Whitebox attack resistant cryptography

Bc. Dušan Klinec

Faculty of Informatics
Masaryk University
Brno

24. 06. 2013

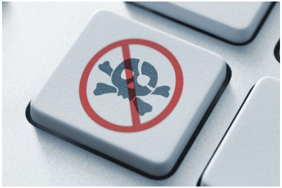
Outline

1. Whitebox cryptography
   - Motivation
   - Whitebox context
2. Implementation
3. Improvements
   - Observations from attacks
   - Solutions
     - Key schedule
     - S-Boxes
     - Diffusion layer
4. Questions and Discussion
   - Oponent's question
     - BGE attack introduction
     - Isomorphism
     - Recovering non-linear parts of IO bijections
   - Questions

Outline

Motivation

- Cryptographic algorithm runs on an untrusted device
- Study resistance to analyzing from cryptographic perspective
    - Digital Rights Management
    - client software running in the cloud (NSA is listening)
    - cryptographic operation performed on smart-card



Credit: http://www.wired.co.uk/

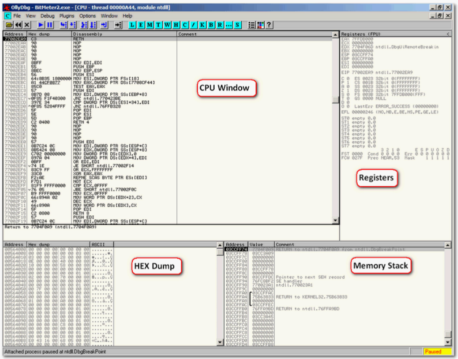
Credit: http://www.businessinsider.com/

Credit: PV079 L08 smartcards

Whitebox context

Strong model of an adversary

- traces steps of algorithm
- sees/modifies memory content
- can modify the binary code of an algorithm
- observe an internal state

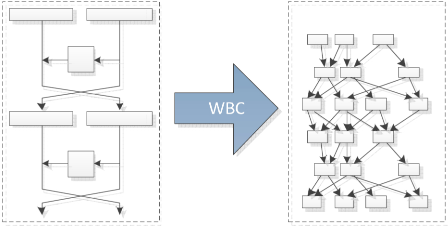


Whitebox transformation

Whitebox transformations

Cipher algorithm is transformed by whitebox transformations to a form, that is more difficult to attack in the whitebox context.

- Similar to *obfuscation*, but has different perspective/goals (cryptographic ones, resist key extraction, inverting)
- Often uses transformation of the algorithm to a network of look-up tables, which are further composed and obfuscated

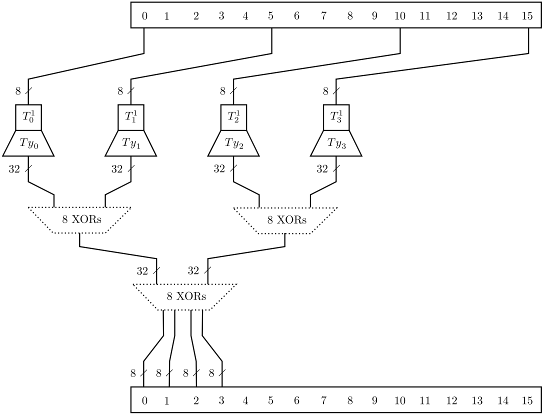


Credit: B. Wyseur

AES transformations

We are interested mainly in AES

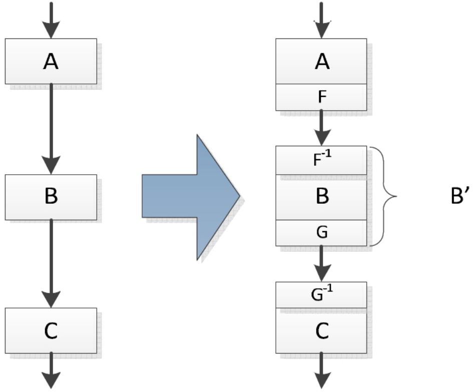
- Table implementation in Rijndael paper, use look-up tables
- *ShiftRows*, *AddRoundKey*, *SubBytes*, *MixColumn*
- secret symmetric key is embedded into look-up tables



Source: `http://eprint.iacr.org/2013/104.pdf`

Whitebox transformations - IO bijections

- tables themselves are vulnerable to an algebraic analysis (extraction of an embedded symmetric key)
- Solution: random input/output bijections



Source: `http://whiteboxcrypto.com/files/2012˙misc.pdf`

Whitebox AES

First implementation by Chow *et al.* in 2002, *White-Box Cryptography and an AES Implementation*.

- uses look-up tables
- uses mentioned protections to resist attacks
- encryption algorithm table size: 752 kB

Broken by Billet *et al.* in 2005, algebraic attack (so called BGE attack)

- recovers non-linear part of IO bijections up to unknown affine part
- further analysis, attacking one round
- using public knowledge of key-invariant building blocks (S-box, MixColumn), extracts round keys
- key schedule is invertible $\rightarrow$ embedded encryption key recovered

Whitebox AES

First implementation by Chow *et al.* in 2002, *White-Box Cryptography and an AES Implementation*.

- uses look-up tables
- uses mentioned protections to resist attacks
- encryption algorithm table size: 752 kB

Broken by Billet *et al.* in 2005, algebraic attack (so called BGE attack)

- recovers non-linear part of IO bijections up to unknown affine part
- further analysis, attacking one round
- using public knowledge of key-invariant building blocks (S-box, MixColumn), extracts round keys
- key schedule is invertible → embedded encryption key recovered

Whitebox dual AES

AES whitebox scheme appeared using *dual* ciphers, by Karroumi in 2011.

Dual AES

- Ciphers E, E' are dual if they are isomorphic.
- $\exists f, g, h\, \forall P, K : f(E_K)(P) = E'_{g(K)}(h(P))$, where f, g, h are bijections, P is plaintext, K is encryption key.
- Thus $E_K(P) = f^{-1}(E'_{g(k)}(h(P))$

Whitebox dual AES scheme

- original paper uses dual AES ciphers, a linear transformation Δ is used to change one dual AES to another.
- in each round/column uses different dual AES
- no published cryptanalysis yet
- claimed resistance to BGE attack 2^{91} computational steps
- we proved it is not the case

Whitebox dual AES

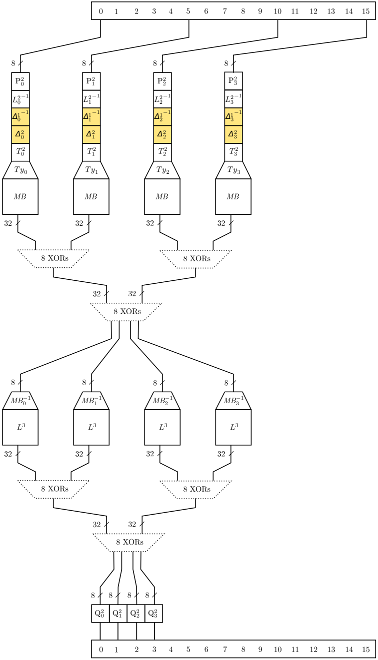
AES whitebox scheme appeared using *dual* ciphers, by Karroumi in 2011.

