A Framework for Automated Biclique Cryptanalysis of Block Ciphers

F. Abed C. Forler E. List S. Lucks J. Wenzel

Bauhaus-Universität Weimar

FSE 2013, Singapore

13.03.2013

Biclique Cryptanalysis

- Biclique = complete bipartite graph, connecting each in a set of starting states $\mathcal{S}$ with each in a set of ending states $\mathcal{C}$ over a sub-cipher
- Introduced by Khovratovich, Rechberger, and Savelieva [KRS11] as formalization of initial structures in splice-and-cut MitM attacks
- First used for preimage attacks on round-reduced SHA-2, Skein and their compression functions
- Adapted for key-recovery attacks on the AES by Bogdanov, Khovratovich and Rechberger [BKR11]

Biclique Cryptanalysis

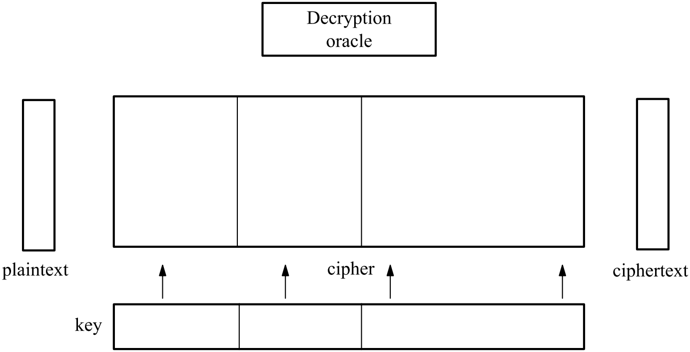
- Many more key-recovery attacks followed since then
  - on SQUARE by Mala [Mal11]
  - on ARIA-256 by Chen and Xue [CX12]
  - on Piccolo by Wang *et al.* [WWY12]
  - on IDEA by Khovratovich, Leurent, and Rechberger [KLR12]
  - HIGHT [HKK11], TWINE by Çoban *et al.* [cKOB12], L-Block by Wang *et al.* [WWYZ12], PRESENT and LED by Jeong *et al.* [JKL+12], KLEIN-64 by Ahmadian *et al.* [ASA13]
- Several approaches and improvements
  - Independent and long bicliques [KRS11, BKR11], probabilistic bicliques [KLR12], bicliques for permutations [Kho12]

Motivation

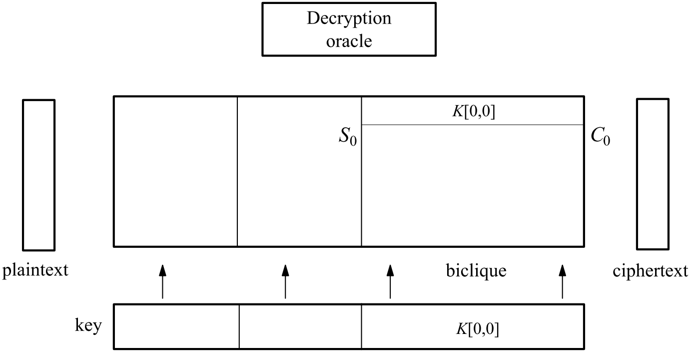
- Initial aim to completely understand the attacks by Bogdanov *et al.*
- Small framework to help the cryptanalyst to find independent bicliques of maximal length
- Consider independent bicliques: generic, independency of differentials = formalized criterion to test

Agenda

Biclique Cryptanalysis – Brief Recall

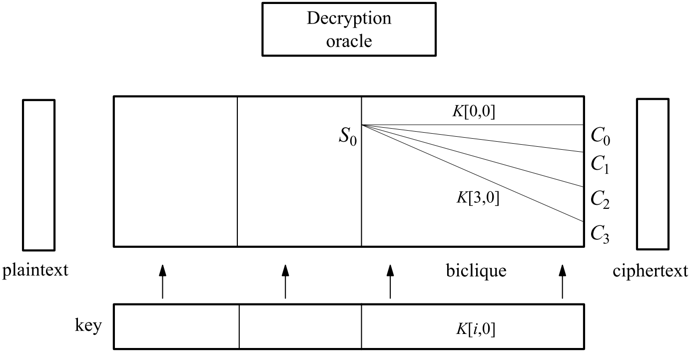


- Given a primitive E, define splitting as in splice-and-cut attack, *e.g.*, $E = \mathcal{B} \circ E_2 \circ E_1$
- Construct biclique around starting state, here over $\mathcal{B}$



