Secure Key Management for Multi-Party Computation in MOZAIK

Enzo Marquet[*], Jerico Moeyersons[+], **Erik Pohle**[†], Michiel Van Kenhove[+], Aysajan Abidin[†], Bruno Volckaert[+]

erik.pohle@esat.kuleuven.be

July 3, 2023

[*] CiTiP, KU Leuven, Belgium
[+] IDLab-imec, Ghent University, Belgium
[†] imec-COSIC, KU Leuven, Belgium

2023 International Workshop on Privacy Engineering – IWPE'23

Introduction

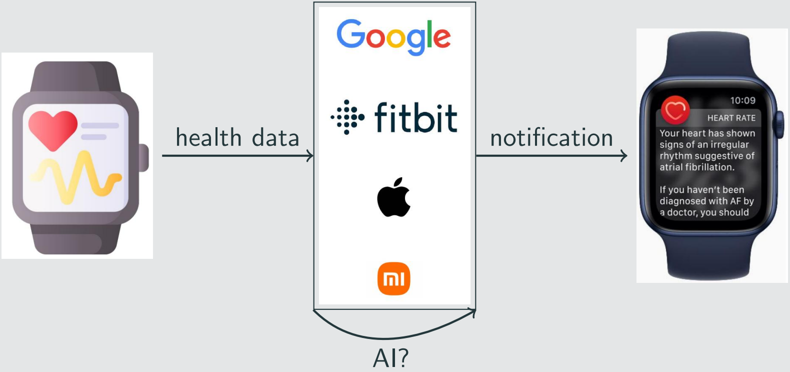
MOZAIK

- Platform for secure data sharing and processing
- Focus on user-control, privacy and GDPR compliance
- Data provided by IoT/embedded devices

MOZAIK

- Platform for secure data sharing and processing
- Focus on user-control, privacy and GDPR compliance
- Data provided by IoT/embedded devices

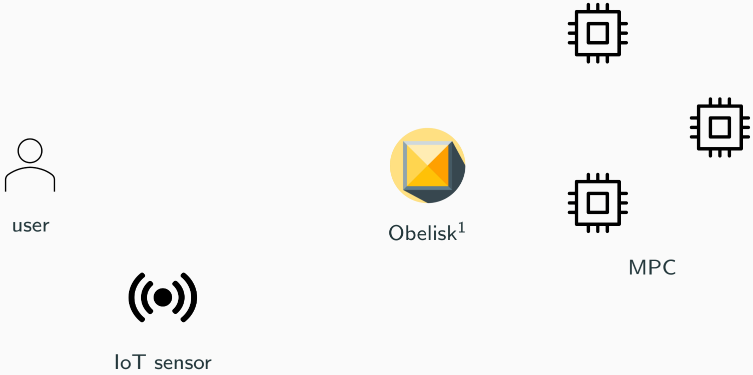
Use case: Heartbeat anomaly detection



Your heart has shown signs of an irregular rhythm suggestive of atrial fibrillation.

If you haven't been diagnosed with AF by a doctor, you should

Architecture



user

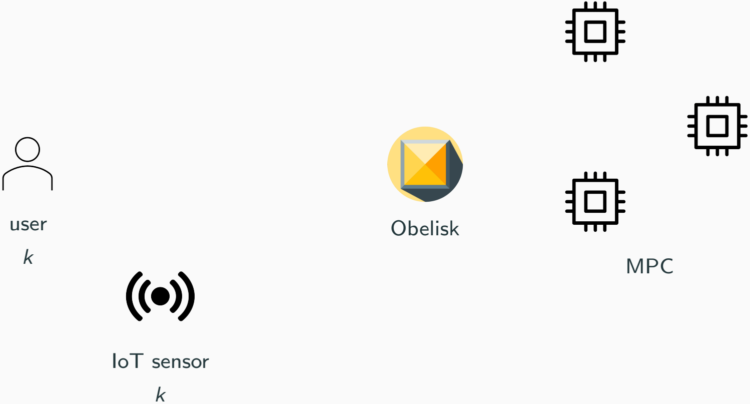
IoT sensor

Obelisk[1]

MPC

[1] https://obelisk.ilabt.imec.be/catalog/home

Architecture



user

k

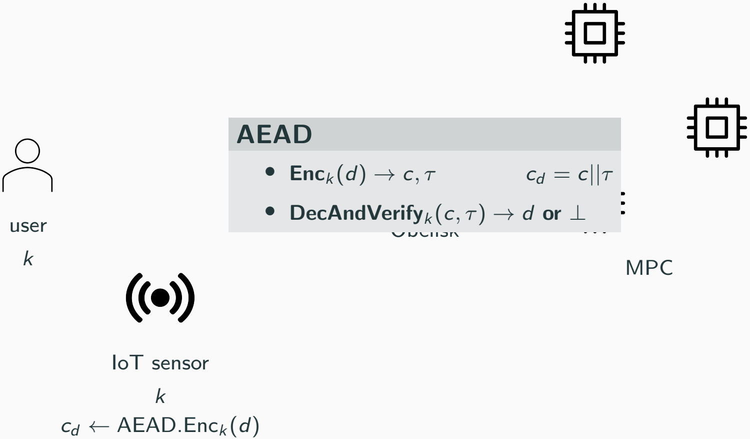
IoT sensor

k

Obelisk

MPC

Architecture

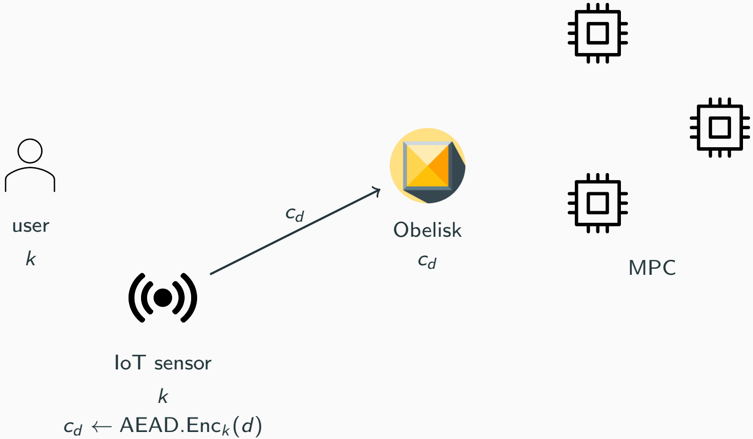


AEAD

- **Enc**$_k(d) \rightarrow c, \tau$ $\qquad c_d = c||\tau$
- **DecAndVerify**$_k(c, \tau) \rightarrow d$ **or** $\perp$

