Limitations of the Meta-Reduction Technique: The Case of Schnorr Signatures

Marc Fischlin [1] **Nils Fleischhacker** [2]

[1]TU Darmstadt

[2]Saarland University, Center for IT-Security, Privacy, and Accountability

June 5, 2014

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

(Informal) Main Results[1]

- Schnorr Signatures are provably secure under the DLOG assumption in the weakly programmable ROM.
- Under the one-more DLOG assumption there does not exist a "single instance" reduction from the DLOG assumption in the non-programmable ROM.
- Eliminating the one-more DLOG assumption from our meta-reduction is highly unlikely.

[1] actual results may vary

Schnorr Signatures [Sch90,Schn91]

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

$\underline{\mathsf{Kgen}(1^\kappa)}$	$\underline{\mathsf{Sign}(\mathsf{sk}, m)}$	$\underline{\mathsf{Vrfy}(\mathsf{pk}, m, \sigma)}$
$\mathsf{sk} \xleftarrow{\$} \mathbb{Z}_q$	$r \xleftarrow{\$} \mathbb{Z}_q$	parse σ as (c, y)
$\mathsf{pk} := g^{\mathsf{sk}}$	$R := g^r$	if $c \stackrel{?}{=} \mathcal{H}(\mathsf{pk}^{-c} g^y, m)$
return $(\mathsf{sk}, \mathsf{pk})$	$c := \mathcal{H}(R, m)$	output 1
	$y := r + \mathsf{sk} \cdot c$	else
	return $\sigma = (c, y)$	output 0

Schnorr Signatures [Sch90,Schn91]

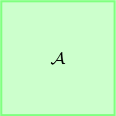
$$\mathbb{G} = \langle g \rangle, H$$

$\underline{\mathsf{Kgen}(1^\kappa)}$	$\underline{\mathsf{Sign}(\mathsf{sk}, m)}$	$\underline{\mathsf{Vrfy}(\mathsf{pk}, m, \sigma)}$
$\mathsf{sk} \xleftarrow{\$} \mathbb{Z}_q$	$r \xleftarrow{\$} \mathbb{Z}_q$	parse σ as (c, y)
$\mathsf{pk} := g^{\mathsf{sk}}$	$R := g^r$	if $c \stackrel{?}{=} \mathcal{H}(\mathsf{pk}^{-c} g^y, m)$
return $(\mathsf{sk}, \mathsf{pk})$	$c := \mathcal{H}(R, m)$	output 1
	$y := r + \mathsf{sk} \cdot c$	else
	return $\sigma = (c, y)$	output 0

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

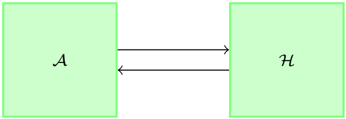
SAARLAND UNIVERSITY
GRADUATE SCHOOL OF COMPUTER SCIENCE
CISPA

Random Oracle Model
with(out) programmability [FLR+10]



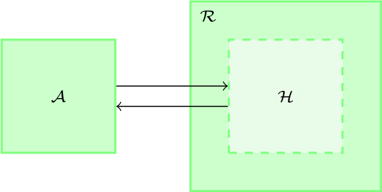
Random Oracle Model

with(out) programmability [FLR+10]



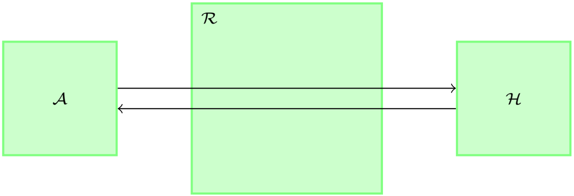
Random Oracle Model

with(out) programmability [FLR+10]



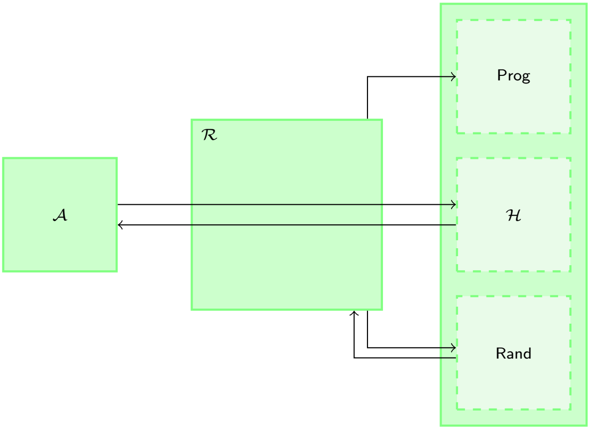
Random Oracle Model

with(out) programmability [FLR+10]



