

باسمه تعالی

گزارش پروژه درس رمزنگاری پیشرفته

**دانشگاه صنعتي شريف**

عنوان:

جستجوی خودکار تمایزگر تفاضل ناممکن

فاز اول

انجام‌دهنده:

پوریا دادخواه  
401201381

# بهار 1401

چکیده

در این پروژه قصد داریم یک تمایزگر خودکار- به این معنی استفاده از یک خانواده الگوریتم مشخص قابل پیاده‌سازی کامپیوتری به جای انجام اعمال دستی و یا نیمه دستی – برای حمله تفاصل ناممکن به الگوریتم رمز simon پیدا کنیم.

به این منظور ابتدا به جزییات بیشتری از حمله تفاضل ناممکن و روند کلی حمله از جمله یافتن بهترین زوج ورودی از طریق جدول DDT، رسیدن به شرط تناقض در تعداد دور مشخص و سپس توسعه به مرحله ورودی-خروجی به صورت قطعی و با احتمال 1، از طریق الگوریتم بازیابی کلید پرداختیم.

سپس با مطالعه مروری مقاله 5 از مراجع، با روند پیاده‌سازی متفاوت حمله تفاضلی با استفاده از constraint programing آشنا شدیم که به جای قرار دادن شرط خود روی تفاضل ورودی و خروجی و استفاده از روش‌های معمول-دستی، نیمه خودکار و وابسته به الگوریتم هدف- برای یافتن تناقض و سپس بازیابی کلید با روش‌های نه‌چندان کارا، مساله را به صورت واحد دیده و با مدل سازی درست به تعریف متغیرها، دامنه‌ها و نهایتا قیود متناسب با وقوع تناقض موردنیاز می‌پردازد و از یک الگوریتم حل مساله بهینه‌ساز- معمولا به صورت جعبه ‌سیاه- مانندSAT-MILP به یافتن تمایزگر مطلوب می‌پردازد.

در آخر با مطالعه مقاله 6 و همچنین مقاله الگوریتم [[1]](#footnote-1)simon با این الگوریتم fiestel و کلمه محور و تابع دور این الگوریتم-متشکل از شیفت های چرخشی، اند بیت‌وایز و xor با کلید دور- و نحوه تولید زیرکلید‌های آشنا شدیم.

هم‌چنین علاوه بر مطالعات نظری فوق سورس‌کد‌های پیاده‌سازی الگوریتم simon، نمونه کد اعمال حمله جستجو خودکار تمایزگر تفاضل ناممکن بر رمز skinny و minizinc به عنوان یکی از ابزارهای موجود برای اعمال الگوریتم cp و حل مساله بهینه‌سازی را بررسی کردیم.

با این مطالعات نهایتا هدف این است که بتوانیم با توجه الگوریتم هدف علاوه بر مشخص کردن متغیر‌ها و فراپارامتر‌های مساله قیود متناسب این الگوریتم را برای دستیابی به یک تناقض پیدا کرده و با استفاده از یک الگوریتم cp حالت مختلف ورودی دادن را تست کرده ( مانند تعداد دور های هربخش از حمله، یا تعداد جفت ورودی‌های ورودی الگوریتم) و تمایزگری را انتخاب کنیم که پس از توسعه و بازیابی کلید، زمان و پیچیدگی مطلوبی در یافتم کلید درست داشته باشد.

از چالش‌های این مسیر هم می‌توان علاوه بر یافتن قیود بهینه سازی متناظر با simon به نحوه استفاده از cp ها اشاره کرد که از آن‌جایی که به سازوکار آن آشنایی نداریم شاید یافتن hyperparameter های مناسب کار ساده ‌ای نباشد و با جا‌به‌جایی یا کم و زیاد کردن قیود عملکرد متوسط دیگری را شاهد باشیم. همچنین چالش‌های نرمال پیاده‌سازی از جمله سازگاری سورس کدهای مختلف و کار با واسط‌ها کاربری کتابخانه‌ها نیز طبیعتا ممکن است چالش ‌برانگیز باشد.

با تشکر

1. Observations on the SIMON block cipher family Stefan Kölbl1, Gregor Leander2, and Tyge Tiessen1 2015 [↑](#footnote-ref-1)