Adaptive Garbled RAM from Laconic Oblivious Transfer

Sanjam Garg
UC Berkeley

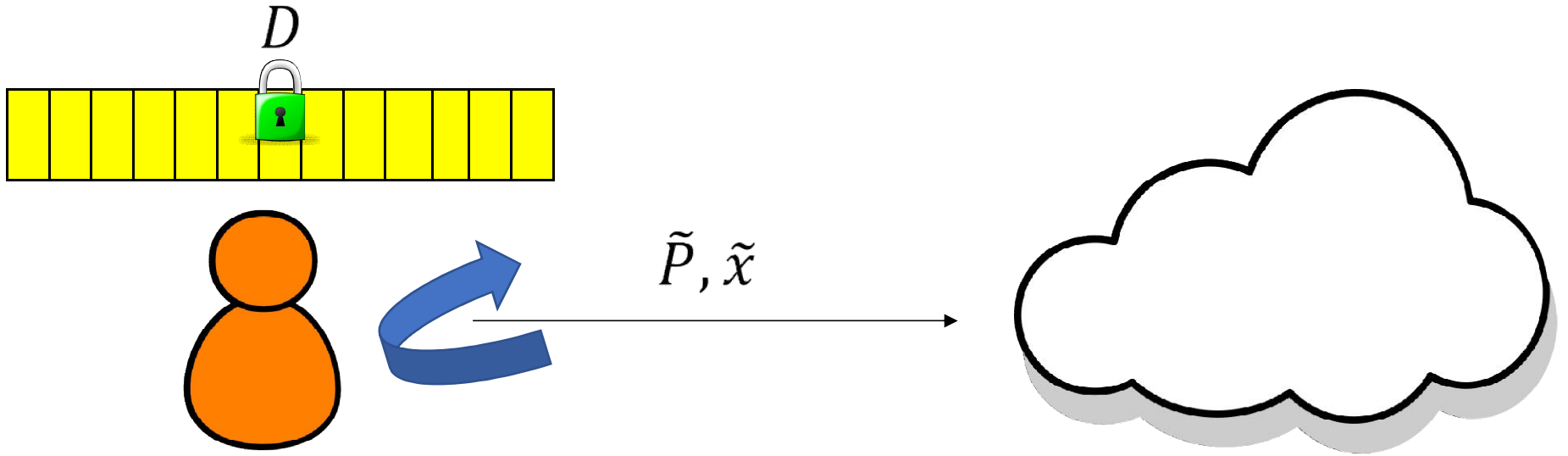
Rafail Ostrovsky
UCLA

Akshayaram Srinivasan
UC Berkeley

Crypto 2018

Garbled RAM

Lu-Ostrovsky 13



$$\left|\tilde{P}\right| = T.poly(\lambda, \log|D|)$$
$$|\tilde{x}| = |x|.poly(\lambda, \log|D|)$$

