On Tight Security Proofs for Schnorr Signatures

Nils Fleischhacker[1] Tibor Jager[2] Dominique Schröder[1]

[1]Saarland University

[2]Horst Görtz Institute for IT Security, Ruhr-University Bochum

December 9, 2014

(Informal) main Result

The security of Schnorr signatures cannot be *tightly* reduced to any *natural* assumption using a *generic* reduction.

The result holds unconditionally.

Schnorr Signatures [Sch90,Schn91]

$$\mathbb{G} = \langle g \rangle, \mathcal{H}$$

$\underline{\mathsf{Kgen}(1^\kappa)}$

$x \xleftarrow{\$} \mathbb{Z}_q$

$X := g^x$

return (x, X)

$\underline{\mathsf{Sign}(x, m)}$

$r \xleftarrow{\$} \mathbb{Z}_q$

$R := g^r$

$c := \mathcal{H}(R, m)$

$y := r + x \cdot c$

return $\sigma = (R, y)$

$\underline{\mathsf{Vrfy}(X, m, \sigma)}$

parse σ as (R, y)

$c := \mathcal{H}(R, m)$

return $g^y \stackrel{?}{=} X^c \cdot R$

- Provably secure under DLOG assumption in the ROM [PS96, PS00].
- Previous impossibility results for tight proofs for DLOG and algebraic reductions [PV05,GBL08,Seu12].

Schnorr Signatures [Sch90,Schn91]

$$\mathbb{G} = \langle g \rangle, \mathcal{H}$$

$\underline{\mathsf{Kgen}(1^\kappa)}$

$x \xleftarrow{\$} \mathbb{Z}_q$

$X := g^x$

return (x, X)

$\underline{\mathsf{Sign}(x, m)}$

$r \xleftarrow{\$} \mathbb{Z}_q$

$R := g^r$

$c := \mathcal{H}(R, m)$

$y := r + x \cdot c$

return $\sigma = (R, y)$

$\underline{\mathsf{Vrfy}(X, m, \sigma)}$

parse σ as (R, y)

$c := \mathcal{H}(R, m)$

return $g^y \stackrel{?}{=} X^c \cdot R$

- Provably secure under DLOG assumption in the ROM [PS96, PS00].
- Previous impossibility results for tight proofs for DLOG and algebraic reductions [PV05,GBL08,Seu12].

Schnorr Signatures [Sch90,Schn91]

$$\mathbb{G} = \langle g \rangle, \mathcal{H}$$

$\underline{\mathsf{Kgen}(1^\kappa)}$

$x \xleftarrow{\$} \mathbb{Z}_q$

$X := g^x$

return (x, X)

$\underline{\mathsf{Sign}(x, m)}$

$r \xleftarrow{\$} \mathbb{Z}_q$

$R := g^r$

$c := \mathcal{H}(R, m)$

$y := r + x \cdot c$

return $\sigma = (R, y)$

$\underline{\mathsf{Vrfy}(X, m, \sigma)}$

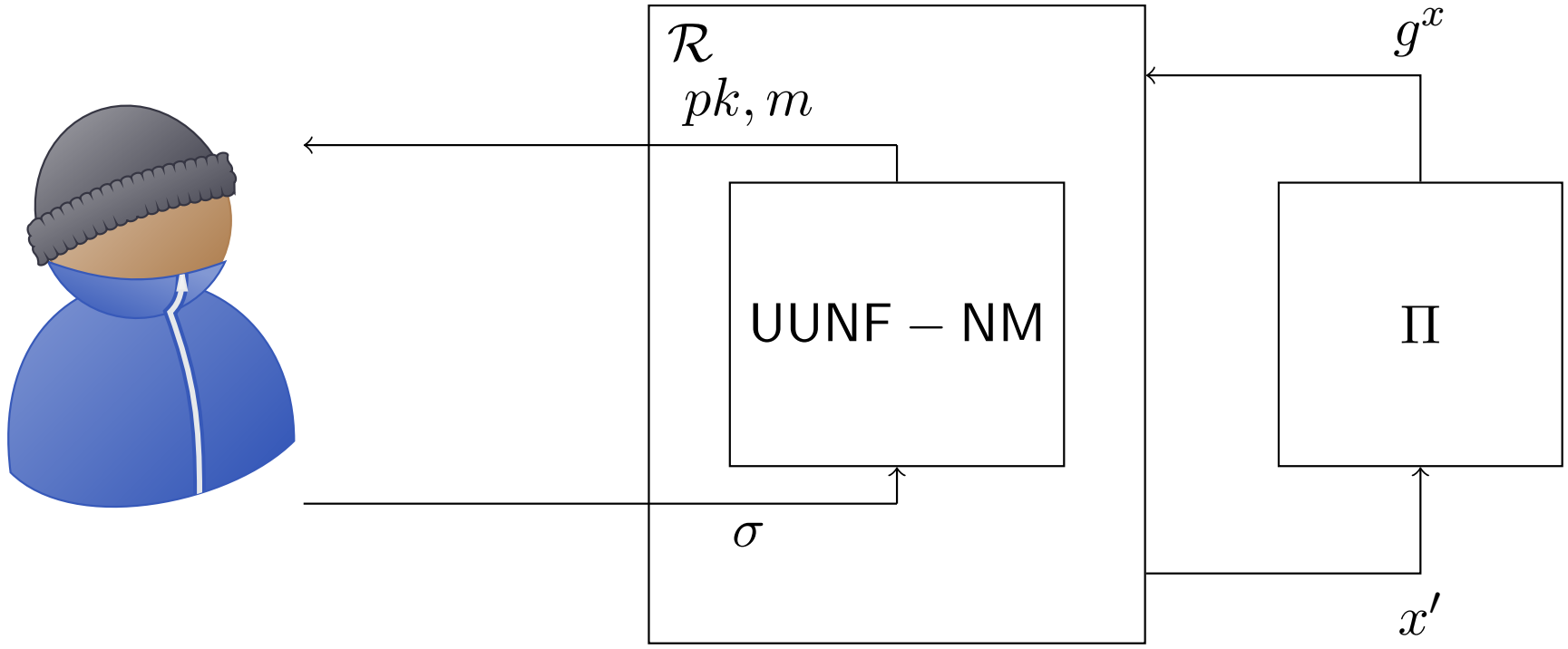
parse σ as (R, y)

$c := \mathcal{H}(R, m)$

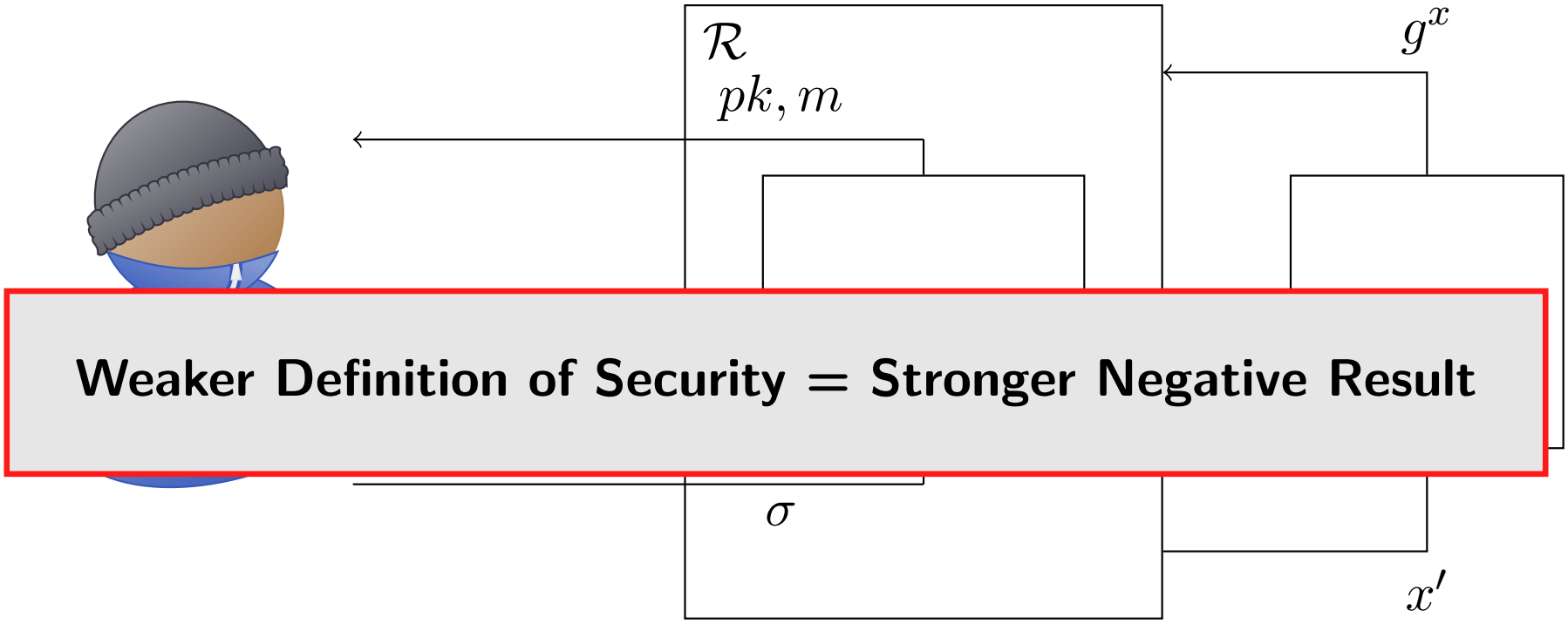
return $g^y \stackrel{?}{=} X^c \cdot R$

