Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)
Кафедра кібербезпеки
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр
(назва освітнього ступеня)

на тему: Дослідження вимог до фізичного та програмного захисту інформації на об'єктах критичної інфраструктури загроз і обмежень

Виконав: студент 6 курсу, групи СБм-61
спеціальності 125 Кібербезпека
(шифр і назва спеціальності)

		Сербичанський С.М
	(підпис)	(прізвище та ініціали)
Керівник		*Скарга-Бандурова І.С.*
	(підпис)	(прізвище та ініціали)
Нормоконтроль		*Лобур Т.Б.*
	(підпис)	(прізвище та ініціали)
Завідувач кафедри		*Загородна Н.В.*
	(підпис)	(прізвище та ініціали)
Рецензент		
	(підпис)	(прізвище та ініціали)

Тернопіль
2022

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра Кібербезпеки
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Загородна Н.В.
(підпис) (прізвище та ініціали)
«____» _____ 2022 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Магістр
(назва освітнього ступеня)

за спеціальністю 125 Кібербезпека
(шифр і назва спеціальності)

Студенту Сербичанському Сергію Миколайовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження вимог до фізичного та програмного захисту інформації на об'єктах критичної інфраструктури загроз і обмежень

Керівник роботи Скарга-Бандурова І.С.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «_25_» _листопада__ 2022 року № _4/7-966_____

2. Термін подання студентом завершеної роботи грудня 2022р.

3. Вихідні дані до роботи Наукові публікації про загрози хмарної безпеки та проблем безпеки хмарних середовищ

4. Зміст роботи (перелік питань, які потрібно розробити): Вступ, 1 Аналіз вимог до фізичного та програмного захисту інформації на об'єктах критичної інфраструктури, 1.1.Аналіз особливостей Smart Grid як об'єктів критичної інфраструктури та інформаційної діяльності, 1.2 Аналіз вразливостей в Smart Grid, 1.3 Існуючі рішення захисту мережі, 1.4 Загроза квантової обробки інформації, 2 Розробка інформаційної моделі Smart Grid для Mesh-клієнтів, 2.1 Теоретичні основи автентифікації з використанням відкритого ключа та дерев Меркла, 2.2 Експериментальне середовище, 2.3 Інформаційна модель в середовищі ns-3, 3 Моделювання програмного захисту інформації на об'єктах критичної інфраструктури, 3.1 Параметри експерименту, 3.2 Модель захисту із використанням дерев Меркла, 3.3 Модель захисту із використанням RSA, 3.4 Результати експерименту, 4 Охорона праці та безпека в надзвичайних ситуаціях, 4.1 Охорона праці, 4.2 Безпека в надзвичайних ситуаціях, Висновки, Перелік використаних джерел.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів) 1 Титульна сторінка. 2 Тема. Мета. Об'єкт. Предмет дослідження. 3 Завдання дослідження. 4. Порівняльна характеристика комунікаційних протоколів, 5. Порівняльна характеристика часу виконання ключів, 6. Приклад існуючої мережі в NS 3, 7. Концептуальна модель меш мережі, 8. Представлення дерева Merkle, 9. Модель стандартної меш мережі, 10. Результати тестування меш мережі розміром в 9 вузлів без проведення автентифікації, 11. Час взаємодії вузлів, 12. Меш мережа розміром в 64 вузла, 13. Таблиця вузлів в меш мережі 14. Результати тестування меш мережі розміром в 64 вузла з проведення автентифікації Tree Merkle, 15. Час автентифікації вузлів, 16. Результати тестування меш мережі розміром в 64 17. Порівняльні характеристики моделювання 18 Висновки. 19 Апробація результатів валіфіаційної роботи. 20 Завершальний слайд.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Осухівська Г.М., к.т.н., доцент		
Безпека в надзвичайних ситуаціях	Клепчик В.М., старший викладач		

7. Дата видачі завдання 14 листопада 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	14.11.2022-15.11.2022	*Виконано*
2.	Підбір наукових джерел про об1єкти критичної інфраструктури	16.11.2022-17.11.2022	*Виконано*
3.	Переклад та опрацювання наукових джерел про	17.22.2022-21.11.2022	*Виконано*
	дослідження методів захисту відомих меш мереж.		
4.	Розробка інформаційної моделі Smart Grid, для меш клієнтів	21.11.2022-23.11.2022	*Виконано*
5.	Проведення експиментів з використання моделей захисту RSA та Tree Merkle.	24.11.2022-27.11.2022	
			Виконано
6.	Оформлення розділу «Аналіз вимог до фізичного		
	та програмного захисту інформації на об’єктах критичної інфраструктури»	28.11.2022-30.11.2022	*Виконано*
7.	Оформлення розділу «Розробка інформаційної моделі		
	Smart Grid для Mesh-клієнтів»	01.12.2022-04.12.2022	*Виконано*
8.	Оформлення розділу «Моделювання програмного захисту інформації на об’єктах критичної інфраструктури»	05.12.2022-07.12.2022	*Виконано*
9.	Виконання завдання до підрозділу «Охорона праці»	08.12.2022-09.12.2022	*Виконано*
10.	Виконання завдання до підрозділу «Безпека в надзви-		
	чайних ситуаціях»	10.12.2022-11.12.2022	*Виконано*
11.	Оформлення кваліфікаційної роботи	12.12.2022-13.12.2022	*Виконано*
12.	Нормоконтроль	14.12.2022-15.12.2022	*Виконано*
13.	Перевірка на плагіат	9.12.2022	*Виконано*
14.	Попередній захист кваліфікаційної роботи	16.12.2022	*Виконано*
15.	Захист кваліфікаційної роботи	.12.2022	

Студент _____ Сербичанський С.М
(підпис) (прізвище та ініціали)

Керівник роботи _____ Скарга-Бандурова І.С.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Дослідження вимог до фізичного та програмного захисту інформації на об’єктах критичної інфраструктури загроз і обмежень // Кваліфікаційна робота освітнього рівня «Магістр» // Сербичанський Сергій Миколайович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп’ютерно-інформаційних систем і програмної інженерії, кафедра кібербезпеки, група СБм-61 // Тернопіль, 2022 // С. 82, рис. – 14, табл. – 6, додат. – 2, бібліогр. – 10.

