Improved Key Recovery Attacks on Reduced-Round AES with Practical Data and Memory Complexities

Achiya Bar-On
Nathan Keller

Orr Dunkelman

Eyal Ronen
Adi Shamir

אוניברסיטת בר-אילן
Bar-Ilan University

אוניברסיטת חיפה
University of Haifa
جامعة حيفا

מכון ויצמן למדע
WEIZMANN INSTITUTE OF SCIENCE

AES

- AES is the best known and most widely used secret key cryptosystem
  - Almost all secure connections on the Internet use AES

- Its security had been analyzed for more than 20 years

- AES has either 10, 12, or 14 rounds depending on the key size (128, 192, 256 bits)

- To date there is no known attack on full AES which is significantly faster than exhaustive search

Analyzing reduced round AES

- Interesting as a platform for analyzing the remaining security margins

- Several Light Weight Cryptosystems and Hash functions use 4 or 5 rounds AES as a building block
  - 4-Round AES: ZORRO, LED and AEZ
  - 5-Round AES: WEM, Hound and ELmD

Analyzing reduced round AES

- There are 3 relevant parameters:
  Time (T), Memory (M) and Data (D)

- To combine these 3 complexity measures it is common to summarize them as a single number max(T,M,D) defined as their Total Complexity

Best attacks on 5 round AES

- Only a few techniques led to successful attacks against 5-round AES

Technique	Complexity Max(T, D, M)	Year
Square	2^{32}	2000
Imp. Differential	2^{32}	2001
Yoyo	2^{32}	2017

Recent attacks on 5 rounds AES

- In 2017 a new technique (the multiple-of-8 attack [GRR, EC'17]) was proposed, and in 2018 Grassi applied a special version of it (the mixture-differentials attack) to 5 round AES
- However, its complexity was not better than previous attacks
- In this work we improve the 20 year old record to 2^{22}

Recent attacks on 5 rounds AES

- In 2017 a new technique (the multiple-of-8 attack [GRR, EC'17]) was proposed, and in 2018 Grassi had applied a special version of it (the mixture-differentials attack) to 5 round AES

- However, its complexity was not better than previous attacks

Best attacks on 5 round AES - updated

Technique	**Complexity Max(T, D, M)**	**Year**
Square	2^{32}	2000
Imp. Differential	2^{32}	2001
Yoyo	2^{32}	2017
Grassi	2^{32}	2018

Our new result

- Breaking the 20 years old 2^{32} barrier by a factor of 1000:

Technique	Complexity Max(T, D, M)	Year
Square	2^{32}	2000
Imp. Differential	2^{32}	2001
Yoyo	2^{32}	2017
Grassi	2^{32}	2018
Our new result	2^{22}	2018

AES structure

- 10, 12, or 14 rounds, where each round of AES consists of:

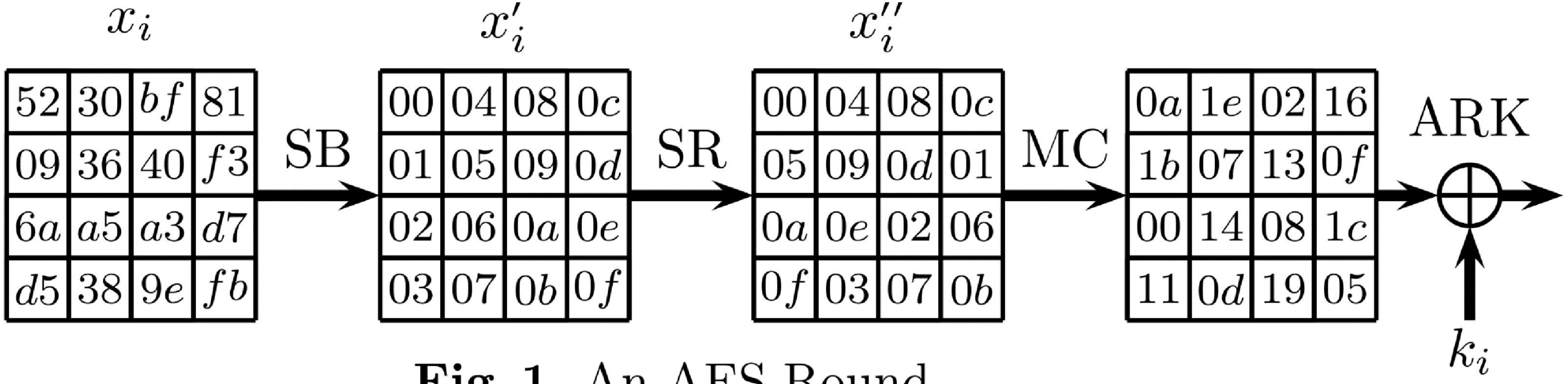
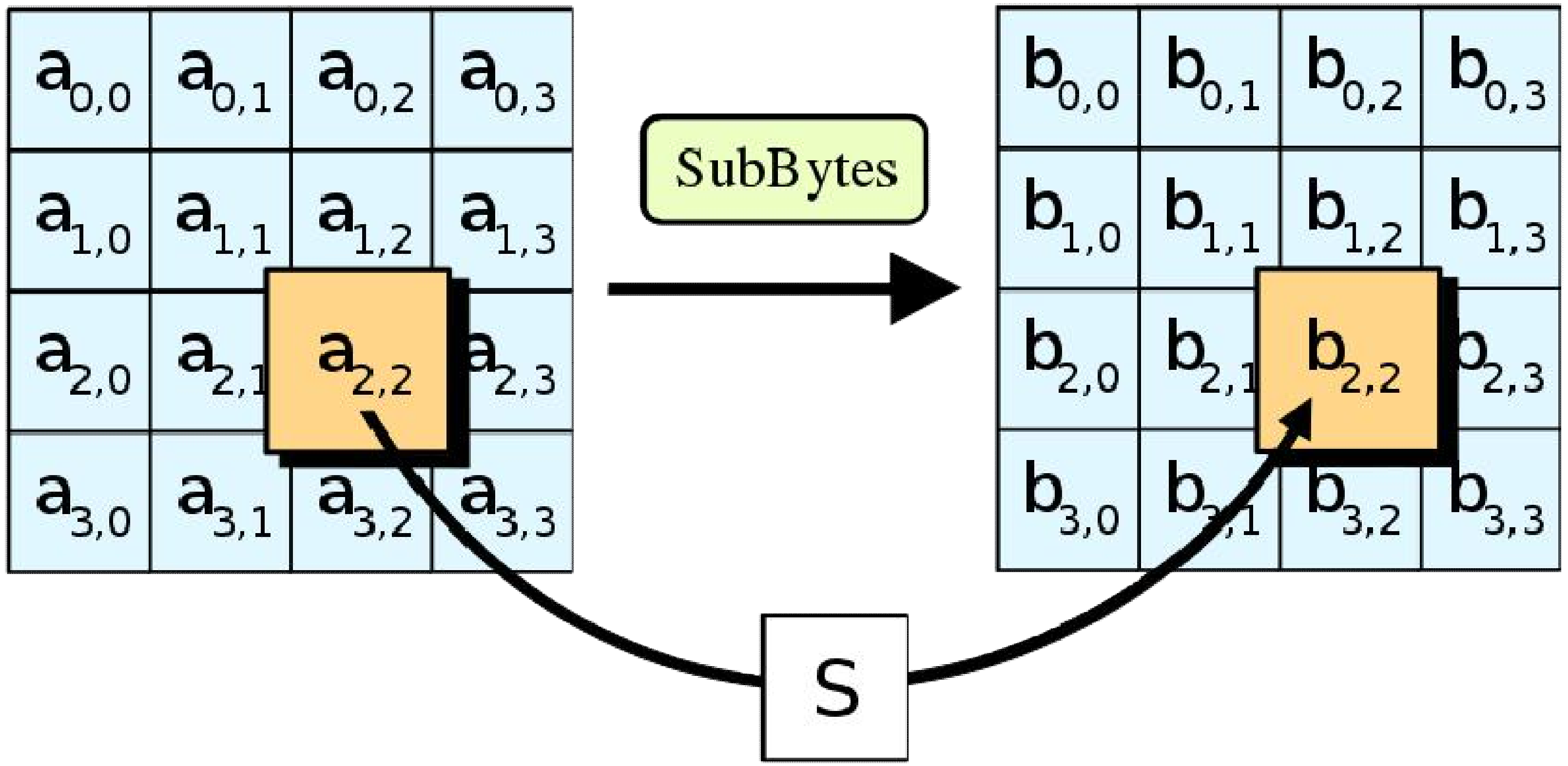


