



ТЕХНИЧЕСКИ УНИВЕРСИТЕТ – СОФИЯ

Факултет по телекомуникации

Катедра „Технологии и мениджмънт на комуникационни системи“

Маг. Ериселда Гури Малай

**РАЗРАБОТВАНЕ НА НОВИ ХАРДУЕРНИ РЕШЕНИЯ ЗА
КИБЕРСИГУРНОСТ НА КОМУНИКАЦИОННИ СИСТЕМИ**

А В Т О Р Е Ф Е Р А Т

**на дисертация за придобиване на образователна и научна степен
"ДОКТОР"**

Област: 5. Технически науки

Професионално направление 5.3 Комуникационна и компютърна техника

Научна специалност "Автоматизация на инженерния труд и системи за
автоматизирано проектиране"

Научен ръководител: доц. д-р инж. Галя Маринова

София, 2026 г.

Дисертацията се обсъжда и насочва за защита от Съвета на катедрата "Технологии и управление на комуникационни системи" към Факултета по телекомуникации на ТУ-София на редовна среща, проведена на 11.06.2026 г.

Публичната защита на дисертацията ще се проведе на 01.10.2026 г. от 15:00 часа в Конферентната зала на БИК на Техническия университет – София на открито заседание на научното жури, определено с нареждане № ОЖ – 5.3 – 55 / 24.06..2026 г. на ректора на ТУ-София, съставено от

1. Доц. д-р инж. Тихомир Брусев - Председател
2. Доц. д-р инж. Ростислав Русев - Научен секретар
3. Проф. д-тн Тодор Стоилов
4. Доц. д-р инж. Адриана Бороджиева
5. Доц. д-р инж. Валентин Мутков

Рецензенти:

1. Доц. д-р инж. Тихомир Брусев
2. Доц. д-р инж. Адриана Бороджиева

Материалите по защитата са достъпни за заинтересованите в офиса на Факултета по телекомуникации на ТУ-София, блок No1, офис No 1306.

Дисертантът е задочен докторант към катедра „Технологии и мениджмънт на комуникационни системи“ на факултет по телекомуникации. Изследванията по дисертационната разработка са направени от автора, като някои от тях са подкрепени от научноизследователски проекти.

Автор: маг. Ериселда Малай

Заглавие: Разработване на нови хардуерни решения за киберсигурност на комуникационни системи.

Тираж: 20 броя

Отпечатано в ИПК на Технически университет – София

I. ОБЩА ХАРАКТЕРИСТИКА НА ДИСЕРТАЦИОННИЯ ТРУД

Актуалност на проблема

Логическото криптиране е техника, предназначена да повиши сигурността на цифровите схеми чрез вграждане на механизми за криптиране. Този подход цели да защити интелектуалната собственост и да защити интегралните схеми (ИС) от обратен инженеринг и хардуерна намеса. Алгоритъмът, предложен в тази работа, използва логическо криптиране, за да постигне стабилна защита за цифрови проекти чрез директно интегриране на криптирането в логическата структура на схемата.

Сърцевината на новия алгоритъм включва вграждане на криптографски трансформации в логическите елементи на схемата, създавайки сигурен и обфусциран проект. Това се постига чрез приложение на комбинация от булеви функции и криптографски ключове за промяна на функционалността на елементите. Получената криптирана схема е устойчива на неоторизиран достъп и обратно инженерство, тъй като логическата функционалност е скрита и достъпна само за тези с правилните ключове за декриптиране.

Нашият подход включва и механизми за управление и разпространение на ключове, гарантирайки, че ключовете за криптиране се обработват сигурно и достъпват само от оторизирани потребители. Демонстрираме ефективността на предложения алгоритъм чрез серия симулации и оценки, подчертавайки неговата устойчивост срещу често срещани вектори на атаки като обратен инженеринг и сондиране.

Предложеният алгоритъм за логическо криптиране осигурява значително подобрение на сигурността на цифровите схеми, като предпазва от кражба на интелектуална собственост и хардуерни атаки, като същевременно запазва функционалността и производителността. Този принос развива областта на хардуерната сигурност чрез въвеждане на практичен и ефективен метод за защита на интегрални схеми чрез логическо криптиране.

Нарастващата сложност и взаимосвързаност на съвременните комуникационни системи засилват нуждата от надеждни механизми за сигурност с ниска латентност. Традиционното софтуерно криптиране, макар и гъвкаво, става все по-уязвимо към хардуерни заплахи като странични канални атаки, обратно инженерство и извличане на логика. Тази дисертация представя разработването, симулацията и оценката на нови техники за криптиране на хардуерно ниво, насочени към подобряване на киберсигурността на комуникационните инфраструктури.

Изследват се три логически базирани метода за криптиране: криптиране, базирано на схеми XOR/XNOR, обфускация чрез мултиплексор (MUX) и последователно криптиране, базирано на тригери (Flip-Flop). Тези техники са реализирани във Verilog/VHDL и интегрирани в стандартни бенчмарк схеми и комуникационни подмодули. Разработва се цялостна рамка за оценка с помощта на инструменти за синтез и симулация (ModelSim, Vivado, Quartus), а производителността се оценява чрез критични метрики: консумация на енергия, допълнителна площ, закъснение, средно разстояние на Хаминг (AHD) и средна чувствителност към повреди (AFS).

Резултатите показват ясни компромиси между ефективността и сигурността. XOR/XNOR криптирането предлага минимално използване на ресурси, но ограничена устойчивост на атаки. Проектите, базирани на MUX, подобряват обфускацията с умерени разходи, докато криптирането, базирано на тригери, осигурява най-висока защита за последователна логика за сметка на по-голяма сложност. Резултатите подкрепят персонализирано внедряване на логика за криптиране в зависимост от изискванията за сигурност и производителност на приложението.

Дисертационният труд допринася с валидирана методология за сигурено хардуерно проектиране в комуникационните системи и предлага насоки за бъдещи подобрения чрез хибридни архитектури, тестване, базирано на изкуствен интелект, и интеграция с квантово-устойчиви криптографски примитиви.

Съществуващите методи за логическо криптиране срещат предизвикателства като лоша стандартизация, ограничена мащабируемост и компромиси между сигурност и производителност (площ, закъснение, хранване). Много от тях са или твърде скъпи за системи в реално време, или недостатъчно надеждни срещу напреднали атаки като SAT решатели, сондаж и инжектиране на грешки. Въпреки потенциала си, няколко ключови въпроса остават нерешени в приложението на логическото криптиране:

- **Ефективна интеграция:** Как логическото криптиране може ефективно да бъде интегрирано в процеса на проектиране на цифрови схеми, без да се компрометира тяхната производителност или функционалност?
- **Устойчивост на атаки:** До каква степен логическото криптиране може да устои на напреднали техники за обратен инженеринг и сондиращи атаки, и как може да бъде количествено измерена и подобрена неговата устойчивост?
- **Управление на ключове:** Кои са най-добрите практики за управление и разпространение на криптиращи ключове по начин, който гарантира както сигурност, така и практичност, особено при мащабни внедрявания?
- **Design Overhead:** Как могат да се оптимизират компромисите между сигурността и допълнителните ресурси за проекта, за да се минимизира въздействието върху площта на схемата, консумацията на енергия и бързодействието?

Тази дисертация адресира тези предизвикателства, като предлага нов алгоритъм за логическо криптиране, който цели да повиши сигурността на цифровите схеми, като същевременно запазва практическите ограничения на проекта. Изследването се фокусира върху разработването на ефективни методи за интегриране на криптиране в логическите елементи, подобряване на устойчивостта на различни вектори на атаки и установяване на стабилни практики за управление на ключове. Чрез комбинация от теоретичен анализ, симулация и емпирична валидация, тази работа цели да напредне в областта на хардуерната сигурност и да предостави практически решения за защита на интегрални схеми от нововъзникващи заплахи. Останалата част от тази дисертация е структурирана така, че да предостави цялостно изследване на хардуерната сигурност чрез логическо криптиране, преминавайки от основни концепции към практическа реализация и анализ.

Цел на дисертацията, основни задачи и методи на изследване

Въз основа на синтеза на прегледаната литература, дисертацията идентифицира ключови предизвикателства и пропуски в областта на логическото криптиране. Те включват:

- Устойчивост към нови методологии за атака, като тези, произтичащи от напредъка в квантовите изчисления.
- Предизвикателства при управлението на ключове и необходимостта от по-ефективни и мащабируеми механизми за криптиране.

Целта на дисертацията е да се проектират хардуерни криптиращи модули с логически структури, да се внедрят в комуникационни хардуерни системи, да се оцени и сравни тяхната производителност чрез ключови хардуерни метрики (мощност, площ, закъснение), да се оцени всеки метод чрез метрики като устойчивост на AHD, AFS и SAT атаки, както и да се предложи нова рамка за оценка на хардуерно криптиране, приложима към комуникационни подсистеми.

За постигане на тази цел са дефинирани следните задачи:

- А) За дефиниране на логически модули и логически структури за криптиране. Да се реализират хардуерни езици за описание на комуникационните хардуерни системи и среда на оценъчна система.
- Б) Да се дефинират ограниченията в производителността на съвременните хардуерни криптиращи подходи в комуникационни схеми.
- В) За да се оцени как различните техники за криптиране на логическо ниво се сравняват по площ, забавяне и разход за хранене.
- Г) Те определят техниката, която предлага най-добрия компромис между силата на криптирането и хардуерната ефективност.
- Д) Да се разработи нова методология за оценка за систематично оценяване на ефективността на хардуерните криптиращи техники в комуникационни системи.
- Е) Да валидира предложените решения.

Това изследване прилага систематична и многостепенна методология за изследване на ефективната интеграция, устойчивостта на сигурността и компромисите между техниките за логическо криптиране в цифровите схеми. Предложената методология комбинира теоретичен анализ, хардуерен проект, симулационна оценка и оценка на сигурността, за да гарантира както академична строгост, така и практическа значимост.

Изследването следва експериментален подход ориентиран към проектирането. Основният фокус е върху разработването и оценката на техники за криптиране на логическо ниво, които подобряват сигурността на веригите, като същевременно запазват приемливи проектни разходи. Три различни методологии за логическо криптиране са проектирани, реализирани и анализирани, за да се осигури цялостно сравнение между структурни, функционални и сигурностни измерения.

Общата методология е структурирана в следните фази:

- 1) Проектиране и интеграция на логическо криптиране
- 2) Хардуерно описание и реализация
- 3) Оценка на представянето
- 4) Оценка на сигурността
- 5) Сравнителен анализ и валидиране.

Таблица 11 Обобщение на характеристиките на изследователските методи

	Метод на изследване	Тип	Цел	Използвани инструменти
1.	Математическо моделиране	Количествени	Формулиране на метрики за сигурност и производителност	MATLAB, Python
2.	Хардуерен проект и реализация	Количествени	Реализиране на архитектури за логическо криптиране	Verilog/VHDL
3.	Логически синтез и симулация	Количествени	Оценка на площта, мощността и закъснението	Vivado / Design Compiler, ModelSim
4.	Анализ на сигурността	Количествени	Измерване на устойчивостта на AHD, AFS и SAT	Python, MiniSAT
5.	Сравнителна оценка	Смесени	Сравнение на техниките за криптиране	MATLAB, Python
6.	Преглед на литературата	Смесени	Идентифициране на тенденции и пропуски в изследванията	IEEE Xplore, Google Scholar

Изследванията върху логическото криптиране включват три различни методологии за събиране на данни:

- 1) **Експериментален подход**, експерименталният подход се използва за наблюдение на въздействието на логическото криптиране върху поведението на веригата при контролирани условия.
- 2) **Подход, базиран на симулация** Подходът, базиран на симулация, позволява детайлен анализ на функционалната коректност, времето, захранването и натоварването на площта чрез хардуерни инструменти за описание и симулация.
- 3) **Подход за оценка и валидиране**. Накрая, подходът за оценка и валидиране се използва за оценка на устойчивостта и ефективността на предложените схеми за криптиране чрез количествени метрики и сравнителен анализ.

Това изследване акцентира върху вземането на решения, базирани на данни, в изследванията на хардуерната сигурност чрез прилагане на математически и статистически анализ, използвайки MATLAB и Python. Тези платформи са използвани за обработка и анализ на експериментални и симулационни данни, получени от логически криптирани цифрови схеми. Нека наборът от данни е дефиниран като:

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N, \quad (2.1)$$

където x_i представлява вектора на независимите параметри на проекта, включително тип на криптиране, дължина на ключа и сложност на схемата, и y_i означава съответните зависими променливи, като консумация на енергия, надразход на площ, забавяне на разпространение, AHD и AFS. Описателен статистически анализ беше използван за характеризиране на разпределението на всяка метрика за производителност и сигурност.

Научна новост

Тази дисертация предоставя унифицирано проучване и сравнителен анализ на техники за логическо криптиране, базирано на XOR/XNOR-, MUX- и Flip-Flop, за хардуерна сигурност. За разлика от предишни изследвания, които се фокусират върху отделни методи, то установява обща рамка за оценка на тяхната ефективност по сигурността, сложността на внедряването и натоварването на

производителността. Работата идентифицира ключови изследователски пропуски и предлага бъдещи насоки за разработване на по-сигурни и ефективни хардуерни криптиращи решения за съвременни интегрални схеми и вградени системи.

Практическа приложимост

Резултатите от тази теза могат да се приложат при проектирането на сигурни интегрални схеми, вградени системи, IoT устройства и комуникационен хардуер. Чрез предоставяне на сравнителен анализ на техники за логическо криптиране, базирани на XOR/XNOR-, MUX- и Flip-Flop, изследването помага на инженери и изследователи да изберат подходящи механизми за сигурност, които балансират защитата, производителността, разтоварването на площта и консумацията на енергия. Предложеният нов подход допринася за разработването на по-сигурни и устойчиви хардуерни системи срещу обратен инженеринг, манипулиране и неотризиран достъп.

Публикации

Основните постижения и резултати от дисертацията са публикувани в 10 научни публикации, представени в следната таблица.

Списания, включени в Scopus, Web of Science и IEEEEXPLORE	2 Статии Номер 1, No 2	ТЕЛЕКОМ (2020) ТЕЛЕКОМ (2021)	Индексирано от IEEE Xplore, Scopus,		
	5 Статии No 6, No 7 No8 No9 No10	TechDefense (2023) TechDefense (2024) TechDefense (2025)	Индексирано от IEEE Xplore, Scopus		
	1 Статии No 4	IWSSIP (2022) IWSSIP (2021 Underline Science Inc.)	Публикувано в Science Direct, индексирано от Scopus.		
Международна конференция и списание	1 Статия No 3	Международна конференция по наука, UBT (2021)			
Глави в книгите	1 глава No 5	Мониторинг и защита на критична инфраструктура чрез безпилотни системи, iOS Press	Индексирано от Scopus		
Цитати	7 Цитати				

Структура и обем на дисертацията

Дисертацията е в обем от 165 страници, включително въведение, 5 глави за решаване на формулирани основни задачи, списък с основните приноси, списък с публикациите по дисертацията и използвана литература. Цитирани са общо 200 литературни източника. Дисертационният труд включва общо 54 фигури и 33 таблици. Номерацията на фигурите и таблиците в автореферата съответства на тези в дисертацията.

II. СЪДЪРЖАНИЕ НА ДИСЕРТАЦИЯТА

ГЛАВА I. Аналитичен преглед на хардуерни методи, техники, мултимедийни инструменти за проектиране на комуникационни схеми

В тази глава се представя подробен преглед на литературата за съществуващите техники за хардуерно криптиране и сигурност. Анализирани са предишни изследвания в областта на логическото криптиране, обфускацията и свързаните с тях контрамерки, като същевременно идентифицира техните силни страни, ограничения и пропуски, които това изследване цели да адресира.

Хардуерната обфускация е изследвана както за FPGA, така и за ASIC проектни потоци, както и за различни нива на абстракция, например при прехвърляне на регистри и IP на ниво елемент. Моделът на сигурност в хардуерната обфускация е подобен на криптирането, но тук разчита на тайната на ключа, а не на типа алгоритъм. Целите на трансформацията и начините, по които те се постигат. Логическото криптиране се извършва чрез въвеждане на специализирани логически структури или портове, контролирани от ключове, които водят до грешни резултати за грешен ключ и обратно. Втората цел е проектът да се скрие чрез структурна трансформация. Тази цел е съществена по две причини:

- 1) защитава самия катинар като блокер, който лесно може да бъде премахнат и също така лесно разпознаваем;
- 2) предотвратява изтичането на секретната информация за проекта, като логическия стил и типа изпълнявана функция

Проектирането на сигурни комуникационни системи включва цялостно разбиране както на софтуерни, така и на хардуерни механизми за сигурност. Въпреки че софтуерната сигурност традиционно е била фокус на повечето решения за киберсигурност, нарастващата сложност на нападателите и еволюцията на вградените технологии наложиха преход към по-хардуерно ориентирани подходи за сигурност. Този раздел разглежда фундаменталните различия, допълващите се роли и компромисите между тези два парадигми.

Критично е да се разберат възможните хакери, които могат да започнат атака, използвайки тези пропуски в сигурността. Това може да помогне на проектантите да разберат възможностите на нападателите и да изберат подходящи контрамерки според операцията и целевия противник. Той може да приеме формата на индивид или организирана група, която иска да получи, унищожи или се намеси в актив, до който няма право да има достъп. Противниците могат да се разделят на вътрешни и външни лица, като се вземе предвид процесът на проектиране на интегрални схеми и участващите компании. В този раздел са изобразени на различни етапи от процеса на проектиране на SoC. Литературният обзор в областта на хардуерната сигурност представя разнообразни техники и гледни точки, но в тази развиваща се област могат да бъдат идентифицирани няколко силни страни, ограничения и противоречия.

ГЛАВА II. Изследователски методи за хардуерно криптиране, техники и мултимедийни инструменти за проектиране на комуникационни вериги

В тази глава е обобщено описанието, фокусирано върху теоретичния фон и изследователската методология. Той въвежда основните принципи на криптографията и хардуерната сигурност, обяснява моделите на потенциални атаки и описва методологиите, инструментите и проектантските подходи, използвани в изследването.

2.1 Методология на изследване

Това изследване прилага систематична и многостепенна методология за изследване на ефективната интеграция, сигурността и компромисите между техниките за логическо криптиране в цифровите схеми. За да гарантира както академична строгост, така и практическа значимост, предложената методология комбинира теоретичен анализ, хардуерно проектиране, симулационна оценка и оценка на сигурността.

Изследването следва проектно-ориентиран експериментален подход. Основният фокус е върху разработването и оценката на техники за криптиране на логическо ниво, които подобряват сигурността на схемите, като същевременно запазват приемливи проектни разходи. Проектирани, реализирани и анализирани са три различни методологии за логическо криптиране, за да се осигури цялостно сравнение между структурни, функционални и сигурностни измерения.

Общата методология е структурирана в следните фази:

- 1) Проектиране и интеграция на логическо криптиране
- 2) Хардуерно описание и реализация
- 3) Оценка на представянето
- 4) Оценка на сигурността
- 5) Сравнителен анализ и валидиране.

Таблица 11 Обобщение на характеристиките на изследователските методи

	Метод на изследване	Тип	Цел	Използвани инструменти
1.	Математическо моделиране	Количествени	Формулиране на метрики за сигурност и производителност	MATLAB, Python
2.	Хардуерно проектиране и реализация	Количествени	Реализиране на архитектури за логическо криптиране	Verilog/VHDL
3.	Логически синтез и симулация	Количествени	Оценка на площта, мощността и разходката за време	Vivado / Design Compiler, ModelSim
4.	Анализ на сигурността	Количествени	Определяне на устойчивостта на AHD, AFS и SAT	Python, MiniSAT
5.	Сравнителна оценка	Смесени	Сравнение на техниките за криптиране	MATLAB, Python
6.	Преглед на литературата	Смесени	Идентифициране на тенденции и пропуски в изследванията	IEEE Xplore, Google Scholar

Изследванията върху логическото криптиране включват три различни методологии за събиране на данни:

- 1) **Експерименталният** подход, експерименталният подход, се използва за наблюдение на въздействието на логическото криптиране върху поведението на веригата при контролирани условия.
- 2) **Подходът, базиран на** симулация, позволява детайлен анализ на функционалната коректност, времето, храненето и натоварването на площта чрез хардуерни инструменти за описание и симулация.

- 3) **Подход за оценка и валидиране.** Накрая, подходът за оценка и валидация се използва за оценка на устойчивостта и ефективността на предложените схеми за криптиране чрез количествени метрики и сравнителен анализ.

2.2 Фактори, влияещи върху логическото криптиране при проектиране на програмируеми логически матрици (FPGA)

Този раздел показва, че логическото криптиране води до съществена промяна на проектните характеристики на системите, базирани на FPGA, като въвежда допълнителна логика, зависима от ключове, сложност на маршрутизацията и активност при превключване. В този раздел обяснете факторите, влияещи върху конвенционалните FPGA дизайни, които имат засилено въздействие при интегриране на механизмите за криптиране. Разбирането на тези фактори е от съществено значение за балансиране на сигурността, енергийната ефективност, натоварването на производителността и устойчивостта на хардуерни атаки.

2.3 Методи за проектиране на архитектура за логическо криптиране

Разработени са три архитектури за логическо криптиране, които представят както комбинаторни, така и последователни стратегии за заключване: Нека некриптирана цифрова схема бъде моделирана като тютъл (кортеж):

$$C=(X,Y,F) \quad (2.1)$$

където

- $X=\{x_1,x_2,...,x_n\}$ означава множеството от първични входове,
- $y=\{y_1,y_2,...,y_m\}$ означава множеството от първични изходи,
- $F: \{0,1\}^n \rightarrow \{0,1\}^m$ представлява булевата функционалност.

Криптирана (заключена) схема се дефинира като:

$$C_L = (X, K, Y, F_L) \quad (2.2)$$

където

- $K=\{k_1,k_2,...,k_p\}$ е векторът на секретния ключ,
- $F_L: \{0,1\}^{n+p} \rightarrow \{0,1\}^m$ е криптираната булева функция.

