Timing Attacks against the Syndrome Inversion in code-based Cryptosystems

Falko Strenzke

Cryptography and Computeralgebra, Department of Computer Science,
Technische Universität Darmstadt, Germany,
fstrenzke@crypto-source.de

June 5, 2013

Introduction

- Topic: recovery of the secret key of a code-based McEliece or Niederreiter cryptosystem through a timing side-channel
- Practical local timing attack
- Combination of three different vulnerabilities

Introduction

- Topic: recovery of the secret key of a code-based McEliece or Niederreiter cryptosystem through a timing side-channel
- Practical local timing attack
- Combination of three different vulnerabilities

Introduction

- Topic: recovery of the secret key of a code-based McEliece or Niederreiter cryptosystem through a timing side-channel
- Practical local timing attack
- Combination of three different vulnerabilities

- Parameters of a Goppa Code
  - irreducible polynomial $g(Y) \in \mathbb{F}_{2^m}[Y]$ of degree t (the Goppa Polynomial)
  - support $\Gamma = (\alpha_0, \alpha_1, \ldots, \alpha_{n-1})$, a *permutation* of $\mathbb{F}_{2^m}$, where $n = 2^m$
- Properties of the Code
  - the code has length n (code word length) ,
  - dimension $k = n - mt$ (message length) and
  - can correct up to t errors.
  - a parity check matrix H, where $cH^\top = 0$ if $c \in C$
  - example for secure parameters: $n = 2048$, $t = 50$ for 100 bit security

Goppa Codes

- Parameters of a Goppa Code
  - irreducible polynomial $g(Y) \in \mathbb{F}_{2^m}[Y]$ of degree t (the Goppa Polynomial)
  - support $\Gamma = (\alpha_0, \alpha_1, \ldots, \alpha_{n-1})$, a *permutation* of $\mathbb{F}_{2^m}$, where $n = 2^m$
- Properties of the Code
  - the code has length n (code word length) ,
  - dimension $k = n - mt$ (message length) and
  - can correct up to t errors.
  - a parity check matrix H, where $cH^\top = 0$ if $c \in C$
  - example for secure parameters: $n = 2048$, $t = 50$ for 100 bit security

Goppa Codes

- Parameters of a Goppa Code
  - irreducible polynomial $g(Y) \in \mathbb{F}_{2^m}[Y]$ of degree t (the Goppa Polynomial)
  - support $\Gamma = (\alpha_0, \alpha_1, \dots, \alpha_{n-1})$, a *permutation* of $\mathbb{F}_{2^m}$, where $n = 2^m$
- Properties of the Code
  - the code has length n (code word length) ,
  - dimension $k = n - mt$ (message length) and
  - can correct up to t errors.
  - a parity check matrix H, where $cH^\top = 0$ if $c \in C$
  - example for secure parameters: $n = 2048$, $t = 50$ for 100 bit security

Goppa Codes

- Parameters of a Goppa Code
  - irreducible polynomial $g(Y) \in \mathbb{F}_{2^m}[Y]$ of degree t (the Goppa Polynomial)
  - support $\Gamma = (\alpha_0, \alpha_1, \ldots, \alpha_{n-1})$, a *permutation* of $\mathbb{F}_{2^m}$, where $n = 2^m$
- Properties of the Code
  - the code has length n (code word length) ,
  - dimension $k = n - mt$ (message length) and
  - can correct up to t errors.
  - a parity check matrix H, where $cH^\top = 0$ if $c \in \mathcal{C}$
  - example for secure parameters: $n = 2048$, $t = 50$ for 100 bit security

Goppa Codes

- Parameters of a Goppa Code
  - irreducible polynomial $g(Y) \in \mathbb{F}_{2^m}[Y]$ of degree t (the Goppa Polynomial)
  - support $\Gamma = (\alpha_0, \alpha_1, \ldots, \alpha_{n-1})$, a *permutation* of $\mathbb{F}_{2^m}$, where $n = 2^m$
- Properties of the Code
  - the code has length n (code word length) ,
  - dimension $k = n - mt$ (message length) and
  - can correct up to t errors.
  - a parity check matrix H, where $cH^\top = 0$ if $c \in \mathcal{C}$
  - example for secure parameters: $n = 2048$, $t = 50$ for 100 bit security

Goppa Codes

- Parameters of a Goppa Code
  - irreducible polynomial $g(Y) \in \mathbb{F}_{2^m}[Y]$ of degree t (the Goppa Polynomial)
  - support $\Gamma = (\alpha_0, \alpha_1, \ldots, \alpha_{n-1})$, a *permutation* of $\mathbb{F}_{2^m}$, where $n = 2^m$
- Properties of the Code
  - the code has length n (code word length) ,
  - dimension $k = n - mt$ (message length) and
  - can correct up to t errors.
  - a parity check matrix H, where $cH^\top = 0$ if $c \in \mathcal{C}$
  - example for secure parameters: $n = 2048$, $t = 50$ for 100 bit security

Goppa Codes

- Parameters of a Goppa Code
  - irreducible polynomial $g(Y) \in \mathbb{F}_{2^m}[Y]$ of degree t (the Goppa Polynomial)
  - support $\Gamma = (\alpha_0, \alpha_1, \ldots, \alpha_{n-1})$, a *permutation* of $\mathbb{F}_{2^m}$, where $n = 2^m$
- Properties of the Code
  - the code has length n (code word length) ,
  - dimension $k = n - mt$ (message length) and
  - can correct up to t errors.
  - a parity check matrix H, where $cH^\top = 0$ if $c \in \mathcal{C}$
  - example for secure parameters: $n = 2048$, $t = 50$ for 100 bit security

Goppa Codes

- Parameters of a Goppa Code
  - irreducible polynomial $g(Y) \in \mathbb{F}_{2^m}[Y]$ of degree t (the Goppa Polynomial)
  - support $\Gamma = (\alpha_0, \alpha_1, \dots, \alpha_{n-1})$, a *permutation* of $\mathbb{F}_{2^m}$, where $n = 2^m$
- Properties of the Code
  - the code has length n (code word length) ,
  - dimension $k = n - mt$ (message length) and
  - can correct up to t errors.
  - a parity check matrix H, where $cH^\top = 0$ if $c \in \mathcal{C}$
  - example for secure parameters: $n = 2048$, $t = 50$ for 100 bit security

Goppa Codes

- Parameters of a Goppa Code
  - irreducible polynomial $g(Y) \in \mathbb{F}_{2^m}[Y]$ of degree t (the Goppa Polynomial)
  - support $\Gamma = (\alpha_0, \alpha_1, \ldots, \alpha_{n-1})$, a *permutation* of $\mathbb{F}_{2^m}$, where $n = 2^m$
- Properties of the Code
  - the code has length n (code word length) ,
  - dimension $k = n - mt$ (message length) and
  - can correct up to t errors.
  - a parity check matrix H, where $cH^\top = 0$ if $c \in \mathcal{C}$
  - example for secure parameters: $n = 2048$, $t = 50$ for 100 bit security

The McEliece PKC

- key generation
  - choose the parameters n and t
  - generate randomly $g(Y)$ and Γ (determining the secret the code)
  - for this private code $\mathcal{C}_s$ one has a generator matrix G_s
  - the public key is $G_p = [\mathbb{I}|G_p'] = TG_s$
- encryption: $\vec{z} = \vec{m}G_p + \vec{e}$, $\mathrm{wt}\left(\vec{e}\right) = t$
- decryption: syndrome decoding

The McEliece PKC

- key generation
  - choose the parameters n and t
  - generate randomly $g(Y)$ and Γ (determining the secret the code)
  - for this private code $\mathcal{C}_s$ one has a generator matrix G_s
  - the public key is $G_p = [\mathbb{I}|G_p'] = TG_s$
- encryption: $\vec{z} = \vec{m}G_p + \vec{e}$, $\mathrm{wt}\left(\vec{e}\right) = t$
- decryption: syndrome decoding

The McEliece PKC

- key generation
  - choose the parameters n and t
  - generate randomly $g(Y)$ and Γ (determining the secret the code)
  - for this private code $\mathcal{C}_s$ one has a generator matrix G_s
  - the public key is $G_p = [\mathbb{I}|G_p'] = TG_s$
- encryption: $\vec{z} = \vec{m}G_p + \vec{e}$, $\mathrm{wt}(\vec{e}) = t$
- decryption: syndrome decoding

The McEliece PKC

- key generation
  - choose the parameters n and t
  - generate randomly $g(Y)$ and Γ (determining the secret the code)
  - for this private code $\mathcal{C}_s$ one has a generator matrix G_s
  - the public key is $G_p = [\mathbb{I}|G'_p] = TG_s$
- encryption: $\vec{z} = \vec{m}G_p + \vec{e}$, $\mathrm{wt}(\vec{e}) = t$
- decryption: syndrome decoding

The McEliece PKC

- key generation
  - choose the parameters n and t
  - generate randomly $g(Y)$ and Γ (determining the secret the code)
  - for this private code $\mathcal{C}_s$ one has a generator matrix G_s
  - the public key is $G_p = [\mathbb{I}|G'_p] = TG_s$
- encryption: $\vec{z} = \vec{m}G_p + \vec{e}$, $\mathrm{wt}(\vec{e}) = t$
- decryption: syndrome decoding

The McEliece PKC

- key generation
  - choose the parameters n and t
  - generate randomly $g(Y)$ and Γ (determining the secret the code)
  - for this private code $\mathcal{C}_s$ one has a generator matrix G_s
  - the public key is $G_p = [\mathbb{I}|G'_p] = TG_s$
- encryption: $\vec{z} = \vec{m}G_p + \vec{e}$, $\mathrm{wt}\left(\vec{e}\right) = t$
- decryption: syndrome decoding

The McEliece PKC

- key generation
  - choose the parameters n and t
  - generate randomly $g(Y)$ and Γ (determining the secret the code)
  - for this private code $\mathcal{C}_s$ one has a generator matrix G_s
  - the public key is $G_p = [\mathbb{I}|G_p'] = TG_s$
- encryption: $\vec{z} = \vec{m}G_p + \vec{e}$, $\mathrm{wt}\left(\vec{e}\right) = t$
- decryption: syndrome decoding

Syndrome Decoding

- secret key: $g(Y)$, $\Gamma = (\alpha_0, \alpha_1, \ldots, \alpha_{n-1})$
- input: distorted codeword $\vec{e} \oplus \vec{c}$
- output: error vector $\vec{e} \in \mathbb{F}_{2^m}^n$, $\mathrm{wt}(\vec{e}) = t$ chosen during encryption
- $S(Y) \leftarrow \underbrace{(\vec{e} \oplus \vec{c})H^\top}_{\in \mathbb{F}_{2^m}^t} \left(Y^{t-1}, \cdots, Y, 1\right)^\top$
- $\tau(Y) \leftarrow \sqrt{S^{-1}(Y) + Y} \bmod g(Y)$ // by EEA
- $(\alpha(Y), \beta(Y)) \leftarrow \mathrm{EEA}(g(Y), \tau(Y))$
- $\sigma(Y) \leftarrow \alpha^2(Y) + Y\beta^2(Y)$
- $e_i \leftarrow 1$ iff $\sigma(\alpha_i) = 0$

Syndrome Decoding

- secret key: $g(Y)$, $\Gamma = (\alpha_0, \alpha_1, \ldots, \alpha_{n-1})$
- input: distorted codeword $\vec{e} \oplus \vec{c}$
- output: error vector $\vec{e} \in \mathbb{F}_{2^m}^n$, $\mathrm{wt}(\vec{e}) = t$ chosen during encryption
- $S(Y) \leftarrow \underbrace{(\vec{e} \oplus \vec{c})H^\top}_{\in \mathbb{F}_{2^m}^t} \left(Y^{t-1}, \cdots, Y, 1\right)^\top$
- $\tau(Y) \leftarrow \sqrt{S^{-1}(Y) + Y} \bmod g(Y)$ // by EEA
- $(\alpha(Y), \beta(Y)) \leftarrow \mathrm{EEA}(g(Y), \tau(Y))$
- $\sigma(Y) \leftarrow \alpha^2(Y) + Y\beta^2(Y)$
- $e_i \leftarrow 1$ iff $\sigma(\alpha_i) = 0$

Syndrome Decoding

- secret key: $g(Y)$, $\Gamma = (\alpha_0, \alpha_1, \ldots, \alpha_{n-1})$
- input: distorted codeword $\vec{e} \oplus \vec{c}$
- output: error vector $\vec{e} \in \mathbb{F}_{2^m}^n$, $\mathrm{wt}(\vec{e}) = t$ chosen during encryption

- $S(Y) \leftarrow \underbrace{(\vec{e} \oplus \vec{c})H^\top}_{\in \mathbb{F}_{2^m}^t} \left(Y^{t-1}, \cdots, Y, 1\right)^\top$
- $\tau(Y) \leftarrow \sqrt{S^{-1}(Y) + Y} \bmod g(Y)$ // by EEA
- $(\alpha(Y), \beta(Y)) \leftarrow \mathrm{EEA}(g(Y), \tau(Y))$
- $\sigma(Y) \leftarrow \alpha^2(Y) + Y\beta^2(Y)$
- $e_i \leftarrow 1$ iff $\sigma(\alpha_i) = 0$

Syndrome Decoding

- secret key: $g(Y)$, $\Gamma = (\alpha_0, \alpha_1, \ldots, \alpha_{n-1})$
- input: distorted codeword $\vec{e} \oplus \vec{c}$
- output: error vector $\vec{e} \in \mathbb{F}_{2^m}^n$, $\mathrm{wt}(\vec{e}) = t$ chosen during encryption
- $S(Y) \leftarrow \underbrace{(\vec{e} \oplus \vec{c})H^\top}_{\in \mathbb{F}_{2^m}^t} (Y^{t-1}, \cdots, Y, 1)^\top$
- $\tau(Y) \leftarrow \sqrt{S^{-1}(Y) + Y} \bmod g(Y)$ // by EEA
- $(\alpha(Y), \beta(Y)) \leftarrow \mathrm{EEA}(g(Y), \tau(Y))$
- $\sigma(Y) \leftarrow \alpha^2(Y) + Y\beta^2(Y)$
- $e_i \leftarrow 1$ iff $\sigma(\alpha_i) = 0$

Syndrome Decoding

- secret key: $g(Y)$, $\Gamma = (\alpha_0, \alpha_1, \ldots, \alpha_{n-1})$
- input: distorted codeword $\vec{e} \oplus \vec{c}$
- output: error vector $\vec{e} \in \mathbb{F}_{2^m}^n$, $\mathrm{wt}(\vec{e}) = t$ chosen during encryption
- $S(Y) \leftarrow \underbrace{(\vec{e} \oplus \vec{c})H^\top}_{\in \mathbb{F}_{2^m}^t} \left(Y^{t-1}, \cdots, Y, 1\right)^\top$
- $\tau(Y) \leftarrow \sqrt{S^{-1}(Y) + Y} \bmod g(Y)$ // by EEA
- $(\alpha(Y), \beta(Y)) \leftarrow \mathrm{EEA}(g(Y), \tau(Y))$
- $\sigma(Y) \leftarrow \alpha^2(Y) + Y\beta^2(Y)$
- $e_i \leftarrow 1$ iff $\sigma(\alpha_i) = 0$

