System Designs for End-to-End Privacy
Functionality, Performance, & Accessibility

Anwar Hithnawi

Based on joint work with: Lukas Burkhalter, Alexander Viand, Nicolas Kuechler, and Hidde Lycklama.
Data Driven World

- 2010: Data volume
- 2011: Data volume
- 2012: Data volume
- 2013: Data volume
- 2014: Data volume
- 2015: Data volume
- 2016: Data volume
- 2017: Data volume
- 2018: Data volume
- 2019: Data volume
- 2020: Data volume
- 2021: Data volume
- 2022: Data volume
- 2023: Data volume
- 2024: Data volume
- 2025: Data volume

- 2010: 181 ZB
- 2011: 79 ZB
- 2012: Data volume
- 2013: Data volume
- 2014: Data volume
- 2015: Data volume
- 2016: Data volume
- 2017: Data volume
- 2018: Data volume
- 2019: Data volume
- 2020: Data volume
- 2021: Data volume
- 2022: Data volume
- 2023: Data volume
- 2024: Data volume
- 2025: Data volume

Data volume from 2010 to 2025, with significant growth between 2018 and 2021, reaching a peak of 181 ZB in 2025.
Sensitive Data

Finance
Health
Government
Personal
Morgan Stanley settles personal data breach lawsuit for $60 million

Data Breaches Keep Happening. So Why Don’t You Do Something?

Capital One Data Breach Compromises Data of Over 100 Million

All 3 Billion Yahoo Accounts Were Affected by 2013 Attack

The Government Uses ‘Near Perfect Surveillance’ Data on Americans
Congressional hearings are urgently needed to address location tracking.
By THE EDITORIAL BOARD

Grindr and OkCupid Spread Personal Details, Study Says
Norwegian research raises questions about whether certain of sharing of information violate data privacy laws in Europe and the United States.

You Should Be Freaking Out About Privacy
Nothing to hide, nothing to fear? Think again.

Technology
Data broker shared billions of location records with District during pandemic
The bulk sales of location data have fueled a debate over public health and privacy
Data Breaches

Data Misuses
~ 1.245 Billion

The number of data records stolen in 2020

143,000,000

57,000,000

330,000,000

533,000,000

Equifax

Uber

Twitter

Facebook
Where the sensitive information is concentrated, that is where the spies will go. This is just a fact of life.

former NSA official Ken Silva.
Where the sensitive information is concentrated, that is where the spies will go. This is just a fact of life. “

former NSA official Ken Silva.
End-to-End Encrypted Systems
End-to-End Encrypted Systems

- Graphical representation of secure communication flow
- Encrypted data transfer
- Secure cloud storage

isc7UUS+/IouYEaELEAT
f4xAGfqYs3k
1G0jPok8nn
More Applications?
End-to-End Security

Data in transit
Secure communication
End-to-End Security

data in transit
secure communication
Data in transit
secure communication

Data at rest
secure storage

End-to-End Security
End-to-End Security

- **data at rest**
  - secure storage
- **data in transit**
  - secure communication
- **data in use**
  - secure computation
Modern Cryptography

End-to-End Security

- data at rest: secure storage
- data in transit: secure communication
- data in use: secure computation
Modern Cryptography

- data at rest
  - secure storage

Conventional Crypto
- Encryption & Digital Signature

End-to-End Security
- data in transit
  - secure communication
- data in use
  - secure computation
Modern Cryptography

Ubiquitous Adoption

Conventional Crypto

Encryption & Digital Signature

End-to-End Security

data at rest
secure storage

data in transit
secure communication

data in use
secure computation
Modern Cryptography

Ubiquitous Adoption

Conventional Crypto
- Encryption & Digital Signature

End-to-End Security
- data in transit: secure communication
- data at rest: secure storage
- data in use: secure computation

Advanced Crypto
- Homomorphic Encryption
- Secure Multi-party Computation
- Zero Knowledge Proofs

Modern Cryptography
Modern Cryptography

Ubiquitous Adoption
Conventional Crypto
Encryption & Digital Signature

End-to-End Security

data at rest
secure storage

data in transit
secure communication

data in use
secure computation

Advanced Crypto

• Homomorphic Encryption
• Secure Multi-party Computation
• Zero Knowledge Proofs

Just Starting
Fully Homomorphic Encryption

Enables computation on encrypted data

\[ f(x) \leftarrow \text{Decrypt} \]

\[ X \rightarrow \text{Encrypt} \]

\[ \text{KeyGen} \]

\[ k_p \uparrow \]

\[ k_s \downarrow \]
Fully Homomorphic Encryption

Enables **computation** on encrypted data

Delegate the **processing** of data without giving away **access** to it
40 Years of FHE History

