Akelarre

Akelarre

- Block cipher
- Combines features of 2 strong ciphers
  - IDEA — "mixed mode" arithmetic
  - RC5 — keyed rotations
- Goal is a more efficient strong cipher
- Proposed in 1996, broken within a year
  - "Two rights sometimes make a wrong"

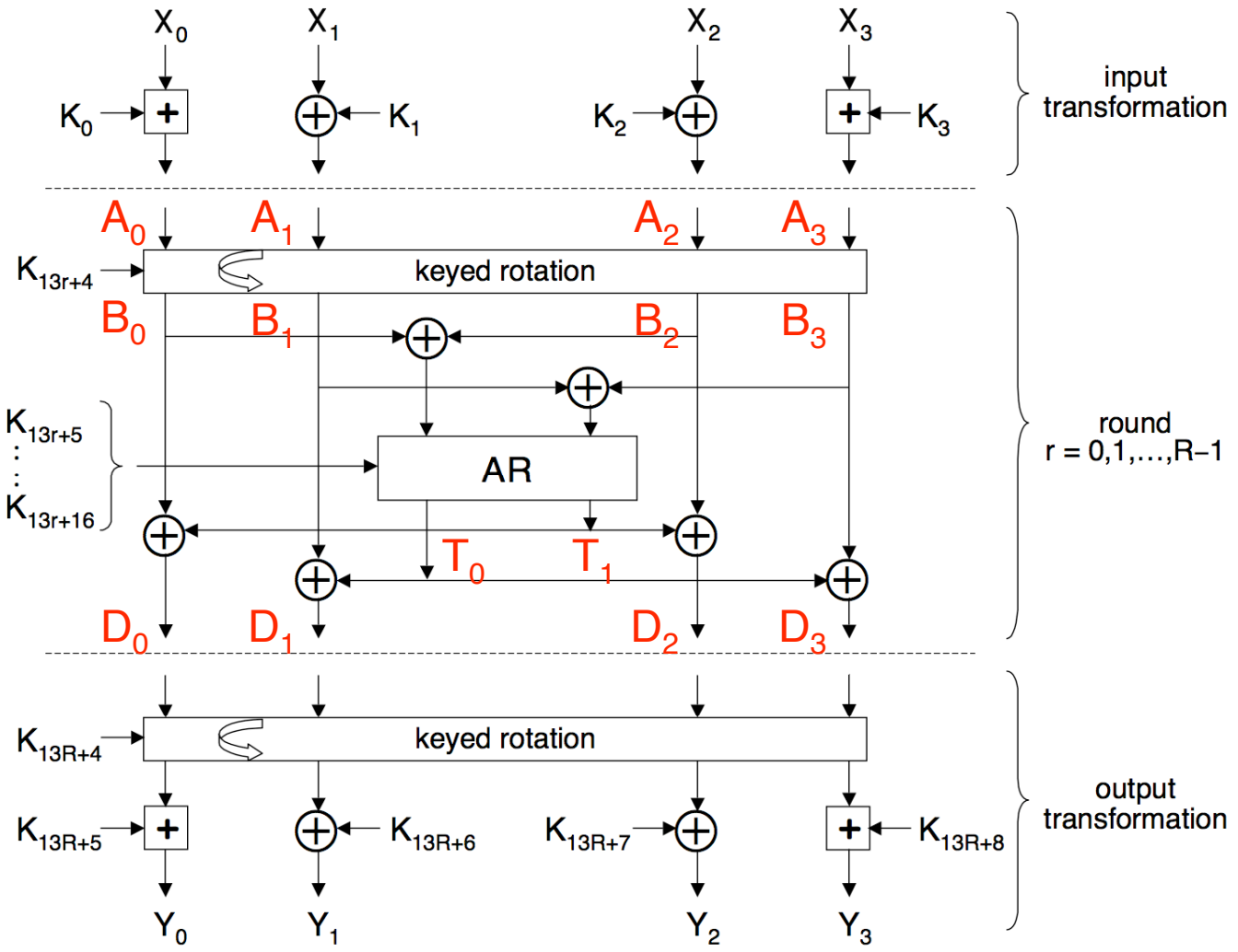
Akelarre Cipher

- Variable number of rounds
  - Authors conjecture 4 rounds is secure
  - Actually, insecure for any number of rounds!
- Block length is 128 bits
- Key length can be any multiple of 64 bits
  - We assume 128-bit key
  - No more secure for larger key
- Algorithm based on 32-bit words

