Stochastic Secret Sharing with 1-Bit Shares and Applications to MPC

Benny Applebaum **Eliran Kachlon**

TEL AVIV UNIVERSITY

Sharing a Bit by Bits

astic Secret Sharing
it Shares
and Ap MPC

Benny Applebaum

TEL AVIV UNIVERSITY

Combiners

Sharing a Bit by Bits

Sharing a Bit by Bits

Combiners

PIR

Sharing a Bit by Bits

Combiners

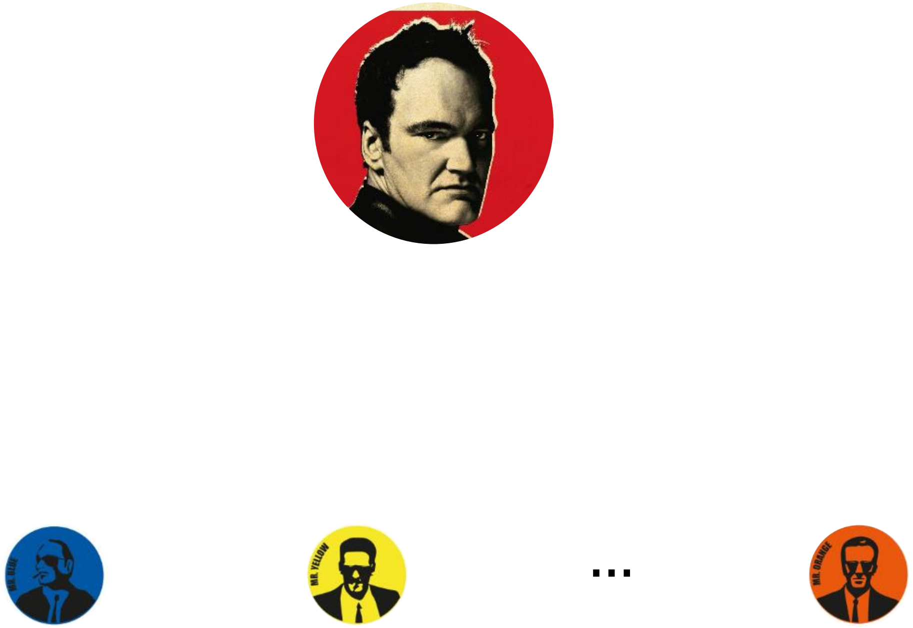
Secure computation of **Boolean** circuits

PIR

Dream Version: "Shamir over $\mathbb{F}_2$"

Dream Version: "Shamir over $\mathbb{F}_2$"

Dream Version: “Shamir over $\mathbb{F}_2$”

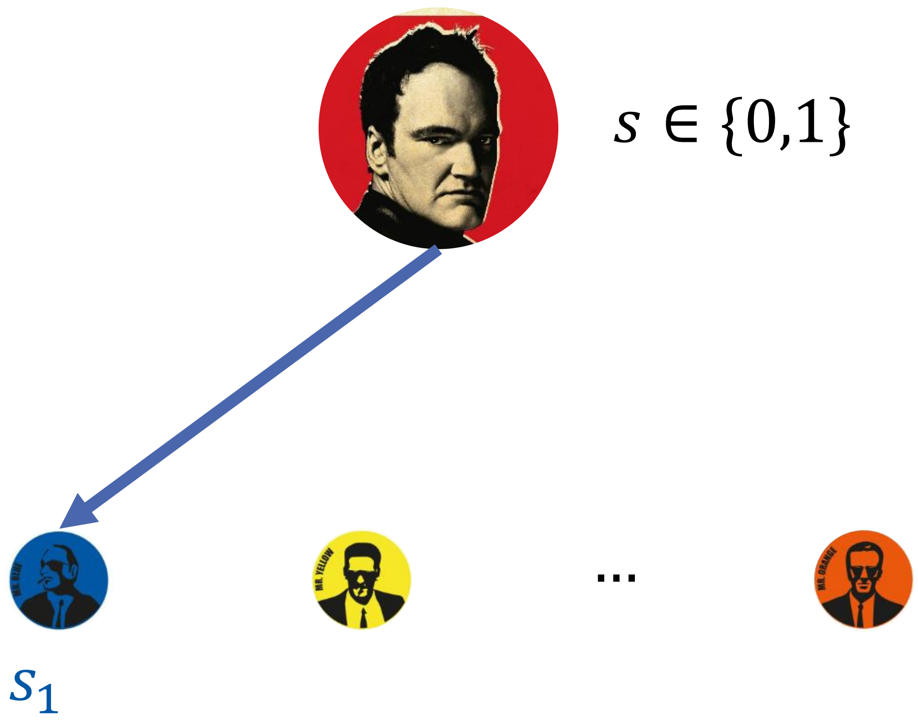


Dream Version: “Shamir over $\mathbb{F}_2$”

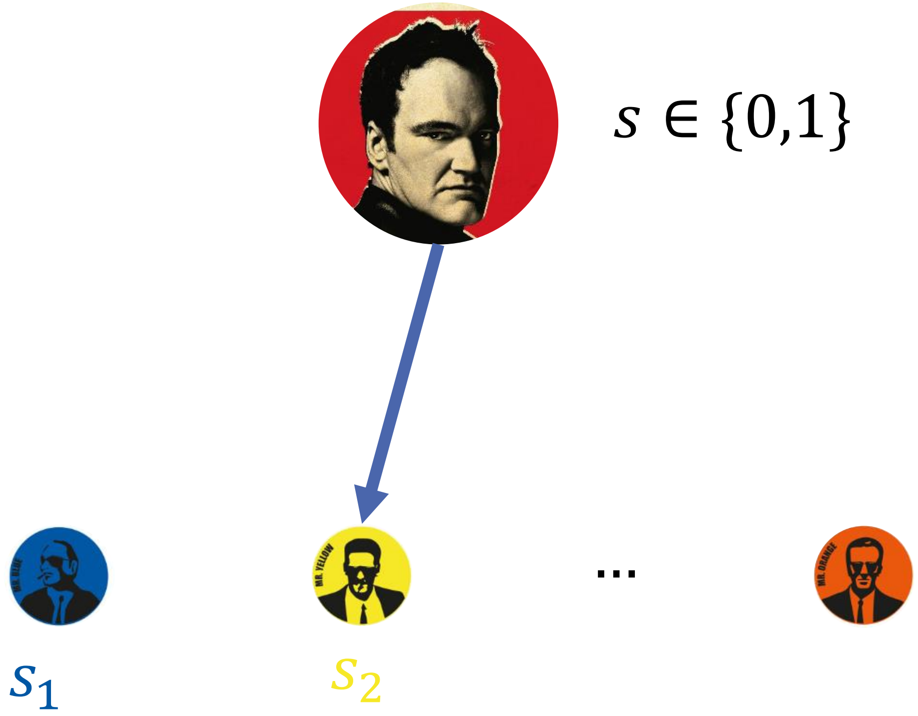
$s \in \{0,1\}$

…

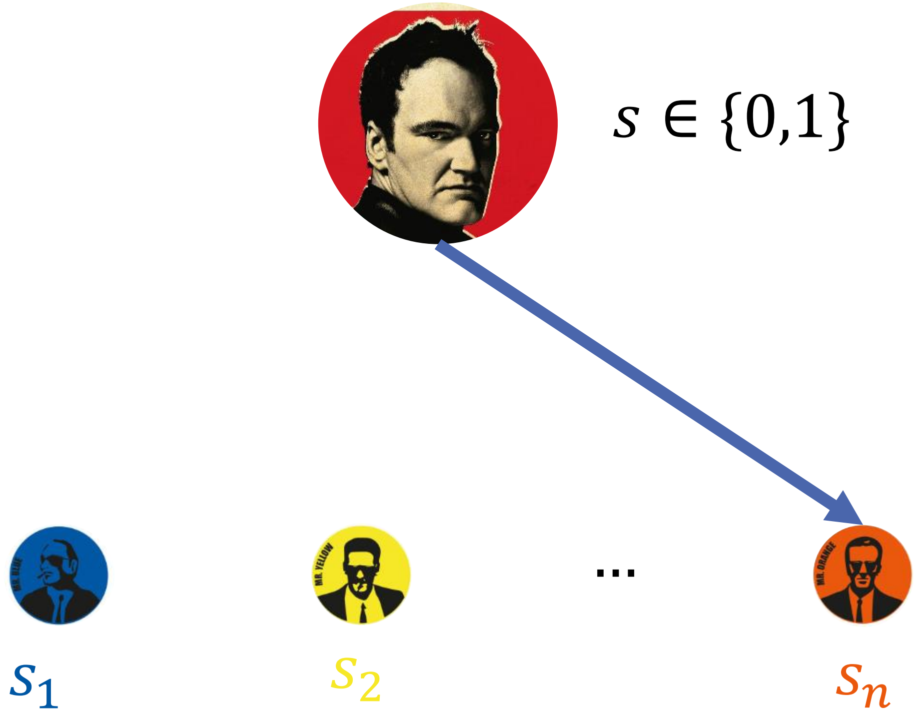
Dream Version: "Shamir over $\mathbb{F}_2$"



Dream Version: “Shamir over $\mathbb{F}_2$”



Dream Version: "Shamir over $\mathbb{F}_2$"



Dream Version: “Shamir over $\mathbb{F}_2$”

$s \in \{0,1\}$

1-bit shares!

s_1

s_2

…

s_n

Dream Version: “Shamir over $\mathbb{F}_2$”

- t-privacy

$s \in \{0,1\}$

1-bit shares!

