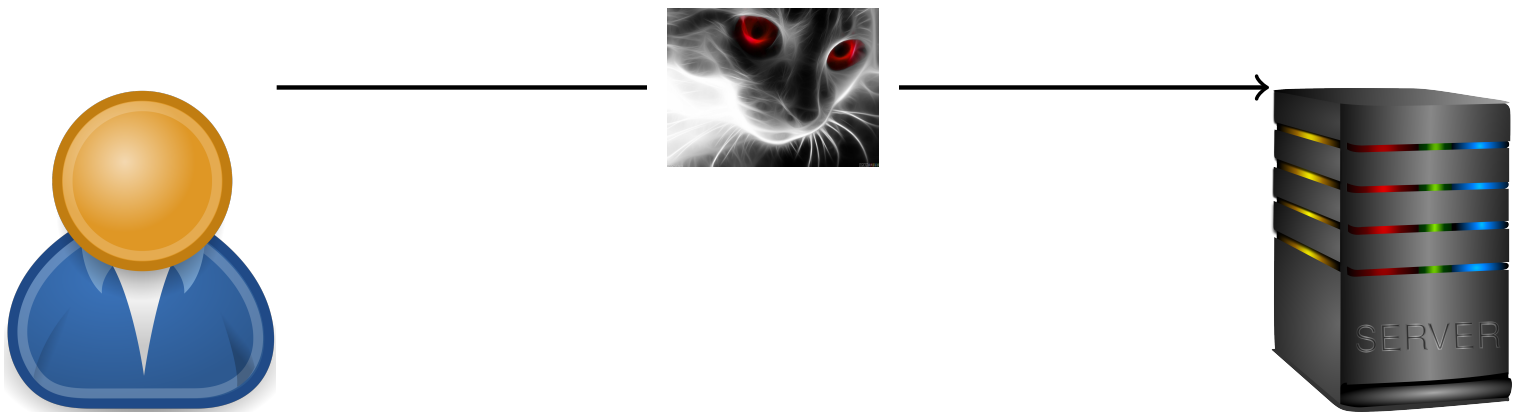
FHE Circuit Privacy Almost For Free

Florian Bourse, Rafaël Del Pino, Michele Minelli, and Hoeteck Wee

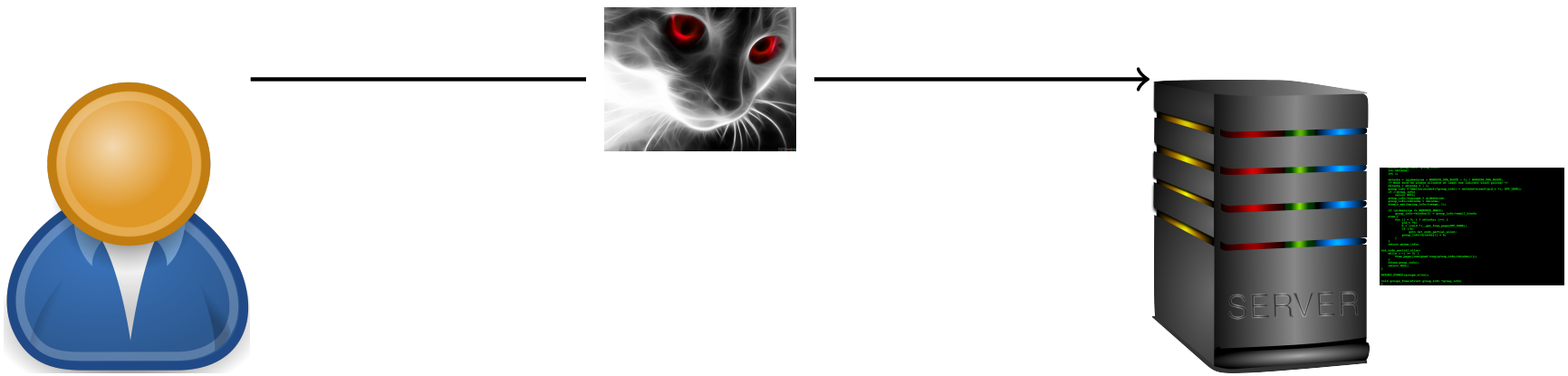
CNRS, École normale supérieure, INRIA, PSL, Paris, France

