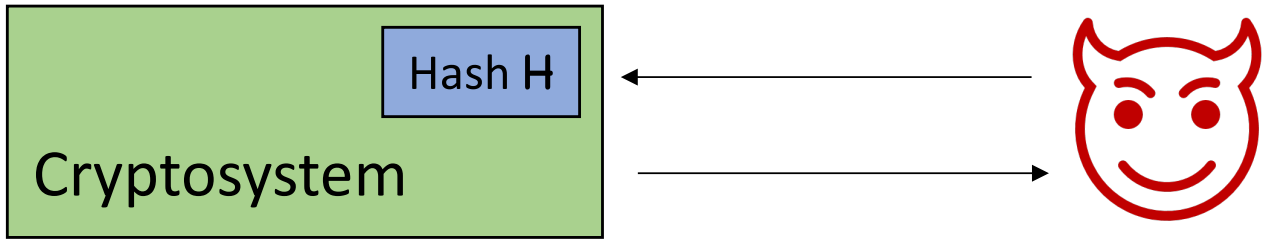
Augmented Random Oracles

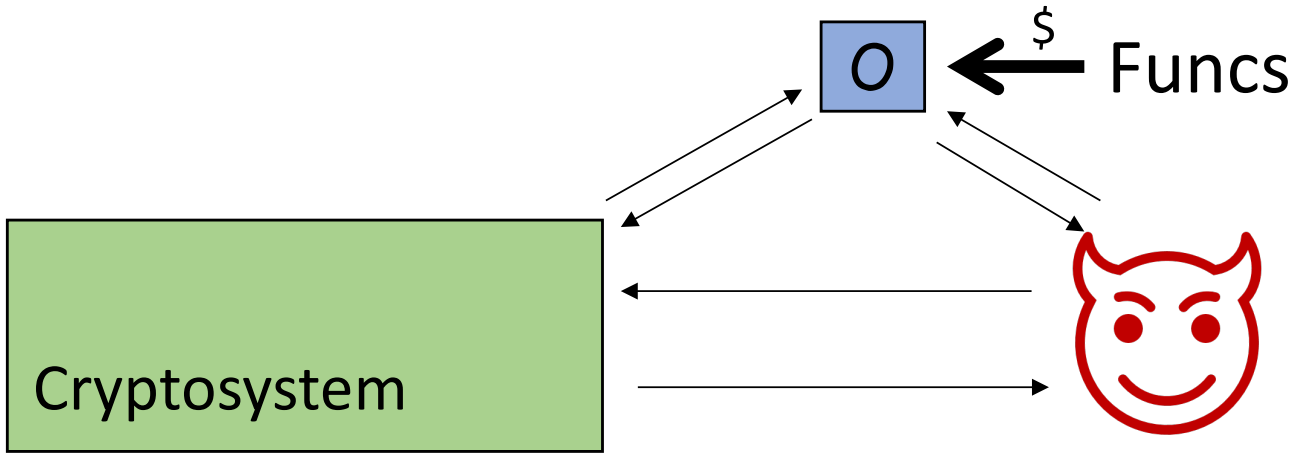
Mark Zhandry (NTT Research & Princeton University)



Sometimes can’t prove security. Then what?

Random Oracle Model (ROM)

[Bellare-Rogaway'93]



Idea: Prove security in ROM, then hope security translates to concrete hash e.g. SHA2

[Canetti-Goldreich-Halevi'98]
ROM uninstantiability: $\exists$ scheme S st:
(1) S^{O} secure in ROM, but
(2) $\forall$ concrete H, S^{H} insecure

Since CGH'98, numerous other uninstantiabilities: [Dent'02, Goldwasser-Kalai'03, Bellare-Boldyreva-Palacio'04, Maurer-Renner-Holenstein'04, Black'06, Brzuska-Farshim-Mittleback'15]

Despite these works, the ROM remains widely used

Our goal: Design a model that avoids uninstantiability results, while still allowing proofs beyond the standard model

Case study:
Encrypt-with-Hash (EwH)

EwH [Bellare-Boldyreva-O'Neill'07]:

PKE ➡ $c = Enc(m ; H(pk||m))$

Thm [BBO'07]: If PKE is IND-CPA → EwH is secure deterministic encryption in random oracle model

Thm [Brzuska-Farshim-Mittelbach'14]: Under suitable assumptions, $\exists$ IND-CPA PKE s.t. EwH is insecure for **any** hash function

Proof sketch: Assume IND-CPA PKE'. Construct new PKE

$c = Enc'(m ; r),\ P_{m,r}$

```
P_{m,r}( <H> ) {
    if H(m)==r: return m;
    else: return ⊥;
}
```

