Compressing Vector OLE

Elette Boyle, *Geoffroy Couteau,* Niv Gilboa, Yuval Ishai

IDC HERZLIYA

Karlsruher Institut für Technologie

Ben-Gurion University of the Negev

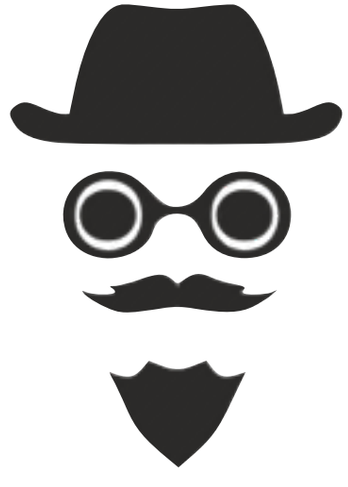
TECHNION Israel Institute of Technology

Secure Communication

M

Secure Communication

M xor K



M

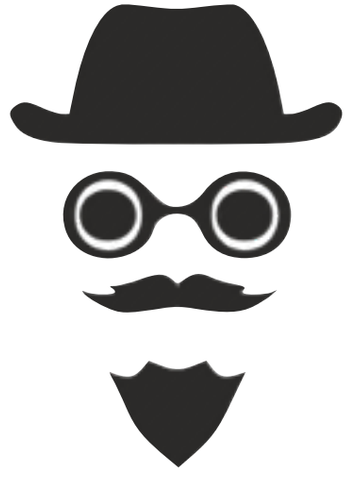
K

Shannon, 1944

K

Secure Communication

M xor K



M

K

Shannon, 1944

K

Long, random equality correlation

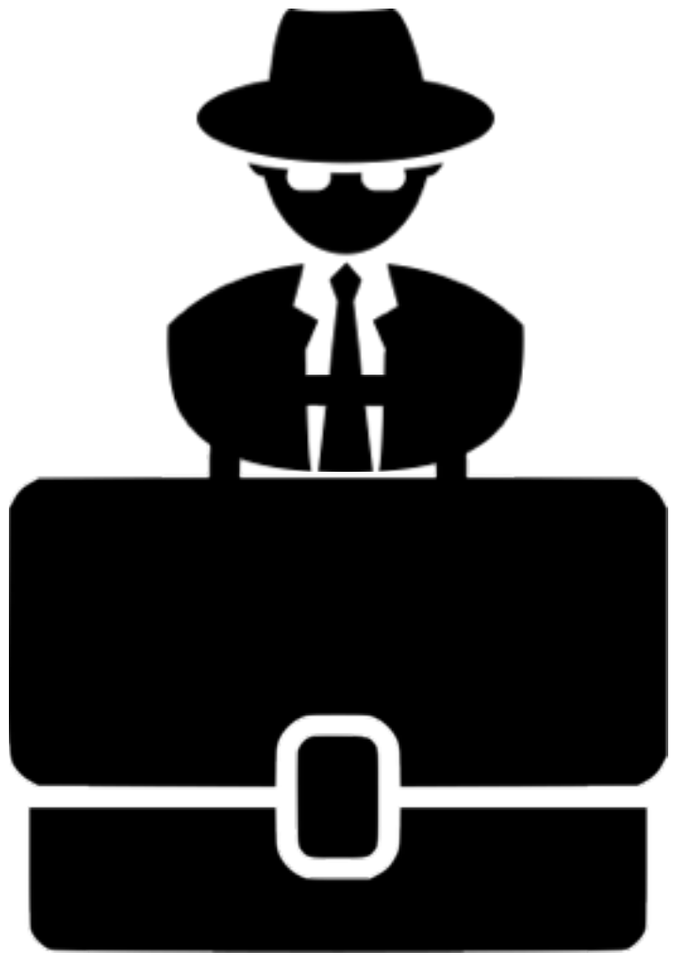
$\Longleftrightarrow$ perfect secure communication

