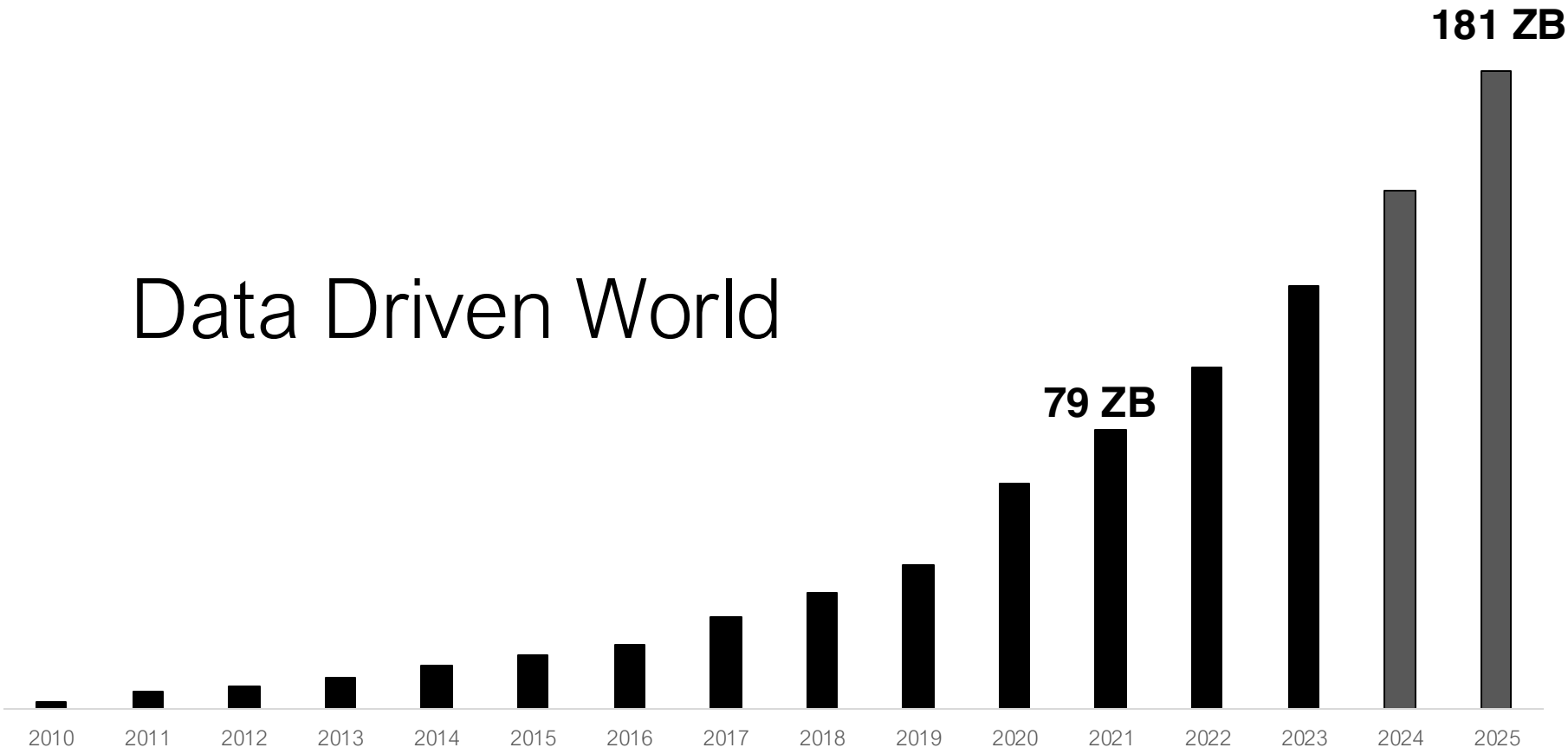
Democratizing Privacy-Preserving Computation

Anwar Hithnawi

ETH zürich

Data Driven World



Sensitive Data

Smart Homes

Genetics

Dating

Geolocation

Finance

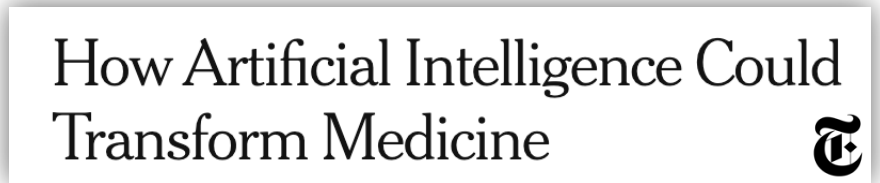
Health

Government

Personal


PARTNER CONTENT JORIS TOONDERS, YONEGO
WIRED
DATA IS THE NEW OIL OF THE DIGITAL ECONOMY


INNOVATION
Why Big Data Is The New Natural Resource
Forbes


How Artificial Intelligence Could Transform Medicine

PARTNER CONTENT JORIS TOONDERS, YONEGO

WIRED

DATA IS THE NEW OIL OF THE DIGITAL ECONOMY

INNOVATION

Why Big Data Is The New Natural Resource Forbes

How Artificial Intelligence Could Transform Medicine

You Should Be Freaking Out About Privacy

Nothing to hide, nothing to fear? Think again.

Grindr and OkCupid Spread Personal Details, Study Says

Norwegian research raises questions about whether certain ways of sharing of information violate data privacy laws in Europe the United States.

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

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

~ 1.245 Billion

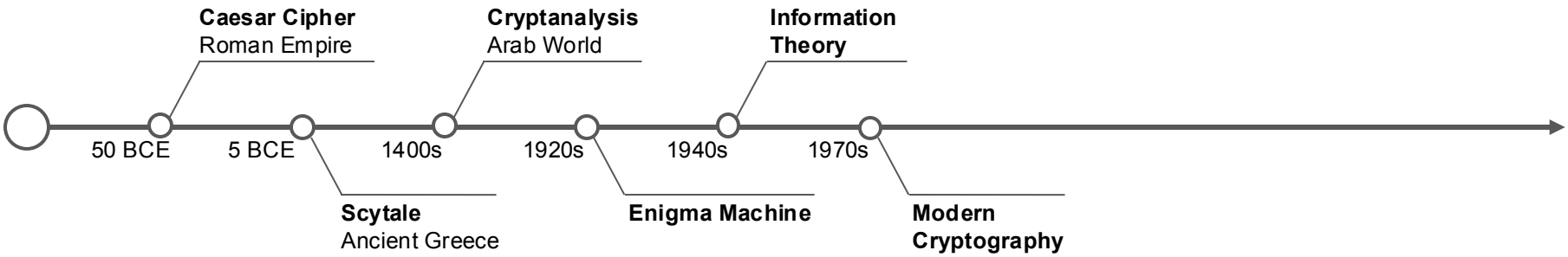
The number of data records
stolen in 2020

~ **1.245** Billion

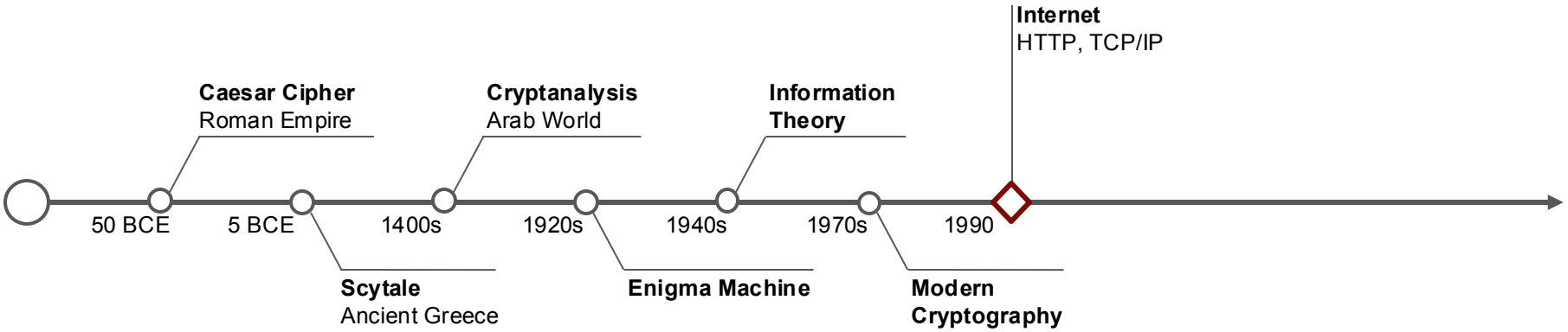
The number of data records **stolen** in 2020

143,000,000	57,000,000	330,000,000	533,000,000
EQUIFAX	Uber		
2017	2017	2018	2019

