COMPARISON OF MILP SOLVERS
WITH USING RELATED-KEY DIFFERENTİAL ATTACK

Halil İbrahim Kaplan

TDD / Kripto Analiz Laboratuvarı

halil.kaplan@tubitak.gov.tr

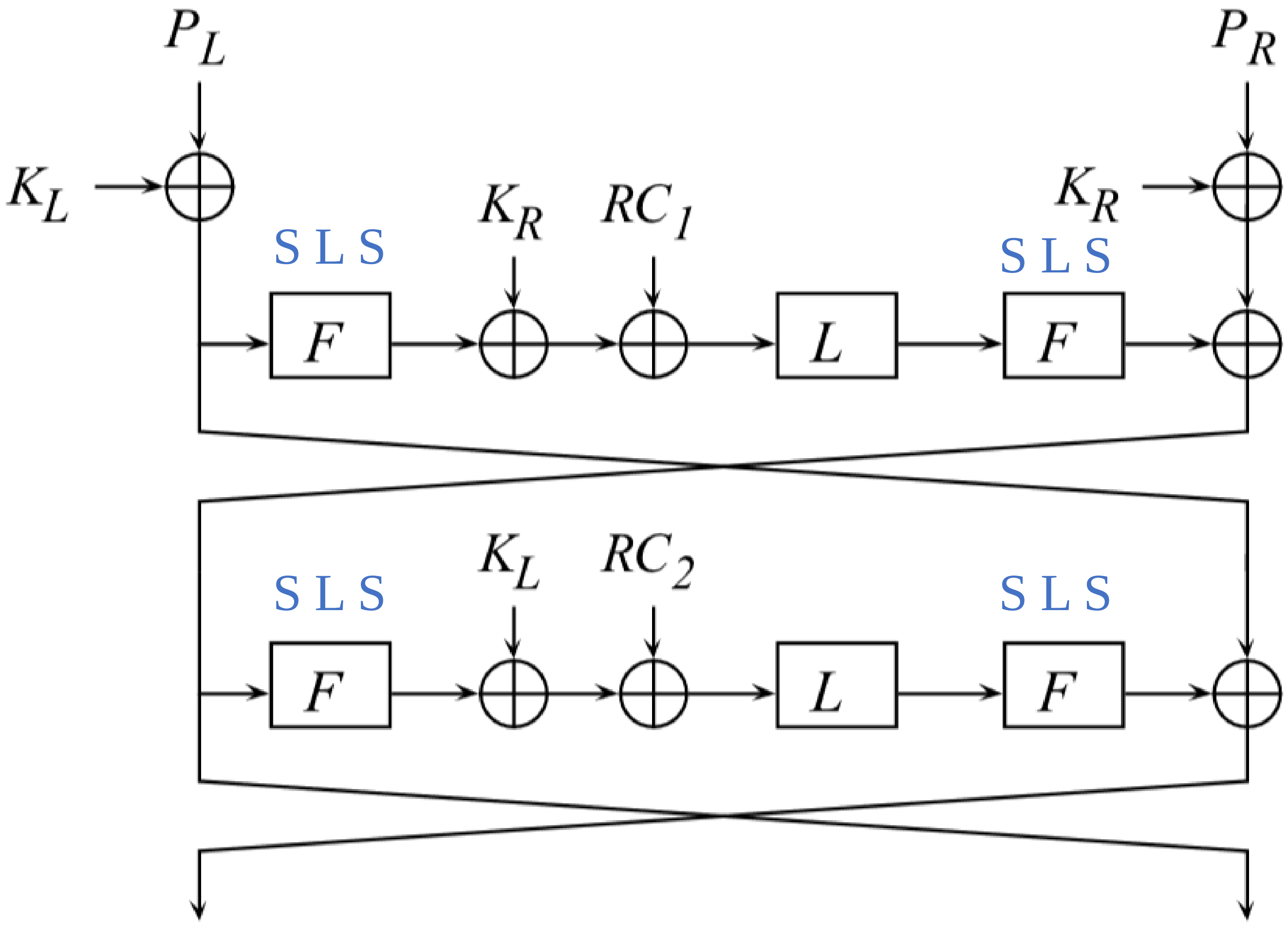
2024

TÜBİTAK
BİLGEM

Overview

1. ITUbee Block Cipher
2. MILP Model of Related-Key Differential Attack
3. GLPK Solver
4. HiGHS Solver
5. Gurobi Solver
6. CPLEX Solver
7. Comparison

ITUbee Block Cipher



The encryption operation proceeds as follows:

1. $X_1 \leftarrow P_L \oplus K_L$ and $X_0 \leftarrow P_R \oplus K_R$.
2. for $i = 1...20$ do
   (a) if $i \in \{1, 3, ..., 19\}$
      i. $RK \leftarrow K_R$.
   (b) else
      i. $RK \leftarrow K_L$.
   (c) $X_{i+1} \leftarrow X_{i-1} \oplus F(L(RK \oplus RC_i \oplus F(X_i)))$. Note that 16-bit round constant is added to the rightmost 16 bits.
3. $C_L \leftarrow X_{20} \oplus K_R$ and $C_R \leftarrow X_{21} \oplus K_L$.