- Provably secure under DLOG assumption in the ROM [PS96, PS00]. Not tight!
- Previous impossibility results for tight proofs for DLOG and algebraic reductions [PV05,GBL08,Seu12].

Why do we care about tightness?

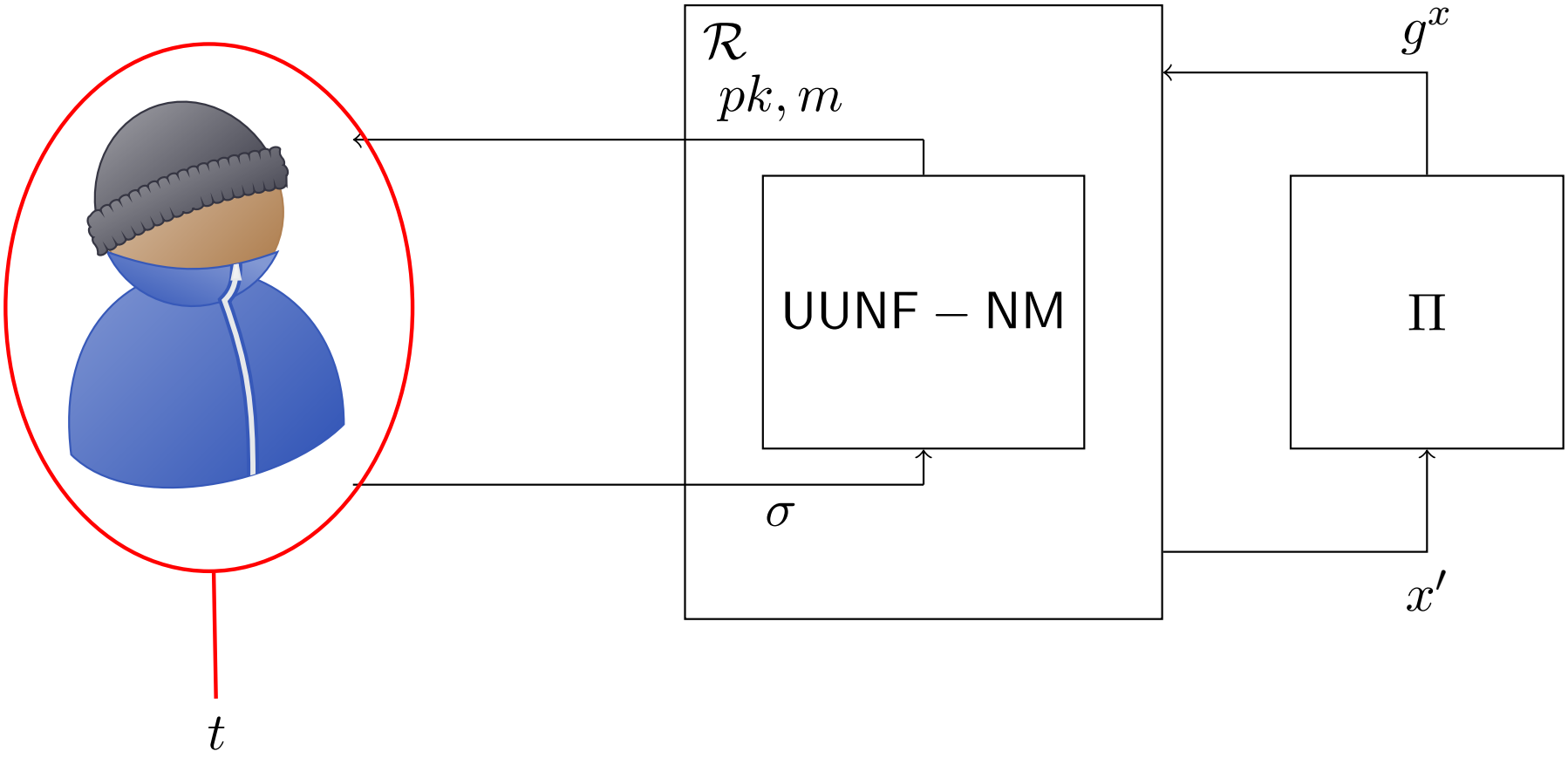


Why do we care about tightness?