- RSA '78
- GM '82
- El-Gamal '85
- Benaloh '94
- Paillier '99
- SYY '00
- BGN '05
- Benaloh '94
- IP '07
- Gentry '09

ACHES & Answer to 30 min

RAD '78

1980

1990

2000

2010

2020
40 Years of FHE History
40 Years of FHE History

- RSA '78
- GM '82
- El-Gamal '85
- Benaloh '94
- Paillier '99
- SYY '00
- BGN '05
- Gentry '09
- SYY '00
- IP '07
- BGN '05
- GSW '13
- TFHE '16
- GSW '13
- BFV '12
- GSW '13
- BGV '11
- ZAMA '20

FHE Hardware Accelerators

- RAD '78
- 1980
- 1990
- 2000
- 2010
- 2010
- 2020

- 0.01 s
- 1980
- 1990
- 2000
- 2010
- 2020
FHE is not yet practical for many applications **but** will soon be practical for a wider set of applications…
FHE is not yet practical for many applications but will soon be practical for a wider set of applications…

Performance gap of modern applications

Facilitate FHE use in real world deployment
Building Encrypted Data Processing Systems

DBMS
- CryptDB
- Blind Seer
- Arx
- Conclave
- ...
Building Encrypted Data Processing Systems

**DBMS**
- CryptDB
- Blind Seer
- Arx
- ...

**Advanced Cryptography**

**Modern Applications**

**Internet of Things**
- Talos
- Pilatus
- Kryptein
- ...

**Analytics**

**Machine Learning**

**Streaming**

**IoT**

**Advanced Cryptography**

**Modern Applications**
Building Encrypted Data Processing Systems

DBMS
- CryptDB
- Blind Seer
- Arx
  - ...
Building Encrypted Data Processing Systems

DBMS

CryptDB

Blind Seer

Arx

...
Building Encrypted Data Processing Systems

DBMS
- CryptDB
- Blind Seer
  - Arx
  - ...  

Analytics
- Seabed
- Senat
- Conclave
- ...  

Machine Learning
- CryptoNets
- Helen
- RoFL
- ...  

Streaming
- TimeCrypt
- Zeph
- Waldo
- ...  

Internet of Things
- Talos
- Pilatus
- Kryptein
- ...  

Additional Systems:
- Pilatus
- Kryptein
End-to-End Encrypted Systems
End-to-End Encrypted Systems
End-to-End Encrypted Systems
End-to-End Encrypted Systems
Privacy protection means more than securing the data …
Data Misuse

use of data for purposes that the user did not agree to
End-to-End Encrypted Systems

Fundamental issue: unrestricted views of the data
End-to-End Encrypted Systems → End-to-End Privacy

Fundamental issue: unrestricted views of the data

privacy enhanced
End-to-End Encrypted Systems → End-to-End Privacy

Fundamental issue: unrestricted views of the data

privacy enhanced

Data Minimization  Purpose Limitation
Goal:

End-to-End Security → End-to-End Privacy
Goal:

End-to-End Security  ➔  End-to-End Privacy

data confidentiality
unauthorized parties
Goal:

End-to-End Security $\rightarrow$ End-to-End Privacy

data confidentiality $\&$ strong privacy guarantees
unauthorized parties $\&$ authorized parties
My Research: Building practical systems that use cryptography to empower users and preserve their privacy & tools to democratize cryptography

- E2E Privacy
- E2E Security

- Talos & Pilatus [SenSys’15 & ‘17]
- Droplet [Usenix Sec’20]
- TimeCrypt [Usenix NSDI’20]
- Zeph [Usenix OSDI’21]
- RoFL [In Submission]
- FHE Compilers [IEEE S&P’21]
- HECO [In Submission]

Privacy Enforcement
Functionality & Performance
Robustness
Accessibility
Zeph

User-centric Model for Privacy

Cryptographically Enforces Privacy

(Usenix OSDI ‘21)
One of Many Scenarios

"Raw Location Data"

Privacy Transformation

"Daily popular running tracks"

Policy

- Temporal
- Spatial
- Population

Day 6
One of Many Scenarios

Privacy Transformation

Policy

Temporal

Spatial

Population

“Daily popular running tracks”

End-to-End Security

Enforcement
Existing End-to-End Encrypted Streaming Pipeline
Existing End-to-End Encrypted Streaming Pipeline
Existing End-to-End Encrypted Streaming Pipeline
Integrate Privacy Controls into Existing Pipelines

Data Producer → Data → Encrypted Data Storage / Processing
Integrate Privacy Controls into Existing Pipelines

Data Producer

Policy Enforcement

Encrypted Data Storage / Processing

Policy

Enforcement

Data
Integrate Privacy Controls into Existing Pipelines

- Policy
- Encrypted Data Storage / Processing
- Data Producer
- Privacy Protocol
State-of-the-art