Правилната функционалност се изпълнява ако и само ако:

$$F_L(X, K_c) = F(X) \quad (2.3)$$

където K_c е правилният ключ за криптиране.

1) Криптиране, базирано на елементи XOR/XNOR,

Елементите XOR/XNOR се вмъкват в стратегически избрани възли в схемата. Коректността на изхода на схемата зависи от приложения секретен ключ. Грешните ключове водят до повредени изходи, осигурявайки защита срещу неототоризирана употреба и обратно инженерство. Криптирането се реализира чрез поставяне на елементите XOR/XNOR в избрани вътрешни възли. Криптираният сигнал се дефинира като: $v_i v_i'$

$$v_i' = v_i \oplus k_i \text{ or } v_i' = \overline{v_i} \oplus k_i \quad (2.4)$$

За грешен ключ $K \neq K_s$ вероятността за изходна повреда се увеличава статистически по логическия конус, увеличавайки функционалната ентропия.

2) Криптиране, базирано на MUX,

Логическото заключване, базирано на мултиплексори, се реализира чрез въвеждане на MUX структури, управлявани чрез ключове, които динамично избират между правилни и грешни пътища на сигнала. Тази техника увеличава функционалната неяснота и усложнява структурния анализ от страна на нападателите. Криптирането, базирано на мултиплексор, въвежда логика на избор, контролирана от ключове:

$$v_i' = k_i \cdot f_{\text{true}}(v_i) + \overline{k_i} \cdot f_{\text{false}}(v_i) \quad (2.5)$$

Където f_{false} е умишлено грешен логически път. Тази структура увеличава неяснотата при обратния инженеринг, като създава множество функционално правдоподобни конфигурации.

3) Криптиране, базирано на тригери

Предлага се нова схема за криптиране, базирана на тригери (Flip-Flop), за последователни схеми. Криптирането се постига чрез вграждане на поведение, контролирано от ключове, в елементи, държащи състоянието, като по този начин се защитават както пътят на данните, така и преходите към състоянието. Този подход засилва устойчивостта срещу атаки, които използват вериги от сканиране и последователно поведение. За последователни схеми еволюцията на състоянието се моделира като:

$$S_{t+1} = G(S_t, X_t) \quad (2.6)$$

Криптираният преход на състоянието става:

$$S_{t+1} = G_L(S_t, X_t, K) \quad (2.7)$$

Поведението на тригера, зависимо от ключа, променя или:

- Функции за обновяване на състоянието, или
- Логика на гейтинг за тактовия генератор или за данните

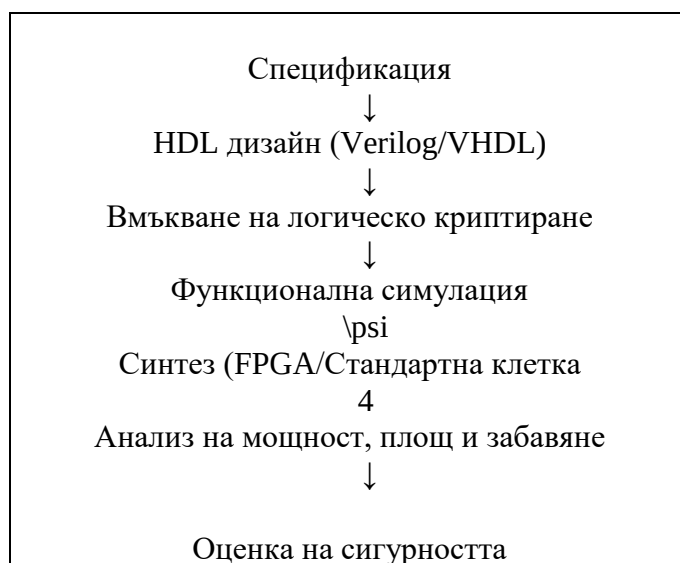
Това въвежда времева зависимост от ключа, значително увеличавайки устойчивостта към сканиращи и последователни атаки.

2.4 Хардуерна реализация

Всички архитектури за криптиране са реализирани с помощта на Verilog/VHDL, за да се гарантира синтезируемост и съвместимост с реални блок-схемите на цифровите проекти.

- Криптиращите елементи са интегрирани в комуникационни хардуерни блокове.
- Проектите се синтезират чрез библиотеки със стандартни клетки.
- Прилагат се унифицирани ограничения за дизайн, за да се гарантира справедливо сравнение между техниките.

Тази фаза на внедряване гарантира, че предложените решения са готови за внедряване и подходящи за интеграция в практически комуникационни подсистеми. Общият процес на хардуерна имплементация, възприет в това изследване, е илюстриран на Фигура 19. Потокът следва структурирана и съвместима с инструментите методология на проектиране, започваща с HDL моделиране и завършваща с оценка след синтеза.



Фигура 19 Хардуерна реализация за логически криптирани проекти върху FPGA

Този алгоритъм гарантира последователност, повторяемост и коректно сравнение между всички техники за криптиране.

- **Интеграция на криптиращи модули**

Криптиращи модули, базирани на елементите XOR/XNOR, заключване чрез мултиплексор MUX и криптиране, базирано на тригери, са интегрирани в комуникационни хардуерни блокове като шифратори, контролери и устройства за обработка на данни. Логиката на криптиране се вмъква в стратегически избрани точки в пътя на данните и контролния път, за да се максимизира въздействието върху сигурността, като същевременно се минимизират допълнителната площ.

Inputs → Functional Logic → Encryption Logic → Outputs

↓

Таен ключ

Ключовите сигнали се съхраняват в защитени регистри и се разпределят вътрешно, за да се предотврати външна наблюдаемост. Криптираният проект дава правилни изходни резултати само когато е приложен правилният ключ.

- **HDL моделиране и симулация**

Всички проекти се описват чрез синтезируем Verilog/VHDL код на ново регистриран трансфер RTL. Където е възможно се предпочита поведенческо описание, което позволява на инструментите за синтез да извършват логическа оптимизация и споделяне на ресурси. Структурните описания се използват селективно в криптиращи модули, за да се поддържа прецизен контрол върху логическото поведение, зависещо от ключа.

Функционалната симулация се извършва за потвърждение на:

- Правилна работа при правилния ключ,
- Изходна повреда при неправилни ключове,
- Липса на функционални нарушения, въведени от криптиране.

• Ограничения при синтез и проектиране

За да се гарантира обективно сравнение криптираните и некриптираните проекти се синтезират чрез еднакви ограничения. За всички реализации се прилагат идентични тактови честоти, нива на напрежение и настройки за оптимизация. В зависимост от целевата платформа се използват библиотеки със стандартни клетки (за ASIC-стил оценка) или FPGA издателски библиотеки.

Основните цели на синтеза включват:

- Минимизиране на добавената вследствие на криптирането площ,
- Запазване на закъсненията,
- Ограничение за увеличената поради добавена логика изразходвана мощността.

Тази хардуерна методология за внедряване гарантира, че:

- Всички архитектури на криптиране са напълно синтезируеми,
- Проектите могат директно да се имплементират на FPGA или ASIC платформи,
- Показателите за производителност и сигурност се определят при идентични условия.

Структурираният поток позволява безпроблемна интеграция на логическото криптиране в практически комуникационни подсистеми, валидирайки възможността на предложените техники за реални хардуерни приложения за сигурност.

Таблица 12. Последователност на имплементация и експериментално мапване на резултатите специфични за инструмента/платформата

Инструмент/Платформа	Етап на проектиране	Генерирани изходи	Таблицы свързани с експерименталните резултати
Xilinx Vivado	RTL синтез, разполагане и маршрутизация анализ на мощността	LUT/FF/DSP/BRAM използване, тайминг, динамична и статична изразходвана мощност	(Допълнителна площ), (Закъснение), (Захранване), (Въздействие на скоростта на превключване)
Intel Quartus Prime	Синтез, напасване, оценка на мощността	Използвани ресурси, Fmax, изразходвана мощност	(Сравнение на реализации върху схеми от различни производители), (Сравнение на закъснението), (Изразходвана мощност за

			реализации върху схеми от различни производители)
Synopsys Design Compiler	синтез на ASIC	Брой гейтове, площ на клетки, генериране на нетлист	(Площ на нивото на елементите), (Анализ на закъсненията)
Synopsys PrimeTime/PX	Анализ на времето и мощността	Изразходвана мощност от тока на утечка, динамична мощност, критични пътища	(Разбиване на захранването на ASIC), (Тайминг на ниво гейт)
ModelSim/XSim	Функционална и RTL симулация	Правилно/неправилно поведение на ключа, валидиране на времедиаграмите	(Валидиране на функционалната коректност)
MATLAB / Python	Статистически анализ и анализ на сигурността	AHD, AFS, SAT резистентност, регресионни резултати	AHD, (AFS), (SAT Runtime), (Анализ на силата на ключовете)

2.5 Хардуерни метрики и оценка на сигурността

Ефективността на техниките за логическо криптиране се оценява чрез комбинация от хардуерни метрики за производителност и мерки за сигурност. Показателите за производителност количествено определят натоварването на имплементацията, въведено от криптирането, докато мерките за сигурност оценяват устойчивостта на криптирания проект срещу различни модели на атаки. Заедно тези метрики предоставят цялостна рамка за оценка за сравнение на различни архитектури на логическото криптиране.

А. Показатели за представяне:

Показателите за производителност измерват въздействието на логическото криптиране върху физическите и времевите характеристики на хардуерния проект.

- Консумация на енергия,
- Областен овърхед (брой логически гейтове, LUT)
- Латентност/Забавяне (критичен път, забавяне на разпространението)

2.6 Стратегия за управление на ключове

Ефективното управление на ключове е от съществено значение, за да се гарантира, че логическото криптиране осигурява силни гаранции за сигурност, като същевременно проектите остават мащабируеми и практични за реално внедряване. В това изследване се анализират стратегиите за управление на ключове по отношение на сигурното съхранение, контролираната активация, мащабируемостта на разпределение и чувствителността към ключовете, което гарантира, че предложените архитектури за криптиране могат да се интегрират в мащабни хардуерни системи извън лабораторни условия.

2.7 Сравнителен анализ и валидиране

За всички предложени техники за логическо криптиране се провежда структуриран сравнителен анализ, използвайки унифициран набор от показатели за производителност и сигурност. Този анализ осигурява обективна и възпроизводима оценка на стратегиите за криптиране при идентични дизайнерски ограничения, позволявайки справедливо сравнение и смислени заключения.

Рамката за оценка, разработена в това изследване, позволява:

- Обективно сравнение на техниките за криптиране, базирани на XOR/XNOR, MUX и Flip-Flop
- Идентифициране на оптимални компромиси между хардуерните надразходи (мощност, площ, закъснение) и сигурност
- Валидиране на предложения метод за криптиране, базиран на Flip-Flop, особено за последователни схеми.

ГЛАВА III: Нова методология, алгоритъм и имплементация на техники за хардуерно криптиране

Тази глава представя внедряването и систематичната оценка на техники за хардуерно логическо криптиране за подобряване на киберсигурността на комуникационни системи. Изследването разглежда три представителни механизма за криптиране: логическо криптиране, базирано на XOR/XNOR, логическо криптиране, базирано на MUX, и криптиране, базирано на тригери. Тези техники вграждат примитиви за сигурност директно в хардуерната тъкан, като по този начин укрепват устойчивостта срещу обратен инженеринг, атаки с инжектиране на грешки и експлоатация на странични канали. Предлага се алгоритмична рамка за оценка на коменталната устойчивост на криптиране и ресурсите за имплементацията. Рамката интегрира логическо обфускиране, ключово-зависими трансформации и криптиране, базирано на състояние, за да оцени сигурността чрез експерименти с инжектиране на грешки, анализ на разстоянието на Хаминг и метрики за разпространение на грешки. Паралелно се анализира влиянието на криптирането върху латентността, консумацията на енергия и разходката на площта, за да се характеризират компромисите в производителността. Целта на тази глава е двустранна: (i) да се идентифицира техниката за криптиране, която осигурява най-силната обфускация и зависимост от ключове, и (ii) да се установи възпроизводима методология за оценка, която да подкрепя информирани проектантски решения за сигурен хардуер в комуникационни системи.

3.1 Рамка за реализация на хардуерна криптираща логика

Прилага се двуслойна рамка за имплементация, която позволява както високо ниво на оценяване, така и хардуерно точно валидиране. Симулацията, базирана на Python, се използва за алгоритмичен анализ и метрични изчисления, докато VHDL-базирана реализация на RTL се прилага за хардуерна верификация.

- **Python симулационен слой:** Оценява силата на криптирането, разстоянието на Хаминг и чувствителността към грешки при контролирани условия.
- **VHDL хардуерен слой:** Реализира и валидира логиката на криптиране на ниво RTL, осигурявайки синтезируемост и коректност.

Това разделяне позволява бързо прототипиране, като същевременно запазва коректността на реалното хардуерно поведение.

- **Симулация на устойчивостта на криптирането, базирана на Python**

Симулационната рамка базирана на Python оценява силата на криптиране чрез следната процедура:

- Генериране на случайни логически схеми и изходни вектори.
- Приложение на криптиране, базирано на XOR/XNOR, MUX и Flip-Flop.
- Изчисляване на HD между оригинални и криптирани изходи.
- Изпълнение на инжекция на бит-флип грешка, при която всеки входен бит се обръща за измерване на изходната чувствителност.

Този подход позволява ефективно, статистически значимо сравнение на схеми за криптиране при идентични условия.

• HDL-базирана хардуерна верификация

RTL реализациите на всички техники за криптиране се разработват с помощта на синтезируем HDL. Унифицираният тестбенч проверява:

- Правилното функционално поведение при правилния ключ.
- Изходна повреда под грешни ключове.
- Разпространение на грешка при инжектиране на грешка.
- Последователност с резултатите на ниво симулация.

Този етап потвърждава практическата жизнеспособност на предложената криптираща логика за внедряване на FPGA и ASIC.

• Сравнителна оценка на техники за криптиране

Трите схеми за криптиране се оценяват по следния начин:

- **Криптиране, базирано на XOR/XNOR:** Въвежда инверсия, зависима от ключа, използвайки елементи XOR/XNOR.
- **Криптиране, базирано на MUX:** Избира между правилни и грешни логически пътища чрез мултиплексори, контролирани с ключове.
- **Криптиране, базирано на тригери:** Включва зависимост от последователни състояния чрез D-тип тригери, въвеждайки времева обфускация.

Работен процес на оценка:

- Генериране случайна логическа схема и тестови вектори.
- Прилагане всеки метод за криптиране.
- Изчисляване на метрики за сигурност (AHD, AFS).
- Сравнение устойчивостта и въздействието върху производителността.

Използвали сме скрипта за генериране на случайна логическа схема и извършване на криптиране с методи XOR/XNOR, MUX и Flip-Flop. След това тези схеми се визуализират и се сравнява тяхната производителност въз основа на разстоянието на Хаминг и чувствителността към повреди.

AHD (за да измери колко далеч са криптираните изходи от оригиналните). Това е средното Хамингово разстояние между оригиналната и криптираната схема. Това се изчислява за всяка входна комбинация чрез сравнение на изходите.

$$AHD = \frac{1}{M} \sum_{j=1}^m HD(O_{correct}^{(j)}, O_{incorrect}^{(j)}) \quad (3.1)$$

m- брой тестови вектори

O^j – изход за тестовия вектор j^{th}

- По-ниските стойности показват, че методът на криптиране запазва повече сходство с оригиналните данни, което потенциално го прави по-малко сигурен.
- По-високите стойности показват по-добра обфускация, което обикновено е по-сигурно.

Средна чувствителност към грешки (AFS) (за симулиране на въздействието на бит-флипове върху всеки метод на криптиране). Той се изчислява чрез симулиране на битови обръщания на входните стойности и измерване на промяната в изхода. Колкото по-висока е чувствителността към грешки, толкова повече криптирането се влияе от бит-флипове.

$$AFS = \frac{1}{N} \sum_{i=1}^N HD(O_{normal}^{(i)}, O_{faulted}^{(i)}) \quad (3.2)$$

N: Брой експерименти с инжектиране на разломи

$O_{normal}^{(i)}, O_{faulted}^{(i)}$: изход за тестовия вектор i^{th}

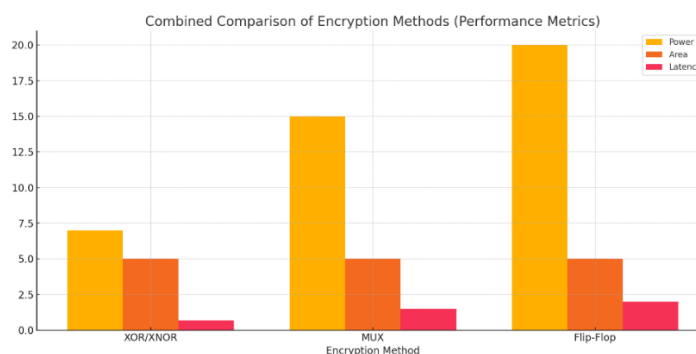
- По-високата чувствителност към грешки означава, че методът на криптиране е по-уязвим към бит-флип, което може да бъде проблем за реални приложения, където грешки могат да възникнат по време на предаване или съхранение.
- По-ниските стойности показват, че криптирането е по-устойчиво и по-малко засегнато от грешки.

Таблица 15 Резултатите за метода на криптиране

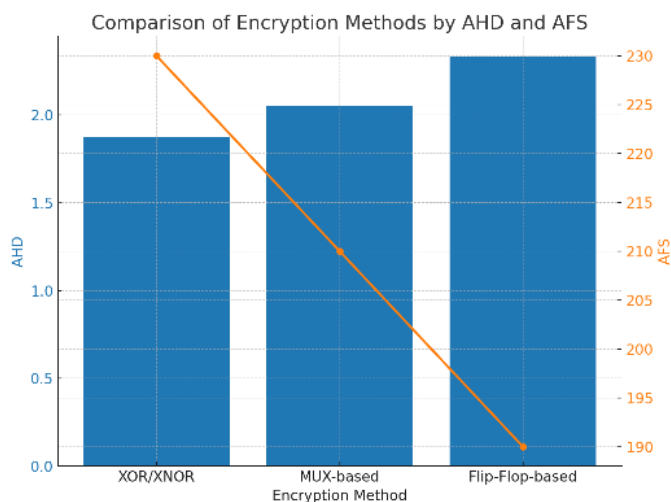
Метод на криптиране	AHD	AFS
XOR/XNOR	1.876	230
Базирано на MUX	2.053	210
Тригери	2.331	190

След симулиране на няколко типа логически схеми, резултатите са както в Таблица 15. Дефинираните параметри са обяснени по следния начин: Най-добрият метод за криптиране, както е показано на фигура 19, е криптирането, базирано на тригери (FLIP-FLOP), защото:

- Той има най-високото разстояние на Хаминг (по-добро замъгляване).
- Той има най-ниска чувствителност към разломи (по-устойчив на дефекти).



Фигура 19 Сравнение на методите за криптиране по показатели за производителност



Фигура 20 Сравнение на методите за криптиране по мертрики за сигурност

Както е показано на фигурите 19, 20. При XOR/XNOR криптирането, изразходваната мощност е най-ниска от всички методи, Площ: Минимална; Ефективен брой на елементите. Латентност: Най-бърз идеал за системи с ниски ресурси и висока скорост. Извод: Най-добро за леки приложения, където ефективността е ключова.

MUX-базирано криптиране, мощност: средно ниво, площ: подобно на XOR, но добавя сложност поради логиката на избора, латентност: умерена латентност от допълнителна логика. Извод: Добър баланс между силата на обфускацията и използването на ресурси. Криптиране, базирано на тригери, мощност: Най-високите тригери са елементи, държащи състояние, и консумират повече. Площ: Съвпада с останалите по брой елементи, но с по-тежки компоненти. Латентност: Най-бавна поради последователния характер и зависимостта от тактовия генератор. XOR/XNOR е оптимален по ефективност. MUX добавя сложност с управляеми разходи. Flip-Flop е сигурен, но с много ресурси, по-подходящ за критични системи, нуждаещи се от силна защита.