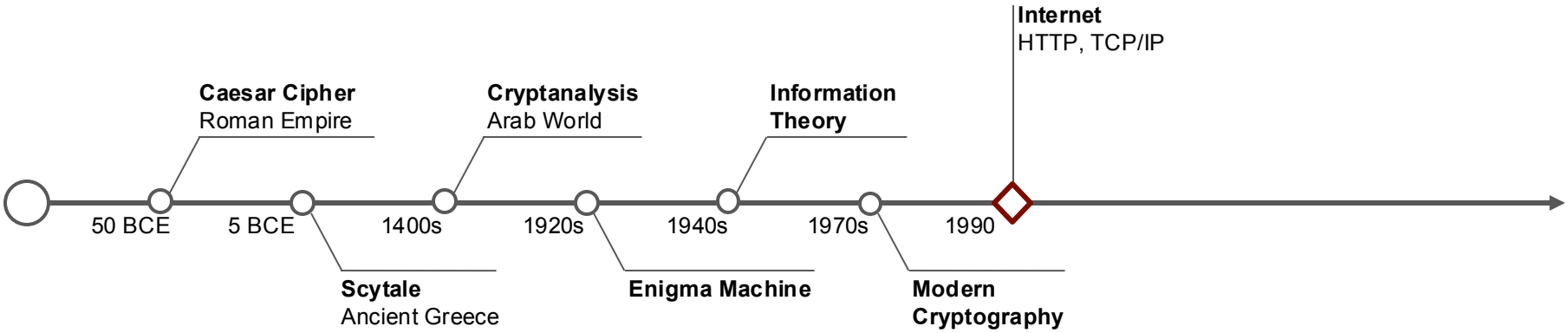
Data Protection: An Age-Old Concern



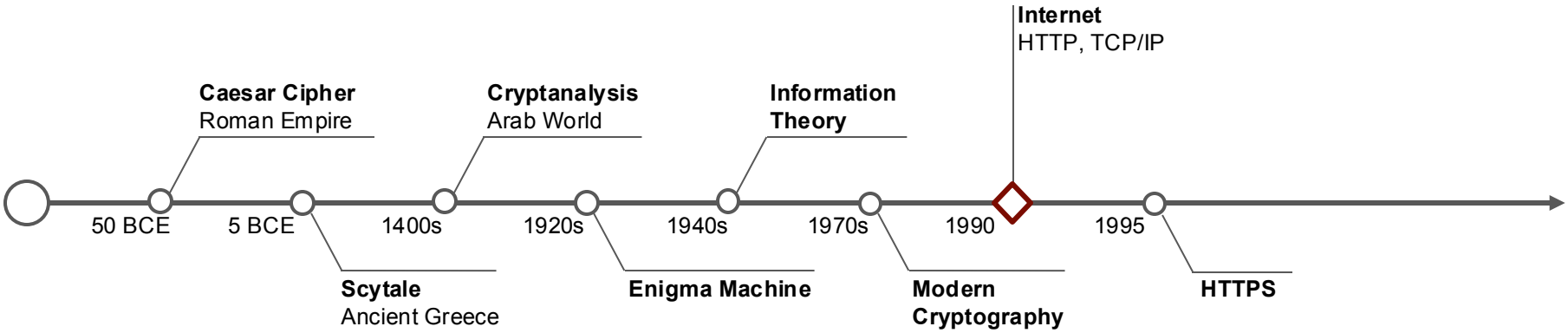
Data Protection: An Age-Old Concern



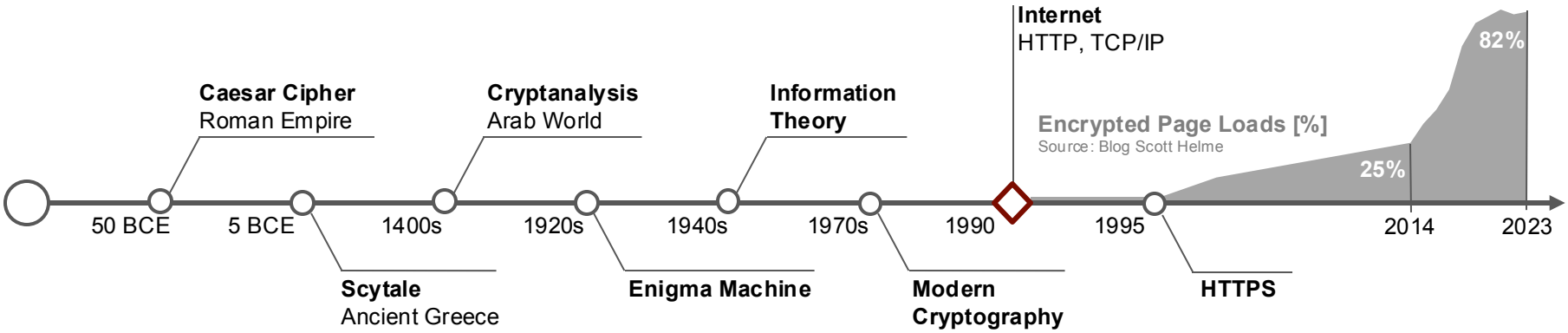
Data Protection: An Age-Old Concern



Data Protection: An Age-Old Concern

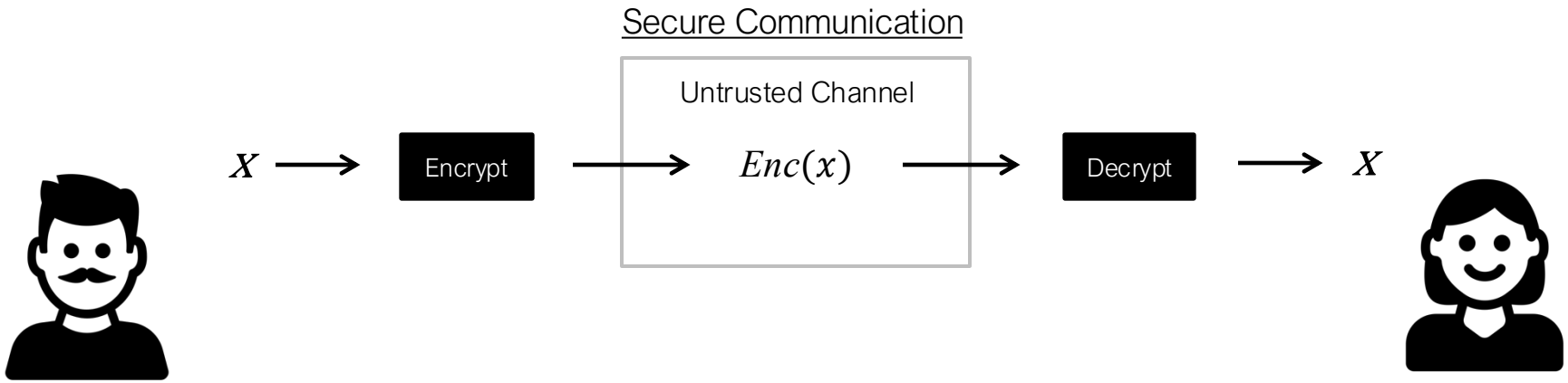


Data Protection: An Age-Old Concern

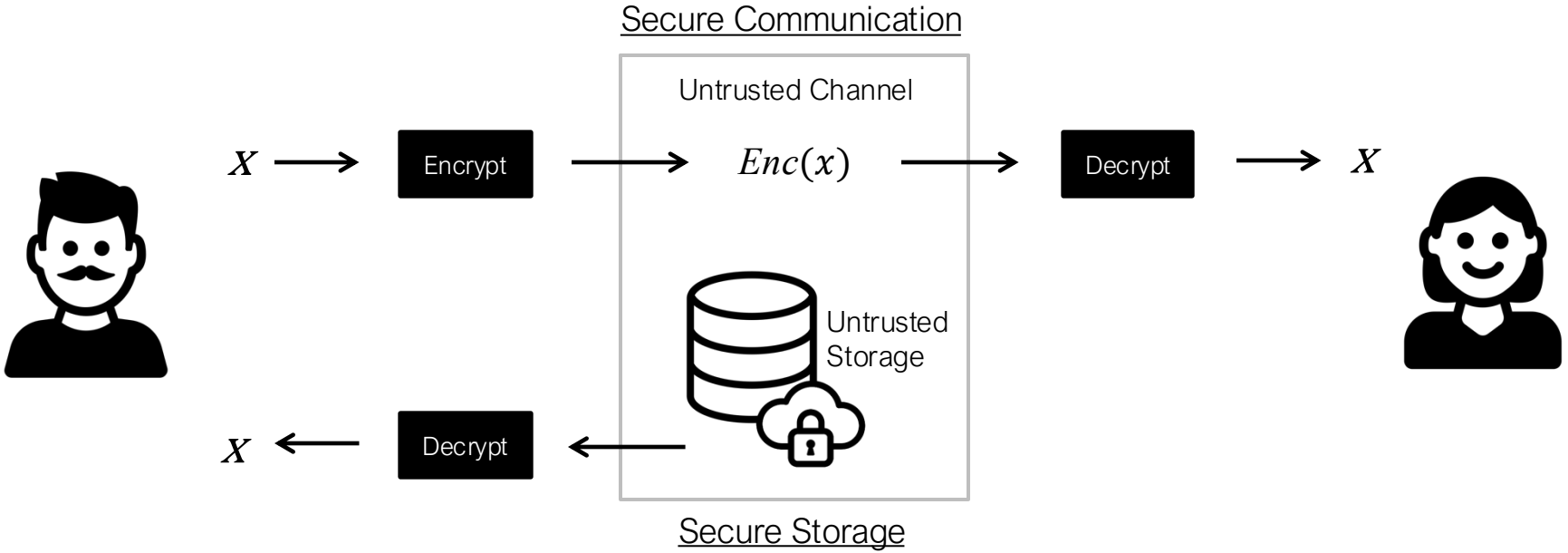


Securing Data: Building Blocks

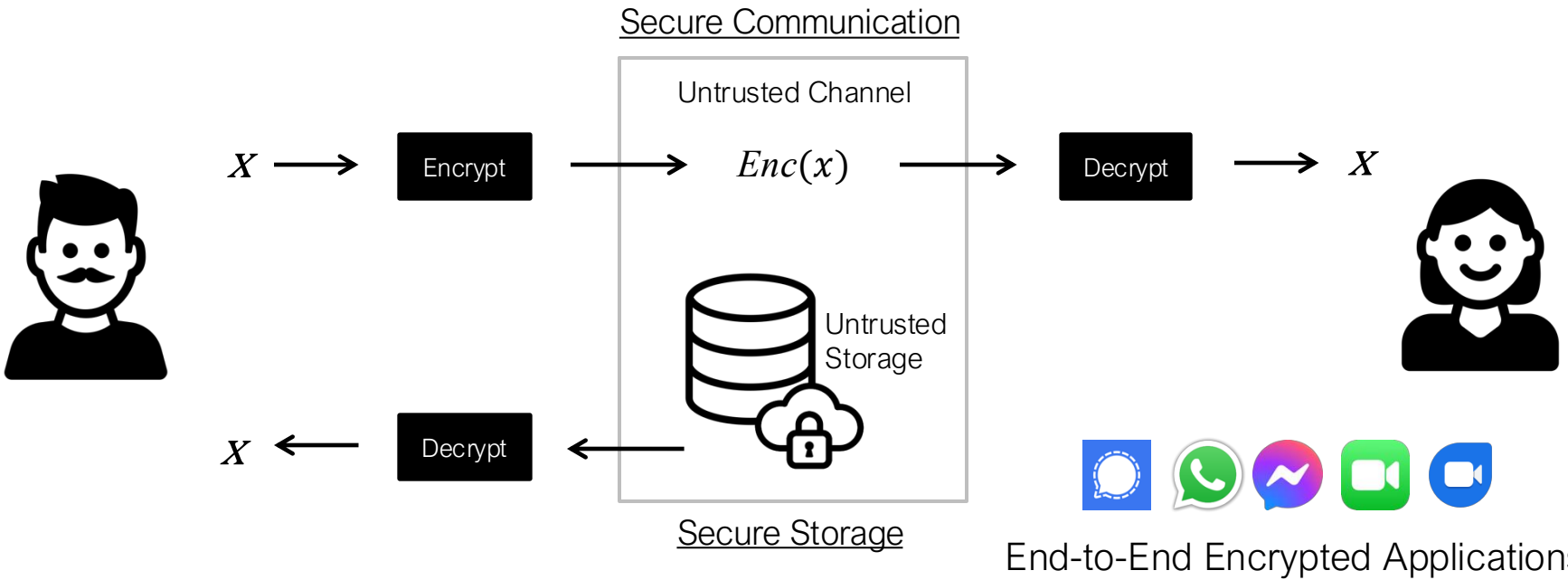
Securing Data: Building Blocks



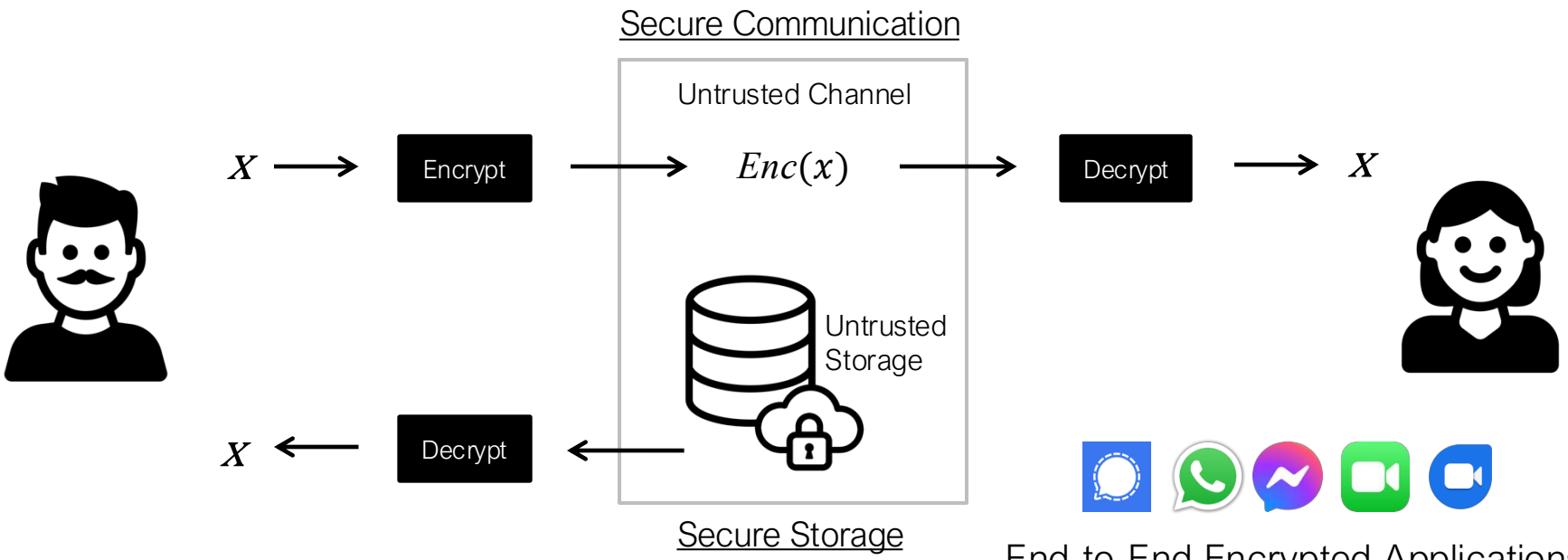
Securing Data: Building Blocks



Securing Data: Building Blocks



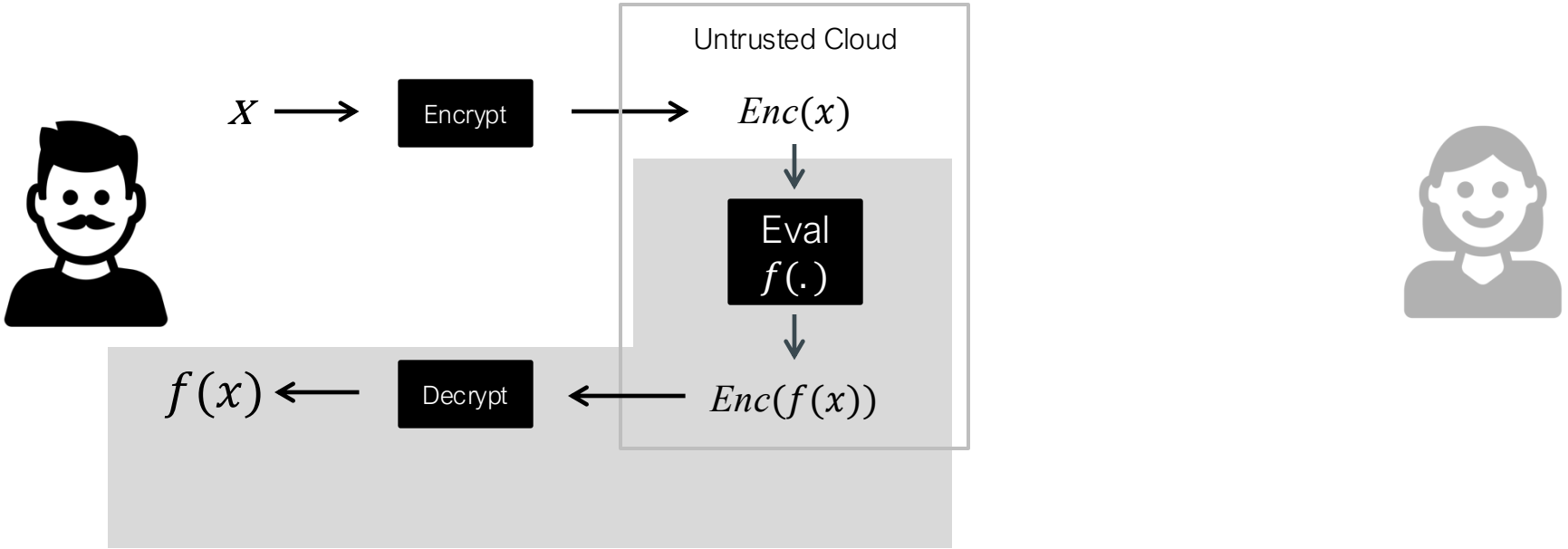
Securing Data: Building Blocks



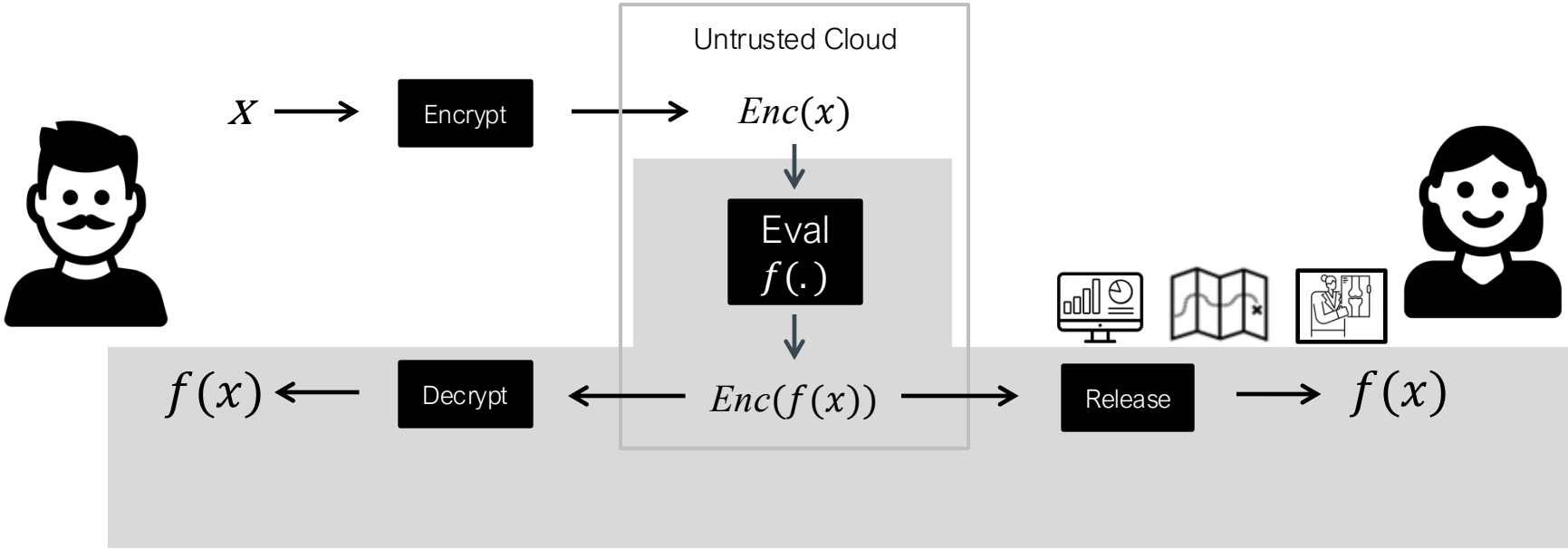
End-to-End Encrypted Applications

More Applications?

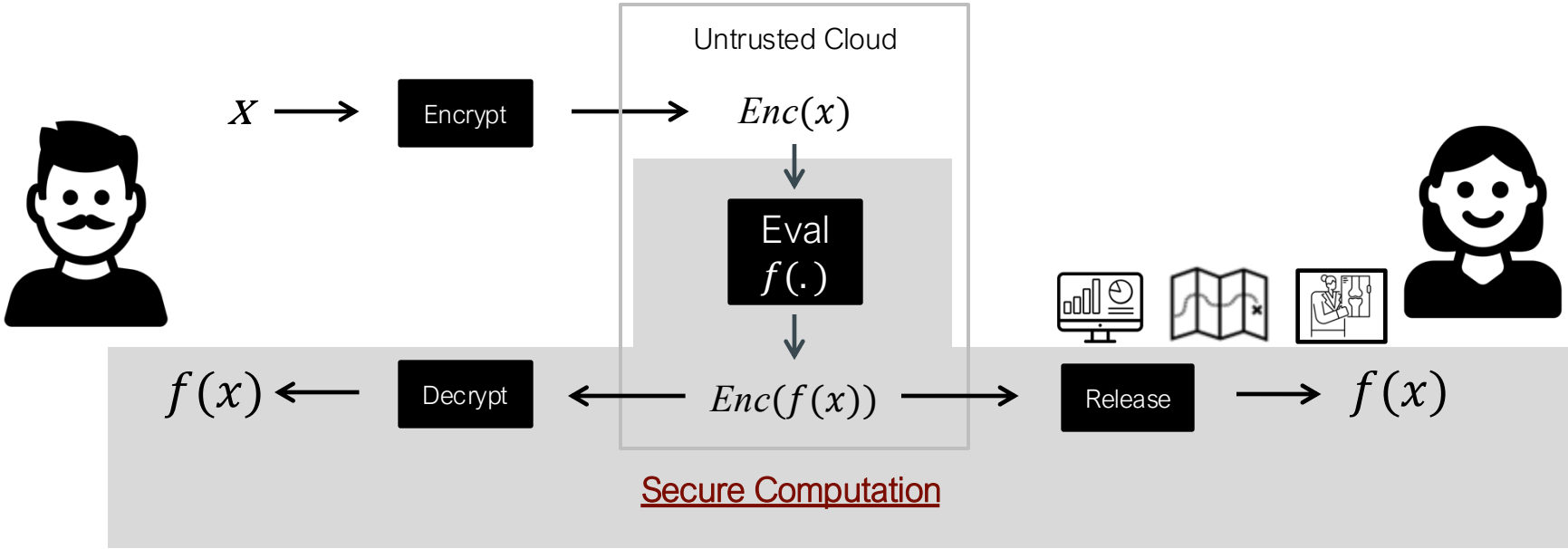
Securing Data in Use: Modern Applications



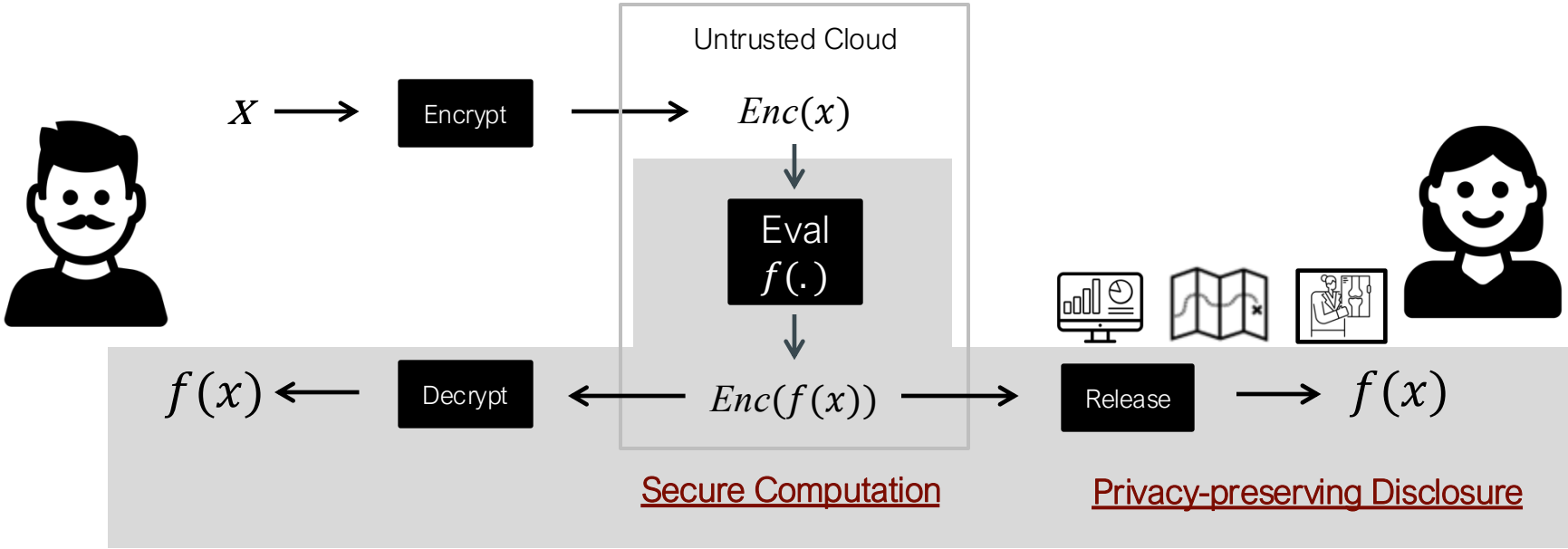
Securing Data in Use: Modern Applications



Securing Data in Use: Modern Applications

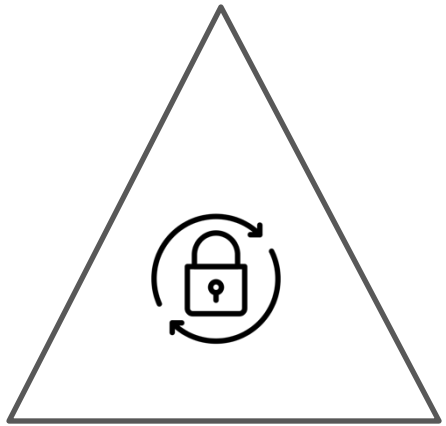


Securing Data in Use: Modern Applications



End-to-End Security

data at rest
secure storage



data in transit
secure communication

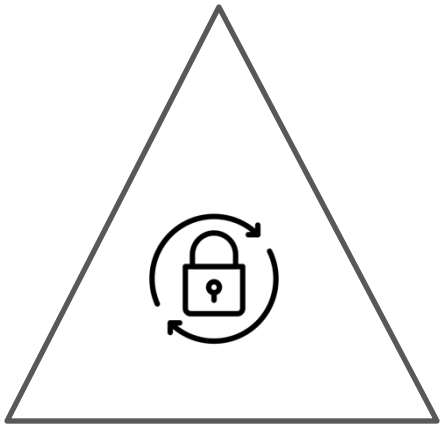
data in use
secure computation

End-to-End Security

Ubiquitous Adoption

Conventional Crypto

Encryption & Digital Signature

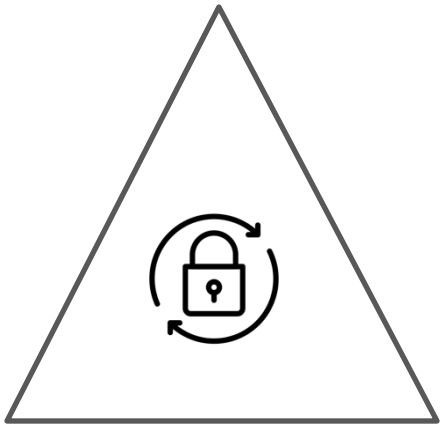


data at rest
secure storage

data in transit
secure communication

data in use
secure computation

End-to-End Security



data at rest
secure storage

Ubiquitous Adoption

Conventional Crypto

Encryption & Digital Signature

Just Starting

Privacy - Enhancing Technologies (PETs)

- Homomorphic Encryption
- Secure Multi-party Computation
- Zero Knowledge Proofs
- Differential Privacy

data in transit
secure communication

data in use
secure computation

~ 40 Years of History

1978
Homomorphic Encryption

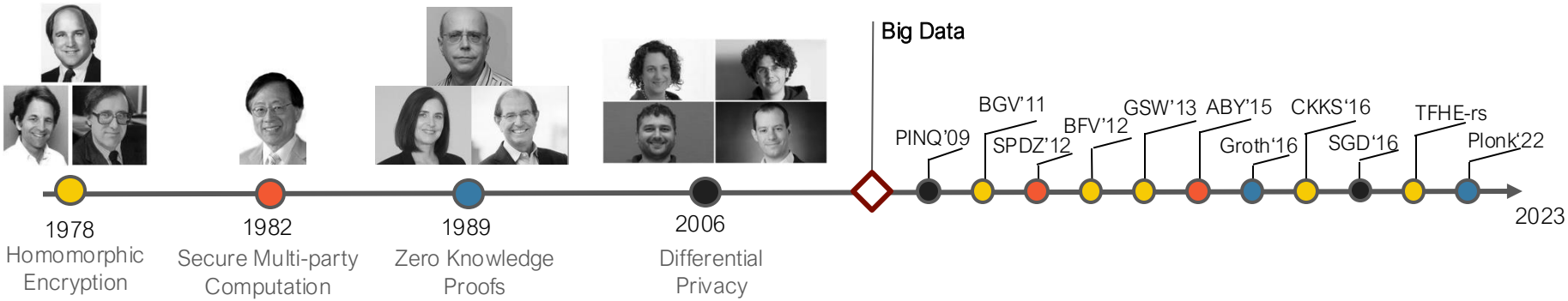
1982
Secure Multi-party Computation

1989
Zero Knowledge Proofs

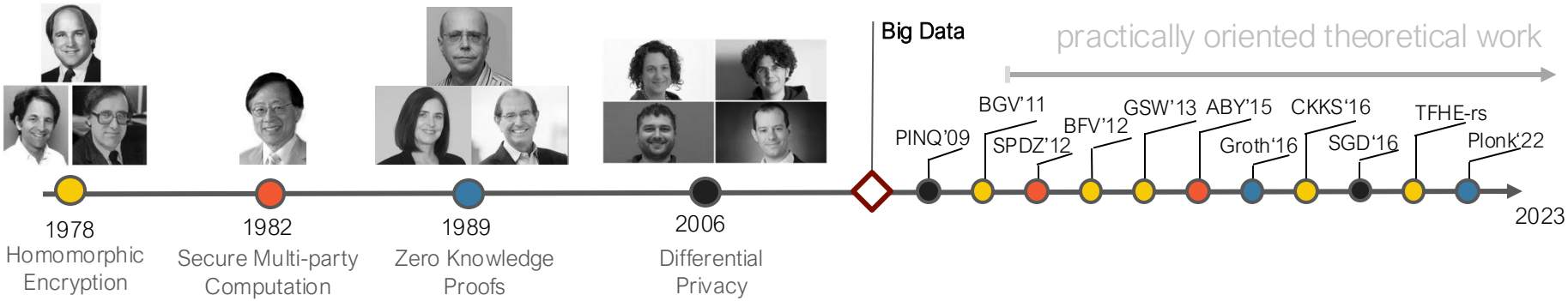
2006
Differential Privacy

2023

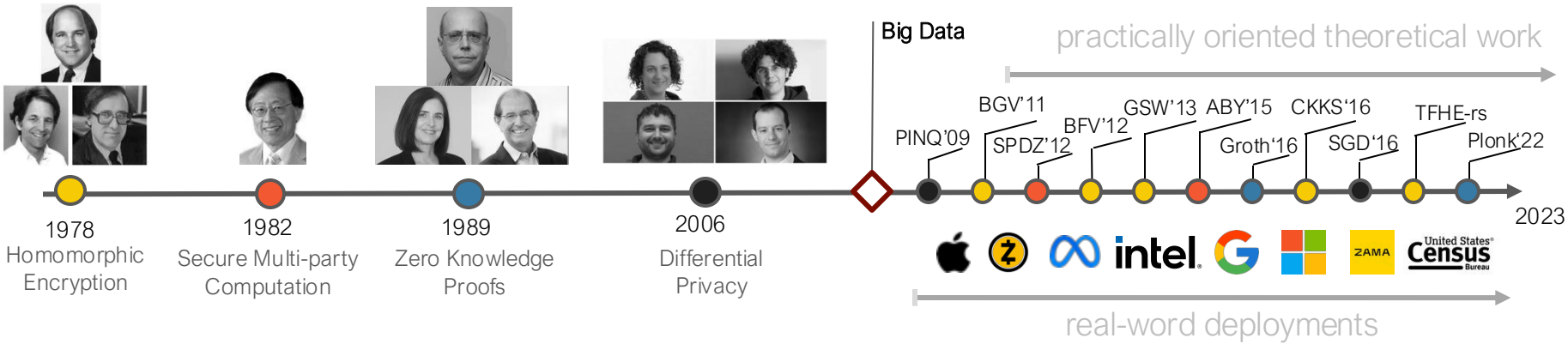
~ 40 Years of History

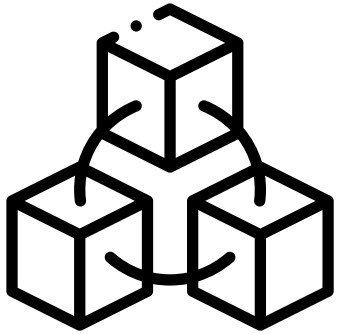


~ 40 Years of History

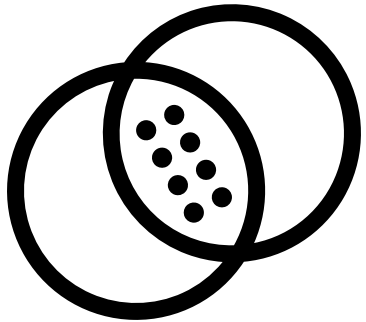


~ 40 Years of History



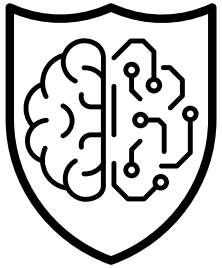


Blockchain

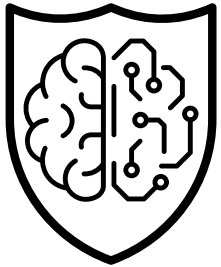


Private Set Intersection

Census



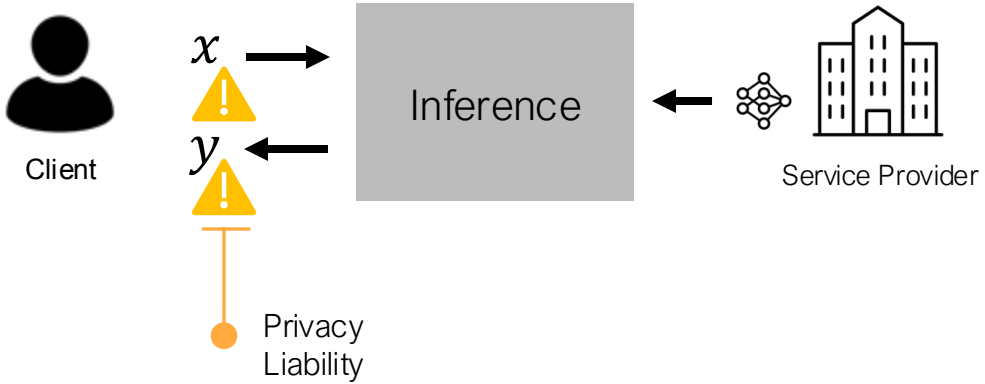
Privacy-Preserving Machine Learning



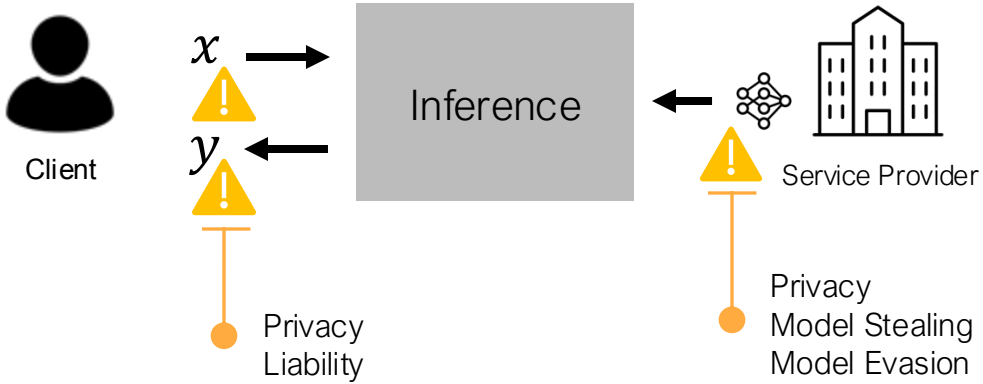
Privacy-Preserving Machine Learning

Private Machine Learning as a Service

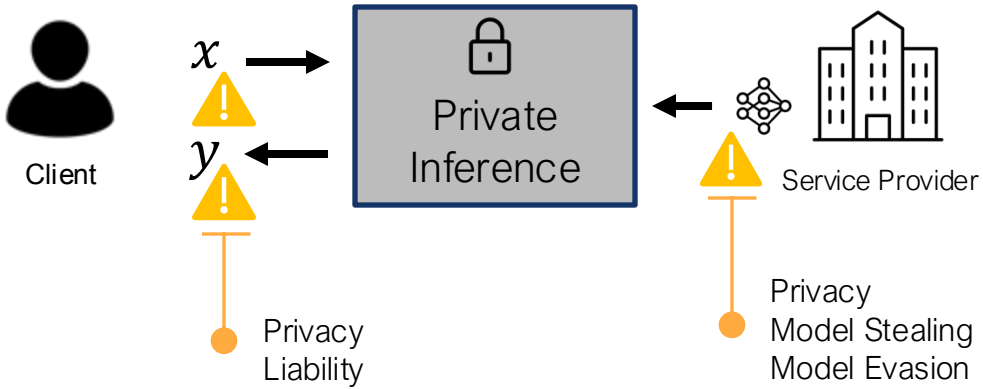
Private Machine Learning as a Service



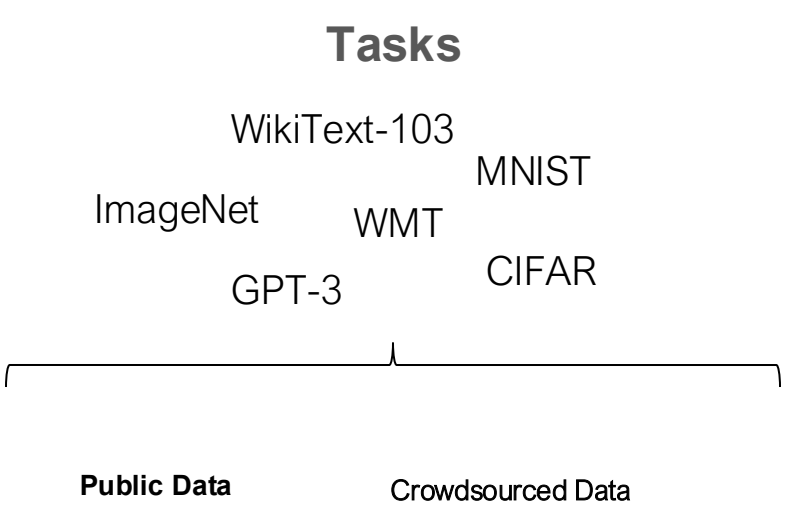
Private Machine Learning as a Service



Private Machine Learning as a Service



Solving tasks where data is **accessible**…



For example: web, books, articles, science, TV, corpus, audiobooks, …

Solving tasks where data is **accessible**…

For example: web, books, articles, science, TV, corpus, audiobooks, …

… however, many important tasks we care about ...