Insecurity of EwH: just feed code of hash function into $P_{m,r}$ ✓

Security of PKE: $P_{m,r}$ reveals m! ✗

Proof sketch: Assume IND-CPA PKE'. Construct new PKE

$c = Enc'(m ; r),\ Obf(P_{m,r})$

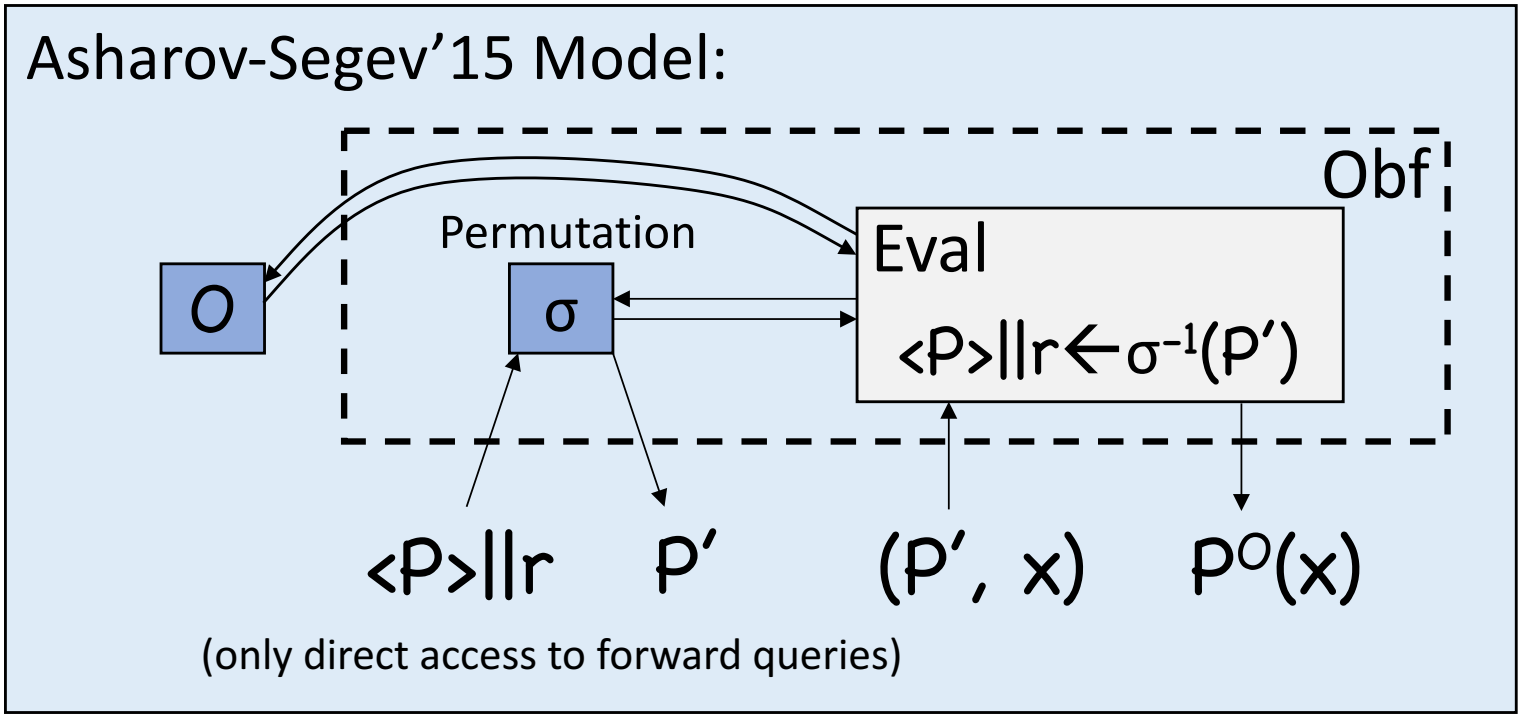
```
P_{m,r}( <H> ) {
    if H(m)==r: return m;
    else: return ⊥;
}
```

Insecurity of EwH: just feed code into obfuscated $P_{m,r}$ ✓

Security of PKE, intuition: given just black-box access to $P_{m,r}$, no way to find accepting input ✓

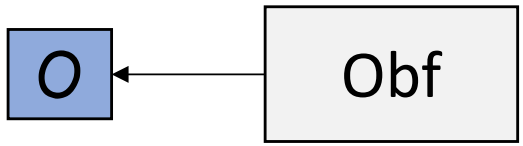
Key Takeaway: ROM uninstantiabilities use that concrete hash functions have code, but random oracles do not. However, they don't care about what the actual code does

