RSAConference™2024

San Francisco | May 6 – 9 | Moscone Center

THE ART OF POSSIBLE

SESSION ID: CRYP-T09A

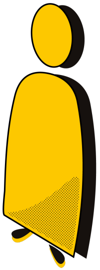
TFHE Public-Key Encryption Revisited

#RSAC

Marc Joye

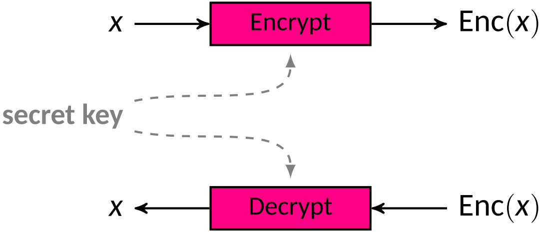
Chief Scientist
Zama, Paris, France

People shouldn’t care about privacy

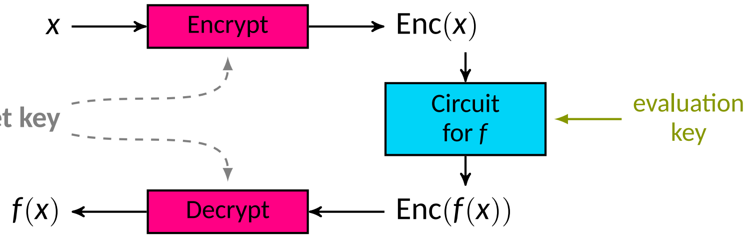


Not because it doesn’t matter, but because it shouldn’t be an issue

Fully Homomorphic Encryption



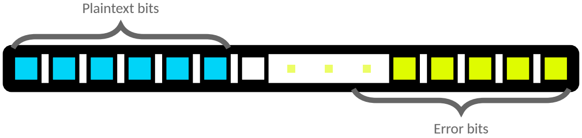
Fully Homomorphic Encryption



Remark: Any private-key FHE scheme can easily be turned into a public-key FHE scheme

Torus-FHE a.k.a. TFHE

secret key: $\mathbf{s} \in \{0,1\}^n$



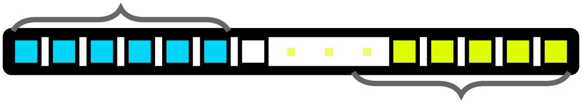
Encryption

1. $\mathbf{a} \xleftarrow{\$} \mathbb{Z}_q^n$ (mask)
2. $\mu := \Delta m + e$ with $e \leftarrow \chi$
3. $b \leftarrow \mu + \langle \mathbf{a}, \mathbf{s} \rangle \pmod q$ (body)

ciphertext: $(\mathbf{a}, b) \in \mathbb{Z}_q^{n+1}$

Torus-FHE a.k.a. TFHE

secret key: $\mathbf{s} \in \{0,1\}^n$



Encryption

1. $\mathbf{a} \stackrel{\$}{\leftarrow} \mathbb{Z}_q^n$ (mask)
2. $\mu := \Delta m + e$ with $e \leftarrow \chi$
3. $b \leftarrow \mu + \langle \mathbf{a}, \mathbf{s} \rangle \pmod{q}$ (body)

Decryption

1. $\mu \leftarrow b - \langle \mathbf{a}, \mathbf{s} \rangle \pmod{q}$
2. round μ and get $m = \lceil \mu / \Delta \rfloor$

(correctness requires $|e| < \Delta/2$)

ciphertext: $(\mathbf{a}, b) \in \mathbb{Z}_q^{n+1}$

From Private-Key to Public-Key Encryption

$$\mathsf{pk} = \left(u_1 \leftarrow [\![0]\!]_{\mathsf{sk}}, \ldots, u_z \leftarrow [\![0]\!]_{\mathsf{sk}}\right)$$

public-key encryption

- $(r_1, \ldots, r_z) \stackrel{\$}{\leftarrow} \{0,1\}^z$
- $S \leftarrow \boxplus_{i=1}^{z} r_i\, u_i$
- $M \leftarrow [\![m]\!]_{\mathsf{sk}}$ ("trivial" encryption)
- return $C \leftarrow S \boxplus M$

Note: $(\mathbf{0}, \Delta m)$ is a trivial TFHE encryption of m

From Private-Key to Public-Key Encryption

$\mathsf{pk} = \left(u_1 \leftarrow [\![0]\!]_{\mathsf{sk}}, \ldots, u_z \leftarrow [\![0]\!]_{\mathsf{sk}}\right)$