❶ Data is encrypted by IoT device

Architecture



user

k

IoT sensor

k

$c_d \leftarrow \text{AEAD.Enc}_k(d)$

c_d

Obelisk

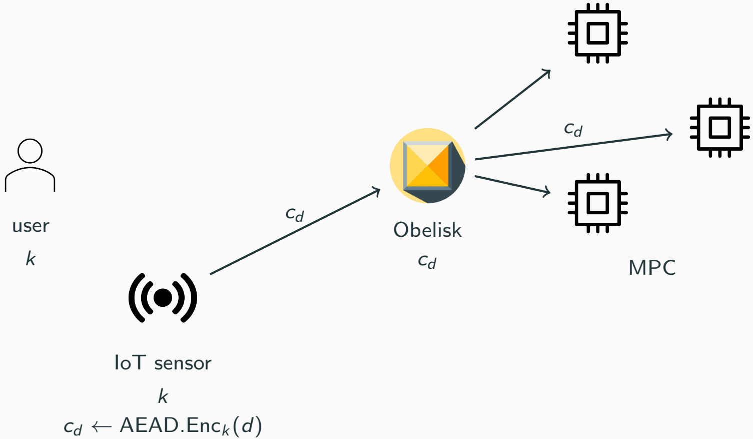
c_d

MPC

❶ Data is encrypted by IoT device

❷ Data is stored in central database layer

Architecture



user

k

IoT sensor

k

$c_d \leftarrow \text{AEAD.Enc}_k(d)$

c_d

Obelisk

c_d

c_d

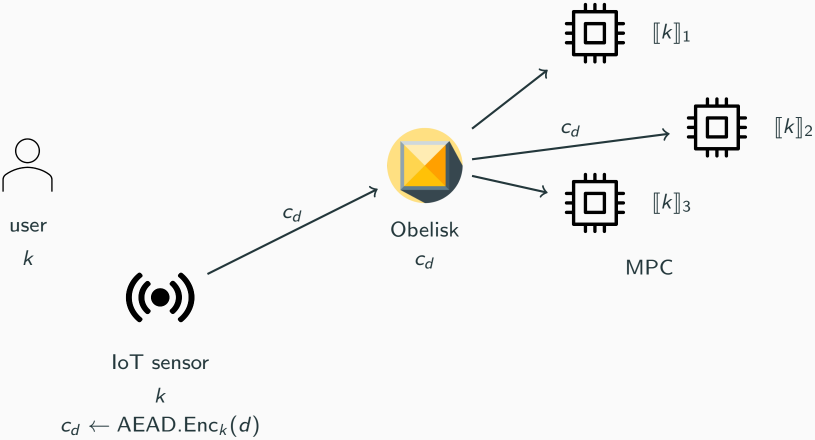
MPC

❶ Data is encrypted by IoT device

❷ Data is stored in central database layer

❸ Data is fetched by MPC parties

Architecture



1. Data is encrypted by IoT device
2. Data is stored in central database layer
3. Data is fetched by MPC parties
4. MPC parties have secret share $[\![k]\!]$

Secret-Sharing & MPC

Secret Sharing

- **Share**$(x) \to [\![x]\!]_1, \ldots, [\![x]\!]_n$
- **Recon**$(\{[\![x]\!]_j\}_{j \in A}) \to x$

set A access structure

Secret-Sharing & MPC

Secret Sharing

- **Share**$(x) \to [\![x]\!]_1, \ldots, [\![x]\!]_n$
- **Recon**$(\{[\![x]\!]_j\}_{j \in A}) \to x$ set A access structure

Examples (where $x \in \mathbb{F}$) finite field $\mathbb{F}$

- Shamir: $[\![x]\!]_i = p(i)$ with $p(0) = x$ p polynomial of degree t

  at least $t + 1$ shares are required to reconstruct

Secret-Sharing & MPC

Secret Sharing

- **Share**$(x) \to [\![x]\!]_1, \ldots, [\![x]\!]_n$
- **Recon**$(\{[\![x]\!]_j\}_{j \in A}) \to x$

set A access structure

Examples (where $x \in \mathbb{F}$)

finite field $\mathbb{F}$

- Shamir: $[\![x]\!]_i = p(i)$ with $p(0) = x$
  at least $t + 1$ shares are required to reconstruct

p polynomial of degree t

- Additive: $[\![x]\!]_1 + \cdots + [\![x]\!]_n = x$

Secret-Sharing & MPC

Secret Sharing

- **Share**$(x) \to [\![x]\!]_1, \ldots, [\![x]\!]_n$
- **Recon**$(\{[\![x]\!]_j\}_{j \in A}) \to x$

set A access structure

Examples (where $x \in \mathbb{F}$)

finite field $\mathbb{F}$

- Shamir: $[\![x]\!]_i = p(i)$ with $p(0) = x$

  at least $t + 1$ shares are required to reconstruct

p polynomial of degree t

- Additive: $[\![x]\!]_1 + \cdots + [\![x]\!]_n = x$

Secure multi-party computation

- Each party P_i has private input x_i
- Public input z

Secret-Sharing & MPC

Secret Sharing

- **Share**$(x) \to [\![x]\!]_1, \dots, [\![x]\!]_n$
- **Recon**$(\{[\![x]\!]_j\}_{j \in A}) \to x$ set A access structure

Examples (where $x \in \mathbb{F}$) finite field $\mathbb{F}$

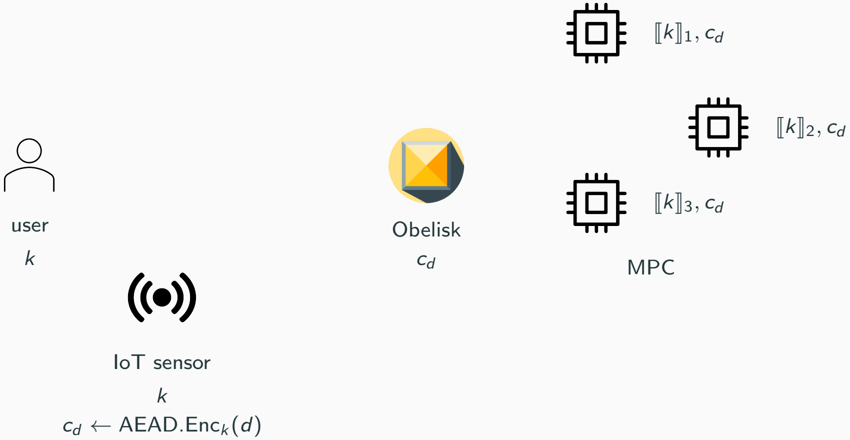
- Shamir: $[\![x]\!]_i = p(i)$ with $p(0) = x$ p polynomial of degree t
  at least $t + 1$ shares are required to reconstruct
- Additive: $[\![x]\!]_1 + \cdots + [\![x]\!]_n = x$

Secure multi-party computation

- Each party P_i has private input x_i
- Public input z
- Compute function $y \leftarrow f(x_1, \dots, x_n, z)$ s.t. no party learns the other inputs

$\Rightarrow$ Distributed protocol

Architecture (cont.)



user

k

IoT sensor

k

$c_d \leftarrow \text{AEAD.Enc}_k(d)$

Obelisk

c_d

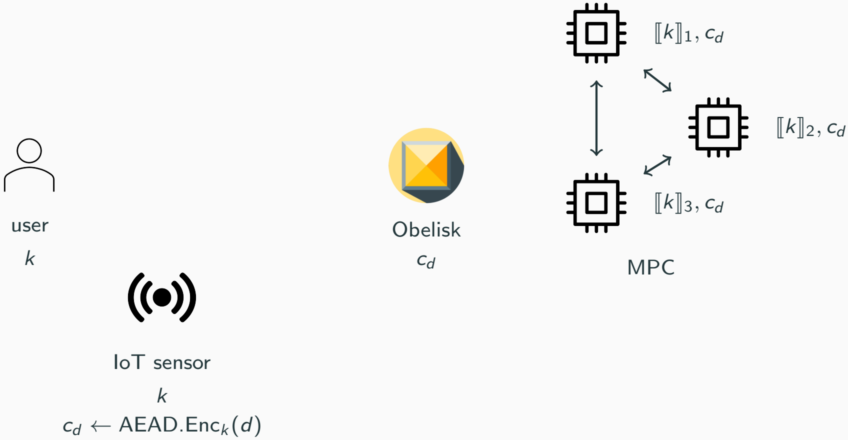
$[\![k]\!]_1, c_d$

$[\![k]\!]_2, c_d$

$[\![k]\!]_3, c_d$

MPC

Architecture (cont.)



