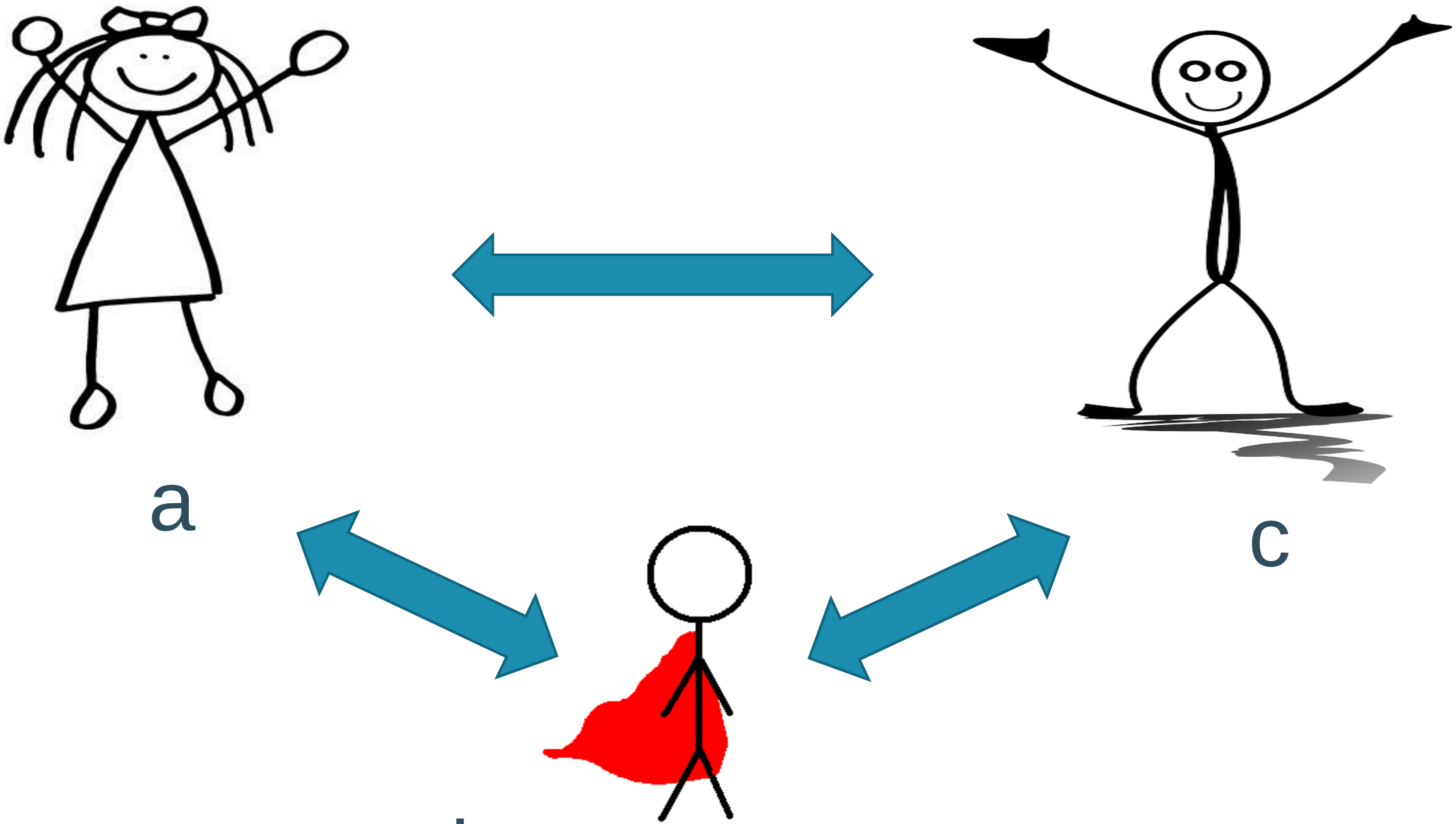
Eurocrypt 2018

Overdrive: Making SPDZ Great Again

Marcel Keller, Valerio Pastro, and **Dragos Rotaru**

University of Bristol, Yale University, KU Leuven

What's all the fuss about?



b

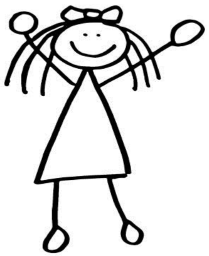
Goal: Compute F(a, b, c)

M. Keller, V. Pastro, **Dragos Rotaru**

imec-Cosic, Dept. Electrical Engineering

KU LEUVEN

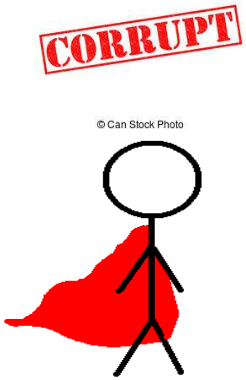
Security Model



- Many parties (up to N)
- Malicious adversary
- Dishonest majority of corrupted parties