...

s_1 s_2 s_n

Dream Version: “Shamir over $\mathbb{F}_2$”

- t-privacy
- $(t+1)$-correctness

$s \in \{0,1\}$

1-bit shares!

s_1

s_2

…

s_n

Dream Version: "Shamir over $\mathbb{F}_2$"

- t-privacy
- $(t+1)$-correctness
- Linear over $\mathbb{F}_2$

$s \in \{0,1\}$

1-bit shares!

s_1

s_2

…

s_n

Dream Version: “Shamir over $\mathbb{F}_2$”

- t-privacy
- $(t+1)$-correctness
- Linear over $\mathbb{F}_2$

$s \in \{0,1\}$

1-bit shares!

$s \Rightarrow$

s_1

s_2

…

s_n

Dream Version: “Shamir over $\mathbb{F}_2$”

- t-privacy
- $(t+1)$-correctness
- Linear over $\mathbb{F}_2$

$s \in \{0,1\}$

1-bit shares!

  … 

$s \Rightarrow \quad s_1 \quad s_2 \quad \cdots \quad s_n$

$s' \Rightarrow \quad s_1' \quad s_2' \quad \cdots \quad s_n'$

Dream Version: "Shamir over $\mathbb{F}_2$"

- t-privacy
- $(t+1)$-correctness
- Linear over $\mathbb{F}_2$

$s \in \{0,1\}$

1-bit shares!

  … 

$s \Rightarrow$	s_1	s_2		s_n
$s' \Rightarrow$	s_1'	s_2'		s_n'
$s + s' \Rightarrow$	$s_1 + s_1'$	$s_2 + s_2'$		$s_n + s_n'$

Dream Version: "Shamir over $\mathbb{F}_2$"

- t-privacy
- $(t + 1)$-correctness
- Linear over $\mathbb{F}_2$
- Multiplicative for $t < n/2$

$s \in \{0,1\}$

1-bit shares!

s_1

s_2

…

s_n

Dream Version: “Shamir over $\mathbb{F}_2$”

- t-privacy
- $(t+1)$-correctness
- Linear over $\mathbb{F}_2$
- Multiplicative for $t < n/2$

$s \in \{0,1\}$

1-bit shares!

…

$s \Rightarrow$ s_1 s_2 s_n

Dream Version: “Shamir over $\mathbb{F}_2$”

- t-privacy
- $(t+1)$-correctness
- Linear over $\mathbb{F}_2$
- Multiplicative for $t < n/2$

$s \in \{0,1\}$

1-bit shares!

…

$s \Rightarrow \quad s_1 \quad s_2 \quad \cdots \quad s_n$

$s' \Rightarrow \quad s_1' \quad s_2' \quad \cdots \quad s_n'$

Dream Version: "Shamir over $\mathbb{F}_2$"

- t-privacy
- $(t+1)$-correctness
- Linear over $\mathbb{F}_2$
- Multiplicative for $t < n/2$

$s \in \{0,1\}$

1-bit shares!

…

$s \Rightarrow$	s_1	s_2		s_n
$s' \Rightarrow$	s'_1	s'_2		s'_n
$s \cdot s' \Leftarrow$	$s_1 \cdot s'_1$	$s_2 \cdot s'_2$		$s_n \cdot s'_n$

Dream Version: “Shamir over $\mathbb{F}_2$”

- t-privacy
- $(t+1)$-correctness
- Linear over $\mathbb{F}_2$
- Multiplicative for $t < n/2$

$s \in \{0,1\}$

1-bit shares!

s_1

s_2

…

s_n

Dream Version: "Shamir over $\mathbb{F}_2$"

- t-privacy
- $(t+1)$-correctness
- Linear over $\mathbb{F}_2$
- Multiplicative for $t < n/2$

$s \in \{0,1\}$

1-bit shares!

s_1

s_2

…

s_n

Problem: In Shamir we have $|\mathbb{F}| > n$

Dream Version: "Shamir over $\mathbb{F}_2$"

- t-privacy
- $(t+1)$-correctness
- Linear over $\mathbb{F}_2$
- Multiplicative for $t < n/2$

$s \in \{0,1\}$

1-bit shares!

s_1

s_2

…

s_n

Problem: In Shamir we have $|\mathbb{F}| > n$

$\Rightarrow \log(n)$-bit shares even for 1-bit secrets!

Dream Version: "Shamir over $\mathbb{F}_2$"

- t-privacy
- $(t+1)$-correctness
- Linear over $\mathbb{F}_2$
- Multiplicative for $t < n/2$

$s \in \{0,1\}$

1-bit shares!

s_1 s_2 … s_n

Problem: In Shamir we have $|\mathbb{F}| > n$

$\Rightarrow \log(n)$-bit shares even for 1-bit secrets!

$\Omega(\log(n))$ is necessary for threshold secret sharing [CDN15]

Ramp Secret Sharing

Ramp Secret Sharing

- t-privacy

Ramp Secret Sharing

- t-privacy
- $(t + \epsilon n)$-correctness

Ramp Secret Sharing

- t-privacy
- $(t + \epsilon n)$-correctness

AG codes [CC06]
Random linear codes [CCGHV07]

Ramp Secret Sharing

- t-privacy
- $(t + \epsilon n)$-correctness
- Linear and multiplicative

AG codes [CC06]
Random linear codes [CCGHV07]

Ramp Secret Sharing

- t-privacy
- $(t + \epsilon n)$-correctness

AG codes [CC06]
Random linear codes [CCGHV07]

- Linear and multiplicative
- Share size = $O(\log(1/\epsilon))$ bits!

Ramp Secret Sharing

- t-privacy
- $(t + \epsilon n)$-correctness

AG codes [CC06]
Random linear codes [CCGHV07]

- Linear and multiplicative
- Share size = $O(\log(1/\epsilon))$ bits!

Ramp Secret Sharing

- t-privacy
- $(t + \epsilon n)$-correctness

AG codes [CC06]
Random linear codes [CCGHV07]

- Linear and multiplicative
- Share size = $O(\log(1/\epsilon))$ bits!

- **Problem:** 1-bit shares require large gap $\epsilon \geq 1/3$ [BGK20]

Ramp Secret Sharing

- t-privacy
- $(t + \epsilon n)$-correctness
- Linear and multiplicative
- Share size = $O(\log(1/\epsilon))$ bits!

AG codes [CC06]
Random linear codes [CCGHV07]

- **Problem:** 1-bit shares require large gap $\epsilon \geq 1/3$ [BGK20]
  - E.g. t-privacy vs. $\left(t + \frac{n}{3}\right)$-correctness

Ramp Secret Sharing

- t-privacy
- $(t + \epsilon n)$-correctness

AG codes [CC06]
Random linear codes [CCGHV07]

- Linear and multiplicative
- Share size = $O(\log(1/\epsilon))$ bits!

- **Problem:** 1-bit shares require large gap $\epsilon \geq 1/3$ [BGK20]
  - E.g. t-privacy vs. $\left(t + \frac{n}{3}\right)$-correctness
  - Worse in existing constructions

1-Bit Shares?

1-Bit Shares?

- Meaningful model for 1-bit shares?

1-Bit Shares?

- Meaningful model for 1-bit shares?
  - Shamir-like features?

1-Bit Shares?

- Meaningful model for 1-bit shares?
  - Shamir-like features?
  - Useful for MPC?

1-Bit Shares?

- Meaningful model for 1-bit shares?
  - Shamir-like features?
  - Useful for MPC?

Yes!

1-Bit Shares?

- Meaningful model for 1-bit shares?
  - Shamir-like features?
  - Useful for MPC?
- Stochastic Secret Sharing

Yes!

