High Performance Computing for Cryptography

M2P SCCI: Ensimag-Ujf

Abdourahmane SAKHO – Ali MKHIDA – Maad EL YADARI

Outline

- Canonical basis to normal basis
- Finding isomorphisms
- Speeding up computations on δ
- Non LUT Implementation
- Recovery of secret key
- Throughput of design at 200 MHz of the AES
- Practical Work

I. Canonical basis to normal basis

I. Canonical basis to normal basis

Sbox

$$S(t) = At^{-1} + B$$

t^{-1} : modulo inversion in $\mathbb{F}_2[X]/X^8 + X^4 + X^3 + X + 1$
Idea :
Find isomorphisms $\delta : \mathbb{F}_{256} \to \mathbb{F}_{16}[X]/X^2 + AX + B$ and compute the modular inversion in $\mathbb{F}_{16}[X]/X^2 + AX + B$

$$t \xrightarrow{\delta} a_1X + a_0 \xrightarrow{inversion} b_1X + b_0 \xrightarrow{\delta^{-1}} t^{-1}$$

I. Canonical basis to normal basis

Sbox

$$S(t) = At^{-1} + B$$

t^{-1} : modulo inversion in $\mathbb{F}_2[X]/X^8 + X^4 + X^3 + X + 1$
Idea :
Find isomorphisms $\delta : \mathbb{F}_{256} \to \mathbb{F}_{16}[X]/X^2 + AX + B$ and compute the modular inversion in $\mathbb{F}_{16}[X]/X^2 + AX + B$

$$t \xrightarrow{\delta} a_1X + a_0 \xrightarrow{inversion} b_1X + b_0 \xrightarrow{\delta^{-1}} t^{-1}$$

Computations in $\mathbb{F}_{16}$ under canonical basis

$$\mathbb{F}_{16} = \mathbb{F}_2[X]/X^4 + X^3 + X^2 + X + 1$$

with α a root of the generating polynomial.
Elements represented in the ordered basis : $\{\alpha^3, \alpha^2, \alpha, 1\}$

I. Canonical basis to normal basis

Normal Basis

Normal basis : $\{\alpha^8, \alpha^4, \alpha^2, \alpha\}$

$1_{\mathbb{F}_{16}} = \alpha^8 + \alpha^4 + \alpha^2 + \alpha.$

$\alpha^3 = \alpha^8, \alpha^5 = \alpha^{10} = \alpha^8 + \alpha^4 + \alpha^2 + \alpha, \alpha^6 = \alpha^{16} = \alpha, \alpha^9 = \alpha^4, \alpha^{12} = \alpha^2$

$x = [a_3, a_2, a_1, a_0]$ and $y = [b_3, b_2, b_1, b_0]$ in $\mathbb{F}_{16}$ where the a_is and b_is are the co-ordinates in the ordered basis.

I. Canonical basis to normal basis

Normal Basis

Normal basis : $\{\alpha^8, \alpha^4, \alpha^2, \alpha\}$

$1_{\mathbb{F}_{16}} = \alpha^8 + \alpha^4 + \alpha^2 + \alpha.$

$\alpha^3 = \alpha^8, \alpha^5 = \alpha^{10} = \alpha^8 + \alpha^4 + \alpha^2 + \alpha, \alpha^6 = \alpha^{16} = \alpha, \alpha^9 = \alpha^4, \alpha^{12} = \alpha^2$

$x = [a_3, a_2, a_1, a_0]$ and $y = [b_3, b_2, b_1, b_0]$ in $\mathbb{F}_{16}$ where the a_is and b_is are the co-ordinates in the ordered basis.

Multiplication

$$
\begin{aligned}
x \times y &= (a_3\alpha^8 + a_2\alpha^4 + a_1\alpha^2 + a_0\alpha) \times (b_3\alpha^8 + b_2\alpha^4 + b_1\alpha^2 + b_0\alpha) \\
&= (a_0b_1 \oplus a_1b_0 \oplus a_2b_2 \oplus \overbrace{a_0b_2 \oplus a_2b_0 \oplus a_1b_3 \oplus a_3b_1})\alpha^8 + \\
&\quad (a_0b_3 \oplus a_3b_0 \oplus a_1b_1 \oplus a_0b_2 \oplus a_2b_0 \oplus a_1b_3 \oplus a_3b_1)\alpha^4 + \\
&\quad (a_2b_3 \oplus a_3b_2 \oplus a_0b_0 \oplus a_0b_2 \oplus a_2b_0 \oplus a_1b_3 \oplus a_3b_1)\alpha^2 + \\
&\quad (a_1b_2 \oplus a_2b_1 \oplus a_3b_3 \oplus a_0b_2 \oplus a_2b_0 \oplus a_1b_3 \oplus a_3b_1)\alpha
\end{aligned}
$$

I. Canonical basis to normal basis

Computations continued

Square

Taking $x = y$ in the above expression gives :

$$a_2\alpha^8 + a_1\alpha^4 + a_0\alpha^2 + a_3\alpha$$

Remarks

A lot of terms in common in the co-ordinates of the ordered basis.
Squaring is permutation of co-ordinates : Comes for free in hardware.
Better suited than canonical basis.

II. Finding isomorphisms

II. Finding isomorphisms

Isomorphism δ

- Look for generators α of $\mathbb{G}_m(\mathbb{F}_2[X]/X^8 + X^4 + X^3 + X + 1)$ and β of $\mathbb{G}_m(\mathbb{F}_{16}[X]/X^2 + AX + B)$.
- α and β roots of the same irreducible polynomial $P(X)$.
- Set $\delta(\alpha) = \beta$, use $\delta(\alpha^i) = \delta(\alpha)^i$.

