Short Memory Method on Koblitz Curves

Katsuyuki Okeya

Tsuyoshi Takagi

Camille Vuillaume

Outline

Elliptic curves

- Interest of ECC
- Binary Method
- NAF Method

Koblitz curves

- Binary Curves
- τ Expansions
- Binary vs. Koblitz

Short Memory

- Normal vs. Polynomial Basis
- Short Memory on Normal Basis
- Short Memory on Polynomial Basis

Outline

Elliptic curves

- Interest of ECC
- Binary Method
- NAF Method

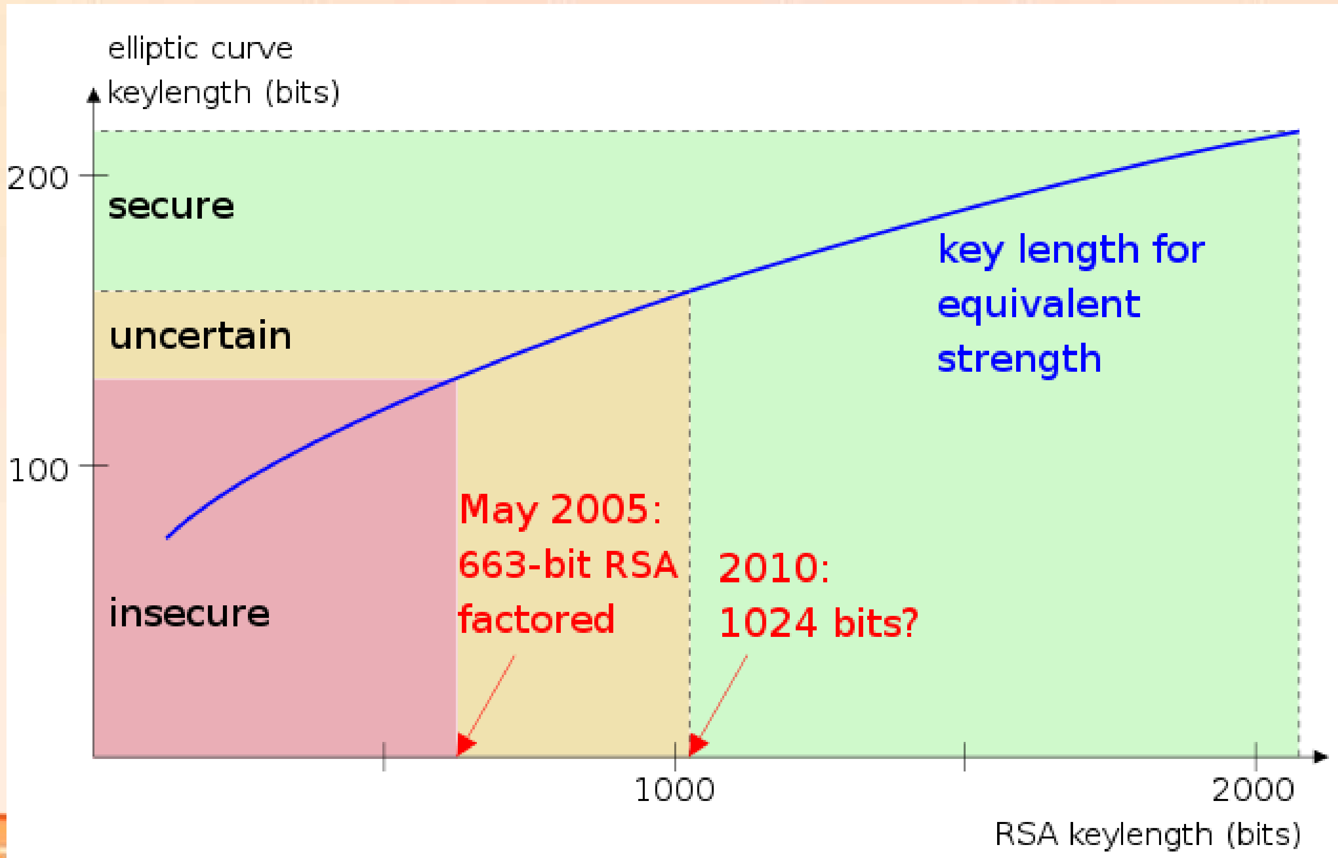
Koblitz curves

Short Memory

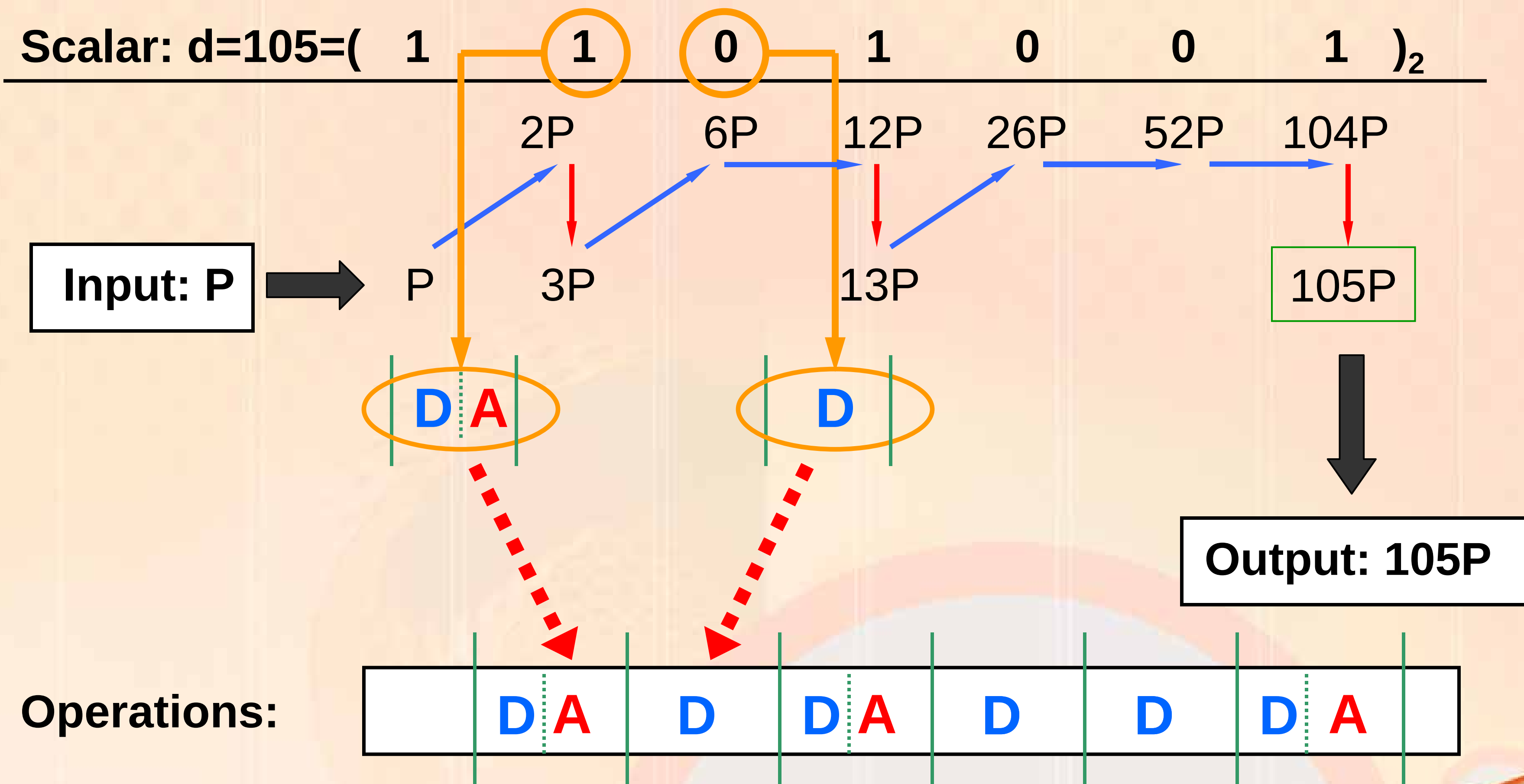
RSA vs. Elliptic Curves

