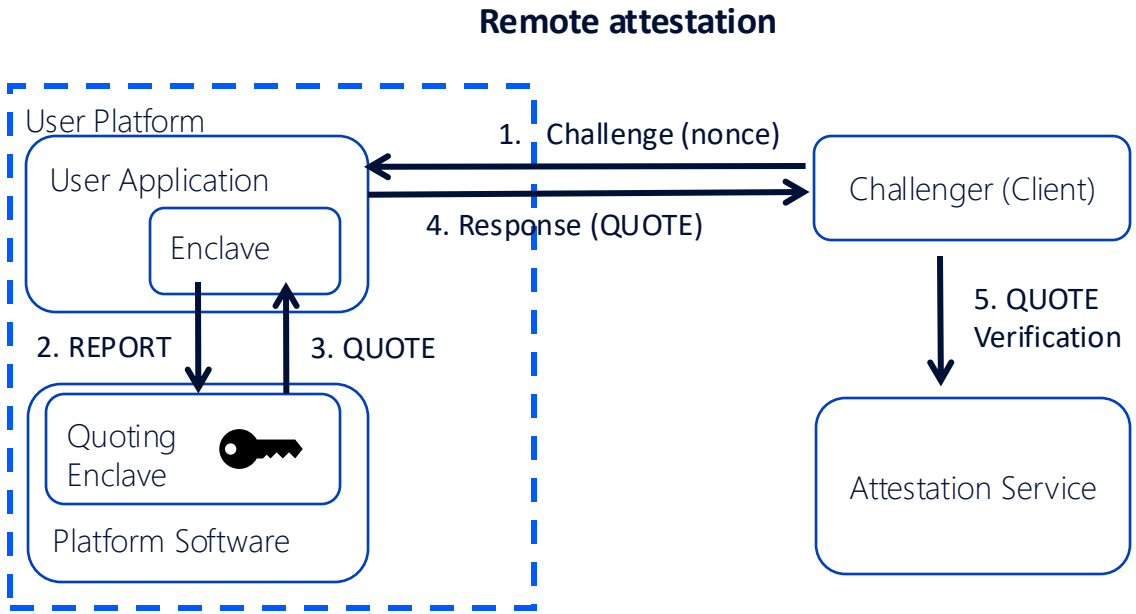
When initiated, an enclave produces an attestation report/quote containing:

- Cryptographic hash of code and data (MRENCLAVE)
- A signature via the attestation key of the hardware

Local attestation: between two enclaves on the same platform

Remote attestation: between a client and an enclave

- Increased confidence that the intended software is running in an SGX enclave with latest TCB version



Actors, Threat Model and Assumptions

Actors

- **Dataset Owner**: Wants to prove to others that a confidential dataset satisfies certain properties without exposing it to others

- **Dataset User**: Wants to obtain guarantees about datasets before purchasing and using them in their application
- **PraaS Provider**: Operates the necessary **software infrastructure** in a cloud setting
- **Cloud Provider**: Provides **hardware infrastructure** with standard security practices and up-to-date TEEs

Threat Model & Assumptions

- No collusion between PraaS/cloud provider and dataset owner/user
- No attacks on TEEs
- Instantiation with Intel SGX
- Supplementary protocols not in scope