3.2 Предложен алгоритъм за оценка на хардуерната логика на криптиране

Входни данни:

$n_inputs \leftarrow$ Брой входове на схемата

$n_gates \leftarrow$ Брой елементи в кръга

$key_bits \leftarrow$ Ключове за криптиране

Резултат:

Криптирана верига, разстояние на Хаминг, оценка на грешки, графики на производителността

Процедури:

1. Дефинирайте логически функции на гейта:

$И(a, b) \leftarrow a И b$

$ИЛИ(a, b) \leftarrow a ИЛИ b$

$XOR(a, b) \leftarrow a XOR b$

$XNOR(a, b) \leftarrow НЕ(XOR b)$

$NOT(a) \leftarrow$ Обърнете a

$MUX(a, b, sel) \leftarrow a$ if $sel = 0$ else b

$D_FLIP_FLOP(d, clk, q) \leftarrow d$, ако $clk = 1$, иначе q

2. Генериране на случайна логическа схема:

$входове \leftarrow [x_0, x_1, \dots, x_n]$

$Проводници \leftarrow$ входове

$Кръг \leftarrow \emptyset$

За $i = 1$ до n_gates прави:

- $gate \leftarrow$ Случайно изберете от $\{AND, OR, XOR, XNOR\}$
 $(a, b) \leftarrow$ Случайни жици от жици
 $Излез \leftarrow "W" + I$
 Добавяне (гейт, a , b , out) към схемата
 Добавяне към кабелите
3. Прилагайте криптиране:
 - a. Криптиране, базирано на XOR/XNOR:

За всеки гейт в схемата:
 Ако гейтът е XOR или XNOR:
 Вмъкване на XOR/XNOR логика, зависима от ключове
 - b. Криптиране, базирано на MUX:

За всеки гейт в схемата:
 Вмъкнете MUX (a , b , key_bit) логика
 - в. Криптиране, базирано на флип-флоп:

За всеки гейт в схемата:
 Заменете изхода с $D_FLIP_FLOP(a, clk=1)$
 4. Оценка на веригата:

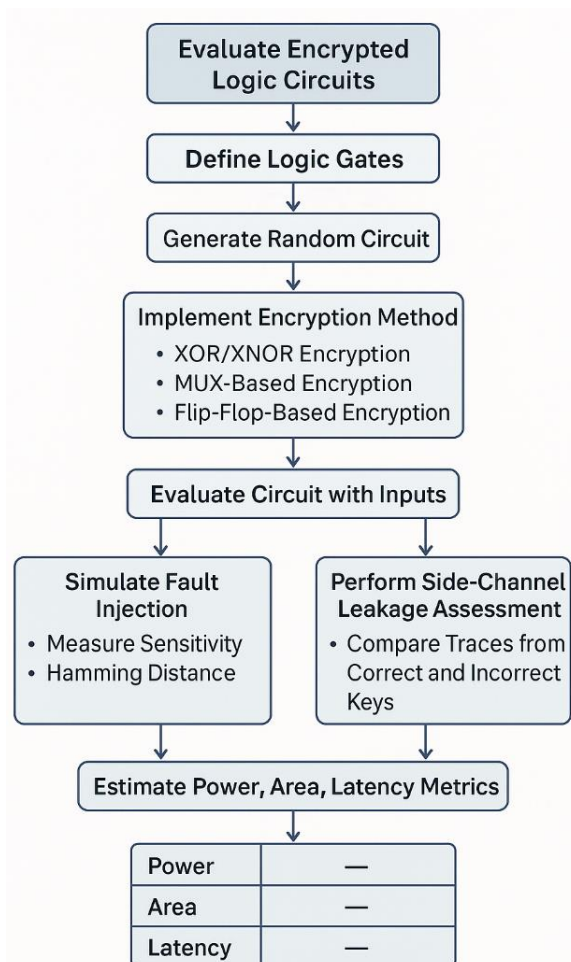
За всеки входен вектор в $\{0,1\}^n$:
 Оценете веригата с правилния ключ
 Стойност на изхода за съхранение
 5. Анализ на разстоянието на Хаминг:

За всеки входен вектор:
 Оценете изхода с правилен ключов $\rightarrow O_correct$
 Оценка на изхода с неправилен клавишен $\rightarrow O_wrong$
 Compute Hamming_Distance($O_correct$, O_wrong)
 6. Анализ на инжектиране на повреди:

$fault_score \leftarrow 0$
 За всеки входен вектор:
 Базова $\leftarrow$ оценка(схема, входове)
 За всеки входен бит:
 Обърнете $\leftarrow$ обърнете един входен бит
 Дефектен $\leftarrow$ оцени (верига, обръщане)
 $fault_score +=$ Hamming_Distance(базово, дефектно)
 7. Оценка на показателите за представяне:

Инициализирайте $gate_weights$:
 $AND/OR = 1$, $XOR/XNOR = 2$, $MUX = 3$, $D_FLIP_FLOP = 4$
 За всеки криптиран гейт в схемата:
 Нагрупуване:
 $Power \leftarrow \sum gate_weight$
 Площ $\leftarrow \sum$ фиксирана врата (например LUT или еквивалент на NAND)
 Латентност $\leftarrow \sum$ закъснението на ниво гейт за всеки тип
 8. Резултати от графика:

Сюжет:
 - Метод на захранване срещу криптиране
 - Метод за криптиране на площ срещу криптиране
 - Латентност срещу метод на криптиране
- Връщане:
 Криптирана верига,
 Резултат за дистанцията на Хаминг,
 Оценка за чувствителност към грешки,
 Графики на мощност, площ и латентност



Фигура 21 Алгоритъм описан с блокова схема

ТАлгоритъмът от фигура 21 предоставя систематична оценка на хардуерната логика на криптиране, като гарантира, че техниките за криптиране на схеми като XOR/XNOR, MUX-базирани и Flip-Flop-базирани, подобряват сигурността на схемите, като същевременно запазват ефективността на производителността. Методологията определя силата на криптирането количествено, което я прави ценна за проектантите на интегрални схеми, експерти по FPGA сигурност и инженери по вградени системи.

3.3 Симулиране на метода на криптиране върху схемите

Предложените техники за криптиране се оценяват с помощта на стандартни ISCAS бенчмарк схеми, които са широко използвани в литературата за валидиране на коректността на логическия проект, устойчивостта на грешки и механизмите за сигурност. По-конкретно, ISCAS-85 комбинационни схеми и ISCAS-89 последователни схеми се използват за оценка на ефективността на логическото криптиране както в комбинаторни, така и в архитектури зависими от състоянието. След получаване на бенчмарк нетлистите, схемите се парсират за извличане на логически елементи, взаимовръзки и елементи на тригера. За ISCAS-85 схемите симулацията включва директна оценка на комбинаторната логика чрез прилагане на булеви функции към входните вектори. За ISCAS-89 схемите последователното поведение се моделира изрично чрез включване на тактови актуализации на състоянието, при които съдържанието на тригера се обновява при всеки тактов цикъл според логиката на преходите на състоянието на схемата. Този симулационен подход, базиран на бенчмаркове, предоставя реалистична и мащабируема оценка на техниките за логическо криптиране, демонстрирайки тяхната приложимост към практически хардуерни проекти, използвани в сигурни комуникационни системи. Таблица 16 обобщава еталонните схеми ISCAS-85 и ISCAS-89, използвани в това изследване, обхващайки различни размери и структурни сложности, за да се осигури представителна оценка на предложените техники за криптиране.

Таблица 16 Обобщение на ISCAS бенчмарк схемите, използвани за оценка на криптирането

Бенчмарк Трасе	Тип схема	Брой входове	Брой изходи	Логически елементи (приблизително)
C432	Комбинаторен (ISCAS-85)	36	7	~160 елемента
C499	Комбинаторен (ISCAS-85)	41	32	~202 елемента
C880	Комбинаторен (ISCAS-85)	60	26	~383 елемента
C1355	Комбинаторен (ISCAS-85)	41	32	~546 елемента
S298	Последователен (ISCAS-89)	3	6	14 FF, ~100 елемента
S344	Последователен (ISCAS-89)	9	11	15 FF, ~160 елемента
S349	Последователен (ISCAS-89)	9	11	15 FF, ~170 елемента
S5378	Последователен (ISCAS-89)	35	49	179 FF, ~2 700 елемента

Изброените еталонни схеми принадлежат към широко използваните пакети ISCAS-85 и ISCAS-89, които са стандартни набори от данни за оценка на цифрови техники за проектиране, тестване и оптимизация. Бенчмарковете ISCAS-85 се състоят от комбинаторни схеми, докато тестовите ISCAS-89 включват последователни схеми с паметни елементи като тригери.

3.4 Блок-схема за цялостна оценка на мерки за сигурност и производителност на логически криптирани бенчмарк схеми

Работният процес след това се разклонява към етап на интеграция на криптиране, където се генерират три криптирани варианта на схемата от базовия проект:

- 1) Логическо криптиране, базирано на XOR/XNOR,
- 2) Логическо криптиране, базирано на MUX,
- 3) Логическо криптиране, базирано на тригери.

Всяка криптирана схема включва ключово-зависими логически модификации, като същевременно запазва функционалната коректност под валидния ключ.

След въвеждането на криптиране, потокът продължава към блока за оценка на сигурността, който съдържа три паралелни пътя за анализ:

- **Анализ на разстоянието чрез Хаминг:** Изходите, произведени при правилния ключ, се сравняват с изходи, генерирани с неправилен ключовш. AND се изчислява за всички тестови вектори, за да се количествено определи функционалната обфускация.
- **Анализ на инжектиране на грешки:** Контролирани грешки (например обръщане на входни битове или блокирани грешки) се инжектират в криптираните вериги. Изходните отклонения се измерват, за да се изчисли AFS, отразявайки устойчивостта към атаки, базирани на грешки.
- **Анализ на чувствителността на ключовете:** Отделните битове на ключа се обръщат, докато входовете остават постоянни. Изходната вариация се измерва, за да се оцени чувствителността към малки промени в тоналността.

Паралелно потокът преминава към блока за оценка на производителността, където всеки криптиран проект се синтезира и реализира чрез FPGA или ASIC инструменти. От докладите след синтез и след разполагането и маршрутизацията се извличат и нормализират консумацията на енергия (статична и динамична), използването на площта и критичното закъснение на пътя спрямо базовия проект. Работният процес навлиза в етапа на оценка на SAT атаките, където нетлистите на ниво гейт на криптираните вериги се подлагат на SAT-базирани деобфускационни атаки. Метрики като време на изпълнение на SAT solver, брой оракулни заявки и тайм-аут събития се записват, за да се определи количествено устойчивостта срещу алгоритмични атаки. Всички изчислени метрики се събират в етапа на агрегиране на резултатите, където данните по бенчмарк и метод за криптиране се компилират в сравнителни таблици и графики. Тези резултати позволяват сравнение между методите и подкрепят анализ на компромиса между сигурност и производителност.

3.5 Алгоритъм: Решение за внедряване, базирано на разходите за сигурност, за логически криптиран комуникационен хардуер

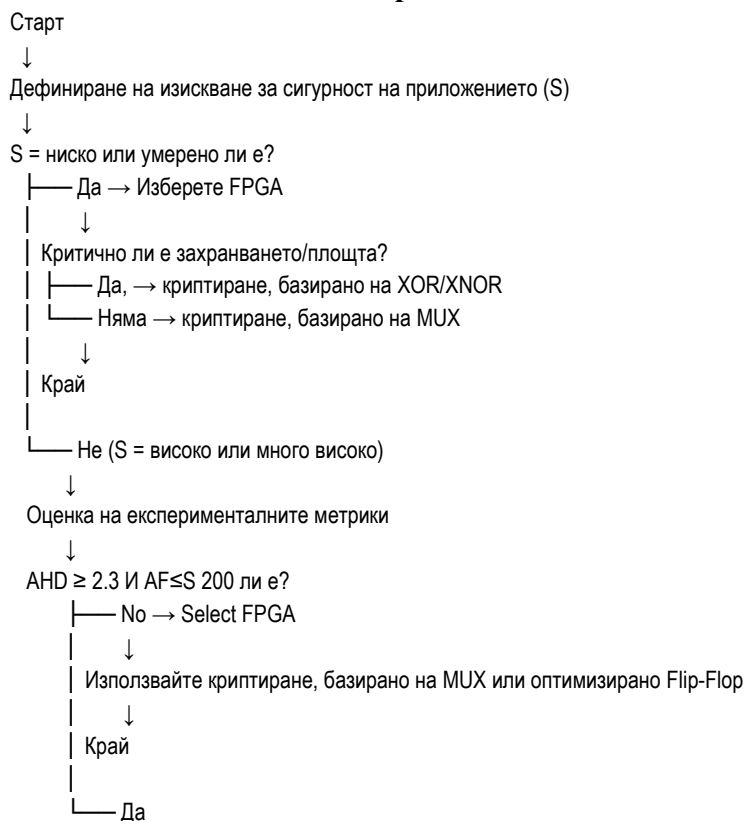
Вход

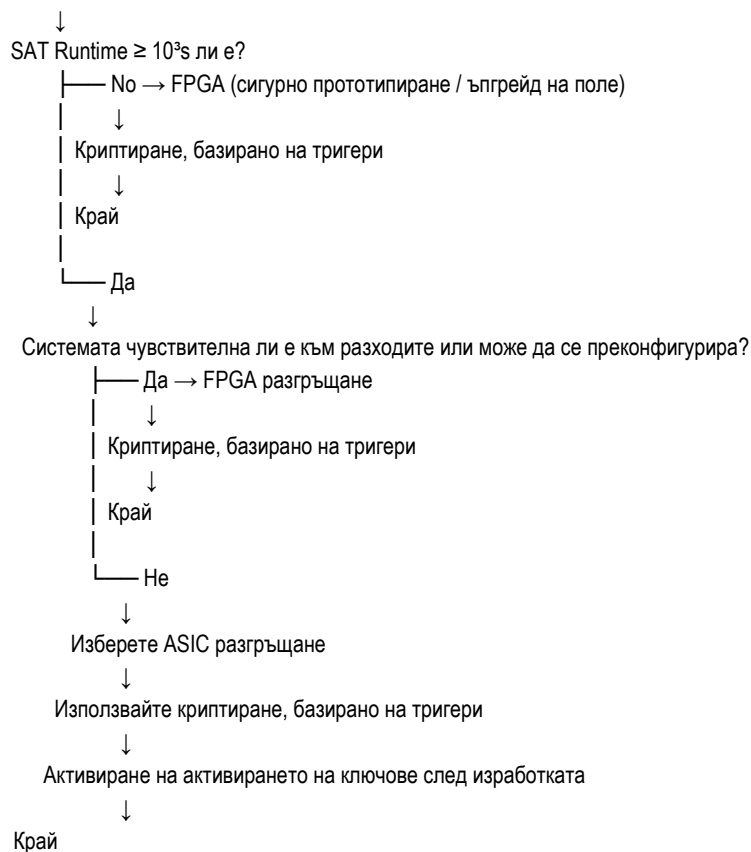
- Изисквания за сигурност S
- Ограничения на разходите и гъвкавостта C
- Експериментални метрики за сигурност:
 - Средно разстояние на Хаминг (AHD)
 - Средна чувствителност към повреди (AFS)
 - TSAT по време на изпълнение на SAT атака

Изход

- Препоръчителна платформа за внедряване (FPGA или ASIC)
- Препоръчителна техника за логическо криптиране

Последователност на вземане на решения





Алгоритъмът формализира процес на вземане на решения, базиран на сигурността, базиран на разходите, за внедряване на логически криптиран комуникационен хардуер, като съвместно взема предвид изискванията за сигурност на приложенията, ограниченията на имплементацията и експериментално измерените метрики за сигурност. Алгоритъмът взема като входни данни нивото на целева сигурност, ограниченията за цена и преконфигурируемост, както и количествените метрики, получени от оценката, а именно AHD, AFS и SAT атака за изпълнение. Праговете за тези метрики се определят въз основа на емпирични резултати, където $AHD \geq 2.3$ показва силна функционална обфускация, $AFS \leq 200$ отразява висока устойчивост на повреди, а времето на изпълнение на $SAT \geq 10^3$ секунди означава значителна устойчивост на алгоритмични атаки за де-обфускация. За приложения с ниска до умерена сигурност се избират леки методи за криптиране, за да се минимизира натоварването. За разлика от това, когато е необходима висока или много висока сигурност и всички прагове на метриката са изпълнени, алгоритъмът приоритизира криптирането, базирано на Flip-Flop, поради поведението, зависещо от състоянието, което значително увеличава корумпирането на изхода при грешни ключове, намалява наблюдаемостта на грешки и значително увеличава сложността на SAT атаките. При тези условия се препоръчва внедряването на ASIC, когато ограниченията на разходите и реконфигурируемостта позволяват, тъй като постоянно логическо свързване и активирането на ключове след изработка допълнително засилват гаранциите за сигурност.

ГЛАВА IV: Оценка на фактора на корупция и секретността при хардуерно логическо криптиране с многобитови ключове

В тази глава описва симулационната среда и процеса на оценка, включително настройката, инструментите и бенчмарковете, използвани за тестване на предложените проекти. Той представя резултатите, сравнявайки представянето, здравината на сигурността и разходите спрямо съществуващите техники.

4.1 Експериментална настройка. Хардуерна криптираща логика

За всички техники за криптиране местоположението на ключовия елемент се променя, за да се провери влиянието на местоположението и типа на криптиращия гейт върху сигурността на схемата. Целта на тази глава е да създаде експериментален тестов стенд за изследване на схеми с различни видове криптиращи елементи, базирани на XOR/XNOR, MUX и Flip-Flop. Целта на изследването е да измери разстоянията на Хаминг при неправилни активационни ключове между грешния и правилния изход, фактора на корупция и секретността. Всички изследвани схеми имат идентична промяна в структурната сложност. Първо, логическите схеми се описват на VHDL език, а по-късно са реализирани 3 типа техники за криптираща логика: криптиране, базирано на XOR/XNOR, криптиране, базирано на MUX, техника Encrypt flip-flop. Реализирани са следните стъпки:

- 1) Разработване на логически криптиращи елементи с помощта на VHDL.
- 2) Разработване на тестови платформи за логически криптиращи елементи.
- 3) Успешно симулиране на проектирани логически криптиращи елементи.
- 4) Разработване на схема чрез сглобяване на логическите криптиращи елементи.

4.2 Логическо криптиране, базирано на анализ на грешки

Сега ще разгледаме как използваме ключовите елементи (като XOR/XNOR) за криптиране на проекта, така че всеки грешен ключ да води до грешен изход. Това е сравнимо със ситуацията, при която схема с дефект, който е възбуден и разпространен към изходите, генерира неправилен изход. Анализът на грешки при тестването на интегрални схеми и логическото криптиране са свързани в следните наблюдения. Тези наблюдения ще се използват за добавяне на елементите XOR/XNOR.

4.3 Криптиране, базирано на XOR/XNOR,

Техниката за криптиране, базирана на XOR/XNOR, вмъква елементи XOR/XNOR (ключови гейтове) в проекта на случаен принцип. При тази техника се променя местоположението на криптиращия елемент, за да се наблюдава как се променят Хаминговото разстояние (HD) [6], факторът на корупция и измамата. Два типа схеми са криптирани: AND/OR (една таблица на истинността) и AND/OR/XOR (таблица с истина с 2 превключвания), за да се определи коя от тях е по-лесна за криптиране. Стойностите на правилния ключов бит за XOR/XNOR криптиращите гейтове се извеждат директно от таблицата на тротовете и именно тази стойност позволява втория запис на изхода. Правилният ключов бит за XOR гейта е "0", а за XNOR гейта е "1". Причината за използването на ключови елементи при тази техника е, че при използване на правилния ключ на входа на схемата се гарантира правилната функционалност на схемата. В случай на намеса от неоторизирани лица с грешен ключ, това ще доведе до грешен изход.

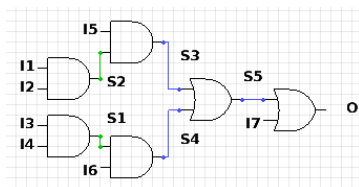
Схеми без криптиране

Първият проект на Фигура 29 няма реализирано криптиране и се състои от комбинаторни компоненти. Първоначалната схема се състои от 8 входно-изходни (7 входа и 1 изход). Състои се от 4 AND, 2 OR логически елемента. След успешната оценка на VHDL проекта, той беше реализиран на развойната система Zed Board. След програмирането на VHDL кода в FPGA устройството, RTL схемите се синтезират в инструмента VIVADO. Вторите оригинални схеми са в конфигурацията на Фигура 5 и тя е съставена от AND/OR/XOR гейтове, където XOR има таблица за истинност с 2 превключвания.

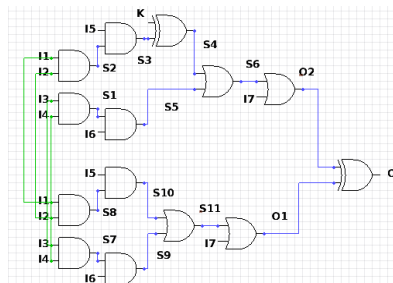
$$SHS_{cc} = \frac{1}{6} = 0.166 \quad (4)$$

- XOR/XNOR криптиране с 1 бит ключ

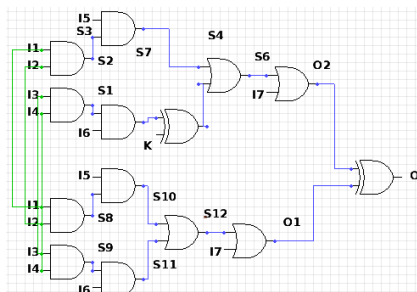
Първата техника за криптиране е базираната на XOR/XNOR, която вмъква на случаен принцип елементи XOR/XNOR (ключови гейтове) в проекта. При тази техника за логическо криптиране ще променим позицията на различните ключови врати, не сме променяли броя на вратите. Причината за използването на ключовите врати е да се гарантира, че правилните ключове се използват като входове, и да се гарантира, че функционалността на схемата е правилна. В случай на неупълномощен потребител, грешният клавиш на входа ще осигури грешна функция на изхода. За да се тества тази техника в оригиналните схеми, се вмъкват, симулират и оценяват елементите XOR с ключов бит К и различни места, виж Фигури 29, 30 и 31.