LHL teaches that $z = (n+1)\,|q|_2 + \kappa$

public-key encryption

- $(r_1, \ldots, r_z) \xleftarrow{\$} \{0,1\}^z$
- $S \leftarrow \boxplus_{i=1}^{z} r_i\, u_i$
- $M \leftarrow [\![m]\!]_{\mathsf{sk}}$ (“trivial” encryption)
- return $C \leftarrow S \boxplus M$

Note: $(\mathbf{0}, \Delta m)$ is a trivial TFHE encryption of m

For typical parameters, the resulting public key pk for TFHE takes 526 kB

This Talk: Public-key variant of TFHE

- Output ciphertexts are of LWE type
  $\rightsquigarrow$ TFHE compatible
- Two useful properties:
  1. public key is much shorter
  2. resulting ciphertexts are less noisy
- Security based on RLWE

Main Tool: 'Special' Vector Convolution

Definition For $\boldsymbol{u} = (u_1, \ldots, u_n), \boldsymbol{v} = (v_1, \ldots, v_n) \in \mathbb{Z}^n$,

$$\boldsymbol{w} = \boldsymbol{u} \circledast \boldsymbol{v} = (\underbrace{\boldsymbol{u} \circledast_1 \boldsymbol{v}}_{=w_1}, \ldots, \underbrace{\boldsymbol{u} \circledast_n \boldsymbol{v}}_{=w_n}) \in \mathbb{Z}^n$$

where

$$w_i = \boldsymbol{u} \circledast_i \boldsymbol{v} = \sum_{j=1}^{i} u_j \, v_{n+j-i} - \sum_{j=i+1}^{n} u_j \, v_{j-i}$$

Main Tool: 'Special' Vector Convolution

Definition For $\boldsymbol{u} = (u_1, \ldots, u_n), \boldsymbol{v} = (v_1, \ldots, v_n) \in \mathbb{Z}^n$,

$$\boldsymbol{w} = \boldsymbol{u} \circledast \boldsymbol{v} = (\underbrace{\boldsymbol{u} \circledast_1 \boldsymbol{v}}_{=w_1}, \ldots, \underbrace{\boldsymbol{u} \circledast_n \boldsymbol{v}}_{=w_n}) \in \mathbb{Z}^n$$

where

$$w_i = \boldsymbol{u} \circledast_i \boldsymbol{v} = \sum_{j=1}^{i} u_j\, v_{n+j-i} - \sum_{j=i+1}^{n} u_j\, v_{j-i}$$

Properties

1. $\boldsymbol{u} \circledast \boldsymbol{v} = \overleftarrow{\boldsymbol{v}} \circledast \overleftarrow{\boldsymbol{u}}$
2. $\boldsymbol{u} \circledast_n \boldsymbol{v} = \langle \boldsymbol{u}, \boldsymbol{v} \rangle$
3. $\langle \boldsymbol{t} \circledast \boldsymbol{u}, \boldsymbol{v} \rangle = \langle \boldsymbol{t} \circledast \boldsymbol{v}, \boldsymbol{u} \rangle$

New TFHE Public-Key Variant

Key generation

1. $\mathbf{s} \xleftarrow{\$} \{0,1\}^n$; $\mathbf{e} \leftarrow \chi$
2. $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^n$
3. $\mathbf{B} \leftarrow \mathbf{A} \circledast \mathbf{s} + \mathbf{e} \pmod{q}$

$\mathrm{pk} = (\mathbf{A}, \mathbf{B})$ and $\mathrm{sk} = \mathbf{s}$

New TFHE Public-Key Variant

Key generation

1. $\boldsymbol{s} \xleftarrow{\$} \{0,1\}^n$; $\boldsymbol{e} \leftarrow \chi$
2. $\boldsymbol{A} \xleftarrow{\$} \mathbb{Z}_q^n$
3. $\boldsymbol{B} \leftarrow \boldsymbol{A} \circledast \boldsymbol{s} + \boldsymbol{e} \pmod{q}$

pk $= (\boldsymbol{A}, \boldsymbol{B})$ and sk $= \boldsymbol{s}$

Encryption of m

1. $\boldsymbol{r} \xleftarrow{\$} \{0,1\}^n$; $\boldsymbol{e_1} \leftarrow \chi^n$; $e_2 \leftarrow \chi$
2. $\boldsymbol{a} \leftarrow \boldsymbol{A} \circledast \boldsymbol{r} + \boldsymbol{e_1}$ (mask)
3. $b \leftarrow \mu + \langle \boldsymbol{B}, \boldsymbol{r} \rangle \pmod{q}$ (body)
   with $\mu := \Delta m + e_2$

