Rijndael

Joan Daemen
Proton World
Belgium

Vincent Rijmen
COSIC
Belgium

Vincent meets Joan

- Where: K.U.Leuven, research group COSIC
- When: Summer '93
- Why: Evaluation of propriety cipher
- What: successful cryptanalysis (under NDA :-(

The Mother of Rijndael: Square

- Need for a 128-bit block cipher with 128-bit keys
  - 56-bit DES key: exhaustive key search feasible
  - 64-bit DES block (and Triple-DES): MAC weaknesses
- Summer-Fall ‘96: Design
  - symmetrical parallel structure
  - byte-oriented
  - no arithmetic operations
- Spring ‘97: Publication
  - Fast Software Encryption Workshop in Haifa, Israel

Our AES Proposal: Rijndael

- Spring '97: early draft of AES call for proposals:
  - key and block lengths 128, 196 and 256 bits
  - we started to work on a Square variant satisfying this
- Summer '97: Official AES call
  - requirement of 192 and 256 bit block lengths removed
  - "would be infeasible to realize"
- June '98: AES submission deadline
  - We baptized our design **Rijndael** (**Rij**men **& Dae**men) and submitted it to NIST

AES Selection Process

- August 98': AES 1 in Ventura (CA)
  - 15 proposals were presented
- Square had made school
  - Rijndael: *son* of Square
  - Crypton (Korea) has the *Square structure*
  - Twofish (Counterpane) uses Square features
- August '99: Announcement of the five finalists
- October 2000: Rijndael announced as AES

What makes Rijndael stand out?

- The symmetric and parallel structure
  - gives implementers a lot of flexibility
  - has not allowed effective cryptanalytic attacks
- Well adapted to modern processors
  - Pentium
  - RISC and parallel processors
- Suited for Smart cards
- Flexible in dedicated hardware

→ Let’s have a look at what’s inside!

Rijndael: what is inside?

- Key and State bytes arranged in rectangular arrays

$k_{0,0}$	$k_{0,1}$	$k_{0,2}$	$k_{0,3}$	$k_{0,4}$	$k_{0,5}$	$k_{0,6}$	$k_{0,7}$
$k_{1,0}$	$k_{1,1}$	$k_{1,2}$	$k_{1,3}$	$k_{1,4}$	$k_{1,5}$	$k_{1,6}$	$k_{1,7}$
$k_{2,0}$	$k_{2,1}$	$k_{2,2}$	$k_{2,3}$	$k_{2,4}$	$k_{2,5}$	$k_{2,6}$	$k_{2,7}$
$k_{3,0}$	$k_{3,1}$	$k_{3,2}$	$k_{3,3}$	$k_{3,4}$	$k_{3,5}$	$k_{3,6}$	$k_{3,7}$

Variable Key size:

16, 24 or 32 bytes

Variable Block size:

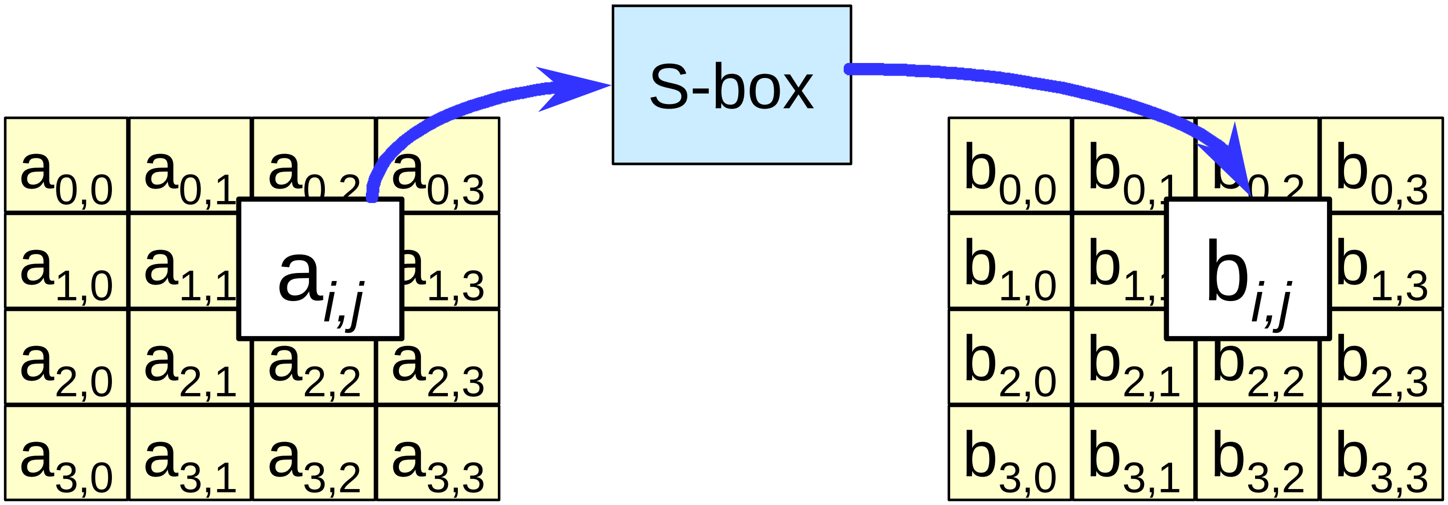
16, 24 or 32 bytes

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$	$a_{0,4}$	$a_{0,5}$	$a_{0,6}$	$a_{0,7}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$	$a_{1,4}$	$a_{1,5}$	$a_{1,6}$	$a_{1,7}$
$a_{2,0}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$	$a_{2,4}$	$a_{2,5}$	$a_{2,6}$	$a_{2,7}$
$a_{3,0}$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$	$a_{3,4}$	$a_{3,5}$	$a_{3,6}$	$a_{3,7}$

Rijndael: Iterated Block Cipher

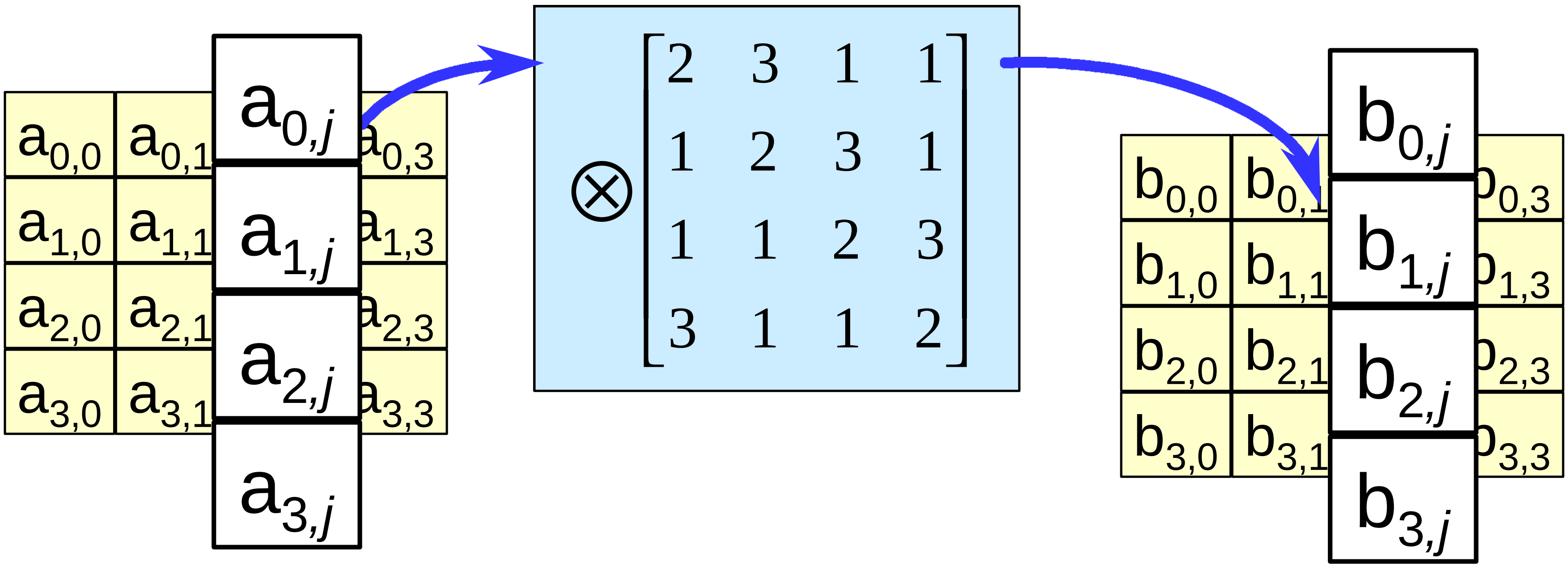
- 10/12/14 times applying the same round function
- Round function: uniform and parallel, composed of 4 steps
- Each step has its own particular function:
  - ByteSub: nonlinearity
  - ShiftRow: inter-column diffusion
  - MixColumn: inter-byte diffusion within columns
  - Round key addition

