Attacking the IETF/ISO Standard for Internal Re-keying CTR-ACPKM

FSE 2023

Orr Dunkelman Shibam Ghosh Eran Lambooij

March 23, 2023

Department of Computer Science, University of Haifa

University of Haifa

Table of contents

Plan of this Section

1. Advanced CryptoPro Key Meshing (ACPKM)
2. Security Issues with the ACPKM Transformation
3. A Related-key Distinguisher on CTR-ACPKM
4. ACPKM is not Misuse Resistant
5. Conclusion

Why Re-Keying?

1. To enhance security, it is common practice to restrict the duration of key use

Why Re-Keying?

1. To enhance security, it is common practice to restrict the duration of key use
2. The encryption key is typically changed after a set amount of encryptions
3. Key lifetime: The maximum amount of data that can be encrypted under a key

Why Re-Keying?

1. To enhance security, it is common practice to restrict the duration of key use
2. The encryption key is typically changed after a set amount of encryptions
3. Key lifetime: The maximum amount of data that can be encrypted under a key
4. Changing a key requires a key-exchange protocol with high computation and communication costs

Why Re-Keying?

1. To enhance security, it is common practice to restrict the duration of key use
2. The encryption key is typically changed after a set amount of encryptions
3. Key lifetime: The maximum amount of data that can be encrypted under a key
4. Changing a key requires a key-exchange protocol with high computation and communication costs
5. Re-keying mechanism: Generating the secret key K_i of the i-th epoch based on the previous keys (suggested by Abdalla and Bellare).

Why Re-Keying?

1. To enhance security, it is common practice to restrict the duration of key use
2. The encryption key is typically changed after a set amount of encryptions
3. Key lifetime: The maximum amount of data that can be encrypted under a key
4. Changing a key requires a key-exchange protocol with high computation and communication costs
5. Re-keying mechanism: Generating the secret key K_i of the i-th epoch based on the previous keys (suggested by Abdalla and Bellare).
6. Types of re-keying mechanisms:
    - The block cipher level (fresh re-keying)
    - The block cipher mode of operation level (internal re-keying)
    - The protocol level (external re-keying)

ACPKM **Internal Re-keying**

- Basic Idea: Call a key update function after encrypting a predefined number of blocks, known as a section
- ACPKM mode was Proposed in CTCrypt'2016
- Counter mode with ACPKM, CTR-ACPKM is Passing through the last formal standardization process in IETF (CFRG)
- Was standardized by ISO (ISO 10116)

ACPKM **Internal Re-keying**

- Basic Idea: Call a key update function after encrypting a predefined number of blocks, known as a section
- ACPKM mode was Proposed in CTCrypt'2016
- Counter mode with ACPKM, CTR-ACPKM is Passing through the last formal standardization process in IETF (CFRG)
- Was standardized by ISO (ISO 10116)

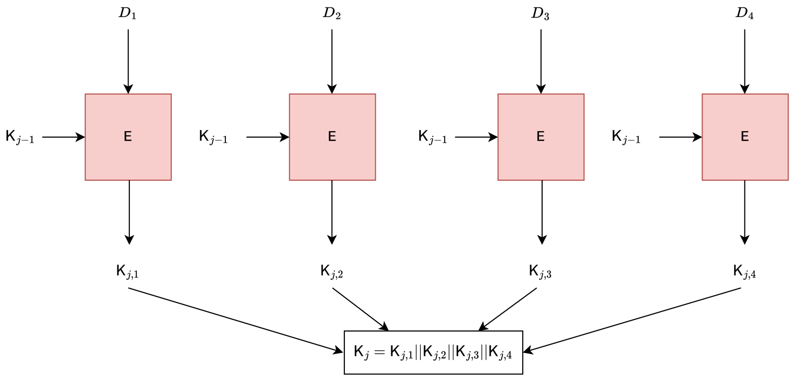
ACPKM method generates a new key in the following way:

$$\mathsf{K}_j = \mathrm{MSB}_\kappa(E_{\mathsf{K}_{j-1}}(D_1)|\cdots|E_{\mathsf{K}_{j-1}}(D_r))$$

where $r = \kappa/n$ and $D_1, D_2, D_3, \ldots, D_r$ are carefully chosen constants