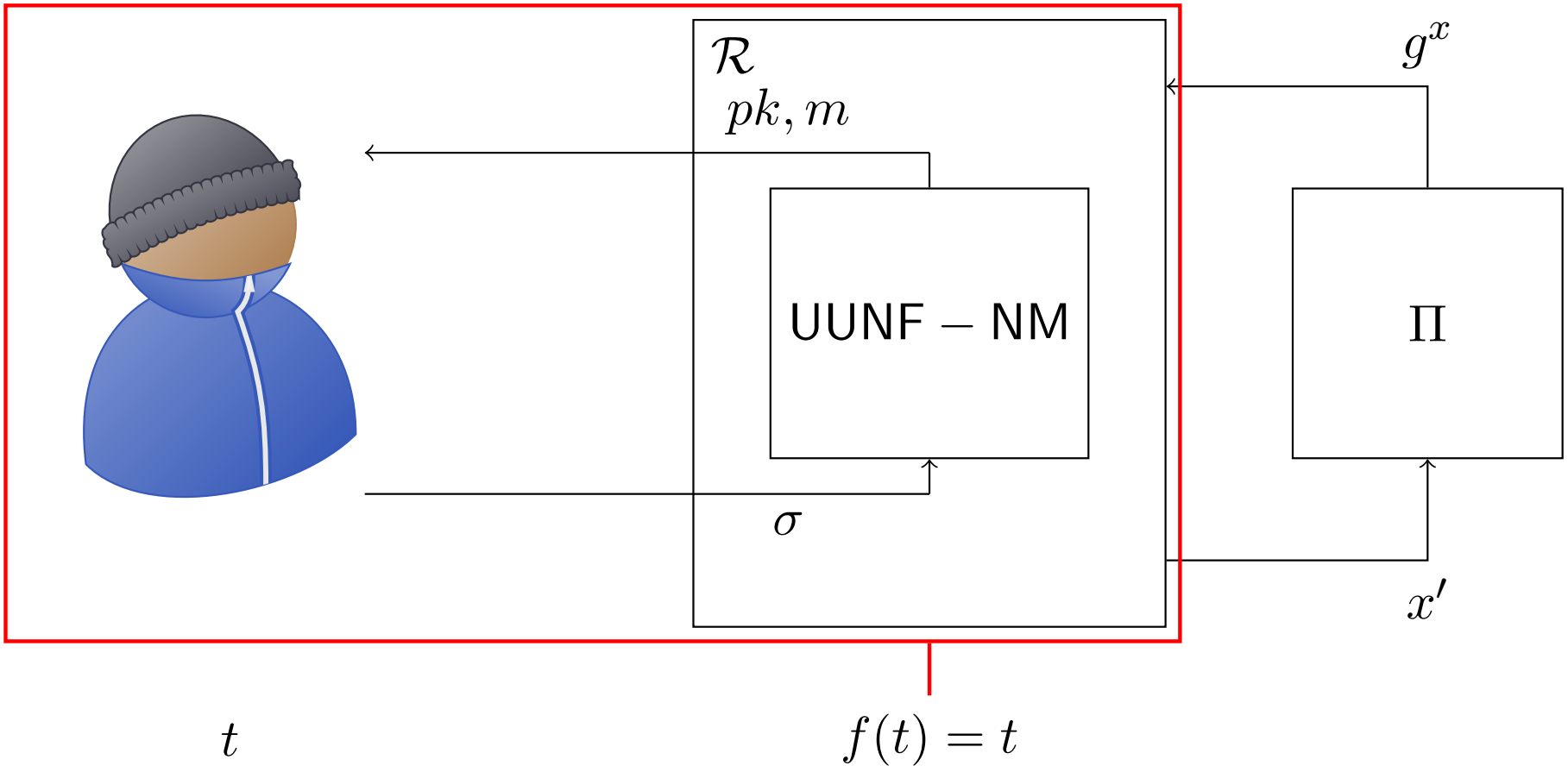


Weaker Definition of Security = Stronger Negative Result

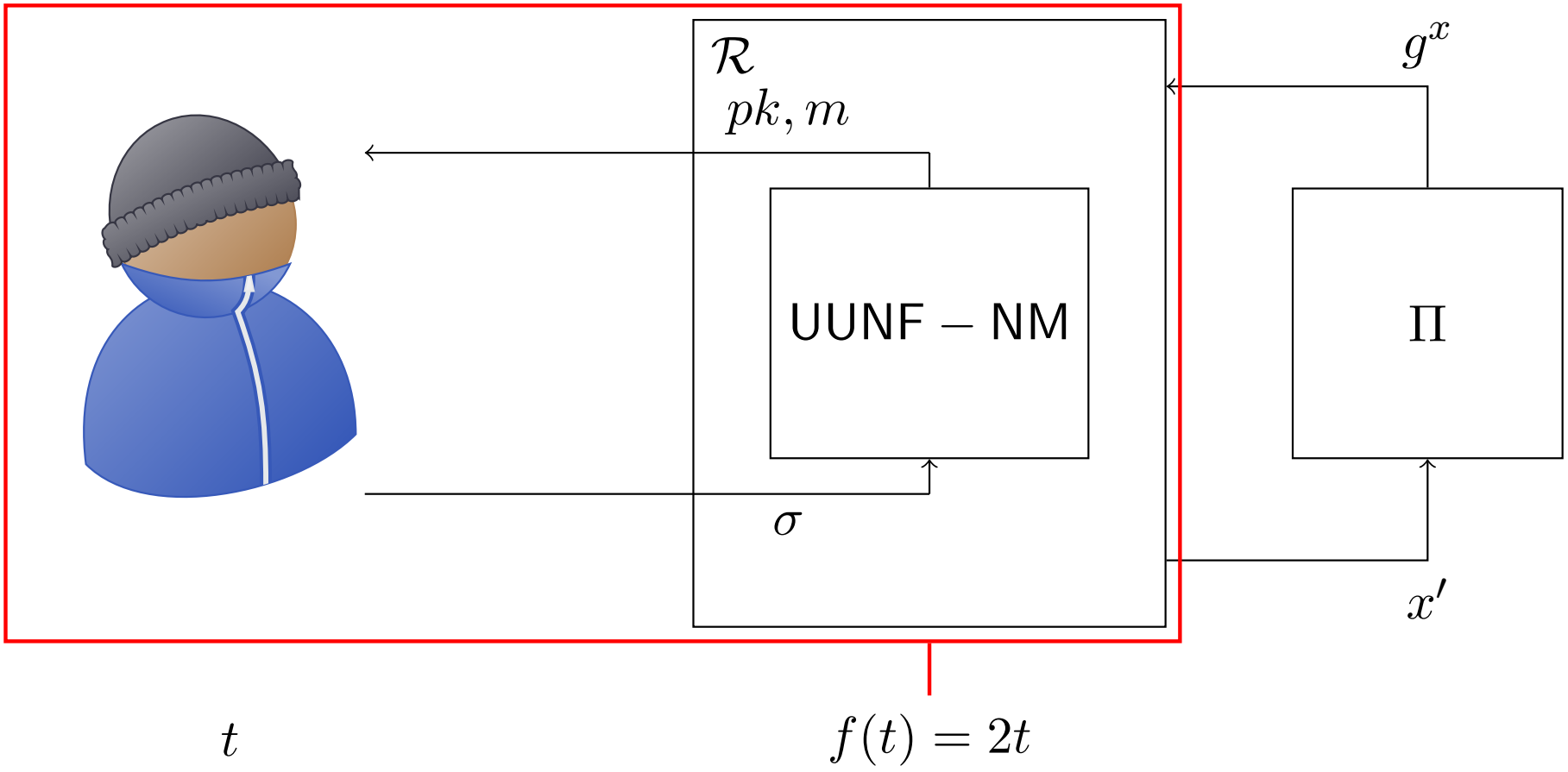
Why do we care about tightness?



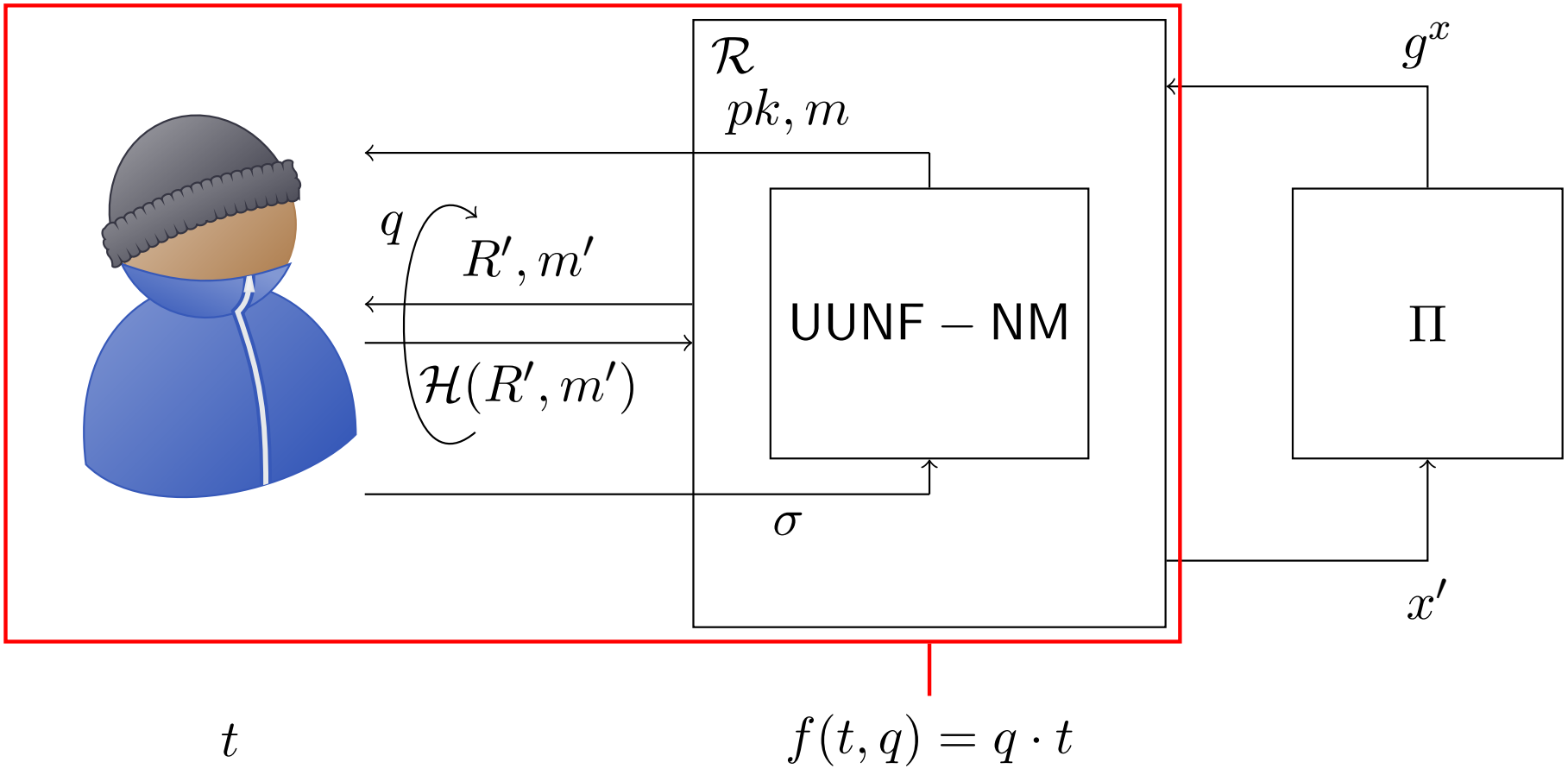
Why do we care about tightness?



