Mars Attacks! Revisited.

Differential Attack 12 Rounds of the MARS Core and Defeating the Complex MARS Key Schedule

INDOCRYPT'11

Michael Gorski, Thomas Knapke, Eik List,
Stefan Lucks, Jakob Wenzel

Bauhaus-University Weimar, Germany

Motivation

What is MARS?

- block cipher with 128 bit block size
- developed 1998 by a team from IBM as a candidate for the Advanced Encryption Standard (AES)
- one of five finalists in the AES competition 2001
- no attacks from 2001 till 2009

Motivation (cont'd)

Why is MARS an interesting subject to study?

- full AES is theoretically broken

Motivation (cont'd)

Why is MARS an interesting subject to study?

- full AES is theoretically broken
- many attacks on AES base on exploiting the relatively weak key schedule of AES

Motivation (cont'd)

Why is MARS an interesting subject to study?

- full AES is theoretically broken
- many attacks on AES base on exploiting the relatively weak key schedule of AES
- MARS structure differs from other ciphers (mixing rounds)

Motivation (cont'd)

Why is MARS an interesting subject to study?

- full AES is theoretically broken
- many attacks on AES base on exploiting the relatively weak key schedule of AES
- MARS structure differs from other ciphers (mixing rounds)
- key scheduler much stronger/ more complex than key scheduler of AES

What we did

We propose two attacks:

- extend 11-round distinguisher by Kelsey et al to 12 core rounds

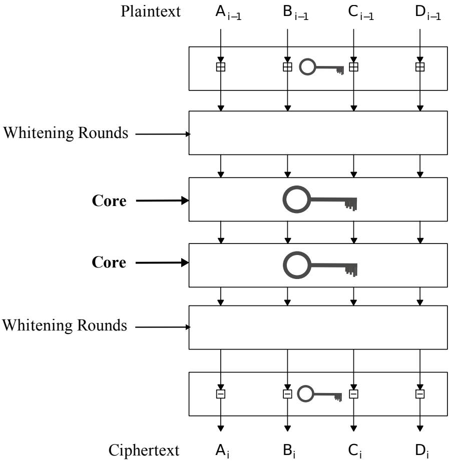
What we did

We propose two attacks:

- extend 11-round distinguisher by Kelsey et al to 12 core rounds
- establish first key recovery attack on the MARS key schedule, using the distinguisher to recover the secret key

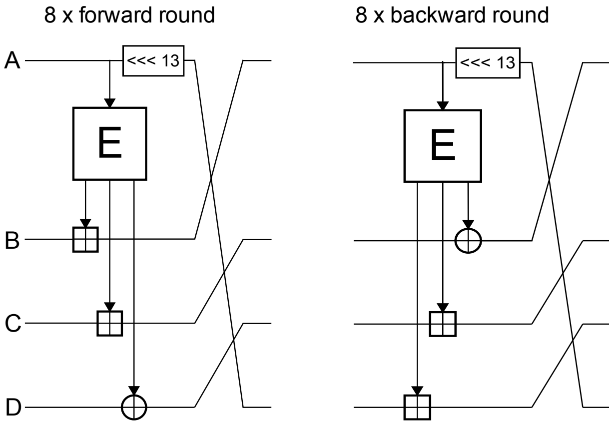
Outline

MARS



- 128 bit block size
- internal state: 4×32 bit words (A, B, C, D)

MARS - Structure of the Core Rounds



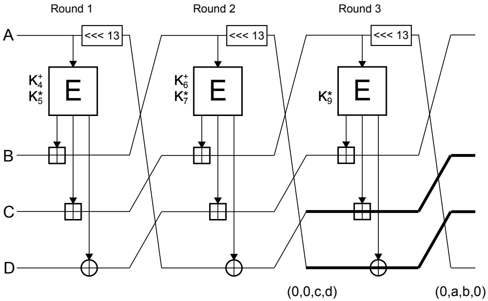
Distinguisher and Subkey Recovery

Exploits differential properties of the MARS core

- 3-round differential characteristic with probability 1
  $(0, 0, 0, \alpha) \rightarrow (\beta, 0, 0, 0)$
- distinguisher uses the 3-rounds characteristic twice, for rounds 4 - 6 and 7 - 9
- differences, if multiplied with a constant, propagate only in the most significant bits (used in round 10)

Distinguisher and Subkey Recovery (cont'd)