Random Oracle Model

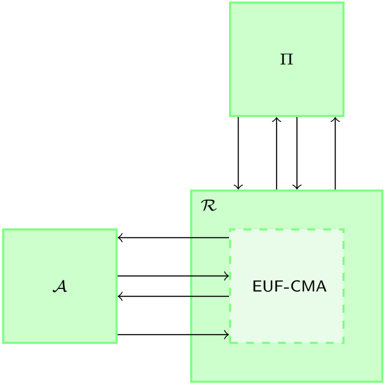
with(out) programmability [FLR+10]



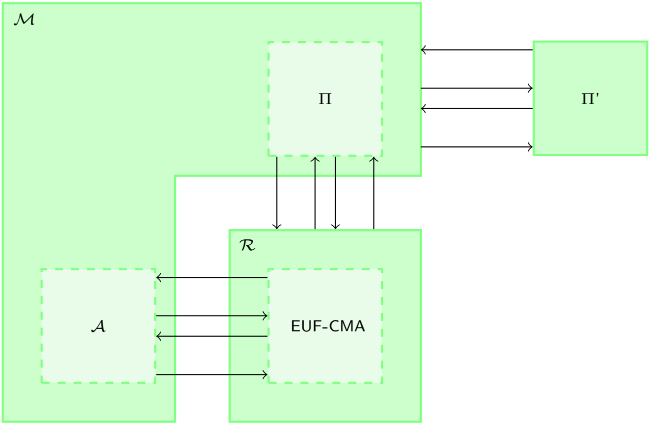
$$\mathsf{Prog}(a, b) \Rightarrow \mathcal{H}(a) \stackrel{\text{def}}{=} \mathsf{Rand}(b)$$

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

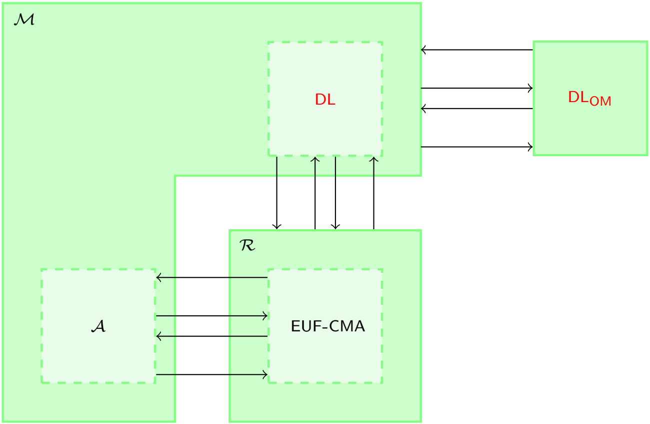
Meta-Reductions [BV98]



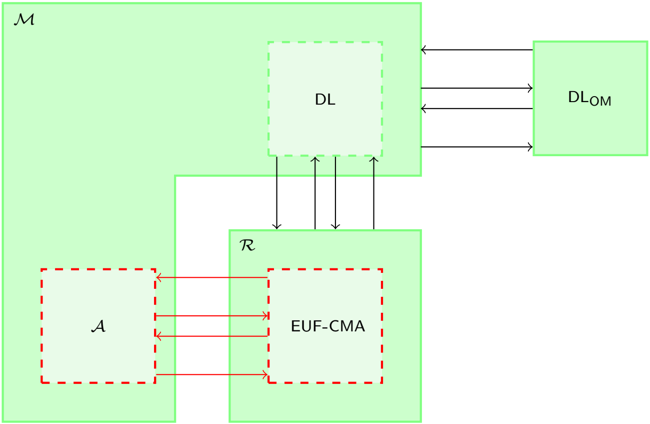
Meta-Reductions [BV98]



Meta-Reductions [BV98]



Meta-Reductions [BV98]