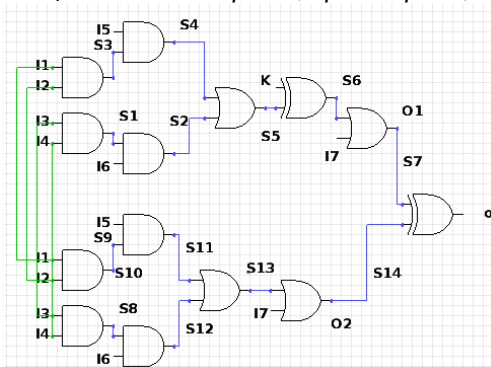
Фигура 29 Начална схема без криптиране



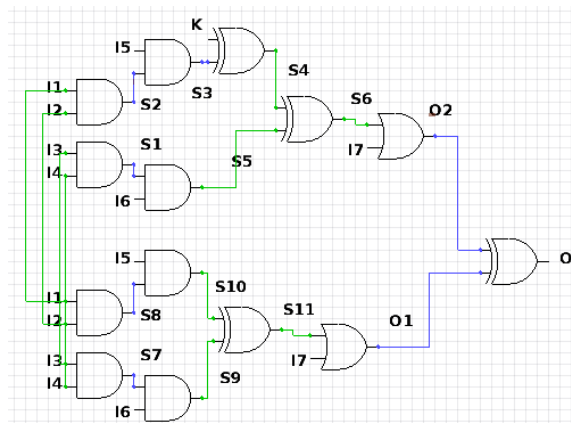
Фигура 30 Първа позиция на 1 XOR гейт, криптиране, базирано на XOR/XNOR



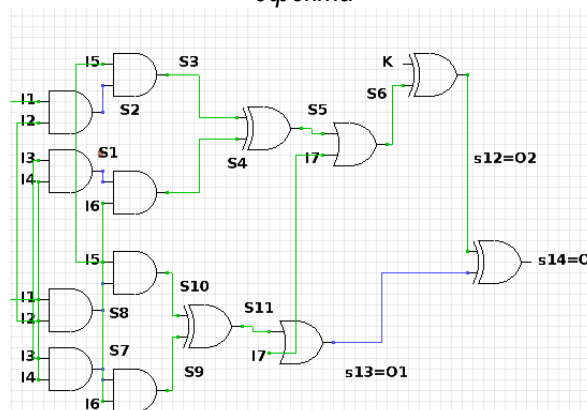
Фигура 31 Втората позиция на 1 XOR врата, криптиране, базирано на XOR/XNOR



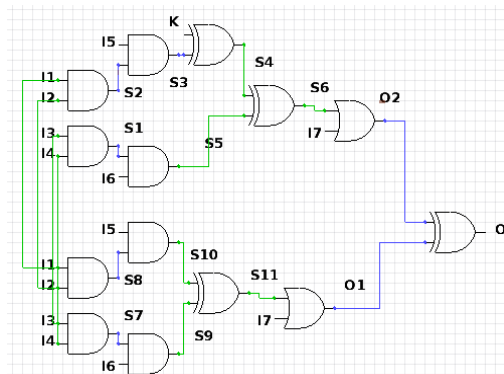
Фигура 32 Трета позиция на елемент XOR, криптиране, базирано на XOR/XNOR



Фигура 33 В първата криптирана схема сменяме гейт от OR на XOR, за да видим промяната и ефекта



Фигура 34 Втората позиция на 1 XOR гейт се променя с криптиране, базирано на XOR, XOR/XNOR



Фигура 35 Схема с криптиране (XOR гейт), В първата криптирана схема се променя гейт от OR на XOR.

Първата схема не е криптирана и се състои от редица последователни и конкатенирани компоненти. 7 входа и 1 изход съставляват 8-те I/O в основната схема. Както се вижда на Фигура 35, той се състои от четири AND врати и два OR гейта. RTL схемите се създаваха, след като VHDL кодът беше реализиран във FPGA устройството.

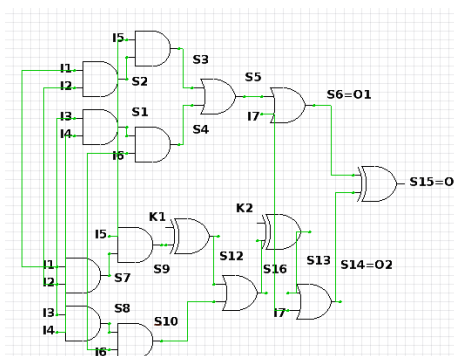
Структурни промени в сложността при сценарии с 2-битово – ключ:

$$SHS_{cc} = \frac{2}{6} = 0.3 \quad (6)$$

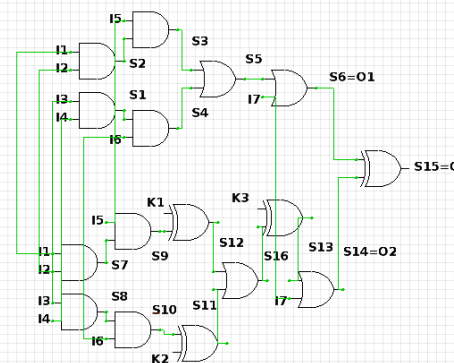
Структурни промени в сложността при сценарии с 3-битово –ключово криптиране:

$$SHS_{cc} = \frac{3}{6} = 0.5 \quad (7)$$

Следвайки поведенческата симулация и логическото проектиране на схемата във VHDL, примерите за оценка на сигурността на различни хардуерни методи за криптиране на логика се симулират и оценяват по отношение на фактори на корупция, измамност и секретност. Първият сценарий за криптиране включва добавяне на два XOR гейта с ключове K1, K2, както е показано на Фигура 36. Блок-диаграмата на Фигура 36 илюстрира как методът на криптиране се разработва чрез произволно вмъкване на XOR/XNOR гейтове с два ключови гейта. Чрез поведенческо моделиране се изследва функционалността на криптиращата схема, базирана на XOR/XNOR. В следващия сценарий 3 XOR/XNOR гейта (ключови гейтове) се въвеждат на случаен принцип в схемата, за да се тества методът на криптиране XOR/XNOR. Схемата на Фигура 37 е създадена чрез три отделни ключови гейта – K1, K2 и K3 – в тази техника за логическо криптиране.



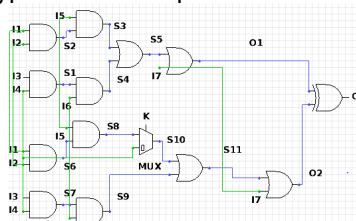
Фигура 36: 2-ключово криптиране, базирано на XOR/XNOR.



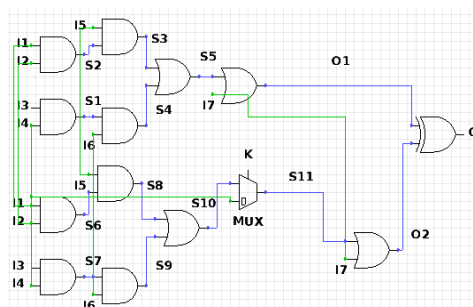
Фигура 37 3-битов ключ за криптиране, базирано на XOR/XNOR,

MUX-базирано криптиране

В този случай, при методите за криптиране, базирани на MUX, се използват 2 входни и 1 контролни битови мултиплексори. В този случай контролният бит на MUX ще действа като ключов бит, а правилните и грешните стойности на ключовия бит се определят от VHDL описанието на мултиплексора. Един от входовете на мултиплексора е правилният сигнал, а другият вход може да идва от всяка друга точка на схемата. Ако няма разрешено използване на веригата, ще се активира грешен вход и резултатите от изходите ще са грешни. Различните места на MUX гейта се оценяват, за да се види как това влияе на сигурността на веригите.

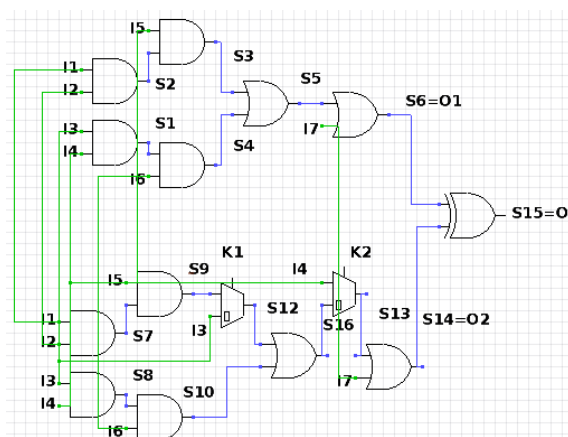


Фигура 38 Първа позиция на 1 MUX портал, криптиране, базирано на MUX

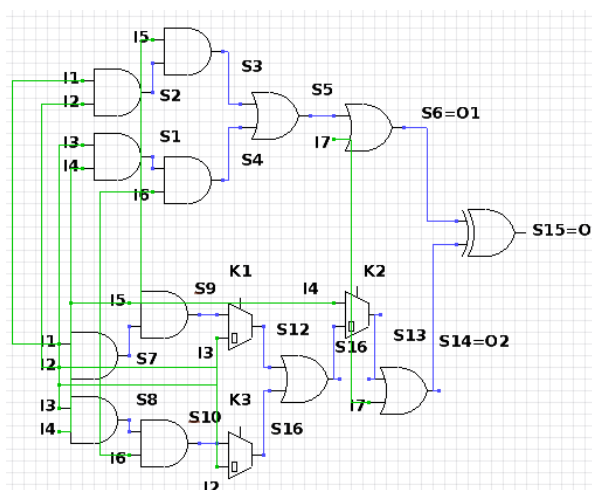


Фигура 39 Втората позиция на 1 MUX портал, криптиране, базирано на MUX

След завършване на внедряването на криптиране, базирано на XOR/XNOR, се симулират и оценяват методите за криптиране, базирани на MUX, както при предишната техника, се добавят 2 и повече мултиплексора за криптиране. Генерират се различни ключови елементи. Въпреки това е трудно да се определи типът на ключовия елемент. Трудността при разграничаването на "истинските връзки" от "фалшивите" определя колко сигурен е методът, базиран на MUX.



Фигура 40 2-битово криптиране, базирано на MUX ключ

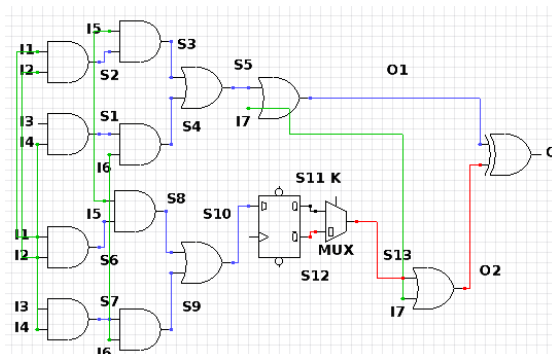


Фигура 41 3-битово криптиране, базирано на MUX ключ

Техника за криптиране на тригера

Методът за криптиране на тригера изисква двата изхода на FF да бъдат свързани с 2 входни MUX. В този случай ключът за сигурност се прилага като вход за избор на линия на MUX. Чрез използване на грешен защитен ключ, MUX ще позволи грешен вход на изхода на схемата. Това свързване на MUX

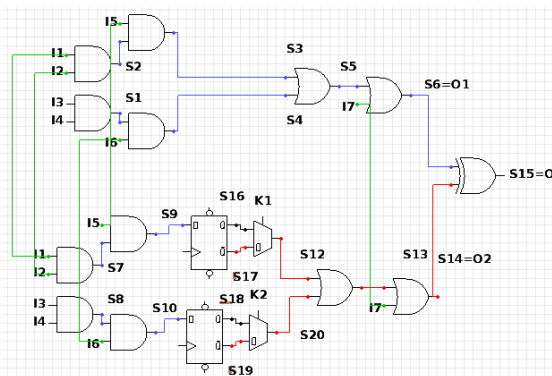
с D-FF е направено, за да не се случи ситуация, че дори с грешен ключ може да получите правилния резултат на изхода на схемата.



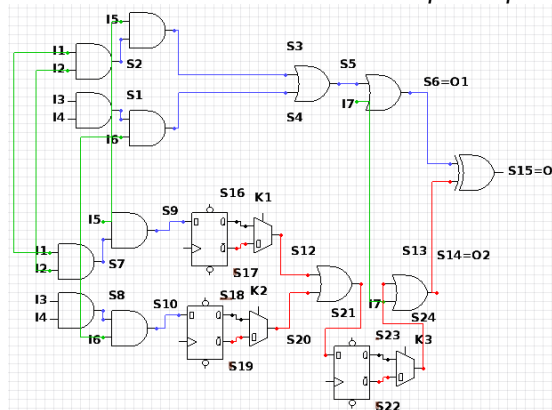
Фигура 42 Техника за криптиране на тригера

При техниката за криптиране на тригера (FF), за да се използват два FF изхода, трябва да бъде свързана с мултиплекс с 2 входа. В този случай входът select line на MUX се прилага с ключа за сигурност. Ако се използва грешен защитен ключ, мултиплексът ще пусне грешния вход на изхода на веригата.

MUX е свързан към D-FF по този начин, за да се избегне възможността да натиснете грешен клавиш и въпреки това да получите желаните резултат на изхода на схемата.



Фигура 43 2-битов ключ FLIP-FLOP-базирано криптиране



Фигура 44 3-битово криптиране, базирано на FLIP-FLOP

НАУЧНИ, НАУЧНО-ПРИЛОЖНИ И ПРИЛОЖНИ ПРИНОСИ

Научните приноси на тази дисертация са:

- 1) Комплексна рамка за оценка на логическо криптиране. Разработена е унифицирана рамка за количествена оценка на логически криптирани схеми, използвайки функционални, базирани на грешки и SAT-базирани показатели за сигурност, както и хардуерни разходи за производителност.
- 2) Въвеждане на композитен рейтинг за сигурност. Предлага се нова композитна метрика за интегриране на множество измерения за сигурност в един резултат, позволяваща директно сравнение между бенчмаркове, стилове на криптиране и плътности на ключови елементи.
- 3) Демонстрация на предимства на криптирането, зависимо от състоянието. Експерименталните резултати потвърждават, че криптирането, базирано на тригери и зависимо от състояние, осигурява по-добра устойчивост на инжектиране на грешки и атаки, базирани на SAT, в сравнение с чисто комбинационните схеми.
- 4) Алгоритъм за вземане на решения, базирани на сигурността и разходите. Предлага се алгоритъм, базиран на праг, който превежда експерименталните метрики за сигурност в детерминирани решения за внедряване на FPGA и ASIC.

Научно-приложните приноси на дисертацията са:

- 5) Сравнителен анализ на техники за логическо криптиране. Структурираното сравнение на техниките за криптиране, базирани на XOR/XNOR, базирани на MUX и Flip-Flop, е представено както в комбинаторни, така и в последователни еталонни схеми.
- 6) Анализ на мащабируемостта на плътността на криптирането. Влиянието на плътността на ключовите елементи върху здравината на сигурността и разходите за реализация се анализира систематично за конфигурации на 2, 5, 10 и 16 ключови елемента.

Приложните приноси на дисертацията са:

- 7) Валидиране, базирано на бенчмарк, с използване на ISCAS схеми. Рамката е валидирана на стандартни ISCAS-85 и ISCAS-89 еталонни схеми, което гарантира възпроизводимост и релевантност за реални дигитални дизайни.
- 8) Хардуерни модели, готови за имплементация. Предоставени са синтезируеми VHDL реализации на всички техники за криптиране, което улеснява интеграцията в практически комуникационни хардуерни подсистеми.

СПИСЪК НА ПУБЛИКАЦИИТЕ ПО ДИСЕРТАЦИЯТА

Публикации:

1. E. Malaj, G.Marinova, Review on hardware solutions for cybersecurity of communication systems, 2020 28th National Conference with International Participation (TELECOM), Date of Conference: 29-30 Oct. 2020, Date Added to IEEE Xplore: 25 December 2020, INSPEC Accession Number: 20304410, DOI: 10.1109/TELCOM50385.2020.9299537, Publisher: IEEE, Conference Location: Sofia, Bulgaria], pp.129-132, IEEE Scopus. Indexed in IEEE Xplore, Scopus <https://ieeexplore.ieee.org/document/9299537>
2. E. Malaj, G.Marinova, Simulated algorithm RSA in Mail Server device, Published in 2021 29th National Conference with International Participation (TELECOM), Date of Conference: 28-29 Oct. 2021, Sofia, Bulgaria, IEEE-Xplore, IEEE Scopus. Indexed in IEEE Xplore, Scopus <https://ieeexplore.ieee.org/document/9659623>

3. E. Malaj, G. Marinova, Review of cybersecurity hardware devices, The International Conference on Science, Technology, Business and Innovation of UBT assembles in Kosovo, Date of Conference: 04-05 Oct. 2021. <https://knowledgecenter.ubt-uni.net/cgi/viewcontent.cgi?article=3183&context=conference>.
4. E. G. Malaj, M. Ibro and G. Marinova, "Experimental setup for Hardware Encryption logic evaluation for e-learning ", 2022 29th International Conference on Systems, Signals and Image Processing (IWSSIP), Sofia, Bulgaria, 2022, pp. 1-4, DOI: 10.1109/IWSSIP55020.2022.9854421. Conference Paper | Publisher: IEEE | Electronic ISBN: 978-1-6654-9578-3 Cited by: Papers (1), IEEE Scopus and WoS Indexed in IEEEExplore, Scopus <https://ieeexplore.ieee.org/document/9854421>.
5. G. Marinova, E. G. Malaj, & M. Ibro, (2023) "Hardware Solution for Cybersecurity of Unmanned Systems Critical Infrastructure", In Monitoring and Protection of Critical Infrastructure by Unmanned Systems, Vol. 63, pp. 127-138. iOS Press. DOI 10.3233/NICSP230010 Book Chapter | Publisher: IOS Press, IEEE Scopus. Indexed in IEEEExplore, Scopus <https://ebooks.iospress.nl/doi/10.3233/NICSP230010>.
6. E. G. Malaj, G. Marinova, "Effect of the Hardware Logical Encryption Type and Location on the Circuit Security ", 2023 IEEE International Workshop on Technologies for Defense and Security (TechDefense), 2023 IEEE Scopus. Indexed in IEEEExplore, Scopus <https://ieeexplore.ieee.org/document/10380923>
7. E. G. Malaj, G. Marinova, (2023) "Study of the Secrecy in Hardware Logical Encryption With Multiple Bit Keys", 2023 IEEE International Workshop on Technologies for Defense and Security (TechDefense), 2023, IEEE Scopus. Indexed in IEEEExplore, Scopus <https://ieeexplore.ieee.org/document/10380831>
8. E. G. Malaj, G. Marinova, (2024) "Identification Hardware Attacks of Integrated Circuit Design Data and Intellectual Property", 2024 IEEE International Workshop on Technologies for Defense and Security (TechDefense), 2024, IEEE Scopus. Indexed in IEEEExplore, Scopus <https://ieeexplore.ieee.org/document/10863477>
9. E. G. Malaj, G. Marinova, (2025) "Review on Hardware Logical Encryption" 2025 IEEE International Workshop on Technologies for Defense and Security (TechDefense), 2025, IEEE Scopus Indexed in IEEEExplore, Scopus
10. E. G. Malaj, (2025) "Algorithmic Framework for Evaluating Hardware Encryption Circuits" 2025 IEEE International Workshop on Technologies for Defense and Security (TechDefense), 2025, IEEE Scopus. Indexed in IEEEExplore, Scopus

SUMMARY

DEVELOPMENT OF NEW HARDWARE SOLUTIONS FOR CYBERSECURITY OF COMMUNICATION SYSTEMS

This thesis investigated the problem of securing communication-oriented hardware systems against reverse engineering, fault injection, and algorithmic de-obfuscation attacks through the use of logic encryption. With the increasing globalization of IC design and manufacturing, hardware intellectual property has become a critical security concern, particularly for communication infrastructure, embedded systems, and safety-critical applications. The central objective of this research was to develop and evaluate a systematic, quantitative framework for assessing the effectiveness and deployment feasibility of logic-encrypted hardware.

To address this objective, the dissertation proposed a unified evaluation methodology that combines functional obfuscation, fault resilience, key sensitivity, SAT attack resistance, and hardware implementation cost. Three representative logic encryption techniques XOR/XNOR-based, MUX-based, and Flip-Flop-based encryption were implemented and evaluated across a wide range of ISCAS-85 combinational and ISCAS-89 sequential benchmark circuits. Encryption density was systematically scaled using 2, 5, 10, and 16 key-gate configurations to capture the security–cost trade-off inherent in logic encryption design.

The experimental results demonstrate that logic encryption effectiveness is strongly dependent on both encryption density and circuit structure. Increasing the number of key-gates consistently improves functional obfuscation, key sensitivity, and resistance to SAT-based attacks. However, these security gains are accompanied by predictable increases in power consumption, area overhead, and delay.

A major finding of this work is the superior security scaling behavior of sequential, state-dependent encryption mechanisms. Flip-Flop-based encryption consistently outperformed purely combinational locking schemes across all benchmarks, particularly for large sequential circuits. The presence of memory elements amplifies key-dependent behavior over time, significantly increasing attack complexity and reducing fault observability.

To enable holistic comparison across benchmarks and encryption strategies, this dissertation introduced a Composite Security Score that integrates multiple security metrics into a single, deployment-oriented measure. This score proved effective in differentiating lightweight, moderate, and high-assurance security configurations and directly supported the formulation of deployment thresholds for FPGA and ASIC platforms.

One of the primary contributions of this thesis is the translation of experimental security analysis into actionable deployment guidelines. The proposed Security Cost Driven Deployment Decision Algorithm maps measured Composite Security Scores to recommended encryption densities and hardware platforms. Low scores favor FPGA deployment with sparse encryption for cost-sensitive applications, while high scores mandate high-density or sequential encryption and ASIC deployment for mission-critical systems.

In conclusion, this thesis demonstrates that effective hardware security cannot be achieved through isolated design choices or single-metric optimization. Instead, robust protection of communication-oriented hardware systems requires a balanced integration of encryption density, circuit structure, and platform-specific constraints. By providing a systematic evaluation framework, quantitative deployment thresholds, and validated experimental results, this research establishes a solid foundation for the design and deployment of secure hardware in an increasingly adversarial environment. The integration of threshold-based decision rules, supported by experimental data and visualized through flowchart-based deployment models, bridges the gap between academic security evaluation and real-world hardware design practice. This approach enables designers to make informed, metric-driven decisions rather than relying on ad-hoc or heuristic security assumptions. The methodologies and insights presented in this work are expected to remain relevant as hardware security challenges continue to evolve, and they provide a strong basis for future research into adaptive, side-channel-aware, and system-level hardware protection mechanisms.



TECHNICAL UNIVERSITY OF SOFIA
Faculty of Telecommunications
Department “Technologies and Management of
Communication Systems”

MSc. Eriselda Guri Malaj

DEVELOPMENT OF NEW HARDWARE SOLUTIONS FOR
CYBERSECURITY OF COMMUNICATION SYSTEMS

EXTENDED SUMMARY (AVTOREFERAT)

of a dissertation for obtaining the educational and scientific degree "DOCTOR"

Field: 5. Technical sciences

Professional field: 5.3 Communication and Computer Engineering

Scientific specialty: Computer-aided Engineering and Systems for Design
Automation.

Scientific supervisor: Assoc. Prof. Dr. Galia Marinova

Sofia, 2026

The dissertation is discussed and directed for defense by the Department Council of the Department of “Technologies and Management of Communication Systems” at the Faculty of Telecommunications of TU-Sofia at a regular meeting held on 11.06.2026.