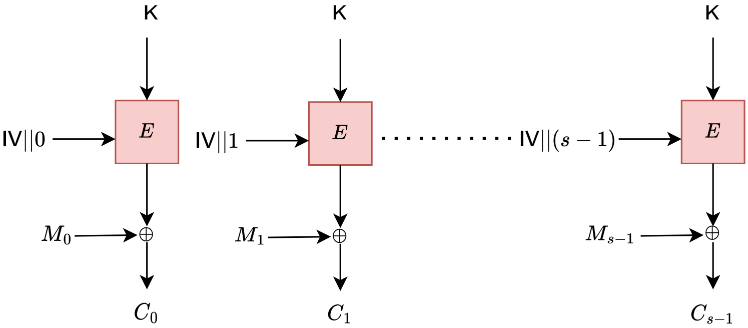
ACPKM **Internal Re-keying**

$\kappa = 4n$



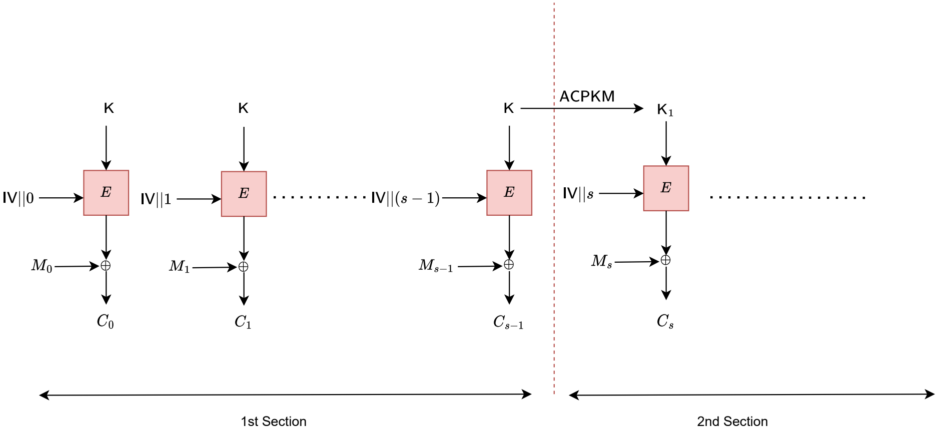
CTR-ACPKM

Section size is s



CTR-ACPKM

Section size is s



Plan of this Section

1. Advanced CryptoPro Key Meshing (ACPKM)
2. Security Issues with the ACPKM Transformation
3. A Related-key Distinguisher on CTR-ACPKM
4. ACPKM is not Misuse Resistant
5. Conclusion

Entropy Loss on ACPKM Transformation

1. ACPKM as a functional graph: Consider the graph $G_{\text{ACPKM}} = (V, E)$, where $V = \{0, 1\}^{\kappa}$ and $E = \{(\mathsf{K}, \text{ACPKM}(\mathsf{K}))\}$
2. A vertex $\mathsf{K} \in V$ is called ν-th iterate image point if $\exists x$ s.t. $(\text{ACPKM})^{\nu}(x) = \mathsf{K}$ (denoted by I^{ν})

Entropy Loss on ACPKM Transformation

1. ACPKM as a functional graph: Consider the graph $G_{\text{ACPKM}} = (V, E)$, where $V = \{0,1\}^\kappa$ and $E = \{(\text{K}, \text{ACPKM}(\text{K}))\}$
2. A vertex $\text{K} \in V$ is called ν-th iterate image point if $\exists x$ s.t. $(\text{ACPKM})^\nu(x) = \text{K}$ (denoted by I^ν)
3. Result on functional graph by Flajolet and Odlyzko: The H_0 entropy of the key-space after s iterations is approximately $\kappa + 1 - \log_2(s)$ where $s \leq 2^{\frac{\kappa}{2}}$

Exhaustive Search For Section Keys

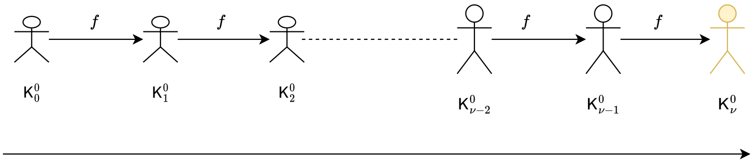
1. Reduced entropy indicates more efficient exhaustive search on later sections

Exhaustive Search For Section Keys

1. Reduced entropy indicates more efficient exhaustive search on later sections
2. K is a valid key for the ν-th section iff $K \in I^{\nu}$

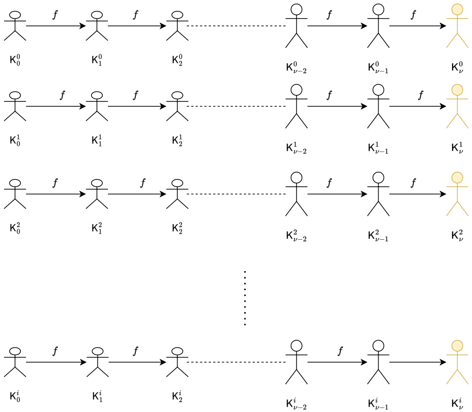
Exhaustive Search For Section Keys

1. Reduced entropy indicates more efficient exhaustive search on later sections
2. K is a valid key for the ν-th section iff $K \in I^{\nu}$



