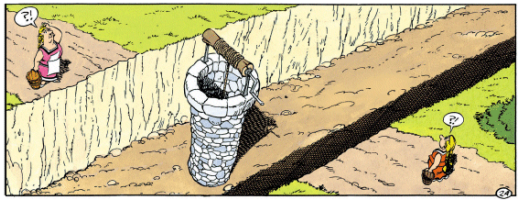
Masking vs. Multiparty Computation: How Large is the Gap for AES?



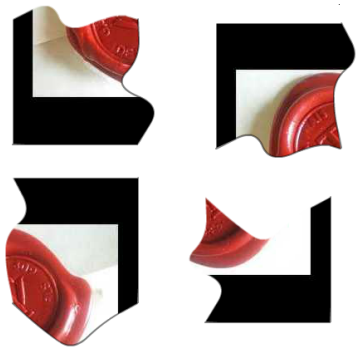
Vincent Grosso[1], François-Xavier Standaert[1], Sebastian Faust[2].

[1] ICTEAM/ELEN/Crypto Group, Université catholique de Louvain, Belgium.
[2] Ecole Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland.

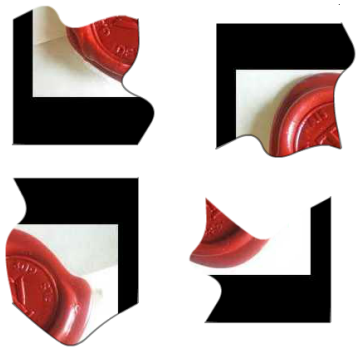
CHES 2013, Santa Barbara, California, USA.

Secret Sharing

Secret Sharing

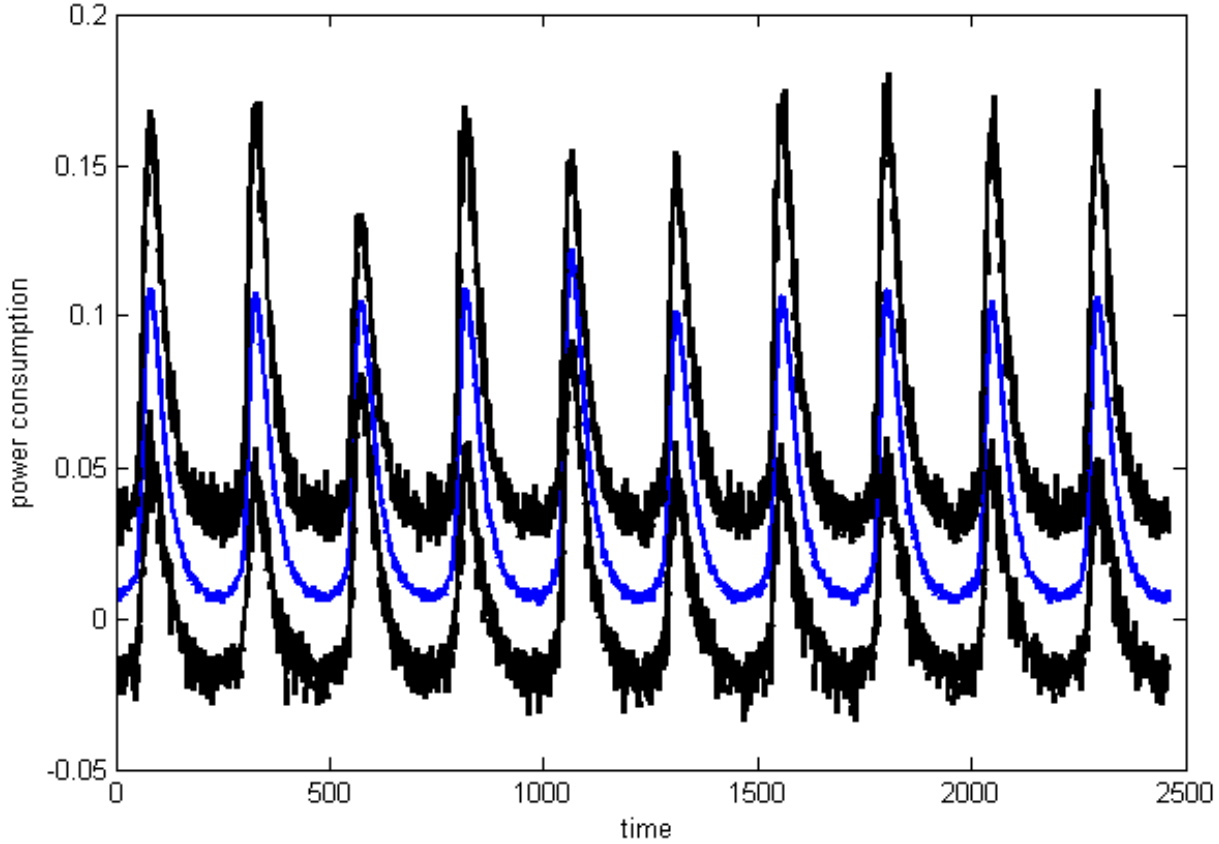


Secret Sharing



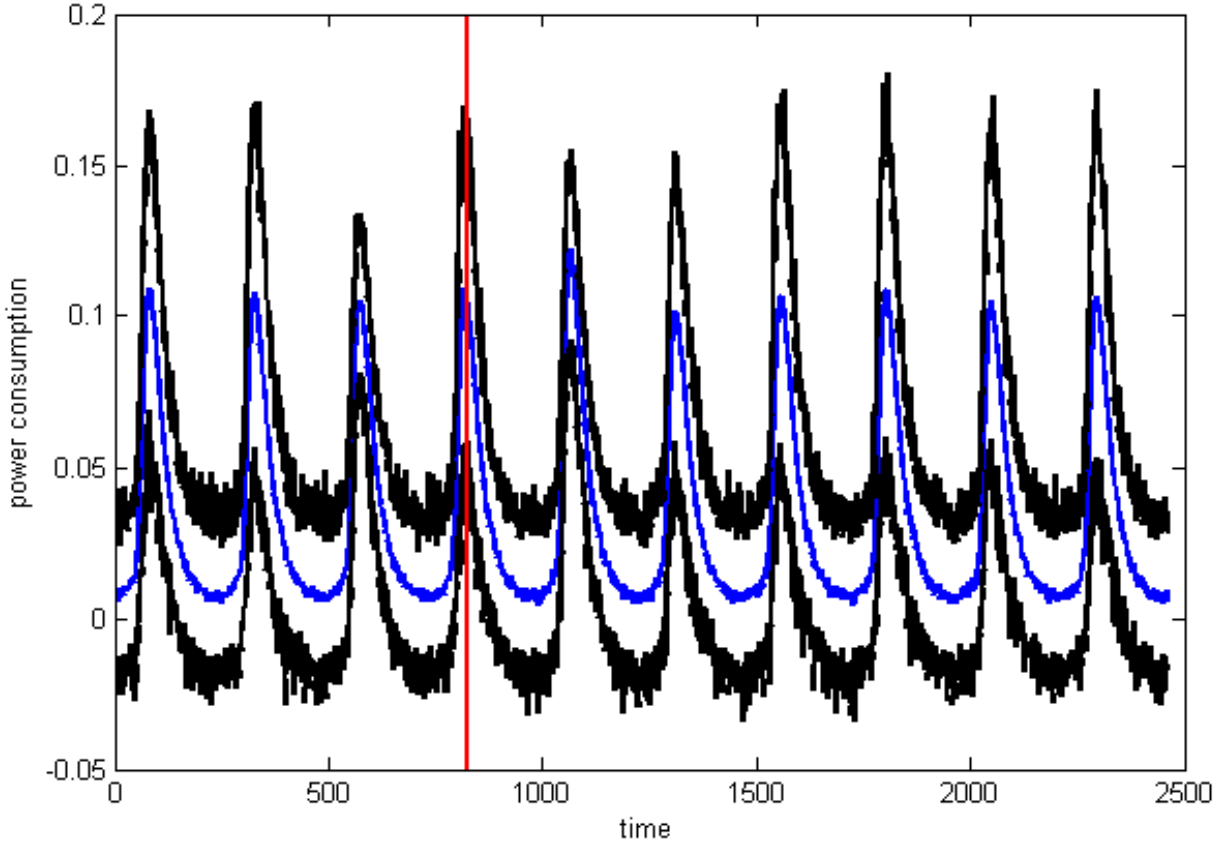
P(|) = P( )

Masking ≃ Computing on Shared Values



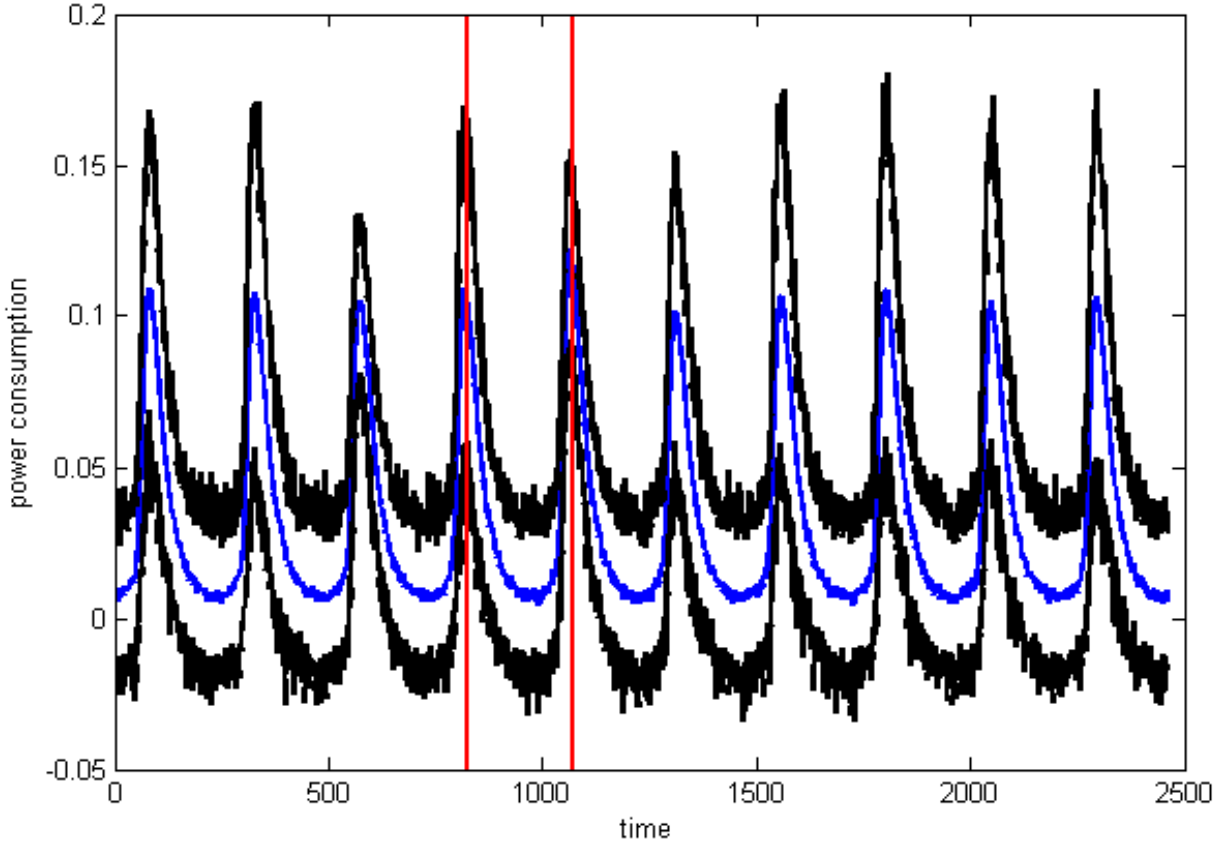
Traces contain information plus some noise.

Masking ≃ Computing on Shared Values



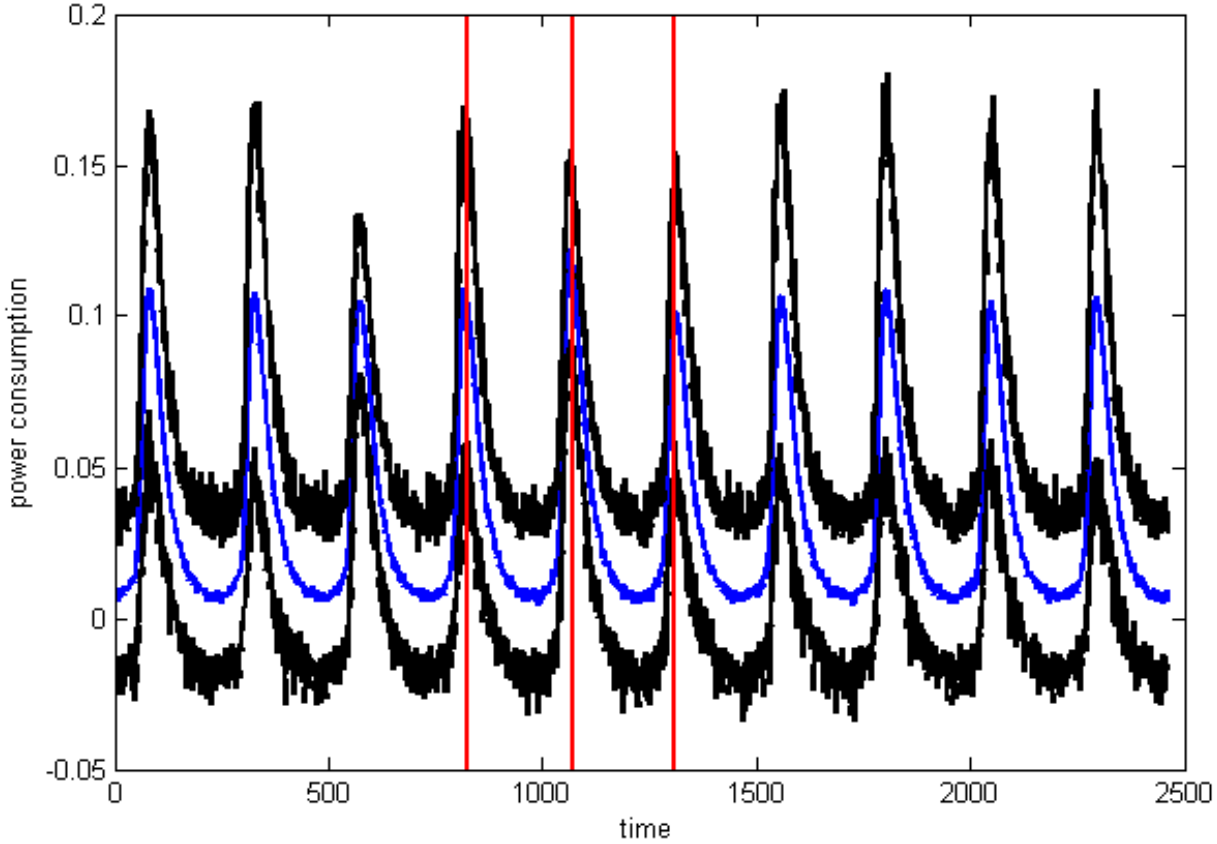
Unprotected device: the leakage of 1 share is needed to mount an attack.