EUF-CMA

pk

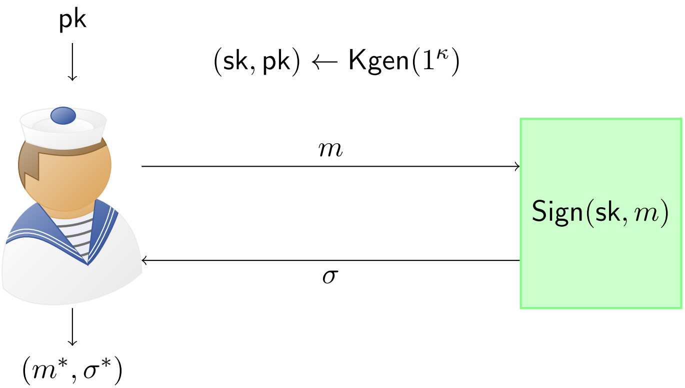
$\downarrow$

$(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Kgen}(1^\kappa)$

$\downarrow$

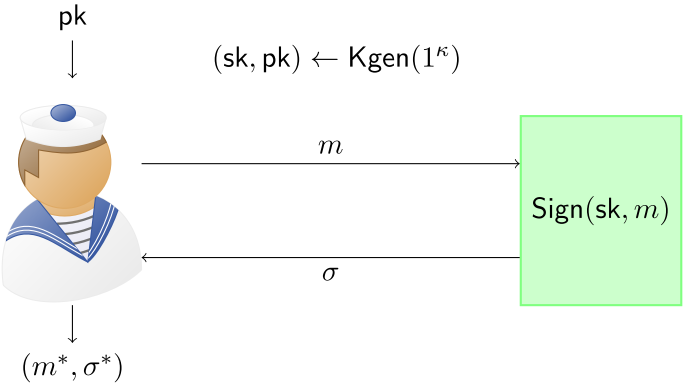
(m^*, σ^*)

EUF-CMA



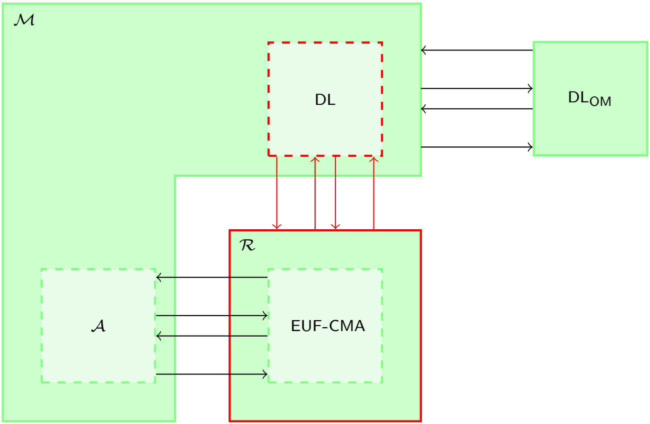
pk
$(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Kgen}(1^\kappa)$
m
$\mathsf{Sign}(\mathsf{sk}, m)$
σ
(m^*, σ^*)

EUF-CMA

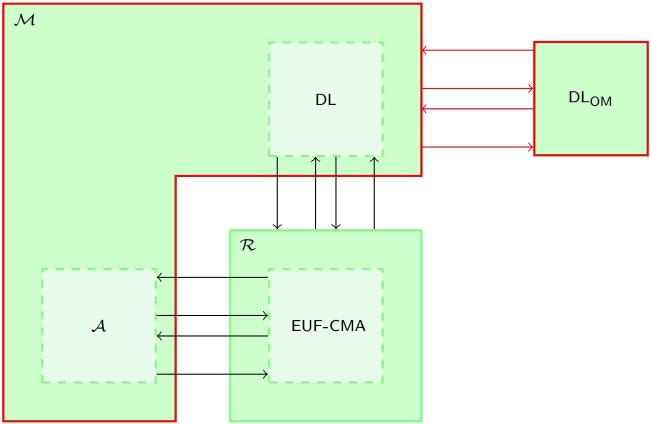


The attacker wins if $\mathsf{Vrfy}(\mathsf{pk}, m^*, \sigma^*) = 1$ and $m \neq m^*$

Meta-Reductions



Meta-Reductions



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

The One-More discrete log problem [BNPS03]