Inaccessible

Health – Cancer, Alzheimer, Dementia, Depression

Finance – Economic growth, Market predictions

Government – Education, Taxes, Immigration, Income

Personal Data – Text Messages, Emails, Photos

Solving tasks where data is **accessible**…

Tasks

WikiText-103

MNIST

ImageNet

WMT

CIFAR

GPT-3

Public Data

Crowdsourced Data

For example: web, books, articles, science, TV, corpus, audiobooks, …

… however, many important tasks we care about ...

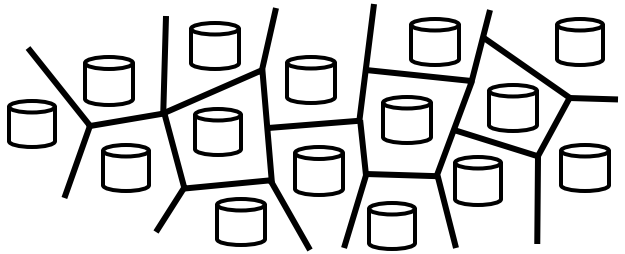
Inaccessible

Health – Cancer, Alzheimer, Dementia, Depression

Finance – Economic growth, Market predictions

Government – Education, Taxes, Immigration, Income

Personal Data – Text Messages, Emails, Photos



Data Silos

- Privacy Laws
- Competition

Solving tasks where data is **accessible**…

… however, many important tasks we care about ...

Tasks

WikiText-103

MNIST

ImageNet

WMT

CIFAR

GPT-3

Public Data

Crowdsourced Data

For example: web, books, articles, science, TV, corpus, audiobooks, …

Inaccessible

Health – Cancer, Alzheimer, Dementia, Depression

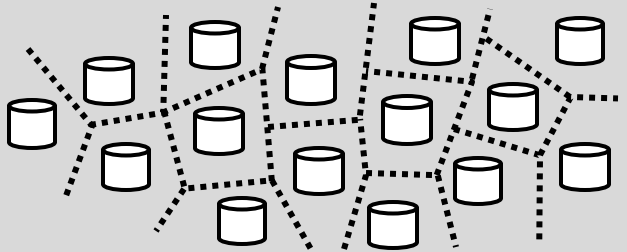
Finance – Economic growth, Market predictions

Government – Education, Taxes, Immigration, Income

Personal Data – Text Messages, Emails, Photos

→ EU Data Governance Act (**DGA**) effective from 2023

facilitate the reuse of protected public-sector data

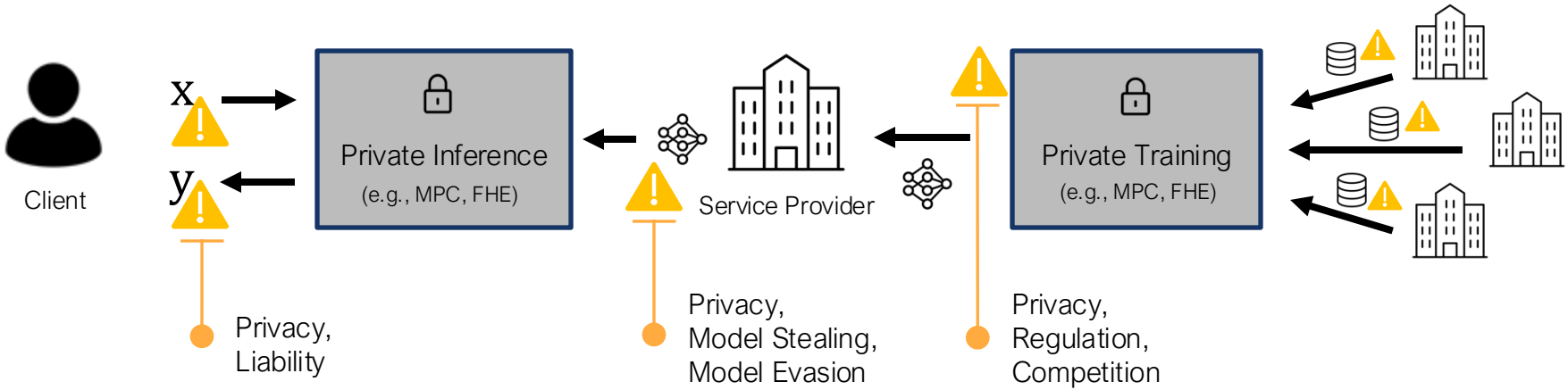


Legal Frameworks and Technologies to facilitate privacy preserving access

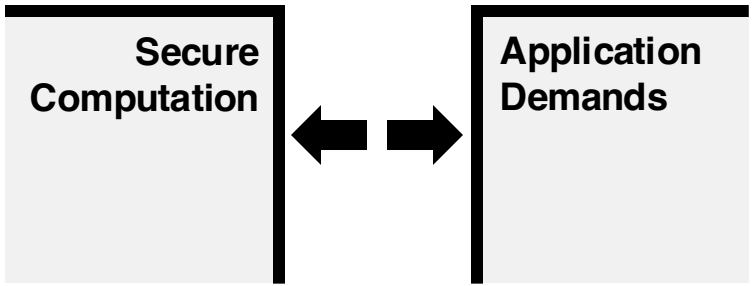
Data Silos

- Privacy Laws
- Competition

Privacy-Preserving Machine Learning

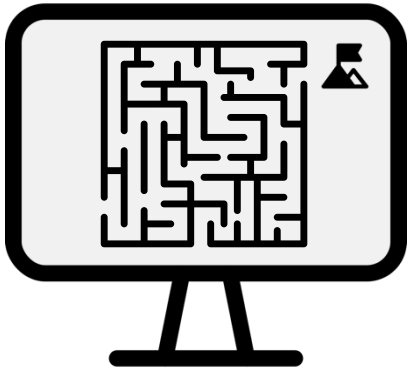


Theory to Practice: Barriers to Broad Adoption



Performance Gap

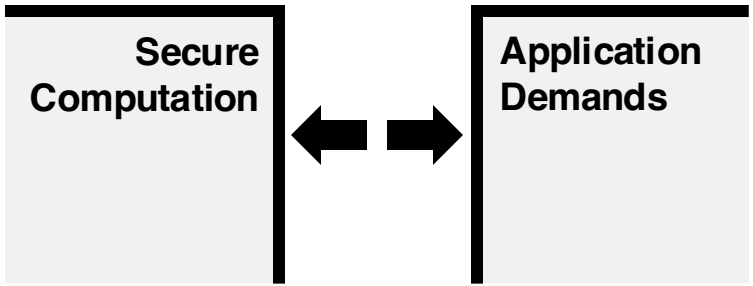
Practical for numerous applications but remains beyond reach for constrained use cases.



Complexity

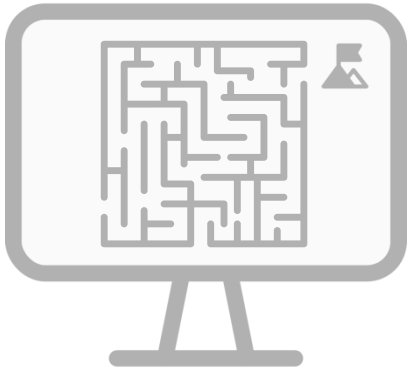
There's a gap between the capabilities of PETs today and organizations' ability to incorporate them into applications.

Theory to Practice: Barriers to Broad Adoption



Performance Gap

Practical for numerous applications but remains beyond reach for constrained use cases.

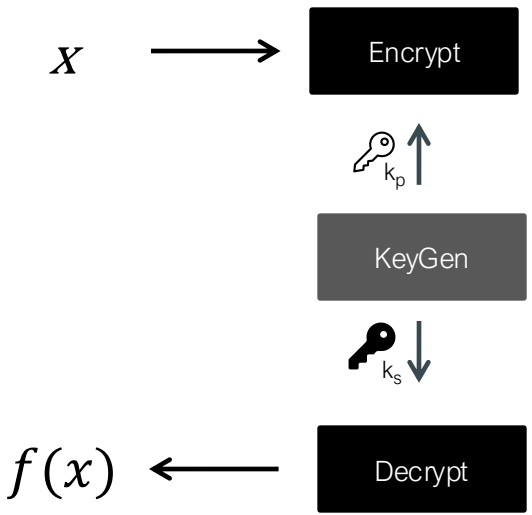


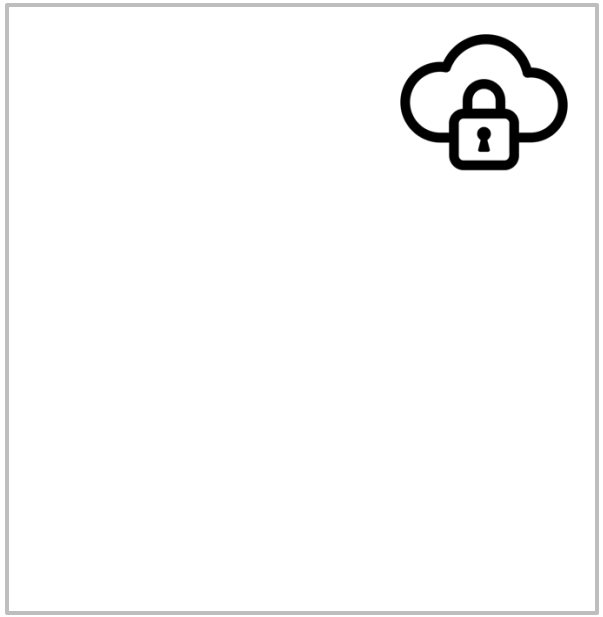
Complexity

There's a gap between the capabilities of PETs today and organizations' ability to incorporate them into applications.