Round step 1: ByteSub



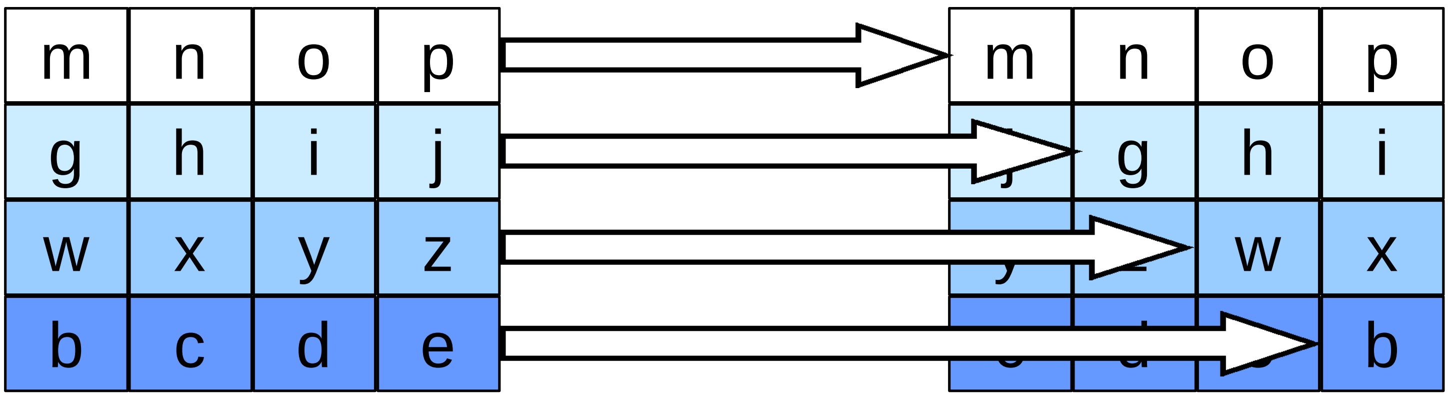
- Bytes are transformed by applying invertible S-box.
- One single S-box for the complete cipher
- High non-linearity

Round step 2: MixColumn



- Bytes in columns are linearly combined
- High intra-column diffusion:
  - based on theory of error-correcting codes