Masking ≃ Computing on Shared Values



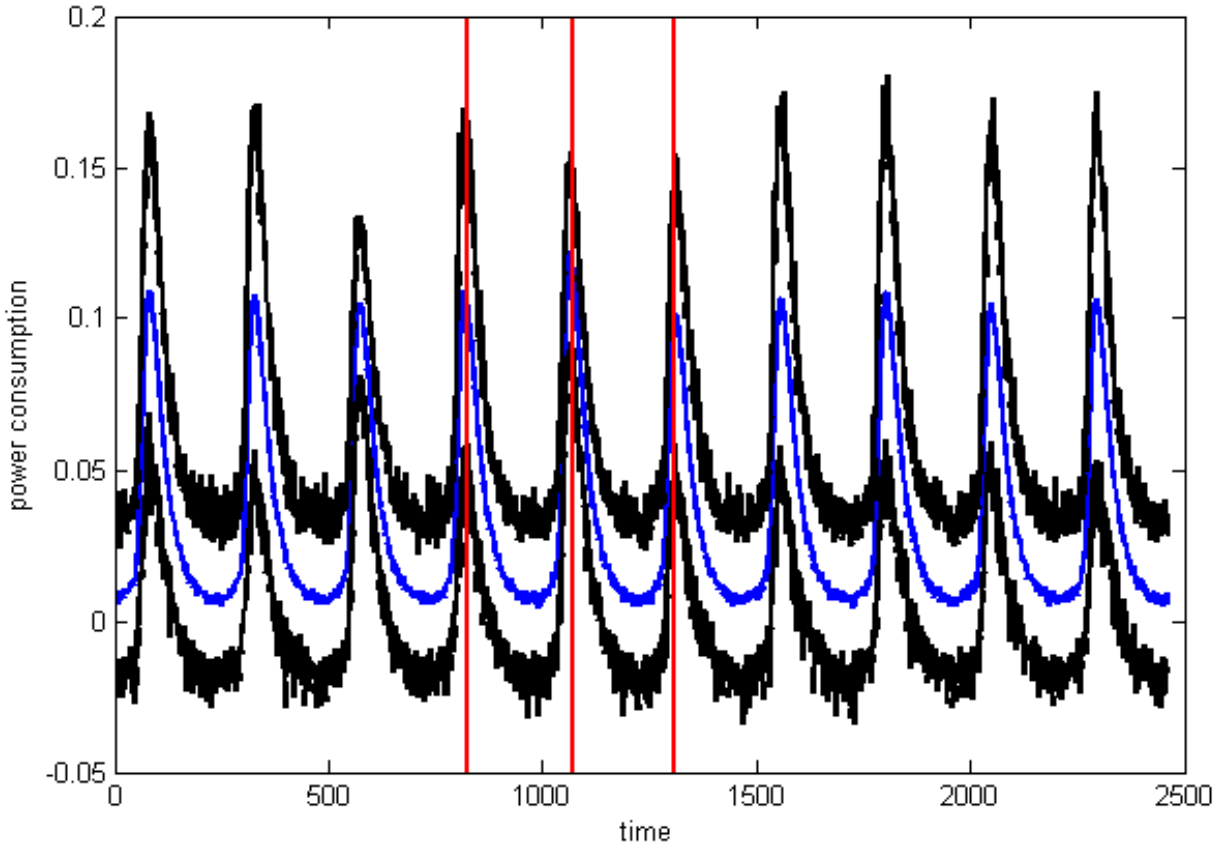
Protected device with 2 shares: ideally the leakage of 2 shares is needed to mount an attack.

Masking ≃ Computing on Shared Values



Protected device with 3 shares: ideally the leakage of 3 shares is needed to mount an attack.

Masking ≃ Computing on Shared Values



Masking order: minimal number of shares of which the leakage has to be exploited minus 1.

Intuition

We have to combine leakage of shares $\Rightarrow$ multiplication noise.

The data complexity of attacks against masking is ideally ($\simeq$ independent leakages) exponential in the masking order with noise as a basis.

Masking Schemes

- Re-computation
- Additive
- Multiplicative
- Affine
- Polynomial/MPC
- Threshold

Masking Schemes

- ~~Re-computation~~: not higher-order
- Additive
- Multiplicative
- ~~Affine~~: not higher-order
- Polynomial/MPC
- ~~Threshold~~: not higher-order

Masking Schemes

- ~~Re-computation~~: not higher-order
- Boolean
- Multiplicative
- ~~Affine~~: not higher-order
- Polynomial/MPC
- ~~Threshold~~: not higher-order

Masking Schemes

- ~~Re-computation~~: not higher-order
- Boolean
- Switch
- ~~Affine~~: not higher-order
- Polynomial/MPC
- ~~Threshold~~: not higher-order

Masking Schemes

- ~~Re-computation~~: not higher-order
- Boolean
  - **Rivain Prouff'10** secure multiplication
  - **Kim Hong Lim'11** subfield
- Switch
- ~~Affine~~: not higher-order
- Polynomial/MPC
- ~~Threshold~~: not higher-order

Masking Schemes

- ~~Re-computation~~: not higher-order
- Boolean
  - **Rivain Prouff'10** secure multiplication
  - **Kim Hong Lim'11** subfield
- Switch
  - **Genelle Prouff Quisquater'11**
- ~~Affine~~: not higher-order
- Polynomial/MPC
- ~~Threshold~~: not higher-order

Masking Schemes

- ~~Re-computation~~: not higher-order
- Boolean
  - **Rivain Prouff'10** secure multiplication
  - **Kim Hong Lim'11** subfield
- Switch
  - **Genelle Prouff Quisquater'11**
- ~~Affine~~: not higher-order
- Polynomial/MPC
  - **Prouff Roche'11**
- ~~Threshold~~: not higher-order

Pros and Cons

- Masking (**RP'10**, **KHL'11**, **GPQ'11**):

Pros:

- efficiency
- well studied

Cons:

- sensitive to glitches

Pros and Cons

- Masking (**RP'10**, **KHL'11**, **GPQ'11**):

Pros:

- efficiency
- well studied

Cons:

- sensitive to glitches

- MPC (**PR'11**):

Pros:

- leaks less information
- glitch-resistance (independent leakage)
- error detection?

Cons:

- more expensive in time
- more expensive in memory
- not clear how much

Contribution

1. What is the cost for each scheme?
   Unified comparison of existing schemes.
2. Can we improve the MPC technique?
   Packed secret sharing.
3. How good must be the randomness?
   Requirements and impact on efficiency.

Methodology

- Target device: ATMEGA644p
- Generic description in C, compiled with avr_gcc
- Supported by assembly subroutines
- Maximum overhead: $< 2\times$ compared to previous work
- Sufficient to obtain an idea on the efficiency of each scheme

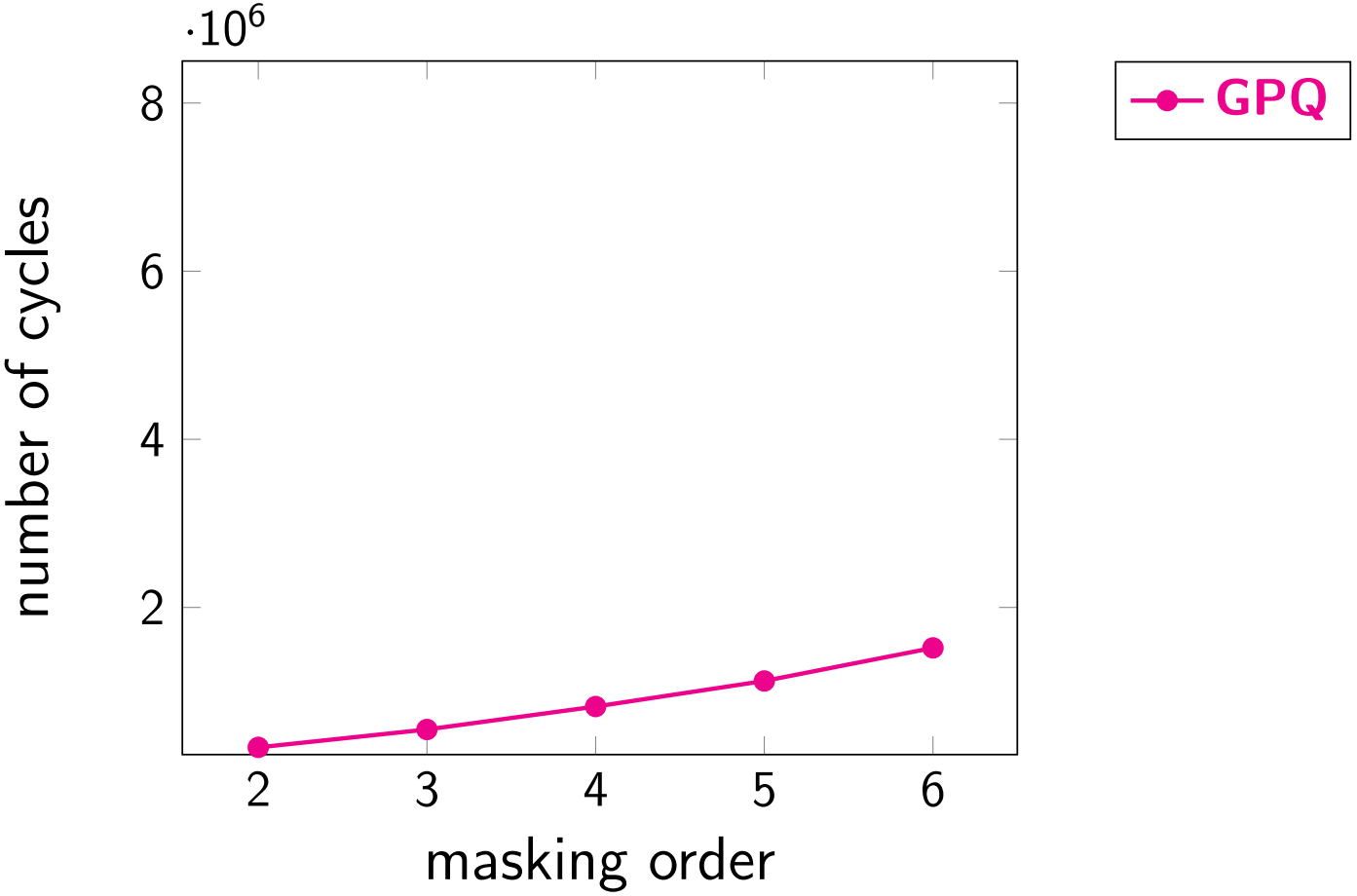
Outline

1. Unified comparison of existing schemes

2. Efficiency improvement with packed secret sharing

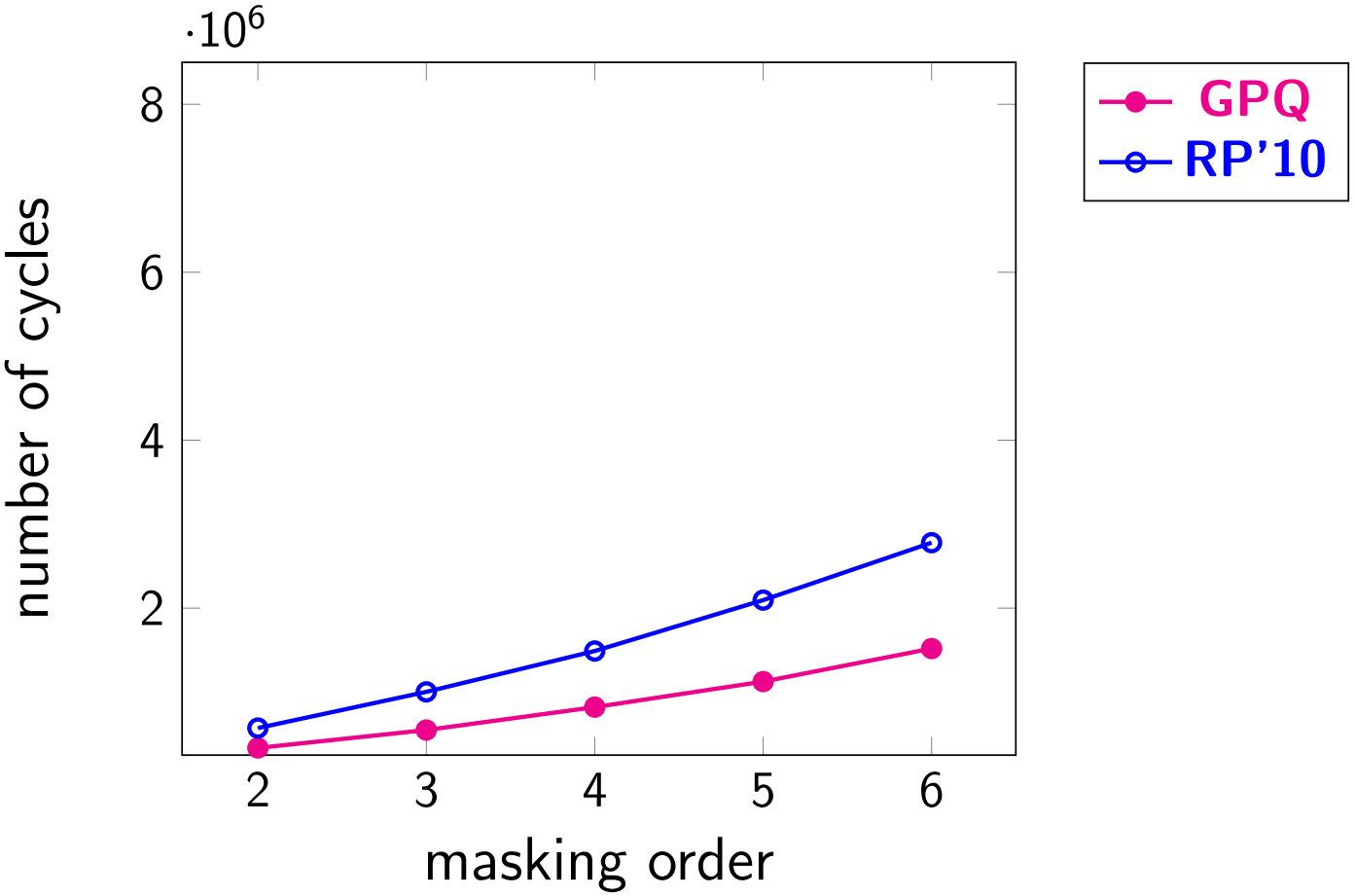
3. Randomness requirements and impact on efficiency of masking schemes

