New Advances in Secure RAM Computation

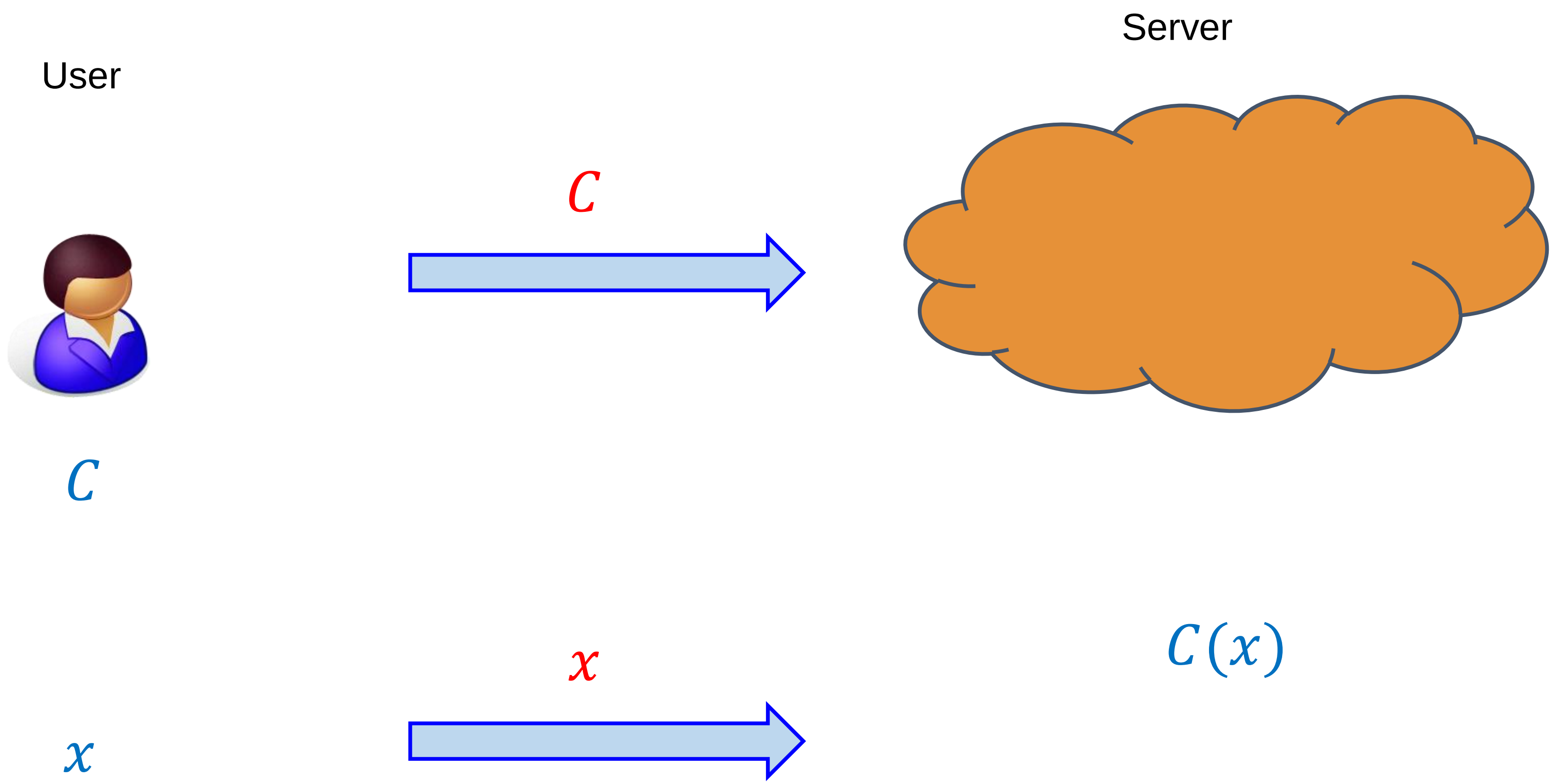
Sanjam Garg

University of California, Berkeley

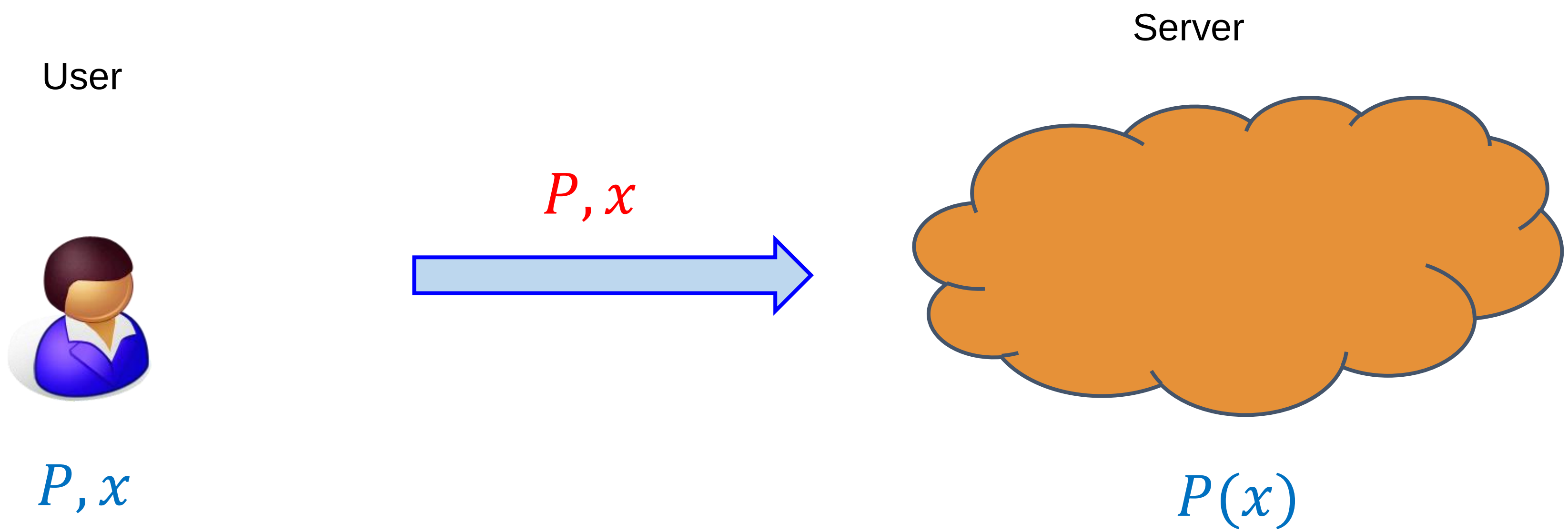
Based on joint works with

Steve Lu, Payman Mohassel, Charalampos Papamanthou, Rafail Ostrovsky and Alessandra Scafuro