The public defense of the dissertation will take place on 01.10.2026 from 15:00 hours in the Conference Hall of the BIC of the Technical University - Sofia at an open meeting of the scientific jury, determined by order № ОЖ – 5.3 – 55 / 24.06..2026 of the Rector of TU-Sofia composed of

1. Assoc. Prof. Ph.D. Tihomir Brusev - President
2. Assoc. Prof. Ph.D. Rostislav Rusev - Scientific Secretary
3. Prof. DSc. Todor Stoilov
4. Assoc. Prof. Ph.D. Adriana Borodzhieva
5. Assoc. Prof. Ph.D. Valentin Mutkov

Reviewers:

1. Assoc. Prof. Ph.D. Tihomir Brusev
2. Assoc. Prof. Ph.D. Adriana Borodzhieva

The materials on the defense are available to those interested in the office of the Faculty of Telecommunications of TU-Sofia, block №1, office № 1306.

The thesis belongs to a part-time doctoral student in the Department of Technology and Management of Communication Systems of the Faculty of Telecommunications. The research in this dissertation was conducted by the author, and some of it was supported by research projects.

Author: MSc. Eriselda Malaj

Title: Development of new hardware solutions for cybersecurity of communication systems.

Circulation: 10 pieces

Printed in the IPC of the Technical University – Sofia

I. GENERAL CHARACTERISTICS OF THE DISSERTATION

Relevance of the problem

Logic encryption is a technique designed to enhance the security of digital circuits by embedding encryption mechanisms within their logic gates. This approach aims to protect intellectual property and secure integrated circuits (ICs) against reverse engineering and hardware tampering. The algorithm proposed in this work leverages logic encryption to achieve robust protection for digital designs by incorporating encryption directly into the circuit's logic structure.

The core of our algorithm involves embedding cryptographic transformations within the logic gates of a circuit, creating a secure and obfuscated design. This is accomplished through the application of a combination of Boolean functions and cryptographic keys to alter the functionality of the gates. The resulting encrypted circuit is resistant to unauthorized access and reverse engineering, as the logic functionality is obscured and only accessible to those with the correct decryption keys.

Our approach also includes mechanisms for key management and distribution, ensuring that the encryption keys are securely handled and accessed only by authorized entities. We demonstrate the effectiveness of the proposed algorithm through a series of simulations and evaluations, highlighting its resilience against common attack vectors such as reverse engineering and probing.

Overall, the proposed logic encryption algorithm provides a significant enhancement to the security of digital circuits, safeguarding against intellectual property theft and hardware-based attacks while maintaining functionality and performance. This contribution advances the field of hardware security by introducing a practical and effective method for protecting integrated circuits through logic encryption.

The growing complexity and interconnectivity of modern communication systems have intensified the need for robust, low-latency security mechanisms. Traditional software-based encryption, while flexible, is increasingly vulnerable to hardware-level threats such as side-channel attacks, reverse engineering, and logic extraction. This thesis presents the development, simulation, and evaluation of novel hardware-level encryption techniques aimed at enhancing the cybersecurity of communication infrastructures.

Three logic-based encryption methods are investigated: XOR/XNOR-based encryption, MUX-based obfuscation, and Flip-Flop-based sequential encryption. These techniques are implemented in Verilog/VHDL and integrated into standard benchmark circuits and communication submodules. A comprehensive evaluation framework is developed using synthesis and simulation tools (ModelSim, Vivado, Quartus), and performance is assessed through critical metrics: power consumption, area overhead, delay, Average Hamming Distance (AHD), and Average Fault Sensitivity (AFS).

Results demonstrate clear trade-offs between efficiency and security. XOR/XNOR encryption offers minimal resource usage but limited resistance to attacks. MUX-based designs improve obfuscation with moderate overhead, while Flip-Flop-based encryption provides the highest protection for sequential logic at the cost of increased complexity. The findings support tailored deployment of encryption logic depending on the application's security and performance requirements.

This thesis contributes a validated methodology for secure hardware design in communication systems and proposes guidelines for future enhancements through hybrid architectures, AI-based testing, and integration with quantum-resilient cryptographic primitives.

Existing logic encryption methods face challenges like poor standardization, limited scalability, and trade-offs between security and performance (area, delay, power). Many are either too costly for real-time systems or not strong enough against advanced attacks like SAT solvers, probing, and fault injection. However, despite its potential, several key issues remain unresolved in the application of logic encryption:

- **Effective Integration:** How can logic encryption be effectively integrated into the design process of digital circuits without compromising their performance or functionality?
- **Resistance to Attacks:** To what extent can logic encryption withstand advanced reverse engineering techniques and probing attacks, and how can its resilience be quantified and enhanced?
- **Key Management:** What are the best practices for managing and distributing encryption keys in a way that ensures both security and practicality, especially in large-scale deployments?
- **Design Overhead:** How can the trade-offs between security and design overhead be optimized to minimize the impact on the circuit's area, power consumption, and speed?

This thesis addresses these challenges by proposing a novel logic encryption algorithm that aims to enhance the security of digital circuits while maintaining practical design constraints. The research focuses on developing efficient methods for integrating encryption into logic gates, improving resistance to various attack vectors, and establishing robust key management practices. Through a combination of theoretical analysis, simulation, and empirical validation, this work seeks to advance the state of the art in hardware security and provide practical solutions for protecting integrated circuits from emerging threats. The remainder of this dissertation is structured to provide a comprehensive exploration of hardware security through logic encryption, progressing from foundational concepts to practical implementation and analysis.

Objective of the dissertation, main tasks and research methods

Based on the synthesis of the reviewed literature, the thesis identifies key challenges and gaps in the field of logical encryption. These include:

- Resistance to new attack methodologies, such as those arising from advances in quantum computing.
- Key management challenges and the need for more efficient and scalable encryption mechanisms.

The objective of the thesis is to design hardware encryption modules using logic structures, to implement them in communication-related hardware systems, to evaluate and compare their performance using key hardware metrics (power, area, delay), to assess the security strength of each method using metrics such as AHD, AFS, and SAT attack resistance and to propose a new hardware encryption evaluation framework applicable to communication subsystems.

To achieve this goal the following tasks are defined:

- A) To define logic modules and logic structures for encryption. To implement hardware description languages for the communication-related hardware systems and creation of an evaluation set-up.
- B) To define the performance limitations of current hardware encryption approaches in communication circuits
- B) To evaluate how different logic-level encryption techniques compare in terms of area, delay, and power overhead?
- Γ) To determine the technique which offers the best trade-off between encryption strength and hardware efficiency.

- Д) To develop a new evaluation methodology to systematically assess the effectiveness of hardware encryption techniques in communication systems
- Е) To validate the solutions proposed.

This research adopts a systematic and multi-stage methodology to investigate the effective integration, security robustness, and performance trade-offs of logic encryption techniques in digital circuits. The proposed methodology combines theoretical analysis, hardware design, simulation-based evaluation, and security assessment to ensure both academic rigor and practical relevance.

The research follows a design-oriented and experimental approach. The core focus is on developing and evaluating logic-level encryption techniques that enhance circuit security while maintaining acceptable design overhead. Three distinct logic encryption methodologies are designed, implemented, and analyzed to provide a comprehensive comparison across structural, functional, and security dimensions. The overall methodology is structured into the following phases:

- А. Logic encryption design and integration
- Б. Hardware description and implementation
- В. Performance evaluation
- Г. Security strength assessment
- Д. Comparative analysis and validation

Table 11 Summary of research methods characteristics

	Research Method	Type	Purpose	Tools Used
1.	Mathematical Modeling	Quantitative	Formulate security and performance metrics	MATLAB, Python
2.	Hardware Design & Implementation	Quantitative	Implement logic encryption architectures	Verilog/VHDL
3.	Logic Synthesis & Simulation	Quantitative	Evaluate area, power, and timing overhead	Vivado / Design Compiler, ModelSim
4.	Security Analysis	Quantitative	Measure AHD, AFS, and SAT resistance	Python, MiniSAT
5.	Comparative Evaluation	Mixed	Compare encryption techniques	MATLAB, Python
6.	Literature Review	Mixed	Identify trends and research gaps	IEEE Xplore, Google Scholar

The research on logic encryption incorporates three distinct data collection methodologies:

- А. **Experimental approach**, The experimental approach is used to observe the impact of logic encryption on circuit behavior under controlled conditions.
- Б. **Simulation-based approach** The simulation-based approach enables detailed analysis of functional correctness, timing, power, and area overhead using hardware description and simulation tools.
- В. **Valuation and validation approach**. Finally, the evaluation and validation approach is employed to assess the security robustness and effectiveness of the proposed encryption schemes through quantitative metrics and comparative analysis.

This study emphasizes data-driven decision-making in hardware security research through the application of mathematical and statistical analysis using MATLAB and Python. These platforms were utilized to process and analyze experimental and simulation data obtained from logic-encrypted digital circuits.

Scientific novelty

This thesis provides a unified survey and comparative analysis of XOR/XNOR-, MUX-, and Flip-Flop-based logical encryption techniques for hardware security. Unlike previous studies that focus on individual methods, it establishes a common framework for evaluating their security effectiveness, implementation complexity, and performance overhead. The work identifies key research gaps and proposes future directions for developing more secure and efficient hardware encryption solutions for modern integrated circuits and embedded systems.

Practical applicability

The results of this thesis can be applied in the design of secure integrated circuits, embedded systems, IoT devices, and communication hardware. By providing a comparative analysis of XOR/XNOR-, MUX-, and Flip-Flop-based logical encryption techniques, the study helps engineers and researchers select suitable security mechanisms that balance protection, performance, area overhead, and power consumption. The findings contribute to the development of more secure and resilient hardware systems against reverse engineering, tampering, and unauthorized access.

Publications

The main achievements and results of the dissertation are published in 10 research works presented in the following table.

Journals listed in Scopus, Web of Science and IEEEEXPLORE	2 Papers Nr. 1, Nr. 2	TELECOM (2020) TELECOM (2021)	<i>Indexed by IEEE Xplore, Scopus,</i>		
	5 Papers Nr. 6, Nr. 7 Nr.8 Nr.9 Nr.10	TechDefense (2023) TechDefense (2024) TechDefense (2025)	Indexed by IEEE Xplore, Scopus		
	1 Paper Nr. 4	IWSSIP (2022) IWSSIP (2021 Underline Science Inc.)	Published in Science Direct, Indexed by Scopus.		
International Conference and Journal	1 Paper Nr. 3	International Conference on Science, UBT (2021)			
Chapters in Books	1 Chapter Nr. 5	<i>Monitoring and Protection of Critical Infrastructure by Unmanned Systems</i> , iOS Press	Indexed by Scopus		
Citations	7 Citations				

Structure and volume of the dissertation

The dissertation is in a volume of 165 pages, including an introduction, of 5 chapters for solving the formulated main tasks, a list of the main contributions, a list of the dissertation publications and used literature. Total 200 of literature sources are cited. The work includes total 54 figures and 33 tables. The numbers of figures and tables in the abstract correspond to those in the dissertation.

II. CONTENT OF THE DISSERTATION

CHAPTER I. Analytical Review of Hardware Encryption Methods, Techniques, multimedia tools for the design Communication Circuits

The chapter presents a detailed literature review of existing hardware encryption and security techniques. It examines prior work in logic encryption, obfuscation, and related countermeasures, while identifying their strengths, limitations, and gaps that this research aims to address.

Overview of Cybersecurity in Communication Systems

Hardware Obfuscation has been investigated for both FPGA and ASIC design flows, and for different levels of abstraction, for example, register transfer and gate-level level IPs. The security model in hardware obfuscation is similar to encryption but here relies on the secret of the key and not on the type of algorithm. The objectives of the transformation and the ways in which they are achieved. Logic encryption is performed by introducing specialized logic structures or key-controlled ports which produce erroneous results for the wrong key and vice versa. The second objective is to conceal the design through structural transformation. This objective is essential for two reasons:

- 1) protects the lock itself, as a blocker, which can be easily removed and can also be easily identified;
- 2) prevents the leakage of design secrets, such as the logical style and the type of function performed.

The design of secure communication systems involves a comprehensive understanding of both software and hardware security mechanisms. While software security has traditionally been the focus of most cybersecurity solutions, the increasing sophistication of attackers and the evolution of embedded technologies have necessitated a shift towards more hardware-oriented security approaches. This section explores the fundamental differences, complementary roles, and trade-offs between these two paradigms.

It's critical to comprehend the possible enemies who might launch an assault using the security flaws. This can assist designers in understanding the capabilities of attackers and selecting appropriate countermeasures based on the operation and targeted adversary. The enemy may take the form of an individual or an organized group that wants to get, destroy, or interfere with an asset that they are not authorized to access. Adversaries can be divided into insiders and outsiders when considering an integrated circuit design process and the companies involved. In this section are depicted at various points during the SoC design process.

The IC design process involves multiple stages, tools, and stakeholders, making it inherently vulnerable to a wide range of security threats. As modern ICs are increasingly designed using globalized supply chains and third-party IP, vulnerabilities can be introduced intentionally or unintentionally at different phases of the design and manufacturing lifecycle. Understanding these vulnerabilities is essential for developing effective hardware security countermeasures such as logic encryption. A SoC vulnerability is a weakness that could be used by an adversary to launch an attack and take control of assets. Each vulnerability must be allocated to one or more appropriate abstraction levels so that it may be effectively detected during the development of the DSeRC framework. An IC design typically involves specification, RTL, gate-level, and physical layout design. The goal of the DSeRC framework is to find vulnerabilities as early in the design flow as possible, since a lengthy development cycle and expensive design can result from a delayed review. Furthermore, if vulnerabilities in one stage are not fixed, they may create new ones when moving from one level to the next. Based on the abstraction layers, vulnerabilities are categorized in this section. The current body of literature in hardware security presents a diverse range of techniques and perspectives, yet several strengths, limitations, and controversies can be identified across this evolving field.

CHAPTER II. Research Methods for Hardware Encryption, Techniques and Multimedia Tools for the design Communication Circuits

In this chapter is summarized the description of focuses on the theoretical background and research methodology. It introduces the fundamental principles of cryptography and hardware security, explains the models of potential attacks, and describes the methodologies, tools, and design approaches used throughout the research.

2.1 Research Methodology - This research adopts a systematic and multi-stage methodology to investigate the effective integration, security robustness, and performance trade-offs of logic encryption techniques in digital circuits. The proposed methodology combines theoretical analysis, hardware design, simulation-based evaluation, and security assessment to ensure both academic rigor and practical relevance.

The research follows a design-oriented and experimental approach. The core focus is on developing and evaluating logic-level encryption techniques that enhance circuit security while maintaining acceptable design overhead. Three distinct logic encryption methodologies are designed, implemented, and analysed to provide a comprehensive comparison across structural, functional, and security dimensions.

The overall methodology is structured into the following phases:

- A. Logic encryption design and integration
- B. Hardware description and implementation
- B. Performance evaluation
- Γ. Security strength assessment
- Δ. Comparative analysis and validation

Table 11 Summary of research methods characteristics

	Research Method	Type	Purpose	Tools Used
1.	Mathematical Modeling	Quantitative	Formulate security and performance metrics	MATLAB, Python
2.	Hardware Design & Implementation	Quantitative	Implement logic encryption architectures	Verilog/VHDL
3.	Logic Synthesis & Simulation	Quantitative	Evaluate area, power, and timing overhead	Vivado / Design Compiler, ModelSim
4.	Security Analysis	Quantitative	Measure AHD, AFS, and SAT resistance	Python, MiniSAT
5.	Comparative Evaluation	Mixed	Compare encryption techniques	MATLAB, Python
6.	Literature Review	Mixed	Identify trends and research gaps	IEEE Xplore, Google Scholar

The research on logic encryption incorporates three distinct data collection methodologies:

- Γ. **Experimental approach**, the experimental approach is used to observe the impact of logic encryption on circuit behavior under controlled conditions.
- Δ. **Simulation-based approach** the simulation-based approach enables detailed analysis of functional correctness, timing, power, and area overhead using hardware description and simulation tools.

- E. **Valuation and validation approach.** Finally, the evaluation and validation approach is employed to assess the security robustness and effectiveness of the proposed encryption schemes through quantitative metrics and comparative analysis.

2.2 Factors Affecting Logic-Encrypted in FPGA Designs

This section defines Logic encryption significantly alters the design characteristics of FPGA-based systems by introducing additional key-dependent logic, routing complexity, and switching activity. In this section explain the factors affecting conventional FPGA designs have an amplified impact when encryption mechanisms are integrated. Understanding these factors is essential for balancing security strength, power efficiency, performance overhead, and resistance to hardware attacks.

2.3 Methods for Logic Encryption Architecture Design

Three logic encryption architectures are developed to represent both combinational and sequential locking strategies: Let an unencrypted digital circuit be modeled as a tuple:

$$C=(X,Y,F) \quad (2.1)$$

where

- $X=\{x_1,x_2,\dots,x_n\}$ denotes the set of primary inputs,
- $Y=\{y_1,y_2,\dots,y_m\}$ denotes the set of primary outputs,
- $F: \{0,1\}^n \rightarrow \{0,1\}^m$ represents the Boolean functionality.

An encrypted (locked) circuit is defined as:

$$C_L = (X, K, Y, F_L) \quad (2.2)$$

where

- $K=\{k_1,k_2,\dots,k_p\}$ is the secret key vector,
- $F_L: \{0,1\}^{n+p} \rightarrow \{0,1\}^m$ is the encrypted Boolean function.

Correct functionality is satisfied if and only if:

$$F_L(X, K_c) = F(X) \quad (2.3)$$

where K_c is the correct encryption key.

1) XOR/XNOR-Based Encryption

XOR/XNOR gates are inserted at strategically selected nodes in the circuit. The correctness of the circuit output depends on the applied secret key. Incorrect keys result in corrupted outputs, providing protection against unauthorized use and reverse engineering. Encryption is implemented by inserting XOR/XNOR gates at selected internal nodes v_i . The encrypted signal v'_i is defined as:

$$v'_i = v_i \oplus k_i \text{ or } v'_i = \overline{v_i \oplus k_i} \quad (2.4)$$

For an incorrect key $K \neq K_c$, the probability of output corruption is statistically amplified across the logic cone, increasing functional entropy.

2)MUX-Based Encryption

Multiplexer-based logic locking is implemented by introducing key-controlled MUX structures that dynamically select between correct and incorrect signal paths. This technique increases functional ambiguity and complicates structural analysis by attackers. Multiplexer-based encryption introduces key-controlled selection logic:

$$v'_i = k_i \cdot f_{\text{true}}(v_i) + \overline{k_i} \cdot f_{\text{false}}(v_i) \quad (2.5)$$

Where f_{false} is a deliberately incorrect logic path. This structure increases ambiguity in reverse engineering by creating multiple functionally plausible configurations.

3)Flip-Flop-Based Encryption

A novel Flip-Flop-based encryption scheme is proposed for sequential circuits. Encryption is achieved by embedding key-controlled behavior into state-holding elements, thereby protecting both the data path and the state transitions. This approach strengthens resistance against attacks that exploit scan chains and sequential behavior. For sequential circuits, state evolution is modeled as:

$$S_{t+1} = G(S_t, X_t) \quad (2.6)$$

The encrypted state transition becomes:

$$S_{t+1} = G_L(S_t, X_t, K) \quad (2.7)$$

Key-dependent Flip-Flop behavior modifies either:

- State update functions, or
- Clock/data gating logic

This introduces temporal dependency on the key, significantly increasing resistance to scan-based and sequential attacks.

2.4 Hardware Implementation

All encryption architectures are implemented using Verilog/VHDL to ensure synthesizability and compatibility with real-world digital design flows.

- Encryption modules are integrated into communication-oriented hardware blocks.
- Designs are synthesized using standard-cell libraries.
- Uniform design constraints are applied to ensure fair comparison across techniques.

This implementation phase ensures that the proposed solutions are implementation-ready and suitable for integration into practical communication subsystems. The overall hardware implementation flow adopted in this research is illustrated in Figure 19. The flow follows a structured and tool-compatible design methodology beginning with HDL modeling and ending with post-synthesis evaluation.

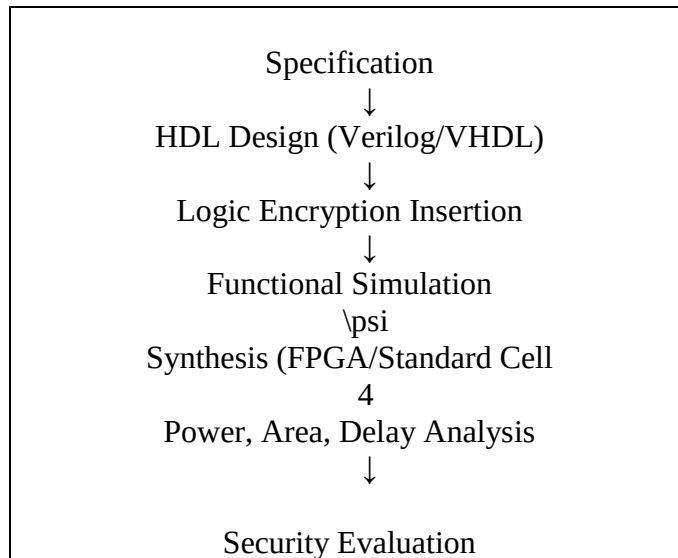


Figure 19 Hardware Implementation Flow for Logic-Encrypted FPGA Designs

This flow ensures consistency, repeatability, and fair comparison among all encryption techniques.

- **Integration of Encryption Modules**

Encryption modules based on XOR/XNOR logic, MUX-based locking, and Flip-Flop-based encryption are integrated into communication-oriented hardware blocks such as encoders, controllers, and data-processing units. The encryption logic is inserted at strategically selected points within the datapath and control path to maximize security impact while minimizing overhead.

Inputs → Functional Logic → Encryption Logic → Outputs



Secret key

Key signals are stored in secure registers and distributed internally to prevent external observability. The encrypted design produces correct outputs only when the correct key is applied.

- **HDL Modeling and Simulation**

All designs are described using synthesizable RTL-level Verilog/VHDL. Behavioral modeling is preferred where possible to allow synthesis tools to perform logic optimization and resource sharing. Structural descriptions are used selectively in encryption modules to maintain precise control over key-dependent logic behavior.