Existing Schemes



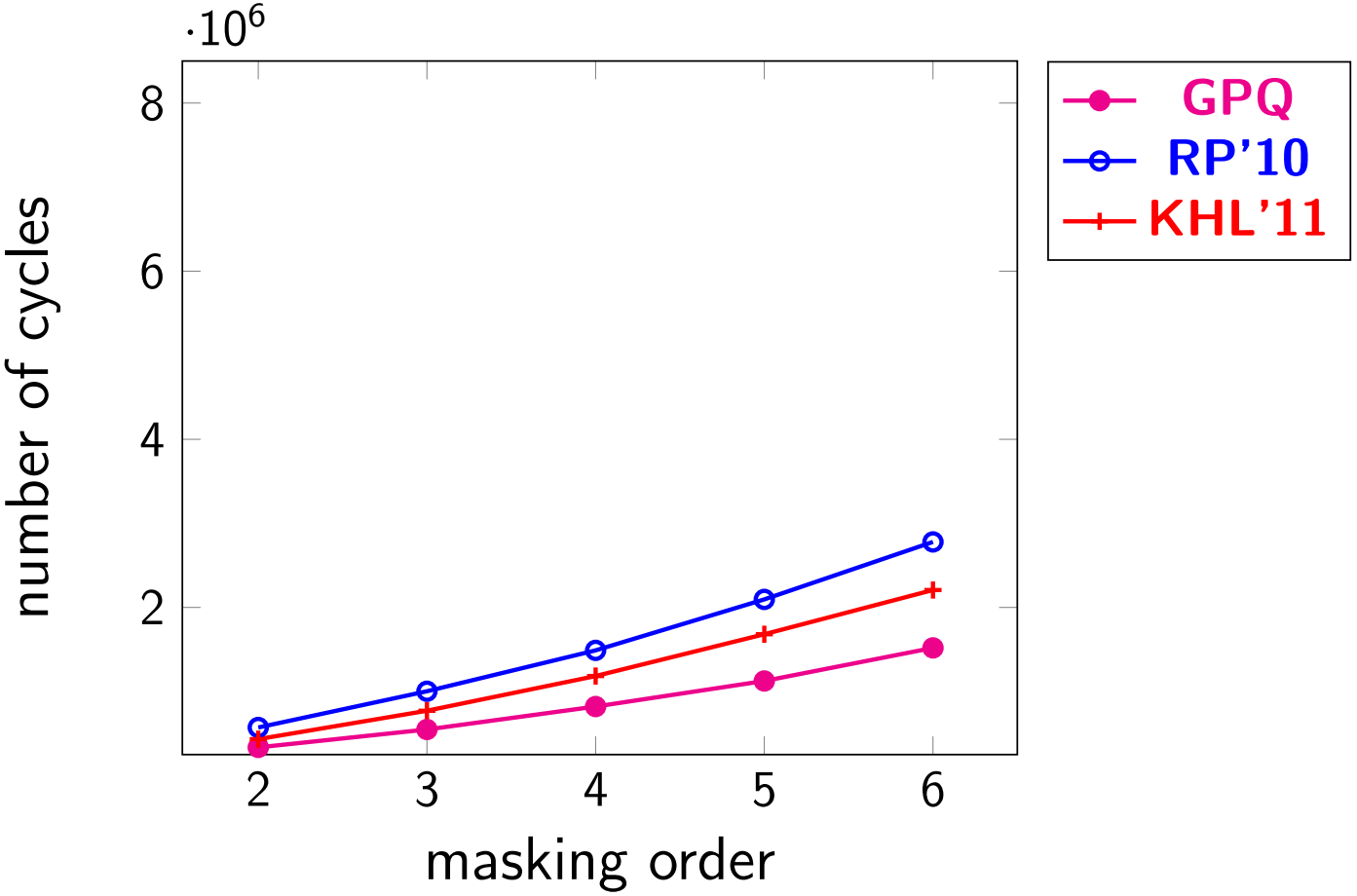
GPQ'11: The most efficient masking scheme for AES, since multiplicative and linear parts are well separated.

Existing Schemes



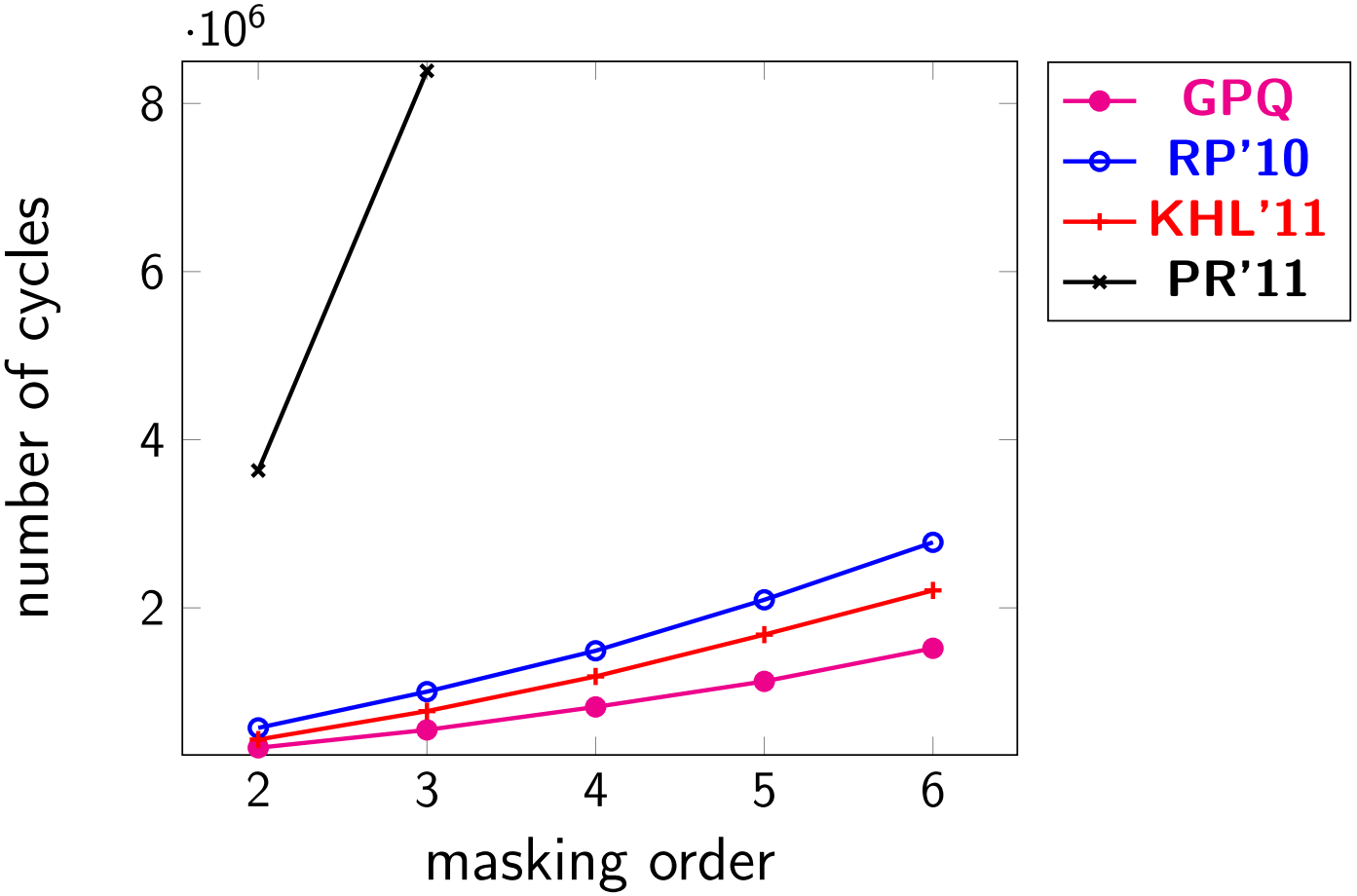
RP'10: secure multiplication with quadratic cost.

Existing Schemes



KHL'11 improvement of **RP'10** by using subfield, secure multiplication with quadratic cost.

Existing Schemes



PR'11 glitches-free solution, secure multiplication with cubic cost.

Outline

1. Unified comparison of existing schemes

2. Efficiency improvement with packed secret sharing

3. Randomness requirements and impact on efficiency of masking schemes

Polynomial Sharing

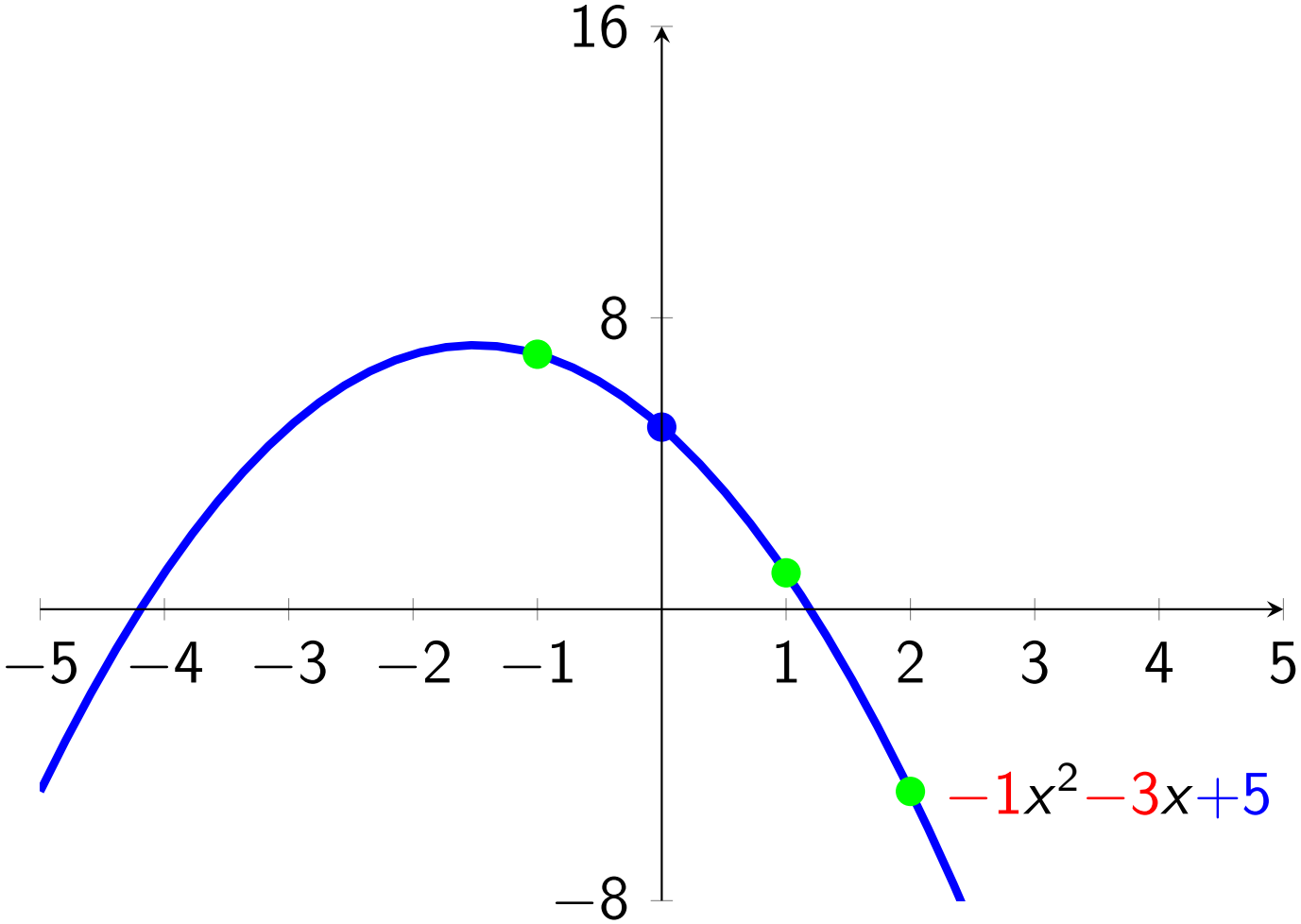
$$P(X) = S + r_1 X + \ldots r_d X^d$$

r_i random values.

$S = P(0)$.