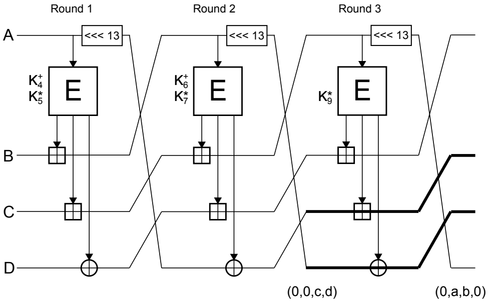
For each of the 2^{154} subkey candidates of the first three rounds do:



1. choose 2^{56} texts with arbitrary differences $(0, a, b, 0)$

Distinguisher and Subkey Recovery (cont'd)

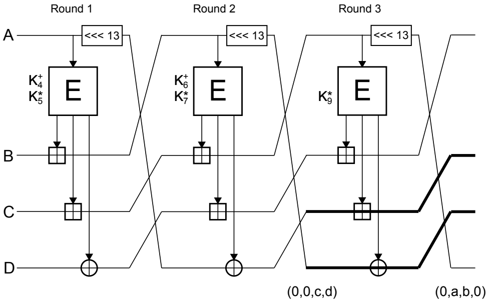
For each of the 2^{154} subkey candidates of the first three rounds do:



1. choose 2^{56} texts with arbitrary differences $(0, a, b, 0)$
2. partially decrypt $(0, a, b, 0)$ to reach (A, B, C, D)

Distinguisher and Subkey Recovery (cont'd)

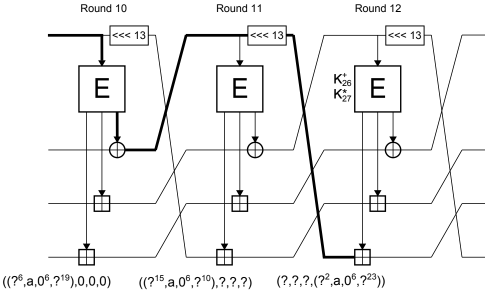
For each of the 2^{154} subkey candidates of the first three rounds do:



1. choose 2^{56} texts with arbitrary differences $(0, a, b, 0)$
2. partially decrypt $(0, a, b, 0)$ to reach (A, B, C, D)
3. create 2^{56} batches with 302 texts each with difference (A, B, C, D) between batches

Distinguisher and Subkey Recovery (cont'd)

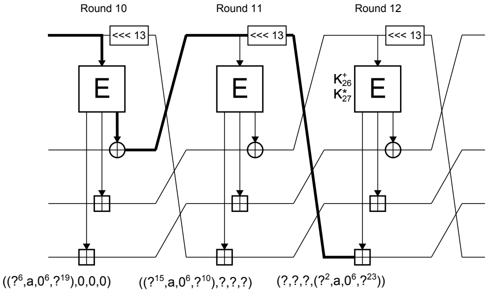
For each of the 2^{154} subkey candidates of the first three rounds do:



5. partially decrypt all ciphertexts with each of the 2^{32} subkey candidates for Round 12 and extract the bit "*a*" for each ciphertext

Distinguisher and Subkey Recovery (cont'd)

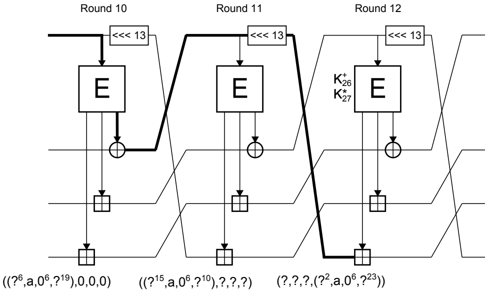
For each of the 2^{154} subkey candidates of the first three rounds do:



5. partially decrypt all ciphertexts with each of the 2^{32} subkey candidates for Round 12 and extract the bit "*a*" for each ciphertext
6. build 2^{56} strings of 302 "*a*" bits for each batch

Distinguisher and Subkey Recovery (cont'd)

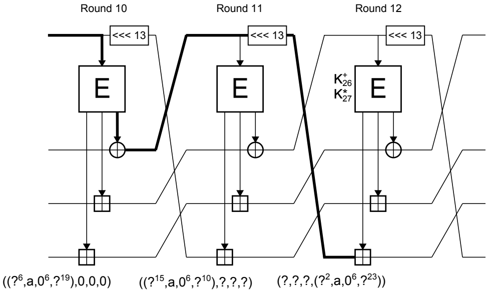
For each of the 2^{154} subkey candidates of the first three rounds do:



7. store and sort the resulting bit strings in order of the chosen plaintexts

Distinguisher and Subkey Recovery (cont'd)

For each of the 2^{154} subkey candidates of the first three rounds do:



7. store and sort the resulting bit strings in order of the chosen plaintexts
8. compare the bit strings pairwise to identify the correct subkey candidate

What we got from the Distinguisher

valid subkeys for

$$\{K_4^+, K_5^*, K_6^+, K_7^*, K_9^*, K_{26}^+, K_{27}^*(9 \text{ bit})\}.$$

MARS Key Schedule

- expands 256-bit secret key to 40 subkeys
- four iterations, each iteration generates 10 round keys

MARS Key Schedule

- expands 256-bit secret key to 40 subkeys
- four iterations, each iteration generates 10 round keys
- uses internal array $T[0 \ldots 14]$ with 15×32-bit words

MARS Key Schedule

- expands 256-bit secret key to 40 subkeys
- four iterations, each iteration generates 10 round keys
- uses internal array $T[0 \ldots 14]$ with 15×32-bit words
- three phases per iteration:
  - linear transformation
  - four stirring rounds
  - removing patterns from multiplication keys

Key Schedule (cont'd)

- **Initialization** ($T[0] \dots T[7] = \text{key};\ T[8] \dots T[14] = 0$)

Key Schedule (cont'd)

- **Initialization** ($T[0] \ldots T[7] =$ key; $T[8] \ldots T[14] = 0$) and four iterations of...
- **Linear transformation**
  for ($i = 0, \ldots, 14$)
  $T[i] = T[i] \oplus ((T[(i-7) \bmod 15] \oplus T[(i-2) \bmod 15]) \lll 3) \oplus (4i+j)$

Key Schedule (cont'd)

- **Initialization** ($T[0] \ldots T[7] = \text{key};\ T[8] \ldots T[14] = 0$) and four iterations of...
- **Linear transformation**
  for ($i = 0, \ldots, 14$)
  $T[i] = T[i] \oplus ((T[(i-7) \bmod 15] \oplus T[(i-2) \bmod 15]) \lll 3) \oplus (4i+j)$
- **Four stirring rounds**
  for ($k = 1, \ldots, 4$)
  for ($i = 0, \ldots, 14$)
  $T[i] = (T[i] + S[\text{low 9 bits of } T[(i-1) \bmod 15]]) \lll 9$

Key Schedule (cont'd)

- **Initialization** ($T[0] \dots T[7] =$ key; $T[8] \dots T[14] = 0$) and four iterations of...
- **Linear transformation**
  for $(i = 0, \dots, 14)$
  $T[i] = T[i] \oplus ((T[(i-7) \bmod 15] \oplus T[(i-2) \bmod 15]) \lll 3) \oplus (4i+j)$
- **Four stirring rounds**
  for $(k = 1, \dots, 4)$
  for $(i = 0, \dots, 14)$
  $T[i] = (T[i] + S[\text{low 9 bits of } T[(i-1) \bmod 15]]) \lll 9$
- **Storing next 10 keys**
  for $(i = 0, \dots, 9)$
  $K[10j + i] = T[4i \bmod 15]$

Key Schedule (cont'd)

- **Initialization** ($T[0]\dots T[7] = \text{key};\ T[8]\dots T[14] = 0$) and four iterations of...
- **Linear transformation**
  for ($i = 0, \dots, 14$)
  $T[i] = T[i] \oplus ((T[(i-7) \bmod 15] \oplus T[(i-2) \bmod 15]) \lll 3) \oplus (4i+j)$
- **Four stirring rounds**
  for ($k = 1, \dots, 4$)
  for ($i = 0, \dots, 14$)
  $T[i] = (T[i] + S[\text{low 9 bits of } T[(i-1) \bmod 15]]) \lll 9$
- **Storing next 10 keys**
  for ($i = 0, \dots, 9$)
  $K[10j + i] = T[4i \bmod 15]$
- **Modification of multiplication keys**

MITM Attack on the MARS Key Schedule

- Kelsey et al. finished after recovering subkeys:
  - subkeys from 3rd and 4th iteration

MITM Attack on the MARS Key Schedule

- Kelsey et al. finished after recovering subkeys:
  - subkeys from 3rd and 4th iteration
- **difficult to invert multiple iterations**