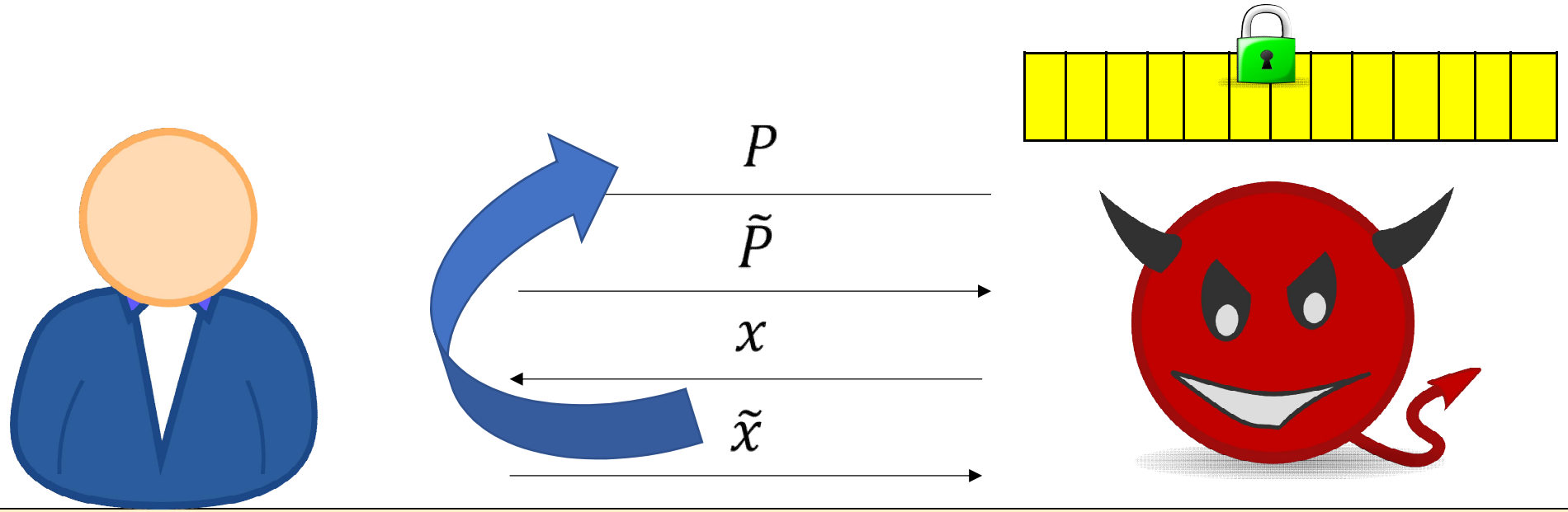
Learns only $P^D(x)$

Selective Security

Long line of work [LO13, GHLOW14, GLOS15, GLO15, LO17]

Adaptive Garbled RAM

[Canetti-Chen-Holmgren-Raykova16, Ananth-Chen-Chung-Lin-Lin16]



Can we construct Adaptive Garbled RAM from standard assumptions?

Prior constructions were either in the random oracle model [BHR12] or based on indistinguishability obfuscation [CCHR16, ACCLL16]

Why is Adaptive GRAM important?

Motivated by the study of Adaptive Garbled Circuits [BHR12,BGG+14,HJO+16,JW16,JKK+17,JSW17,G**S**18]

Applications: One-time programs[GKR08], Online-offline 2PC[LR14], Verifiable Computation[GGP10], Adaptive Compact FE[AS16]

Adaptive GRAM + $O(1)$ round Malicious MPC $=>$
$O(1)$ round Malicious MPC for RAM program in the persistent setting

Prior $O(1)$ round protocols based on standard assumptions [GGMP16,HY16,KY18] did not support persistence in the malicious setting

Our Results

Adaptive GRAM from Laconic OT

Theorem: There exists a construction of Adaptive GRAM from Laconic Oblivious Transfer.

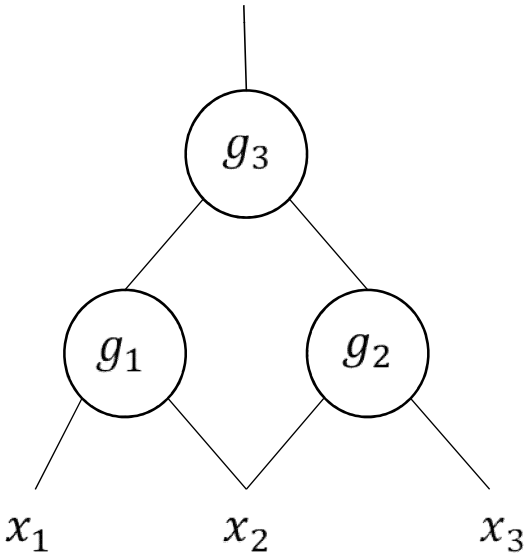
Corollary [CDG+16,DG17,BLSV18,DGHM18]: There exists a construction of Adaptive GRAM based on CDH/Factoring/LWE.