Fig. 1. An AES Round

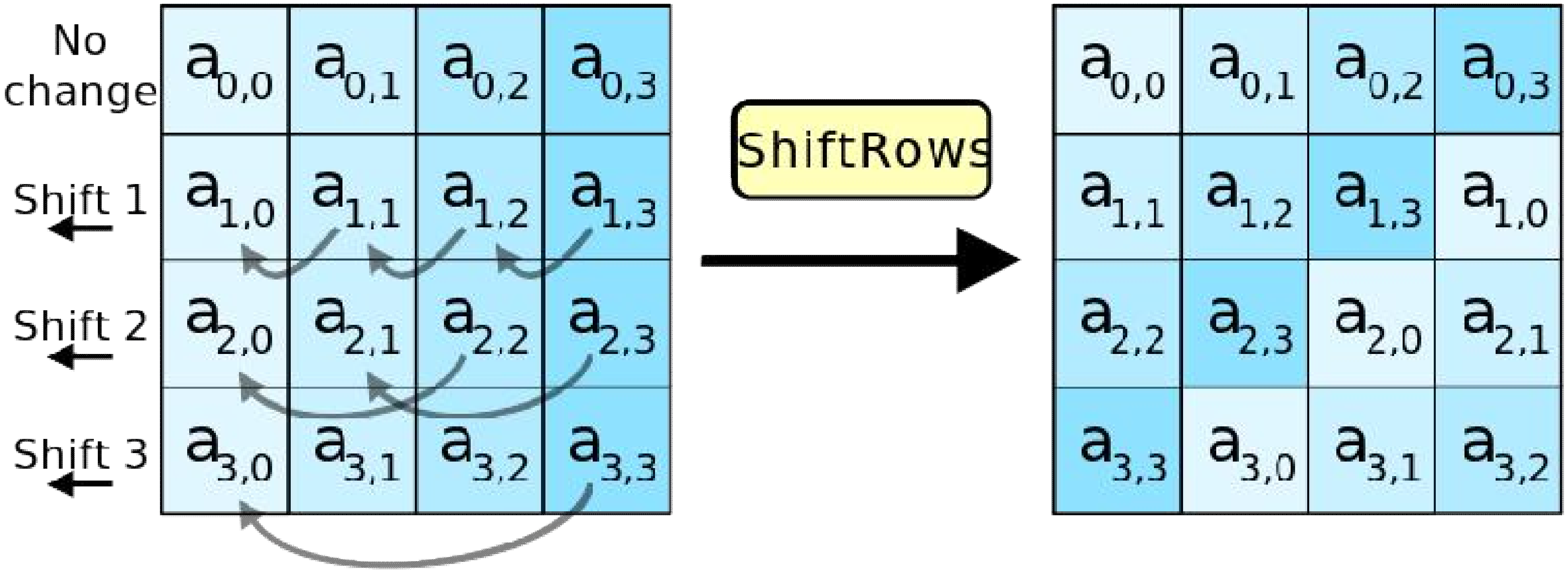
- Extra ARK operation before the first round
- No Mix Column in the last round

SB – SubBytes Operation



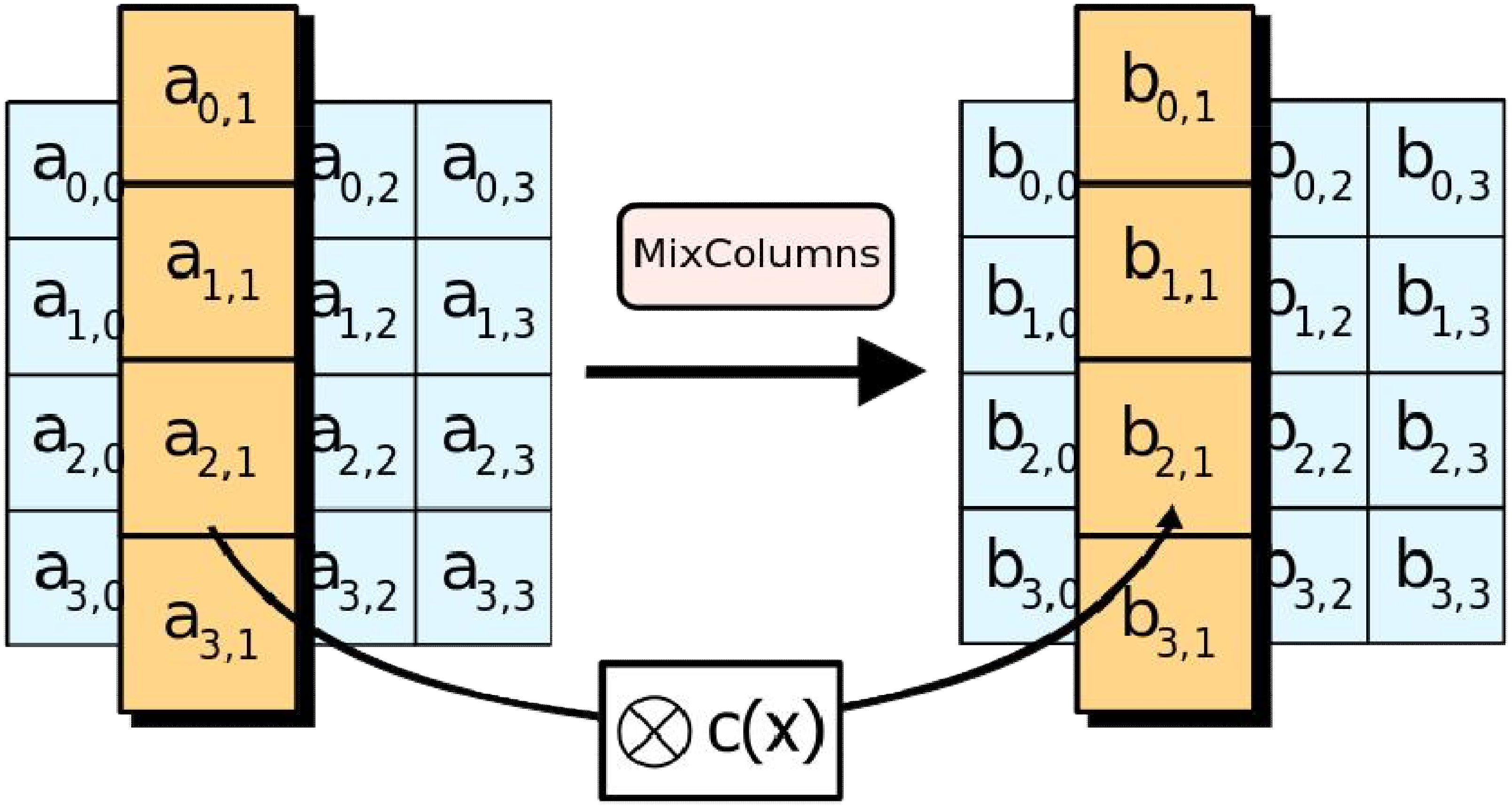
By User:Matt Crypto - Own work, Public Domain,
https://commons.wikimedia.org/w/index.php?curid=1118913

SR – ShiftRows Operation



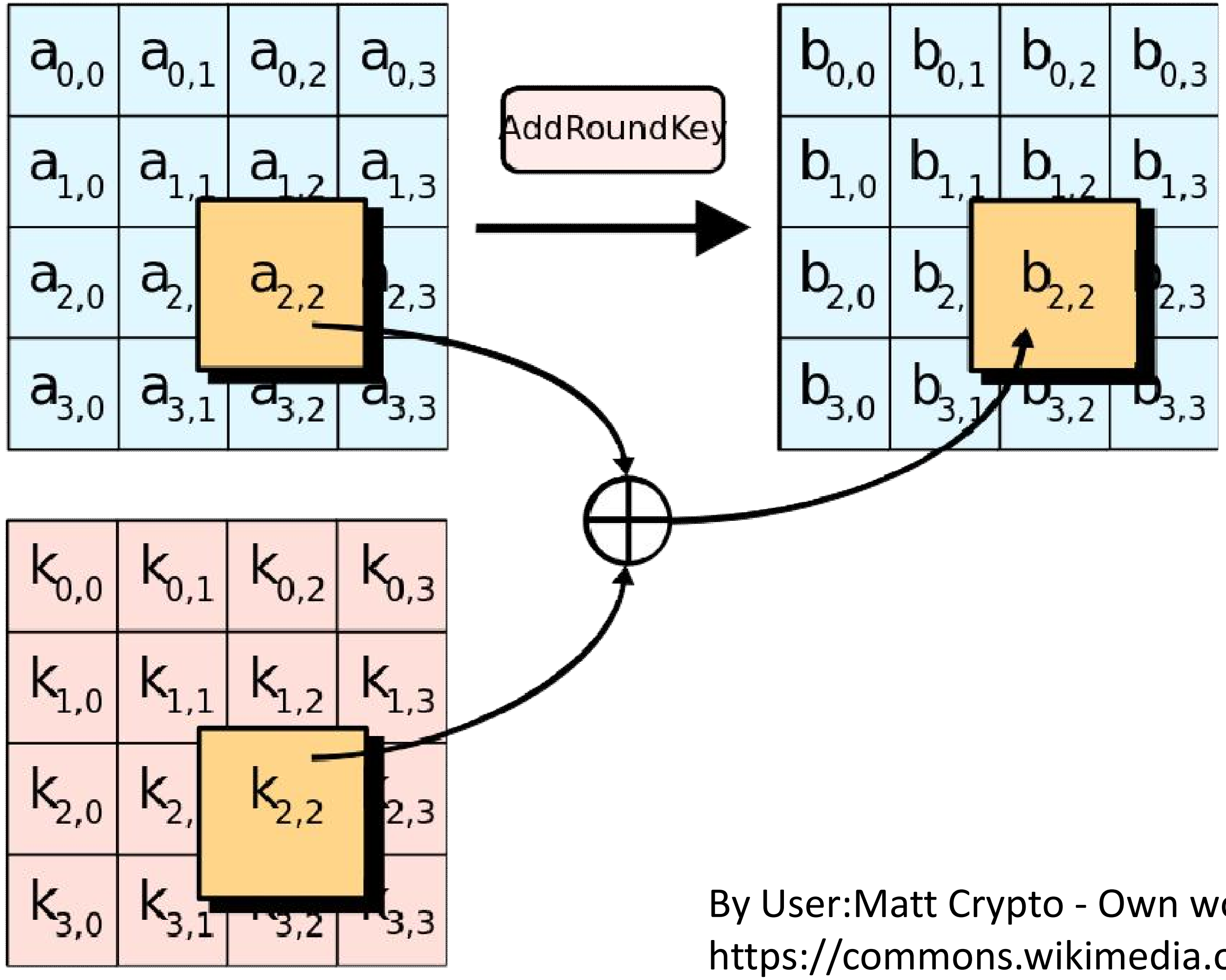
By User:Matt Crypto - Own work, Public Domain,
https://commons.wikimedia.org/w/index.php?curid=1118782