Akelarre Cipher

- To encrypt
  - Input transformation
  - Round 0,1,2,…,R−1
  - Output transformation
- Lots of subkey blocks: 13R + 8
  - Each is a 32-bit word
- Employs addition mod 2^{32} and XOR
- Addition-rotation (AR) structure

Akelarre


X0
X1
X2
X3
K0
K1
K2
K3
input transformation
A0
A1
A2
A3
K13r+4
keyed rotation
B0
B1
B2
B3
K13r+5
K13r+16
AR
T0
T1
D0
D1
D2
D3
round
r = 0,1,…,R−1
K13R+4
keyed rotation
K13R+5
K13R+6
K13R+7
K13R+8
Y0
Y1
Y2
Y3
output transformation

Akelarre Round

- Let (A_0,A_1,A_2,A_3) be input to round r
- And (B_0,B_1,B_2,B_3) is the output
- Then

  $(B_0,B_1,B_2,B_3) = (A_0,A_1,A_2,A_3) <<< \langle K_{13r+4} \rangle_{25\ldots31}$
- Where "<<<" is left rotation and $\langle X \rangle_{i\ldots j}$ means bits i thru j (inclusive) of X
  - Recall, bits numbered left-to-right, from 0

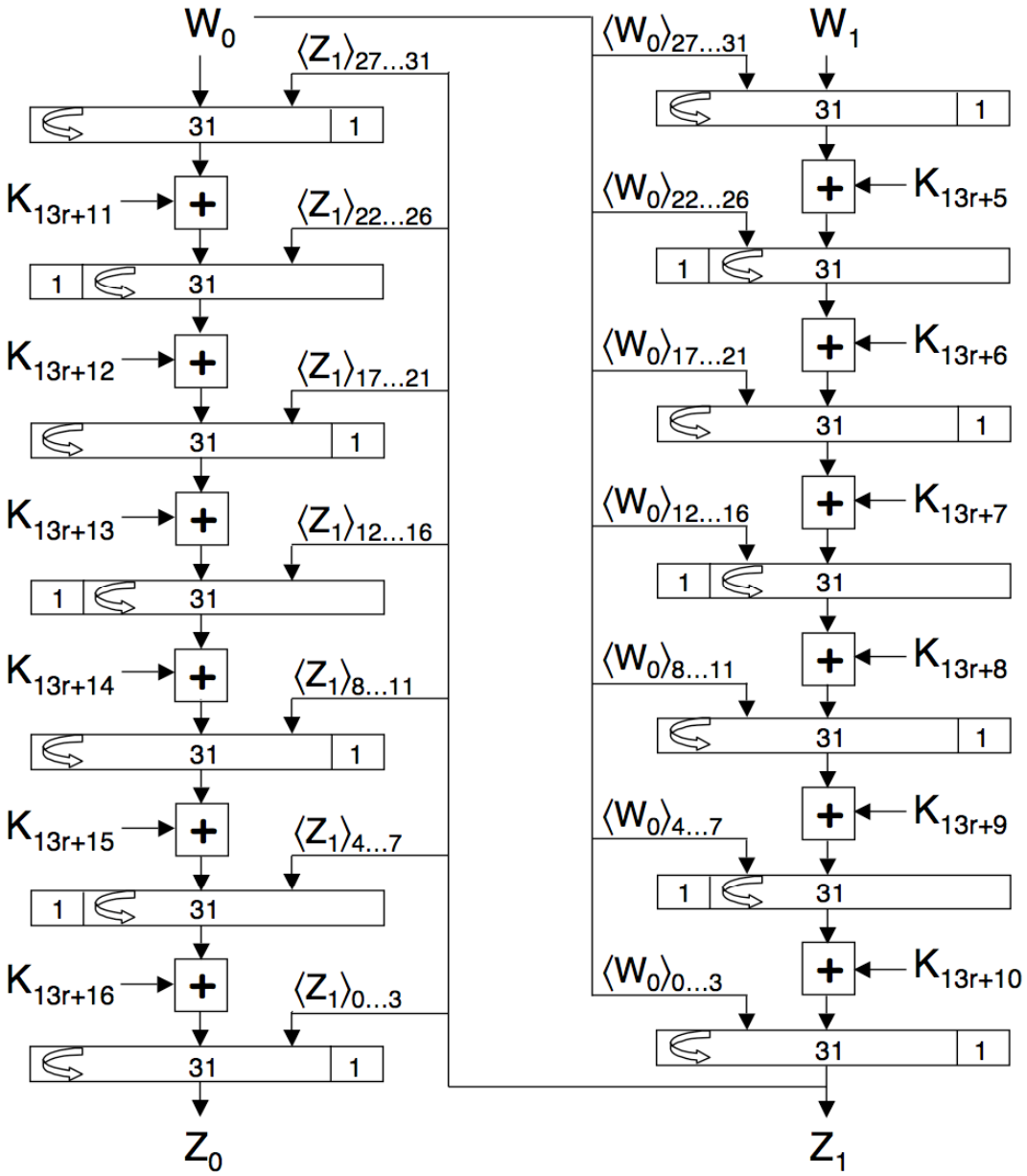