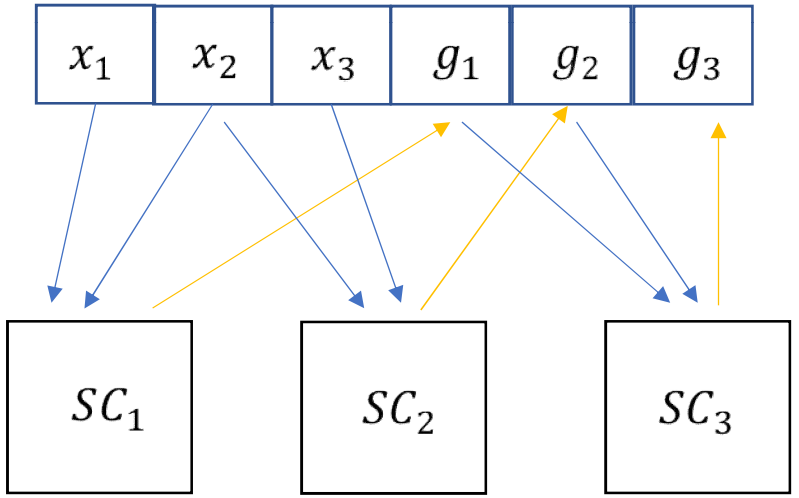
Rest of the talk

- Starting Point: Adaptive Garbled Circuits [Garg-**S** 18]
- Challenges in Extending to the RAM setting
- How to overcome the challenges?

Adaptive Garbled Circuits [Garg-S 18]