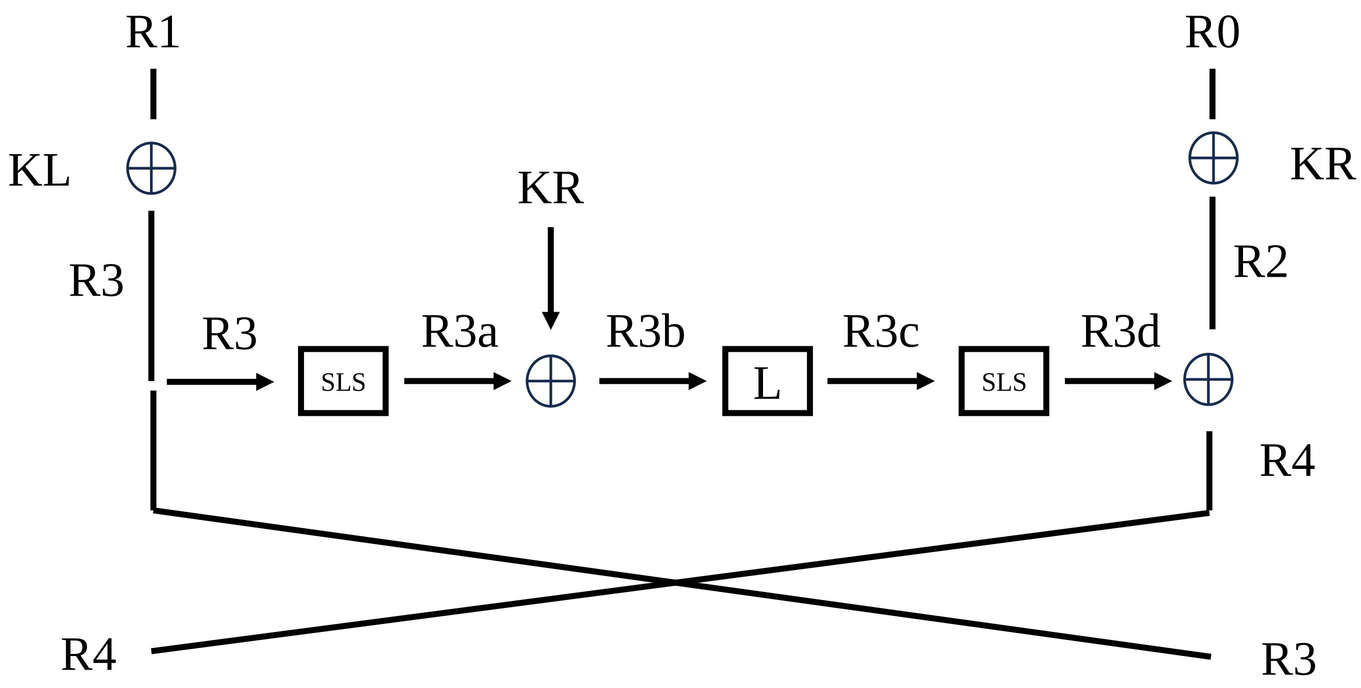
- $F(X) = S(L(S(X)))$.
- $S(a\|b\|c\|d\|e) = s[a]\|s[b]\|s[c]\|s[d]\|s[e]$ where a, b, c, d, e are 8-bit values and s is the S-box used in AES [10].
- $L(a\|b\|c\|d\|e) = (e \oplus a \oplus b)\|(a \oplus b \oplus c)\|(b \oplus c \oplus d)\|(c \oplus d \oplus e)\|(d \oplus e \oplus a)$.

ITUbee offers 80-bit security. Also, the cipher provides the same security level against attacks in related key models as well as in single key attack models. In addition, the cipher has no weak keys.