Dual AES

- Ciphers E, E' are dual if they are isomorphic.
- $\exists f, g, h \, \forall P, K : f\left(E_K\right)(P) = E'_{g(K)}\left(h\left(P\right)\right)$, where f, g, h are bijections, P is plaintext, K is encryption key.
- Thus $E_K(P) = f^{-1}(E'_{g(k)}(h(P))$

Whitebox dual AES scheme

- original paper uses dual AES ciphers, a linear transformation Δ is used to change one dual AES to another.
- in each round/column uses different dual AES
- no published cryptanalysis yet
- claimed resistance to BGE attack 2^{91} computational steps
- we proved it is not the case


0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
P_0^2 P_1^2 P_2^2 P_3^2
$L_0^{2^{-1}}$ $L_1^{2^{-1}}$ $L_2^{2^{-1}}$ $L_3^{2^{-1}}$
$\Delta_0^{1^{-1}}$ $\Delta_1^{1^{-1}}$ $\Delta_2^{1^{-1}}$ $\Delta_3^{1^{-1}}$
Δ_0^2 Δ_1^2 Δ_2^2 Δ_3^2
T_0^2 T_1^2 T_2^2 T_3^2
Ty_0 Ty_1 Ty_2 Ty_3
MB
8 XORs
MB_0^{-1} MB_1^{-1} MB_2^{-1} MB_3^{-1}
L^3
Q_0^2 Q_1^2 Q_2^2 Q_3^2
8
32

Outline

Implementation - whitebox AES scheme

```
// Inverse ShiftRows()
// | 00 04 08 12 |                                | 00 04 08 12 |
// | 01 05 09 13 | --- Shift Rows Inv --->  | 13 01 05 09 |
// | 02 06 10 14 |  (cyclic left right)     | 10 14 02 06 |
// | 03 07 11 15 |                                | 07 11 15 03 |
//
const static int shiftRowsInv[N_BYTES];


// XOR tables
W32XTB eXTab[N_ROUNDS][N_SECTIONS][N_XOR_GROUPS];

// XOR tables for external encodings (input & output, connected to Type I tables)
W32XTB eXTabEx[2][15][4];       // 2 (input, output) * 15 (8,4,2,1) * 4 (32bit * 4 = 128bit)

// Type I - just first round
AES_TB_TYPE1 eTab1[2][N_BYTES];

// Type II tables
AES_TB_TYPE2 eTab2[N_ROUNDS][N_BYTES];

// Type III tables
AES_TB_TYPE3 eTab3[N_ROUNDS][N_BYTES];

// universal encryption/decryption method
void encdec(W128b& state, bool encrypt);

// pure table implementation of encryption of given state
void encrypt(W128b& state);

// pure table implementation of decryption of given state
void decrypt(W128b& state);
```

Implementation - whitebox dual AES scheme

```
// One i iteration corresponds to one column above. One i=0 iteration should look like this
// Every A in next diagram is A_I = A^1_{r+1, I} - simplified syntax
//
// | A_0 (02 T(x)) |   | A_0 (03 T(x)) |   | A_0 (01 T(x)) |   | A_0 (01 T(x)) |
// | A_3 (01 T(x)) | + | A_3 (02 T(x)) | + | A_3 (03 T(x)) | + | A_3 (01 T(x)) |
// | A_2 (02 T(x)) |   | A_2 (01 T(x)) |   | A_2 (02 T(x)) |   | A_2 (03 T(x)) |
// | A_1 (03 T(x)) |   | A_1 (01 T(x)) |   | A_1 (01 T(x)) |   | A_1 (02 T(x)) |
//
int tmpi;
for(tmpi=0; tmpi<4; tmpi++){
    if (encrypt){
        this->AESCipher[ 4* r     + i                  ].applyTinv(mcres[tmpi]);
        this->AESCipher[(4*(r+1)) + POS_MOD(i-tmpi, 4)].applyT(   mcres[tmpi]);
        applyLookupTable(genA1[(4*(r+1)) + POS_MOD(i-tmpi, 4)],    mcres[tmpi]);
        this->AESCipher[(4*(r+1)) + POS_MOD(i-tmpi, 4)].applyTinv(mcres[tmpi]);
        this->AESCipher[ 4* r     + i                  ].applyT(   mcres[tmpi]);
    } else {
        this->AESCipher[ 4* r     + i                  ].applyTinv(mcres[tmpi]);
        this->AESCipher[(4*(r+1)) + POS_MOD(i+tmpi, 4)].applyT(   mcres[tmpi]);
        applyLookupTable(genA2[(4*(r+1)) + POS_MOD(i+tmpi, 4)],    mcres[tmpi]);

        //
        // Compensate affine part of A2 relation
        //
        // A2 is not linear in decryption case, but affine.
        // We have here 4 elements (entering XOR), so from 3 of them
        // we have to subtract affine constant = A2[0].
        // Af(a1+a2+a3+a4) = A*a1 + A*a2 + A*a3 + A*a4 + c
        //                 = Af(a1) + Af(a2)+Af(0) + Af(a3)+Af(0) + Af(a4)+Af(0)
        if (j!=0) {
            mcres[tmpi][0] += genA2[(4*(r+1)) + POS_MOD(i+tmpi, 4)][0];
        }

        this->AESCipher[(4*(r+1)) + POS_MOD(i+tmpi, 4)].applyTinv(mcres[tmpi]);
        this->AESCipher[ 4* r     + i                  ].applyT(   mcres[tmpi]);
    }
```

Implementation - BGE attack

```
cout << "Starting attack phase 1 ..." << endl;
for(r=0; r<9; r++){
    // Init f_00 function in Sr
    for(i=0; i<AES_BYTES; i++){
        Sr[r].S[i%4][i/4].f_00.c1 = 0;
    }

    //
    // Compute f(x,0,0,0) function for each Q_{i,j}
    //
    // x x x x        y_{0,0} y_{1,0} ..
    // 0 0 0 0   R    y_{0,1} y_{1,1} ..
    // 0 0 0 0  --->  y_{0,2} y_{1,2} ..
    // 0 0 0 0        y_{0,3} y_{1,3} ..
    //
    cout << "Generating f_00 for round r="<<r<<endl;
    for(x=0; x<=0xff; x++){
        memset(&state, 0, sizeof(state));       // put 0 everywhere
        state.B[0]=x;   state.B[1]=x;           // init with x values for y_0 in each column
        state.B[2]=x;   state.B[3]=x;           // recall that state array is indexed by rows.

        this->Rbox(state, true, r, true);       // perform R box computation on input & output values
        for(i=0; i<AES_BYTES; i++){
            fction_t & f00 = Sr[r].S[i%4][i/4].f_00;
            f00.f[x] = state.B[i];
            f00.finv[state.B[i]] = x;
        }
    }


    // f(x,0,0,0) finalization - compute hash of f00 function
    for(i=0; i<AES_BYTES; i++){
        Sr[r].S[i%4][i/4].f_00.initHash();
    }
```

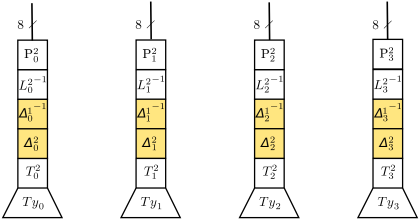
Implementation - BGE attack

We then discovered, that BGE attack works also on the whitebox dual AES scheme! (It should not)

```
   recoverQj; q = 0x88; gamma=0x01;
   recoverQj self-test; r=5; col=3; (y0, y3); P[0].deltaInv=0x03; alfa_{3,0}=0x03
   recoverQj self-test; r=5; col=3; (y0, y3); P[1].deltaInv=0x01; alfa_{3,1}=0x01
   recoverQj self-test; r=5; col=3; (y0, y3); P[2].deltaInv=0x01; alfa_{3,2}=0x01
   recoverQj self-test; r=5; col=3; (y0, y3); P[3].deltaInv=0x02; alfa_{3,3}=0x02
   recoverQj; q = 0x3c; gamma=0x01;

Going to reconstruct encryption key from extracted round keys...
* Round keys extracted from the process, r=3
 0x3d  0x47  0x1e  0x6d  0x80  0x16  0x23  0x7a  0x47  0xfe  0x7e  0x88  0x7d  0x3e  0x44  0x3b

* Round keys extracted from the process, r=4
 0xef  0xa8  0xb6  0xdb  0x44  0x52  0x71  0x0b  0xa5  0x5b  0x25  0xad  0x41  0x7f  0x3b  0x00

* Round keys extracted from the process, r=5
 0xd4  0x7c  0xca  0x11  0xd1  0x83  0xf2  0xf9  0xc6  0x9d  0xb8  0x15  0xf8  0x87  0xbc  0xbc

Recovering cipher key from round keys...
We have correct Rcon! rconIdx=3
RC=2; previousKey:
 0xf2  0x7a  0x59  0x73
 0xc2  0x96  0x35  0x59
 0x95  0xb9  0x80  0xf6
 0xf2  0x43  0x7a  0x7f

RC=1; previousKey:
 0xa0  0x88  0x23  0x2a
 0xfa  0x54  0xa3  0x6c
 0xfe  0x2c  0x39  0x76
 0x17  0xb1  0x39  0x05

RC=0; previousKey:
 0x2b  0x28  0xab  0x09
 0x7e  0xae  0xf7  0xcf
 0x15  0xd2  0x15  0x4f
 0x16  0xa6  0x88  0x3c

Final result:
 0x2b  0x7e  0x15  0x16  0x28  0xae  0xd2  0xa6  0xab  0xf7  0x15  0x88  0x09  0xcf  0x4f  0x3c

Benchmark finished! Total time = 3a s; on average = 58 s; clocktime=57.66 s;
```

What is wrong with dual AES scheme?