1-Bit Shares?

- Meaningful model for 1-bit shares?
  - Shamir-like features?
  - Useful for MPC?
- Stochastic Secret Sharing
  - Probabilistic corruption

Yes!

1-Bit Shares?

- Meaningful model for 1-bit shares?
  - Shamir-like features?
  - Useful for MPC?
- Stochastic Secret Sharing
  - Probabilistic corruption
- Static Secret Sharing

Yes!

1-Bit Shares?

- Meaningful model for 1-bit shares?
  - Shamir-like features?
  - Useful for MPC?
- Stochastic Secret Sharing
  - Probabilistic corruption
- Static Secret Sharing
  - Worst-case corruptions

Yes!

1-Bit Shares?

- Meaningful model for 1-bit shares?
  - Shamir-like features?
  - Useful for MPC?
- Stochastic Secret Sharing
  - Probabilistic corruption
- Static Secret Sharing
  - Worst-case corruptions
- Applications to MPC

Yes!

Stochastic Secret Sharing

Stochastic Secret Sharing

- Stochastic corruption model

Stochastic Secret Sharing

- Stochastic corruption model
  - Every party is corrupted with probability p

Stochastic Secret Sharing

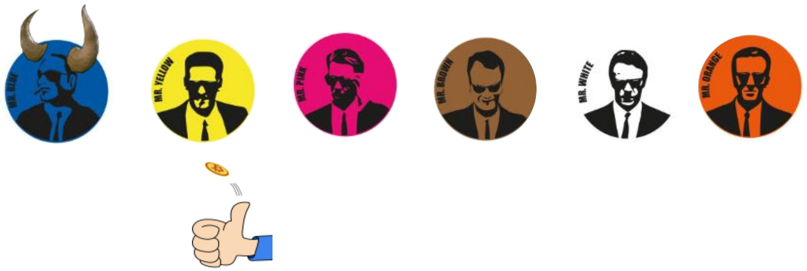
- Stochastic corruption model
  - Every party is corrupted with probability p

Stochastic Secret Sharing

- Stochastic corruption model
  - Every party is corrupted with probability p

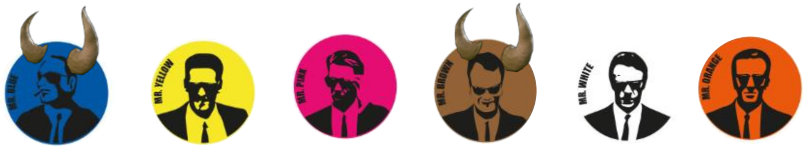
Stochastic Secret Sharing

- Stochastic corruption model
  - Every party is corrupted with probability p



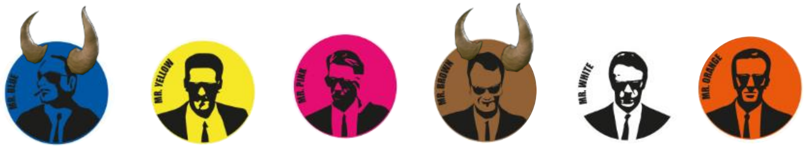
Stochastic Secret Sharing

- Stochastic corruption model
  - Every party is corrupted with probability p



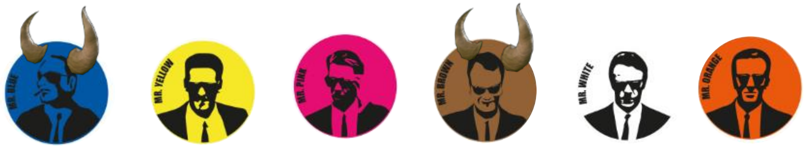
Stochastic Secret Sharing

- Stochastic corruption model
  - Every party is corrupted with probability p
  - Natural in some areas in cryptography



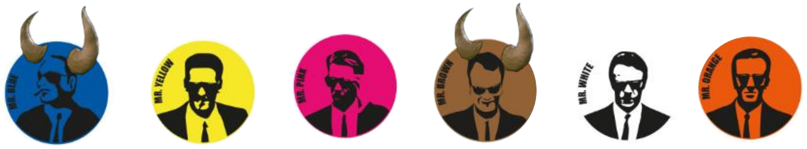
Stochastic Secret Sharing

- Stochastic corruption model
  - Every party is corrupted with probability p
  - Natural in some areas in cryptography
    - Leakage resilient circuits in the random probing model



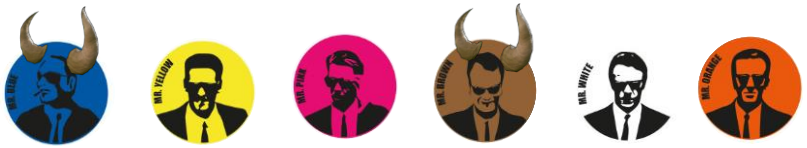
Stochastic Secret Sharing

- Stochastic corruption model
  - Every party is corrupted with probability p
  - Natural in some areas in cryptography
    - Leakage resilient circuits in the random probing model
    - Combiners



Stochastic Secret Sharing

- Stochastic corruption model
  - Every party is corrupted with probability p
  - Natural in some areas in cryptography
    - Leakage resilient circuits in the random probing model
    - Combiners



- Except with negligible error probability over the choice of bad parties:

Stochastic Secret Sharing

- Stochastic corruption model
  - Every party is corrupted with probability p
  - Natural in some areas in cryptography
    - Leakage resilient circuits in the random probing model
    - Combiners
- Except with negligible error probability over the choice of bad parties:
  - **Correctness:** Honest parties can recover s

Stochastic Secret Sharing

- Stochastic corruption model
  - Every party is corrupted with probability p
  - Natural in some areas in cryptography
    - Leakage resilient circuits in the random probing model
    - Combiners

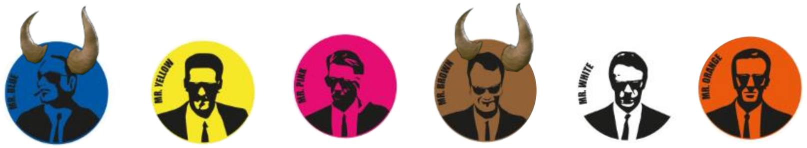
- Except with negligible error probability over the choice of bad parties:
  - **Correctness:** Honest parties can recover s

Stochastic Secret Sharing

- Stochastic corruption model
  - Every party is corrupted with probability p
  - Natural in some areas in cryptography
    - Leakage resilient circuits in the random probing model
    - Combiners
- Except with negligible error probability over the choice of bad parties:
  - **Correctness:** Honest parties can recover s
  - **Privacy:** Corrupt parties have no info about s

Stochastic Secret Sharing

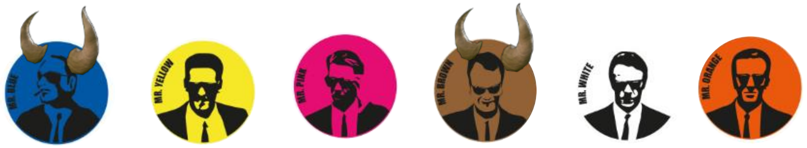
- Stochastic corruption model
  - Every party is corrupted with probability p
  - Natural in some areas in cryptography
    - Leakage resilient circuits in the random probing model
    - Combiners



s_1 s_2 s_3 s_4 s_5 s_6

- Except with negligible error probability over the choice of bad parties:
  - **Correctness:** Honest parties can recover s (=Authorized set)
  - **Privacy:** Corrupt parties have no info about s

Stochastic Secret Sharing



s_1 s_2 s_3 s_4 s_5 s_6

- Stochastic corruption model
  - Every party is corrupted with probability p
  - Natural in some areas in cryptography
    - Leakage resilient circuits in the random probing model
    - Combiners
- Except with negligible error probability over the choice of bad parties:
  - **Correctness:** Honest parties can recover s (=Authorized set)
  - **Privacy:** Corrupt parties have no info about s (=Unauthorized set)

Stochastic Secret Sharing

- Stochastic corruption model
  - Every party is corrupted with probability p
  - Natural in some areas in cryptography
    - Leakage resilient circuits in the random probing model
    - Combiners

- Except with negligible error probability over the choice of bad parties:
  - **Correctness:** Honest parties can recover s (=Authorized set)
  - **Privacy:** Corrupt parties have no info about s (=Unauthorized set)
- $\Rightarrow p < 1/2$

Our Results: SSS with 1-bit shares

Our Results: SSS with 1-bit shares

- n-party SSS with 1-bit shares

Our Results: SSS with 1-bit shares

- n-party SSS with 1-bit shares
  - For every $p < 1/2$

Our Results: SSS with 1-bit shares

- n-party SSS with 1-bit shares
  - For every $p < 1/2$
  - Error $= 2^{-\Omega_p(\sqrt{n})}$