NOKIA BELL LABS

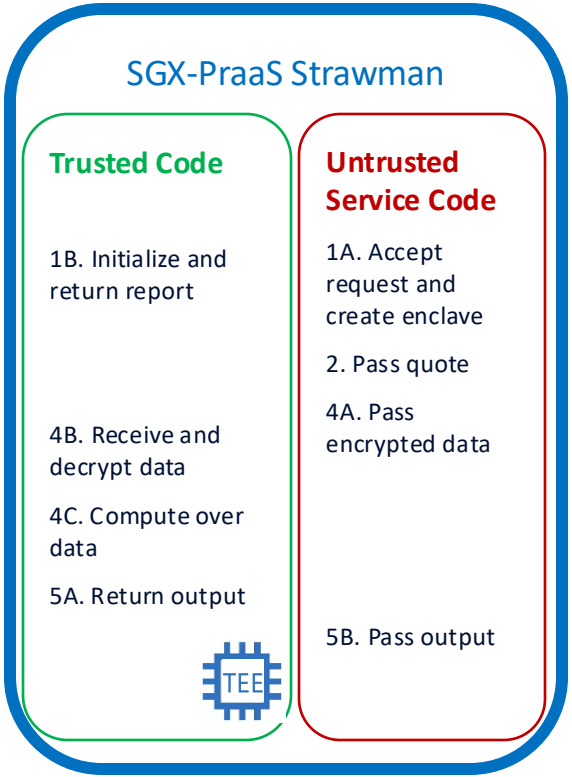
Agenda

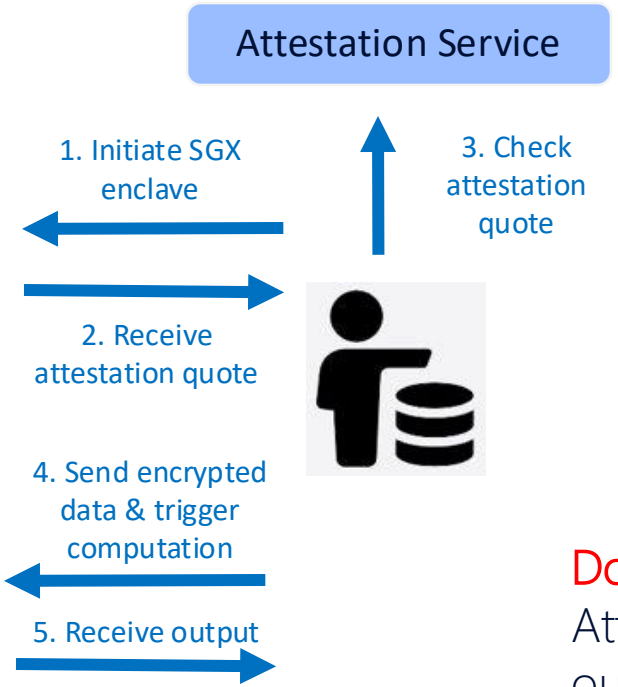
- Motivation
- Background & Assumptions
- PraaS Overview
  - Enclave-signed output
  - Property Computation Functions (PCFs)
- Prototype Implementation and Evaluation

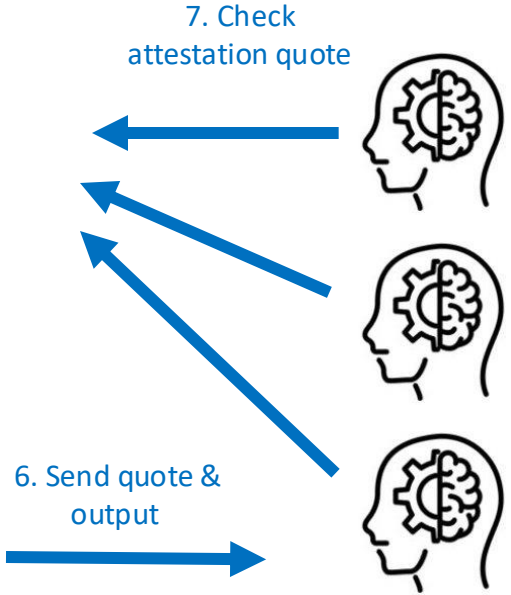
NOKIA BELL LABS

Proof Generation

Strawman







Does not work!

Attestation result and SGX output are not linked together!

NOKIA BELL LABS

Enclave-signed Output

Linking attestation with computation

SGX-PraaS

Trusted Code

1B. Initialize, **generate an ephemeral public/private keypair** and return report **with public key**

4B. Receive and decrypt data

4C. Compute over data

5A. **Sign output** and return it

TEE

Untrusted Service Code

1A. Accept request and create enclave

2. Pass quote **with public key**

4A. Pass encrypted data

5B. Pass **signed output**

Attestation Service

1. Initiate SGX enclave

2. Receive attestation quote **with public key**

3. Check attestation quote **with public key**

4. Send encrypted data & trigger computation

5. Receive **signed output**

6. Send attestation quote **with public key & signed output** to third parties

7. Check attestation quote **with public key**

8. Check output signature **with public key from attestation quote**

Verifiable Proof of Property:
Attestation quote with enclave public key + signed output