Workshop HEAT 2016 — Paris, France
Tuesday, July 5

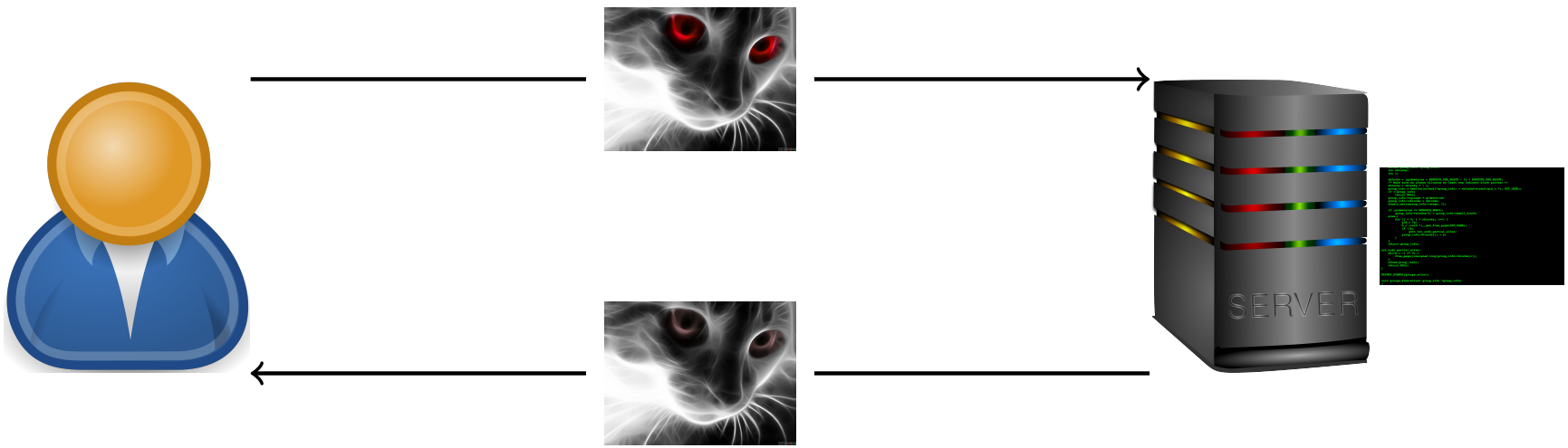
Example: red-eye removal



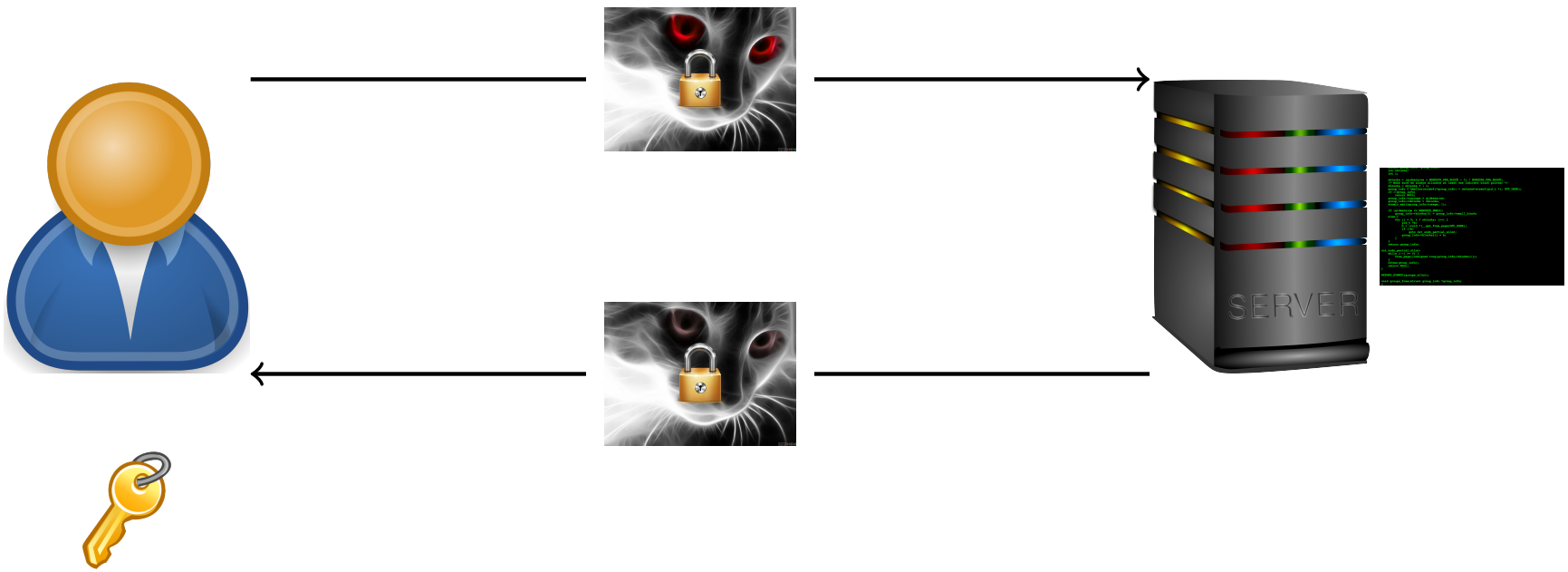
Example: red-eye removal



Example: red-eye removal



Data privacy: FHE



GSW encryption scheme [GenSahWat13]

$$\mathbf{G} = \mathbf{Id}_n \otimes \mathbf{g}, \quad \mathbf{g} = \left(1, 2, \ldots, 2^k\right)$$

$$\mathbf{C} = \mathsf{Enc}(\mu) = \begin{pmatrix} \mathbf{A} \\ \mathbf{sA} + \mathbf{e} \end{pmatrix} + \mu\mathbf{G} \in \mathbb{Z}_q^{n \times m}$$