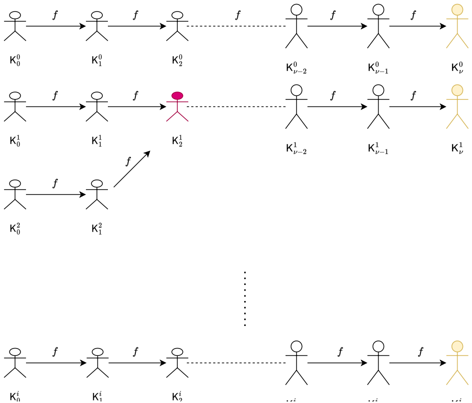
Basic Approach to Find the ν-th Section Keys

A basic approach to find valid section keys for the ν-th section



Basic Approach to Find the ν-th Section Keys

A basic approach to find valid section keys for the ν-th section

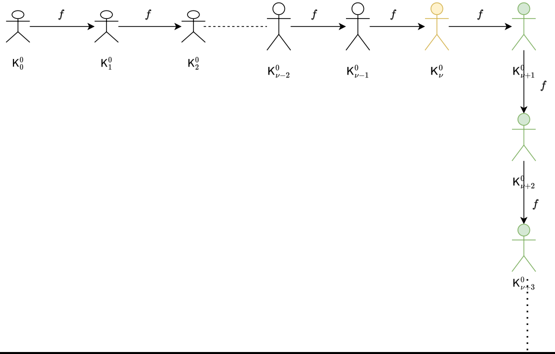


Improved Exhaustive Search

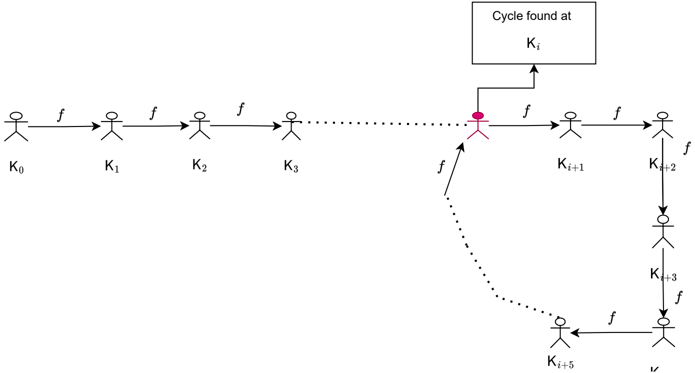
- If $\mathsf{K} \in \mathsf{I}^{\nu}$, then $\exists x$ such that $f^{\nu}(x) = \mathsf{K}$.
- Thus, $f(\mathsf{K}) = f(f^{\nu}(x)) = f^{\nu}(f(x))$.
- So, $f(\mathsf{K})$ is also a valid ν-th section key.

Improved Exhaustive Search

- If $\mathsf{K} \in \mathsf{I}^{\nu}$, then $\exists x$ such that $f^{\nu}(x) = \mathsf{K}$.
- Thus, $f(\mathsf{K}) = f(f^{\nu}(x)) = f^{\nu}(f(x))$.
- So, $f(\mathsf{K})$ is also a valid ν-th section key.



Improved Exhaustive Search



The H_1-Entropy of the ACPKM Transformation

- $P^{\nu}_{\mathsf{K}} = \{x \in \{0,1\}^{\kappa} : f^{\nu}(x) = \mathsf{K}\}$ is the set of master-keys that, after ν sections, can reach the section key K

The H_1-Entropy of the ACPKM Transformation

- $P_K^\nu = \{x \in \{0,1\}^\kappa : f^\nu(x) = K\}$ is the set of master-keys that, after ν sections, can reach the section key K
- The probability that K is a valid *nu*-th key is

$$\Pr_\nu(K) = \frac{|P_K^\nu|}{2^\kappa}$$

The H_1-Entropy of the ACPKM Transformation

- $P_{\mathsf{K}}^{\nu} = \{x \in \{0,1\}^{\kappa} : f^{\nu}(x) = \mathsf{K}\}$ is the set of master-keys that, after ν sections, can reach the section key K
- The probability that K is a valid *nu*-th key is

$$\Pr_{\nu}(\mathsf{K}) = \frac{|P_{\mathsf{K}}^{\nu}|}{2^{\kappa}}$$

- Thus H_1-Entropy or Shannon entropy is

$$H_1(I^{\nu}) = \sum_{\mathsf{K} \in I^{\nu}} \Pr_{\nu}(\mathsf{K}) \log\left(\frac{1}{\Pr_{\nu}(\mathsf{K})}\right)$$

