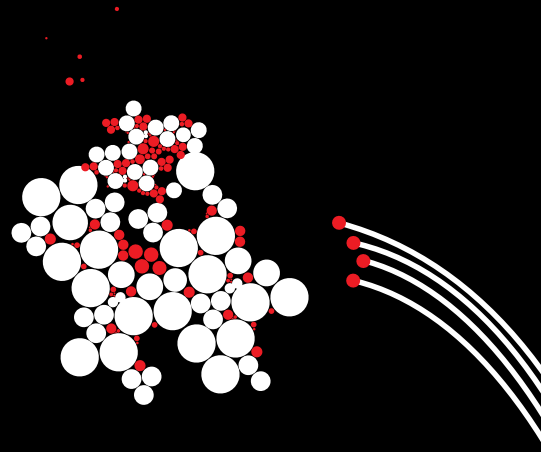
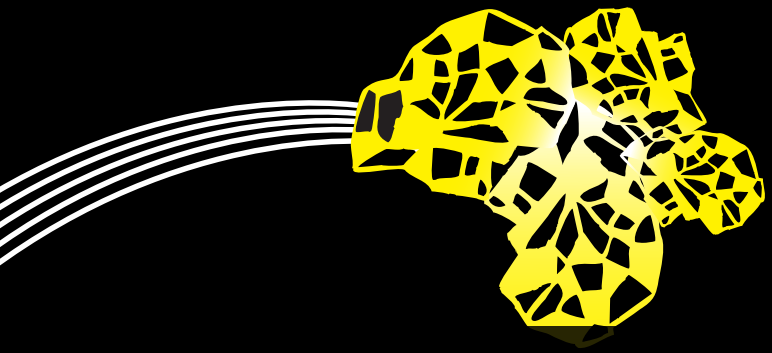
UNIVERSITY OF TWENTE.



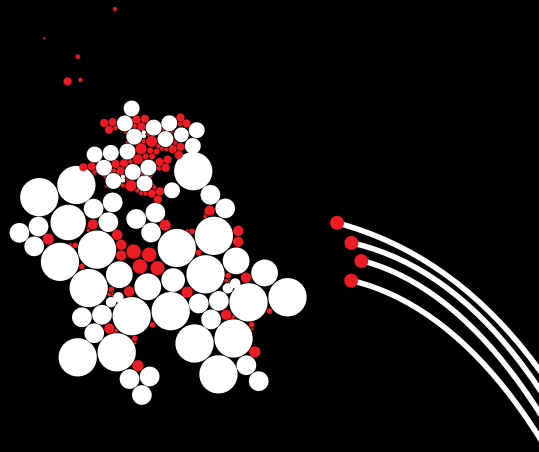
Multi-client Predicate-only Encryption for Conjunctive Equality Tests

Tim van de Kamp Andreas Peter Maarten Everts Willem Jonker



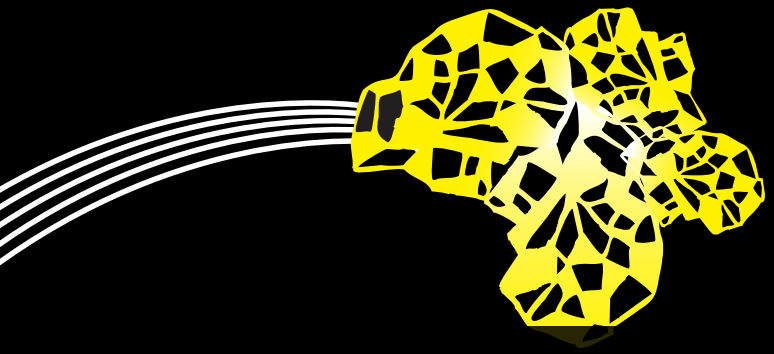
16th International Conference on Cryptology And Network Security, 2017

UNIVERSITY OF TWENTE.



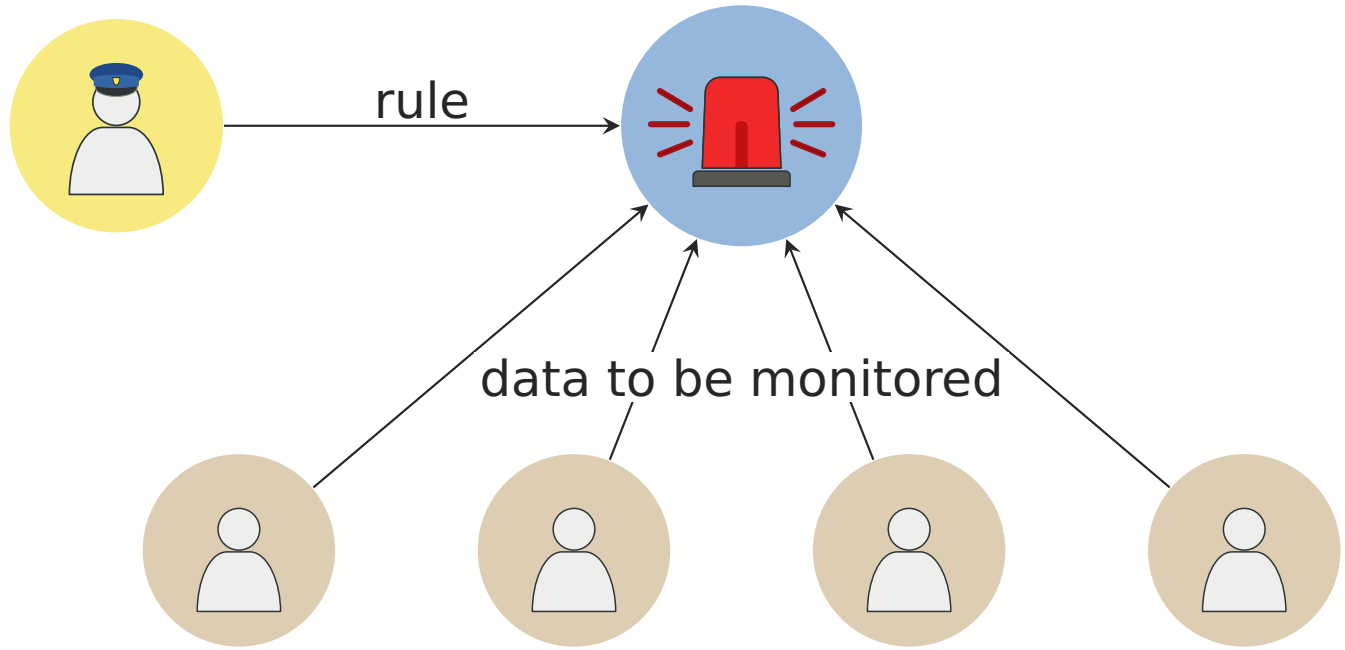
Monitoring over Encrypted Data

Tim van de Kamp **Andreas Peter** **Maarten Everts** **Willem Jonker**



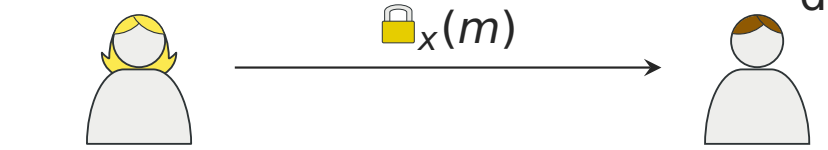
16th International Conference on Cryptology And Network Security, 2017

This Talk: Monitoring over Encrypted Data



Monitoring of sensitive data using sensitive monitoring rules.