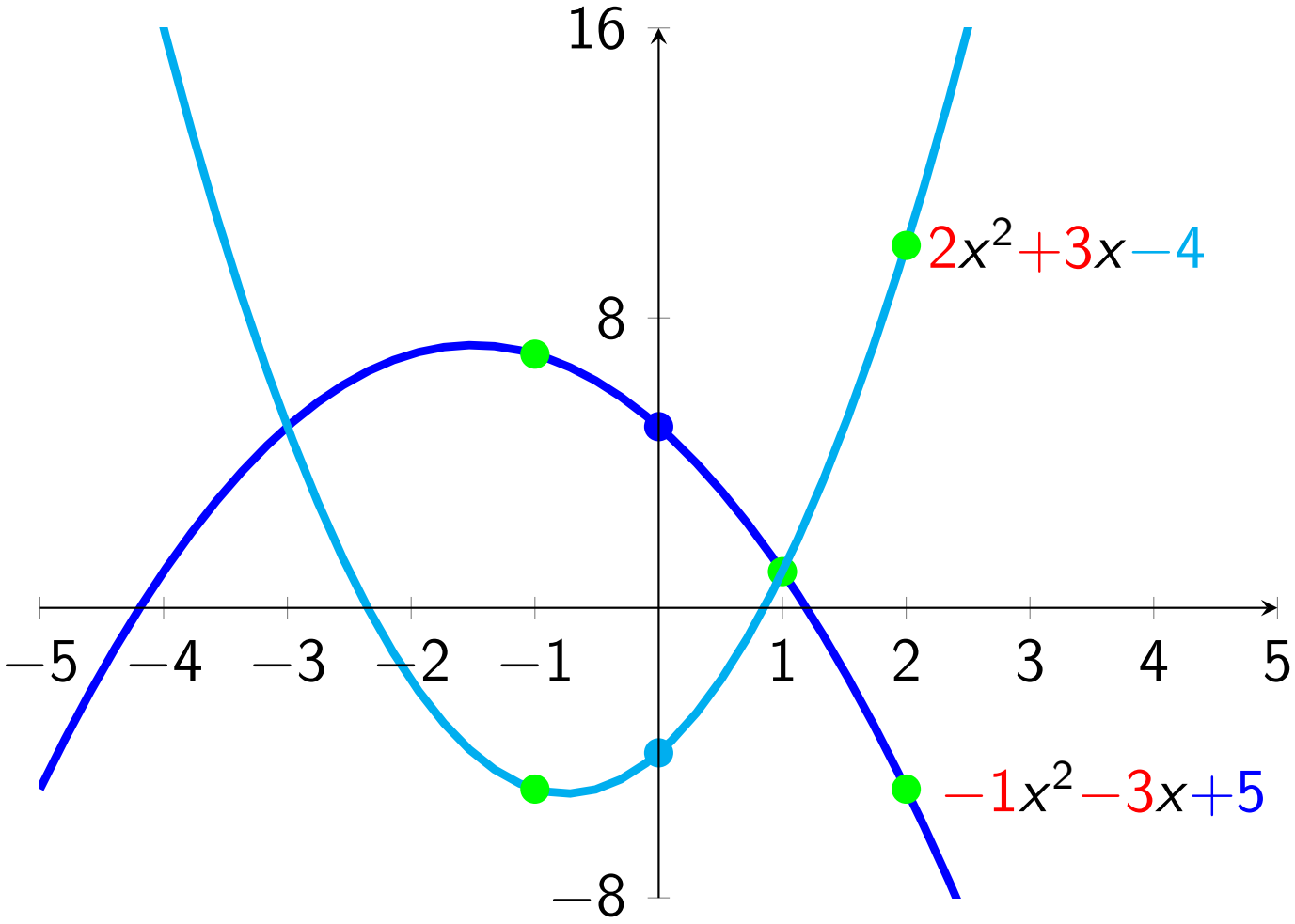
Shares: $P(t_i)$.

Interpolation



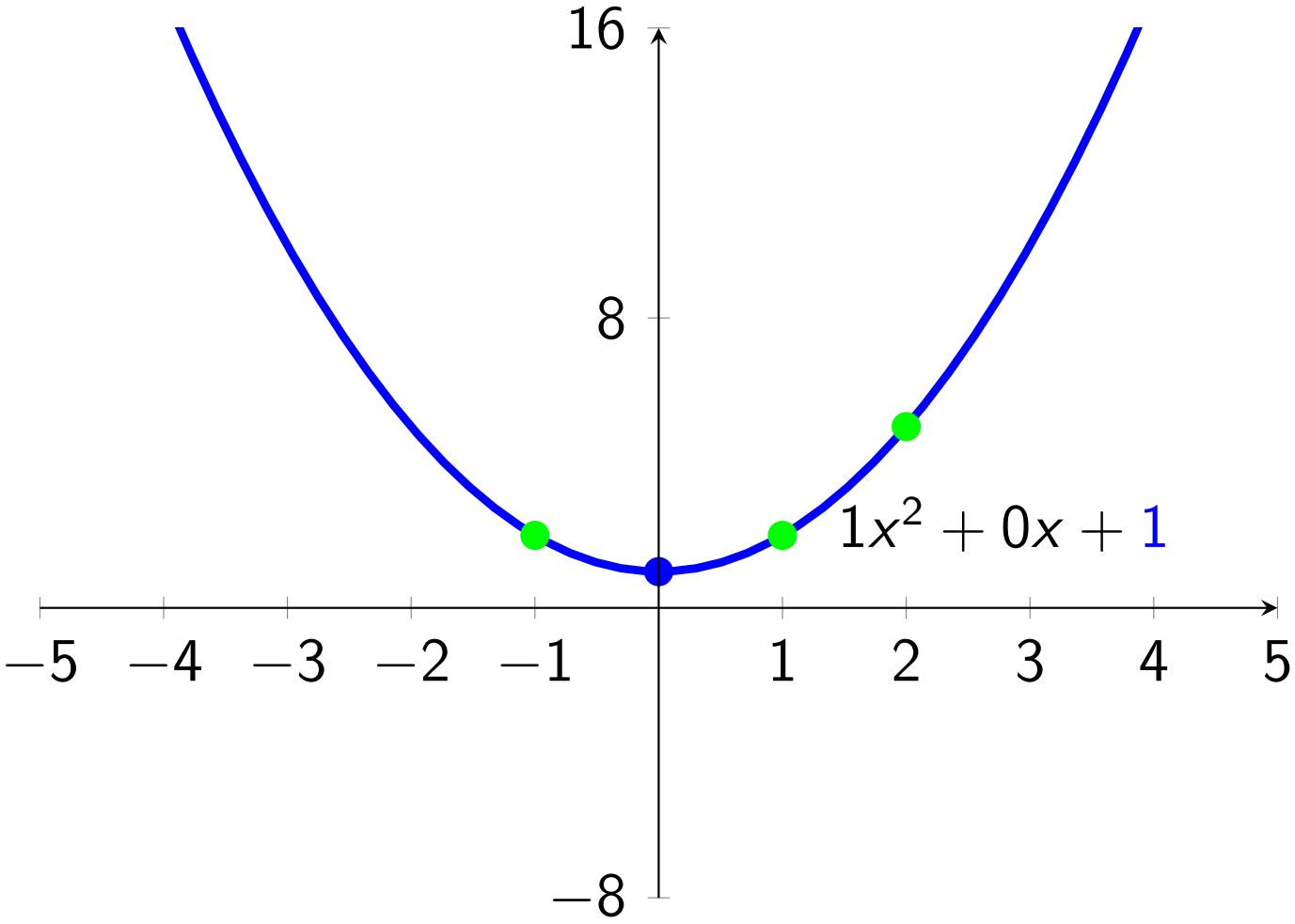
$d+1$ points are sufficient to recover the polynomial.

Addition



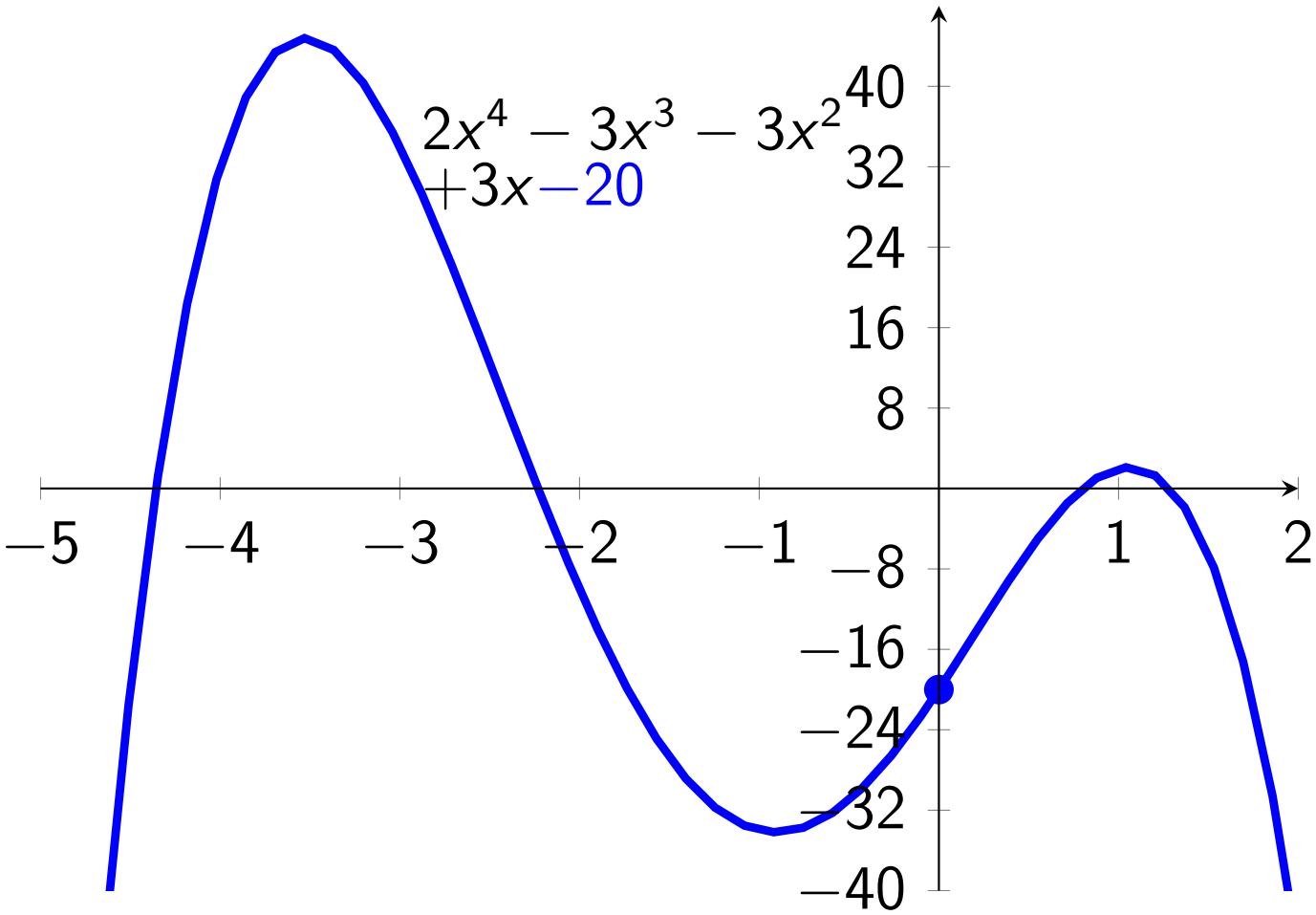
The sum of each couple of shares (located on the same point)

Addition



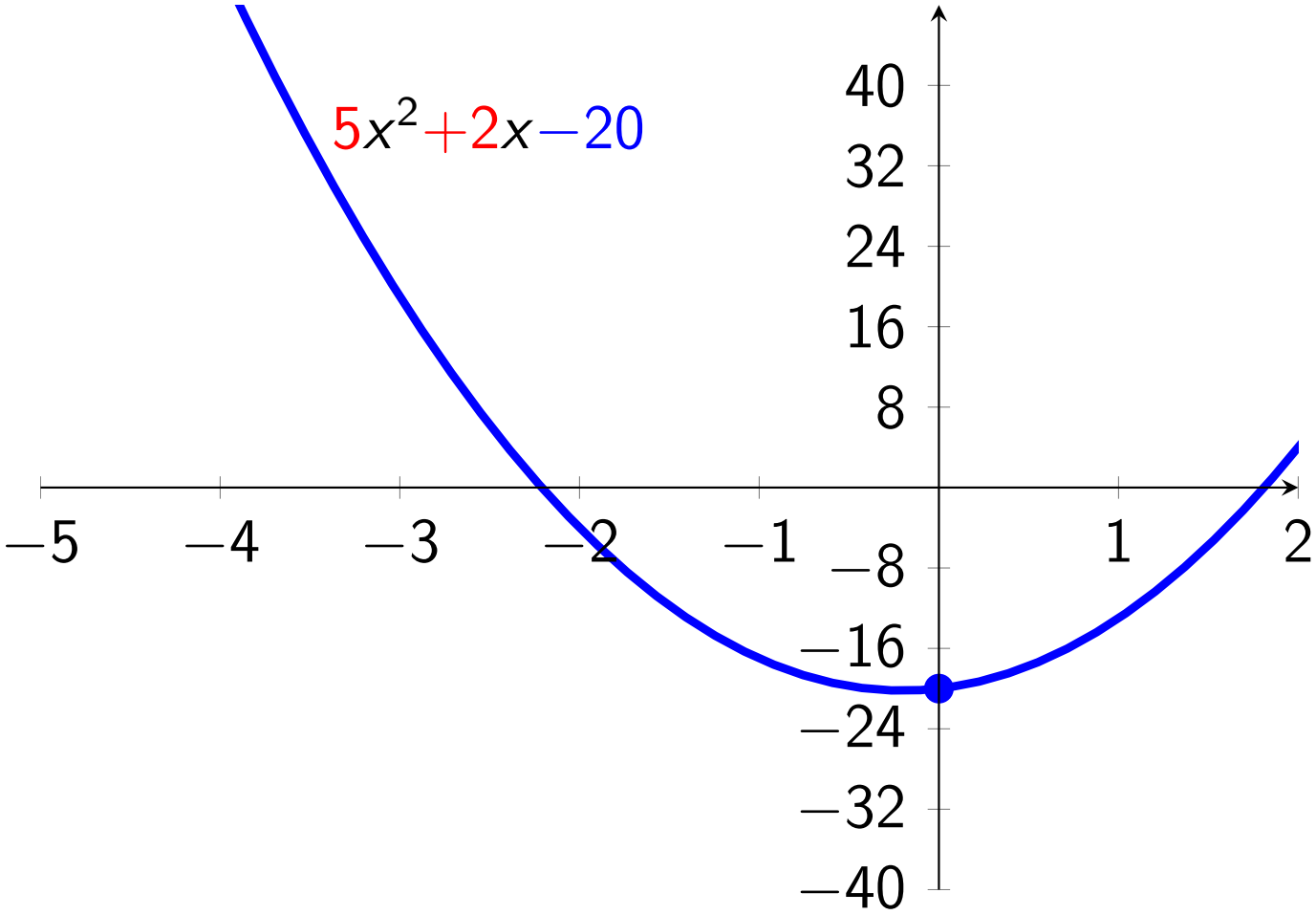
The sum of each couple of shares (located on the same point), is sufficient to sum the secrets.