New TFHE Public-Key Variant

Key generation

1. $\boldsymbol{s} \xleftarrow{\$} \{0,1\}^n$; $\boldsymbol{e} \leftarrow \chi$
2. $\boldsymbol{A} \xleftarrow{\$} \mathbb{Z}_q^n$
3. $\boldsymbol{B} \leftarrow \boldsymbol{A} \circledast \boldsymbol{s} + \boldsymbol{e} \pmod q$

pk $= (\boldsymbol{A}, \boldsymbol{B})$ and sk $= \boldsymbol{s}$

Encryption of m

1. $\boldsymbol{r} \xleftarrow{\$} \{0,1\}^n$; $\boldsymbol{e_1} \leftarrow \chi^n$; $e_2 \leftarrow \chi$
2. $\boldsymbol{a} \leftarrow \boldsymbol{A} \circledast \boldsymbol{r} + \boldsymbol{e_1}$ (mask)
3. $b \leftarrow \mu + \langle \boldsymbol{B}, \boldsymbol{r} \rangle \pmod q$ (body)
   with $\mu := \Delta m + e_2$

Decryption of $(\boldsymbol{a}, b)$

1. $\mu \leftarrow b - \langle \boldsymbol{s}, \boldsymbol{a} \rangle \pmod q$
2. round μ and get $m = \lceil \mu / \Delta \rfloor$

(correctness requires $|e| < \Delta/2$)

Security & Performance

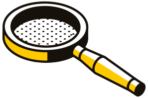
Scheme is semantically secure under the RLWE assumption in $\mathbb{Z}_q[X]/(X^n+1)$ (with n a power of 2)

Note: If $\boldsymbol{u} = (u_1, u_2, \ldots, u_n) \in \mathbb{Z}_q^n \overset{\sim}{\longleftrightarrow} \mathcal{u} = u_1 + u_2 X + \cdots + u_n X^{n-1} \in \mathbb{Z}_q[X]/(X^n+1)$
then $\boldsymbol{u} \circledast \overleftarrow{\boldsymbol{v}} = \boldsymbol{v} \circledast \overleftarrow{\boldsymbol{u}} \cong \mathcal{u} \cdot \mathcal{v}$

Security & Performance

Scheme is semantically secure under the RLWE assumption in $\mathbb{Z}_q[X]/(X^n+1)$ (with n a power of 2)

Note: If $\mathbf{u} = (u_1, u_2, \ldots, u_n) \in \mathbb{Z}_q^n \overset{\sim}{\longleftrightarrow} \mathcal{u} = u_1 + u_2 X + \cdots + u_n X^{n-1} \in \mathbb{Z}_q[X]/(X^n+1)$
then $\mathbf{u} \circledast \overleftarrow{\mathbf{v}} = \mathbf{v} \circledast \overleftarrow{\mathbf{u}} \cong \mathcal{u} \cdot \mathcal{v}$



- For typical parameters, the public key pk only takes 8.2 kB (instead of 526 kB)
- Resulting ciphertexts are also less noisy — typically σ of 2^{44} (instead of $2^{46.5}$)

Generalizations

More general polynomial rings Multiplication in polynomial rings induces a convolution between vectors

- basic scheme relies on $\mathbb{Z}_q[X]/(X^n+1)$ with n a power of 2
- similar schemes with $R_q := \mathbb{Z}_q[X]/(\mathfrak{p})$ for some monic irreducible polynomial $\mathfrak{p}$
  - e.g., cyclotomic polynomials $\mathfrak{p}(X) = \Phi_M(X)$
  - e.g., $\mathfrak{p}(X) = X^{2n} + X^n + 1$ with n a power of 3

More general distributions Private key ***s*** and/or randomizer ***r*** can be drawn in $\{-1, 0, 1\}$, or in small subsets of $\mathbb{Z}_q$