Ключові слова: SMART GRID, MESH NETWORK, АВТЕНТИФІКАЦІЯ, RSA, TREE MERKLE, NS-3, МОДЕЛЬ, КРИТИЧНА ІНФРАСТРУКТУРА.

Кваліфікаційна робота присвячена дослідженню методів захисту об’єкту критичної інфраструктури електромережі, що використовує технологію Smart Grid.

У першому розділі проводиться аналіз особливостей Smart Grid: розгляд структури та основних компонентів. Також проводиться аналіз існуючих вразливостей. Розглядаються існуючі рішення захисту мережі Smart Grid.

В другому розділі розглядаються варіанти використання існуючих рішень для вдосконалення захисту мережі. Також відбувається детальний розгляд автентифікації вузлів в Smart Grid. Також проводиться дослідження алгоритмів автентифікації та розглядається середовище для проведення експериментальних досліджень.

У третьому розділі створено модель захисту з використанням автентифікації в меш мережі використовуючи алгоритми RSA та Tree Merkle. Проведені експериментальні дослідження, що визначають час автентифікації в Smart Grid, використовуючи створені моделі захисту. Наведено результати експериментальних досліджень та сформовані висновки, щодо використання алгоритмів автентифікації.

ANNOTATION

Study of requirements for physical and software protection of information on critical infrastructure objects of threats and restrictions // Qualification work of the educational level "Master" // Serbichanskyi Serhiy Mykolayovych // Ivan Pulyuy Ternopil National Technical University, Faculty of Computer Information Systems and Software of engineering, department of cyber security, SBm-61 group // Ternopil, 2022 // p. 82, fig. – 14, tab. – 6, add. – 2, bibliography – 10.

Keywords: SMART GRID, MESH NETWORK, AUTHENTICATION, RSA, TREE MERKLE, NS-3, MODEL, CRITICAL INFRASTRUCTURE.

The qualification work is devoted to the research of methods of protection of the object of the critical infrastructure of the power grid, which uses Smart Grid technology.

In the first section, an analysis of the features of the Smart Grid is carried out: a review of the structure and main components. An analysis of existing vulnerabilities is also carried out. Existing Smart Grid network protection solutions are reviewed.

The second section considers options for using existing solutions to improve network protection. There is also a detailed consideration of node authentication in the Smart Grid. Authentication algorithms are also being researched and an environment for conducting experimental research is being considered.

In the third section, a protection model was created using authentication in a mesh network using RSA and Tree Merkle algorithms. Experimental studies were carried out, which determine the time of authentication in Smart Grid, using the created protection models. The results of experimental studies and conclusions regarding the use of authentication algorithms are presented.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

AES – advanced encryption standard

AMI – advanced metering infrastructure

DoS – denial of service

ECC – elliptic curve cryptography

LFSR – linear-feedback shift register

LQI – link quality Indication.

PGP – pretty good privacy

PKI – public key infrastructure

POW – proof-of-work

VPN – virtual private network

WMN – wireless mesh networks

WSN – wireless sensor networks

КСЗІ – комплексна система захисту інформації

ЗМІСТ

ВСТУП

Актуальність теми. Інфраструктура електромережі формує функціональну основу сучасного суспільства. Дотепер мережі добре служили людству, але найближчим часом вони досягнуть межі. Якщо залишити так, як є, електромережі не будуть встигати за нашими вимогами. Очікується, що до 2050 року світове споживання електроенергії зросте втричі. Уряди та організації почали розробляти плани впровадження електричних мереж з покращеною функціональністю, надійністю та ефективністю. Ці нові мережі матимуть «передову децентралізовану цифрову інфраструктуру з двосторонніми можливостями для передачі інформації, керування обладнанням і розподілу енергії»[1]. Нова інфраструктура, яку зазвичай називають Smart Grid, зможе краще включати нові форми виробництва енергії, а також буде самовідновлюваною та більш надійною. Однак ці нові системи є складними та мають потенціал для численних вразливостей. Smart Grid об’єднує кілька добре відомих, але відмінних галузей, а саме електроенергетику, інформаційні технології та зв’язок. Ці галузі мають різні пріоритети та цілі. Наприклад, в електротехнічній промисловості найвищим пріоритетом є безпека людини. У галузі інформаційних технологій найвищим пріоритетом є конфіденційність, цілісність і доступність інформації. Для Smart Grid заходи кібербезпеки не повинні заважати безпечній та надійній роботі енергосистеми.

Останні роки, бездротові сітчасті мережі (wireless mesh networks - WMN) привернули багато уваги та стають дедалі популярнішими топологіями завдяки своїй економічній ефективності та надійності. За допомогою топології WMN можна охопити ту саму область, що й типовий WiFi, але з меншою кількістю маршрутизаторів. Це робить їх економічно привабливими у використанні, у тому числі в Smart Grid.