Syndrome Decoding

- secret key: $g(Y)$, $\Gamma = (\alpha_0, \alpha_1, \ldots, \alpha_{n-1})$
- input: distorted codeword $\vec{e} \oplus \vec{c}$
- output: error vector $\vec{e} \in \mathbb{F}_{2^m}^n$, $\mathrm{wt}(\vec{e}) = t$ chosen during encryption

- $S(Y) \leftarrow \underbrace{(\vec{e} \oplus \vec{c})H^\top}_{\in \mathbb{F}_{2^m}^t} \left(Y^{t-1}, \cdots, Y, 1\right)^\top$
- $\tau(Y) \leftarrow \sqrt{S^{-1}(Y) + Y} \bmod g(Y)$ // by EEA
- $(\alpha(Y), \beta(Y)) \leftarrow \mathrm{EEA}(g(Y), \tau(Y))$
- $\sigma(Y) \leftarrow \alpha^2(Y) + Y\beta^2(Y)$
- $e_i \leftarrow 1$ iff $\sigma(\alpha_i) = 0$

Syndrome Decoding

- secret key: $g(Y)$, $\Gamma = (\alpha_0, \alpha_1, \dots, \alpha_{n-1})$
- input: distorted codeword $\vec{e} \oplus \vec{c}$
- output: error vector $\vec{e} \in \mathbb{F}_{2^m}^n$, $\mathrm{wt}(\vec{e}) = t$ chosen during encryption

- $S(Y) \leftarrow \underbrace{(\vec{e} \oplus \vec{c}) H^\top}_{\in \mathbb{F}_{2^m}^t} \left(Y^{t-1}, \cdots, Y, 1\right)^\top$
- $\tau(Y) \leftarrow \sqrt{S^{-1}(Y) + Y} \bmod g(Y)$ // by EEA
- $(\alpha(Y), \beta(Y)) \leftarrow \mathrm{EEA}(g(Y), \tau(Y))$
- $\sigma(Y) \leftarrow \alpha^2(Y) + Y\beta^2(Y)$
- $e_i \leftarrow 1$ iff $\sigma(\alpha_i) = 0$

Syndrome Decoding

- secret key: $g(Y)$, $\Gamma = (\alpha_0, \alpha_1, \ldots, \alpha_{n-1})$
- input: distorted codeword $\vec{e} \oplus \vec{c}$
- output: error vector $\vec{e} \in \mathbb{F}_{2^m}^n$, $\mathrm{wt}(\vec{e}) = t$ chosen during encryption

- $S(Y) \leftarrow \underbrace{(\vec{e} \oplus \vec{c})H^\top}_{\in \mathbb{F}_{2^m}^t} \left(Y^{t-1}, \cdots, Y, 1\right)^\top$
- $\tau(Y) \leftarrow \sqrt{S^{-1}(Y) + Y} \bmod g(Y)$ // by EEA
- $(\alpha(Y), \beta(Y)) \leftarrow \mathrm{EEA}(g(Y), \tau(Y))$
- $\sigma(Y) \leftarrow \alpha^2(Y) + Y\beta^2(Y)$
- $e_i \leftarrow 1$ iff $\sigma(\alpha_i) = 0$

Error Positions and Support Elements

$$
\begin{array}{l|cccccccccccc}
\vec{e} = & (& 0 & 0 & \ldots & 0 & 1 & 0 & \ldots & 0 & 1 & 0 & \ldots &) \\
\text{indexes:} & & 0 & 1 & \ldots & & f_1 & & & & f_2 & & & \\
\hline
 & & & & & & \epsilon_1 & & & & \epsilon_2 & & & \\
 & & & & & & = \alpha_{f_1} & & & & = \alpha_{f_2} & & &
\end{array}
$$

- $\sigma(Y) = \prod_{i=0}^{w-1}(\alpha_{f_i} - Y)$
- $\Gamma = (\alpha_0, \alpha_1, \ldots \alpha_{n-1})$

Error Positions and Support Elements

$\vec{e} =$	(	0	0	...	0	1	0	...	0	1	0	...	)
indexes:		0	1	...		f_1				f_2			
						ϵ_1				ϵ_2			
						$= \alpha_{f_1}$				$= \alpha_{f_2}$			

- $\sigma(Y) = \prod_{i=0}^{w-1}(\alpha_{f_i} - Y)$
- $\Gamma = (\alpha_0, \alpha_1, \ldots \alpha_{n-1})$

Vulnerability against weight 4 error vectors

- previous work (PQCrypto 2010, Strenzke):
  - input $w = 4$ error vectors $\rightarrow$ measure decryption time
  - time $\rightarrow N$ (number of iterations in the key equation solving EEA)
  - $N = 1 \rightarrow \sum_{i=1}^{4} \epsilon_i \neq 0$
  - $N = 0 \rightarrow \sum_{i=1}^{4} \epsilon_i = 0$
  - two Problems:
    - insufficient information
    - practicality as timing attack left open

Vulnerability against weight 4 error vectors

- previous work (PQCrypto 2010, Strenzke):
  - input $w = 4$ error vectors $\rightarrow$ measure decryption time
  - time $\rightarrow$ N (number of iterations in the key equation solving EEA)
  - $N = 1 \rightarrow \sum_{i=1}^{4} \epsilon_i \neq 0$
  - $N = 0 \rightarrow \sum_{i=1}^{4} \epsilon_i = 0$
  - two Problems:
    - insufficient information
    - practicality as timing attack left open

Vulnerability against weight 4 error vectors

- previous work (PQCrypto 2010, Strenzke):
  - input $w = 4$ error vectors $\rightarrow$ measure decryption time
  - time $\rightarrow N$ (number of iterations in the key equation solving EEA)
  - $N = 1 \rightarrow \sum_{i=1}^{4} \epsilon_i \neq 0$
  - $N = 0 \rightarrow \sum_{i=1}^{4} \epsilon_i = 0$
  - two Problems:
    - insufficient information
    - practicality as timing attack left open

Vulnerability against weight 4 error vectors

- previous work (PQCrypto 2010, Strenzke):
  - input $w = 4$ error vectors $\rightarrow$ measure decryption time
  - time $\rightarrow N$ (number of iterations in the key equation solving EEA)
  - $N = 1 \rightarrow \sum_{i=1}^{4} \epsilon_i \neq 0$
  - $N = 0 \rightarrow \sum_{i=1}^{4} \epsilon_i = 0$
  - two Problems:
    - insufficient information
    - practicality as timing attack left open

Vulnerability against weight 4 error vectors

- previous work (PQCrypto 2010, Strenzke):
  - input $w = 4$ error vectors $\rightarrow$ measure decryption time
  - time $\rightarrow N$ (number of iterations in the key equation solving EEA)
  - $N = 1 \rightarrow \sum_{i=1}^{4} \epsilon_i \neq 0$
  - $N = 0 \rightarrow \sum_{i=1}^{4} \epsilon_i = 0$
  - two Problems:
    - insufficient information
    - practicality as timing attack left open

Vulnerability against weight 4 error vectors

- previous work (PQCrypto 2010, Strenzke):
  - input $w = 4$ error vectors $\rightarrow$ measure decryption time
  - time $\rightarrow$ N (number of iterations in the key equation solving EEA)
  - $N = 1 \rightarrow \sum_{i=1}^{4} \epsilon_i \neq 0$
  - $N = 0 \rightarrow \sum_{i=1}^{4} \epsilon_i = 0$
  - two Problems:
    - insufficient information
    - practicality as timing attack left open

Vulnerability against weight 4 error vectors

- previous work (PQCrypto 2010, Strenzke):
  - input $w = 4$ error vectors $\rightarrow$ measure decryption time
  - time $\rightarrow$ N (number of iterations in the key equation solving EEA)
  - $N = 1 \rightarrow \sum_{i=1}^{4} \epsilon_i \neq 0$
  - $N = 0 \rightarrow \sum_{i=1}^{4} \epsilon_i = 0$
  - two Problems:
    - insufficient information
    - practicality as timing attack left open

Vulnerability against weight 4 error vectors

- previous work (PQCrypto 2010, Strenzke):
  - input $w = 4$ error vectors $\rightarrow$ measure decryption time
  - time $\rightarrow N$ (number of iterations in the key equation solving EEA)
  - $N = 1 \rightarrow \sum_{i=1}^{4} \epsilon_i \neq 0$
  - $N = 0 \rightarrow \sum_{i=1}^{4} \epsilon_i = 0$
  - two Problems:
    - insufficient information
    - practicality as timing attack left open

New Source of timing differences for weight 4 error vectors

- Syndrome

$$S(Y) \equiv \sum_{i=1}^{w} \frac{1}{Y \oplus \epsilon_i} \equiv \frac{\Omega(Y)}{\sigma(Y)} \mod g(Y)$$

- Known about the syndrome inversion EEA: If $w \leq t/2$
- then break once $\deg(r_i(Y)) \leq (t/2) - 1$
- to find $\sigma(Y)$ as the output of EEA
- $\rightarrow$ information about an intermediate iteration

New Source of timing differences for weight 4 error vectors

- Syndrome

$$S(Y) \equiv \sum_{i=1}^{w} \frac{1}{Y \oplus \epsilon_i} \equiv \frac{\Omega(Y)}{\sigma(Y)} \bmod g(Y)$$

- Known about the syndrome inversion EEA: If $w \leq t/2$
- then break once $\deg(r_i(Y)) \leq (t/2) - 1$
- to find $\sigma(Y)$ as the output of EEA
- $\rightarrow$ information about an intermediate iteration

- Syndrome

$$S(Y) \equiv \sum_{i=1}^{w} \frac{1}{Y \oplus \epsilon_i} \equiv \frac{\Omega(Y)}{\sigma(Y)} \bmod g(Y)$$

- Known about the syndrome inversion EEA: If $w \leq t/2$
- then break once $\deg(r_i(Y)) \leq (t/2) - 1$
- to find $\sigma(Y)$ as the output of EEA
- $\rightarrow$ information about an intermediate iteration

New Source of timing differences for weight 4 error vectors

- Syndrome

$$S(Y) \equiv \sum_{i=1}^{w} \frac{1}{Y \oplus \epsilon_i} \equiv \frac{\Omega(Y)}{\sigma(Y)} \mod g(Y)$$

- Known about the syndrome inversion EEA: If $w \leq t/2$
- then break once $\deg(r_i(Y)) \leq (t/2) - 1$
- to find $\sigma(Y)$ as the output of EEA
- $\rightarrow$ information about an intermediate iteration

New Source of timing differences for weight 4 error vectors

- Syndrome

$$S(Y) \equiv \sum_{i=1}^{w} \frac{1}{Y \oplus \epsilon_i} \equiv \frac{\Omega(Y)}{\sigma(Y)} \bmod g(Y)$$

- Known about the syndrome inversion EEA: If $w \leq t/2$
- then break once $\deg(r_i(Y)) \leq (t/2) - 1$
- to find $\sigma(Y)$ as the output of EEA
- $\rightarrow$ information about an intermediate iteration

The Syndrome Inversion EEA for $w = 4$

$b_{-1} \leftarrow 0,\ b_0 \leftarrow 1,\ r_{-1} \leftarrow g(Y),\ r_0 \leftarrow S(Y),\ i \leftarrow 0$
while $\deg(r_i) > 0$ **do**
    $i \leftarrow i + 1$
    $(q_i(Y), r_i(Y)) \leftarrow r_{i-2}(Y)/r_{i-1}(Y)$
    $b_i(Y) \leftarrow b_{i-2}(Y) + q_i(Y)b_{i-1}(Y)$
end while

we know: $\exists i : \sigma(Y) = b_i(Y) \wedge \Omega(Y) = r_i(Y)$

$$S(Y) \equiv \sum_{i=1}^{4} \frac{1}{Y \oplus \epsilon_i} \equiv \frac{\Omega(Y)}{\sigma(Y)} \equiv \frac{\sigma_3 Y^2 \oplus \sigma_1}{Y^4 \oplus \sigma_3 Y^3 \oplus \sigma_2 Y^2 \oplus \sigma_1 Y \oplus \sigma_0} \mod g(Y)$$

i	$\deg(q_i(Y))$	$\deg(b_i(Y))$	$\deg(r_i(Y))$
1	1	1	t-2
2	1	2	t-3
3	1	3	t-4
4	1	4	2 0
5	t - 6	t - 2	1
6	1	t - 1	0

$\sigma_3 = \epsilon_1 \oplus \epsilon_2 \oplus \epsilon_3 \oplus \epsilon_4 = 0 \Rightarrow i = 5, 6$ skipped

$b_{-1} \leftarrow 0,\ b_0 \leftarrow 1,\ r_{-1} \leftarrow g(Y),\ r_0 \leftarrow S(Y),\ i \leftarrow 0$
while $\deg(r_i) > 0$ **do**
    $i \leftarrow i + 1$
    $(q_i(Y), r_i(Y)) \leftarrow r_{i-2}(Y)/r_{i-1}(Y)$
    $b_i(Y) \leftarrow b_{i-2}(Y) + q_i(Y)b_{i-1}(Y)$
end while

we know: $\exists i : \sigma(Y) = b_i(Y) \wedge \Omega(Y) = r_i(Y)$

$$S(Y) \equiv \sum_{i=1}^{4} \frac{1}{Y \oplus \epsilon_i} \equiv \frac{\Omega(Y)}{\sigma(Y)} \equiv \frac{\sigma_3 Y^2 \oplus \sigma_1}{Y^4 \oplus \sigma_3 Y^3 \oplus \sigma_2 Y^2 \oplus \sigma_1 Y \oplus \sigma_0} \mod g(Y)$$

i	$\deg(q_i(Y))$	$\deg(b_i(Y))$	$\deg(r_i(Y))$
1	1	1	t-2
2	1	2	t-3
3	1	3	t-4
4	1	4	2 0
5	t - 6	t - 2	1
6	1	t - 1	0

$\sigma_3 = \epsilon_1 \oplus \epsilon_2 \oplus \epsilon_3 \oplus \epsilon_4 = 0 \Rightarrow i = 5, 6$ skipped