4.3 Related Key Differential Attacks

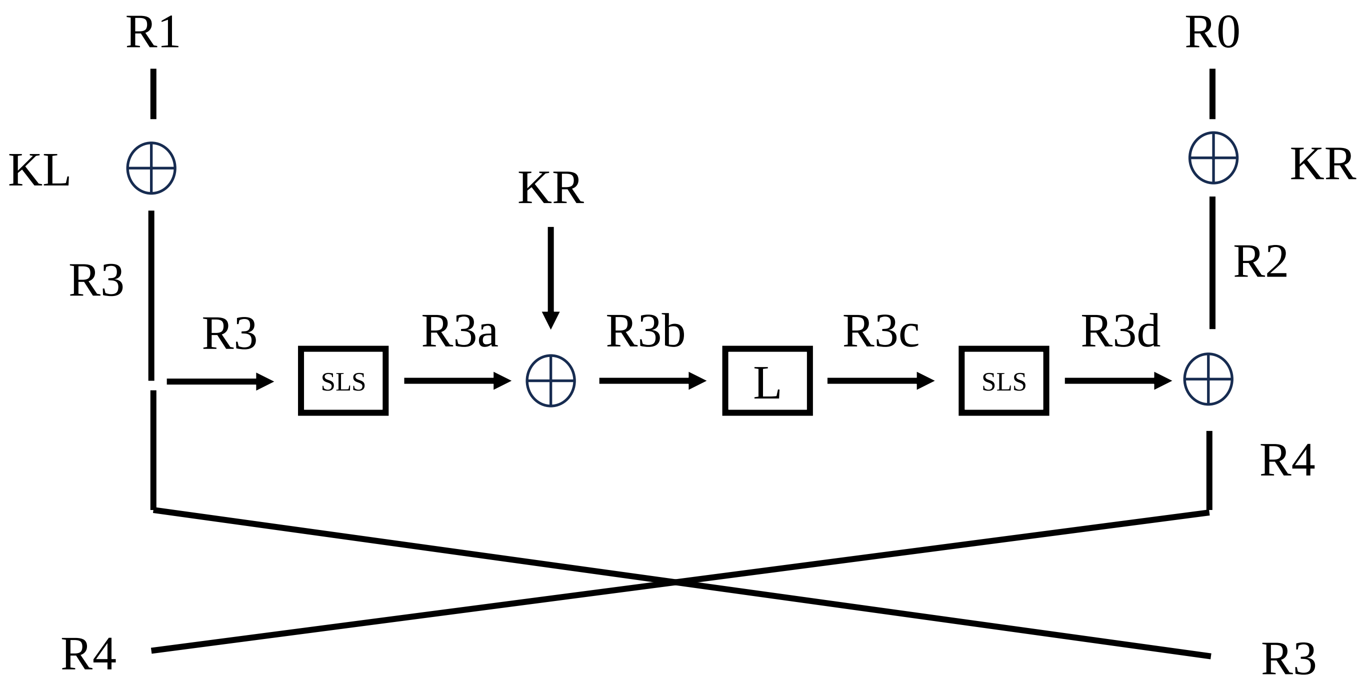
In the related key attack model, the adversary is able to collect plaintext and ciphertext pairs under related keys. In our algorithm, we divide the master key into two parts K_L and K_R and we use these parts between F functions in the rounds. It is trivial to see that when there is a difference in the key used between F functions, then at least one of the F functions will be active. In the best case for the adversary, the difference will be only one part of the key K_L or K_R. As a result, in two consecutive rounds there exists at least one active F function in the case of the related key attack. The probabilities of differentials for the F function is less than 2^{-17} as given in the Section 4.1. Thus, for 10 consecutive rounds the probabilities will be less than $(2^{-17})^5 = 2^{-85}$ which is not usable in an attack.

MILP Model of Related-Key Differential Attack



Minimize: R3 + R3a + R3c + R3d

MILP Model of Related-Key Differential Attack



Add Constraint : KL + KR ≥ 1

XOR:

$$x^{\oplus}_{in_1} + x^{\oplus}_{in_2} + x^{\oplus}_{out} \geq 2d^{\oplus} \quad ,$$
$$d^{\oplus} \geq x^{\oplus}_{in_1} \quad ,$$
$$d^{\oplus} \geq x^{\oplus}_{in_2} \quad ,$$
$$d^{\oplus} \geq x^{\oplus}_{out} \quad .$$