Yao's garbled circuits



RAM analogue of Garbled circuits

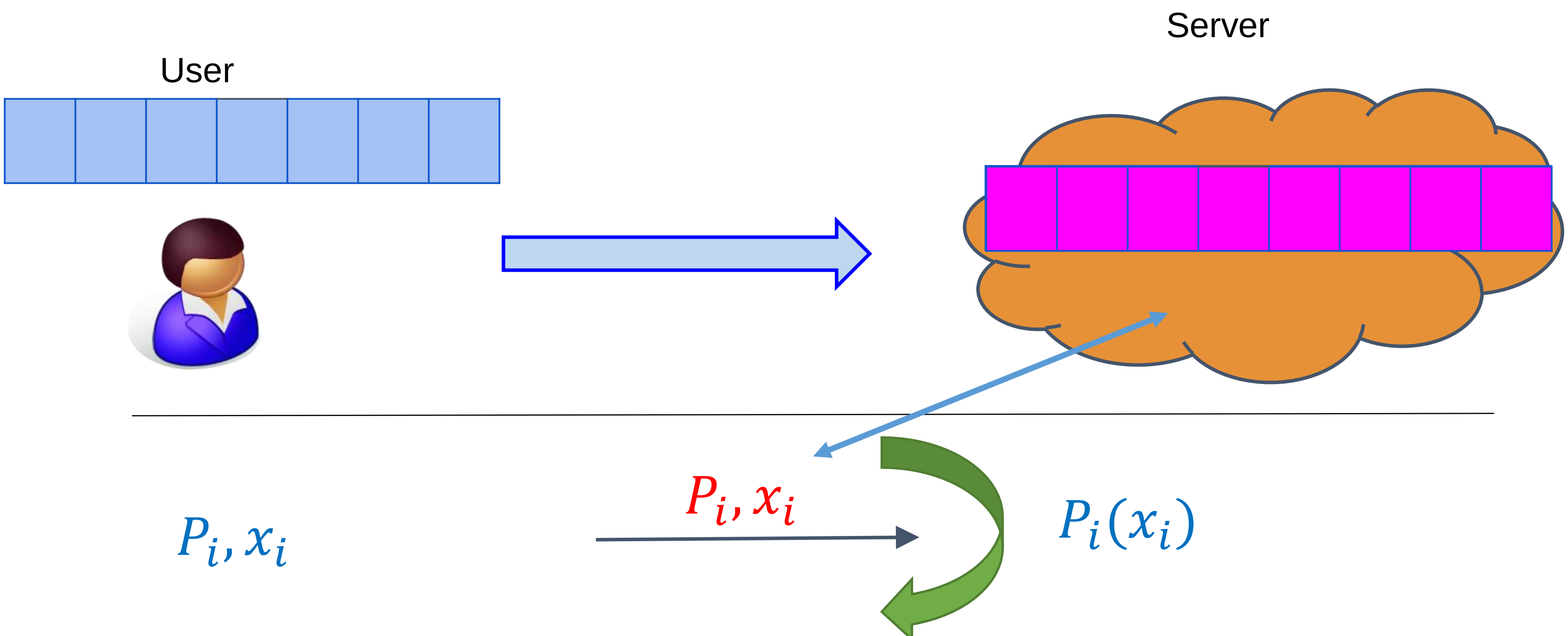


If the running time of the program P is T then the corresponding circuit is of size T^3.

Communication complexity and computational complexity of both parties grows with T^3.

