RUHR-UNIVERSITÄT BOCHUM
Efficient Implementations of MQPKS on Constrained Devices
Peter Czypek,Stefan Heyse, Enrico Thomae

Efficient Implementations of MQPKS on Constrained Devices

Peter Czypek, Stefan Heyse, Enrico Thomae

CHES2012

11.09.2012

Motivation

- Quantum computers can solve Discrete Logarithm problem and Factorization problem
- Alternatives must be found
- MQ based cryptography is one solution
- Many MQ schemes were partially or fully broken in the past
- Few implementations exist of the remaining schemes
- Fair comparison of schemes was only possible theoretically

Goals

- Implement
  - all currently secure schemes
  - with the same security level
  - configurable code
  - including all currently known optimizations

- Show that MQ schemes are a good alternative to current schemes?

MQ Signature Schemes - Basics

- `sign()` maps the message to signature with the secret key
- `verify()` maps the signature to message with the public key
- If the verification result is not the original message, the signature is invalid
- `sign` and `verify` are inverses of each other
- `verify(sign(message)) = message`

$$\begin{tikzcd} \text{message } y \arrow[r, bend left, "\texttt{sign}"] & \text{signature } x \arrow[l, bend left, "\texttt{verify}"] \end{tikzcd}$$

MQ Signature Schemes - Basics

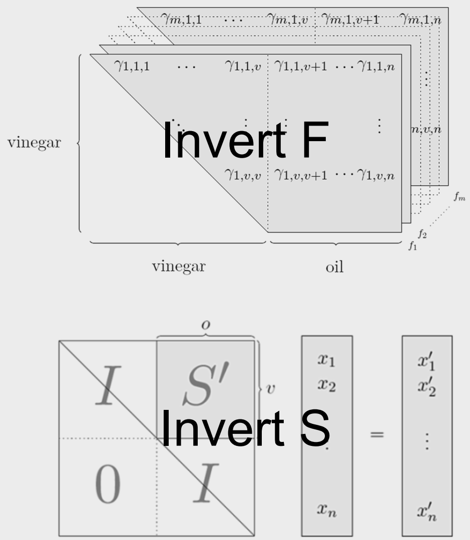
- Four maps exist in a general MQ scheme: *P*, *S*, *F*, and *T*
- *P* is the composition of *S*, *F*, and *T* and is the public key, $P = T \circ F \circ S$
- *S*, *F*, and *T* are the secret key

Inversion of *P* is hard because *P* is a large MQ system

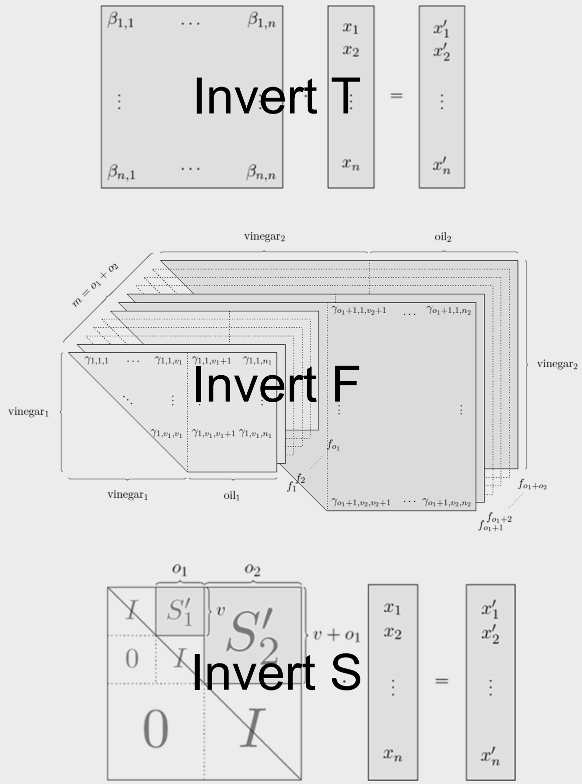
$$
\begin{tikzcd}
\text{signature } x \in \mathbb{F}_q^n \arrow[rr, "\mathcal{P}"] \arrow[rr, bend left=40, "\texttt{verify}"] \arrow[d, "S"'] & & \text{message } y \in \mathbb{F}_q^m \arrow[ll, bend right=60] \arrow[ll, bend left=80, "\texttt{sign}"] \\
x' \in \mathbb{F}_q^n \arrow[rr, "\mathcal{F}"'] & & x'' \in \mathbb{F}_q^m \arrow[u, "T"']
\end{tikzcd}
$$

Schemes

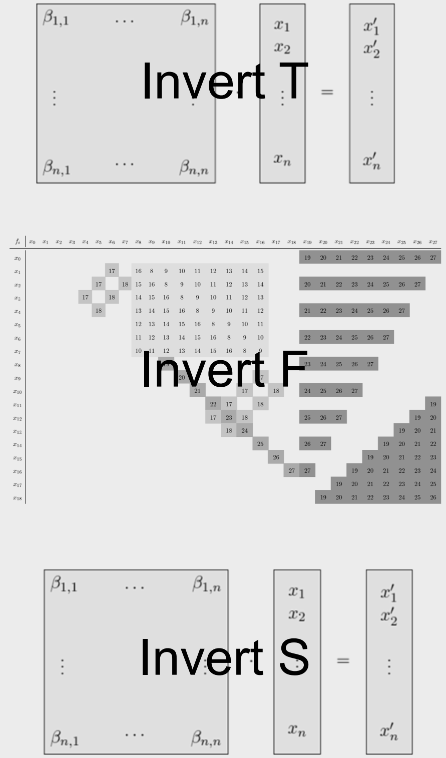
UOV



Rainbow



enTTS



RUB

Linear Maps

- Maps or transformations can also be seen as functions
- There exist two types of maps in MQ schemes: linear and MQ maps

- Linear maps mix variables and therefore “hide” existing structure

$$\begin{pmatrix} \beta_{1,1} & \cdots & \beta_{1,n} \\ \vdots & & \vdots \\ \beta_{n,1} & \cdots & \beta_{n,n} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} x'_1 \\ x'_2 \\ \vdots \\ x'_n \end{pmatrix}$$

$$x'_i = \beta_{i,1}x_1 + \beta_{i,2}x_2 + \ ... + \beta_{i,n}x_n$$

Peter Czypek,Stefan Heyse, Enrico Thomae

Inverting Linear Maps

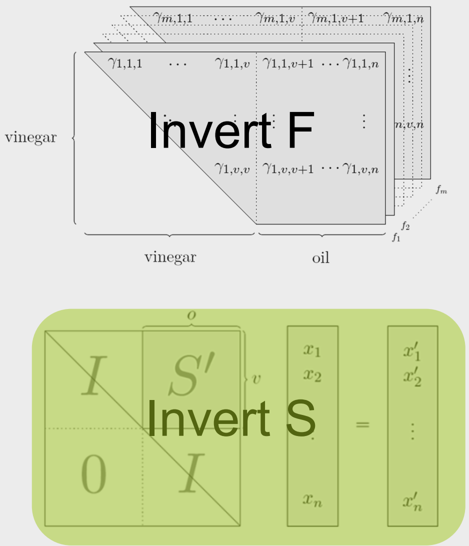
- *S* and *T* can be inverted by matrix inversion
- Matrix inversion can be done by Gaussian elimination algorithm for each column of identity matrix
- Inversion of a linear map is matrix vector multiplication with the inverse

$$\underbrace{\begin{pmatrix} \beta_{1,1} & \cdots & \beta_{1,n} \\ \vdots & & \vdots \\ \beta_{n,1} & \cdots & \beta_{n,n} \end{pmatrix}}_{T^{-1}} \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} x'_1 \\ x'_2 \\ \vdots \\ x'_n \end{pmatrix}$$

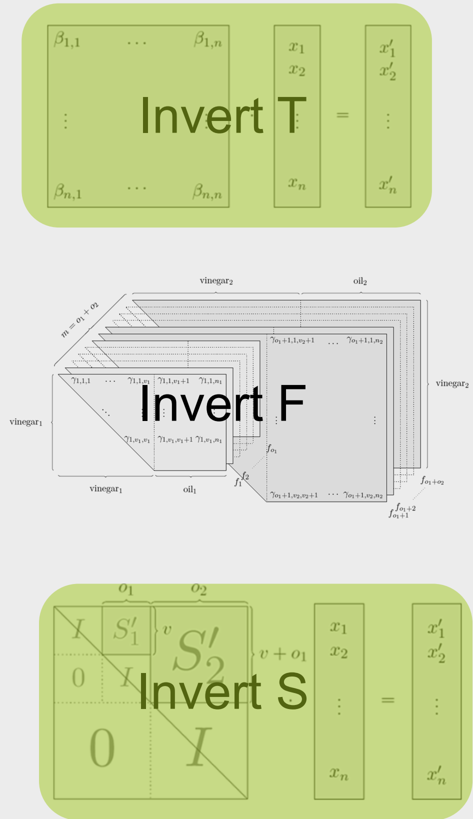
Peter Czypek,Stefan Heyse, Enrico Thomae