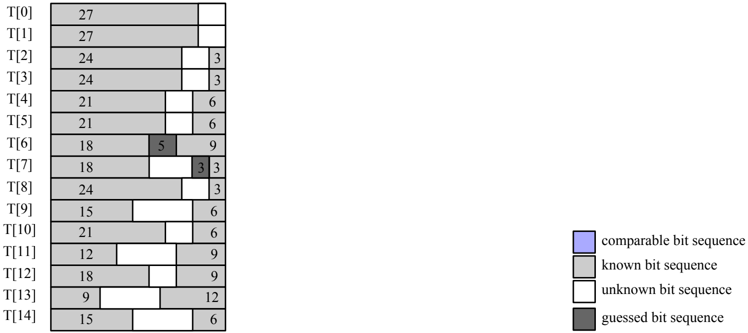
MITM Attack on the MARS Key Schedule

- Kelsey et al. finished after recovering subkeys:
    - subkeys from 3rd and 4th iteration
- **difficult to invert multiple iterations**
- idea: mount a Meet-in-the-Middle-Attack on the first iteration

Key Schedule (cont'd)

- **Initialization** ($T[0] \dots T[7] =$ key; $T[8] \dots T[14] = 0$) and four iterations of...
- **Linear transformation**
  for $(i = 0, \dots, 14)$
  $T[i] = T[i] \oplus ((T[(i-7) \bmod 15] \oplus T[(i-2) \bmod 15]) \lll 3) \oplus (4i+j)$
- **Four stirring rounds**
  for $(k = 1, \dots, 4)$
  for $(i = 0, \dots, 14)$
  $T[i] = (T[i] + S[\text{low 9 bits of } T[(i-1) \bmod 15]]) \lll 9$
- **Storing next 10 keys**
  for $(i = 0, \dots, 9)$
  $K[10j + i] = T[4i \bmod 15]$
- **Modification of multiplication keys**

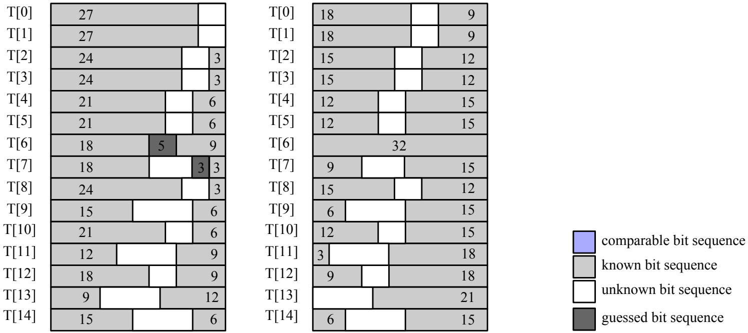
MITM - Forward Step



- Linear Transformation:

$$T[i] = T[i] \oplus ((T[i - 7 \bmod 15] \oplus T[i - 2 \bmod 15]) \lll 3) \oplus (4i + j)$$

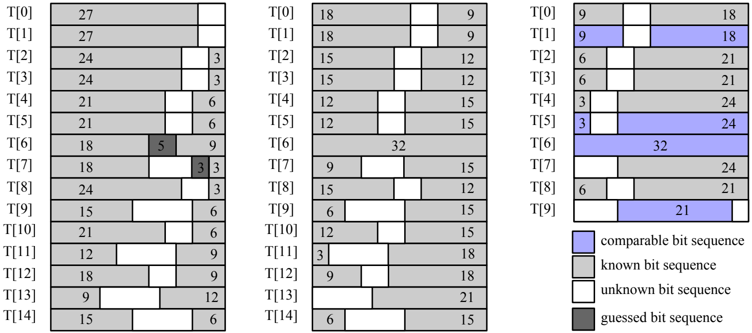
MITM - Forward Step



- First Stirring Round:

  $T[i] = (T[i] + S[\text{low 9 bits of } T[i-1 \bmod 15]]) \lll 9$

MITM - Forward Step



- Second Stirring Round:

  $T[i] = (T[i] + S[\text{low 9 bits of } T[i-1 \bmod 15]]) \lll 9$

MITM - Backward Step

- our distinguisher recovers five subkeys from first iteration: $\{K_4^+, K_5^*, K_6^+, K_7^*, K_9^*\}$

MITM - Backward Step

- our distinguisher recovers five subkeys from first iteration: $\{K_4^+, K_5^*, K_6^+, K_7^*, K_9^*\}$
- attack uses four subkeys that are mapped to $T[i]$s as follows: $\{K_4^+, K_5^*, K_6^+, K_9^*\} \rightarrow \{T[1], T[5], T[9], T[6]\}$