More Ambitious: Garbled RAM

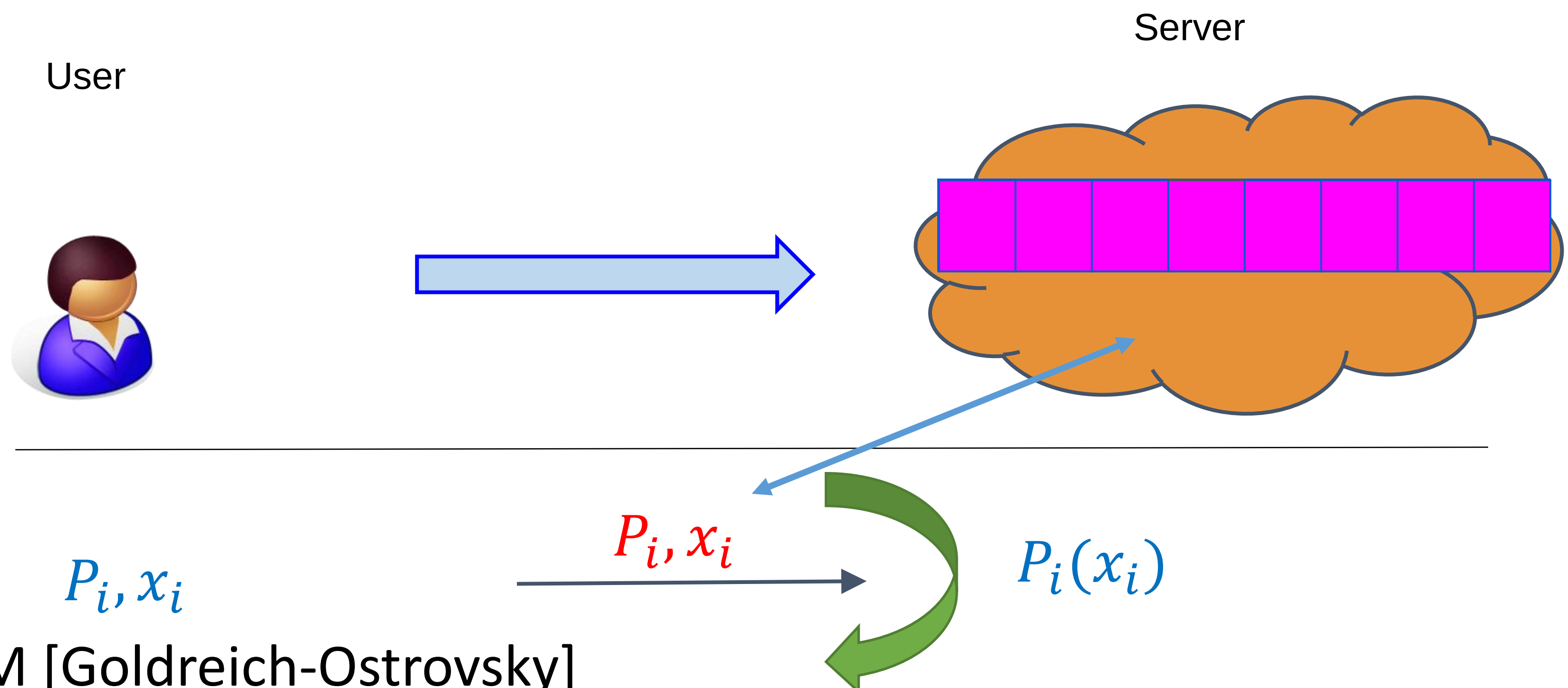
[LO13,GHLORW14]



- Size of garbled database is $\tilde{O}(|D|)$
- Communication and computation cost grows in $\tilde{O}(T_i)$

More Ambitious: Garbled RAM

[LO13,GHLORW14]



ORAM [Goldreich-Ostrovsky]

- Full-security: Server learns nothing but the output
- Unprotected Memory Access (UMA): Server learns access pattern.

Putting in context – Secure Computation

- Traditional protocols – have large round complexity
  - Linear in running time [OS97, GKKKMR12 …]

- Seeking an analogue of Yao's garbled circuits
  - Non-interactive

Landscape: Garbled RAM

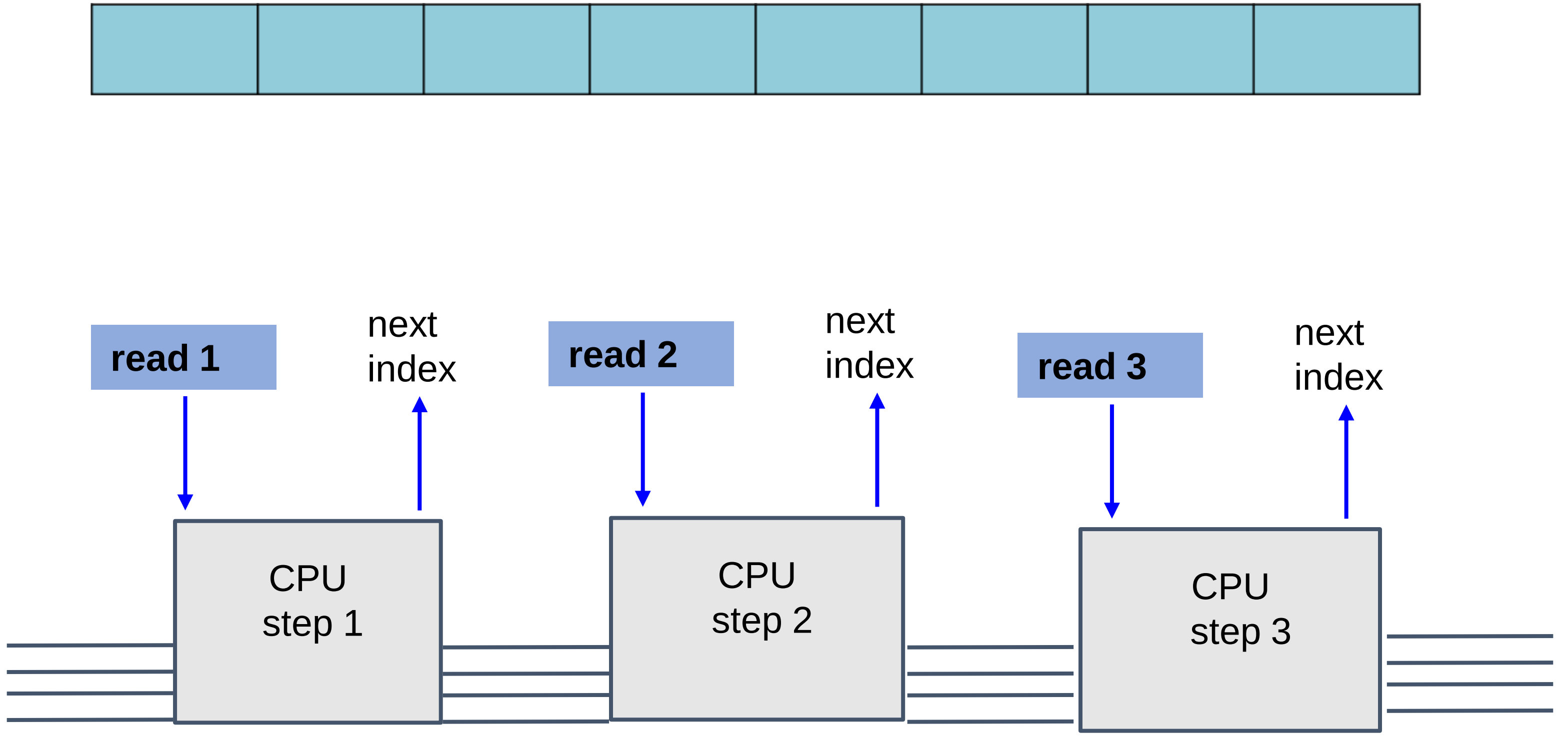
- Heuristic construction from OWFs [LO13]
  - Circularity Issue
  - Fixed using IBE [GHLORS14]
- Construction from OWFs [GLOS15]
- Using only black-box use of OWFs[GLO15]
  - OWF can't be modeled as a random oracle

