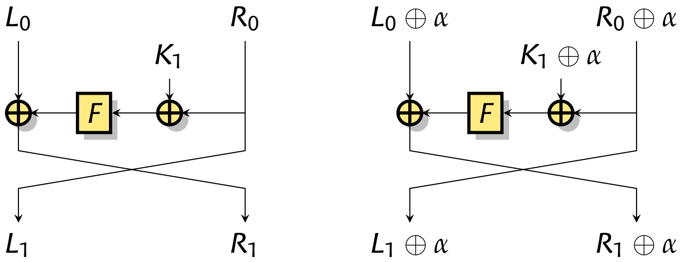
Another Look at Complementation Properties

Charles Bouillaguet, Orr Dunkelman,
Gaëtan Leurent, Pierre-Alain Fouque

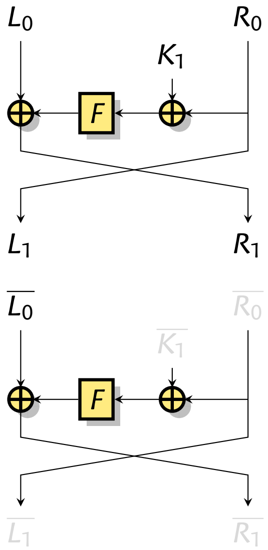
École Normale Supérieure
Paris, France



DES's Complementation Property

- If the key is bitwise complemented, so are all the subkeys.
  $K \to K_1, K_2, \ldots, K_{16}$ and
  $\overline{K} \to \overline{K_1}, \overline{K_2}, \ldots, \overline{K_{16}}$
- If the state is also complemented the input to the F function is the same.
- Therefore the output is the same.
  $R'_1 = \overline{L_0} \oplus F(\overline{K_1} \oplus \overline{R_0})$
- DES's complementation property:

$$\overline{DES_{\overline{K}}(\overline{P})} = DES_K(P)$$



DES's Complementation Property

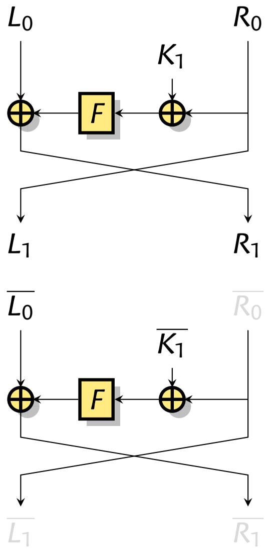
- If the key is bitwise complemented, so are all the subkeys.
  $K \to K_1, K_2, \ldots, K_{16}$ and
  $\overline{K} \to \overline{K_1}, \overline{K_2}, \ldots, \overline{K_{16}}$

- If the state is also complemented the input to the F function is the same.

- Therefore the output is the same.
  $R_1' = \overline{L_0} \oplus F(\overline{K_1} \oplus \overline{R_0})$

- DES's complementation property:

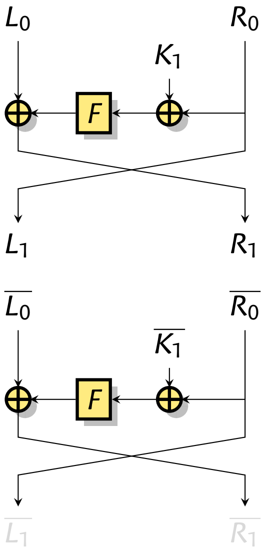
$$\overline{DES_{\overline{K}}(\overline{P})} = DES_K(P)$$



DES's Complementation Property

- If the key is bitwise complemented, so are all the subkeys.
  $K \rightarrow K_1, K_2, \ldots, K_{16}$ and
  $\overline{K} \rightarrow \overline{K_1}, \overline{K_2}, \ldots, \overline{K_{16}}$
- If the state is also complemented the input to the F function is the same.
- Therefore the output is the same.
  $R_1' = \overline{L_0} \oplus F(\overline{K_1} \oplus \overline{R_0})$
- DES's complementation property:

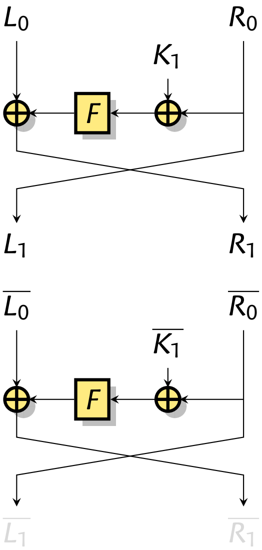
$$\overline{DES_{\overline{K}}(\overline{P})} = DES_K(P)$$



DES's Complementation Property

- If the key is bitwise complemented, so are all the subkeys.
  $K \to K_1, K_2, \ldots, K_{16}$ and
  $\overline{K} \to \overline{K_1}, \overline{K_2}, \ldots, \overline{K_{16}}$
- If the state is also complemented the input to the F function is the same.
- Therefore the output is the same.
  $R_1' = \overline{L_0} \oplus F(K_1 \oplus R_0)$
- DES's complementation property:

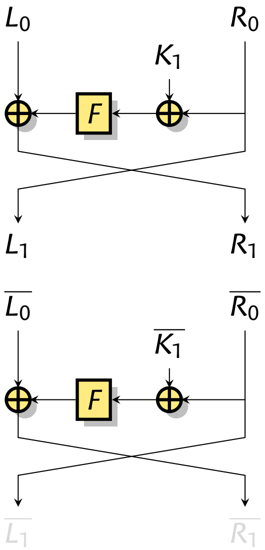
$$\overline{DES_{\overline{K}}(\overline{P})} = DES_K(P)$$



DES's Complementation Property

- If the key is bitwise complemented, so are all the subkeys.
  $K \rightarrow K_1, K_2, \ldots, K_{16}$ and
  $\overline{K} \rightarrow \overline{K_1}, \overline{K_2}, \ldots, \overline{K_{16}}$
- If the state is also complemented the input to the F function is the same.
- Therefore the output is the same.
  $R_1' = \overline{L_0 \oplus F(K_1 \oplus R_0)}$
- DES's complementation property:

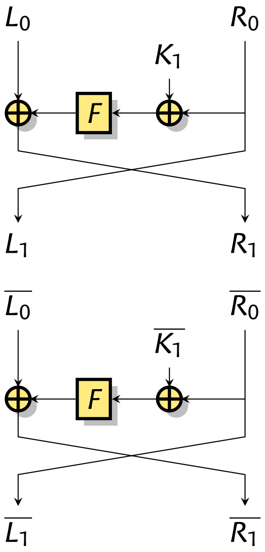
$$\overline{DES_{\overline{K}}(\overline{P})} = DES_K(P)$$



DES's Complementation Property

- If the key is bitwise complemented, so are all the subkeys.
  $K \to K_1, K_2, \ldots, K_{16}$ and
  $\overline{K} \to \overline{K_1}, \overline{K_2}, \ldots, \overline{K_{16}}$
- If the state is also complemented the input to the F function is the same.
- Therefore the output is the same.
  $R_1' = \overline{R_1}$
- DES's complementation property:

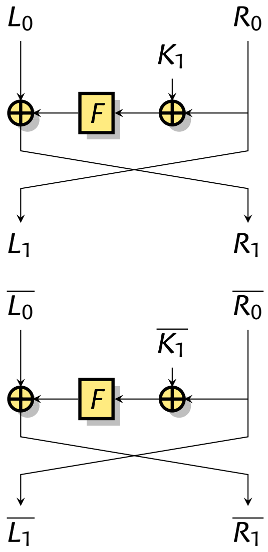
$$\overline{DES_{\overline{K}}(\overline{P})} = DES_K(P)$$



DES's Complementation Property

- If the key is bitwise complemented, so are all the subkeys.
  $K \to K_1, K_2, \ldots, K_{16}$ and
  $\overline{K} \to \overline{K_1}, \overline{K_2}, \ldots, \overline{K_{16}}$
- If the state is also complemented the input to the F function is the same.
- Therefore the output is the same.
  $R'_1 = \overline{R_1}$
- DES's complementation property:

$$\overline{DES_{\overline{K}}(\overline{P})} = DES_K(P)$$



Other similar properties

- Complementation property on LOKI:
  $E_{K \oplus \alpha}(P \oplus \alpha) = E_K(P) \oplus \alpha$
- Equivalent keys of TEA:
  $E_{K \oplus \Delta_{\text{msb}}}(P) = E_K(P)$
- Pseudo-collisions in CHI:
  $CF(\overline{H}, \overline{M}) = CF(H, M)$
- Pseudo-collisions in MD5:
  $CF(H \oplus \Delta_{\text{msb}}, M) = CF(H, M)$ with probability 2^{-48}

Generalization of the complementation property

Definition (Self-similarity relation in a block cipher)

Invertible and easy to compute transformations ϕ, ψ and θ such that:
$\forall K, P: \quad E_{\psi(K)}(\phi(P)) = \theta(E_K(P))$

Definition (Self-similarity relation in a compression function)

Invertible and easy to compute transformations ϕ, ψ and θ such that:
$\forall H, M: \quad CF(\phi(H), \psi(M)) = \theta(CF(H, M))$

- We also consider probabilistic relations.
- Broad definition.
  - Related key differential.
  - Related key slide attack.
  - Rotational cryptanalysis.