Background: Predicate Encryption



decryption $\begin{cases} m & \text{if } f_y(x) = \text{TRUE} \\ \perp & \text{if } f_y(x) = \text{FALSE} \end{cases}$

message: m
label: x

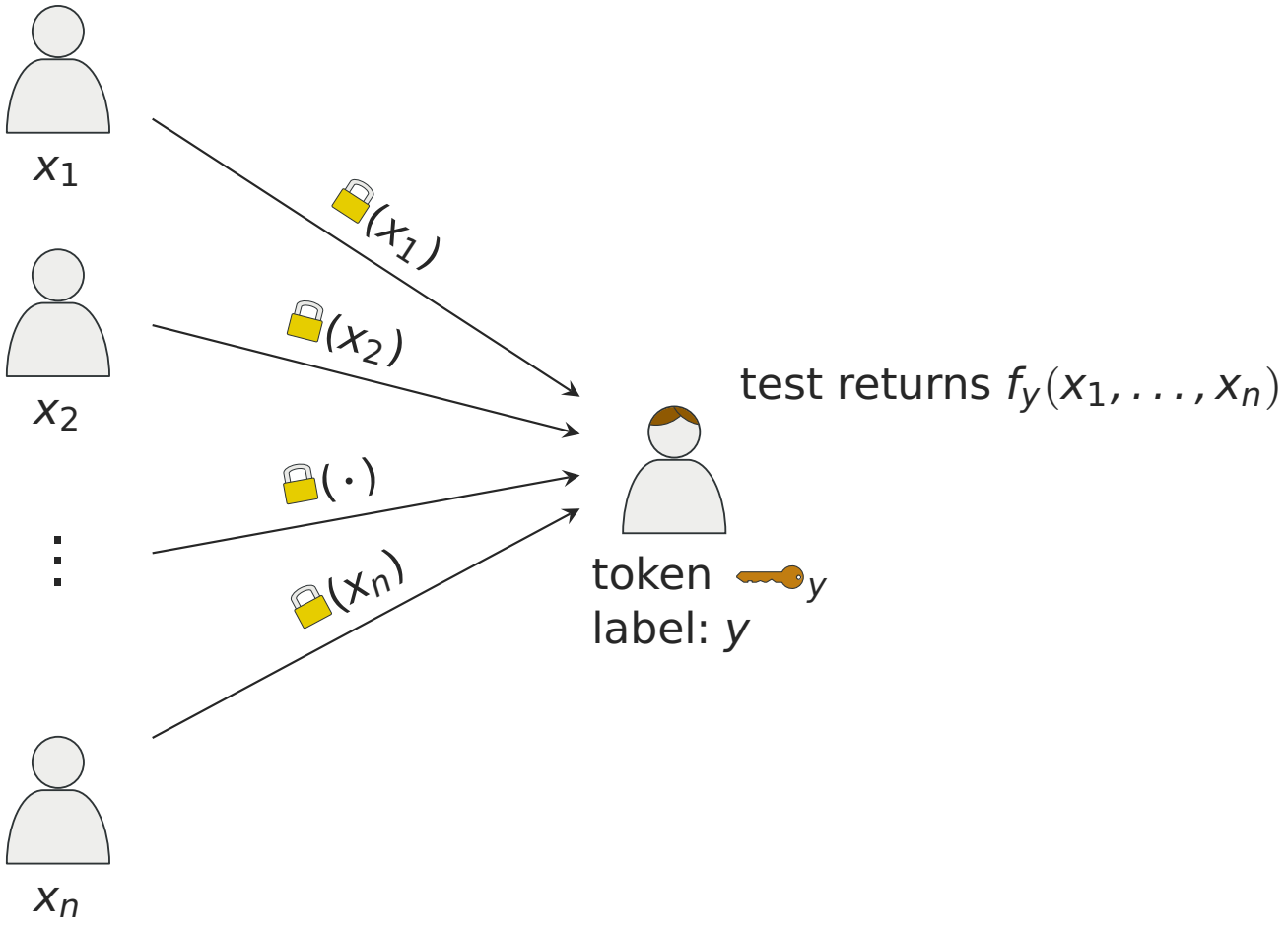
decryption key $_y$
label: y

Predicate encryption for relation $R(x, y)$.

Examples

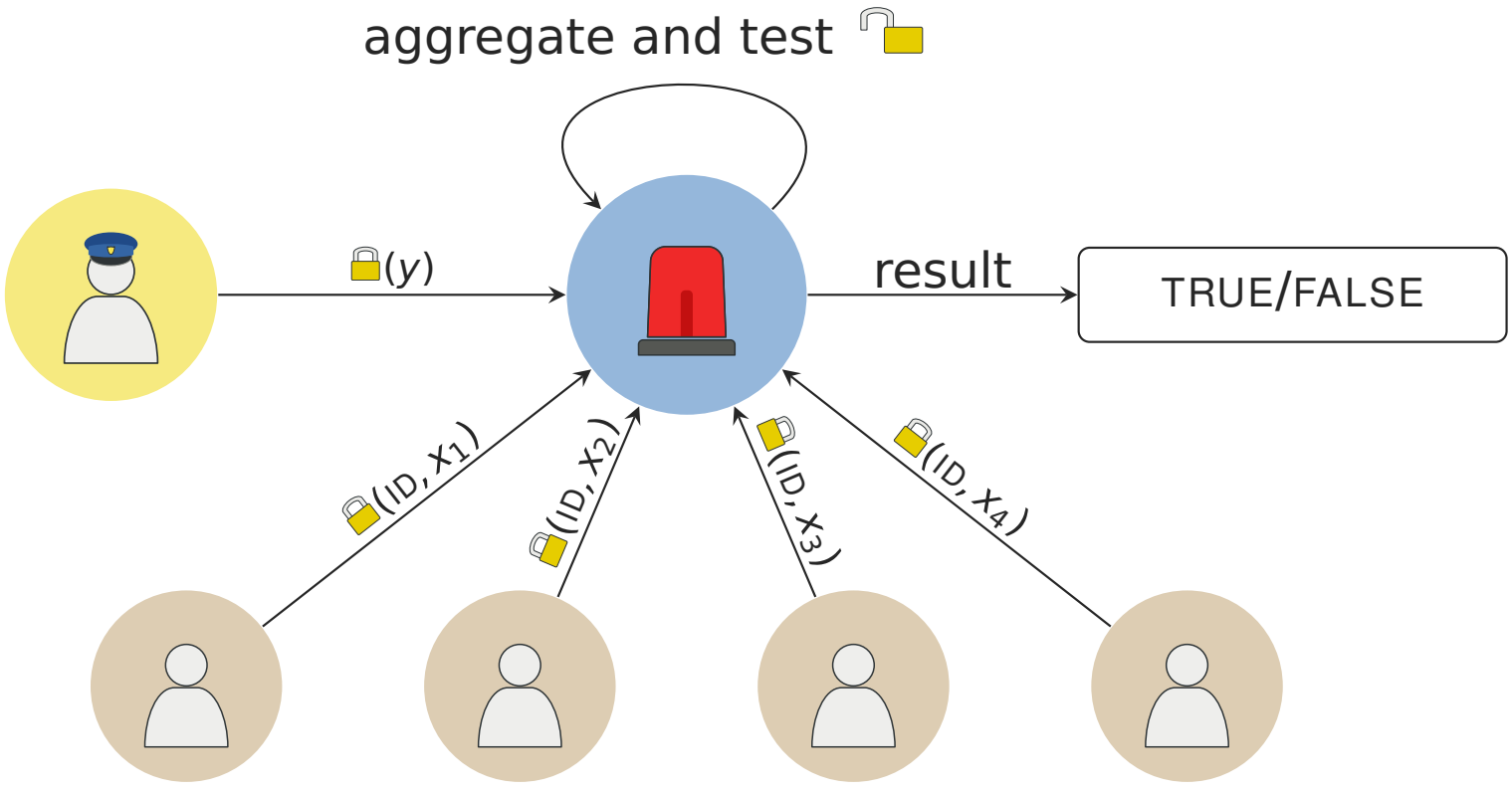
- Identity-based encryption
- Attribute-based encryption
- Hidden vector encryption
- Inner-product predicate encryption