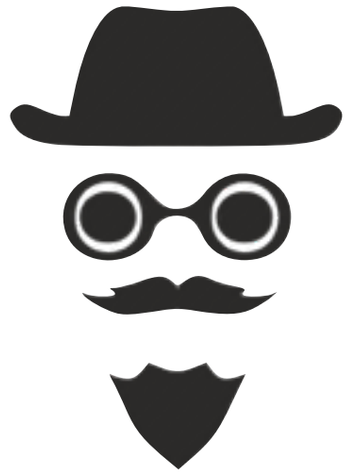
Secure Communication

M

K

M xor K





K

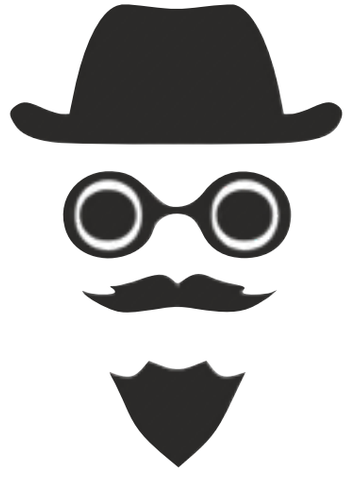
Secure Communication

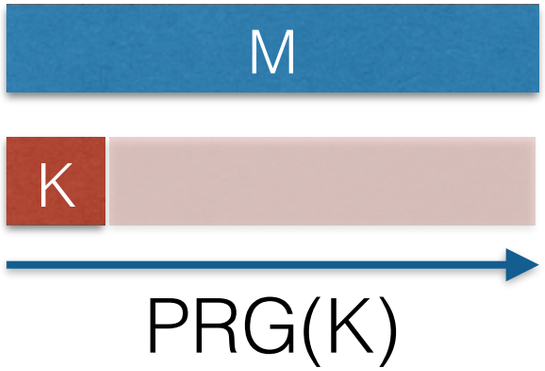
M

K

K

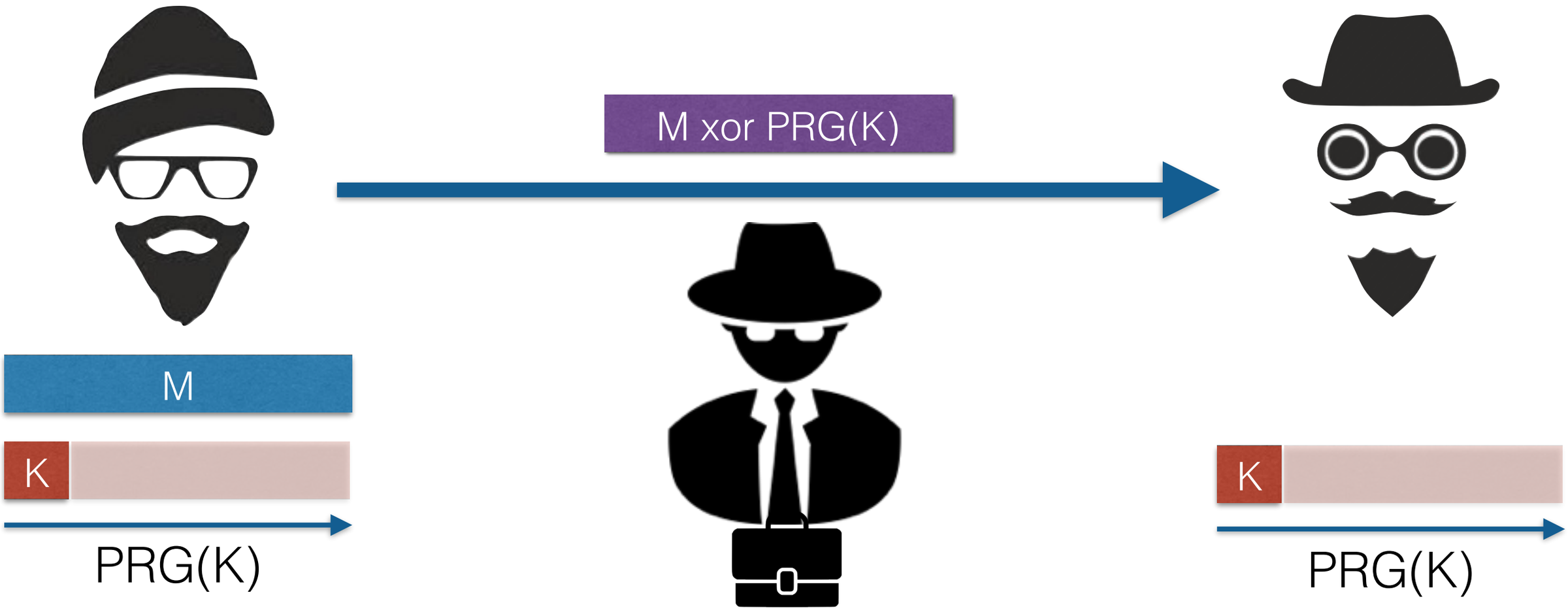
Secure Communication





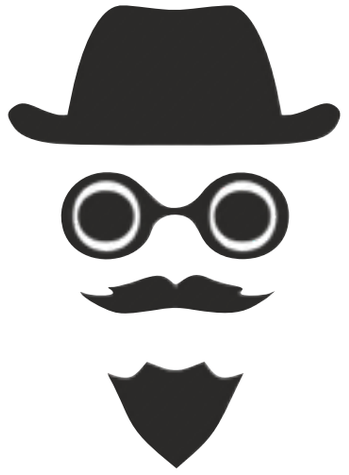
Short keys generate long (pseudo)-random correlations, enabling secure communication

Secure Communication



Short keys generate long (pseudo)-random correlations, enabling secure communication