$b_{-1} \leftarrow 0,\ b_0 \leftarrow 1,\ r_{-1} \leftarrow g(Y),\ r_0 \leftarrow S(Y),\ i \leftarrow 0$
while $\deg(r_i) > 0$ **do**
    $i \leftarrow i + 1$
    $(q_i(Y), r_i(Y)) \leftarrow r_{i-2}(Y)/r_{i-1}(Y)$
    $b_i(Y) \leftarrow b_{i-2}(Y) + q_i(Y)b_{i-1}(Y)$
end while

we know: $\exists i : \sigma(Y) = b_i(Y) \wedge \Omega(Y) = r_i(Y)$

$$S(Y) \equiv \sum_{i=1}^{4} \frac{1}{Y \oplus \epsilon_i} \equiv \frac{\Omega(Y)}{\sigma(Y)} \equiv \frac{\sigma_3 Y^2 \oplus \sigma_1}{Y^4 \oplus \sigma_3 Y^3 \oplus \sigma_2 Y^2 \oplus \sigma_1 Y \oplus \sigma_0} \mod g(Y)$$

i	$\deg(q_i(Y))$	$\deg(b_i(Y))$	$\deg(r_i(Y))$
1	1	1	t-2
2	1	2	t-3
3	1	3	t-4
4	1	4	2 0
5	t - 6	t - 2	1
6	1	t - 1	0

$\sigma_3 = \epsilon_1 \oplus \epsilon_2 \oplus \epsilon_3 \oplus \epsilon_4 = 0 \Rightarrow i = 5, 6$ skipped

$b_{-1} \leftarrow 0,\ b_0 \leftarrow 1,\ r_{-1} \leftarrow g(Y),\ r_0 \leftarrow S(Y),\ i \leftarrow 0$
while $\deg(r_i) > 0$ **do**
  $i \leftarrow i + 1$
  $(q_i(Y), r_i(Y)) \leftarrow r_{i-2}(Y)/r_{i-1}(Y)$
  $b_i(Y) \leftarrow b_{i-2}(Y) + q_i(Y) b_{i-1}(Y)$
end while

we know: $\exists i : \sigma(Y) = b_i(Y) \wedge \Omega(Y) = r_i(Y)$

$$S(Y) \equiv \sum_{i=1}^{4} \frac{1}{Y \oplus \epsilon_i} \equiv \frac{\Omega(Y)}{\sigma(Y)} \equiv \frac{\sigma_3 Y^2 \oplus \sigma_1}{Y^4 \oplus \sigma_3 Y^3 \oplus \sigma_2 Y^2 \oplus \sigma_1 Y \oplus \sigma_0} \mod g(Y)$$

i	$\deg(q_i(Y))$	$\deg(b_i(Y))$	$\deg(r_i(Y))$
1	1	1	t-2
2	1	2	t-3
3	1	3	t-4
4	1	4	2 0
5	t - 6	t - 2	1
6	1	t - 1	0

$\sigma_3 = \epsilon_1 \oplus \epsilon_2 \oplus \epsilon_3 \oplus \epsilon_4 = 0 \Rightarrow i = 5, 6$ skipped

$b_{-1} \leftarrow 0,\ b_0 \leftarrow 1,\ r_{-1} \leftarrow g(Y),\ r_0 \leftarrow S(Y),\ i \leftarrow 0$

while $\deg(r_i) > 0$ **do**

  $i \leftarrow i + 1$

  $(q_i(Y), r_i(Y)) \leftarrow r_{i-2}(Y)/r_{i-1}(Y)$

  $b_i(Y) \leftarrow b_{i-2}(Y) + q_i(Y) b_{i-1}(Y)$

end while

we know: $\exists i : \sigma(Y) = b_i(Y) \wedge \Omega(Y) = r_i(Y)$

$$S(Y) \equiv \sum_{i=1}^{4} \frac{1}{Y \oplus \epsilon_i} \equiv \frac{\Omega(Y)}{\sigma(Y)} \equiv \frac{\sigma_3 Y^2 \oplus \sigma_1}{Y^4 \oplus \sigma_3 Y^3 \oplus \sigma_2 Y^2 \oplus \sigma_1 Y \oplus \sigma_0} \mod g(Y)$$

i	$\deg(q_i(Y))$	$\deg(b_i(Y))$	$\deg(r_i(Y))$
1	1	1	t-2
2	1	2	t-3
3	1	3	t-4
4	1	4	2 0
5	t - 6	t - 2	1
6	1	t - 1	0

$\sigma_3 = \epsilon_1 \oplus \epsilon_2 \oplus \epsilon_3 \oplus \epsilon_4 = 0 \Rightarrow i = 5, 6$ skipped

The Syndrome Inversion EEA for $w = 4$

$b_{-1} \leftarrow 0,\ b_0 \leftarrow 1,\ r_{-1} \leftarrow g(Y),\ r_0 \leftarrow S(Y),\ i \leftarrow 0$
while $\deg(r_i) > 0$ **do**
  $i \leftarrow i + 1$
  $(q_i(Y), r_i(Y)) \leftarrow r_{i-2}(Y)/r_{i-1}(Y)$
  $b_i(Y) \leftarrow b_{i-2}(Y) + q_i(Y) b_{i-1}(Y)$
end while

we know: $\exists i : \sigma(Y) = b_i(Y) \wedge \Omega(Y) = r_i(Y)$

$$S(Y) \equiv \sum_{i=1}^{4} \frac{1}{Y \oplus \epsilon_i} \equiv \frac{\Omega(Y)}{\sigma(Y)} \equiv \frac{\sigma_3 Y^2 \oplus \sigma_1}{Y^4 \oplus \sigma_3 Y^3 \oplus \sigma_2 Y^2 \oplus \sigma_1 Y \oplus \sigma_0} \mod g(Y)$$

i	$\deg(q_i(Y))$	$\deg(b_i(Y))$	$\deg(r_i(Y))$
1	1	1	t-2
2	1	2	t-3
3	1	3	t-4
4	1	4	2 0
5	t - 6	t - 2	1
6	1	t - 1	0

$\sigma_3 = \epsilon_1 \oplus \epsilon_2 \oplus \epsilon_3 \oplus \epsilon_4 = 0 \Rightarrow i = 5, 6$ skipped

Weight 6 Vulnerability

$$S(Y) \equiv \frac{\sigma_5 Y^4 \oplus \sigma_3 Y^2 \oplus \sigma_1}{Y^6 \oplus \sigma_5 Y^5 \oplus \sigma_4 Y^4 \oplus \sigma_3 Y^3 \oplus \sigma_2 Y^2 \oplus \sigma_1 Y + \sigma_0} \bmod g(Y),$$

- $\sigma_5 = \sum_{i=1}^{6} \epsilon_i$
- $\sigma_3 = \sum_{j=3}^{6} \sum_{k=1}^{j-1} \sum_{l=1}^{k-1} \epsilon_j \epsilon_k \epsilon_l$

Weight 6 Vulnerability

$$S(Y) \equiv \frac{\sigma_5 Y^4 \oplus \sigma_3 Y^2 \oplus \sigma_1}{Y^6 \oplus \sigma_5 Y^5 \oplus \sigma_4 Y^4 \oplus \sigma_3 Y^3 \oplus \sigma_2 Y^2 \oplus \sigma_1 Y + \sigma_0} \mod g(Y),$$

- $\sigma_5 = \sum_{i=1}^{6} \epsilon_i$
- $\sigma_3 = \sum_{j=3}^{6} \sum_{k=1}^{j-1} \sum_{l=1}^{k-1} \epsilon_j \epsilon_k \epsilon_l$

Weight 1 Vulnerability

- if $\epsilon_1 = 0$, one fewer iteration in polynomial division inside the syndrome inversion EEA
- z with $\alpha_z = 0$ becomes known

Weight 1 Vulnerability

- if $\epsilon_1 = 0$, one fewer iteration in polynomial division inside the syndrome inversion EEA
- z with $\alpha_z = 0$ becomes known

Building the Attack

- always: maximal rank from $w = 4$ is $n - m - 1$
- most of the times: knowledge about z (with $\alpha_z = 0$) increases rank to $n - m$

α_0	α_1	$\dots$	α_i	$\dots$	α_{n-m-3}	α_{n-m-2}	β_0	$\dots$	β_{m-1}
1	0	$\dots$	0	$\dots$	0	0	X	$\dots$	X
$\vdots$									
0	0	$\dots$	1	$\dots$	0	0	X	$\dots$	X
$\vdots$									
0	0	$\dots$	0	$\dots$	0	1	X	$\dots$	X

- $\alpha_i = \sum_{j \in B_i} \beta_j$

Building the Attack

- always: maximal rank from $w = 4$ is $n - m - 1$
- most of the times: knowledge about z (with $\alpha_z = 0$) increases rank to $n - m$

α_0	α_1	$\dots$	α_i	$\dots$	α_{n-m-3}	α_{n-m-2}	β_0	$\dots$	β_{m-1}
1	0	$\dots$	0	$\dots$	0	0	X	$\dots$	X
$\vdots$									
0	0	$\dots$	1	$\dots$	0	0	X	$\dots$	X
$\vdots$									
0	0	$\dots$	0	$\dots$	0	1	X	$\dots$	X

- $\alpha_i = \sum_{j \in B_i} \beta_j$

Building the Attack

- always: maximal rank from $w = 4$ is $n - m - 1$
- most of the times: knowledge about z (with $\alpha_z = 0$) increases rank to $n - m$

α_0	α_1	$\dots$	α_i	$\dots$	α_{n-m-3}	α_{n-m-2}	β_0	$\dots$	β_{m-1}
1	0	$\dots$	0	$\dots$	0	0	X	$\dots$	X
$\vdots$									
0	0	$\dots$	1	$\dots$	0	0	X	$\dots$	X
$\vdots$									
0	0	$\dots$	0	$\dots$	0	1	X	$\dots$	X

- $\alpha_i = \sum_{j \in B_i} \beta_j$

Building the Attack

- always: maximal rank from $w = 4$ is $n - m - 1$
- most of the times: knowledge about z (with $\alpha_z = 0$) increases rank to $n - m$

α_0	α_1	$\dots$	α_i	$\dots$	α_{n-m-3}	α_{n-m-2}	β_0	$\dots$	β_{m-1}
1	0	$\dots$	0	$\dots$	0	0	X	$\dots$	X
$\vdots$									
0	0	$\dots$	1	$\dots$	0	0	X	$\dots$	X
$\vdots$									
0	0	$\dots$	0	$\dots$	0	1	X	$\dots$	X

- $\alpha_i = \sum_{j \in B_i} \beta_j$

Collecting cubic Equations

$$\Omega(Y) = \sigma_5 Y^4 \oplus \sigma_3 Y^2 \oplus \sigma_1$$

$$\begin{array}{llllllll}
C_1: & \beta_3 & \leftarrow & \beta_0, & \beta_1, & \beta_2 & \\
C_2: & \beta_4 & \leftarrow & \beta_0, & \beta_1, & \beta_2, & \beta_3 \\
\vdots & \vdots & \vdots & \vdots & \vdots & & \\
C_{m-3}: & \beta_{m-1} & \leftarrow & \beta_0, & \beta_1, & \dots & \beta_{m-2}
\end{array}$$

- for $i = 1, \dots, 6$: $\epsilon_i \in \mathrm{span}(\beta_0, \beta_1, \beta_2, \beta_3)$
- $\sigma_5 = \sum_{j=1}^{6} \epsilon_j = \sum_{j=1}^{6} \sum_{i \in B_{f_j}} \beta_i = 0 \Leftrightarrow$ count of each β_i across ϵ_i *even*
- $\sigma_3 = \sum_{j=3}^{6} \sum_{k=1}^{j-1} \sum_{l=1}^{k-1} \epsilon_j \epsilon_k \epsilon_l = 0$ (through timing)
- $\rightarrow \sum_{j=3}^{6} \sum_{k=1}^{j-1} \sum_{l=1}^{k-1} \left(\sum_{i \in B_{f_j}} \beta_i\right) \left(\sum_{i \in B_{f_k}} \beta_i\right) \left(\sum_{i \in B_{f_l}} \beta_i\right) = 0$
- count of β_3 is 2:

$$a\beta_3^2 + b\beta_3 + c = 0$$

Collecting cubic Equations

$$\Omega(Y) = \sigma_5 Y^4 \oplus \sigma_3 Y^2 \oplus \sigma_1$$

$$\begin{array}{llllll} C_1: & \beta_3 & \leftarrow & \beta_0, & \beta_1, & \beta_2 \\ C_2: & \beta_4 & \leftarrow & \beta_0, & \beta_1, & \beta_2, \quad \beta_3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \\ C_{m-3}: & \beta_{m-1} & \leftarrow & \beta_0, & \beta_1, & \dots \quad \beta_{m-2} \end{array}$$

- for $i = 1, \dots, 6$: $\epsilon_i \in \mathrm{span}(\beta_0, \beta_1, \beta_2, \beta_3)$
- $\sigma_5 = \sum_{j=1}^{6} \epsilon_j = \sum_{j=1}^{6} \sum_{i \in B_{f_j}} \beta_i = 0 \Leftrightarrow$ count of each β_i across ϵ_i *even*
- $\sigma_3 = \sum_{j=3}^{6} \sum_{k=1}^{j-1} \sum_{l=1}^{k-1} \epsilon_j \epsilon_k \epsilon_l = 0$ (through timing)
- $\rightarrow \sum_{j=3}^{6} \sum_{k=1}^{j-1} \sum_{l=1}^{k-1} \left(\sum_{i \in B_{f_j}} \beta_i \right) \left(\sum_{i \in B_{f_k}} \beta_i \right) \left(\sum_{i \in B_{f_l}} \beta_i \right) = 0$
- count of β_3 is 2:

$$a\beta_3^2 + b\beta_3 + c = 0$$

Collecting cubic Equations

$$\Omega(Y) = \sigma_5 Y^4 \oplus \sigma_3 Y^2 \oplus \sigma_1$$

$$\begin{array}{llllllll}
C_1: & \beta_3 & \leftarrow & \beta_0, & \beta_1, & \beta_2 & \\
C_2: & \beta_4 & \leftarrow & \beta_0, & \beta_1, & \beta_2, & \beta_3 \\
\vdots & \vdots & \vdots & \vdots & \vdots & & \\
C_{m-3}: & \beta_{m-1} & \leftarrow & \beta_0, & \beta_1, & \ldots & \beta_{m-2}
\end{array}$$