A new observation: The H_1 entropy loss

AES: Key Size = 32, Block Size = 16					
steps	H_0	H_1	$\log_2(\kappa) - H_0$	$\log_2(\kappa) - H_1$	$H_1 - H_0$
0	31.338262	31.172745	0.661738	0.827255	-0.165517
1	30.906223	30.654303	1.093777	1.345697	-0.251920
2	30.581405	30.274630	1.418595	1.725370	-0.306775
3	30.319969	29.974669	1.680031	2.025331	-0.345300
4	30.100699	29.726603	1.899301	2.273397	-0.374096
5	29.911633	29.515048	2.088367	2.484952	-0.396585
6	29.745322	29.330610	2.254678	2.669390	-0.414712
7	29.596806	29.167126	2.403194	2.832874	-0.429680

A new observation: The H_1 entropy loss

Simon: Key Size = 32, Block Size = 16					
steps	H_0	H_1	$\log_2(\kappa) - H_0$	$\log_2(\kappa) - H_1$	$H_1 - H_0$
0	31.338258	31.172739	0.661742	0.827261	-0.165519
1	30.906216	30.654282	1.093784	1.345718	-0.251934
2	30.581411	30.274611	1.418589	1.725389	-0.306800
3	30.319954	29.974645	1.680046	2.025355	-0.345309
4	30.100679	29.726576	1.899321	2.273424	-0.374103
5	29.911625	29.515037	2.088375	2.484963	-0.396588
6	29.745328	29.330618	2.254672	2.669382	-0.414710
7	29.596808	29.167133	2.403192	2.832867	-0.429675

Attack Motivated by H_1-entropy Loss

- Loss of H_1-entropy indicates non-uniform distribution of master-keys among valid section keys

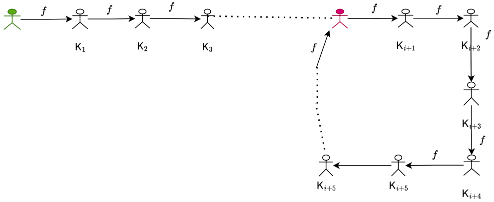
Attack Motivated by H_1-entropy Loss

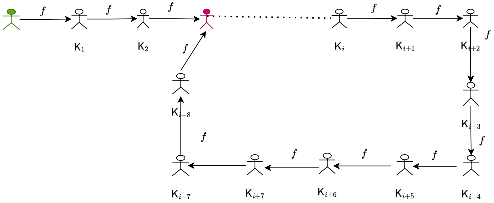
- Loss of H_1-entropy indicates non-uniform distribution of master-keys among valid section keys
- Some section keys cover more master keys than others
- Keys that cover more master keys have a higher probability of being correct ν-th section keys

Attack Motivated by H_1-entropy Loss

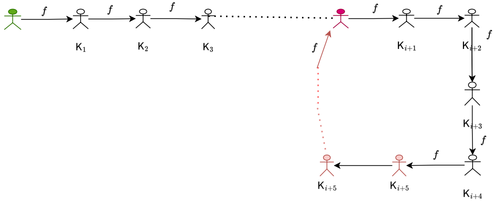
- Loss of H_1-entropy indicates non-uniform distribution of master-keys among valid section keys
- Some section keys cover more master keys than others
- Keys that cover more master keys have a higher probability of being correct ν-th section keys
- We can look for these keys by checking for larger $|P^{\nu}_{\mathsf{K}}|$

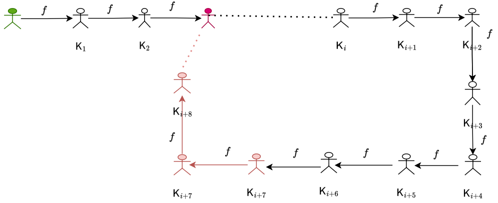
Attack Motivated by H_1-entropy Loss





Attack Motivated by H_1-entropy Loss





AES: Key Size = 32, Block Size = 16, $\nu = 256$

Iteration	Avg. covered key	Avg. computation	Effectiveness	Total covered key
1	$2^{24.40}$	$2^{16.42}$	$2^{7.98}$	$2^{24.40}$
2	$2^{23.71}$	$2^{16.46}$	$2^{7.34}$	$2^{25.09}$
3	$2^{23.12}$	$2^{16.38}$	$2^{6.74}$	$2^{25.42}$
4	$2^{22.64}$	$2^{16.46}$	$2^{6.18}$	$2^{25.61}$
8	$2^{21.98}$	$2^{16.53}$	$2^{5.44}$	$2^{25.99}$
16	$2^{21.19}$	$2^{16.38}$	$2^{4.80}$	$2^{26.50}$
32	$2^{20.78}$	$2^{16.53}$	$2^{4.24}$	$2^{26.99}$
64	$2^{20.35}$	$2^{16.41}$	$2^{3.93}$	$2^{27.49}$
128	$2^{19.76}$	$2^{16.38}$	$2^{3.37}$	$2^{27.89}$
256	$2^{19.44}$	$2^{16.51}$	$2^{2.93}$	$2^{28.33}$
512	$2^{16.69}$	$2^{16.33}$	$2^{0.35}$	$2^{28.82}$