Functional simulation is performed to verify:

- Correct operation under the correct key
- Output corruption under incorrect keys
- Absence of functional violations introduced by encryption

- **Synthesis and Design Constraints**

Encrypted and unencrypted designs are synthesized using uniform constraints to ensure objective comparison. Identical clock frequencies, voltage levels, and optimization settings are applied across all implementations. Standard-cell libraries (for ASIC-style evaluation) or FPGA vendor libraries are used depending on the target platform.

Key synthesis objectives include:

- Minimizing encryption-induced area overhead
- Preserving timing closure
- Limiting power increase due to added logic

This hardware implementation methodology ensures that:

- All encryption architectures are fully synthesizable
- Designs can be directly deployed on FPGA or ASIC platforms
- Performance and security metrics are measured under identical conditions

The structured flow enables seamless integration of logic encryption into practical communication subsystems, validating the feasibility of the proposed techniques for real-world hardware security applications.

Table 12. Tool-Specific Implementation Flow and Experimental Results Mapping

Tool / Platform	Design Stage	Outputs Generated	Linked Experimental Results Tables
Xilinx Vivado	RTL synthesis, place & route, power analysis	LUT/FF/DSP/BRAM utilization, timing slack, dynamic & static power	(Area Overhead), (Delay), (Power), (Toggle-Rate Impact)
Intel Quartus Prime	Synthesis, fitting, power estimation	Logic utilization, Fmax, early & post-fit power	(Cross-Vendor Area), (Timing Comparison), (Cross-Vendor Power)
Synopsys Design Compiler	ASIC-style synthesis	Gate count, cell area, netlist generation	(Gate-Level Area), (Timing Analysis)
Synopsys PrimeTime / PX	Timing and power analysis	Leakage power, dynamic power, critical paths	(ASIC Power Breakdown), (Gate-Level Timing)
ModelSim / XSim	Functional & RTL simulation	Correct/incorrect key behavior, waveform validation	(Functional Correctness Validation)
MATLAB / Python	Statistical & security analysis	AHD, AFS, SAT resistance, regression results	AHD, (AFS), (SAT Runtime), (Key Strength Analysis)

2.5 Hardware Metrics and Security Evaluation

The effectiveness of logic encryption techniques is evaluated using a combination of hardware performance metrics and security metrics. Performance metrics quantify the implementation overhead introduced by encryption, while security metrics assess the robustness of the encrypted design against various attack models. Together, these metrics provide a comprehensive evaluation framework for comparing different logic encryption architectures.

A. Performance Metrics:

Performance metrics measure the impact of logic encryption on the physical and temporal characteristics of the hardware design.

- Power Consumption
- Area Overhead (number of logic gates, LUTs)
- Latency/Delay (critical path, propagation delay)

2.6 Key Management Strategy

Effective key management is essential to ensure that logic encryption techniques provide strong security guarantees while remaining scalable and practical for real-world deployment. In this research, key management strategies are analyzed with respect to secure storage, controlled activation, scalable distribution, and key sensitivity, ensuring that the proposed encryption architectures can be integrated into large-scale hardware systems beyond laboratory settings.

2.7 Comparative Analysis and Validation

A structured comparative analysis is conducted across all proposed logic encryption techniques using a unified set of performance and security metrics. This analysis ensures an objective and reproducible evaluation of the encryption strategies under identical design constraints, enabling fair comparison and meaningful conclusions.

The evaluation framework developed in this research supports:

- Objective comparison of XOR/XNOR-based, MUX-based, and Flip-Flop-based encryption techniques
- Identification of optimal trade-offs between hardware overhead (power, area, delay) and security strength
- Validation of the proposed Flip-Flop-based encryption method, particularly for sequential circuits

CHAPTER III: New Methodology, Algorithm and Implementation of Hardware Encryption Techniques

This chapter presents the implementation and systematic evaluation of hardware logic encryption techniques for enhancing the cybersecurity of communication systems. The study investigates three representative encryption mechanisms: XOR/XNOR-based logic encryption, MUX-based logic encryption, and Flip-Flop-based encryption. These techniques embed security primitives directly into the hardware fabric, thereby strengthening resistance against reverse engineering, fault injection attacks, and side-channel exploitation.

An algorithmic evaluation framework is proposed to quantitatively assess encryption robustness and implementation overhead. The framework integrates logic obfuscation, key-dependent transformations, and state-based encryption to evaluate security strength through fault injection experiments, Hamming distance analysis, and error propagation metrics. In parallel, the impact of encryption on latency, power consumption, and area overhead is analyzed to characterize performance trade-offs. The objective of this chapter is twofold: (i) to identify the encryption technique that provides the strongest obfuscation and key dependency, and (ii) to establish a reproducible evaluation methodology that supports informed design decisions for secure hardware in communication systems.

3.1 Implementation Framework for Hardware Encryption Logic

A dual-layer implementation framework is adopted to enable both high-level evaluation and hardware-accurate validation. Python-based simulation is used for algorithmic analysis and metric computation, while VHDL-based RTL implementation is employed for hardware verification.

- **Python Simulation Layer:** Evaluates encryption strength, Hamming distance, and fault sensitivity under controlled conditions.
- **VHDL Hardware Layer:** Implements and validates encryption logic at the RTL level, ensuring synthesizability and correctness.

This separation enables rapid prototyping while maintaining fidelity to real hardware behavior.

- **Python-Based Simulation of Encryption Robustness**

The Python simulation framework evaluates encryption strength through the following procedure:

- Generation of random logic circuits and output vectors.
- Application of XOR/XNOR-based, MUX-based, and Flip-Flop-based encryption.
- Computation of HD between original and encrypted outputs.
- Execution of bit-flip fault injection, where each input bit is inverted to measure output sensitivity.

This approach enables efficient, statistically meaningful comparison of encryption schemes under identical conditions.

- **HDL-Based Hardware Verification**

RTL implementations of all encryption techniques are developed using synthesizable HDL. A unified testbench verifies:

- Correct functional behavior under the correct key.
- Output corruption under incorrect keys.
- Error propagation under fault injection.
- Consistency with simulation-level results.

This stage confirms the practical viability of the proposed encryption logic for FPGA and ASIC deployment.

- **Comparative Evaluation of Encryption Techniques**

The three encryption schemes are evaluated as follows:

- **XOR/XNOR-Based Encryption:** Introduces key-dependent inversion using XOR/XNOR gates.
- **MUX-Based Encryption:** Selects between correct and incorrect logic paths using key-controlled multiplexers.
- **Flip-Flop-Based Encryption:** Incorporates sequential state dependency using D-type flip-flops, introducing temporal obfuscation.

Evaluation Workflow

- Generate a random logic circuit and test vectors.
- Apply each encryption method.

- Compute security metrics (AHD, AFS).
- Compare robustness and performance impact.

We have used the script has generated a random logic circuit and performs encryption with XOR/XNOR, MUX, and Flip-Flop methods. It have then visualize these circuits and compare their performance based on Hamming distance and fault sensitivity.

AHD (to measure how far apart the encrypted outputs are from the original outputs). It is the average Hamming distance between the original circuit and the encrypted circuit. This is calculated for each input combination by comparing the outputs.

$$AHD = \frac{1}{M} \sum_{j=1}^m HD(O_{correct}^{(j)}, O_{incorrect}^{(j)}) \quad (3.1)$$

m- number of test vectors

O^j - output for the j^{th} test vector

- Lower values indicate that the encryption method retains more similarity with the original data, potentially making it less secure.
- Higher values indicate better obfuscation, which is generally more secure.

Average Fault Sensitivity (AFS) (to simulate the impact of bit-flips on each encryption method). It is calculated by simulating bit-flips in the input values and measuring the change in output. The higher the fault sensitivity, the more the encryption is impacted by bit-flips.

$$AFS = \frac{1}{N} \sum_{i=1}^N HD(O_{normal}^{(i)}, O_{faulted}^{(i)}) \quad (3.2)$$

N: Number of fault injection experimentes

$O_{normal}^{(i)}, O_{faulted}^{(i)}$: output for the i^{th} test vector

- Higher fault sensitivity means the encryption method is more vulnerable to bit-flips, which could be a concern for real-world applications where errors may occur during transmission or storage.
- Lower values indicate that the encryption is more robust and less affected by errors.

Table 15 The results for Encryption Method

Encryption Method	AHD	AFS
XOR/XNOR	1.876	230
MUX-based	2.053	210
Flip-Flop-based	2.331	190

After simulating several types of logical circuits, the results are as in Table 15. The defined parameters are explained as follows: The best encryption method as shown in figure 19 is FLIP-FLOP-based encryption, because:

- It has the highest Hamming Distance (better obfuscation).
- It has the lowest fault Sensitivity (more resilient to faults)

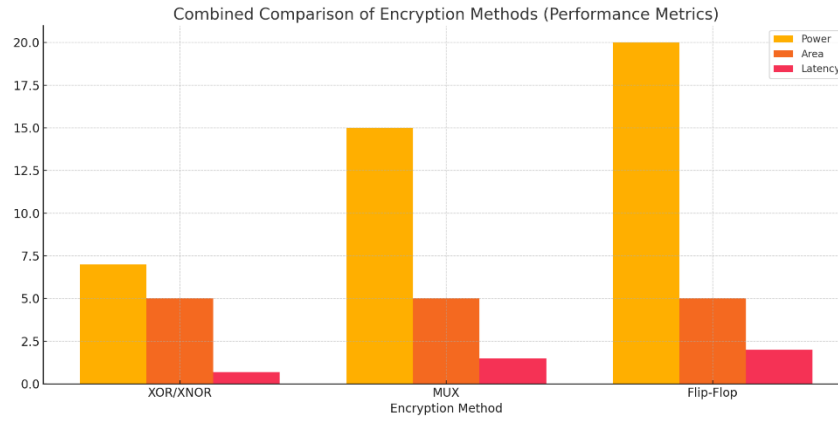


Figure 19 Comparison of Encryption Methods by Performance Metrics

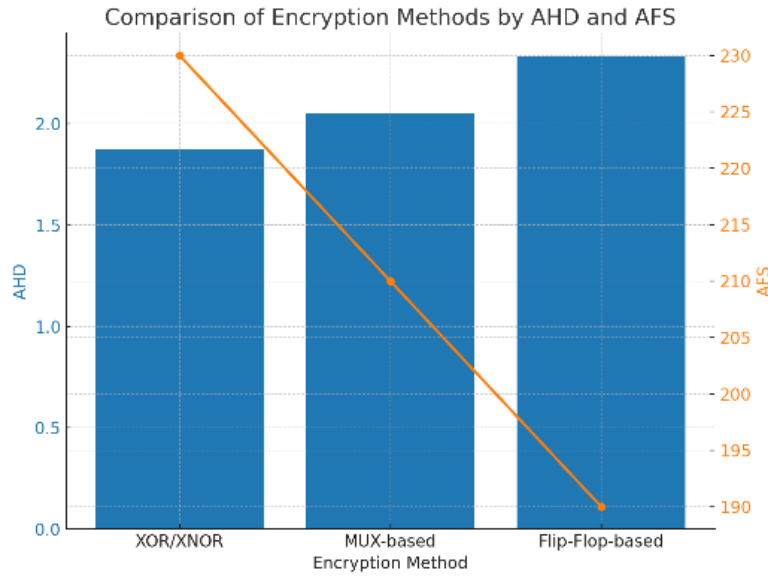


Figure 20 Comparison of Encryption Methods by Security Metrics

As shown at figure 19, 20. XOR/XNOR Encryption, Power: Lowest of all methods, Area: Minimal; efficient in gate count. Latency: Fastest ideal for low-resource, high speed systems. Takeaway: Best for lightweight applications where efficiency is key.

MUX-based Encryption, Power: Medium level, Area: Like XOR, but adds complexity due to selection logic, Latency: Moderate latency from additional logic. Takeaway: A good balance between obfuscation strength and resource usage.

Flip-Flop-based Encryption, Power: Highest flip-flops are state-holding elements and consume more. Area: Matches the others in gate count, but heavier components. Latency: Slowest due to sequential nature and clock dependency.

XOR/XNOR is optimal for efficiency. MUX adds complexity with manageable overhead. Flip-Flop is secure but resource-heavy, better for critical systems needing robust protection.

3.2 Proposed Algorithm for Evaluating Hardware Encryption Logic

Input:

n_inputs $\leftarrow$ Number of circuit inputs
 n_gates $\leftarrow$ Number of gates in the circuit
 key_bits $\leftarrow$ Encryption key bits

Output:

Encrypted Circuit, Hamming Distance, Fault Score, Performance Plots

Procedure:

1. Define Logic Gate Functions:
 - $AND(a, b) \leftarrow a \text{ AND } b$
 - $OR(a, b) \leftarrow a \text{ OR } b$
 - $XOR(a, b) \leftarrow a \text{ XOR } b$
 - $XNOR(a, b) \leftarrow NOT(a \text{ XOR } b)$
 - $NOT(a) \leftarrow \text{Invert } a$
 - $MUX(a, b, sel) \leftarrow a \text{ if } sel = 0 \text{ else } b$
 - $D_FLIP_FLOP(d, clk, q) \leftarrow d \text{ if } clk = 1 \text{ else } q$
2. Generate Random Logic Circuit:
 - $inputs \leftarrow [x_0, x_1, \dots, x_n]$
 - $wires \leftarrow inputs$
 - $circuit \leftarrow \emptyset$
 - For $i = 1$ to n_gates do:
 - $gate \leftarrow \text{Randomly select from } \{AND, OR, XOR, XNOR\}$
 - $(a, b) \leftarrow \text{Random wires from wires}$
 - $out \leftarrow "w" + i$
 - Append $(gate, a, b, out)$ to circuit
 - Add out to wires
3. Apply Encryption:
 - a. XOR/XNOR-based Encryption:
 - For each gate in circuit:
 - If gate is XOR or XNOR:
 - Insert key-dependent XOR/XNOR logic
 - b. MUX-based Encryption:
 - For each gate in circuit:
 - Insert $MUX(a, b, key_bit)$ logic
 - c. Flip-Flop-based Encryption:
 - For each gate in circuit:
 - Replace output with $D_FLIP_FLOP(a, clk=1)$
4. Circuit Evaluation:
 - For each input vector in $\{0,1\}^n$:
 - Evaluate circuit with correct key
 - Store output value
5. Hamming Distance Analysis:
 - For each input vector:
 - Evaluate output with correct key $\rightarrow O_correct$
 - Evaluate output with incorrect key $\rightarrow O_wrong$
 - Compute $Hamming_Distance(O_correct, O_wrong)$
6. Fault Injection Analysis:
 - $fault_score \leftarrow 0$
 - For each input vector:
 - $baseline \leftarrow evaluate(circuit, inputs)$
 - For each input bit:
 - $flip \leftarrow \text{invert one input bit}$
 - $faulty \leftarrow evaluate(circuit, flip)$
 - $fault_score += Hamming_Distance(baseline, faulty)$
7. Performance Metrics Estimation:
 - Initialize $gate_weights$:
 - $AND/OR = 1, XOR/XNOR = 2, MUX = 3, D_FLIP_FLOP = 4$
 - For each encrypted gate in circuit:
 - Accumulate:
 - $Power \leftarrow \sum gate_weight$

$Area \leftarrow \sum \text{fixed gate area (e.g., LUTs or NAND equivalent)}$

$Latency \leftarrow \sum \text{gate-level delay per type}$

8. Plot Results:

Plot:

- Power vs. Encryption Method
- Area vs. Encryption Method
- Latency vs. Encryption Method

Return:

Encrypted Circuit,
Hamming Distance Score,
Fault Sensitivity Score,
Plots of Power, Area, and Latency

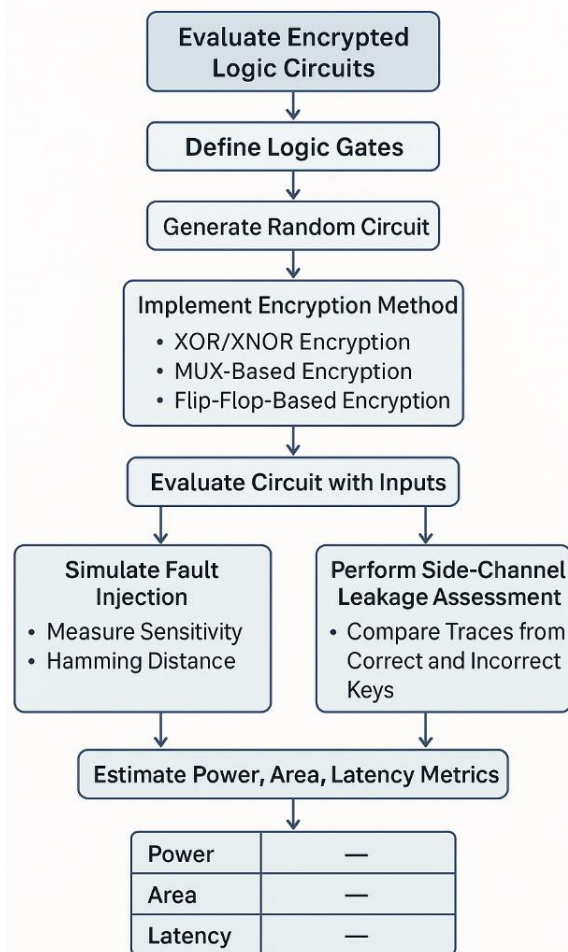


Figure 21 Algorithm Block Schema

This algorithm at figure 21 provides a systematic evaluation of hardware encryption logic, ensuring that encryption techniques like XOR/XNOR, MUX-based, and Flip-Flop-based enhance circuit security while maintaining performance efficiency. The methodology quantifies encryption strength, making it valuable for IC designers, FPGA security experts, and embedded system engineers.

3.3 Simulating the Encryption Method on the Circuits

The proposed encryption techniques are evaluated using standard ISCAS benchmark circuits, which are widely adopted in the literature for validating logic design correctness, fault tolerance, and security mechanisms. Specifically, ISCAS-85 combinational circuits and ISCAS-89 sequential circuits are employed to assess the effectiveness of logic encryption across both combinational and state-dependent architectures.

After obtaining the benchmark netlists, the circuits are parsed to extract logic gates, interconnections, and flip-flop elements. For ISCAS-85 circuits, simulation involves direct evaluation of combinational logic by applying Boolean gate functions to the input vectors. For ISCAS-89 circuits, sequential behavior is modeled explicitly by incorporating clock-driven state updates, where flip-flop contents are updated on each clock cycle according to the circuit's state transition logic.

This benchmark-driven simulation approach provides a realistic and scalable evaluation of logic encryption techniques, demonstrating their applicability to practical hardware designs used in communication-oriented secure systems. Table 16 summarizes the ISCAS-85 and ISCAS-89 benchmark circuits used in this study, covering a range of circuit sizes and structural complexities to ensure representative evaluation of the proposed encryption techniques.

Table 16 Summary of ISCAS Benchmark Circuits Used for Encryption Evaluation

Benchmark Circuit	Circuit Type	Number of Inputs	Number of Outputs	Logic Elements (Approx.)
c432	Combinational (ISCAS-85)	36	7	~160 gates
c499	Combinational (ISCAS-85)	41	32	~202 gates
c880	Combinational (ISCAS-85)	60	26	~383 gates
c1355	Combinational (ISCAS-85)	41	32	~546 gates
s298	Sequential (ISCAS-89)	3	6	14 FFs, ~100 gates
s344	Sequential (ISCAS-89)	9	11	15 FFs, ~160 gates
s349	Sequential (ISCAS-89)	9	11	15 FFs, ~170 gates
s5378	Sequential (ISCAS-89)	35	49	179 FFs, ~2,700 gates

The benchmark circuits listed belong to the widely used ISCAS-85 and ISCAS-89 suites, which are standard datasets for evaluating digital circuit design, testing, and optimization techniques. The ISCAS-85 benchmarks consist of combinational circuits, while the ISCAS-89 benchmarks include sequential circuits with memory elements such as flip-flops.

3.4 Flowchart for Comprehensive Security and Performance Metric Evaluation of Logic-Encrypted Benchmark Circuits

The workflow then branches into an encryption integration stage, where three encrypted circuit variants are generated from the baseline design:

- A. XOR/XNOR-based logic encryption
- B. MUX-based logic encryption
- B. Flip-Flop-based logic encryption

Each encrypted circuit incorporates key-dependent logic modifications while preserving functional correctness under the valid key.

Following encryption insertion, the flow proceeds to the security evaluation block, which contains three parallel analysis paths:

- **Hamming Distance Analysis:** Outputs produced under the correct key are compared with outputs generated using incorrect keys. The AHD is computed across all test vectors to quantify functional obfuscation.
- **Fault Injection Analysis:** Controlled faults (e.g., input bit flips or stuck-at faults) are injected into the encrypted circuits. Output deviations are measured to compute the AFS, reflecting resistance to fault-based attacks.
- **Key Sensitivity Analysis:** Individual key bits are flipped while holding inputs constant. Output variation is measured to assess sensitivity to small key changes.

In parallel, the flow transitions to the performance evaluation block, where each encrypted design is synthesized and implemented using FPGA or ASIC toolchains. From post-synthesis and post-place-and-route reports, power consumption (static and dynamic), area utilization, and critical path delay are extracted and normalized against the baseline design. The workflow then enters the SAT attack evaluation stage, where gate-level netlists of the encrypted circuits are subjected to SAT-based de-obfuscation attacks. Metrics such as SAT solver runtime, number of oracle queries, and timeout events are recorded to quantify algorithmic attack resistance. All computed metrics are collected in the results aggregation stage, where per-benchmark and per-encryption-method data are compiled into comparative tables and plots. These results enable cross-method comparison and support security–performance trade-off analysis.