Akelarre Round

- Let (T_0,T_1) be output of AR structure
- Then $(T_0,T_1) = AR(B_0 \oplus B_2, B_1 \oplus B_3)$
  - We ignore dependence of AR on subkey
- Let (D_0,D_1,D_2,D_3) be output of round r
- Then

  $(D_0,D_1,D_2,D_3) = (B_0 \oplus T_1, B_1 \oplus T_0, B_2 \oplus T_1, B_3 \oplus T_0)$
- Block (D_0,D_1,D_2,D_3) is input to next round
  - Except for final round

AR Structure

- Note: W_1 processed first
- Rotate 31 bits, with 1 bit fixed

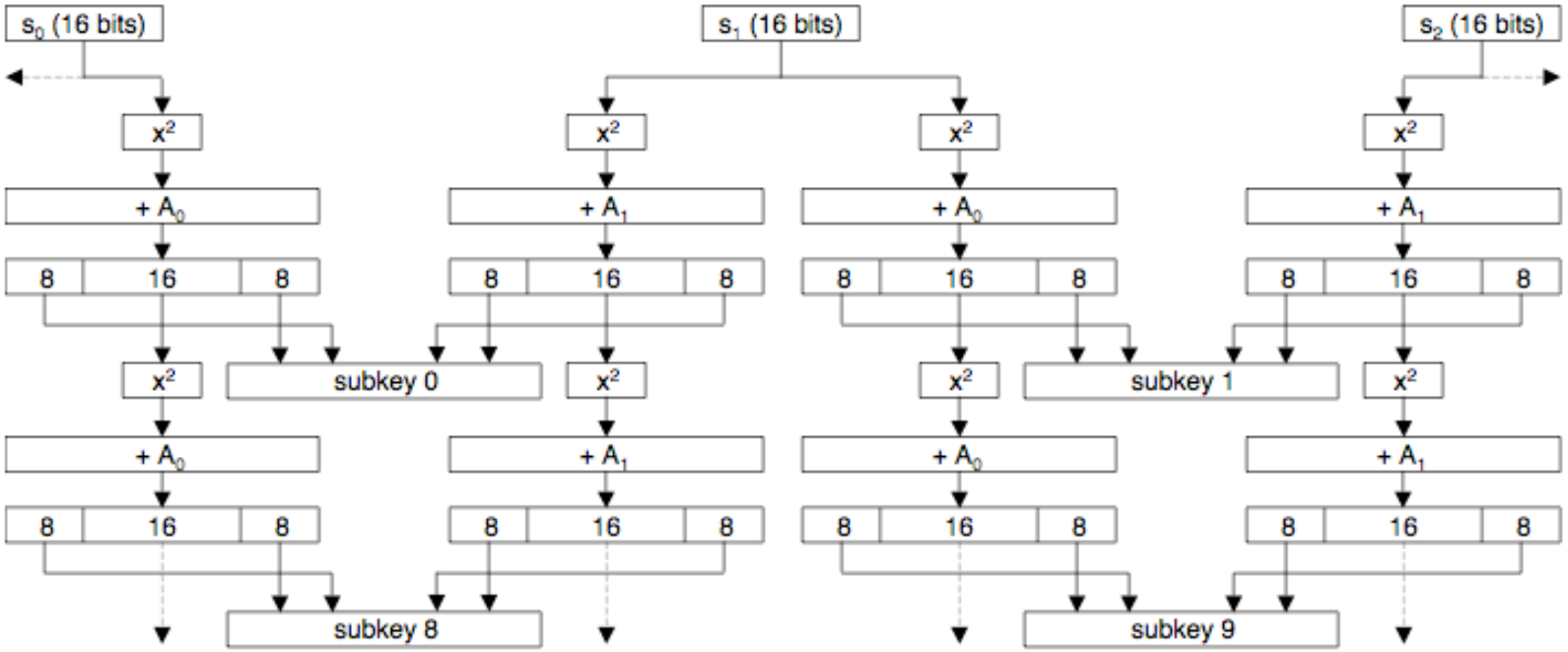


AR Structure

- For example, first step is
  $((\langle W_1 \rangle_{0\ldots30} <<< \langle W_0 \rangle_{27\ldots31}), \langle W_1 \rangle_{31\ldots31})$
- One pass thru AR is 14 half-rounds
  - Like 7 rounds of typical block cipher
  - But each Akalarre "round" is very simple
- AR structure is heart of Akelarre
- How to attack AR structure?