Schemes

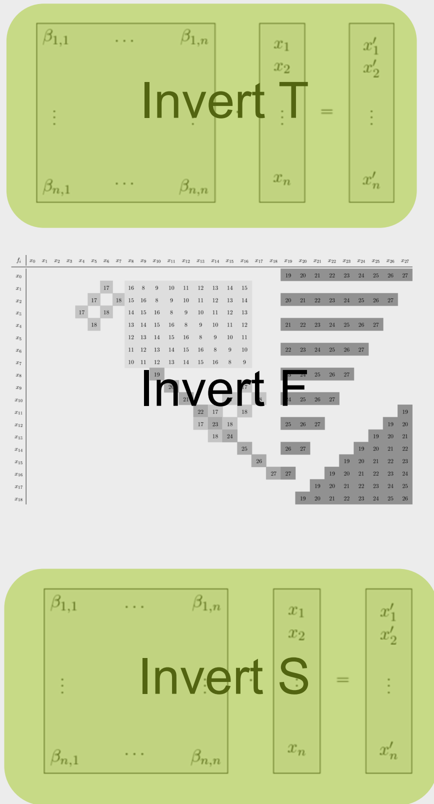
UOV



Rainbow



enTTS



Peter Czypek,Stefan Heyse, Enrico Thomae

MQ Maps

- *F* and *P* are MQ maps

$$\begin{tikzcd}
\text{signature } x \in \mathbb{F}_q^n \arrow[r, "\mathcal{P}"] \arrow[d, "S"'] & \text{message } y \in \mathbb{F}_q^m \\
x' \in \mathbb{F}_q^n \arrow[r, "\mathcal{F}"'] & x'' \in \mathbb{F}_q^m \arrow[u, "T"']
\end{tikzcd}$$

- *P* has no special structure and is large, therefore hard to invert

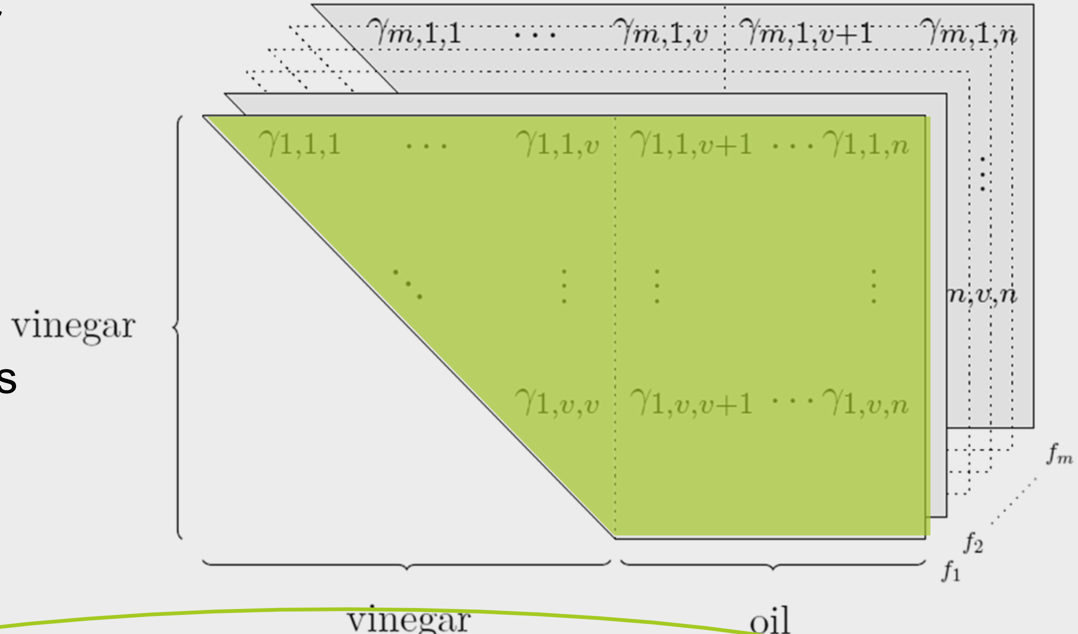
$$3\, x_1x_1 + 8\, x_1x_2 + 5x_1x_3 + 8\, x_2x_2 + 6x_2x_3 + 2x_3x_3 = m_1$$

$$1\, x_1x_1 + 7\, x_1x_2 + 9x_1x_3 + 3\, x_2x_2 + 7x_2x_3 + 2x_3x_3 = m_2$$

- A special structure in *F* is necessary to allow easy inversion
- This special structure is hidden by *S* and *T*

Inverting Central Maps - UOV

- Two variable groups: Oil & Vinegar
- Fix vinegar variables to make system linear
- A quadratic linear equation system remains after fixing
- Apply Gaussian elimination to get a solution for the oil variables



$x_1 \cdots x_4 \leftarrow$ fixed random values $r_1 \cdots r_4$

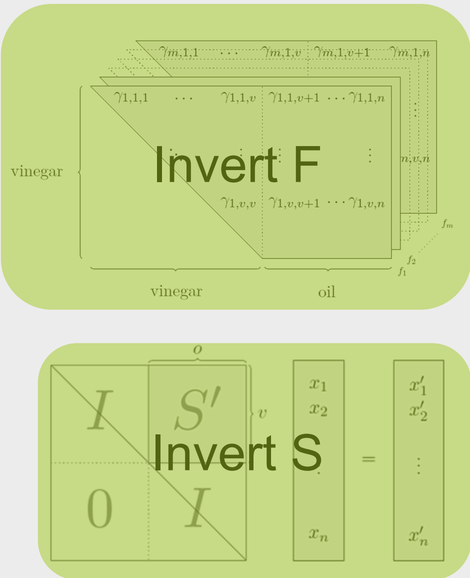
$$
\begin{aligned}
f_i = \underbrace{\begin{aligned}&\gamma_{i,1,1}r_1r_1 + \gamma_{i,1,2}r_1r_2 + \gamma_{i,1,3}r_1r_3 + \gamma_{i,1,4}r_1r_4\\ &+\gamma_{i,2,2}r_2r_2 + \gamma_{i,2,3}r_2r_3 + \gamma_{i,2,4}r_2r_4\\ &+\gamma_{i,3,3}r_3r_3 + \gamma_{i,3,4}r_3r_4\\ &+\gamma_{i,4,4}r_4r_4\end{aligned}}_{\sum \text{constant terms} = v_i}
\underbrace{\begin{aligned}&+\gamma_{i,1,5}r_1x_5 + \gamma_{i,1,6}r_1x_6\\ &+\gamma_{i,2,5}r_2x_5 + \gamma_{i,2,6}r_2x_6\\ &+\gamma_{i,3,5}r_3x_5 + \gamma_{i,3,6}r_3x_6\\ &+\gamma_{i,4,5}r_4x_5 + \gamma_{i,4,6}r_4x_6\end{aligned}}_{\sum \text{linear terms} = c_{i,5}x_5, c_{i,6}x6}
\end{aligned}
$$

$$
\begin{pmatrix} c_{1,5} & c_{1,6} \\ c_{2,5} & c_{2,6} \end{pmatrix} \cdot \begin{pmatrix} x_5 \\ x_6 \end{pmatrix} = \begin{pmatrix} m_1 - v_1 \\ m_2 - v_2 \end{pmatrix}
$$

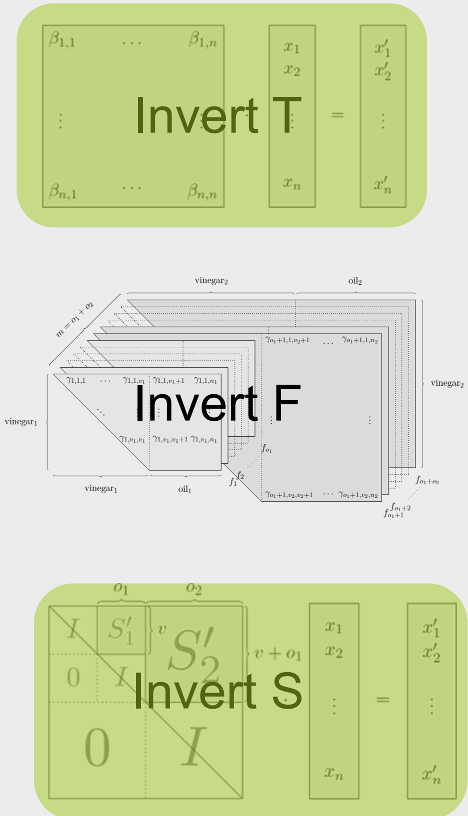
Peter Czypek,Stefan Heyse, Enrico Thomae