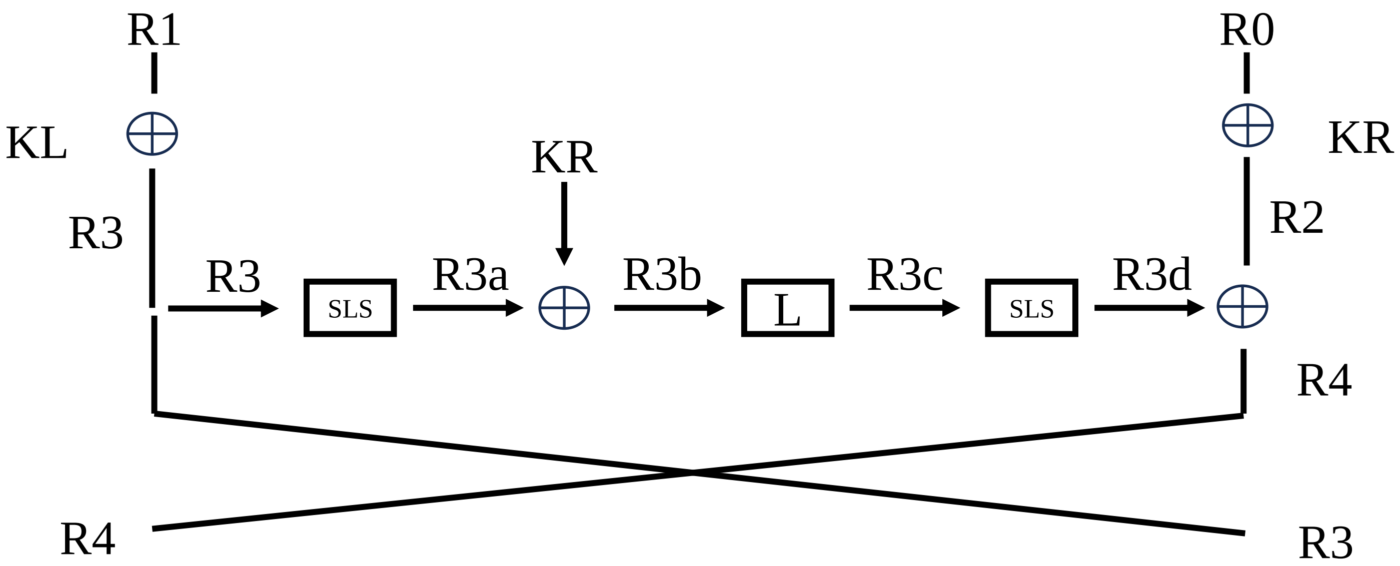
```
def xor(p,in1,in2,out,dummy):
    for i in range(len(in1)):
        p.add_constraint(in1[i]+in2[i]+out[i]-2*dummy[i] >=0)
        p.add_constraint(dummy[i]- in1[i] >= 0)
        p.add_constraint(dummy[i]- in2[i] >= 0)
        p.add_constraint(dummy[i]- out[i] >= 0)
```

L Function:

```
pp = Polyhedron(vertices=[

[ 0,0,0,0,0,0,0,0,0,0 ],
[ 0,0,0,0,1,1,0,0,1,1 ],
[ 0,0,0,1,0,0,0,1,1,1 ],
[ 0,0,0,1,1,1,0,1,0,0 ],
[ 0,0,0,1,1,1,0,1,1,1 ],
[ 0,0,1,0,0,0,1,1,1,0 ],
[ 0,0,1,0,1,1,1,1,0,1 ],
[ 0,0,1,0,1,1,1,1,1,1 ],
[ 0,0,1,1,0,0,1,0,0,1 ],
[ 0,0,1,1,1,1,1,0,1,0 ],
[ 0,0,1,1,1,1,1,0,1,1 ],
[ 0,0,1,1,0,0,1,1,1,1 ],
[ 0,0,1,1,1,1,1,1,1,1 ],
[ 0,0,1,1,1,1,1,1,1,0 ],
```