Fully Homomorphic Encryption

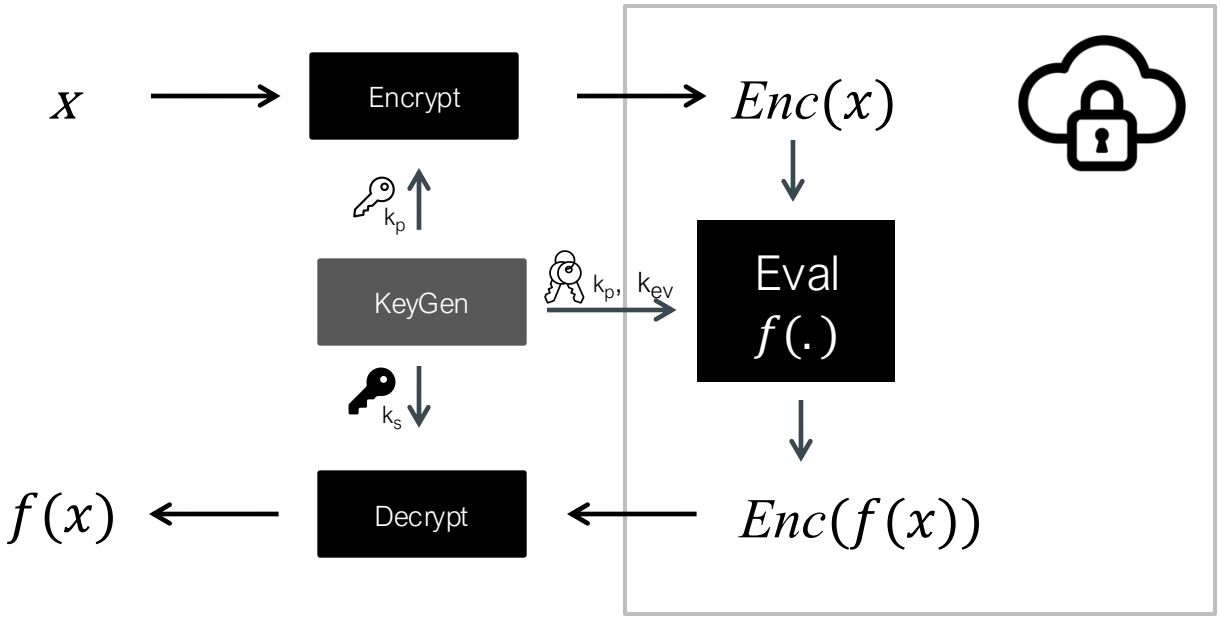
Enables computation on encrypted data





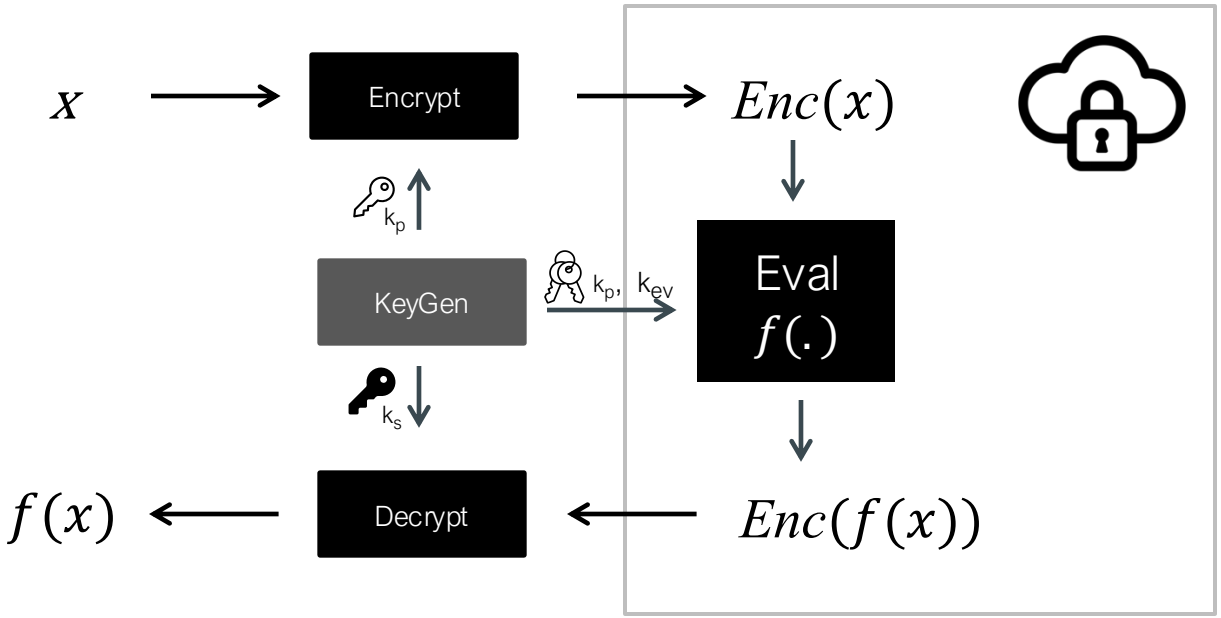
Fully Homomorphic Encryption

Enables computation on encrypted data



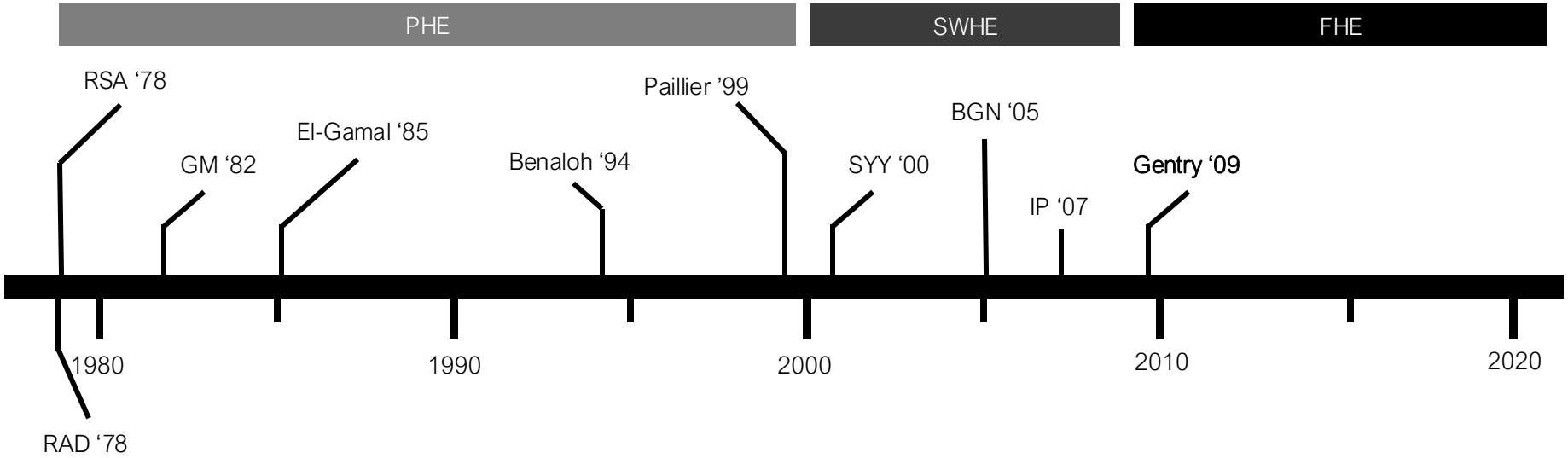
Fully Homomorphic Encryption

Enables **computation** on encrypted data

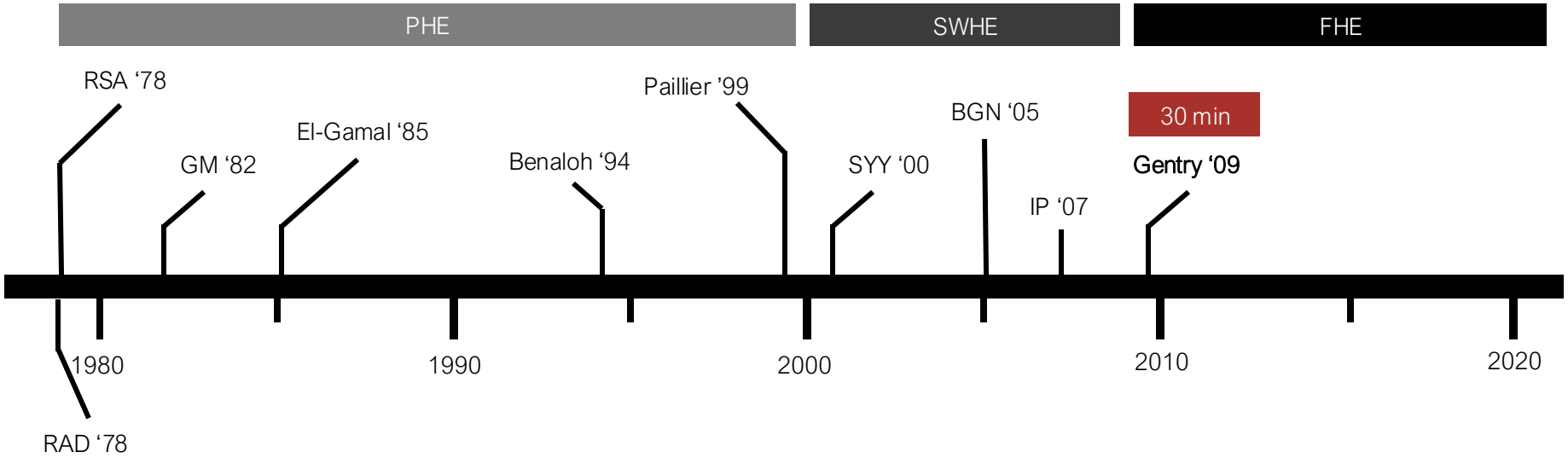


Delegate the **processing** of data without giving away **access** to it

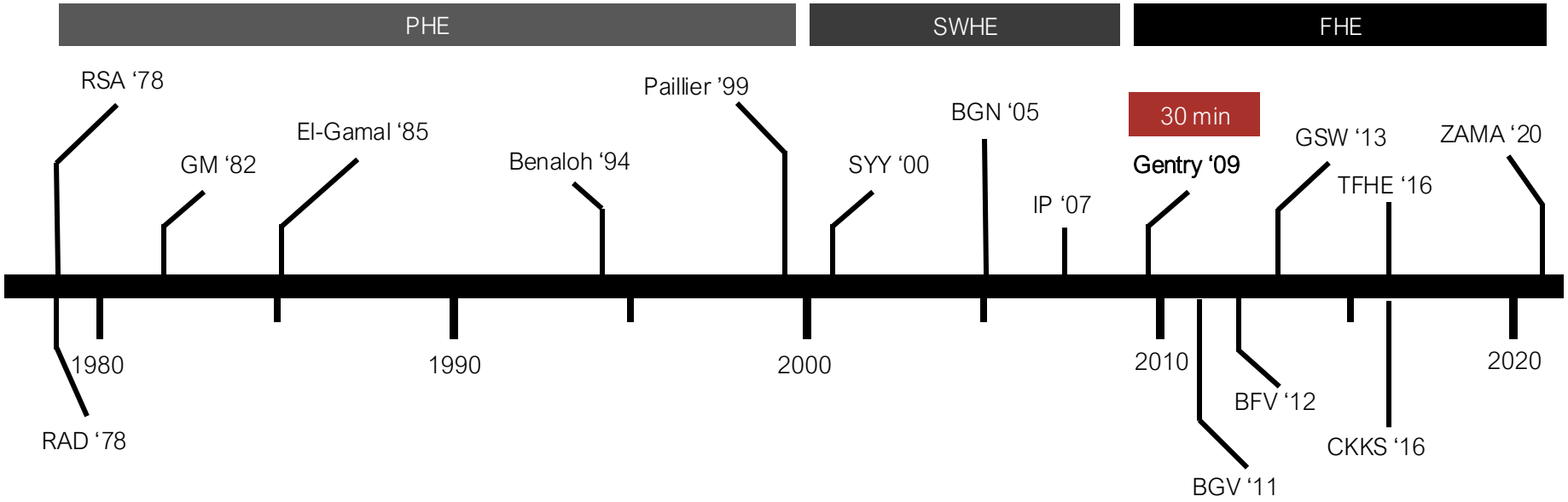
40 Years of FHE History



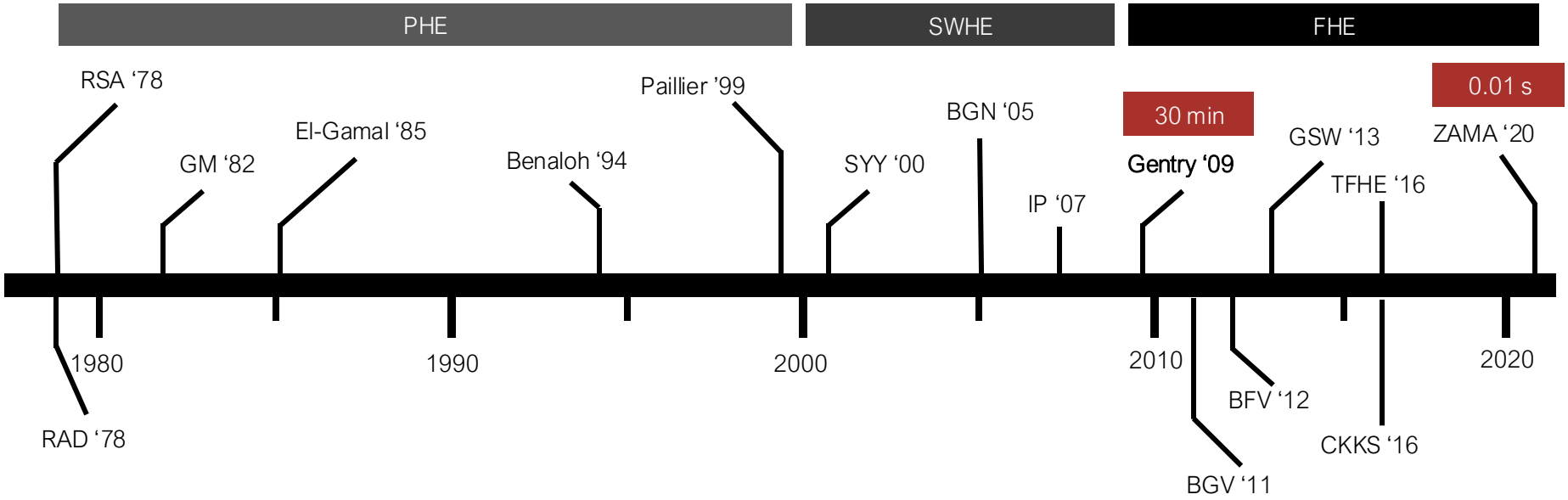
40 Years of FHE History



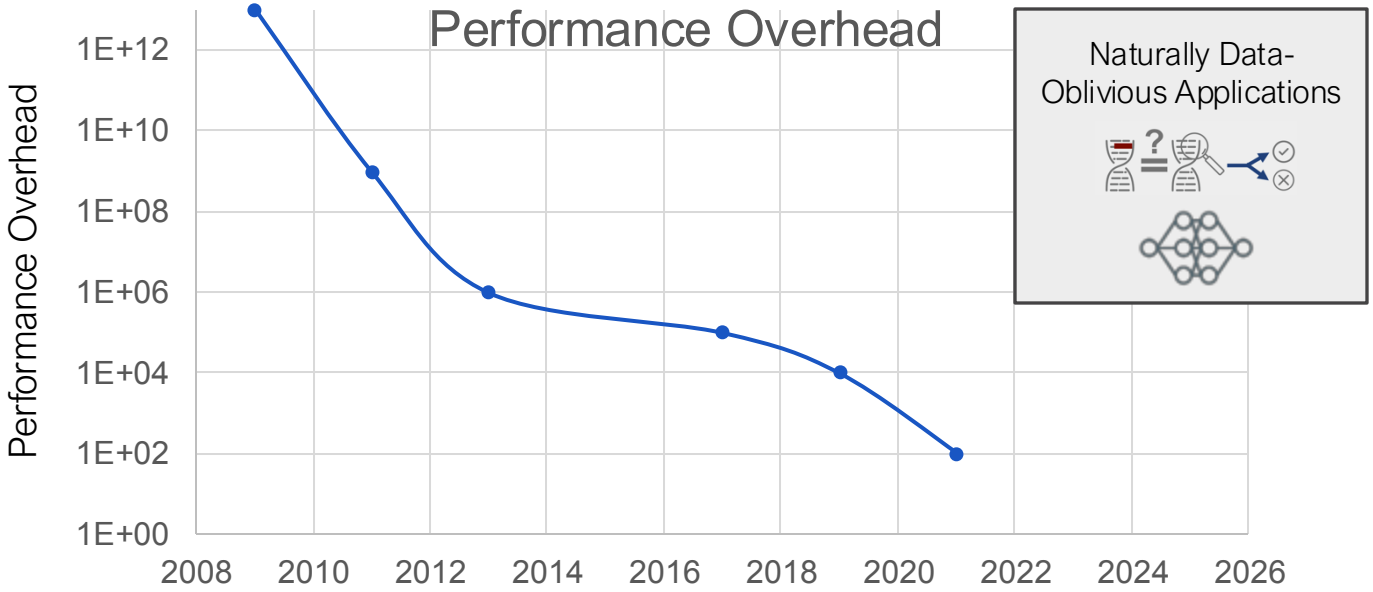
40 Years of FHE History



40 Years of FHE History



Fully Homomorphic Encryption



Graph Adapted from Kristin Lauter, Talk @PriCon2020

Real-world use Started to Emerge

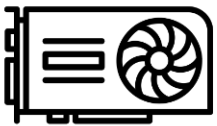
Apple Live Caller ID Lookup
(Private Information Retrieval)

Microsoft Edge Password Monitor
(Private Set Intersection)

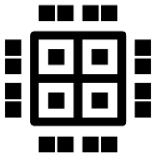
FHE Commercialization

Hardware Acceleration for FHE

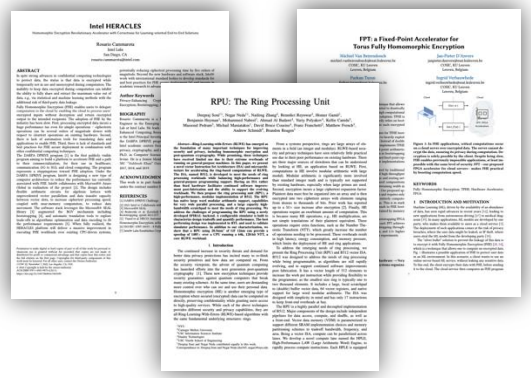
GPU



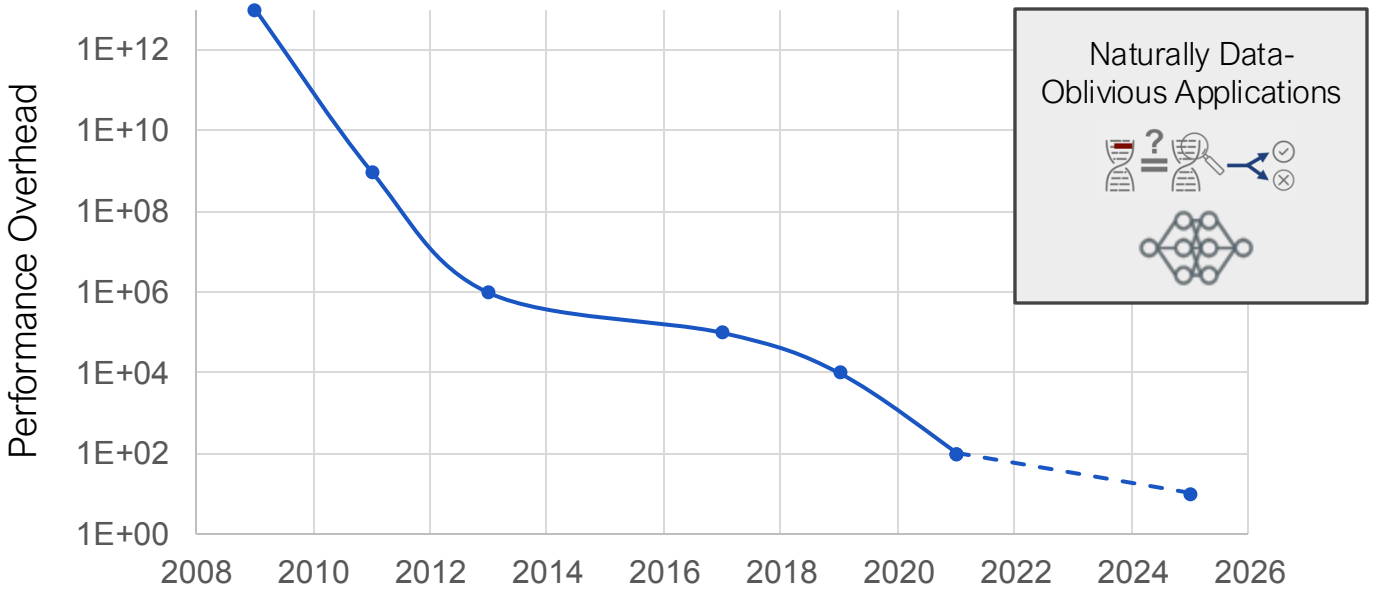
FPGA



ASIC

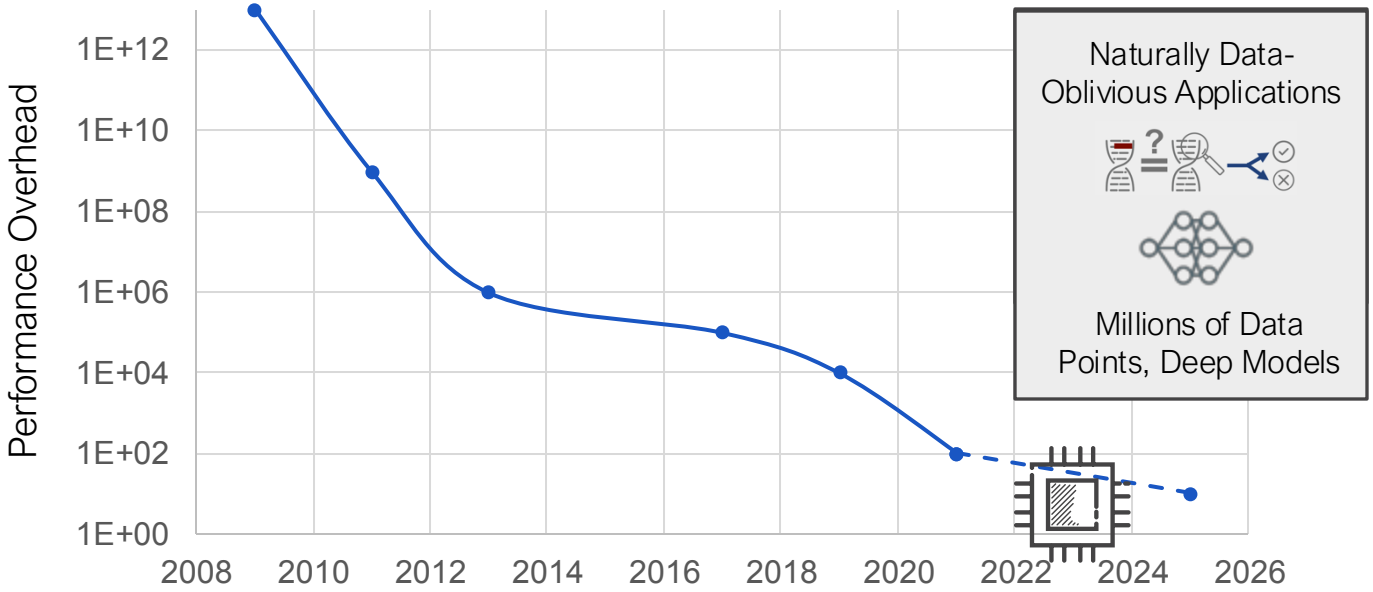


Fully Homomorphic Encryption



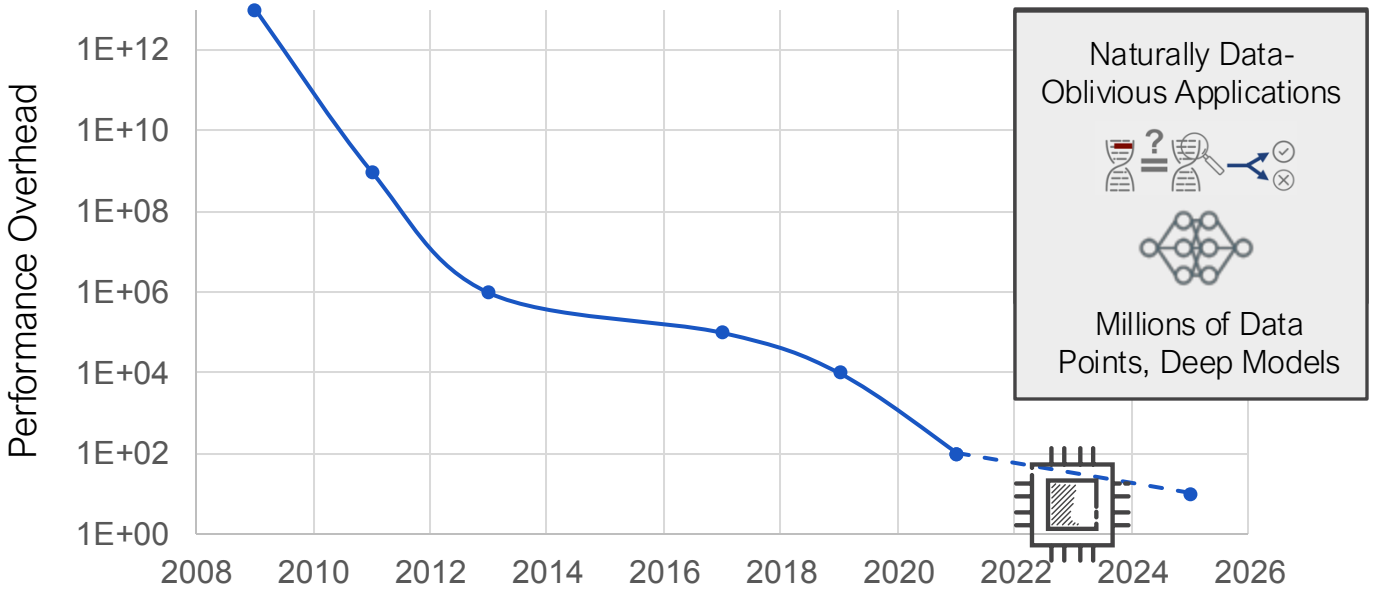
Graph Adapted from Kristin Lauter, Talk @PriCon2020

Fully Homomorphic Encryption



Graph Adapted from Kristin Lauter, Talk @PriCon2020

Performance Gap
Fully Homomorphic Encryption



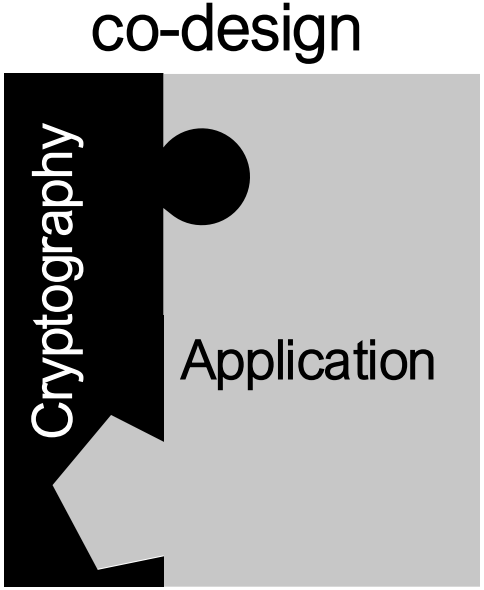
Highly Interactive Applications

Constrained Environments

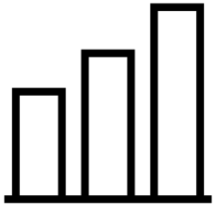
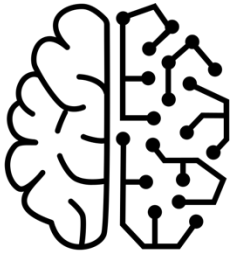
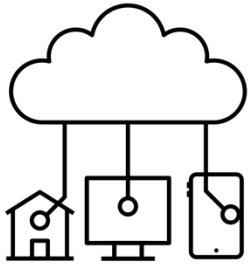
Graph Adapted from Kristin Lauter, Talk @PriCon2020

Approach to Efficiency

Empower Constrained Environments with Encrypted Data Processing.



Encrypted Data Processing Systems

DBMS	Analytics	Machine Learning	Streaming	Internet of Things
				
CryptDB	Seabed	**Arc**	**TimeCrypt**	**Talos**
Blind Seer	Seanat	Helen	**Zeph**	**Pilatus**
Arx	Conclave	**RoFL**	Waldo	Kryptein
…	…	…	…	…

…

Encrypted Data Processing Systems

DBMS

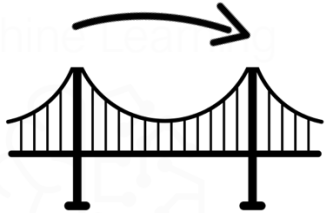
CryptDB

Blind Seer

Arx

…

Advanced Cryptography



Modern Applications

co-design

…

…

…

Internet of Things

…

Talos

Pilatus

Kryptein

…

Encrypted Data Processing Systems

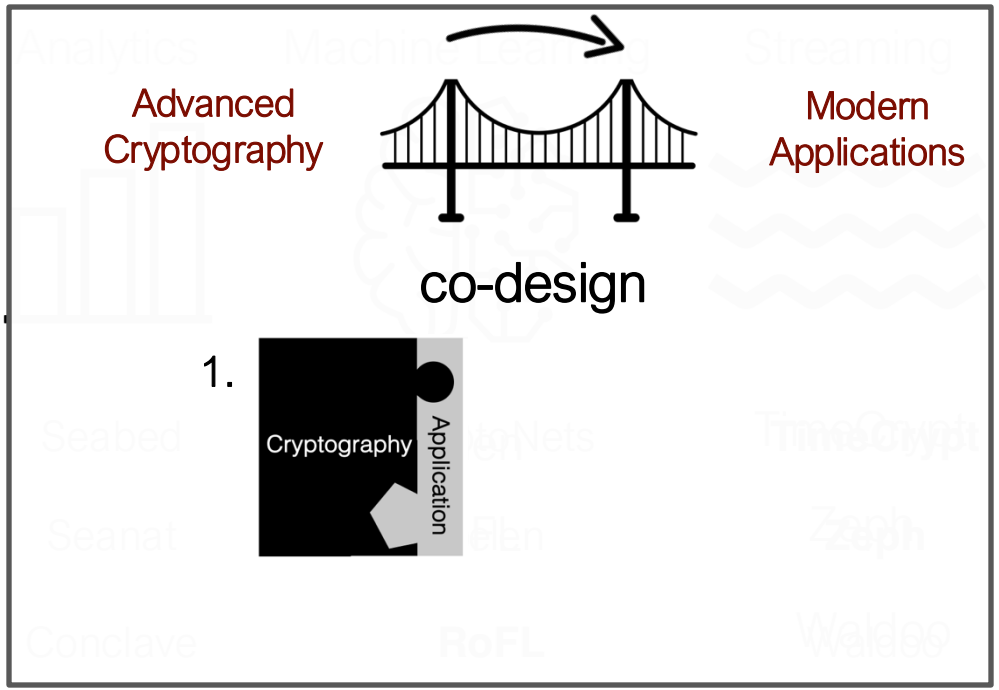
DBMS

CryptDB

Blind Seer

Arx

…

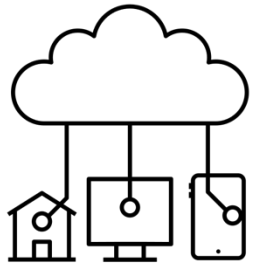


…

…

…

Internet of Things



…

Talos

Pilatus

Kryptein

…

Encrypted Data Processing Systems

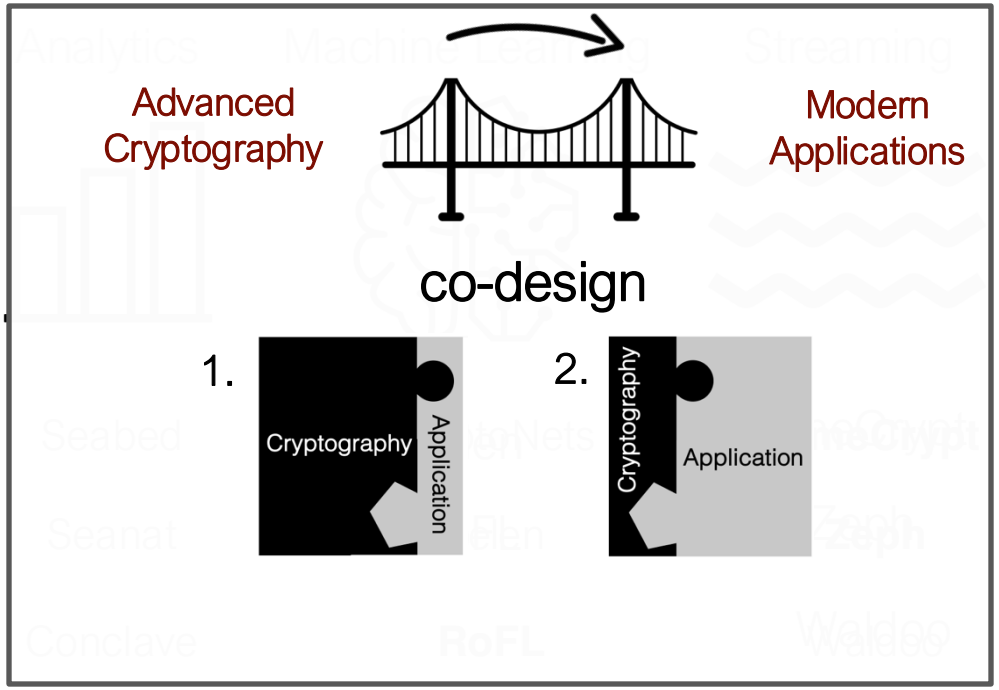
DBMS

CryptDB

Blind Seer

Arx

…



…

…

…

Internet of Things

…

Talos

Pilatus

Kryptein

…

Encrypted Data Processing Systems

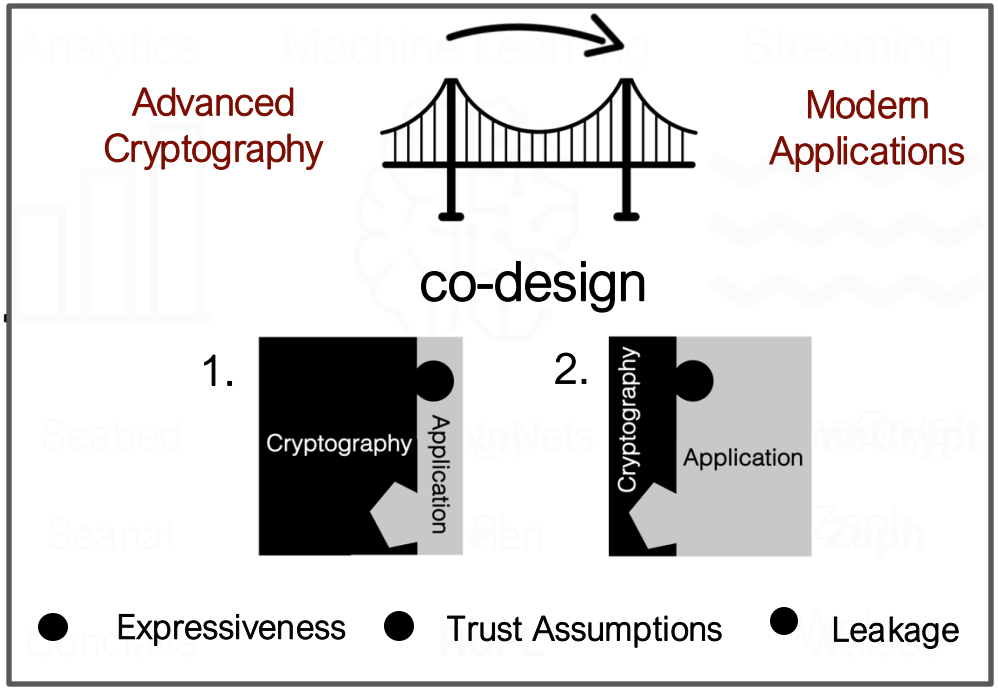
DBMS

CryptDB

Blind Seer

Arx

…



…

…

…

Internet of Things

…

Talos

Pilatus

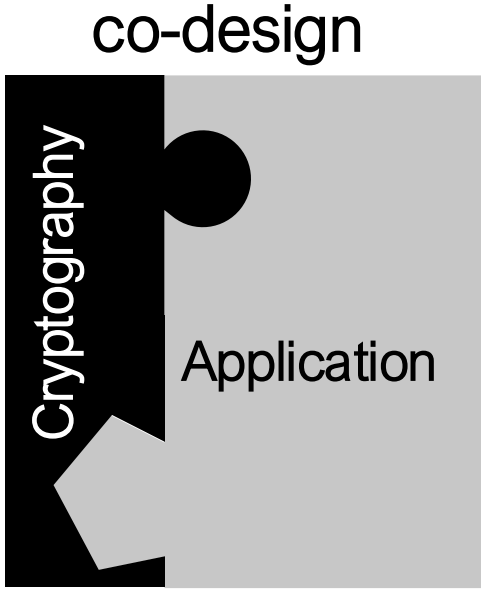
Kryptein

…

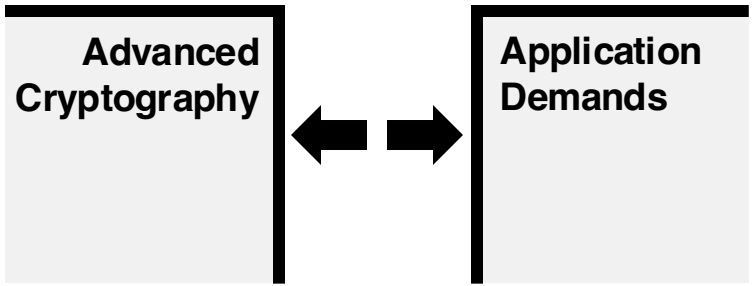
Approach to Efficiency

Pros.
- enhanced performance
- targeted functionality

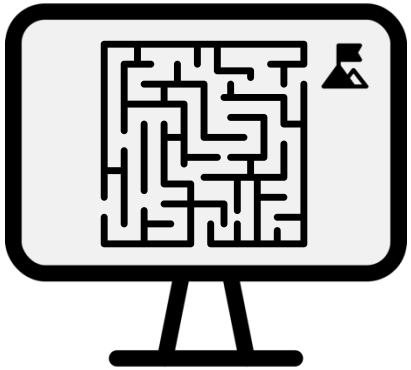
Cons.
- limited flexibility
- poor interoperability