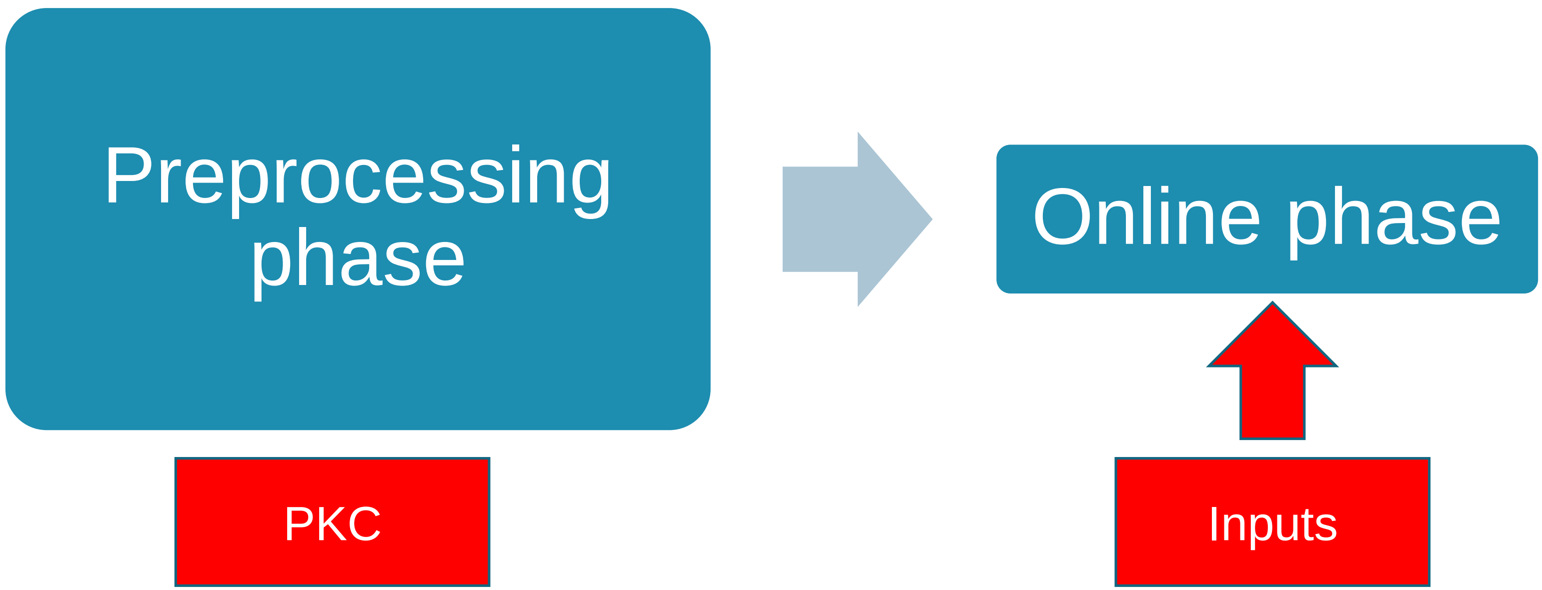
M. Keller, V. Pastro, **Dragos Rotaru**
imec-Cosic, Dept. Electrical Engineering
KU LEUVEN

Security Model



- Many parties (up to N)
- Malicious adversary
- Dishonest majority of corrupted parties

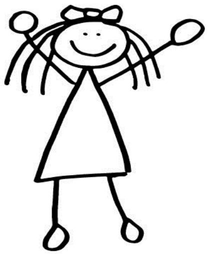
Malicious MPC protocols



SPDZ, TinyOT, BDOZa, MASCOT

M. Keller, V. Pastro, **Dragos Rotaru** imec-Cosic, Dept. Electrical Engineering KU LEUVEN

Secret share then authenticate

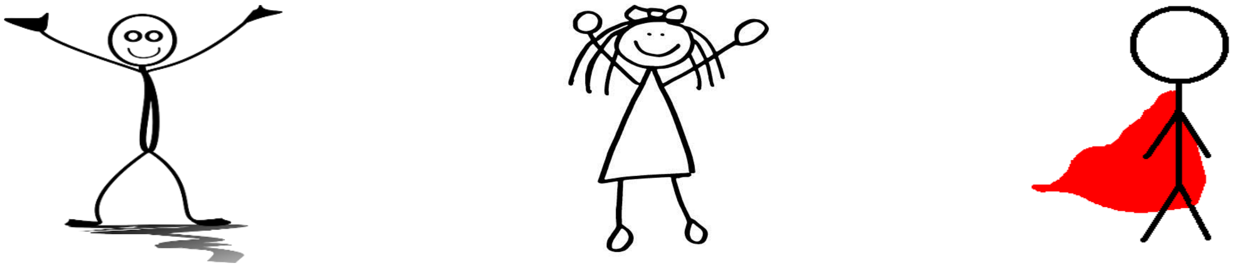


$$\alpha_1 + \alpha_2 + \alpha_3 = \alpha$$

$$x_1 + x_2 + x_3 = x$$

$$\gamma(x)_1 + \gamma(x)_2 + \gamma(x)_3 = \alpha x$$

Secret share then authenticate

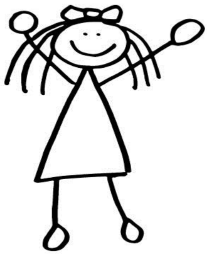


$$\alpha_1 + \alpha_2 + \alpha_3 = \alpha$$

$$(x+y)_1 + (x+y)_2 + (x+y)_3 = x+y$$

$$\gamma(x+y)_1 + \gamma(x+y)_2 + \gamma(x+y)_3 = \alpha(x+y)$$

Secret share then authenticate



But we want to multiply!