Secure Computation



m_0

m_1

$\cdots$

x_0

x_1

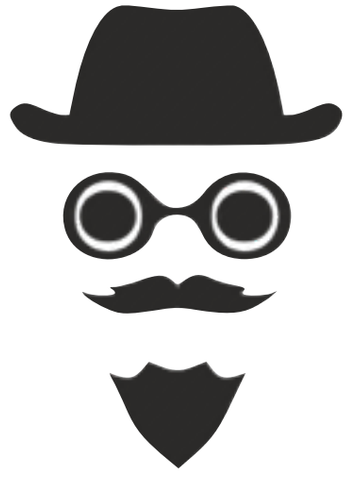
$y_0 = f_0(x_0, x_1)$

$y_1 = f_1(x_0, x_1)$

Secure Computation

m_0

m_1



$\cdots$

x_0

x_1

$y_0 = f_0(x_0, x_1)$

$y_1 = f_1(x_0, x_1)$

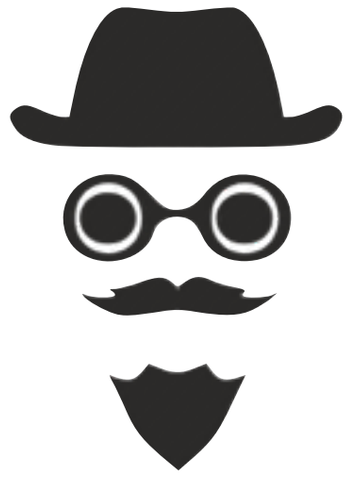
K0

K1

Secure Computation

m_0

m_1



$\cdots$

x_0 x_1

Beaver, 1995

$y_0 = f_0(x_0, x_1)$ $y_1 = f_1(x_0, x_1)$

K0 K1

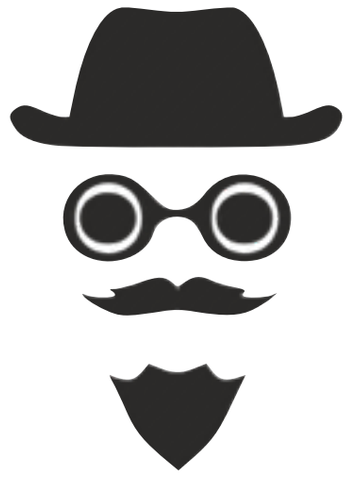
Long, random ‘more complex’ correlation

$\Longleftrightarrow$ perfect secure computation

Secure Computation

m_0

m_1



$\dots$

x_0 x_1

$y_0 = f_0(x_0, x_1)$ $y_1 = f_1(x_0, x_1)$

Beaver, 1995

K0 K1

Long, random ‘more complex’ correlation

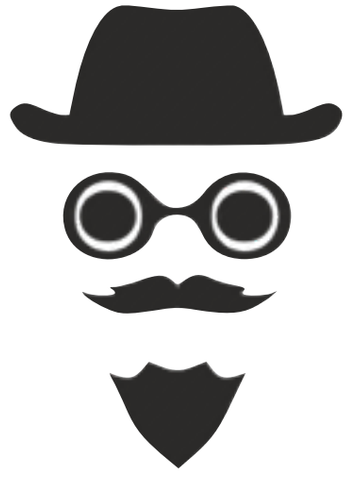
$\Longleftrightarrow$ perfect secure computation

[Bea95, Bea97, IPS09, BDOZ11, BIKO12, NNOB12, DPSZ12, IKMOP13 DZ13, DLT14, BIKK14, LOS14, FKOS15, DZ16, KOS16, DNNR17, C18…]

Secure Computation

m_0

m_1



x_0

$y_0 = f_0(x_0, x_1)$

K0

x_1

$y_1 = f_1(x_0, x_1)$

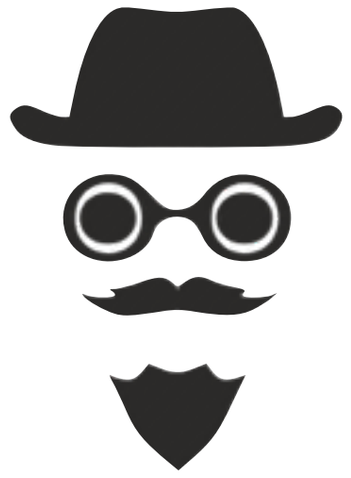
K1

[Bea95, Bea97, IPS09, BDOZ11, BIKO12, NNOB12, DPSZ12, IKMOP13 DZ13, DLT14, BIKK14, LOS14, FKOS15, DZ16, KOS16, DNNR17, C18…]

Higher Degree Pseudorandom Correlations

m_0

m_1



$\cdots$

x_0 x_1

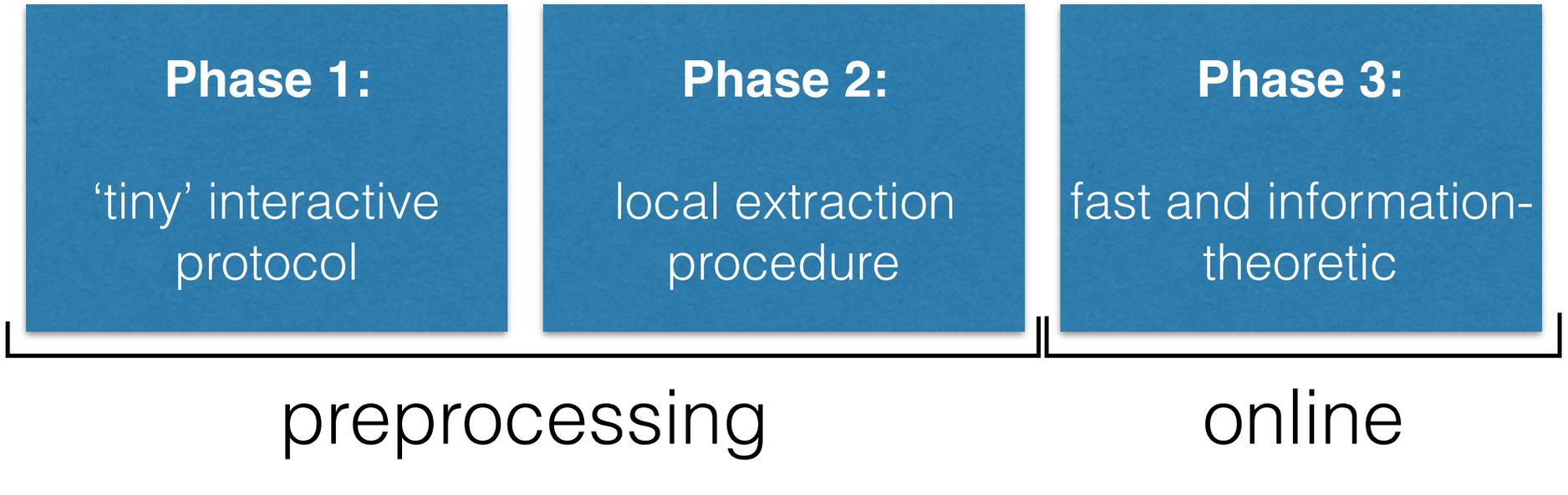
$y_0 = f_0(x_0, x_1)$ $y_1 = f_1(x_0, x_1)$

PRG0(K0)

Long pseudorandom correlation
gives comp. secure computation

PRG1(K1)

Why would it be so cool?