Our Results: SSS with 1-bit shares

- n-party SSS with 1-bit shares
  - For every $p < 1/2$
  - Error $= 2^{-\Omega_p(\sqrt{n})}$
  - Linear and multiplicative

Our Results: SSS with 1-bit shares

- n-party SSS with 1-bit shares
  - For every $p < 1/2$
  - Error $= 2^{-\Omega_p(\sqrt{n})}$
  - Linear and multiplicative
  - Based on Reed-Muller code

Our Results: SSS with 1-bit shares

- n-party SSS with 1-bit shares
  - For every $p < 1/2$
  - Error $= 2^{-\Omega_p(\sqrt{n})}$
  - Linear and multiplicative
  - Based on Reed-Muller code

Our Results: SSS with 1-bit shares

- n-party SSS with 1-bit shares
  - For every $p < 1/2$
  - Error $= 2^{-\Omega_p(\sqrt{n})}$
  - Linear and multiplicative
  - Based on Reed-Muller code

Our Results: SSS with 1-bit shares

$s \in \{0,1\}$

- n-party SSS with 1-bit shares
  - For every $p < 1/2$
  - Error $= 2^{-\Omega_p(\sqrt{n})}$
  - Linear and multiplicative
  - Based on Reed-Muller code

…

Our Results: SSS with 1-bit shares

$s \in \{0,1\}$
$f(x_1, \dots, x_m)$
Degree-$m/2$
$f(0, \dots, 0) = s$

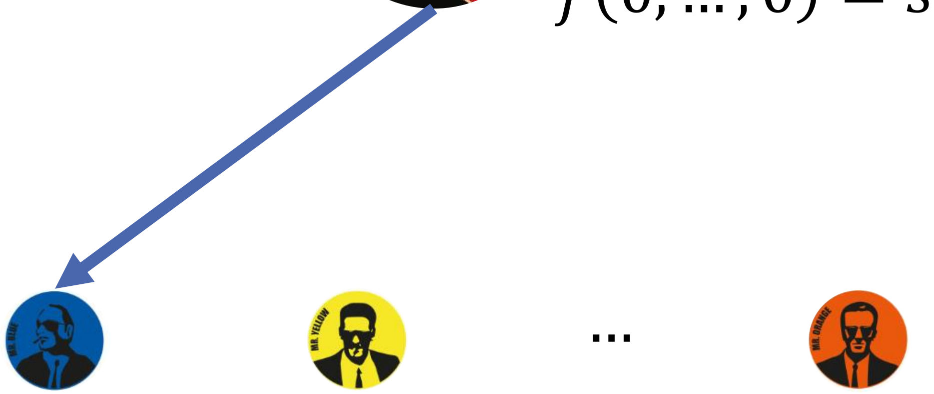
- n-party SSS with 1-bit shares
  - For every $p < 1/2$
  - Error $= 2^{-\Omega_p(\sqrt{n})}$
  - Linear and multiplicative
  - Based on Reed-Muller code



…


Our Results: SSS with 1-bit shares

$s \in \{0,1\}$
$f(x_1, \dots, x_m)$
Degree-$m/2$
$f(0, \dots, 0) = s$

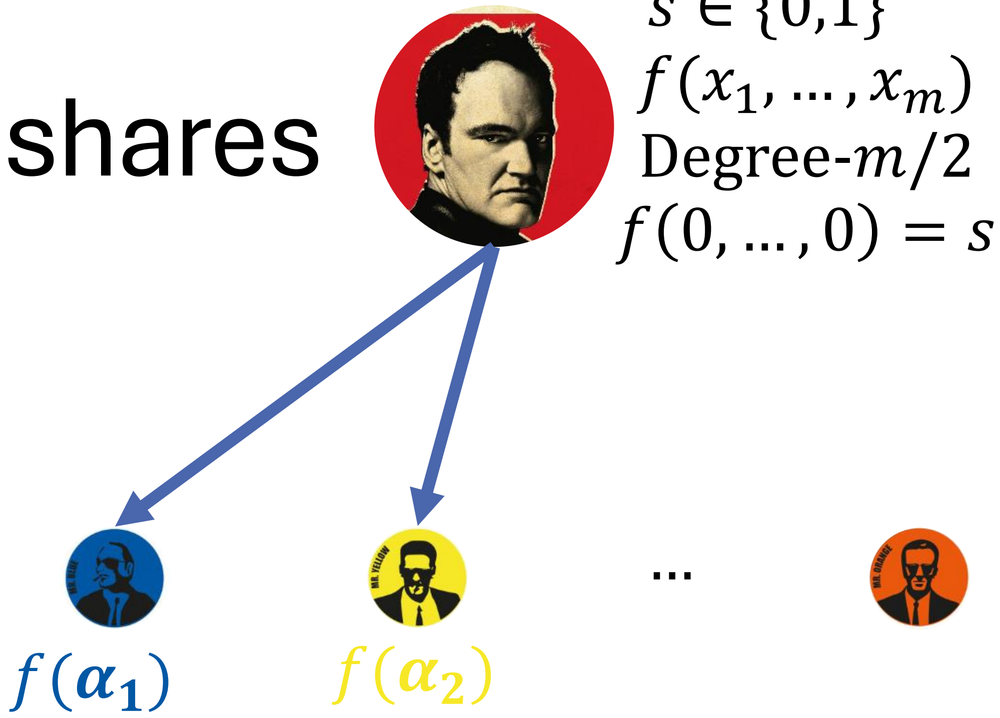


$f(\boldsymbol{\alpha}_1)$

- n-party SSS with 1-bit shares
  - For every $p < 1/2$
  - Error $= 2^{-\Omega_p(\sqrt{n})}$
  - Linear and multiplicative
  - Based on Reed-Muller code

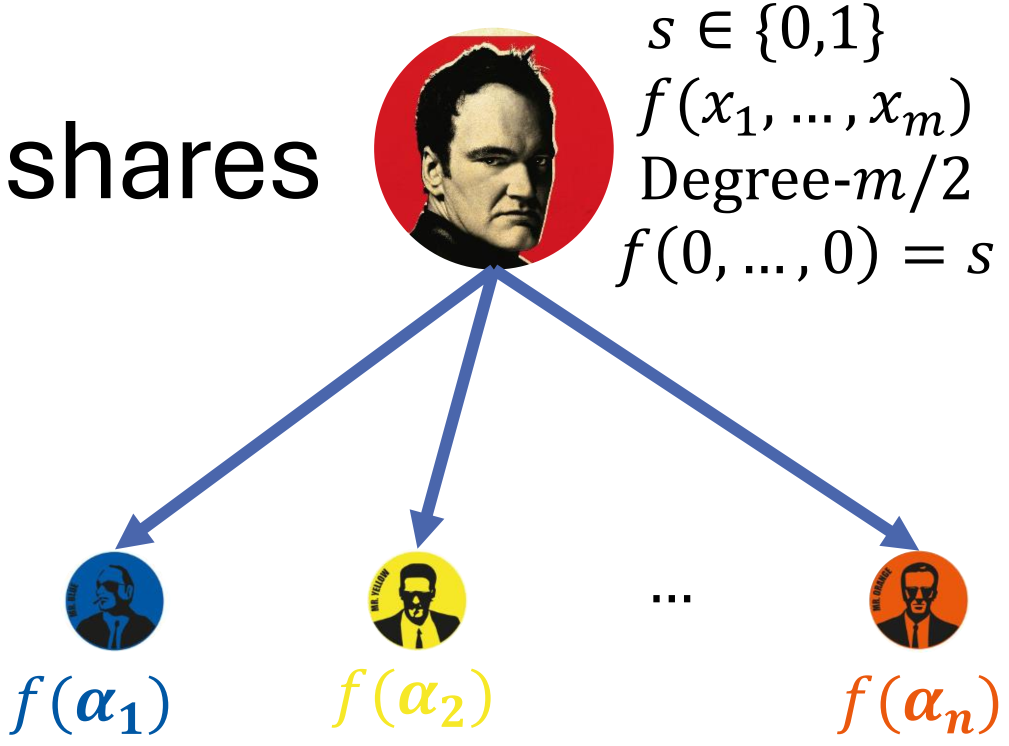
Our Results: SSS with 1-bit shares

- n-party SSS with 1-bit shares
  - For every $p < 1/2$
  - Error $= 2^{-\Omega_p(\sqrt{n})}$
  - Linear and multiplicative
  - Based on Reed-Muller code