Our goal: Design model where O does have code, namely instruction to make query

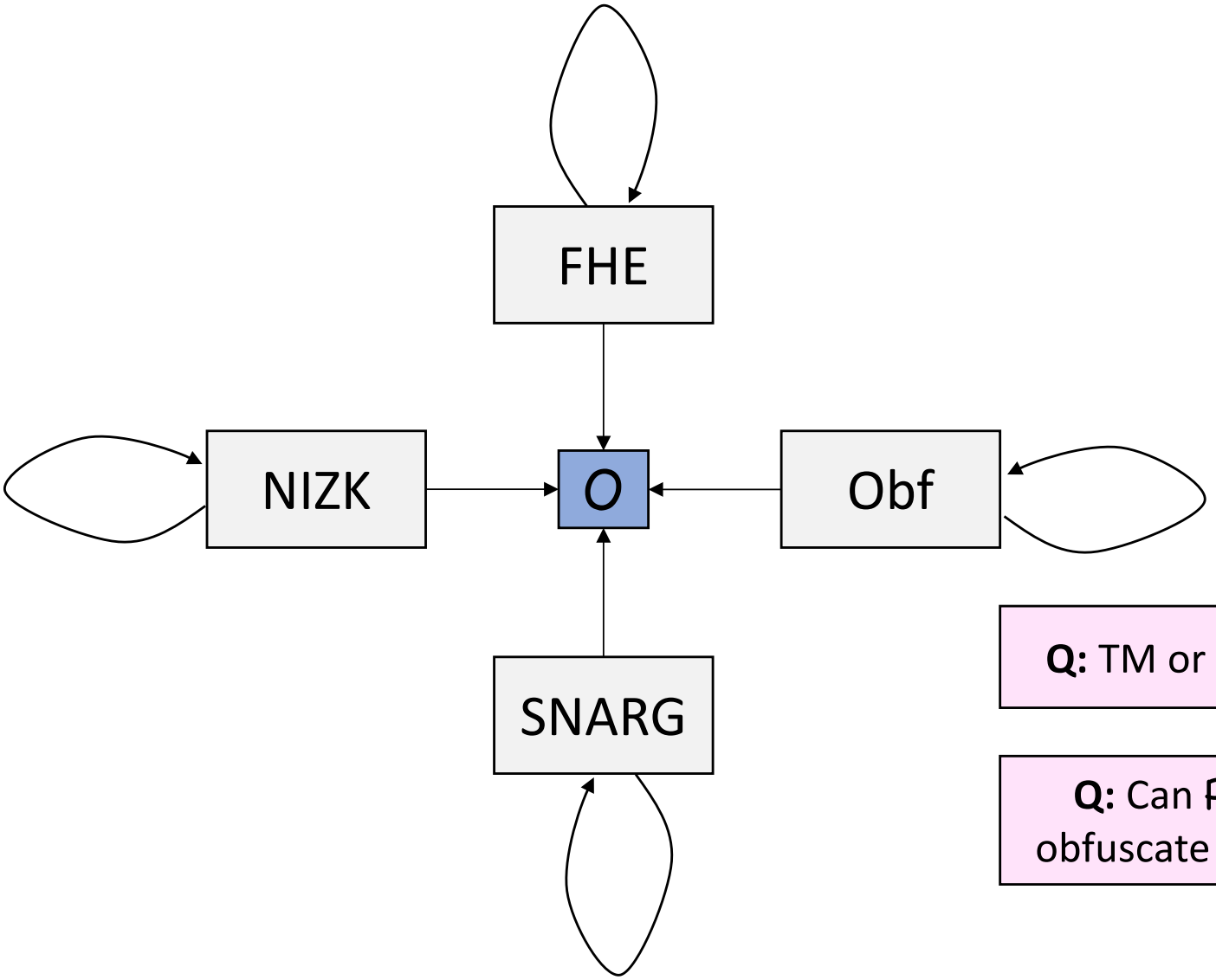


Thm [AS'15] (informal): Limits on the power of obfuscation