- Reduced overall communication
- ‘On demand’ (or ‘business-card’) MPC
- Hiding communication pattern

Previous Works

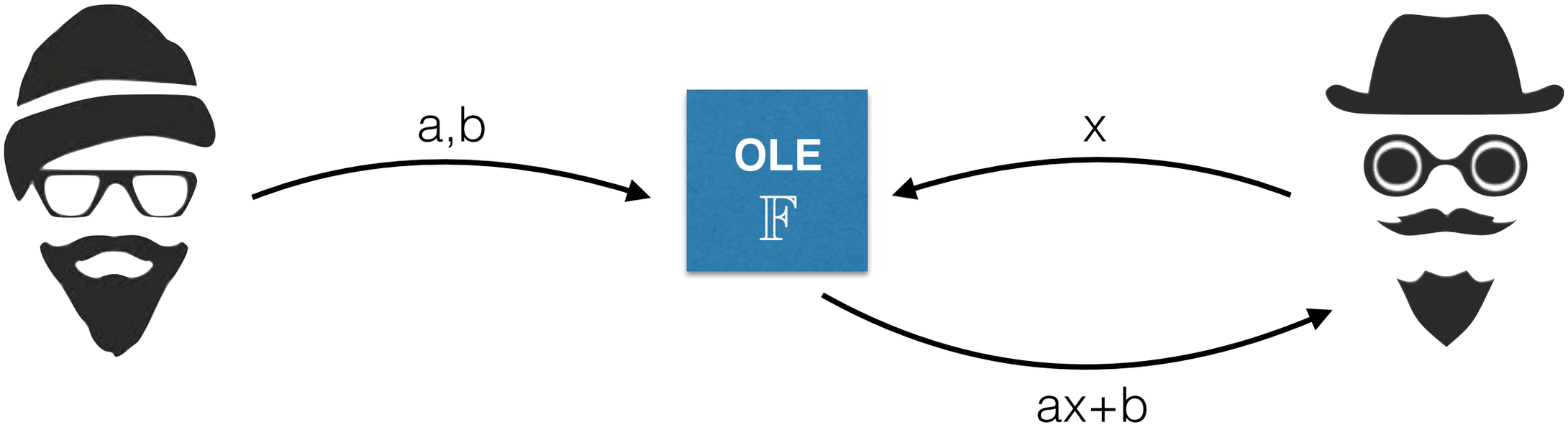
- GI99: multiparty linear correlations
- CDI05: generalization (pseudorandom Shamir)
- HIJKR16: arbitrary correlations, from iO
- B*C*GIO17: additive correlations, from HSS
- Scho18: relaxed form of additive correlations, key-homomorphic PRF

Previous Works

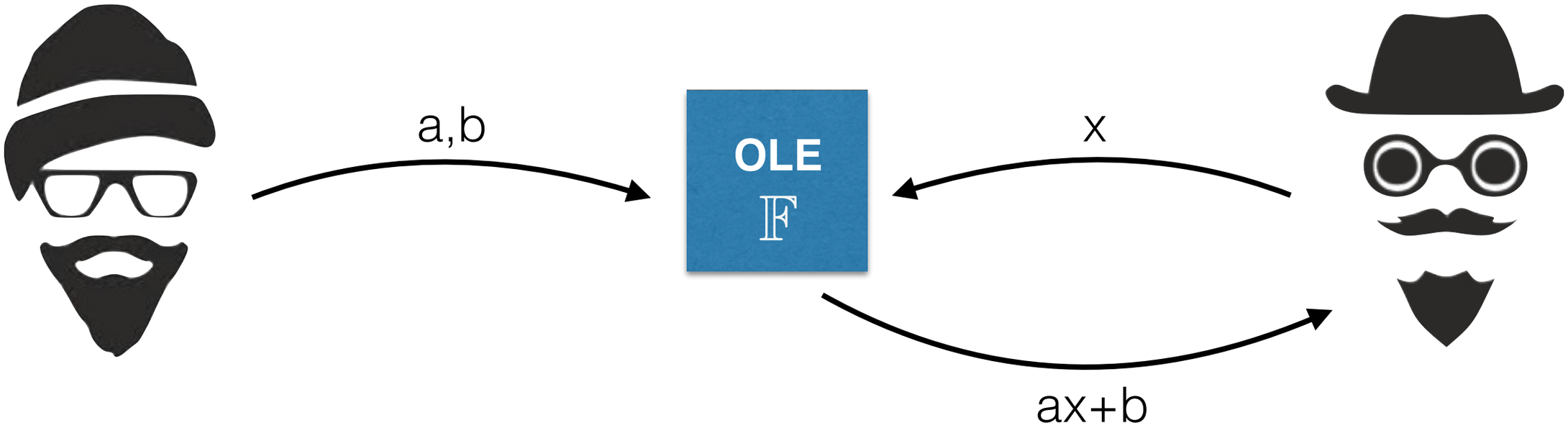
- GI99: multiparty linear correlations
- CDI05: generalization (pseudorandom Shamir)
- HIJKR16: arbitrary correlations, from iO
- B*C*GIO17: additive correlations, from HSS
- Scho18: relaxed form of additive correlations, key-homomorphic PRF