Success Probability of the Attack

- We prove that $E(|\cup_{\mathsf{K}\in\mathcal{K}^{\nu}} P_{\mathsf{K}}^{\nu}|) \geq |\mathcal{K}^{\nu}|\nu$
- A section ν in the range $2^{\kappa/4} \leq \nu < 2^{\kappa/2}$ is expected to cover $2^{3\kappa/4}$ master-keys.
- Thus one iteration suggests an attack with time complexity $2^{\kappa/2}$ and success rate $2^{-\kappa/4}$.

AES: Key Size = 32, Block Size = 16

Section(ν)	Avg. covered key	Avg. computation	Effectiveness
16	$2^{20.49}$	$2^{16.49}$	$2^{3.99}$
32	$2^{21.49}$	$2^{16.46}$	$2^{5.03}$
64	$2^{22.51}$	$2^{16.53}$	$2^{5.97}$
128	$2^{23.39}$	$2^{16.41}$	$2^{6.99}$
256	$2^{24.37}$	$2^{16.39}$	$2^{7.99}$
512	$2^{25.40}$	$2^{16.42}$	$2^{8.99}$

Plan of this Section

1. Advanced CryptoPro Key Meshing (ACPKM)
2. Security Issues with the ACPKM Transformation
3. A Related-key Distinguisher on CTR-ACPKM
4. ACPKM is not Misuse Resistant
5. Conclusion

Related-key Distinguisher

- Consider a CTR-ACPKM instance with section size s
- Suppose the master-key is K

Related-key Distinguisher

- Consider a CTR-ACPKM instance with section size s
- Suppose the master-key is K

- Consider another CTR-ACPKM instance with section size s'
- Choose the master-key $K' = \text{ACPKM}(K)$

Related-key Distinguisher

- Consider a CTR-ACPKM instance with section size s
- Suppose the master-key is K

- Consider another CTR-ACPKM instance with section size s'
- Choose the master-key $\mathsf{K}' = \mathsf{ACPKM}(\mathsf{K})$
- $2s > s' > s$

Related-key Distinguisher

- Consider a CTR-ACPKM instance with section size s
- Suppose the master-key is K
- Choose message-nonce pair $(\mathsf{IV}, \mathsf{M}_1)$
- Let $\text{CTR-ACPKM}(\mathsf{IV}, \mathsf{M}_1) = \mathsf{C}_1$

Related-key Distinguisher

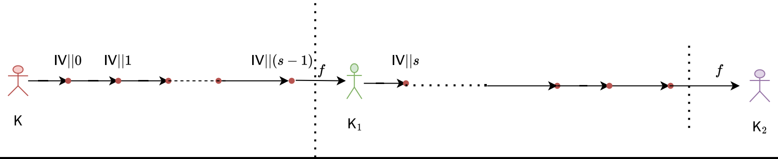
- Consider a CTR-ACPKM instance with section size s
- Suppose the master-key is K
- Choose message-nonce pair $(\mathsf{IV}, \mathsf{M}_1)$
- Let $\text{CTR-ACPKM}(\mathsf{IV}, \mathsf{M}_1) = \mathsf{C}_1$

- Consider another CTR-ACPKM instance with section size s'
- Choose the master-key $\mathsf{K}' = \mathsf{ACPKM}(\mathsf{K})$
- $2s > s' > s$
- Choose message-nonce pair $(\mathsf{IV}, \mathsf{M}_2)$
- Let $\text{CTR-ACPKM}(\mathsf{IV}, \mathsf{M}_2) = \mathsf{C}_2$