Multi-client Predicate-only Encryption – Concept

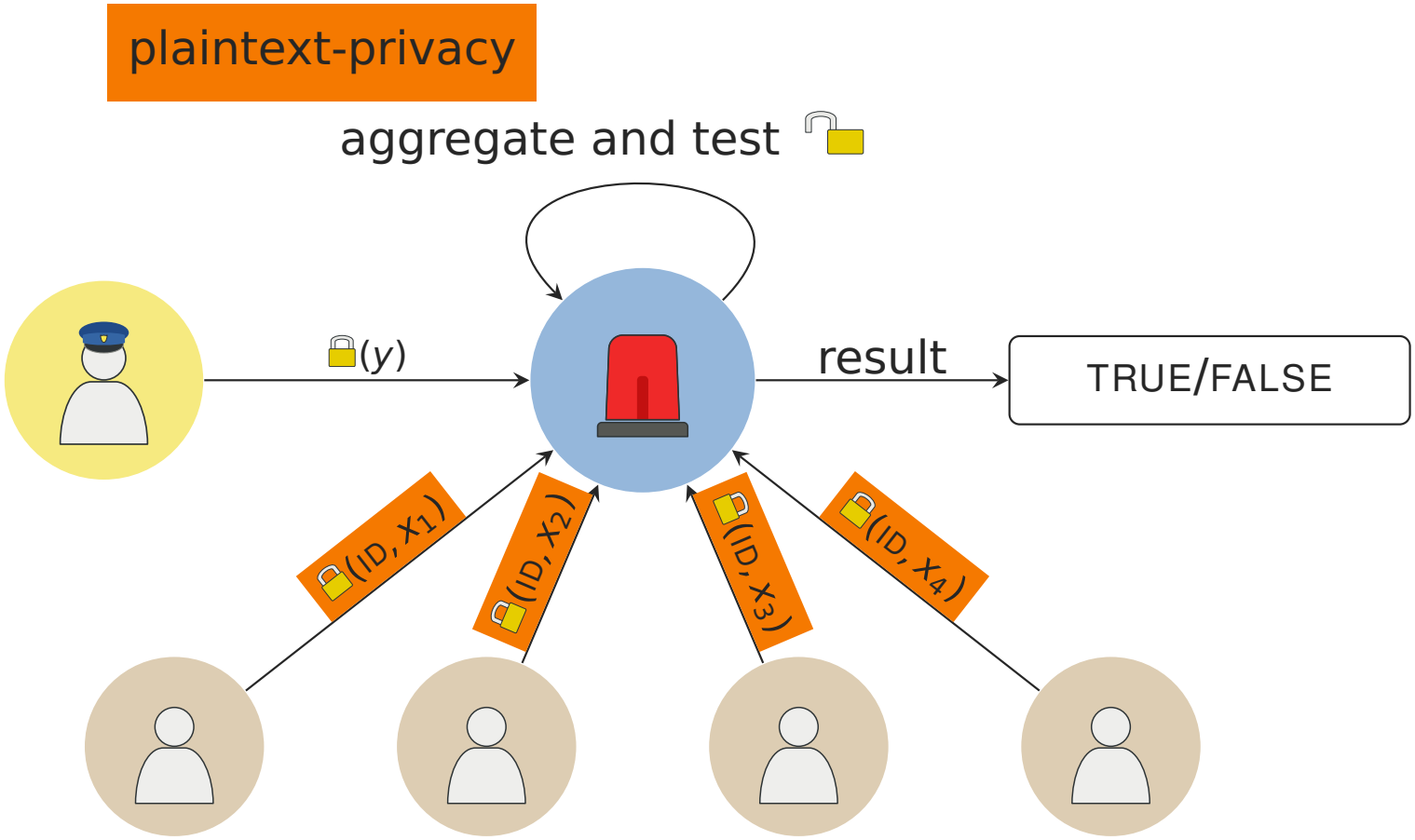


Multi-client predicate-only encryption for relation $R(x_1, \ldots, x_n, y)$.