Schemes

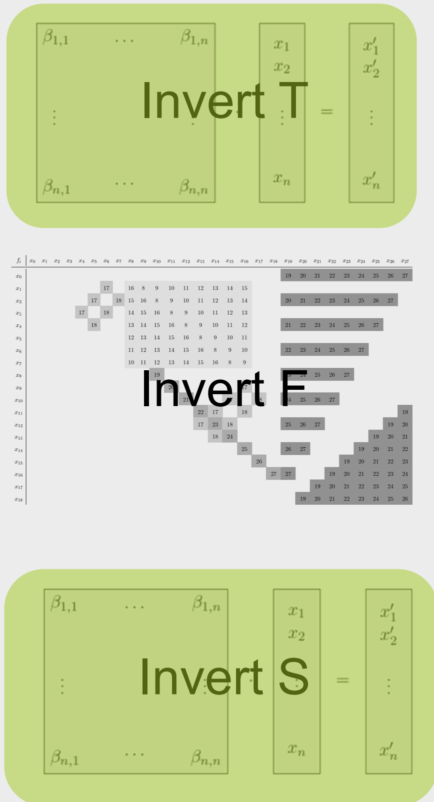
UOV



Rainbow

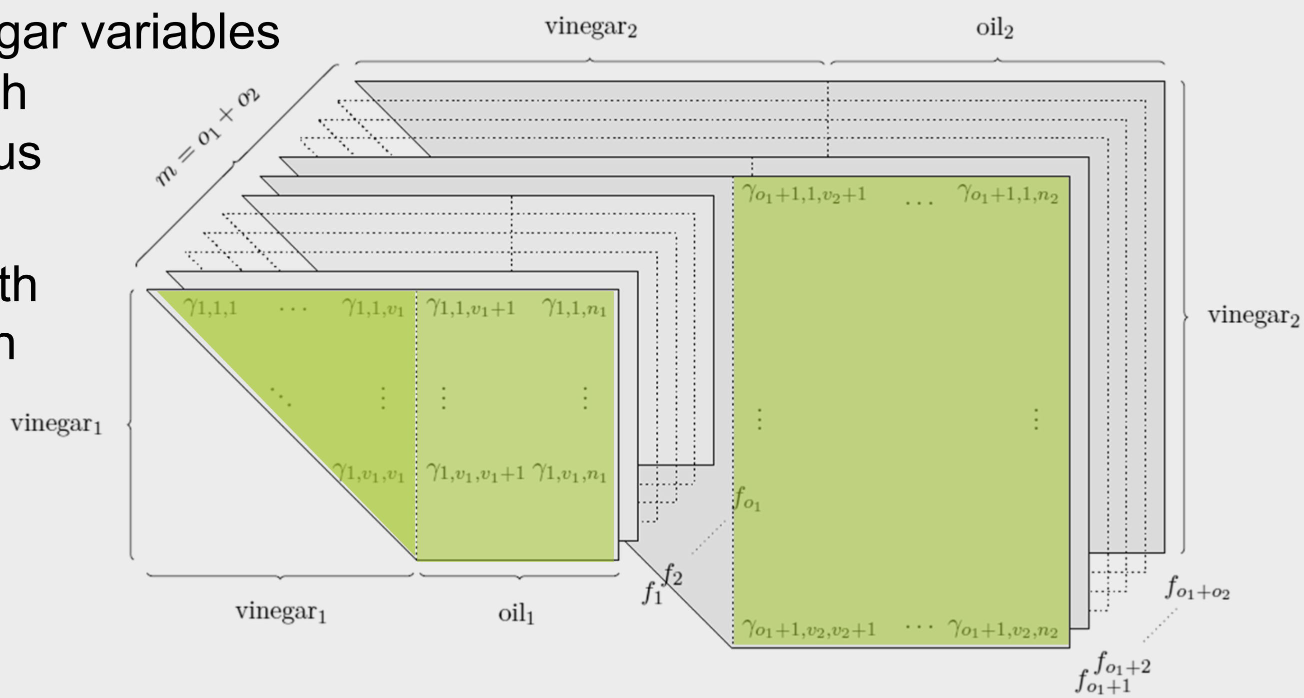


enTTS



Inverting Central Maps - Rainbow

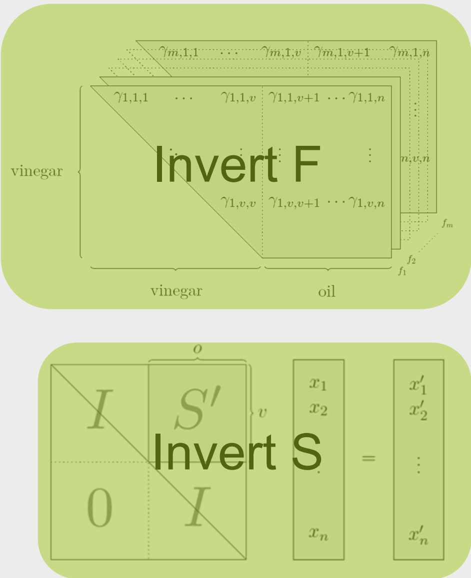
- Two or more layers (like a Rainbow)
- Solve first layer as normal UOV instance
- In next layer fix vinegar variables not randomly but with solution from previous layer
- Solve layer again with Gaussian elimination



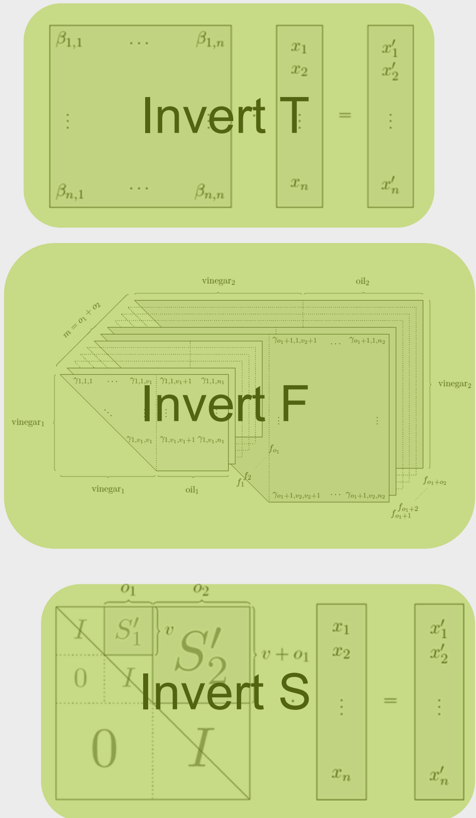
Rainbow(3,2,4) : $x_1\ x_2\ x_3$ | $x_4\ x_5$ | $x_6\ x_7\ x_8\ x_9$

Schemes

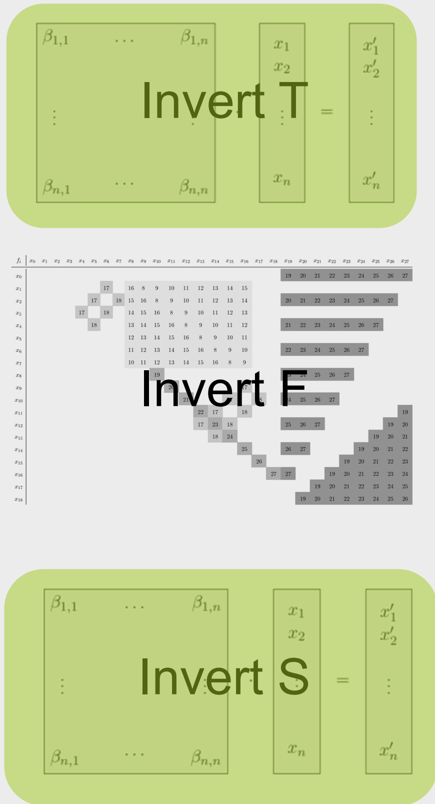
UOV



Rainbow



enTTS



Inverting Central Maps - enTTS

$$f_i = x_i + \sum_{j=1}^{2\ell-3} \gamma_{ij} x_j x_{2\ell-2+(i+j+1 \bmod 2\ell-1)} \qquad \text{for } 2\ell-2 \leq i \leq 4\ell-4,$$

$$f_i = x_i + \sum_{j=1}^{\ell-2} \gamma_{ij} x_{i+j-(4\ell-3)} x_{i-j-2\ell} + \sum_{j=\ell-1}^{2\ell-3} \gamma_{ij} x_{i+j-3\ell+3} x_{i-j+\ell-2}$$

$$\text{for } i = 4\ell-3 \text{ or } 4\ell-2,$$

$$f_i = x_i + \gamma_{i0} x_{i-2\ell+1} x_{i-2\ell-1} + \sum_{j=4\ell-1}^{i} \gamma_{i,j-(4\ell-2)} x_{2(i-j)} x_j$$

$$+ \sum_{j=i+1}^{6\ell-3} \gamma_{i,j-(4\ell-2)} x_{4\ell-1+i-j} x_j \qquad \text{for } 4\ell-1 \leq i \leq 6\ell-3$$

Peter Czypek,Stefan Heyse, Enrico Thomae

Inverting Central Maps – enTTS

f_i	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}	x_{18}	x_{19}	x_{20}	x_{21}	x_{22}	x_{23}	x_{24}	x_{25}	x_{26}	x_{27}
x_0																				19	20	21	22	23	24	25	26	27
x_1							17		16	8	9	10	11	12	13	14	15											
x_2						17		18	15	16	8	9	10	11	12	13	14			20	21	22	23	24	25	26	27	
x_3					17		18		14	15	16	8	9	10	11	12	13											
x_4						18			13	14	15	16	8	9	10	11	12			21	22	23	24	25	26	27		
x_5									12	13	14	15	16	8	9	10	11											
x_6									11	12	13	14	15	16	8	9	10			22	23	24	25	26	27			
x_7									10	11	12	13	14	15	16	8	9											
x_8											19									23	24	25	26	27				
x_9												20					17											
x_{10}													21			17		18		24	25	26	27					
x_{11}														22	17		18											19
x_{12}														17	23	18				25	26	27					19	20
x_{13}															18	24										19	20	21
x_{14}																	25			26	27				19	20	21	22
x_{15}																		26						19	20	21	22	23
x_{16}																			27	27			19	20	21	22	23	24
x_{17}																						19	20	21	22	23	24	25
x_{18}																					19	20	21	22	23	24	25	26