Generalization of the complementation property

Definition (Self-similarity relation in a block cipher)

Invertible and easy to compute transformations ϕ, ψ and θ such that:
$\forall K, P: \quad E_{\psi(K)}(\phi(P)) = \theta(E_K(P))$

Definition (Self-similarity relation in a compression function)

Invertible and easy to compute transformations ϕ, ψ and θ such that:
$\forall H, M: \quad CF(\phi(H), \psi(M)) = \theta(CF(H, M))$

- We also consider probabilistic relations.
- Broad definition.
  - Related key differential.
  - Related key slide attack.
  - Rotational cryptanalysis.

Our results

- Attacks on *Lesamnta*.
  - For any number of round.
  - Collision attack in $2^{n/4}$ on the compression function.
  - Improved herding attack on the hash function.
- Related key differential attack on XTEA.
  - Attack on 36 rounds.
  - 50 rounds for a class of weak keys.
- Rotational relations in *ESSENCE*.
- Algebraic relations in $\mathcal{PURE}$.
- Results on first round *SHAvite-3*$_{512}$ with weak salt.

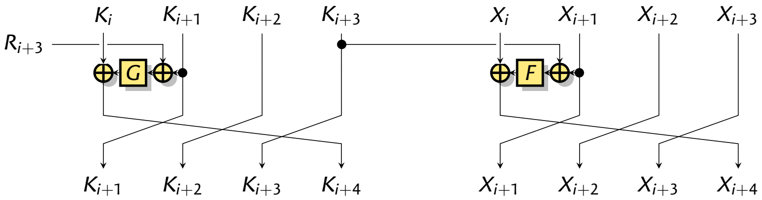
Outline

Lesamnta

- First round SHA-3 candidate
- Merkle-Damgård with an MMO compression function
- Generalized Feistel
- Round function is AES-based

Shoichi Hirose, Hidenori Kuwakado, Hirotaka Yoshida
SHA-3 Proposal: Lesamnta
Submission to the NIST SHA-3 competition

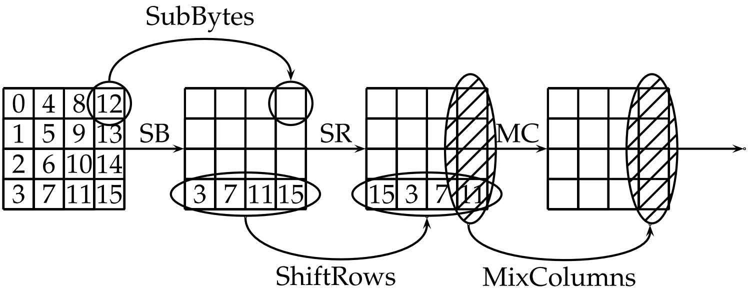
Lesamnta (cont.)



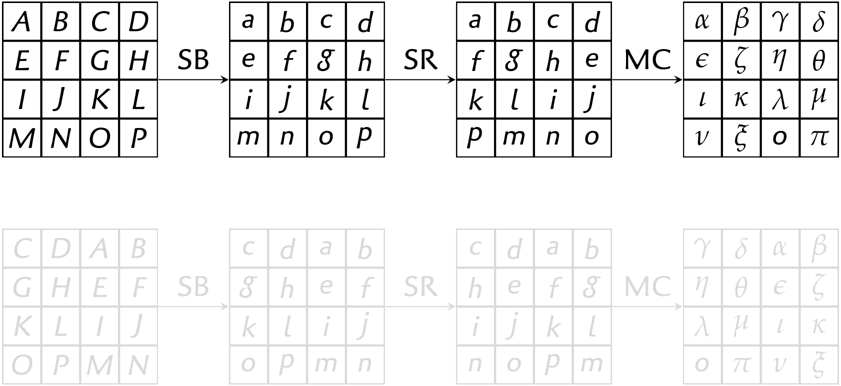
$$X_{i+4} = X_i \oplus F\left(X_{i+1} \oplus K_{i+3}\right)$$
$$K_{i+4} = K_i \oplus G\left(K_{i+1} \oplus R_{i+3}\right).$$

- Message loaded to $K_{-3}, K_{-2}, K_{-1}, K_0$
- Chaining value loaded to $X_{-3}, X_{-2}, X_{-1}, X_0$
- F and G AES-based