MC – MixColumn Operation



By User:Matt Crypto - Own work, Public Domain,
https://commons.wikimedia.org/w/index.php?curid=1118874

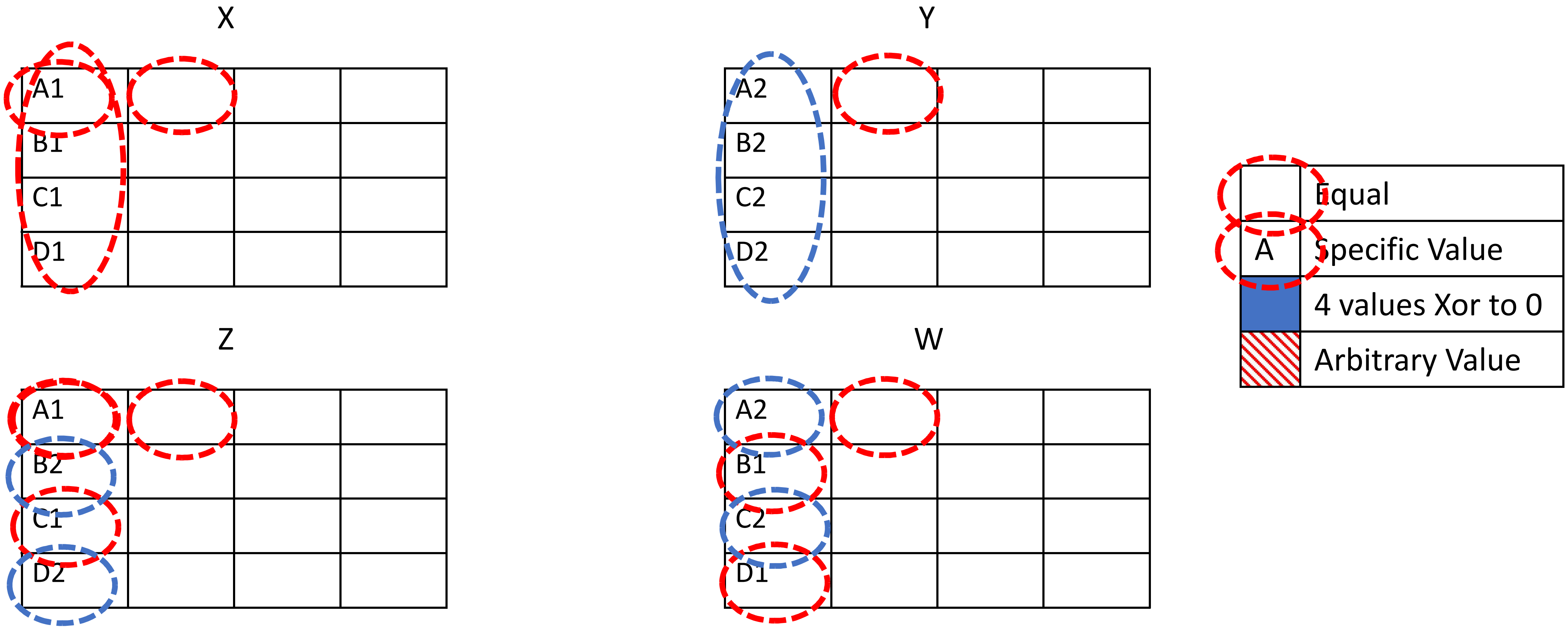
ARK – Add Round Key Operation



By User:Matt Crypto - Own work, Public Domain,
https://commons.wikimedia.org/w/index.php?curid=1118831

The notation of mixtures (Grassi et. al 2017)

- What is a mixture of an AES state pair (x,y)?