Privacy Protocols

Public: aggregation + masking
State-of-the-art

Privacy Protocols

Policy

Public: aggregation + masking

Privacy Protocol

Policy

Public: differential privacy

Privacy Protocol
Our Approach

State-of-the-art

Privacy Protocols

1. Public: aggregation + masking

2. Public: differential privacy

Privacy Platform

generic data upload for multiple privacy transformations
Data Producer

Data

Encrypted Data Storage / Processing
1. **Compatibility with Existing Systems**

Data Producer

Data

Encrypted Data Storage / Processing
1. **Compatibility with Existing Systems**

2. **Data with Heterogeneous Privacy Policies**

   A  
   B  
   C  

Encrypted Data Storage / Processing
1. **Compatibility** with Existing Systems

2. Data with **Heterogeneous** Privacy Policies

A  
B  
C  

Data Producer

Data
1. **Compatibility** with Existing Systems

2. **Data with Heterogeneous Privacy Policies**

3. Allow **Transformations** on Encrypted Data
Zeph’s Approach
Zeph’s Approach

Data Plane

Data Producer

Data

Encrypted Data Storage / Processing

Privacy Plane
Zeph’s Approach

Privacy Plane

Privacy Controller

Data Plane

Data Producer

Encrypted Data Storage / Processing

Policy
Zeph’s Approach

Data Plane

Data Producer

Encrypted Data Storage / Processing

Privacy Plane

Privacy Controller

Privacy Transformation

Privacy Compliant “Public” Views

Policy

“Public” Views

Data
Zeph’s Approach

Data Plane

Data Producer

Encrypted Data Storage / Processing

Privacy Plane

Privacy Controller

Privacy Transformation

Privacy Compliant “Public” Views

Policy

"Public" Views
Zeph’s Approach

Data Plane

Privacy Plane

Data Producer

Privacy Controller

Policy

Data

Encrypted Data Storage / Processing

Privacy Transformation

Privacy Compliant “Public” Views

Token
Zeph’s Approach

Data Plane

Privacy Plane

Privacy Transformation
Privacy Compliant “Public” Views

Federated Privacy Control

Privacy Controller

Token

Data Producer

Data

Encrypted Data Storage / Processing

Policy

Token

Privacy Controller

Privacy Transformation
Cryptographic Privacy Tokens

\[ \trianglerightarrow Enc(\triangle, \bullet) \rightarrow \triangle \]
Cryptographic Privacy Tokens

\[ \triangle \rightarrow Enc(\triangle, \text{key}) \rightarrow \triangle \rightarrow f(\triangle) \]
Cryptographic Privacy Tokens

\[ \triangle \rightarrow Enc(\triangle, k) \rightarrow \triangle \rightarrow f(\triangle) \rightarrow Dec(\triangle, k) \]
Cryptographic Privacy Tokens

\[ \triangle \rightarrow Enc(\triangle, \bullet) \rightarrow \triangle \rightarrow \color{gray} f(\triangle) \rightarrow Dec(\triangle, \bullet) \rightarrow \triangle \]

Unfettered view of the data
Cryptographic Privacy Tokens

Unfettered view of the data

\[ \triangle \rightarrow Enc(\triangle, \mathcal{K}) \rightarrow f(\triangle) \rightarrow Dec(\triangledown, \mathcal{K}) \rightarrow \triangledown \]

\[ \mathcal{K} \rightarrow f'(\mathcal{K}) \]
Cryptographic Privacy Tokens

\[ \triangle \rightarrow Enc(\triangle, \bullet) \rightarrow \triangle \rightarrow f(\triangle) \rightarrow Dec(\triangle, \bullet) \rightarrow \triangle \]

\[ \bullet \rightarrow f'(\bullet) \rightarrow \text{Unfettered view of the data} \]
Cryptographic Privacy Tokens

Unfettered view of the data

Privacy enhanced view of the data

\[ Enc(\triangle, \text{key}) \rightarrow f(\triangle) \rightarrow Dec(\triangledown, \text{key}) \]

\[ f'(\text{key}) \rightarrow \]
Zeph’s Threat Model and Assumptions

**Data Plane**
- Data Producer
- Encrypted Data Storage / Processing
- Fully Trusted

**Privacy Plane**
- Policy
- Privacy Controller
- Token
- Privacy Transformation
- Privacy Compliant
- “Public” Views
- Honest-But-Curious

PKI + \(<\alpha\) colluding parties

Federated Privacy Control

(time \(\mathbb{0}, \text{data}\))
Privacy Transformations
Existing Privacy Transformations

“Practical” Privacy Tools
- Data Masking
  - Pseudonymization
  - Tokenization
  - Perturbation
  - Redaction
  - Date Shifting