Мета і задачі дослідження**.** У цій роботі розглядається один з аспектів безпеки Smart Grid, а саме використання криптографічних систем з відкритим ключем в обладнанні електромережі в контексті потенційної майбутньої загрози квантової комп'ютерної атаки, та аргументується використання дерев Merkle на відміну від криптографічних систем із відкритим ключем у Smart Grid, зокрема, коли вони використовуються для автентифікації бездротових пристроїв.

В роботі буде використано симулятор мережі ns-3, в якому планується реалізувати дві схеми автентифікації: криптографічної системи з відкритим ключем, а саме RSA і дерева Merkle. Порівняння базуватиметься на наступних параметрах: час обчислення, використання пам'яті та кількість необхідних обмінів.

Задача полягає в тому, щоб показати, що дерева Merkle є альтернативою відкритому ключу з точки зору обчислювального часу та функціональності, але оскільки вони стійкі до майбутніх атак з боку квантового комп'ютера, мають потерційну перевагу у безпеці.

Об'єкт дослідження. Алгоритми автентифікації в мережах Smart Grid.

Предмет дослідження. Ефективність алгоритмів автентифікації в мереж Smart Grid.

Наукова новизна одержаних результатів кваліфікаційної роботи полягає у аналізі та дослідженні методів автентифікації у мережах Smart Grid, враховуючи такі параметри як швидкість, об'єм пам'яті, що потребує пристрої та складність виконання алгоритмів враховуючі загальнопоширені методи взлому, а також стійкість до взлому із використанням квантових обчислень. Обґрунтування використання дерева Merkle як алгоритму для автентифікації у мережах Smart Grid. Особистий внесок полягає у розробці та тестуванні моделей, що дозволяють вирішити поставлені задачі.

Практичне значення одержаних результатів. Полягає в тому, що основні наукові положення дисертації реалізовані у виді розрахункових моделей та

програмних засобів, направлені на розширення можливостей при розробці новітніх систем захисту в мережах Smart Grid.

Апробація результатів роботи. Основні результати проведених досліджень обговорювались на: Міжнародній науковій конференції „Іван Пулюй: життя в ім’я науки та України“ (до 175-ліття від дня народження) (м.Тернопіль), X науково-технічній конференції «Інформаційні моделі, системи та технології» (м.Тернопіль), XI Міжнародній науково-технічній конференції молодих учених та студентів «Актуальні задачі сучасних технологій» (м.Тернопіль).

Публікації. Основні результати кваліфікаційної роботи опубліковано у праці конференції (див. Додаток А).

РОЗДІЛ 1. АНАЛІЗ ВИМОГ ДО ФІЗИЧНОГО ТА ПРОГРАМНОГО ЗАХИСТУ ІНФОРМАЦІЇ НА ОБ'ЄКТАХ КРИТИЧНОЇ ІНФРАСТРУКТУРИ. ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕНЬ

Існують різні стандарти умов та правил для захисту критичної інфраструктури. Зокрема до найуживаніших стандартів, які використовують як для звичайних підприємств так і для підприємств критичної інфраструктури є ISO/IEC 27001[2]. Для об'єктів критичної інфраструктури за законом «Про основні засади забезпечення кібербезпеки України» об'єкти критичної інфраструктури повинні використовувати КСЗІ[3]. Звісно будь які існуючі стандарти, збірники вимог та інші мають свої недоліки, про те дозволяють створювати та розробляти надійні системи безпеки, що здатні протистояти більшості наявних загроз. Об'єкти критичної інфраструктури є над важливими для існування держави.

Проте використання нових технологій до прикладу держави у смартфоні (Дії), використання мереж Smart Grid, новітньої технології для ефективного розподілу електроенергії в мережах живлення провокує і нові небезпеки які не є описаними в жодних із стандартів.

Використання мереж Smart Grid є потребою у зв'язку із збільшенням попиту на електроенергію у Європейському континенті. Так як мережа Smart Grid дозволяє не створюючи нові системи генерування, ефективно розподіляти електроенергію за рахунок швидкого інформування місць генерації електроенергії. Також мережа здатна забезпечувати ефективну розподіл електроенергії під час стихійних лих, військових дій тощо…

Мережа Smart Grid є ефективним засобом, що використовує пристрої збору та передачі інформації не використовуючи загального сервера. Проте виникають ризики використанні підроблених пристроїв, що будуть під'єднанні до мережі, компрометація даних, що передаються, з втратою конфіденційності користувачів. Так як критична інфраструктура є глобальною і її модифікація передбачає суттєві витрати на модифікацію необхідно також

враховувати суттєвий розвиток квантових технологій обчислення, що здатні нівелювати використання протоколів із відкритим ключем.

1.1 Аналіз особливостей Smart Grid як об’єктів критичної інфраструктури та інформаційної діяльності

Smart grid (розумна електромережа) - інтелектуальна електрична мережа, в якій існує зв'язок між усіма учасниками енергетичного ринку, спрямована на надання енергетичних послуг, зниження витрат і підвищення ефективності, а також інтеграцію розподілених джерел енергії, включаючи відновлювану енергію. джерела.

Основною особливістю мережі Smart Grid на відміну від інших об’єктів критичної інфраструктури є відсутність певного централізованого об’єкту, відповідно використання технологій захисту мережі, захист екрануванням від перехоплення даних за межею об’єкта не можливо реалізувати.

Мережа Smart Grid використовує датчики, які зчитують та аналізують дані про використання електроенергії в окремих господарствах, будинках чи навіть квартирах. Використовуючи технології меш мереж передають дані між собою, а також повідомляють місця генерації електроенергії в тих чи інших районах. Що в свою чергу дозволяє збільшити кількість непостійних джерел живлення таких як зелена енергетика (вітряки, сонячні панелі).