Speed — ECC are 30 times faster than RSA…

Memory — …and require 6.5 less memory

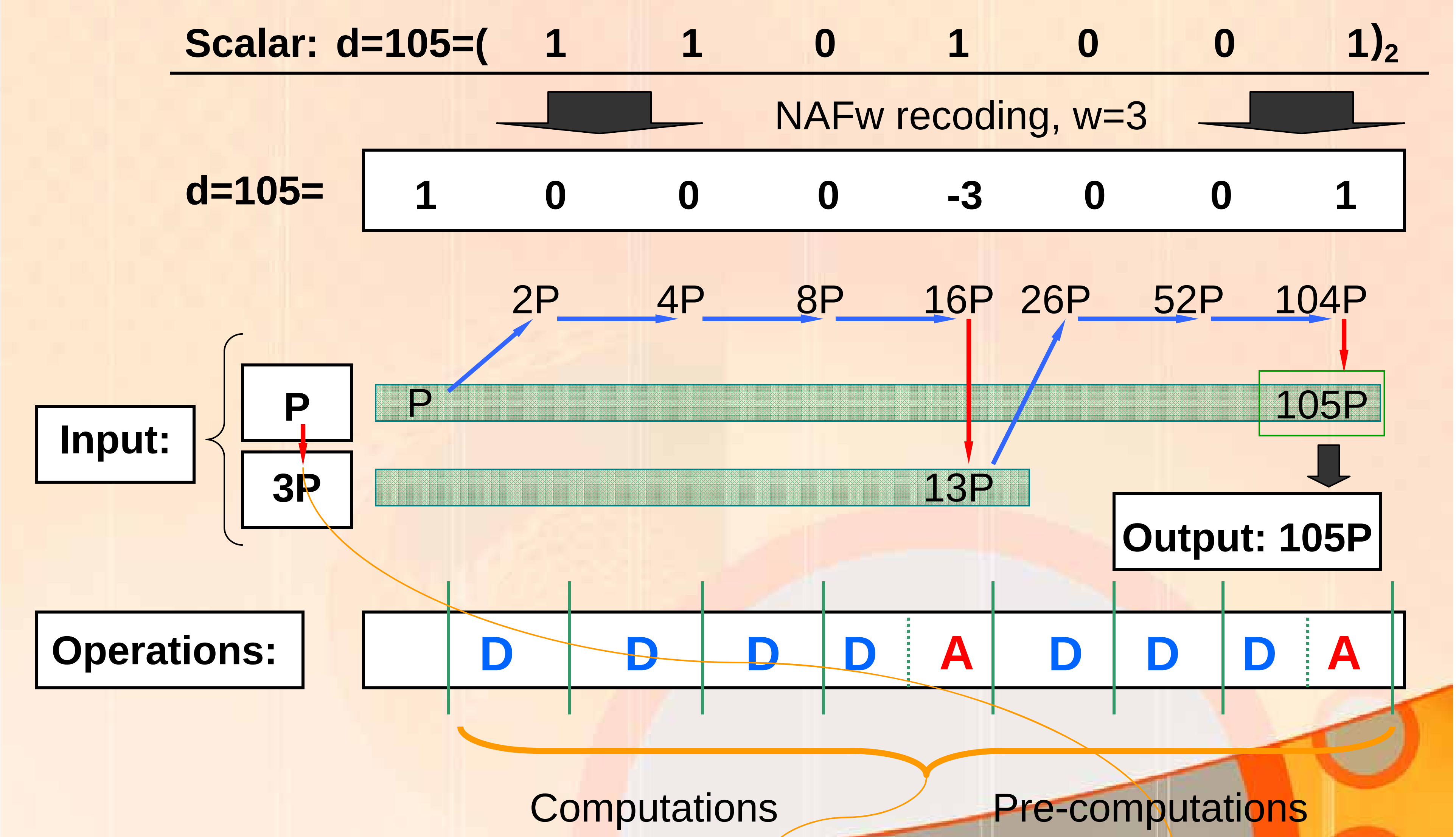


Binary Method



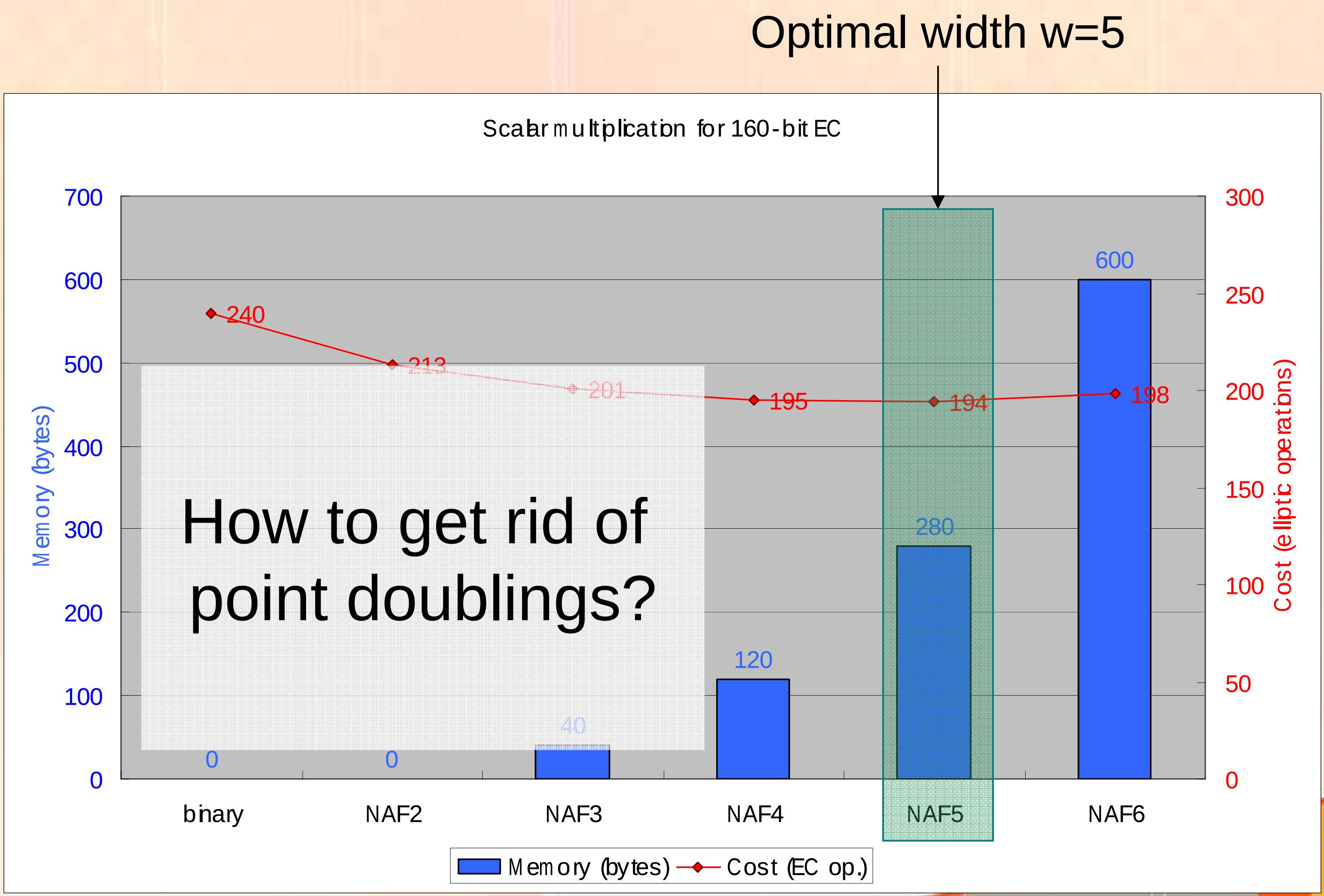
On average: (n-1)D+n/2A

NAFw Method



On average: $nD + n/(w+1)A + (2^{w-2}-1)A$

Comparison – NAFw Method



Maximal speed-up: less than 20% ☹

Outline

Elliptic curves

Koblitz curves

Binary Curves

τ Expansions

Binary vs. Koblitz

Short Memory

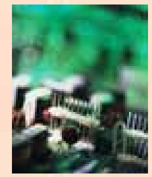
Binary Curves

Coprocessor-less architecture

Lower production cost, cheaper design ☺

Can use AES acceleration hardware

Re-use existing design ☺



Binary Curves

$y^2\ xy=x^3+ax^2+b,\ a,b\in F_{2m}$

Koblitz Curves

$y^2+xy=x^3+ax+1,\ a=\{0,1\}$

Slow without AES hardware or fast processor

With coprocessor, prime curves are faster ☹

Well-suited for mobile phone CPU (32-bit RISC)

Faster than general binary curves ☺☺

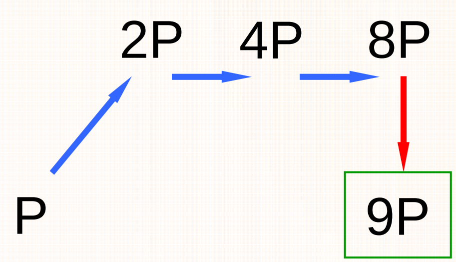
Binary and τ-Expansions

Binary curves

P = (x,y) → 2P = (x',y')

= $1*2^3$ + $1*2^0$

d=9=(1 0 0 1)$_2$



Koblitz curves

P = (x,y) → 2P = (x',y')