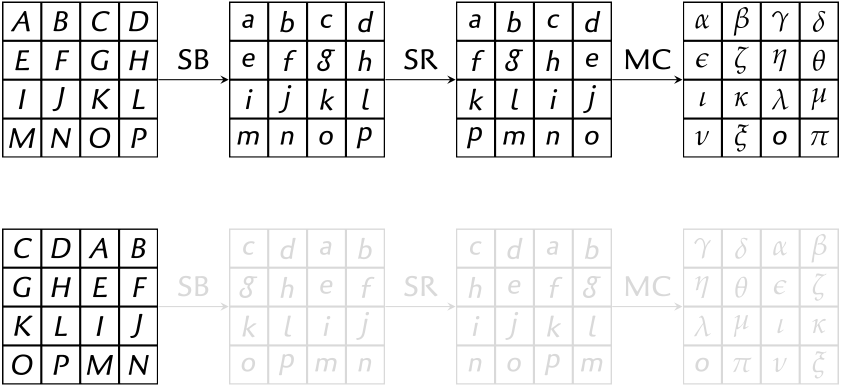
The AES Round function



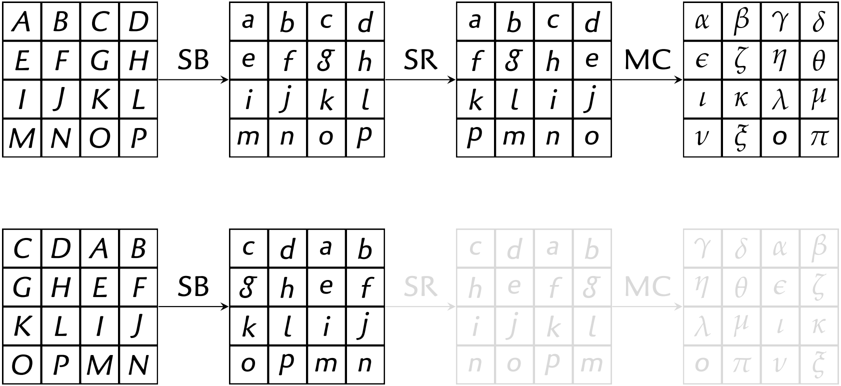
Some Interesting Properties of AES [LSWD04]



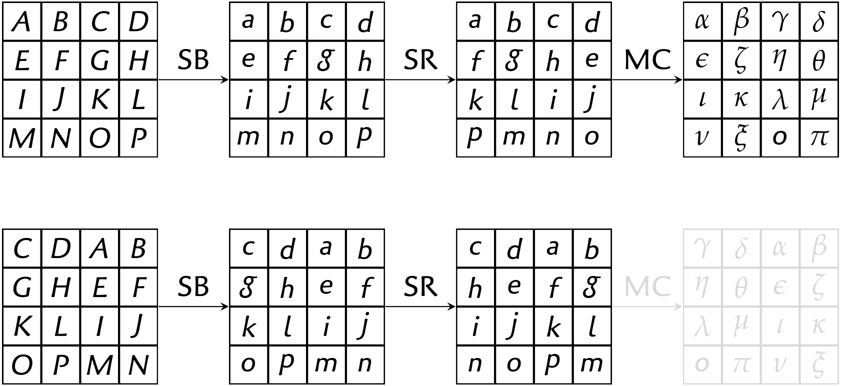
Some Interesting Properties of AES [LSWD04]



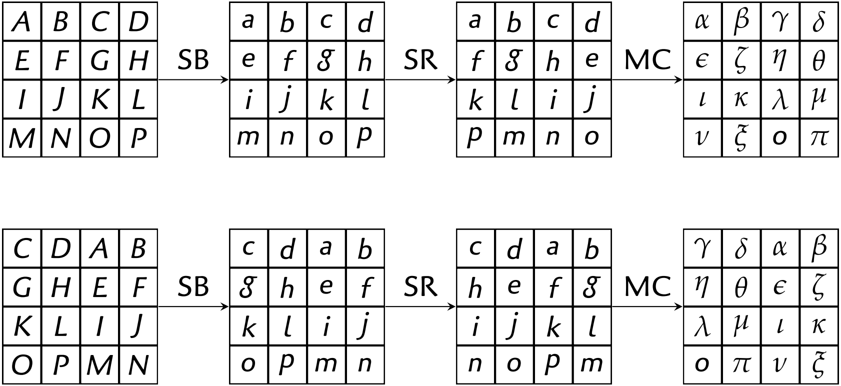
Some Interesting Properties of AES [LSWD04]



Some Interesting Properties of AES [LSWD04]



Some Interesting Properties of AES [LSWD04]

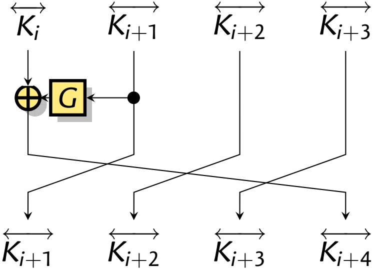


Some Interesting Properties of Lesamnta's F and G

- *Lesamnta*'s F posses similar properties:
  $F(X, Y) = (Z, W) \Rightarrow F(Y, X) = (W, Z)$.
- The same is true for G as well:
  $G(X, Y) = (Z, W) \Rightarrow G(Y, X) = (W, Z)$.
- Let $\overleftrightarrow{(a, b)} = (b, a)$
  - $F(\overleftrightarrow{x}) = \overleftrightarrow{F(x)}$
  - $G(\overleftrightarrow{x}) = \overleftrightarrow{G(x)}$

Complementation-like property in Lesamnta

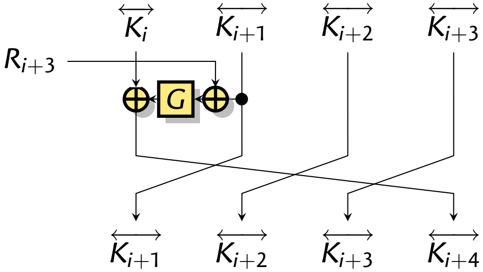
- Can we use this in the key-schedule?