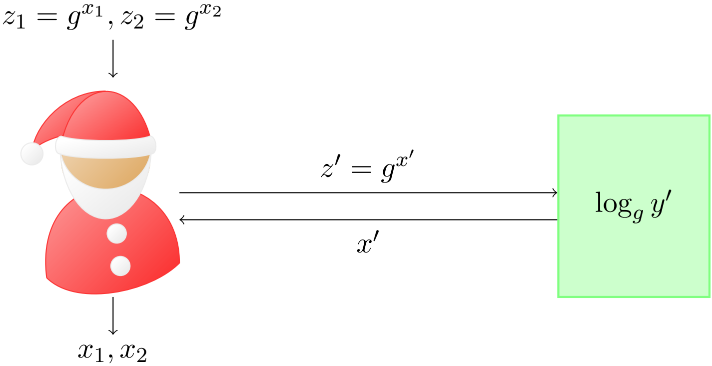
$$z_1 = g^{x_1}, z_2 = g^{x_2}$$

$$\downarrow$$

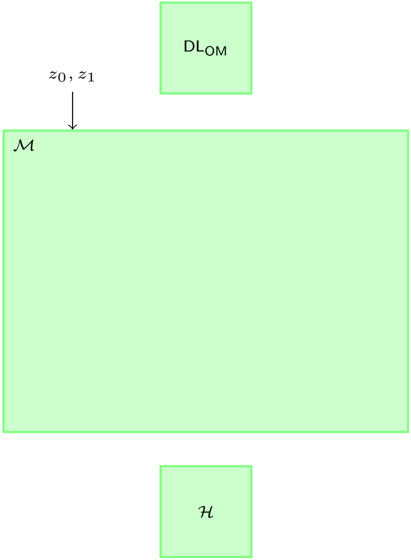
$$\downarrow$$

$$x_1, x_2$$

The One-More discrete log problem [BNPS03]



In the non-programmable ROM



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

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

$R := g^r$

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

$y := r + \mathsf{sk} \cdot c$

return $\sigma = (c, y)$

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

parse σ as (c, y)

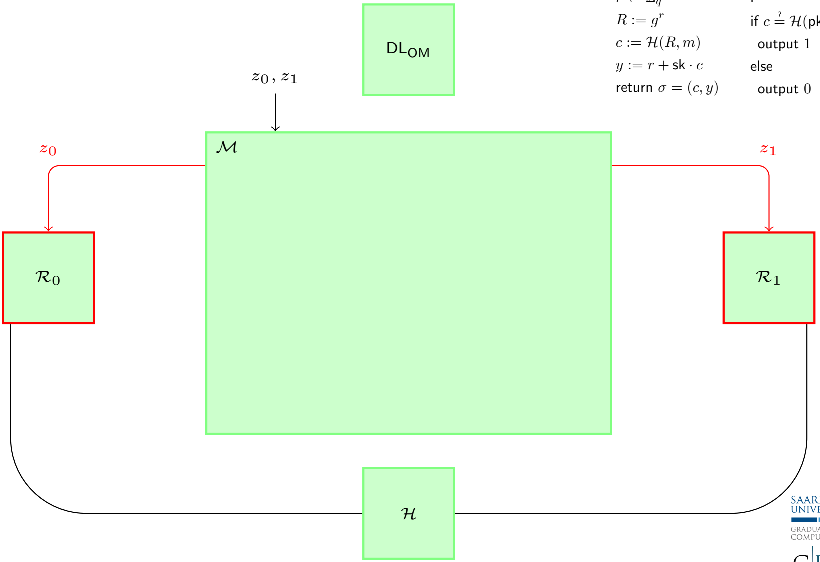
if $c \stackrel{?}{=} \mathcal{H}(\mathsf{pk}^{-c} g^y, m)$

output 1

else

output 0

In the non-programmable ROM



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

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

$R := g^r$

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

$y := r + \mathsf{sk} \cdot c$

return $\sigma = (c, y)$

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

parse σ as (c, y)