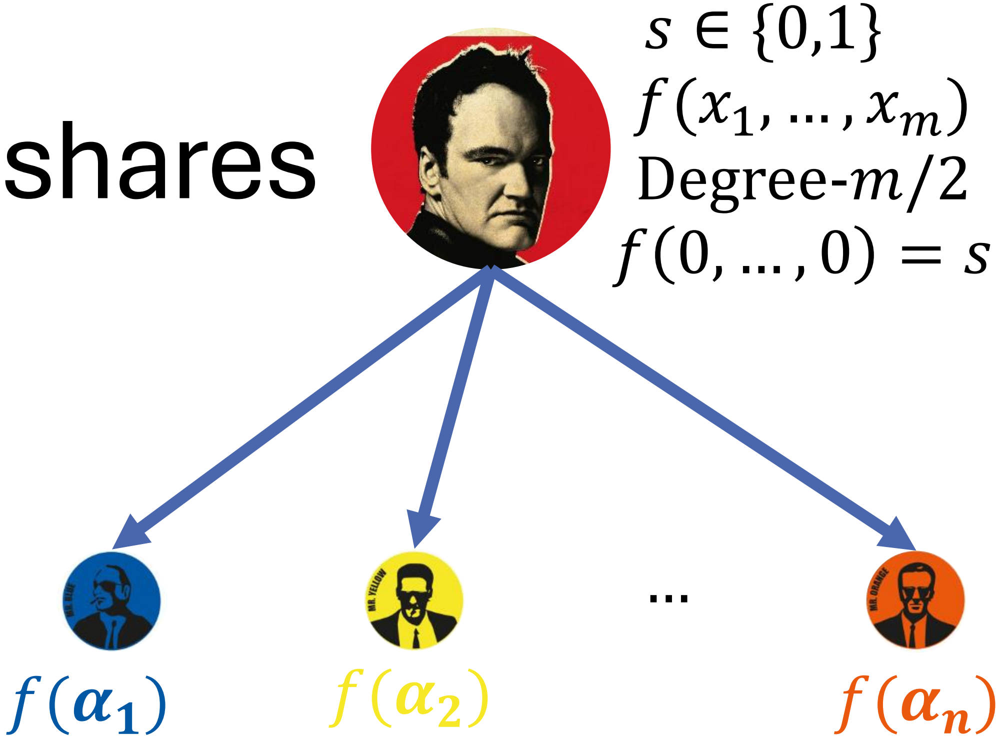
Our Results: SSS with 1-bit shares

- n-party SSS with 1-bit shares
  - For every $p < 1/2$
  - Error $= 2^{-\Omega_p(\sqrt{n})}$
  - Linear and multiplicative
  - Based on Reed-Muller code



Our Results: SSS with 1-bit shares

- n-party SSS with 1-bit shares
  - For every $p < 1/2$
  - Error $= 2^{-\Omega_p(\sqrt{n})}$
  - Linear and multiplicative
  - Based on Reed-Muller code



Our Results: SSS with 1-bit shares

- n-party SSS with 1-bit shares
  - For every $p < 1/2$
  - Error $= 2^{-\Omega_p(\sqrt{n})}$
  - Linear and multiplicative
  - Based on Reed-Muller code

Linear code C and $C^{\perp}$ behave well over $BEC(p)$

$\Downarrow$

p-stochastic secret sharing

Our Results: SSS with 1-bit shares

- n-party SSS with 1-bit shares
  - For every $p < 1/2$
  - Error $= 2^{-\Omega_p(\sqrt{n})}$
  - Linear and multiplicative
  - Based on Reed-Muller code

$BEC(p)$

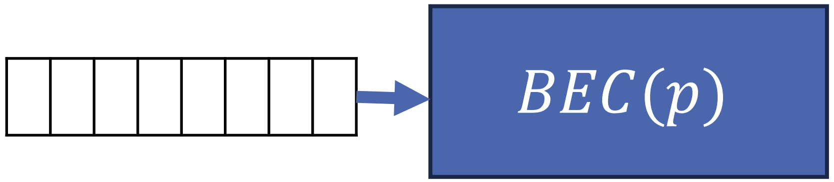
Linear code C and $C^{\perp}$ behave well over $BEC(p)$

⇓

p-stochastic secret sharing

Our Results: SSS with 1-bit shares

- n-party SSS with 1-bit shares
  - For every $p < 1/2$
  - Error $= 2^{-\Omega_p(\sqrt{n})}$
  - Linear and multiplicative
  - Based on Reed-Muller code



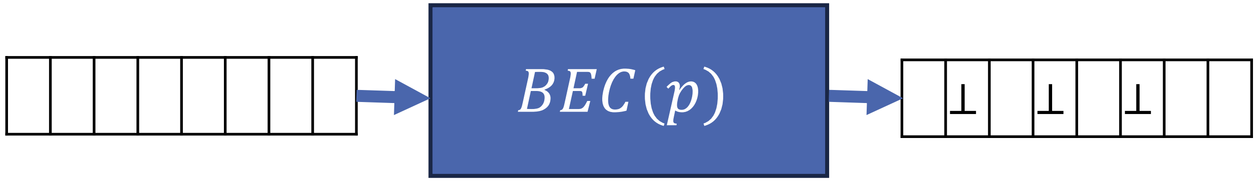
Linear code C and $C^{\perp}$ behave well over $BEC(p)$

$\Downarrow$

p-stochastic secret sharing

Our Results: SSS with 1-bit shares

- n-party SSS with 1-bit shares
  - For every $p < 1/2$
  - Error $= 2^{-\Omega_p(\sqrt{n})}$
  - Linear and multiplicative
  - Based on Reed-Muller code



Linear code C and $C^{\perp}$ behave well over $BEC(p)$

p-stochastic secret sharing

Our Results: SSS with 1-bit shares

- n-party SSS with 1-bit shares
  - For every $p < 1/2$
  - Error $= 2^{-\Omega_p(\sqrt{n})}$
  - Linear and multiplicative
  - Based on Reed-Muller code

Reed-Muller achieves capacity over $BEC(p)$ [KKMPSU17]

Linear code C and $C^{\perp}$ behave well over $BEC(p)$

p-stochastic secret sharing

Our Results: SSS with 1-bit shares

- n-party SSS with 1-bit shares
  - For every $p < 1/2$
  - Error $= 2^{-\Omega_p(\sqrt{n})}$
  - Linear and multiplicative
  - Based on Reed-Muller code

Our Results: SSS with 1-bit shares

- n-party SSS with 1-bit shares
  - For every $p < 1/2$
  - Error $= 2^{-\Omega_p(\sqrt{n})}$
  - Linear and multiplicative
  - Based on Reed-Muller code

- Probabilistic construction of SSS with1-bit shares

Our Results: SSS with 1-bit shares

- n-party SSS with 1-bit shares
  - For every $p < 1/2$
  - Error $= 2^{-\Omega_p(\sqrt{n})}$
  - Linear and multiplicative
  - Based on Reed-Muller code

- Probabilistic construction of SSS with1-bit shares
  - For every $p < 1/2$

Our Results: SSS with 1-bit shares

- n-party SSS with 1-bit shares
  - For every $p < 1/2$
  - Error $= 2^{-\Omega_p(\sqrt{n})}$
  - Linear and multiplicative
  - Based on Reed-Muller code

- Probabilistic construction of SSS with1-bit shares
  - For every $p < 1/2$
  - Error $= 2^{-\Omega_p(n)}$

Our Results: SSS with 1-bit shares

- n-party SSS with 1-bit shares
  - For every $p < 1/2$
  - Error $= 2^{-\Omega_p(\sqrt{n})}$
  - Linear and multiplicative
  - Based on Reed-Muller code

- Probabilistic construction of SSS with1-bit shares
  - For every $p < 1/2$
  - Error $= 2^{-\Omega_p(n)}$
  - Linear

Our Results: SSS with 1-bit shares

- n-party SSS with 1-bit shares
  - For every $p < 1/2$
  - Error $= 2^{-\Omega_p(\sqrt{n})}$
  - Linear and multiplicative
  - Based on Reed-Muller code

- Probabilistic construction of SSS with1-bit shares
  - For every $p < 1/2$
  - Error $= 2^{-\Omega_p(n)}$
  - Linear

Our Results: SSS with 1-bit shares

- n-party SSS with 1-bit shares
  - For every $p < 1/2$
  - Error $= 2^{-\Omega_p(\sqrt{n})}$
  - Linear and multiplicative
  - Based on Reed-Muller code

- Probabilistic construction of SSS with1-bit shares
  - For every $p < 1/2$
  - Error $= 2^{-\Omega_p(n)}$
  - Linear
  - Not multiplicative

Our Results: SSS with 1-bit shares

- n-party SSS with 1-bit shares
  - For every $p < 1/2$
  - Error $= 2^{-\Omega_p(\sqrt{n})}$
  - Linear and multiplicative
  - Based on Reed-Muller code

- Probabilistic construction of SSS with1-bit shares
  - For every $p < 1/2$
  - Error $= 2^{-\Omega_p(n)}$
  - Linear
  - Not multiplicative

Linear-size circuit!