Multiplication



To recover the secret we need $2d + 1$ points, since the degree of the polynomial product is $2d$.

Multiplication

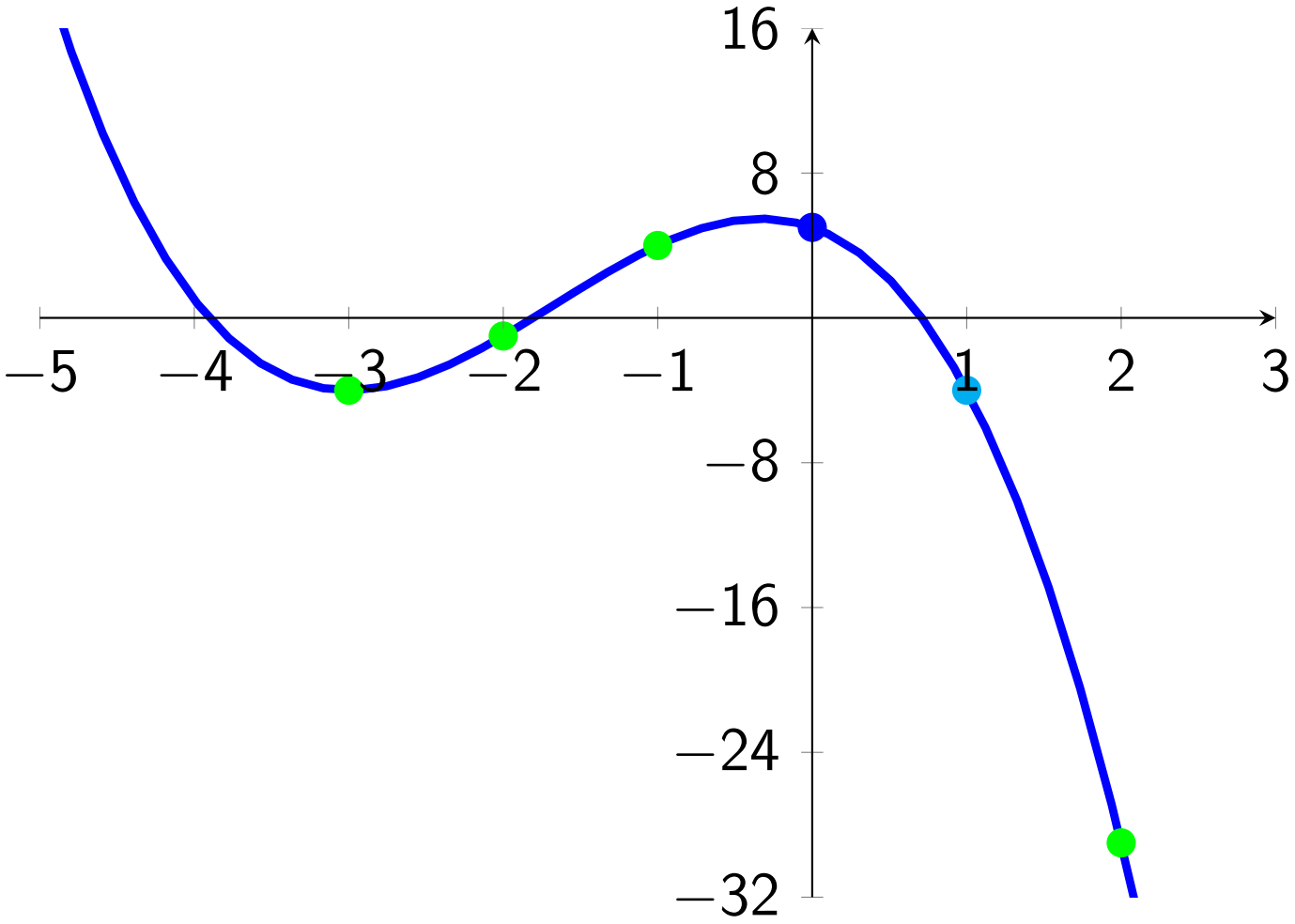


And a secure way to reduce the degree of the polynomial.

Motivation

- $2d + 1$ points of the polynomial are used for the shares, 1 point is used for the secret, all others are unused.
- Hide several, say t, secrets in the polynomial and have $2(d + t) - 1$ shares and keep the d-order masking.
- Secrets are hidden in different locations.

Packed Secret Polynomial



Intuition

Perform the computation on the secrets in "parallel" rather than sequentially.

Intuition

Perform the computation on the secrets in "parallel" rather than sequentially.

Let t be the number of secrets, packed secret sharing is interesting when:

$$Cost(\text{t packed}) < \text{t} \times Cost(\text{single secret})$$

Intuition

Perform the computation on the secrets in "parallel" rather than sequentially.

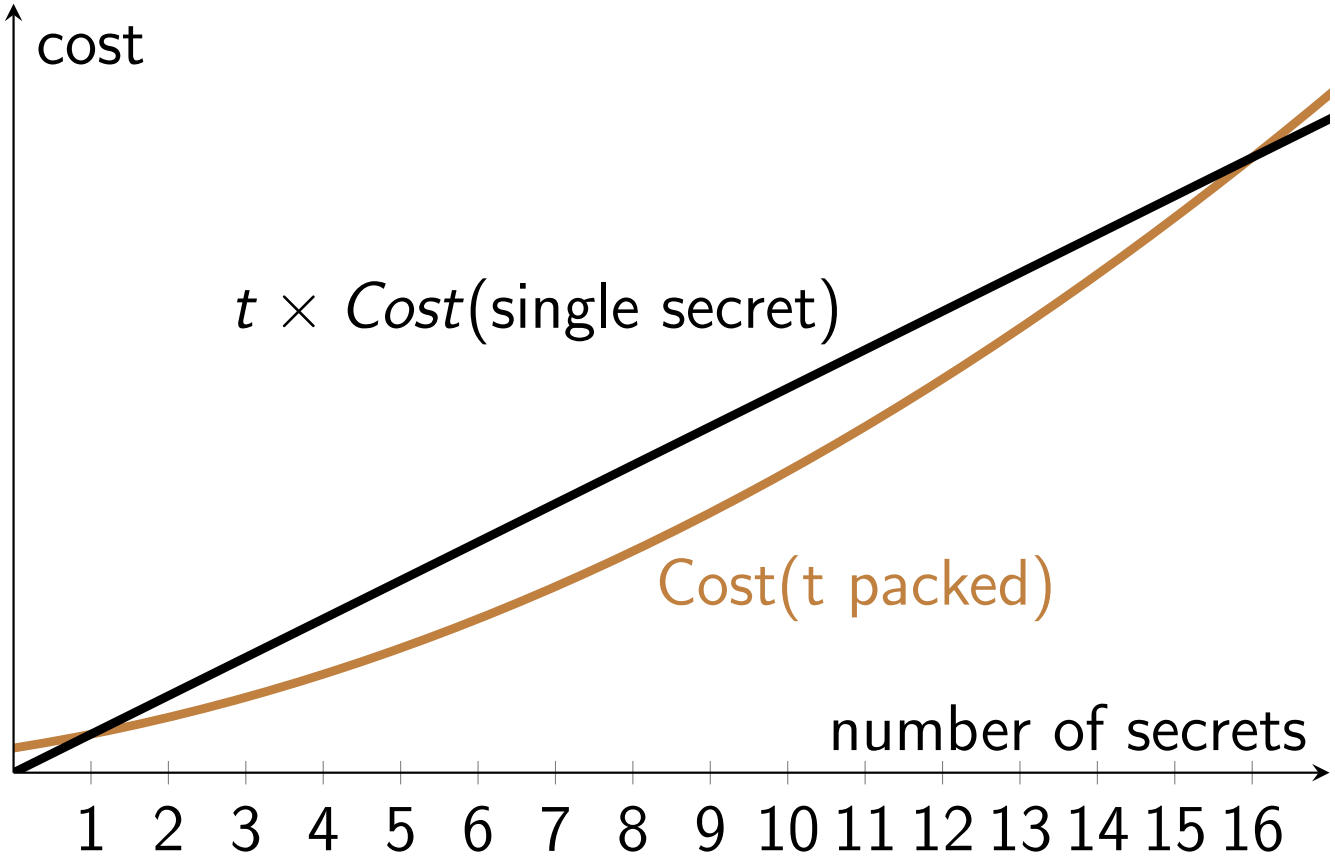
Let t be the number of secrets, packed secret sharing is interesting when:

$$Cost(\text{t packed}) < \text{t} \times Cost(\text{single secret})$$

Let d be the masking order and the cost of the algorithm be quadratic in the number of shares.

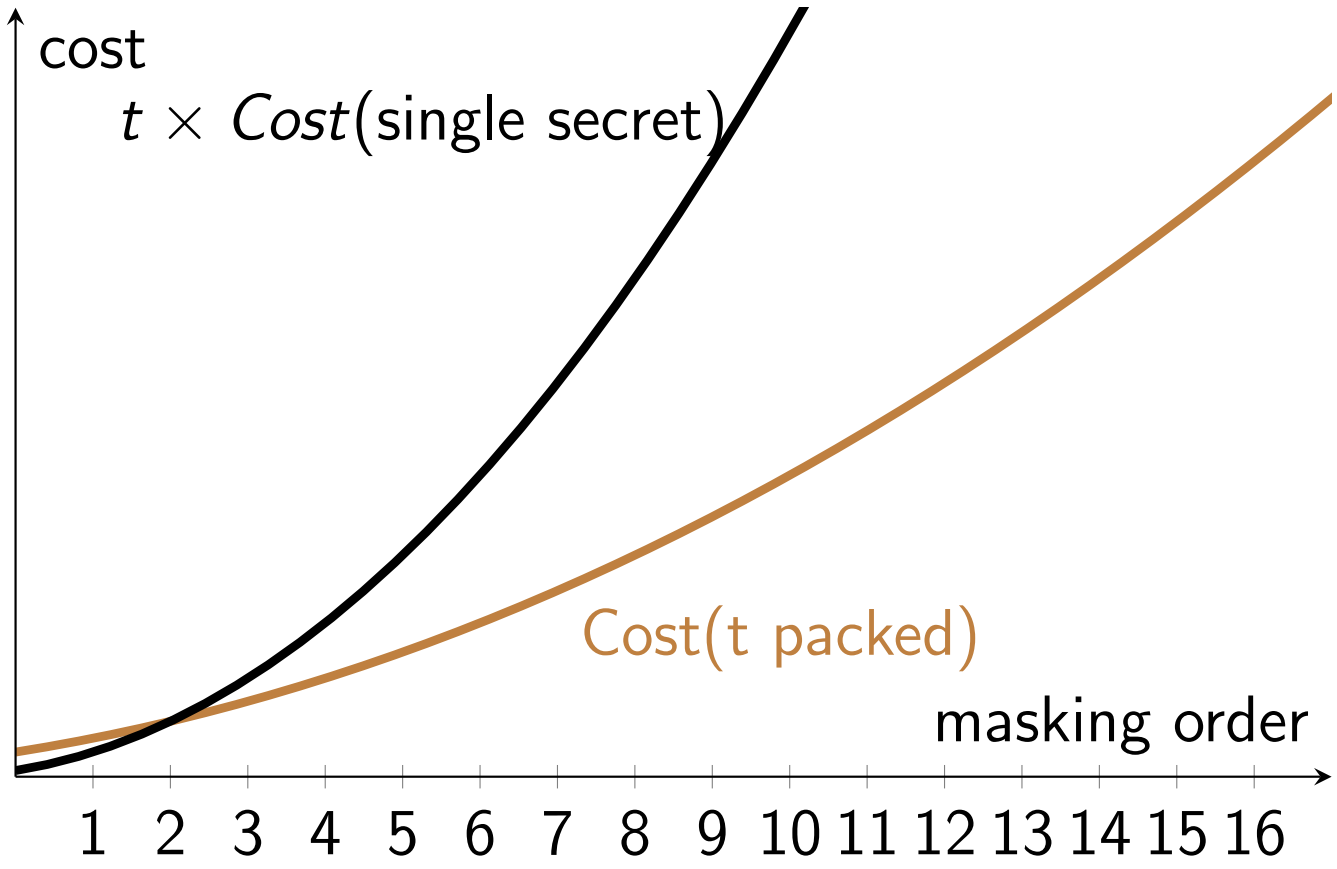
$$(t + d)^2 < t \times (d + 1)^2$$

Example: Fixed Masking Order, $d = 4$



For any fixed masking order, there exists an interval of number of secrets for which packing is interesting.

Example: Fixed Number of Secrets, $t = 4$



For any fixed number of secrets, the bigger the masking order is, the more interesting is the packing technique.