Data Generalization
- Bucketing
- Time Resolution
- Population

Formal Privacy Models
Existing Privacy Transformations

**“Practical” Privacy Tools**
- Data Masking
  - Pseudonymization
  - Tokenization
  - Perturbation
  - Redaction
  - Date Shifting

**Data Generalization**
- Bucketing
- Time Resolution
- Population

**Formal Privacy Models**
- Differential Privacy (DP)
  - Additive Noise Mechanism
  - Others (e.g., ME, SVT)
- k - Anonymity
How Zeph augments existing System Designs
How Zeph augments existing System Designs

Data Producer

Data Producer

Data Producer

Streaming Jobs

Orchestrator

Stream Processing Platform

Schema Registry

Schema
How Zeph augments existing System Designs
How Zeph augments existing System Designs

Privacy Controller

Data Producer

Token

Data

Schema

Stream Processing Platform

Streaming Jobs Orchestrator

Streaming Jobs

Privacy Transformation

Schema

Streaming Query

Schema

Schema Registry
How Zeph augments existing System Designs

Stream Processing Platform
- Policy Manager
- Streaming Jobs Orchestrator

Streaming Jobs
- Privacy Transformation
- Schema

Privacy Controller
- Token
- Data Producer
- Schema

Data Producer
- Data
- Schema

Schema Registry

Kafka
Contributions

User-centric privacy
- Keep End-User Control Simple

Privacy Orchestration
- Organize Privacy Transformations

Cryptographic Enforcement
- Cryptographic Privacy Tokens
Contributions

User-centric privacy
Keep End-User Control Simple

Privacy Orchestration
Organize Privacy Transformations

Cryptographic Enforcement
Cryptographic Privacy Tokens

Mapping
Audit
Privacy Transformations

Privacy Compatible Matching

Stream View Query 1
Stream View Query 2

Privacy Enhanced View

Cryptographic Token
Encrypted Data
Cryptographic Enforcement of Privacy

1) **Confidentiality** of data

2) Transformation **Authorization** by Privacy Controller

3) **Compute** transformation on confidential data

4) Privacy Controller is **efficient** and **independent** of data
Cryptographic Enforcement of Privacy

1) **Confidentiality** of data

2) Transformation **Authorization** by Privacy Controller

3) **Compute** transformation on confidential data

4) Privacy Controller is **efficient** and **independent** of data

---

**Additive Homomorphic Secret Sharing**

\[
\text{Share } x \\
\x_1 \rightarrow \text{Eval}_f(x_1) \rightarrow y_1 \\
\x_2 \rightarrow \text{Eval}_f(x_2) \rightarrow y_2 \\
\oplus \rightarrow f(x)
\]
1) Confidentiality of data

2) Transformation Authorization by Privacy Controller

3) Compute transformation on confidential data

4) Privacy Controller is efficient and independent of data

Additive Homomorphic Secret Sharing

Data Producer
Cryptographic Enforcement of Privacy

1) **Confidentiality** of data

2) Transformation **Authorization** by Privacy Controller

3) **Compute** transformation on confidential data

4) Privacy Controller is **efficient** and **independent** of data

Additive Homomorphic Secret Sharing

- **Data Producer**
- **Privacy Controller**

\[ f'(\cdot) \]
Cryptographic Enforcement of Privacy

1) **Confidentiality** of data

2) Transformation **Authorization** by Privacy Controller

3) **Compute** transformation on confidential data

4) Privacy Controller is **efficient** and **independent** of data

Additive Homomorphic Secret Sharing

- **Data Producer**
- **Privacy Controller**
- **Server**

Privacy enhanced view of the data
Privacy Transformations

**Practical** Privacy Tools

- Data Masking
  - Pseudonymization
  - Tokenization
  - Perturbation
  - Redaction
  - Date Shifting

- Data Generalization

**Formal Privacy Models**

- Differential Privacy (DP)
  - Additive Noise Mechanism
  - Others (e.g., SVT)

- k - Anonymity

\[ T'(\cdot) = \sum \]
Privacy Transformations

“Practical” Privacy Tools

- Data Masking
  - Pseudonymization
  - Tokenization
  - Perturbation
  - Redaction
  - Date Shifting

- Data Generalization
  - Bucketing
  - Time Resolution
  - Population

Formal Privacy Models

- Differential Privacy (DP)
  - Additive Noise Mechanism
  - Others (e.g., SVT)

- k-Anonymity
Additive Homomorphic Privacy Transformations

\[ T'(\cdot) = \sum \]

**Data Producer**
- A
- B

**Keys**
- A
- B

**Privacy Controller** - operates on keys

**Ciphertexts**
- \[ T'(\cdot) \]

**Operate on data**

**T'(\cdot)**
Additive Homomorphic Privacy Transformations

\[ T'(\cdot) = \sum \]