GSW encryption scheme [GenSahWat13]

$$\mathbf{G} = \mathbf{Id}_n \otimes \mathbf{g}, \quad \mathbf{g} = \left(1, 2, \ldots, 2^k\right)$$

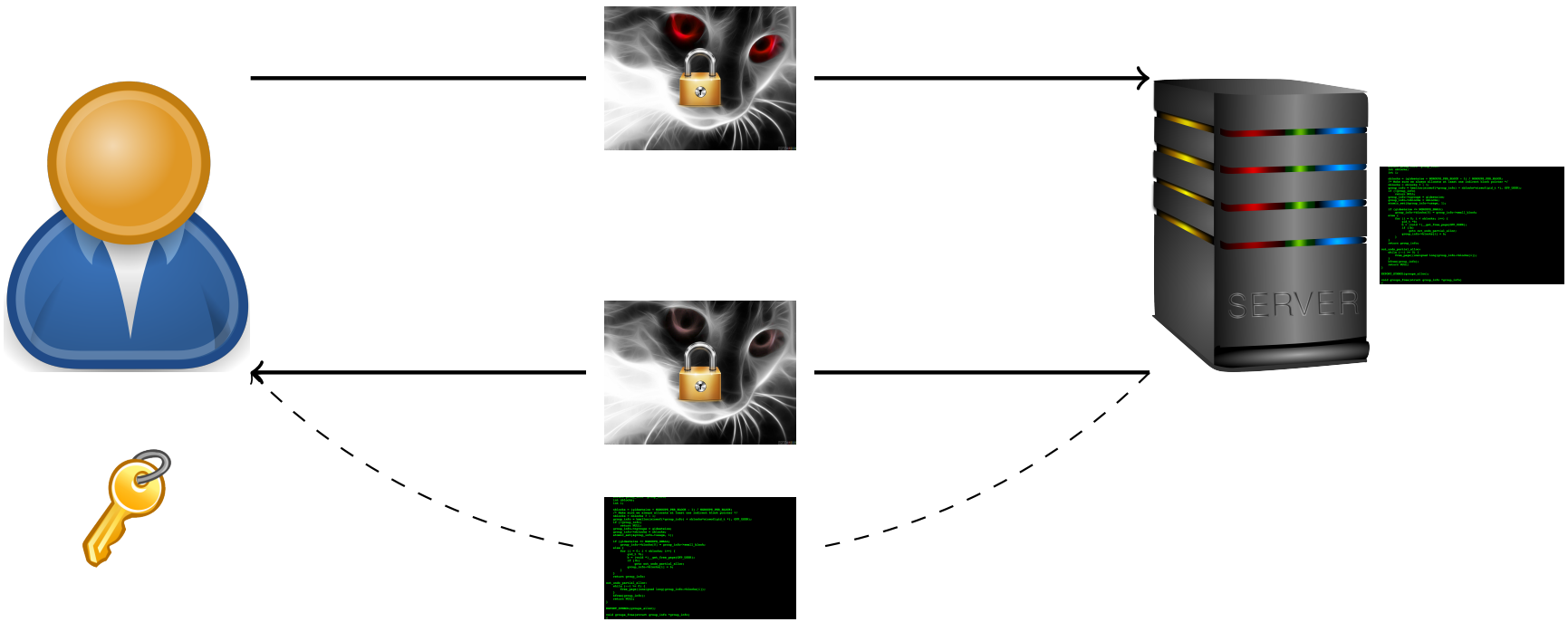
$$\mathbf{C} = \mathsf{Enc}(\mu) = \begin{pmatrix} \mathbf{A} \\ \mathbf{sA} + \mathbf{e} \end{pmatrix} + \mu\mathbf{G} \in \mathbb{Z}_q^{n \times m}$$

Sum $\mathsf{Enc}(\mu_1) + \mathsf{Enc}(\mu_2)$

Product $\mathsf{Enc}(\mu_1) \cdot \mathbf{G}^{-1}(\mathsf{Enc}(\mu_2))$

where $\forall\ \mathbf{v} \in \mathbb{Z}_q^n$, $\mathbf{G}^{-1}(\mathbf{v}) \in \mathbb{Z}_q^m$ is *small* and s.t. $\mathbf{G} \cdot \mathbf{G}^{-1}(\mathbf{v}) = \mathbf{v}$