Мережа Smart Grid є повністю автоматичною та включає в себе системи моніторингу та управління, а також системи відновлення після збоїв в роботі.

Дозволяє в режимі реального часу передавати між клієнтами та виробниками електроенергії, аналізуючи та враховуючи типи виробництва електроенергії.

Захисту потребує кожен окремий пристрій, що збирає інформацію в електромережі. Так як пристрої розміщуються в різних територіях, які не можливо виокреми в окрему безпечну зону, необхідно забезпечувати і

фізичний захист кожного пристрою так і програмний захист, щоб забезпечити систему від перехоплення, компрометації та зміни даних.

1.1.1 Case studies

В Індії в 31 липня 2012 року стався найбільший у світі збій електроенергії, від якого постраждали 700 мільйонів людей, у тому числі столиця країни Нью-Делі [5]. Потяги та системи метро зупинилися, а приватні підприємства та лікарні приготували свої генератори. Сотні шахтарів застрягли у вугільних шахтах у Західній Бенгалії. Незважаючи на рекордні розміри, блекаут не був зовсім несподіваним. В Індії хронічно не вистачає електроенергії, і в індійських містах регулярно відбуваються тимчасові відключення електроенергії.

Причина 1: Не спрацювала система екстрених відключень;

Причина 2: Не прорахована зміна потреби використання електроенергії;

Причина 3: Відсутність розподілення систем та пріоритетності відключення електроенергії в надзвичайних умовах.

10 листопада 2009 року через сильний шторм, що накрив весь бразильський штат Парана та половину Парагваю або ж за іншою версією хакерською атакою, передача електроенергії від «Ітайпу» раптово перервалася. Ураганні вітри пошкодили три магістральні високовольтні лінії електропередачі (сама ГЕС не постраждала). Аварія призвела до масових відключень електроенергії, повністю зануривши Парагвай у пітьму, а також залишивши Сан-Паулу, Ріо-де-Жанейро, Еспіріту-Санту та інші бразильські міста без світла на кілька годин. Загалом без електрики залишилося майже 50 мільйонів людей.

Парагвай залишився без електроенергії не маючи резервних джерел генерування електроенергії, в той самий час Бразилія маючи інші засоби генерування не втратила енергосистему повністю, проте через відсутність чіткого розподілу регіони які залишились без постачання, отримали

електроенергію тільки тоді коли було відновлено постачання з «Ітайпу». З чого можна зробити висновки, навіть за умов недостатньої генерації електроенергії, з допомогою ефективних систем моніторингу та розподілу можна зменшити негативний ефект та забезпечити електроенергією критичну інфраструктуру.

14 серпня 2003 року каскадний збій в електромережі занурив у темряву понад 50 мільйонів людей на північному сході США та Канади. Це було найзначніше відключення електроенергії в Північній Америці з економічними наслідками у десять мільярдів доларів.

В штаті Огайо серія великих коливань потужності в діапазоні від 2000 до 4000 мегават вплинула на з'єднання мереж Онтаріо в Мічигані та Нью-Йорку. У результаті 14 серпня північно-східна енергосистема Сполучених Штатів і частини енергосистеми Онтаріо почали відключатися. Приблизно 61 800 мегават споживацького навантаження було перервано, що вплинуло на населення понад 50 мільйонів людей. Відновлювальні роботи тривали більше дев'яти днів до завершення надзвичайного стану 22 серпня.

Відключення електроенергії 2003 року було спровоковано простим заростанням дерев в Огайо. А також перенавантаження електромережі виникло в зв'язку із спекотною погодою, та збільшення кількості використовуваних кондиціонерів, а також відключенням деяких елктрогенеруючих станцій.

FirstEnergy, яка складається з семи комунальних компаній, була доменом, де сталася помилка. Правила Північноамериканської ради з регулювання електроенергетики (NERC) вимагають, щоб комунальні послуги повідомляли своїх сусідів про брак електроенергії. FirstEnergy цього не зробила. Лінії електропередач зазвичай розширюються та звужуються через вплив навантаження та погоди. Цього дня в районі Клівленд-Акрон, штат Огайо, провисання високовольтних ліній електропередач увійшло в контакт із зарослими гілками дерев, що спричинило замикання на землю.

Збій цих ліній електропередачі згодом спричинив стрибки напруги та відключення на лініях нижчої напруги. Потім ці несправності спричинили перенапруги та відключення сусідніх високовольтних ліній електропередачі, що о 4:06 EDT спричинило неконтрольований каскадний збій.

Причина 1: нездатність FirstEnergy оцінити та зрозуміти недоліки своєї системи, зокрема щодо нестабільності напруги. FirstEnergy не працював із відповідними критеріями напруги.

Причина 2: Неадекватне усвідомлення ситуації у FirstEnergy.

Причина 3: неспроможність організації надійності об’єднаної мережі забезпечити ефективну діагностичну підтримку в реальному часі.

Підсумовуючи можна стверджувати, що в усіх випадках найбільших блекаутів в історії людства були різні причини вплив яких можна було б зменшити використовуючи децентралізовану цифрову інфраструктуру з двосторонніми можливостями для передачі інформації, керування обладнанням і розподілу енергії або ж і зовсім не допустити.

1.1.2 Структура та основні компонети Smart Grid

Основна мережа Smart Grid побудована ієрархічно: домашні мережі (HAN), мережі сусідів (NAN) і глобальні мережі (WAN).

NAN отримує дані з кількох мереж HAN і забезпечує магістраль для передачі даних постачальникам електроенергії. Розгортання передової інфраструктури вимірювання (AMI) починається з встановлення інтелектуальних лічильників на місцях. Інтелектуальні лічильники надають детальну інформацію про споживання електроенергії в місцях обслуговування та зазвичай утворюють бездротові мережі.