- for $i = 1, \ldots, 6$: $\epsilon_i \in \mathrm{span}(\beta_0, \beta_1, \beta_2, \beta_3)$
- $\sigma_5 = \sum_{j=1}^{6} \epsilon_j = \sum_{j=1}^{6} \sum_{i \in B_{f_j}} \beta_i = 0 \Leftrightarrow$ count of each β_i across ϵ_i *even*
- $\sigma_3 = \sum_{j=3}^{6} \sum_{k=1}^{j-1} \sum_{l=1}^{k-1} \epsilon_j \epsilon_k \epsilon_l = 0$ (through timing)
- $\rightarrow \sum_{j=3}^{6} \sum_{k=1}^{j-1} \sum_{l=1}^{k-1} \left(\sum_{i \in B_{f_j}} \beta_i\right) \left(\sum_{i \in B_{f_k}} \beta_i\right) \left(\sum_{i \in B_{f_l}} \beta_i\right) = 0$
- count of β_3 is 2:

$$a\beta_3^2 + b\beta_3 + c = 0$$

Collecting cubic Equations

$\Omega(Y) = \sigma_5 Y^4 \oplus \sigma_3 Y^2 \oplus \sigma_1$

$$
\begin{array}{llllllll}
C_1: & \beta_3 & \leftarrow & \beta_0, & \beta_1, & \beta_2 & \\
C_2: & \beta_4 & \leftarrow & \beta_0, & \beta_1, & \beta_2, & \beta_3 \\
\vdots & \vdots & \vdots & \vdots & \vdots & & \\
C_{m-3}: & \beta_{m-1} & \leftarrow & \beta_0, & \beta_1, & \dots & \beta_{m-2}
\end{array}
$$

- for $i = 1, \dots, 6$: $\epsilon_i \in \mathrm{span}(\beta_0, \beta_1, \beta_2, \beta_3)$
- $\sigma_5 = \sum_{j=1}^{6} \epsilon_j = \sum_{j=1}^{6} \sum_{i \in B_{f_j}} \beta_i = 0 \Leftrightarrow$ count of each β_i across ϵ_i *even*
- $\sigma_3 = \sum_{j=3}^{6} \sum_{k=1}^{j-1} \sum_{l=1}^{k-1} \epsilon_j \epsilon_k \epsilon_l = 0$ (through timing)
- $\rightarrow \sum_{j=3}^{6} \sum_{k=1}^{j-1} \sum_{l=1}^{k-1} \left(\sum_{i \in B_{f_j}} \beta_i\right) \left(\sum_{i \in B_{f_k}} \beta_i\right) \left(\sum_{i \in B_{f_l}} \beta_i\right) = 0$
- count of β_3 is 2:

$$a\beta_3^2 + b\beta_3 + c = 0$$

Collecting cubic Equations

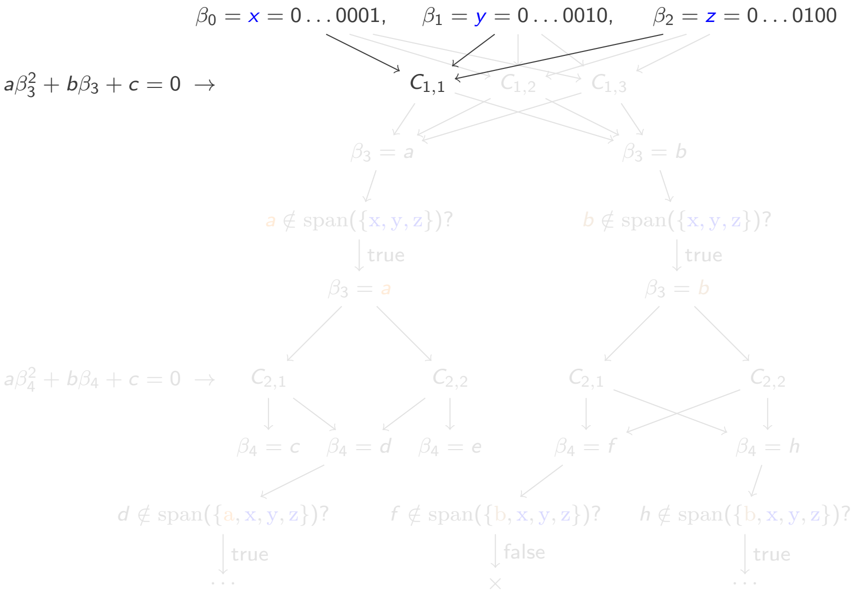
$$\Omega(Y) = \sigma_5 Y^4 \oplus \sigma_3 Y^2 \oplus \sigma_1$$

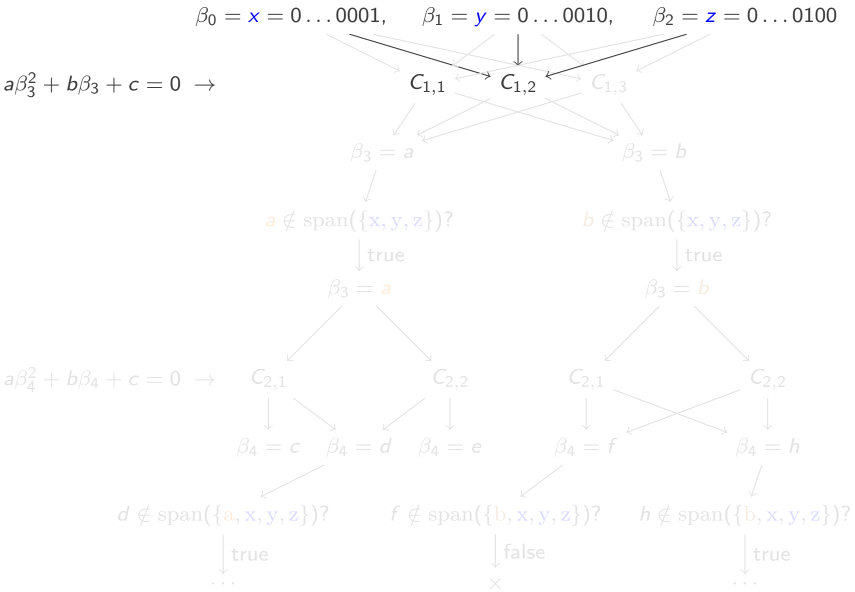
$$\begin{array}{lcccccc}
C_1: & \beta_3 & \leftarrow & \beta_0, & \beta_1, & \beta_2 & \\
C_2: & \beta_4 & \leftarrow & \beta_0, & \beta_1, & \beta_2, & \beta_3 \\
\vdots & \vdots & \vdots & \vdots & \vdots & & \\
C_{m-3}: & \beta_{m-1} & \leftarrow & \beta_0, & \beta_1, & \dots & \beta_{m-2}
\end{array}$$

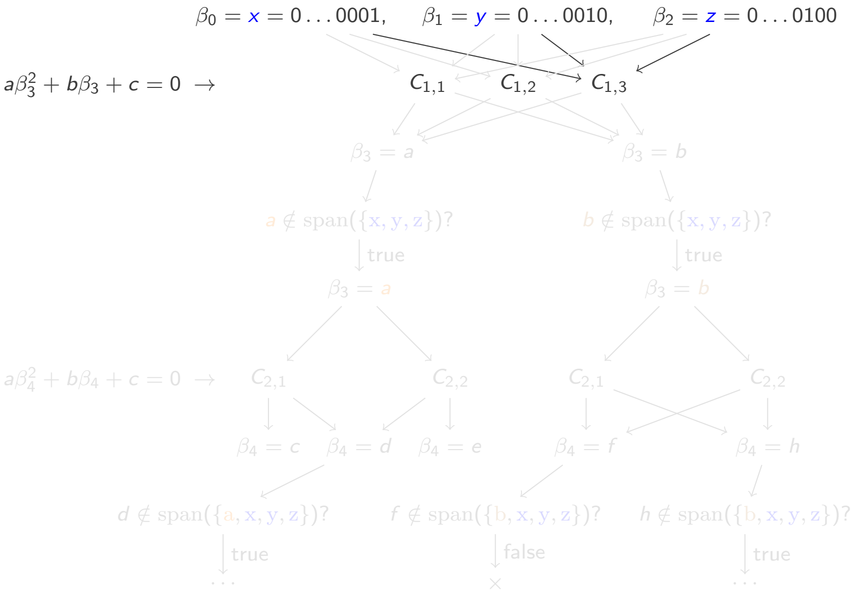
- for $i = 1, \dots, 6$: $\epsilon_i \in \mathrm{span}(\beta_0, \beta_1, \beta_2, \beta_3)$
- $\sigma_5 = \sum_{j=1}^{6} \epsilon_j = \sum_{j=1}^{6} \sum_{i \in B_{f_j}} \beta_i = 0 \Leftrightarrow$ count of each β_i across ϵ_i *even*
- $\sigma_3 = \sum_{j=3}^{6} \sum_{k=1}^{j-1} \sum_{l=1}^{k-1} \epsilon_j \epsilon_k \epsilon_l = 0$ (through timing)
- $\rightarrow \sum_{j=3}^{6} \sum_{k=1}^{j-1} \sum_{l=1}^{k-1} \left(\sum_{i \in B_{f_j}} \beta_i\right) \left(\sum_{i \in B_{f_k}} \beta_i\right) \left(\sum_{i \in B_{f_l}} \beta_i\right) = 0$
- count of β_3 is 2:

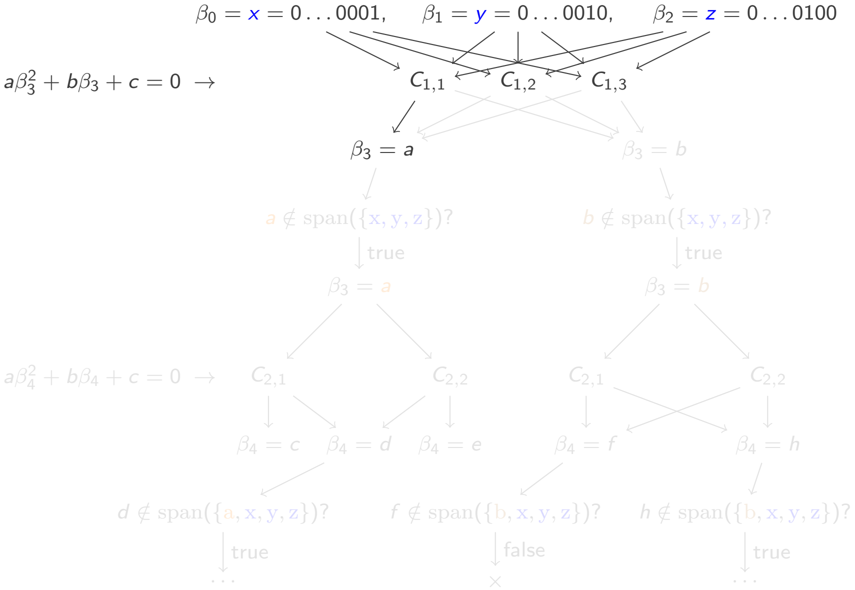
$$a\beta_3^2 + b\beta_3 + c = 0$$

$$\beta_0 = x = 0\ldots0001, \quad \beta_1 = y = 0\ldots0010, \quad \beta_2 = z = 0\ldots0100$$

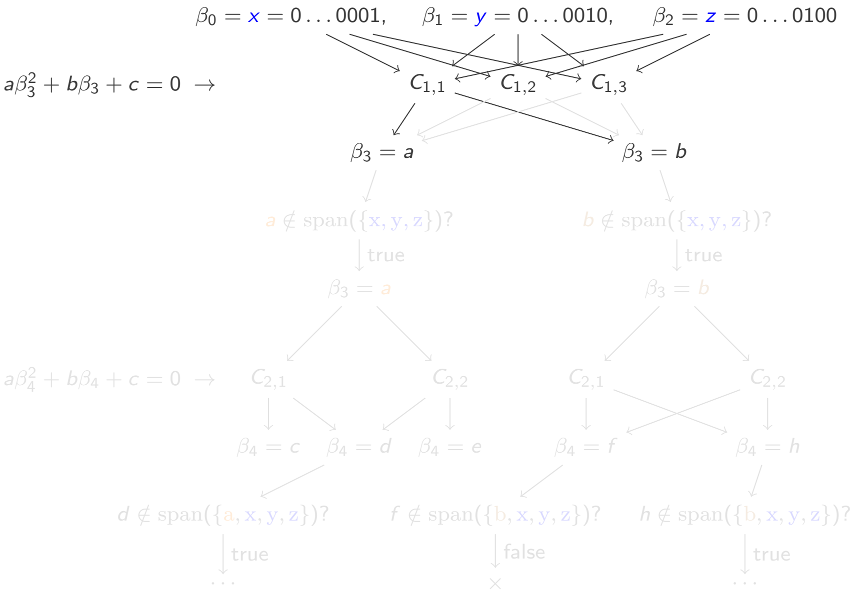

$\beta_0 = x = 0\ldots0001,\quad \beta_1 = y = 0\ldots0010,\quad \beta_2 = z = 0\ldots0100$
$a\beta_3^2 + b\beta_3 + c = 0 \rightarrow$
$C_{1,1}$
$C_{1,2}$
$C_{1,3}$
$\beta_3 = a$
$\beta_3 = b$
$a \notin \mathrm{span}(\{x, y, z\})?$
$b \notin \mathrm{span}(\{x, y, z\})?$
true
true
$\beta_3 = a$
$\beta_3 = b$
$a\beta_4^2 + b\beta_4 + c = 0 \rightarrow$
$C_{2,1}$
$C_{2,2}$
$C_{2,1}$
$C_{2,2}$
$\beta_4 = c$
$\beta_4 = d$
$\beta_4 = e$
$\beta_4 = f$
$\beta_4 = h$
$d \notin \mathrm{span}(\{a, x, y, z\})?$
$f \notin \mathrm{span}(\{b, x, y, z\})?$
$h \notin \mathrm{span}(\{b, x, y, z\})?$
true
false
true
$\ldots$
$\times$
$\ldots$


$\beta_0 = x = 0\ldots0001,\quad \beta_1 = y = 0\ldots0010,\quad \beta_2 = z = 0\ldots0100$
$a\beta_3^2 + b\beta_3 + c = 0 \rightarrow$
$C_{1,1}$
$C_{1,2}$
$C_{1,3}$
$\beta_3 = a$
$\beta_3 = b$
$a \notin \mathrm{span}(\{x, y, z\})$?
$b \notin \mathrm{span}(\{x, y, z\})$?
true
true
$\beta_3 = a$
$\beta_3 = b$
$a\beta_4^2 + b\beta_4 + c = 0 \rightarrow$
$C_{2,1}$
$C_{2,2}$
$C_{2,1}$
$C_{2,2}$
$\beta_4 = c$
$\beta_4 = d$
$\beta_4 = e$
$\beta_4 = f$
$\beta_4 = h$
$d \notin \mathrm{span}(\{a, x, y, z\})$?
$f \notin \mathrm{span}(\{b, x, y, z\})$?
$h \notin \mathrm{span}(\{b, x, y, z\})$?
true
false
true
. . .
. . .


