Automatic Cryptanalysis of Block Ciphers with CP
A case study: related key differential cryptanalysis

David Gerault

LIMOS, University Clermont Auvergne
This presentation is inspired by 4 papers written with Pascal Lafourcade, Marine Minier, Christine Solnon, Siwei Sun, Qianqian Yang, Yosuke Todo, Kexin Qiao, Lei Hu

Summer school on Real World Crypto
Keyed permutation $E: \{0,1\}^K \times \{0,1\}^P \rightarrow \{0,1\}^P$. Generally simple function iterated $n$ times.

Expected Property

Indistinguishable from a random permutation if $K$ is unknown
Attacking a block cipher

The attacker can encrypt messages of his choice and tries to recover the hidden key $K$. 

\[ f \overset{?}{=} E \text{ or random permutation } \pi? \]

Distinguishing from $\pi \equiv$ recovering $K$
The attacker choses $\delta K$ (but $K$ remains hidden)
- Allowed by certain protocol/real life applications
- A block cipher should be secure in the related key model
- **The best published attacks against AES are related key**
Related Key Attack

\[ X \xrightarrow{f_K} C \]
\[ X' = X \oplus \delta X \xrightarrow{f_K \oplus \delta K} C' \]
\[ \delta C? \]

Distribution of \( \delta C \) for chosen \( \delta X, \delta K \) and random \( X \) and \( K \)...

If \( f = \pi \) ?
If \( f = E \) ?

David Gerault (LIMOS, University Clermont Auvergne) This presentation is inspired by 4 papers written with Pascal Lafourcade, Marine Minier, Christine Solnon, Siwei Sun, Qianqian Yang, Yosuke Todo, Kexin Qiao, Lei Hu

Automatic Cryptanalysis of Block Ciphers with CP Summer school on Real World Crypto
Related Key Attack

\[ X' = X \oplus \delta X \]

\[ f_K \oplus \delta K \]

\[ C' \]

Distribution of \( \delta C \) for chosen \( \delta X, \delta K \) and random \( X \) and \( K \)...

If \( f = \pi \) ? Uniform
If \( f = E \) ? Not uniform!

Distinguishing attack

The attacker requires many encryptions with input difference \( \delta X, \delta K \) and observes whether there is a bias in the distribution of \( \delta C \)...
Differential characteristics

The higher the bias $\text{Pr}[(\delta X, \delta K) \rightarrow \delta C]$, the better the attack!

Differential characteristics (i.e. propagation patterns $(\delta X, \delta K) \rightarrow \delta C$) with optimal probability are needed, but difficult to find!

- Fix $\delta X, \delta K$
- Apply known propagation rules to obtain the most likely $\delta C$
We did it! With CP

Holy Grail

“Constraint programming represents one of the closest approaches computer science has yet made to the holy grail of programming: the user states the problem, the computer solves it.” (E. Freuder)
Variables
Define variables on given domains
- $[23..42]$ \( x \)
- bool \( y \)
- array $[1..N,1..M]$ of floats $\delta$ ...

Constraints
Define relations between these variables as constraints
- \( x + y < 5 \)
- \( \text{sum(AllVariables)} = 10 \)
- Table: list of allowed tuples \((a, b, c) \in \{(2, 3, 4), (1, 7, 2)\}\)

Objective function
(optional) Define an objective function to optimize
- Maximize($\text{Sum}(\delta)$)
Why another automatic tool?

Other automatic tools exist

- SAT
- Mixed Integer Linear Programming (MILP)
- ...

Question: Why yet another one?
Why another automatic tool?

Other automatic tools exist

- SAT  Boolean variables
- Mixed Integer Linear Programming (MILP)  Linear inequalities
- ...

**Question: Why yet another one?**
**Response: Generalization!**

**CP**

- No limitations on variables nor constraints
- Uses algorithms from the other methods
- There exist tools translating from CP to the others
Related Work & Contributions: AES

Standard since 2000

Problem
Finding optimal RK differential characteristics on AES-128, AES-192 and AES-256

Previous work
- **Biryukov et al., 2010**: Branch & Bound → Several hours (AES-128), several weeks (AES-192)
- **Fouque et al., 2013**: Graph traversal → 30 minutes, 60 Gb memory, 12 cores (AES-128)
Related Work & Contributions: AES

Standard since 2000

Problem
Finding optimal RK differential characteristics on AES-128, AES-192 and AES-256

Previous work
- Biryukov et al., 2010: Branch & Bound
  → Several hours (AES-128), several weeks (AES-192)
- Fouque et al., 2013: Graph traversal
  → 30 minutes, 60 Gb memory, 12 cores (AES-128)