if $c \stackrel{?}{=} \mathcal{H}(\mathsf{pk}^{-c} g^y, m)$

output 1

else

output 0

SAARLAND UNIVERSITY

GRADUATE SCHOOL OF COMPUTER SCIENCE

CISPA

Center for IT-Security, Privacy and Accountability

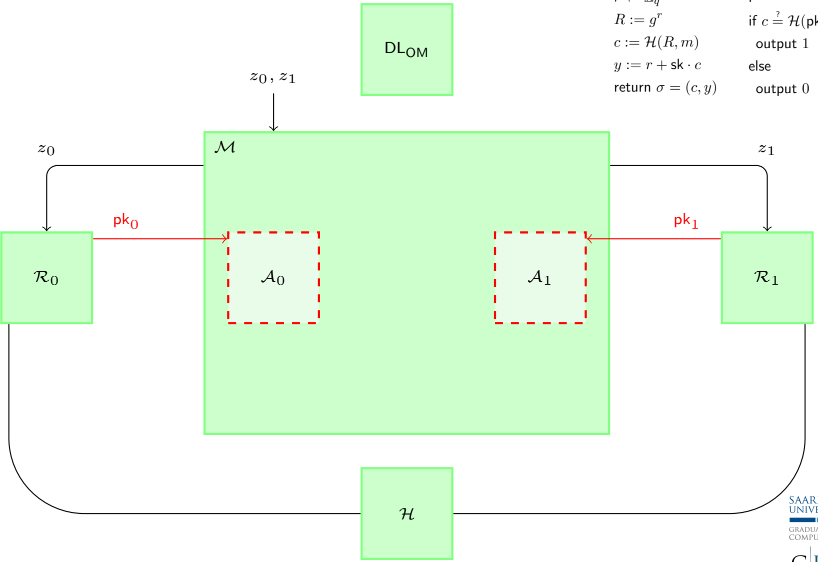
In the non-programmable ROM

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

$r \xleftarrow{\$} \mathbb{Z}_q$
$R := g^r$
$c := \mathcal{H}(R, m)$
$y := r + \mathsf{sk} \cdot c$
return $\sigma = (c, y)$

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

parse σ as (c, y)
if $c \stackrel{?}{=} \mathcal{H}(\mathsf{pk}^{-c} g^y, m)$
  output 1
else
  output 0



In the non-programmable ROM

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

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

$R := g^r$

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

$y := r + \mathsf{sk} \cdot c$

return $\sigma = (c, y)$

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

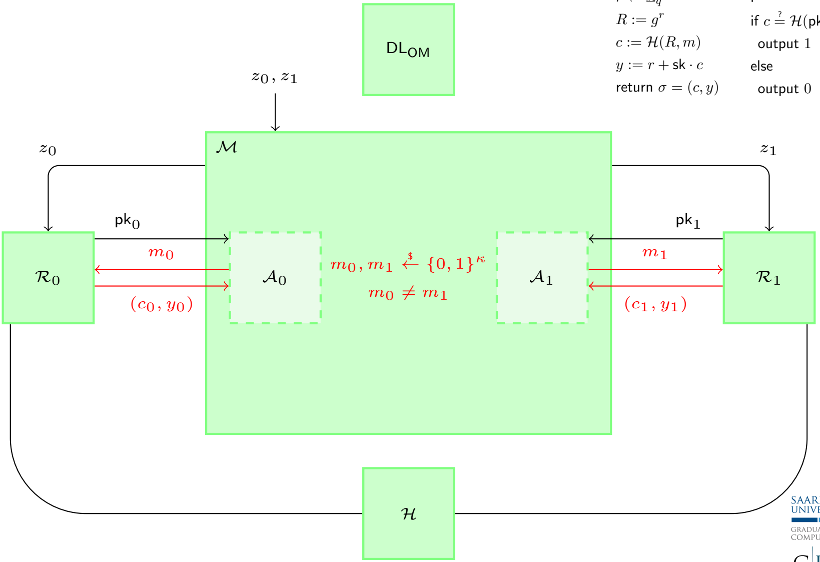
parse σ as (c, y)