β0 = x = 0...0001,
β1 = y = 0...0010,
β2 = z = 0...0100
aβ3² + bβ3 + c = 0 →
C1,1
C1,2
C1,3
β3 = a
β3 = b
a ∉ span({x, y, z})?
b ∉ span({x, y, z})?
true
true
β3 = a
β3 = b
aβ4² + bβ4 + c = 0 →
C2,1
C2,2
C2,1
C2,2
β4 = c
β4 = d
β4 = e
β4 = f
β4 = h
d ∉ span({a, x, y, z})?
f ∉ span({b, x, y, z})?
h ∉ span({b, x, y, z})?
true
false
true
...
×
...

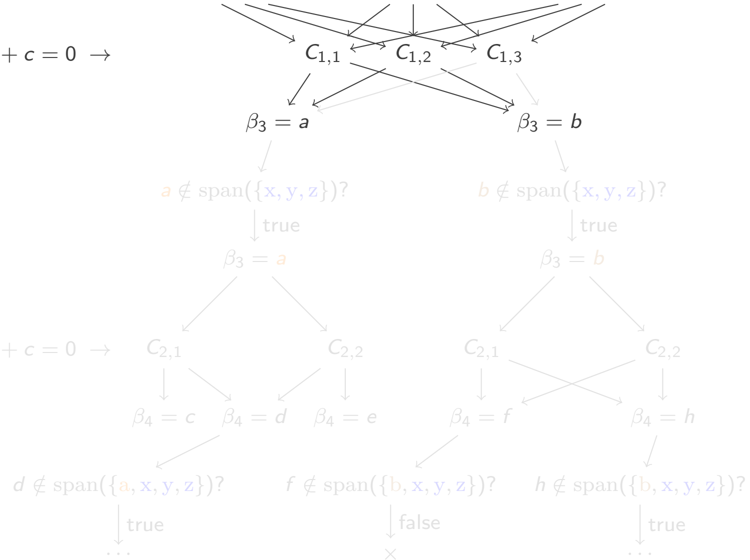


$\beta_0 = x = 0\ldots0001$, $\beta_1 = y = 0\ldots0010$, $\beta_2 = z = 0\ldots0100$
$a\beta_3^2 + b\beta_3 + c = 0 \rightarrow$
$C_{1,1}$ $C_{1,2}$ $C_{1,3}$
$\beta_3 = a$
$\beta_3 = b$
$a \notin \mathrm{span}(\{x, y, z\})$?
$b \notin \mathrm{span}(\{x, y, z\})$?
true
true
$\beta_3 = a$
$\beta_3 = b$
$a\beta_4^2 + b\beta_4 + c = 0 \rightarrow$
$C_{2,1}$ $C_{2,2}$ $C_{2,1}$ $C_{2,2}$
$\beta_4 = c$ $\beta_4 = d$ $\beta_4 = e$ $\beta_4 = f$ $\beta_4 = h$
$d \notin \mathrm{span}(\{a, x, y, z\})$?
$f \notin \mathrm{span}(\{b, x, y, z\})$?
$h \notin \mathrm{span}(\{b, x, y, z\})$?
true
false
true
$\ldots$
$\times$
$\ldots$

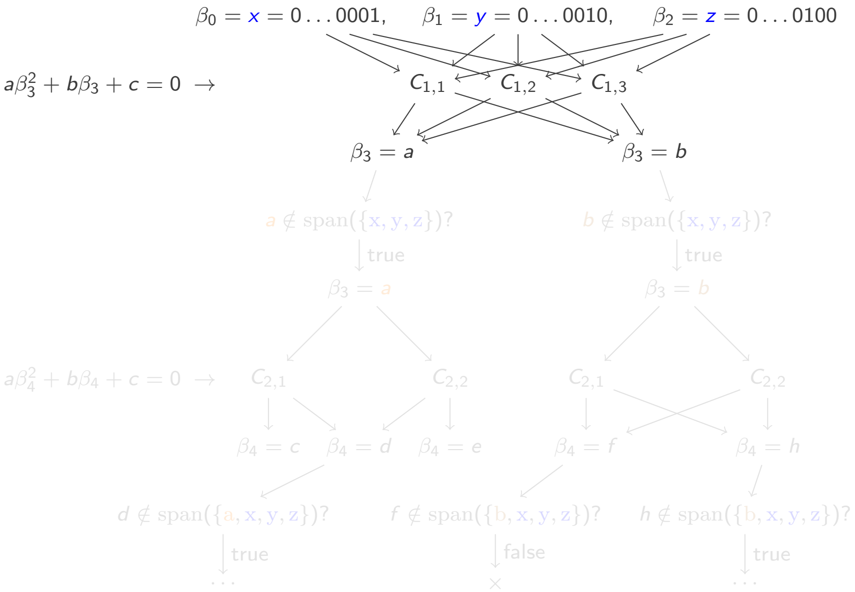

β0 = x = 0...0001, β1 = y = 0...0010, β2 = z = 0...0100
aβ3² + bβ3 + c = 0 →
C1,1 C1,2 C1,3
β3 = a
β3 = b
a ∉ span({x, y, z})?
b ∉ span({x, y, z})?
true
true
β3 = a
β3 = b
aβ4² + bβ4 + c = 0 →
C2,1 C2,2 C2,1 C2,2
β4 = c β4 = d β4 = e β4 = f β4 = h
d ∉ span({a, x, y, z})?
f ∉ span({b, x, y, z})?
h ∉ span({b, x, y, z})?
true
false
true
...
×
...

$\beta_0 = x = 0\ldots0001, \quad \beta_1 = y = 0\ldots0010, \quad \beta_2 = z = 0\ldots0100$

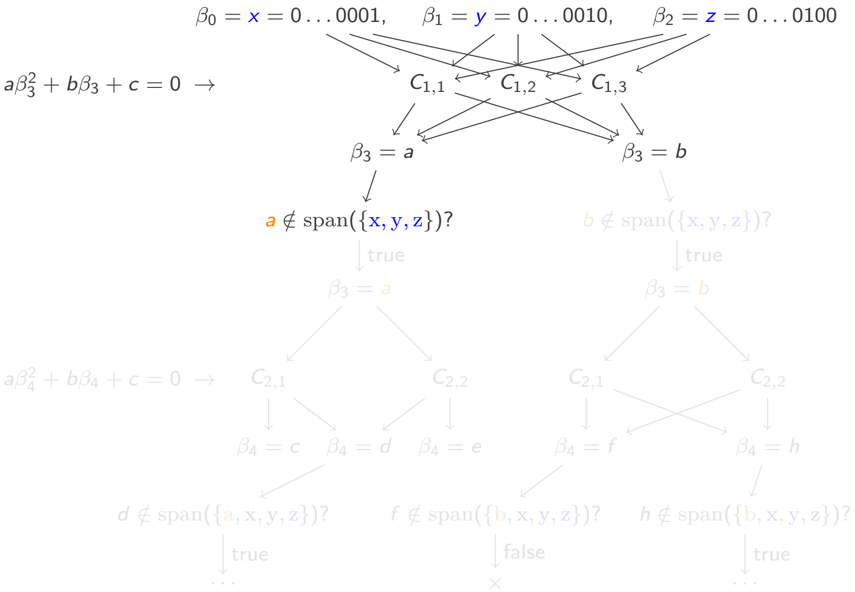
$a\beta_3^2 + b\beta_3 + c = 0 \rightarrow$



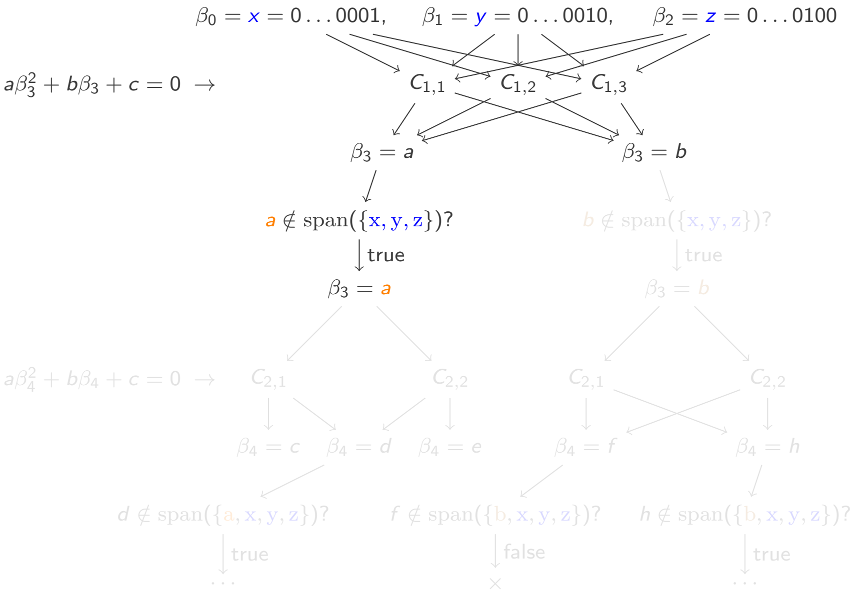
$a\beta_4^2 + b\beta_4 + c = 0 \rightarrow$



$\beta_0 = x = 0\ldots0001$, $\beta_1 = y = 0\ldots0010$, $\beta_2 = z = 0\ldots0100$
$a\beta_3^2 + b\beta_3 + c = 0 \rightarrow$
$C_{1,1}$ $C_{1,2}$ $C_{1,3}$
$\beta_3 = a$ $\beta_3 = b$
$a \notin \mathrm{span}(\{x, y, z\})$?
$b \notin \mathrm{span}(\{x, y, z\})$?
true
true
$\beta_3 = a$ $\beta_3 = b$
$a\beta_4^2 + b\beta_4 + c = 0 \rightarrow$
$C_{2,1}$ $C_{2,2}$ $C_{2,1}$ $C_{2,2}$
$\beta_4 = c$ $\beta_4 = d$ $\beta_4 = e$ $\beta_4 = f$ $\beta_4 = h$
$d \notin \mathrm{span}(\{a, x, y, z\})$?
$f \notin \mathrm{span}(\{b, x, y, z\})$?
$h \notin \mathrm{span}(\{b, x, y, z\})$?
true
false
true
$\ldots$
$\times$
$\ldots$



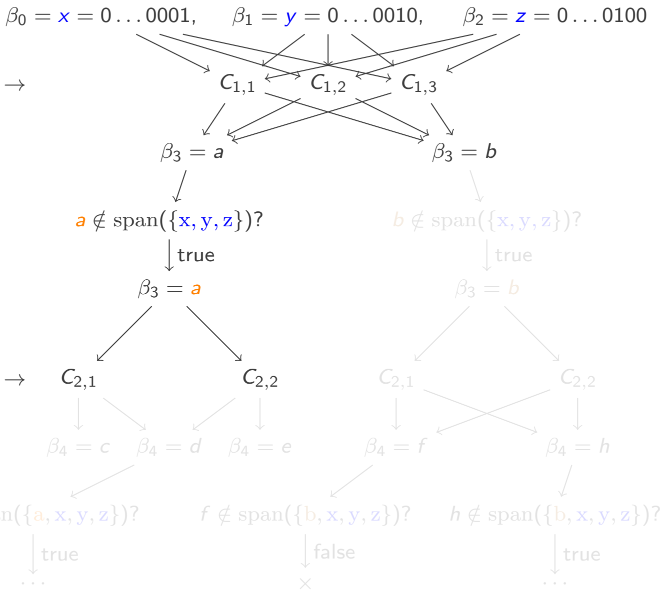
β0 = x = 0...0001,
β1 = y = 0...0010,
β2 = z = 0...0100
aβ3² + bβ3 + c = 0 →
C1,1
C1,2
C1,3
β3 = a
β3 = b
a ∉ span({x, y, z})?
b ∉ span({x, y, z})?
true
true
β3 = a
β3 = b
aβ4² + bβ4 + c = 0 →
C2,1
C2,2
C2,1
C2,2
β4 = c
β4 = d
β4 = e
β4 = f
β4 = h
d ∉ span({a, x, y, z})?
f ∉ span({b, x, y, z})?
h ∉ span({b, x, y, z})?
true
false
true
...
...

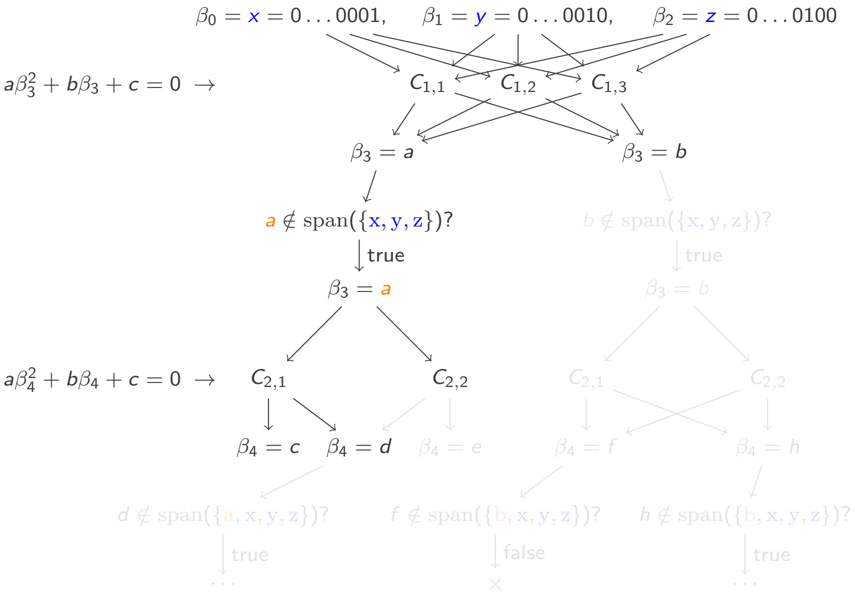


$\beta_0 = x = 0\ldots0001, \quad \beta_1 = y = 0\ldots0010, \quad \beta_2 = z = 0\ldots0100$
$a\beta_3^2 + b\beta_3 + c = 0 \rightarrow$
$C_{1,1}$ $C_{1,2}$ $C_{1,3}$
$\beta_3 = a$
$\beta_3 = b$
$a \notin \text{span}(\{x, y, z\})$?
true
$\beta_3 = a$
$b \notin \text{span}(\{x, y, z\})$?
true
$\beta_3 = b$
$a\beta_4^2 + b\beta_4 + c = 0 \rightarrow$
$C_{2,1}$ $C_{2,2}$ $C_{2,1}$ $C_{2,2}$
$\beta_4 = c$ $\beta_4 = d$ $\beta_4 = e$ $\beta_4 = f$ $\beta_4 = h$
$d \notin \text{span}(\{a, x, y, z\})$?
true
$\ldots$
$f \notin \text{span}(\{b, x, y, z\})$?
false
$\times$
$h \notin \text{span}(\{b, x, y, z\})$?
true
$\ldots$