- BGE attack considers dual AES implementation as a normal ones
  - same irreducible polynomial defining field
  - same generator of the field
- so why it works?
  - linear mapping Δ transforming one dual AES to another dual AES
  - linear mapping Δ can be merged with non-linear random bijections
  - removed in the attack, has no effect whatsoever
  - proof in master thesis



What is wrong with dual AES scheme?

$$Q_{i,j}^{r\ \prime}\left(\bigoplus_{l=0}^{3}\Delta(\alpha_{l,j})\cdot\left(\Delta\times A\times\Delta^{-1}\left(\left(\Delta\circ P_{i,l}^{r\ \prime\prime}\left(x_{i,l}\right)\oplus\Delta\left(k_{i,l}\right)\right)^{-1^{\Delta}\ \mathrm{GF}\left(2^{8}\right)}\right)\oplus\Delta\left(c\right)\right)\right)$$

$$Q_{i,j}^{r\ \prime}\circ\Delta\left(\bigoplus_{l=0}^{3}\alpha_{l,j}\cdot\left(A\times\Delta^{-1}\left(\left(\Delta\circ P_{i,l}^{r\ \prime\prime}\left(x_{i,l}\right)\oplus\Delta\left(k_{i,l}\right)\right)^{-1^{\Delta}\ \mathrm{GF}\left(2^{8}\right)}\right)\oplus c\right)\right)$$

$$Q_{i,j}^{r\ \prime}\circ\Delta\left(\bigoplus_{l=0}^{3}\alpha_{l,j}\cdot\left(A\times\Delta^{-1}\left(\left(\Delta\left(P_{i,l}^{r\ \prime\prime}\left(x_{i,l}\right)\oplus k_{i,l}\right)\right)^{-1^{\Delta}\ \mathrm{GF}\left(2^{8}\right)}\right)\oplus c\right)\right)$$

$$Q_{i,j}^{r\ \prime}\circ\Delta\left(\bigoplus_{l=0}^{3}\alpha_{l,j}\cdot\left(A\times\Delta^{-1}\left(\Delta\left(P_{i,l}^{r\ \prime\prime}\left(x_{i,l}\right)\oplus k_{i,l}\right)^{-1\ \mathrm{GF}\left(2^{8}\right)}\right)\oplus c\right)\right)$$

$$Q_{i,j}^{r\ \prime}\circ\Delta\left(\bigoplus_{l=0}^{3}\alpha_{l,j}\cdot\left(A\times\left(\left(P_{i,l}^{r\ \prime\prime}\left(x_{i,l}\right)\oplus k_{i,l}\right)^{-1\ \mathrm{GF}\left(2^{8}\right)}\right)\oplus c\right)\right)$$

$$Q_{i,j}^{r\ \prime}\circ\Delta\left(\bigoplus_{l=0}^{3}\alpha_{l,j}\cdot\left(A\times\left(\left(P_{i,l}^{r\ \prime\prime}\left(x_{i,l}\right)\oplus k_{i,l}\right)^{-1\ \mathrm{GF}\left(2^{8}\right)}\right)\oplus c\right)\right)$$

$$Q_{i,j}^{r\ \prime}\circ\Delta\circ R_{i,j}^{\prime}\left(x_{i,0},x_{i,1},x_{i,2},x_{i,3}\right)$$

Outline

Main observations from attacks

- mostly algebraic attacks
- key schedule reversibility is a weakness
- attacks use a public knowledge of static building blocks (S-boxes, MixColumns)

Kerckhoffs's principle

According to Kerckhoffs's principle, cipher security should be based on the secrecy of a secret key not the design of the cipher.

Solutions

- AES is not suitable for whitebox context (no secure whitebox implementation exists)
- design a new cipher for whitebox context
- transform key-invariant building blocks to key-dependent
  - preserve same security level
  - add randomness
  - neglect hardware implementation issues

Suggested modifications

- non-invertible key schedule
- key-dependent S-boxes
- stronger diffusion layer (larger, key-dependent)

Key schedule

Issues:

- 2 consecutive round keys $\rightarrow$ we obtain all round keys
- attack does not need to attack on each round

Solution:

- use hash function to derive round keys
- use expensive hash function (e.g., KPDF2)
- whitebox context $\rightarrow$ each round key has to be considered as a separate, strong encryption key
- $$k_i^r = \begin{cases} hash_{N_{bc},N_{sha}}(key, salt)_i & \text{if } r = 0 \\ hash_{N_{bc},N_{sha}}(k^{r-1} \mid\mid key, salt)_i & \text{otherwise} \end{cases} \tag{2}$$

S-boxes

- Key-invariant S-Box is used in BGE attack
- Use concept of key-dependent S-boxes (Blowfish, Twofish)

Twofish S-boxes

$$s_{0,k_0,k_1}(x) = q_1\left[q_0\left[q_0\left[x\right] \oplus k_0\right] \oplus k_1\right]$$
$$s_{1,k_2,k_3}(x) = q_0\left[q_0\left[q_1\left[x\right] \oplus k_2\right] \oplus k_3\right]$$
$$s_{2,k_4,k_5}(x) = q_1\left[q_1\left[q_0\left[x\right] \oplus k_4\right] \oplus k_5\right]$$
$$s_{3,k_6,k_7}(x) = q_0\left[q_1\left[q_1\left[x\right] \oplus k_6\right] \oplus k_7\right]$$

Where q_0, q_1 are fixed permutations.

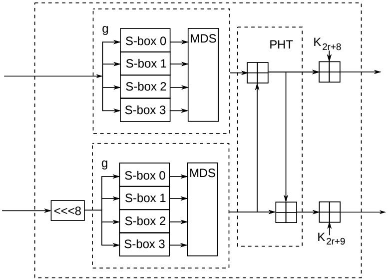


Figure: Twofish core

Diffusion layer

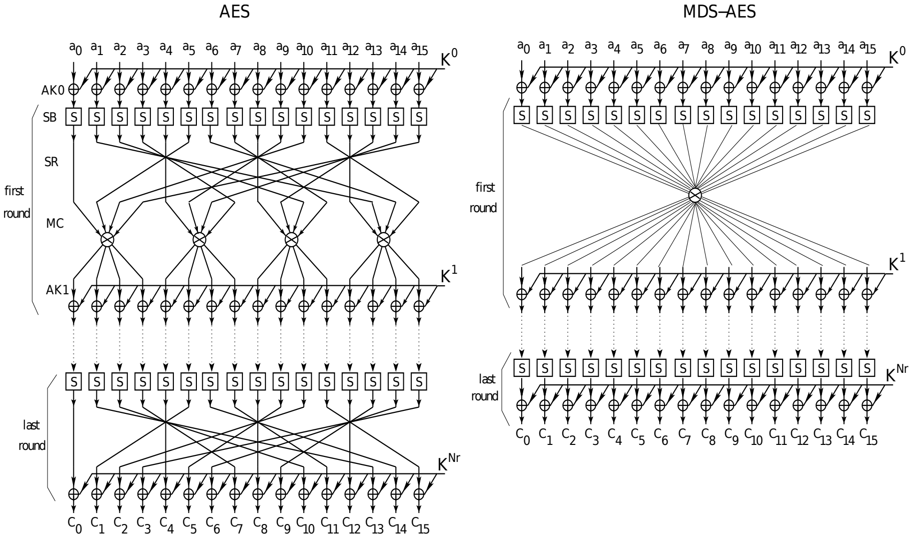
Issues:

- 32 bit diffusion is too small (invertion attack)
- $8 \rightarrow 32$ table type II
- limitation - HW implementation concerns
- low dependency of output on input within one round
  - one output byte depends on 4 input bytes (16 in total)
  - *ShiftRows* effect is negligible in whitebox context

Solutions:

- neglect HW implementation performance
- increase diffusion layer on whole round
- more possible diffusion layers with same level of security (MDS codes), key dependency

Diffusion layer



Credit: Jr., Jorge Nakahara and Abrahao, Elcio, *A New Involutory MDS Matrix for the AES*

Security gain

- BGE attack is not possible anymore
  - naive mounting would require to try all possible key-dependent combinations of building blocks (diffusion, S-Boxes)
  - no key extraction due to key schedule modification
- Inverting the cipher is much more difficult
  - function is now too wide
  - each output byte depends on each input byte within one round
  - stronger diffusion
- our opinion: key-dependency and randomization are important concepts

Drawbacks:

- new cipher $\rightarrow$ need to analyze (statistical) blackbox properties
- no backward compatibility
- lower throughput, computationally more intensive
- increased implementation (tables) size

Outline

Oponent's question

Question

Explain the following equation: $\widetilde{Q}(\psi(g)) = g(0),\ g \in \mathcal{S}$

- It is core of the first part of the BGE attack (recovering non-linear part of IO bijections)
- Need to explain whole first part of the attack.

BGE attack

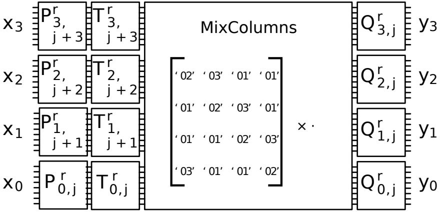
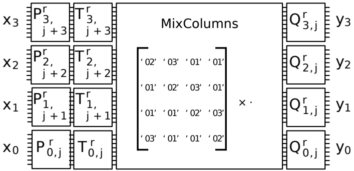


Figure: WB AES round from BGE attack perspective, one of R^r_j, $j = 0, \dots, 3$

$$
\begin{aligned}
y_0\left(x_0, c_1, c_2, c_3\right) &= Q^r_{0,j}\left(\alpha T^r_{0,j}\left(P^r_{0,j}\left(x_0\right)\right) \oplus \beta_{c_1,c_2,c_3}\right) \\
&= Q^r_{0,j} \circ \oplus_{\beta_{c_1,c_2,c_3}} \circ \alpha \cdot T^r_{0,j} \circ P^r_{0,j}\left(x_0\right)
\end{aligned} \tag{4}
$$

Fix $c_2 = c_3 = 0$ (WLOG).



$$y_0\left(x_0, c_1\right) = Q^r_{0,j} \circ \oplus_{\beta_{c_1}} \circ \alpha \cdot T^r_{0,j} \circ P^r_{0,j}\left(x_0\right)$$

$$\begin{aligned}
& y_0\left(x_0, b\right) \circ y_0^{-1}\left(x_0, a\right) = \\
& = \left(Q_{0,j} \circ \oplus_{\beta_b} \circ \alpha \cdot T_{0,j} \circ P_{0,j}\right) \circ \left(P_{0,j}^{-1} \circ \left(\alpha \cdot T_{0,j}\right)^{-1} \circ \oplus_{\beta_a} \circ Q_{0,j}^{-1}\right) \\
& = Q_{0,j} \circ \oplus_{\beta_b} \circ \oplus_{\beta_a} \circ Q_{0,j}^{-1} \\
& = Q_{0,j} \circ \oplus_{(\beta_b \oplus \beta_a)} \circ Q_{0,j}^{-1} \\
& = Q_{0,j} \circ \oplus_{\beta_c} \circ Q_{0,j}^{-1}
\end{aligned}$$

Isomorphism

Isomorphism

$$\varphi \ : \ \begin{array}{ccc} (\mathcal{S}, \circ) & \longrightarrow & (\mathrm{GF}(2)^8, \oplus) \\ Q \circ \oplus_\beta \circ Q^{-1} & \longmapsto & [\beta] \end{array}$$

- $f_1, f_2 \in \mathcal{S}$, then also $f_2 \circ f_1 \in \mathcal{S}$
- $f_2 \circ f_1 = (Q \circ \oplus_{\beta_2} \circ Q^{-1}) \circ (Q \circ \oplus_{\beta_1} \circ Q^{-1}) = Q \circ \oplus_{(\beta_1 \oplus \beta_2)} \circ Q^{-1}$
- $\varphi(f_1 \circ f_2) = \varphi(f_1) \oplus \varphi(f_2)$

φ is isomorphism, but we don't know it.

- we have set $\mathcal{S}$ of functions f of the form $Q \circ \oplus_\beta \circ Q^{-1}$
- we don't know β for some $f \in \mathcal{S}$

Isomorphism

Isomorphism

$$\varphi : \begin{array}{ccc} (\mathcal{S}, \circ) & \longrightarrow & (\mathrm{GF}(2)^8, \oplus) \\ Q \circ \oplus_\beta \circ Q^{-1} & \longmapsto & [\beta] \end{array}$$

- $f_1, f_2 \in \mathcal{S}$, then also $f_2 \circ f_1 \in \mathcal{S}$
- $f_2 \circ f_1 = (Q \circ \oplus_{\beta_2} \circ Q^{-1}) \circ (Q \circ \oplus_{\beta_1} \circ Q^{-1}) = Q \circ \oplus_{(\beta_1 \oplus \beta_2)} \circ Q^{-1}$
- $\varphi(f_1 \circ f_2) = \varphi(f_1) \oplus \varphi(f_2)$

φ is isomorphism, but we don't know it.

- we have set $\mathcal{S}$ of functions f of the form $Q \circ \oplus_\beta \circ Q^{-1}$
- we don't know β for some $f \in \mathcal{S}$

Isomorphism

Isomorphism

$$\varphi \ : \ \begin{array}{ccc} (\mathcal{S}, \circ) & \longrightarrow & \left(\mathrm{GF}\left(2\right)^{8}, \oplus\right) \\ Q \circ \oplus_{\beta} \circ Q^{-1} & \longmapsto & [\beta] \end{array}$$

- $f_1, f_2 \in \mathcal{S}$, then also $f_2 \circ f_1 \in \mathcal{S}$
- $f_2 \circ f_1 = (Q \circ \oplus_{\beta_2} \circ Q^{-1}) \circ (Q \circ \oplus_{\beta_1} \circ Q^{-1}) = Q \circ \oplus_{(\beta_1 \oplus \beta_2)} \circ Q^{-1}$
- $\varphi(f_1 \circ f_2) = \varphi(f_1) \oplus \varphi(f_2)$

φ is isomorphism, but we don't know it.

- we have set $\mathcal{S}$ of functions f of the form $Q \circ \oplus_{\beta} \circ Q^{-1}$
- we don't know β for some $f \in \mathcal{S}$

Isomorphism

Isomorphism

$$\varphi \;:\; \begin{array}{ccc} (\mathcal{S}, \circ) & \longrightarrow & (\mathrm{GF}(2)^8, \oplus) \\ Q \circ \oplus_\beta \circ Q^{-1} & \longmapsto & [\beta] \end{array}$$

- $f_1, f_2 \in \mathcal{S}$, then also $f_2 \circ f_1 \in \mathcal{S}$
- $f_2 \circ f_1 = (Q \circ \oplus_{\beta_2} \circ Q^{-1}) \circ (Q \circ \oplus_{\beta_1} \circ Q^{-1}) = Q \circ \oplus_{(\beta_1 \oplus \beta_2)} \circ Q^{-1}$
- $\varphi(f_1 \circ f_2) = \varphi(f_1) \oplus \varphi(f_2)$

φ is isomorphism, but we don't know it.