if $c \stackrel{?}{=} \mathcal{H}(\mathsf{pk}^{-c} g^y, m)$

output 1

else

output 0



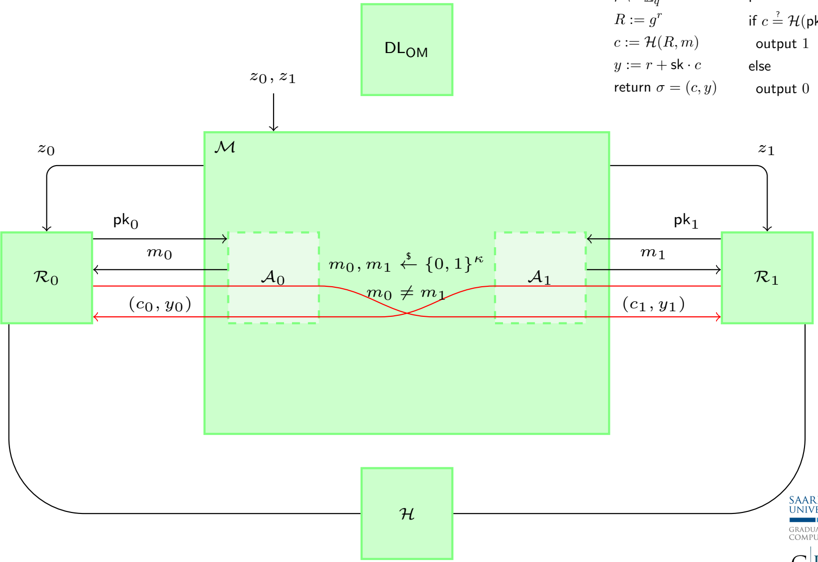
In the non-programmable ROM

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

$r \xleftarrow{\$} \mathbb{Z}_q$
$R := g^r$
$c := \mathcal{H}(R, m)$
$y := r + \mathsf{sk} \cdot c$
return $\sigma = (c, y)$

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

parse σ as (c, y)
if $c \stackrel{?}{=} \mathcal{H}(\mathsf{pk}^{-c} g^y, m)$
  output 1
else
  output 0



In the non-programmable ROM

Sign(sk, m)

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

$R := g^r$

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

$y := r + \mathsf{sk} \cdot c$

return $\sigma = (c, y)$

Vrfy(pk, m, σ)

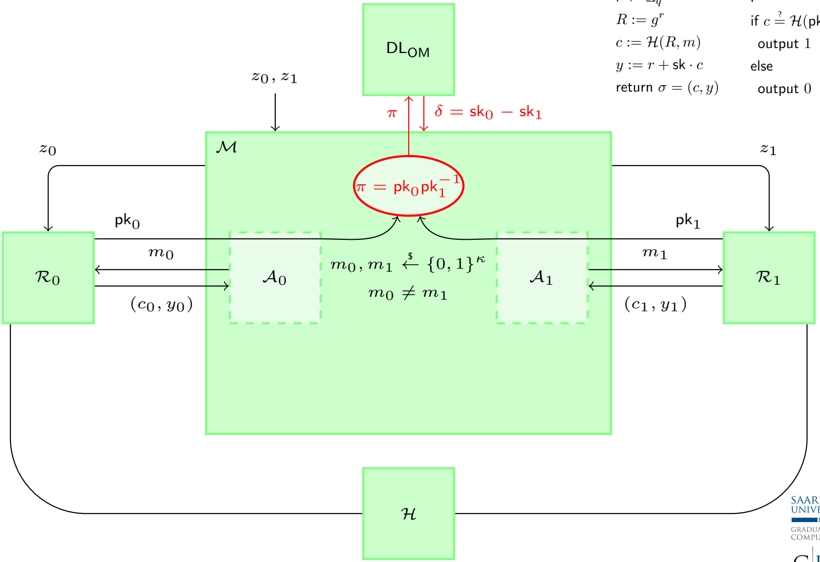
parse σ as (c, y)