Let’s do it – what do we need?

M. Keller, V. Pastro, **Dragos Rotaru** imec-Cosic, Dept. Electrical Engineering
KU LEUVEN

Let’s do it – what do we need?

M. Keller, V. Pastro, **Dragos Rotaru** imec-Cosic, Dept. Electrical Engineering

Let’s do it – what do we need?

Let’s do it – what do we need?

What we have done

Fastest triple generation!

How to multiply shared inputs with triples (Beaver's Trick)

$$\begin{aligned} x \cdot y &= (x + a - a) \cdot (y + b - b) \\ &= (x + a) \cdot (y + b) - (y + b) \cdot a - (x + a) \cdot b + a \cdot b \end{aligned}$$

M. Keller, V. Pastro, **Dragos Rotaru** imec-Cosic, Dept. Electrical Engineering
KU LEUVEN

How to multiply shared inputs with triples (Beaver's Trick)

$$x \cdot y = (x + a - a) \cdot (y + b - b)$$

$$= (x + a) \cdot (y + b) - (y + b) \cdot a - (x + a) \cdot b + a \cdot b$$

Masked and opened

Random secret triple

M. Keller, V. Pastro, **Dragos Rotaru** imec-Cosic, Dept. Electrical Engineering

Revisit, improve, revisit…

BDOZa (BDOZ'11)

Semi-homomorphic encryption

SPDZ-1 (DPSZ'12)

Depth-1 SHE (NTL), ZK Proof

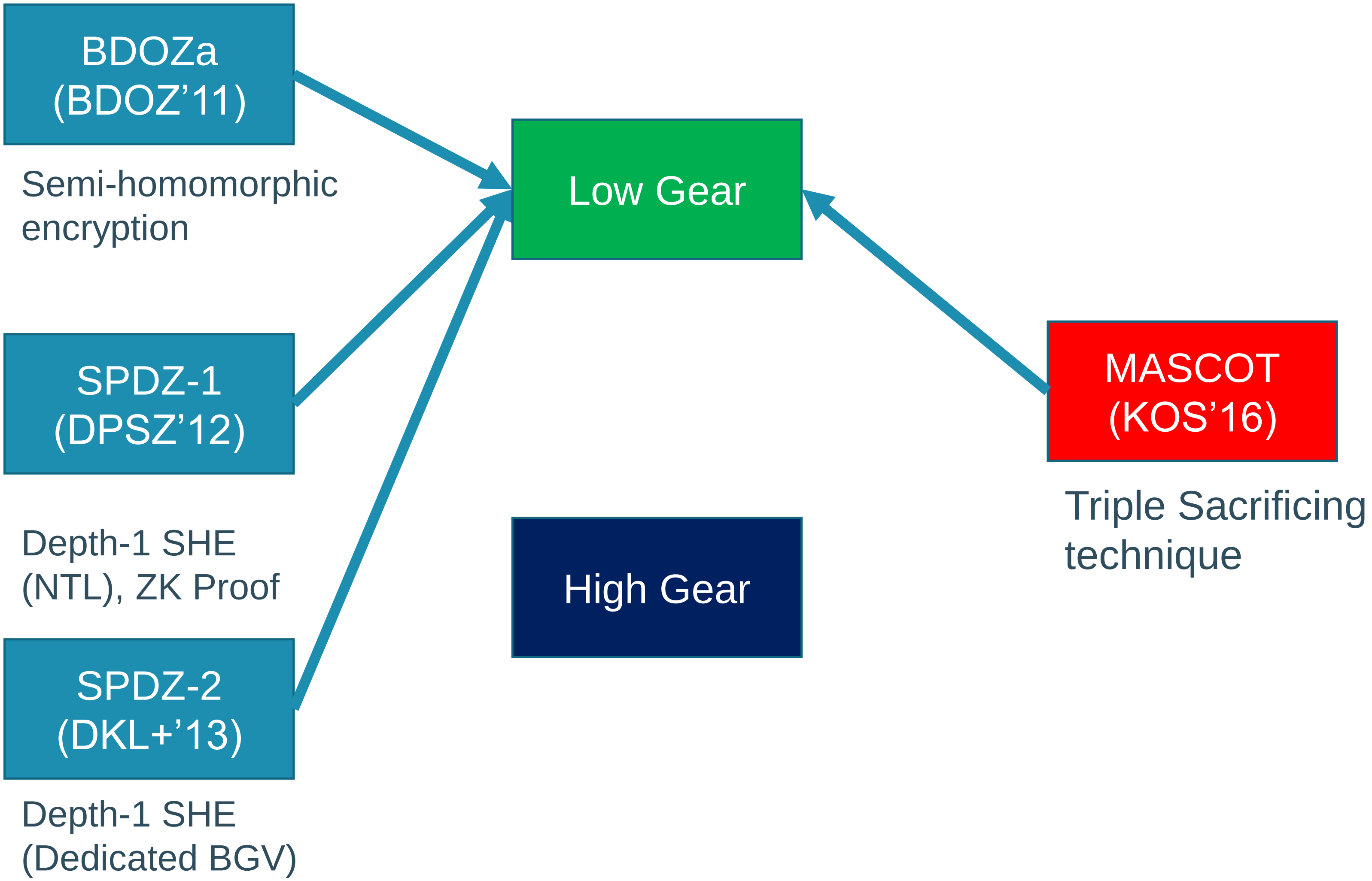
SPDZ-2 (DKL+'13)