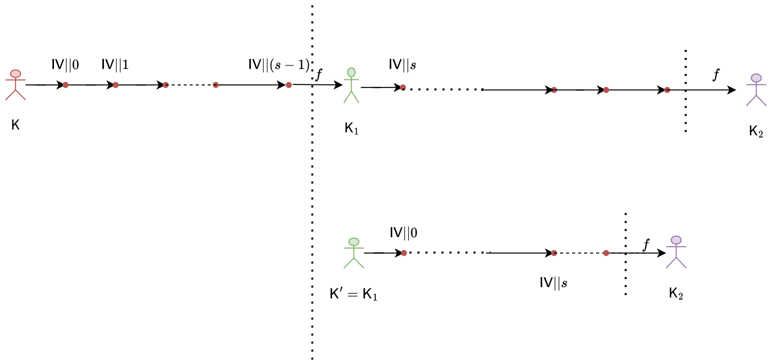
Related-key Distinguisher

- Consider a CTR-ACPKM instance with section size s
- Suppose the master-key is K
- Choose message-nonce pair (IV, M_1)
- Let CTR-ACPKM(IV, M_1) = C_1

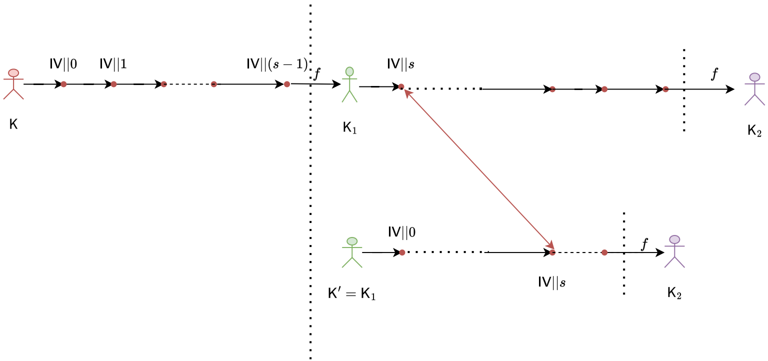
- Consider another CTR-ACPKM instance with section size s'
- Choose the master-key $K' = \text{ACPKM}(K)$
- $2s > s' > s$
- Choose message-nonce pair (IV, M_2)
- Let CTR-ACPKM(IV, M_2) = C_2



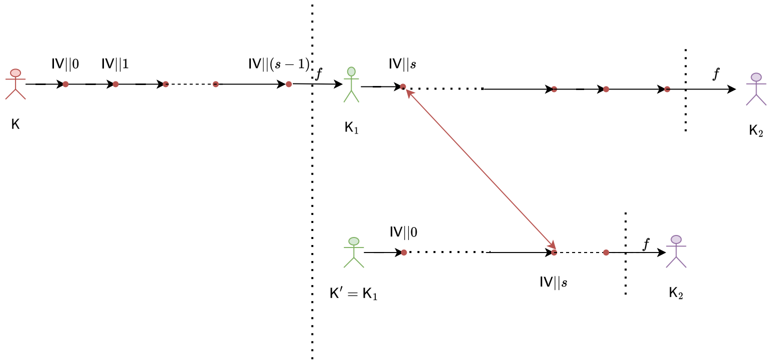
Related-key Distinguisher



Related-key Distinguisher



Related-key Distinguisher



$$E_{K'}(\mathrm{INC}^{s}_{\frac{n}{2}}(\mathrm{IV}\|0^{\frac{n}{2}})) = E_{K_1}(\mathrm{INC}^{s}_{\frac{n}{2}}(\mathrm{IV}\|0^{\frac{n}{2}})) \implies C_1[s] \oplus C_2[s] = M_1[s] \oplus M_2[s]$$

Plan of this Section

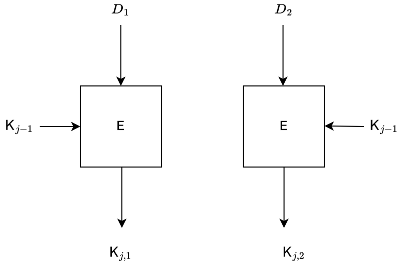
1. Advanced CryptoPro Key Meshing (ACPKM)
2. Security Issues with the ACPKM Transformation
3. A Related-key Distinguisher on CTR-ACPKM
4. ACPKM is not Misuse Resistant
5. Conclusion

CTR-ACPKM **with Weak Block Ciphers**

Consider the case where $\kappa = 2n$

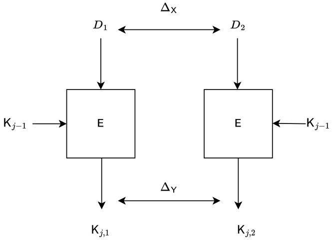
CTR-ACPKM **with Weak Block Ciphers**

Consider the case where $\kappa = 2n$



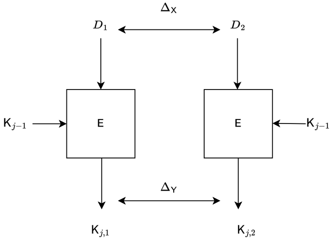
$$\mathsf{K}_j = \mathsf{K}_{j,1} || \mathsf{K}_{j,2}$$

What happens if $\Delta_X \xrightarrow{p} \Delta_Y$?