if $c \stackrel{?}{=} \mathcal{H}(\mathsf{pk}^{-c} g^y, m)$

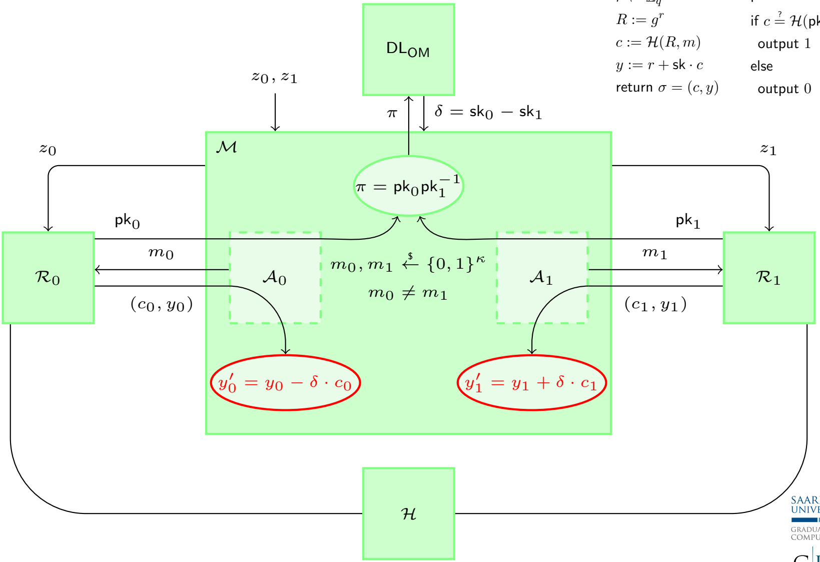
output 1

else

output 0



In the non-programmable ROM



<u>$\mathsf{Sign}(\mathsf{sk}, m)$</u>

$r \xleftarrow{\$} \mathbb{Z}_q$
$R := g^r$
$c := \mathcal{H}(R, m)$
$y := r + \mathsf{sk} \cdot c$
return $\sigma = (c, y)$

<u>$\mathsf{Vrfy}(\mathsf{pk}, m, \sigma)$</u>

parse σ as (c, y)
if $c \stackrel{?}{=} \mathcal{H}(\mathsf{pk}^{-c} g^y, m)$
  output 1
else
  output 0

In the non-programmable ROM

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

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

$R := g^r$

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

$y := r + \mathsf{sk} \cdot c$

return $\sigma = (c, y)$

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

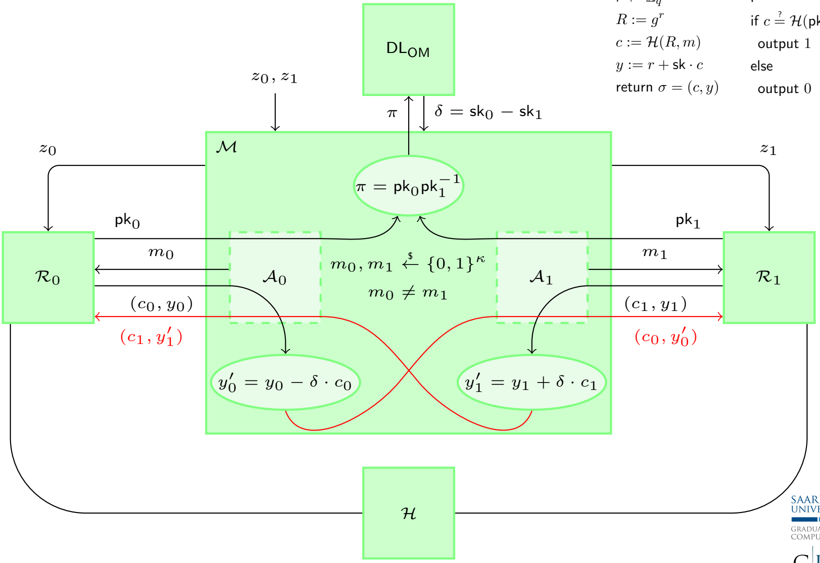
parse σ as (c, y)