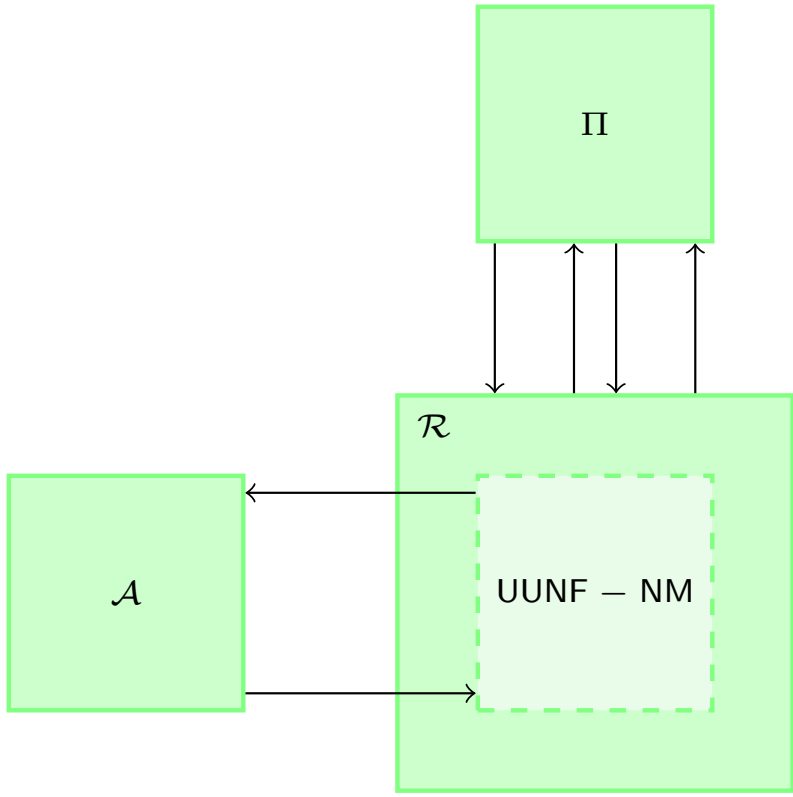
Why do we care about tightness?



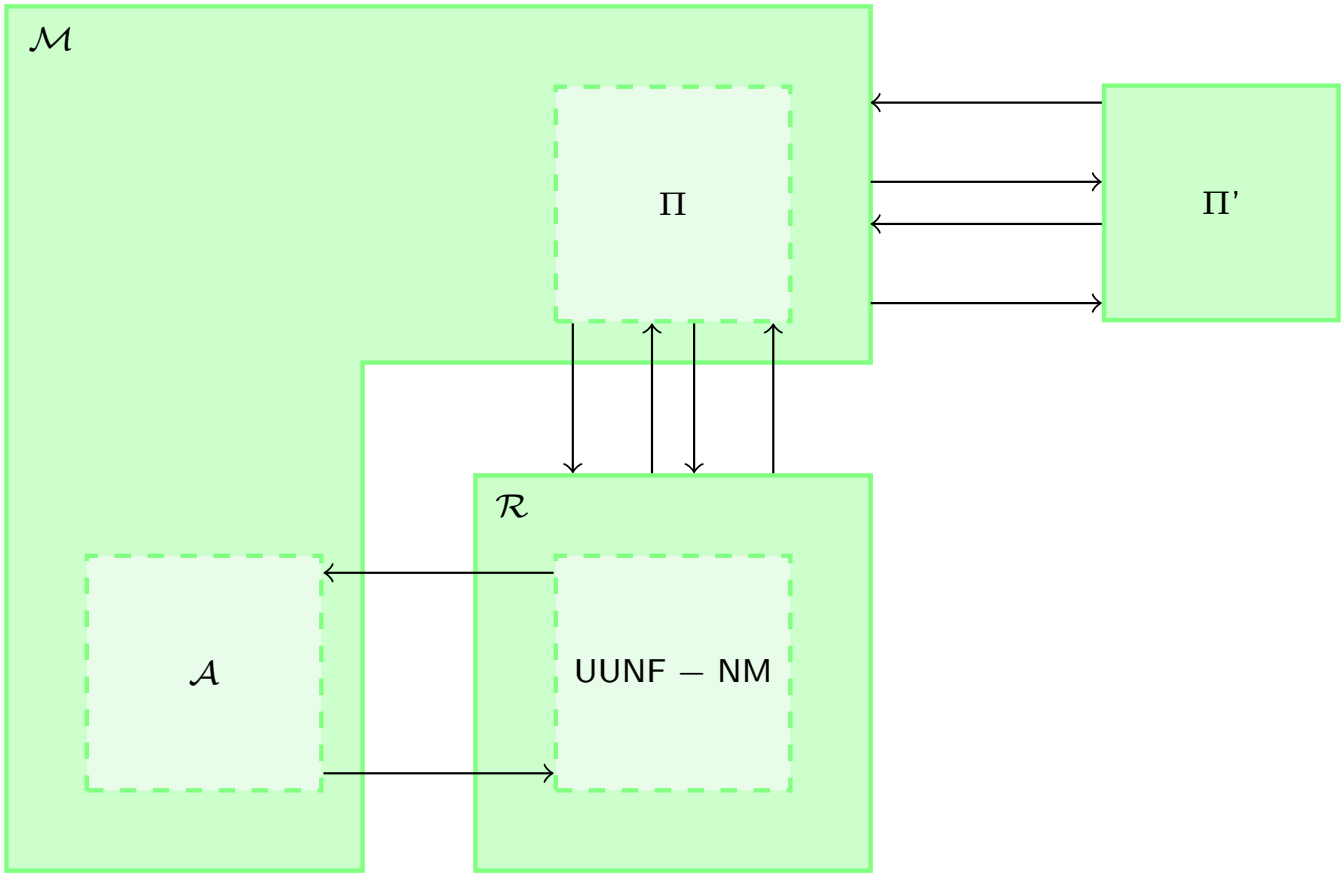
Why do we care about tightness?



Meta-Reductions [BV98]



Meta-Reductions [BV98]



Previous Work on Lower Bounds

	PV05	
Bound	$\frac{1}{q^{1/2}}$	
Reduction	algebraic (OM)DL	
Assumption	OMDL	

Previous Work on Lower Bounds

	PV05	GBL08	
Bound	$\frac{1}{q^{1/2}}$	$\frac{1}{q^{2/3}}$	
Reduction	algebraic (OM)DL	algebraic (OM)DL	
Assumption	OMDL	OMDL	

Previous Work on Lower Bounds

	PV05	GBL08	Seurin12	
Bound	$\frac{1}{q^{1/2}}$	$\frac{1}{q^{2/3}}$	$O(\frac{1}{q})$	
Reduction	algebraic (OM)DL	algebraic (OM)DL	algebraic (OM)DL	
Assumption	OMDL	OMDL	OMDL	

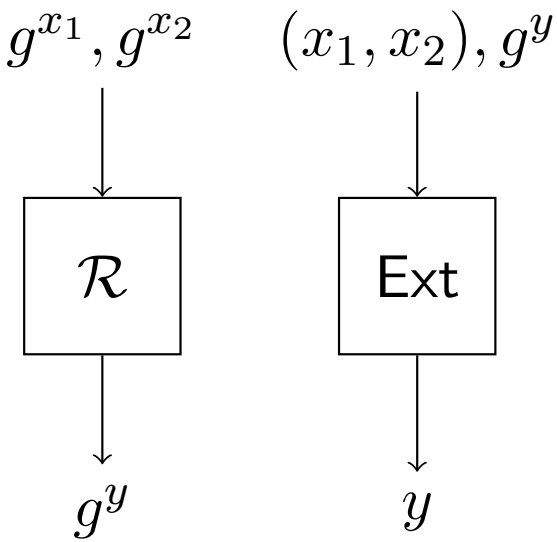
Previous Work on Lower Bounds

	PV05	GBL08	Seurin12	Our Work
Bound	$\frac{1}{q^{1/2}}$	$\frac{1}{q^{2/3}}$	$O(\frac{1}{q})$	$O(\frac{1}{q})$
Reduction	algebraic (OM)DL	algebraic (OM)DL	algebraic (OM)DL	generic representation invariant
Assumption	OMDL	OMDL	OMDL	none