Peter Czypek,Stefan Heyse, Enrico Thomae

Inverting Central Maps - enTTS

- enTTS Layer 1:
  - Fix x_1 to x_7 randomly
  - Multiply with coefficients to get a LES
  - Solve with Gaussian elimination

$$\begin{pmatrix}
1 & p_{8,1}x_1 & p_{8,2}x_2 & p_{8,3}x_3 & p_{8,4}x_4 & p_{8,5}x_5 & p_{8,6}x_6 & p_{8,7}x_7 & 0 \\
0 & 1 & p_{9,1}x_1 & p_{9,2}x_2 & p_{9,3}x_3 & p_{9,4}x_4 & p_{9,5}x_5 & p_{9,6}x_6 & p_{9,7}x_7 \\
p_{10,7}x_7 & 0 & 1 & p_{10,1}x_1 & p_{10,2}x_2 & p_{10,3}x_3 & p_{10,4}x_4 & p_{10,5}x_5 & p_{10,6}x_6 \\
p_{11,6}x_6 & p_{11,7}x_7 & 0 & 1 & p_{11,1}x_1 & p_{11,2}x_2 & p_{11,3}x_3 & p_{11,4}x_4 & p_{11,5}x_5 \\
p_{12,5}x_5 & p_{12,6}x_6 & p_{12,7}x_7 & 0 & 1 & p_{12,1}x_1 & p_{12,2}x_2 & p_{12,3}x_3 & p_{12,4}x_4 \\
p_{13,4}x_4 & p_{13,5}x_5 & p_{13,6}x_6 & p_{13,7}x_7 & 0 & 1 & p_{13,1}x_1 & p_{13,2}x_2 & p_{13,3}x_3 \\
p_{14,3}x_3 & p_{14,4}x_4 & p_{14,5}x_5 & p_{14,6}x_6 & p_{14,7}x_7 & 0 & 1 & p_{14,1}x_1 & p_{14,2}x_2 \\
p_{15,2}x_2 & p_{15,3}x_3 & p_{15,4}x_4 & p_{15,5}x_5 & p_{15,6}x_6 & p_{15,7}x_7 & 0 & 1 & p_{15,1}x_1 \\
p_{16,1}x_1 & p_{16,2}x_2 & p_{16,3}x_3 & p_{16,4}x_4 & p_{16,5}x_5 & p_{16,6}x_6 & p_{16,7}x_7 & 0 & 1
\end{pmatrix} \cdot \begin{pmatrix} x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \\ x_{13} \\ x_{14} \\ x_{15} \\ x_{16} \end{pmatrix} = \begin{pmatrix} y_8 \\ y_9 \\ y_{10} \\ y_{11} \\ y_{12} \\ y_{13} \\ y_{14} \\ y_{15} \\ y_{16} \end{pmatrix}$$

enTTS(20,28) : $x_0\ x_1\ x_2\ x_3\ x_4\ x_5\ x_6\ x_7\ x_8\ x_9\ x_{10}\ x_{11}\ x_{12}\ x_{13}\ x_{14}\ x_{15}\ x_{16}\ x_{17}\ x_{18}\ x_{19}\ x_{20}\ x_{21}\ x_{22}\ x_{23}\ x_{24}\ x_{25}\ x_{26}\ x_{27}$

Inverting Central Maps - enTTS

- enTTS Layer 2:
  - Can be solved directly

$$
\begin{aligned}
x_{17} &= y_{17} - p_{17,1}x_1x_6 - p_{17,2}x_2x_5 - p_{17,3}x_3x_4 - p_{17,4}x_9x_{16} - p_{17,5}x_{10}x_{15} - p_{17,6}x_{11}x_{14} - p_{17,7}x_{12}x_{13} \\
x_{18} &= y_{18} - p_{18,1}x_2x_7 - p_{18,2}x_3x_6 - p_{18,3}x_4x_5 - p_{18,4}x_{10}x_{17} - p_{18,5}x_{11}x_{16} - p_{18,6}x_{12}x_{15} - p_{18,7}x_{13}x_{14}
\end{aligned}
$$

enTTS(20,28) : x_0 x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9 x_{10} x_{11} x_{12} x_{13} x_{14} x_{15} x_{16} x_{17} x_{18} x_{19} x_{20} x_{21} x_{22} x_{23} x_{24} x_{25} x_{26} x_{27}

Inverting Central Maps - enTTS

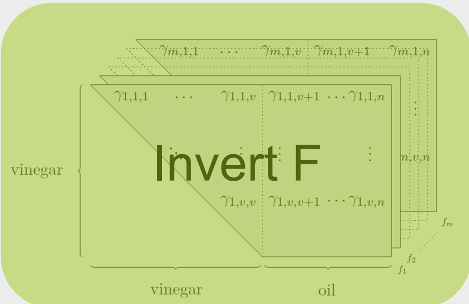
- enTTS Layer 3:
  - Fix x_0 randomly
  - Multiply already known values with coefficients to get a LES
  - Solve LES

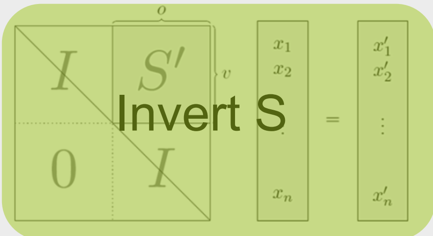
$$
\begin{pmatrix}
1+p_{19,1}x_0 & p_{19,2}x_{18} & p_{19,3}x_{17} & p_{19,4}x_{16} & p_{19,5}x_{15} & p_{19,6}x_{14} & p_{19,7}x_{13} & p_{19,8}x_{12} & p_{19,9}x_{11} \\
p_{20,1}x_2 & 1+p_{20,2}x_0 & p_{20,3}x_{18} & p_{20,4}x_{17} & p_{20,5}x_{16} & p_{20,6}x_{15} & p_{20,7}x_{14} & p_{20,8}x_{13} & p_{20,9}x_{12} \\
p_{21,1}x_4 & p_{21,2}x_2 & 1+p_{21,3}x_0 & p_{21,4}x_{18} & p_{21,5}x_{17} & p_{21,6}x_{16} & p_{21,7}x_{15} & p_{21,8}x_{14} & p_{21,9}x_{13} \\
p_{22,1}x_6 & p_{22,2}x_4 & p_{22,3}x_2 & 1+p_{22,4}x_0 & p_{22,5}x_{18} & p_{22,6}x_{17} & p_{22,7}x_{16} & p_{22,8}x_{15} & p_{22,9}x_{14} \\
p_{23,1}x_8 & p_{23,2}x_6 & p_{23,3}x_4 & p_{23,4}x_2 & 1+p_{23,5}x_0 & p_{23,6}x_{18} & p_{23,7}x_{17} & p_{23,8}x_{16} & p_{23,9}x_{15} \\
p_{24,1}x_{10} & p_{24,2}x_8 & p_{24,3}x_6 & p_{24,4}x_4 & p_{24,5}x_2 & 1+p_{24,6}x_0 & p_{24,7}x_{18} & p_{24,8}x_{17} & p_{24,9}x_{16} \\
p_{25,1}x_{12} & p_{25,2}x_{10} & p_{25,3}x_8 & p_{25,4}x_6 & p_{25,5}x_4 & p_{25,6}x_2 & 1+p_{25,7}x_0 & p_{25,8}x_{18} & p_{25,9}x_{17} \\
p_{26,1}x_{14} & p_{26,2}x_{12} & p_{26,3}x_{10} & p_{26,4}x_8 & p_{26,5}x_6 & p_{26,6}x_4 & p_{26,7}x_2 & 1+p_{26,8}x_0 & p_{26,9}x_{18} \\
p_{27,1}x_{16} & p_{27,2}x_{14} & p_{27,3}x_{12} & p_{27,4}x_{10} & p_{27,5}x_8 & p_{27,6}x_6 & p_{27,7}x_4 & p_{27,8}x_2 & 1+p_{27,9}x_0
\end{pmatrix}
\cdot
\begin{pmatrix}
x_{19} \\ x_{20} \\ x_{21} \\ x_{22} \\ x_{23} \\ x_{24} \\ x_{25} \\ x_{26} \\ x_{27}
\end{pmatrix}
=
\begin{pmatrix}
y_{19} - p_{19,0}x_8x_{10} \\
y_{20} - p_{20,0}x_9x_{11} \\
y_{21} - p_{21,0}x_{10}x_{12} \\
y_{22} - p_{22,0}x_{11}x_{13} \\
y_{23} - p_{23,0}x_{12}x_{14} \\
y_{24} - p_{24,0}x_{13}x_{15} \\
y_{25} - p_{25,0}x_{14}x_{16} \\
y_{26} - p_{26,0}x_{15}x_{17} \\
y_{27} - p_{27,0}x_{16}x_{18}
\end{pmatrix}
$$

enTTS(20,28) : $x_0\ x_1\ x_2\ x_3\ x_4\ x_5\ x_6\ x_7\ x_8\ x_9\ x_{10}\ x_{11}\ x_{12}\ x_{13}\ x_{14}\ x_{15}\ x_{16}\ x_{17}\ x_{18}\ x_{19}\ x_{20}\ x_{21}\ x_{22}\ x_{23}\ x_{24}\ x_{25}\ x_{26}\ x_{27}$