Елементи Smart Grid це датчики та вимірювачі, що надають оцінку стабільності енергосистеми, попереджують крадіжку електроенергії, проводять моніторинг стану обладнання. Включать пристрої для вимірювання часу споживання та в ціноутворення, а також управління захисними

механізмами електромережі.

На рівні кінцевих користувачів, знаходять цифрові лічильники, які саме зчитують кількість електроспоживання, та передають дані в глобальну мережу.

А також має змогу вмикати та вимикати електроживлення у кінцевого користувача, тим самим реагуючи на надзвичайний стан в електромережах.

Також ведуться дебати про можливість кінцевого користувача налаштовувати енергоживлення враховуючи вартість електрики статичних пристроїв, до прикладу бойлер. Це дає нам ранній погляд на проблеми, які можуть виникнути під час переходу на Smart Grid.

1.1.3 Бездротові сітчасті мережі (Mesh-мережі) в Smart Grid і проблеми їх кібербезпеки

У бездротовій сітчастій мережі кожен вузол є одноранговим, відповідно є відсутній спеціальний вузол базової станції (сервер). Потім повідомлення може бути направлено через кожен вузол до кінцевого пункту призначення. Це робить їх більш надійними, оскільки кожен вузол має лише передавати дані наступному вузлу, і кожен вузол зазвичай підключений до кількох інших вузлів.

Додаткова надійність полягає в тому, що може бути кілька маршрутів від джерела до поглинача. Здатність mesh-мережі створювати та змінювати маршрути також динамічно підвищує надійність бездротового з’єднання. Недоліком відсутності центральної інфраструктури є те, що кожен вузол буде складнішим, а також дорожчим (як з точки зору грошової вартості, так і потужності, необхідної для їх роботи). Для пристроїв, що живляться від батареї, це може бути важливою проблемою.

Існує чотири основні обмеження на WMN, а саме:

– обчислювальна потужність;
– час автономної роботи;

– мобільність і пропускна здатність.

Пропускну здатність WMN можна збільшити, додавши більше вузлів. Зазвичай WMN налаштовується на трьох рівнях. Перший рівень — це Mesh Clients, які підключаються до мережі через Mesh Routers на другому рівні. Потім Mesh-маршрутизатори підключаються до Mesh Gateways на третьому рівні, що забезпечує доступ до інших мереж, таких як Інтернет.

WMN надзвичайно вразливі до атак через їх топологію, що динамічно змінюється, відсутність традиційної інфраструктури безпеки та бездротовий характер. Відомо, що бездротовий зв’язок типів IEEE 802.11 і 802.15 дуже легко підслухати сигнал. Безкоштовні мережеві сніфери, такі як Kismet і WireShark, дозволяють бачити мережеві пакети, які стоять за мережевими зв’язками. Сніфінг пакетів є повністю пасивним, тобто цільова система безпосередньо не бачить ці дії. Пасивні атаки можуть порушити конфіденційність. WMN також вразливі до активних атак, коли цільова система безпосередньо бачить ці дії.

Стандарт ZigBee визначає набір комунікаційних протоколів для бездротових мереж малого радіусу дії з низькою швидкістю передачі даних. Пристрої ZigBee працюють у діапазонах частот 868 МГц, 915 МГц і 2,4 ГГц і мають максимальну швидкість передачі даних 250 К біт на секунду. ZigBee забезпечує надзвичайно низьке споживання та ефективність (завдяки адаптованому робочому циклу, низькій швидкості та радіозв’язку з низьким покриттям) і дозволяє створювати великомасштабні мережі для WPN, що робить його одним із найзручніших стандартів для цієї мети, порівняння з іншими протоколами наведено на рисунку 1.1[4].

Standard	ZigBee/IEEE 802.15.4	Bluetooth	UWB	IEEE 802.11 b/g
Working frequency	868/915 MHz, 2.4GHz	2.4 GHz	3.1 - 10.6 GHz	2.4 GHz
Range (m)	30 – 75+	10 – 30	~10	30 – 100 +
Data rate	20/40/250 kbps	1 Mbps	100+ Mbps	2 – 54 Mbps
Devices	255 – 65k	8		50 – 200
Power consumption	~1 mW	~40 – 100 mW	~80 – 300 mW	~160 mW – 600W
Cost ($US)	~2 – 5	~4 – 5	~5 – 10	~20 – 50

Рисунок 1.1 – Порівняльна характеристика комунікаційних протоколів

Стандарт ZigBee складається з чотирьох рівнів мережевого протоколу. Фізичний рівень (PHY). Це рівень, який забезпечує можливості передачі даних. Для пристроїв визначено три різні стани: передача, прийом і сплячий режим. Це дозволяє пристрою економити енергію, коли робочі цикли визначені та пристрій перебуває в режимі сну. Він також характеризує якість/потужність зв’язку отриманого сигналу зв’язку – Link Quality Indication (LQI). Канальний рівень. Це рівень, відповідальний за передачу даних між вузлами мережі. Він представляє на підрівні MAC структуру суперкадру. Суперкадр визначається часом між двома маяками, відправленими координатором мережі. Суперкадр можна розділити між активним і неактивним періодами, це разом із сплячим станом дозволяє економити енергію під час неактивних періодів. Наступним є рівень NWK, який відповідає за керування формуванням мережі та маршрутизацією. Нарешті, верхній рівень — це рівень програм (APL), на якому розміщено об’єкт програми.

Перші два рівні, PHY і MAC, рівні визначаються стандартом IEEE 802.15.4. ZigBee дотримується стандарту 802.15.4, але потім виходить за його рамки, реалізуючи два верхніх рівні.