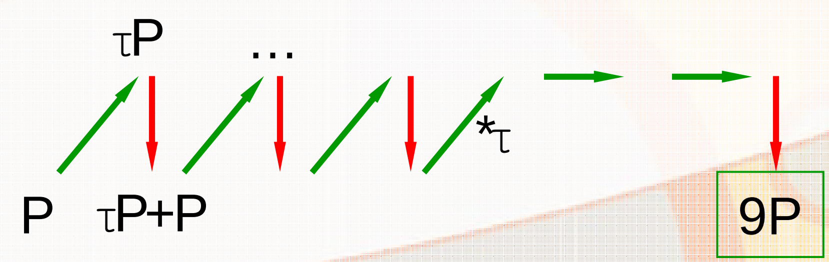
P = (x,y) → τP = (x²,y²) 10x faster

$2P = -\tau^2 P + \mu\tau P$

$\tau = (\mu + \sqrt{-1})/7$

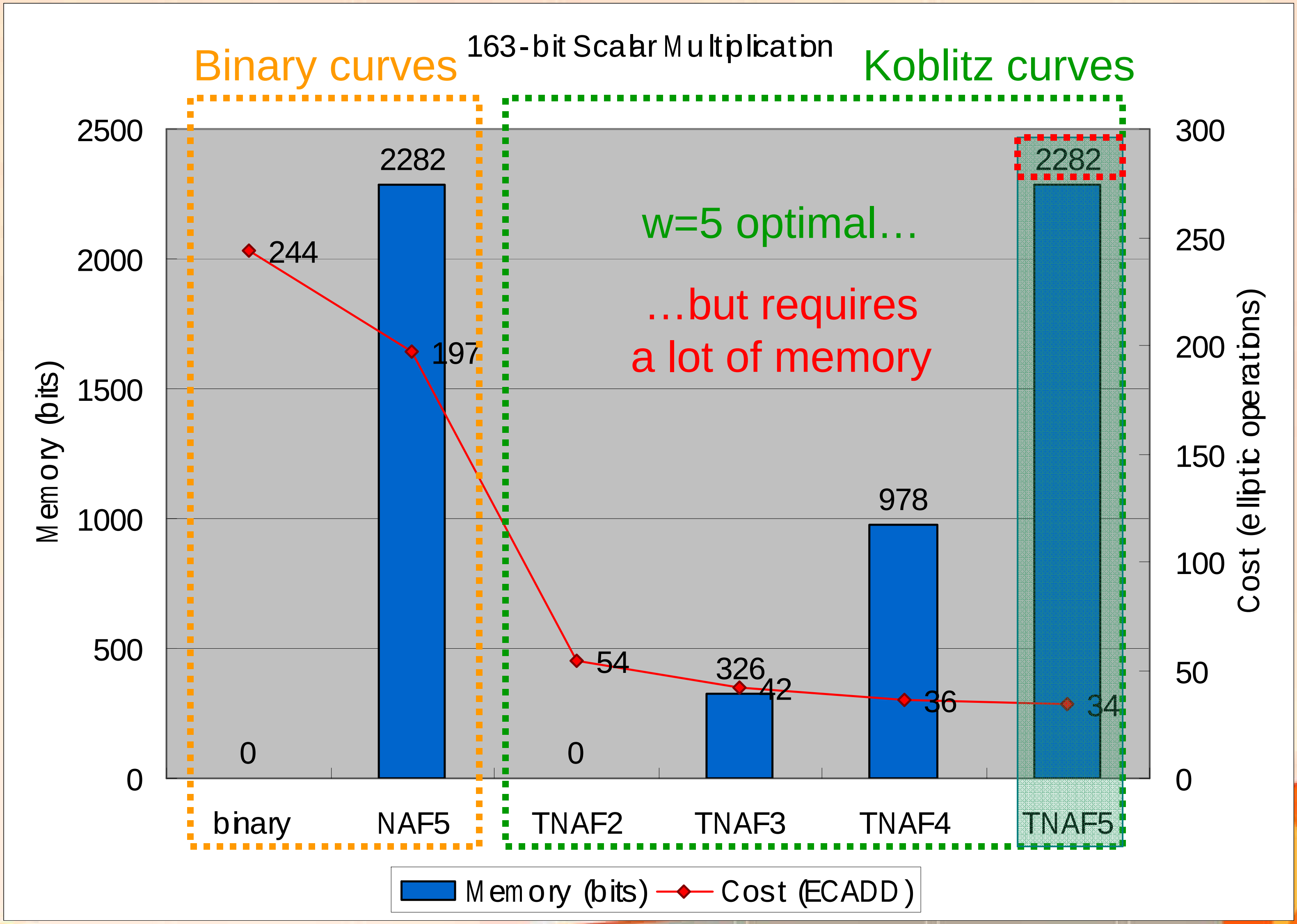
= $1*\tau^6 + 1*\tau^5 + 1*\tau^4 + 1*\tau^3$ + $1*\tau^0$

d=9=(1 1 1 1 0 0 1)$_\tau$



τ-and-add: no point doubling

Binary vs. Koblitz Curves



Outline

Elliptic curves

Koblitz curves

Short Memory

- Normal vs. Polynomial Basis
- Short Memory on Normal Basis
- Short Memory with Mixed Bases

Normal vs. Polynomial Basis

Polynomial Basis	Normal Basis
$p=p_{m-1}X^{m-1}+\ldots+p_1X+p_0$, where $p_i \in \{0,1\}$	$b=b_0\beta_0+b_1\beta_1+\ldots+b_{m-1}\beta_{m-1}$, where $\beta_i^2=\beta_{i+1}$ and $\beta_{m-1}^2=\beta_0$
Fast reduction with trinomials or pentanom...	No r... ☺
Fast...	☹☹
Fa...	☺☺
S...	...Hardware