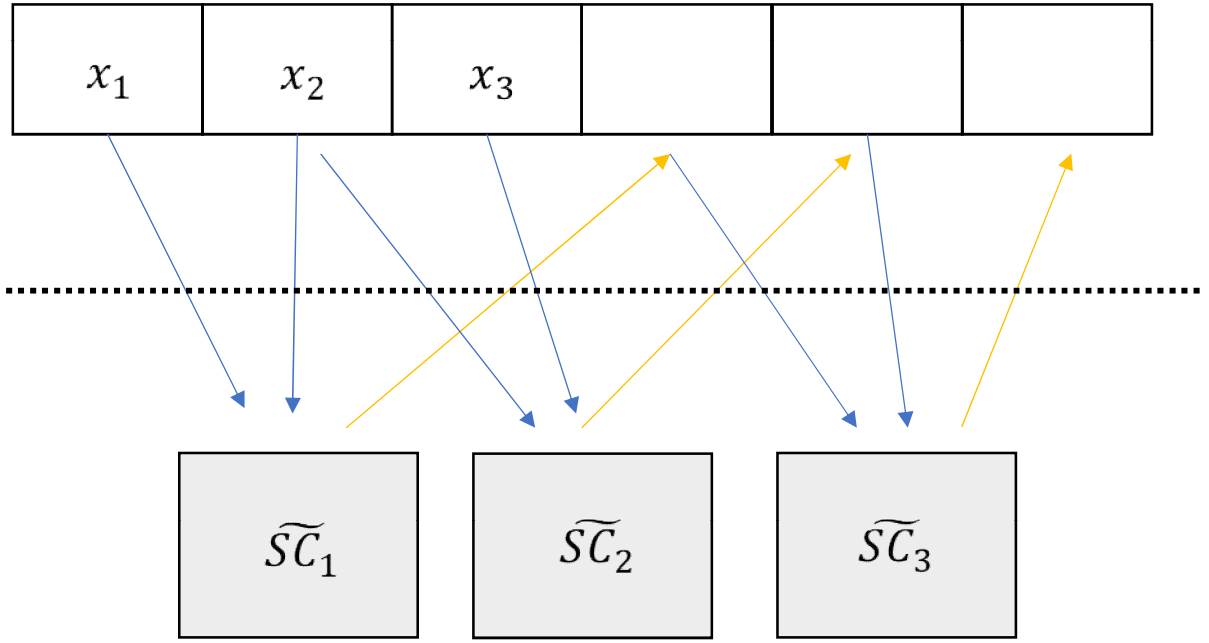
Alternate View of a Boolean Circuit





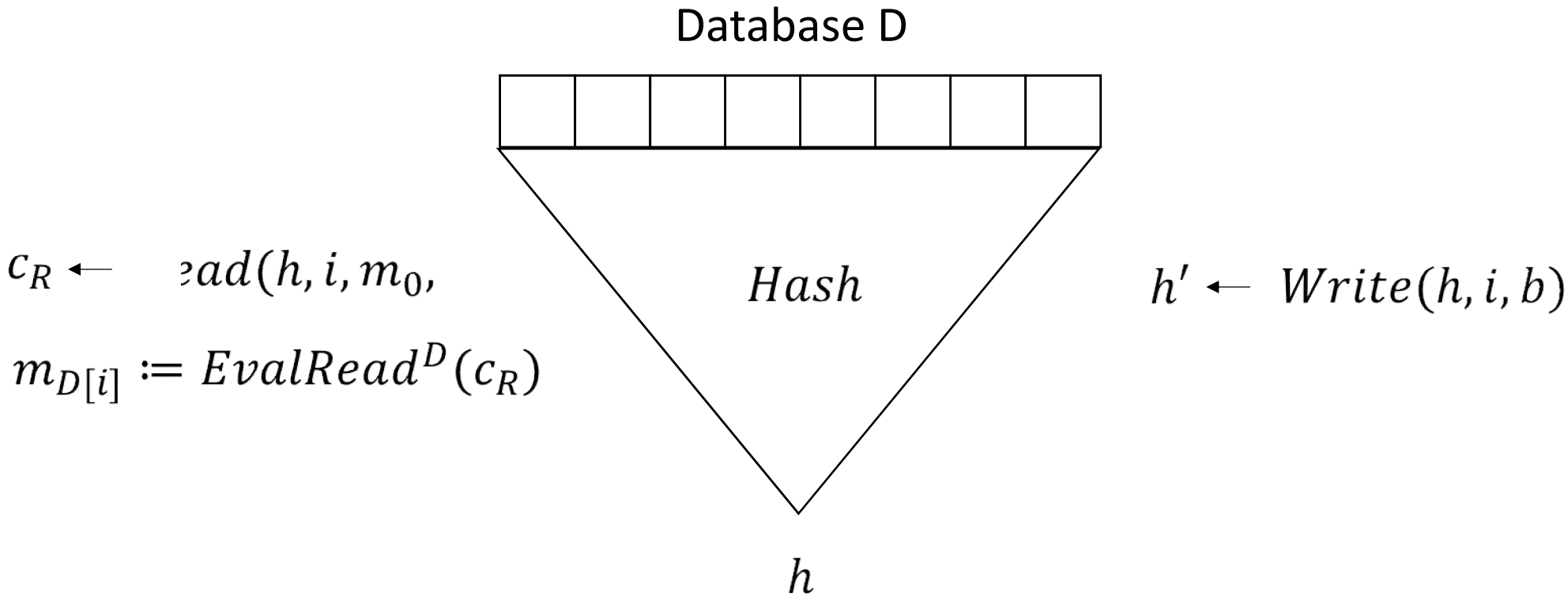
Garbling Step Circuits

Access the database via Laconic OT



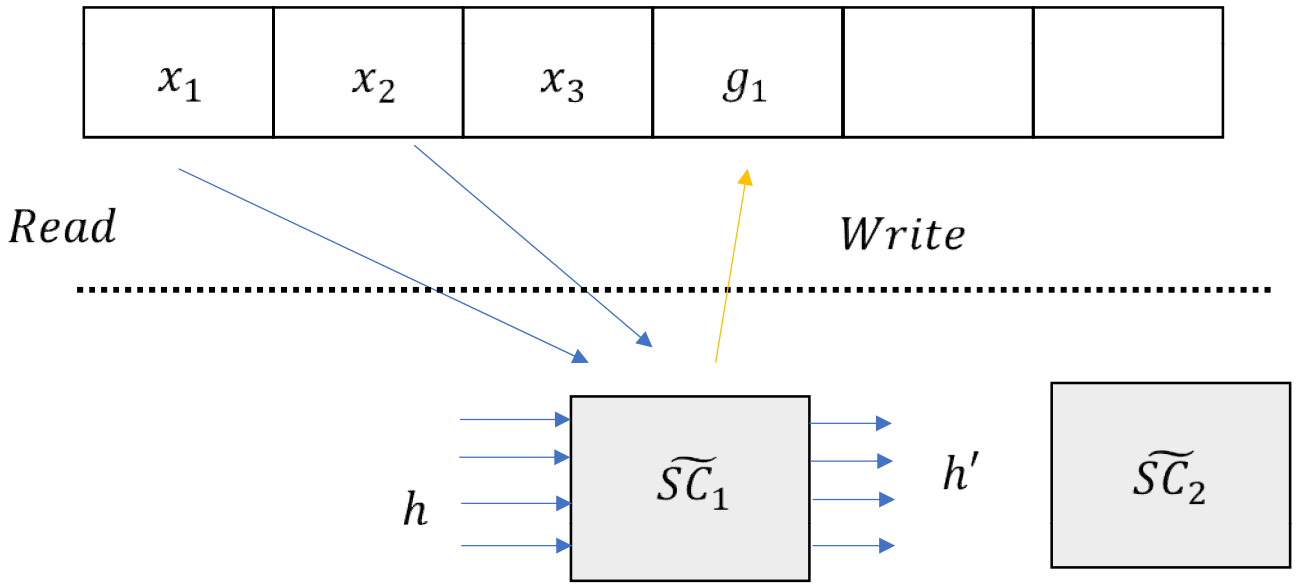
Updatable Laconic Oblivious Transfer

[Cho-Dottling-Garg-Gupta-Miao-Polychroniadou 17]