Існує кілька переваг використання протоколу ZigBee порівняно з іншими протоколами для WSN.

Однією з головних переваг є те, що ZigBee стандартизовано на всіх рівнях, це гарантує сумісність продуктів від різних виробників.

Іншою перевагою є потужність сітки, пристрої, як правило, з’єднуються з кожним ближнім пристроєм, що робить кожен вузол мережі доступним з будь-якого іншого вузла та розширює мережу географічно, а також забезпечує самовідновлення, якщо кращий шлях до вузла там не вдається це інший шлях для досягнення вузла. Чим більше у вас пристроїв, тим надійніша мережа.

Низьке споживання енергії та робота в мережі навіть без батареї (Green Power). Пристрої для збору енергії не мають акумуляторів, одержуючи їх шляхом вилучення необхідної енергії з навколишнього середовища (шляхом натискання на рух, світло, п’єзо/тиск або ефект Пельтьє). Це особливо ефективно для пристроїв, які лише іноді підключаються до мережі (коли вони мають живлення), і дозволяє цим пристроям безпечно вмикатися та виключатися з мережі, тому вони можуть бути вимкнені більшу частину часу та не потребувати енергії.

Завдяки високій масштабованості мережі ZigBee можуть працювати з тисячами пристроїв, і вони спілкуватимуться один з одним за найкращим доступним шляхом.

Certicom — це криптографічна компанія, яка надає послуги автентифікації в мережах ZigBee. Вони мають багато патентів, пов’язаних із системою криптографії з еліптичним ключем (ECC). ECC — це криптосистема з відкритим ключем, заснована на проблемі дискретного журналу. Certicom використовує ECC для їх автентифікації. Якби квантовий комп’ютер був доступний завтра, усі розумні лічильники, які використовують цю систему, стали б небезпечними.

1.2 Аналіз вразливостей в Smart Grid

Розумні електромережі потенційно можуть мати багато ризиків, і вони можуть вплинути не лише на організації, але й на постійних клієнтів. Ці ризики можуть становити суттєву загрозу конфіденційності людей, як-от

конфіденційна інформація про клієнтів, ризик викрадення інформації або припинення бізнесу назавжди. Ці ризики виникають не лише під час користування Інтернетом, але й впливають на клієнтів удома, тоді як зловмисники можуть збирати особисту інформацію.

Відмова в обслуговуванні (DoS) — це стратегічна атака, і будь-які атаки на доступність є частиною DoS-атаки. Що стосується Smart Grid, доступні провідні служби для Smart Grids, що означає, що Smart Grid має шанс отримати атаку типу «відмова в обслуговуванні». Підключення Smart Grid має бути безпечним і надійним. Підключення підключення має бути надійним і безпечним, оскільки Smart Grid розподіляє з'єднання між незліченними пристроями на більшій території за допомогою систем розподіленої архітектури. Якщо (Dos) атака відбудеться на Smart Grid, вона зазнає величезних втрат.

Розповсюдження зловмисного програмного забезпечення: основний ризик, з яким Smart Grid може зіткнутися через розповсюдження зловмисного програмного забезпечення, що викликає серйозне занепокоєння. Зловмисники можуть розробити зловмисне програмне забезпечення, яке можна використати для зараження серверів організації, а також зараження пристроїв. Використовуючи розповсюдження зловмисного програмного забезпечення, зловмисник може маніпулювати функціями пристроїв або систем, що дозволить зловмисникам отримати доступ для збору конфіденційної інформації.

Прослуховування та аналіз трафіку: Прослуховування та аналіз трафіку є типами атак спуфінгу. зловмисник може отримати конфіденційну інформацію, відстежуючи мережевий трафік. Smart Grid зіткнеться з цим ризиком через велику мережу, яку вона містить, Smart Grid включає багато мережевих вузлів, і важко підтримувати пристрої, підключені до великої мережі. Розумна мережа створює найбільший ризик викрадення даних, що є основною проблемою для захисту даних у всьому світі.

1.3 Існуючі рішення захисту мережі

Для покращення рівня безпеки в Smart Grid можна виділити декілька базових способів:

Шифрування. Найвищий доступний стандарт шифрування відомий як AES (Advanced Encryption Standard) 256-біт і використовується найбільш рекомендованими провайдерами VPN. 256-бітне шифрування, настільки безпечний, що його можна використовувати в банках та урядових установах в усьому світі для забезпечення безпеки своїх даних.

Smart Grid потребує захисту від зловмисного програмного забезпечення, оскільки вбудована система та системи загального призначення, підключені до Smart Grid, повинні бути захищені та захищені від кібератак. Для вбудованої системи потрібен ключ виробника, який можна використовувати для захисту продукту для перевірки програмного забезпечення. Основна причина безпеки вбудованої системи полягає в тому, що вбудовані системи доступні лише для запуску програмного забезпечення, яке надається виробником, і для перевірки програмного забезпечення потрібен ключ виробника, тоді як системи загального призначення підтримують програмне забезпечення сторонніх розробників, наприклад антивірусне програмне забезпечення постійно оновлюватиме антивірусне програмне забезпечення.

Мережа Smart Grid потребує вищої пропускної здатності для зв’язку, що також означає, що для автентифікації можна використовувати методи криптографії. Підтримка методів криптографії збільшить вартість, хоча вони забезпечують чудовий механізм автентифікації.

1.3.1 Автентифікація в Smart Grid

Зазвичай механізм автентифікації в no-WMN використовує централізовану систему, яка керує доступом на основі списків і сертифікатів.