$$b=b_0\beta_0+b_1\beta_1+\ldots+b_{m-2}\beta_{m-2}+b_{m-1}\beta_{m-1}$$

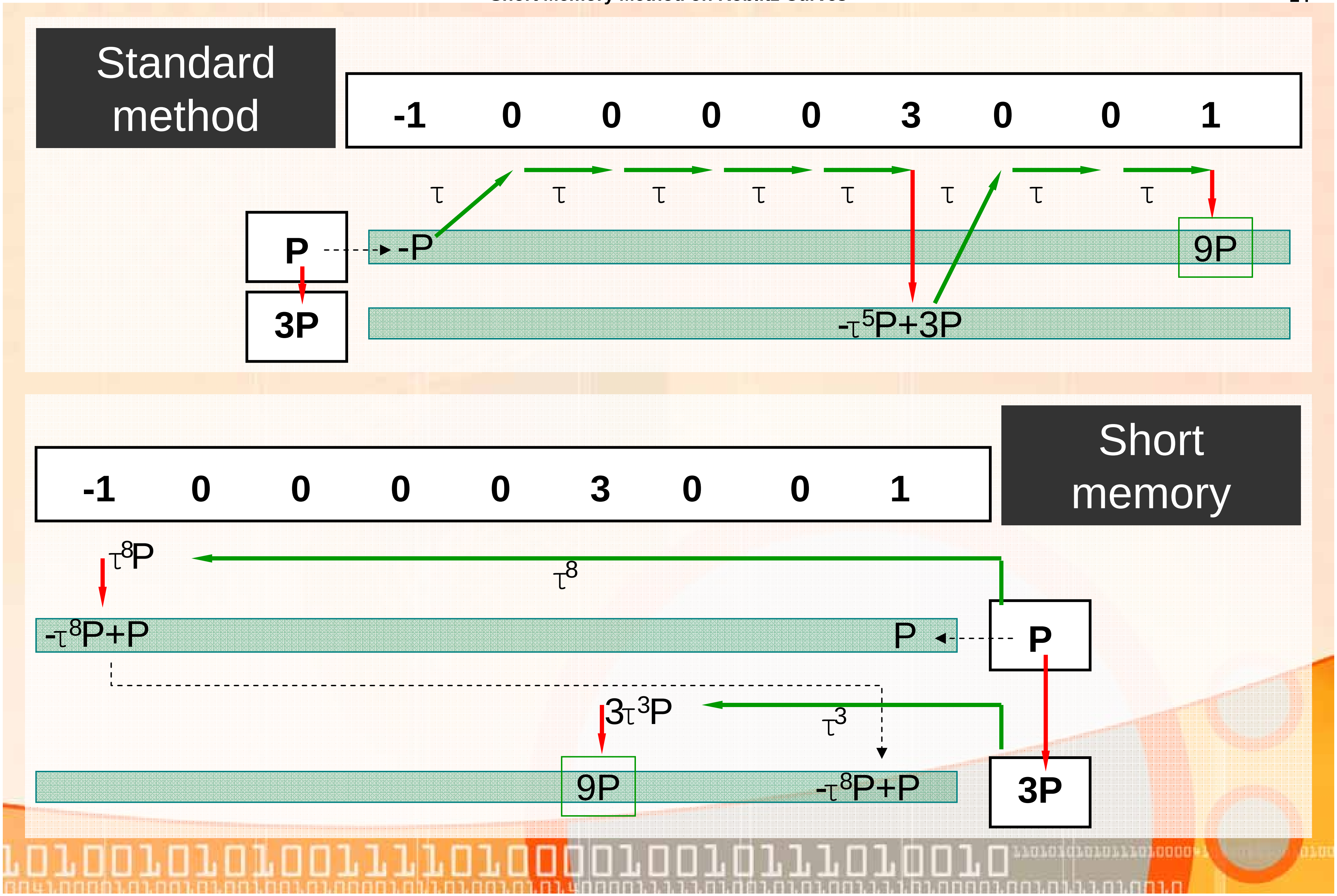
$$b^2=b_0\beta_0^2+b_1\beta_1^2+\ldots+b_{m-2}\beta_{m-2}^2+b_{m-1}\beta_{m-1}^2$$

$$= b_{m-1}\beta_0+b_0\beta_1+b_1\beta_2+\ldots+b_{m-2}\beta_{m-1}$$

Interesting for Koblitz curves (τ)

Mixed approach?