Encrypting Multiple Plaintexts

- Encryption of Z plaintexts
  - Naïve approach $\rightsquigarrow Z(n+1)|q|_2$ bits
  - Packing technique $\rightsquigarrow \left(\lceil Z/n \rceil\, n + Z\right)|q|_2$ bits

e.g., for $Z = n \implies 2n|q|_2$ bits vs. $n(n+1)|q|_2 \approx n^2|q|_2$ bits

Encrypting Multiple Plaintexts

- Encryption of Z plaintexts
  - Naïve approach $\rightsquigarrow Z(n+1)|q|_2$ bits
  - Packing technique $\rightsquigarrow \left(\lceil Z/n \rceil\, n + Z\right)|q|_2$ bits

e.g., for $Z = n \implies 2n|q|_2$ bits vs. $n(n+1)|q|_2 \approx n^2|q|_2$ bits

- "Mask can be shared"

Public-key encryption of $m_1, \ldots, m_Z$

1. $\boldsymbol{r} \xleftarrow{\$} \{0,1\}^n$; $\boldsymbol{e_1} \leftarrow \chi^n$; $\boldsymbol{e_2} \leftarrow \chi^Z$
2. $\boldsymbol{a} \leftarrow \boldsymbol{A} \circledast \boldsymbol{r} + \boldsymbol{e_1}$ (mask)
3. $\begin{cases} b_1 \leftarrow \Delta m_1 + e_{2,1} + \langle \boldsymbol{B}, \boldsymbol{r} \rangle \pmod{q} \\ b_j \leftarrow \Delta m_j + e_{2,j} + (\boldsymbol{B} \circledast \boldsymbol{r})_{i_j} \pmod{q} \end{cases}$ (for $2 \leq j \leq Z$)

- $(\boldsymbol{a}, b_1)$ is an LWE ciphertext
- $(\Psi_{i_j}(\boldsymbol{a}), b_j)$ are LWE ciphertexts for some public maps Ψ_{i_j}

ZAMA

Conclusion

NEW SCHEME

- Public-key variant of TFHE with ciphertexts as LWE samples
  - ✓ significantly smaller public-key size
  - ✓ lower noise in resulting ciphertexts
  - ✓ provably secure under the RLWE assumption
- Generalizations and extensions
- Packing technique and companion conversion

Conclusion

NEW SCHEME

- Public-key variant of TFHE with ciphertexts as LWE samples
  - ✓ significantly smaller public-key size
  - ✓ lower noise in resulting ciphertexts
  - ✓ provably secure under the RLWE assumption
- Generalizations and extensions
- Packing technique and companion conversion

APPLICATION

- Integrated in fhEVM (private smart-contract protocol)

Conclusion

NEW SCHEME

- Public-key variant of TFHE with ciphertexts as LWE samples
  - ✓ significantly smaller public-key size
  - ✓ lower noise in resulting ciphertexts
  - ✓ provably secure under the RLWE assumption
- Generalizations and extensions
- Packing technique and companion conversion

APPLICATION

- Integrated in fhEVM (private smart-contract protocol)

RSAConference™2024

San Francisco | May 6 – 9 | Moscone Center

THE ART OF POSSIBLE

SESSION ID: CRYP-T09B

Differential Privacy for Free? Harnessing the Noise in Approximate Homomorphic Encryption