CTR-ACPKM **with Weak Block Ciphers**

What happens if $\Delta_X \xrightarrow{p} \Delta_Y$?



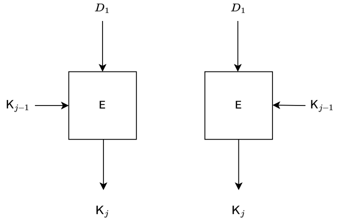
- With probability p, $K_{j,1}||K_{j,2} = K_{j,1}||K_{j,1} \oplus \Delta_Y$
- We find such a output difference by seeing $O(1/p)$ sections in time $O(2^n/p)$

CTR-ACPKM **with Weak Block Ciphers**

What happens if $0 \xrightarrow[\Delta_K]{p} 0$?

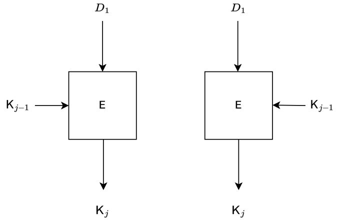
CTR-ACPKM **with Weak Block Ciphers**

What happens if $0 \xrightarrow[\Delta_K]{p} 0$?



CTR-ACPKM **with Weak Block Ciphers**

What happens if $0 \xrightarrow[\Delta_K]{p} 0$?



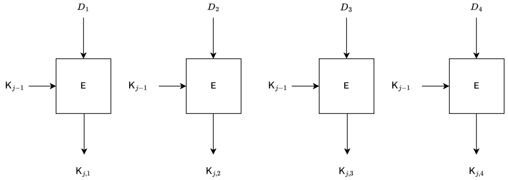
- Key entropy drops by about 0.66 bits in 1st update for a random function
- TEA's related-key properties lead to a drop of almost 2.34 bits in key entropy in the 1st update

CTR-ACPKM **with Weak Block Ciphers**

- For the case where $\kappa = 4n$, we get even better attack

CTR-ACPKM **with Weak Block Ciphers**

- For the case where $\kappa = 4n$, we get even better attack
- Here we can choose $\binom{4}{2}$ pairs from $\{D_1, D_2, D_3, D_4\}$

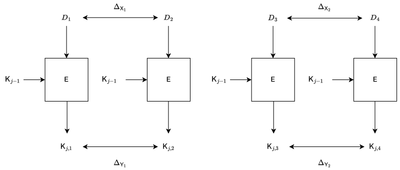


CTR-ACPKM **with Weak Block Ciphers**

What happens if $\Delta_{X_1} \xrightarrow{p_1} \Delta_{Y_1}$ and $\Delta_{X_2} \xrightarrow{p_2} \Delta_{Y_2}$

CTR-ACPKM **with Weak Block Ciphers**

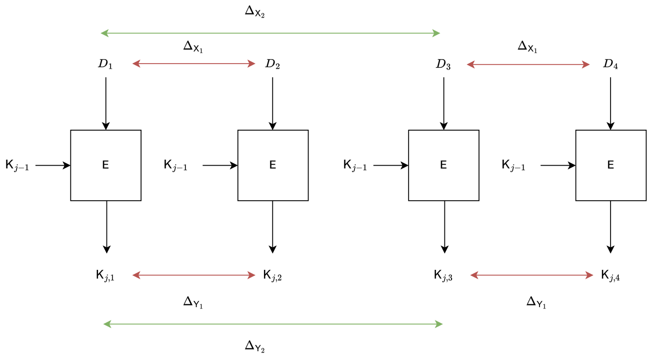
What happens if $\Delta_{X_1} \xrightarrow{p_1} \Delta_{Y_1}$ and $\Delta_{X_2} \xrightarrow{p_2} \Delta_{Y_2}$



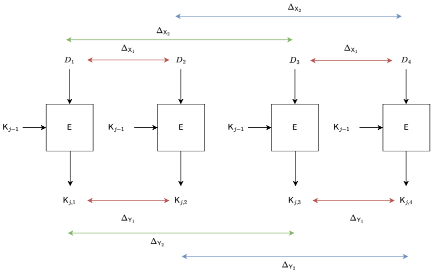
With probability $p_1 p_2$, the section key

$$K_j = K_{j,1}||K_{j,2}||K_{j,3}||K_{j,4} = K_{j,1}||K_{j,1} \oplus \Delta_{Y_1}||K_{j,3}||K_{j,3} \oplus \Delta_{Y_2}$$