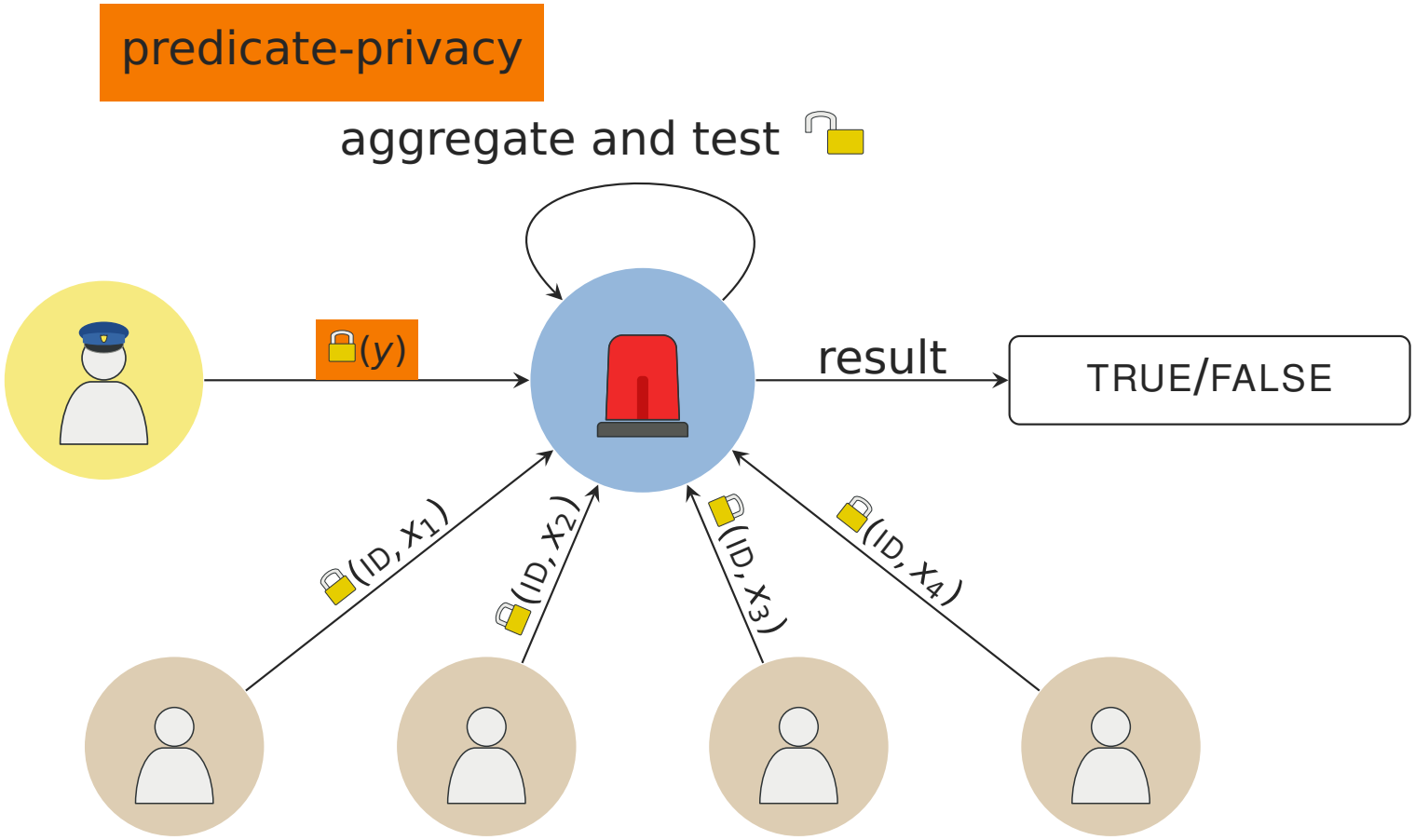
Multi-client Predicate-only Encryption – Security



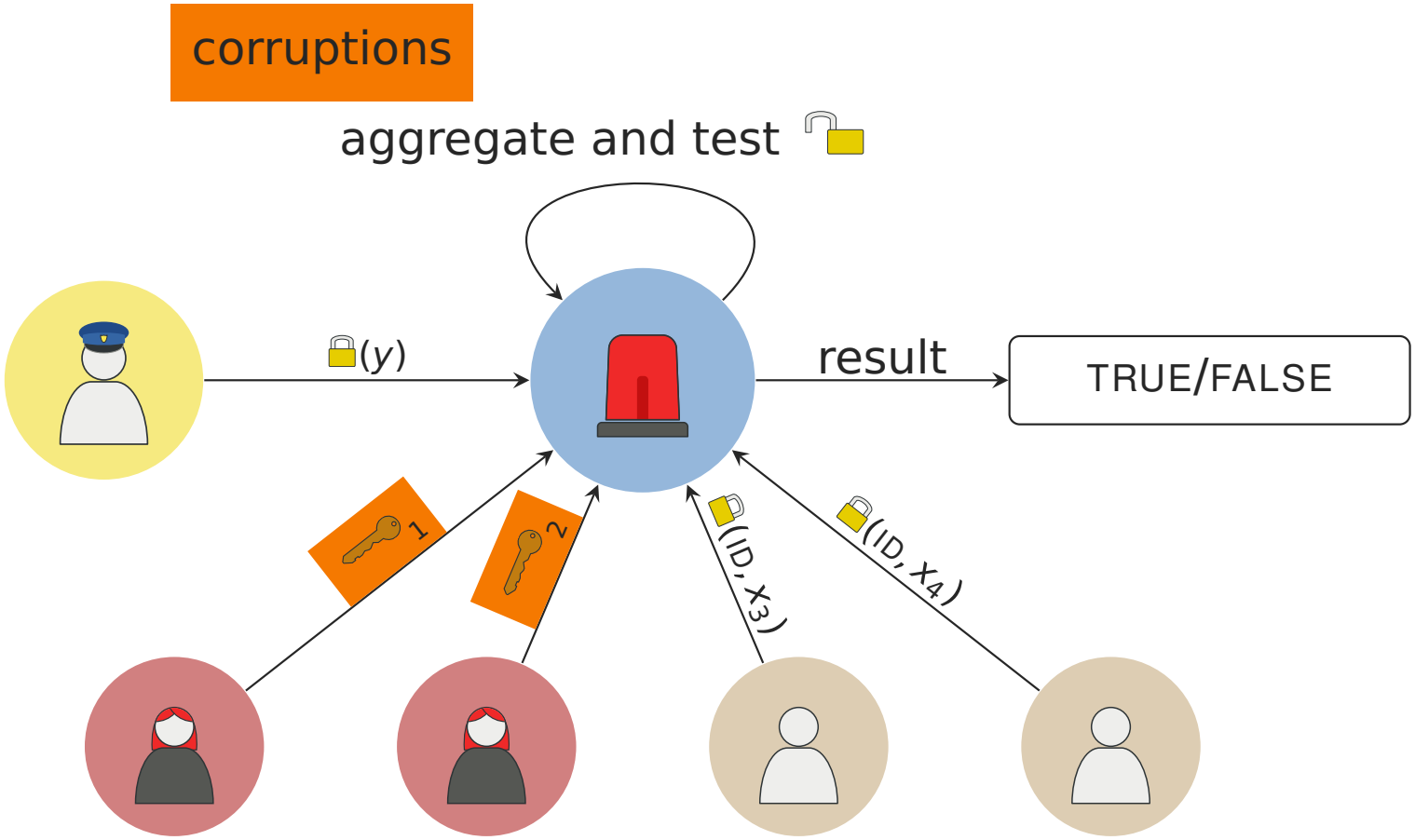
Multi-client Predicate-only Encryption – Security