MILP Model of Related-Key Differential Attack

```
Toplam aktif s-kutusu:  16.0
KL-KR              {0: 0.0, 1: 1.0, 2: 0.0, 3: 1.0, 4: 0.0} {0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0}

R1-R0              {0: 0.0, 1: 1.0, 2: 0.0, 3: 1.0, 4: 0.0} {0: 1.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 1.0}

R3-R2              {0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0} {0: 1.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 1.0}

R3a-R3b-R3c-R3d    {0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0} {0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0} {0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0} {0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0}

R4-R3              {0: 1.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 1.0} {0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0}

R4a-R4b-R4c-R4d    {0: 0.0, 1: 1.0, 2: 0.0, 3: 1.0, 4: 0.0} {0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0} {0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0} {0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0}

R5-R4              {0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0} {0: 1.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 1.0}

R5a-R5b-R5c-R5d    {0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0} {0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0} {0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0} {0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0}

R6-R5              {0: 1.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 1.0} {0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0}

R6a-R6b-R6c-R6d    {0: 0.0, 1: 1.0, 2: 0.0, 3: 1.0, 4: 0.0} {0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0} {0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0} {0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0}

R7-R6              {0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0} {0: 1.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 1.0}

R7a-R7b-R7c-R7d    {0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0} {0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0} {0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0} {0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0}

R8-R7              {0: 1.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 1.0} {0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0}

R8a-R8b-R8c-R8d    {0: 0.0, 1: 1.0, 2: 0.0, 3: 1.0, 4: 0.0} {0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0} {0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0} {0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0}

R9-R8              {0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0} {0: 1.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 1.0}

R9a-R9b-R9c-R9d    {0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0} {0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0} {0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0} {0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0}

R10-R9             {0: 1.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 1.0} {0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0}

R10a-R10b-R10c-R10d {0: 0.0, 1: 1.0, 2: 0.0, 3: 1.0, 4: 0.0} {0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0} {0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0} {0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0}

R11-R10            {0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0} {0: 1.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 1.0}
```




```
Toplam aktif s-kutusu:  16.0
KL-KR            {0: 0.0, 1: 1.0, 2: 0.0, 3: 1.0, 4: 0.0} {0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0}

R1-R0            {0: 0.0, 1: 1.0, 2: 0.0, 3: 1.0, 4: 0.0} {0: 1.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 1.0}

R3-R2            {0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0} {0: 1.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 1.0}

R3a-R3b-R3c-R3d  {0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0} {0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0}
[redacted]       {0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0} {0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0}

R4-R3            {0: 1.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 1.0} {0: 0.0, 1: 0.0, 2: 0.0, 3: 0.0, 4: 0.0}
```

GLPK (GNU Linear Programming Kit)

Introduction | Downloading | Documentation | Mailin

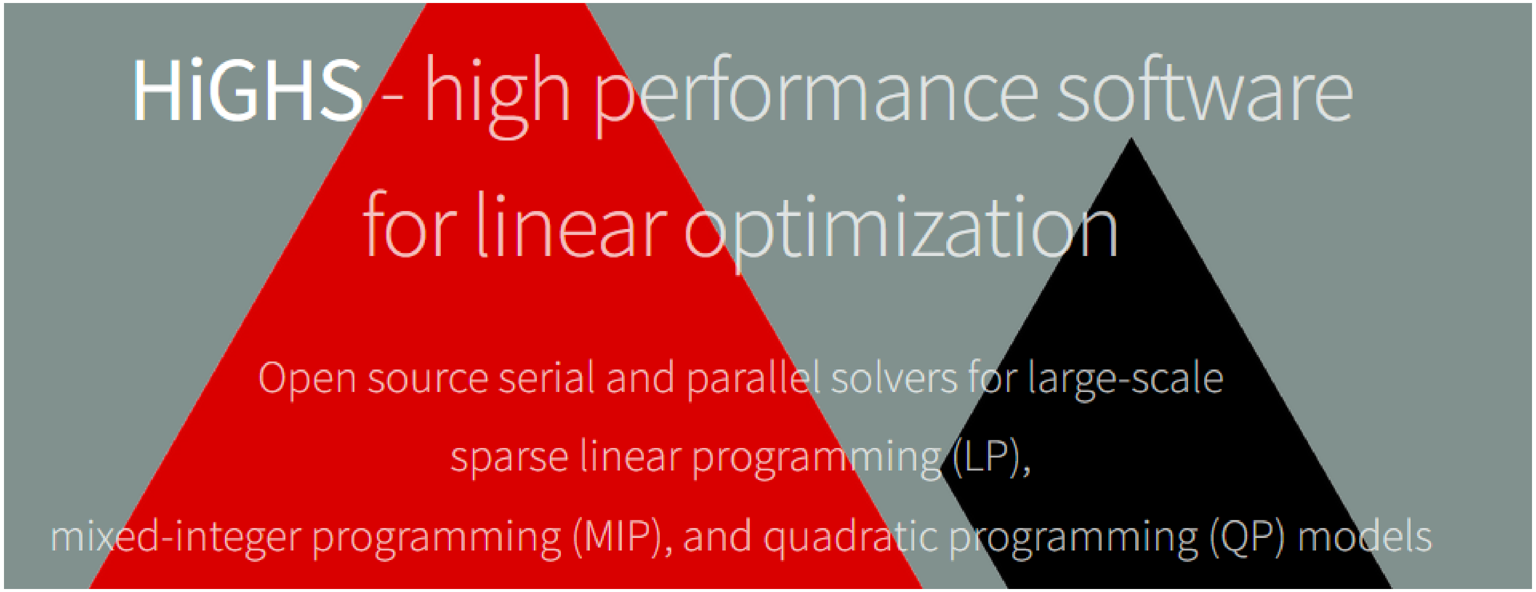
Introduction to GLPK

The GLPK (GNU Linear Programming Kit) package is intended for solving large-scale linear programming
the form of a callable library.