Short Memory – Normal Basis



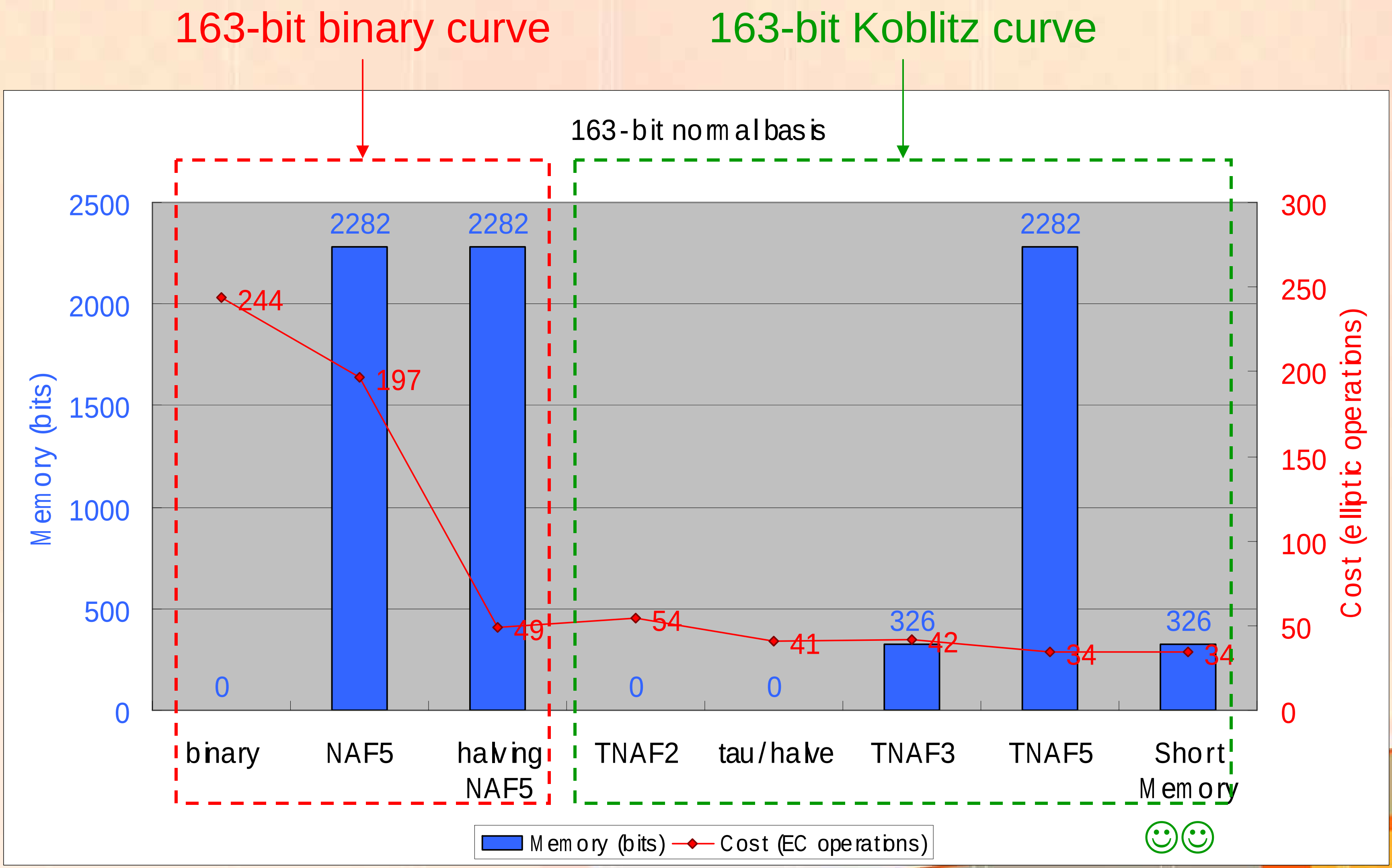
Sequential Precomputations

u	$\alpha_u = u \bmod \tau^5$	Binary representation of α_u
1	1	1
3	$\tau-3$	τ^2-1
5	$\tau-1$	$\tau-1$
7	$\tau+1$	$\tau+1$
9	$2\tau-3$	$-\tau^4-\tau-1 = -\tau^4-(\tau+1)$
11	$2\tau-1$	$-\tau^3+\tau^2-1 = -\tau^3+\tau^2-1$
13	$2\tau+1$	$-\tau^3+\tau^2+1 = -\tau^2(\tau-1)+1$
15	$-3\tau+1$	$\tau^3-\tau^2-\tau+1 = -(-\tau^3+\tau^2-1)-\tau$

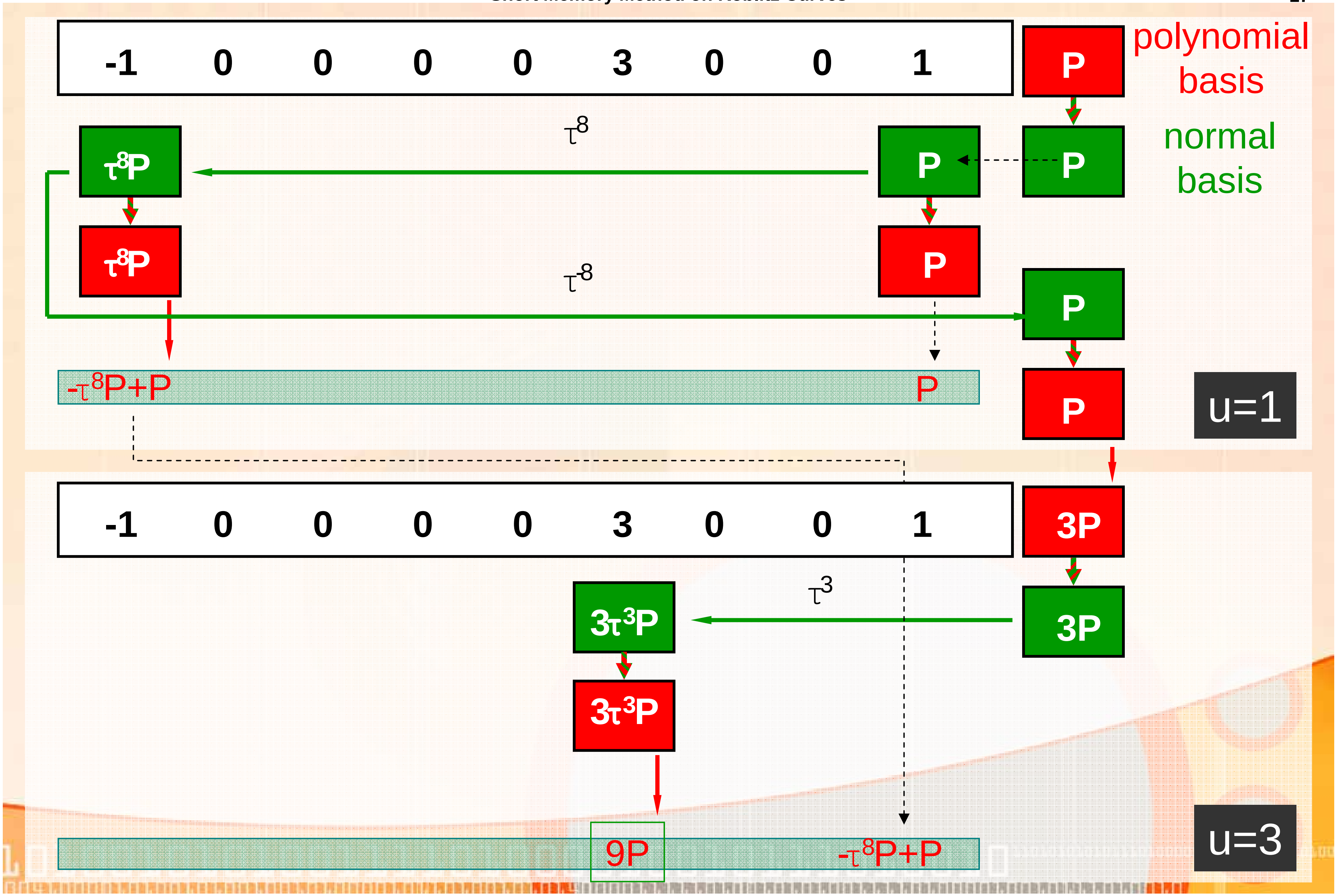
Precomputations

$\alpha_1 P = P$	$\alpha_3 P = \tau^2 P + P$	$\alpha_{11} P = -\tau^3 P + \alpha_3 P$	$\alpha_{15} P = -\alpha_{11} P - \tau P$
$\alpha_5 P = \tau P - P$	$\alpha_{13} P = -\tau^2 \alpha_5 P + P$	$\alpha_7 P = \tau P + P$	$\alpha_9 P = -\tau^4 P - \alpha_7 P$