Algebraic vs. Generic Reductions

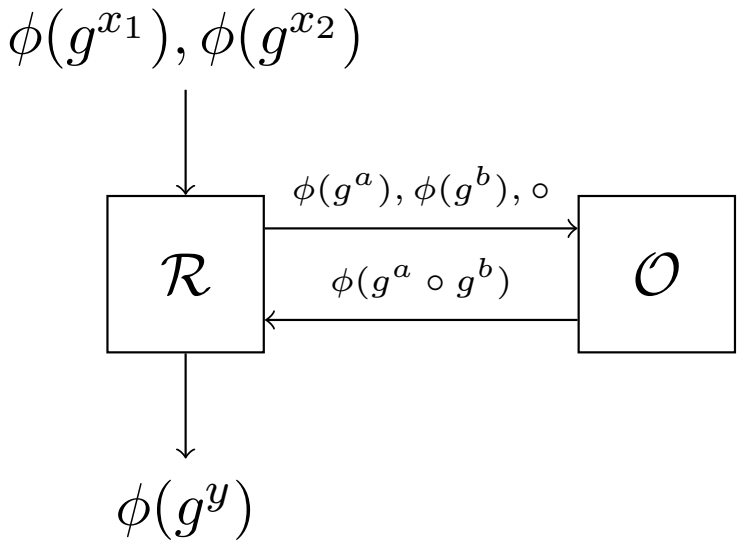
An algebraic reduction only computes group elements using group operations.

$\langle \mathbb{G}, g \rangle$

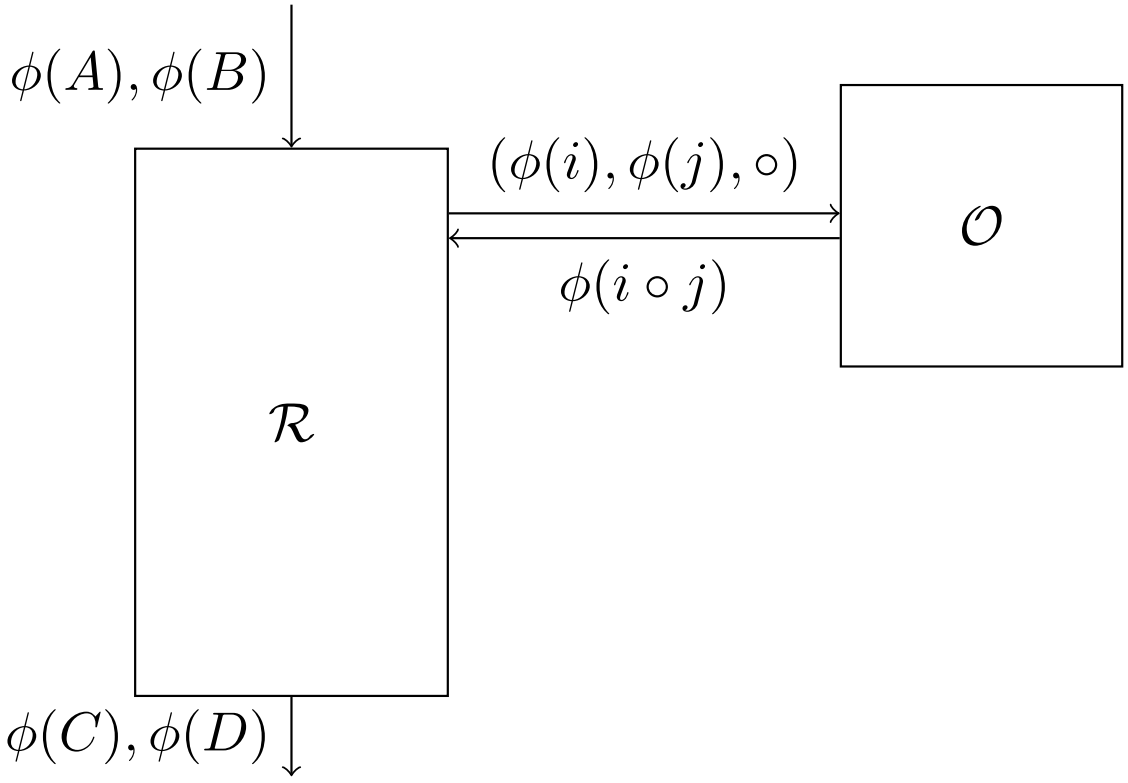


A generic reduction works regardless of the representation of the group.

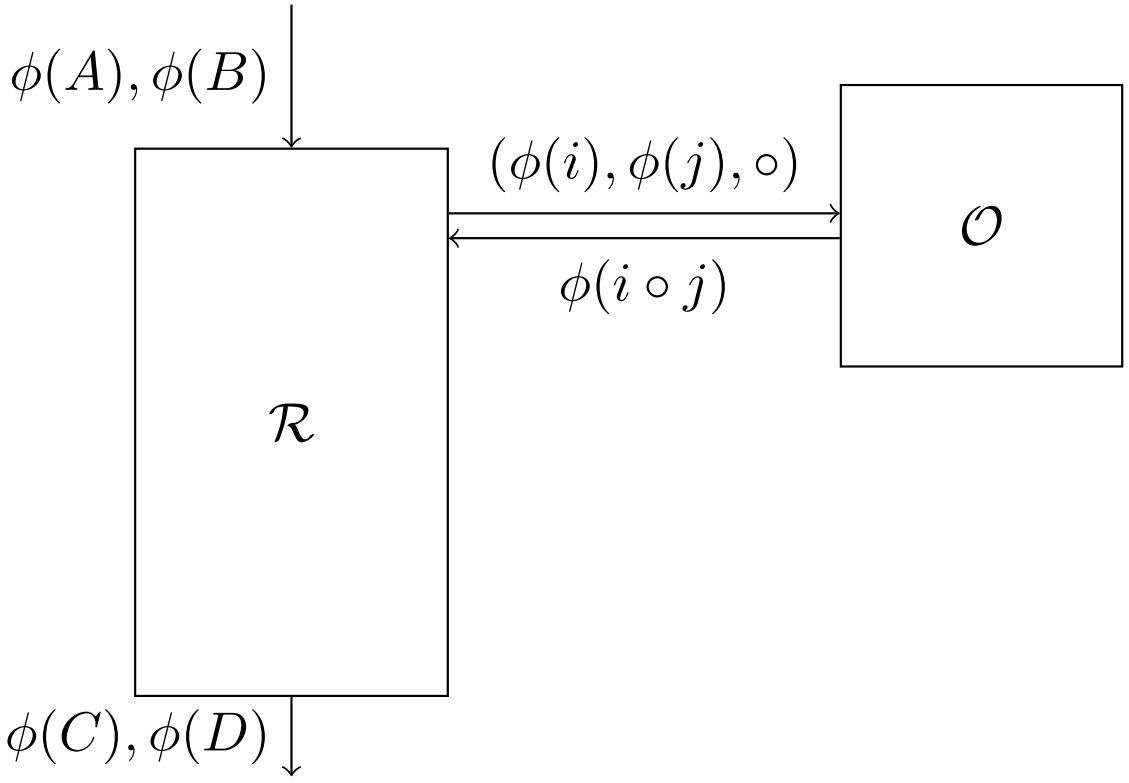
$\phi : \mathbb{G} \to \{0,1\}^{2n}$



So... GGM? No!