- No, because of the constants
- On the other hand, the constants are almost symmetric...

Complementation-like property in Lesamnta

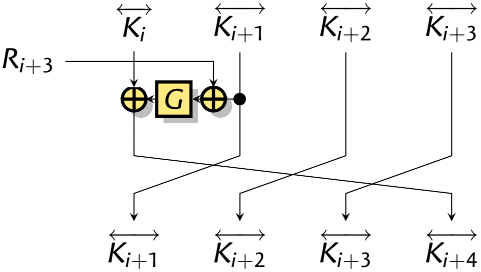
- Can we use this in the key-schedule?



- No, because of the constants
- On the other hand, the constants are almost symmetric...

Complementation-like property in Lesamnta

- Can we use this in the key-schedule?



- No, because of the constants
- On the other hand, the constants are almost symmetric...

Lesamnta's constants

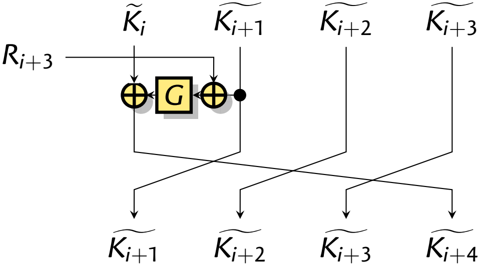
- $R_i = (2i, 2i+1)$
- $R_i \oplus \overleftrightarrow{R_i} = (1,1)$
- Let $\widetilde{(a,b)} = \overleftrightarrow{(a,b)} \oplus (1,1) = (b \oplus 1, a \oplus 1)$
- $\widetilde{R_i} = R_i$

Lesamnta's constants

- $R_i = (2i, 2i+1)$
- $R_i \oplus \overleftrightarrow{R_i} = (1,1)$
- Let $\widetilde{(a,b)} = \overleftrightarrow{(a,b)} \oplus (1,1) = (b \oplus 1, a \oplus 1)$
- $\widetilde{R_i} = R_i$

Complementation-like property in Lesamnta, part II

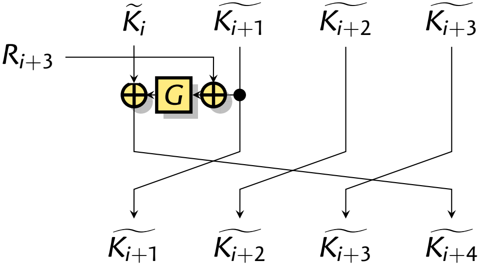
- Can we use this in the key-schedule?



- $\widetilde{K_{i+1}} \oplus R_{i+3} = \overleftrightarrow{K_{i+1} \oplus R_{i+3}}$
- $G(\widetilde{K_{i+1}} \oplus R_{i+3}) = \overleftrightarrow{G(K_{i+1} \oplus R_{i+3})}$
- $\widetilde{K_i} \oplus G(\widetilde{K_{i+1}} \oplus R_{i+3}) = K_i \oplus G(\widetilde{K_{i+1}} \oplus R_{i+3}) = \widetilde{K_{i+4}}$

Complementation-like property in Lesamnta, part II

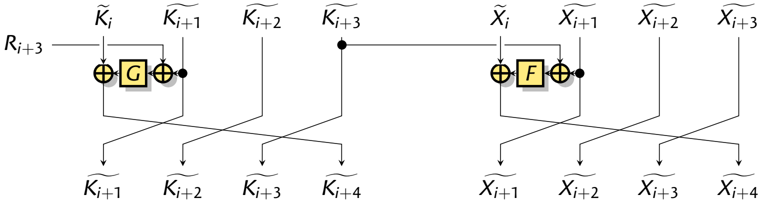
- Can we use this in the key-schedule?



- $\widetilde{K_{i+1}} \oplus R_{i+3} = \overleftrightarrow{K_{i+1} \oplus R_{i+3}}$
- $G(\widetilde{K_{i+1}} \oplus R_{i+3}) = \overleftrightarrow{G(K_{i+1} \oplus R_{i+3})}$
- $\widetilde{K_i} \oplus G(\widetilde{K_{i+1}} \oplus R_{i+3}) = K_i \oplus G(\widetilde{K_{i+1} \oplus R_{i+3}}) = \widetilde{K_{i+4}}$

Complementation-like property in Lesamnta, part II

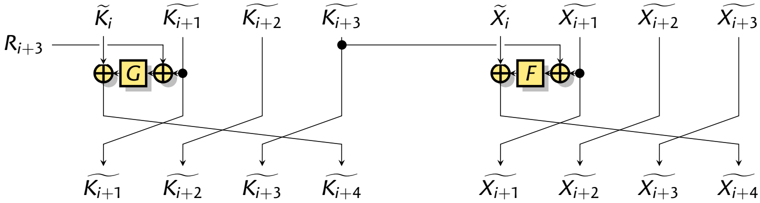
- Can we use this in the full compression function?