Standard corruptions?

Standard corruptions?

- Static secret sharing

Standard corruptions?

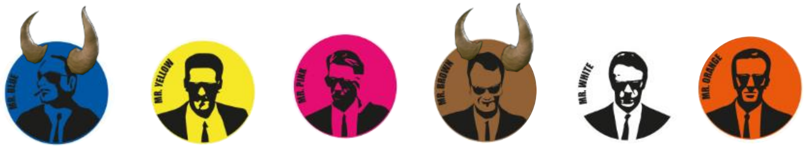
- Static secret sharing
  - Threshold t

Standard corruptions?

- Static secret sharing
  - Threshold t
  - Public randomness R

Standard corruptions?

- Static secret sharing
  - Threshold t
  - Public randomness R



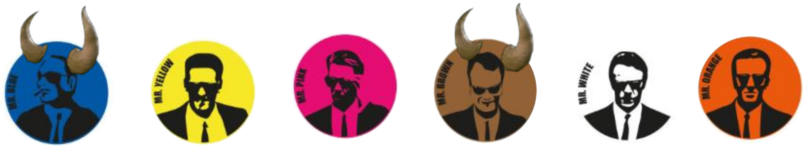
Standard corruptions?

- Static secret sharing
  - Threshold t
  - Public randomness R

Standard corruptions?

R

- Static secret sharing
  - Threshold t
  - Public randomness R



s_1 s_2 s_3 s_4 s_5 s_6

Standard corruptions?

R

- Static secret sharing
  - Threshold t
  - Public randomness R
  - For every t corrupt parties, except with negligible probability over R:

s_1 s_2 s_3 s_4 s_5 s_6

Standard corruptions?

R

- Static secret sharing
  - Threshold t
  - Public randomness R
  - For every t corrupt parties, except with negligible probability over R:
  - **Correctness:** Honest parties can recover s

s_1 s_2 s_3 s_4 s_5 s_6

Standard corruptions?

R

s_1 s_2 s_3 s_4 s_5 s_6

- Static secret sharing
  - Threshold t
  - Public randomness R
  - For every t corrupt parties, except with negligible probability over R:
  - **Correctness:** Honest parties can recover s
  - **Privacy:** Corrupt parties have no info about s

Standard corruptions?

R

- Static secret sharing
  - Threshold t
  - Public randomness R
  - For every t corrupt parties, except with negligible probability over R:
  - **Correctness:** Honest parties can recover s (=Authorized set)
  - **Privacy:** Corrupt parties have no info about s

s_1 s_2 s_3 s_4 s_5 s_6

Standard corruptions?

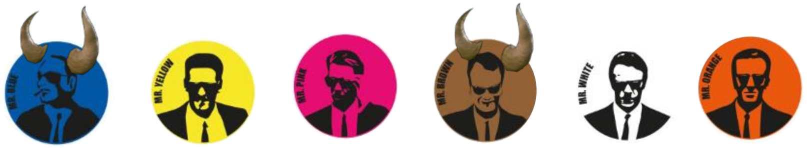
R

s_1 s_2 s_3 s_4 s_5 s_6

- Static secret sharing
  - Threshold t
  - Public randomness R
  - For every t corrupt parties, except with negligible probability over R:
  - **Correctness:** Honest parties can recover s (=Authorized set)
  - **Privacy:** Corrupt parties have no info about s (=Unauthorized set)

Standard corruptions?

R



s_1 s_2 s_3 s_4 s_5 s_6

- Static secret sharing
  - Threshold t
  - Public randomness R
  - For every t corrupt parties, except with negligible probability over R:
  - **Correctness:** Honest parties can recover s (=Authorized set)
  - **Privacy:** Corrupt parties have no info about s (=Unauthorized set)

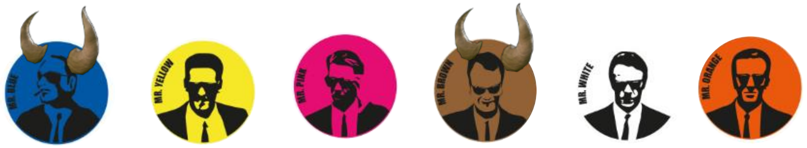
  - Public randomness chosen once and for all

Standard corruptions?

R

- Static secret sharing
  - Threshold t
  - Public randomness R
  - For every t corrupt parties, except with negligible probability over R:
  - **Correctness:** Honest parties can recover s (=Authorized set)
  - **Privacy:** Corrupt parties have no info about s (=Unauthorized set)

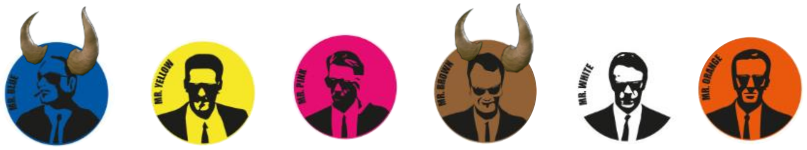
  - Public randomness chosen once and for all
    - Can be reused by the dealer and other dealers



s_1 s_2 s_3 s_4 s_5 s_6

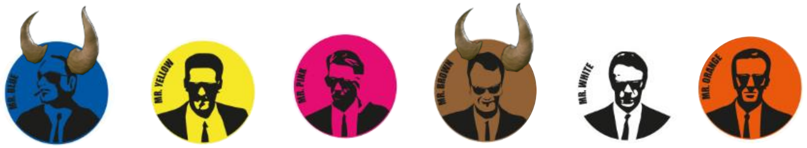
From Stochastic to Static

R



From Stochastic to Static

R



p-stochastic secret sharing with error δ

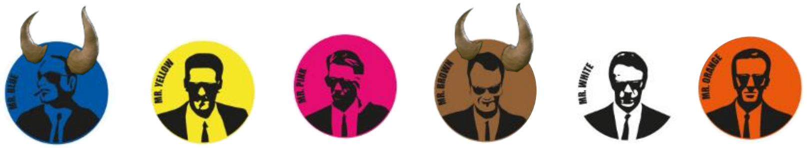
From Stochastic to Static

R

p-stochastic secret sharing with error δ

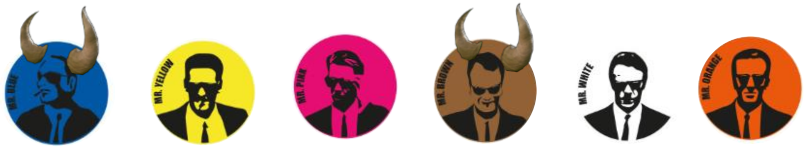
↓

Static secret sharing
threshold $t = \lfloor pn \rfloor$
error 2δ



From Stochastic to Static

R



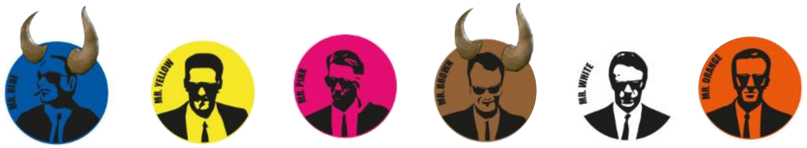
p-stochastic secret sharing with error δ

Permute the parties!

Static secret sharing threshold $t = \lfloor pn \rfloor$ error 2δ

From Stochastic to Static

π



p-stochastic secret sharing with error δ

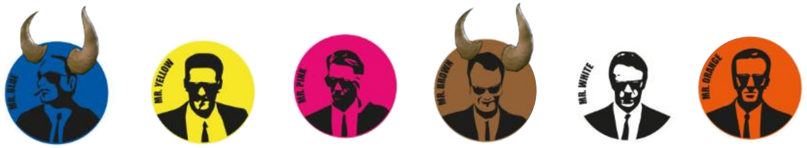
↓

Permute the parties!

Static secret sharing
threshold $t = \lfloor pn \rfloor$
error 2δ

From Stochastic to Static

π



$s_{\pi(1)}$

p-stochastic secret sharing with error δ

↓ Permute the parties!