- we have set $\mathcal{S}$ of functions f of the form $Q \circ \oplus_\beta \circ Q^{-1}$
- we don't know β for some $f \in \mathcal{S}$

Isomorphism

$$\varphi \ : \ \begin{array}{ccc} (\mathcal{S}, \circ) & \longrightarrow & (\mathrm{GF}\,(2)^8, \oplus) \\ Q \circ \oplus_\beta \circ Q^{-1} & \longmapsto & [\beta] \end{array}$$

Imagine we know β:

- Select a tuple $(f_1, \dots, f_8)$, $f_i \in \mathcal{S}$, s.t. $(\varphi(f_1), \dots, \varphi(f_8)) = ([e_i])_{i=1,\dots,8}$ is a standard base of $\mathrm{GF}\,(2)^8$
- It holds $f_i = Q \circ \oplus_{2^{i-1}} \circ Q^{-1}$, so that $([\varphi(f_i)])_{i=0,\dots,8} = ([e_i])_{i=0,\dots,8}$
- Then $(f_1, \dots, f_8)$ is a base of $(\mathcal{S}, \circ)$, so it holds: $\forall f \in \mathcal{S}, \exists!(\varepsilon_1, \dots, \varepsilon_8) \in \{0,1\}^8,\ f = f_8^{\varepsilon_8} \circ f_7^{\varepsilon_7} \circ \dots \circ f_1^{\varepsilon_1}$
- Then $\varphi(f) = \varphi(f_8^{\varepsilon_8}) \oplus \varphi(f_7^{\varepsilon_7}) \oplus \dots \oplus \varphi(f_1^{\varepsilon_1})$
- Then $\varphi(f) = [e_8]^{\varepsilon_8} \oplus [e_7]^{\varepsilon_7} \oplus \dots \oplus [e_1]^{\varepsilon_1}$
- But we don't know φ

Isomorphism

$$\varphi \;:\; \begin{array}{ccc} (\mathcal{S}, \circ) & \longrightarrow & (\mathrm{GF}(2)^8, \oplus) \\ Q \circ \oplus_\beta \circ Q^{-1} & \longmapsto & [\beta] \end{array}$$

Imagine we know β:

- Select a tuple $(f_1, \dots, f_8)$, $f_i \in \mathcal{S}$, s.t. $(\varphi(f_1), \dots, \varphi(f_8)) = ([e_i])_{i=1,\dots,8}$ is a standard base of $\mathrm{GF}(2)^8$
- It holds $f_i = Q \circ \oplus_{2^{i-1}} \circ Q^{-1}$, so that $([\varphi(f_i)])_{i=0,\dots,8} = ([e_i])_{i=0,\dots,8}$
- Then $(f_1, \dots, f_8)$ is a base of $(\mathcal{S}, \circ)$, so it holds: $\forall f \in \mathcal{S}, \exists!(\varepsilon_1, \dots, \varepsilon_8) \in \{0,1\}^8,\ f = f_8^{\varepsilon_8} \circ f_7^{\varepsilon_7} \circ \dots \circ f_1^{\varepsilon_1}$
- Then $\varphi(f) = \varphi(f_8^{\varepsilon_8}) \oplus \varphi(f_7^{\varepsilon_7}) \oplus \dots \oplus \varphi(f_1^{\varepsilon_1})$
- Then $\varphi(f) = [e_8]^{\varepsilon_8} \oplus [e_7]^{\varepsilon_7} \oplus \dots \oplus [e_1]^{\varepsilon_1}$
- But we don't know φ

Isomorphism

$$\varphi \; : \; \begin{array}{ccc} (\mathcal{S}, \circ) & \longrightarrow & (\mathrm{GF}(2)^8, \oplus) \\ Q \circ \oplus_\beta \circ Q^{-1} & \longmapsto & [\beta] \end{array}$$

Imagine we know β:

- Select a tuple $(f_1, \ldots, f_8)$, $f_i \in \mathcal{S}$, s.t. $(\varphi(f_1), \ldots, \varphi(f_8)) = ([e_i])_{i=1,\ldots,8}$ is a standard base of $\mathrm{GF}(2)^8$
- It holds $f_i = Q \circ \oplus_{2^{i-1}} \circ Q^{-1}$, so that $([\varphi(f_i)])_{i=0,\ldots,8} = ([e_i])_{i=0,\ldots,8}$
- Then $(f_1, \ldots, f_8)$ is a base of $(\mathcal{S}, \circ)$, so it holds: $\forall f \in \mathcal{S}, \exists!(\varepsilon_1, \ldots, \varepsilon_8) \in \{0,1\}^8$, $f = f_8^{\varepsilon_8} \circ f_7^{\varepsilon_7} \circ \cdots \circ f_1^{\varepsilon_1}$
- Then $\varphi(f) = \varphi(f_8^{\varepsilon_8}) \oplus \varphi(f_7^{\varepsilon_7}) \oplus \cdots \oplus \varphi(f_1^{\varepsilon_1})$
- Then $\varphi(f) = [e_8]^{\varepsilon_8} \oplus [e_7]^{\varepsilon_7} \oplus \cdots \oplus [e_1]^{\varepsilon_1}$
- But we don't know φ

Isomorphism

$$\varphi \;:\; \begin{array}{ccc} (\mathcal{S}, \circ) & \longrightarrow & (\mathrm{GF}(2)^8, \oplus) \\ Q \circ \oplus_\beta \circ Q^{-1} & \longmapsto & [\beta] \end{array}$$

Imagine we know β:

- Select a tuple $(f_1, \ldots, f_8)$, $f_i \in \mathcal{S}$, s.t. $(\varphi(f_1), \ldots, \varphi(f_8)) = ([e_i])_{i=1,\ldots,8}$ is a standard base of $\mathrm{GF}(2)^8$
- It holds $f_i = Q \circ \oplus_{2^{i-1}} \circ Q^{-1}$, so that $([\varphi(f_i)])_{i=0,\ldots,8} = ([e_i])_{i=0,\ldots,8}$
- Then $(f_1, \ldots, f_8)$ is a base of $(\mathcal{S}, \circ)$, so it holds: $\forall f \in \mathcal{S}, \exists!(\varepsilon_1, \ldots, \varepsilon_8) \in \{0,1\}^8,\ f = f_8^{\varepsilon_8} \circ f_7^{\varepsilon_7} \circ \cdots \circ f_1^{\varepsilon_1}$
- Then $\varphi(f) = \varphi(f_8^{\varepsilon_8}) \oplus \varphi(f_7^{\varepsilon_7}) \oplus \cdots \oplus \varphi(f_1^{\varepsilon_1})$
- Then $\varphi(f) = [e_8]^{\varepsilon_8} \oplus [e_7]^{\varepsilon_7} \oplus \cdots \oplus [e_1]^{\varepsilon_1}$
- But we don't know φ

Isomorphism

$$\varphi \;:\; \begin{array}{ccc} (\mathcal{S}, \circ) & \longrightarrow & (\mathrm{GF}(2)^8, \oplus) \\ Q \circ \oplus_\beta \circ Q^{-1} & \longmapsto & [\beta] \end{array}$$

Imagine we know β:

- Select a tuple $(f_1, \ldots, f_8)$, $f_i \in \mathcal{S}$, s.t. $(\varphi(f_1), \ldots, \varphi(f_8)) = ([e_i])_{i=1,\ldots,8}$ is a standard base of $\mathrm{GF}(2)^8$
- It holds $f_i = Q \circ \oplus_{2^{i-1}} \circ Q^{-1}$, so that $([\varphi(f_i)])_{i=0,\ldots,8} = ([e_i])_{i=0,\ldots,8}$
- Then $(f_1, \ldots, f_8)$ is a base of $(\mathcal{S}, \circ)$, so it holds: $\forall f \in \mathcal{S}, \exists!(\varepsilon_1, \ldots, \varepsilon_8) \in \{0,1\}^8,\ f = f_8^{\varepsilon_8} \circ f_7^{\varepsilon_7} \circ \cdots \circ f_1^{\varepsilon_1}$
- Then $\varphi(f) = \varphi(f_8^{\varepsilon_8}) \oplus \varphi(f_7^{\varepsilon_7}) \oplus \cdots \oplus \varphi(f_1^{\varepsilon_1})$
- Then $\varphi(f) = [e_8]^{\varepsilon_8} \oplus [e_7]^{\varepsilon_7} \oplus \cdots \oplus [e_1]^{\varepsilon_1}$
- But we don't know φ

Isomorphism

$$\varphi \; : \; \begin{array}{ccc} (\mathcal{S}, \circ) & \longrightarrow & (\mathrm{GF}(2)^8, \oplus) \\ Q \circ \oplus_\beta \circ Q^{-1} & \longmapsto & [\beta] \end{array}$$