Privacy Controller - operates on keys

Privacy Compliant “Public” View

Data Masking
Redaction
Perturbation
Generalization

Data Producers

Ciphertexts

\(\text{Keys} \quad A \quad B \quad C \quad D\)

\(\text{Data} \quad \text{Producer} \quad - C \quad - D\)

\(\text{Ciphertexts} \quad \text{Data} \quad \text{Producer} \quad - A \quad - B\)

\(T'(\cdot) \quad \text{operate on data}\)
Additive Homomorphomic Privacy Transformations

Ciphertexts

A - B

A + C

Data Masking
Redaction

Generalization
Population

Data Masking
Perturbation

Privacy Controller - operates on keys

Privacy Compliant
"Public" View

T'(·) = ∑

T'(·)

Data Producer

A - C

B - D

Data Producer

A - A

B - B

Ciphertexts

operate on data
Additive Homomorphic Privacy Transformations

\[ T'(\cdot) = \sum \]

**Privacy Controller** - operates on keys

Data Masking

Redaction

Keys

\( A \quad C \)

\( B \quad D \)

Data Producer

\( A \quad C \quad D \)

Ciphertexts

Data Masking

Population

\( A + C \)

Data Masking

Perturbation

\( A + C + \Delta \)

Privacy Compliant

“Public” View

Ciphertexts

Data Masking

Redaction

Data Producer

\( A \quad C \quad D \)

\( B \quad A \quad D \)

Data Producer

\( A \quad - C \quad - D \)

\( A \quad - A \quad - B \)

\( B \quad A \quad D \)

**T'**

\( T'(\cdot) \)

operate on data
Additive Homomorphic Privacy Transformations

\[ T'(\cdot) = \sum \]

Data Masking
Redaction

Data Masking
Perturbation

Generalization
Population

Data Producer

Keys

Privacy Controller - operates on keys

Privacy Compliant
"Public" View

\[ T'(\cdot) \]

operate on data

operate on data
Enable Federated Privacy Control

“multiple Data Producers - one Privacy Controller”
Enable Federated Privacy Control

“multiple Data Producers - multiple Privacy Controllers”
Enable Federated Privacy Control

"multiple Data Producers - multiple Privacy Controllers"

Secure Aggregation Protocol
Only reveal the aggregation of the keys to the server
Zeph Implementation and Evaluation

Data Producer & Privacy Controller

Privacy Transformation

Apache Kafka Streams

Java

Rust Programming Language
Zeph Implementation and Evaluation

Data Producer & Privacy Controller

Privacy Transformation

Fitness App
Website Analytics
Smart Car

Streams
Zeph Implementation and Evaluation

Data Producer & Privacy Controller

Privacy Transformation

Fitness App  Website Analytics  Smart Car
Web Analytics: End-to-End Benchmark

![Graph showing latency vs. data producers/privacy controllers for Plaintext and Zeph.](image-url)
Web Analytics: End-to-End Benchmark

- Modest Overhead
- 1 sec compared to 0.46 sec
My Research: Building practical systems that use cryptography to empower users and preserve their privacy & tools to democratize cryptography
TimeCrypt (Usenix NSDI '20)

Encrypted Time Series Database
Can we enable encrypted data processing for time series workloads?
Can we enable encrypted data processing for **time series** workloads?
Time Series Data is Emerging Everywhere

Time-ordered observations of a quantitative characteristic of an individual or phenomenon taken at successive points in time.

Financial    Health Monitoring    Smart Grid    Internet of Things    DevOps/Telemetry
Time Series Data is Emerging Everywhere

Time-ordered observations of a quantitative characteristic of an individual or phenomenon taken at successive points in time.

high resolution sensitive data!

Financial  Health Monitoring  Smart Grid  Internet of Things  DevOps/Telemetry
Time Series Databases

**Time Series Workloads**
- Primarily INSERTS to recent time interval
- Statistical queries over time ranges
- Single writer

**Performance Requirements**
- High throughput writes
- Data compaction (aging out data)
- Scale with data volume and velocity
Data Sources

Services
Constrained Devices

Large volume

TS Access Control Semantics

Low-latency

Functionality of TSDB

Data Sources

Services
Services

Data Sources

Constrained Devices

Large volume

Functionality of TSDB

Low-latency

TS Access Control Semantics

Services
Cryptographic Access Control

Data Sources

Services

Doctor

Health Insurance
Cryptographic Access Control

Data Sources

Services

Doctor

Health Insurance
Cryptographic Access Control

The services **learned** more than they **needed**!
Cryptographic Access Control

The services *learned* more than they *needed*!