Theory to Practice: Barriers to Broad Adoption

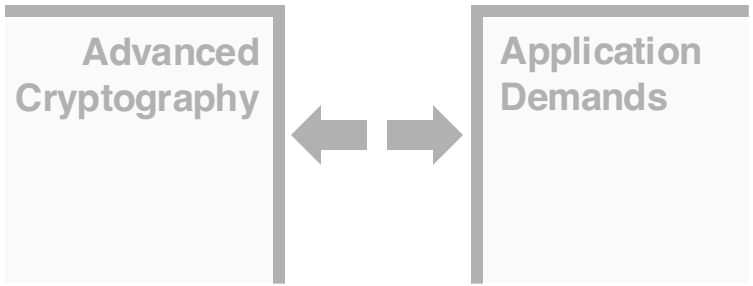


Performance Gap

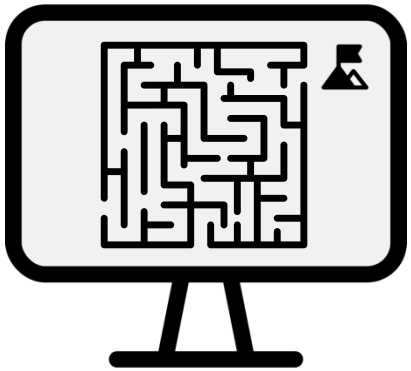


Complexity

Theory to Practice: Barriers to Broad Adoption



Performance Gap



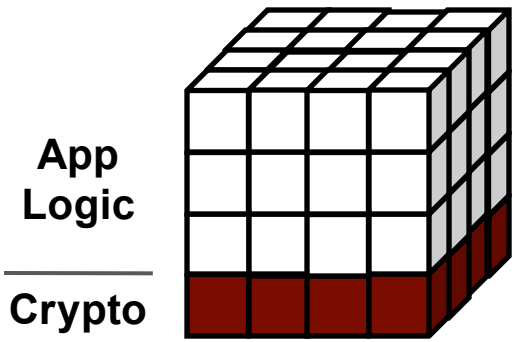
Complexity

Developing and Deploying Privacy-preserving Applications is

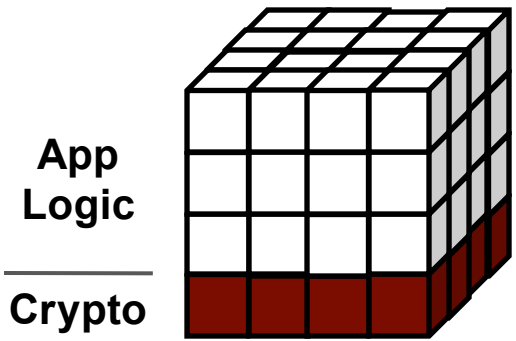
Notoriously Hard

What does “developing these applications” entail?

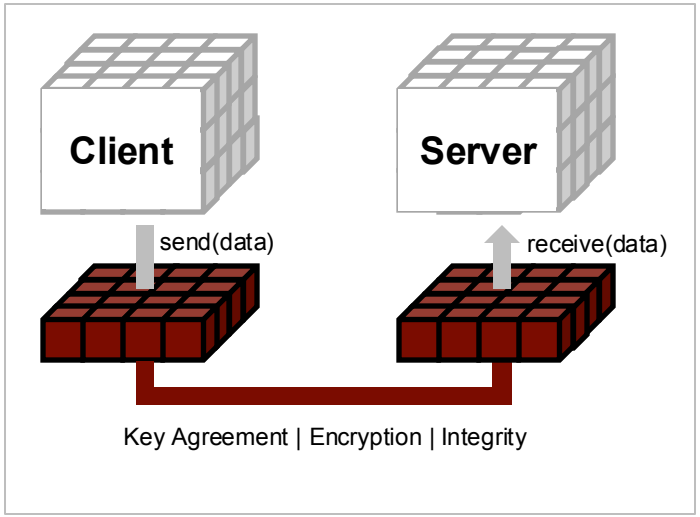
Conventional Cryptography



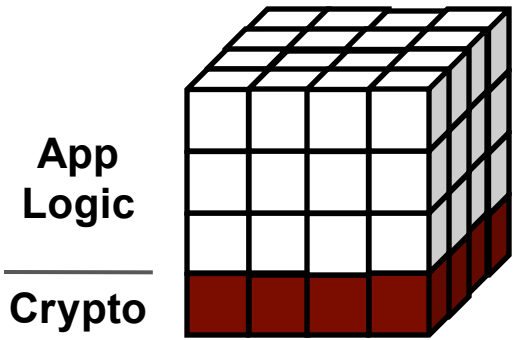
Conventional Cryptography



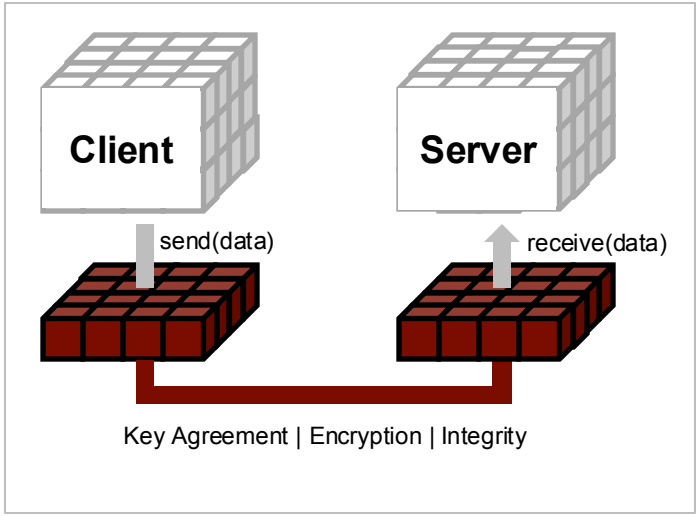
Secure Communication



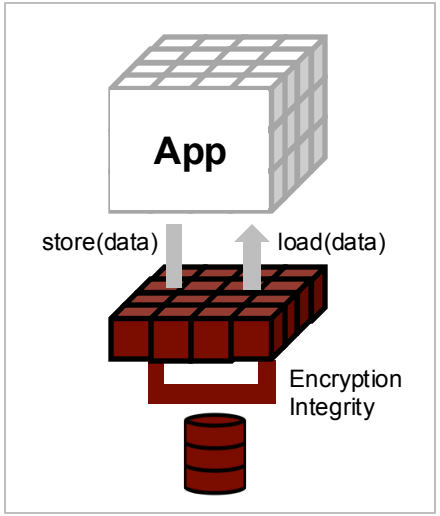
Conventional Cryptography



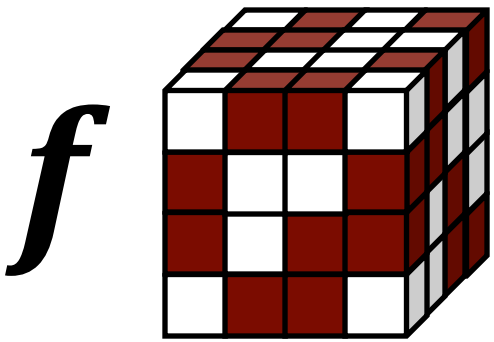
Secure Communication



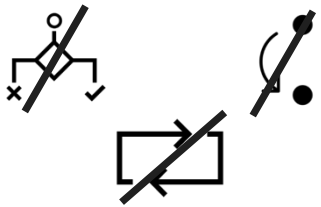
Secure Storage



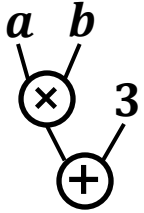
Advanced Cryptography: Secure Computation



Crypto 



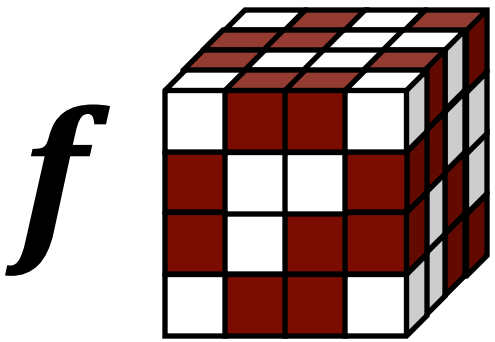
Data Oblivious

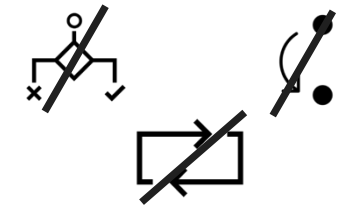


Arithmetization

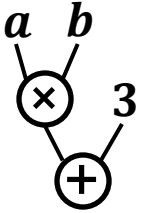
Noise

Advanced Cryptography: Secure Computation





Data Oblivious



Arithmetization

Noise

Functionality and performance depend on f's representation:

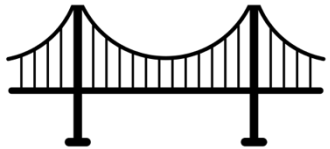
- How do we express f
- How do we optimize f

Usable Fully Homomorphic Encryption

(IEEE S&P'21, USENIX Security'23)

Usable FHE

Advanced Cryptography



Programming Languages

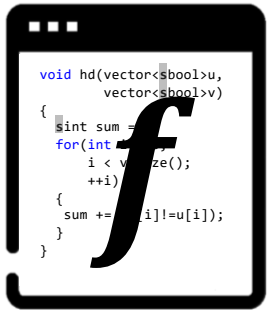
1. What makes developing FHE applications hard?

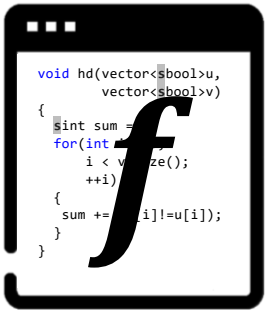
[IEEE S&P'21]

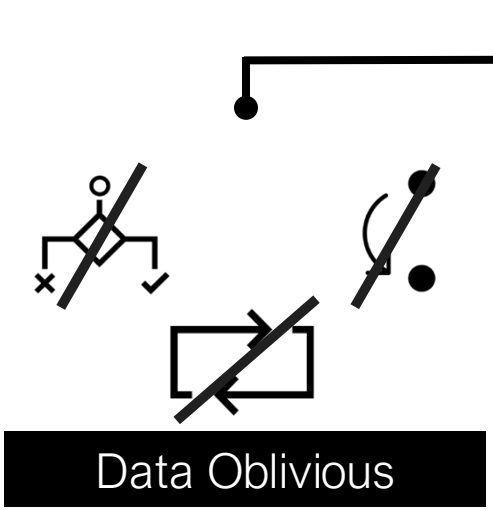
2. How can compilers address these complexities?

[USENIX Security'23]

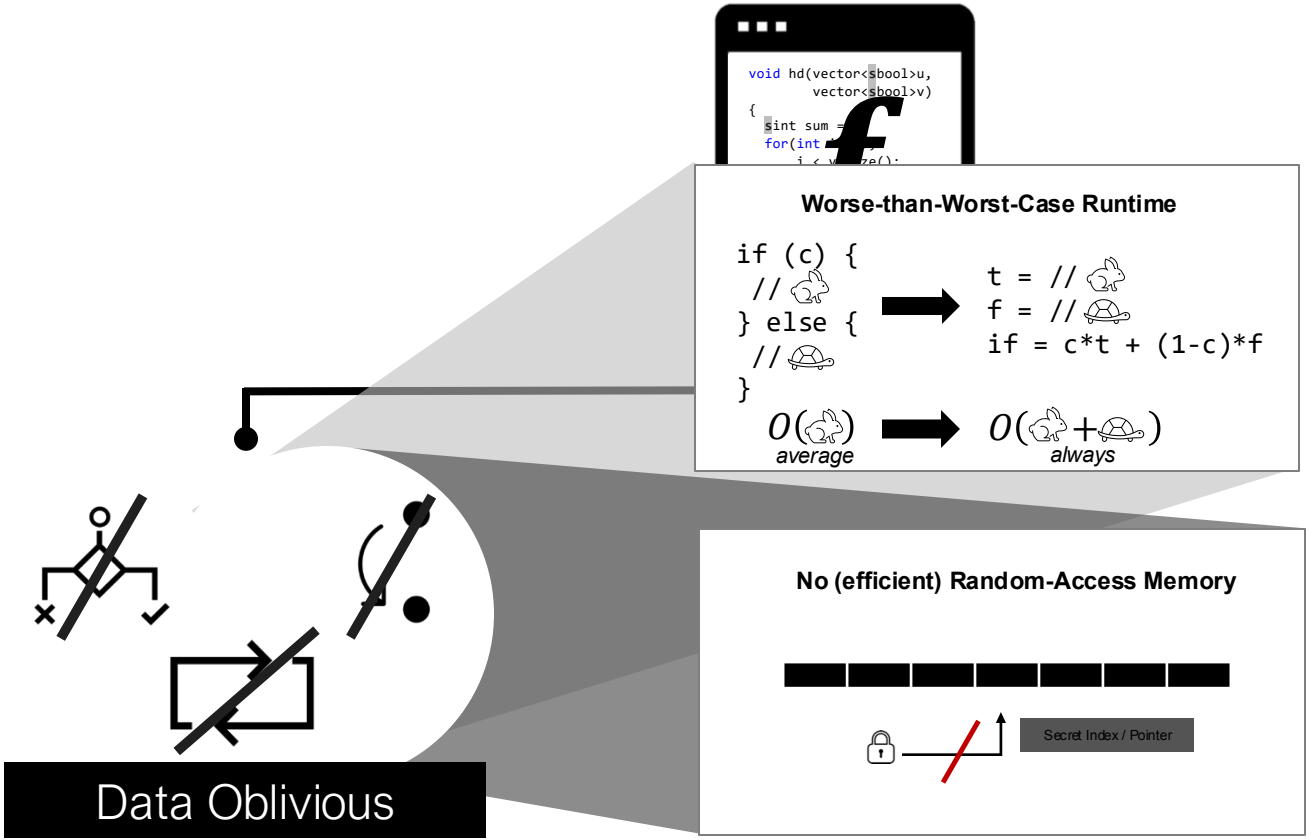
Fully Homomorphic Encryption Programming Paradigm

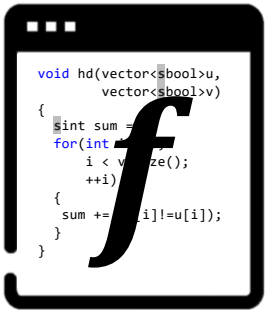


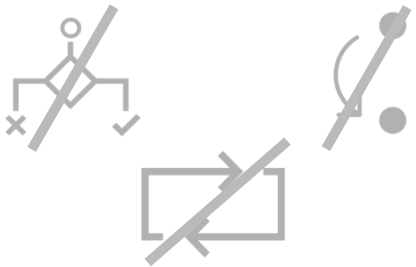




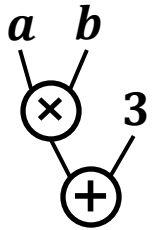
Data Oblivious


void hd(vector<sbool>u,
vector<sbool>v)
{
for(int
Worse-than-Worst-Case Runtime
if (c) {
} else {
}
t = //
f = //
if = c*t + (1-c)*f
average
always
No (efficient) Random-Access Memory
Secret Index / Pointer
Data Oblivious

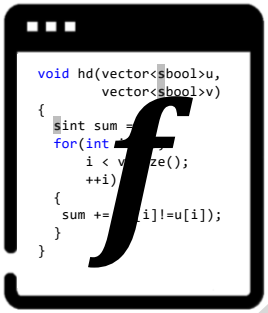


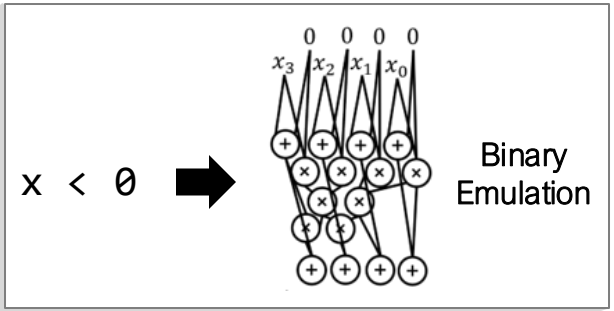


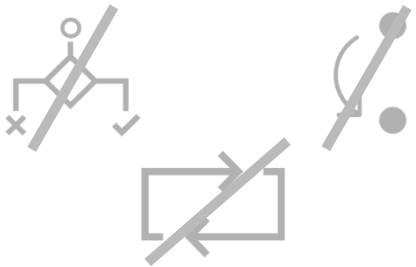
Data Oblivious



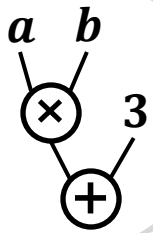
Arithmetization



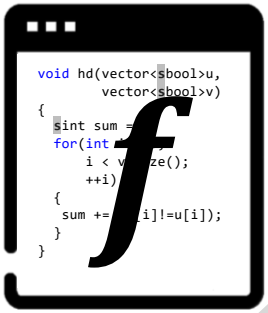


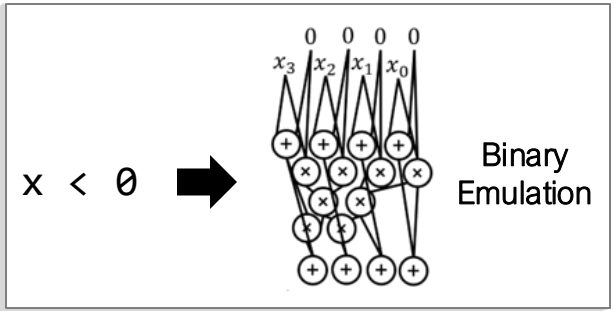


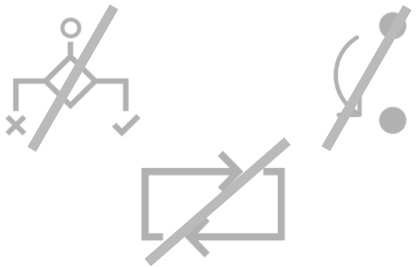
Data Oblivious



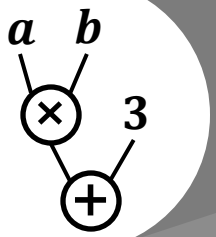
Arithmetization



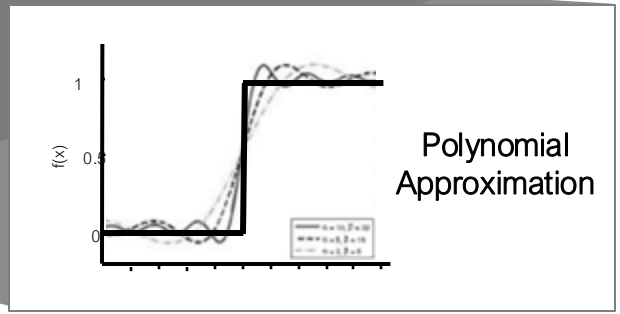


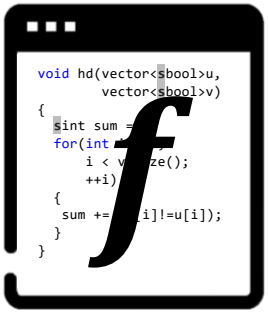


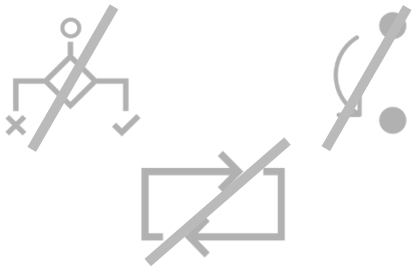
Data Oblivious



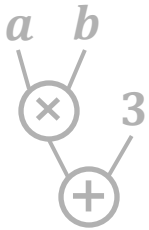
Arithmetization



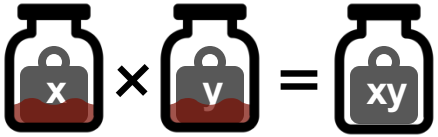




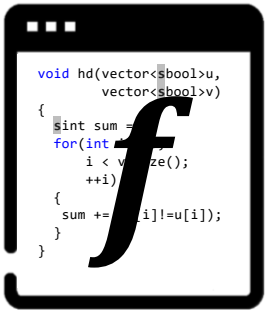
Data Oblivious

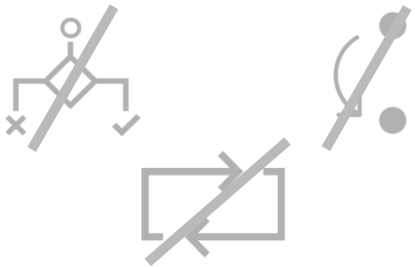


Arithmetization

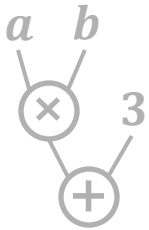


Noise Management

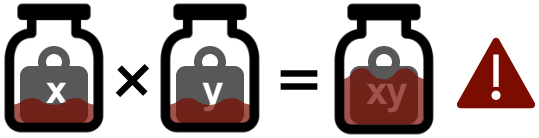




Data Oblivious



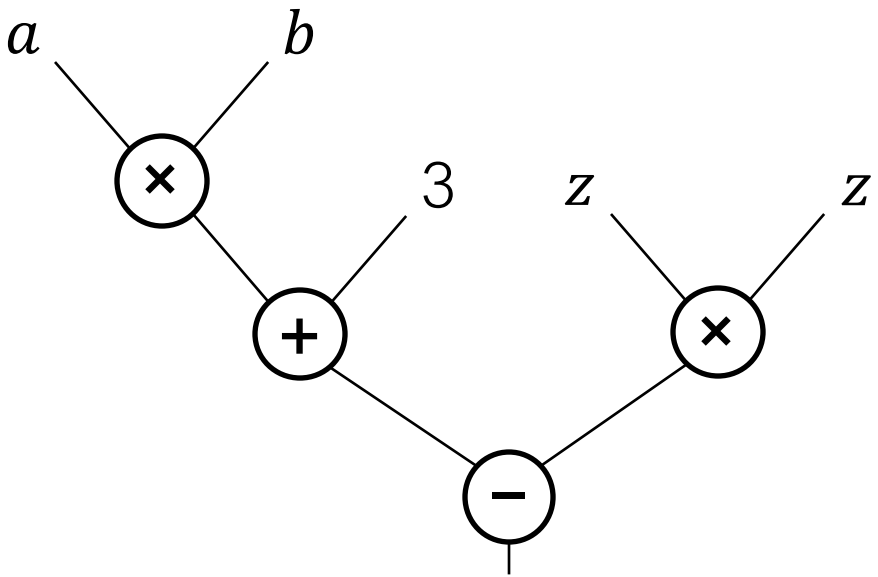
Arithmetization



Noise Management

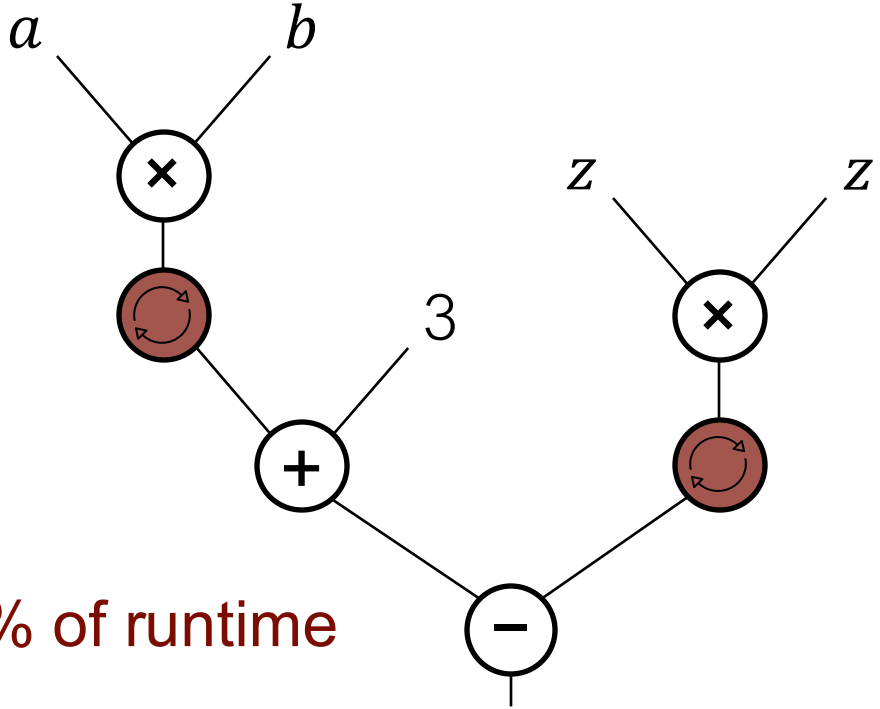
FHE Noise Management

```
void f(...)
{
 ctxt ab = a*b + 3;
 ctxt r = ab – z*z;
 return r;
}
```