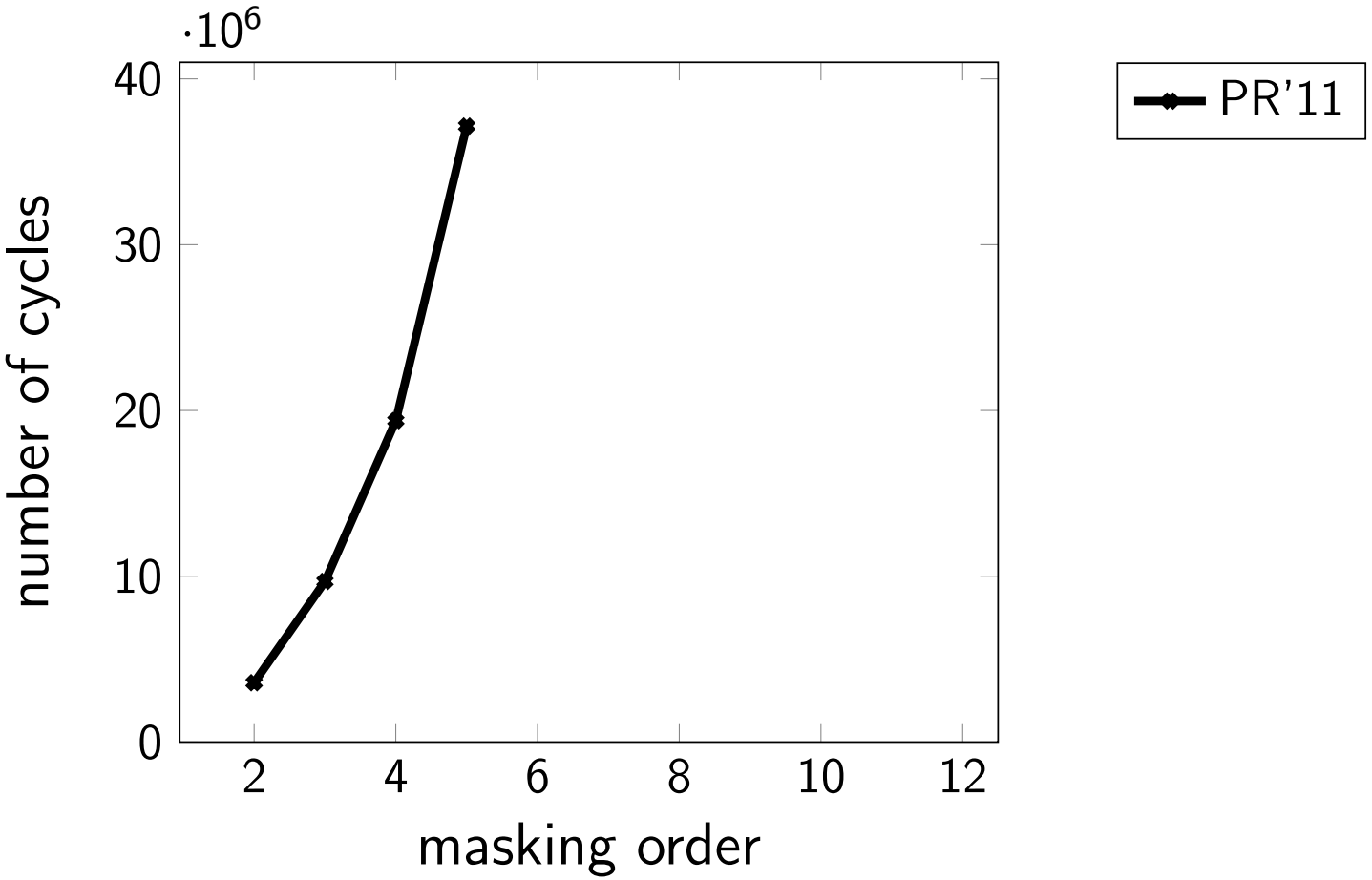
Issues

- **PR'11** multiplication is not suitable for packing, use Damgård et al. multiplication
- ShiftRows: how to move location of secrets
- MixColumns: how to combine sensitive values hidden in the same polynomial

Switch

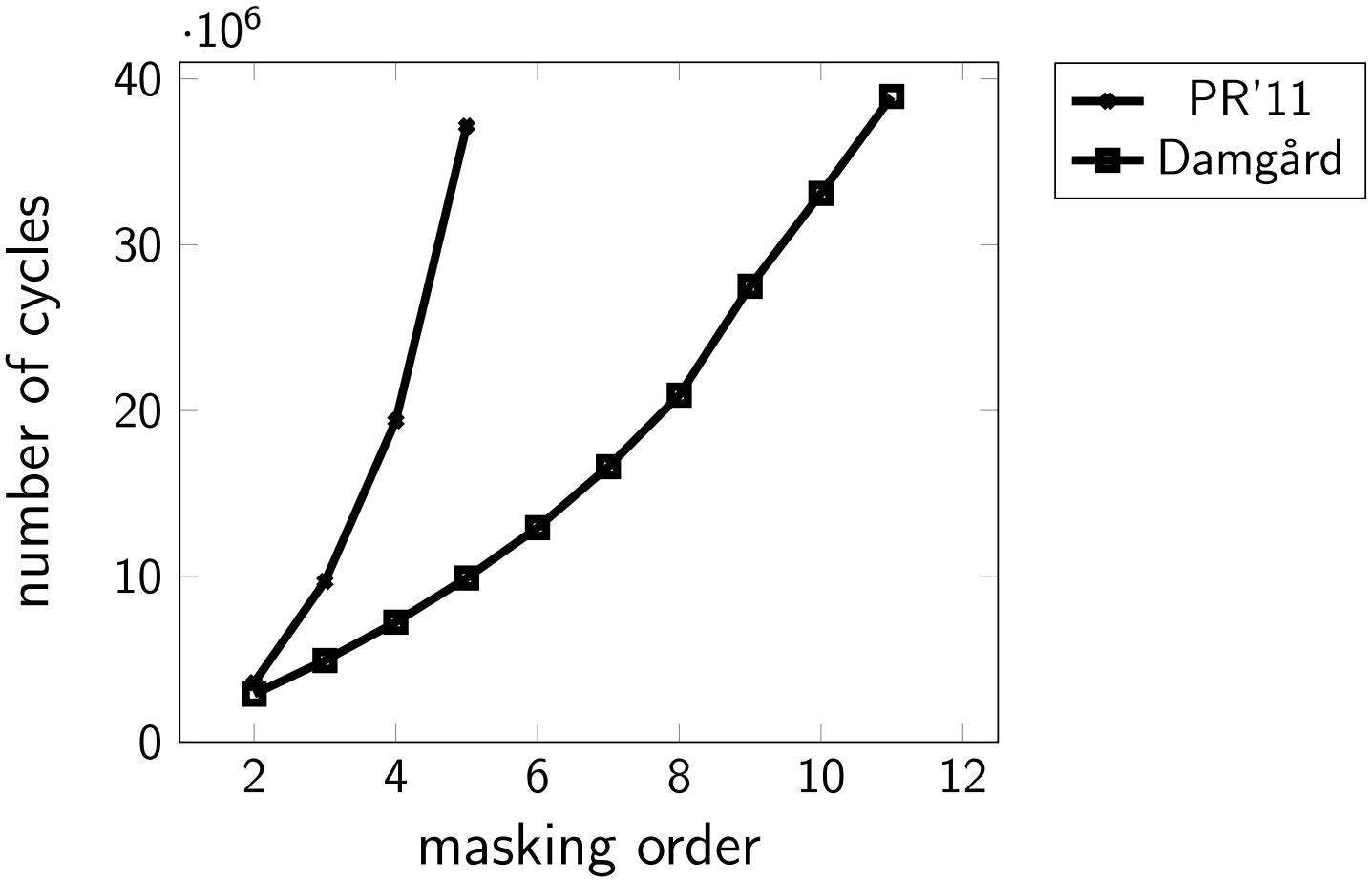
- Solution switch between packed and single secret polynomials
- Packed polynomials for the inversion
- Single secret polynomials for the linear parts

Packed Secrets Sharing vs. Single Secret



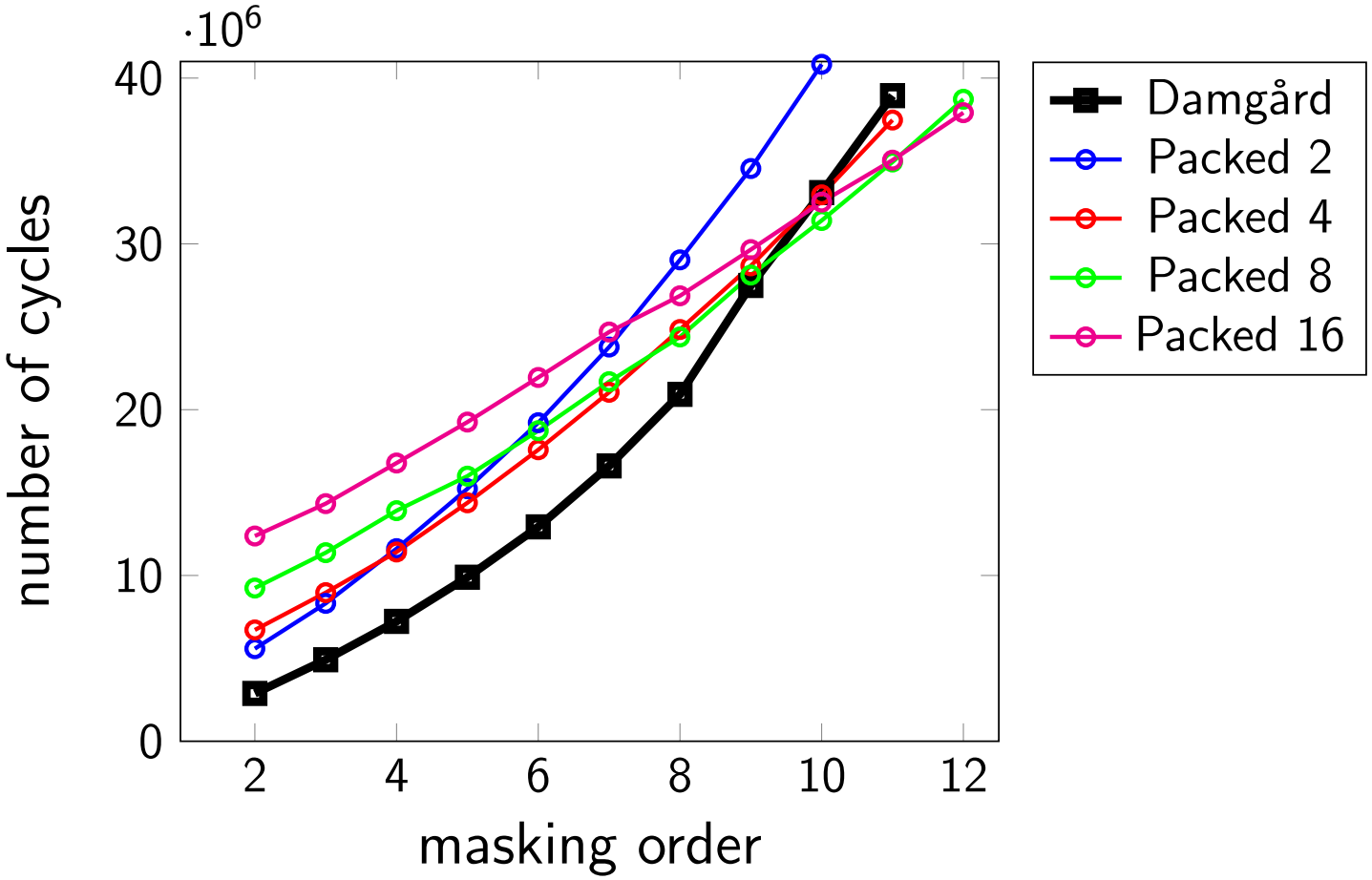
PR'11 cubic complexity.

Packed Secrets Sharing vs. Single Secret



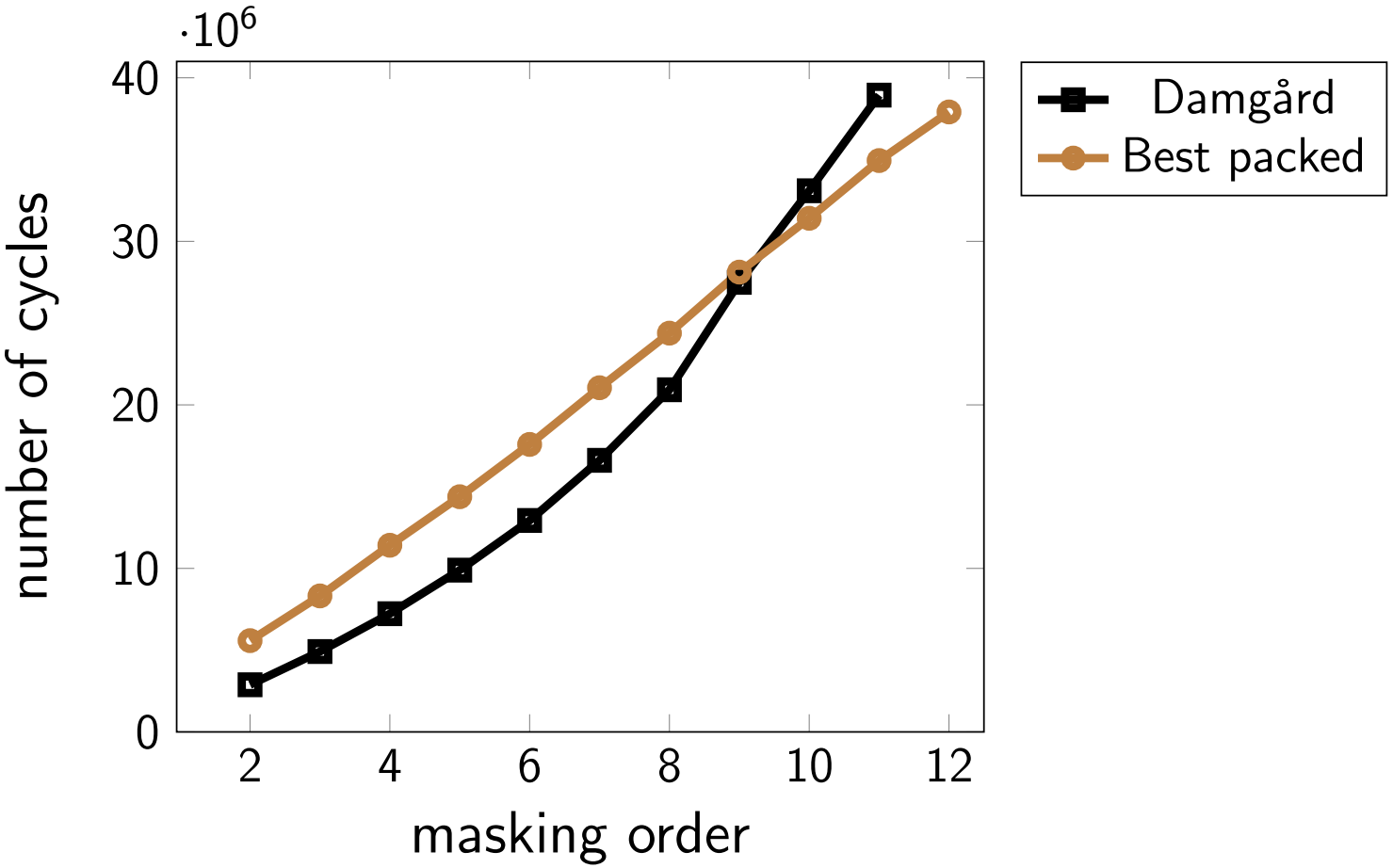
New multiplication method has quadratic complexity.