Protecting the algorithm: circuit privacy



Leakage in the error term: toy example

Given $\mathbf{s}$, and 3 encryptions of 0:

$$\mathbf{C_1} = \begin{pmatrix} \mathbf{A}_1 \\ \mathbf{sA}_1 + \mathbf{e}_1 \end{pmatrix}, \mathbf{C_2} = \begin{pmatrix} \mathbf{A}_2 \\ \mathbf{sA}_2 + \mathbf{e}_2 \end{pmatrix}, \mathbf{C_3} = \begin{pmatrix} \mathbf{A}_3 \\ \mathbf{sA}_3 + \mathbf{e}_3 \end{pmatrix}.$$

Leakage in the error term: toy example

Given $\mathbf{s}$, and 3 encryptions of 0:

$$\mathbf{C_1} = \begin{pmatrix} \mathbf{A}_1 \\ \mathbf{sA}_1 + \mathbf{e}_1 \end{pmatrix}, \mathbf{C_2} = \begin{pmatrix} \mathbf{A}_2 \\ \mathbf{sA}_2 + \mathbf{e}_2 \end{pmatrix}, \mathbf{C_3} = \begin{pmatrix} \mathbf{A}_3 \\ \mathbf{sA}_3 + \mathbf{e}_3 \end{pmatrix}.$$

$\mathbf{C_i} + \mathbf{C_j}$ leaks i and j:

Leakage in the error term: toy example

Given $\mathbf{s}$, and 3 encryptions of 0:

$$\mathbf{C_1} = \begin{pmatrix} \mathbf{A}_1 \\ \mathbf{sA}_1 + \mathbf{e}_1 \end{pmatrix}, \mathbf{C_2} = \begin{pmatrix} \mathbf{A}_2 \\ \mathbf{sA}_2 + \mathbf{e}_2 \end{pmatrix}, \mathbf{C_3} = \begin{pmatrix} \mathbf{A}_3 \\ \mathbf{sA}_3 + \mathbf{e}_3 \end{pmatrix}.$$

$\mathbf{C_i} + \mathbf{C_j}$ leaks i and j:
The error term is $\mathbf{e}_i + \mathbf{e}_j$!

Protecting the algorithm: circuit privacy

Eval$(f, \mathbf{C}_1, \ldots, \mathbf{C}_\ell)$ should reveal nothing on f but $f(\mu_1, \ldots, \mu_\ell)$.

Noise flooding [Gen09]

$$\mathbf{C}_f = \mathsf{Eval}(f, \mathbf{C}_1, \ldots, \mathbf{C}_n),$$

- $\mathbf{C}_f = \mathbf{C}_f + \begin{pmatrix} \mathbf{0} \\ \mathbf{e}' \end{pmatrix}, \quad q \gg \mathbf{e}' \gg \mathbf{e}_f$

Pros Destroys all information contained in the noise

Cons Requires superpolynomial modulus, not multi-hop

Soak-spin-repeat [DucSte16]

$$\mathbf{C}_f = \mathsf{Eval}(f, \mathbf{C}_1, \ldots, \mathbf{C}_n),$$

- $\mathbf{C}_f = \mathbf{C}_f + \begin{pmatrix} \mathbf{0} \\ \mathbf{e}' \end{pmatrix}, \quad \mathbf{e}' \approx \mathbf{e}_f$
- $\mathbf{C}_f = \mathsf{Eval}(\mathsf{Dec}(\cdot, \mathbf{C_f}), \mathsf{Enc}(sk)))$
- Repeat $O(\lambda)$ times

Pros Works with polynomial modulus, multi-hop

Cons Requires circular security (bootstrapping)

Our approach [BDMW16]

$$\mathbf{C}_f = \mathsf{Eval}(f, \mathbf{C}_1, \ldots, \mathbf{C}_n),$$

- $\mathbf{C}_f = \mathbf{C}_f + \begin{pmatrix} \mathbf{0} \\ \mathbf{e}' \end{pmatrix}, \quad \mathbf{e}' \approx \mathbf{e}_f$

Pros Polynomial modulus, no circular security, multi-hop

Cons Only for NC^1 evaluations on GSW, leaks $|f|$

Variant on discrete Gaussian leftover hash lemma

For any small $\mathbf{e} \in \mathbb{Z}_q^k, v \in \mathbb{Z}_q$,

$$\left\langle \mathbf{e}, \mathbf{g}^{-1}(v) \right\rangle + e' \approx_s y,$$

where

- $\mathbf{g}^{-1}(v)$ is a *discrete Gaussian* conditioned on the fact that $\left\langle \mathbf{g}, \mathbf{g}^{-1}(v) \right\rangle = v$,
- y is a *discrete Gaussian* with parameter $\widetilde{O}(\|\mathbf{e}\|)$.