FHE Noise Management

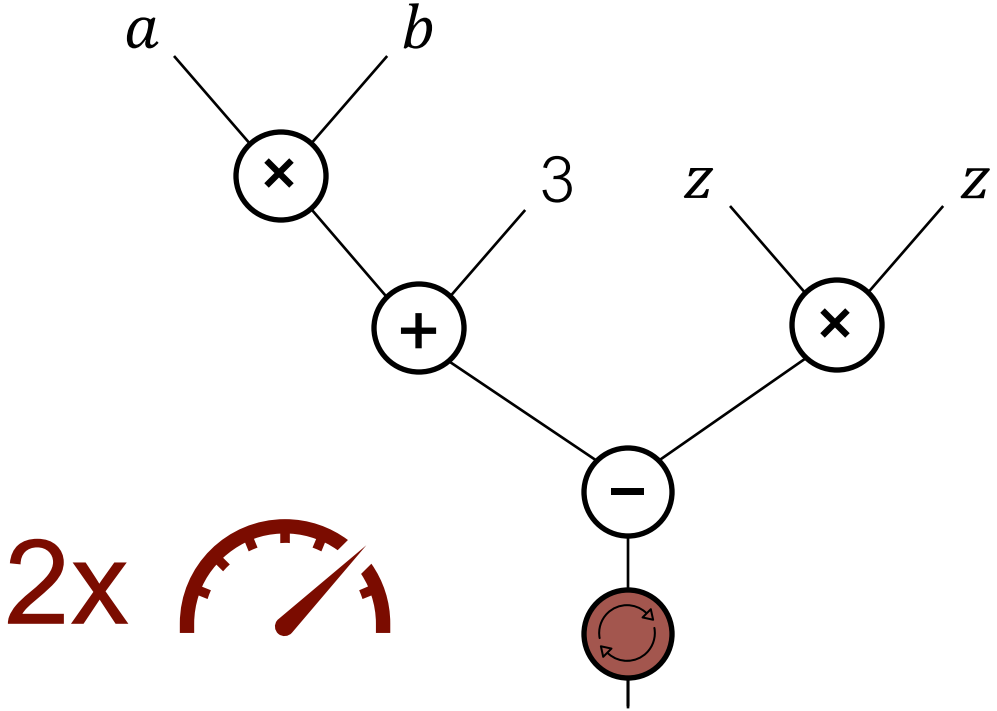
```
void f(...)
{
 ctxt ab = a*b + 3;
 ctxt r = ab - z*z;
 return r;
}
```



90% of runtime

FHE Noise Management

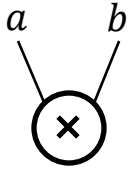
```
void f(...)
{
 ctxt ab = a*b + 3;
 ctxt r = ab – z*z;
 return r;
}
```



Accessibility
FHE Developer Tooling

Evolution of FHE Tools

Evolution of FHE Tools

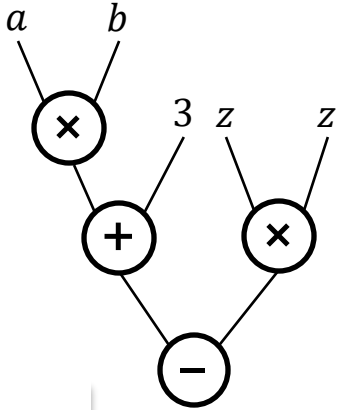


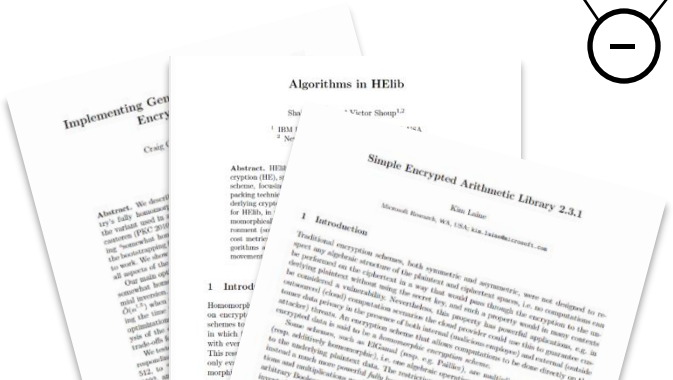
Implementing Gentry's Fully-Homomorphic Encryption Scheme

Craig Gentry* and Shai Halevi*

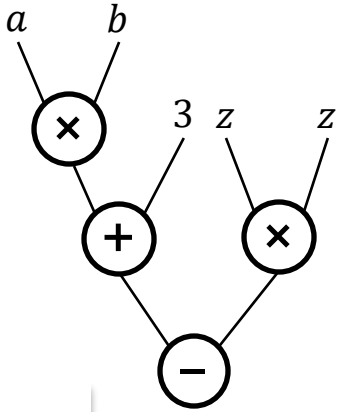
IBM Research

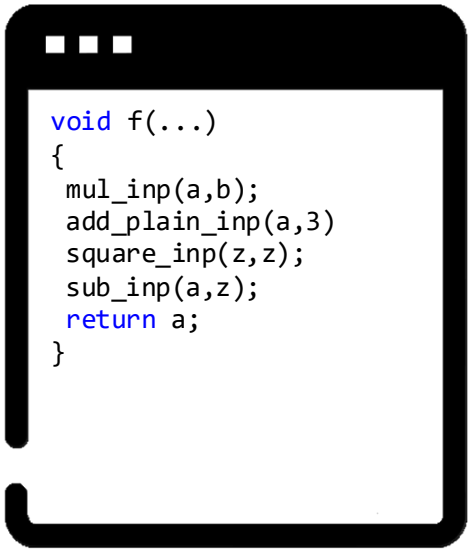
Evolution of FHE Tools



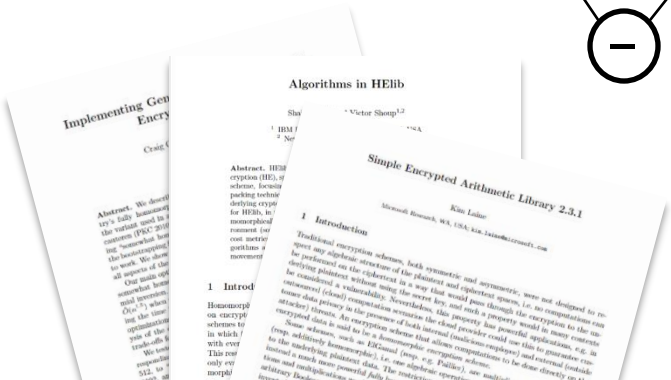


Evolution of FHE Tools

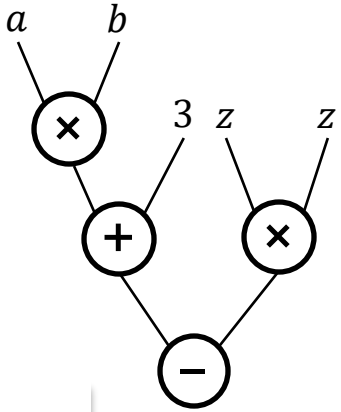


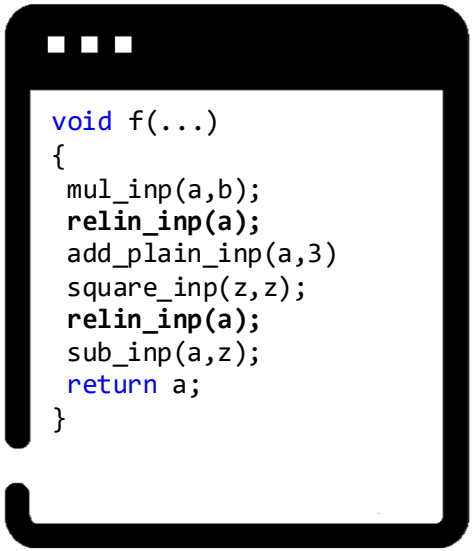


```
void f(...)
{
 mul_inp(a,b);
 add_plain_inp(a,3)
 square_inp(z,z);
 sub_inp(a,z);
 return a;
}
```

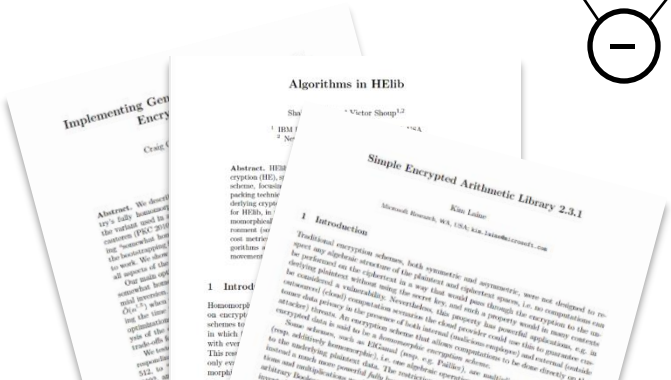


Evolution of FHE Tools

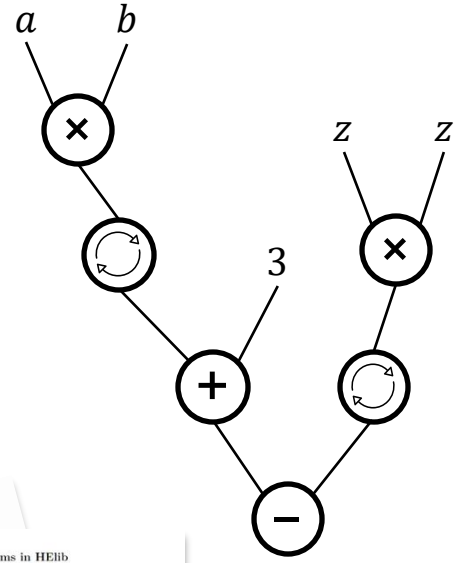




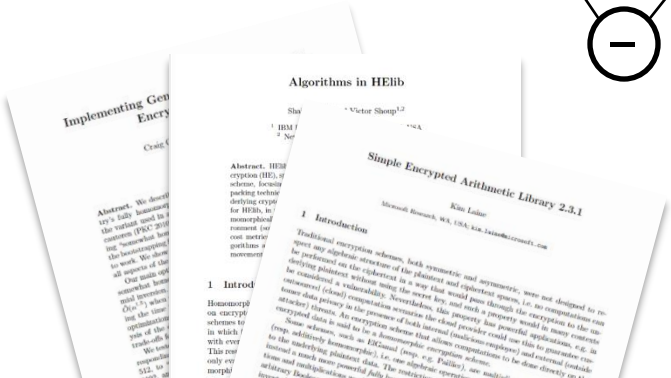
```
void f(...)
{
 mul_inp(a,b);
 relin_inp(a);
 add_plain_inp(a,3)
 square_inp(z,z);
 relin_inp(a);
 sub_inp(a,z);
 return a;
}
```



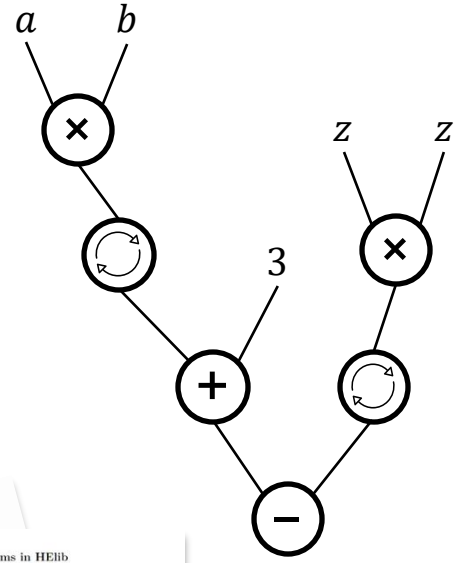
Evolution of FHE Tools

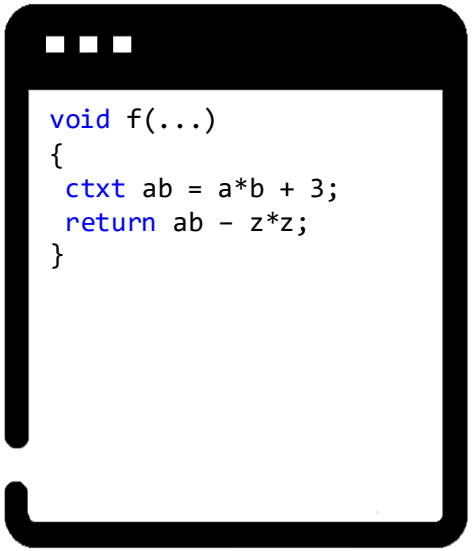



```
void f(...)
{
 mul_inp(a,b);
 relin_inp(a);
 add_plain_inp(a,3)
 square_inp(z,z);
 relin_inp(a);
 sub_inp(a,z);
 return a;
}
```

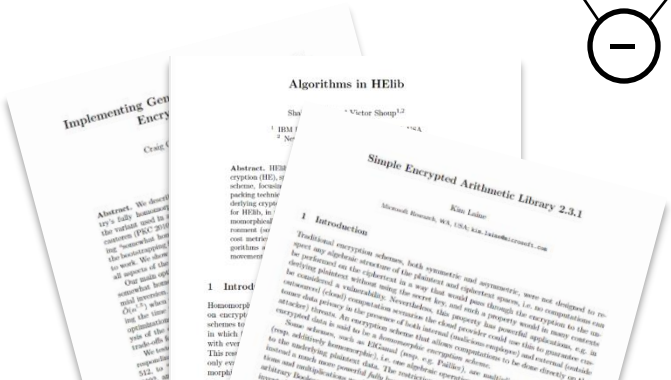


Evolution of FHE Tools

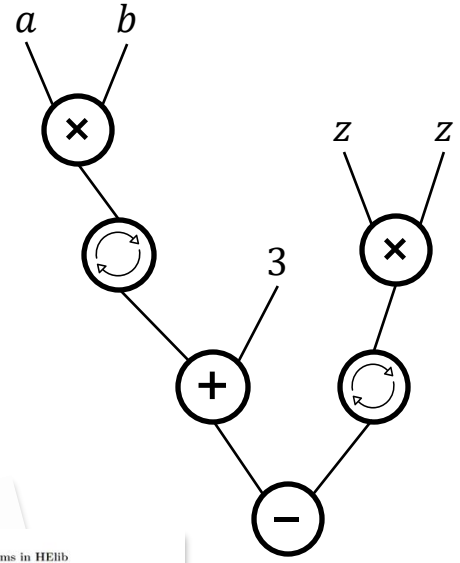




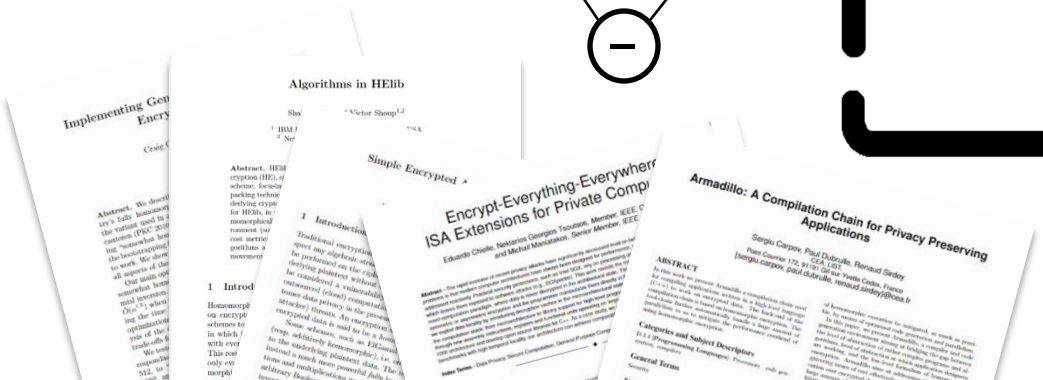
```
void f(...)
{
 ctxt ab = a*b + 3;
 return ab – z*z;
}
```



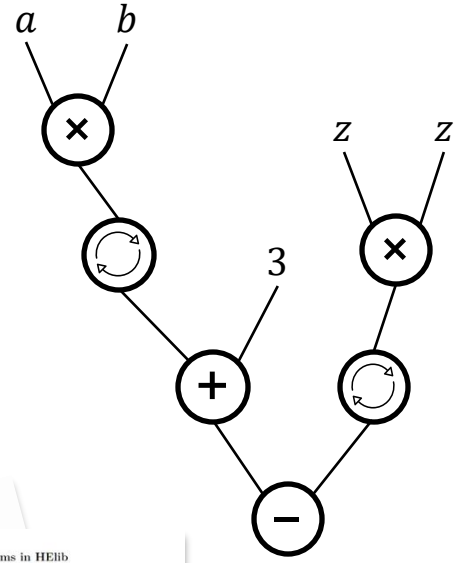
Evolution of FHE Tools



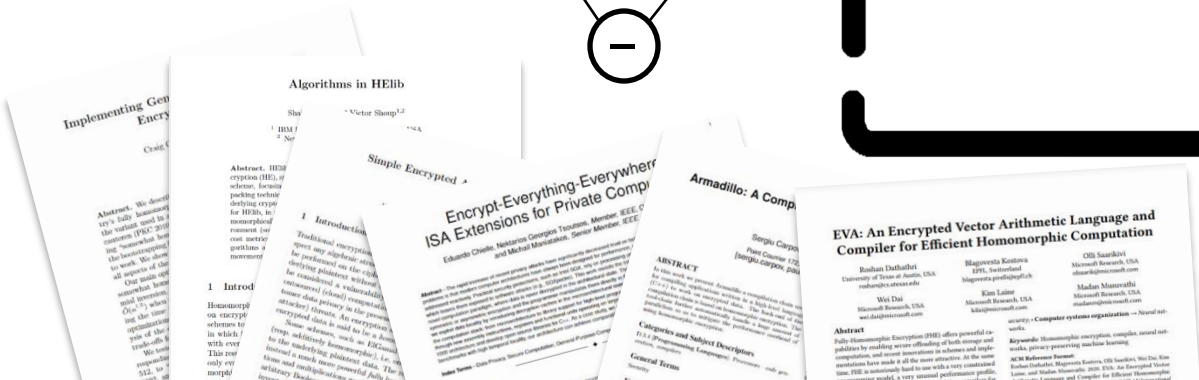

```
void f(...)
{
 ctxt ab = a*b + 3;
 return ab – z*z;
}
```



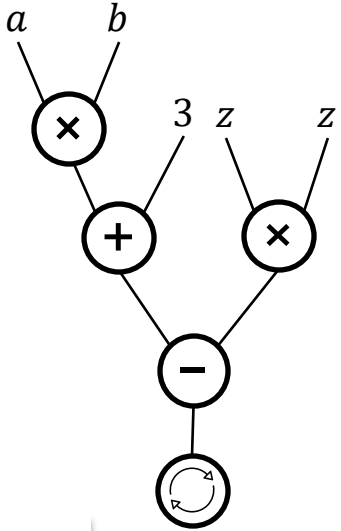
Evolution of FHE Tools



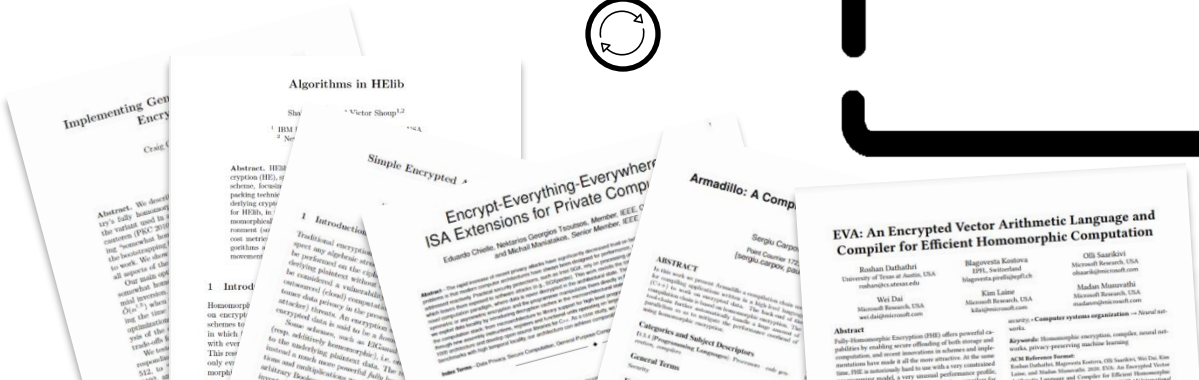

```
void f(...)
{
 ctxt ab = a*b + 3;
 return ab – z*z;
}
```



Evolution of FHE Tools

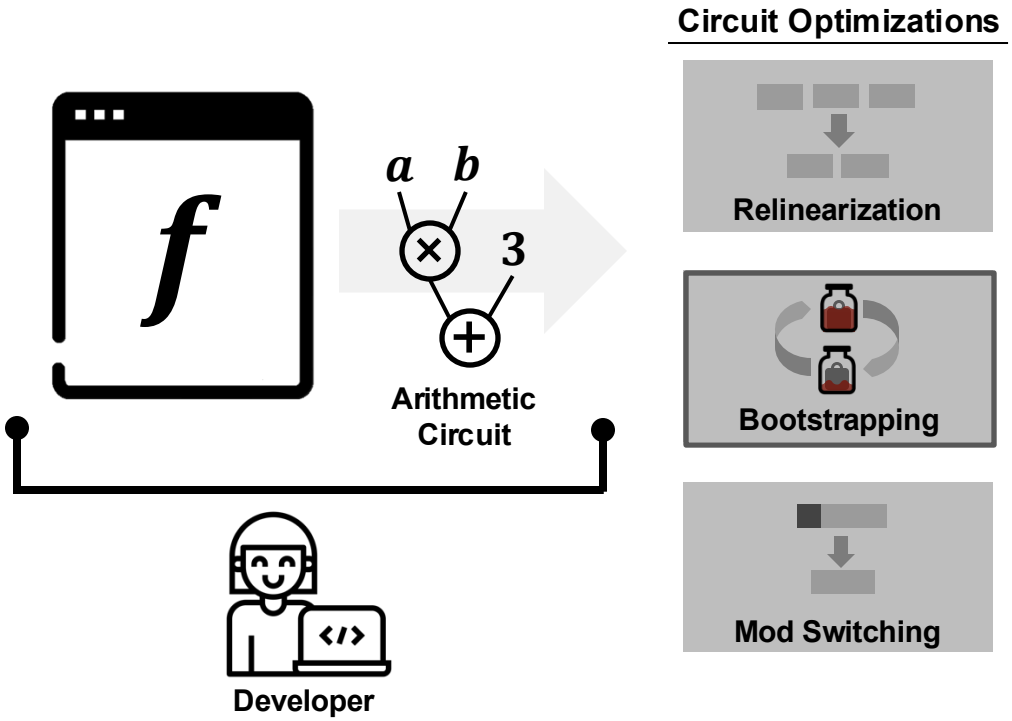



```
void f(...)
{
 ctxt ab = a*b + 3;
 return ab – z*z;
}
```