The evolution of mixtures under AES

- Consider the following 4 inputs to round i

X

A1			
B1			
C1			
D1			

Y

A2			
B2			
C2			
D2			

Z

A1			
B2			
C1			
D2			

W

A2			
B1			
C2			
D1			

	Equal
A	Specific Value
	4 values Xor to 0
	Arbitrary Value

The evolution of mixtures under AES

- Round i after Sub Byte

X

A1*			
B1*			
C1*			
D1*			

Y

A2*			
B2*			
C2*			
D2*			

Z

A1*			
B2*			
C1*			
D2*			

W

A2*			
B1*			
C2*			
D1*			

	Equal
A	Specific Value
	4 values Xor to 0
	Arbitrary Value

The evolution of mixtures under AES

- Round i after Shift Rows

X

A1*			
			B1*
		C1*	
	D1*		

Y

A2*			
			B2*
		C2*	
	D2*		

Z

A1*			
			B2*
		C1*	
	D2*		

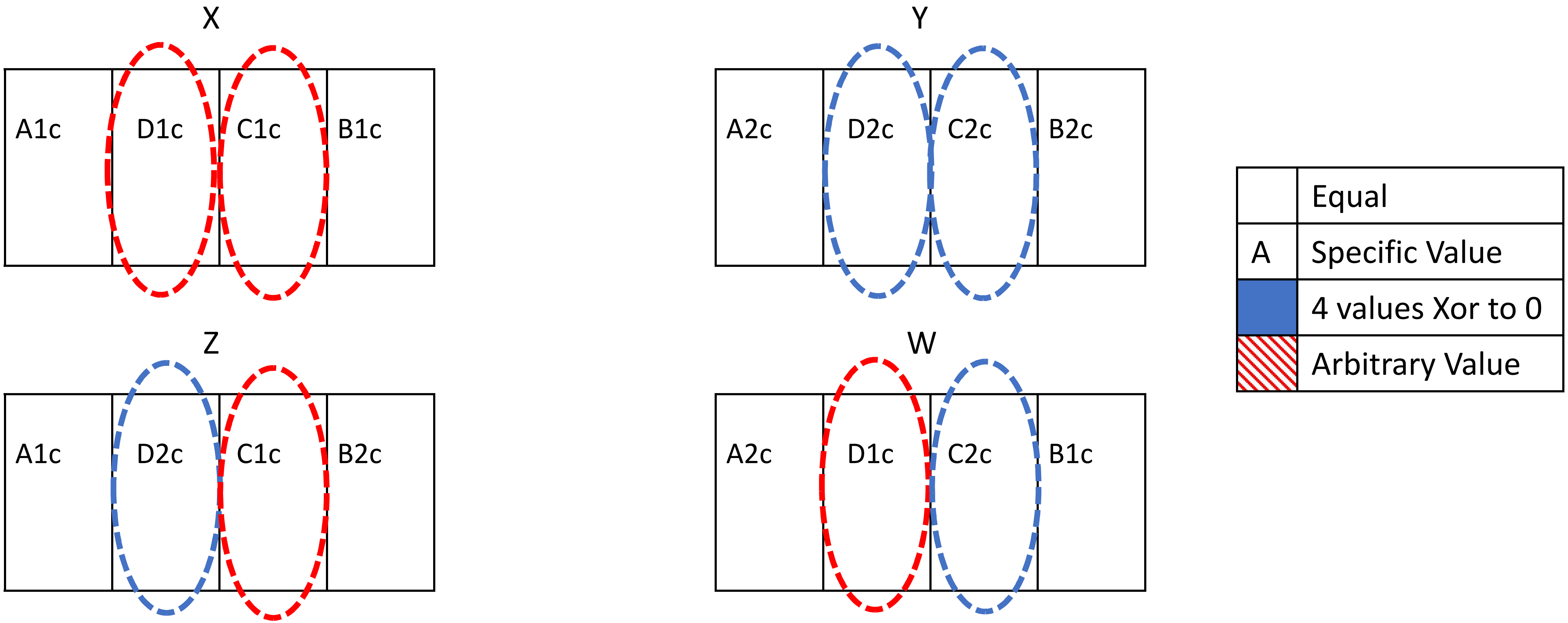
W

A2*			
			B1*
		C2*	
	D1*		

	Equal
A	Specific Value
(blue)	4 values Xor to 0
(red hatched)	Arbitrary Value

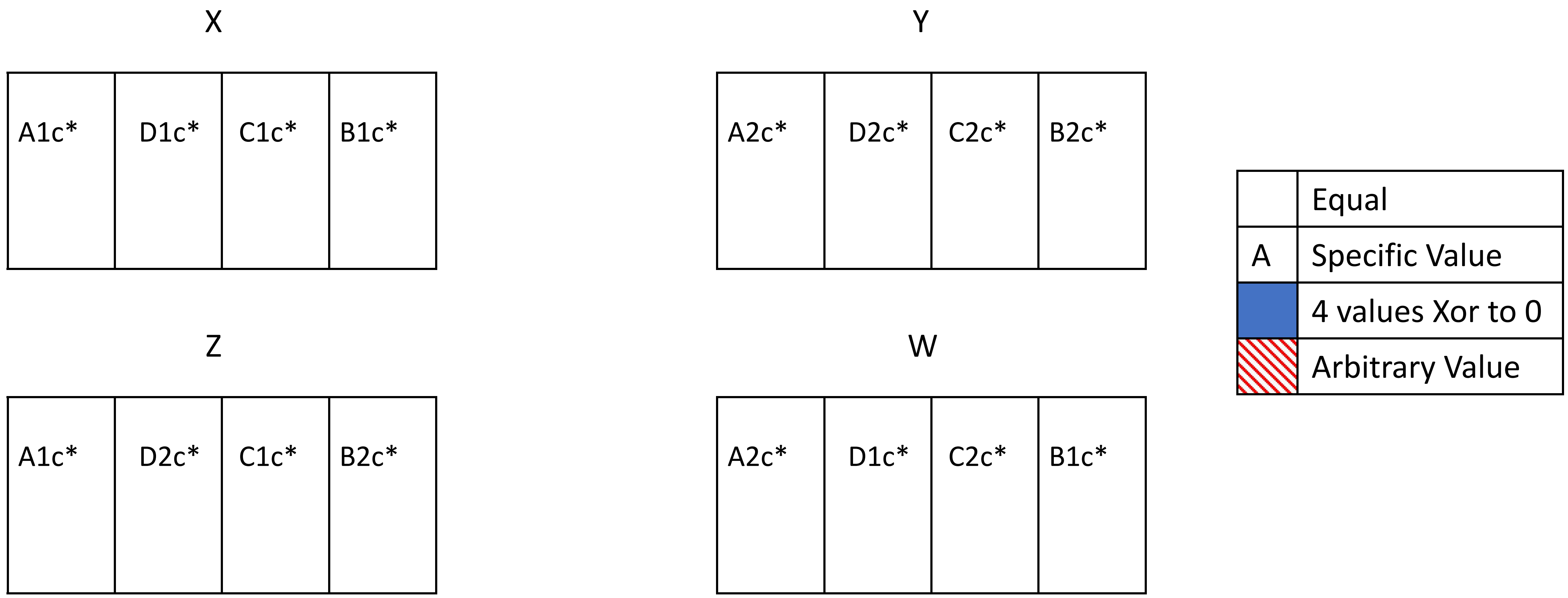
The evolution of mixtures under AES

- Round i after Mix Column



The evolution of mixtures under AES

- Round i after Add Round Key



The evolution of mixtures under AES

- Input to round i+1

X

A1c*	D1c*	C1c*	B1c*

Y

A2c*	D2c*	C2c*	B2c*

Z

A1c*	D2c*	C1c*	B2c*

W

A2c*	D1c*	C2c*	B1c*

	Equal
A	Specific Value
	4 values Xor to 0
	Arbitrary Value

The evolution of mixtures under AES

- Round i+1 after Sub Byte

X

A1c’	D1c’	C1c’	B1c’

Y

A2c’	D2c’	C2c’	B2c’

Z

A1c’	D2c’	C1c’	B2c’

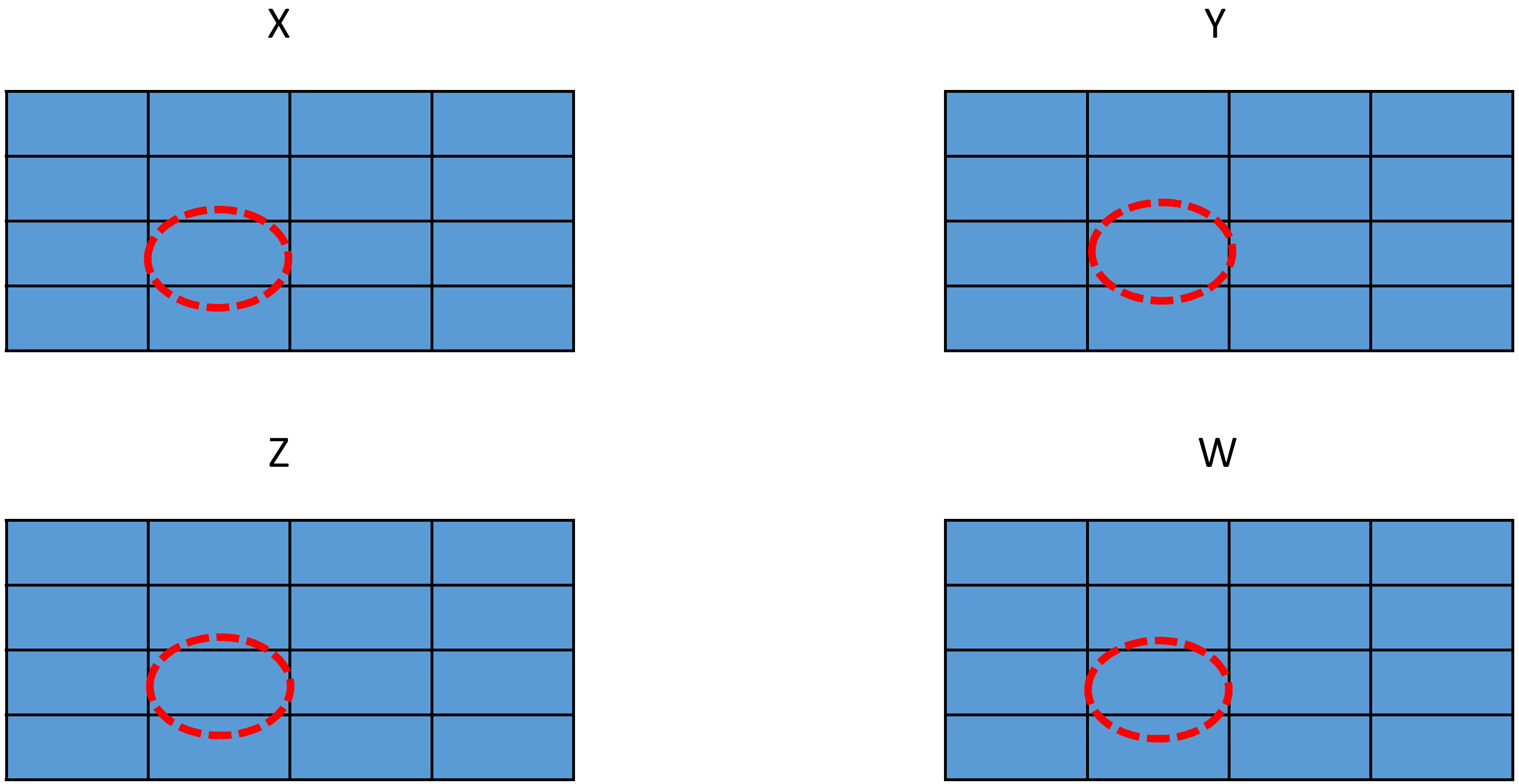
W

A2c’	D1c’	C2c’	B1c’

	Equal
A	Specific Value
	4 values Xor to 0
	Arbitrary Value

The evolution of mixtures under AES

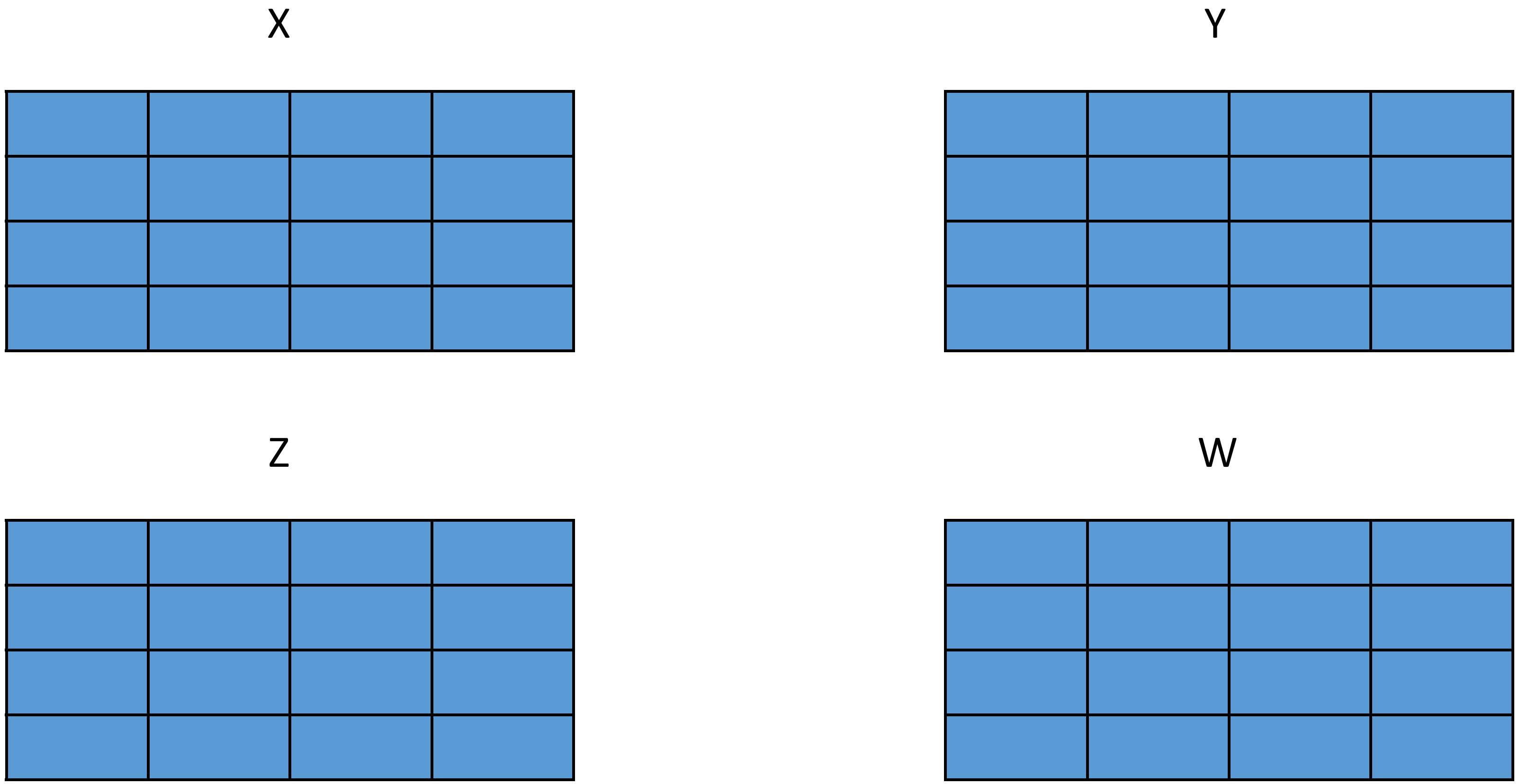
- Implies weaker property in round i+1 after Sub Byte