Our results
- 25 minutes (AES-128), 24 hours (AES-192), 30 minutes (AES-256)
- New (better) differential characteristics on all versions
- Disproved incorrect one found in previous work
Related Work & Contributions: Midori

Lightweigh block cipher, 2015

Problem
Finding optimal RK differential characteristics on Midori-64 and Midori-128

Previous work
- Midori-64: Dong, 2016: Custom algorithm → 14 rounds (out of 16), $2^{116}$ operations
- Midori-128: Not done
Related Work & Contributions: Midori

Lightweight block cipher, 2015

Problem
Finding optimal RK differential characteristics on Midori-64 and Midori-128

Previous work
- **Midori-64:** Dong, 2016: Custom algorithm → 14 rounds (out of 16), $2^{116}$ operations
- **Midori-128:** Not done

Our results (Indocrypt 2016)
- Few hours
- Full round for both versions
- Practical attacks:
  - **Midori-64:** $2^{35}$
  - **Midori-128:** $2^{43}$
Other directions: FSE2017

Problem
Searching for integral, zero-correlation linear, and impossible differential distinguisher on various block ciphers

Results
- PRESENT, HIGHT, SKINNY
- Reproduced results from the literature
- New distinguisher on SKINNY
Conclusion and future challenges

- CP is readable and easy to use
- It is less error prone than custom code
- It performs better than other approaches
- It generalizes MILP and SAT
- **Use CP!**
Thank you for your attention
Other ways to improve a CP model

- **Variable ordering**: Starting with the most constrained one
- **Value choice**: If you want to minimize a sum, affecting variables to 0 first is a good idea
- **BlackBox heuristics**: domain over weighted degree, etc...
- **Restarts**: Reseed the BlackBox strategy after some time
- **Other methods**: The power of **MiniZinc**
- **Parallel solving**: Not trivial but can help
2 steps solving

**Step 1: boolean abstraction**

\[ \Delta = 0 \]
\[ \Delta = 1 \]

Find candidate solutions

**Step 2: actual byte values**

\[ \delta = 0 \]
\[ \delta \neq 0 \]

Check their consistency

---

**Step 1**

Step1(n) gives an output \( \mathcal{O} = (\Delta X, \Delta K, \Delta C) \) and the corresponding difference propagation path, such that the number of Sboxes is minimal.

**Step 2**

Step2(\( \mathcal{O} \)) returns a probability and the difference values along the path if \( \mathcal{O} \) is consistent, 0 otherwise.
Modelling properly

Straightforward modelling
With a naive approach, more than 90 millions \textit{inconsistent} step 1 solutions found for 4 rounds of AES-128 with 11 active SBoxes

More elaborate modelling
With a more suble approach, 0 inconsistent solution
Example: XOR Constraint

(white = 0, colored ≠ 0)

Byte values

\[ \delta_A \oplus \delta_B = \delta_C \]

Boolean abstraction

\[ \Delta_A \oplus \Delta_B = \Delta_C \]

Inferring equalities

XORs introduce a lot of branching, but storing information about equality or difference during step 1 helps filtering a lot!
Example: XOR Constraint

(white = 0, colored ≠ 0)

Inferring equalities

XORs introduce a lot of branching, but storing information about equality or difference during step 1 helps filtering a lot!
With which software

**Specific solver: Highly customizable**

Fine-grained tuning: table constraint heuristics, custom constraints etc...

- Choco (Java)
- Gecode (C++)
- Sunny-CP (portfolio)
- Chuffed (Uses SAT techniques)
- and many more...

**MiniZinc: More generic**

- CP language, compiled to FlatZinc
- Read by many solvers, including SAT and MILP solvers
- MiniZinc competition
More details

**Choco: General structure**

- **Solver:** Solver s = new Solver("Example solver");
- **Variables:** IntVar X = VF.bounded(0, 5, s);
- **Constraints:** s.post(ICF.arithm(X, "!=" , 3);
- **Heuristics:** s.set(ISF.domOverWDeg(allvars, someSeed));
- **Solve:** s.findSolution();

**MiniZinc: General structure**

- **Variables:** var 0..5: X;
- **Constraints:** constraint X = 5;
- **Heuristics and solve:** solve:: int_search(allVars, dom_w_deg, indomain_min, complete) satisfy;
Case study: PRESENT (Bogdanov, 2007)

Problem

Search for optimal differential characteristics, i.e. difference propagation patterns with the highest possible probability.