- Not talk about succinct constructions based on iO [CHJV14, BGT14, LP14, KLW15, CH15, CCCLLZ15...]

Outline of the rest of the talk

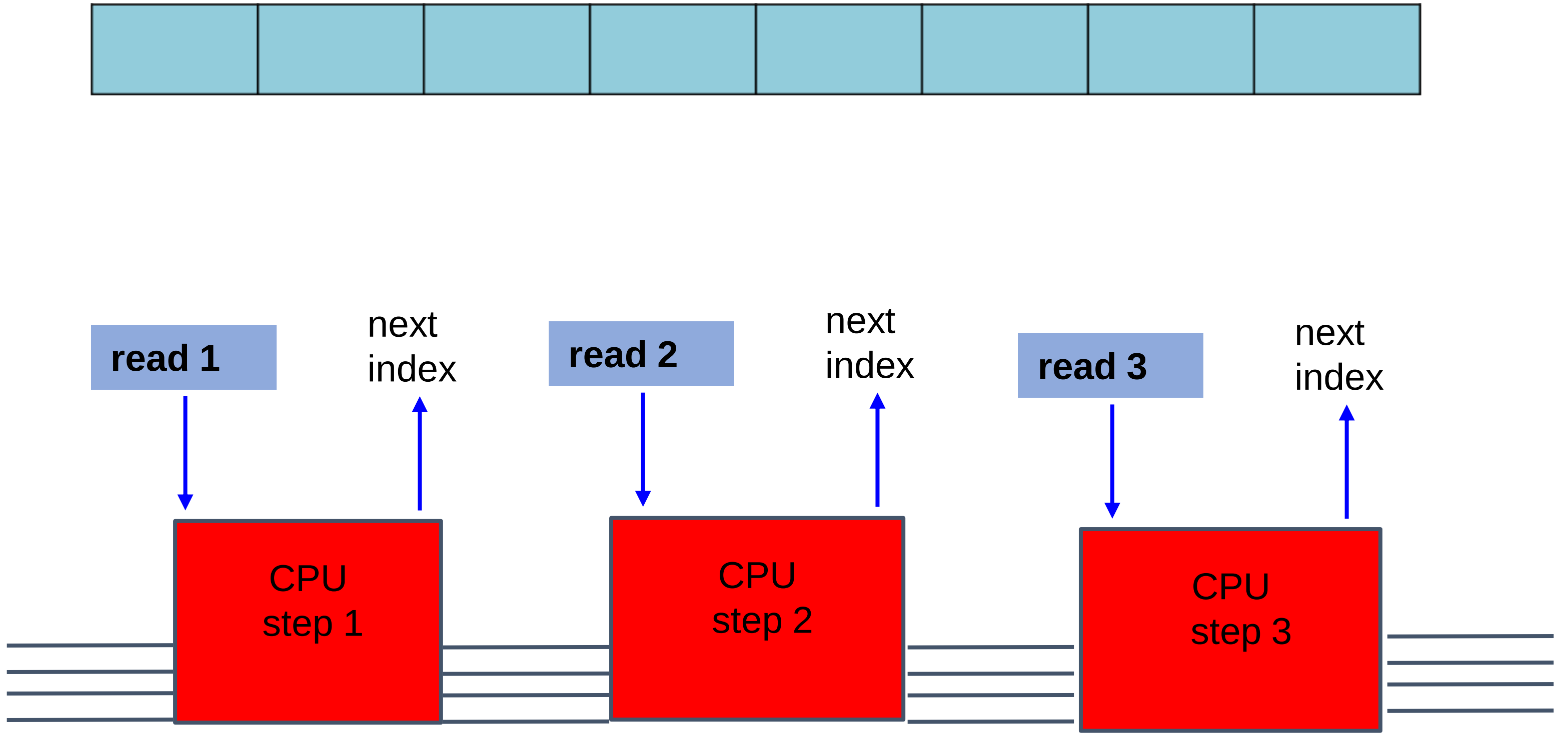
- RAM model
- LO13 approach
- Technical bottleneck in realizing black-box construction
- High level idea of black-box construction [GLO15]
- Extensions [GMP15,GM15,GGMP15,GP15]