GLPK supports the *GNU MathProg modeling language*, which is a subset of the AMPL language.

The GLPK package includes the following main components:

- primal and dual simplex methods
- primal-dual interior-point method
- branch-and-cut method
- translator for GNU MathProg
- application program interface (API)
- stand-alone LP/MIP solver



Solvers

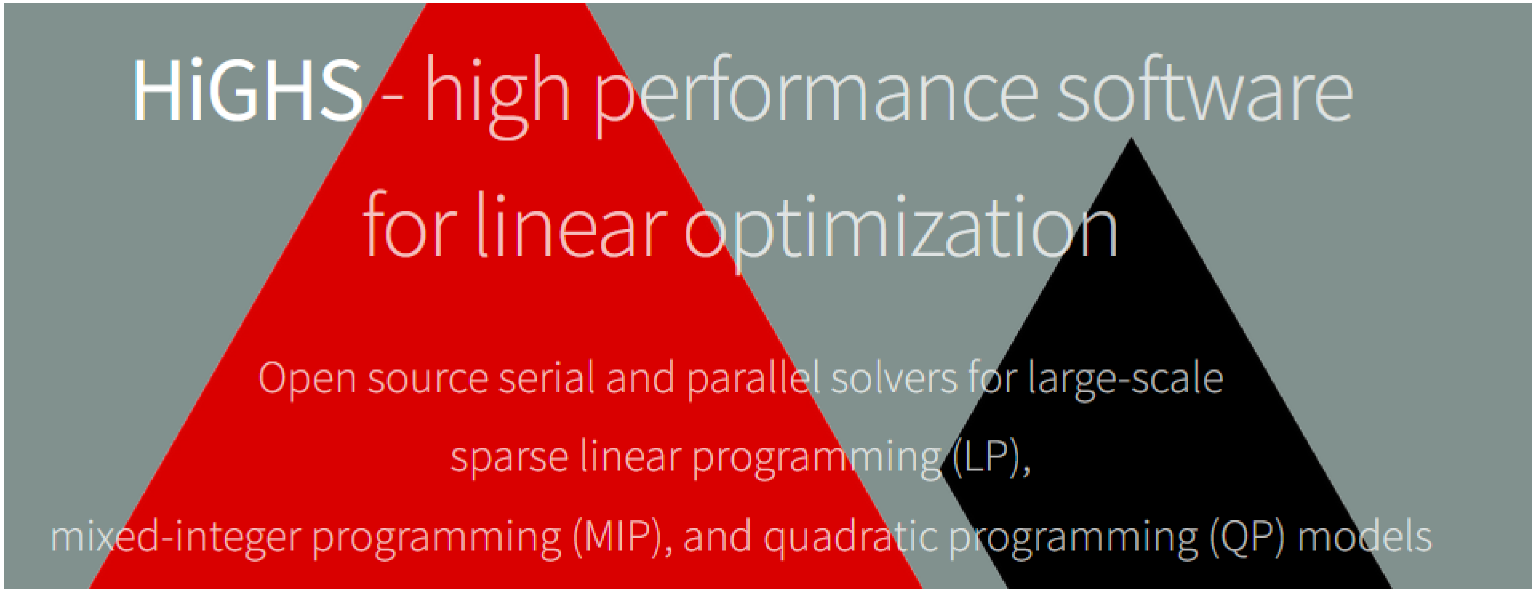
LP

HiGHS has implementations of the three main solution techniques for LP. HiGHS will choose the most appropriate technique for a given problem, but this can be over-ridden by setting the option solver.

Simplex

HiGHS has efficient implementations of both the primal and dual simplex methods, although the dual simplex solver is likely to be faster and is more robust, so is used by default. The novel features of the dual simplex solver are described in

HiGHS Solver



Interior point

HiGHS has one interior point (IPM) solver based on the preconditioned conjugate gradient method, as discussed in

Implementation of an interior point method with basis preconditioning, Mathematical Programming Computation, 12, 603-635, 2020. DOI: 10.1007/s12532-020-00181-8.

This solver is serial. An interior point solver based on direct factorization is being developed.

Setting the option **solver** to "ipm" forces the IPM solver to be used

Primal-dual hybrid gradient method

HiGHS includes the cuPDLP-C primal-dual hybrid gradient method for LP (PDLP). Currently this only runs on CPU, so it is unlikely to be competitive with the HiGHS interior point or simplex solvers. Enabling HiGHS to run PDLP on a GPU is work in progress.