- $K_i \to \widetilde{K}_i$
- $\widetilde{X_{i+1}} \oplus \widetilde{K_{i+3}} = \overleftrightarrow{X_{i+1} \oplus K_{i+3}}$
- $F(\widetilde{X_{i+1}} \oplus \widetilde{K_{i+3}}) = \overleftrightarrow{F(X_{i+1} \oplus K_{i+3})}$
- $\widetilde{X}_i \oplus F(\widetilde{X_{i+1}} \oplus \widetilde{K_{i+3}}) = \widetilde{X_i \oplus F(X_{i+1} \oplus K_{i+3})} = \widetilde{X_{i+4}}$

Complementation-like property in Lesamnta, part II

- Can we use this in the full compression function?



- $K_i \rightarrow \widetilde{K_i}$
- $\widetilde{X_{i+1}} \oplus \widetilde{K_{i+3}} = \overleftrightarrow{X_{i+1} \oplus K_{i+3}}$
- $F(\widetilde{X_{i+1}} \oplus \widetilde{K_{i+3}}) = \overleftrightarrow{F(X_{i+1} \oplus K_{i+3})}$
- $\widetilde{X_i} \oplus F(\widetilde{X_{i+1}} \oplus \widetilde{K_{i+3}}) = X_i \oplus F(\widetilde{X_{i+1} \oplus K_{i+3}}) = \widetilde{X_{i+4}}$

Some Really Interesting Property of Lesamnta

- $CF(\widetilde{X}, \widetilde{K}) = \overleftrightarrow{CF(X, K)}$
- If $\widetilde{X} = X$ and $\widetilde{K} = K$, then $\overleftrightarrow{CF(X, K)} = CF(X, K)$
  - The output is in a subspace of size $2^{n/2}$.
- Collision in the compression function in time $2^{n/4}$
- Second-preimage on weak messages
- Improved herding attack
  - $2^{n/2}$ instead of $2^{2n/3}$

Some Really Interesting Property of Lesamnta

- $CF(\widetilde{X}, \widetilde{K}) = \overleftrightarrow{CF(X, K)}$
- If $\widetilde{X} = X$ and $\widetilde{K} = K$, then $\overleftrightarrow{CF(X, K)} = CF(X, K)$
  - The output is in a subspace of size $2^{n/2}$.
- Collision in the compression function in time $2^{n/4}$
- Second-preimage on weak messages
- Improved herding attack
  - $2^{n/2}$ instead of $2^{2n/3}$

Some Really Interesting Property of Lesamnta

- $CF(\widetilde{X}, \widetilde{K}) = \overleftrightarrow{CF(X, K)}$
- If $\widetilde{X} = X$ and $\widetilde{K} = K$, then $\overleftrightarrow{CF(X, K)} = CF(X, K)$
  - The output is in a subspace of size $2^{n/2}$.
- Collision in the compression function in time $2^{n/4}$
- Second-preimage on weak messages
- Improved herding attack
  - $2^{n/2}$ instead of $2^{2n/3}$

XTEA

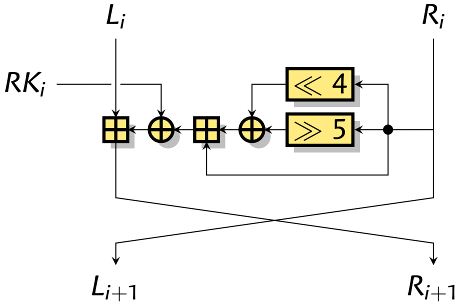
- Lightweight block cipher.
- Successor to TEA with a more complex key schedule to avoid RK.
- Feistel Design
- Implemented in the Linux kernel

David Wheeler, Roger Needham
Tea extensions
Technical report, 1997

XTEA

```
void encipher(int num_rounds, u32 v[2], u32 const k[4]) {
    int i;
    u32 v0=v[0], v1=v[1], sum=0, delta=0x9E3779B9;
    for (i=0; i < num_rounds; i++) {
        v0 += (((v1 << 4) ^ (v1 >> 5)) + v1)
              ^ (sum + k[sum & 3]);
        sum += delta;
        v1 += (((v0 << 4) ^ (v0 >> 5)) + v0)
              ^ (sum + k[(sum>>11) & 3]);
    }
    v[0]=v0; v[1]=v1;
}
```