Round step 3: ShiftRow



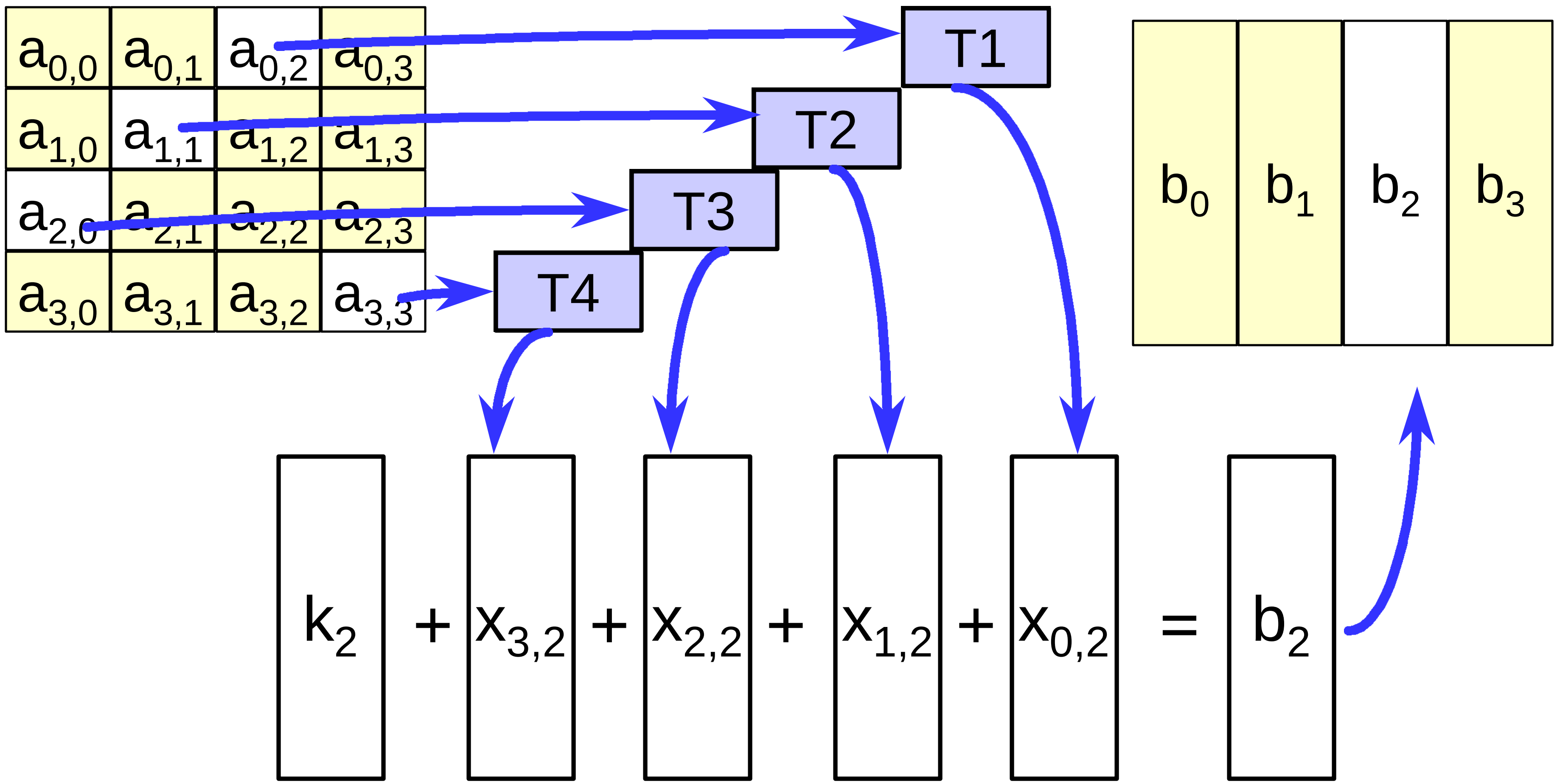
- Rows are shifted over 4 different offsets
- High diffusion over multiple rounds:
  - Interaction with MixColumn

Round step 4: Key addition

$$\begin{array}{|c|c|c|c|}\hline a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ \hline a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} \\ \hline a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} \\ \hline a_{3,0} & a_{3,1} & a_{3,2} & a_{3,3} \\ \hline \end{array} + \begin{array}{|c|c|c|c|}\hline k_{0,0} & k_{0,1} & k_{0,2} & k_{0,3} \\ \hline k_{1,0} & k_{1,1} & k_{1,2} & k_{1,3} \\ \hline k_{2,0} & k_{2,1} & k_{2,2} & k_{2,3} \\ \hline k_{3,0} & k_{3,1} & k_{3,2} & k_{3,3} \\ \hline \end{array} = \begin{array}{|c|c|c|c|}\hline b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} \\ \hline b_{1,0} & b_{1,1} & b_{1,2} & b_{1,3} \\ \hline b_{2,0} & b_{2,1} & b_{2,2} & b_{2,3} \\ \hline b_{3,0} & b_{3,1} & b_{3,2} & b_{3,3} \\ \hline \end{array}$$