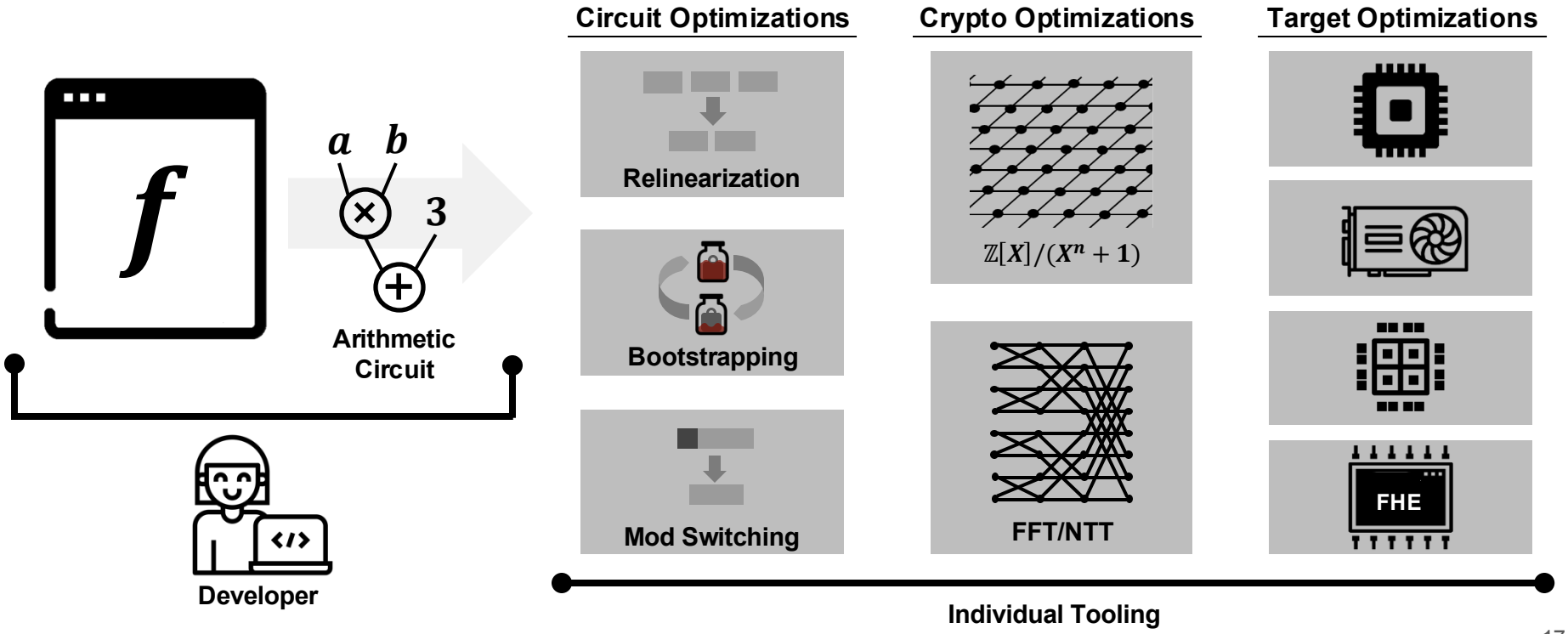


Existing tools make important contributions,
but are very **narrowly focussed**

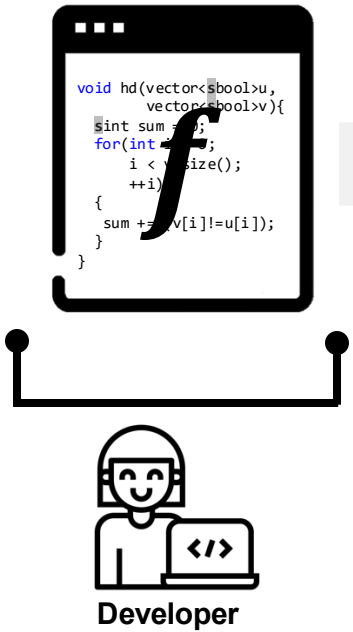
Developing FHE Applications

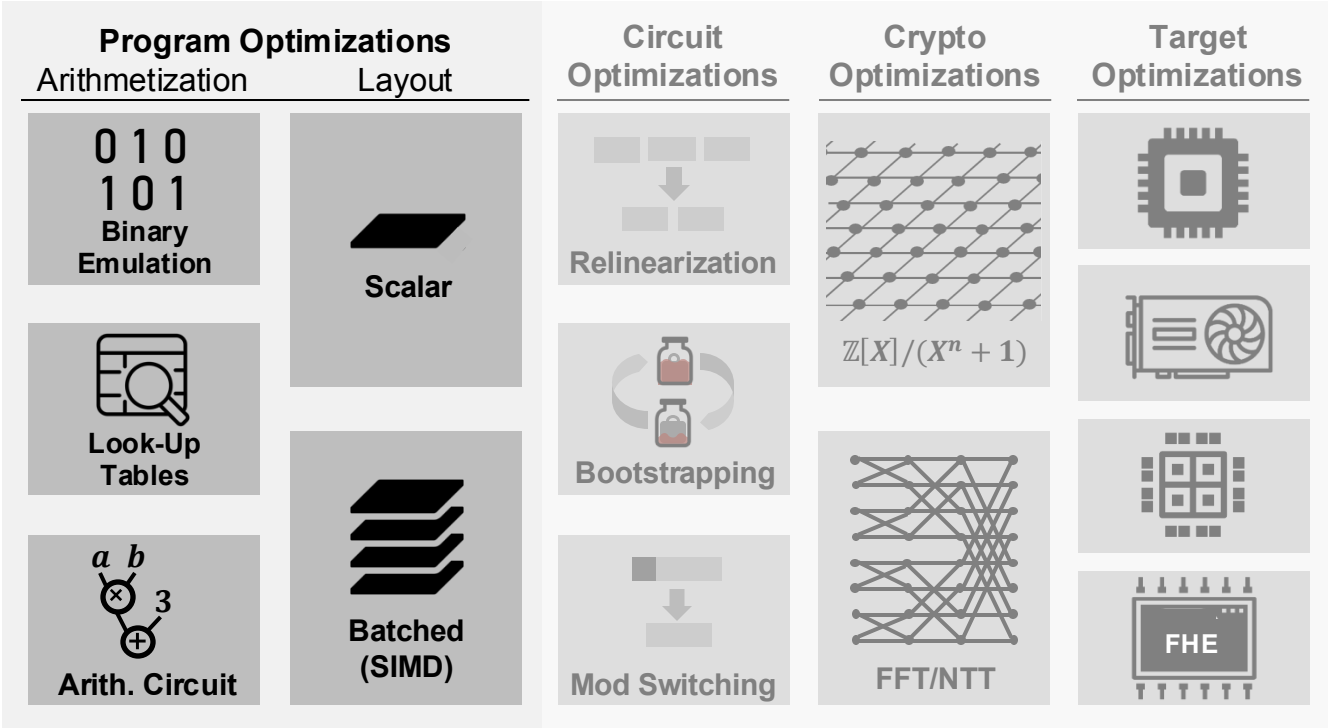


Developing FHE Applications



HECO


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


Program Optimizations
Arithmetization
Layout
0 1 0
1 0 1
Binary Emulation
Look-Up Tables
a b
3
Arith. Circuit
Scalar
Batched (SIMD)
Circuit Optimizations
Relinearization
Bootstrapping
Mod Switching
Crypto Optimizations
$\mathbb{Z}[X]/(X^n+1)$
FFT/NTT
Target Optimizations
FHE

SIMD-like Parallelism

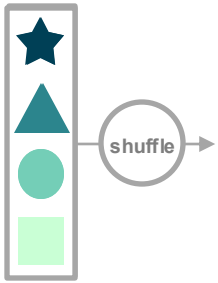
Standard C++

```
int[] foo(int[] x,int[]
y){

   int[] r;
   for(i = 0; i < 6; ++i){
      r[i] = x[i] * y[i]
   }
   return r;

}
```

Batched FHE

```
int[] foo(int[] a,int[]
b){

   return a * b;

}
```

SIMD Batching



No efficient free permutation or scatter/gather

SIMD-like Parallelism

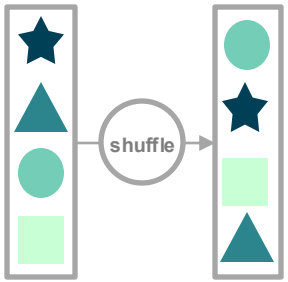
Standard C++

```
int[] foo(int[] x,int[]
y){

   int[] r;
   for(i = 0; i < 6; ++i){
      r[i] = x[i] * y[i]
   }
   return r;

}
```

Batched FHE

```
int[] foo(int[] a,int[]
b){

   return a * b;

}
```

SIMD Batching



No efficient free permutation or scatter/gather

Only cyclical rotations

SIMD-like Parallelism

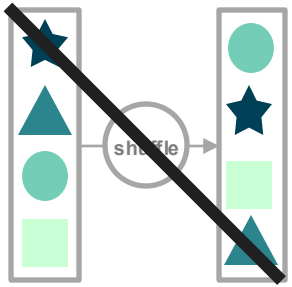
Standard C++

```
int[] foo(int[] x,int[]
y){

   int[] r;
   for(i = 0; i < 6; ++i){
      r[i] = x[i] * y[i]
   }
   return r;

}
```

Batched FHE

```
int[] foo(int[] a,int[]
b){

   return a * b;

}
```

SIMD Batching



No efficient free permutation or scatter/gather

Only cyclical rotations

SIMD-like Parallelism

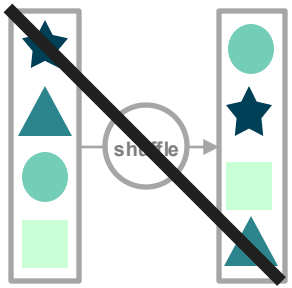
Standard C++

```
int[] foo(int[] x,int[]
y){

   int[] r;
   for(i = 0; i < 6; ++i){
      r[i] = x[i] * y[i]
   }
   return r;

}
```

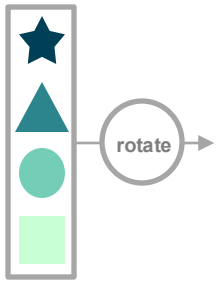
Batched FHE

```
int[] foo(int[] a,int[]
b){

   return a * b;

}
```

SIMD Batching



No efficient free permutation or scatter/gather



Only cyclical rotations

SIMD-like Parallelism

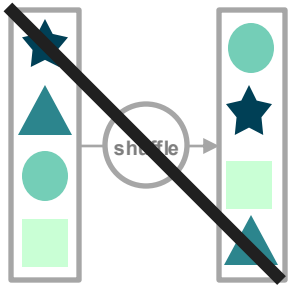
Standard C++

```
int[] foo(int[] x,int[]
y){

   int[] r;
   for(i = 0; i < 6; ++i){
      r[i] = x[i] * y[i]
   }
   return r;

}
```

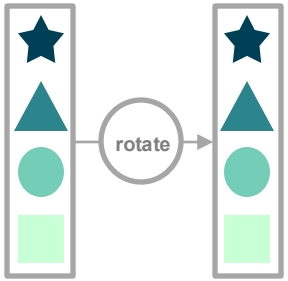
Batched FHE

```
int[] foo(int[] a,int[]
b){

   return a * b;

}
```

SIMD Batching



No efficient free permutation or scatter/gather



Only cyclical rotations

SIMD-like Parallelism

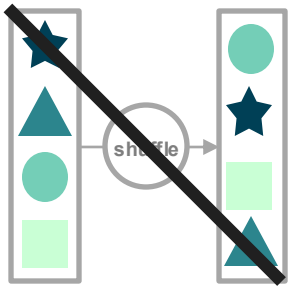
Standard C++

```
int[] foo(int[] x,int[]
y){

   int[] r;
   for(i = 0; i < 6; ++i){
      r[i] = x[i] * y[i]
   }
   return r;

}
```

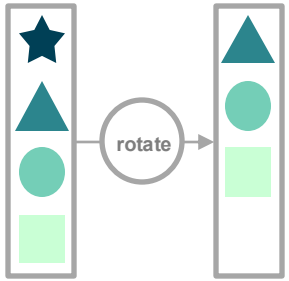
Batched FHE

```
int[] foo(int[] a,int[]
b){

   return a * b;

}
```

SIMD Batching



No efficient free permutation or scatter/gather



Only cyclical rotations

SIMD-like Parallelism

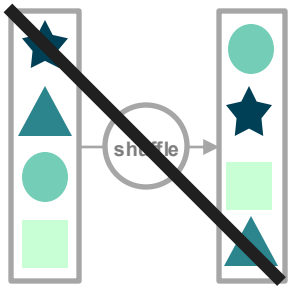
Standard C++

```
int[] foo(int[] x,int[]
y){

   int[] r;
   for(i = 0; i < 6; ++i){
      r[i] = x[i] * y[i]
   }
   return r;

}
```

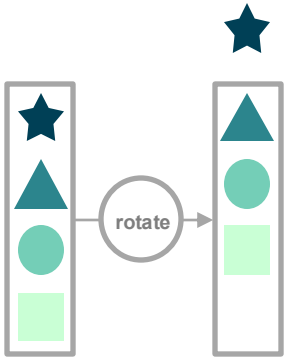
Batched FHE

```
int[] foo(int[] a,int[]
b){

   return a * b;

}
```

SIMD Batching

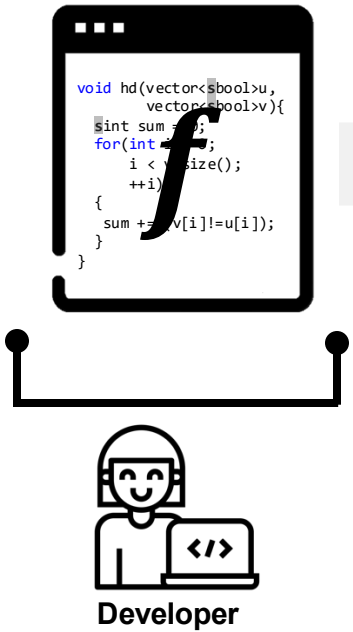


No efficient free permutation or scatter/gather



Only cyclical rotations

HECO: Transform High-level Programs to Efficient FHE Solutions



Program Optimizations

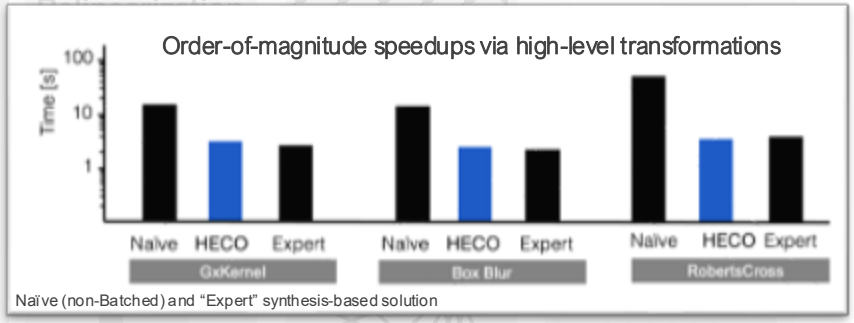
Arithmetization: 0 1 0 / 1 0 1 Binary Emulation; Look-Up Tables; $a\ b$ / $\times$ / $+$ / 3 Arith. Circuit

Layout: Scalar; Batched (SIMD)

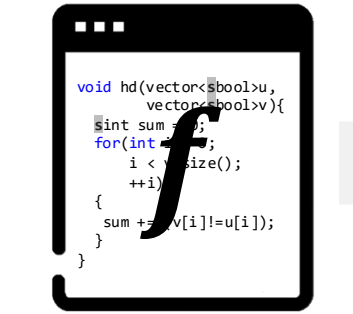
Circuit Optimizations: Mod Switching

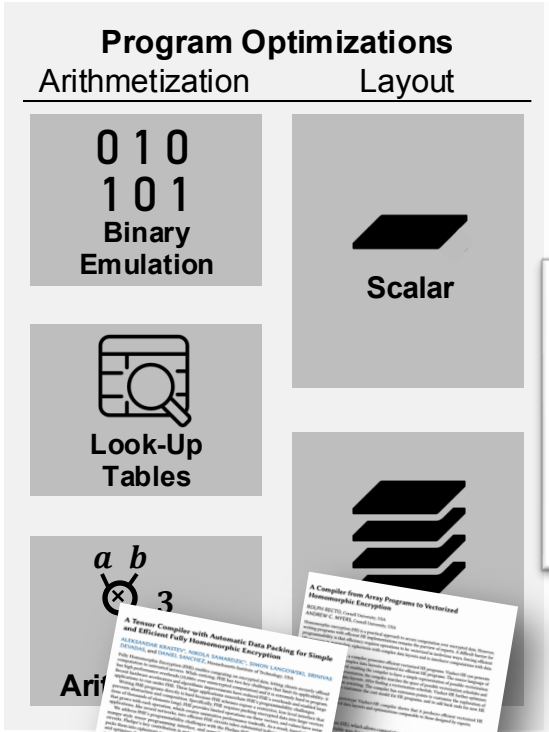
Crypto Optimizations: FFT/NTT

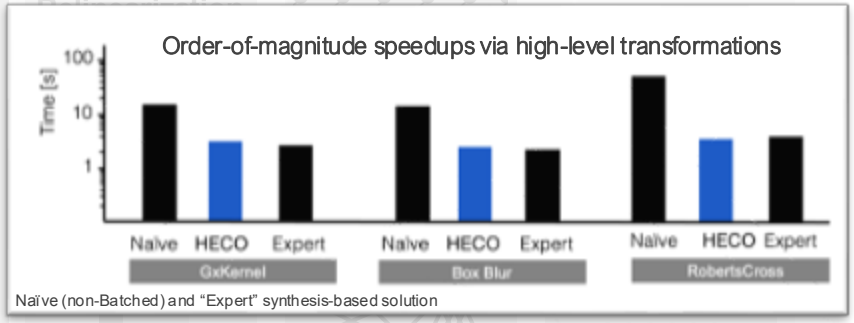
Target Optimizations: FHE



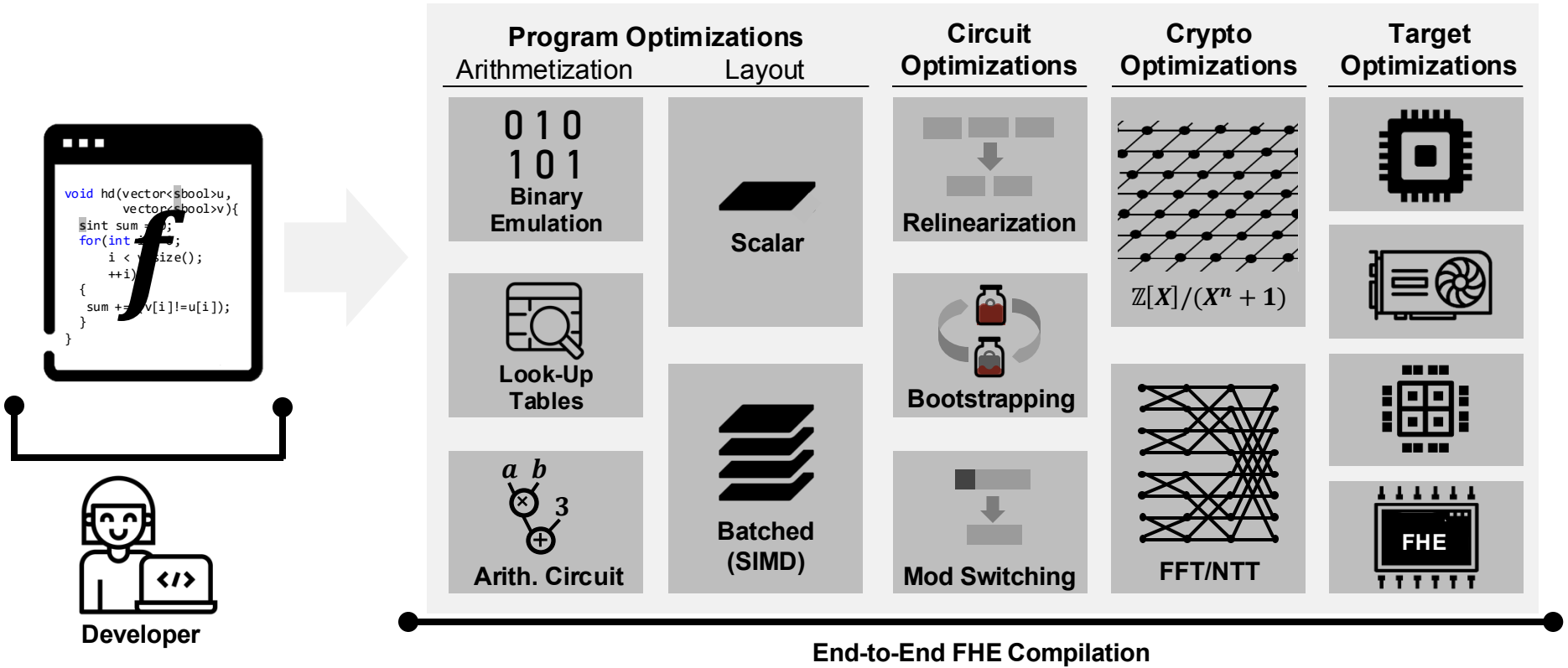
HECO: Transform High-level Programs to Efficient FHE Solutions







HECO: End-to-End FHE Compilation



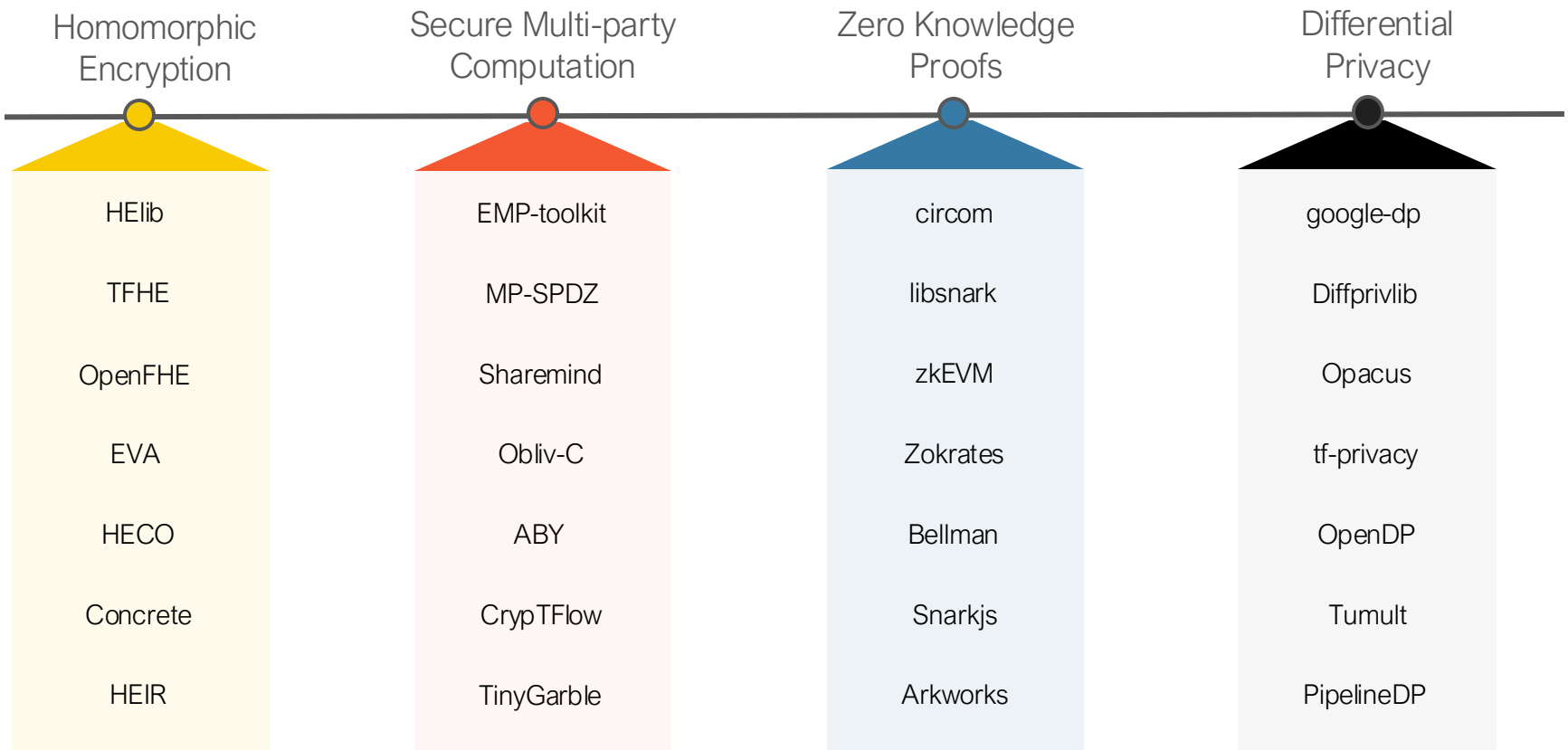
Standardizing the FHE Ecosystem

HEIR: Working Group on Compilers & Accelerators (heir.dev/community/)