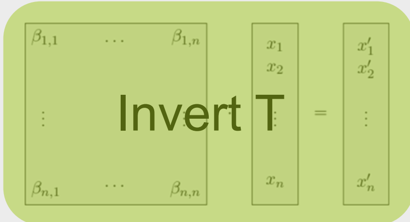
Schemes

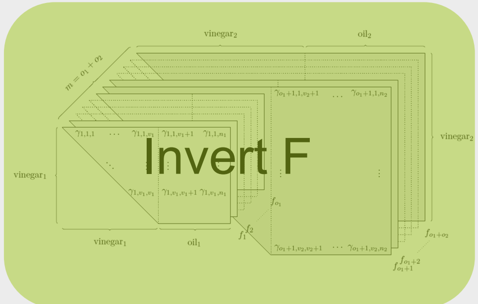
UOV

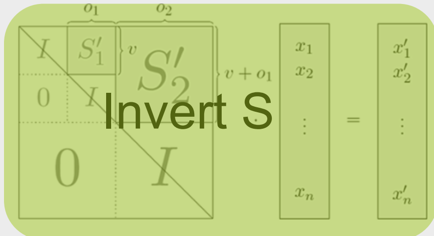




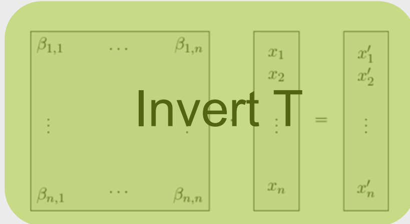
Rainbow

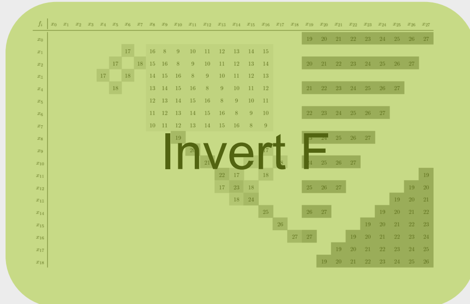


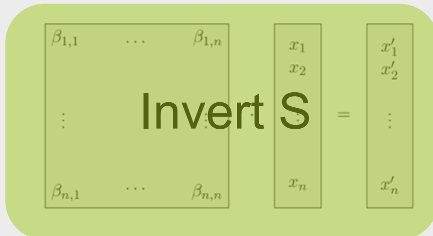




enTTS







Optimizations - Reduced Polynomials

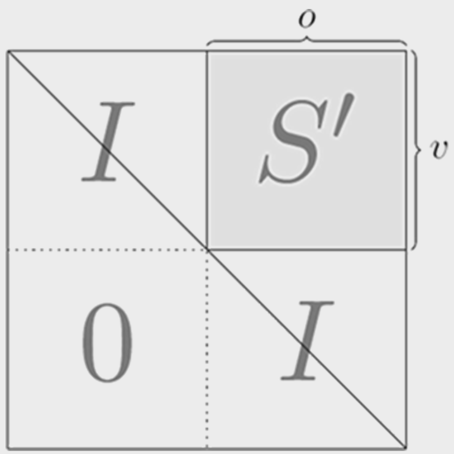
- Leaving out linear and constant terms in polynomials saves time and space
- Can be applied to UOV and Rainbow

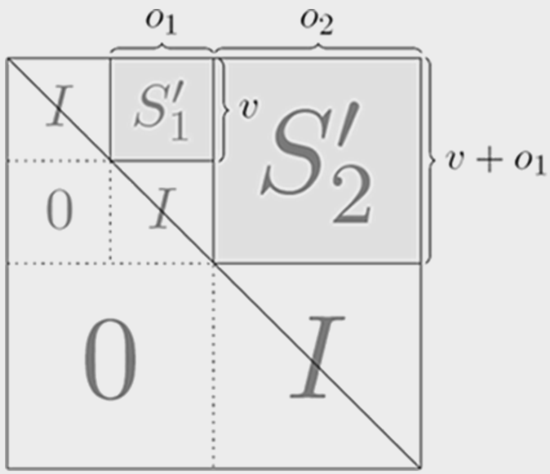
$$p^{(k)}(x_1,\ldots,x_n) := \sum_{1\le i\le j\le n} \gamma_{ij}^{(k)} x_i x_j + \cancel{\sum_{1\le i\le n} \beta_i^{(k)}} x_i + \cancel{\alpha^{(k)}}$$

- In the linear transformations the constant parts are also left out

Optimizations - Self Invertible Linear Maps

- In case of UOV and Rainbow S can be chosen of the form:

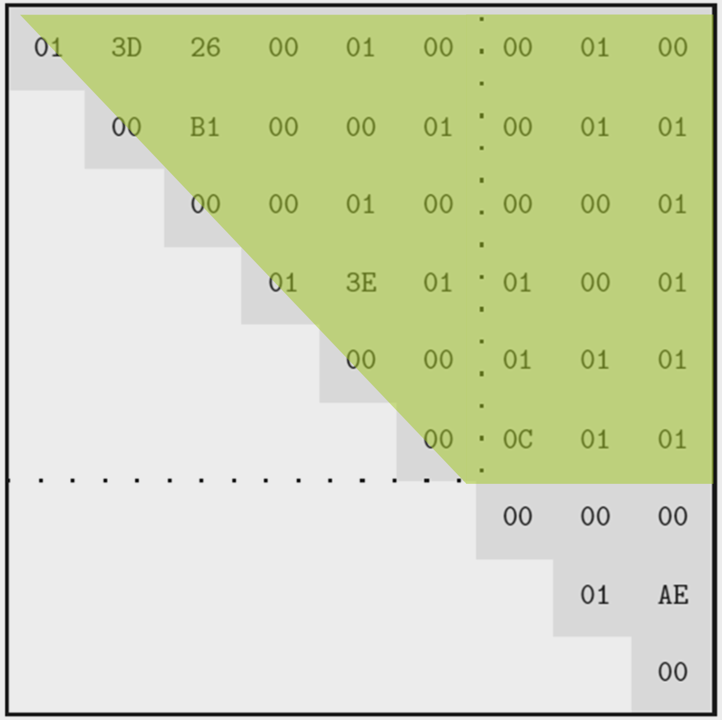




- S is self invertible $S^{-1} = S$,so no inversion is necessary.
- Multiplications in UOV signature generation are reduced from $n \cdot n$ to $o \cdot v$
- Private key is smaller

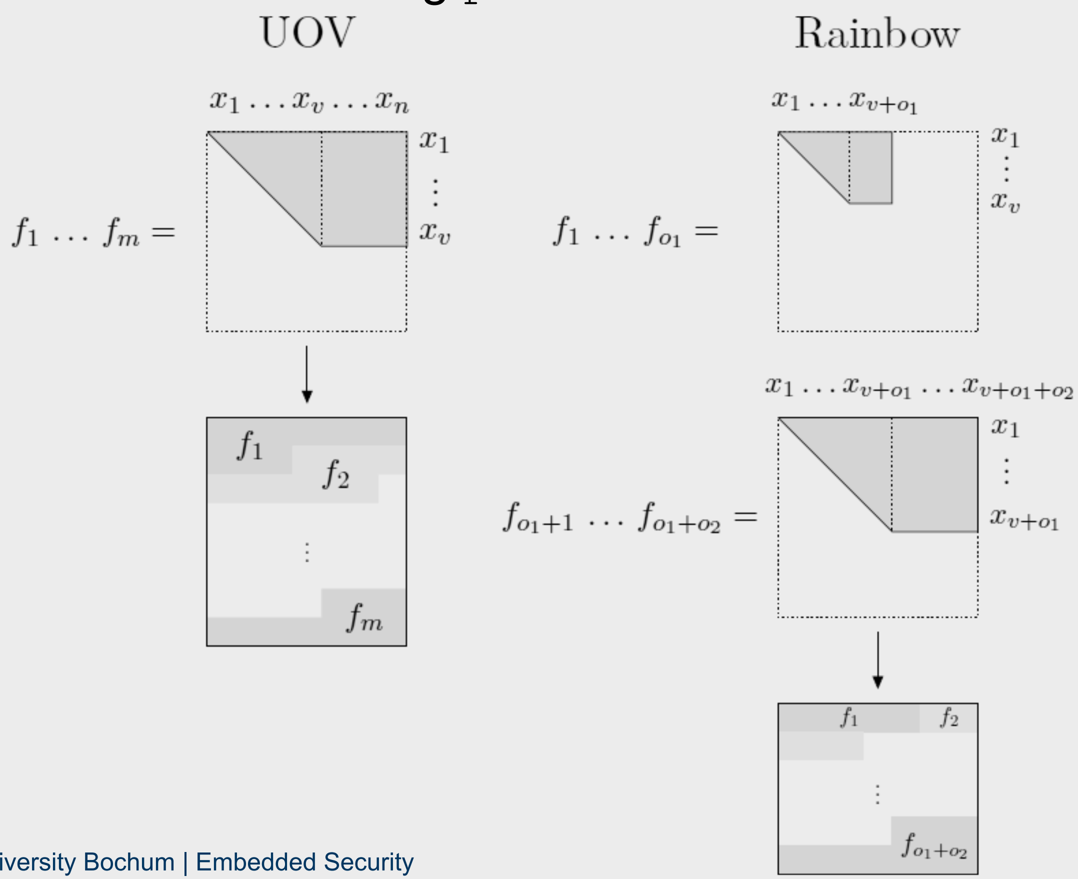
Optimizations - 0/1 UOV

- 0/1 UOV is an optimization for UOV
- Petzold, Thomae, Wolf et. al showed that large parts of the public key can be chosen randomly fixed
- This part can be treated as a system parameter and is not part of the public key anymore
- Faster verification is possible because the arithmetic in GF(2) is easier: 1= copy or 0 = not
  - An additional check is necessary if an element is from GF(2) or GF(2^8)
- Key generation: First choose *P* and then calculate *F*