user

k

IoT sensor

k

$c_d \leftarrow \mathsf{AEAD.Enc}_k(d)$

Obelisk

c_d

$[\![k]\!]_1, c_d$

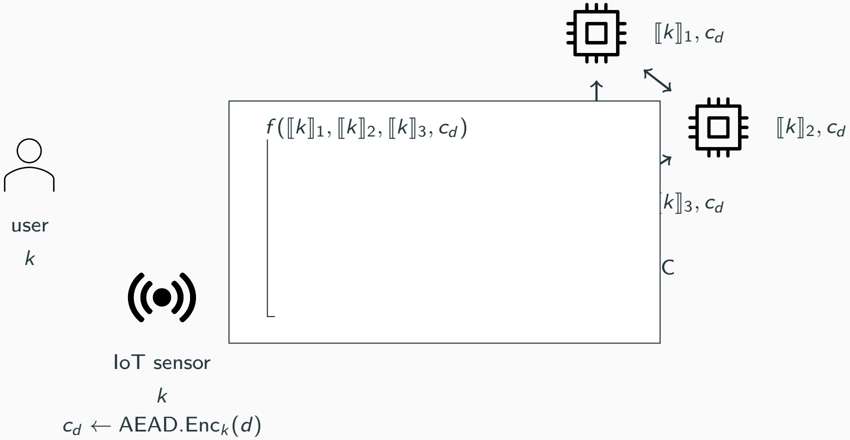
$[\![k]\!]_2, c_d$

$[\![k]\!]_3, c_d$

MPC

❺ Data is processed using MPC

Architecture (cont.)



$[\![k]\!]_1, c_d$

$[\![k]\!]_2, c_d$

$k]\!]_3, c_d$

C

$f([\![k]\!]_1, [\![k]\!]_2, [\![k]\!]_3, c_d)$

user

k

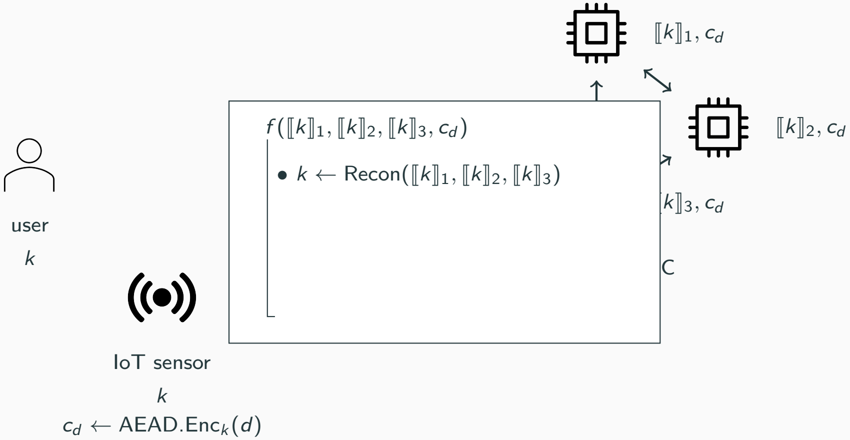
IoT sensor

k

$c_d \leftarrow \text{AEAD.Enc}_k(d)$

❺ Data is processed using MPC

Architecture (cont.)



$\llbracket k \rrbracket_1, c_d$

$\llbracket k \rrbracket_2, c_d$

$k \rrbracket_3, c_d$

C

$f(\llbracket k \rrbracket_1, \llbracket k \rrbracket_2, \llbracket k \rrbracket_3, c_d)$

- $k \leftarrow \text{Recon}(\llbracket k \rrbracket_1, \llbracket k \rrbracket_2, \llbracket k \rrbracket_3)$

user

k

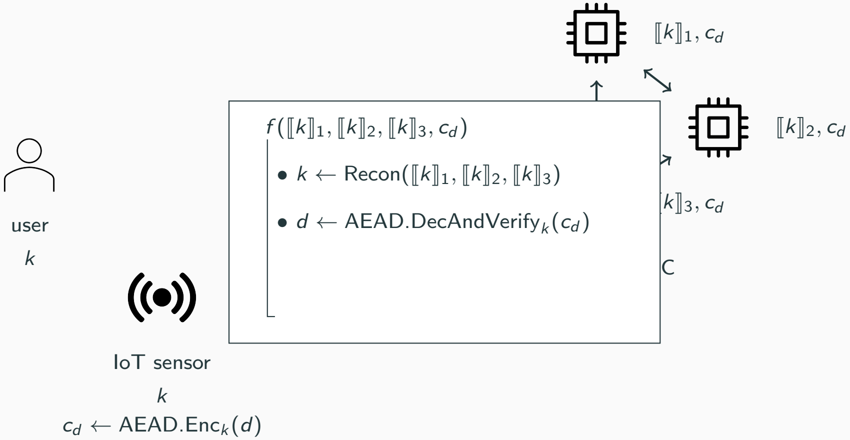
IoT sensor

k

$c_d \leftarrow \text{AEAD.Enc}_k(d)$

❺ Data is processed using MPC

Architecture (cont.)

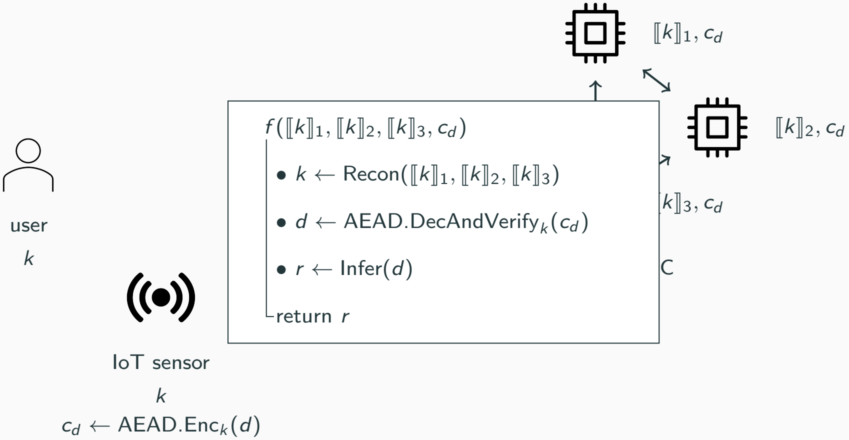


$f(⟦k⟧_1, ⟦k⟧_2, ⟦k⟧_3, c_d)$

- $k \leftarrow \mathsf{Recon}(⟦k⟧_1, ⟦k⟧_2, ⟦k⟧_3)$
- $d \leftarrow \mathsf{AEAD.DecAndVerify}_k(c_d)$

❺ Data is processed using MPC

Architecture (cont.)