RAM Model



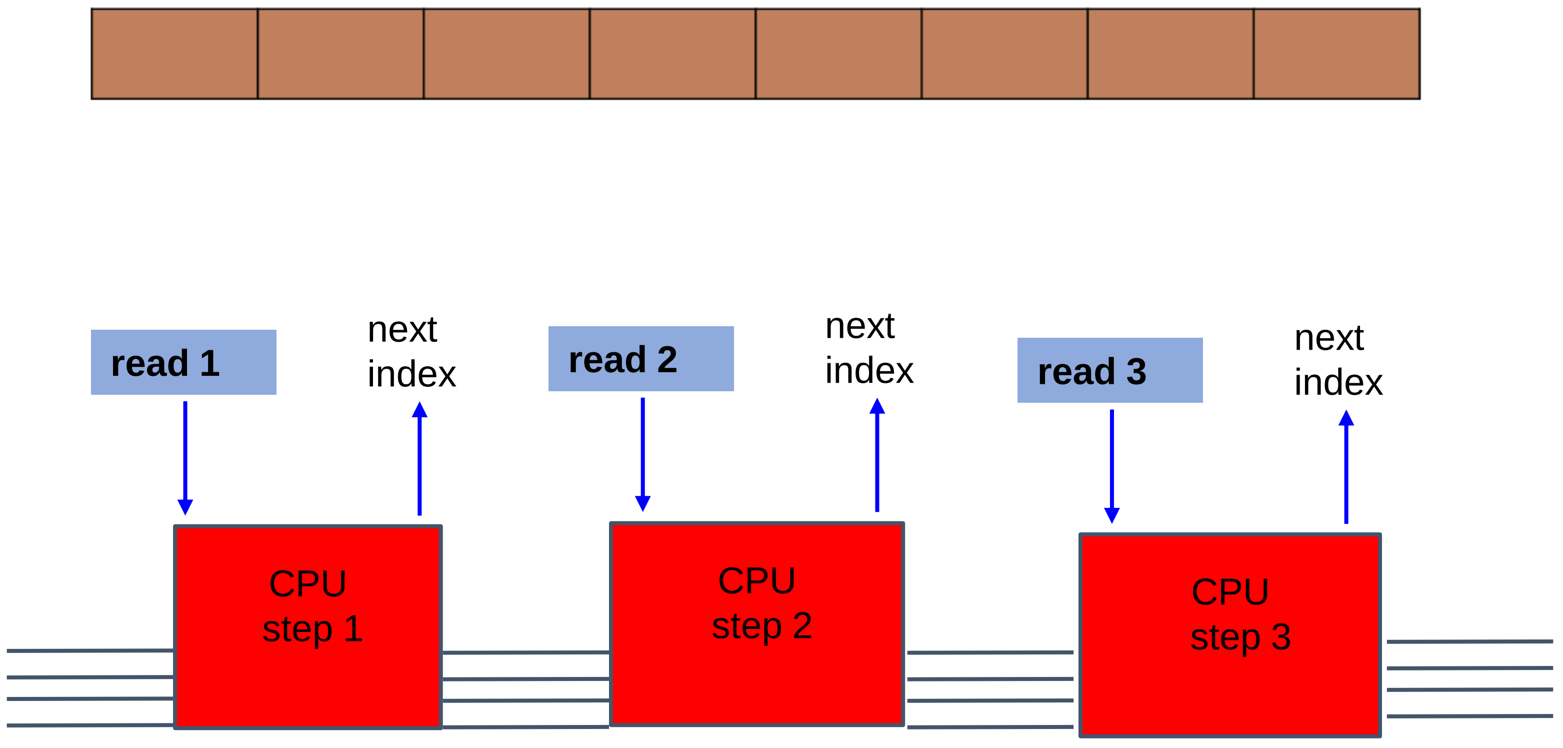
Writes require additional work but let's ignore that!

LO13 approach



Use garbled circuits!

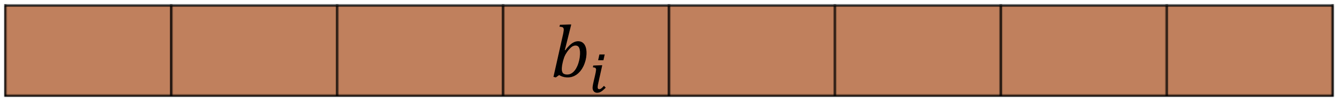
LO13 approach



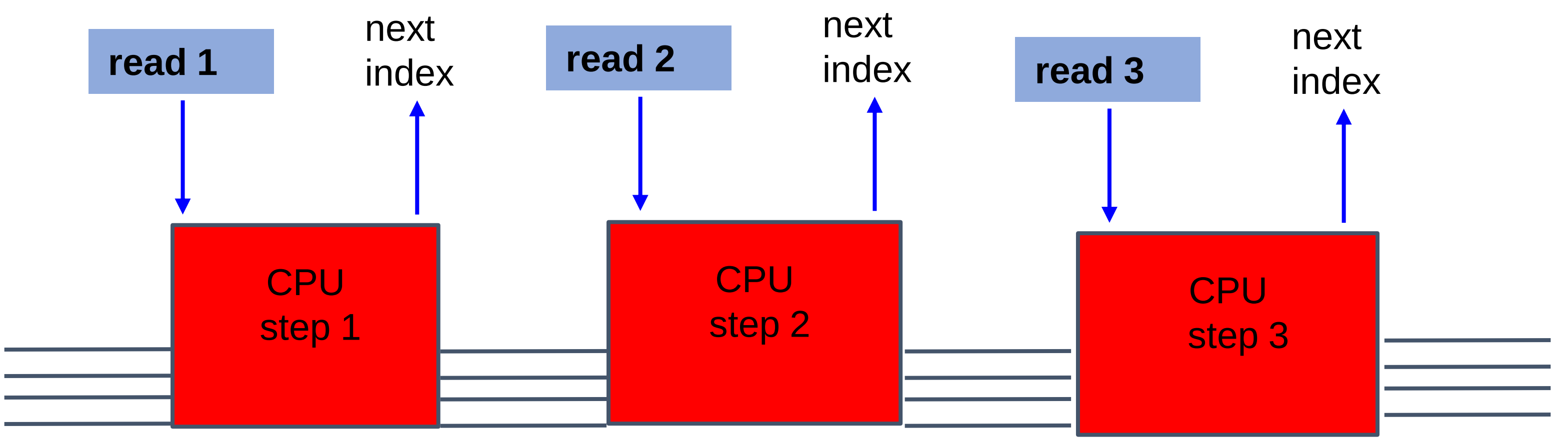
1) Somehow encrypt memory
2) translate table