#RSAC

Tabitha Ogilvie

Intel Labs; Royal Holloway, University of London

Disclaimer

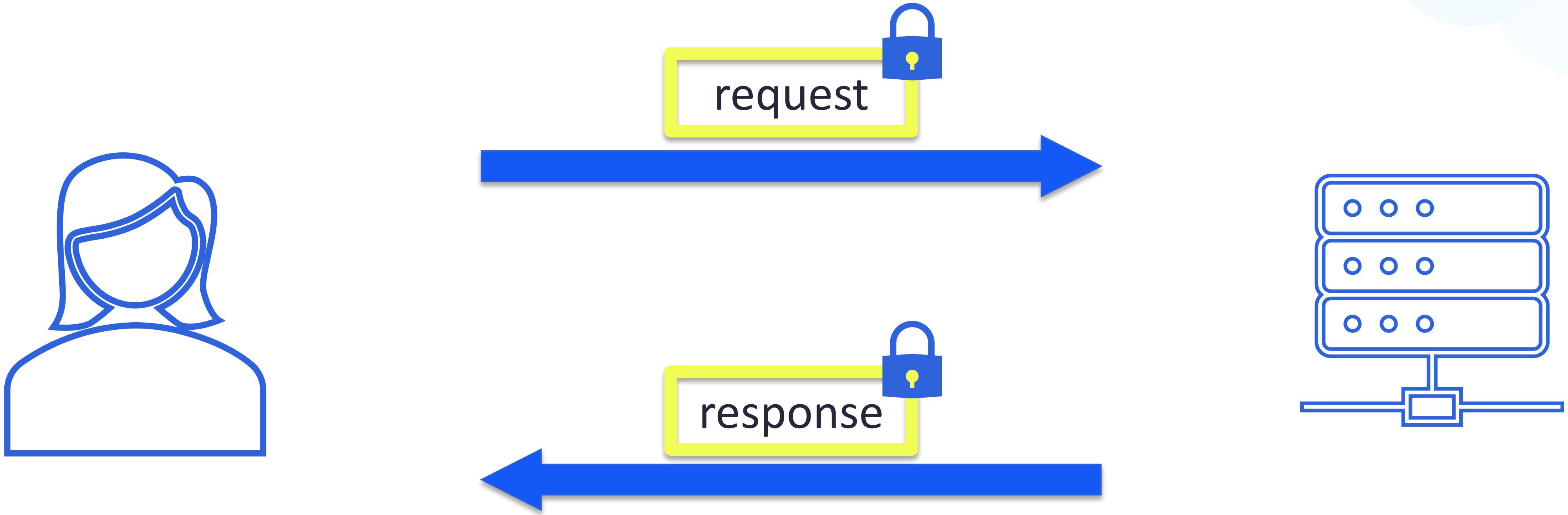
Presentations are intended for educational purposes only and do not replace independent professional judgment. Statements of fact and opinions expressed are those of the presenters individually and, unless expressly stated to the contrary, are not the opinion or position of RSA Conference™ or any other co-sponsors. RSA Conference does not endorse or approve, and assumes no responsibility for, the content, accuracy or completeness of the information presented.

Attendees should note that sessions may be audio- or video-recorded and may be published in various media, including print, audio and video formats without further notice. The presentation template and any media capture are subject to copyright protection.