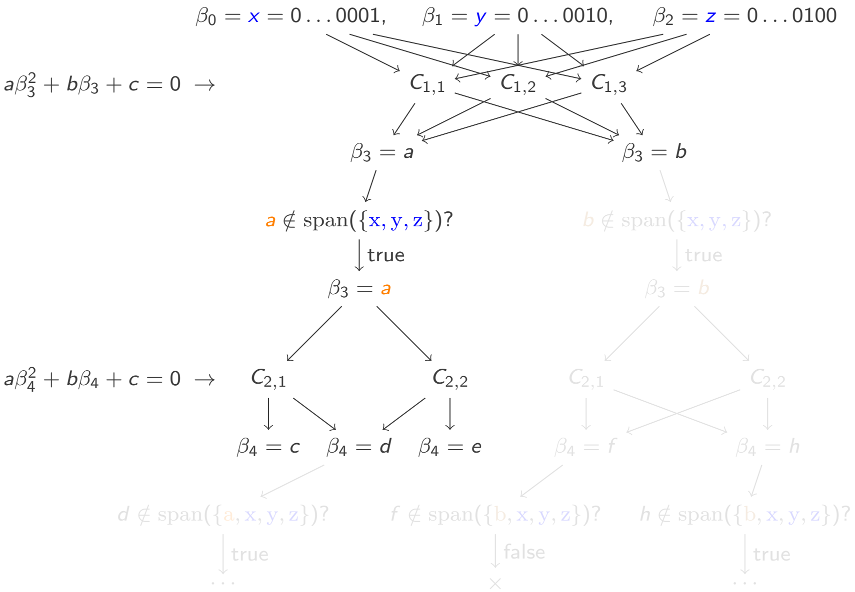
$a\beta_3^2 + b\beta_3 + c = 0 \rightarrow$

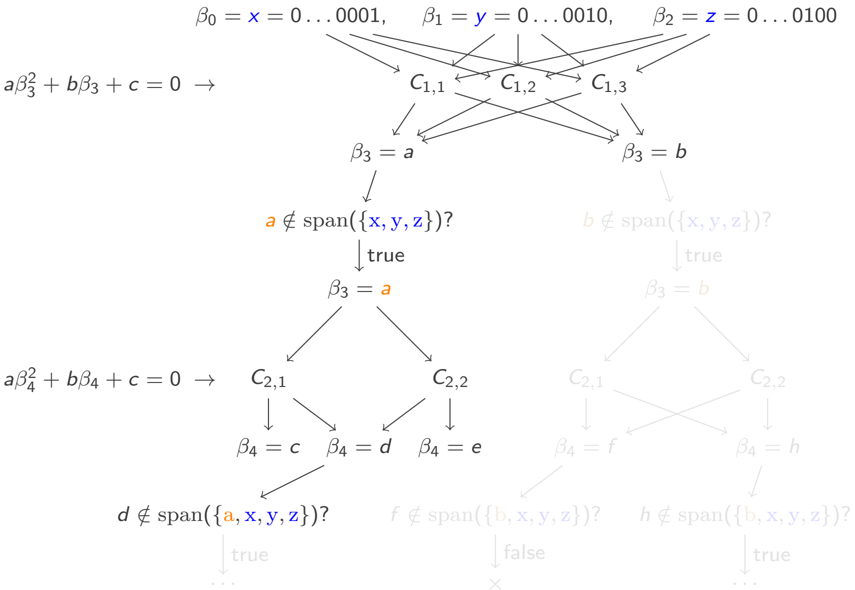
$a\beta_4^2 + b\beta_4 + c = 0 \rightarrow$

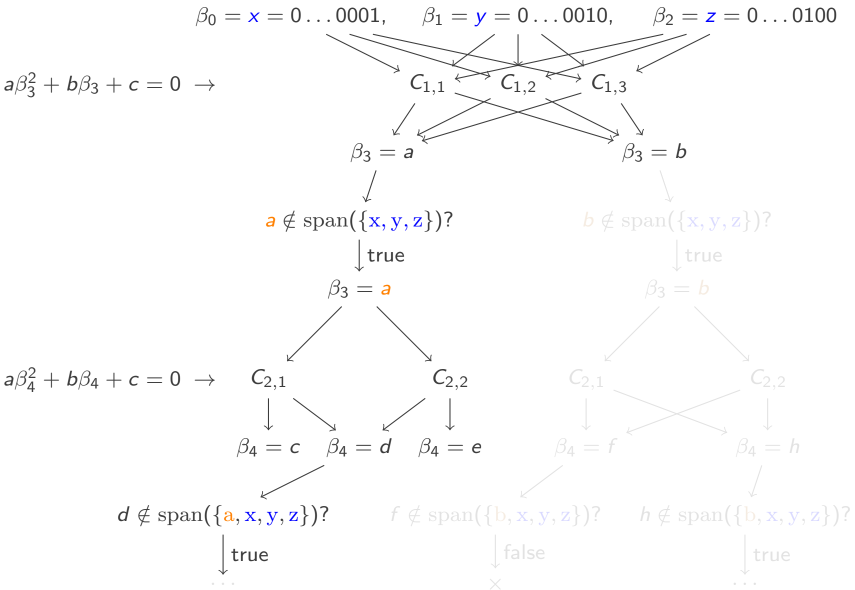


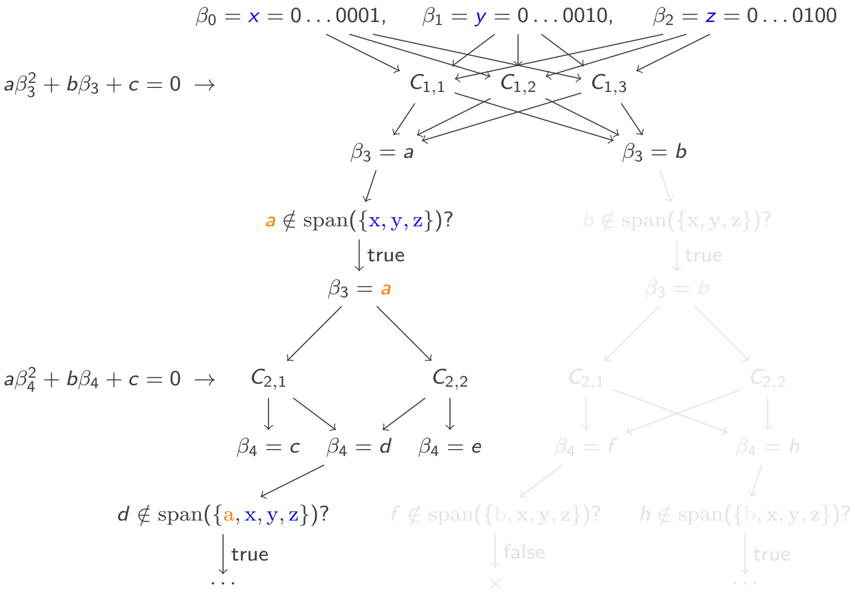


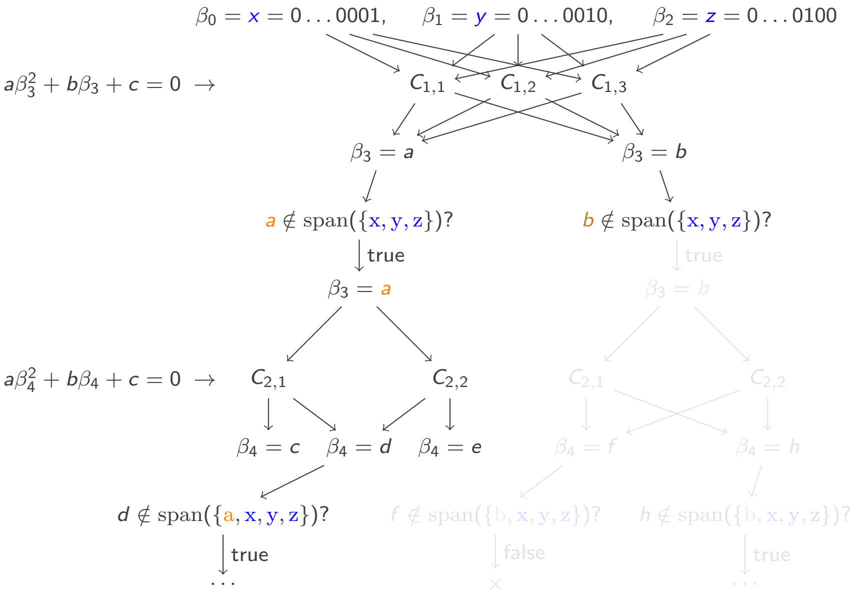
β0 = x = 0...0001, β1 = y = 0...0010, β2 = z = 0...0100
aβ3² + bβ3 + c = 0 →
C1,1 C1,2 C1,3
β3 = a
β3 = b
a ∉ span({x, y, z})?
true
β3 = a
b ∉ span({x, y, z})?
true
β3 = b
aβ4² + bβ4 + c = 0 →
C2,1 C2,2
C2,1 C2,2
β4 = c β4 = d β4 = e
β4 = f β4 = h
d ∉ span({a, x, y, z})?
true
...
f ∉ span({b, x, y, z})?
false
×
h ∉ span({b, x, y, z})?
true
...


$\beta_0 = x = 0\ldots0001$, $\beta_1 = y = 0\ldots0010$, $\beta_2 = z = 0\ldots0100$
$a\beta_3^2 + b\beta_3 + c = 0 \rightarrow$
$C_{1,1}$ $C_{1,2}$ $C_{1,3}$
$\beta_3 = a$
$\beta_3 = b$
$a \notin \mathrm{span}(\{x, y, z\})$?
$b \notin \mathrm{span}(\{x, y, z\})$?
true
true
$\beta_3 = a$
$\beta_3 = b$
$a\beta_4^2 + b\beta_4 + c = 0 \rightarrow$
$C_{2,1}$ $C_{2,2}$ $C_{2,1}$ $C_{2,2}$
$\beta_4 = c$ $\beta_4 = d$ $\beta_4 = e$ $\beta_4 = f$ $\beta_4 = h$
$d \notin \mathrm{span}(\{a, x, y, z\})$?
$f \notin \mathrm{span}(\{b, x, y, z\})$?
$h \notin \mathrm{span}(\{b, x, y, z\})$?
true
false
true
...
×
...

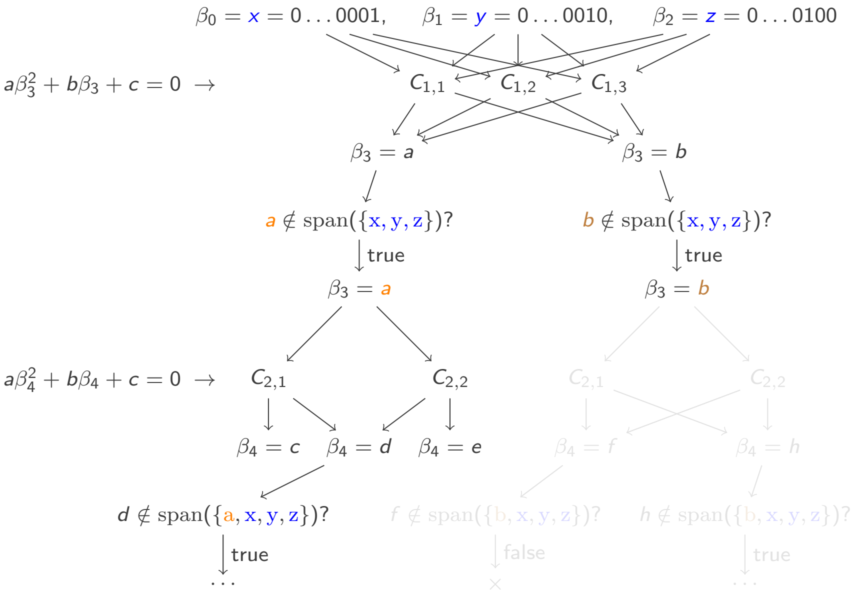

$\beta_0 = x = 0\ldots0001,$ $\beta_1 = y = 0\ldots0010,$ $\beta_2 = z = 0\ldots0100$
$a\beta_3^2 + b\beta_3 + c = 0 \rightarrow$
$C_{1,1}$ $C_{1,2}$ $C_{1,3}$
$\beta_3 = a$ $\beta_3 = b$
$a \notin \mathrm{span}(\{x, y, z\})?$
$b \notin \mathrm{span}(\{x, y, z\})?$
true
true
$\beta_3 = a$ $\beta_3 = b$
$a\beta_4^2 + b\beta_4 + c = 0 \rightarrow$
$C_{2,1}$ $C_{2,2}$ $C_{2,1}$ $C_{2,2}$
$\beta_4 = c$ $\beta_4 = d$ $\beta_4 = e$ $\beta_4 = f$ $\beta_4 = h$
$d \notin \mathrm{span}(\{a, x, y, z\})?$
$f \notin \mathrm{span}(\{b, x, y, z\})?$
$h \notin \mathrm{span}(\{b, x, y, z\})?$
true
false
true
...
...