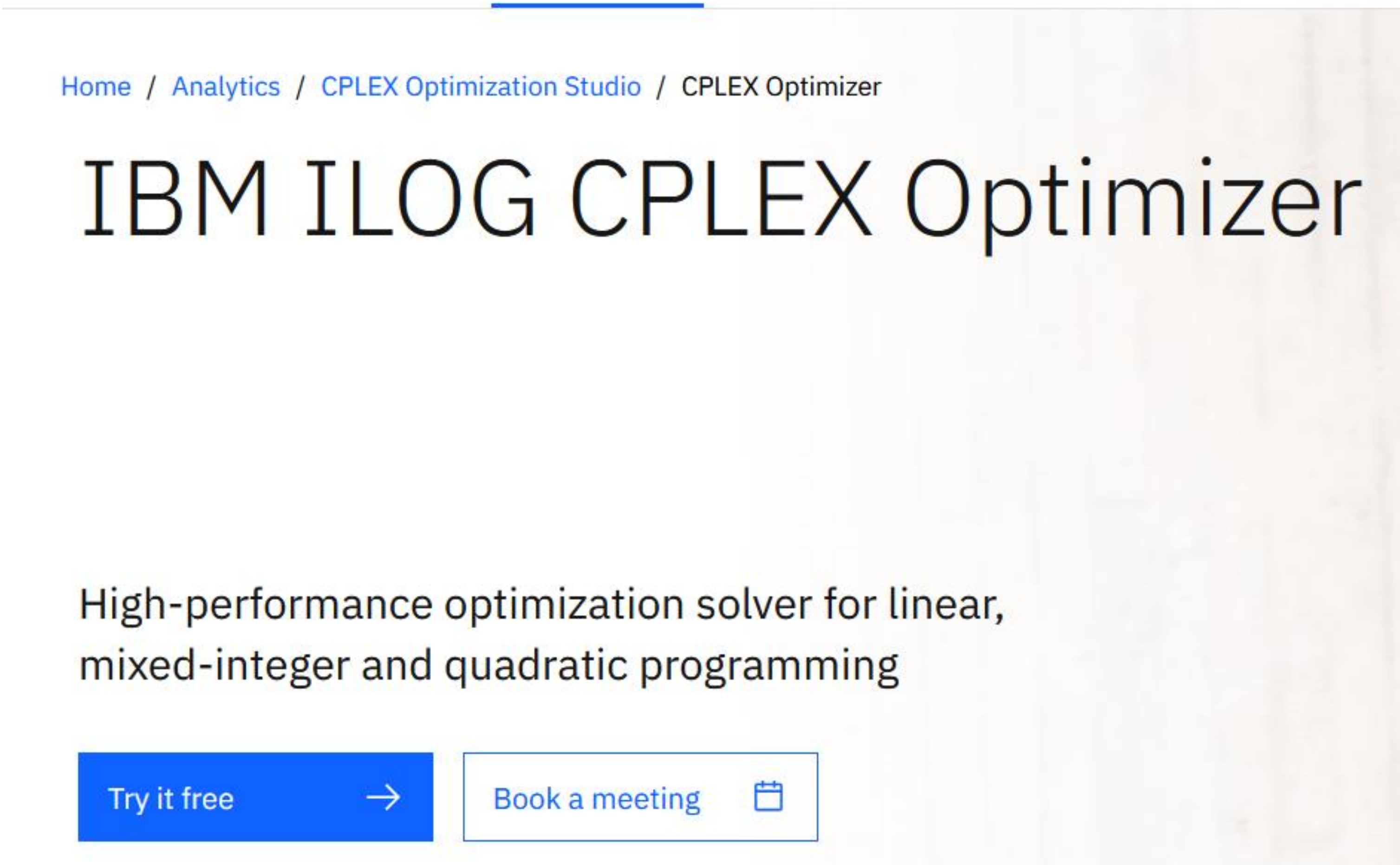
GUROBI OPTIMIZER

The World's Fastest Solver

Unmatched Performance

With our powerful algorithms, you can add complexity to your model to better represent the real world, and still solve your model within the available time.

- The performance gap grows as model size and difficulty increase.
- Gurobi has a history of making continual improvements across a range of problem types, with a more than 75x speedup on MILP since version 1.1.