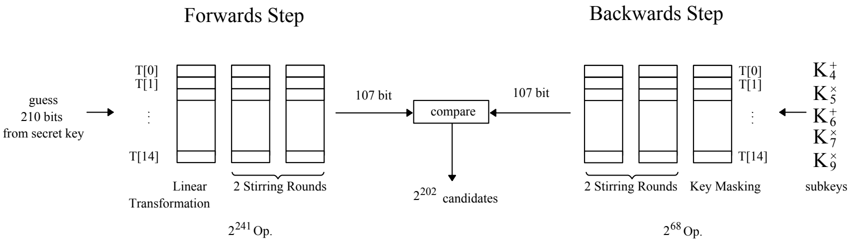
MITM - Backward Step

- **Modification of multiplication keys**
  - invert multiplication keys $\{K_5^*, K_9^*\}$
  - lookup table for K $\rightarrow$ T projections
  - max. $102 \approx 2^7$ candidates
  - 2^{14} candidates

MITM - Backward Step

- **Modification of multiplication keys**
  - invert multiplication keys $\{K_5^*, K_9^*\}$
  - lookup table for K $\rightarrow$ T projections
  - max. $102 \approx 2^7$ candidates
  - 2^{14} candidates
- **Two stirring rounds backwards**
  - require least significant nine bits for each of our four words $T[i]$ for each stirring round
  - know the bits for $T[6]$ after guessing $T[5]$
  - $2^{9 \cdot 3 \cdot 2}$ op. $= 2^{54}$ op.
- $2^{14} \cdot 2^{54}$ op. $= 2^{68}$ op. for backward step

MITM Attack on the MARS Key Schedule



Attack Analysis

Distinguisher operations:

- 2^{65} Texts $\cdot 2^{186}$ Keys $\cdot 3$ Executions $\approx 2^{252}$ *Encryptions*
- 3 executions are required as one 3-round differential for round 7-9 has probability $\neq 1$

Attack Analysis

Forward step:

- guessing the bits of $T[0] \ldots T[7]$: 2^{210}
- guessing 5 bit of $T[6]$ and 3 bit of $T[7]$: 2^{8}
- carry bit for 23 additions: 2^{23}
- **summarize:** 2^{241}

Attack Analysis

Forward step:

- guessing the bits of $T[0] \ldots T[7]$: 2^{210}
- guessing 5 bit of $T[6]$ and 3 bit of $T[7]$: 2^{8}
- carry bit for 23 additions: 2^{23}
- **summarize:** 2^{241}

Backward step:

- nine bits for $T[0]$, $T[4]$, $T[8]$ (two stirring rounds): 2^{54}
- multiplication keys (from possible table entries): 2^{14}
- **summarize:** 2^{68}

Attack Analysis

- probability of finding a matching pair of 107 bits is 2^{-107}.

Attack Analysis

- probability of finding a matching pair of 107 bits is 2^{-107}.
- combine forward and backward step:

$$2^{241} \cdot 2^{68} \cdot 2^{-107} = 2^{202}.$$

Attack Analysis

- probability of finding a matching pair of 107 bits is 2^{-107}.
- combine forward and backward step:

$$2^{241} \cdot 2^{68} \cdot 2^{-107} = 2^{202}.$$

- We gather 2^{202} candidates for 210 bits of the secret key
- $2^{202} \cdot 2^{46} = 2^{248}$ Op. for final testing

Conclusion

- we have ...
  - extended the 11-round attack by Kelsey et al to a differential attack on 12 rounds
  - suggested a MITM attack on the MARS key schedule that allows to recover the secret key more efficiently than exhaustive search

Recent Attacks on MARS/Analysis

Type	Rounds	Texts	Bytes	Op.	Reference
Differential	**12C**	2^{65}	2^{69}	2^{252}	**this work**
Amp. Boomerang	11C	2^{65}	2^{70}	2^{229}	[KKS00]
Amp. Boomerang	6M, 6C	2^{69}	2^{73}	2^{197}	[KS00]
MITM	16M, 5C	8	2^{236}	2^{232}	[KS00]
Diff. MITM	16M, 5C	2^{50}	2^{197}	2^{247}	[KS00]
Impossible Diff.	8C	-	-	-	[BF00]
Differential	8M, 8C	2^{105}	2^{109}	2^{231}	[Pes09]

Table: Op: operations, C: core rounds, M: mixing rounds