Static secret sharing
threshold $t = \lfloor pn \rfloor$
error 2δ

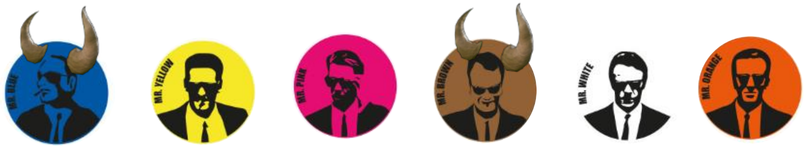
From Stochastic to Static

π

p-stochastic secret sharing with error δ

Permute the parties!

Static secret sharing
threshold $t = \lfloor pn \rfloor$
error 2δ



$s_{\pi(1)}$ $s_{\pi(2)}$

From Stochastic to Static

π

p-stochastic secret sharing
with error δ

Permute the parties!

Static secret sharing
threshold $t = \lfloor pn \rfloor$
error 2δ

$s_{\pi(1)}$ $s_{\pi(2)}$ $s_{\pi(3)}$ $s_{\pi(4)}$ $s_{\pi(5)}$ $s_{\pi(6)}$

From Stochastic to Static

From Stochastic to Static

- n-party static SS with 1-bit shares
  - Threshold $t = 0.499n$
  - Error $= 2^{-\Omega(\sqrt{n})}$
  - Linear and multiplicative
  - Based on Reed-Muller code

From Stochastic to Static

- n-party static SS with 1-bit shares
  - Threshold $t = 0.499n$
  - Error $= 2^{-\Omega(\sqrt{n})}$
  - Linear and multiplicative
  - Based on Reed-Muller code

- Probabilistic construction with 1-bit shares
  - Threshold $t = 0.499n$
  - Error $= 2^{-\Omega(n)}$
  - Linear
  - Not multiplicative

From Stochastic to Static

- n-party static SS with 1-bit shares
  - Threshold $t = 0.499n$
  - Error $= 2^{-\Omega(\sqrt{n})}$
  - Linear and multiplicative
  - Based on Reed-Muller code

Optimal communication!

- Probabilistic construction with 1-bit shares
  - Threshold $t = 0.499n$
  - Error $= 2^{-\Omega(n)}$
  - Linear
  - Not multiplicative

From Stochastic to Static

- n-party static SS with 1-bit shares
  - Threshold $t = 0.499n$
  - Error $= 2^{-\Omega(\sqrt{n})}$
  - Linear and multiplicative
  - Based on Reed-Muller code

Optimal communication!

- Probabilistic construction with 1-bit share
  - Threshold $t = 0.499n$
  - Error $= 2^{-\Omega(n)}$
  - Linear
  - Not multiplicative

Optimal computation! Amortized linear-size circuits!

Applications: Communication Efficient MPC

Applications: Communication Efficient MPC

Statistical MPC for **Boolean circuits**

Applications: Communication Efficient MPC

Statistical MPC for **Boolean circuits**

- Passive static security against $t = 0.499n$ parties

Applications: Communication Efficient MPC

Statistical MPC for **Boolean circuits**

- Passive static security against $t = 0.499n$ parties
- **Offline phase**: $O(|C|)$ bits per party

Applications: Communication Efficient MPC

Statistical MPC for **Boolean circuits**

- Passive static security against $t = 0.499n$ parties
- **Offline phase**: $O(|C|)$ bits per party
- **Online phase:** 1-bit per gate!

Applications: Communication Efficient MPC

Statistical MPC for **Boolean circuits**

- Passive static security against $t = 0.499n$ parties
- **Offline phase**: $O(|C|)$ bits per party
- **Online phase:** 1**-**bit per gate!
- **Total communication:** $O(|C| \cdot n)$ bits

Applications: Communication Efficient MPC

Statistical MPC for **Boolean circuits**

- Passive static security against $t = 0.499n$ parties
- **Offline phase**: $O(|C|)$ bits per party
- **Online phase:** 1-bit per gate!
- **Total communication:** $O(|C| \cdot n)$ bits
- **Error**: $2^{-\Omega(\sqrt{n})}$

Applications: Communication Efficient MPC

Statistical MPC for **Boolean circuits**

- Passive static security against $t = 0.499n$ parties
- **Offline phase**: $O(|C|)$ bits per party
- **Online phase:** 1-bit per gate!
- **Total communication:** $O(|C| \cdot n)$ bits
- **Error**: $2^{-\Omega(\sqrt{n})}$
- $O(|C| \cdot \log(n))$ time in RAM model

Applications: Communication Efficient MPC

Statistical MPC for **Boolean circuits**

- Passive static security against $t = 0.499n$ parties
- **Offline phase**: $O(|C|)$ bits per party
- **Online phase:** 1**-**bit per gate!
- **Total communication:** $O(|C| \cdot n)$ bits
- **Error**: $2^{-\Omega(\sqrt{n})}$
- $O(|C| \cdot \log(n))$ time in RAM model

Circuit size
$O(|C|)$ per party!

Applications: Communication Efficient MPC

Statistical MPC for **Boolean circuits**

- Passive static security against $t = 0.499n$ parties
- **Offline phase:** $O(|C|)$ bits per party
- **Online phase:** 1-bit per gate!
- **Total communication:** $O(|C| \cdot n)$ bits
- **Error:** $2^{-\Omega(\sqrt{n})}$
- $O(|C| \cdot \log(n))$ time in RAM model

Circuit size
$O(|C|)$ per party!

[Damgård,Ishai,Krøigaard10]
[Genkin,Ishai,Polychroniadou15]
[Garay,Ishai,Ostrovsky,Zikas17]
[Goyal, Polychroniadou,Song21]

Applications: Communication Efficient MPC

Statistical MPC for **Boolean circuits**

- Passive static security against $t = 0.499n$ parties
- **Offline phase**: $O(|C|)$ bits per party
- **Online phase:** 1-bit per gate!
- **Total communication:** $O(|C| \cdot n)$ bits

$O(|C| \cdot \log(n))$ bits

- **Error**: $2^{-\Omega(\sqrt{n})}$
- $O(|C| \cdot \log(n))$ time in RAM model

Circuit size
$O(|C|)$ per party!

[Damgård,Ishai,Krøigaard10]
[Genkin,Ishai,Polychroniadou15]
[Garay,Ishai,Ostrovsky,Zikas17]
[Goyal, Polychroniadou,Song21]

Applications: Communication Efficient MPC

Statistical MPC for **Boolean circuits**

- Passive static security against $t = 0.499n$ parties
- **Offline phase**: $O(|C|)$ bits per party
- **Online phase:** 1-bit per gate!
- **Total communication:** $O(|C| \cdot n)$ bits
- **Error**: $2^{-\Omega(\sqrt{n})}$
- $O(|C| \cdot \log(n))$ time in RAM model

$O(|C| \cdot \log(n))$ bits

Perfect security

Circuit size
$O(|C|)$ per party!

[Damgård,Ishai,Krøigaard10]
[Genkin,Ishai,Polychroniadou15]
[Garay,Ishai,Ostrovsky,Zikas17]
[Goyal, Polychroniadou,Song21]

Applications: Communication Efficient MPC

Statistical MPC for **Boolean circuits**

- Passive static security against $t = 0.499n$ parties
- **Offline phase**: $O(|C|)$ bits per party
- **Online phase:** 1-bit per gate!

Total communication: $O(|C| \cdot \log^3 n)$

Error: $negl(n)$

- $O(|C| \cdot \log(n))$ time in RAM model

$O(|C| \cdot \log(n))$ bits

Perfect security

Circuit size

$O(|C|)$ per party!

[Damgård,Ishai,Krøigaard10]
[Genkin,Ishai,Polychroniadou15]
[Garay,Ishai,Ostrovsky,Zikas17]
[Goyal, Polychroniadou,Song21]

Applications: Communication Efficient MPC

Statistical MPC for **Boolean circuits**

- Passive static security against $t = 0.499n$ parties
- **Offline phase:** $O(|C|)$ bits per party
- **Online phase:** 1-bit per gate!