Multi-client Predicate-only Encryption – Security

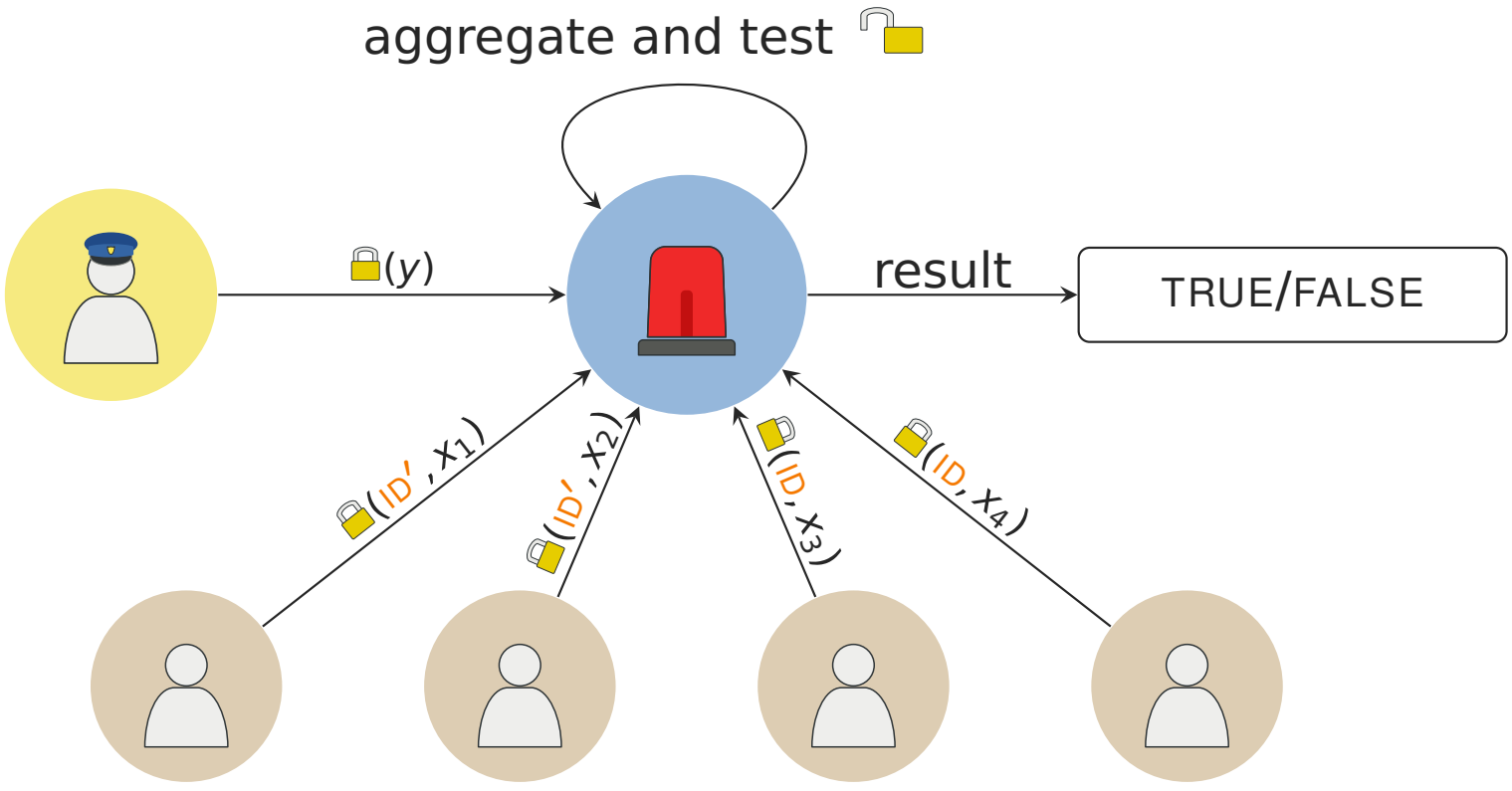


Multi-client Predicate-only Encryption – Security

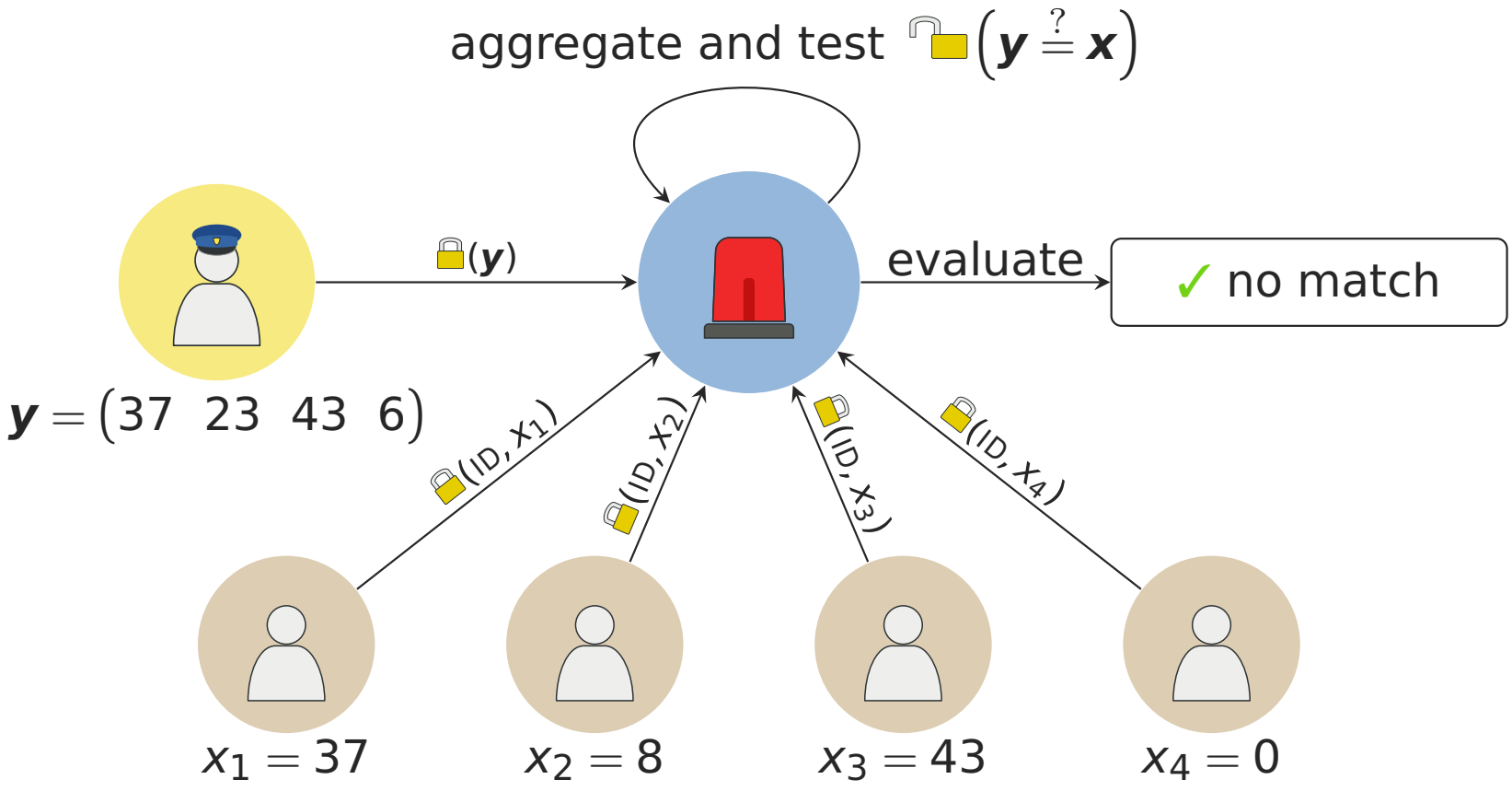


Multi-client Predicate-only Encryption – Security

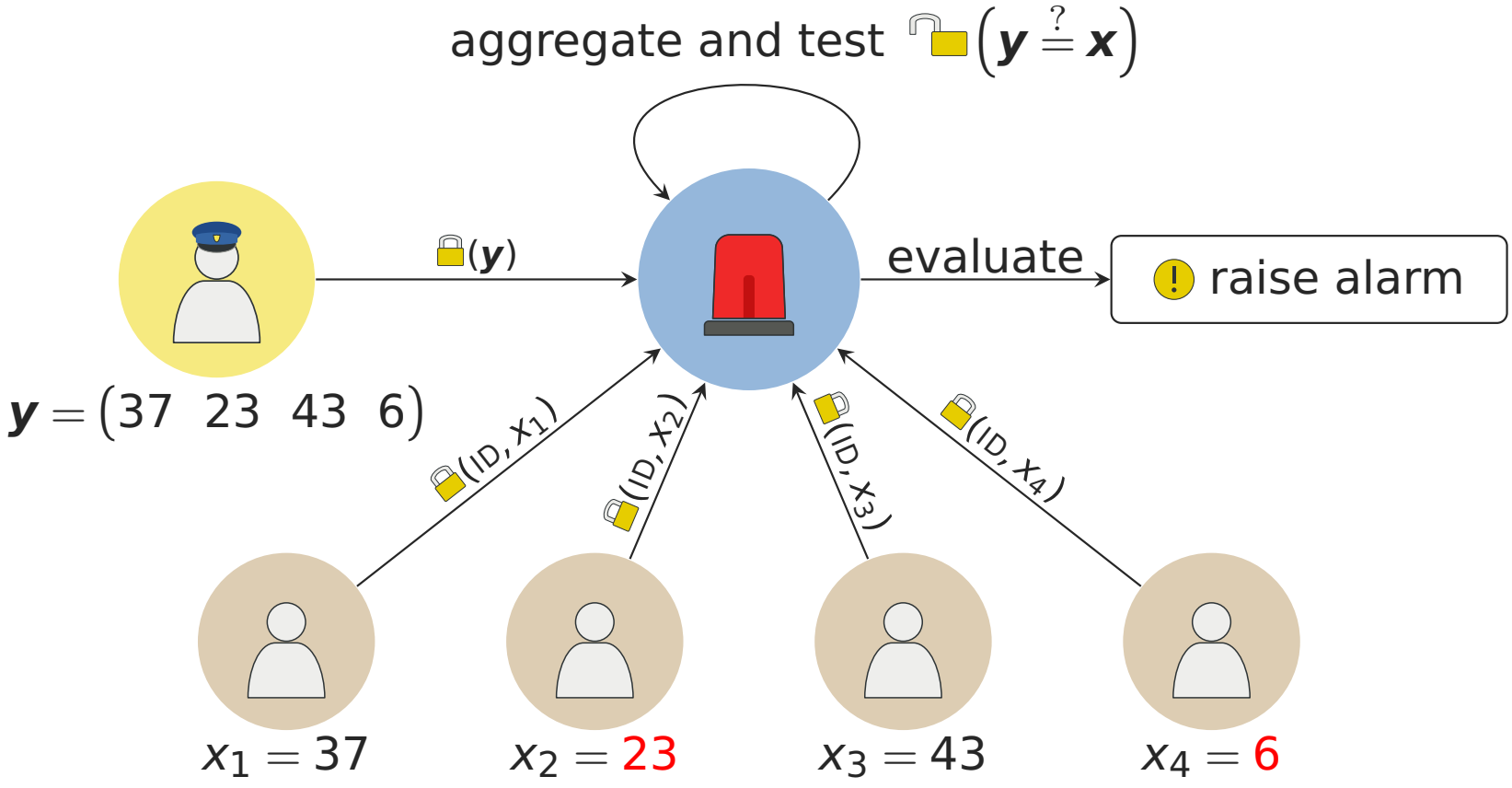
mix-and-match attacks prevention



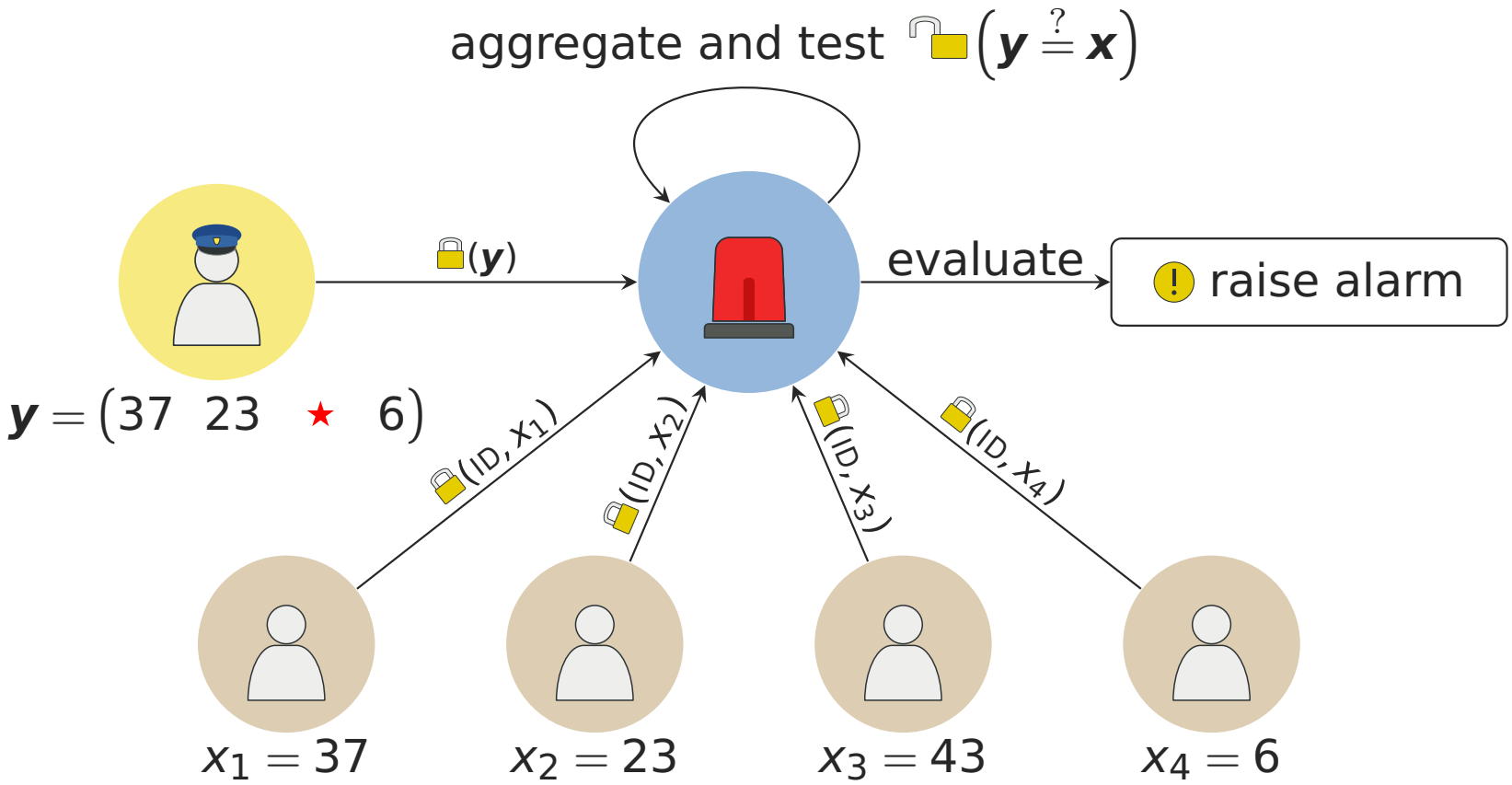
Construction: Schematic Overview



Construction: Schematic Overview



Construction: Schematic Overview



Construction: Simplified & Highlights

Setup(1^λ)

- prime-order asymmetric pairing $e\colon \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$
- hash function $H\colon \{0,1\}^* \to \mathbb{G}_1$
- $\mathsf{usk}_i = ({g_1}^{\alpha_i}, \quad)$
- $\mathsf{msk} = \left\{({g_2}^{\alpha_i}, \quad)\right\}_{i\in[n]}$

Construction: Simplified & Highlights

Setup(1^λ)

- prime-order asymmetric pairing $e\colon \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$
- hash function $H\colon \{0,1\}^* \to \mathbb{G}_1$
- $\mathsf{usk}_i = (g_1^{\alpha_i}, \quad)$
- $\mathsf{msk} = \left\{(g_2^{\alpha_i}, \quad)\right\}_{i\in[n]}$