Оскільки WMN децентралізовані, використання таких серверів не завжди можливо. З цією метою криптографію з відкритим ключем можна використовувати для автентифікації. Рекомендації NIST пропонують використовувати криптографію з відкритим ключем у Smart Grid, зокрема як основу для операцій автентифікації. У криптографії з симетричним ключем той самий ключ використовується для шифрування та дешифрування. Ці шифри швидкі та надійні, але є проблематичними щодо безпечного розподілу ключів. Тобто, з однаковим ключем на обох кінцях, якщо будь-який ключ скомпрометовано, вся система зламана. Відкритий ключ, який іноді називають асиметричною криптографією, використовує два різні ключі, тобто пару відкритий-приватний ключ. Це має перевагу в тому, що якщо будь-який ключ зламано, система не зламана. Відкритий ключ був розроблений у 1960-70-х роках урядовими та академічними дослідниками. Зважаючи на те, що симетричні шифри налічують тисячі років, відкритий ключ є дуже новим. Криптосистеми з відкритим ключем є дуже надійними та потужними, і якщо їх правильно впровадити, вони пропонують один із найкращих доступних на сьогодні захистів. За своєю суттю сила системи відкритих ключів ґрунтується на математичних задачах, які дуже складно вирішити; а саме, проблема факторизації або проблема дискретних логарифмів. Ці теми будуть розглянуті далі.

1.3.2 Автентифікація з використанням публічного ключа

Як і будь-яка інша схема шифрування, автентифікація відкритого ключа базується на певному алгоритмі. Існує кілька добре вивчених, безпечних і надійних алгоритмів, найпоширенішими з яких є RSA та DSA. На відміну від загальновідомих (симетричних або секретних) алгоритмів шифрування, алгоритми шифрування з відкритим ключем працюють з двома окремими ключами. Ці два ключі утворюють пару, яка є індивідуальною для кожного користувача.

RSA отримує свою безпеку через труднощі розкладання великих цілих чисел, які є добутком двох великих простих чисел. Помножити ці два числа легко, але визначення початкових простих чисел із загальної суми або розкладання на множники вважається неможливим через час, який займе використання навіть сучасних суперкомп'ютерів.

Алгоритм RSA реалізовується за наступними кроками:

Виберіть два великих простих числа, x і p. Прості числа мають бути великими.

– обчислити n = x * y;

– обчислити функцію ϕ (n) = (x -1)(y-1);

– виберіть ціле число e таке, що є простим до ϕ (n) та 1 < e < ϕ (n);

– обчислити d, d=1 m o d ϕ(n). D можна знайти за допомогою розширеного алгоритму Евкліда;

– пара (n, d) становить закритий ключ;

– дано відкритий текст P, представлений у вигляді числа, зашифрованого тексту C розраховується як: C = P^e m o d n;

Використання закритого ключа (n , d), відкритий текст можна знайти за допомогою:

$P = C^d$ m o d n.

1.3.3 Аргументи проти використання публічного ключа в Smart Grid

Відкритий ключ має багато накладних витрат з точки зору ресурсів, включаючи час і енергію. Багато пристроїв у цих мережах можуть мати брак ресурсів. Самі по собі це важливі питання, але в центрі уваги цієї роботи є загроза квантового комп'ютера. Інформаційна безпека викликає занепокоєння не тільки в тому, чи безпечна вона сьогодні, але чи буде вона безпечною через 10-30 років. Свого часу німецька машина Enigma була найсучаснішим у шифруванні даних, сьогодні її зламати – це складна задача домашнього завдання для студентів. Обладнання, встановлене в електричній мережі, має

залишатися на місці протягом багатьох років, і в Smart Grid воно має залишатися в безпеці під час роботи. Інтелектуальну електромережу потрібно будувати довговічною, і, наскільки це можливо, вона має бути «захищеною від майбутнього». З цієї причини дослідження потенційних проблем безпеки, пов'язаних з мережами, становить великий дослідницький інтерес на цьому етапі розвитку Smart Grid.

1.4 Загроза квантової обробки інформації

Для безпеки відкритого ключа існує загроза реалізації квантового комп'ютеру який може зламати проблему факторизації або дискретного журналу за поліноміальний час.

Ця робота зосереджена на проблемі факторизації, але варто згадати кілька добре відомих і широко використовуваних задач, а саме Діффі-Хеллмана, Ель-Гамаля та криптографія еліптичної кривої (ECC) є прикладами систем, які базуються на проблемі дискретного журналу. Більшість досліджень квантових обчислень все ще є теоретичними, але «Як тільки квантові комп'ютери стануть реальністю, усі загальноприйняті системи шифрування з відкритим ключем стануть абсолютно незахищеними». Практичні наслідки цього були б далекосяжними. Донедавна ця так звана квантова перевага або квантова «вищість» була лише теорією. Однак у 2019 році Google використав квантовий комп'ютер для виконання конкретного обчислювального завдання всього за 200 секунд. На те саме завдання, за словами компанії, найпотужнішому на той час цифровому суперкомп'ютеру знадобилося б 10 000 років .

Квантові обчислення — це використання квантових явищ, таких як суперпозиція та заплутаність, для виконання обчислень. Основною одиницею квантового комп'ютера є квантовий біт (або скорочено кубіт).