Some intuition

$$\left\langle \mathbf{e}, \mathbf{g}^{-1}(v) \right\rangle + e' \approx_s y,$$

Some intuition

$$\left\langle \mathbf{e}, \mathbf{g}^{-1}(v) \right\rangle + e' \approx_s y,$$

- A sum of Gaussians is Gaussian:
  $\left\langle \mathbf{e}, \mathbf{g}^{-1}(v) \right\rangle$ is a discrete Gaussian over a certain lattice.

Some intuition

$$\left\langle \mathbf{e}, \mathbf{g}^{-1}(v) \right\rangle + e' \approx_s y,$$

- A sum of Gaussians is Gaussian:
  $\left\langle \mathbf{e}, \mathbf{g}^{-1}(v) \right\rangle$ is a discrete Gaussian over a certain lattice.
- The second term ensures that the support is $\mathbb{Z}$.

Slightly more technical

$$\left\langle \mathbf{e}, \mathbf{g}^{-1}(v) \right\rangle + e' \approx_s y,$$

Let $\Lambda = \{\mathbf{x} \mid \langle \mathbf{g}, \mathbf{x} \rangle = 0 \bmod q\}$,

Slightly more technical

$$\left\langle \mathbf{e}, \mathbf{g}^{-1}(v) \right\rangle + e' \approx_s y,$$

Let $\Lambda = \{\mathbf{x} \mid \langle \mathbf{g}, \mathbf{x} \rangle = 0 \bmod q\}$,
$\Lambda_{\mathbf{e}} = \{(\mathbf{x}, t) \in \Lambda \times \mathbb{Z} \mid \langle \mathbf{x}, \mathbf{e} \rangle + t = 0\}$.

Slightly more technical

$$\left\langle \mathbf{e}, \mathbf{g}^{-1}(v) \right\rangle + e' \approx_s y,$$

Let $\Lambda = \{\mathbf{x} \mid \langle \mathbf{g}, \mathbf{x} \rangle = 0 \bmod q\}$,
$\Lambda_{\mathbf{e}} = \{(\mathbf{x}, t) \in \Lambda \times \mathbb{Z} \mid \langle \mathbf{x}, \mathbf{e} \rangle + t = 0\}$.
$(\mathbf{g}^{-1}(v), e')$ is a Gaussian on a coset of $\Lambda_{\mathbf{e}}$.

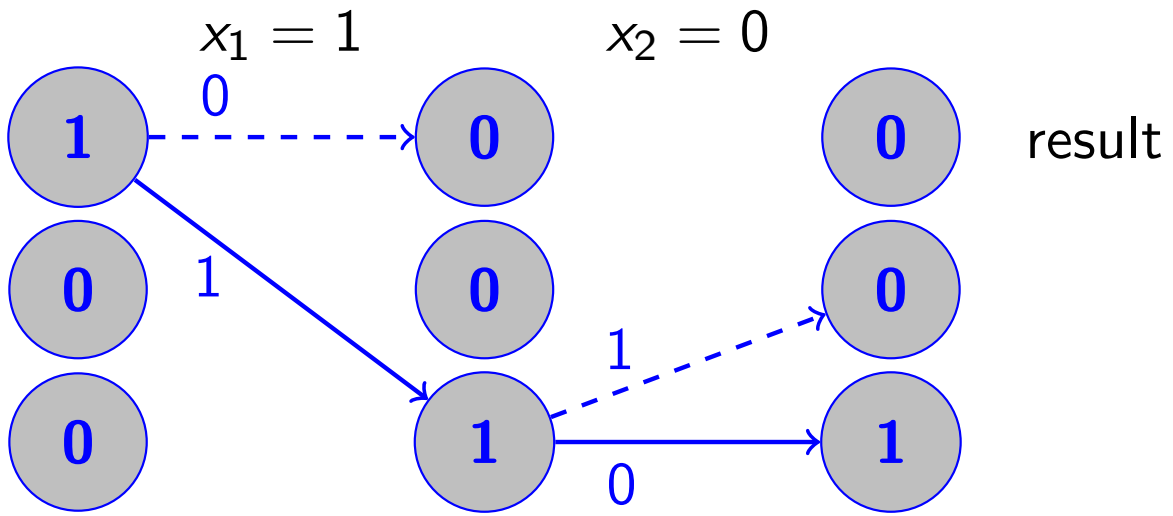
Slightly more technical

$$\left\langle \mathbf{e}, \mathbf{g}^{-1}(v) \right\rangle + e' \approx_s y,$$

Let $\Lambda = \{\mathbf{x} \mid \langle \mathbf{g}, \mathbf{x} \rangle = 0 \bmod q\}$,
$\Lambda_{\mathbf{e}} = \{(\mathbf{x}, t) \in \Lambda \times \mathbb{Z} \mid \langle \mathbf{x}, \mathbf{e} \rangle + t = 0\}$.
$(\mathbf{g}^{-1}(v), e')$ is a Gaussian on a coset of $\Lambda_{\mathbf{e}}$.
What we show:

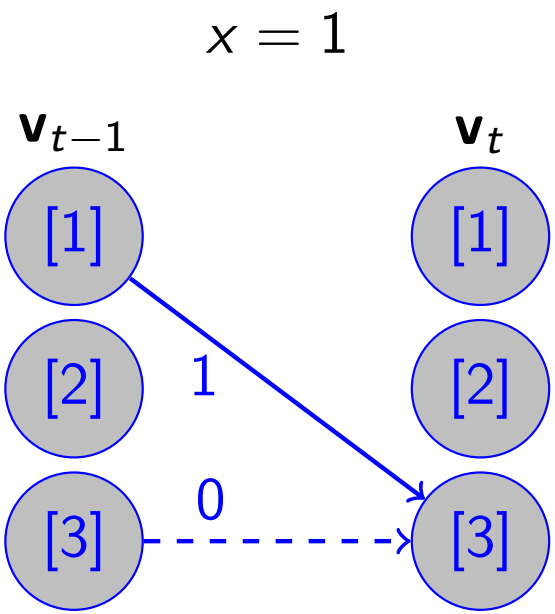
$$\eta_\varepsilon(\Lambda_{\mathbf{e}}) = \widetilde{O}(\|\mathbf{e}\|)$$

Branching programs



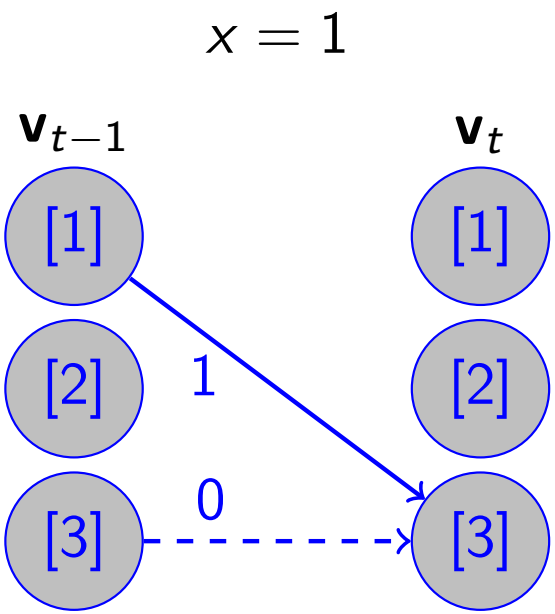
$$\text{result} = \left(\pi_{L,x_{var(L)}} \circ \ldots \circ \pi_{1,x_{var(1)}}(1) \stackrel{?}{=} 1\right)$$

One step of computation



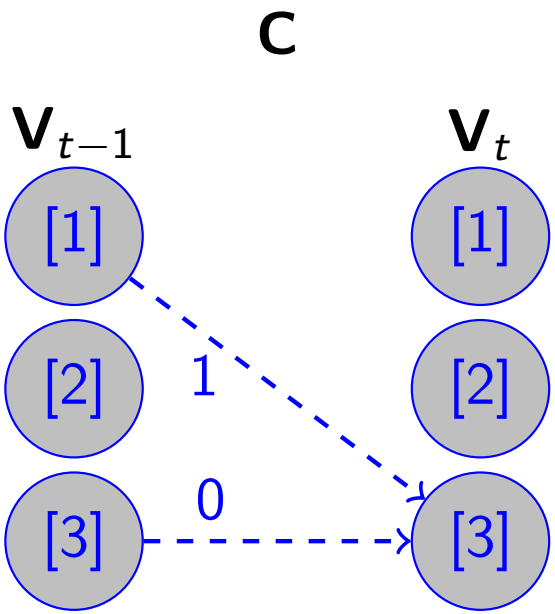
$$v_t[i] = v_{t-1}[\pi_{t,x}^{-1}(i)]$$

One step of computation



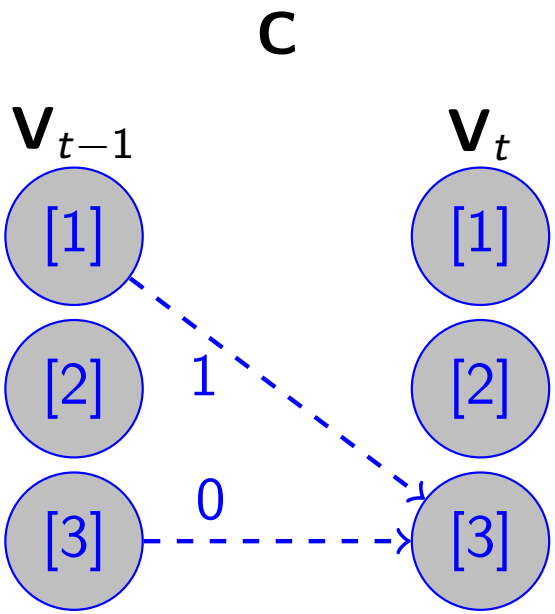
$$v_t[i] = v_{t-1}[\pi_{t,x}^{-1}(i)]$$
$$= x \cdot v_{t-1}[\pi_{t,1}^{-1}(i)] + (1-x) \cdot v_{t-1}[\pi_{t,0}^{-1}(i)]$$