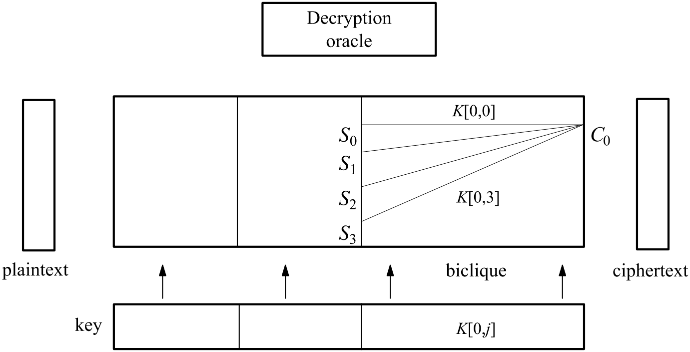
- Choose a *base computation* $\{S_0, K[0,0], C_0\}$:

$$S_0 \xrightarrow[\mathcal{B}]{K[0,0]} C_0$$



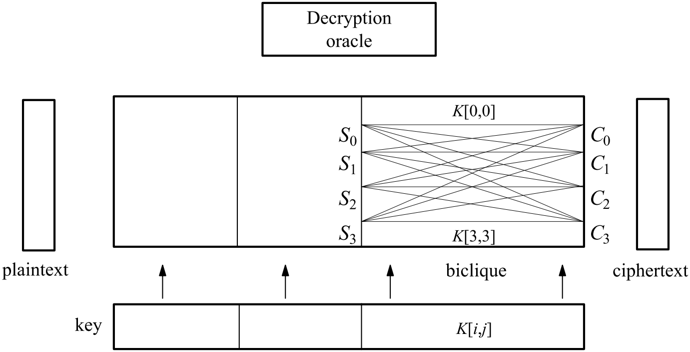
- Find 2^d good (forward) Δ_i-differentials, and compute 2^d times:

$$S_0 \xrightarrow[\mathcal{B}]{K[i,0]} C_i \quad \equiv \quad S_0 \xrightarrow[\mathcal{B}]{K[0,0] \oplus \Delta_i^K} C_0 \oplus \Delta_i$$



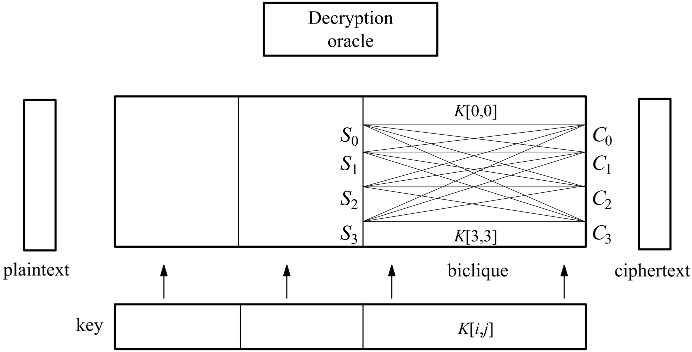
- Find 2^d good (backward) ∇_j-differentials, and compute 2^d times:

$$S_j \xleftarrow[\mathcal{B}]{K[0,j]} C_0 \quad \equiv \quad S_0 \oplus \nabla_j \xleftarrow[\mathcal{B}]{K[0,0] \oplus \nabla_j^K} C_0$$

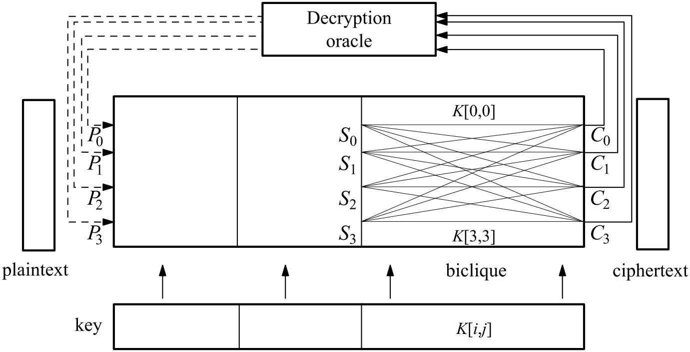


- If the trails are *independent* (do not share active non-linear operations), it applies $\forall i, j \in \{0, \ldots, 2^d - 1\}$:

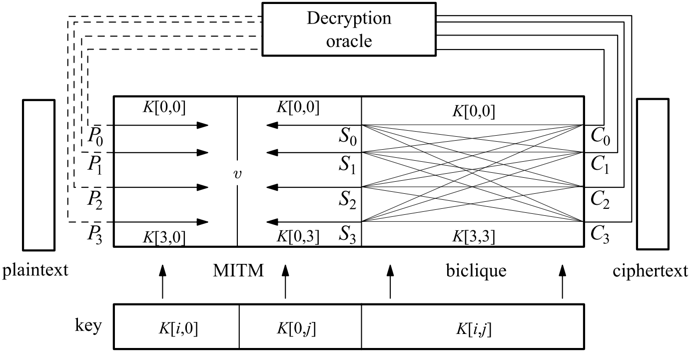
$$S_j \xrightarrow[\mathcal{B}]{K[i,j]} C_i \quad \equiv \quad S_0 \oplus \nabla_j \xrightarrow[\mathcal{B}]{K[0,0] \oplus \Delta_i^K \oplus \nabla_j^K} C_0 \oplus \Delta_i$$



- Test 2^{2d} keys with only $2 \cdot 2^d$ computations in the biclique

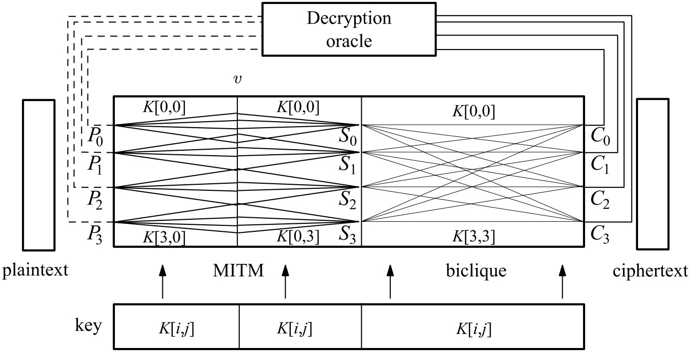


- For 2^d ciphertexts C_i, request the corresponding plaintexts P_i from an oracle



- Compute and store 2^d values $v_{i,0}$ in forward direction
- Compute and store 2^d values $v_{0,j}$ in backward direction

$$\forall i: \quad P_i \xrightarrow[E_1]{K[i,0]} \overrightarrow{v_{i,0}} \quad \text{and} \quad \forall j: \quad \overleftarrow{v_{0,j}} \xleftarrow[E_2^{-1}]{K[0,j]} S_j.$$



- For remaining $2^{2d} - 2 \cdot 2^d$ key candidates $K[i,j]$, only recompute the parts, where the trails with $K[i,j]$ differ from those with $K[i,0]$ or $K[0,j]$

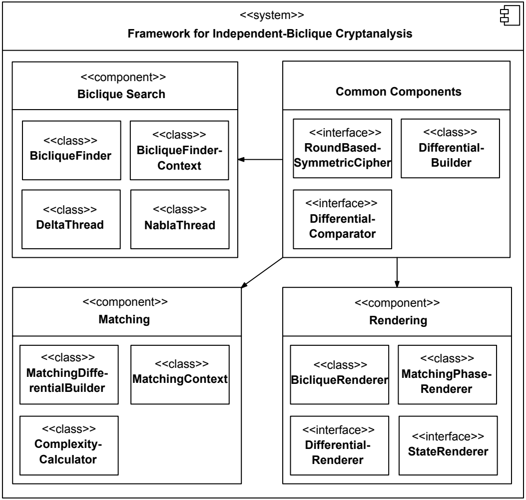
$$\forall i,j \neq 0: \quad P_i \xrightarrow[E_1]{K[i,j]} \overrightarrow{v_{i,j}} \quad \text{and} \quad \overleftarrow{v_{i,j}} \xleftarrow[E_2^{-1}]{K[i,j]} S_j.$$