Total communication: $O(|C| \cdot \log^3 n)$
Error: $negl(n)$

- $O(|C| \cdot \log(n))$ time in RAM model
- Works even in the **real-time model**

$O(|C| \cdot \log(n))$ bits

Perfect security

[Damgård,Ishai,Krøigaard10]
[Genkin,Ishai,Polychroniadou15]
[Garay,Ishai,Ostrovsky,Zikas17]
[Goyal, Polychroniadou,Song21]

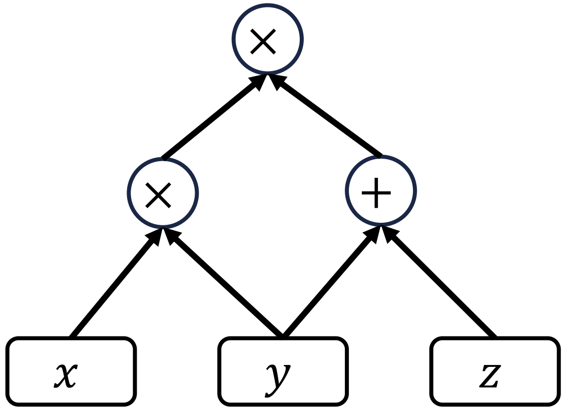
Applications: Communication Efficient MPC

Statistical MPC for **Boolean circuits**

- Passive static security against $t = 0.499n$ parties
- **Offline phase**: $O(|C|)$ bits per party
- **Online phase:** 1-bit per gate!
-
-

Total communication: $O(|C| \cdot \log^3 n)$
Error: $negl(n)$

- $O(|C| \cdot \log(n))$ time in RAM model
- Works even in the **real-time model**



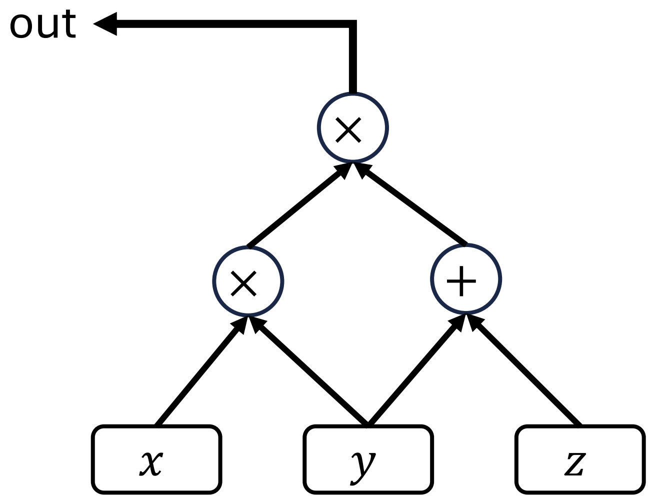
Applications: Communication Efficient MPC

Statistical MPC for **Boolean circuits**

- Passive static security against $t = 0.499n$ parties
- **Offline phase:** $O(|C|)$ bits per party
- **Online phase:** 1-bit per gate!

Total communication: $O(|C| \cdot \log^3 n)$
Error: $negl(n)$

- $O(|C| \cdot \log(n))$ time in RAM model
- Works even in the **real-time model**



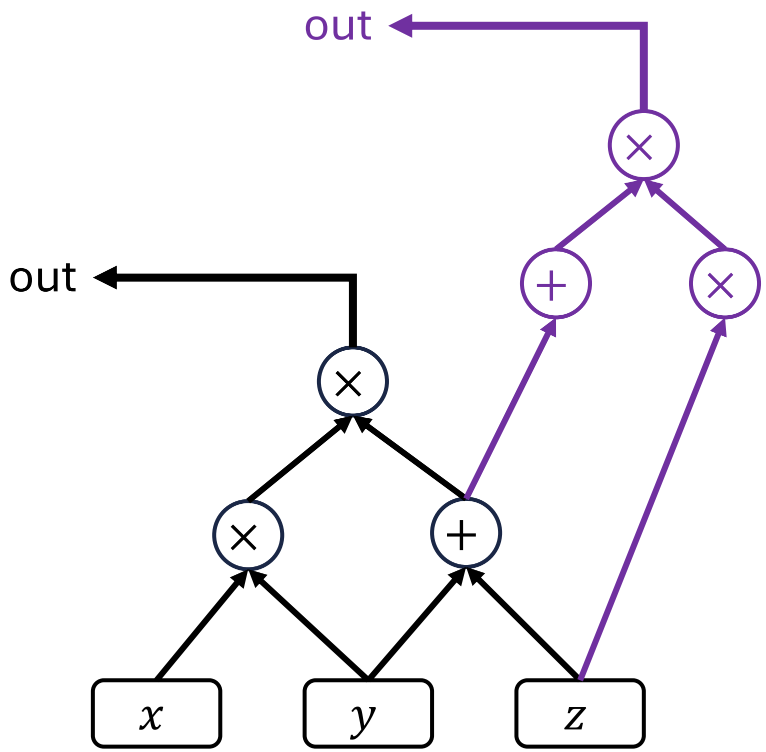
Applications: Communication Efficient MPC

Statistical MPC for **Boolean circuits**

- Passive static security against $t = 0.499n$ parties
- **Offline phase**: $O(|C|)$ bits per party
- **Online phase:** 1-bit per gate!

Total communication: $O(|C| \cdot \log^3 n)$
Error: $negl(n)$

- $O(|C| \cdot \log(n))$ time in RAM model
- Works even in the **real-time model**



Conclusion

Conclusion

- Stochastic secret sharing

Conclusion

- Stochastic secret sharing
  - Stochastic corruption model

Conclusion

- Stochastic secret sharing
  - Stochastic corruption model
  - Linear codes $C, C^{\perp}$ behave well over BEC $\Rightarrow$ Stochastic secret sharing

Conclusion

- Stochastic secret sharing
  - Stochastic corruption model
  - Linear codes $C, C^{\perp}$ behave well over BEC ⇒ Stochastic secret sharing
  - 1-bit shares: Reed-Muller, random linear codes

Conclusion

- Stochastic secret sharing
  - Stochastic corruption model
  - Linear codes $C, C^{\perp}$ behave well over BEC ⇒ Stochastic secret sharing
  - 1-bit shares: Reed-Muller, random linear codes
- Static secret sharing

Conclusion

- Stochastic secret sharing
  - Stochastic corruption model
  - Linear codes $C, C^{\perp}$ behave well over BEC ⇒ Stochastic secret sharing
  - 1-bit shares: Reed-Muller, random linear codes
- Static secret sharing
  - Stochastic ⇒ static

Conclusion

- Stochastic secret sharing
  - Stochastic corruption model
  - Linear codes $C, C^{\perp}$ behave well over BEC $\Rightarrow$ Stochastic secret sharing
  - 1-bit shares: Reed-Muller, random linear codes
- Static secret sharing
  - Stochastic $\Rightarrow$ static
  - 1-bit shares: Reed-Muller, random linear codes

Conclusion

- Stochastic secret sharing
  - Stochastic corruption model
  - Linear codes $C, C^{\perp}$ behave well over BEC ⇒ Stochastic secret sharing
  - 1-bit shares: Reed-Muller, random linear codes
- Static secret sharing
  - Stochastic ⇒ static
  - 1-bit shares: Reed-Muller, random linear codes
- Applications to MPC

Conclusion

- Stochastic secret sharing
  - Stochastic corruption model
  - Linear codes $C, C^{\perp}$ behave well over BEC ⇒ Stochastic secret sharing
  - 1-bit shares: Reed-Muller, random linear codes
- Static secret sharing
  - Stochastic ⇒ static
  - 1-bit shares: Reed-Muller, random linear codes
- Applications to MPC
  - Secure computation of Boolean circuits

Conclusion

- Stochastic secret sharing
  - Stochastic corruption model
  - Linear codes $C, C^{\perp}$ behave well over BEC $\Rightarrow$ Stochastic secret sharing
  - 1-bit shares: Reed-Muller, random linear codes
- Static secret sharing
  - Stochastic $\Rightarrow$ static
  - 1-bit shares: Reed-Muller, random linear codes
- Applications to MPC
  - Secure computation of Boolean circuits
  - Online communication: 1-bit per gate per party

Conclusion

- Stochastic secret sharing
  - Stochastic corruption model
  - Linear codes $C, C^{\perp}$ behave well over BEC $\Rightarrow$ Stochastic secret sharing
  - 1-bit shares: Reed-Muller, random linear codes
- Static secret sharing
  - Stochastic $\Rightarrow$ static
  - 1-bit shares: Reed-Muller, random linear codes
- Applications to MPC
  - Secure computation of Boolean circuits
  - Online communication: 1-bit per gate per party
  - Real-time model

Conclusion

- Stochastic secret sharing
  - Stochastic corruption model
  - Linear codes $C, C^{\perp}$ behave well over BEC $\Rightarrow$ Stochastic secret sharing
  - 1-bit shares: Reed-Muller, random linear codes
- Static secret sharing
  - Stochastic $\Rightarrow$ static
  - 1-bit shares: Reed-Muller, random linear codes
- Applications to MPC
  - Secure computation of Boolean circuits
  - Online communication: 1-bit per gate per party
  - Real-time model

Thank You!