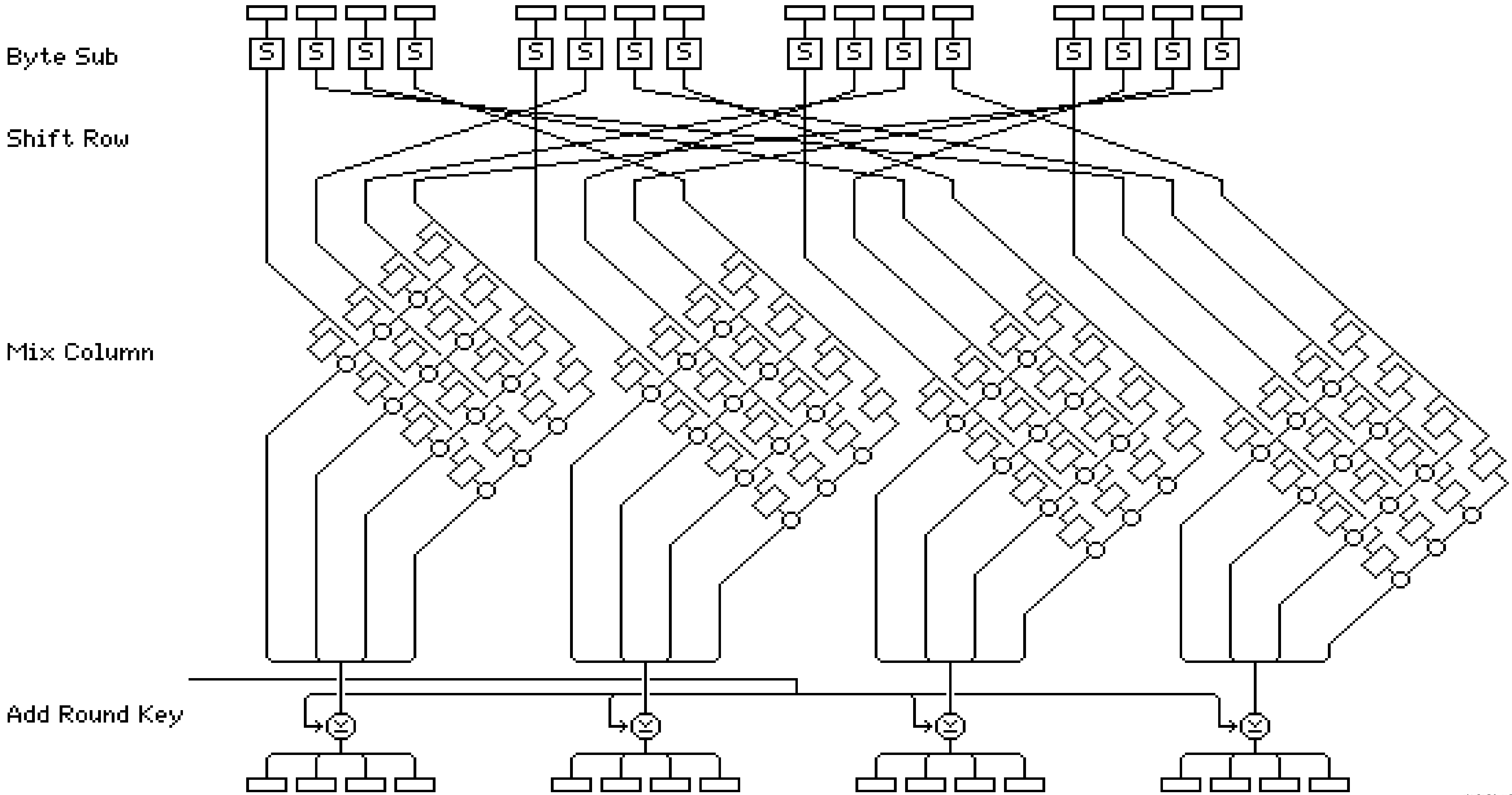
- Makes round function key-dependent
- Computation of round keys: “*keep it simple*”
  - small number of operations
  - small amount of memory

Rijndael on Modern Processors



Round function: just 16 table-lookups and EXORS

Rijndael in Hardware



Future of AES/Rijndael

- AES
  - US Government Administration
  - IPSEC
  - commercial file encryption products
  - Banking (DIGIPASS, ...)
  - ...
- Rijndael
  - UMTS
  - Windows
  - ...

We like to thank

- NIST: for the open way in which the AES process was conducted
- Rijndael Programmers: for showing that it can be efficiently implemented
  - Antoon Bosselaers, Paulo Barretto, Cryptix, Brian Gladman, Geoffrey Keating, Helger Lipmaa, Kazumaro Aoki, Mitsuru Matsui, a.m.o.
- Anyone who motivated us by expressing their interest in our work