if $c \stackrel{?}{=} \mathcal{H}(\mathsf{pk}^{-c} g^y, m)$

output 1

else

output 0



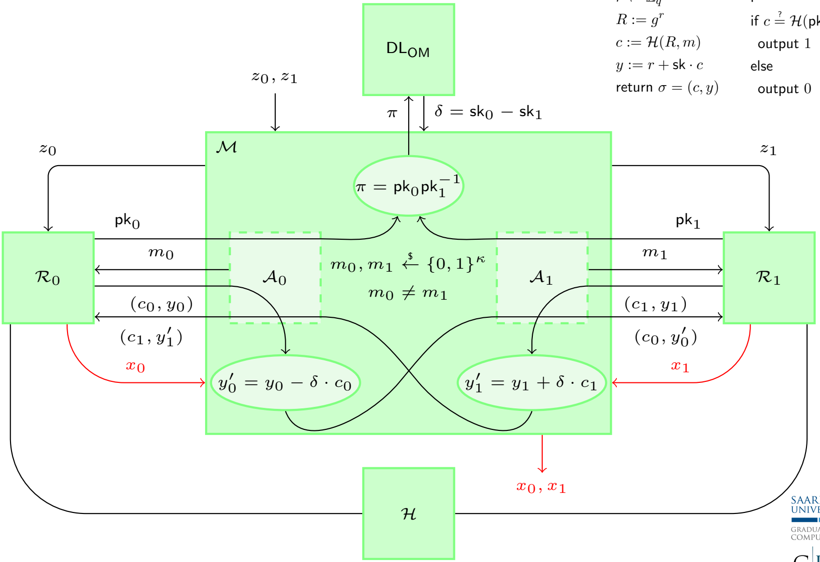
In the non-programmable ROM

<u>$\mathsf{Sign}(\mathsf{sk}, m)$</u>

- $r \xleftarrow{\$} \mathbb{Z}_q$
- $R := g^r$
- $c := \mathcal{H}(R, m)$
- $y := r + \mathsf{sk} \cdot c$
- return $\sigma = (c, y)$

<u>$\mathsf{Vrfy}(\mathsf{pk}, m, \sigma)$</u>

- parse σ as (c, y)
- if $c \stackrel{?}{=} \mathcal{H}(\mathsf{pk}^{-c} g^y, m)$
  - output 1
- else
  - output 0



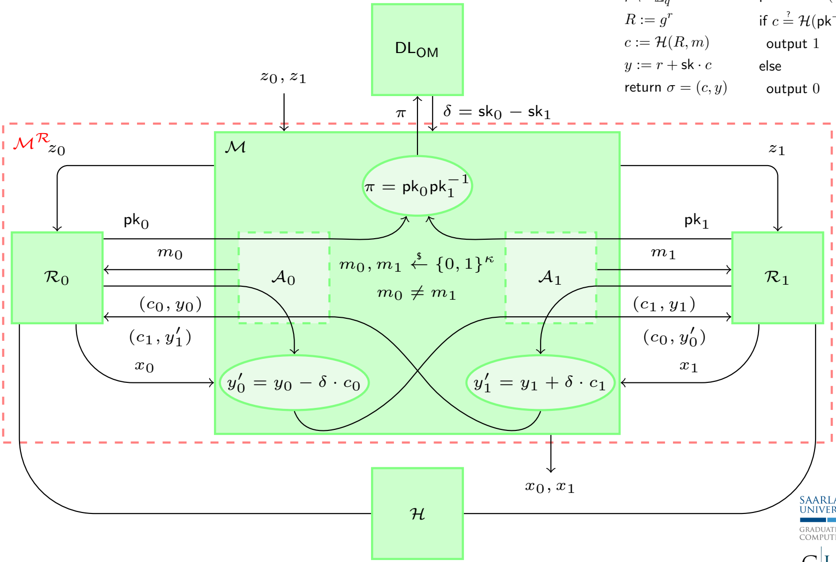
In the non-programmable ROM

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

- $r \xleftarrow{\$} \mathbb{Z}_q$
- $R := g^r$
- $c := \mathcal{H}(R, m)$
- $y := r + \mathsf{sk} \cdot c$
- return $\sigma = (c, y)$

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

- parse σ as (c, y)
- if $c \stackrel{?}{=} \mathcal{H}(\mathsf{pk}^{-c} g^y, m)$
  - output 1
- else
  - output 0



SAARLAND UNIVERSITY GRADUATE SCHOOL OF COMPUTER SCIENCE

CISPA Center for IT-Security, Privacy and Accountability

Can we do better?

- Probably not.
- Going one (meta-)level deeper and using a meta-meta-reduction, we can show that removing the one-more discrete log assumption would (constructively) imply an adversary against the signature scheme.

So, what does this mean?

Under the One-More Discrete Log Assumption, no single instance reductions from the discrete log Problem can exist for Schnorr signatures, if they do not program the random oracle. A relaxed notion of programmability, however, is sufficient.

The result is optimal in the sense that removing the assumption proves to be extremely unlikely.

Open Problems

- We rule out DLOG reductions, but what about CDH,...
- Possibly even interactive assumptions?

Thank You!

Nils Fleischhacker
fleischhacker@cs.uni-saarland.de

Full version available on eprint
`http://eprint.iacr.org/2013/140`

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