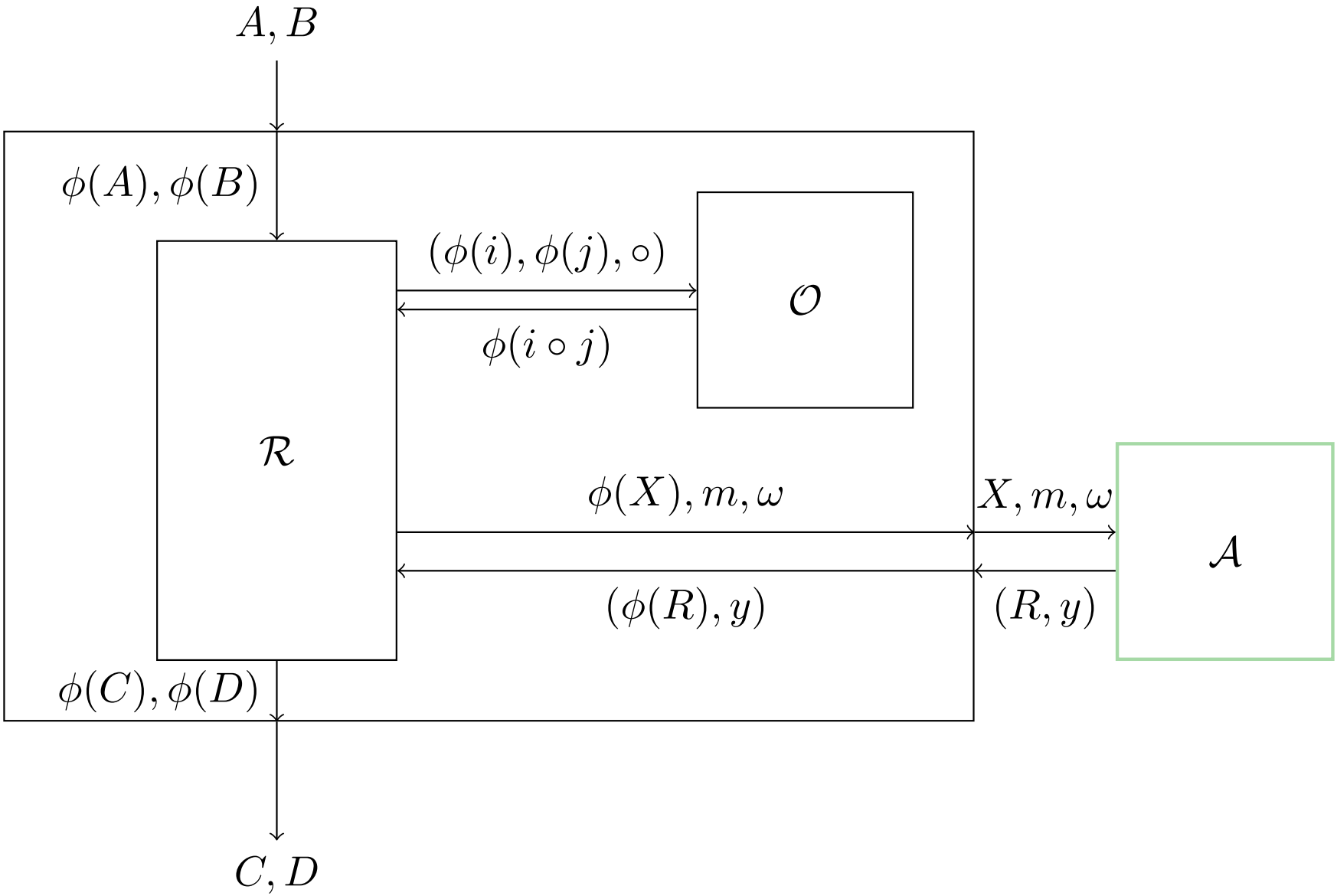

$\phi(A), \phi(B)$
$(\phi(i), \phi(j), \circ)$
$\mathcal{O}$
$\phi(i \circ j)$
$\mathcal{R}$
$\phi(C), \phi(D)$

So... GGM? No!

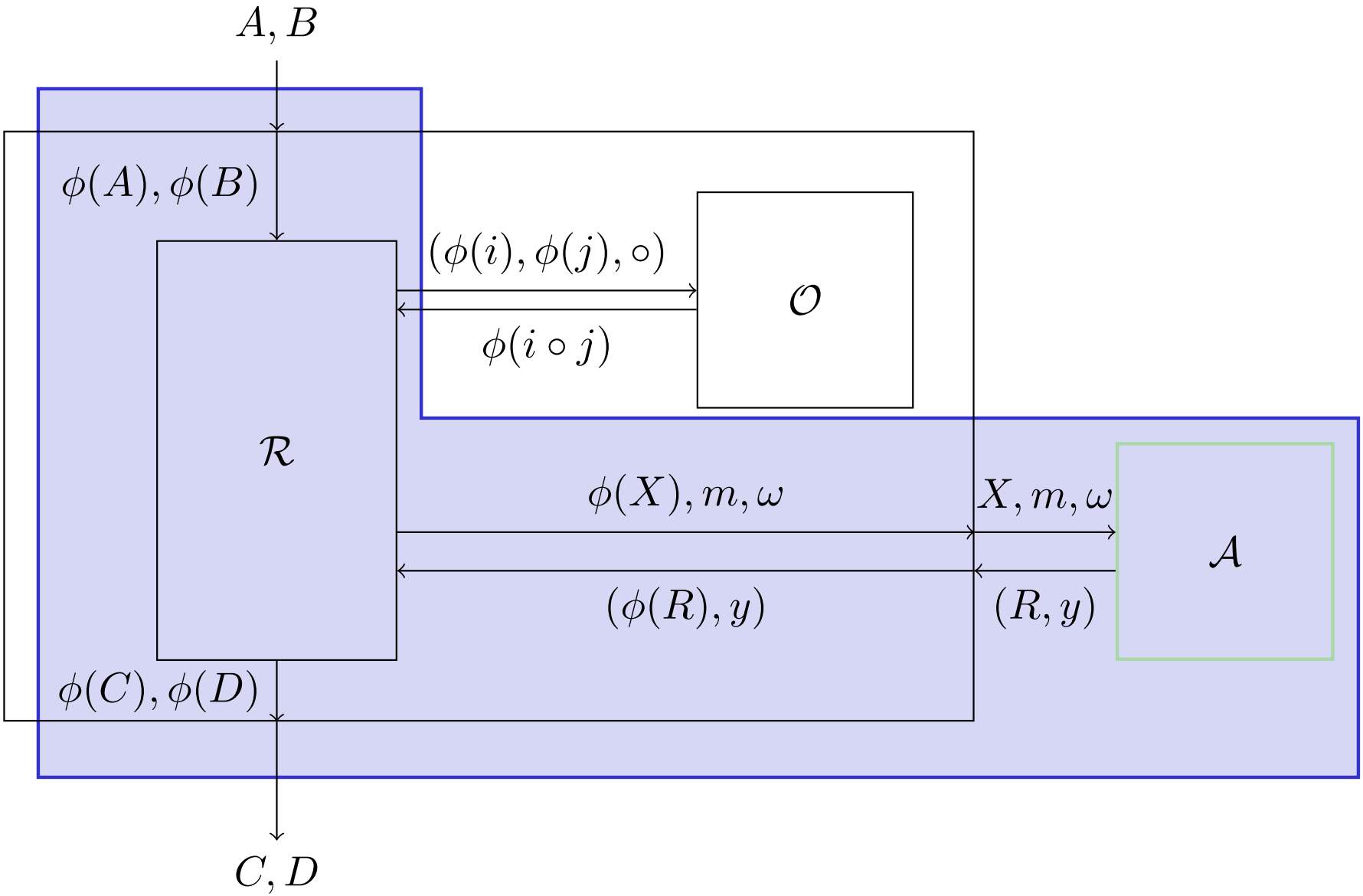


$\mathcal{A}$

So... GGM? No!


A, B
$\phi(A), \phi(B)$
$(\phi(i), \phi(j), \circ)$
$\phi(i \circ j)$
$\mathcal{O}$
$\mathcal{R}$
$\phi(X), m, \omega$
X, m, ω
$\mathcal{A}$
$(\phi(R), y)$
(R, y)
$\phi(C), \phi(D)$
C, D

So... GGM? No!



Ok, so how does it work?