Fine grained data protection

Data Sources

Services

Doctor

Health Insurance
Cryptographic Access Control

Support **time series** access control semantics
- Time
- Resolution
- Attribute
Cryptographic Access Control

Support **time series** access control semantics
- Time
- Resolution
- Attribute

Data Sources

Duration of Sickness

Services

Policy( )

Doctor

Health Insurance
Cryptographic Access Control

Support **time series** access control semantics
- Time
- Resolution
- Attribute

Data Sources

- Monthly Resolution
- Duration of Sickness

Services

- Doctor
- Health Insurance

Policy( )
# TimeCrypt

Encrypted Time Series Database that enables scalable computation over large volumes of encrypted time series data.

<table>
<thead>
<tr>
<th>Functionalities</th>
<th>Statistical queries and lifecycle operations over encrypted data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cryptographic Access Control</td>
<td>Fine-grained access policies over time, resolution, and attributes</td>
</tr>
<tr>
<td>Security</td>
<td>Data Secrecy -- Homomorphic Encryption</td>
</tr>
<tr>
<td></td>
<td>Function Integrity -- Homomorphic MACs</td>
</tr>
<tr>
<td>Efficiency</td>
<td>Interactive queries over large scale data</td>
</tr>
</tbody>
</table>
Large-Scale Challenges

Lots of Data:
Supporting “big data” computations

Fine-grained Access Control:
Scalability as data and the number of data consumers grow

E.g., Partial Homomorphic Encryption [Paillier]
E.g., Attribute Based Encryption [KP-ABE]
Large-Scale Challenges

Lots of Data:
Supporting “big data” computations

Fine-grained Access Control:
Scalability as data and the number of data consumers grow

E.g., Partial Homomorphic Encryption [Paillier]
E.g., Attribute Based Encryption [KP-ABE]

Solution that supports both fine-grained access control and computations over large-scale encrypted data
| Efficiency | Additive Symmetric Homomorphic Encryption [Castelluccia] |
Building Blocks

Efficiency

Additive Symmetric Homomorphic Encryption [Castelluccia]

Key stream: \( k_0, k_1, k_2, k_3, k_4, k_5, \ldots \)
Building Blocks

Efficiency

Additive Symmetric Homomorphic Encryption [Castelluccia]

Key stream: \(k_0, k_1, k_2, k_3, k_4, k_5, \ldots\)

\[ m_0 + k_0 \quad m_1 + k_1 \quad m_2 + k_2 \quad \ldots \quad \text{modulo } M \]
Additive Symmetric Homomorphic Encryption [Castelluccia]

Key stream: $k_0, k_1, k_2, k_3, k_4, k_5, \ldots$

$m_0 + k_0 \quad m_1 + k_1 \quad m_2 + k_2 \quad \ldots \quad \text{modulo } M$

$m_0 + m_1 + k_0 + k_1$

Time
Building Blocks

Additive Symmetric Homomorphic Encryption [Castelluccia]
- Dec. cost $O(n) \rightarrow$ Key cancelling $O(1)$

Key stream: $k_0, k_1, k_2, k_3, k_4, k_5, \ldots$

$$m_0 + k_0 \quad m_1 + k_1 \quad m_2 + k_2 \quad \ldots \mod M$$

$m_0 + m_1 + k_0 + k_1$
Building Blocks

Additive Symmetric Homomorphic Encryption [Castelluccia]
- Dec. cost $O(n) \rightarrow$ Key cancelling $O(1)$

Key stream: $k_0, k_1, k_2, k_3, k_4, k_5, \ldots$

$\Delta$

$m_0 + k_0 - k_1 \quad m_0 + k_1 - k_2 \quad m_2 + k_2 - k_3 \ldots \mod M$

$m_0 + m_1 + k_0 - k_1 + k_1 - k_2$

Time
Building Blocks

Efficiency

Additive Symmetric Homomorphic Encryption [Castelluccia]
- Dec. cost $O(n)$ → Key cancelling $O(1)$
- Key Identifiers → Time-encoded key-streams

Key stream: $k_0, k_1, k_2, k_3, k_4, k_5, \ldots$

$m_0 + k_0 - k_1$
$m_0 + k_1 - k_2$

$m_2 + k_2 - k_3$

$\cdots$ modulo $M$

$m_0 + m_1 + k_0 - k_1 + k_1 - k_2$

Time
### Building Blocks

<table>
<thead>
<tr>
<th>Efficiency</th>
<th>Additive Symmetric Homomorphic Encryption [Castelluccia]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Dec. cost $O(n)$ → Key cancelling $O(1)$</td>
</tr>
<tr>
<td></td>
<td>• Key Identifiers → Time-encoded key-streams</td>
</tr>
<tr>
<td>Expressiveness</td>
<td>Aggregatable Digests</td>
</tr>
</tbody>
</table>

## Building Blocks

### Efficiency

<table>
<thead>
<tr>
<th>Additive Symmetric Homomorphic Encryption [Castelluccia]</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Dec. cost $O(n)$ $\rightarrow$ Key cancelling $O(1)$</td>
</tr>
<tr>
<td>- Key Identifiers $\rightarrow$ Time-encoded key-streams</td>
</tr>
</tbody>
</table>