Initial idea: prove security in AS'15 model instead of ROM

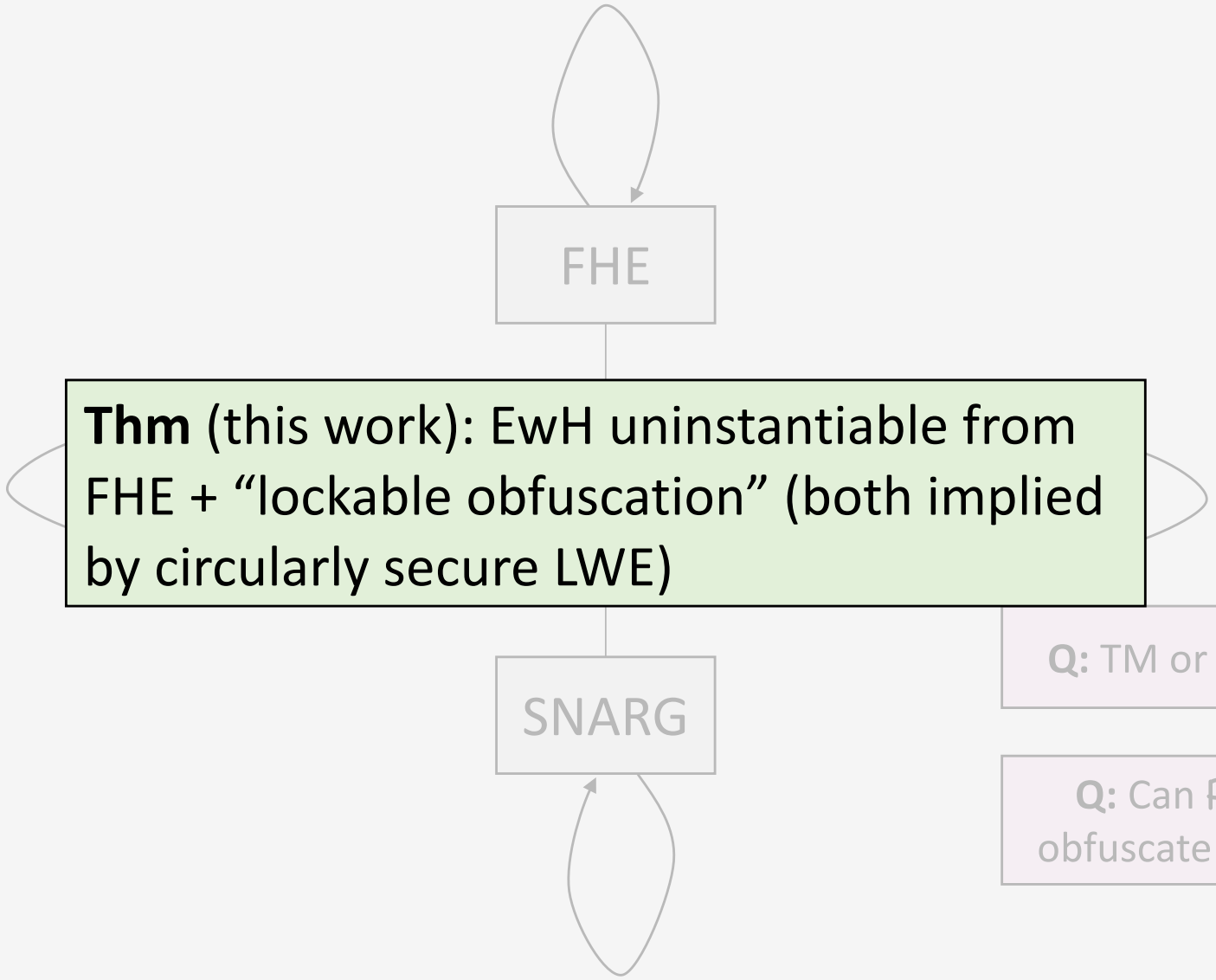


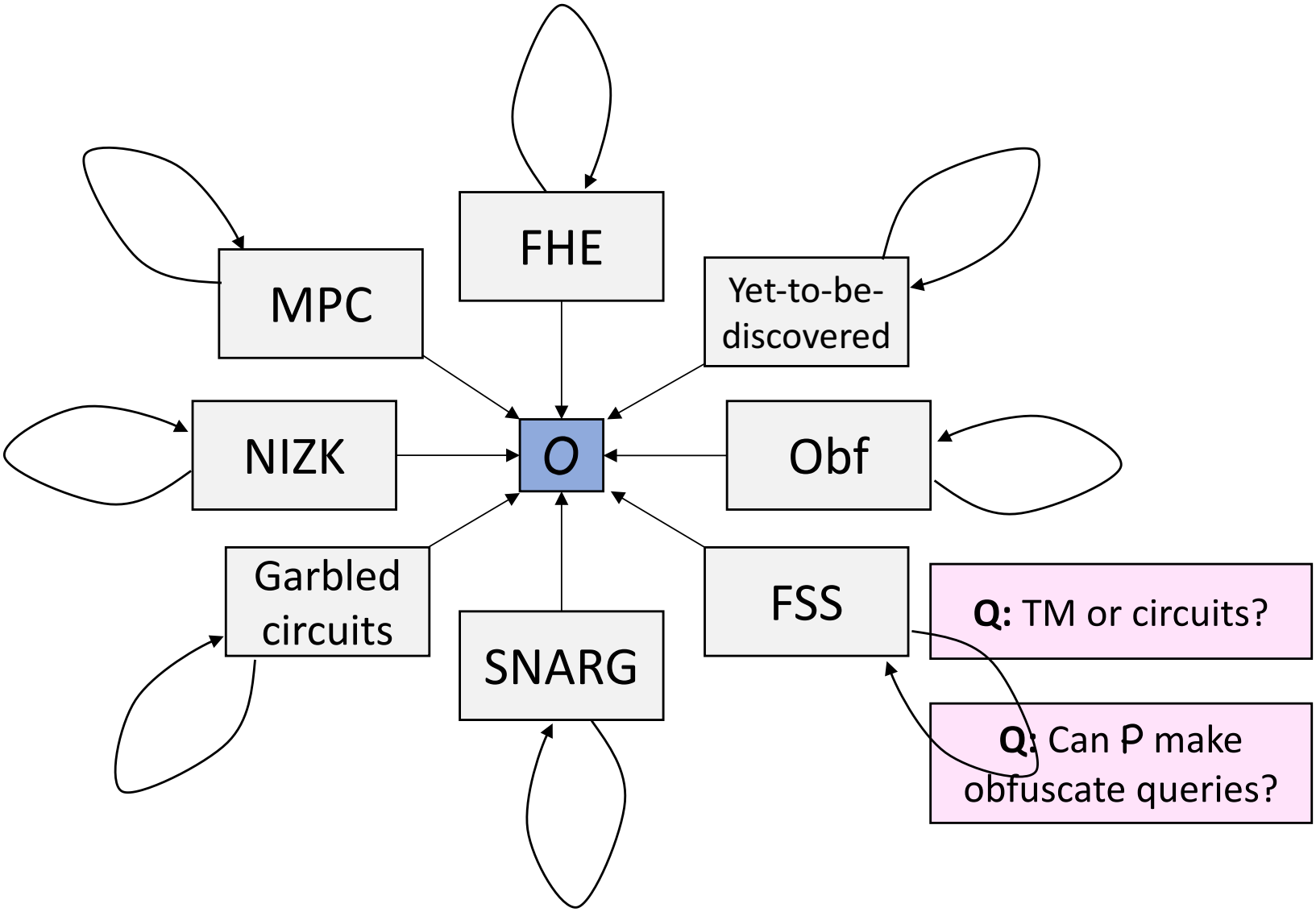
Security in AS'15 model → resilience to BFM'14 techniques