Imagine we know β:

- Select a tuple $(f_1, \ldots, f_8)$, $f_i \in \mathcal{S}$, s.t. $(\varphi(f_1), \ldots, \varphi(f_8)) = ([e_i])_{i=1,\ldots,8}$ is a standard base of $\mathrm{GF}(2)^8$
- It holds $f_i = Q \circ \oplus_{2^{i-1}} \circ Q^{-1}$, so that $([\varphi(f_i)])_{i=0,\ldots,8} = ([e_i])_{i=0,\ldots,8}$
- Then $(f_1, \ldots, f_8)$ is a base of $(\mathcal{S}, \circ)$, so it holds: $\forall f \in \mathcal{S}, \exists!(\varepsilon_1, \ldots, \varepsilon_8) \in \{0,1\}^8,\; f = f_8^{\varepsilon_8} \circ f_7^{\varepsilon_7} \circ \cdots \circ f_1^{\varepsilon_1}$
- Then $\varphi(f) = \varphi(f_8^{\varepsilon_8}) \oplus \varphi(f_7^{\varepsilon_7}) \oplus \cdots \oplus \varphi(f_1^{\varepsilon_1})$
- Then $\varphi(f) = [e_8]^{\varepsilon_8} \oplus [e_7]^{\varepsilon_7} \oplus \cdots \oplus [e_1]^{\varepsilon_1}$
- But we don't know φ

Isomorphism

$$\varphi \;:\; \begin{array}{ccc} (\mathcal{S}, \circ) & \longrightarrow & (\mathrm{GF}(2)^8, \oplus) \\ Q \circ \oplus_\beta \circ Q^{-1} & \longmapsto & [\beta] \end{array}$$

Imagine we know β:

- Select a tuple $(f_1, \ldots, f_8)$, $f_i \in \mathcal{S}$, s.t. $(\varphi(f_1), \ldots, \varphi(f_8)) = ([e_i])_{i=1,\ldots,8}$ is a standard base of $\mathrm{GF}(2)^8$
- It holds $f_i = Q \circ \oplus_{2^{i-1}} \circ Q^{-1}$, so that $([\varphi(f_i)])_{i=0,\ldots,8} = ([e_i])_{i=0,\ldots,8}$
- Then $(f_1, \ldots, f_8)$ is a base of $(\mathcal{S}, \circ)$, so it holds: $\forall f \in \mathcal{S}, \exists!(\varepsilon_1, \ldots, \varepsilon_8) \in \{0,1\}^8,\ f = f_8^{\varepsilon_8} \circ f_7^{\varepsilon_7} \circ \cdots \circ f_1^{\varepsilon_1}$
- Then $\varphi(f) = \varphi(f_8^{\varepsilon_8}) \oplus \varphi(f_7^{\varepsilon_7}) \oplus \cdots \oplus \varphi(f_1^{\varepsilon_1})$
- Then $\varphi(f) = [e_8]^{\varepsilon_8} \oplus [e_7]^{\varepsilon_7} \oplus \cdots \oplus [e_1]^{\varepsilon_1}$
- But we don't know φ

Isomorphism

$$\varphi \ : \ \begin{array}{ccc} (\mathcal{S}, \circ) & \longrightarrow & (\mathrm{GF}(2)^8, \oplus) \\ Q \circ \oplus_\beta \circ Q^{-1} & \longmapsto & [\beta] \end{array}$$

- Select an arbitrary tuple $(f_i)_{i=1,\dots,8}$ that form a base of $\mathcal{S}$
- $(f_i)_{i=1,\dots,8}$ is arbitrary, thus $([\varphi(f_i)])_{i=0,\dots,8} \neq ([e_i])_{i=0,\dots,8}$ in general
- $\varphi(f_i) = [\beta_i]$ in general
- thus $\varphi(f) = \bigoplus_{i=1}^{8}[\beta_i]$, $([\beta_i])_{i=0,\dots,8}$ is some base of $\mathrm{GF}(2)^8$
- Instead we define $\psi(f_i) = [e_i]$, for some base $(f_i)_{i=1,\dots,8}$
  - $(f_i)_{i=1,\dots,8}$ is standard base of $\mathcal{S}$
  - $(\psi(f_i))_{i=1,\dots,8} = ([e_i])_{i=1,\dots,8}$ is base of $\mathrm{GF}(2)^8$
- Then exists (unknown) base change matrix L, s.t. $L[e_i] = [\beta_i]$
- So we can define $\psi = L^{-1} \circ \varphi$

Isomorphism

$$\varphi : \begin{array}{ccc} (\mathcal{S}, \circ) & \longrightarrow & (\mathrm{GF}(2)^8, \oplus) \\ Q \circ \oplus_\beta \circ Q^{-1} & \longmapsto & [\beta] \end{array}$$

- Select an arbitrary tuple $(f_i)_{i=1,\ldots,8}$ that form a base of $\mathcal{S}$
- $(f_i)_{i=1,\ldots,8}$ is arbitrary, thus $([\varphi(f_i)])_{i=0,\ldots,8} \neq ([e_i])_{i=0,\ldots,8}$ in general
- $\varphi(f_i) = [\beta_i]$ in general
- thus $\varphi(f) = \bigoplus_{i=1}^{8}[\beta_i]$, $([\beta_i])_{i=0,\ldots,8}$ is some base of $\mathrm{GF}(2)^8$
- Instead we define $\psi(f_i) = [e_i]$, for some base $(f_i)_{i=1,\ldots,8}$
  - $(f_i)_{i=1,\ldots,8}$ is standard base of $\mathcal{S}$
  - $(\psi(f_i))_{i=1,\ldots,8} = ([e_i])_{i=1,\ldots,8}$ is base of $\mathrm{GF}(2)^8$
- Then exists (unknown) base change matrix L, s.t. $L[e_i] = [\beta_i]$
- So we can define $\psi = L^{-1} \circ \varphi$

Isomorphism

$$\varphi \; : \; \begin{array}{ccc} (\mathcal{S}, \circ) & \longrightarrow & (\mathrm{GF}(2)^8, \oplus) \\ Q \circ \bigoplus_\beta \circ Q^{-1} & \longmapsto & [\beta] \end{array}$$

- Select an arbitrary tuple $(f_i)_{i=1,\dots,8}$ that form a base of $\mathcal{S}$
- $(f_i)_{i=1,\dots,8}$ is arbitrary, thus $([\varphi(f_i)])_{i=0,\dots,8} \neq ([e_i])_{i=0,\dots,8}$ in general
- $\varphi(f_i) = [\beta_i]$ in general
- thus $\varphi(f) = \bigoplus_{i=1}^{8}[\beta_i]$, $([\beta_i])_{i=0,\dots,8}$ is some base of $\mathrm{GF}(2)^8$
- Instead we define $\psi(f_i) = [e_i]$, for some base $(f_i)_{i=1,\dots,8}$
  - $(f_i)_{i=1,\dots,8}$ is standard base of $\mathcal{S}$
  - $(\psi(f_i))_{i=1,\dots,8} = ([e_i])_{i=1,\dots,8}$ is base of $\mathrm{GF}(2)^8$
- Then exists (unknown) base change matrix L, s.t. $L[e_i] = [\beta_i]$
- So we can define $\psi = L^{-1} \circ \varphi$

Isomorphism

$$\varphi \; : \; \begin{array}{ccc} (\mathcal{S}, \circ) & \longrightarrow & (\mathrm{GF}(2)^8, \oplus) \\ Q \circ \oplus_\beta \circ Q^{-1} & \longmapsto & [\beta] \end{array}$$