3.5 Algorithm: Security Cost Driven Deployment Decision for Logic-Encrypted Communication Hardware

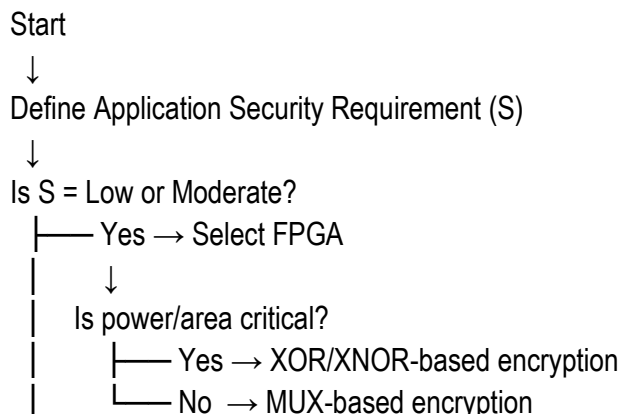
Input

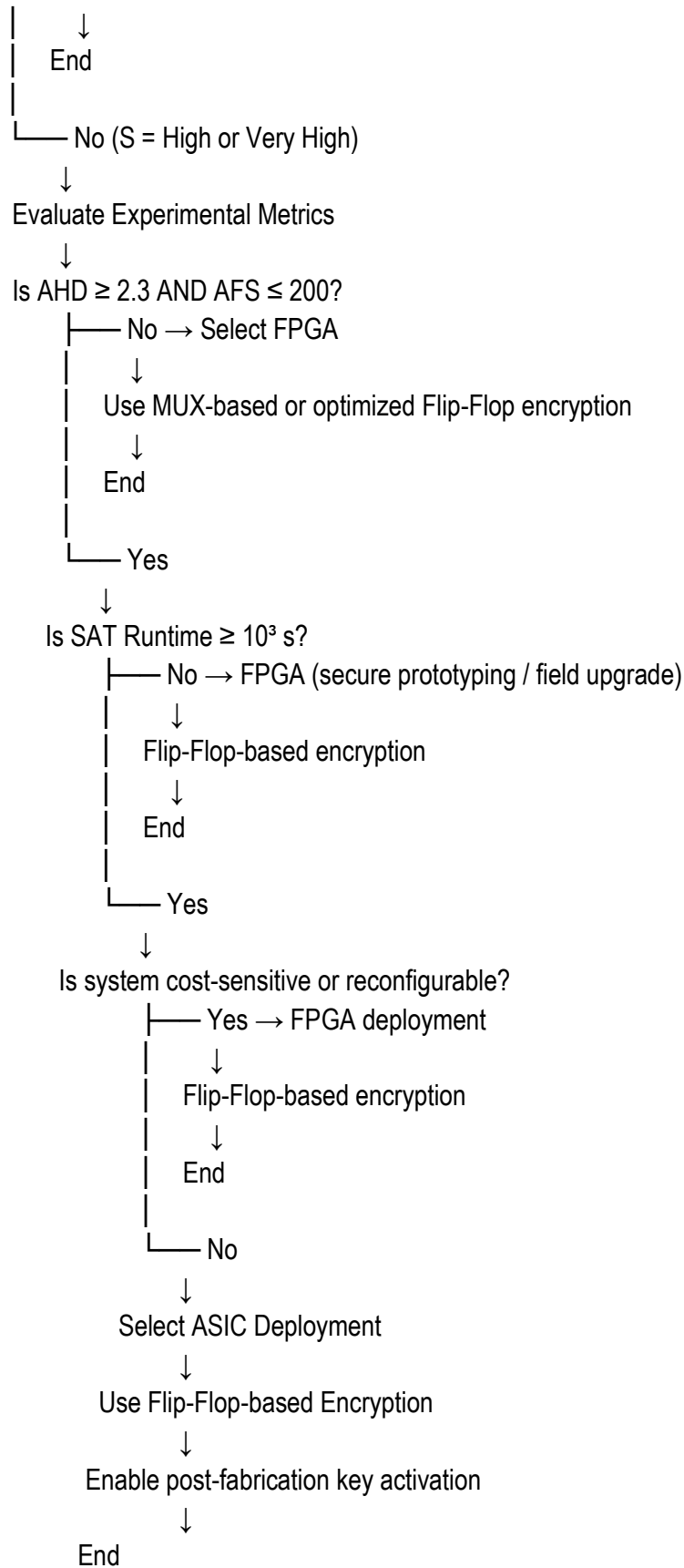
- Security requirements S
- Cost and flexibility constraints C
- Experimental security metrics:
 - Average Hamming Distance (AHD)
 - Average Fault Sensitivity (AFS)
 - SAT attack runtime $TSAT$

Output

- Recommended deployment platform (FPGA or ASIC)
- Recommended logic encryption technique

Decision Flow





Algorithm formalizes a security–cost–driven decision process for deploying logic-encrypted communication hardware by jointly considering application security requirements, implementation constraints, and experimentally measured security metrics. The algorithm takes as inputs the target security level, cost and reconfigurability constraints, and quantitative metrics derived from evaluation, namely AHD, AFS, and SAT

attack runtime. Thresholds for these metrics are established based on empirical results, where $AHD \geq 2.3$ indicates strong functional obfuscation, $AFS \leq 200$ reflects high fault resilience, and $SAT \text{ runtime} \geq 10^3$ seconds signifies substantial resistance to algorithmic de-obfuscation attacks. For low-to-moderate security applications, lightweight encryption methods are selected to minimize overhead. In contrast, when high or very-high security is required and all metric thresholds are satisfied, the algorithm prioritizes Flip-Flop-based encryption due to its state-dependent behavior, which significantly increases output corruption under incorrect keys, reduces fault observability, and dramatically increases SAT attack complexity. Under these conditions, ASIC deployment is recommended when cost and reconfigurability constraints permit, as permanent logic binding and post-fabrication key activation further strengthen security guarantees.

CHAPTER IV: Evaluation of Corruption Factor and Secrecy in Hardware Logic Encryption with Multi-Bit Keys

In this chapter describes the simulation environment and evaluation process, including the setup, tools, and benchmarks used to test the proposed designs. It presents the results, comparing performance, security strength, and overhead against existing techniques.

4.1 Experimental setup Hardware Encryption logic

For all encryption techniques, the location of the key gate is changed in order to verify the impact of the location and type of encryption gate on the security of the circuit. This chapter's goal is to create an experimental test bench for investigating circuits with different types of encryptions gates XOR/XNOR, MUX and Flip-Flop based. The research aims to measure the hamming distances for incorrect activation keys between the incorrect and correct output, the corruption factor and the secrecy. All studied circuits have an identical structural complexity change. First, the logic circuits are described VHDL language, and later 3 types of encryption logic techniques are implemented: XOR/XNOR-based encryption, MUX-based encryption, Encrypt flip-flop technique. The following steps:

- A. Develop logic encryption gates using VHDL.
- B. Develop Test benches for logic encryption gates.
- B. Successfully simulate designed logic encryption gates.
- Γ. Develop a circuit by assembling the logic encryption gates.

4.2 Logic Encryption Based on Fault Analysis

Now, we'll go over how we using key-gates (such XOR/XNOR) to encrypt a design so that any incorrect key results in an incorrect output. This is comparable to the circumstance in which a circuit with a fault that has been excited and spread to the outputs generates an incorrect output. The fault analysis in IC testing and logic encryption are related in the following observations. These observations will be used to add XOR/XNOR gates.

4.3 XOR/XNOR-based encryption

The XOR/XNOR-based encryption technique randomly inserts XOR/XNOR gates (key-gates) into the design. In this technique the location of the encryption gate is changed to observe how the Hamming Distance (HD) [6], corruption factor and deceptiveness change. Two types of circuits are encrypted: AND/OR (one toggle truth table) and AND/OR/XOR (2 toggle truth table) based, in order to determine which of them is easier for encrypting. The values of the correct key bit for the XOR/NXOR encrypting gates is directly deduced from the troth table of the gates and it's the value which lets the second entry at the output. The correct key-bit for the XOR gate is "0" and for the NXOR gate is "1". The reason for using key gates in this technique is that in

case of using the rights keys at the entrance of the circuit, it will ensure the correct functionality of the circuit. In case of interventions by unauthorized persons with the wrong key, will result a wrong output.

Circuits with no encryption

The first design in Figure 4 has no encryption implemented and it is composed of combinatorial components. The initial circuit consists of 8 I/O (7 inputs and 1 output). It consists of 4 AND, 2 OR logic gates [1]. After the VHDL design was evaluated successfully, it was implemented on the Zed Board device. After the implementation of the VHDL code in the FPGA device, the RTL schematics is synthesized in VIVADO tool. The second original circuits are in the setup in Figure 5 and it is composed of AND/OR/XOR gates, where XOR has a 2 toggle truth table.

$$SHScc = \frac{1}{6} = 0.166 \text{ (4)}$$

• XOR/XNOR encryption with 1 key bit

The first encryption technique is the XOR/XNOR-based, which randomly inserts XOR/XNOR gates (key-gates) into the design. In this logic encryption technique, we will change the position of different key gates, we didn't change the number of gates. The reason for using the key gates is to ensure that the right keys are used as inputs, and guarantee that the functionality of the circuit is correct. In cases of any not authorized user, the wrong key at the input will provide the wrong function at the output. To test this technique in the original circuits, XOR gates with key bit K and different locations are inserted, simulated and evaluated, see Figures 29, 30 and 31.

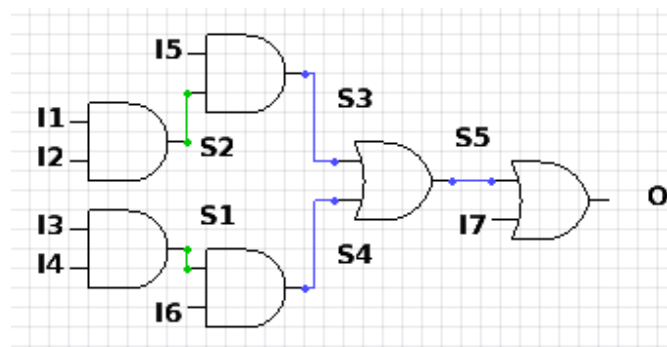


Figure 29 Initial circuit with no encryption

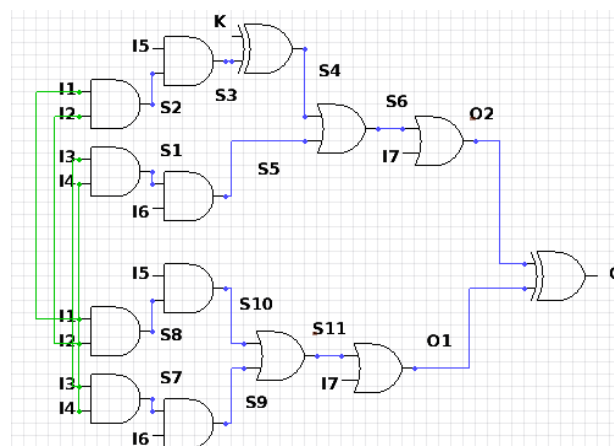


Figure 30 The first position of 1 XOR Gate, XOR/XNOR-based encryption

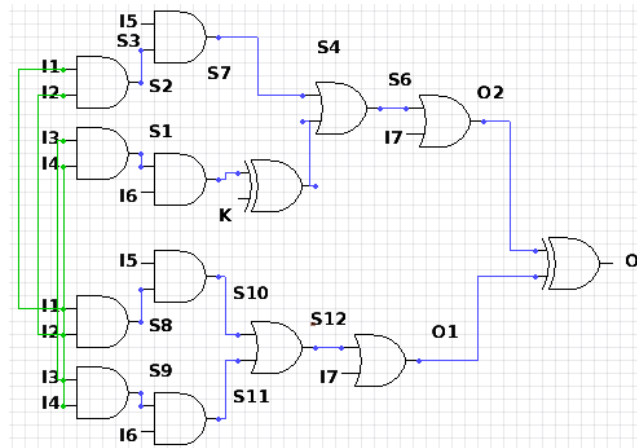


Figure 31 The second position of 1 XOR Gate, XOR/XNOR-based encryption

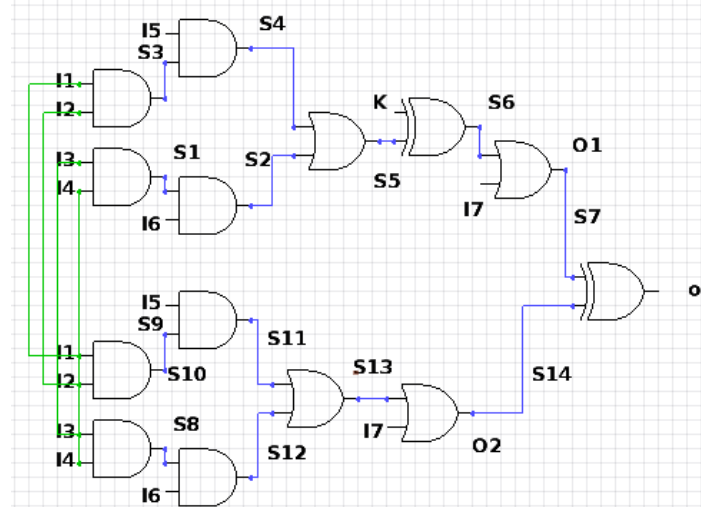


Figure 32 The third position of 1 XOR Gate, XOR/XNOR-based encryption

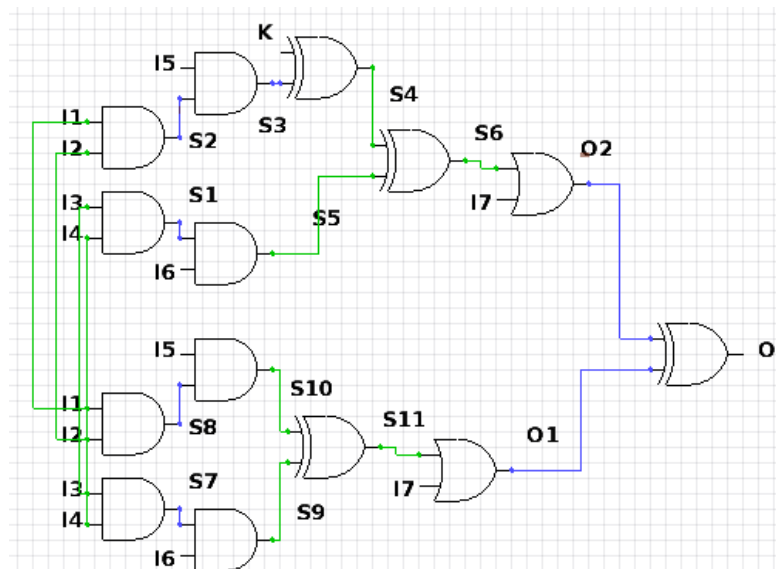


Figure 33 In the first encrypted circuit, we change a gate from OR to XOR to see the change and the effect

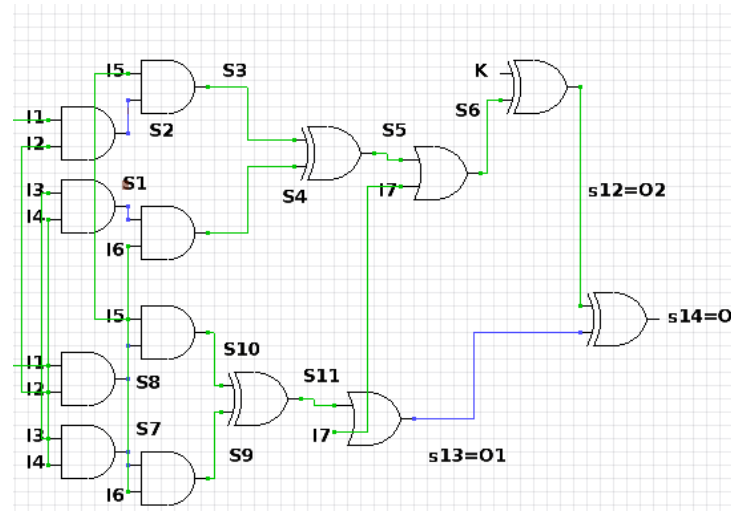


Figure 34 The second position of 1 XOR Gate we change the OR gate with XOR, XOR/XNOR-based encryption

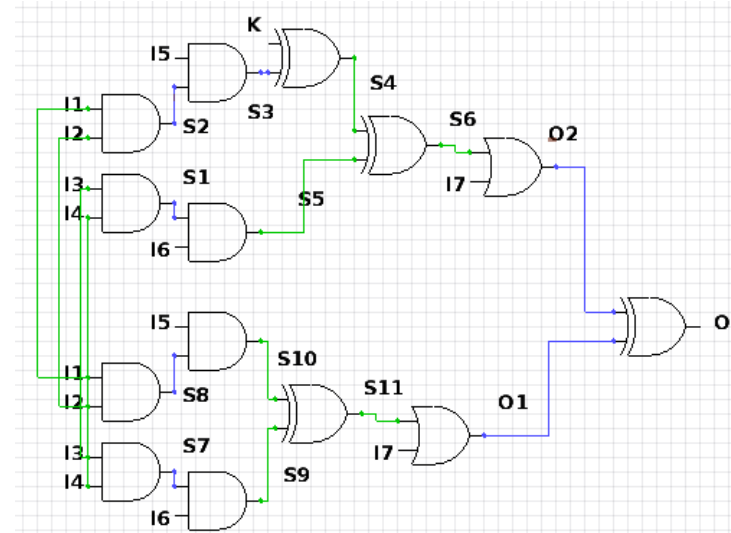


Figure 35 Circuit with encryption (XOR gate), In the first encrypted circuit, we change a gate from OR to XOR.

The first circuit non encrypted and is made up of a number of sequential and concatenated components. 7 inputs and 1 output make up the 8 I/O in the basic circuit. As seen in Figure 35, it consists of four AND gates and two OR gate. RTL schematics were created once the VHDL code was implemented in the FPGA device. [8]

Structural complexity changes for 2 bit –key encryption scenarios:

$$SHScc = \frac{2}{6} = 0.3 \quad (6)$$

Structural complexity changes for 3 bit –key encryption scenarios:

$$SHScc = \frac{3}{6} = 0.5 \quad (7)$$

Following the behavioral simulation and logic design of the circuit in VHDL, the examples to gauge the security of various hardware logic encryption methods are simulated and evaluated in terms of corruption and deceptiveness factors and secrecy. The first encryption scenario consists in adding two XOR gates with K1, K2 keys as shown in Figure 36.

The block diagram in Figure 36 illustrates how the encryption method is developed through arbitrary inserting XOR/XNOR gates with two key-gates. Using behavioral modelling, the XOR/XNOR-based encryption circuit's functionality is examined. In the next scenario 3 XOR/XNOR gates (key gates) are randomly introduced into the circuit to test the XOR/XNOR encryption method. The circuit in Figure 37 is created utilizing three separate key gates, K1, K2, and K3, in this logic encryption technique.

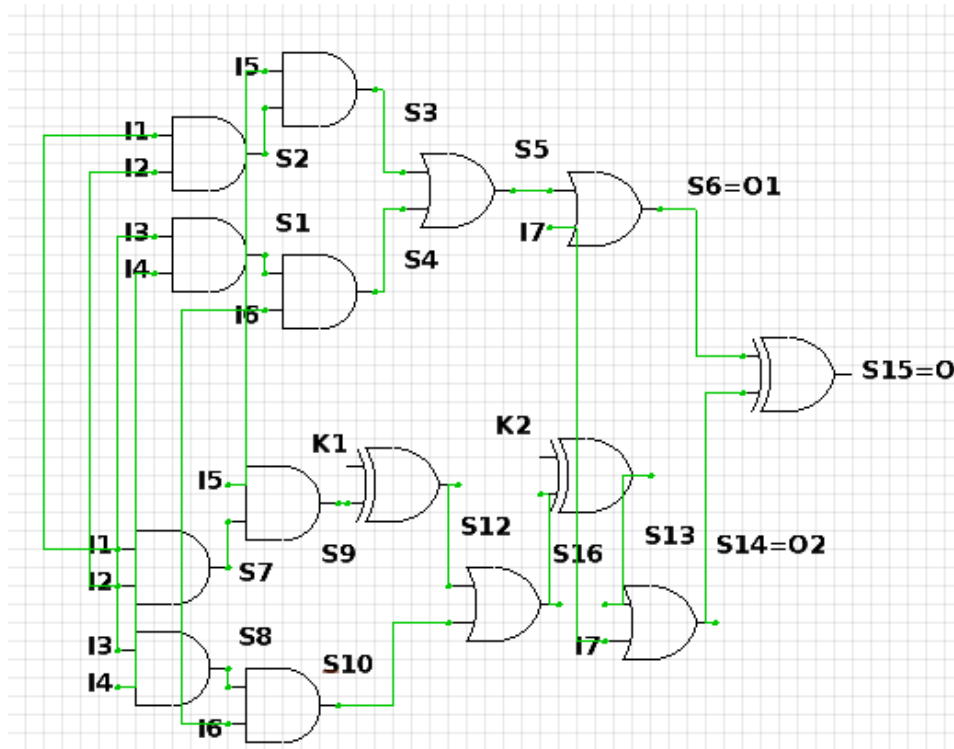


Figure 36 2 key XOR/XNOR-based encryption

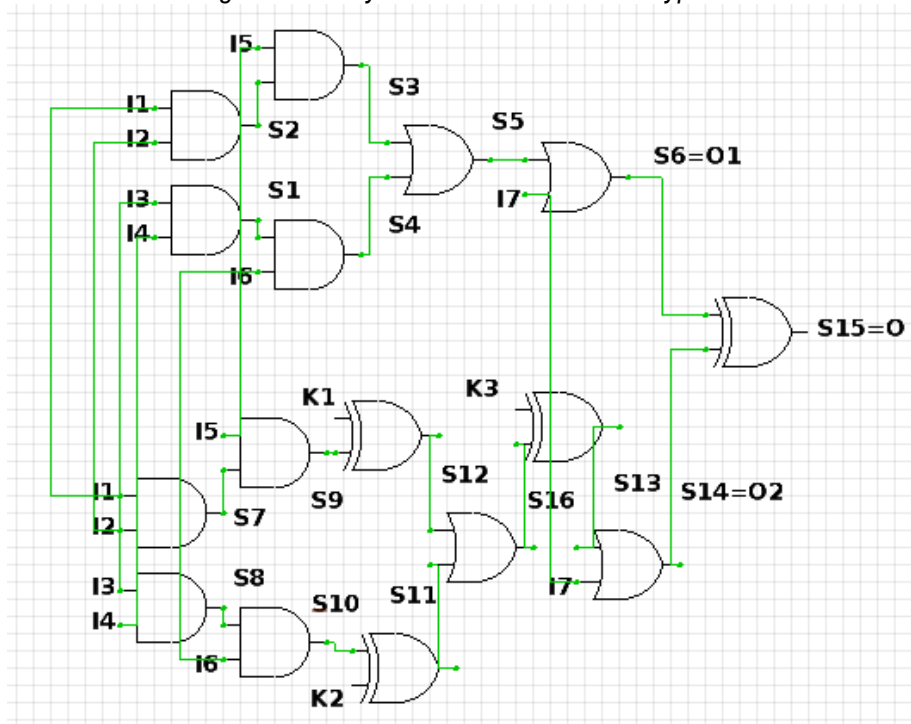


Figure 37 3bit key XOR/XNOR-based encryption

MUX-based encryption

In this case, with the MUX-based encryption methods, 2 input and 1 control bit multiplexors are used. In this case, the control bit of the MUX will act as a key bit and the right and wrong values of the key bit are determined by the VHDL description of the multiplexor. One of the inputs of the multiplexor is the correct signal and the other entry can come from any other point of the circuit. If there is no authorized usage of the circuit, the wrong input will be activated and the results of the circuit outputs will be wrong. Different locations of the MUX gate are evaluated to see how it affects the security of the circuits.