Depth-1 SHE (Dedicated BGV)

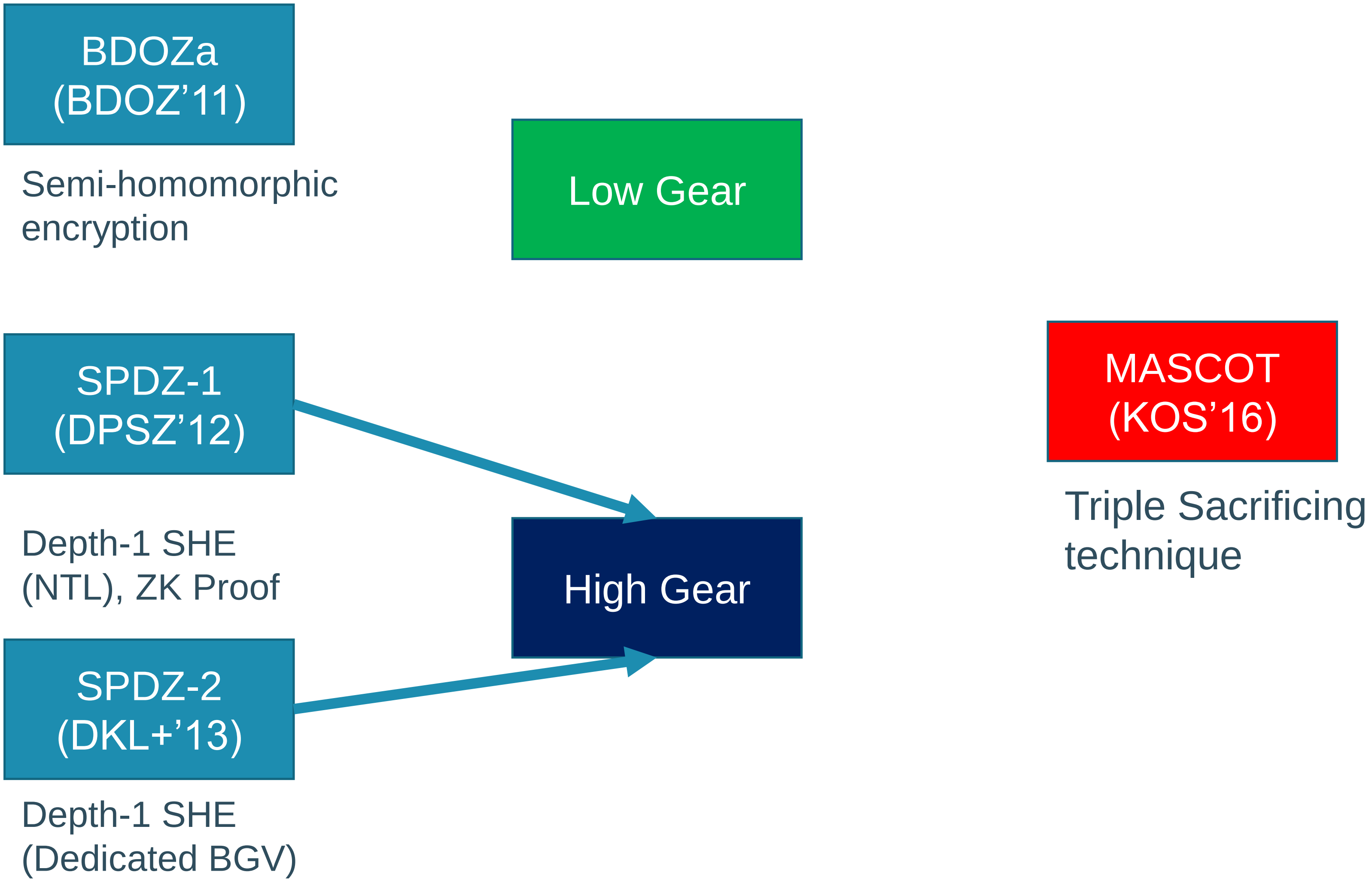
MASCOT (KOS'16)