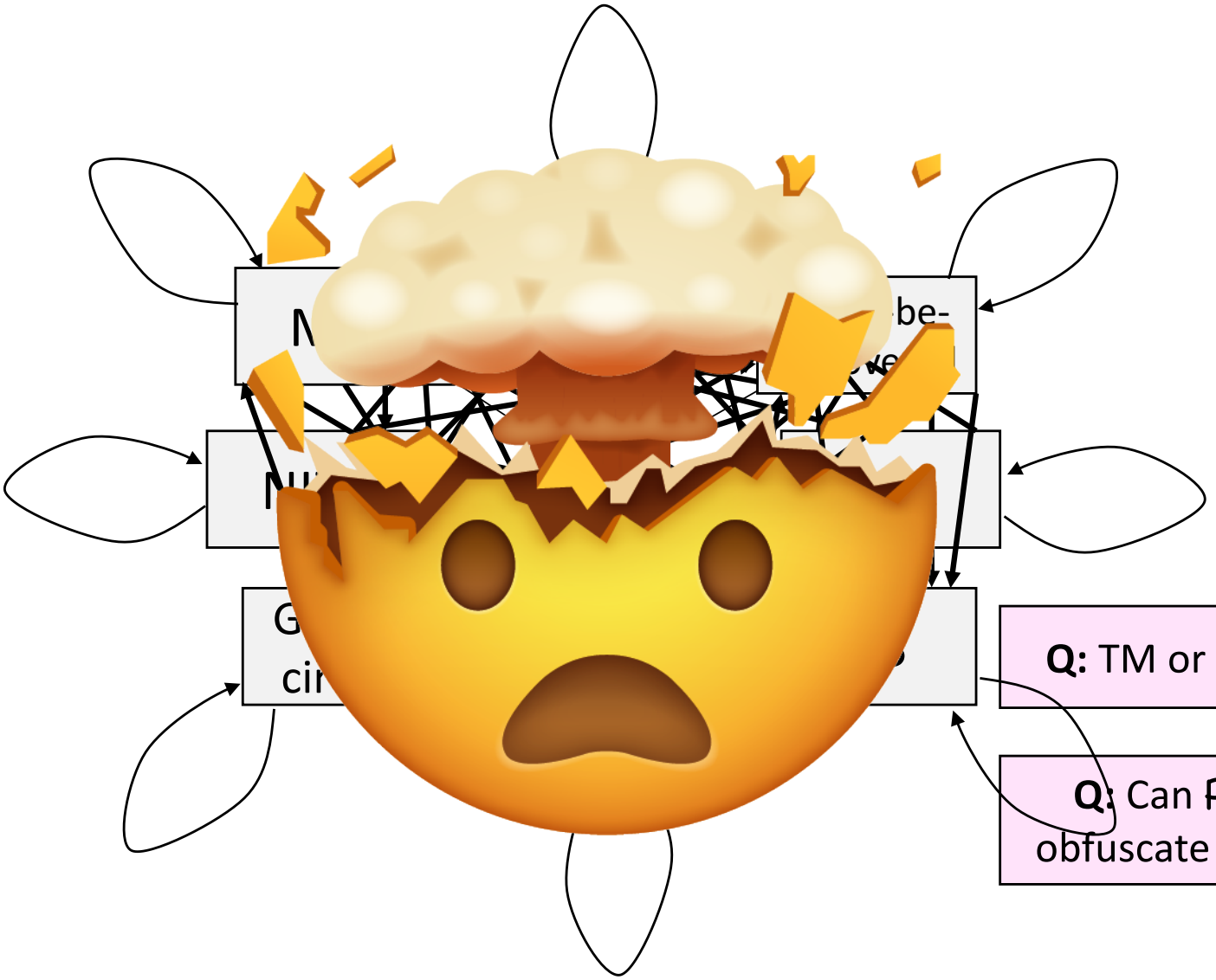


Q: TM or circuits?

Q: Can $\mathcal{P}$ make obfuscate queries?


FHE
Thm (this work): EwH uninstantiable from FHE + “lockable obfuscation” (both implied by circularly secure LWE)