- Select an arbitrary tuple $(f_i)_{i=1,\dots,8}$ that form a base of $\mathcal{S}$
- $(f_i)_{i=1,\dots,8}$ is arbitrary, thus $([\varphi(f_i)])_{i=0,\dots,8} \neq ([e_i])_{i=0,\dots,8}$ in general
- $\varphi(f_i) = [\beta_i]$ in general
- thus $\varphi(f) = \bigoplus_{i=1}^{8} [\beta_i]$, $([\beta_i])_{i=0,\dots,8}$ is some base of $\mathrm{GF}(2)^8$
- Instead we define $\psi(f_i) = [e_i]$, for some base $(f_i)_{i=1,\dots,8}$
    - $(f_i)_{i=1,\dots,8}$ is standard base of $\mathcal{S}$
    - $(\psi(f_i))_{i=1,\dots,8} = ([e_i])_{i=1,\dots,8}$ is base of $\mathrm{GF}(2)^8$
- Then exists (unknown) base change matrix L, s.t. $L[e_i] = [\beta_i]$
- So we can define $\psi = L^{-1} \circ \varphi$

Isomorphism

$$\varphi : \begin{array}{ccc} (\mathcal{S}, \circ) & \longrightarrow & (\mathrm{GF}(2)^8, \oplus) \\ Q \circ \oplus_\beta \circ Q^{-1} & \longmapsto & [\beta] \end{array}$$

- Select an arbitrary tuple $(f_i)_{i=1,\dots,8}$ that form a base of $\mathcal{S}$
- $(f_i)_{i=1,\dots,8}$ is arbitrary, thus $([\varphi(f_i)])_{i=0,\dots,8} \neq ([e_i])_{i=0,\dots,8}$ in general
- $\varphi(f_i) = [\beta_i]$ in general
- thus $\varphi(f) = \bigoplus_{i=1}^{8}[\beta_i]$, $([\beta_i])_{i=0,\dots,8}$ is some base of $\mathrm{GF}(2)^8$
- Instead we define $\psi(f_i) = [e_i]$, for some base $(f_i)_{i=1,\dots,8}$
  - $(f_i)_{i=1,\dots,8}$ is standard base of $\mathcal{S}$
  - $(\psi(f_i))_{i=1,\dots,8} = ([e_i])_{i=1,\dots,8}$ is base of $\mathrm{GF}(2)^8$
- Then exists (unknown) base change matrix L, s.t. $L[e_i] = [\beta_i]$
- So we can define $\psi = L^{-1} \circ \varphi$

Isomorphism

$$\varphi \ : \ \begin{array}{ccc} (\mathcal{S}, \circ) & \longrightarrow & (\mathrm{GF}(2)^8, \oplus) \\ Q \circ \oplus_\beta \circ Q^{-1} & \longmapsto & [\beta] \end{array}$$

- Select an arbitrary tuple $(f_i)_{i=1,\dots,8}$ that form a base of $\mathcal{S}$
- $(f_i)_{i=1,\dots,8}$ is arbitrary, thus $([\varphi(f_i)])_{i=0,\dots,8} \neq ([e_i])_{i=0,\dots,8}$ in general
- $\varphi(f_i) = [\beta_i]$ in general
- thus $\varphi(f) = \bigoplus_{i=1}^{8}[\beta_i]$, $([\beta_i])_{i=0,\dots,8}$ is some base of $\mathrm{GF}(2)^8$
- Instead we define $\psi(f_i) = [e_i]$, for some base $(f_i)_{i=1,\dots,8}$
  - $(f_i)_{i=1,\dots,8}$ is standard base of $\mathcal{S}$
  - $(\psi(f_i))_{i=1,\dots,8} = ([e_i])_{i=1,\dots,8}$ is base of $\mathrm{GF}(2)^8$
- Then exists (unknown) base change matrix L, s.t. $L[e_i] = [\beta_i]$
- So we can define $\psi = L^{-1} \circ \varphi$

Commutative diagram

Isomorphism

$$\varphi \; : \; \begin{array}{ccc} (\mathcal{S}, \circ) & \longrightarrow & (\mathrm{GF}\,(2)^8, \oplus) \\ Q \circ \oplus_\beta \circ Q^{-1} & \longmapsto & [\beta] \end{array}$$

Let's have $f \in \mathcal{S}$. Recall $\psi = L^{-1} \circ \varphi$

$$\begin{tikzcd} \mathrm{GF}(2^8) \arrow[d, "f"'] \\ \mathrm{GF}(2^8) \end{tikzcd}$$

Commutative diagram

Isomorphism

$$\varphi \;:\; \begin{array}{ccc} (\mathcal{S}, \circ) & \longrightarrow & (\mathrm{GF}(2)^8, \oplus) \\ Q \circ \oplus_\beta \circ Q^{-1} & \longmapsto & [\beta] \end{array}$$

Let's have $f \in \mathcal{S}$. Recall $\psi = L^{-1} \circ \varphi$

$$\begin{tikzcd} \mathrm{GF}(2^8) \arrow[r, "Q^{-1}"] \arrow[d, "f"'] & {} \arrow[d, "\oplus_{\varphi(f)}"'] \\ \mathrm{GF}(2^8) & {} \arrow[l, "Q"'] \end{tikzcd}$$

$f = Q \circ \oplus_{\varphi(f)} \circ Q^{-1}$

Commutative diagram

Isomorphism

$$\varphi \;:\; \begin{array}{ccc} (\mathcal{S}, \circ) & \longrightarrow & (\mathrm{GF}\,(2)^8, \oplus) \\ Q \circ \oplus_\beta \circ Q^{-1} & \longmapsto & [\beta] \end{array}$$

Let's have $f \in \mathcal{S}$. Recall $\psi = L^{-1} \circ \varphi$

$$\begin{tikzcd}
\mathrm{GF}(2^8) \arrow[r, "Q^{-1}"] \arrow[d, "f"'] & {} \arrow[r, "L^{-1}"] \arrow[d, "\oplus_{\varphi(f)}"'] & {} \arrow[d, "\oplus_{\psi(f)}"'] \\
\mathrm{GF}(2^8) & {} \arrow[l, "Q"'] & {} \arrow[l, "L"']
\end{tikzcd}$$

$$f = Q \circ \oplus_{\varphi(f)} \circ Q^{-1} = Q \circ L \circ \oplus_{\psi(f)} \circ L^{-1} \circ Q^{-1}$$

Commutative diagram

Isomorphism

$$\varphi : \begin{array}{ccc} (\mathcal{S}, \circ) & \longrightarrow & (\mathsf{GF}(2)^8, \oplus) \\ Q \circ \oplus_\beta \circ Q^{-1} & \longmapsto & [\beta] \end{array}$$

Let's have $f \in \mathcal{S}$. Recall $\psi = L^{-1} \circ \varphi$

$$\begin{tikzcd}
\mathrm{GF}(2^8) \arrow[r, "Q^{-1}"] \arrow[d, "f"'] & \cdot \arrow[r, "L^{-1}"] \arrow[d, "\oplus_{\varphi(f)}"] & \cdot \arrow[r, "\oplus_{(Q \circ L)^{-1}(['00'])}"] \arrow[d, "\oplus_{\psi(f)}"] & \cdot \arrow[d, "\oplus_{\psi(f)}"] \\
\mathrm{GF}(2^8) & \cdot \arrow[l, "Q"'] & \cdot \arrow[l, "L"'] & \cdot \arrow[l, "\oplus_{(Q \circ L)^{-1}(['00'])}"']
\end{tikzcd}$$

$$f = Q \circ \oplus_{\varphi(f)} \circ Q^{-1} = Q \circ L \circ \oplus_{\psi(f)} \circ L^{-1} \circ Q^{-1}$$

Commutative diagram

Isomorphism

$$\varphi \; : \; \begin{array}{ccc} (\mathcal{S}, \circ) & \longrightarrow & (\mathrm{GF}\,(2)^8, \oplus) \\ Q \circ \oplus_\beta \circ Q^{-1} & \longmapsto & [\beta] \end{array}$$