Key Schedule

- Assuming 128-bit key
- See textbook for details



Subkeys

- Encryption and decryption subkeys

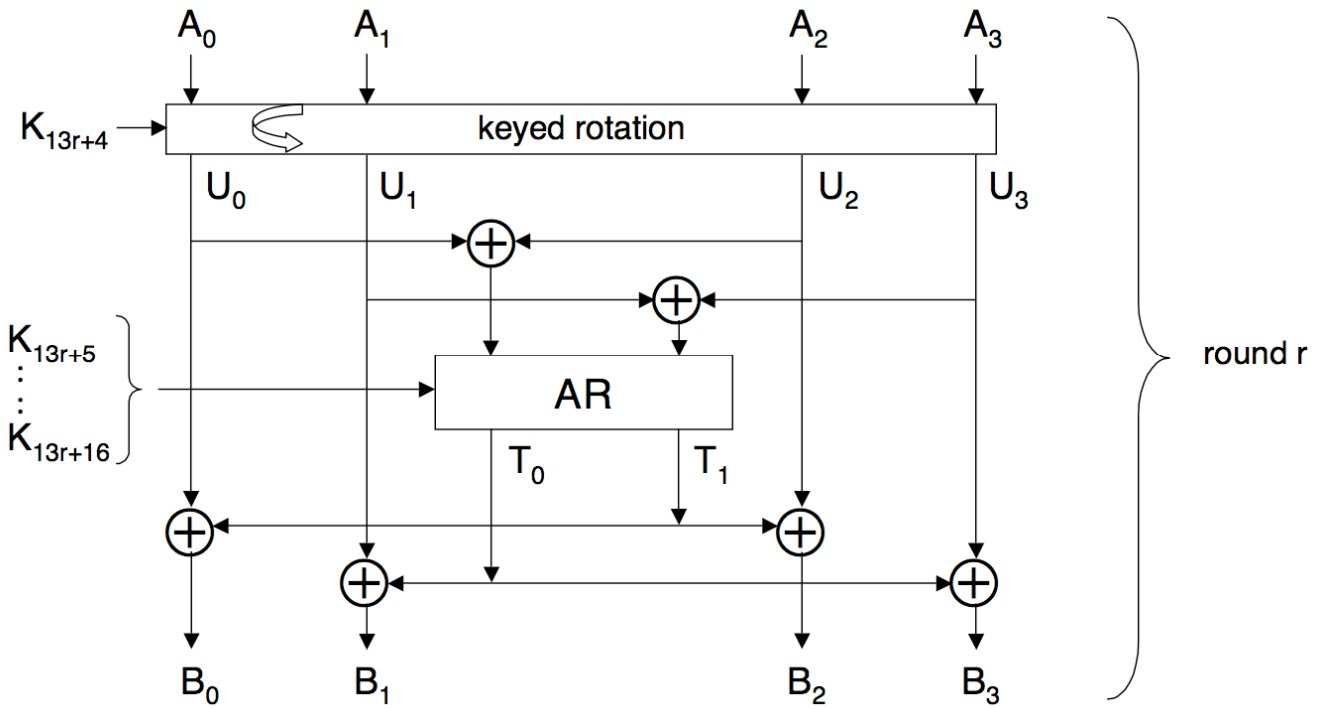
Transformation	Encryption Subkeys	Decryption Subkeys
Input	K_0	$-K_{13R+5}$
	K_1	K_{13R+6}
	K_2	K_{13R+7}
	K_3	$-K_{13R+8}$
Round $r = 0, 1, \ldots, R-1$	K_{13r+4}	$\mathrm{neg}(K_{13(R-r)+4})$
	K_{13r+5}	$K_{13(R-r-1)+5}$
	K_{13r+6}	$K_{13(R-r-1)+6}$
	$\vdots$	$\vdots$
	K_{13r+16}	$K_{13(R-r-1)+16}$
Output	K_{13R+4}	$\mathrm{neg}(K_4)$
	K_{13R+5}	$-K_0$
	K_{13R+6}	K_1
	K_{13R+7}	K_2
	K_{13R+8}	$-K_3$