LO13 approach

STEP 1: garbling/encrypting of the memory



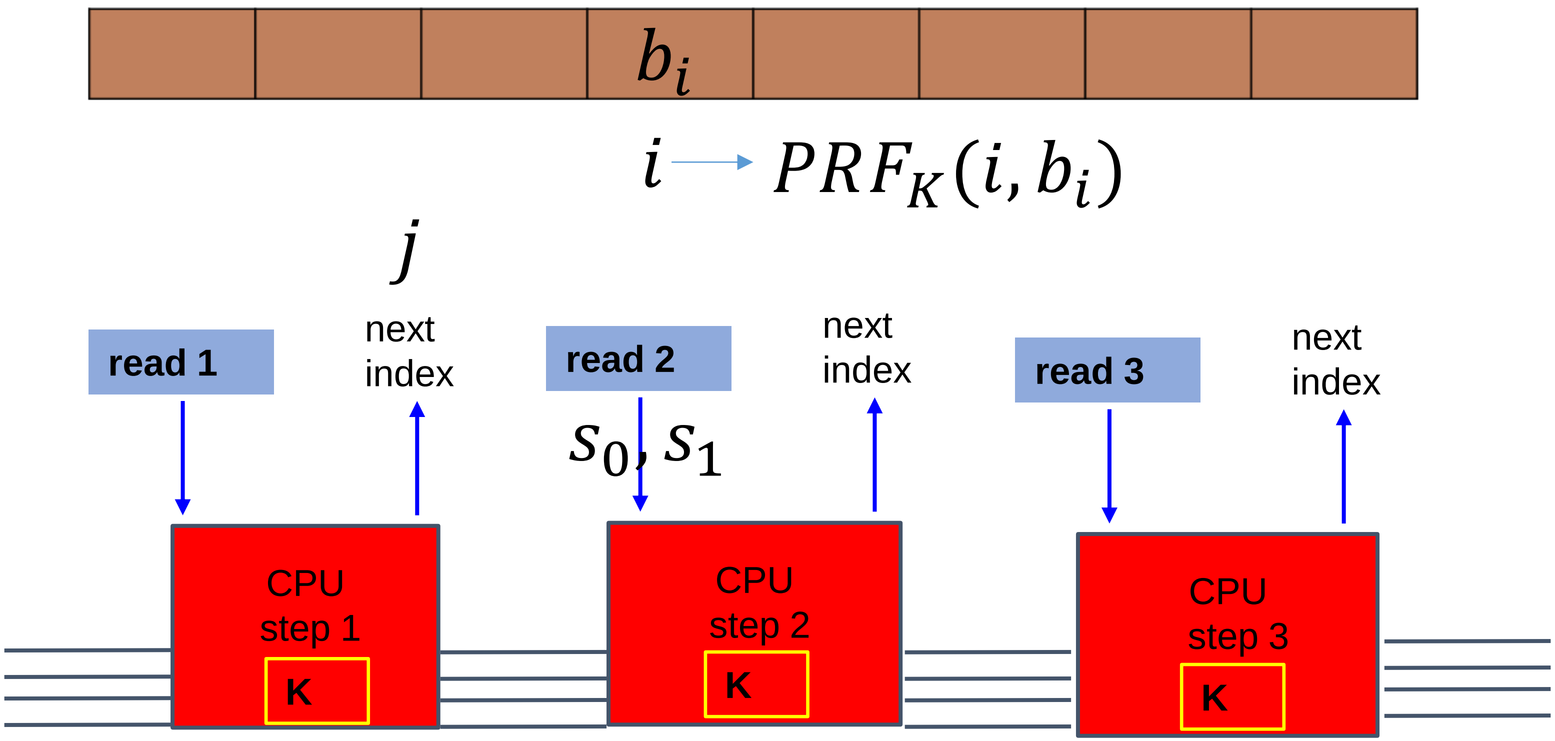
$$i \rightarrow PRF_K(i, b_i)$$



- PRF key **K to garble**

LO13 approach

STEP 2: translate table



$Enc(PRF_K(j,0), s_0)$

$Enc(PRF_K(j,1), s_1)$

- PRF key **K to garble**