Relevance

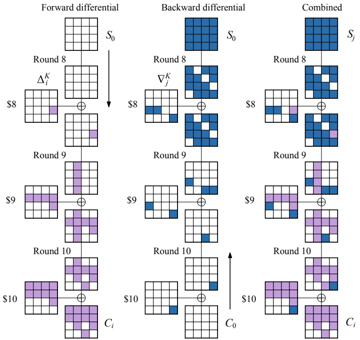
- Low computational advantage if using exhaustive matching-with-precomputations, usually factor of 2-16
- "Bruteforce-like cryptanalysis is not able to conclude that a particular target has a cryptanalytic weakness" (Jia, Rechberger, and Wang [JRW11])
- More general, to derive a lower computational bound for individiual ciphers

Our Framework

Structure


<<system>>
Framework for Independent-Biclique Cryptanalysis
<<component>>
Biclique Search
<<class>> BicliqueFinder
<<class>> BicliqueFinder-Context
<<class>> DeltaThread
<<class>> NablaThread
Common Components
<<interface>> RoundBased-SymmetricCipher
<<class>> Differential-Builder
<<interface>> Differential-Comparator
<<component>>
Matching
<<class>> MatchingDiffe-rentialBuilder
<<class>> MatchingContext
<<class>> Complexity-Calculator
<<component>>
Rendering
<<class>> BicliqueRenderer
<<class>> MatchingPhase-Renderer
<<interface>> Differential-Renderer
<<interface>> StateRenderer

Biclique Search



- Finding a pair of differentials (Δ_i, ∇_j), which share no active components in non-linear operations

Biclique Search (cont'd)

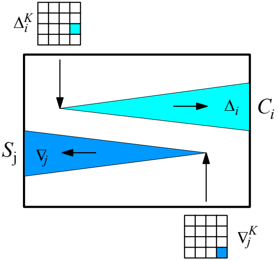
Number of possible differentials

- Example: for a key size $k = 128$ bits and a biclique dimension $d = 8$, one could test

$$\binom{k}{d} = \frac{k!}{d!(k-d)!} = \binom{128}{8} \approx 1.43 \cdot 10^{12}$$

- Reduce time and memory complexity by considering nibble- or byte-wise operating primitives
- **Nibble-wise** primitives: $\binom{\lceil k/4 \rceil}{\lceil d/4 \rceil} = \binom{32}{2} = 496$
- **Byte-wise** primitives: $\binom{\lceil k/8 \rceil}{\lceil d/8 \rceil} = \binom{16}{1} = 16$

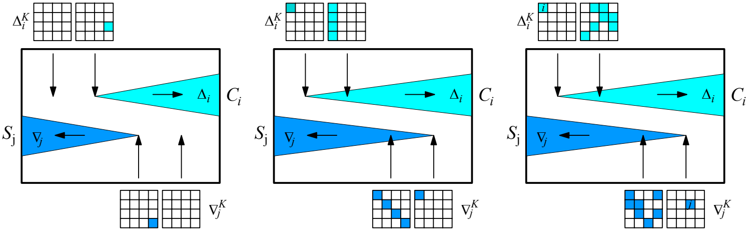
How to Insert Key Differences



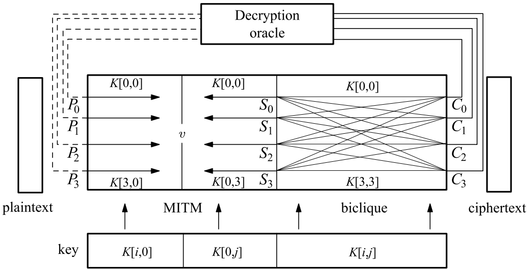
- Affect as little parts of the state as possible
- $\Rightarrow$ inject sub-key differences with least possible hamming weight at the beginning of Δ- and at the end of ∇-differentials
- If $|k| > n$, regard k consecutive sub-key bits as starting key difference

How to Insert Key Differences (cont'd)