### Expressiveness

<table>
<thead>
<tr>
<th>Aggregatable Digests</th>
</tr>
</thead>
<tbody>
<tr>
<td>Known encodings: If we can compute sum privately, then we can compute $f(\cdot)$ privately</td>
</tr>
</tbody>
</table>

Known encodings: If we can compute sum privately, then we can compute $f(\cdot)$ privately

- **digest** $\rightarrow$ vector of encodings of the underlying data
Building Blocks

**Efficiency**

Additive Symmetric Homomorphic Encryption [Castelluccia]
- Dec. cost $O(n) \rightarrow$ Key cancelling $O(1)$
- Key Identifiers $\rightarrow$ Time-encoded key-streams

**Expressiveness**

Aggregatable Digests

Known encodings: If we can compute sum privately, then we can compute $f(\cdot)$ privately

- Vector of encodings of the underling data
- **Statistical queries**: average, sum, count, variance, min/max, histograms, least-squares regression, stochastic gradient descent, heavy hitters …
## Building Blocks

### Efficiency

**Additive Symmetric Homomorphic Encryption** [Castelluccia]
- Dec. cost $O(n) \rightarrow$ Key cancelling $O(1)$
- Key Identifiers $\rightarrow$ Time-encoded key-streams

### Expressiveness

**Aggregatable Digests**

Known encodings: If we can compute sum privately, then we can compute $f(\cdot)$ privately

- **Statistical queries**: average, sum, count, variance, min/max, histograms, least-squares regression, stochastic gradient descent, heavy hitters ...
## Building Blocks

<table>
<thead>
<tr>
<th>Efficiency</th>
<th>Additive Symmetric Homomorphic Encryption [Castelluccia]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>- Dec. cost $O(n)$ → Key cancelling $O(1)$</td>
</tr>
<tr>
<td></td>
<td>- Key Identifiers → Time-encoded key-streams</td>
</tr>
<tr>
<td>Expressiveness</td>
<td>Aggregatable Digests</td>
</tr>
<tr>
<td></td>
<td>Known encodings: If we can compute sum privately, then we can compute $f(\cdot)$ privately</td>
</tr>
<tr>
<td>Access Control</td>
<td>New Key Derivation Construction</td>
</tr>
</tbody>
</table>
## Building Blocks

<table>
<thead>
<tr>
<th>Efficiency</th>
<th>Additive Symmetric Homomorphic Encryption [Castelluccia]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>- Dec. cost $O(n)$ $\rightarrow$ Key cancelling $O(1)$</td>
</tr>
<tr>
<td></td>
<td>- Key Identifiers $\rightarrow$ Time-encoded key-streams</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Expressiveness</th>
<th>Aggregatable Digests</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Known encodings: If we can compute sum privately, then we can compute $f(\cdot)$ privately</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Access Control</th>
<th>New Key Derivation Construction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><img src="image" alt="Diagram" /></td>
</tr>
</tbody>
</table>

- $hash()$ indicator
- $hash_{\text{left}}()$ indicator
- $hash_{\text{right}}()$ indicator
- $t_0$ indicator
- $t_7$ indicator
- Shared token
- Shared keys
- KDF
- Hash functions
- Binary hash tree
- Time
TimeCrypt System Architecture
System Performance

Throughput under load of 4/1 read-write ratio, 49k streams
System Performance

Throughput under load of 4/1 read-write ratio, 49k streams

Throughput [records/s]

Throughput under load of plaintext
2% slowdown compared to plaintext
System Performance

Throughput under load of 4/1 read-write ratio, 49k streams

2% slowdown compared to plaintext

Throughput [records/s]

Ingest

Throughput under load of 4/1 read-write ratio, 49k streams
System Performance

Throughput under load of 4/1 read-write ratio, 49k streams

- 2% slowdown compared to plaintext

Throughput [records/s]

- Throughput [ops/s]
My Research: Building practical systems that use cryptography to empower users and preserve their privacy & tools to democratize cryptography

E2E Security

E2E Privacy

- FHE Compilers [IEEE S&P’21]
- HECO [In Submission]
- RoFL [In Submission]
- Talos & Pilatus [SenSys’15 & ‘17]
- Droplet [Usenix Sec’20]
- TimeCrypt [Usenix NSDI’20]
- Zeph [Usenix OSDI’21]

Privacy Enforcement

Functionality & Performance

Robustness

Accessibility
Fully Homomorphic Encryption Accessibility

(IEEE S&P ’21)

Advanced Cryptography

Programming Languages
FHE holds huge potential to transforming privacy

Finally “practical” - Real world use of FHE started to emerge
Developing FHE Applications is Notoriously Hard
Usable FHE