Technical Bottleneck in Black-Box

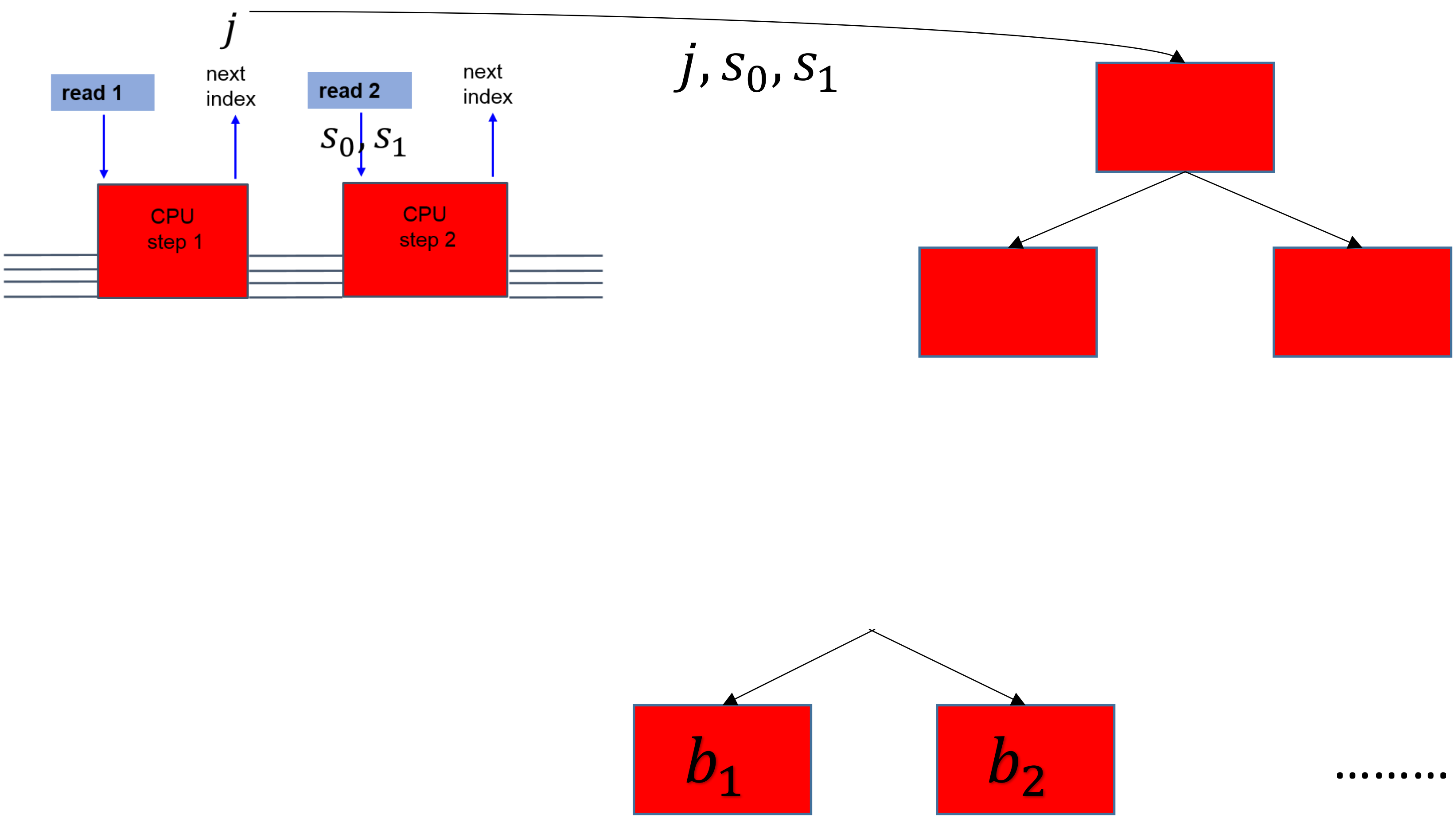
- The data needs to be encrypted so that the server doesn't learn it!

- CPU step garbled circuits need to decrypt the read values internally
  - Need of black-box use of cryptography seems inherent