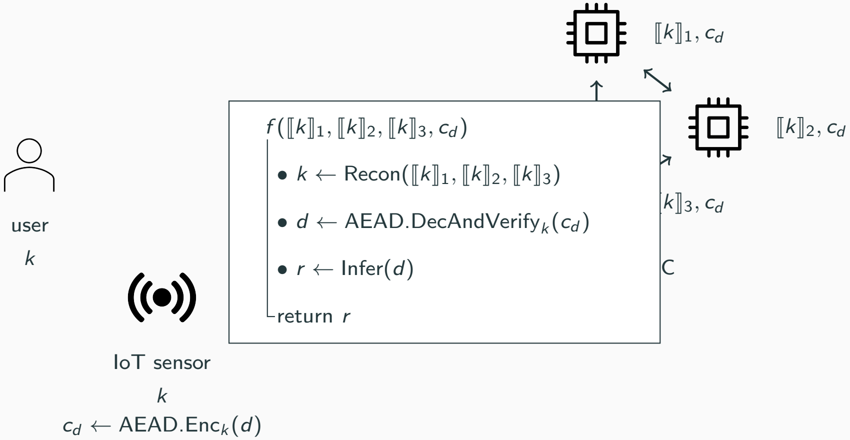


$\llbracket k \rrbracket_1, c_d$
$\llbracket k \rrbracket_2, c_d$
$k \rrbracket_3, c_d$
C
user
k
IoT sensor
k
$c_d \leftarrow \text{AEAD.Enc}_k(d)$

```
$f(\llbracket k \rrbracket_1, \llbracket k \rrbracket_2, \llbracket k \rrbracket_3, c_d)$
  • $k \leftarrow \text{Recon}(\llbracket k \rrbracket_1, \llbracket k \rrbracket_2, \llbracket k \rrbracket_3)$
  • $d \leftarrow \text{AEAD.DecAndVerify}_k(c_d)$
  • $r \leftarrow \text{Infer}(d)$
  return $r$
```

❺ Data is processed using MPC

Architecture (cont.)



```
$f([\![k]\!]_1, [\![k]\!]_2, [\![k]\!]_3, c_d)$
    • $k \leftarrow \text{Recon}([\![k]\!]_1, [\![k]\!]_2, [\![k]\!]_3)$
    • $d \leftarrow \text{AEAD.DecAndVerify}_k(c_d)$
    • $r \leftarrow \text{Infer}(d)$
    return $r$
```

❺ Data is processed using MPC

⇒ Central key: user's symmetric key k and shares $[\![k]\!]$

Key Management and Distribution of $[\![k]\!]$

Key Management and Distribution of $[\![k]\!]$

Goal

- Securely distribute $[\![k]\!]_i$ to MPC party P_i
- Securely recover the result r

Key Management and Distribution of $[\![k]\!]$

Goal

- Securely distribute $[\![k]\!]_i$ to MPC party P_i
- Securely recover the result r

Tools/Assumptions

- IoT device managed/controlled by user
- PKI: user & MPC parties have public keys

Key Management and Distribution of $[\![k]\!]$

Goal

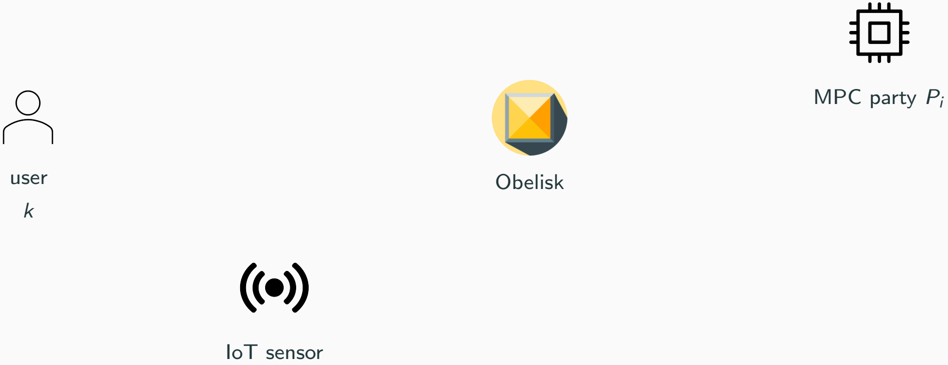
- Securely distribute $[\![k]\!]_i$ to MPC party P_i
- Securely recover the result r

Tools/Assumptions

- IoT device managed/controlled by user
- PKI: user & MPC parties have public keys
- Adversary controls
  - Some users
  - The database
  - Up to t MPC parties

MPC/secret-sharing threshold t

Key Generation



user

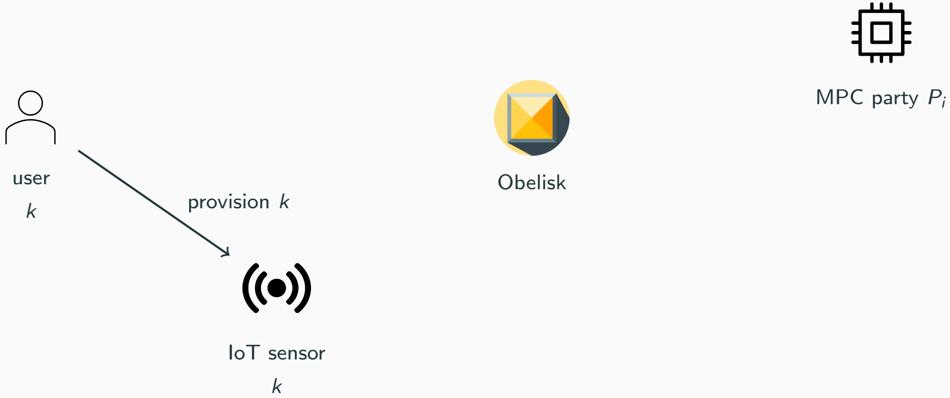
k

Obelisk

MPC party P_i

IoT sensor

Key Generation



user

k

provision k

IoT sensor

k

Obelisk

MPC party P_i

Data collection (loop)

MPC party P_i

user

k

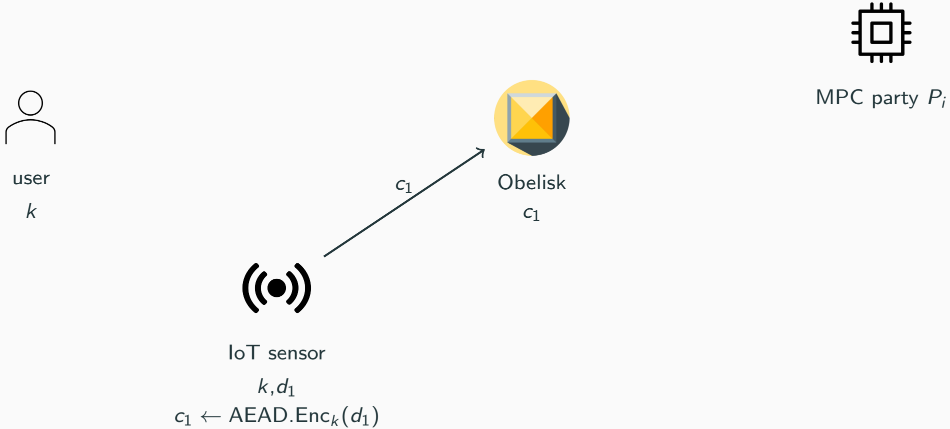
Obelisk

IoT sensor

k, d_1

$c_1 \leftarrow \mathrm{AEAD.Enc}_k(d_1)$

Data collection (loop)



user

k

c_1

Obelisk

c_1

MPC party P_i

IoT sensor

k, d_1

$c_1 \leftarrow \mathsf{AEAD.Enc}_k(d_1)$

Data collection (loop)