Implementation - Central Map Memory Mapping

- Keys are saved without zeros
- Serial read out using `pointer++`



Implementation – Exponential Representation

- GF(2^8) arithmetic with table look up
- Multiplication is addition in exponent mod (2^m-1)

mul(a,b) = exp(log(a)+log(b) mod (2^m-1)) 3 `pgm_read()`

- Saving memory access by keeping temporary results in exponential representation when next operation is a multiplication

mul(mul(a,b) , c) = exp(log[exp(log(a)+log(b) mod (2^m-1))]+log[c] mod (2^m-1)) 6 `pgm_read()`

mul(mul(a,b) , c) = exp((log(a)+log(b) mod (2^m-1)) +log[c] mod (2^m-1)) 4 `pgm_read()`

- Keys are saved in exponential representation, too.

Implementation – Generic Code

- Heavy use of #define
- Code generator for enTTS
- Increasing parameters is very easy

```
/* ----------- SIZES --------------- */
#define __O __M
#define __N (uint16_t)(__V+__O)
#define __LENGTH_OV (uint16_t)((__O*__V)+((__V
#define __LENGTH_F (uint16_t)__LENGTH_OV*__M
#define __LENGTH_L (uint16_t)__N*__N
#define __LENGTH_P (uint16_t)(__M*(__N*(__N+1)
#define __D (uint16_t)((__V*(__V+1))/2)
#define __D2 (uint16_t)((__O*(__O+1))/2)
```

```
for(m=0; m<__M; m++) //all polynomials
{
    i=0;
    oil[m]=message[m]; //copy message to oil, because gauss awaits it in there later
    for(k=0; k<__V; k++)
    {
        for(j=k; j<__V; j++) // read in coeffitiens of F in exponential representation
        {
            oil[m] ^= mul_x_ee(vinegar_quadrat[i++],pgm_read_byte_far((pointer_f++)));
        }
        for(j=0; j<__O; j++)  //vinegar x oil, both in exponential form
        {
            lgs[(m*__M)+j] ^= mul_x_ee(vinegar[k],pgm_read_byte_far((pointer_f++)));
        }
    }
}
```

Comparison – Parameter Choice

- Due to the 8bit micro controller GF(2^8) was chosen as the field
- To be able to compare the schemes on equal conditions parameters for equal security levels are necessary
- For every scheme exist different attacks

Scheme	Security	Parameter	Direct attack	Band Separation	MinRank	HighRank	Kipnis-Shamir	Reconciliation
UOV (o, v)	2^{64}	$(21, 28)$	2^{67} $(g = 1)$	-	-	-	2^{66}	2^{131} $(k = 2)$
	2^{80}	$(28, 37)$	2^{85} $(g = 1)$	-	-	-	2^{83}	2^{166} $(k = 2)$
	2^{128}	$(44, 59)$	2^{130} $(g = 1)$	-	-	-	2^{134}	2^{256} $(k = 2)$
Rainbow (v, o_1, o_2)	2^{64}	$(15, 10, 10)$	2^{67} $(g = 1)$	2^{70}	2^{141}	2^{93}	2^{125}	2^{242} $(k = 6)$
	2^{80}	$(18, 13, 14)$	2^{85} $(g = 1)$	2^{81}	2^{167}	2^{126}	2^{143}	2^{254} $(k = 5)$
	2^{128}	$(36, 21, 22)$	2^{131} $(g = 2)$	2^{131}	2^{313}	2^{192}	2^{290}	2^{523} $(k = 7)$
enTTS (ℓ, m, n)	2^{64}	$(7, 28, 40)$	2^{89} $(g = 1)$	2^{68}	2^{126}	2^{117}	2^{127}	-
	2^{80}	$(9, 36, 52)$	2^{110} $(g = 2)$	2^{85}	2^{159}	2^{151}	2^{160}	-
	2^{128}	$(15, 60, 88)$	2^{176} $(g = 3)$	2^{131}	2^{258}	2^{249}	2^{259}	-

Peter Czypek,Stefan Heyse, Enrico Thomae

Comparison - Sign

	Scheme	n	m	private Key [Byte]	Parameter [Byte]	Clockcyles x 1000	Time[ms] @32MHz	Code Size [Byte]
	enTTS(5, 20, 28)	28	20	1351	*	153	4.79	12890
	enTTS(5, 20, 28)[YCCC06]	28	20	1417	*	568[1]	17.75[2]	-
2^{64}	UOV(21, 28)	49	21	21462	*	1,615	50.49	2188
	0/1 UOV(21, 28)	49	21	12936	8526	1,577	49.29	2258
	Rainbow(15, 10, 10)	35	20	9250	*	848	26.51	4162
	enTTS(7, 28, 40)	40	28	2731	*	332	10.37	24898
2^{80}	UOV(28, 37)	65	28	49728	*	3,637	113.66	2188
	0/1 UOV(28, 37)	65	28	30044	19684	3,526	110.20	2258
	Rainbow(18, 13, 14)	45	27	19682	*	1,740	54.38	4162
	enTTS(9, 36, 52)	52	36	4591	*	609	19.03	41232
2^{128}	UOV(44, 59)	103	44	194700	*	13,314	416.07	2188
	0/1 UOV(44, 59)	103	44	116820	77880	12,782	399.43	2258
	Rainbow(36, 21, 22)	79	43	97675	*	8,227	257.11	4162
	enTTS(15, 60, 88)	88	60	13051	*	2,142	66.94	116698

Comparison - Verify

	Scheme	n	m	public Key [Byte]	Parameter [Byte]	Clockcyles x 1000	Time[ms] @32MHz	Code Size [Byte]
	enTTS(5, 20, 28)	28	20	8120	*	1,126	35.22	827
	enTTS(5, 20, 28)[YCCC06]	28	20	8680	*	5,808[1]	181.5[2]	-
2^{64}	UOV(21, 28)	49	21	25725	*	1,690	52.83	466
	0/1 UOV(21, 28)	49	21	4851	20874	1,395	43.60	578
	Rainbow(15, 10, 10)	35	20	12600	*	1,010	31.58	466
	enTTS(7, 28, 40)	40	28	22960	*	2,558	79.95	827
2^{80}	UOV(28, 37)	65	28	60060	*	3,911	122.23	466
	0/1 UOV(28, 37)	65	28	11368	48692	3,211	100.37	578
	Rainbow(18, 13, 14)	45	27	27945	*	2,214	69.19	466
	enTTS(9, 36, 52)	52	36	49608	*	6,658	208.07	827
2^{128}	UOV(44, 59)	103	44	235664	*	14,134	441.70	466
	0/1 UOV(44, 59)	103	44	43560	192104	13,569	424.04	578
	Rainbow(36, 21, 22)	79	43	135880	*	9,216	288.01	466
	enTTS(15, 60, 88)	88	60	234960	*	3,0789	962.17	827

Comparison – Other Schemes

- Our implementations:
  - enTTS(5,20,28) [security < 2^{64}] sign in 4.79 ms / verify 35.22 ms
  - enTTS(9,36,52) [2^{80}] sign in 19.03 ms / verify in 208.07 ms
  - Rainbow(18,13,17) [2^{80}] sign in 54.38 ms / verify in 69.19 ms

- Other schemes:

Method	Time[ms]@32MHz sign	Time[ms]@32MHz verify
enTTS(5, 20, 28)[YCCC06]	17.75[1]	181.5[1]
ECC-P160 (SECG) [GPW+04]	203[1]	203[1]
ECC-P192 (SECG) [GPW+04]	310[1]	310[1]
ECC-P224 (SECG) [GPW+04]	548[1]	548[1]
RSA-1024 [GPW+04]	2,748[1]	108[1]
RSA-2048 [GPW+04]	20,815[1]	485[1]
NTRU-251-127-31 sign [DPP08]	143[1]	-

Peter Czypek,Stefan Heyse, Enrico Thomae

Conclusion

Rainbow

Verification time

Public key size

enTTS

Signature time

Secret key size

UOV & 0/1 UOV

Code size

(Public key size)

Future aspects

- 0/1 UOV could be improved by using a generated or cyclic system parameter instead a fixed one
- 0/1 UOV could save 8 elements in one byte instead of saving 1 bit in a byte
- The focus of this work was on fast schemes, the code size / time trade-off could be investigated further
- Assembler implementations could speed up the schemes even more

