Declarative approach for Symmetric Cryptography

Decrypt

anr agence nationale de la recherche

Appel : ANR-18-CE39-0007

Année : 2018

Instrument : : AAPG 2018

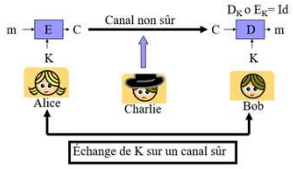
Contact :
marine.minier@loria.fr

COORDINATEUR : Marine Minier (LORIA)

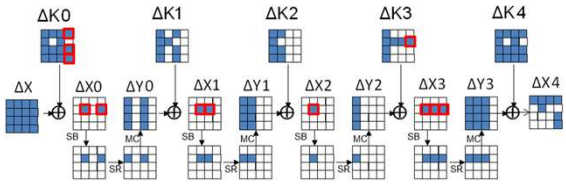
PARTENAIRES : IRISA, LIMOS, LIRIS, LORIA, TASC

Résumé (3 lignes max) :
Ce projet a pour but de proposer un langage déclaratif dédié aux problèmes de cryptanalyses en cryptographie symétrique en utilisant la programmation par contraintes (CP) pour simplifier la représentation des attaques, pour améliorer des attaques existantes et pour construire de nouvelles primitives de chiffrement symétrique.

CONTEXTE ET OBJECTIFS

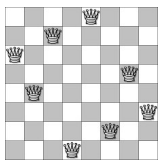


La cryptographie symétrique est une pierre angulaire de la sécurité informatique. Contrairement à la clé publique, en cryptographie symétrique, les deux parties doivent partager une clé commune pour communiquer. Les primitives les plus courantes en cryptographie symétrique sont les chiffrements par flots et les chiffrements par bloc qui garantissent la confidentialité des communications et les fonctions de hachage pour l'intégrité.



Au cours des cinq dernières années, de nombreux résultats de recherche ont tenté d'attaquer ces primitives à l'aide d'outils automatiques tels que la programmation linéaire mixte en nombres entiers (MILP) ou les solveurs de satisfiabilité booléenne (SAT). Ce projet vise à proposer une approche déclarative par programmation par contraintes (CP) pour simplifier la façon dont les attaques en cryptographie symétrique sont modélisées et ainsi dépasser les résultats cryptanalytiques existants notamment pour la cryptanalyse différentielle.

```
int n = 8;
Model model = new Model(n + "-queens problem");
IntVar[] vars = new IntVar[n];
for(int q = 0; q < n; q++){
    vars[q] = model.intVar("Q_"+q, 1, n);
}
for(int i = 0; i < n-1; i++){
    for(int j = i + 1; j < n; j++){
        model.arithm(vars[i], "!=",vars[j]).post();
        model.arithm(vars[i], "!=", vars[j], "-", j - i).post();
        model.arithm(vars[i], "!=", vars[j], "+", j - i).post();
    }
}
Solution solution = model.getSolver().findSolution();
if(solution != null){
    System.out.println(solution.toString());
}
```

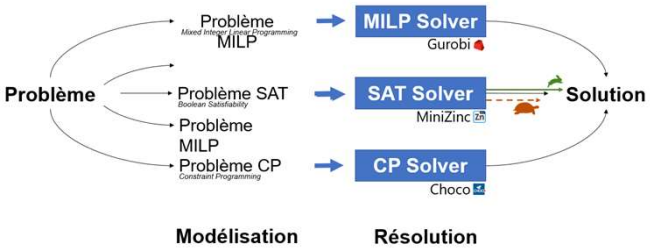


MÉTHODOLOGIE ET RÉSULTATS

Méthodologie : Pour mener à bien une attaque en cryptographie symétrique, on va programmer 2 étapes :

- une première étape abstrait les valeurs de mots en bits et on cherche à minimiser le nombre de bits à 1 ;
- une deuxième étape va utiliser ces chemins minimaux pour chercher des valeurs sur les mots tout en maximisant une fonction objectif.

La première étape peut être efficacement résolue avec MILP, SAT, CP ou des outils dédiés tandis que CP surpasse toutes les autres méthodes pour la deuxième étape. La manière de modéliser la première étape reste cependant très importante et détermine les temps d'exécution de celle-ci.



Résultats majeurs du projet :

-) Nouvelle contrainte : AbstractXOR [CP 2020]

-) Nouvelles attaques implémenter : Boomerang [ToSC 2020 – Best Paper], Division Property [SAC 2021]

-) Amélioration d'attaques différentielles : sur l'AES [AI 2020], sur SKINNY [ACNS 2021], sur Rijndael [à paraître]

-) TAGADA : un outil automatique pour trouver des bornes de sécurité en cryptanalyse différentielle [CP 2021]