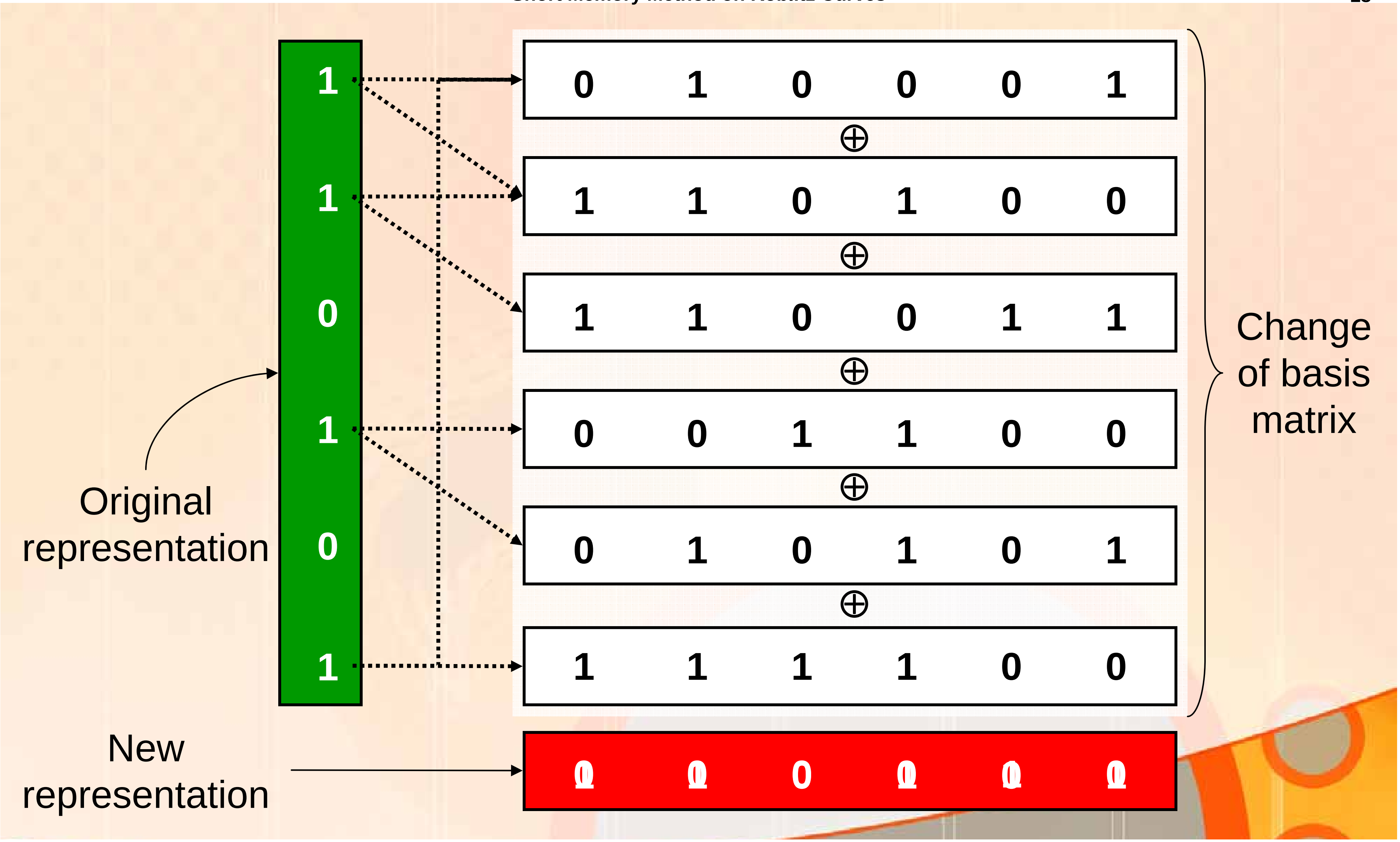
Performance, Hardware



Short Memory - Mixed Bases

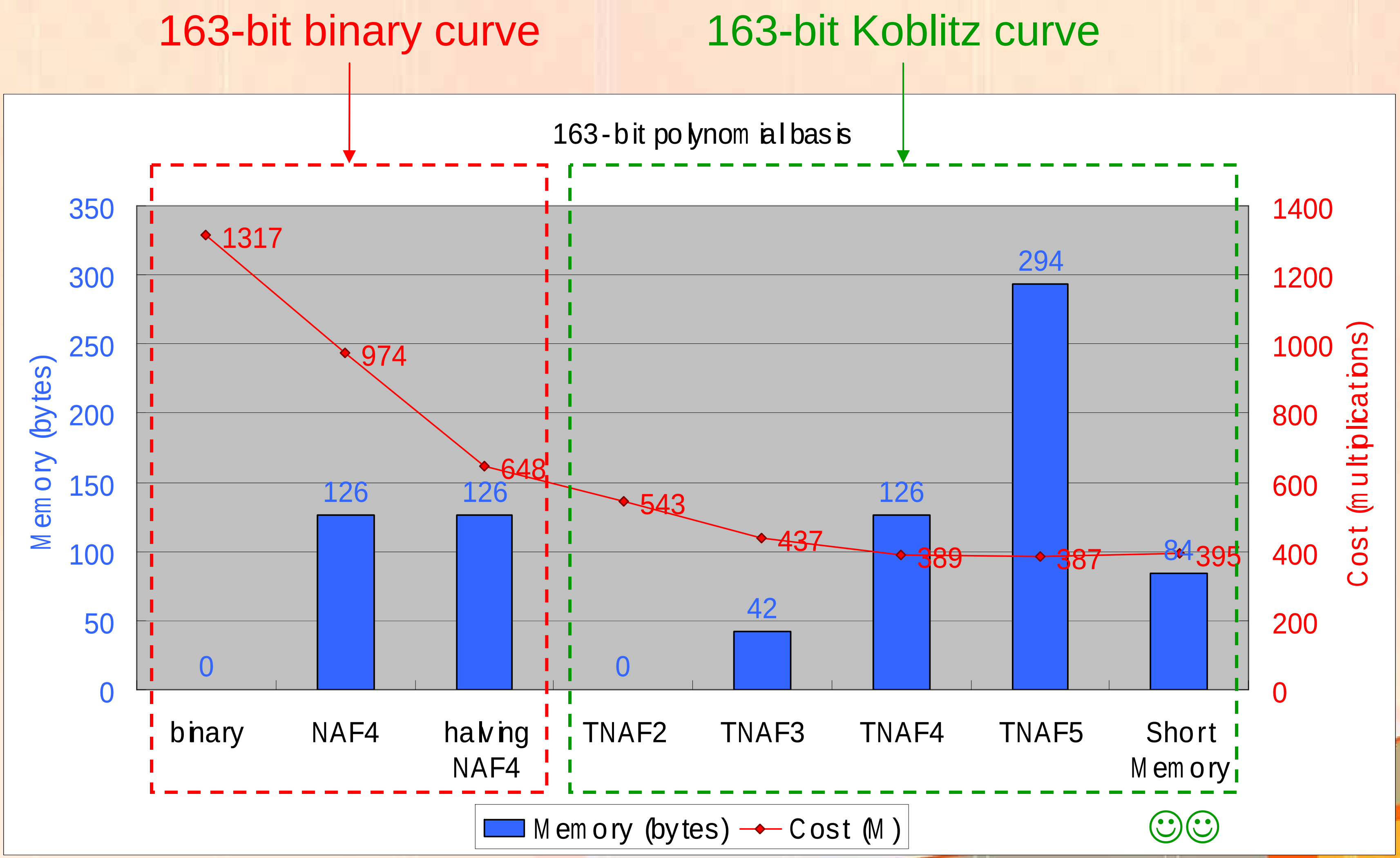


Change of Basis



Cyclic shift not explicitly computed