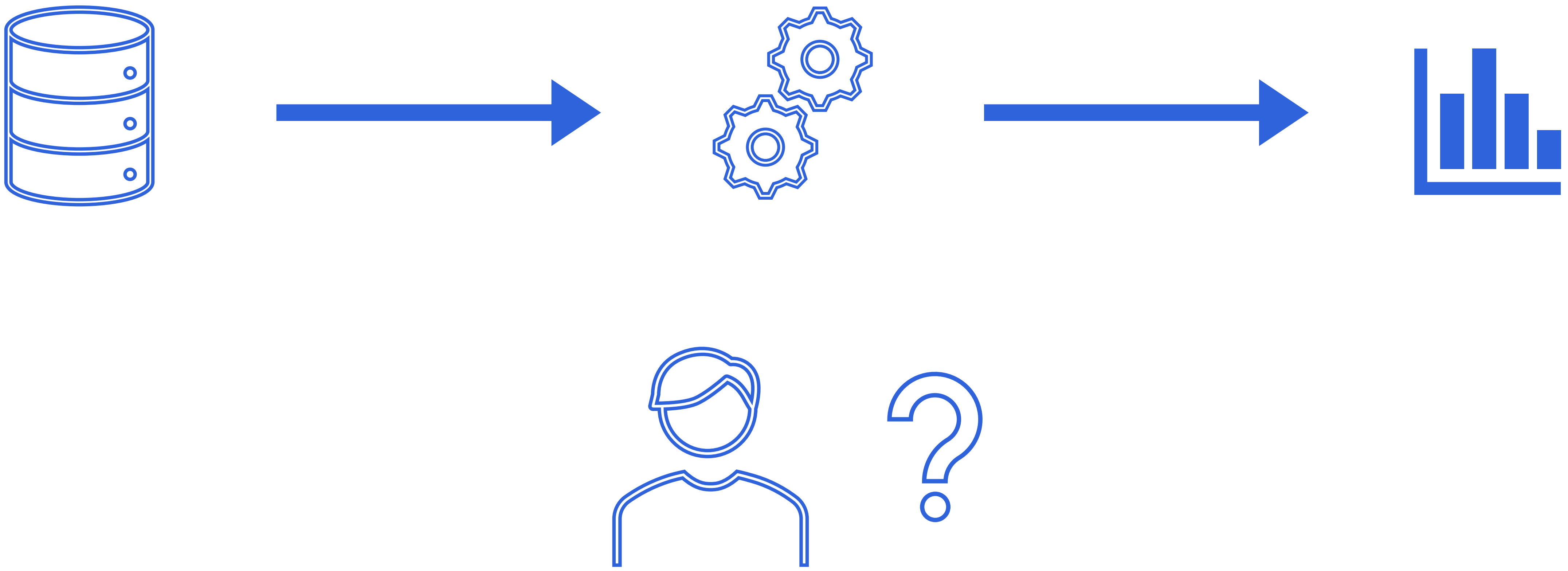
RSAConference™2024

Introduction

Homomorphic Encryption



Differential Privacy



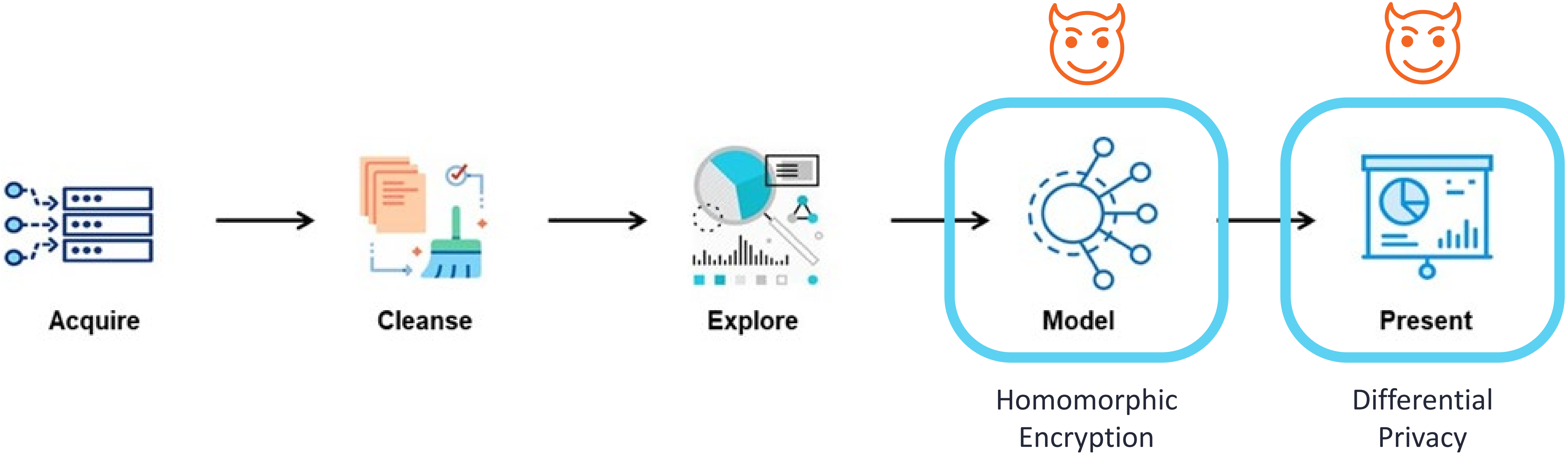


Image credit: https://www.xtivia.com/blog/data-science-pipeline-guidelines/

Our Contributions

Under the hood: noise!

- Popular Homomorphic Encryption schemes rely on the Learning with Errors problem
  - this means we add noise during encryption which grows during computation
- We achieve Differential Privacy by adding noise which obscures any single individual

Q: Can the noise in Homomorphic Encryption be used to give differential privacy "for free"?

Homomorphic Encryption

Differential Privacy

A: yes!

But it's very challenging

Complications

- Homomorphic Encryption noise is typically small
- Homomorphic Encryption noise is difficult to model
- Homomorphic Encryption noise is message dependent
- Homomorphic Encryption noise exposure can compromise security

Conclusion

Apply What We've Talked About Today

- What is the data pipeline for your organization?
  - Where is data being exposed along the way?
- Could you integrate Privacy Enhancing Technologies?
  - Homomorphic Encryption, Differential Privacy
- Does your use of these technologies make sense in the context of the entire pipeline?

RSAConference™2024

Thank you for your attention!

Tabitha Ogilvie