RUHR-UNIVERSITÄT BOCHUM

Peter Czypek,Stefan Heyse, Enrico Thomae

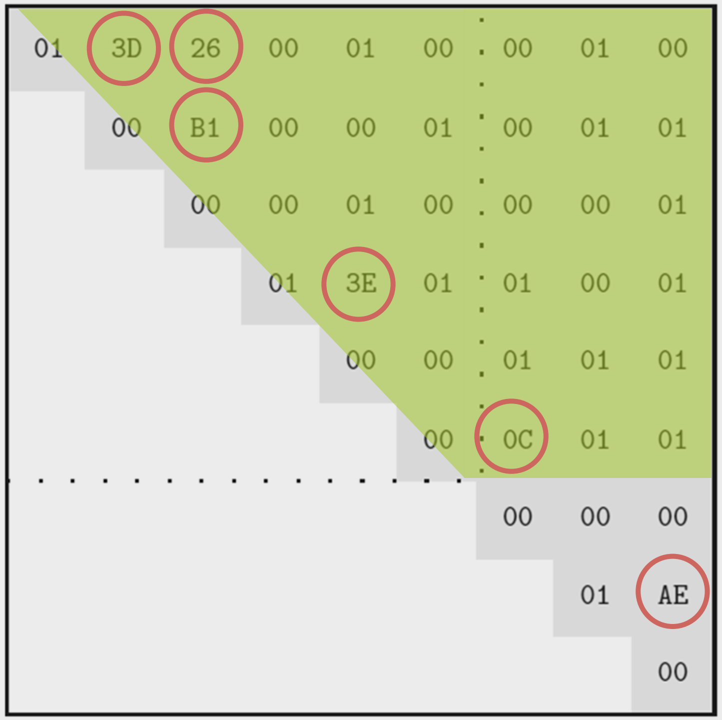
Thank you for your attention.
Any Questions?

Peter Czypek,Stefan Heyse, Enrico Thomae

Optimizations - 0/1 UOV

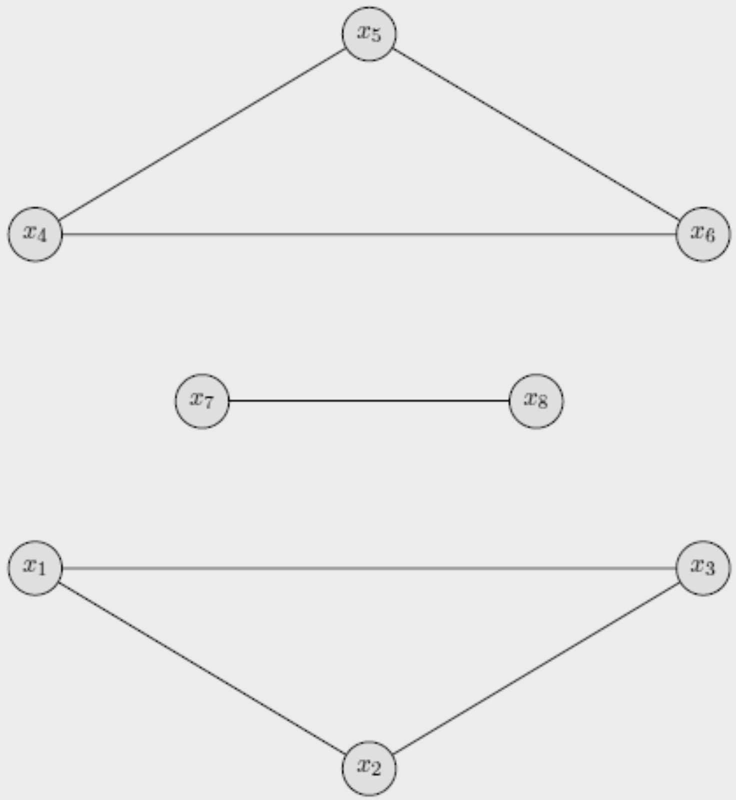
- To prevent a reduction of the key to elements only from GF(2), a special monomial ordering is necessary

- Elements must be combined in a way that even when many GF(2^8) elements are fixed the key has still elements from GF(2^8)



Peter Czypek,Stefan Heyse, Enrico Thomae

0/1 UOV Key Gen – Complementary Turań Graph



$$\ldots + x_1x_2 + x_1x_3 + x_2x_3 + x_4x_5 + x_4x_6 + x_5x_8 + x_7x_8$$

Peter Czypek,Stefan Heyse, Enrico Thomae

0/1 UOV Key Gen – Choosing S

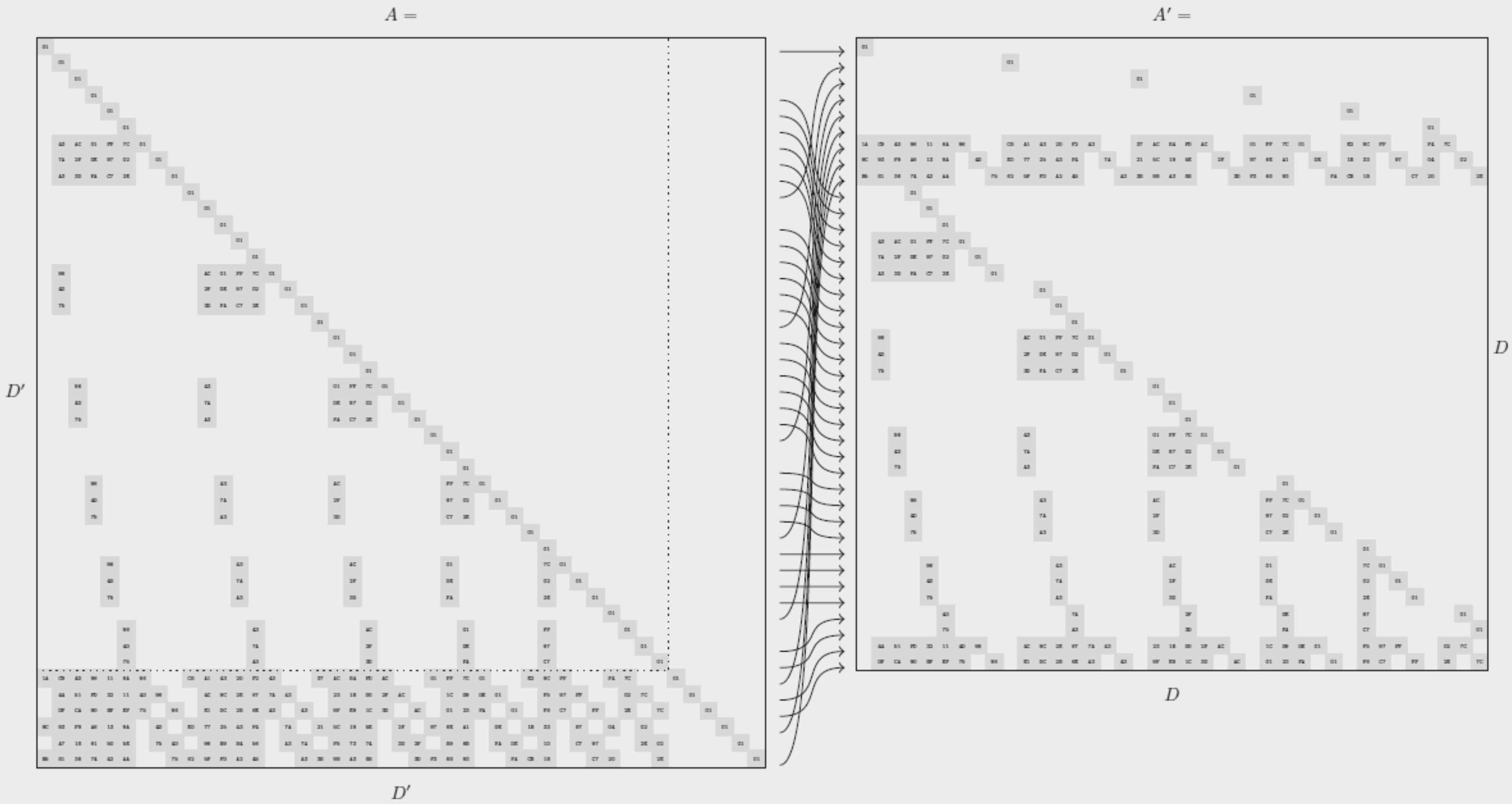
$$S = S^{-1} = \begin{array}{c c} \begin{bmatrix} 01 & & & & & & 47 & A8 & 9E \\ & 01 & & & & & A1 & 55 & AD \\ & & 01 & & & & 3D & F4 & 0A \\ & & & 01 & & & C2 & 9F & 2B \\ & & & & 01 & & 8E & 01 & C5 \\ & & & & & 01 & 54 & 83 & 12 \\ & & & & & & 01 & & \\ & & & & & & & 01 & \\ & & & & & & & & 01 \end{bmatrix} & n \\ n & \end{array}$$