Encrypt($\mathsf{usk}_i, \mathsf{ID}, x_i$)

$$\mathsf{ct}_i = \left(\quad , g_1^{r_i}, g_1^{\alpha_i x_i r_i} \quad \right)$$

GenToken($\mathsf{msk}, \boldsymbol{y}$)

$$\mathsf{tk}_{\boldsymbol{y}} = \left(\left\{ g_2^{u_i}, g_2^{\alpha_i y_i u_i} \right\}_{i\in[n]}, \quad \right)$$

Test($\mathsf{tk}_{\boldsymbol{y}}, \{\mathsf{ct}_i\}_{i\in[n]}$)

$$\prod_{i\in[n]} e\left(g_1^{\alpha_i x_i r_i} \quad , g_2^{u_i}\right) \stackrel{?}{=} \prod_{i\in[n]} e\left(g_1^{r_i}, g_2^{\alpha_i y_i u_i}\right)$$

Construction: Simplified & Highlights

Setup(1^λ)

- prime-order asymmetric pairing $e\colon \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$
- hash function $H\colon \{0,1\}^* \to \mathbb{G}_1$
- $\mathsf{usk}_i = (g_1^{\alpha_i}, \delta_i)$
- $\mathsf{msk} = \left\{(g_2^{\alpha_i}, g_2^{\delta_i})\right\}_{i\in[n]}$

Encrypt(usk_i, ID, x_i)

$\mathsf{ct}_i = \left(H(\mathsf{ID}), g_1^{r_i}, g_1^{\alpha_i x_i r_i} H(\mathsf{ID})^{\delta_i}\right)$

GenToken(msk, $\boldsymbol{y}$)

$$\mathsf{tk}_{\boldsymbol{y}} = \left(\left\{g_2^{u_i}, g_2^{\alpha_i y_i u_i}\right\}_{i\in[n]}, \prod_{i\in[n]} (g_2^{\delta_i})^{u_i} \right)$$