SNARG
Q: TM or circuits?
Q: Can P make obfuscate queries?

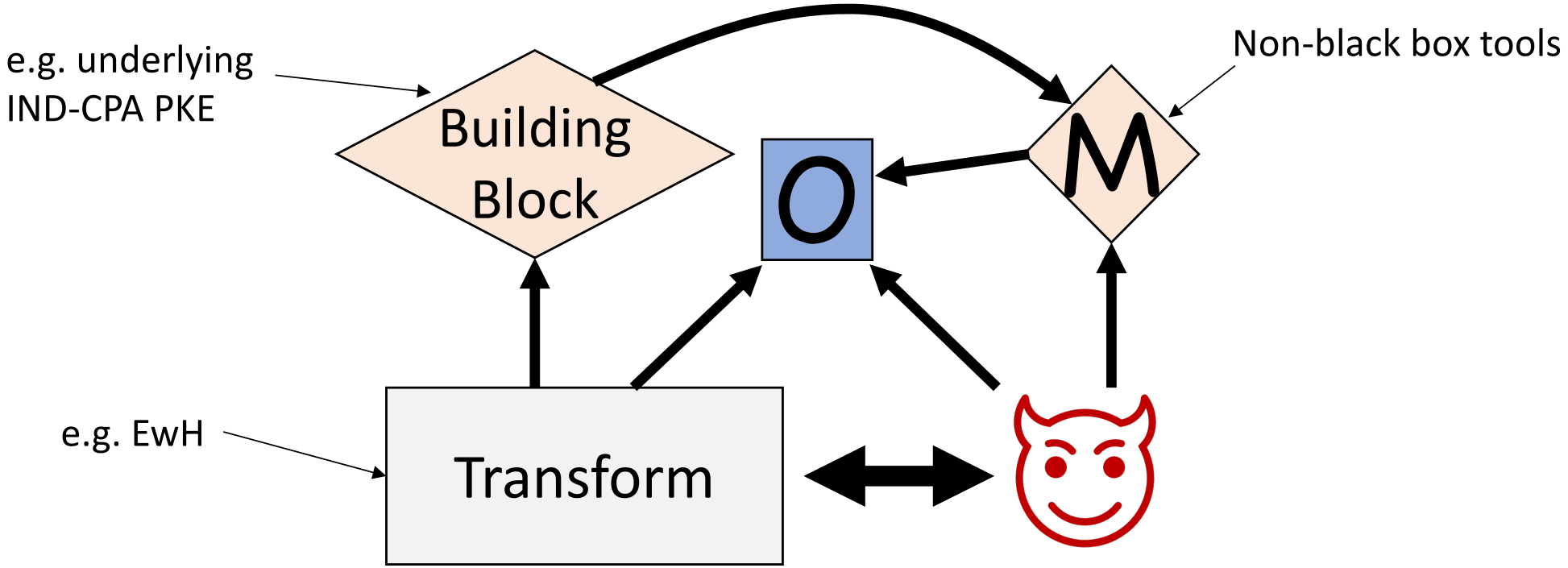

FHE
MPC
Yet-to-be-discovered
NIZK
O
Obf
Garbled circuits
SNARG
FSS
Q: TM or circuits?
Q: Can P make obfuscate queries?