Performance, Software



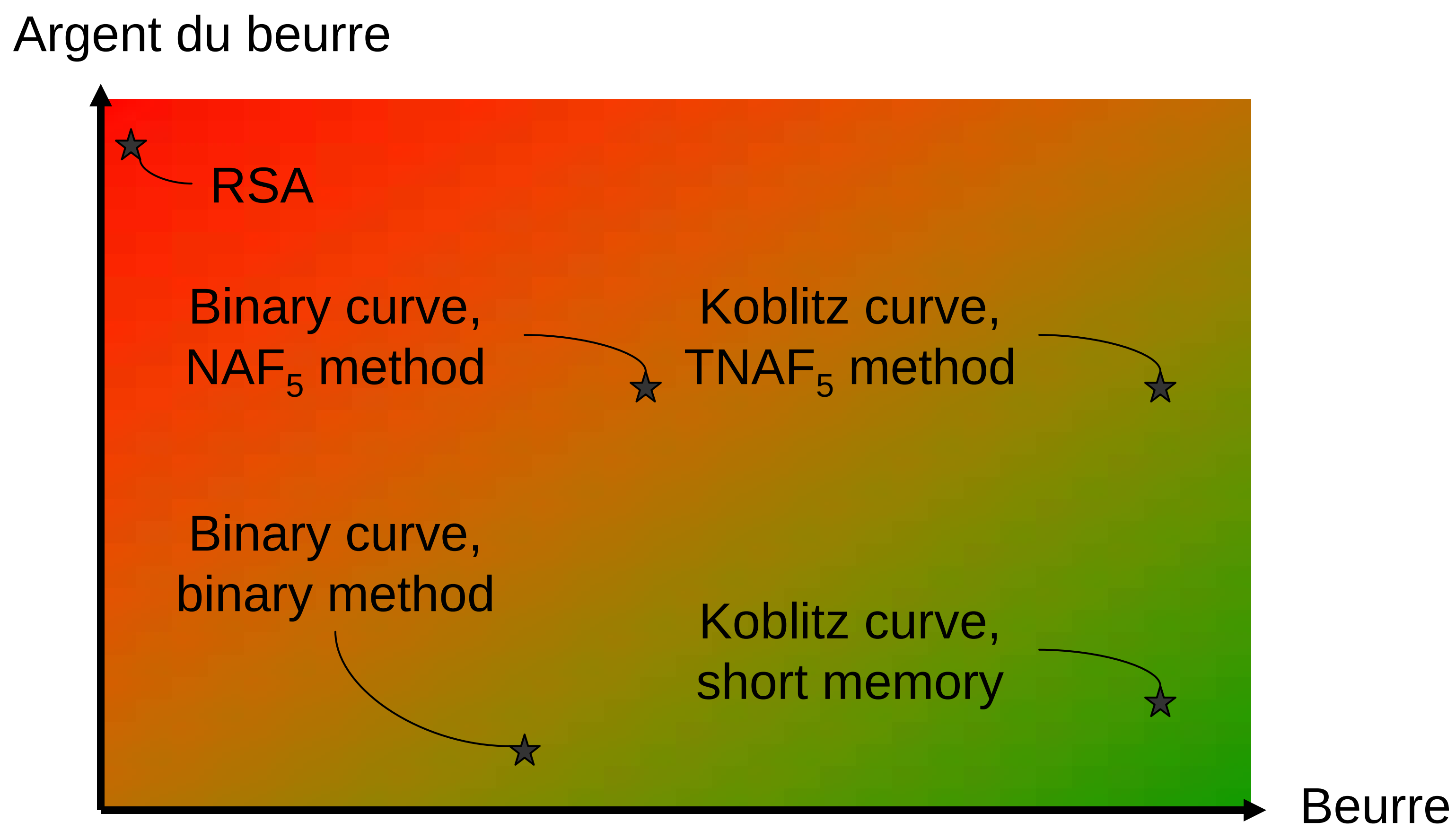
Extensions, open problems

Side channel & fault attacks

Change of basis

Other curves

Recap



"Le beurre et l'argent du beurre"

Questions & Comments