Triple Sacrificing technique

Revisit, improve, revisit…



Revisit, improve, revisit…



 M. Keller, V. Pastro, **Dragos Rotaru** imec-Cosic, Dept. Electrical Engineering KU LEUVEN

LAN Timings

	Triples/s	Security	BGV impl.	$\log_2(\lvert\mathbb{F}_p\rvert)$
SPDZ-1 [DKL+12]	79	40-bit active	NTL	64
SPDZ-2 [DKL+13]	36	40-bit active	specific	64
SPDZ-1 (ours)	12,000	40-bit active	specific	64
MASCOT [KOS16]	5,100	64-bit active	$\perp$	128
Low Gear (Section 3)	30,000	64-bit active	specific	128

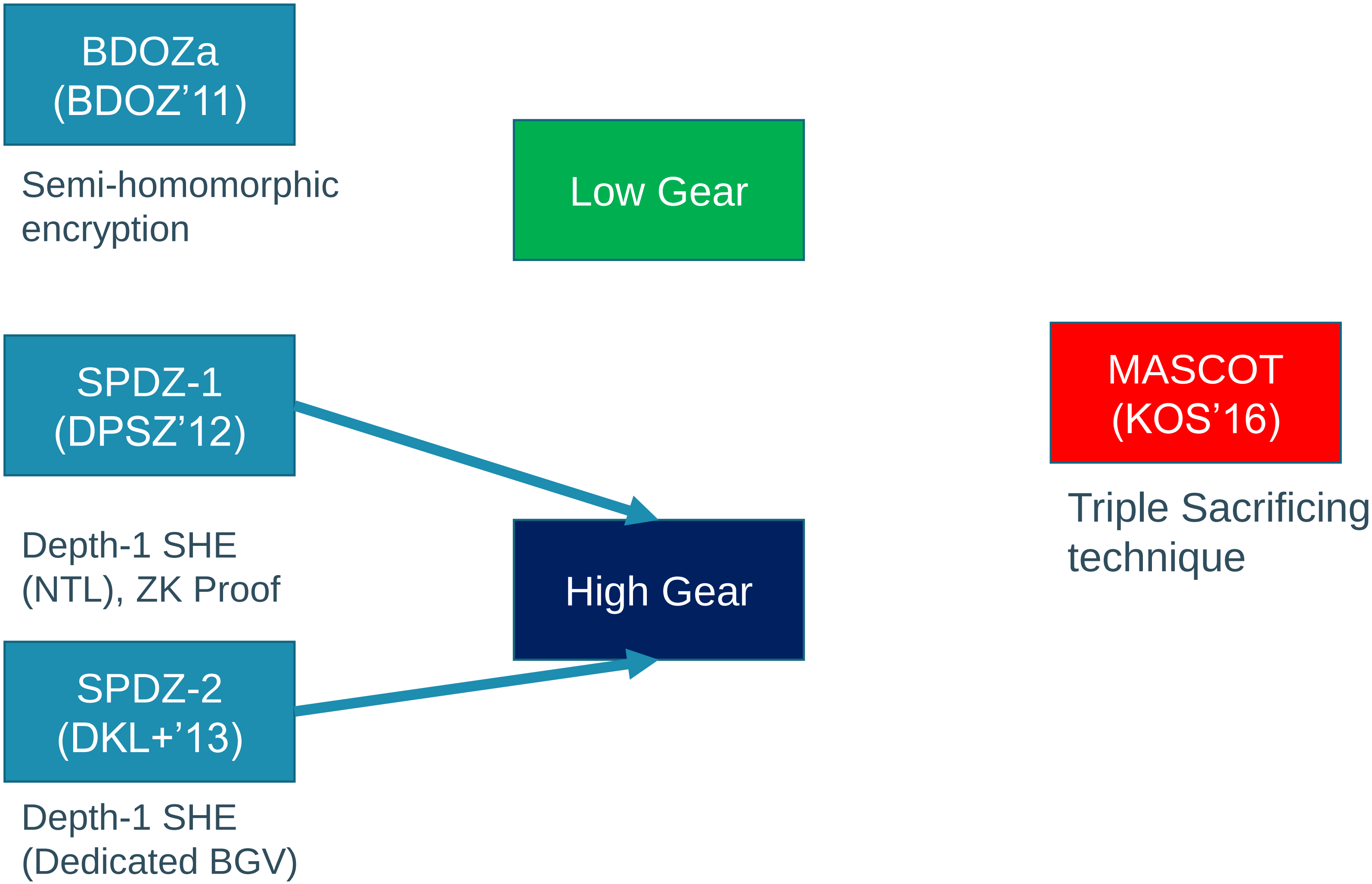
M. Keller, V. Pastro, **Dragos Rotaru** imec-Cosic, Dept. Electrical Engineering KU LEUVEN