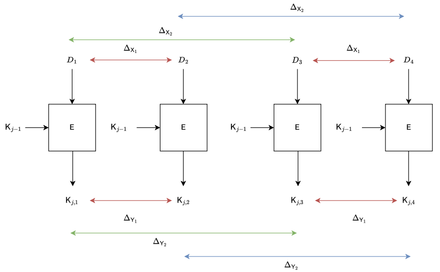
CTR-ACPKM **with Weak Block Ciphers**



CTR-ACPKM **with Weak Block Ciphers**

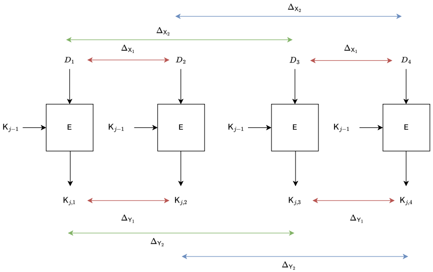


CTR-ACPKM with Weak Block Ciphers



- $K_j = K_{j,1} || K_{j,1} \oplus \Delta_{Y_1} || K_{j,3} || K_{j,3} \oplus \Delta_{Y_1}$ with probability p_1^2
- $K_j = K_{j,1} || K_{j,2} || K_{j,1} \oplus \Delta_{Y_2} || K_{j,2} \oplus \Delta_{Y_2}$ with probability p_2^2

CTR-ACPKM with Weak Block Ciphers



- $K_j = K_{j,1} || K_{j,1} \oplus \Delta_{Y_1} || K_{j,3} || K_{j,3} \oplus \Delta_{Y_1}$ with probability p_1^2
- $K_j = K_{j,1} || K_{j,2} || K_{j,1} \oplus \Delta_{Y_2} || K_{j,2} \oplus \Delta_{Y_2}$ with probability p_2^2
- We note that, in RFC 8645: $D_1 \oplus D_2 = D_3 \oplus D_4$, $D_1 \oplus D_3 = D_2 \oplus D_4$ and $D_1 \oplus D_4 = D_2 \oplus D_3$.

Plan of this Section

1. Advanced CryptoPro Key Meshing (ACPKM)
2. Security Issues with the ACPKM Transformation
3. A Related-key Distinguisher on CTR-ACPKM
4. ACPKM is not Misuse Resistant
5. Conclusion

Conclusion

1. Attacks based on H_0-entropy loss
   - Proposed an improved exhaustive search for the section keys
   - Key collision attack in the multi-user setting
   - Key-recovery attack in the multi-user setting

Conclusion

1. Attacks based on H_0-entropy loss
   - Proposed an improved exhaustive search for the section keys
   - Key collision attack in the multi-user setting
   - Key-recovery attack in the multi-user setting
2. Importance of H_1-entropy loss
   - H_1-entropy loss is much more effective than H_0-entropy loss
   - Proposed a novel master-key recovery attack based on H_1-entropy loss

Conclusion

1. Attacks based on H_0-entropy loss
   - Proposed an improved exhaustive search for the section keys
   - Key collision attack in the multi-user setting
   - Key-recovery attack in the multi-user setting
2. Importance of H_1-entropy loss
   - H_1-entropy loss is much more effective than H_0-entropy loss
   - Proposed a novel master-key recovery attack based on H_1-entropy loss
3. Related-key distinguisher on the CTR-ACPKM mode
   - Independent of the underlying primitive

Conclusion

1. Attacks based on H_0-entropy loss
   - Proposed an improved exhaustive search for the section keys
   - Key collision attack in the multi-user setting
   - Key-recovery attack in the multi-user setting
2. Importance of H_1-entropy loss
   - H_1-entropy loss is much more effective than H_0-entropy loss
   - Proposed a novel master-key recovery attack based on H_1-entropy loss
3. Related-key distinguisher on the CTR-ACPKM mode
   - Independent of the underlying primitive
4. Attacks based on faulty or backdoored implementations of CTR-ACPKM
   - A malicious designer may further harm the mode
   - Attacks based on specific related-key differential property

Recommendations for the Use of ACPKM

1. Using ACPKM without changes can be acceptable in some cases:
   - Large initial key size
   - Implementation issues addressed
   - Appropriate warnings should be added to standards if still used

Recommendations for the Use of ACPKM

1. Using ACPKM without changes can be acceptable in some cases:
   - Large initial key size
   - Implementation issues addressed
   - Appropriate warnings should be added to standards if still used
2. Russian standards GOST 28147-89 (Magma) and Kuznyechik suggested for the use with ACPKM and CPKM
   - GOST has several related key differential properties
   - Multiple works suggest hidden design rationale in Kuznyechik
   - Design rationale of these ciphers is unknown

See the paper for other attacks...

Thank You for your attention!

Any questions?