Packed Secrets Sharing vs. Single Secret



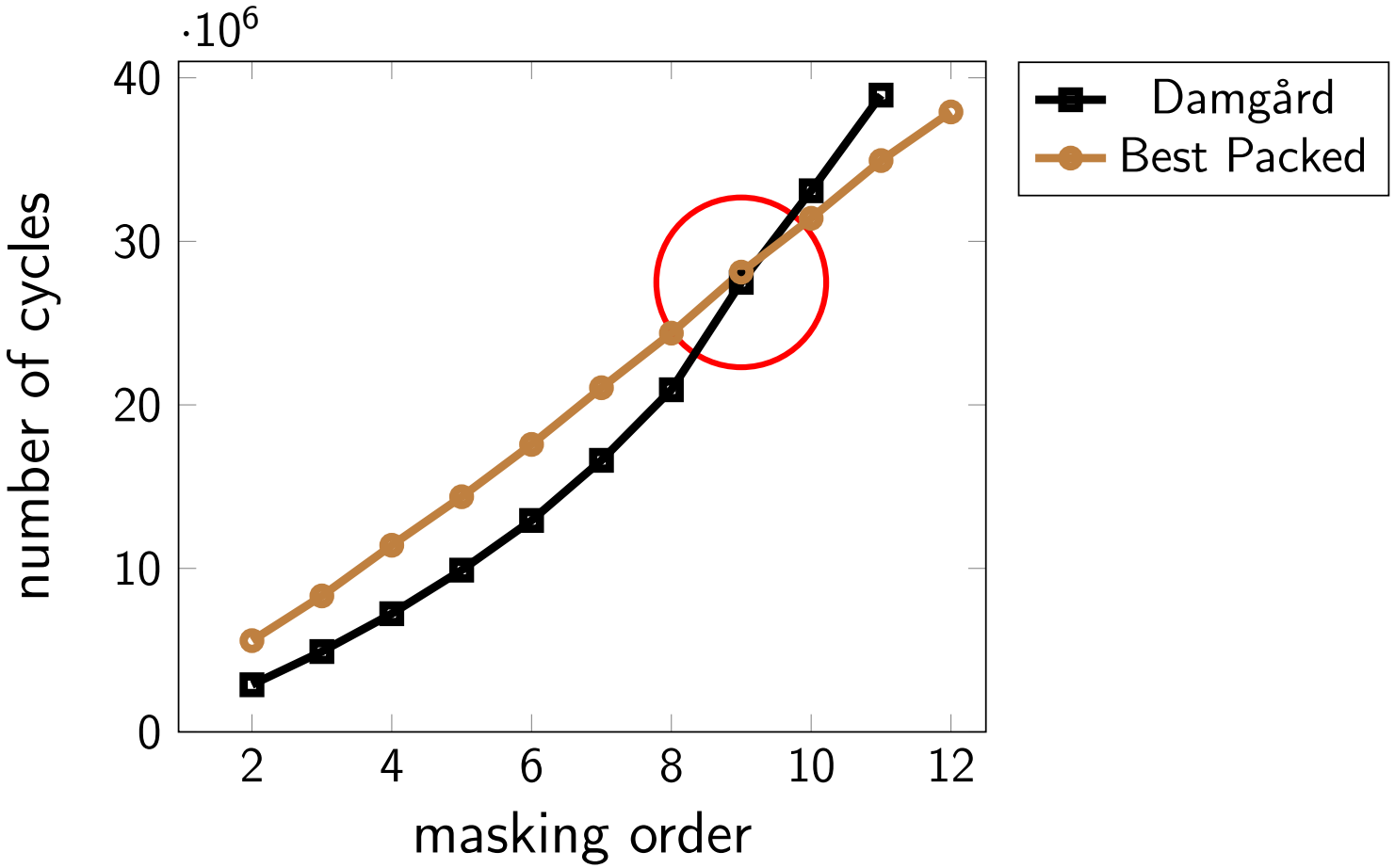
Packed for number of secrets 16's divisor. As expected the best t depends on d.

Packed Secrets Sharing vs. Single Secret



Minimum of packed secrets have quasi-linear complexity.

Packed Secrets Sharing vs. Single Secret



Cross around 10, unrealistic for contemporary devices.

Outline

1. Unified comparison of existing schemes

2. Efficiency improvement with packed secret sharing

3. Randomness requirements and impact on efficiency of masking schemes

Motivation

- In the previous experiment, we considered "free" random generator.
- Proof of security requires uniform randomness.
- In embedded systems, uniform randomness is expensive.

Intuition

What happen when of using non uniform randomness?

Intuition

What happen when of using non uniform randomness?

- predictable randomness, such as counter, such an imperfection of the randomness is that for low noise levels, all the masks will be recovered with probability one.

Intuition

What happen when of using non uniform randomness?

- predictable randomness, such as counter, such an imperfection of the randomness is that for low noise levels, all the masks will be recovered with probability one.
- slightly biased randomness, such biases directly create a lower-order weakness.

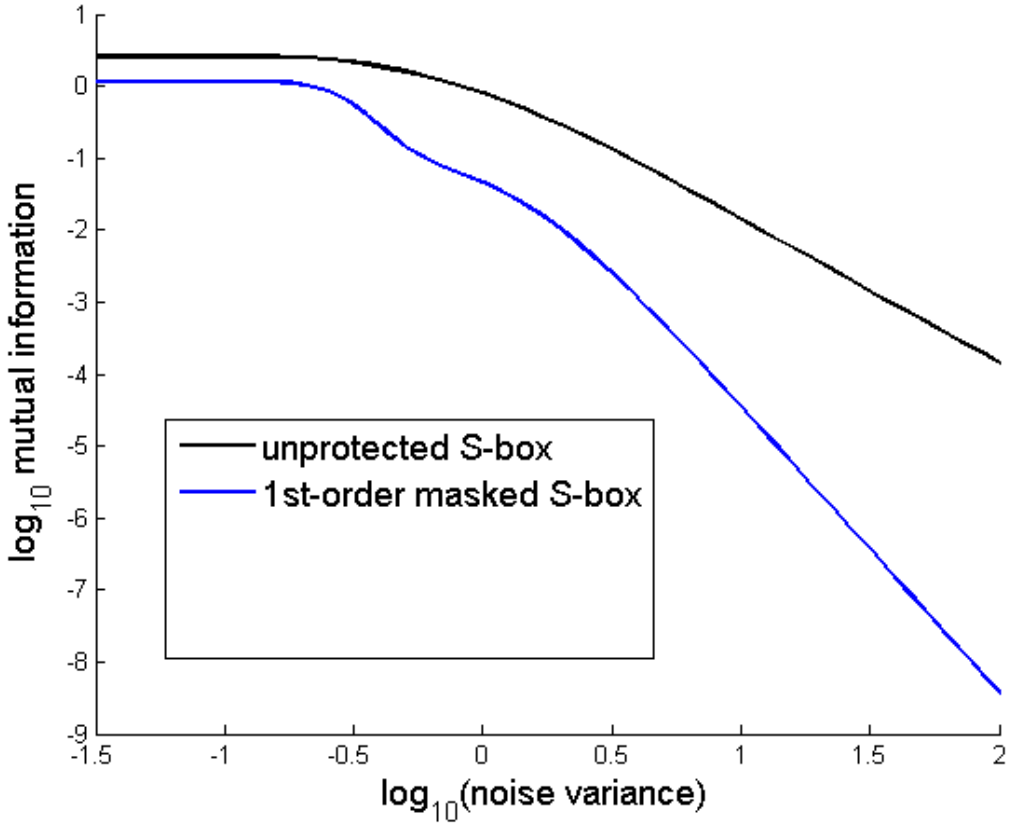
Intuition

What happen when of using non uniform randomness?

- predictable randomness, such as counter, such an imperfection of the randomness is that for low noise levels, all the masks will be recovered with probability one.
- slightly biased randomness, such biases directly create a lower-order weakness.

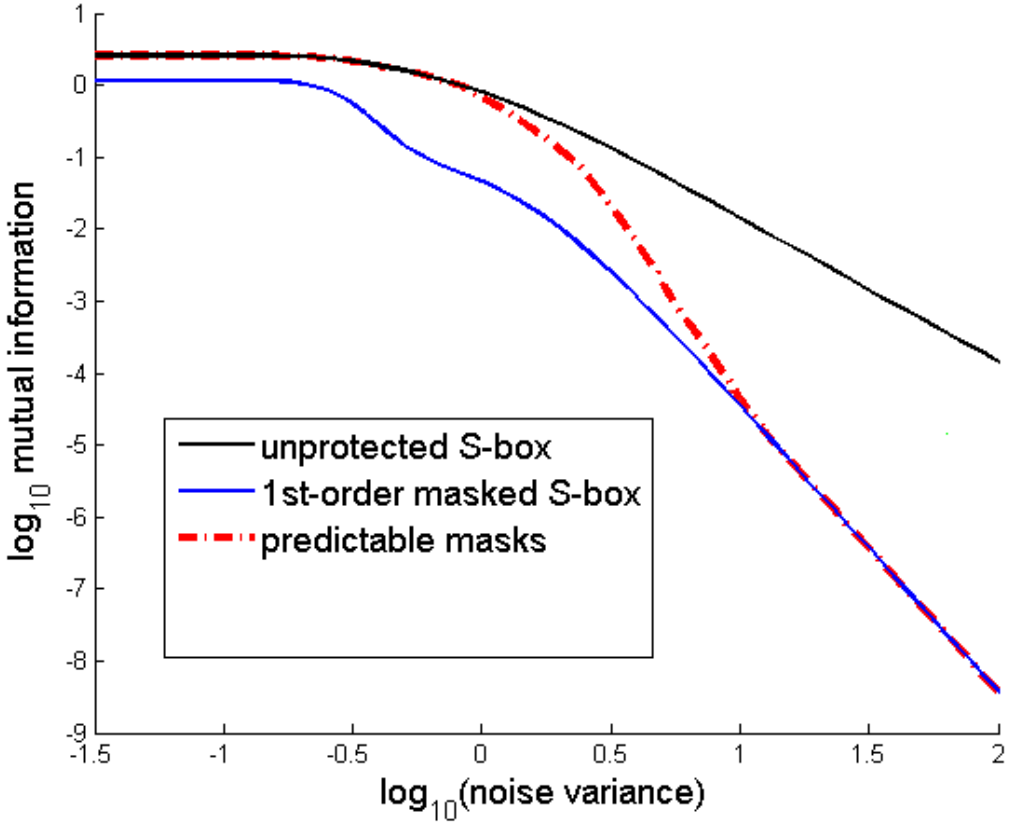
$\Rightarrow$ Compute mutual information between subkey and leakage.

Non Uniform Randomness



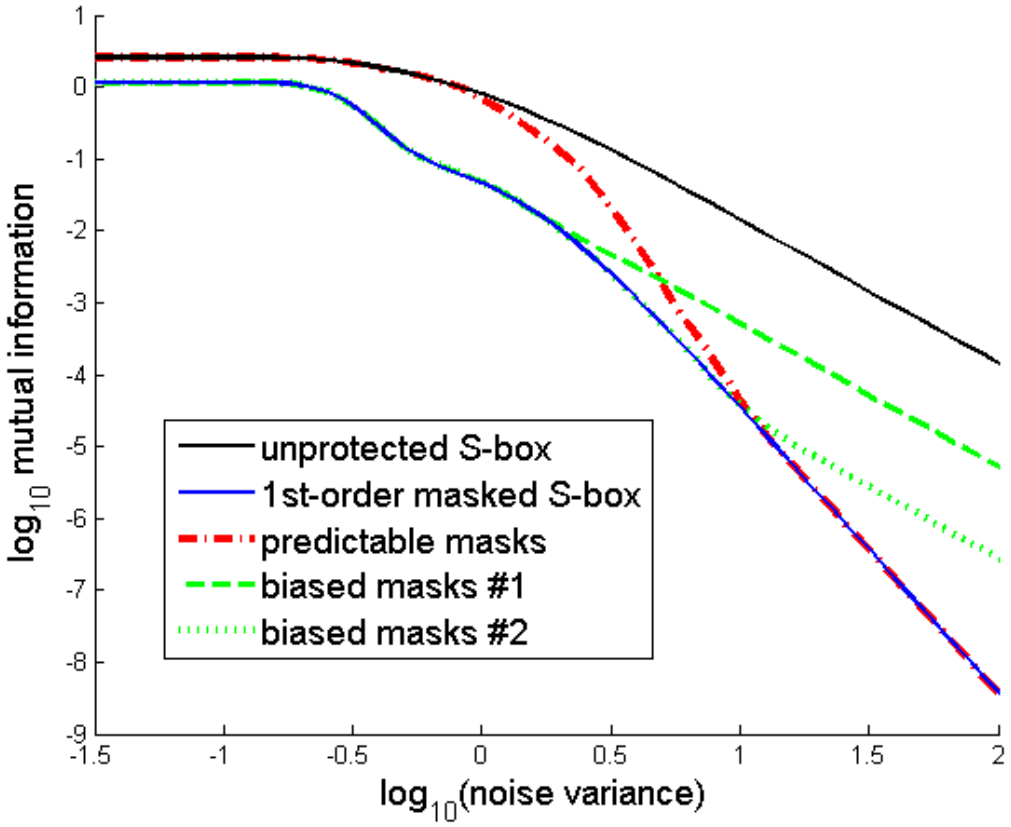
Black curve unprotected case, blue curve curve expected from first order masking.