Non-linear, 2-party: no efficient solution

Oblivious Linear Evaluation

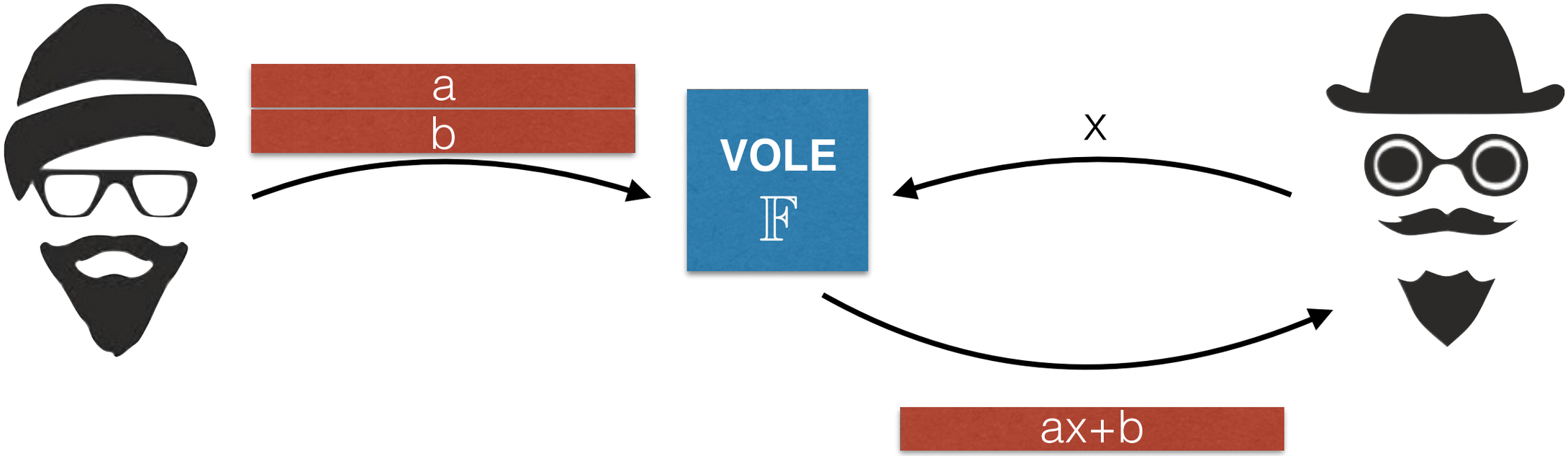


Oblivious Linear Evaluation

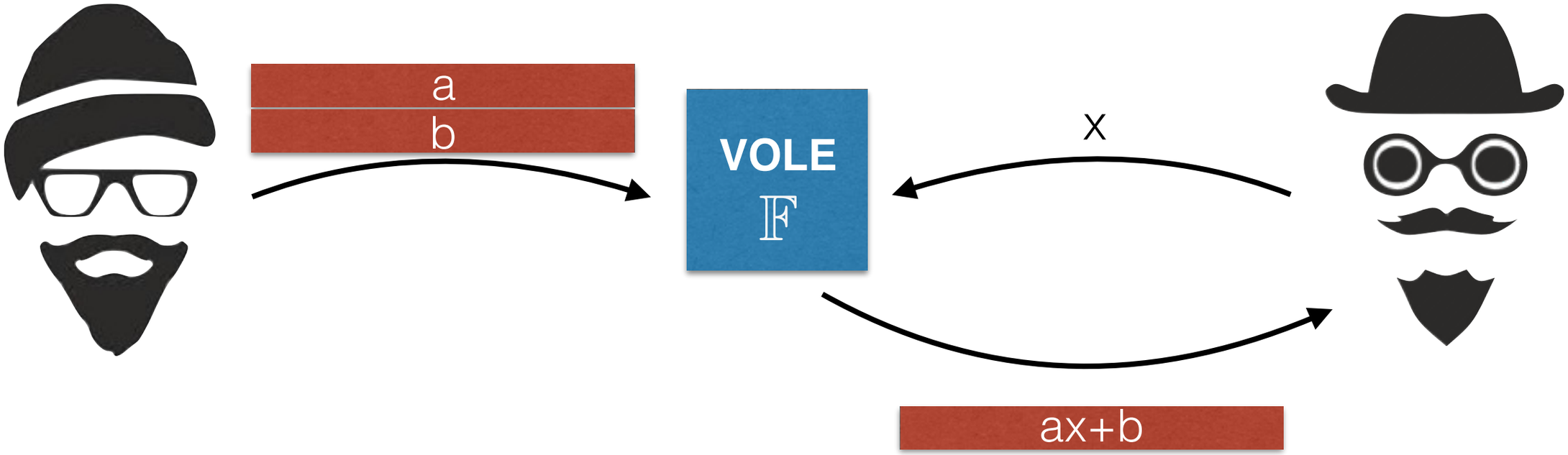


- Natural generalization of OT to large fields
- Enables secure computation of arithmetic circuits
- ‘Dream goal’ of correlation compression