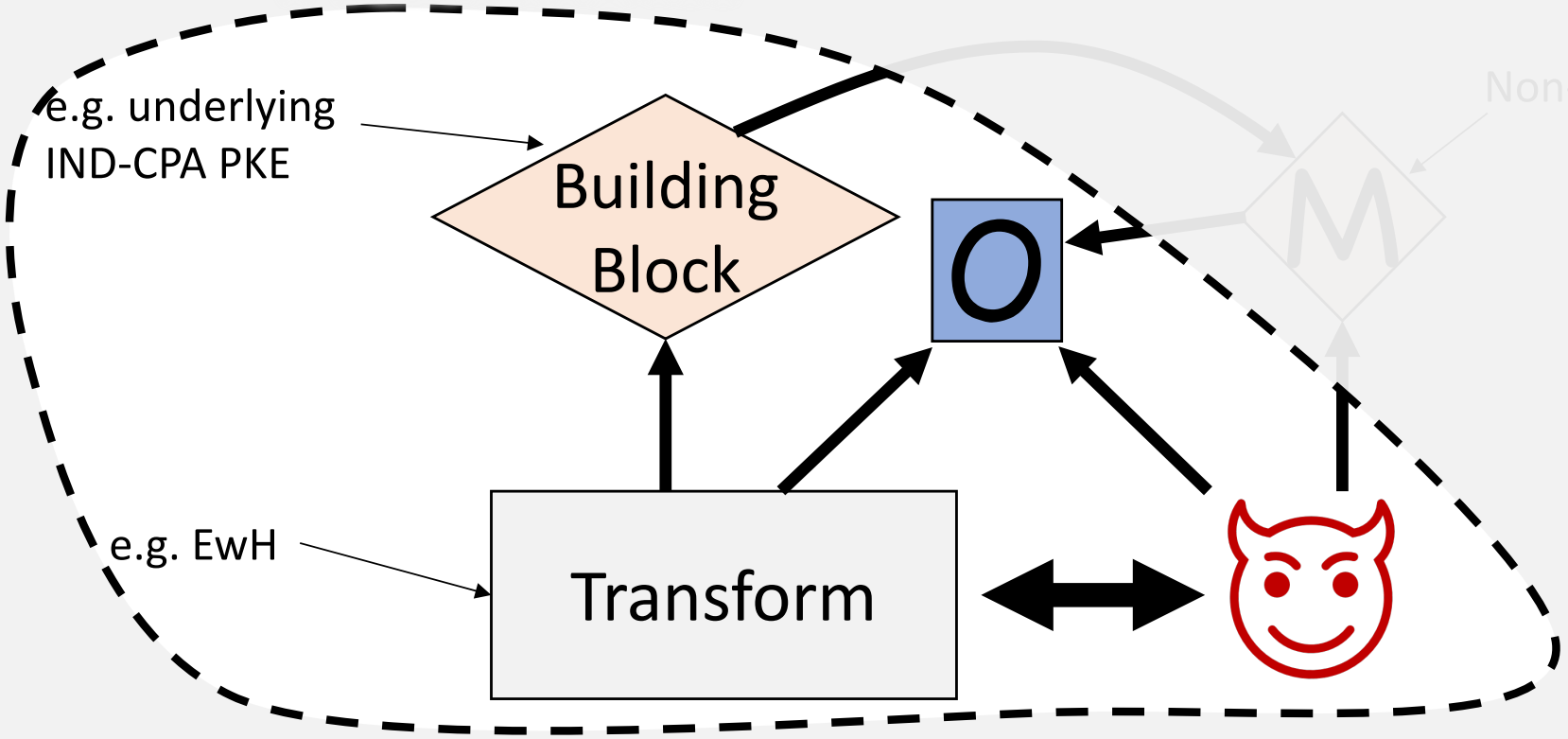
Q: TM or circuits?

Q: Can P make obfuscate queries?