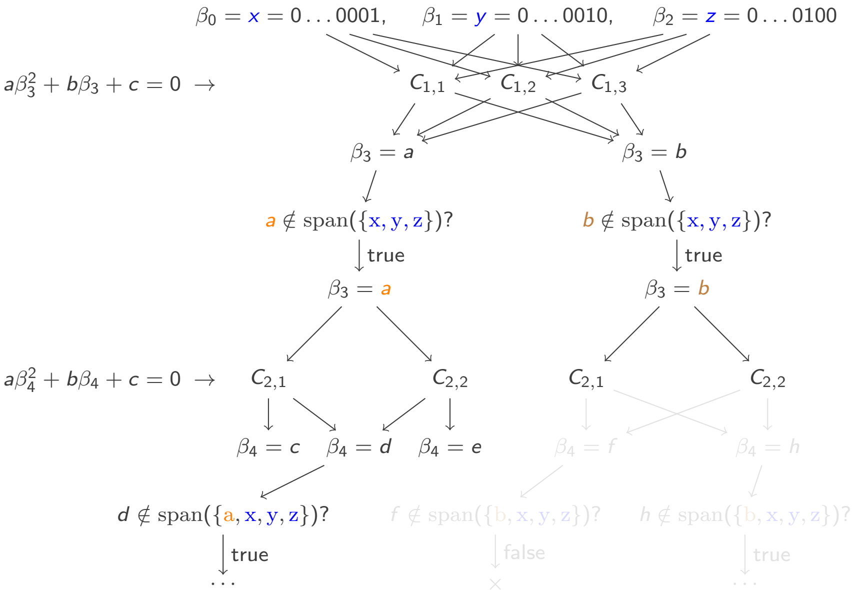

$\beta_0 = x = 0\dots0001$, $\beta_1 = y = 0\dots0010$, $\beta_2 = z = 0\dots0100$
$a\beta_3^2 + b\beta_3 + c = 0 \rightarrow$
$C_{1,1}$ $C_{1,2}$ $C_{1,3}$
$\beta_3 = a$
$\beta_3 = b$
$a \notin \mathrm{span}(\{x, y, z\})$?
$b \notin \mathrm{span}(\{x, y, z\})$?
true
true
$\beta_3 = a$
$\beta_3 = b$
$a\beta_4^2 + b\beta_4 + c = 0 \rightarrow$
$C_{2,1}$ $C_{2,2}$ $C_{2,1}$ $C_{2,2}$
$\beta_4 = c$ $\beta_4 = d$ $\beta_4 = e$ $\beta_4 = f$ $\beta_4 = h$
$d \notin \mathrm{span}(\{a, x, y, z\})$?
$f \notin \mathrm{span}(\{b, x, y, z\})$?
$h \notin \mathrm{span}(\{b, x, y, z\})$?
true
false
true

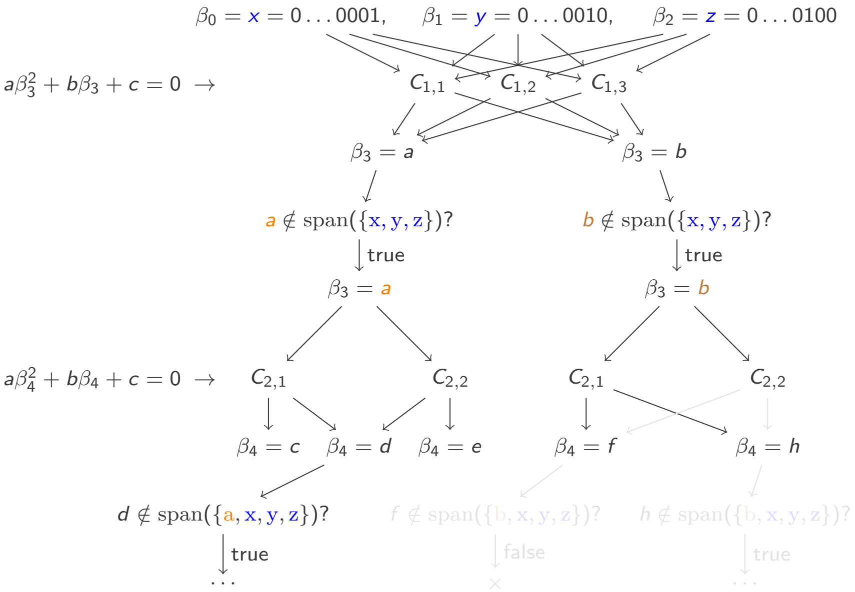

β0 = x = 0...0001, β1 = y = 0...0010, β2 = z = 0...0100
aβ3² + bβ3 + c = 0 →
C1,1 C1,2 C1,3
β3 = a β3 = b
a ∉ span({x, y, z})?
true
β3 = a
aβ4² + bβ4 + c = 0 →
C2,1 C2,2
β4 = c β4 = d β4 = e
d ∉ span({a, x, y, z})?
true
...
b ∉ span({x, y, z})?
true
β3 = b
C2,1 C2,2
β4 = f β4 = h
f ∉ span({b, x, y, z})?
false
h ∉ span({b, x, y, z})?
true
...

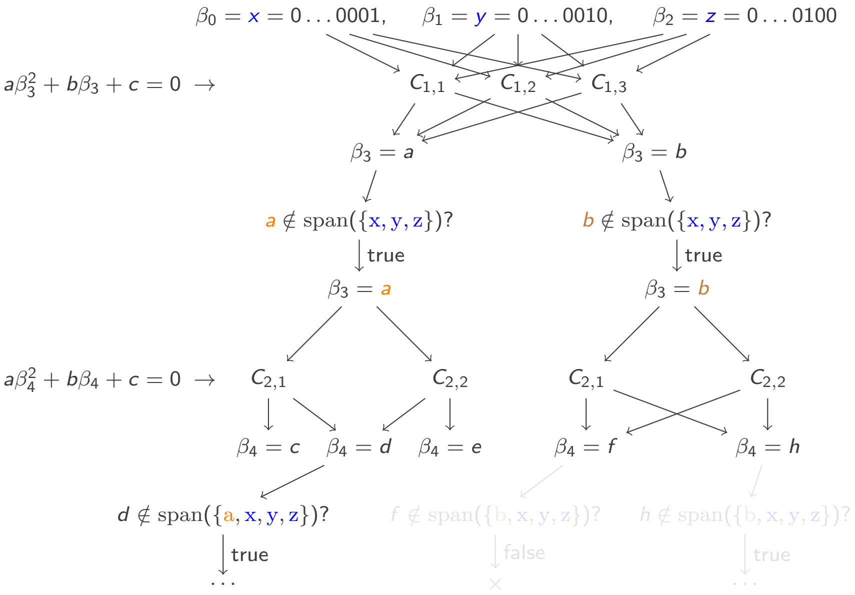


$\beta_0 = x = 0\ldots0001$, $\beta_1 = y = 0\ldots0010$, $\beta_2 = z = 0\ldots0100$
$a\beta_3^2 + b\beta_3 + c = 0 \rightarrow$
$C_{1,1}$ $C_{1,2}$ $C_{1,3}$
$\beta_3 = a$ $\beta_3 = b$
$a \notin \mathrm{span}(\{x, y, z\})$?
$b \notin \mathrm{span}(\{x, y, z\})$?
true
true
$\beta_3 = a$
$\beta_3 = b$
$a\beta_4^2 + b\beta_4 + c = 0 \rightarrow$
$C_{2,1}$ $C_{2,2}$ $C_{2,1}$ $C_{2,2}$
$\beta_4 = c$ $\beta_4 = d$ $\beta_4 = e$ $\beta_4 = f$ $\beta_4 = h$
$d \notin \mathrm{span}(\{a, x, y, z\})$?
$f \notin \mathrm{span}(\{b, x, y, z\})$?
$h \notin \mathrm{span}(\{b, x, y, z\})$?
true
false
true
$\ldots$
$\ldots$

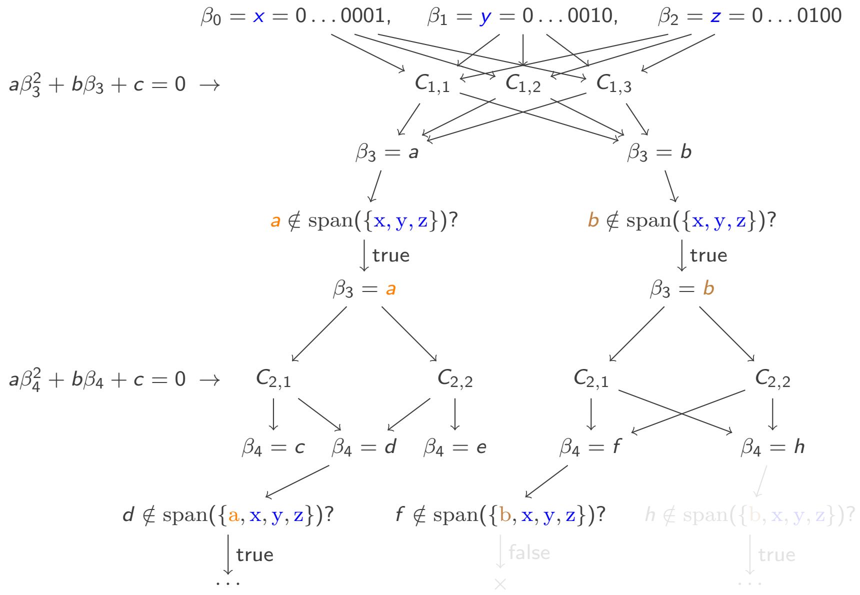

$\beta_0 = x = 0\ldots0001$, $\beta_1 = y = 0\ldots0010$, $\beta_2 = z = 0\ldots0100$
$a\beta_3^2 + b\beta_3 + c = 0 \rightarrow$
$C_{1,1}$ $C_{1,2}$ $C_{1,3}$
$\beta_3 = a$
$\beta_3 = b$
$a \notin \mathrm{span}(\{x, y, z\})$?
$b \notin \mathrm{span}(\{x, y, z\})$?
true
true
$\beta_3 = a$
$\beta_3 = b$
$a\beta_4^2 + b\beta_4 + c = 0 \rightarrow$
$C_{2,1}$ $C_{2,2}$
$C_{2,1}$ $C_{2,2}$
$\beta_4 = c$ $\beta_4 = d$ $\beta_4 = e$
$\beta_4 = f$ $\beta_4 = h$
$d \notin \mathrm{span}(\{a, x, y, z\})$?
$f \notin \mathrm{span}(\{b, x, y, z\})$?
$h \notin \mathrm{span}(\{b, x, y, z\})$?
true
false
true
...
...


$\beta_0 = x = 0\ldots0001$, $\beta_1 = y = 0\ldots0010$, $\beta_2 = z = 0\ldots0100$
$a\beta_3^2 + b\beta_3 + c = 0 \rightarrow$
$C_{1,1}$ $C_{1,2}$ $C_{1,3}$
$\beta_3 = a$ $\beta_3 = b$
$a \notin \mathrm{span}(\{x, y, z\})$?
$b \notin \mathrm{span}(\{x, y, z\})$?
true
true
$\beta_3 = a$ $\beta_3 = b$
$a\beta_4^2 + b\beta_4 + c = 0 \rightarrow$
$C_{2,1}$ $C_{2,2}$ $C_{2,1}$ $C_{2,2}$
$\beta_4 = c$ $\beta_4 = d$ $\beta_4 = e$ $\beta_4 = f$ $\beta_4 = h$
$d \notin \mathrm{span}(\{a, x, y, z\})$?
$f \notin \mathrm{span}(\{b, x, y, z\})$?
$h \notin \mathrm{span}(\{b, x, y, z\})$?
true
false
true
...
×
...

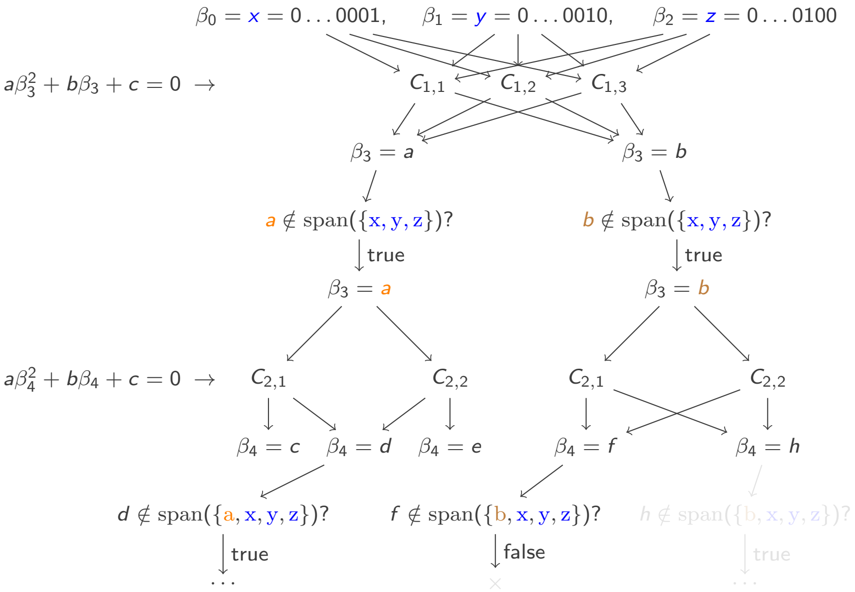

β0 = x = 0...0001, β1 = y = 0...0010, β2 = z = 0...0100
aβ3² + bβ3 + c = 0 →
C1,1 C1,2 C1,3
β3 = a
β3 = b
a ∉ span({x, y, z})?
b ∉ span({x, y, z})?
true
true
β3 = a
β3 = b
aβ4² + bβ4 + c = 0 →
C2,1 C2,2 C2,1 C2,2
β4 = c
β4 = d
β4 = e
β4 = f
β4 = h
d ∉ span({a, x, y, z})?
f ∉ span({b, x, y, z})?
h ∉ span({b, x, y, z})?
true
false
true
...
×
...

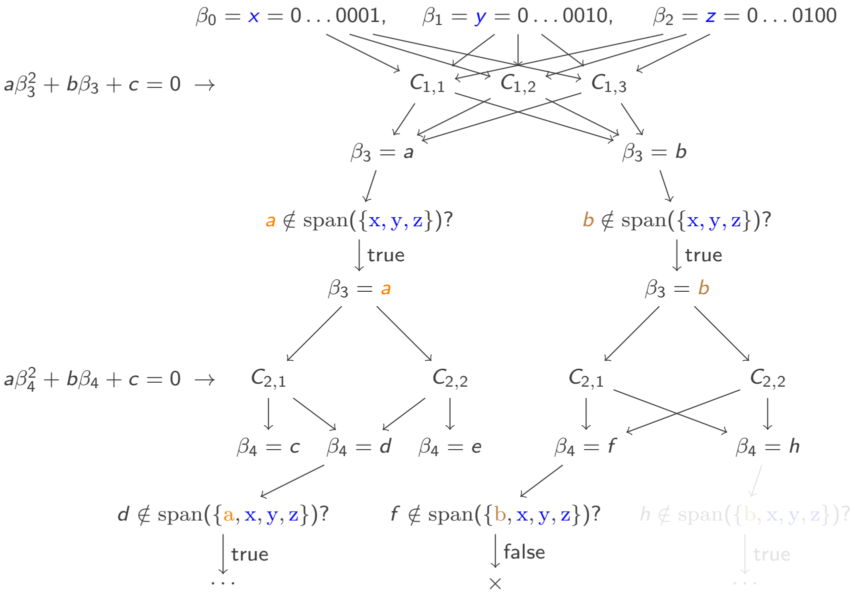


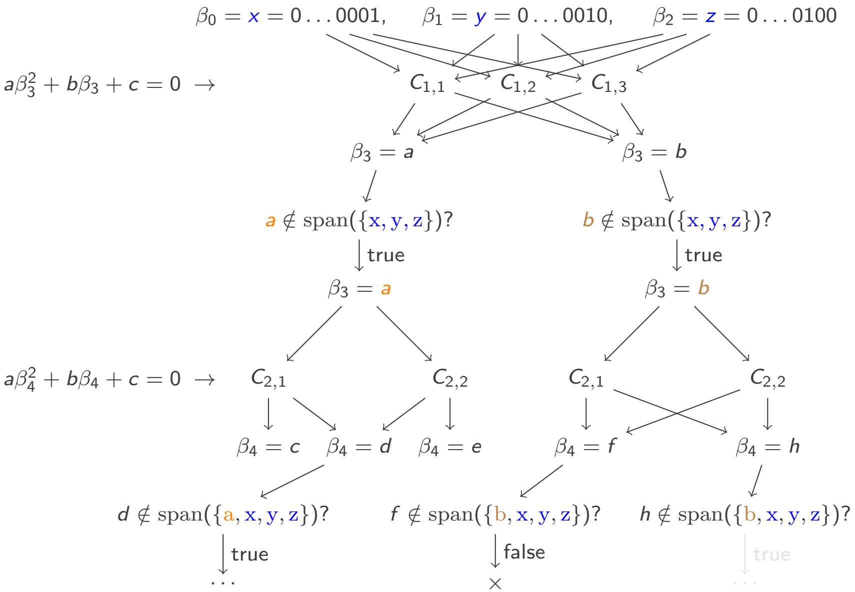
$\beta_0 = x = 0\ldots0001,\quad \beta_1 = y = 0\ldots0010,\quad \beta_2 = z = 0\ldots0100$
$a\beta_3^2 + b\beta_3 + c = 0 \rightarrow$
$C_{1,1}$ $C_{1,2}$ $C_{1,3}$
$\beta_3 = a$ $\beta_3 = b$
$a \notin \mathrm{span}(\{x, y, z\})?$
true
$b \notin \mathrm{span}(\{x, y, z\})?$
true
$\beta_3 = a$ $\beta_3 = b$
$a\beta_4^2 + b\beta_4 + c = 0 \rightarrow$
$C_{2,1}$ $C_{2,2}$ $C_{2,1}$ $C_{2,2}$
$\beta_4 = c$ $\beta_4 = d$ $\beta_4 = e$ $\beta_4 = f$ $\beta_4 = h$
$d \notin \mathrm{span}(\{a, x, y, z\})?$
true
...
$f \notin \mathrm{span}(\{b, x, y, z\})?$
false
$h \notin \mathrm{span}(\{b, x, y, z\})?$
true
...