1. Inject difference in minimum number of bit/byte/nibbles
2. Inject equal difference in more bit/byte/nibbles in the hope of canceling out in the round transformation
3. Provide option to use more sophisticated custom differences, leave specification to user since testing all possibilities is infeasible



Matching



- All rounds and parts of the state are tested to identify a splitting point v between E_1 and E_2 for a matching with minimum number of bits/bytes/nibbles to recompute

General

Properties

- Compute and store Δ-differentials, compute ∇-differentials and test each pair for independency
- If stored Δ-differentials do not fit in memory, the biclique search is performed in iterations
- Round-wise encryption/decryption necessary
- To inject sub-key differences, one needs invertible key schedule (applies for AES-like ciphers, many lightweight ciphers etc.)
- For others, secret-key differences are used as fallback
- $\Rightarrow$ provide interface for ciphers implementations

Usage

- Two applications as entry points for biclique search and matching
- Biclique search takes as arguments:
  - target cipher
  - strategy to build starting key differences
  - cipher-dependent strategy to locate non-linear operations in order to test differentials
  - biclique dimension
  - maximum number of tested rounds
- Matching arguments:
  - target cipher
  - serialized biclique
- Biclique and matching sequence are rendered as PDF
- Resulting computational complexity is output to the user

Our Results

Primitive	Rounds	Biclique rounds	Computational complexity	Data complexity	Memory complexity
AES-128	10 (full)	3	$2^{126.72}$	2^{72}	2^{8}
AES-192	12 (full)	4	$2^{190.28}$	2^{48}	2^{8}
AES-256	14 (full)	4	$2^{254.53}$	2^{64}	2^{8}
BKSQ-96	10 (full)	3	$2^{94.94}$	2^{80}	2^{8}
BKSQ-144	14 (full)	4	$2^{143.03}$	2^{80}	2^{8}
BKSQ-192	18 (full)	5	$2^{191.00}$	2^{96}	2^{8}
LED-64	30/32	7	$2^{63.03}$	2^{56}	2^{8}
LED-128	48 (full)	12	$2^{127.23}$	2^{64}	2^{8}
KHAZAD	8 (full)	3	$2^{127.28}$	2^{64}	2^{8}
PRESENT-80	25 (full)	4	$2^{79.45}$	2^{60}	2^{8}
PRESENT-128	31 (full)	4	$2^{127.37}$	2^{44}	2^{8}
KLEIN-64	12 (full)	2	$2^{63.08}$	2^{32}	2^{8}
KLEIN-80	16 (full)	3	$2^{79.18}$	2^{40}	2^{8}
KLEIN-96	20 (full)	3	$2^{95.18}$	2^{32}	2^{8}
$PRINCE_{core}$	10 (full)	1	$2^{62.72}$	2^{40}	2^{8}

Previous Results

Primitive	Rounds	Data complexity (Texts)	Computations /Success rate (Encryptions)	Memory complexity (Texts)	Biclique rounds	Ref.
AES-128	8/10	$2^{126.33}$	$2^{124.97}$	2^{102}	5	[BKR11]
	8/10	2^{127}	$2^{125.64}$	2^{32}	5	[BKR11]
	8/10	2^{88}	$2^{125.34}$	2^{8}	3	[BKR11]
	10 (full)	2^{88}	$2^{126.18}$	2^{8}	3	[BKR11]
AES-192	9/12	2^{80}	$2^{188.8}$	2^{8}	4	[BKR11]
	12 (full)	2^{80}	$2^{190.164}$	2^{8}	4	[BKR11]
AES-256	9/14	2^{120}	$2^{253.1}$	2^{8}	6	[BKR11]
	9/14	2^{120}	$2^{251.92}$	2^{8}	4	[BKR11]
	14 (full)	2^{40}	$2^{254.42}$	2^{8}	4	[BKR11]
SQUARE	8 (full)	2^{48}	$2^{125.9}$	2^{8}	2	[Mal11]
ARIA-256	16(full)	2^{80}	$2^{255.2}$	n. a	2	[CX12]
Piccolo-80	25 (full)	2^{48}	$2^{78.95}$	n. a.	6	[WWY12]
Piccolo-128	28/31	2^{24}	$2^{126.79}$	n. a	6	[WWY12]
IDEA	7.5/8.5	2^{52}	$2^{123.9}$	2^{7}	1.5	[KLR12]
	8.5 (full)	2^{52}	$2^{126.06}$	2^{3}	1.5	[KLR12]
	8.5 (full)	2^{59}	$2^{125.97}$	2^{3}	1.5	[KLR12]
HIGHT	32 (full)	—	$2^{126.4}$	—	8	[HKK11]
TWINE-80	36 (full)	2^{60}	$2^{79.10}$	2^{8}	8	[cKOB12]
TWINE-128	36 (full)	2^{60}	$2^{126.82}$	2^{8}	11	[cKOB12]
L-Block	32 (full)	2^{52}	$2^{78.40}$	2^{4}	8	[WWYZ12]
KLEIN-64	12 (full)	2^{39}	$2^{62.84}$	$2^{4.5}$	3	[ASA13]