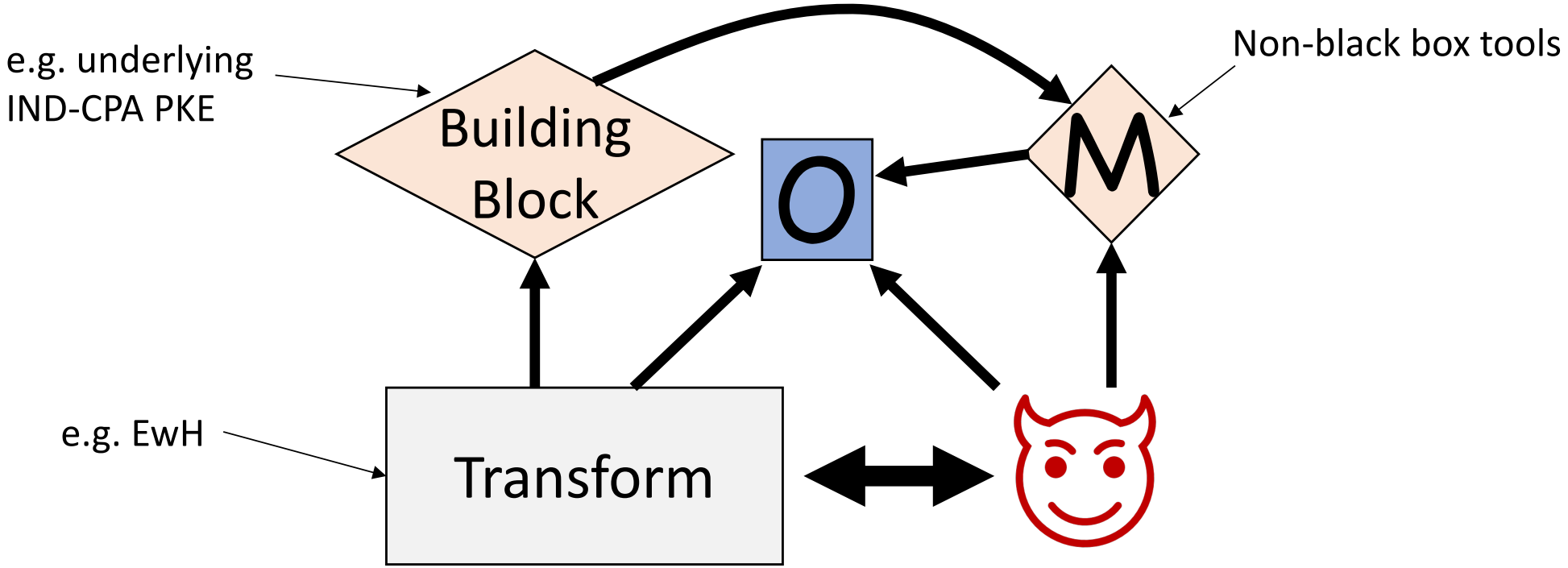
This Work: Augmented Random Oracle Model (AROM)



Def: Transform is secure in AROM if security holds **for all** possible building blocks (meeting prescribed security notion) and **all** efficient M


Plain ROM
e.g. underlying
IND-CPA PKE
Building
Block
O
e.g. EwH
Transform

This Work: Augmented Random Oracle Model (AROM)



Def: Transform is secure in AROM if security holds **for all** possible building blocks (meeting prescribed security notion) and **all** efficient M

Q: How to prove security in the AROM?

Can still do standard-model reductions. But anything else?

Q: Can the AROM prove anything beyond standard model?

Challenges:

- Observability: adversary may "hide" queries to O inside queries to M
- Programmability: reprogrammed O will be inconsistent with M

Thm (this work): **Lossy** PKE → EwH secure in AROM

[Wichs'13]: unlikely to prove in standard model

Thm (this work): Statistically sound public coin proof → Fiat-Shamir secure in the AROM

[Bitansky-Dachman-Soled-Garg-Jain-Kalai-López-Alt-Wichs'13]: unlikely to prove in standard model

Thm (this work):Lossy PKE → CCA-secure encryption in AROM

Not known in standard model

Idea: statistical properties of base cryptosystem
→ can brute-force O,M to observe/program O

Related Work:

- Non-programmable ROM [Nielsen'02, Fischlin-Lehmann-Ristenpart'10]
- Non-observable ROM [Ananth-Bhaskar'12]
- Universal computational extractors (UCE) [Bellare-Hoang-Keelveedhi'13]
- [Canetti'97,…]: instantiate certain ROM properties from well-established tools
- [Boneh-Boyen'04,…]: remove ROM from cryptosystem

AROM: Only model designed specifically based on uninstantiability results

Thanks!