II. Finding isomorphisms

Isomorphism δ

- Look for generators α of $\mathbb{G}_m(\mathbb{F}_2[X]/X^8+X^4+X^3+X+1)$ and β of $\mathbb{G}_m(\mathbb{F}_{16}[X]/X^2+AX+B)$.
- α and β roots of the same irreducible polynomial $P(X)$.
- Set $\delta(\alpha)=\beta$, use $\delta(\alpha^i)=\delta(\alpha)^i$.

Finding β using δ of the lecture

- $2^8-1=255=3\times 5\times 17$, any non-identity element whose order is co-prime to $\{3,5,17\}$ is a generator.
- $\delta(X^2)=[1,0,0,1]X+[0,0,0,0]$. order($\delta(X)$)$=255 \implies$ order$(\delta(X^2))=255$.
- $\beta=[1,0,0,1]X+[0,0,0,0] \implies P(\beta)=[0,0,0,0]X+[0,0,0,0]$ where $P(X)=X^8+X^4+X^3+X+1$.
- $\delta_2(X)=[1,0,0,1]X+[0,0,0,0]$ and use $\delta_2(X^i)=\delta_2(X)^i$.

II. Finding isomorphisms

δ_2 derived from δ

$$\delta_2 = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \qquad \delta_2^{-1} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \end{pmatrix}$$

III. Speeding up computations on δ

III. Speeding up computations on δ

$\delta(v), \delta^{-1}(v)$

Computations are matrix multiplication, can be performed in parallel.

III. Speeding up computations on δ

$\delta(v), \delta^{-1}(v)$

Computations are matrix multiplication, can be performed in parallel.

$\delta(v), \delta^{-1}(v)$

Computations are matrix multiplication, can be performed in parallel. One can look for δ with low hamming weight.

IV. Non LUT Implementation

IV. Non LUT Implementation

Non-LUT proposition

Further use of composite field arithmetic

Idea :

Look to cut the longest path to reduce maximum computational delay i.e. find δ_a and δ_b st $\delta(\cdot) = \delta_b(\delta_a(\cdot))$

IV. Non LUT Implementation

Non-LUT proposition

Further use of composite field arithmetic

Idea :

Look to cut the longest path to reduce maximum computational delay i.e. find δ_a and δ_b st $\delta(\cdot) = \delta_b(\delta_a(\cdot))$

- Find $\delta_a : \mathbb{F}_{16} \to \mathbb{F}_4[X]/X^2 + CX + D$.
- Inversion done in $\mathbb{F}_4[X]/X^2 + CX + D$ where computations done in $\mathbb{F}_4$.
- $\gamma \in \mathbb{F}_4 \implies \gamma^3 = 1 \implies \gamma^{-1} = \gamma^2$.

IV. Non LUT Implementation

Non-LUT proposition

Further use of composite field arithmetic

Idea :

Look to cut the longest path to reduce maximum computational delay i.e. find δ_a and δ_b st $\delta(\cdot) = \delta_b(\delta_a(\cdot))$

- Find $\delta_a : \mathbb{F}_{16} \to \mathbb{F}_4[X]/X^2 + CX + D$.
- Inversion done in $\mathbb{F}_4[X]/X^2 + CX + D$ where computations done in $\mathbb{F}_4$.
- $\gamma \in \mathbb{F}_4 \implies \gamma^3 = 1 \implies \gamma^{-1} = \gamma^2$.
- Inversion comes for free in hardware so no LUT needed.

$$t \xrightarrow{\delta_b} a_1X+a_0 \xrightarrow{\delta_a} \{b_1X+b_0\} \xrightarrow{inverse} \{c_1X+c_0\} \xrightarrow{\delta_a^{-1}} (a_1X+a_0)^{-1} \xrightarrow{\delta_b^{-1}} t^{-1}$$

V. Recovery of secret key

- **PROOF ON BOARD**

VI. Throughput of design at 200 MHz of the AES

- **PROOF ON BOARD**

VII. Practical work

Exhaustive search | Hellman's TMTO | Distinguished Points TMTO

Input :
-Plaintext .
-Cipher text.

Output :
-The encryption key.

Encrypt the plaintext with all the possible keys and compare with cipher text.

For a key of 24 bits of entropy we got: 78 seconds in average => 5,525 for a key with 32 bits of entropy.

Exhaustive search | **Hellman's TMTO** | Distinguished Points TMTO

- Offline computation: the plaintext => the list of (start point, end point).
- Online computation : the plaintext, the cipher text and the list => the Key ?

Some tests for a key with 16 bits of entropy : m = 10 000 and t= 10:

Execution time : offline	Execution time : online	Found the key	False alarm
1.929seconde	0.05seconde	key found	0
1.896seconde	0.092seconde	Key found	3
1.9seconde	0.004seconde	key found	0

Exhaustive search | **Hellman's TMTO** | Distinguished Points TMTO

For an 32 bits of entropy for the key : m = 10 and t= 100 000:

Execution time : offline	Execution time : online	Found the key	False alarm
14.956seconde	122.934seconde	Not found	220
22.652seconde	247.143seconde	Not found	90

For an 32 bits of entropy for the key : m = 100 000 and t= 10:

Execution time : offline	Execution time : online	Found the key	False alarm
5.364seconde	34.917seconde	Not found	0
3.656seconde	12.11seconde	Not found	0

Exhaustive search | Hellman's TMTO | **Distinguished Points TMTO**

m = 10 and t= 100 000:

Execution time offline	Execution time online	Found the key	False alarm
11.234seconds	2.035seconds	Not found	5
8.078seconds	1.96seconds	Not found	3

m = 100 000 and t= 100 :

Execution time offline	Execution time online	Found the key	False alarm
103.662seconds	0.002seconds	Not found	0
112.268seconds	0.001seconds	Not found	0

Thanks for your attention ...