Akelarre Attack

- AR structure is heart of Akelarre
- Incredibly, can "bypass" AR structure
- Idea of attack is simple
- Details not quite so simple
- Recovering plaintext is not easy
  - But it can be done

Akelarre

- Consider round r of Akelarre



Akelarre Attack

- Let

  $A = (A_0,A_1,A_2,A_3)$ be input to round r

  $U = (U_0,U_1,U_2,U_3)$ be result of rotation

  $T = (T_0,T_1)$ be output of AR structure

  $B = (B_0,B_1,B_2,B_3)$ be output of round r

- Denote $A = (a_0,a_1,\ldots,a_{127})$
  - And similarly for U,T,B

Akelarre Attack

- Let ℓ be size of keyed rotation
- Then U = (A <<< ℓ) and

  $(B_0,B_1,B_2,B_3) = (U_0 \oplus T_1, U_1 \oplus T_0, U_2 \oplus T_1, U_3 \oplus T_0)$
- It follows that

  $B_0 \oplus B_2 = U_0 \oplus U_2$ and $B_1 \oplus B_3 = U_1 \oplus U_3$
- AR structure has vanished!
  - The crucial observation for attack

Akalarre Attack

- We have

  $(U_0 \oplus U_2, U_1 \oplus U_3) = (A_0 \oplus A_2, A_1 \oplus A_3) <<< \ell \pmod{64}$

  where subscripts are taken mod 128

- Then input to round related to output:

  $(B_0 \oplus B_2, B_1 \oplus B_3) = (A_0 \oplus A_2, A_1 \oplus A_3) <<< \ell \pmod{64}$

- Neither key nor AR structure appears!

Akalarre Attack

- But this is only one round
- Must extend thru multiple rounds
  - Only changes the rotation amount
- And input/output transformations
  - Rotation and addition/XOR of **subkey**
- Let (X_0,X_1,X_2,X_3) be plaintext and (Y_0,Y_1,Y_2,Y_3) corresponding ciphertext

Akelarre Attack

- Not difficult to show that

$$((Y_0 - K_{13R+5}) \oplus Y_2 \oplus K_{13R+7},$$
$$Y_1 + K_{13R+6} \oplus (Y_2 - K_{13R+8}))$$
$$= ((X_0 + K_0) \oplus X_2 \oplus K_2,$$
$$X_1 \oplus K_1 \oplus (X_3 + K_3)) \lll L \pmod{64}$$

  where L is sum of all rotations

- See textbook for details…

Akelarre Attack

- Assuming known plaintext, we have

  $((Y_0 - K_{13R+5}) \oplus Y_2 \oplus K_{13R+7},$
  $\quad Y_1 + K_{13R+6} \oplus (Y_2 - K_{13R+8}))$
  $= ((X_0 + K_0) \oplus X_2 \oplus K_2,$
  $\quad X_1 \oplus K_1 \oplus (X_3 + K_3)) <<< L \pmod{64}$

  with unknown L and 8 unknown subkeys
- Provided sufficient known plaintext
  - We can then solve for all of the unknowns
  - Just 5 known plaintext blocks is enough

Akelarre Attack

- After solving, we obtain equations

  $((Y_0 - K_{13R+5}) \oplus Y_2 \oplus K_{13R+7},$
  $\quad Y_1 + K_{13R+6} \oplus (Y_2 - K_{13R+8}))$
  $= ((X_0 + K_0) \oplus X_2 \oplus K_2,$
  $\quad X_1 \oplus K_1 \oplus (X_3 + K_3)) <<< L \pmod{64}$

  where plaintext X is the only unknown
- Reduces number of possible plaintexts
- Can then solve for plaintext X in many cases
  - For example, if X is known to be English
  - Or if X is known to be (random) ASCII

Akelarre Attack

- Only a small part of key is recovered
- But we can recover plaintext directly from ciphertext
- Recovering plaintext in this way is not trivial
  - Homework problems give simplified examples
- But, much better than exhaustive key search
- Attack is practical in many cases!

Improved Akelarre

- Attack bypasses virtually all of the complexity of the algorithm
  - That is, the AR structure
- Must force attacker to deal with the AR structure
- There is an "improved" Akelarre
  - Ake98 — also weak

Akelarre Conclusions

- A seriously bad cipher!
- Conceptually easy attack
  - Details not quite so straightforward
- Akelarre shows cipher design is tricky
- Aside: Knuth analyzed a "super-random" number generator and found it was bad
  - Concluded that "random number should not be generated with a method chosen at random"
  - Similar rule applies to ciphers!