Test($\mathsf{tk}_{\boldsymbol{y}}$, $\{\mathsf{ct}_i\}_{i\in[n]}$)

$$\prod_{i\in[n]} e\left(g_1^{\alpha_i x_i r_i} H(\mathsf{ID})^{\delta_i}, g_2^{u_i}\right) \stackrel{?}{=} \prod_{i\in[n]} e\left(g_1^{r_i}, g_2^{\alpha_i y_i u_i}\right) e\left(H(\mathsf{ID}), \prod_{i\in[n]} (g_2^{\delta_i})^{u_i}\right)$$

Construction: Simplified & Highlights

Setup(1^λ)

- prime-order asymmetric pairing $e\colon \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$
- hash function $H\colon \{0,1\}^* \to \mathbb{G}_1$
- $\mathsf{usk}_i = (g_1{}^{\alpha_i}, \delta_i)$
- $\mathsf{msk} = \left\{(g_2{}^{\alpha_i}, g_2^{\delta_i})\right\}_{i\in[n]}$

Encrypt($\mathsf{usk}_i, \mathsf{ID}, x_i$)

$\mathsf{ct}_i = \left(H(\mathsf{ID}), g_1^{r_i}, g_1{}^{\alpha_i \pi_i(x_i) r_i} H(\mathsf{ID})^{\delta_i}\right)$

GenToken($\mathsf{msk}, \boldsymbol{y}$)

$$\mathsf{tk}_{\boldsymbol{y}} = \left(\left\{g_2^{u_i}, g_2{}^{\alpha_i \pi_i(y_i) u_i}\right\}_{i\in[n]}, \prod_{i\in[n]} (g_2^{\delta_i})^{u_i} \right)$$

Test($\mathsf{tk}_{\boldsymbol{y}}, \{\mathsf{ct}_i\}_{i\in[n]}$)