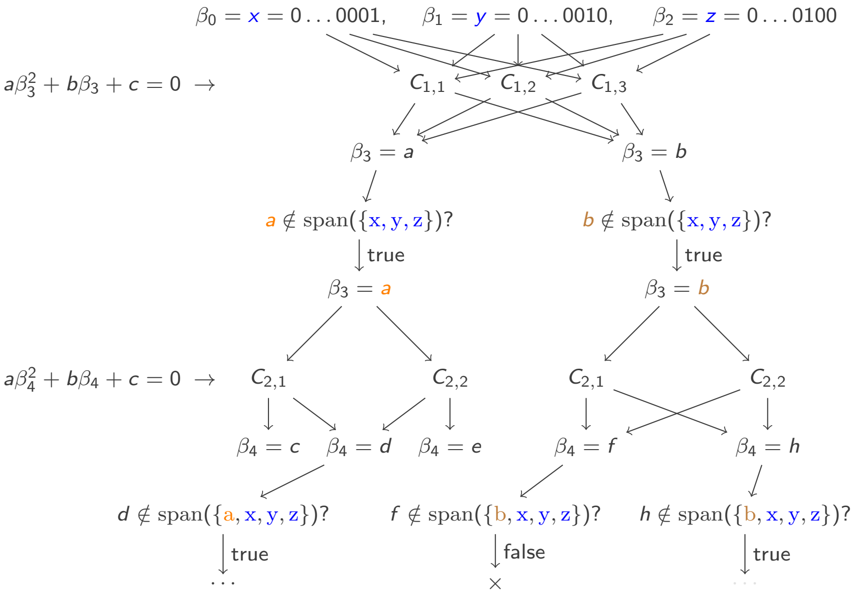


β0 = x = 0...0001,
β1 = y = 0...0010,
β2 = z = 0...0100
aβ3² + bβ3 + c = 0 →
C1,1
C1,2
C1,3
β3 = a
β3 = b
a ∉ span({x, y, z})?
b ∉ span({x, y, z})?
true
true
β3 = a
β3 = b
aβ4² + bβ4 + c = 0 →
C2,1
C2,2
C2,1
C2,2
β4 = c
β4 = d
β4 = e
β4 = f
β4 = h
d ∉ span({a, x, y, z})?
f ∉ span({b, x, y, z})?
h ∉ span({b, x, y, z})?
true
false
true
...
...

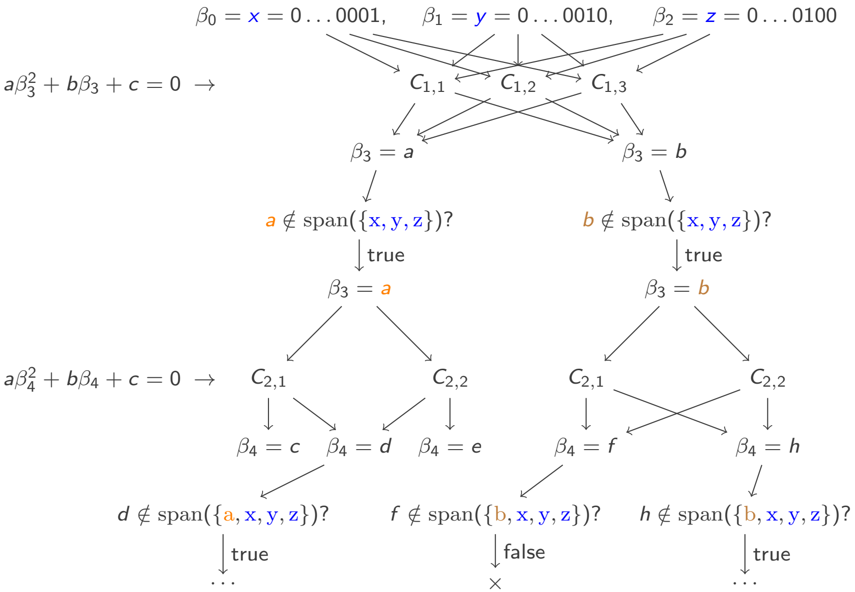

β0 = x = 0...0001, β1 = y = 0...0010, β2 = z = 0...0100
aβ3² + bβ3 + c = 0 →
C1,1 C1,2 C1,3
β3 = a
β3 = b
a ∉ span({x, y, z})?
b ∉ span({x, y, z})?
true
true
β3 = a
β3 = b
aβ4² + bβ4 + c = 0 →
C2,1 C2,2 C2,1 C2,2
β4 = c β4 = d β4 = e β4 = f β4 = h
d ∉ span({a, x, y, z})?
f ∉ span({b, x, y, z})?
h ∉ span({b, x, y, z})?
true
false
true
...
...


β0 = x = 0...0001, β1 = y = 0...0010, β2 = z = 0...0100
aβ3² + bβ3 + c = 0 →
C1,1 C1,2 C1,3
β3 = a
β3 = b
a ∉ span({x, y, z})?
b ∉ span({x, y, z})?
true
true
β3 = a
β3 = b
aβ4² + bβ4 + c = 0 →
C2,1 C2,2 C2,1 C2,2
β4 = c
β4 = d
β4 = e
β4 = f
β4 = h
d ∉ span({a, x, y, z})?
f ∉ span({b, x, y, z})?
h ∉ span({b, x, y, z})?
true
false
true
. . .
×
. . .


$\beta_0 = x = 0\ldots0001$, $\beta_1 = y = 0\ldots0010$, $\beta_2 = z = 0\ldots0100$
$a\beta_3^2 + b\beta_3 + c = 0 \rightarrow$
$C_{1,1}$ $C_{1,2}$ $C_{1,3}$
$\beta_3 = a$ $\beta_3 = b$
$a \notin \mathrm{span}(\{x, y, z\})$?
$b \notin \mathrm{span}(\{x, y, z\})$?
true
true
$\beta_3 = a$ $\beta_3 = b$
$a\beta_4^2 + b\beta_4 + c = 0 \rightarrow$
$C_{2,1}$ $C_{2,2}$ $C_{2,1}$ $C_{2,2}$
$\beta_4 = c$ $\beta_4 = d$ $\beta_4 = e$ $\beta_4 = f$ $\beta_4 = h$
$d \notin \mathrm{span}(\{a, x, y, z\})$?
$f \notin \mathrm{span}(\{b, x, y, z\})$?
$h \notin \mathrm{span}(\{b, x, y, z\})$?
true
false
true
$\ldots$
$\times$
$\ldots$



$\beta_0 = x = 0\dots0001$, $\beta_1 = y = 0\dots0010$, $\beta_2 = z = 0\dots0100$
$a\beta_3^2 + b\beta_3 + c = 0$ →
$C_{1,1}$ $C_{1,2}$ $C_{1,3}$
$\beta_3 = a$ $\beta_3 = b$
$a \notin \mathrm{span}(\{x, y, z\})$? $b \notin \mathrm{span}(\{x, y, z\})$?
true
true
$\beta_3 = a$ $\beta_3 = b$
$a\beta_4^2 + b\beta_4 + c = 0$ →
$C_{2,1}$ $C_{2,2}$ $C_{2,1}$ $C_{2,2}$
$\beta_4 = c$ $\beta_4 = d$ $\beta_4 = e$ $\beta_4 = f$ $\beta_4 = h$
$d \notin \mathrm{span}(\{a, x, y, z\})$? $f \notin \mathrm{span}(\{b, x, y, z\})$? $h \notin \mathrm{span}(\{b, x, y, z\})$?
true
false
true
$\dots$
×


$\beta_0 = x = 0\ldots0001$, $\beta_1 = y = 0\ldots0010$, $\beta_2 = z = 0\ldots0100$
$a\beta_3^2 + b\beta_3 + c = 0 \rightarrow$
$C_{1,1}$ $C_{1,2}$ $C_{1,3}$
$\beta_3 = a$ $\beta_3 = b$
$a \notin \mathrm{span}(\{x, y, z\})$?
$b \notin \mathrm{span}(\{x, y, z\})$?
true
true
$\beta_3 = a$ $\beta_3 = b$
$a\beta_4^2 + b\beta_4 + c = 0 \rightarrow$
$C_{2,1}$ $C_{2,2}$ $C_{2,1}$ $C_{2,2}$
$\beta_4 = c$ $\beta_4 = d$ $\beta_4 = e$ $\beta_4 = f$ $\beta_4 = h$
$d \notin \mathrm{span}(\{a, x, y, z\})$?
$f \notin \mathrm{span}(\{b, x, y, z\})$?
$h \notin \mathrm{span}(\{b, x, y, z\})$?
true
false
true
$\ldots$
$\times$
$\ldots$

	$m = 9,\ t = 33$	$m = 10,\ t = 40$
cycles gap $w = 1$	≈ 400	≈ 600
cycles gap $w = 4$	$\approx 13,000$	$\approx 19,000$
cycles gap $w = 6$	$\approx 17,000$	$\approx 23,000$
number of queries for $w = 1$	3,575,494	11,782,695
number of queries for $w = 4$	1,517,253	2,869,424
number of queries for $w = 6$	374,927	1,837,125
(worst case) number of final verifications	$\approx 8,000$	$\approx 2,000$
(worst case) running time for solving on 1 GHz x86 CPU	3h	28h

$w = 6$ equation counts were 1, 2, 4, 8, 16 , 16 . . .

Countermeasures

possibilities:

- try to achieve constant execution time for EEA
  - very difficult in software
- enforce constant running time for low weight ciphertexts through delay
  - doesn't cover power analysis
- add (pseudo) random error before decryption
  - change security level resp. code parameters – acceptance?
  - interaction with other countermeasures?

Countermeasures

possibilities:

- try to achieve constant execution time for EEA
  - very difficult in software
- enforce constant running time for low weight ciphertexts through delay
  - doesn't cover power analysis
- add (pseudo) random error before decryption
  - change security level resp. code parameters – acceptance?
  - interaction with other countermeasures?

possibilities:

- try to achieve constant execution time for EEA
  - very difficult in software
- enforce constant running time for low weight ciphertexts through delay
  - doesn't cover power analysis
- add (pseudo) random error before decryption
  - change security level resp. code parameters – acceptance?
  - interaction with other countermeasures?

Countermeasures

possibilities:

- try to achieve constant execution time for EEA
  - very difficult in software
- enforce constant running time for low weight ciphertexts through delay
  - doesn't cover power analysis
- add (pseudo) random error before decryption
  - change security level resp. code parameters – acceptance?
  - interaction with other countermeasures?

Countermeasures

possibilities:

- try to achieve constant execution time for EEA
    - very difficult in software
- enforce constant running time for low weight ciphertexts through delay
    - doesn't cover power analysis
- add (pseudo) random error before decryption
    - change security level resp. code parameters – acceptance?
    - interaction with other countermeasures?

Countermeasures

possibilities:

- try to achieve constant execution time for EEA
  - very difficult in software
- enforce constant running time for low weight ciphertexts through delay
  - doesn't cover power analysis
- add (pseudo) random error before decryption
  - change security level resp. code parameters – acceptance?
  - interaction with other countermeasures?

Countermeasures

possibilities:

- try to achieve constant execution time for EEA
  - very difficult in software
- enforce constant running time for low weight ciphertexts through delay
  - doesn't cover power analysis
- add (pseudo) random error before decryption
  - change security level resp. code parameters – acceptance?
  - interaction with other countermeasures?

Conclusion

- practical attack against secret permutation / support
- medium computational effort
- potential for remote attack (maybe without $w = 1$, i.e. α_z)
- first practical key-aimed timing attack against code-based cryptosystems
- related work of mine: for a *specific* choice of the root-finding algorithm practical key-aimed attacks also seem likely

Conclusion

- practical attack against secret permutation / support
- medium computational effort
- potential for remote attack (maybe without $w = 1$, i.e. α_z)
- first practical key-aimed timing attack against code-based cryptosystems
- related work of mine: for a *specific* choice of the root-finding algorithm practical key-aimed attacks also seem likely

- practical attack against secret permutation / support
- medium computational effort
- potential for remote attack (maybe without $w = 1$, i.e. α_z)
- first practical key-aimed timing attack against code-based cryptosystems
- related work of mine: for a *specific* choice of the root-finding algorithm practical key-aimed attacks also seem likely

- practical attack against secret permutation / support
- medium computational effort
- potential for remote attack (maybe without $w = 1$, i.e. α_z)
- first practical key-aimed timing attack against code-based cryptosystems
- related work of mine: for a *specific* choice of the root-finding algorithm practical key-aimed attacks also seem likely

Conclusion

- practical attack against secret permutation / support
- medium computational effort
- potential for remote attack (maybe without $w = 1$, i.e. α_z)
- first practical key-aimed timing attack against code-based cryptosystems
- related work of mine: for a *specific* choice of the root-finding algorithm practical key-aimed attacks also seem likely

Thank you!