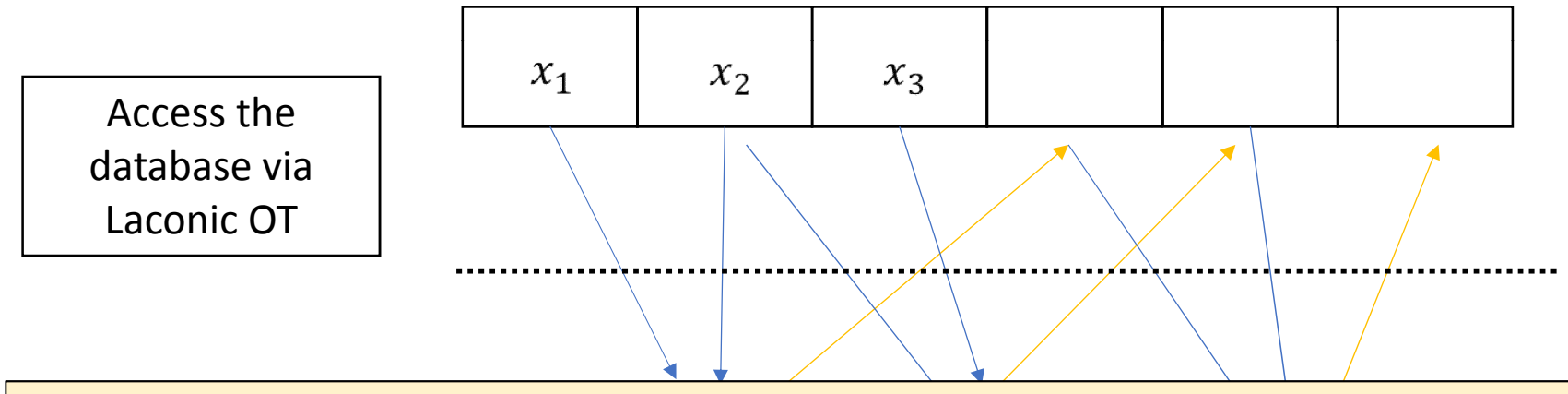


Theorem[CDG+16,DG17,BLSV18,DGHM18]: Assuming CDH/Factoring/LWE, there exists a construction of updatable laconic OT.

Using Laconic OT to access the database



Challenges in the RAM setting



Challenge-1: How to protect the database?