0/1 UOV Key Gen – Choosing B from GF2

$$B = \begin{bmatrix} 01 & & & 01 & 00 & 00 & 00 & 01 & 00 & 00 & 01 & 00 & 00 & 01 & 00 & 00 & 00 & 01 & 00 & 01 & 01 & 00 & 01 & 00 & 00 & 00 & 01 & 01 & 01 & 00 & 01 & 00 & 01 & 01 & 01 & 01 & 01 & 00 & 00 \\ & 01 & & 00 & 01 & 00 & 00 & 00 & 01 & 00 & 01 & 01 & 01 & 00 & 00 & 00 & 01 & 00 & 00 & 00 & 00 & 00 & 00 & 01 & 00 & 00 & 01 & 01 & 00 & 01 & 01 & 01 & 01 & 01 & 01 & 01 & 00 & 00 & 01 \\ & & 01 & 00 & 00 & 01 & 01 & 00 & 00 & 00 & 01 & 00 & 00 & 01 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 01 & 01 & 00 & 00 & 00 & 01 & 01 & 01 & 00 & 01 & 01 & 00 & 00 & 00 & 00 & 00 & 01 & 01 \end{bmatrix} \quad m$$

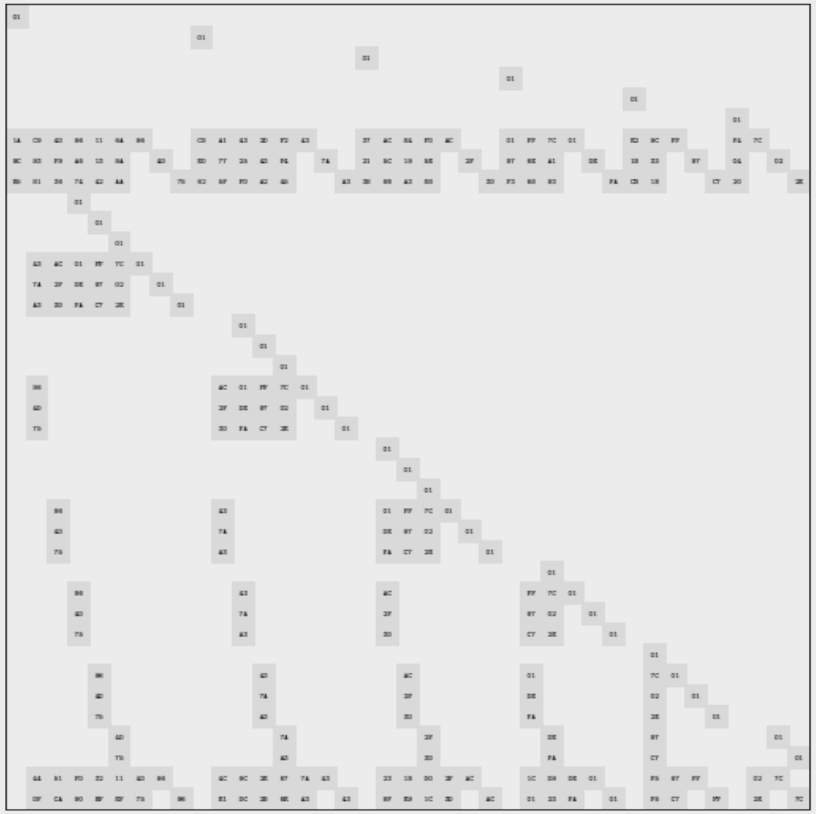
$$D$$

0/1 UOV Key Gen – Calculating A

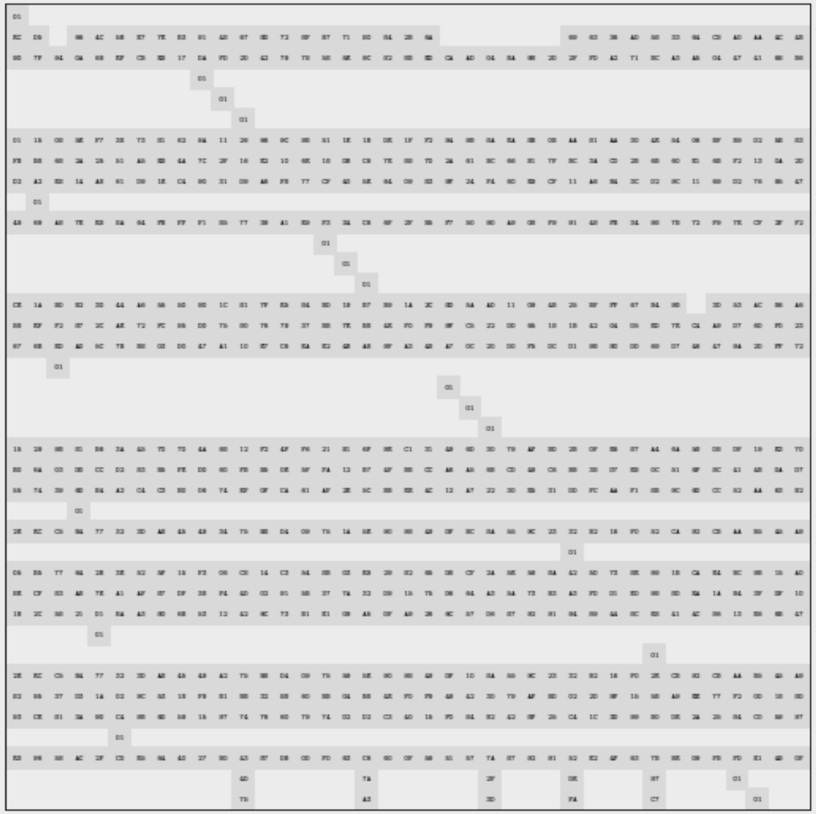


0/1 UOV Key Gen – Inverting A

$A' =$



$A'^{-1} =$



0/1 UOV Key Gen – Calculating F and P

$$\mathcal{F} = B \cdot A'^{-1}$$

$$f_1, f_2, f_3 = \left[\begin{array}{cccccc|ccc} 01 & 3D & 26 & 00 & 01 & 00 & A8 & 8D & E5 \\ & 00 & B1 & 00 & 00 & 01 & B4 & B6 & CB \\ & & 00 & 00 & 01 & 00 & 0E & C8 & 3B \\ & & & 01 & 3E & 01 & 33 & 09 & DB \\ & & & & 00 & 00 & 05 & B0 & C1 \\ & & & & & 00 & 4E & A5 & 5B \end{array}\right] \quad \left[\begin{array}{cccccc|ccc} 00 & 27 & 57 & 00 & 01 & 01 & E7 & 1D & C1 \\ & 01 & 5F & 00 & 01 & 00 & 3D & 51 & DD \\ & & 00 & 00 & 00 & 01 & FD & 66 & 04 \\ & & & 00 & 82 & 01 & FD & 6C & 8F \\ & & & & 01 & 01 & 2A & BA & C8 \\ & & & & & 00 & 6E & 2A & 75 \end{array}\right] \quad \left[\begin{array}{cccccc|ccc} 00 & 7E & 8F & 00 & 01 & 00 & 47 & 76 & 66 \\ & 00 & 94 & 00 & 00 & 00 & 53 & D2 & 4D \\ & & 01 & 01 & 01 & 00 & CE & 90 & EF \\ & & & 00 & A4 & 01 & FE & 00 & DF \\ & & & & 00 & 01 & E2 & D2 & 47 \\ & & & & & 01 & 60 & 49 & 3D \end{array}\right]$$

$$\mathcal{P} = \mathcal{F} \cdot A$$

$$p_1, p_2, p_3 = \left[\begin{array}{cccccc|ccc} 01 & 3D & 26 & 00 & 01 & 00 & 00 & 01 & 00 \\ & 00 & B1 & 00 & 00 & 01 & 00 & 01 & 01 \\ & & 00 & 00 & 01 & 00 & 00 & 00 & 01 \\ & & & 01 & 3E & 01 & 01 & 00 & 01 \\ & & & & 00 & 00 & 01 & 01 & 01 \\ & & & & & 00 & DC & 01 & 01 \\ \hline & & & & & & 00 & 00 & 00 \\ & & & & & & & 01 & AE \\ & & & & & & & & 00 \end{array}\right] \quad \left[\begin{array}{cccccc|ccc} 00 & 27 & 57 & 00 & 01 & 01 & 01 & 00 & 00 \\ & 01 & 5F & 00 & 01 & 00 & 00 & 00 & 00 \\ & & 00 & 00 & 00 & 01 & 00 & 00 & 01 \\ & & & 00 & 82 & 01 & 00 & 01 & 01 \\ & & & & 01 & 01 & 01 & 01 & 01 \\ & & & & & 00 & AA & 01 & 00 \\ \hline & & & & & & 00 & 00 & 01 \\ & & & & & & & 00 & 4F \\ & & & & & & & & 01 \end{array}\right] \quad \left[\begin{array}{cccccc|ccc} 00 & 7E & 8F & 00 & 01 & 00 & 00 & 01 & 00 \\ & 00 & 94 & 00 & 00 & 00 & 00 & 00 & 00 \\ & & 01 & 01 & 01 & 00 & 00 & 00 & 01 \\ & & & 00 & A4 & 01 & 01 & 00 & 01 \\ & & & & 00 & 01 & 00 & 00 & 00 \\ & & & & & 01 & 6E & 00 & 00 \\ \hline & & & & & & 01 & 01 & 01 \\ & & & & & & & 00 & 45 \\ & & & & & & & & 00 \end{array}\right]$$