Vector Oblivious Linear Evaluation

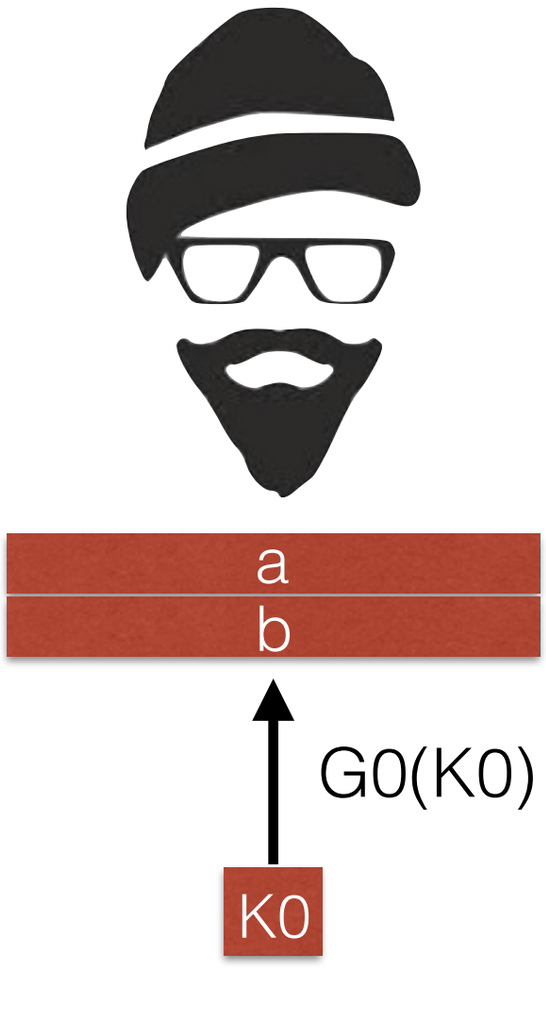


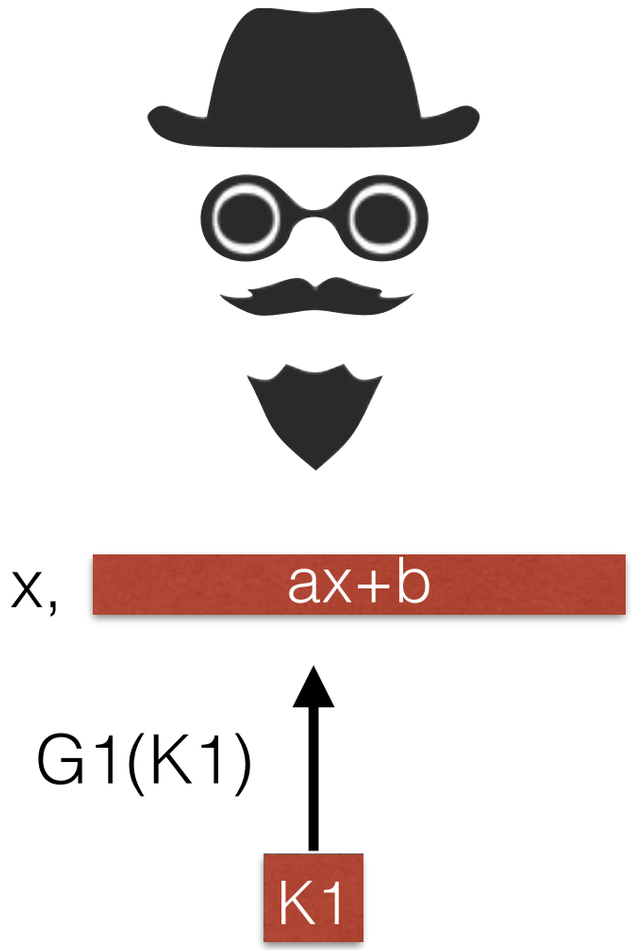
Vector Oblivious Linear Evaluation



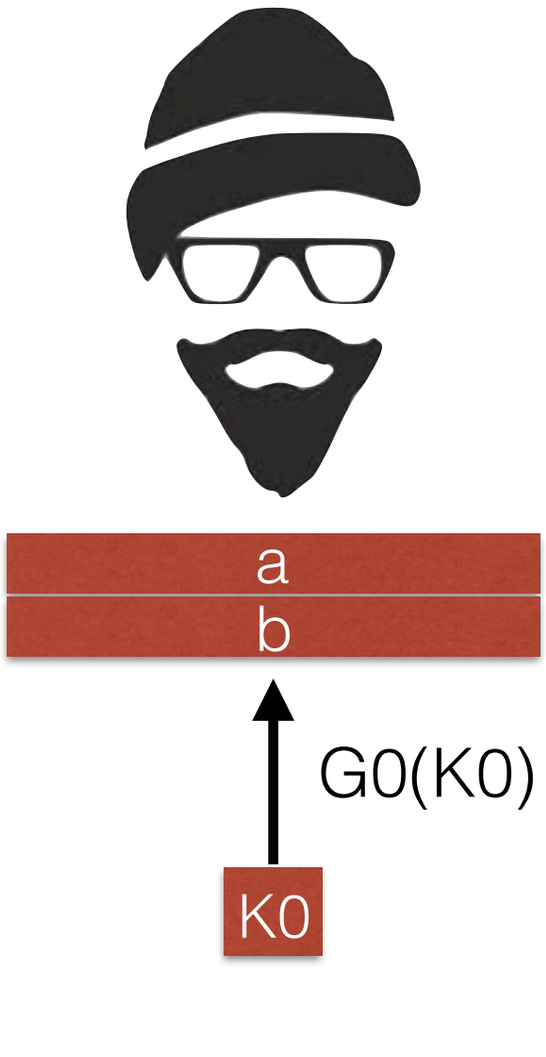
- Generalizes OLE the same way string-OT generalizes OT
- Still enjoys many cool applications for secure computation (more details later)

This Work: Efficient Compression for Vector-OLE Correlation





This Work: Efficient Compression for Vector-OLE Correlation

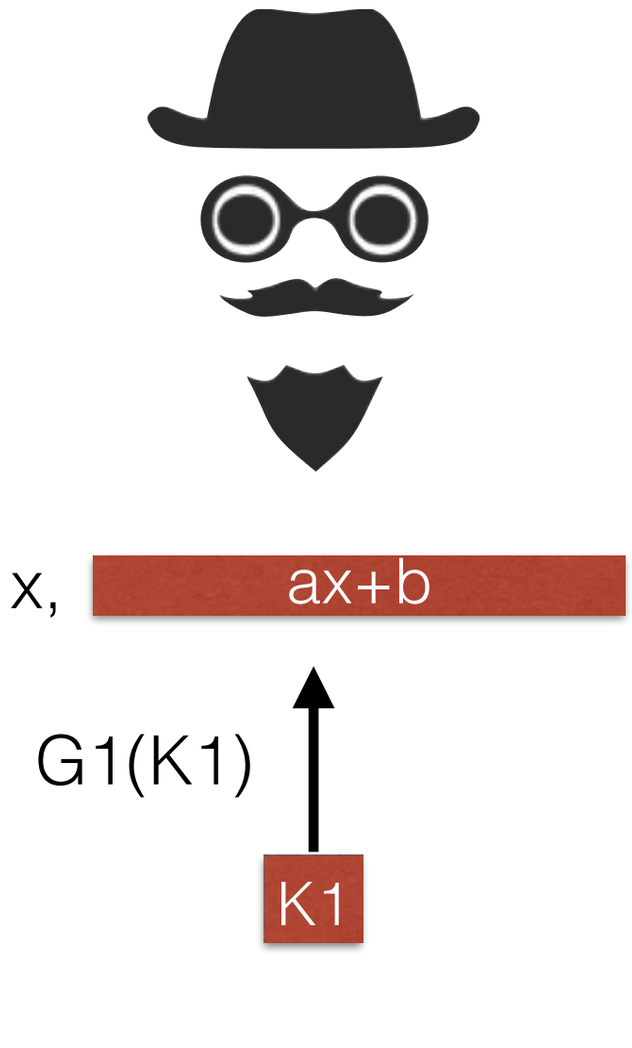


Security:

- Given K1, cannot distinguish (a,b) from random subject to G1(K1) = ax+b
- Given K0, learn nothing about x

Our Result:

An efficient* VOLE generator over any field under LPN



Applications

‘Direct’ applications:

- Rate-1/2 Vector-OLE (better rate from LPN would be a breakthrough)
- Large fan-out MPC
  - Matrix-vector multiplication
  - Nearest neighbor search

‘High-end’ applications:

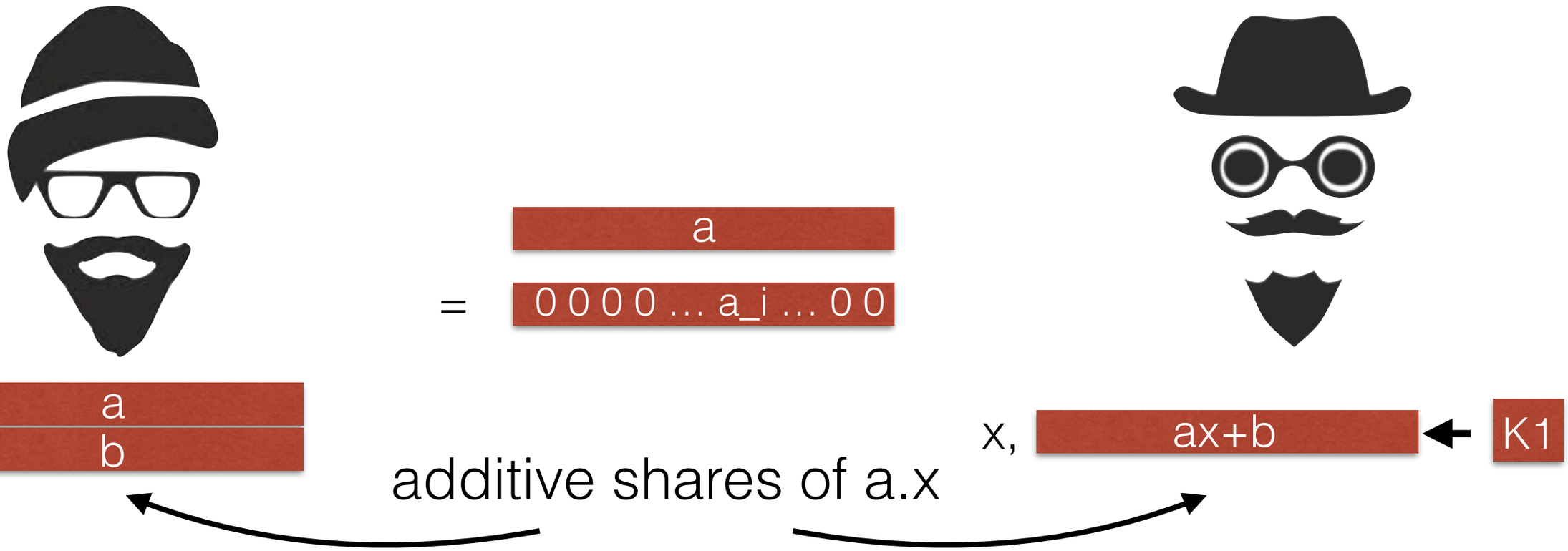
- Constant-depth arithmetic computation via arithmetic garbling
- Reusable non-interactive zero-knowledge in the preprocessing model

Our Method

1. *Sparse* VOLE-generator $\xrightarrow{\text{LPN}}$ pseudorandom VOLE-generator

Our Method

1. *Sparse* VOLE-generator $\xrightarrow{\text{LPN}}$ pseudorandom VOLE-generator

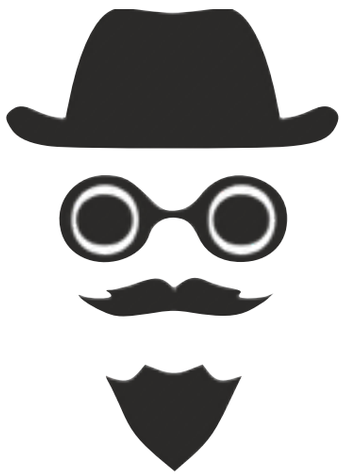


Our Method

1. *Sparse* VOLE-generator $\xrightarrow{\text{LPN}}$ pseudorandom VOLE-generator

a

= 0 0 0 0 ... a_i ... 0 0



K0 → a / b

x, ax+b → K1

additive shares of a.x

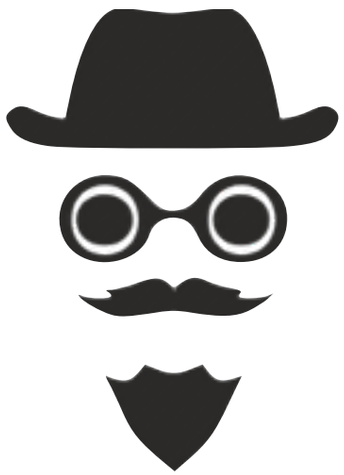
LPN: secret **M** + sparse error ~= random

Our Method

1. *Sparse* VOLE-generator $\xrightarrow{\text{LPN}}$ pseudorandom VOLE-generator

a

= 0 0 0 0 ... a_i ... 0 0



K0 → a / b

c d

additive shares of a.x

x, ax+b ← K1

cx+d

LPN: secret **M** + sparse error ~= random

Our Method

a’
b’
c
d

K0

c
M
+
a’

d
M
+
b’

K1

x,
a’x+b’
cx+d

cx+d
M
+
a’x+b’