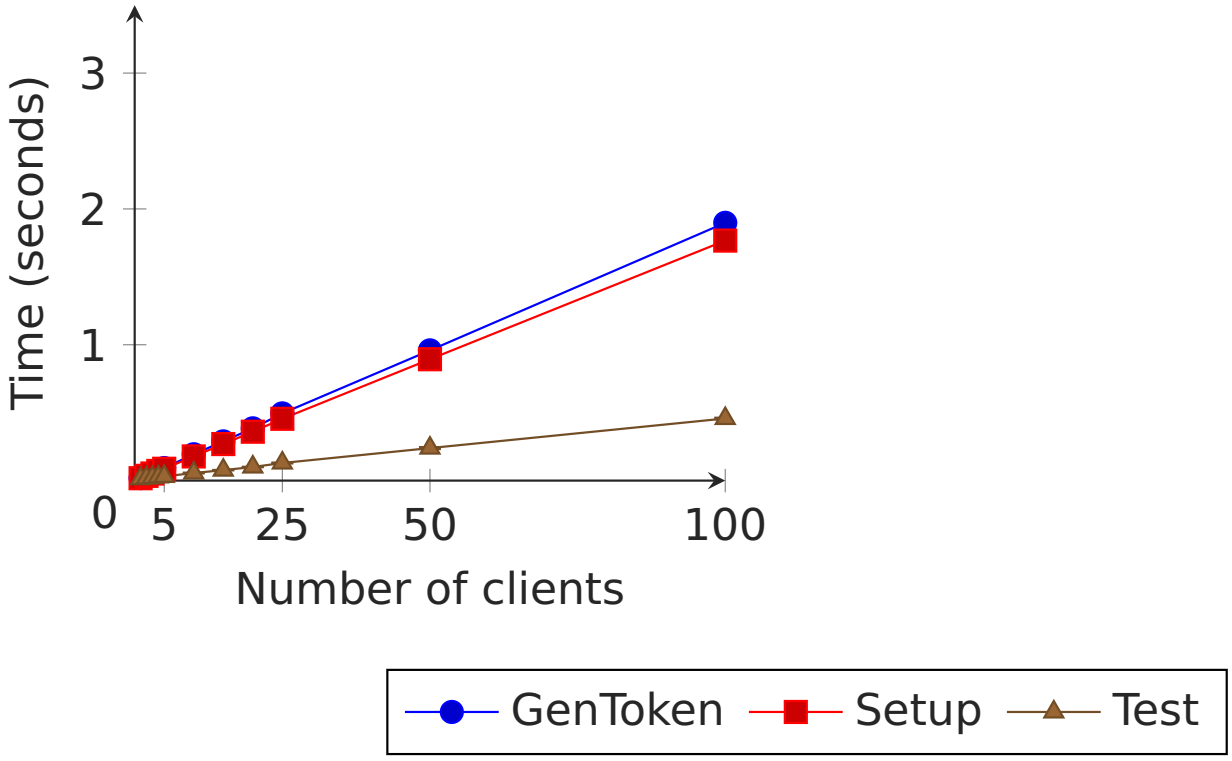
$$\prod_{i\in[n]} e\left(g_1{}^{\alpha_i \pi_i(x_i) r_i} H(\mathsf{ID})^{\delta_i}, g_2^{u_i}\right) \stackrel{?}{=} \prod_{i\in[n]} e\left(g_1^{r_i}, g_2{}^{\alpha_i \pi_i(y_i) u_i}\right) e\left(H(\mathsf{ID}), \prod_{i\in[n]} (g_2^{\delta_i})^{u_i}\right)$$

Evaluation

Proof-of-concept implementation in Go [CRIPTIM].

MNT-159 curve

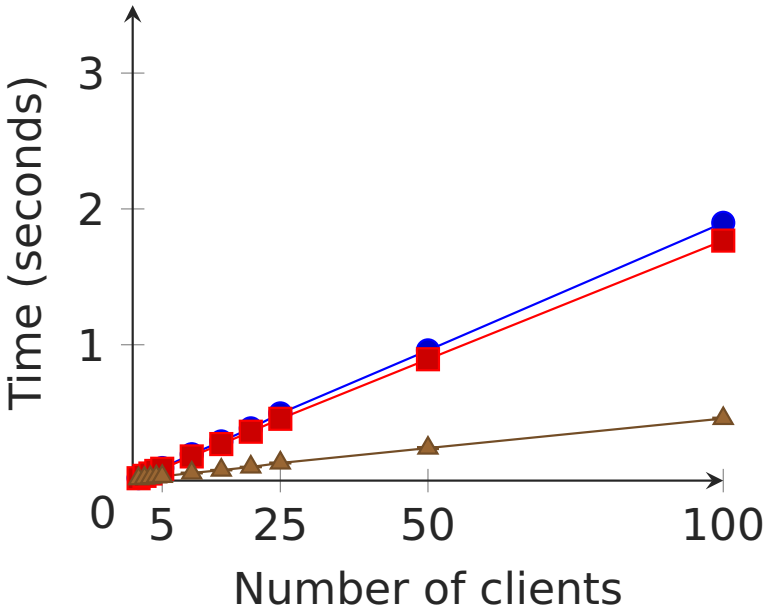
Encrypt (single client): 2.6 ms



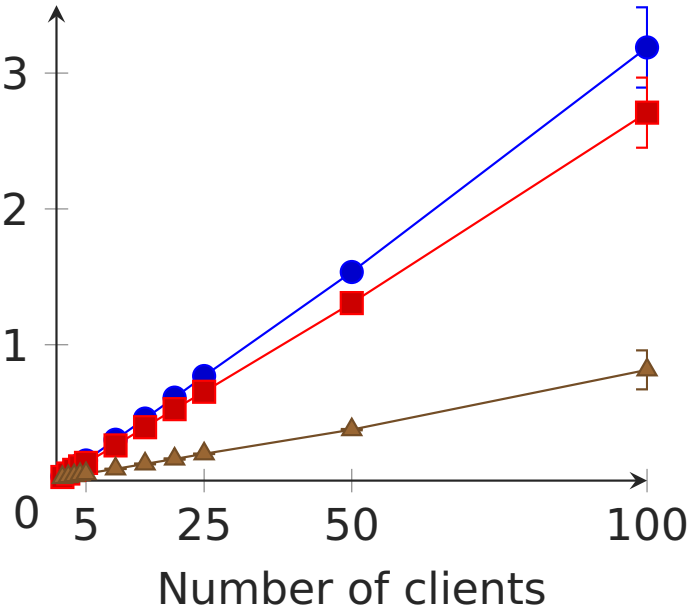
Evaluation

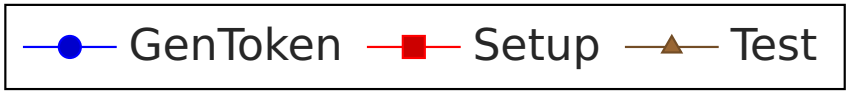
Proof-of-concept implementation in Go [CRIPTIM].

MNT-159 curve
Encrypt (single client): 2.6 ms



MNT-224 curve
Encrypt (single client): 4.4 ms





Summary

- Monitoring over encrypted data
- Defined multi-client predicate-only encryption
- Simple and efficient construction for conjunctive equality tests

Summary

- Monitoring over encrypted data
- Defined multi-client predicate-only encryption
- Simple and efficient construction for conjunctive equality tests

Interested?

Contact: t.r.vandekamp@utwente.nl

References

[CRIPTIM] *Implementations of Private Information Sharing Schemes*. CRIPTIM consortium. URL: https://github.com/CRIPTIM/.