MPC party P_i

user

k

Obelisk

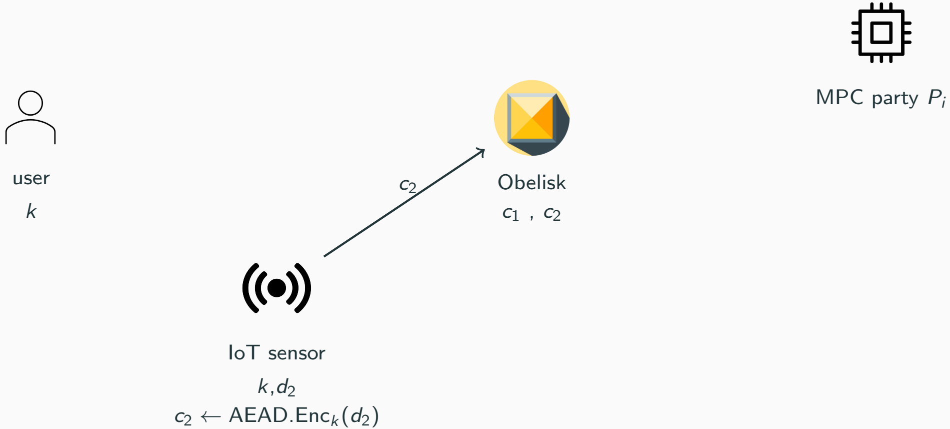
c_1

IoT sensor

k, d_2

$c_2 \leftarrow \text{AEAD.Enc}_k(d_2)$

Data collection (loop)



user

k

c_2

Obelisk

c_1 , c_2

MPC party P_i

IoT sensor

k, d_2

$c_2 \leftarrow \text{AEAD.Enc}_k(d_2)$

Compute setup

- user selects n MPC parties and secret sharing scheme

MPC party P_i

sk_i, pk_i

user

k

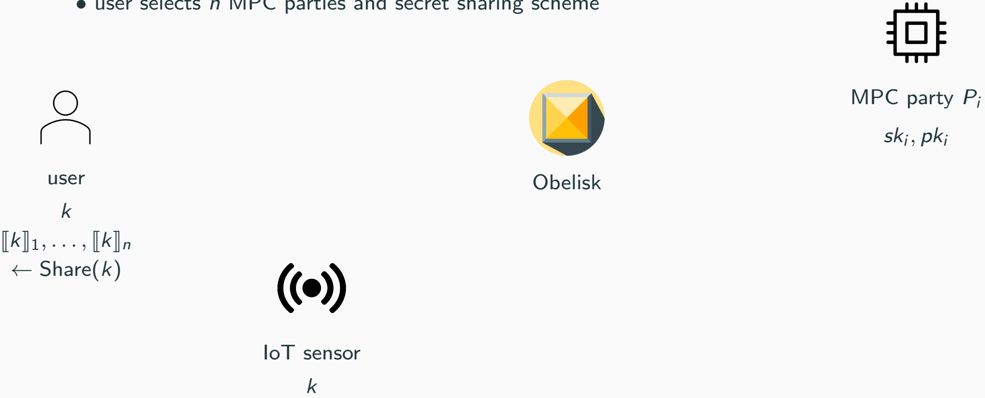
Obelisk

IoT sensor

k

Compute setup

- user selects n MPC parties and secret sharing scheme



user

k

$[\![k]\!]_1, \ldots, [\![k]\!]_n \leftarrow \text{Share}(k)$

Obelisk

MPC party P_i

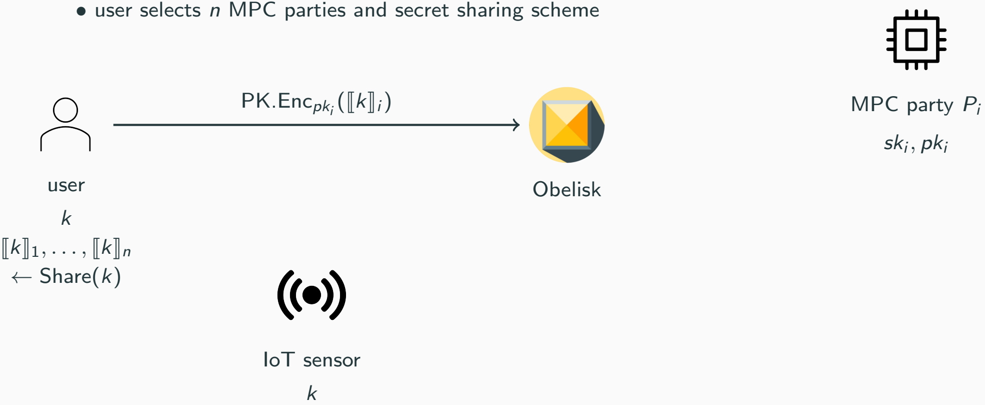
sk_i, pk_i

IoT sensor

k

Compute setup

- user selects n MPC parties and secret sharing scheme



$\mathsf{PK.Enc}_{pk_i}(\llbracket k \rrbracket_i)$

user

k

$\llbracket k \rrbracket_1, \ldots, \llbracket k \rrbracket_n$
$\leftarrow \mathsf{Share}(k)$

Obelisk

MPC party P_i

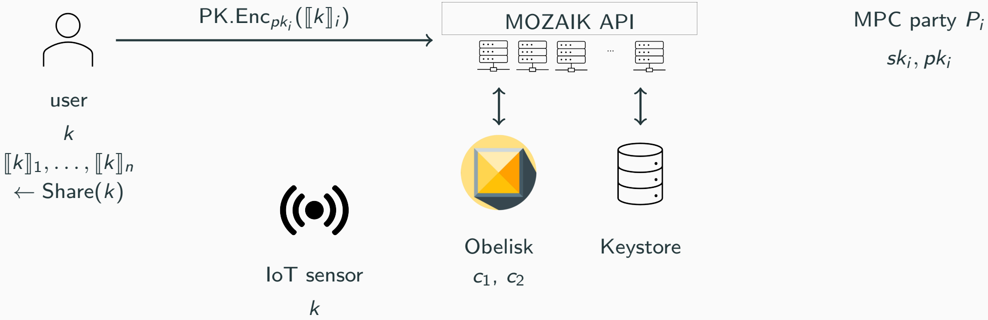
sk_i, pk_i

IoT sensor

k

Compute setup

- user selects n MPC parties and secret sharing scheme
- defense in-depth: separate databases in secure containers