Advanced Cryptography  Programming Languages
Usable FHE

1. What makes developing FHE applications hard?
2. How can compilers address these complexities?
Functionality and performance depend on $f$’s representation:

- How do we express $f$
- How do we optimize $f$
Functionality and performance depend on $f$’s representation:

- How do we express $f$
- How do we optimize $f$

Optimizing this transformation yields better FHE efficiency
FHE Programming Paradigm

- Approximations
- Optimizations
- No If/Else
- No Loops
- SIMD Batching
Data Independence

No Jumps  No Loops  No If/Else

Standard C++

```c
int foo(int a, int b) {
    if(a < b) {
        return a * b;
    } else {
        return a + b;
    }
}
```

FHE

```c
int foo(int a, int b) {
    int c = a < b;
    int i = a * b;
    int e = a + b;
    return c*i + (1-c)*e;
}
```

Always worst-case performance
SIMD-like Parallelism

Could get orders-of-magnitude performance difference between different batching schemes.
Complex Design Space

Polynomial Functions

Schemes: BFV, BGV, CKKS
Complex Design Space

Polynomial Functions

Schemes: BFV, BGV, CKKS

Arbitrary Computation

Schemes: FHEW, TFHE

~1 OoM slowdown
Complex Design Space

Polynomial Functions

Schemes: BFV, BGV, CKKS

Arbitrary Computation

Schemes: FHEW, TFHE

~1 OoM slowdown

Parameter Selection

Cost Model
<table>
<thead>
<tr>
<th>Performance</th>
<th>More complex than overhead of underlying FHE operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programming</td>
<td>Wide gap between naïve implementations &amp; expert solutions</td>
</tr>
<tr>
<td>Paradigm</td>
<td></td>
</tr>
<tr>
<td>Compilers</td>
<td>Transform high level programs to efficient FHE circuits</td>
</tr>
</tbody>
</table>
Democratizing Fully Homomorphic Encryption

Developer with no crypto expertise

Automatically generate efficient and secure FHE for any custom workloads?
FHE Paradigm
Transform high-level programs to efficient FHE solutions

```cpp
void hd(vector<bool>u,
        vector<bool>v)
{
    int sum = 0;
    for(int i = 0;
        i < v.size();
        ++i)
    {
        sum += (v[i]!=u[i]);
    }
}
```
FHE Paradigm
Transform high-level programs to efficient FHE solutions

```cpp
void hd(vector<sbool> u, vector<sbool> v)
{
    int sum = 0;
    for(int i = 0; i < v.size(); ++i)
    {
        sum += (v[i] != u[i]);
    }
}
```
**FHE Paradigm**
Transform high-level programs to efficient FHE solutions

```cpp
void hd(vector<bool> u, vector<bool> v)
{
    int sum = 0;
    for(int i = 0; i < v.size(); ++i)
    {
        sum += (v[i]!=u[i]);
    }
}
```

**Architecture**
End-to-end compilation stack for FHE

- **Application**
- **Circuits**
- **Schemes**
- **Platforms**

CKKS, TFHE, BGV, BFV, ...

**HECO**
FHE Paradigm
Transform high-level programs to efficient FHE solutions

```
void hd(vector<bool> u, vector<bool> v)
{
  int sum = 0;
  for (int i = 0; i < v.size(); ++i)
  {
    sum += (v[i]!=u[i]);
  }
}
```

Architecture
End-to-end compilation stack for FHE

- **Application**
- **Circuits**
- **Schemes**
- **Platforms**

CKKS, TFHE, BGV, BFV, ...

E2E Automatic Optimization
What’s Next?
HECO
open source, **automated** end-to-end optimization for FHE
HECO

open source, automated end-to-end optimization for FHE

Cryptography : Primitives for Verifiable Computation
HECO

open source, automated end-to-end optimization for FHE

Cryptography: Primitives for Verifiable Computation

Systems: **Target HW directly**, generating code for CPU/GPU, upcoming dedicated FHE accelerators and heterogenous deployments using a mix of these.
End-to-End Privacy

Data with Heterogenous Privacy Restrictions

Diverse Data Consumers
End-to-End Privacy

Privacy Management
End-to-End Privacy

Privacy Management

Cryptographically Enforced Privacy

Data with Heterogenous Privacy Restrictions

Privacy Control Logic
Purpose Limitation - Data Minimization - Auditing - Deletion

Diverse Data Consumers
Secure and Robust Collaborative Learning

Secure Decentralized Learning

Secure Federated Learning
Secure and Robust Collaborative Learning

Problem: Model integrity

Secure Decentralized Learning

Secure Federated Learning
Retrofit privacy in the fabric of modern systems

Privacy-preserving, functional, and performant systems
End.