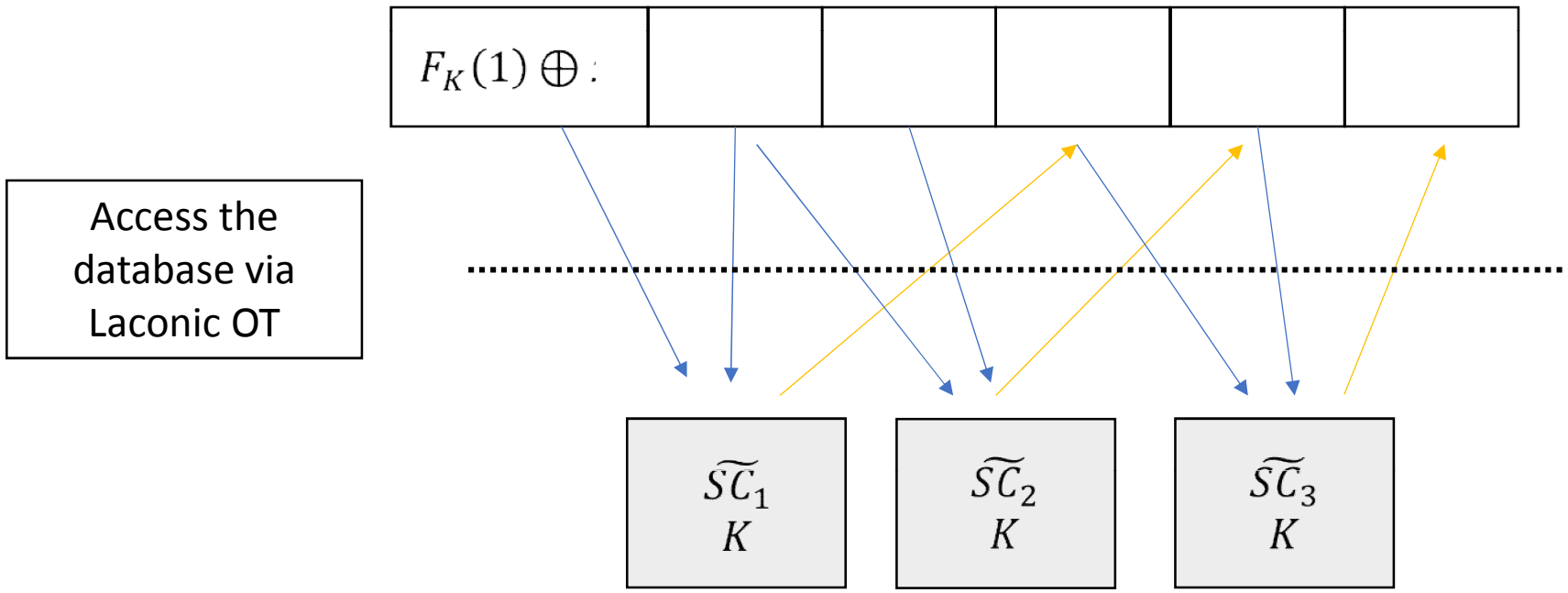
Challenge-2: How to protect the access pattern?

In the adaptive setting, more sophisticated tools are needed.

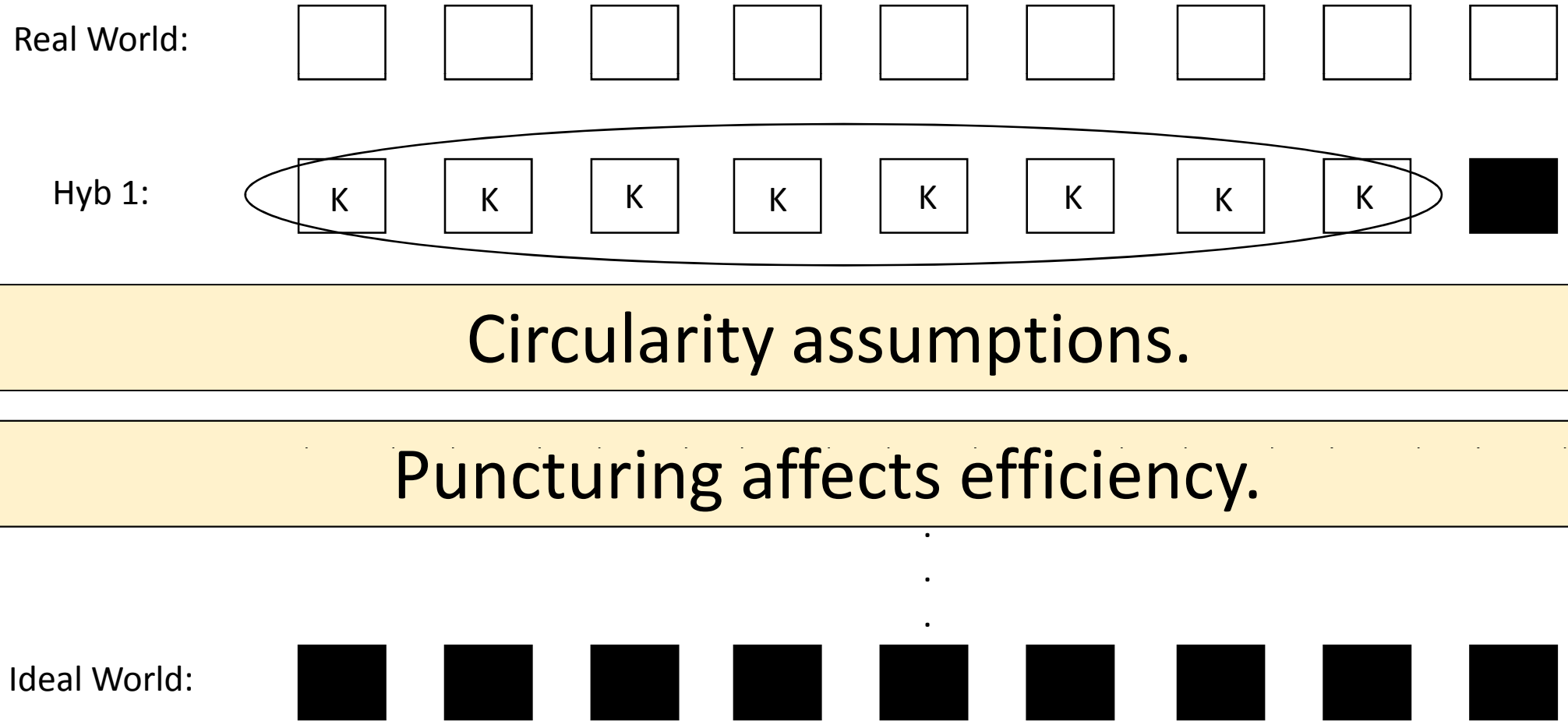
In the selective setting [GHLOW14], transforming from unprotected memory access to full security is done via a ORAM scheme and symmetric encryption.