LAN Timings

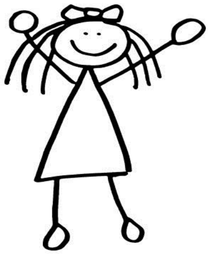
	Triples/s	Security	BGV impl.	$\log_2(\lvert\mathbb{F}_p\rvert)$
SPDZ-1 [DKL+12]	79	40-bit active	NTL	64
SPDZ-2 [DKL+13]	36	40-bit active	specific	64
SPDZ-1 (ours)	12,000	40-bit active	specific	64
MASCOT [KOS16]	5,100	64-bit active	$\perp$	128
Low Gear (Section 3)	30,000	64-bit active	specific	128

M. Keller, V. Pastro, **Dragos Rotaru** imec-Cosic, Dept. Electrical Engineering
KU LEUVEN

Revisit, improve, revisit…



SPDZ-1 recap



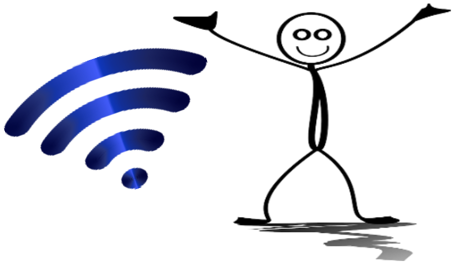
Enc(a[1])
Enc(b[1])

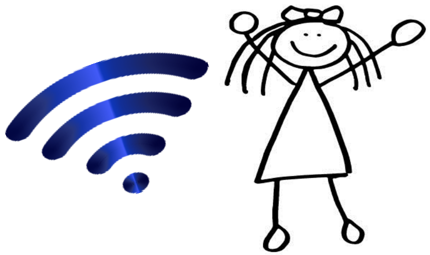
Enc(a[2])
Enc(b[2])

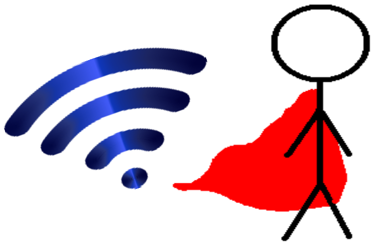
Enc(a[3])
Enc(b[3])

M. Keller, V. Pastro, **Dragos Rotaru** imec-Cosic, Dept. Electrical Engineering KU LEUVEN

SPDZ-1 recap







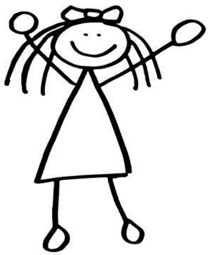
Enc(a[1])
Enc(b[1])

Enc(a[2])
Enc(b[2])

Enc(a[3])
Enc(b[3])

M. Keller, V. Pastro, **Dragos Rotaru** imec-Cosic, Dept. Electrical Engineering
KU LEUVEN

SPDZ-1 recap



Enc(a[1])
Enc(b[1])

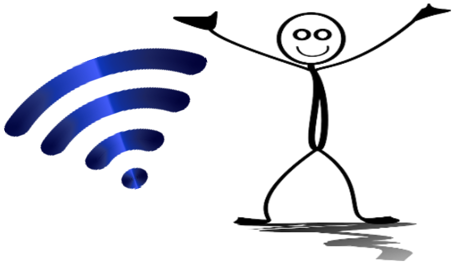
Enc(a[2])
Enc(b[2])

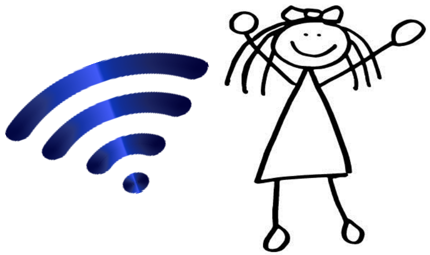
Enc(a[3])
Enc(b[3])

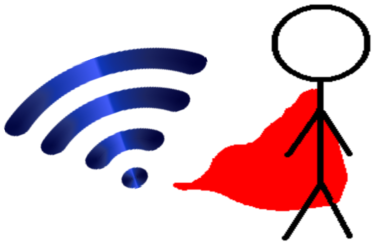
$$C = \left(\sum_i \mathsf{Enc}(a_i)\right) \boxdot \left(\sum_i \mathsf{Enc}(b_i)\right)$$

M. Keller, V. Pastro, **Dragos Rotaru** imec-Cosic, Dept. Electrical Engineering KU LEUVEN

SPDZ-1 recap







Enc(a[1])
Enc(b[1])

Enc(a[2])
Enc(b[2])

Enc(a[3])
Enc(b[3])

$$C = \left(\sum_i \mathsf{Enc}(a_i)\right) \boxdot \left(\sum_i \mathsf{Enc}(b_i)\right)$$

C[1] + C[2] + C[3] = C

- Parties may lie about their plaintext - incorrect decryption, reveal info about secret keys.
- Need to add ZK proofs for bounding the plaintext

M. Keller, V. Pastro, **Dragos Rotaru**
imec-Cosic, Dept. Electrical Engineering
KU LEUVEN

How to 0-knowledge

I know my eX!

Not sure. Let's verify!