Our Method

a’
b’
c
d

K0

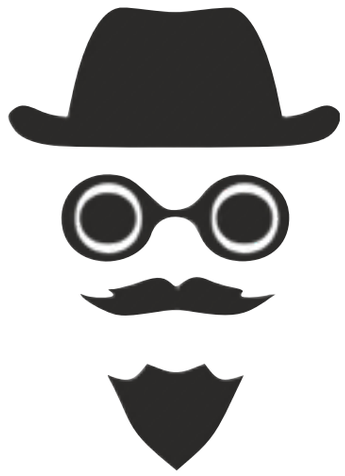
a

c
M
+
a’
d
M
+
b’

b

K1

x,
a’x+b’
cx+d



cx+d
M
+
a’x+b’

ax+b

Sparse Vector-OLE Generator

2. Multi-point function secret sharing $\Longrightarrow$ sparse vector-OLE generator

- Function Secret Sharing secretly shares a function between two parties

- FSS is hard in general

- For restricted functions (point functions), it can be done very efficiently from symmetric primitives [GI14, BGI15, BGI16]

- FSS for k-point functions is basically k-sparse VOLE-generator

Improvements

- Arbitrary stretch from dual-LPN
- Linear-time evaluation from batch codes
- Improvements from LPN-friendly linear codes
- Numerous heuristic optimizations

Improvements

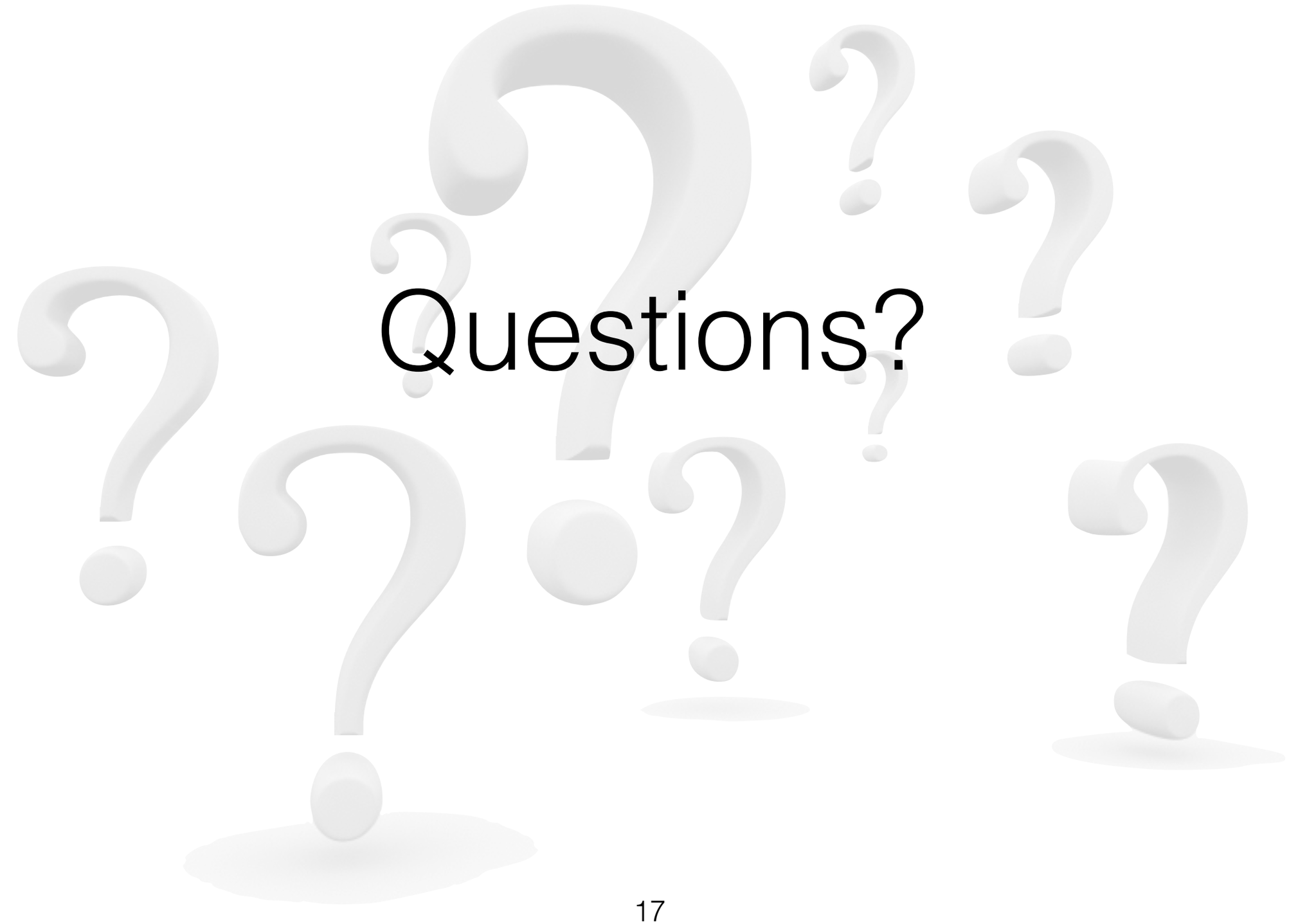
- Arbitrary stretch from dual-LPN
- Linear-time evaluation from batch codes
- Improvements from LPN-friendly linear codes
- Numerous heuristic optimizations

Efficiency (estimation): a million entries over a 128-bit field in 26 milliseconds (one core, standard laptop)

Conclusion

- Even for direct VOLE application, very competitive with alternative approaches
  - ADINZ17: ~5μs/OLE (asymptotic rate 1/3)
  - Our work: ~50ns/OLE (asymptotic rate 1/2)
- Distributed setup can be made very efficiently [Ds17]: for 1 million entries and 128-bit field, 69Kb and 34ms
- Unique communication pattern (local decompression)
- Large number of applications

Thank you for your attention



Questions?