	Equal
A	Specific Value
	4 values Xor to 0
	Arbitrary Value

The evolution of mixtures under AES

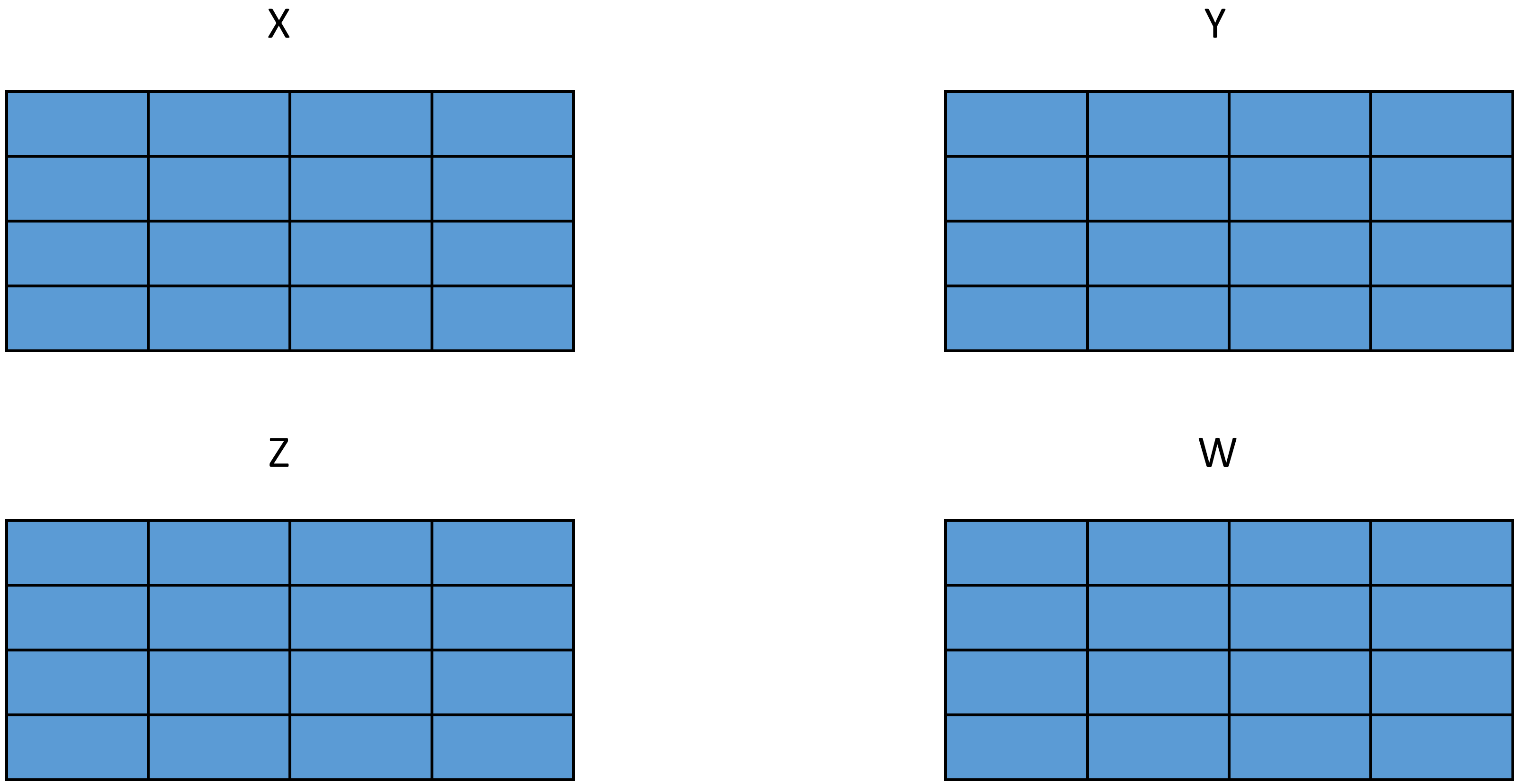
- Round i+1 after Shift Row, Mix Column and ARK



	Equal
A	Specific Value
	4 values Xor to 0
	Arbitrary Value

The evolution of mixtures under AES

- Input to round i+2

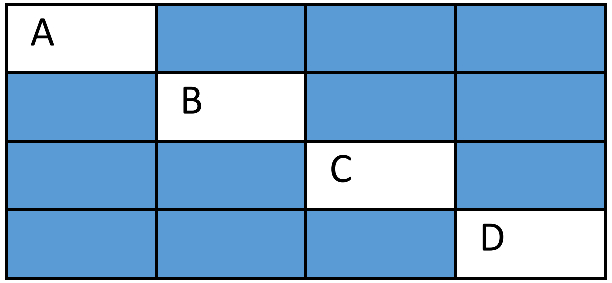


	Equal
A	Specific Value
	4 values Xor to 0
	Arbitrary Value

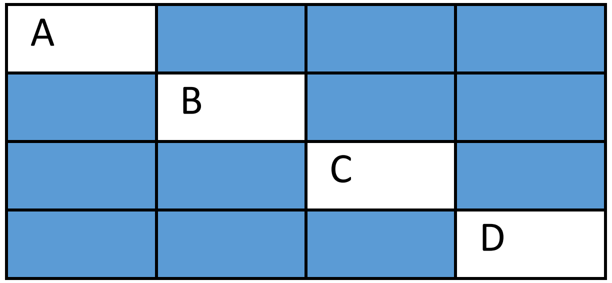
Extending this property to 4 rounds

- Assume states (X,Y) are equal in one of their diagonals

X

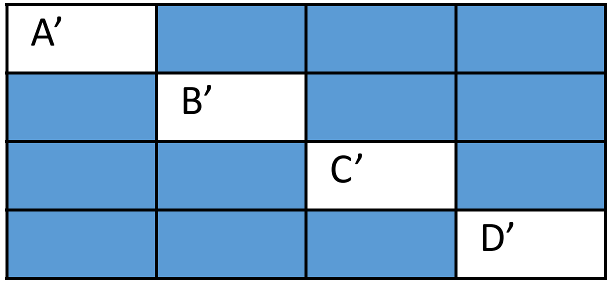


Y

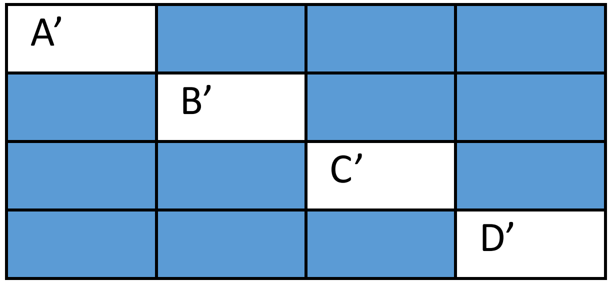


- Then:

Z



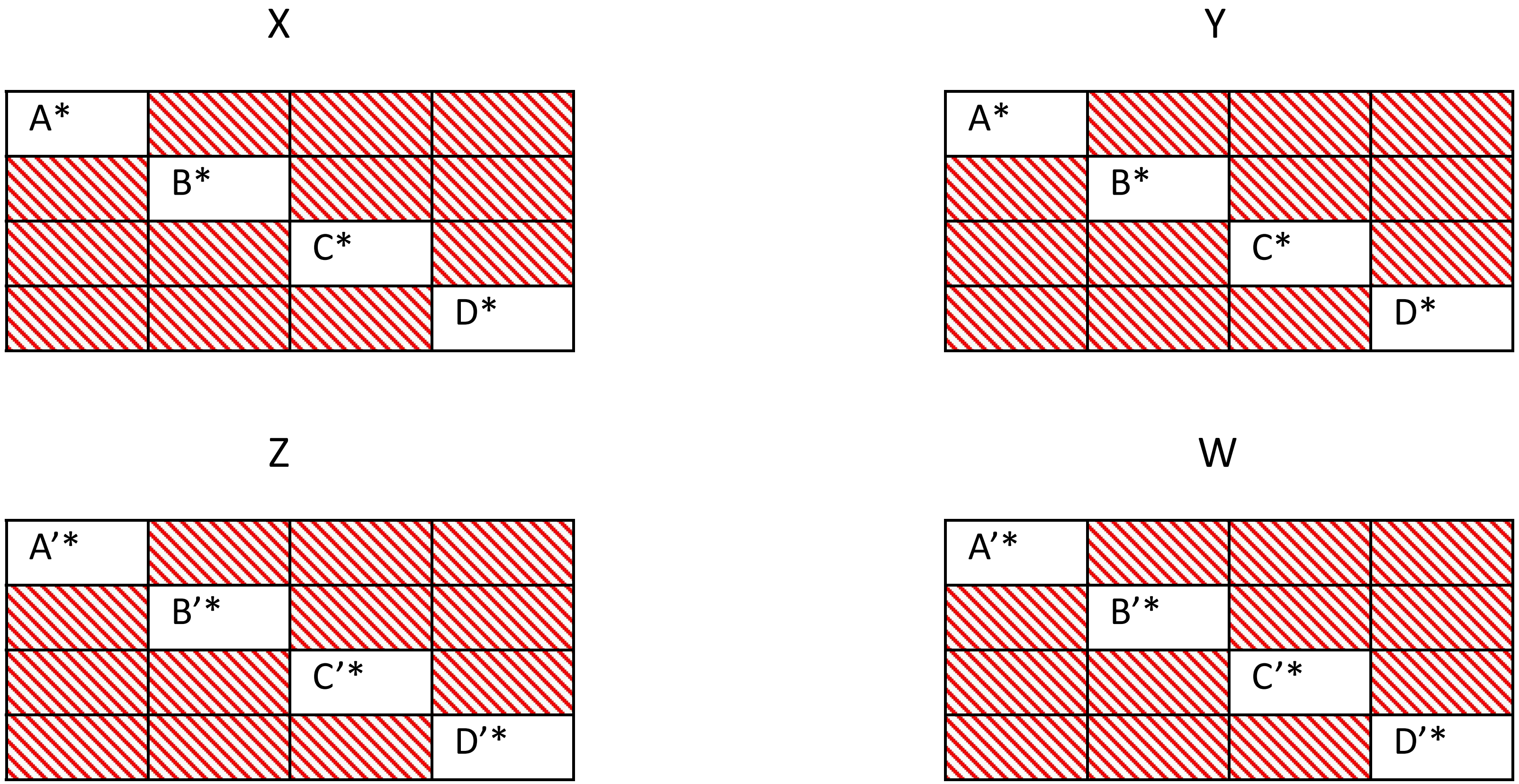
W



	Equal
A	Specific Value
	4 values Xor to 0
	Arbitrary Value

Extending this property to 4 rounds

- Round i+2 after Sub Byte



	Equal
A	Specific Value
	4 values Xor to 0
	Arbitrary Value

Extending this property to 4 rounds

- Round i+2 after Shift rows

X

A*			
B*			
C*			
D*			

Y

A*			
B*			
C*			
D*			

Z

A'*			
B'*			
C'*			
D'*			

W

A'*			
B'*			
C'*			
D'*			

	Equal
A	Specific Value
	4 values Xor to 0
	Arbitrary Value

Extending this property to 4 rounds

- Round i+2 after Mix Column

X

A°			
B°			
C°			
D°			

Y

A°			
B°			
C°			
D°			

Z

A°'			
B°'			
C°'			
D°'			

W

A°'			
B°'			
C°'			
D°'			

	Equal
A	Specific Value
	4 values Xor to 0
	Arbitrary Value

Extending this property to 4 rounds

- Round i+2 after Add Round Key

X

A*			
B*			
C*			
D*			

Y

A*			
B*			
C*			
D*			

Z

A*’			
B*’			
C*’			
D*’			

W

A*’			
B*’			
C*’			
D*’			

	Equal
A	Specific Value
	4 values Xor to 0
	Arbitrary Value

Extending this property to 4 rounds

- Then in the input to round i+3 we get

X

A*			
B*			
C*			
D*			

Y

A*			
B*			
C*			
D*			

Z

A*’			
B*’			
C*’			
D*’			

W

A*’			
B*’			
C*’			
D*’			

	Equal
A	Specific Value
	4 values Xor to 0
	Arbitrary Value

Extending this property to 4 rounds

- Round i+3 after sub byte

X

A^			
B^			
C^			
D^			

Y

A^			
B^			
C^			
D^			

Z

A^'			
B^'			
C^'			
D^'			

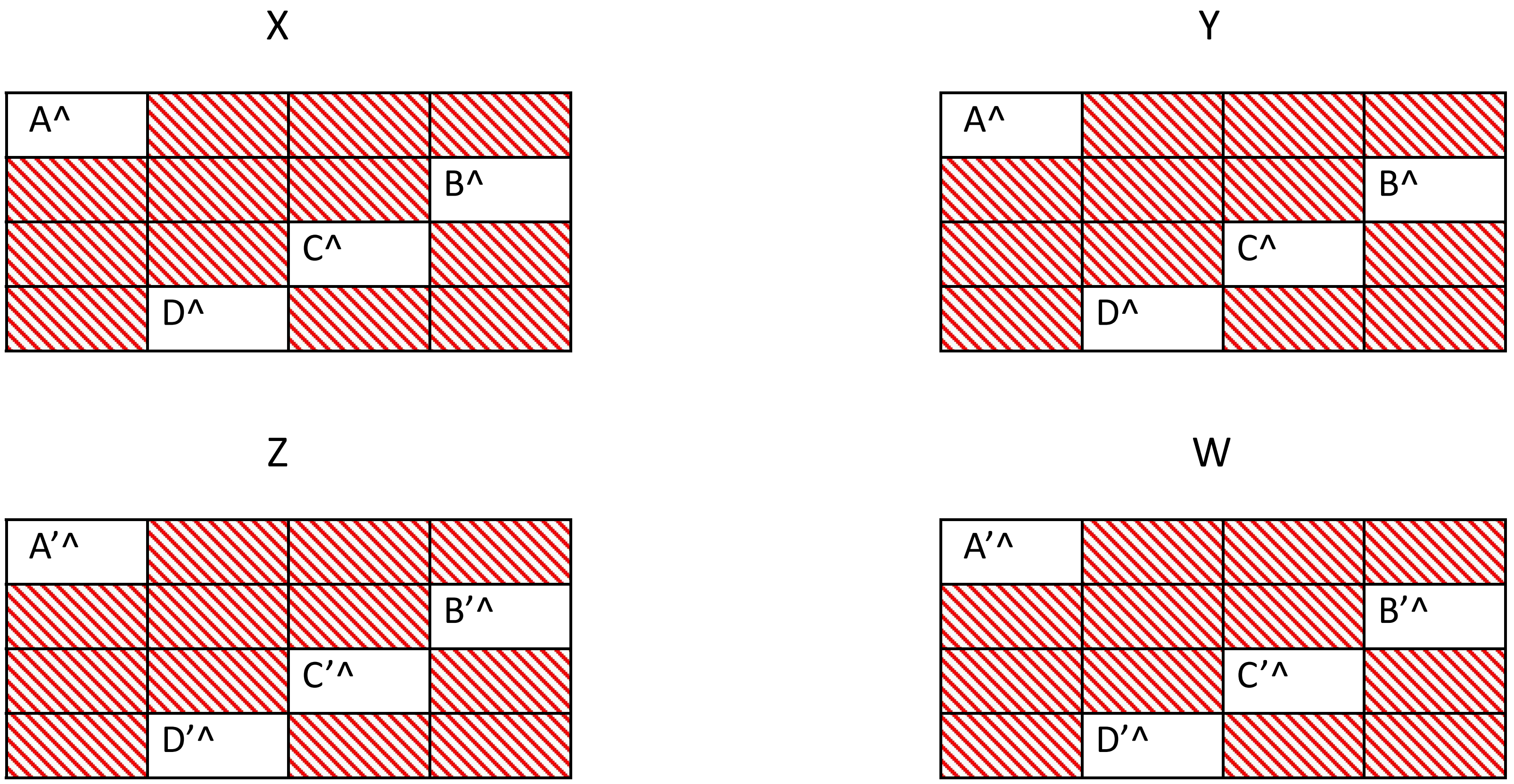
W

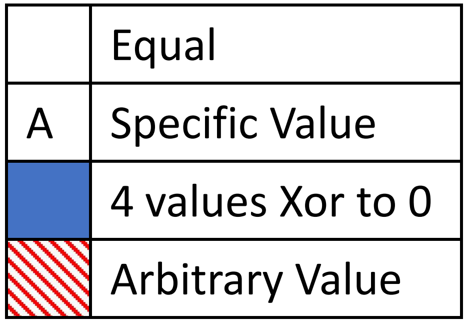
A^'			
B^'			
C^'			
D^'			