Commitment f'(r)

Challenge: E

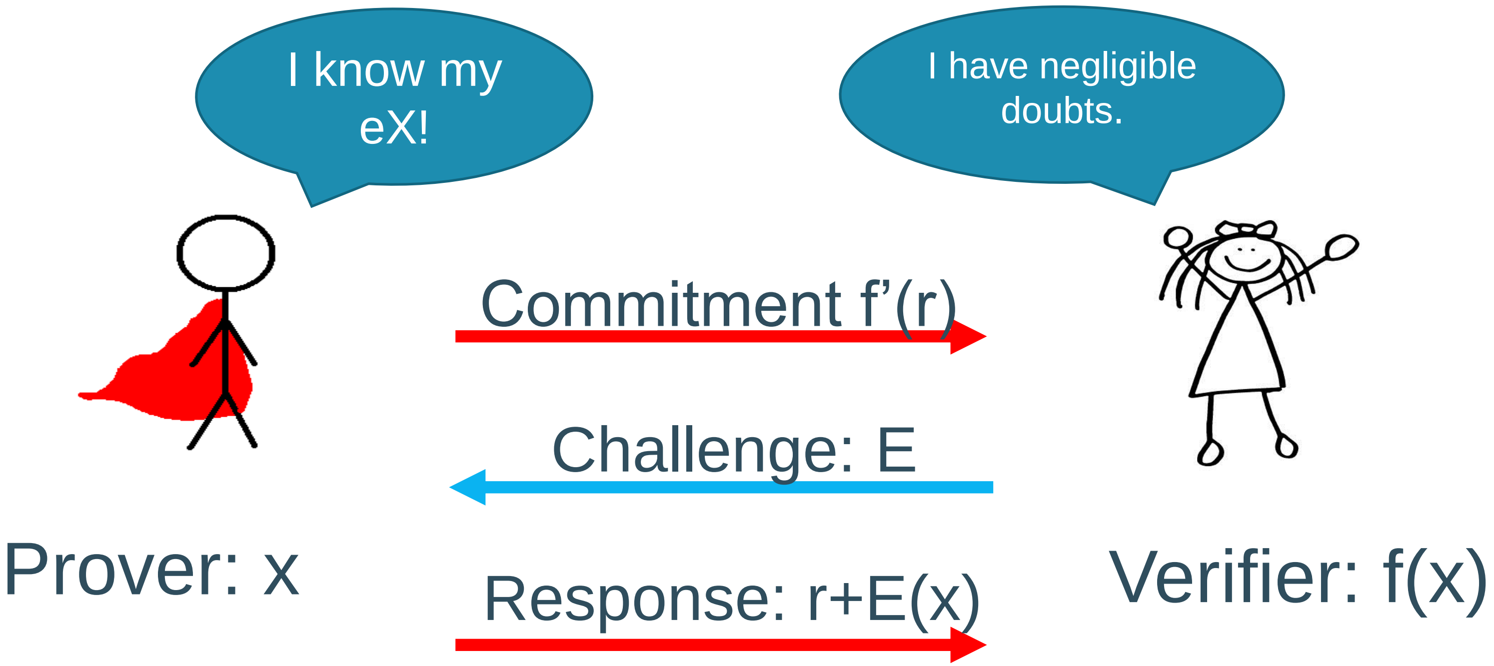
Response: r+E(x)

Prover: x

Verifier: f(x)

M. Keller, V. Pastro, **Dragos Rotaru** imec-Cosic, Dept. Electrical Engineering KU LEUVEN

How to 0-knowledge



- f'(r+E(x)) = f'(r)+E(f(x))
- r+E(x) is bounded
- r >> x, r/x is called slack

M. Keller, V. Pastro, **Dragos Rotaru**
imec-Cosic, Dept. Electrical Engineering

To slack or not to slack

- ZKPoPk: to prove that x < B we need an encryption scheme which supports plaintexts < B * slack

Slack is:
- ~2^50 for 40-bit security
- ~2^100 for 128-bit security

M. Keller, V. Pastro, **Dragos Rotaru** imec-Cosic, Dept. Electrical Engineering KU LEUVEN

To slack or not to slack

- ZKPoPk: to prove that x < B we need an encryption scheme which supports plaintexts < B * slack

Slack is:

- ~2^50 for 40-bit security
- ~2^100 for 128-bit security

- Improve the ZK slack analysis.
- With depth-1 BGV the slack becomes tiny tiny because of the modulus switching.

M. Keller, V. Pastro, **Dragos Rotaru** imec-Cosic, Dept. Electrical Engineering KU LEUVEN

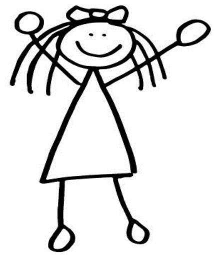
Some ciphertexts need no slack

SPDZ			sec	$\log(\lvert\mathbb{F}_p\rvert)$
1 [DPSZ12]	1 [CDXY17]	2 [DKL+13]		
330	330	332	40	64
572	572	N/A	64	128

Table 1. Ciphertext modulus bit length ($\log(q)$) for two parties.