Homomorphic evaluation



$$\mathbf{V}_t[i] = \mathbf{C} \cdot \mathbf{G}^{-1}(\mathbf{V}_{t-1}[\pi_{t,1}^{-1}(i)]) + (\mathbf{G} - \mathbf{C}) \cdot \mathbf{G}^{-1}(\mathbf{V}_{t-1}[\pi_{t,0}^{-1}(i)])$$

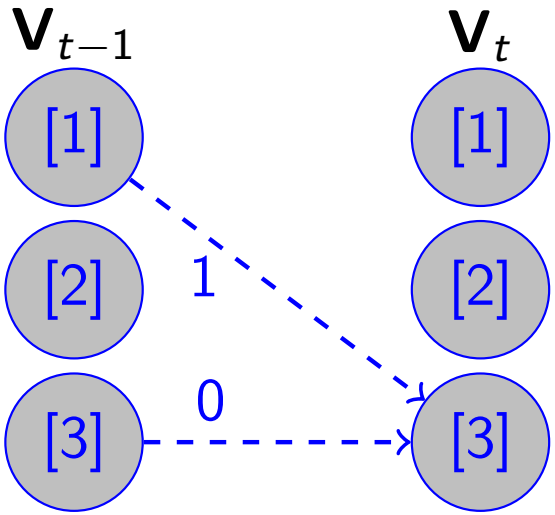
Homomorphic evaluation



$$\mathbf{V}_t[i] = \mathbf{C} \cdot \mathbf{G}^{-1}(\mathbf{V}_{t-1}[\pi_{t,1}^{-1}(i)]) + (\mathbf{G} - \mathbf{C}) \cdot \mathbf{G}^{-1}(\mathbf{V}_{t-1}[\pi_{t,0}^{-1}(i)]) + \begin{pmatrix} \mathbf{0} \\ \mathbf{e}' \end{pmatrix}$$

Homomorphic evaluation

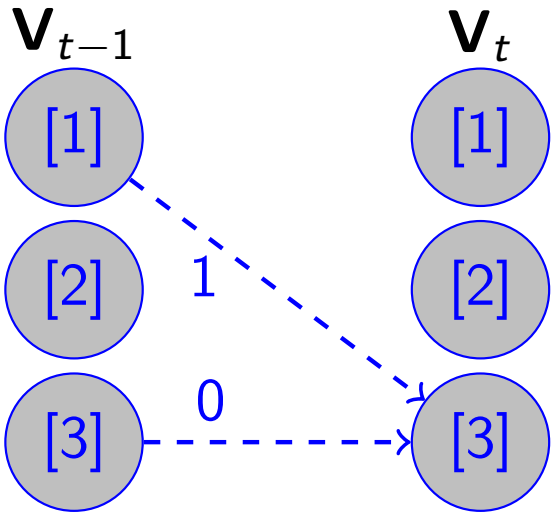
$$\mathbf{C} = \hat{\mathbf{A}} + x\mathbf{G}$$



$$\mathbf{V}_t[i] = \mathbf{V}_{t-1}[\pi_{t,x}^{-1}(i)] + \hat{\mathbf{A}} \cdot \left(\mathbf{G}^{-1}(\mathbf{V}_{t-1}[\pi_{t,1}^{-1}(i)]) - \mathbf{G}^{-1}(\mathbf{V}_{t-1}[\pi_{t,0}^{-1}(i)])\right) + \begin{pmatrix} \mathbf{0} \\ \mathbf{e}' \end{pmatrix}$$

Homomorphic evaluation

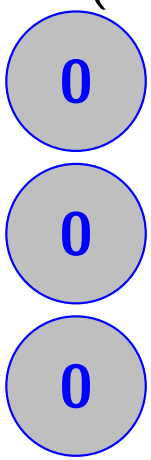
$$\mathbf{C} = \hat{\mathbf{A}} + x\mathbf{G}$$



$$\mathbf{V}_t[i] \approx_s \mathbf{V}_{t-1}[\pi_{t,x}^{-1}(i)] + \begin{pmatrix} \mathbf{B} \\ \mathbf{sB} + \mathbf{y} \end{pmatrix}$$