NOKIA BELL LABS

Property Computation Functions (PCFs)

- Extract a desired property from a dataset
  - Statistical properties, formatting, internal consistency, anonymization, existence of PII/copyrighted material, …

- Envisioned as a catalogue of useful functions to be picked from
  - **Examples:** sampling, non-repetition + sampling, statistics, sampling + statistics, …

- Available to both dataset owners and potential dataset users
  - Dataset owners inspect to check if it is leaking confidential data
  - Dataset users inspect to verify it is computing the desired property
  - Both can reject if not satisfied

NOKIA BELL LABS

Enclave Templates

- Most of enclave code is generic
  - Build scripts, declarations, common libraries
- Several common steps for Proof-of-Property
  - Initialization with ephemeral public/private keypair, receiving encrypted data, signing output

Enclave template

Property Computation Function = Enclave template + custom property logic

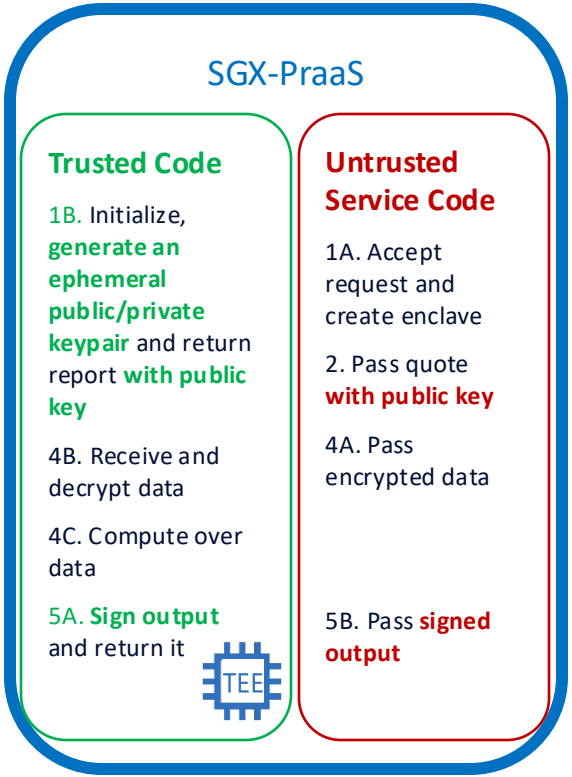
Faster development

Easier customization

Easier reproducibility

NOKIA BELL LABS

Architecture Details



Enclave template

System code

(Untrusted) Service code

- Enclave initialization (1A)
- Proxying encrypted data between client and enclave (2, 4A, 5B)

Client code

- Attestation result check
- Signature check

Enclave code for Proof-of-Property computation

- General steps: pub/private keypair generation (1B), data decryption (4B), output signing(5A)
- Custom step: property extraction (4C)

User-provided code

Agenda

- Motivation
- Background & Assumptions
- PraaS Overview
- Prototype Implementation and Evaluation
  - Common operations
  - Static datasets with sampling
  - Streaming datasets with statistics

NOKIA BELL LABS

Prototype

Implementation and evaluation setup

C/C++ implementation

- Service code (<1K + JSON library)
- Enclave templates (~1K lines of code)
- Python client (~500 lines of code)
- 4 PCFs each with ~100-225 lines of custom code

Python implementation (with gramine libOS)

- Service code with <600 lines of code
- Python client with ~400 lines of code
- 4 PCFs each with ~45 lines of custom code

Evaluation

- **Sampling** for **static datasets** (up to 5M hashes)
- **Statistics** for **streaming data** (up to 200K integers/second)

Setup

- Azure Confidential Computing instance DC2sv3 (2vCPUs and 16GB RAM)
- With Microsoft Attestation Service

➢ Cost: ~0.16 Euro/hour

NOKIA BELL LABS

Common Operations

Across enclave types and dataset sizes (milliseconds), 20+ runs

- Setting up the enclave at the server
  - Initiating the enclave, obtaining the attestation report, getting a quote, …
- Verification of the quote at the client
  - Contacting the Attestation Service Provider with the quote