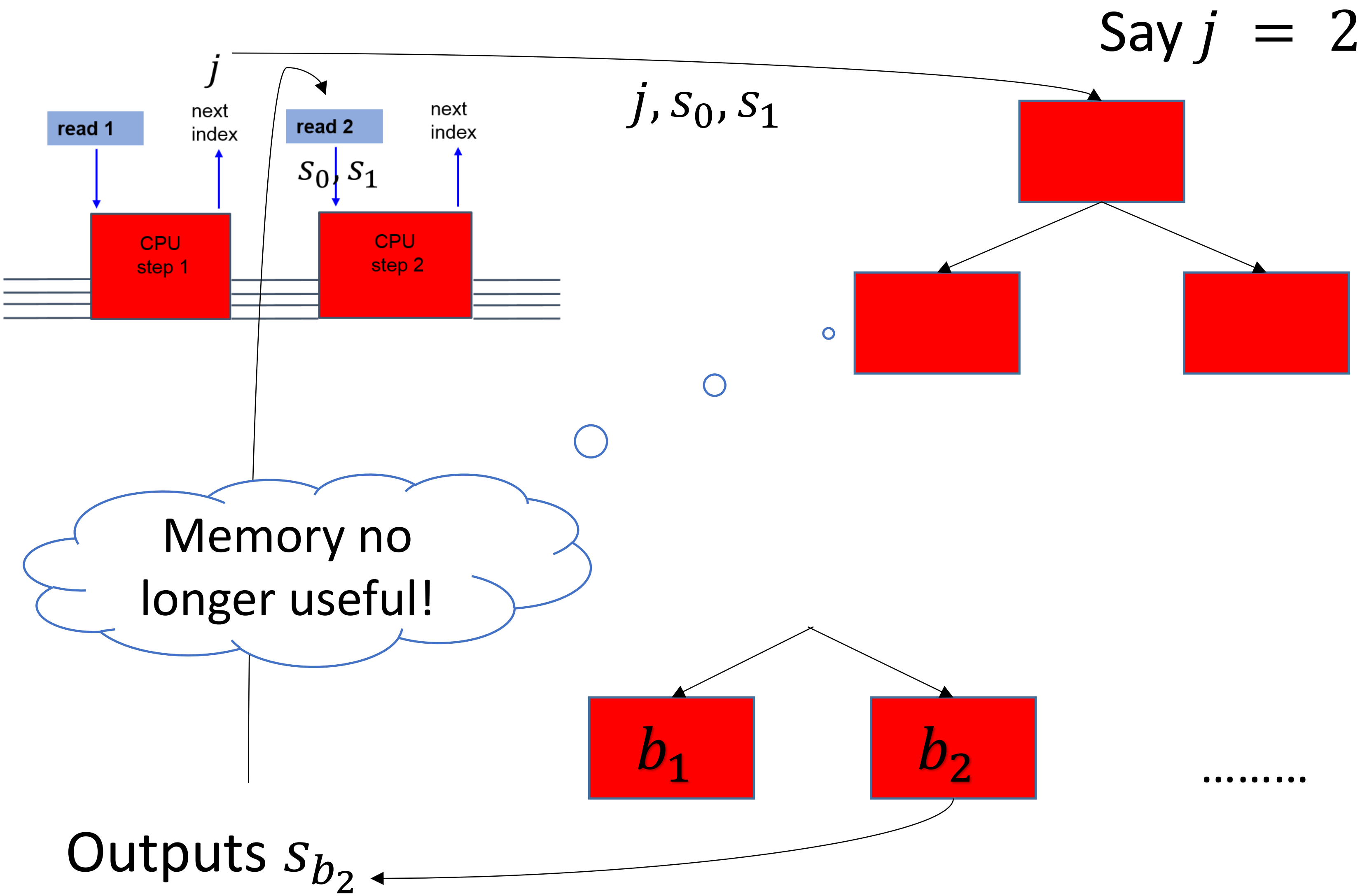
GLO15 high level idea

- Garbled memory comprises of a collection of garbled circuits with data values hardwired in them
- Read implemented by a sub-routine call
    - Control flow is passed to memory circuits

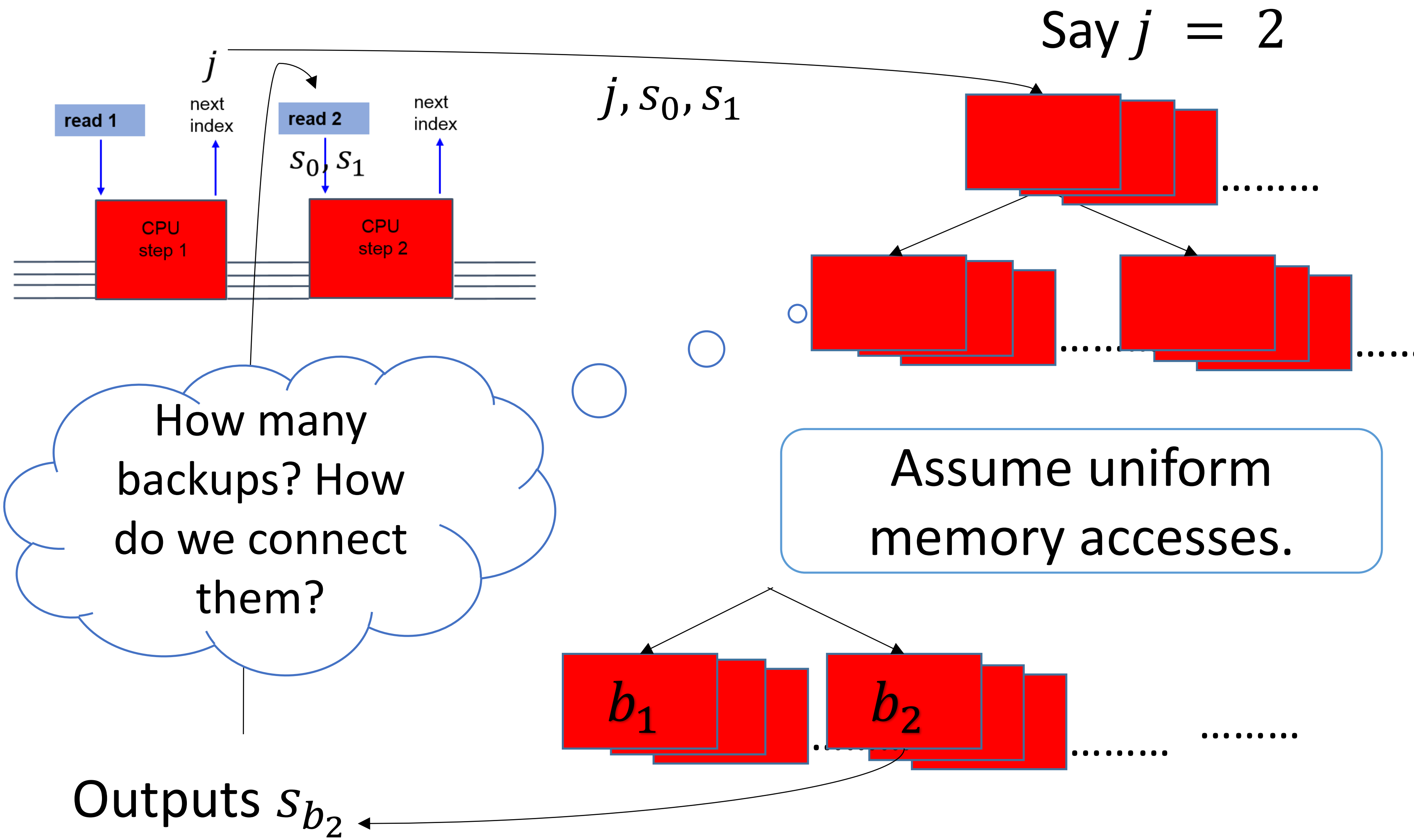
GLO15 – for one read only



GLO15 – for one read only



GLO15 – for m reads only



Conclusion and Open Problems

- Secure Computation for RAM programs

  Round Efficient

  And Black Box
- Important for crypto for big data
- Theoretically practical secure computation.

Thanks!