$\mathsf{PK.Enc}_{pk_i}([\![k]\!]_i)$

MOZAIK API

user

k

$[\![k]\!]_1, \ldots, [\![k]\!]_n$
$\leftarrow \mathsf{Share}(k)$

IoT sensor

k

Obelisk

c_1, c_2

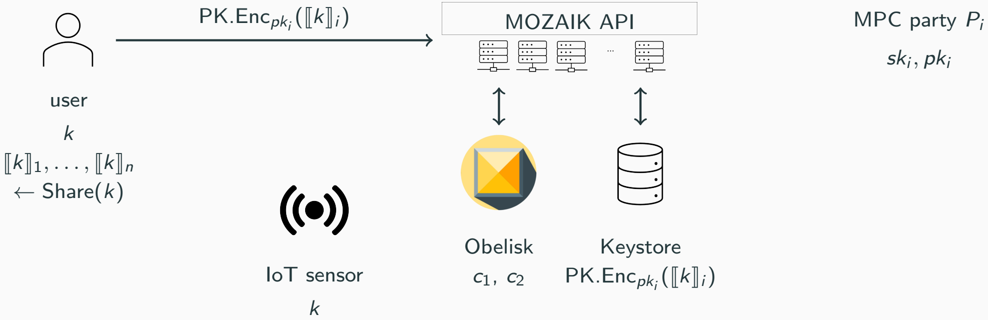
Keystore

MPC party P_i

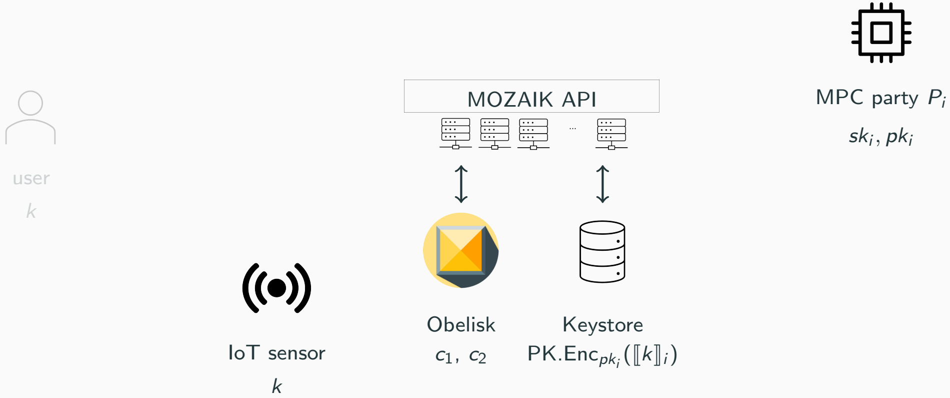
sk_i, pk_i

Compute setup

- user selects n MPC parties and secret sharing scheme
- defense in-depth: separate databases in secure containers


$\text{PK.Enc}_{pk_i}(\llbracket k \rrbracket_i)$
MOZAIK API
user
k
$\llbracket k \rrbracket_1, \ldots, \llbracket k \rrbracket_n$
$\leftarrow \text{Share}(k)$
IoT sensor
k
Obelisk
c_1, c_2
Keystore
$\text{PK.Enc}_{pk_i}(\llbracket k \rrbracket_i)$
MPC party P_i
sk_i, pk_i

Compute



user

k

IoT sensor

k

MOZAIK API

Obelisk

c_1, c_2

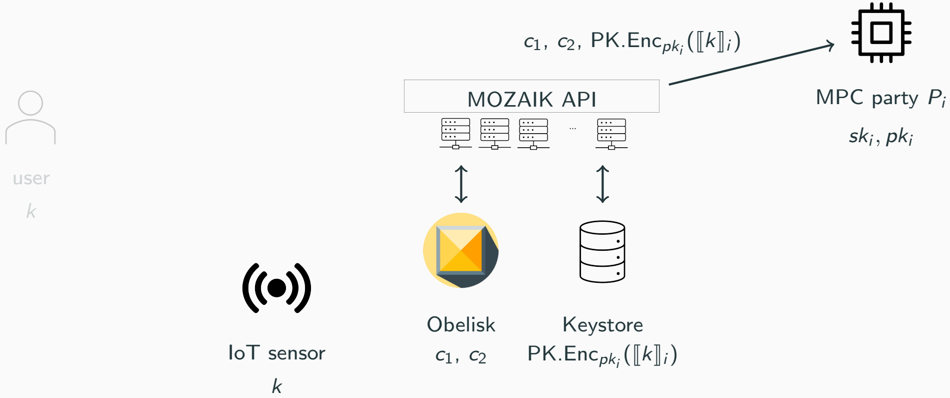
Keystore

$\mathrm{PK.Enc}_{pk_i}(\llbracket k \rrbracket_i)$

MPC party P_i

sk_i, pk_i

Compute



c_1, c_2, $\mathsf{PK.Enc}_{pk_i}(\llbracket k \rrbracket_i)$

MPC party P_i

sk_i, pk_i

user

k

MOZAIK API

IoT sensor

k

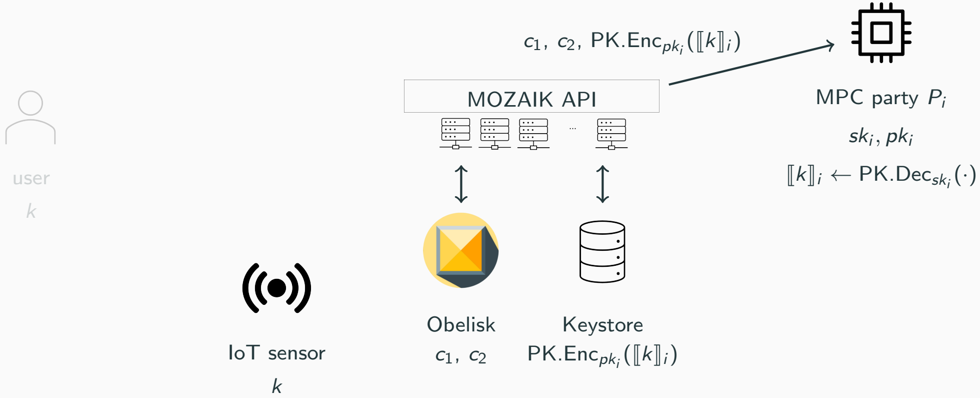
Obelisk

c_1, c_2

Keystore

$\mathsf{PK.Enc}_{pk_i}(\llbracket k \rrbracket_i)$

Compute



$c_1,\ c_2,\ \mathsf{PK.Enc}_{pk_i}(\llbracket k \rrbracket_i)$

MOZAIK API

user

k

IoT sensor

k

Obelisk

$c_1,\ c_2$

Keystore

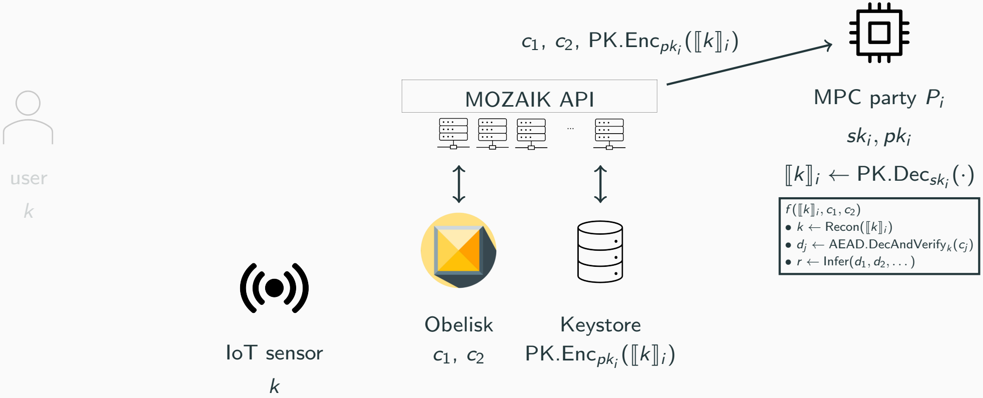
$\mathsf{PK.Enc}_{pk_i}(\llbracket k \rrbracket_i)$