Protecting the Database

Prior Approaches: Location based Encryption



[GS18]- Hybrid Argument



Our Approach: Timed Encryption

$$c \leftarrow Enc(time, k, msg)$$

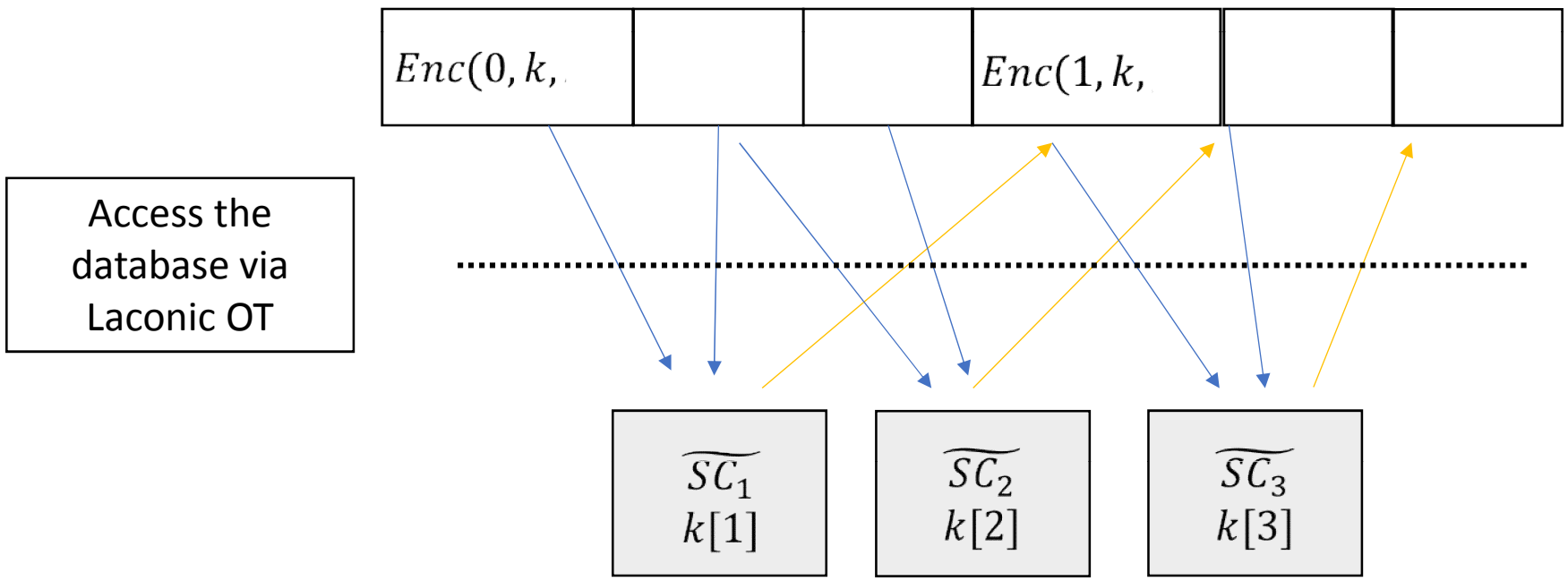
$$k[time'] \leftarrow KeyCons(time', k)$$