Let's have $f \in \mathcal{S}$. Recall $\psi = L^{-1} \circ \varphi$

$$\begin{tikzcd}
\mathrm{GF}(2^8) \arrow[r, "Q^{-1}"] \arrow[d, "f"'] & {} \arrow[r, "L^{-1}"] \arrow[d, "\oplus_{\varphi(f)}"] & {} \arrow[r, "\oplus_{(Q \circ L)^{-1}([\text{'}00\text{'}])}"] \arrow[d, "\oplus_{\psi(f)}"] & {} \arrow[d, "\oplus_{\psi(f)}"] \\
\mathrm{GF}(2^8) & {} \arrow[l, "Q"'] & {} \arrow[l, "L"'] & {} \arrow[l, "\oplus_{(Q \circ L)^{-1}([\text{'}00\text{'}])}"'] \arrow[ll, bend left, "A"'] \arrow[lll, bend left, "\widetilde{Q}"']
\end{tikzcd}$$

$$f = Q \circ \oplus_{\varphi(f)} \circ Q^{-1} = Q \circ L \circ \oplus_{\psi(f)} \circ L^{-1} \circ Q^{-1} = \widetilde{Q} \circ \oplus_{\psi(f)} \circ \widetilde{Q}^{-1}$$

Commutative diagram

$$
\begin{tikzcd}
GF(2^8) \arrow[r, "Q^{-1}"] \arrow[d, "f"'] & {} \arrow[r, "L^{-1}"] \arrow[d, "\oplus_{\varphi(f)}"'] & {} \arrow[r, "\oplus_{(Q\circ L)^{-1}(['00'])}"] \arrow[d, "\oplus_{\psi(f)}"'] & {} \arrow[d, "\oplus_{\psi(f)}"'] \\
GF(2^8) & {} \arrow[l, "Q"'] & {} \arrow[l, "L"'] & {} \arrow[l, "\oplus_{(Q\circ L)^{-1}(['00'])}"'] \arrow[ll, bend left, "A"'] \arrow[lll, bend left=40, "\widetilde{Q}"']
\end{tikzcd}
$$

- Define $A(x) = L(x \oplus (Q \circ L)^{-1}(['00'])) = L(x) \oplus Q^{-1}(['00'])$
- Define $\widetilde{Q} = Q \circ A$
- $f = Q \circ \oplus_{\varphi(f)} \circ Q^{-1} = Q \circ L \circ \oplus_{\psi(f)} \circ L^{-1} \circ Q^{-1} = \widetilde{Q} \circ \oplus_{\psi(f)} \circ \widetilde{Q}^{-1}$

Commutative diagram

$$
\begin{tikzcd}
GF(2^8) \arrow[r, "Q^{-1}"] \arrow[d, "f"'] & {} \arrow[r, "L^{-1}"] \arrow[d, "\oplus_{\varphi(f)}"'] & {} \arrow[r, "\oplus_{(Q \circ L)^{-1}(['00'])}"] \arrow[d, "\oplus_{\psi(f)}"'] & {} \arrow[d, "\oplus_{\psi(f)}"'] \\
GF(2^8) & {} \arrow[l, "Q"'] & {} \arrow[l, "L"'] & {} \arrow[l, "\oplus_{(Q \circ L)^{-1}(['00'])}"'] \arrow[ll, shift left=6, "A"'] \arrow[lll, shift left=12, "\widetilde{Q}"']
\end{tikzcd}
$$

- Observe $\widetilde{Q}^{-1}('00') = Q(L(0) \oplus Q^{-1}(['00'])) = Q(Q^{-1}(['00'])) = ['00']$ L is linear (unknown), A defined in this way so this holds (artificial)!
- $f = Q \circ \oplus_{\varphi(f)} \circ Q^{-1} = \widetilde{Q} \circ \oplus_{\psi(f)} \circ \widetilde{Q}^{-1}$, from commutative diagram
- $f('00') = \widetilde{Q}(\psi(f) \oplus \widetilde{Q}^{-1}('00')) = \widetilde{Q}(\psi(f))$
- Observe $\widetilde{Q}^{-1} \circ Q = A^{-1}$

Commutative diagram

$$
\begin{tikzcd}[column sep=huge, row sep=huge]
GF(2^8) \arrow[r, "Q^{-1}"] \arrow[d, "f"'] & {} \arrow[r, "L^{-1}"] \arrow[d, "\oplus_{\varphi(f)}"'] & {} \arrow[r, "\oplus_{(Q \circ L)^{-1}(['00'])}"] \arrow[d, "\oplus_{\psi(f)}"'] & {} \arrow[d, "\oplus_{\psi(f)}"'] \\
GF(2^8) & {} \arrow[l, "Q"'] & {} \arrow[l, "L"'] & {} \arrow[l, "\oplus_{(Q \circ L)^{-1}(['00'])}"'] \arrow[ll, bend left, "A"] \arrow[lll, bend left=40, "\widetilde{Q}"]
\end{tikzcd}
$$

- Observe $\widetilde{Q}^{-1}('00') = Q(L(0) \oplus Q^{-1}(['00'])) = Q(Q^{-1}(['00'])) = ['00']$ L is linear (unknown), A defined in this way so this holds (artificial)!
- $f = Q \circ \oplus_{\varphi(f)} \circ Q^{-1} = \widetilde{Q} \circ \oplus_{\psi(f)} \circ \widetilde{Q}^{-1}$, from commutative diagram
- $f('00') = \widetilde{Q}(\psi(f) \oplus \widetilde{Q}^{-1}('00')) = \widetilde{Q}(\psi(f))$
- Observe $\widetilde{Q}^{-1} \circ Q = A^{-1}$

Commutative diagram

$$\begin{tikzcd}[column sep=huge, row sep=huge]
GF(2^8) \arrow[r, "Q^{-1}"] \arrow[d, "f"'] & {} \arrow[r, "L^{-1}"] \arrow[d, "\oplus_{\varphi(f)}"'] & {} \arrow[r, "\oplus_{(Q \circ L)^{-1}(['00'])}"] \arrow[d, "\oplus_{\psi(f)}"'] & {} \arrow[d, "\oplus_{\psi(f)}"'] \\
GF(2^8) & {} \arrow[l, "Q"'] & {} \arrow[l, "L"'] & {} \arrow[l, "\oplus_{(Q \circ L)^{-1}(['00'])}"'] \arrow[ll, bend left, "A"] \arrow[lll, bend left=40, "\widetilde{Q}"]
\end{tikzcd}$$

- Observe $\widetilde{Q}^{-1}('00') = Q(L(0) \oplus Q^{-1}(['00'])) = Q(Q^{-1}(['00'])) = ['00']$
  L is linear (unknown), A defined in this way so this holds (artificial)!
- $f = Q \circ \oplus_{\varphi(f)} \circ Q^{-1} = \widetilde{Q} \circ \oplus_{\psi(f)} \circ \widetilde{Q}^{-1}$, from commutative diagram
- $f('00') = \widetilde{Q}(\psi(f) \oplus \widetilde{Q}^{-1}('00')) = \widetilde{Q}(\psi(f))$
- Observe $\widetilde{Q}^{-1} \circ Q = A^{-1}$

Commutative diagram

$$
\begin{tikzcd}
\mathrm{GF}(2^8) \arrow[r, "Q^{-1}"] \arrow[d, "f"'] & \cdot \arrow[r, "L^{-1}"] \arrow[d, "\oplus_{\varphi(f)}"'] & \cdot \arrow[r, "\oplus_{(Q\circ L)^{-1}(['00'])}"] \arrow[d, "\oplus_{\psi(f)}"'] & \cdot \arrow[d, "\oplus_{\psi(f)}"'] \\
\mathrm{GF}(2^8) & \cdot \arrow[l, "Q"'] & \cdot \arrow[l, "L"'] & \cdot \arrow[l, "\oplus_{(Q\circ L)^{-1}(['00'])}"'] \arrow[ll, bend left, "A"] \arrow[lll, bend left=40, "\widetilde{Q}"]
\end{tikzcd}
$$

- Observe $\widetilde{Q}^{-1}('00') = Q(L(0) \oplus Q^{-1}(['00'])) = Q(Q^{-1}(['00'])) = ['00']$ L is linear (unknown), A defined in this way so this holds (artificial)!
- $f = Q \circ \oplus_{\varphi(f)} \circ Q^{-1} = \widetilde{Q} \circ \oplus_{\psi(f)} \circ \widetilde{Q}^{-1}$, from commutative diagram
- $f('00') = \widetilde{Q}(\psi(f) \oplus \widetilde{Q}^{-1}('00')) = \widetilde{Q}(\psi(f))$
- Observe $\widetilde{Q}^{-1} \circ Q = A^{-1}$

Questions?

Thank you for your attention