MPC party P_i

sk_i, pk_i

$\llbracket k \rrbracket_i \leftarrow \mathsf{PK.Dec}_{sk_i}(\cdot)$

Compute



$c_1,\ c_2,\ \mathsf{PK.Enc}_{pk_i}(\llbracket k \rrbracket_i)$

MOZAIK API

MPC party P_i

sk_i, pk_i

$\llbracket k \rrbracket_i \leftarrow \mathsf{PK.Dec}_{sk_i}(\cdot)$

$f(\llbracket k \rrbracket_i, c_1, c_2)$

- $k \leftarrow \mathsf{Recon}(\llbracket k \rrbracket_i)$
- $d_j \leftarrow \mathsf{AEAD.DecAndVerify}_k(c_j)$
- $r \leftarrow \mathsf{Infer}(d_1, d_2, \dots)$

user

k

IoT sensor

k

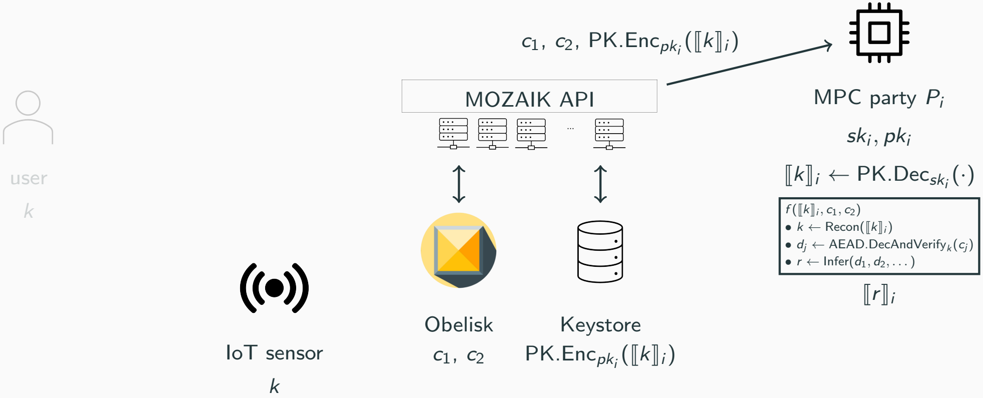
Obelisk

$c_1,\ c_2$

Keystore

$\mathsf{PK.Enc}_{pk_i}(\llbracket k \rrbracket_i)$

Compute



user

k

IoT sensor

k

MOZAIK API

Obelisk

$c_1,\ c_2$

Keystore

$\mathsf{PK.Enc}_{pk_i}(\llbracket k \rrbracket_i)$

$c_1,\ c_2,\ \mathsf{PK.Enc}_{pk_i}(\llbracket k \rrbracket_i)$

MPC party P_i

sk_i, pk_i

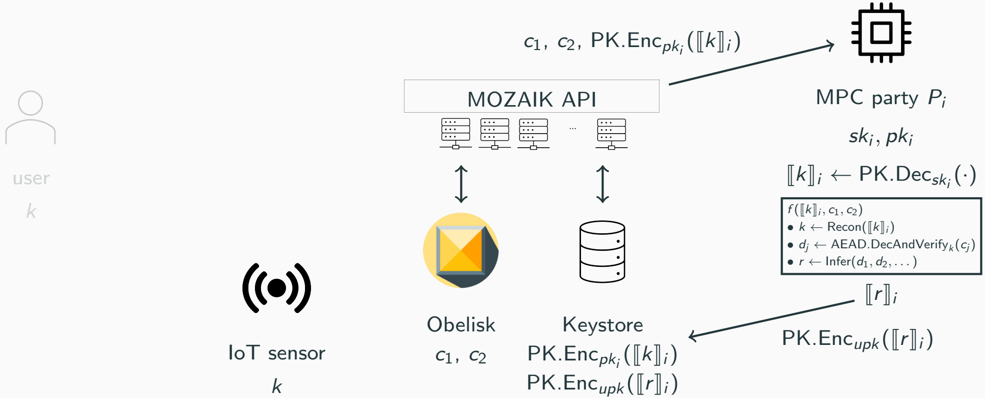
$\llbracket k \rrbracket_i \leftarrow \mathsf{PK.Dec}_{sk_i}(\cdot)$

$f(\llbracket k \rrbracket_i, c_1, c_2)$

- $k \leftarrow \mathsf{Recon}(\llbracket k \rrbracket_i)$
- $d_j \leftarrow \mathsf{AEAD.DecAndVerify}_k(c_j)$
- $r \leftarrow \mathsf{Infer}(d_1, d_2, \ldots)$

$\llbracket r \rrbracket_i$

Compute



user

k

IoT sensor

k

MOZAIK API

Obelisk

$c_1, \; c_2$

Keystore

$\mathsf{PK.Enc}_{pk_i}(\llbracket k \rrbracket_i)$

$\mathsf{PK.Enc}_{upk}(\llbracket r \rrbracket_i)$

$c_1, \; c_2, \; \mathsf{PK.Enc}_{pk_i}(\llbracket k \rrbracket_i)$

MPC party P_i

sk_i, pk_i

$\llbracket k \rrbracket_i \leftarrow \mathsf{PK.Dec}_{sk_i}(\cdot)$

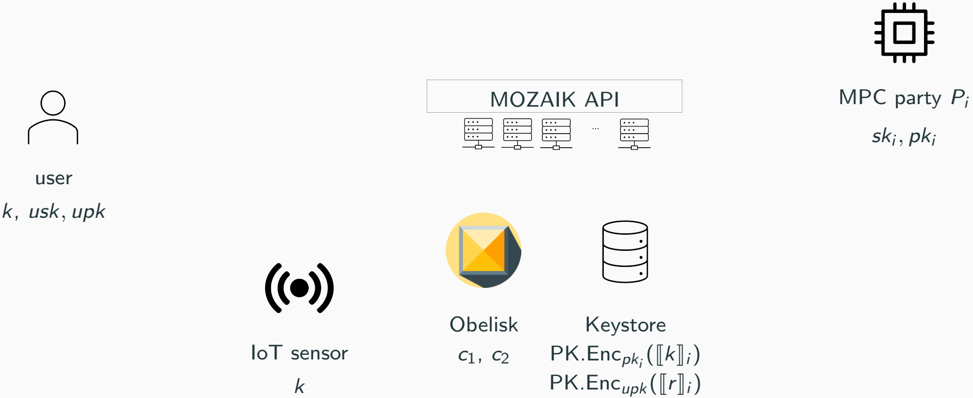
$f(\llbracket k \rrbracket_i, c_1, c_2)$

- $k \leftarrow \mathsf{Recon}(\llbracket k \rrbracket_i)$
- $d_j \leftarrow \mathsf{AEAD.DecAndVerify}_k(c_j)$
- $r \leftarrow \mathsf{Infer}(d_1, d_2, \ldots)$