End

Questions?

Zahra Ahmadian, Mahmoud Salmasizadeh, and Mohammad Reza Aref.
Biclique Cryptanalysis of the Full-Round KLEIN Block Cipher.
Cryptology ePrint Archive, Report 2013/097, 2013.
http://eprint.iacr.org/.

Andrey Bogdanov, Dmitry Khovratovich, and Christian Rechberger.
Biclique Cryptanalysis of the Full AES.
In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT*, volume 7073 of *Lecture Notes in Computer Science*, pages 344–371. Springer, 2011.

Mustafa Çoban, Ferhat Karakoç, and Özkan Boztaş.
Biclique Cryptanalysis of TWINE.
Cryptology ePrint Archive, Report 2012/422, 2012.
http://eprint.iacr.org/.

Shaozhen Chen and Tianmin Xu.
Biclique Attack of the Full ARIA-256.
IACR Cryptology ePrint Archive, 2012:11, 2012.

Deukjo Hong, Bonwook Koo, and Daesung Kwon.
Biclique Attack on the Full HIGHT.
In Howon Kim, editor, *ICISC*, volume 7259 of *Lecture Notes in Computer Science*, pages 365–374. Springer, 2011.

Kitae Jeong, HyungChul Kang, Changhoon Lee, Jaechul Sung, and Seokhie Hong.
Biclique Cryptanalysis of Lightweight Block Ciphers PRESENT, Piccolo and LED.
IACR Cryptology ePrint Archive, 2012:621, 2012.

Keting Jia, Christian Rechberger, and Xiaoyun Wang.
Green Cryptanalysis: Meet-in-the-Middle Key-Recovery for the Full KASUMI Cipher.
Cryptology ePrint Archive, Report 2011/466, 2011.
http://eprint.iacr.org/.

Dmitry Khovratovich.
Bicliques for Permutations: Collision and Preimage Attacks in Stronger Settings.
In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT*, volume 7658 of *Lecture Notes in Computer Science*, pages 544–561. Springer, 2012.

Dmitry Khovratovich, Gaëtan Leurent, and Christian Rechberger.
Narrow-Bicliques: Cryptanalysis of Full IDEA.
In *EUROCRYPT*, pages 392–410, 2012.

Dmitry Khovratovich, Christian Rechberger, and Alexandra Savelieva.
Bicliques for Preimages: Attacks on Skein-512 and the SHA-2 Family.
Cryptology ePrint Archive, Report 2011/286, 2011.
http://eprint.iacr.org/.

Hamid Mala.
Biclique Cryptanalysis of the Block Cipher SQUARE.
Cryptology ePrint Archive, Report 2011/500, 2011.
http://eprint.iacr.org/.

Yanfeng Wang, Wenling Wu, and Xiaoli Yu.
Biclique Cryptanalysis of Reduced-Round Piccolo Block Cipher.
In Mark Dermot Ryan, Ben Smyth, and Guilin Wang, editors, *ISPEC*, volume 7232 of *Lecture Notes in Computer Science*, pages 337–352. Springer, 2012.

Yanfeng Wang, Wenling Wu, Xiaoli Yu, and Lei Zhang.
Security on LBlock against Biclique Cryptanalysis.
In Dong Hoon Lee and Moti Yung, editors, *WISA*, volume 7690 of *Lecture Notes in Computer Science*, pages 1–14. Springer, 2012.