	Equal
A	Specific Value
	4 values Xor to 0
	Arbitrary Value

Extending this property to 4 rounds

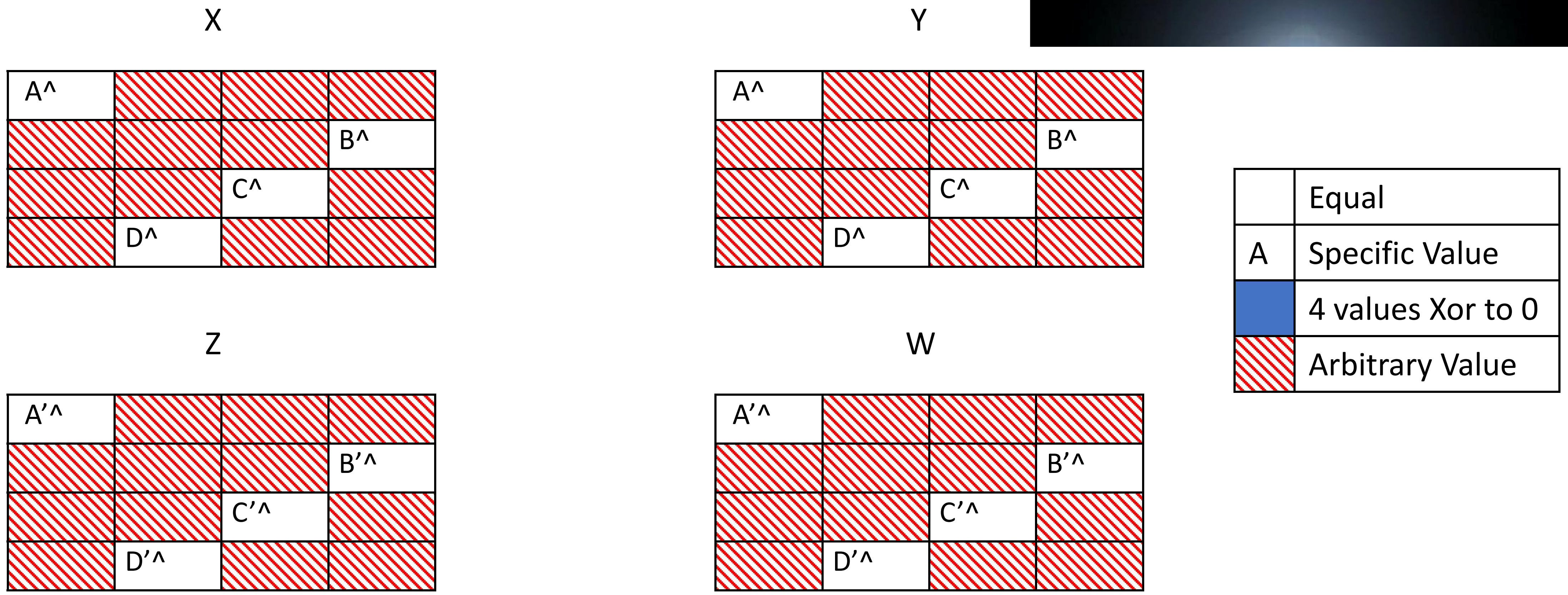
- Round i+3 after Shift Rows and before Mix Column





AES 4 Round Distinguisher

- Last round of AES has no Mix Column



A 5 Round AES Attack (Grassi 18)

- Precede the 4 round distinguisher with an extra round before it
- We encrypt all possible values of A,B,C,D

A			
	B		
		C	
			D

	Equal
A	Specific Value
	4 values Xor to 0
	Arbitrary Value

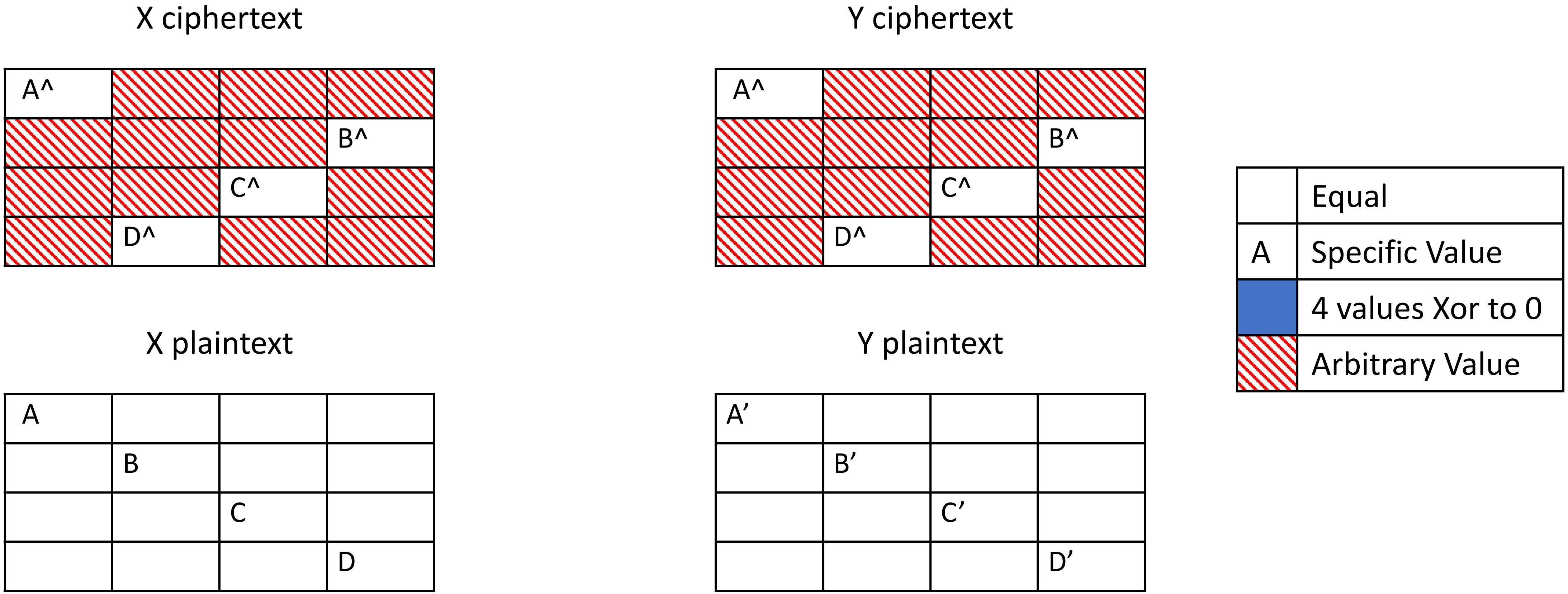
- Then as input to round 1 we get:

A’			
B’			
C’			
D’			

A’, B’, C’, and D’ is a permutation of A, B, C, D which depends only on 4 key bytes

A 5 Round AES Attack [Grassi 18]

- We look for a “good ciphertext pair”, and get the plaintext



A 5 Round AES Attack [Grassi 18]

- For all 2^{32} possible key bytes: partially encrypt (AKR, SB, SR, MC)

X partial round encryption

A*			
B*			
C*			
D*			

Y partial round encryption

A'*			
B'*			
C'*			
D'*			

X plaintext

A			
	B		
		C	
			D

Y plaintext

A'			
	B'		
		C'	
			D'

	Equal
A	Specific Value
	4 values Xor to 0
	Arbitrary Value

A 5 Round AES Attack [Grassi 18]

- Create a state mixture Z, W

X partial round encryption

A*			
B*			
C*			
D*			

Y partial round encryption

A'*			
B'*			
C'*			
D'*			

Z partial round encryption

A*			
B'*			
C*			
D'*			

W partial round encryption

A'*			
B*			
C'*			
D*			

	Equal
A	Specific Value
(blue)	4 values Xor to 0
(red hatched)	Arbitrary Value

A 5 Round AES Attack [Grassi 18]

- Partially decrypt Z and W

Z plaintext

A°			
	B°		
		C°	
			D°

W plaintext

A°’			
	B°’		
		C°’	
			D°’