XTEA



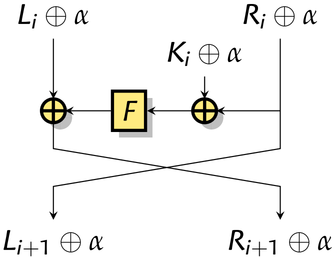
$$L_{i+1} = R_i$$
$$R_{i+1} = L_i \boxplus (F(R_i) \oplus RK_i)$$

- $F(x) = ((x \ll 4) \oplus (x \gg 5)) \boxplus x$

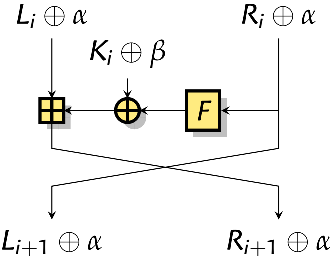
- 128 bit key: K_0, K_1, K_2, K_3
- $RK_{2i} = (\quad i \quad \cdot \delta) \boxplus K_{((i \cdot \delta) \gg 11) \bmod 4}$
- $RK_{2i+1} = ((i+1) \cdot \delta) \boxplus K_{((i+1) \cdot \delta) \bmod 4}$

- 64 rounds

A Simple RK Differential



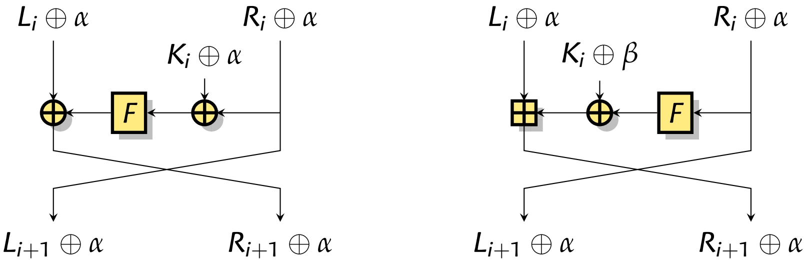
Complementation property



RK iterative differential on XTEA

- $F : \alpha \rightsquigarrow \beta$
- $\alpha = 2^{31}, \beta = 2^{31} + 2^{26}$
- Prob. 1/2.

A Simple RK Differential

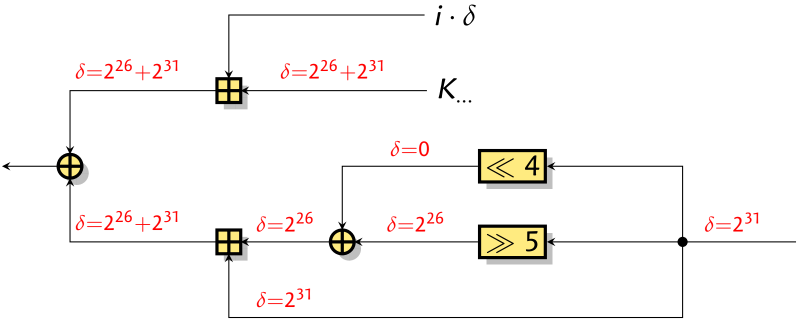


Complementation property

RK iterative differential on XTEA

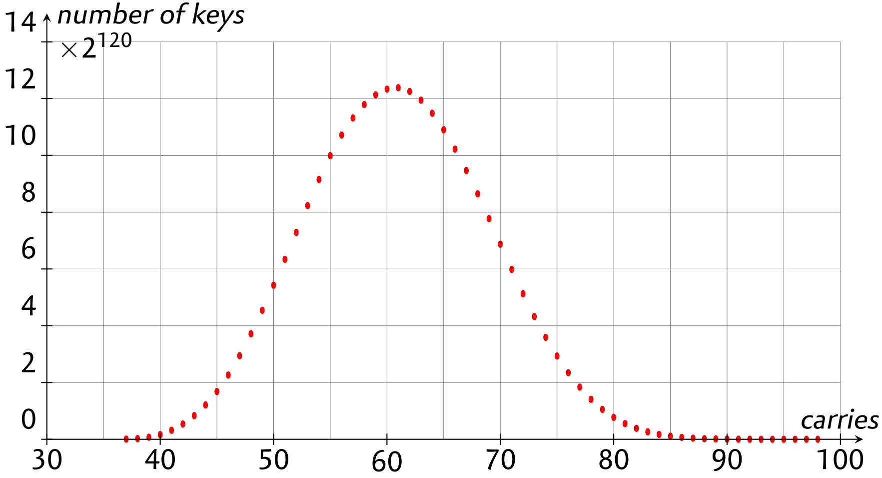
- $F : \alpha \rightsquigarrow \beta$
- $\alpha = 2^{31}, \beta = 2^{31} + 2^{26}$
- Prob. 1/2.