Vanilla Reductions

$$\underline{\text{PROC } \mathcal{A}(X, m, \omega)}$$

$$(R_1, \ldots, R_q) \leftarrow \mathbb{G}^q$$

for all $i \in [q]$

$$\quad c_i = \mathcal{H}(R_i, m)$$

$$\alpha \leftarrow [q]$$

$$y := \log_g X^{c_\alpha} R_\alpha$$

return (R_α, y).

Ok, so how does it work?

Vanilla Reductions

Vanilla Reduction:

- Runs $\mathcal{A}$ once
- Does not rewind

Result:

- Rules out all generic vanilla reductions
- Even tight reductions

Ok, so how does it work?

Vanilla Reductions

$$\underline{\text{PROC } \mathcal{A}(X, m, \omega)}$$

$$(R_1, \ldots, R_q) \leftarrow \mathbb{G}^q$$

for all $i \in [q]$

$$\quad c_i = \mathcal{H}(R_i, m)$$

$$\alpha \leftarrow [q]$$

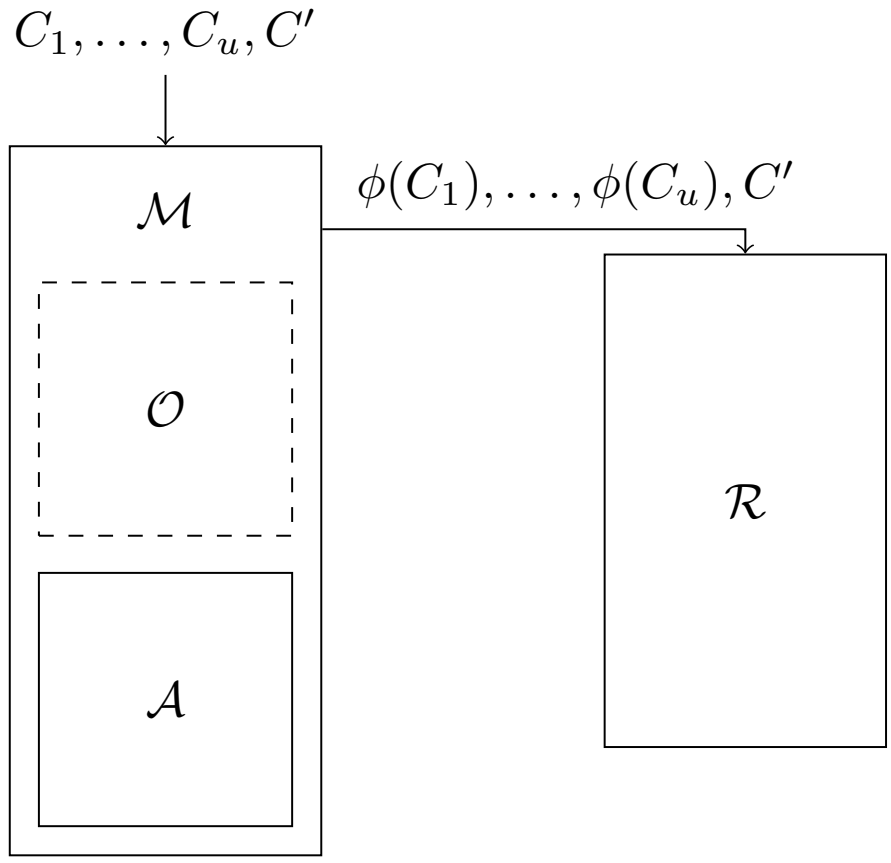
$$y := \log_g X^{c_\alpha} R_\alpha$$

return (R_α, y).

Ok, so how does it work?

Vanilla Reductions

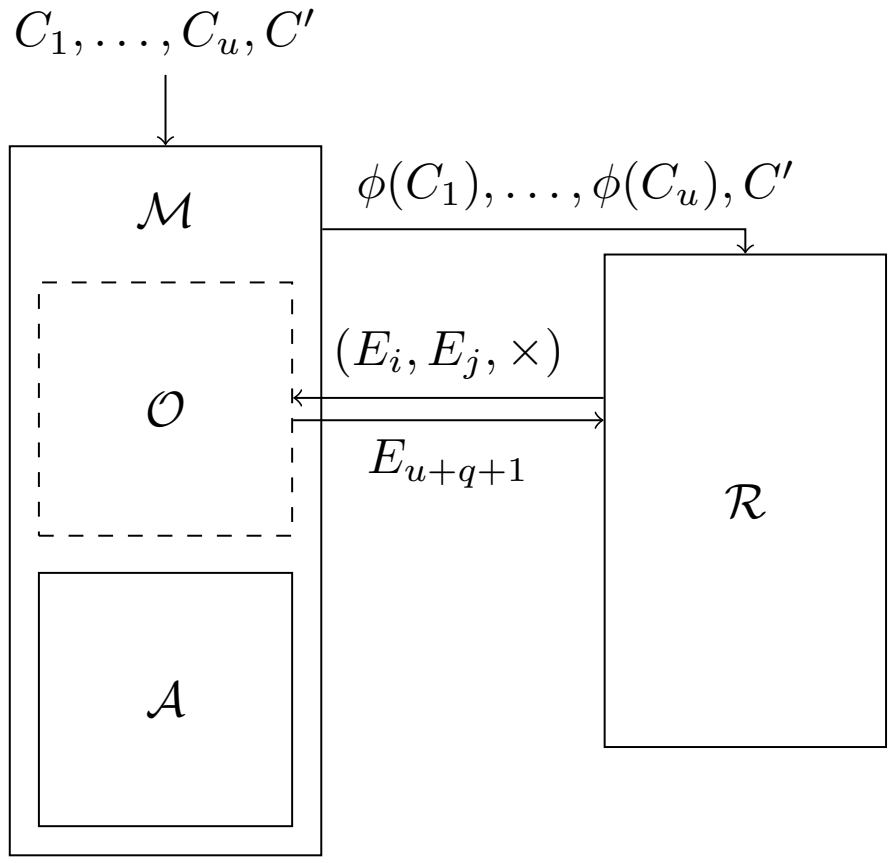
$\mathcal{L}^{\mathbb{G}}$	$\mathcal{L}^{E}$
C_1	E_1
$\vdots$	$\vdots$
C_u	E_u
R_1	E_{u+1}
$\vdots$	$\vdots$
R_q	E_{u+q}



Ok, so how does it work?

Vanilla Reductions

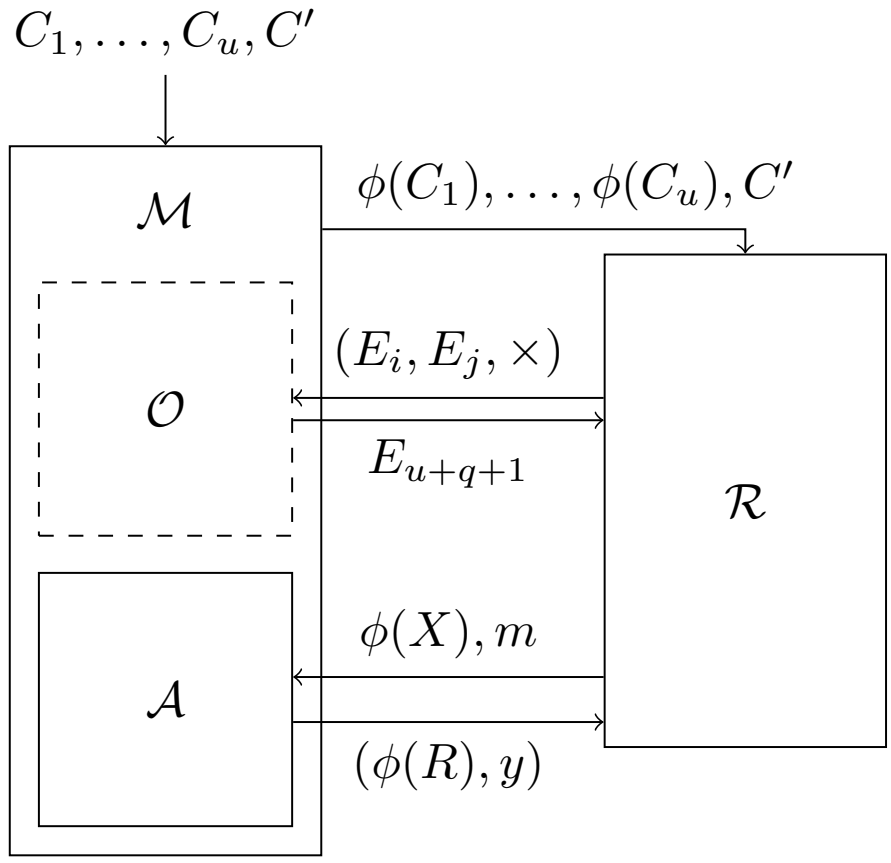
$\mathcal{L}^{\mathbb{G}}$	$\mathcal{L}^{E}$
C_1	E_1
$\vdots$	$\vdots$
C_u	E_u
R_1	E_{u+1}
$\vdots$	$\vdots$
R_q	E_{u+q}
A	E_{u+q+1}



Ok, so how does it work?