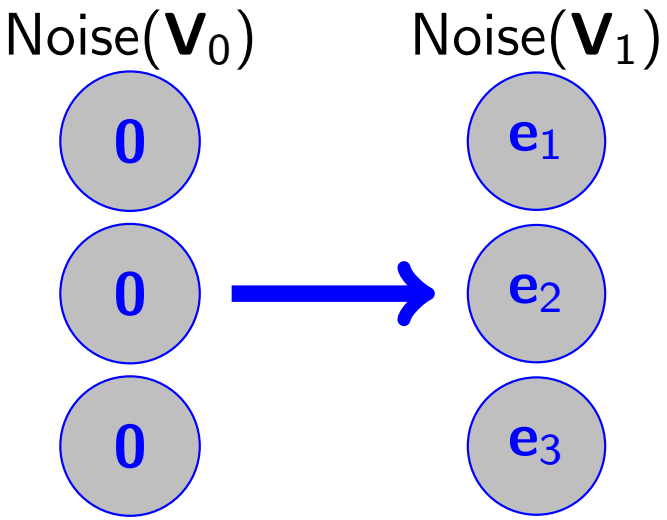
Circuit privacy by induction

$\mathsf{Noise}(\mathbf{V}_0)$



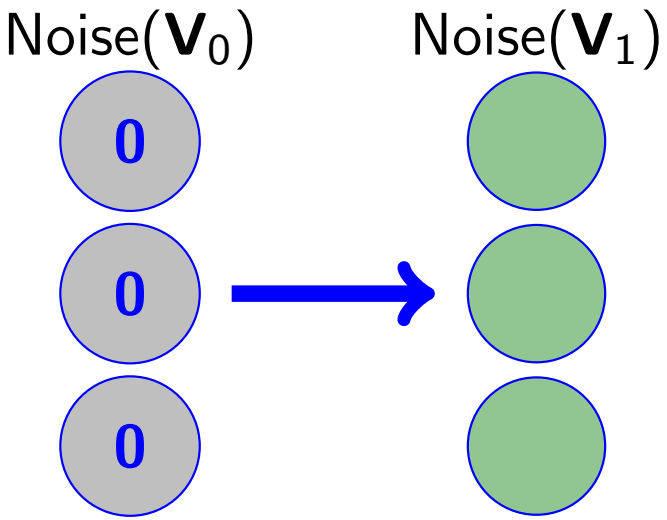
$$\mathsf{Noise}(\mathbf{V}_t[i]) \approx_s \mathsf{Noise}(\mathbf{V}_{t-1}[\pi^{-1}_{1,x_1}(i)]) + \mathbf{y}$$

Circuit privacy by induction



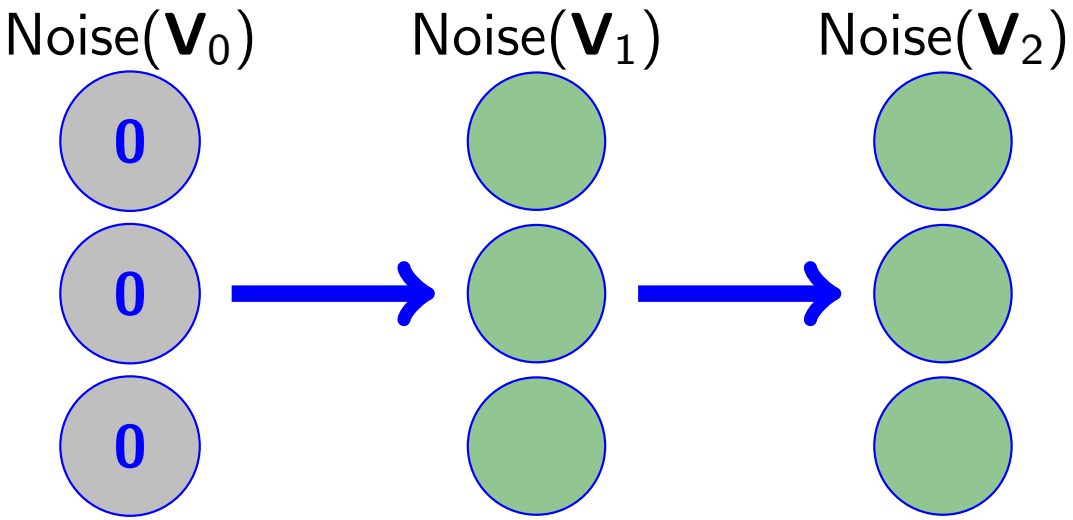
$$\text{Noise}(\mathbf{V}_t[i]) \approx_s \text{Noise}(\mathbf{V}_{t-1}[\pi^{-1}_{1,x_1}(i)]) + \mathbf{y}$$

Circuit privacy by induction



$$\text{Noise}(\mathbf{V}_t[i]) \approx_s \text{Noise}(\mathbf{V}_{t-1}[\pi^{-1}_{1,x_1}(i)]) + \mathbf{y}$$

Circuit privacy by induction



$$\text{Noise}(\mathbf{V}_t[i]) \approx_s \text{Noise}(\mathbf{V}_{t-1}[\pi^{-1}_{1,x_1}(i)]) + \mathbf{y}$$

Noise term independent of computation !

Thank you!

Questions?