Home / Analytics / CPLEX Optimization Studio / CPLEX Optimizer
IBM ILOG CPLEX Optimizer
High-performance optimization solver for linear, mixed-integer and quadratic programming
Try it free
Book a meeting

Comparison

PROBLEM	SOLVER	TIME(sec)	SPEED
Related-Key Differential Cryptanalysis of ITUbee algoirthm (8 round) 48649 constraints 320 variables	GLPK	257	x (base)
	HiGHS	231	1.1x
	SCIP	86	2.9x
	Gurobi	37	6.9x
	CPLEX	16	16x

CPU: 11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz
RAM: 32,0 GB

Comparison

PROBLEM	SOLVER	TIME(sec)	SPEED
Related-Key Differential Cryptanalysis of ITUbee algoirthm (10 round) 60801 constraints 390 variables	GLPK	1610	x (base)
	HiGHS	321	5x
	SCIP	202	7.9x
	Gurobi	34	47x
	CPLEX	40	40x

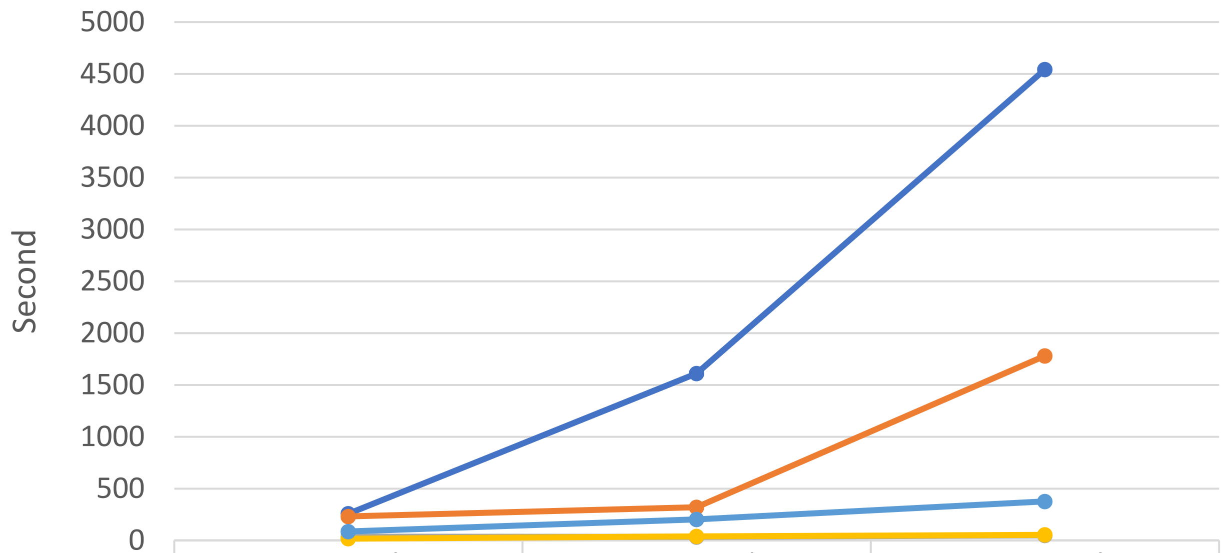
CPU: 11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz
RAM: 32,0 GB

Comparison

PROBLEM	SOLVER	TIME(sec)	SPEED
Related-Key Differential Cryptanalysis of ITUbee algoirthm (12 round) 72953 constraints 460 variables	GLPK	4543	x (base)
	HiGHS	1780	2.5x
	SCIP	376	12x
	Gurobi	48	94x
	CPLEX	53	85x

CPU: 11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz
RAM: 32,0 GB

Comparison



	8 round	10 round	12 round
GLPK	257	1610	4543
HiGHS	231	321	1780
Gurobi	37	34	48
CPLEX	16	40	53
SCIP	86	202	376

CPU: 11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz

RAM: 32,0 GB

Referances

- Karakoç, F., Demirci, H., & Harmancı, A. E. (2013). ITUbee: a software oriented lightweight block cipher. In *Lightweight Cryptography for Security and Privacy: Second International Workshop, LightSec 2013, Gebze, Turkey, May 6-7, 2013, Revised Selected Papers 2* (pp. 16-27). Springer Berlin Heidelberg.

- Mouha, N., Wang, Q., Gu, D., & Preneel, B. (2012). Differential and linear cryptanalysis using mixed-integer linear programming. In *Information Security and Cryptology: 7th International Conference, Inscrypt 2011, Beijing, China, November 30–December 3, 2011. Revised Selected Papers 7* (pp. 57-76). Springer Berlin Heidelberg.

- Developers, S. (2020). SageMath. *the Sage Mathematics Software System (Version 9.0). Retrieved July*, *21*, 2022.

- Gurobi, I. (2013). Gurobi Optimizer 5.0. *Gurobi*.

- Manual, C. U. S. (1987). Ibm ilog cplex optimization studio. *Version*, *12*(1987-2018), 1.