Vanilla Reductions

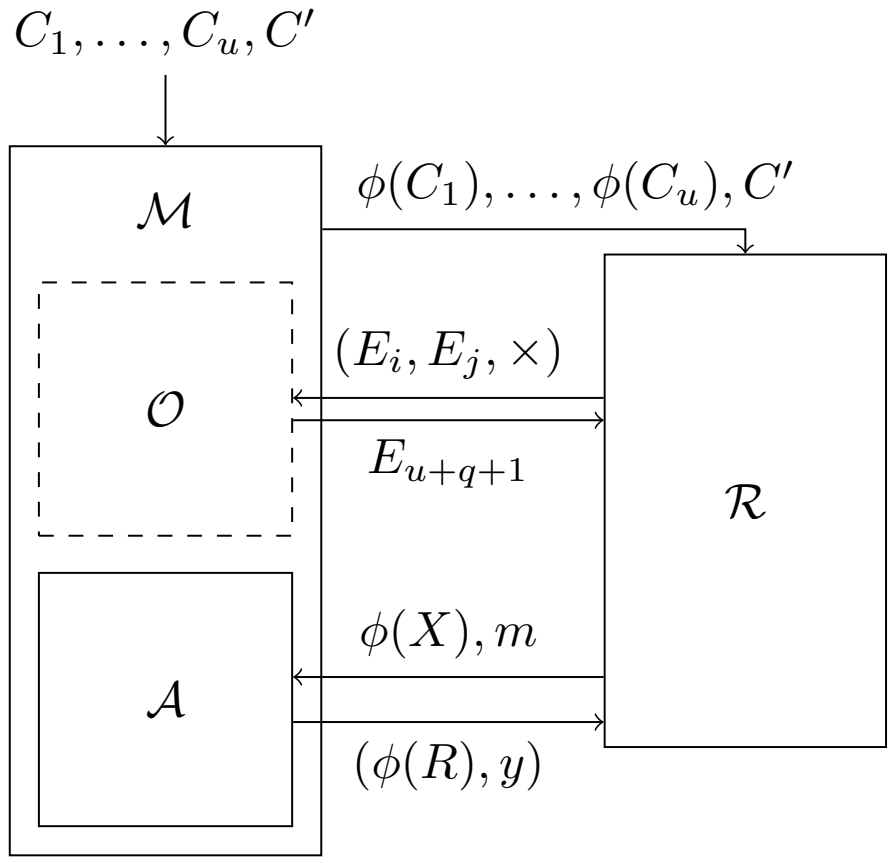
$\mathcal{L}^{\mathbb{G}}$	$\mathcal{L}^{E}$
C_1	E_1
$\vdots$	$\vdots$
C_u	E_u
R_1	E_{u+1}
$\vdots$	$\vdots$
R_q	E_{u+q}
A	E_{u+q+1}



Ok, so how does it work?

Vanilla Reductions

$\mathcal{L}^{\mathbb{G}}$	$\mathcal{L}^{E}$	$\mathcal{L}^{V}$
C_1	E_1	$(1, 0, ...)$
$\vdots$	$\vdots$	$\vdots$
C_u	E_u	$(..., 0, 1, 0, ...)$
R_1	E_{u+1}	$(..., 0, 1, 0, ...)$
$\vdots$	$\vdots$	$\vdots$
R_q	E_{u+q}	$(..., 0, 1)$
A	E_{u+q+1}	$V_i + V_j$



Ok, so how does it work?

Vanilla Reductions

$\mathcal{L}^{\mathbb{G}}$	$\mathcal{L}^{E}$	$\mathcal{L}^{V}$
$\vdots$	$\vdots$	$\vdots$
R_α	$E_{u+\alpha}$	$V_{u+\alpha}$
$\vdots$	$\vdots$	$\vdots$
G_{u+q+1}	E_{u+q+1}	V_{u+q+1}

$\underline{\text{PROC } \mathcal{A}(\phi(X), m, \omega) :}$

for all $i \in [q]$

$\quad c_i = \mathcal{R}.\mathcal{H}(\phi(R_i), m)$

$\alpha \leftarrow [q]$

$y := \log_g X^{c_\alpha} R_\alpha$

Ok, so how does it work?

Vanilla Reductions

$\mathcal{L}^{\mathbb{G}}$	$\mathcal{L}^{E}$	$\mathcal{L}^{V}$
$\vdots$	$\vdots$	$\vdots$
R_{α}	$E_{u+\alpha}$	$V_{u+\alpha}$
$\vdots$	$\vdots$	$\vdots$
G_{u+q+1}	E_{u+q+1}	V_{u+q+1}

$\underline{\text{PROC } \mathcal{A}(\phi(X), m, \omega) :}$

for all $i \in [q]$

$c_i = \mathcal{R}.\mathcal{H}(\phi(R_i), m)$

$\alpha \leftarrow [q]$

$y \leftarrow \mathbb{Z}_p \quad ; \quad R^*_{\alpha} := g^y X^{-c_\alpha}$

Ok, so how does it work?

Vanilla Reductions

$\mathcal{L}^{\mathbb{G}}$	$\mathcal{L}^{E}$	$\mathcal{L}^{V}$
$\vdots$	$\vdots$	$\vdots$
R^*_α	$E_{u+\alpha}$	$V_{u+\alpha}$
$\vdots$	$\vdots$	$\vdots$
G_{u+q+1}	E_{u+q+1}	V_{u+q+1}

$\underline{\text{PROC } \mathcal{A}(\phi(X), m, \omega) :}$

for all $i \in [q]$

$\quad c_i = \mathcal{R}.\mathcal{H}(\phi(R_i), m)$

$\alpha \leftarrow [q]$

$y \leftarrow \mathbb{Z}_p \quad ; \quad R^*_\alpha := g^y X^{-c_\alpha}$

Ok, so how does it work?

Vanilla Reductions

$\mathcal{L}^{\mathbb{G}}$	$\mathcal{L}^{E}$	$\mathcal{L}^{V}$
$\vdots$	$\vdots$	$\vdots$
R^*_α	$E_{u+\alpha}$	$V_{u+\alpha}$
$\vdots$	$\vdots$	$\vdots$
G_{u+q+1}	E_{u+q+1}	V_{u+q+1}

$\underline{\text{PROC } \mathcal{A}(\phi(X), m, \omega) :}$

for all $i \in [q]$

$c_i = \mathcal{R}.\mathcal{H}(\phi(R_i), m)$

$\alpha \leftarrow [q]$

$y \leftarrow \mathbb{Z}_p \quad ; \quad R^*_\alpha := g^y X^{-c_\alpha}$

for $j = 1, \ldots, |\mathcal{L}^{\mathbb{G}}|$ do

$$G_i := \prod_{j=1}^{u+q} V_i^j \cdot G_j$$

return $(y, \phi(R^*_\alpha))$

Will this not trip up the Reduction?

$\mathcal{R}$ is only able to notice the reprogramming if there exist i, j such that

$$G_i = G_j$$

before the reprogramming and

$$G_i \neq G_j$$

after reprogramming, or the other way round. This happens with probability at most

$$\frac{2(u + q + t_{\mathcal{R}})^2}{p} \leq \text{negl}$$

Summary & Conclusion

The security of Schnorr signatures cannot be reduced to any *representation invariant* assumption *tighter than* $O(1/q)$ using a *generic* fully blackbox reduction.

Thank You!

Nils Fleischhacker
fleischhacker@cs.uni-saarland.de

SAARLAND UNIVERSITY
GRADUATE SCHOOL OF COMPUTER SCIENCE
CISPA Center for Information Security, Privacy and Accountability