$\llbracket r \rrbracket_i$

$\mathsf{PK.Enc}_{upk}(\llbracket r \rrbracket_i)$

Fetch results



user

$k, \; usk, upk$

IoT sensor

k

MOZAIK API

$\cdots$

Obelisk

$c_1, \; c_2$

Keystore

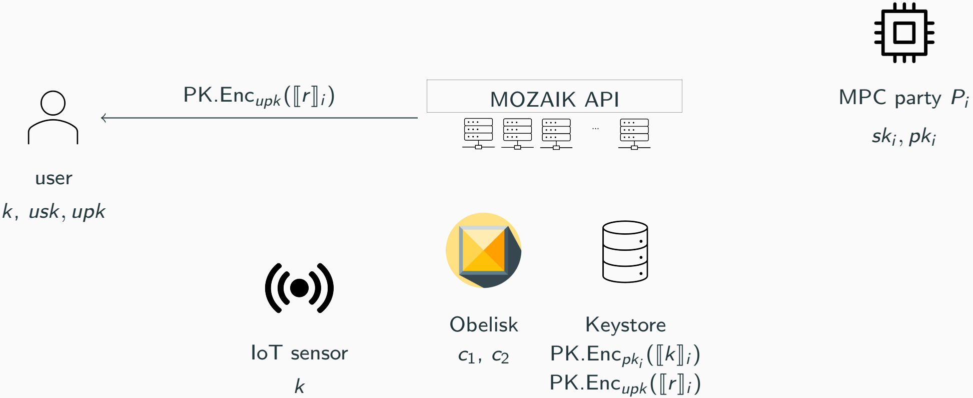
$\mathrm{PK.Enc}_{pk_i}(\llbracket k \rrbracket_i)$

$\mathrm{PK.Enc}_{upk}(\llbracket r \rrbracket_i)$

MPC party P_i

sk_i, pk_i

Fetch results



$\mathsf{PK.Enc}_{upk}(\llbracket r \rrbracket_i)$

MOZAIK API

MPC party P_i

sk_i, pk_i

user

$k,\ usk, upk$

IoT sensor

k

Obelisk

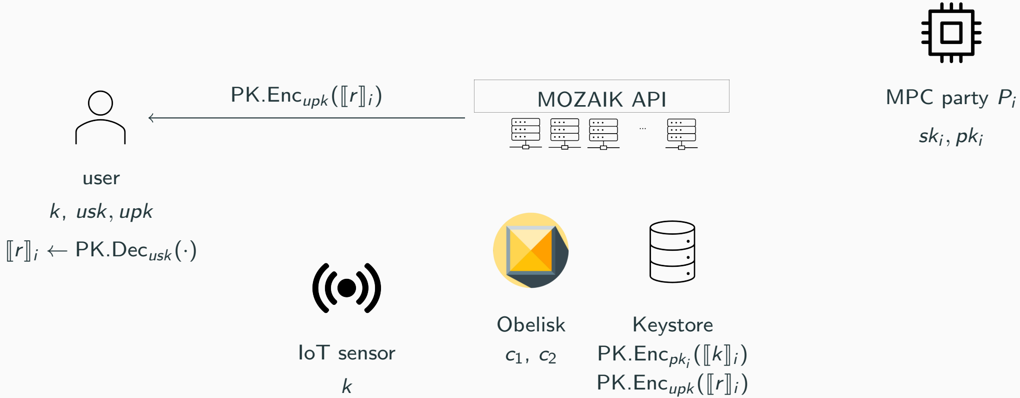
$c_1,\ c_2$

Keystore

$\mathsf{PK.Enc}_{pk_i}(\llbracket k \rrbracket_i)$

$\mathsf{PK.Enc}_{upk}(\llbracket r \rrbracket_i)$

Fetch results



$\mathsf{PK.Enc}_{upk}(\llbracket r \rrbracket_i)$

MOZAIK API

MPC party P_i

sk_i, pk_i

user

$k,\ usk, upk$

$\llbracket r \rrbracket_i \leftarrow \mathsf{PK.Dec}_{usk}(\cdot)$

IoT sensor

k

Obelisk

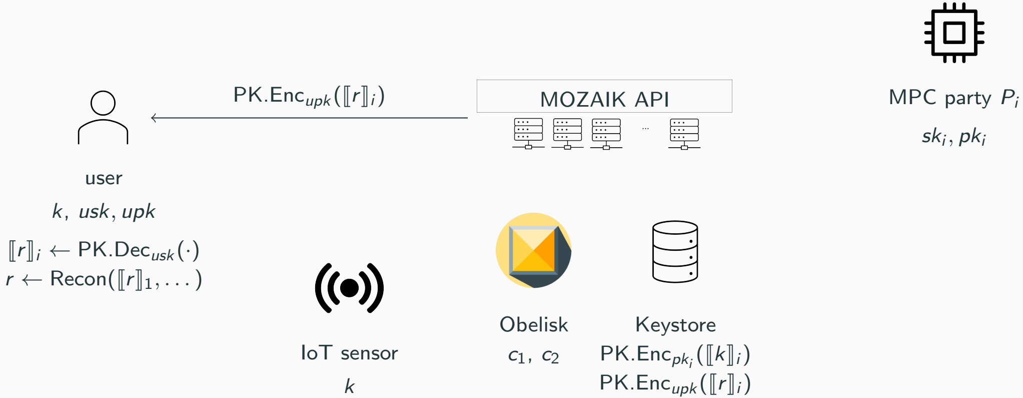
$c_1,\ c_2$

Keystore

$\mathsf{PK.Enc}_{pk_i}(\llbracket k \rrbracket_i)$

$\mathsf{PK.Enc}_{upk}(\llbracket r \rrbracket_i)$

Fetch results



$\mathsf{PK.Enc}_{upk}(\llbracket r \rrbracket_i)$

MOZAIK API

MPC party P_i

sk_i, pk_i

user

$k,\ usk, upk$

$\llbracket r \rrbracket_i \leftarrow \mathsf{PK.Dec}_{usk}(\cdot)$

$r \leftarrow \mathsf{Recon}(\llbracket r \rrbracket_1, \dots)$

IoT sensor

k

Obelisk

$c_1,\ c_2$

Keystore

$\mathsf{PK.Enc}_{pk_i}(\llbracket k \rrbracket_i)$

$\mathsf{PK.Enc}_{upk}(\llbracket r \rrbracket_i)$

Conclusion

Secure

- User in control of k and choice of MPC parties

Conclusion

Secure

- User in control of k and choice of MPC parties
- Shares $[\![k]\!]$ and $[\![r]\!]$ end-to-end encrypted

Conclusion

Secure

- User in control of k and choice of MPC parties
- Shares $[\![k]\!]$ and $[\![r]\!]$ end-to-end encrypted
- Keystore/database cannot reconstruct k or obtain d

Conclusion

Secure

- User in control of k and choice of MPC parties
- Shares $[\![k]\!]$ and $[\![r]\!]$ end-to-end encrypted
- Keystore/database cannot reconstruct k or obtain d
- Key-related data remains at third-parties only during use

Conclusion

Secure

- User in control of k and choice of MPC parties
- Shares $[\![k]\!]$ and $[\![r]\!]$ end-to-end encrypted
- Keystore/database cannot reconstruct k or obtain d
- Key-related data remains at third-parties only during use

Flexible

- Immediate data collection
- User can be offline during processing

Backup

Instantiations

AEAD

- IoT-friendly: Ascon, SKINNY, GIFT-COFB
- MPC-friendly: CTR-tHtMAC-MiMC
- Standards: AES-GCM(-SIV)

PK

- Any CCA-secure scheme, e.g., CRYSTALS-KYBER