Difference propagation



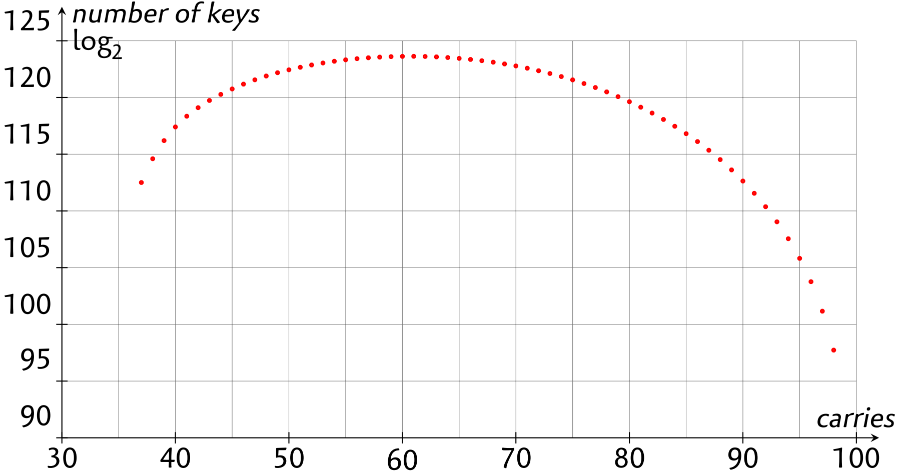
- Modular differences
- With prob. 1 / 3, the XOR-difference is the same
- For a given key, we can compute the XOR-difference
  - $p = 2^{-1}$ if no carries
  - $p = 2^{-1-c}$ if c carries.

Key Distribution



- number of keys with a given prob. (rounds 20–50)
- 48% of the keys have less than 60 carries

Key Distribution



- number of keys with a given prob. (rounds 20–50) ($\log_2$)
- Some keys have only 37 carries

36 Rounds Attack

- Consider rounds 20–55
- Rounds 51–55 only use K_2 and K_3
- Take 2^{62} message pairs
- Partial decrypt by guessing K_2 and K_3
  - If the key is in the 48% weak keys, at least one good pair for 20–50
  - Good pair gives carry pattern
- If it fails, then the key is not in the weak class
  - 52 % of the keyspace remaining.
- Complexity:

Rounds	Data	Time
36	2^{62}	2^{127}
37	$2^{64-\epsilon}$	2^{127}

50 Rounds Attack for Weak Keys

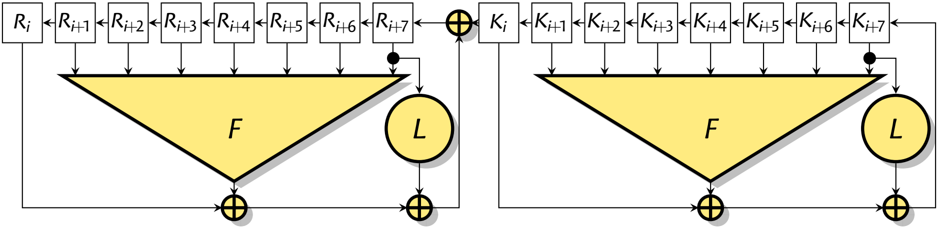
- Consider rounds 10–59
- Rounds 56–59 only use K_0 and K_1
- There is a class of weak keys with 60 carries in 10–55
  - $2^{107.5}$ weak keys out of 2^{128}
- Complexity
  - Data 2^{62}
  - Time 2^{126}
- Recent improvement (WiP)
  - 53 rounds
  - Data 2^{62}
  - Time 2^{99}

ESSENCE

- First round SHA-3 candidate
- Merkle-Damgård with a Davies-Meyer compression function
- Shift-Register based design

Jason Worth Martin
ESSENCE: A Candidate Hashing Algorithm for the NIST Competition
Submission to the NIST SHA-3 competition

ESSENCE



- 32 rounds.
- Message loaded to $K_{-7}, \ldots, K_0$.
- Chaining value loaded to $R_{-7}, \ldots, R_0$.
- F is non-linear bit-wise.
- L is linear based on a LFSR.

Self-similarity property in ESSENCE

- Since L is LFSR based, a rotation can give a slide
  - $\text{LFSR}(x^{\lll 1}) = \text{LFSR}(x)^{\lll 1}$ with prob. $1/4$
- L is the only non-bitwise operation.
  - $\textit{ESSENCE}\text{-round}(R^{\lll 1}, K^{\lll 1}) = \textit{ESSENCE}\text{-round}(R, K)^{\lll 1}$ with prob. $1/4$
- $CF(H^{\lll 1}, M^{\lll 1}) = CF(H, M)^{\lll 1}$ with prob. 2^{-128}
  - We can construct a good pair for a cost of 2^{48}

Conclusion

- Sometimes, a simple relation can go through a function
- The constant are used to avoid this...
  - But sometimes the constants are weak
- Nice properties when the self-similarity relations have fixed points.