	Enclave setup	Quote verification
Sampling	~1800 ms	~175 ms
Nonrep. + Sampling	~1844 ms	~174 ms
Statistics	~424 ms	~183 ms
Sampling + Statistics	~443 ms	~184 ms

➢Independent of the dataset size

➢Depends on the enclave config
  ➢Sampling: Heap 256K pages, Stack 8K

➢(More or less) constant time

NOKIA BELL LABS

Sampling for Static Datasets

Client-side latencies (seconds), 20+ runs

	1M	2M	3M	4M	5M
Encryption of dataset	~6.2 s	~ 12.4 s	~18.7 s	~24.9 s	~31.2 s
Transmission & waiting for result	~47.1 s	~94.0 s	~141.3 s	~188.3 s	~235.5 s
Property computation (Server-side)	~0.03 s	~0.05 s	~0.08 s	~0.11 s	~0.14 s
Signature on result (Server-side)	~0.02 s	~0.04 s	~0.05 s	~0.07 s	~0.09 s
Signature verification	~0.07 s	~0.14 s	~0.20 s	~0.27 s	~0.35 s
Total	~53.4 s	~106.7 s	~160.3 s	~213.6 s	~267.1 s

1 hash = 65B
- SHA256
- 0x16 encoding
- "\n"

5M hashes ~= 310 MB

NOKIA BELL LABS

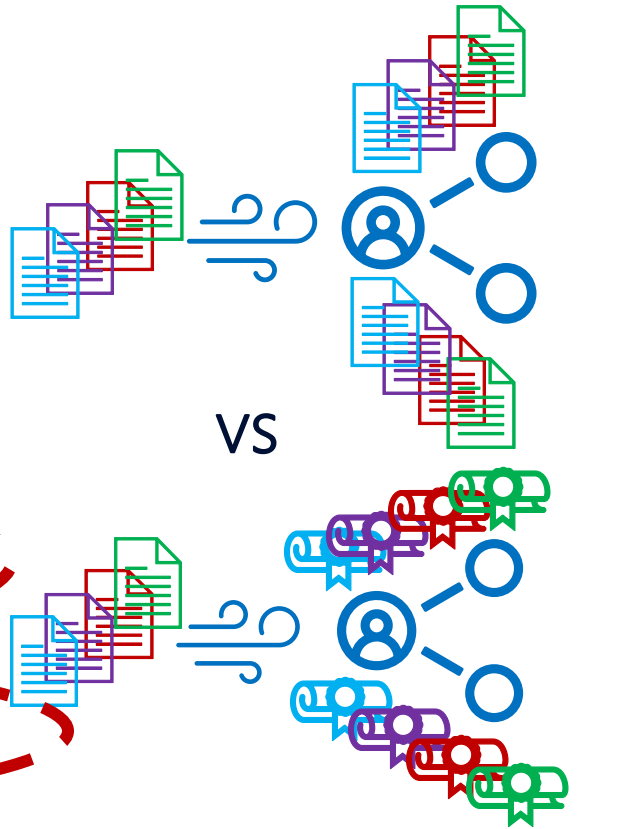
Statistics for Streaming Datasets

Client-side latency (milliseconds), 20+ runs, 100 batches

- Broker sends to subscribers only the batches with certain statistics to reduce bandwidth
- Subscribers want assurance the broker is filtering correctly

	50K/sec	100K/sec	150K/sec	200K/sec
Proof latency	~211 ms	~411 ms	~620 ms	~813 ms
Batch size	195KB	390KB	585KB	780KB
Proof size	<1KB	<1KB	<1KB	<1KB

- Near real-time latency
- Bandwidth reduction



NOKIA BELL LABS

Summary & Open Issues

Summary

PraaS: Verifiable proofs of dataset properties using Intel SGX

- Enable increased interaction among dataset owners and potential users without trust relations
- High performance and low latency for static and streaming datasets with low cost
- Easy customizability with enclave templates for C/C++ and python

- Source code available: https://github.com/Nokia-Bell-Labs/proof-as-a-service

Open issues

- Leakage of sensitive information if only interested in the property
  - Differential privacy
- Inspection of property computation logic may not catch covert channels that require collusion

NOKIA BELL LABS

NOKIA
BELL
LABS