$$\mathrm{m}sg \leftarrow Dec(k[time'], c)\ if\ time' \geq time$$

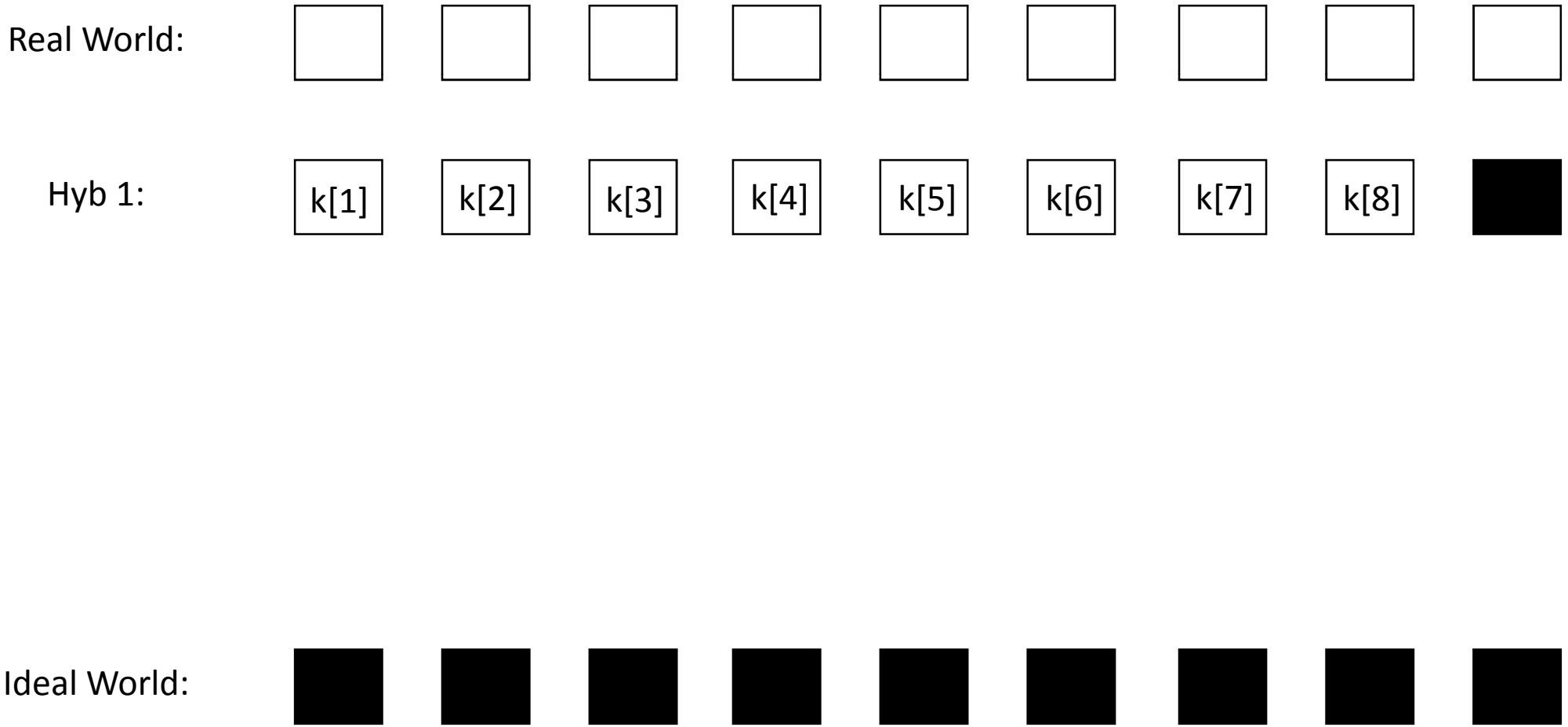
$$(k[time'], c) \approx_c (k[time'], Enc(time, k, 0))\ if\ time' < time$$

Theorem: There is a construction of timed Encryption from one-way functions.

Using Timed Encryption



Revisiting the Hybrid Argument



Conclusion

- We give a construction of Adaptive Garbled RAM from CDH/Factoring/LWE.
- We obtain the first O(1) round malicious MPC for RAM programs in the persistent setting from standard assumptions.
- Open question: Can we remove public-key assumptions?

Thank you!

https://eprint.iacr.org/2018/549