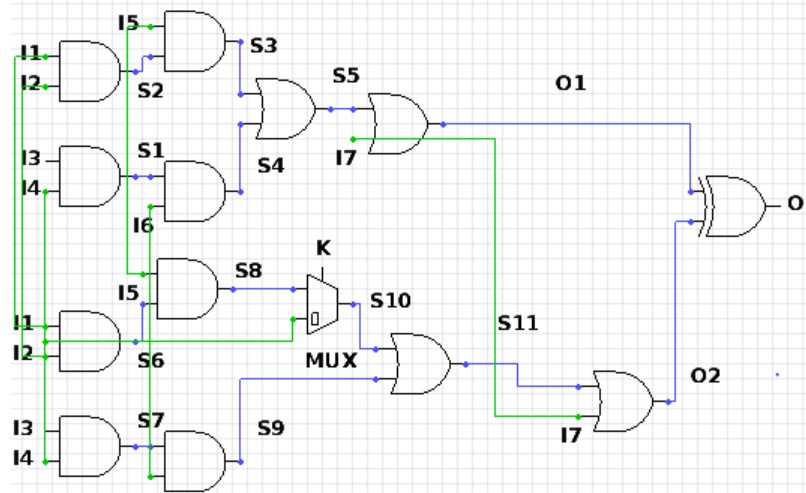


Figure 38 The first position of 1 MUX Gate, MUX-based encryption

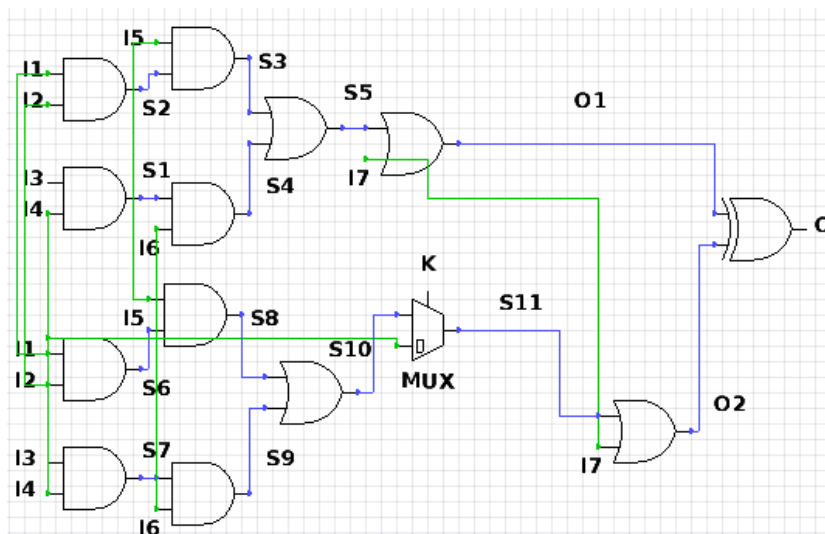


Figure 39 The second position of 1 MUX Gate, MUX-based encryption

After completing the implementation of XOR/XNOR based encryption, the technology MUX-based encryption methods are simulated and evaluated, like the previous technique, 2 and more encryption multiplexors are added. Various key gate locations are generated. It is challenging to recognize the type of the key gate, nevertheless. The difficulty of telling the „true wires“ from the „false wires“ determines how secure is the MUX-based method.

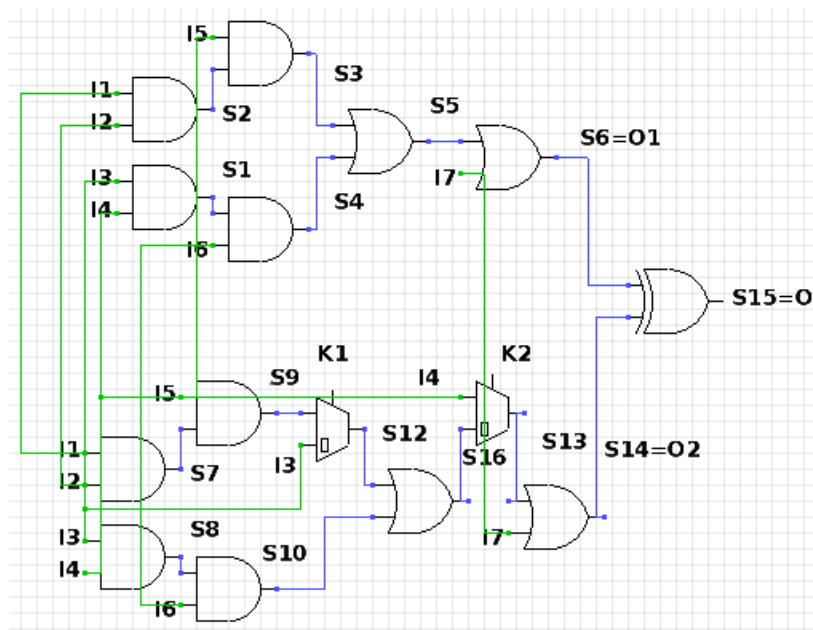


Figure 40 2bit key MUX-based encryption

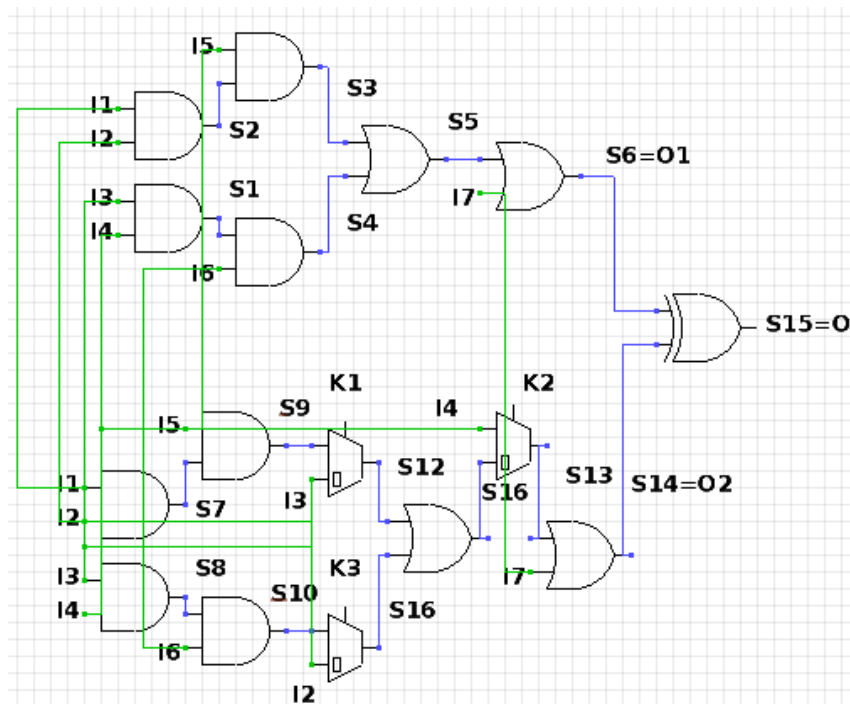


Figure 41 3-bit key MUX-based encryption

Encrypt flip-flop technique

The encrypt flip-flop technique requires the two outputs of the FF to be connected with 2 input MUX. In this case, the security key is applied as the select line input of the MUX. By using the wrong security key, the MUX will allow the wrong input at the output of the circuit. This connection of the MUX with the D-FF is done in order not to occur the scenario that even with the wrong key you might have the right result at the output of the circuit.

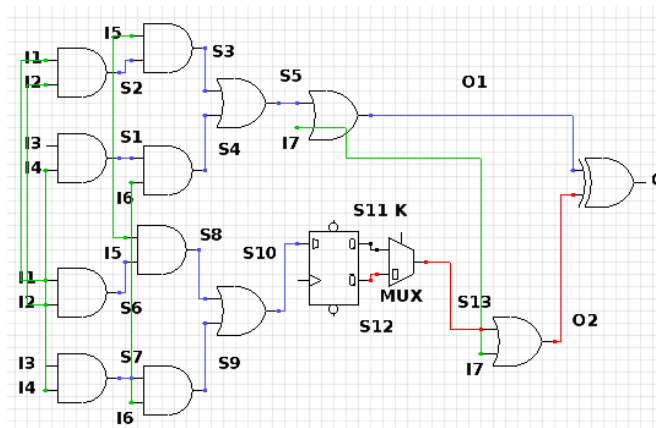


Figure 42 Encrypt flip-flop technique

The encrypt flip-flop (FF) technique in order to use two FF outputs need to be coupled with a 2 input MUX. In this instance, the choose line input of the MUX is applied with the security key. If the incorrect security key is used, the MUX will let the incorrect input at the circuit's output.

The MUX is connected to the D-FF in this way to avoid the possibility that you might press the wrong key and yet get the desired result at the circuit's output.

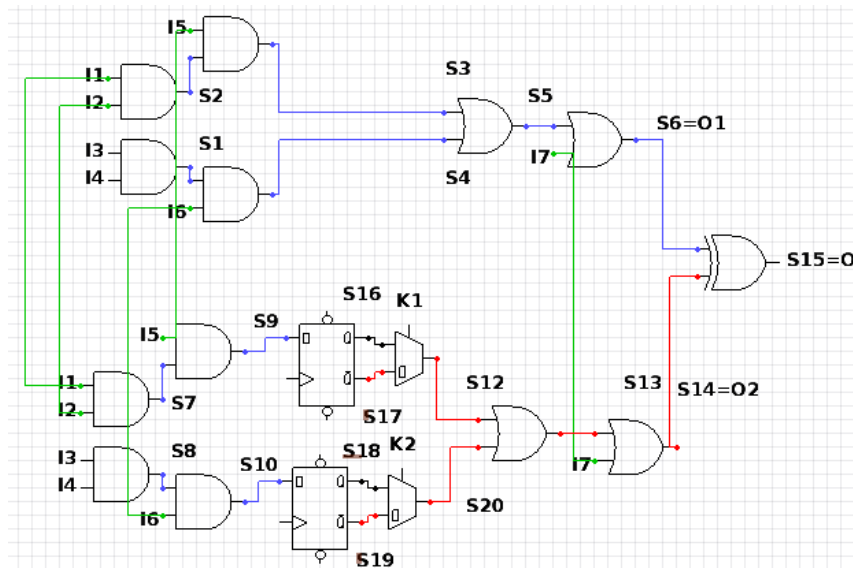


Figure 43 2-bit key FLIP-FLOP-based encryption

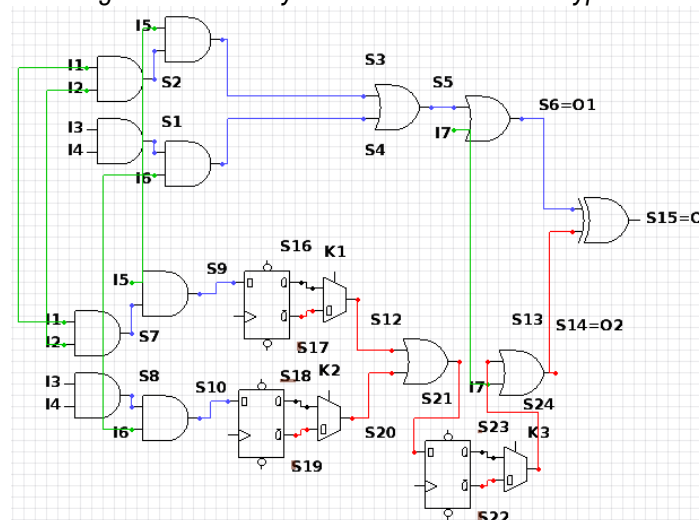


Figure 44 3-bit key FLIP-FLOP-based encryption

SCIENTIFIC-APPLIED AND APPLIED CONTRIBUTIONS

The scientific contributions of this thesis are:

- A. **Comprehensive Evaluation Framework for Logic Encryption**
A unified framework is developed to quantitatively evaluate logic-encrypted circuits using functional, fault-based, and SAT-based security metrics alongside hardware performance overhead.
- B. **Introduction of a Composite Security Score**
A novel composite metric is proposed to integrate multiple security dimensions into a single score, enabling direct comparison across benchmarks, encryption styles, and key-gate densities.
- B. **Demonstration of State-Dependent Encryption Advantages**
Experimental results confirm that Flip-Flop-based, state-dependent encryption provides superior resistance to fault injection and SAT-based attacks compared to purely combinational schemes.
- Г. **Security–Cost Driven Deployment Decision Algorithm**
A threshold-based algorithm is proposed to translate experimental security metrics into deterministic FPGA and ASIC deployment decisions.

The application-scientific contributions of the thesis are:

- Д. **Comparative Analysis of Logic Encryption Techniques**
A structured comparison of XOR/XNOR-based, MUX-based, and Flip-Flop-based encryption techniques is presented across both combinational and sequential benchmark circuits.
- E. **Scalability Analysis of Encryption Density**
The impact of key-gate density on security strength and implementation cost is systematically analyzed for 2, 5, 10, and 16 key-gate configurations.

The application contributions of the thesis are:

- Ж. **Benchmark-Driven Validation Using ISCAS Circuits**
The framework is validated on standard ISCAS-85 and ISCAS-89 benchmark circuits, ensuring reproducibility and relevance to real-world digital designs.
- 3. **Implementation-Ready Hardware Models**
Synthesizable VHDL implementations of all encryption techniques are provided, facilitating integration into practical communication-oriented hardware subsystems.

LIST OF PUBLICATIONS ON THE DISSERTATION

Publications:

1. E. Malaj, G.Marinova, Review on hardware solutions for cybersecurity of communication systems, 2020 28th National Conference with International Participation (TELECOM), Date of Conference: 29-30 Oct. 2020, Date Added to IEEE Xplore: 25 December 2020, INSPEC Accession Number: 20304410, DOI: 10.1109/TELCOM50385.2020.9299537, Publisher: IEEE, Conference Location: Sofia, Bulgaria], pp.129-132, IEEE Scopus. Indexed in IEEE Xplore, Scopus <https://ieeexplore.ieee.org/document/9299537>
2. E. Malaj, G.Marinova, Simulated algorithm RSA in Mail Server device, Published in: 2021 29th National Conference with International Participation (TELECOM), Date of Conference: 28-29 Oct. 2021, Sofia, Bulgaria, IEEE-Xplore, IEEE Scopus. Indexed in IEEE Xplore, Scopus <https://ieeexplore.ieee.org/document/9659623>

3. E. Malaj, G. Marinova, Review of cybersecurity hardware devices, The International Conference on Science, Technology, Business and Innovation of UBT assembles in Kosovo, Date of Conference: 04-05 Oct. 2021. <https://knowledgecenter.ubt-uni.net/cgi/viewcontent.cgi?article=3183&context=conference>.
4. E. G. Malaj, M. Ibro and G. Marinova, "Experimental setup for Hardware Encryption logic evaluation for e-learning", 2022 29th International Conference on Systems, Signals and Image Processing (IWSSIP), Sofia, Bulgaria, 2022, pp. 1-4, DOI: 10.1109/IWSSIP55020.2022.9854421. Conference Paper | Publisher: IEEE | Electronic ISBN: 978-1-6654-9578-3 Cited by: Papers (1), IEEE Scopus and WoS Indexed in IEEEExplore, Scopus <https://ieeexplore.ieee.org/document/9854421>.
5. G. Marinova, E. G. Malaj, & M. Ibro, (2023) "Hardware Solution for Cybersecurity of Unmanned Systems Critical Infrastructure", In Monitoring and Protection of Critical Infrastructure by Unmanned Systems, Vol. 63, pp. 127-138. IOS Press. DOI 10.3233/NICSP230010 Book Chapter | Publisher: IOS Press, IEEE Scopus. Indexed in IEEEExplore, Scopus <https://ebooks.iospress.nl/doi/10.3233/NICSP230010>.
6. E. G. Malaj, G. Marinova, "Effect of the Hardware Logical Encryption Type and Location on the Circuit Security ", 2023 IEEE International Workshop on Technologies for Defense and Security (TechDefense), 2023 IEEE Scopus. Indexed in IEEEExplore, Scopus <https://ieeexplore.ieee.org/document/10380923>
7. E. G. Malaj, G. Marinova, (2023) "Study of the Secrecy in Hardware Logical Encryption With Multiple Bit Keys", 2023 IEEE International Workshop on Technologies for Defense and Security (TechDefense), 2023, IEEE Scopus. Indexed in IEEEExplore, Scopus <https://ieeexplore.ieee.org/document/10380831>
8. E. G. Malaj, G. Marinova, (2024) "Identification Hardware Attacks of Integrated Circuit Design Data and Intellectual Property", 2024 IEEE International Workshop on Technologies for Defense and Security (TechDefense), 2024, IEEE Scopus. Indexed in IEEEExplore, Scopus <https://ieeexplore.ieee.org/document/10863477>
9. E. G. Malaj, G. Marinova, (2025) "Review on Hardware Logical Encryption" 2025 IEEE International Workshop on Technologies for Defense and Security (TechDefense), 2025, IEEE Scopus Indexed in IEEEExplore, Scopus
10. E. G. Malaj, (2025) "Algorithmic Framework for Evaluating Hardware Encryption Circuits" 2025 IEEE International Workshop on Technologies for Defense and Security (TechDefense), 2025, IEEE Scopus. Indexed in IEEEExplore, Scopus

SUMMARY

DEVELOPMENT OF NEW HARDWARE SOLUTIONS FOR CYBERSECURITY OF COMMUNICATION SYSTEMS

This thesis investigated the problem of securing communication-oriented hardware systems against reverse engineering, fault injection, and algorithmic de-obfuscation attacks through the use of logic encryption. With the increasing globalization of IC design and manufacturing, hardware intellectual property has become a critical security concern, particularly for communication infrastructure, embedded systems, and safety-critical applications. The central objective of this research was to develop and evaluate a systematic, quantitative framework for assessing the effectiveness and deployment feasibility of logic-encrypted hardware.

To address this objective, the dissertation proposed a unified evaluation methodology that combines functional obfuscation, fault resilience, key sensitivity, SAT attack resistance, and hardware implementation cost. Three representative logic encryption techniques XOR/XNOR-based, MUX-based, and Flip-Flop-based encryption were implemented and evaluated across a wide range of ISCAS-85 combinational and ISCAS-89

sequential benchmark circuits. Encryption density was systematically scaled using 2, 5, 10, and 16 key-gate configurations to capture the security–cost trade-off inherent in logic encryption design.

The experimental results demonstrate that logic encryption effectiveness is strongly dependent on both encryption density and circuit structure. Increasing the number of key-gates consistently improves functional obfuscation, key sensitivity, and resistance to SAT-based attacks. However, these security gains are accompanied by predictable increases in power consumption, area overhead, and delay.

A major finding of this work is the superior security scaling behavior of sequential, state-dependent encryption mechanisms. Flip-Flop-based encryption consistently outperformed purely combinational locking schemes across all benchmarks, particularly for large sequential circuits. The presence of memory elements amplifies key-dependent behavior over time, significantly increasing attack complexity and reducing fault observability.

To enable holistic comparison across benchmarks and encryption strategies, this dissertation introduced a Composite Security Score that integrates multiple security metrics into a single, deployment-oriented measure. This score proved effective in differentiating lightweight, moderate, and high-assurance security configurations and directly supported the formulation of deployment thresholds for FPGA and ASIC platforms.

One of the primary contributions of this thesis is the translation of experimental security analysis into actionable deployment guidelines. The proposed Security Cost Driven Deployment Decision Algorithm maps measured Composite Security Scores to recommended encryption densities and hardware platforms. Low scores favor FPGA deployment with sparse encryption for cost-sensitive applications, while high scores mandate high-density or sequential encryption and ASIC deployment for mission-critical systems.

In conclusion, this thesis demonstrates that effective hardware security cannot be achieved through isolated design choices or single-metric optimization. Instead, robust protection of communication-oriented hardware systems requires a balanced integration of encryption density, circuit structure, and platform-specific constraints. By providing a systematic evaluation framework, quantitative deployment thresholds, and validated experimental results, this research establishes a solid foundation for the design and deployment of secure hardware in an increasingly adversarial environment. The integration of threshold-based decision rules, supported by experimental data and visualized through flowchart-based deployment models, bridges the gap between academic security evaluation and real-world hardware design practice. This approach enables designers to make informed, metric-driven decisions rather than relying on ad-hoc or heuristic security assumptions. The methodologies and insights presented in this work are expected to remain relevant as hardware security challenges continue to evolve, and they provide a strong basis for future research into adaptive, side-channel-aware, and system-level hardware protection mechanisms.