- Open design meeting every two weeks
- Participants from across industry and academia
  - Companies: Google, Intel
  - Startups: Zama, Cryptolab
  - University: ETH Zurich, KU Leuven
  - Hardware developers: Optalysys, Niobium (Galois)

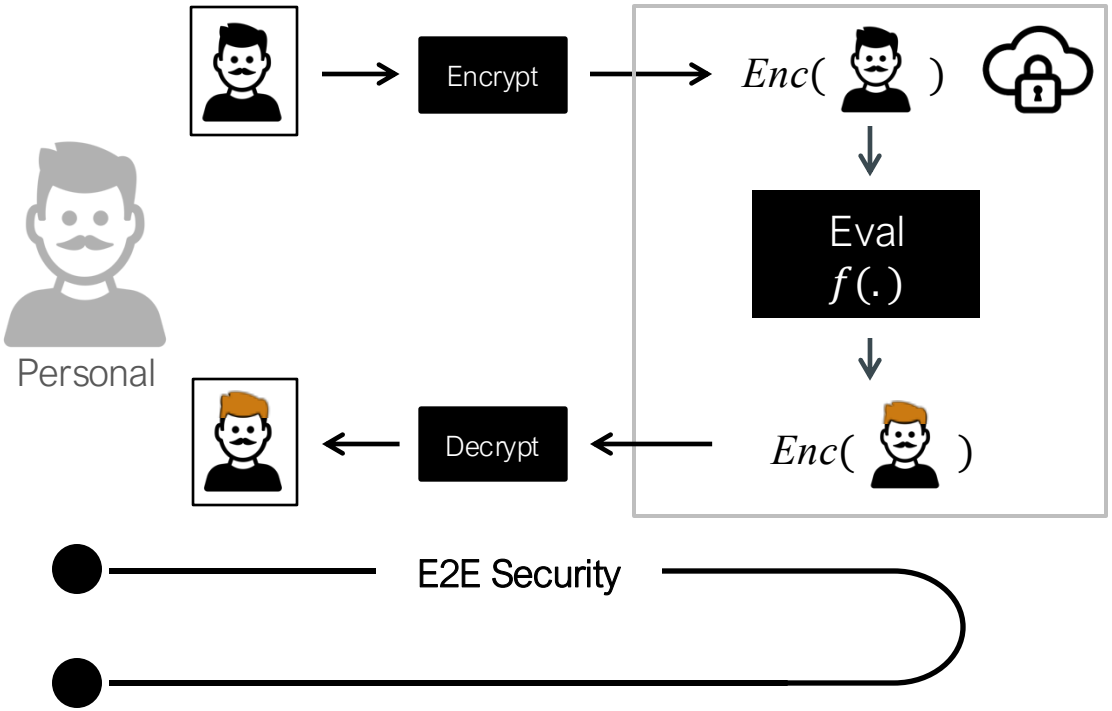
Meeting calendar

Future Directions in the Evolution of Secure Computation Tools


Homomorphic Encryption
Secure Multi-party Computation
Zero Knowledge Proofs
Differential Privacy
HElib
TFHE
OpenFHE
EVA
HECO
Concrete
HEIR
EMP-toolkit
MP-SPDZ
Sharemind
Obliv-C
ABY
CrypTFlow
TinyGarble
circom
libsnark
zkEVM
Zokrates
Bellman
Snarkjs
Arkworks
google-dp
Diffprivlib
Opacus
tf-privacy
OpenDP
Tumult
PipelineDP

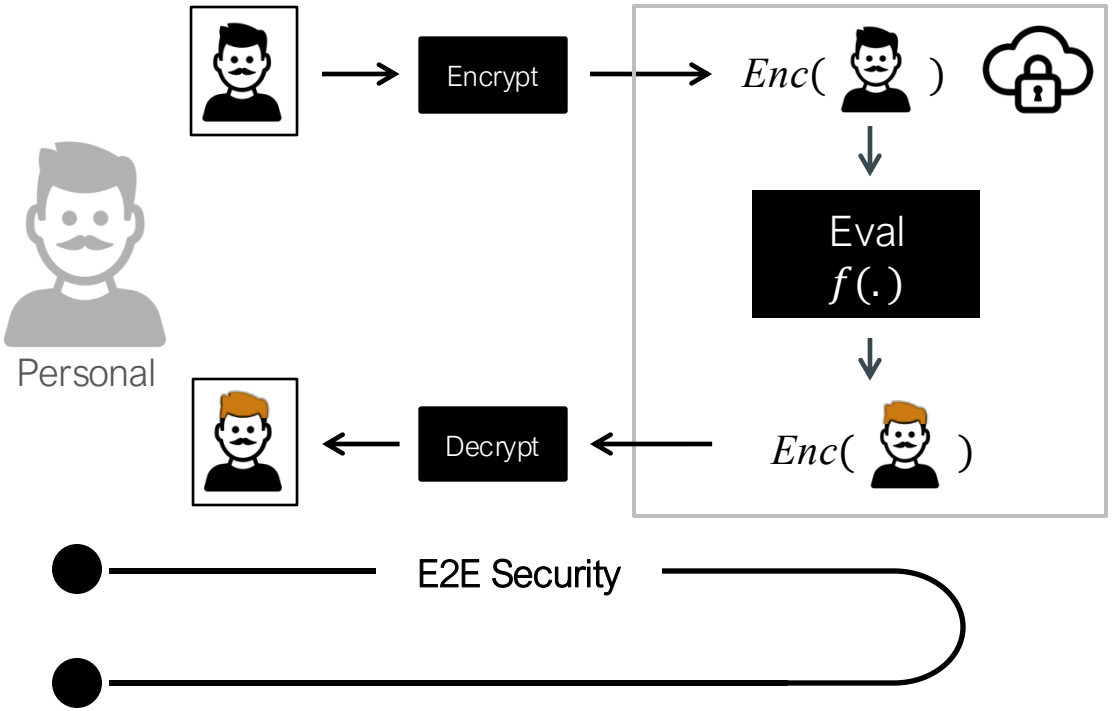
Secure Computation

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

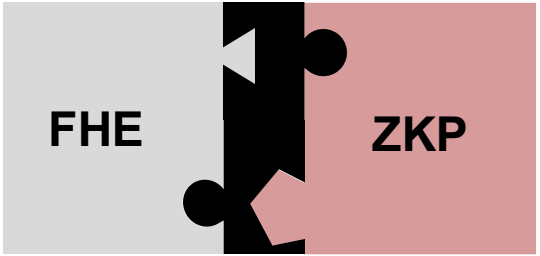


Secure Computation

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

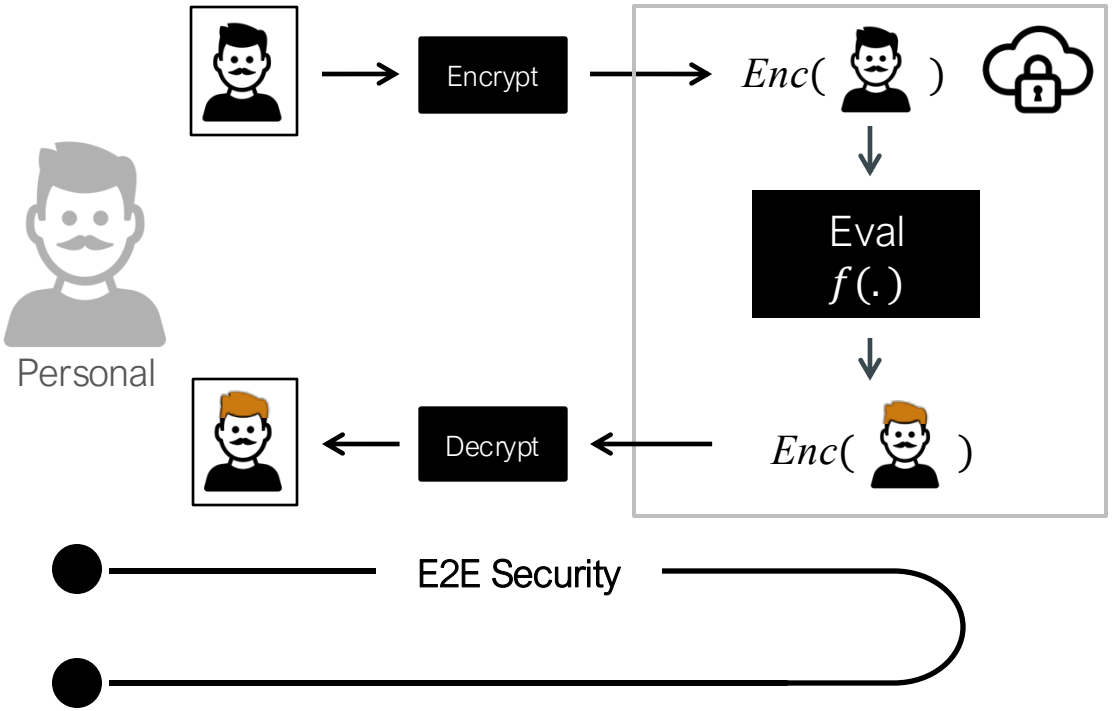


vFHE: Verifiable Fully Homomorphic Encryption. WAHC'24

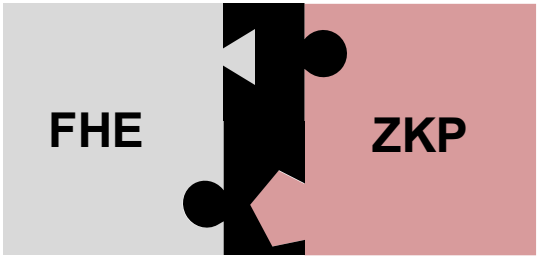


Secure Computation

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



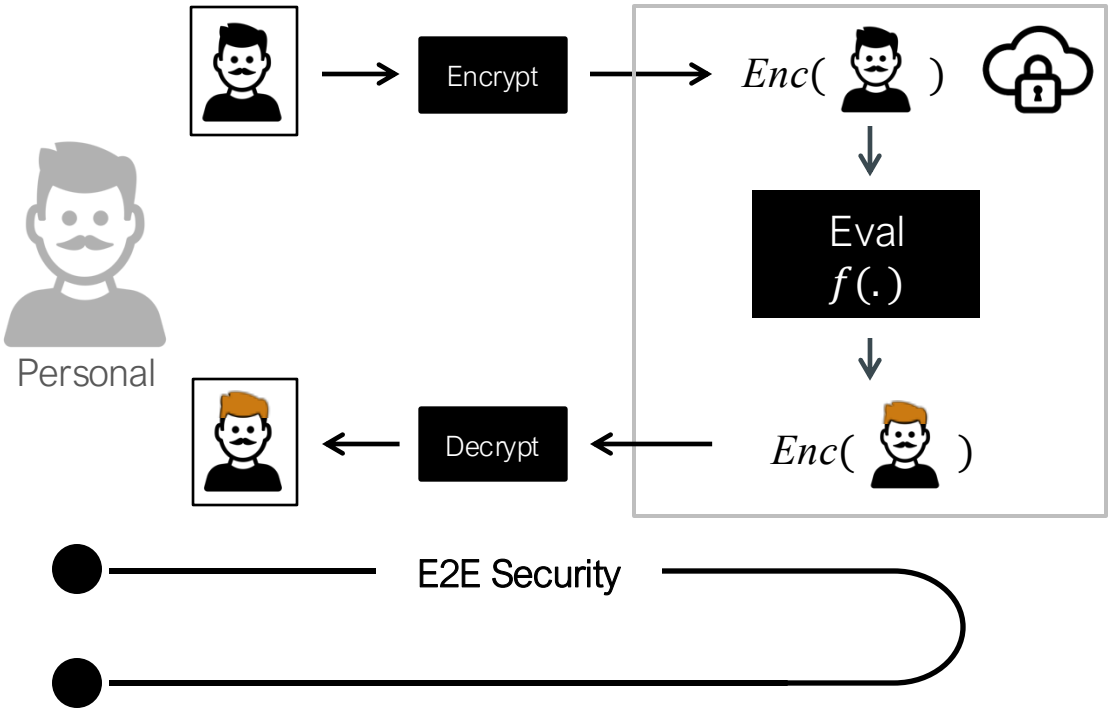
vFHE: Verifiable Fully Homomorphic Encryption. WAHC'24



Hybrid Compilation

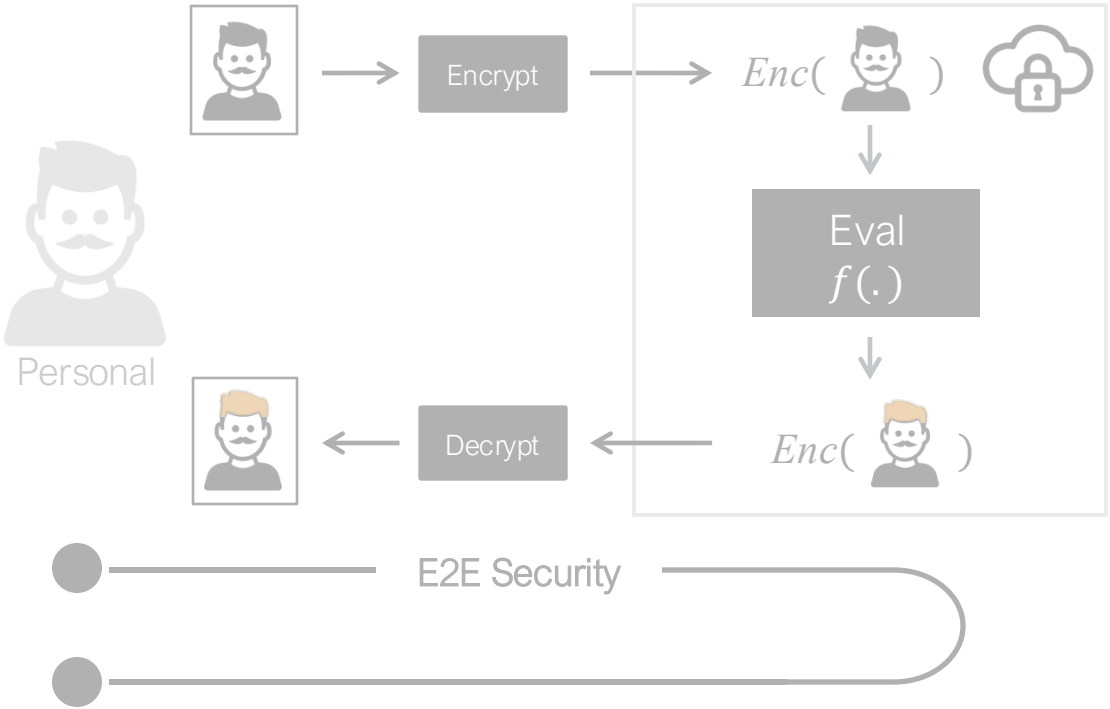
Secure Computation

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



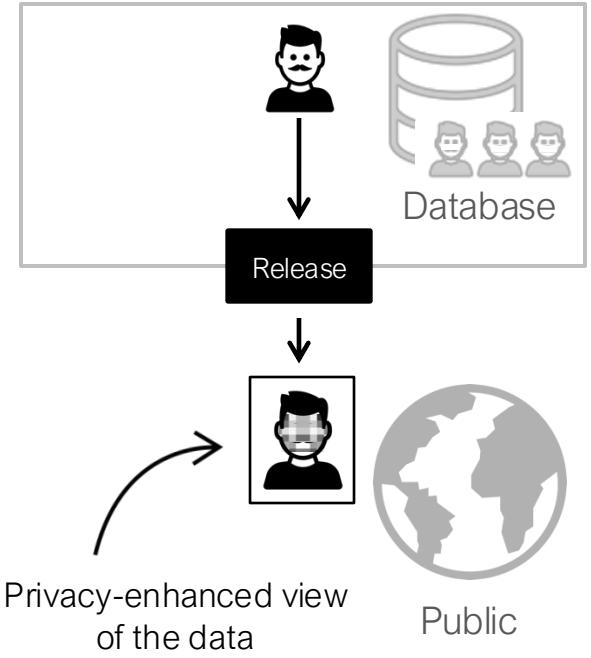
Secure Computation

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



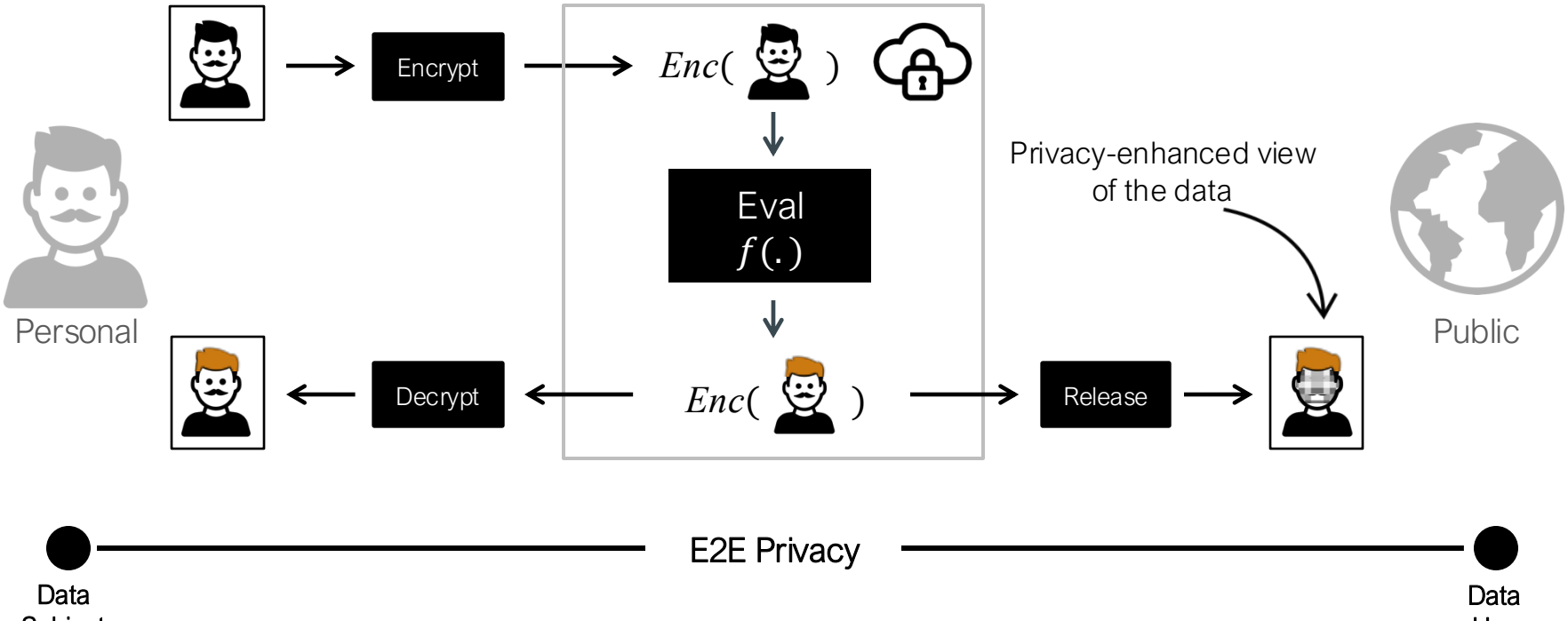
Releasing Data

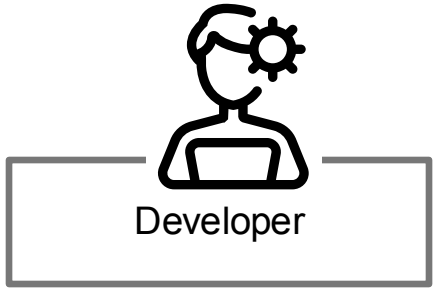
Differential Privacy



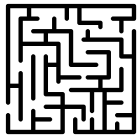
End-to-End Privacy

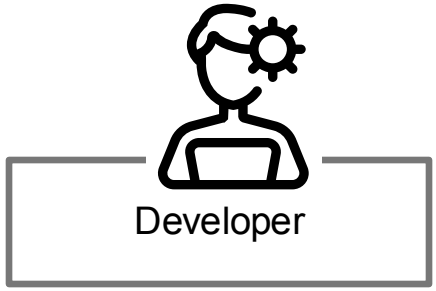
Homomorphic Encryption | Secure Multi-party Computation | Zero Knowledge Proofs | Differential Privacy



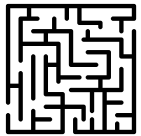


Accessibility





Accessibility

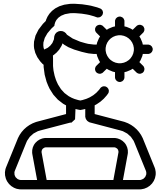


Hybrid Compilation

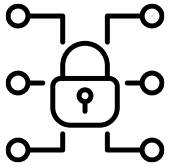
FHE ZKP

MPC

User

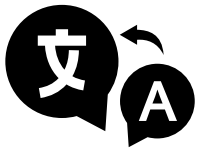


Developer



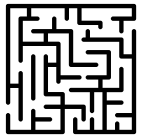
System

Mapping Guarantees to Secure Primitives



Privacy-Transparency Dichotomy

Accessibility

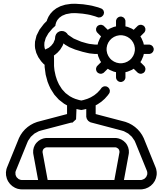


Hybrid Compilation

FHE ZKP

MPC

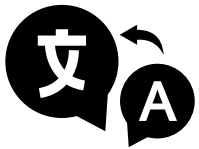
User



Developer

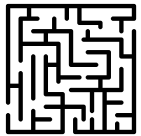
System

Mapping Guarantees to Secure Primitives



Privacy-Transparency Dichotomy

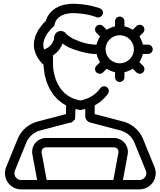
Accessibility



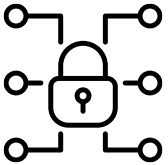
Hybrid Compilation

FHE ZKP MPC

User

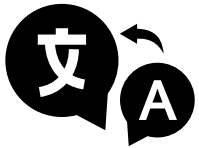


Developer



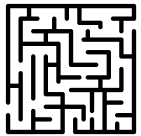
System

Mapping Guarantees to Secure Primitives



Privacy-Transparency Dichotomy

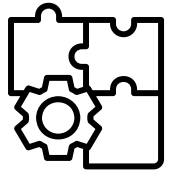
Accessibility



Hybrid Compilation

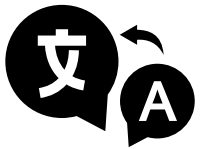
FHE ZKP MPC

Composability

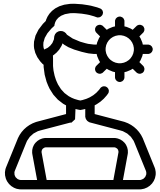


User

Mapping Guarantees to Secure Primitives

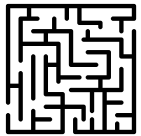


Privacy-Transparency Dichotomy



Developer

Accessibility



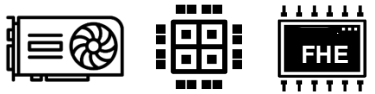
Hybrid Compilation

FHE ZKP MPC

System

Composability

Secure Computation on Heterogeneous Hardware



Work aims to **democratize access to privacy-preserving computation** with new tools, systems, and abstractions.

Acknowledgments

Students


Nicolas Küchler


Hidde Lycklama


Alexander Viand


Miro Haller


Patrick Jattke


Christian Knabenhans

Sponsors