Non Uniform Randomness



Red curve predictable case, combined attacks, reduce the noise, recover the masks values, similar to unprotected.

Non Uniform Randomness



Green curves biased generator, first order leaks, like zero value issue.

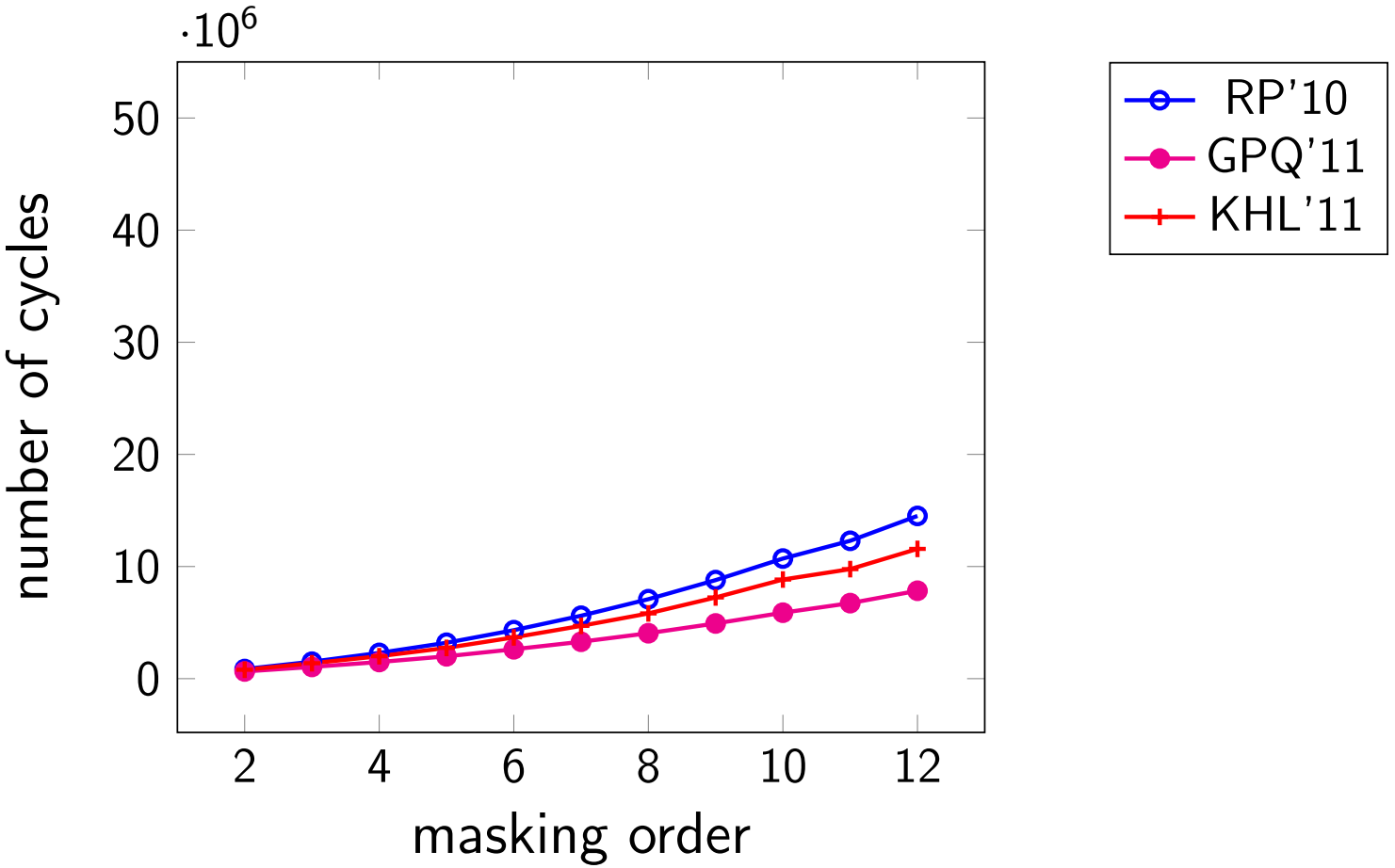
Cost for the Implementation?

- How to produce uniform random values:
  - wait values from a TRNG
  - few rounds of a good permutation
  - hash function
  - ...

Cost for the Implementation?

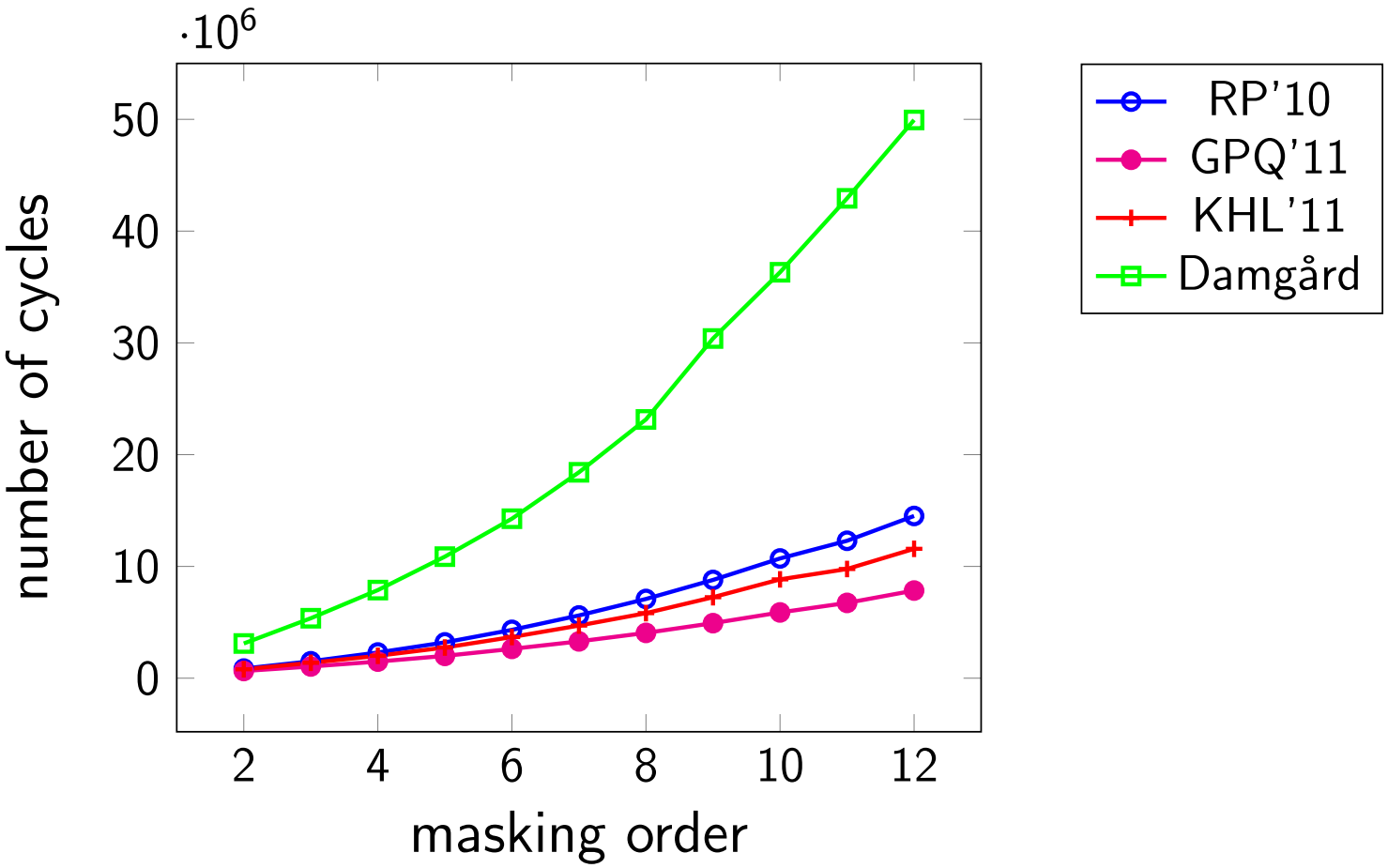
- How to produce uniform random values:
  - wait values from a TRNG
  - few rounds of a good permutation
  - hash function
  - ...
- Cost: around 10 clock cycles per random byte.

Randomness Impact



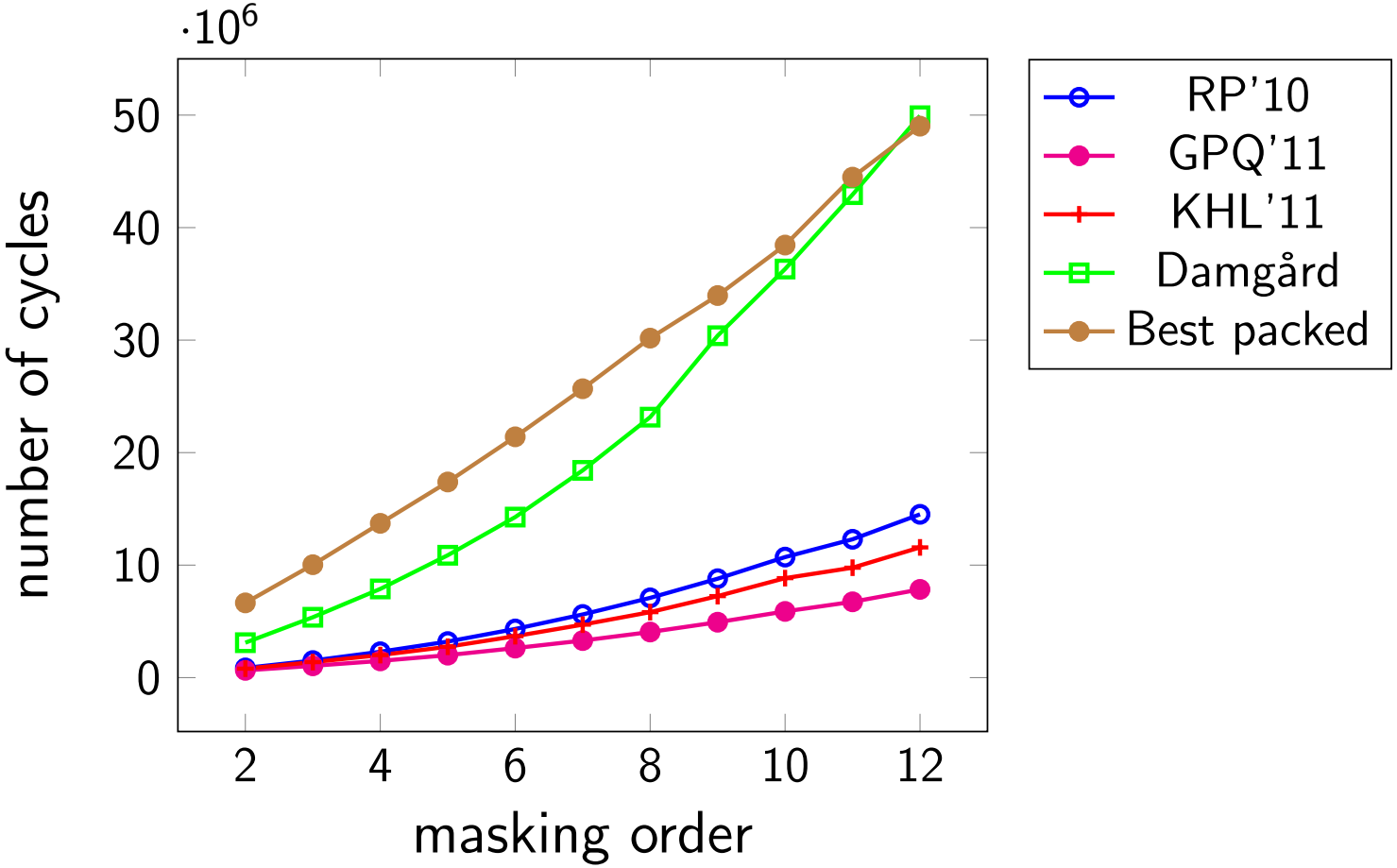
The order of schemes does not change. We just add a little overhead on performances $< 5/4$.

Randomness Impact



The MPC stays far from masking, even with quadratic multiplication.

Randomness Impact



Packed technique is interesting shortly after. Due to the switch between packed and single secret polynomials that uses lot of randomness.

Conclusion

- Unified comparison of masking scheme $\Rightarrow$ allows designers to choose a scheme in function of security and performance.

Conclusion

- Unified comparison of masking scheme $\Rightarrow$ allows designers to choose a scheme in function of security and performance.
- Packing technique theoretically interesting, but concrete gains only appear for large order (maybe interesting in the longer term).

Conclusion

- Unified comparison of masking scheme $\Rightarrow$ allows designers to choose a scheme in function of security and performance.
- Packing technique theoretically interesting, but concrete gains only appear for large order (maybe interesting in the longer term).
- Randomness is not the most expensive part of masked implementation, but is not negligible.