M. Keller, V. Pastro, **Dragos Rotaru** imec-Cosic, Dept. Electrical Engineering

High Gear: SPDZ-1 with global proof



V(P(Alice)) V(P(Bob))	V(P(Bob)) V(P(Charlie)	V(P(Alice)) V(P(Charlie))
V(P(Alice) +P(Bob))	V(P(Bob)+P(Charlie))	+ V(P(Alice)+P(Charlie))

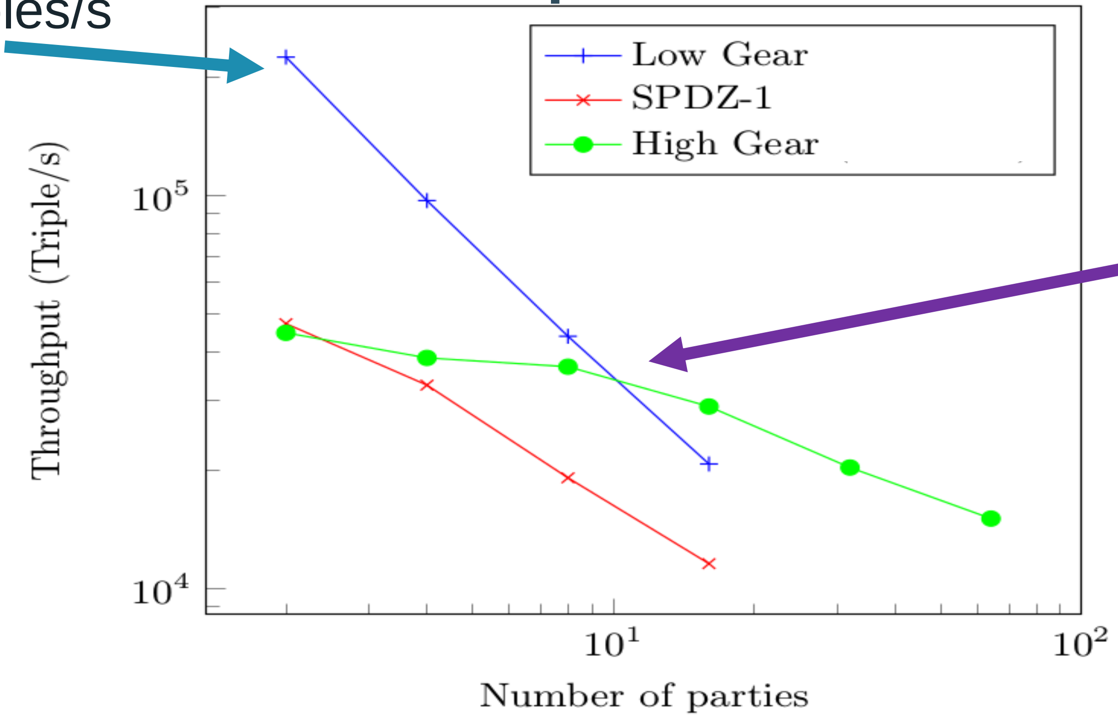
High Gear: SPDZ-1 with global proof

	Triples/s	Security	$\log_2(\lvert\mathbb{F}_p\rvert)$
SPDZ-1 (ours)	6,400	64-bit active	128
High Gear (Section 4)	5,600	64-bit active	128

M. Keller, V. Pastro, **Dragos Rotaru**
imec-Cosic, Dept. Electrical Engineering
KU LEUVEN

Low Gear vs High Gear, the tipping point

224k Triples/s



6 parties

Fig. 13. Triple generation for a 128 bit prime field with 64 bit statistical security on AWS r4.16xlarge instances.

64 CPUs, 488Gb RAM, 25Gb Network

M. Keller, V. Pastro, **Dragos Rotaru** imec-Cosic, Dept. Electrical Engineering
KU LEUVEN

100 party Vickrey Auction

AWS instance	Time	Cost per party
t2.nano	9.0 seconds	$0.000017
c4.8xlarge	1.4 seconds	$0.000741

Table 6. Online phase of Vickrey auction with 100 parties, each inputting one bid.

	Time	Cost per party
MASCOT [KOS16]	1,300 seconds	$0.190
High Gear (Section 4)	98 seconds	$0.014

Table 7. Offline phase of Vickrey auction with 100 parties, each inputting one bid.

AWS m3.2xlarge

8 CPUs, 30Gb RAM, 10Gb Network

Code lives on the internetz

https://github.com/bristolcrypto/SPDZ-2

Open problem alert:

- In the Low Gear protocol we assumed semi-homomorphic BGV is a linear only encryption scheme.
- Can you create ciphertexts which decrypt to non-linear plaintexts without the KS info? Known as linear target malleability [BCI+13] or linear only encryption [BISW17].

Thank you!

Thank you!

- Questions?

Tiny advert: SCALE at TPMPC

- SCALE (Secure Computation Algorithms from Leuven)
- We do a better analysis of the ZK proofs involved.
- Pre-processing phase coupled with the online phase.
- Compiler is documented, people can read how to use it.
- Others bells and whistles.

M. Keller, V. Pastro, **Dragos Rotaru**
imec-Cosic, Dept. Electrical Engineering
KU LEUVEN