Z partial round encryption

A*			
B’*			
C*			
D’*			

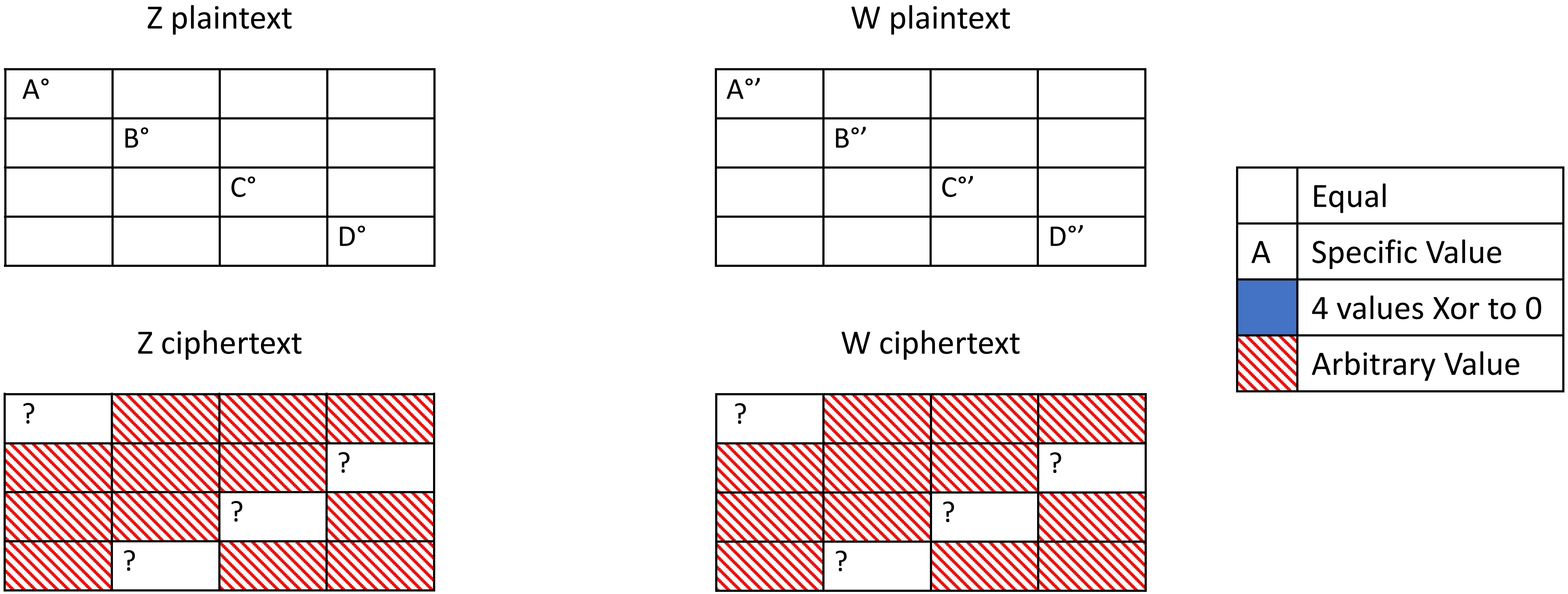
W partial round encryption

A’*			
B*			
C’*			
D*			

	Equal
A	Specific Value
	4 values Xor to 0
	Arbitrary Value

A 5 Round AES Attack [Grassi 18]

- Get Z and W ciphertexts, and check the equality condition



Our attack ideas

Attack	Complexity
Grassi’s original attack	$T=2^{32}$, $D=2^{32}$, $M=2^{32}$

Our attack ideas

Attack	Complexity
Grassi's original attack	$T=2^{32}$, $D=2^{32}$, $M=2^{32}$
Reduce data to get one "good mixture"	$T=2^{47}$, $D=2^{24}$, $M=2^{24}$

Our attack ideas

Attack	Complexity
Grassi's original attack	$T=2^{32}$, $D=2^{32}$, $M=2^{32}$
Reduce data to get one "good mixture"	$T=2^{47}$, $D=2^{24}$, $M=2^{24}$
Switch order to iterate over pairs	$T=2^{33}$, $D=2^{24}$, $M=2^{24}$

Our attack ideas

Attack	Complexity
Grassi's original attack	T=2^{32}, D=2^{32}, M=2^{32}
Reduce data to get one "good mixture"	T=2^{47}, D=2^{24}, M=2^{24}
Switch order to iterate over pairs	T=2^{33}, D=2^{24}, M=2^{24}
Use precomputed table	T=2^{29}, D=2^{24}, M=2^{24}

Our attack ideas

Attack	**Complexity**
Grassi's original attack	$T=2^{32}$, $D=2^{32}$, $M=2^{32}$
Reduce data to get one "good mixture"	$T=2^{47}$, $D=2^{24}$, $M=2^{24}$
Switch order to iterate over pairs	$T=2^{33}$, $D=2^{24}$, $M=2^{24}$
Use precomputed table	$T=2^{29}$, $D=2^{24}$, $M=2^{24}$
Smart selection of input structure	$T=2^{22}$, $D=2^{22}$, $M=2^{22}$

Idea 1 - Reduce Data: The good

- There are many mixtures, but we only need one of them
- Grassi used 2^{32} data
  - 2^{32} encryptions -> 2^{63} pairs -> 2^{31} good pairs

- We use only 2^{24} data
  - 2^{24} encryptions -> 2^{47} pairs -> 2^{15} good pairs
  - For each key and mixture type:
    We have the mixture in our data with probability $(2^{24}/2^{32})^2 = 2^{-16}$
  - There are 2^{15} pairs and 7 mixture types:
    We have a good mixture with probability $1-(1-2^{-16})^{(7*2\char`^15)}$ ~0.97

Idea 1 - Reduce Data: The bad

- We can thus reduce the data complexity
- However, we need to go over all 2^{15} pairs
  - So now $T = 2^{32} * 2^{15} = 2^{47}$
- This is only a time \ data tradeoff:
  - We reduce the data by a factor of 2^8
  - While increasing the time by a factor of 2^{15}

Idea 2 – Switch Order: The good

- We can change the order of operations, iterating over all pairs of pairs:
  - If we have a mixture after ARK, SB , SR and MC operations:
    $$X_0'' \oplus Y_0'' \oplus Z_0'' \oplus W_0'' = 0$$
  - Holds for each byte separately, depending on a single key byte
    $$SB(X_{0,0} \oplus k_0) \oplus SB(Y_{0,0} \oplus k_0) \oplus SB(Z_{0,0} \oplus k_0) \oplus SB(W_{0,0} \oplus k_0) = 0$$
  - Can find a suggestion for each of the 4 key bytes independently
  - Take the 4 key bytes and check for mixture after 1 round

Idea 2 – Switch Order: The bad

- For each pair of pairs (quartet) we can get a 4 key bytes suggestion with $4*2^8$ S-Box applications
  - 2^{24} encryptions -> 2^{47} pairs -> 2^{15} “good pairs”
  - 2^{29} quartets * 4 * 2^8 S box = 2^{39} S-Box ~ 2^{33} encryptions

Idea 3 - Precomputed Table

- We can use an optimized precomputed table
- Consider quartet of bytes of the form (0, a, b, c)
  - For each quartet we find a k such as: $SB(k) \oplus SB(a \oplus k) \oplus SB(b \oplus k) \oplus SB(c \oplus k) = 0$
  - We get (0, a, b, c) by $(0, \mathrm{y} \oplus x, \mathrm{z} \oplus x, \mathrm{w} \oplus x)$
- We get a table of size 2^{24}
  - The order is irrelevant so we can arrange in increasing order: save a factor of 6 to get $\sim 2^{(21.4)}$
  - Precomputation can be optimize to use $\sim 2^{24}$ S Box applications

Idea 4 – Smart Input Structure

- So far we get data and memory 2^{24} and time 2^{29}
- We can use just $2^{22.25}$ data by a smarter choice of input

A			
	B		
		C	

- E.g., A and B can get all 2^8 values each, C gets $2^{6.25}$ possible values
- We get a boost of 2^8 to the mixture probability from 2^{-63} to 2^{-55}
- 3 possible mixtures instead of 7, so in total $3 * 2^{-55}$

Idea 4 – Smart Input Structure

- What is the probability of a mixture?
- $2^{22.25}$ encryptions -> $2^{43.5}$ pairs -> 2^{86} pairs of pairs
- Number of mixture $2^{86} * 3*2^{-55} = 3*2^{31}$
- With “decent” probability we will get at least one “good mixture”
- We use hash tables of the ciphertext to sort the pairs

- Only get 3 bytes of key for each diagonal
  - By applying the same technique on the other diagonals we can recover 13 key bytes and brute force the rest of the key

Our Observation 5

- Data \ Memory trade off
- We can check for zero diff also in SR(Col(1)) and SR(Col(2)) …

- We can check 4 diagonals
  - Increase probability of success by 4
  - Amount of quartets = date^4
  - Reduces the data only by 4^(1/4) = sqrt(2)
  - Increases the amount of memory by factor of 4

Experimental Verification of Our Attack

- We have experimentally verified our theoretic analysis
  - 4 possible amounts of data
  - 200 different keys for each amount
  - Calculated the partial and full key recovery probability

Amount Of Data	3 Byte recovery probability	Full Key recovery probability
2^{22}	0.5	0.031
$2^{22.25}$	0.715	0.187
$2^{22.5}$	0.935	0.715
2^{23}	1	1

Extending to 7 round AES

Technique	Rounds	Data	Memory	Time
Gilbert-Minier	7	2^32	2^80	2^144
Demirci-Selcuk	7	2^99	2^98	2^99
Demirci-Selcuk	7	2^32	>2^100	>2^100
Square	7 (192-bit)	2^36	2^36	2^155
Square	7 (256-bit)	2^36	2^36	2^171

Extending to 7 round AES

Technique	Rounds	Data	Memory	Time
Gilbert-Minier	7	2^{32}	2^{80}	2^{144}
Demirci-Selcuk	7	2^{99}	2^{98}	2^{99}
Demirci-Selcuk	7	2^{32}	$>2^{100}$	$>2^{100}$
Square	7 (192-bit)	2^{36}	2^{36}	2^{155}
Square	7 (256-bit)	2^{36}	2^{36}	2^{171}
Mixture (our)	7 (192-bit)	2^{27}	2^{32}	2^{152}
Mixture (our)	7 (192+256)	2^{27}	2^{40}	2^{144}

Summary and open questions

- We broke a 20 year old attack complexity barrier on 5 round AES, improving it by a factor of 1000
- We obtained an improved "practical data and memory" attack on 7 round AES
- Is it possible to extend our new attacks to larger versions of AES?
- Can our results be used to attack schemes which use reduced 4/5 round AES as a component?


so does anyone have any questions?
"
"
"
©hugh