Кожен двійковий біт, який використовується в сучасних цифрових

комп’ютерах, представляє значення нуля або одиниці, кубіти представляють і нуль, і одиницю (або деяку комбінацію двох) одночасно. Це явище називається суперпозицією. Квантова заплутаність - це особливий зв'язок між парами або групами квантових елементів. Зміна стану одного елемента миттєво впливає на інші заплутані елементи — незалежно від відстані між ними. Складні обчислювальні завдання схожі на пошук виходу з лабіринту. Традиційний комп’ютер намагався б втекти, дотримуючись кожного шляху послідовно, поки не досяг би виходу. Суперпозиція, навпаки, дозволяє квантовому комп’ютеру спробувати всі шляхи одночасно. Це значно скорочує час на пошук рішення. Практика зміни алгоритмів показує, що це є дуже тривалим та складним процесом, а такі глобальні технології як Smart Grid, не можуть швидко змінюватися, тому в побудові безпеки слід враховувати і загрози квантової оброки інформації.

РОЗДІЛ 2. РОЗРОБКА ІНФОРМАЦІЙНОЇ МОДЕЛІ SMART GRID ДЛЯ MESH-КЛІЄНТІВ

Результати проведеного аналізу моделей, методів й інструментальних засобів, для моделювання таких як: Packet Trace, GN3 та інші не дозволяють реалізовувати подібні моделі, а також показали, що у відомих публікаціях не вирішеною є задача автентифікації, використовуючи дерева Merkle.

В даному контексті можна виділити 3 основні задачі магістерської роботи:

1) розробка методу автентифікації в меш мережах, який дозволив би збільшити ефективність та безпеку пристроїв, що використовуються в мережах Smart Grid.

2) Розробка стендової(лабораторної) моделі для проведення експериментів та оцінки ефективності існуючих алгоритмів автентифікації в меш мережах. А саме оцінки швидкодії та використання пам’яті пристроїв. Відповідно і потреби в системних характеристиках пристроїв, що дозволять ефективний захист, швидкодію.

Для проведення досліджень доцільно застосовувати методи експериментальних досліджень, на основі лабораторної моделі меш мереж з якої буде можливість оцінювати алгоритми автентифікації.

Також спостереження дадуть достатню кількість інформації про швидкість роботи меш мереж, надійність та об’єм пам’яті який необхідним.

Моделювання різних умов дасть достатню кількість інформації для того, щоб визначити наскільки розроблена модель відповідає потребам в мережах Smart Grid та чи доцільно використовувати систему, що розроблюється.

Отримані результати можна використовувати для розробки надійних пристроїв, що є складовою Smart Grid. Та надають змогу краще зрозуміти ефективність поширених алгоритмів автентифікації, в меш мережах.

Доцільно припустити, що отримані результати можна використовувати не тільки в Smart Grid, а й в інших технологіях, що будуть засновуватися на меш мережах. До прикладу в розумних будинках, або ж в логістичних хабах.

2.1 Теоретичні основи автентифікації з використанням відкритого ключа та дерев Меркла

Асиметричне шифрування (тобто шифрування з відкритим ключем або криптографія з відкритим ключем), також відоме як асиметрична криптографія, використовується для захисту файлів, каталогів і цілих пристроїв від несанкціонованого доступу та для обміну секретними повідомленнями, автентифікації. Це робиться за допомогою ключів для шифрування та дешифрування.

Щоб запустити асиметричне шифрування, одержувач генерує приватний і відкритий ключ. Партнер по спілкуванню може отримати доступ до відкритого ключа . Ця проста передача відбувається через центр сертифікації або сервер ключів, де зберігається ключ. Відправник шифрує своє повідомлення за допомогою відкритого ключа, а потім може надіслати його одержувачу як зашифрований текст. Після того, як це повідомлення було зашифровано, його може розшифрувати лише одержувач за допомогою свого закритого ключа . Тому, в принципі, ви можете вільно вибирати, який канал зв'язку використовувати; навіть якщо зашифроване повідомлення буде перехоплено зловмисником, його вміст залишиться таємним.

Ця одностороння функція є основною ідеєю асиметричної криптосистеми . Два ключі повністю незалежні один від одного. Навіть якщо зловмисник має доступ до відкритого ключа, він не може використовувати його, щоб зробити будь-які висновки щодо закритого ключа. Щоб переконатися в цьому, відкритий ключ використовує чітко визначені прості множники , які перемножуються разом і дають однозначний результат.

Приватний ключ , з іншого боку, працює виключно з результатом цього обчислення. Майже неможливо зробити будь-які висновки про те, які фактори були використані для отримання цього значення, оскільки існує незліченна кількість можливостей того, як воно могло бути досягнуте. На сьогоднішній день не існує жодної математичної процедури чи алгоритму, який би спростив це зворотне обчислення.

Для кращого пояснення алгоритму роботи криптографічних протоколів з відкритим ключем варто змоделювати користувачів (Аліса – користувач 1, Боб – користувач 2, Том – зловмисник).

Криптографія з відкритим ключем має три основні переваги:

- конфіденційність : лише Боб може прочитати повідомлення Аліси.
- автентичність : Аліса може «підписати» своє повідомлення, тому Боб знає, що лише Аліса могла його надіслати. Він також знає, що Том не міг підробити повідомлення під час передавання.
- невідмовність : Аліса не може заперечити, що вона написала (або принаймні побачила) вміст повідомлення пізніше.

Ці переваги розкрили багато застосувань для криптографії з відкритим ключем, від PGP і HTTPS. Він також використовується для сертифікатів захищеної оболонки, що дозволяє адміністраторам підключатися до будь-яких серверів, не запам’ятовуючи своїх паролів.

Існують певні занепокоєння, пов’язані з постійним використанням шифрування з відкритим ключем, включаючи адміністрування сертифікатів. Цифрові ключі, які використовуються для шифрування та підпису повідомлень, упаковані в цифрові сертифікати, які видають центри сертифікації (CA) — довірені центри для перевірки особи. Ця система відома як інфраструктура відкритих ключів (PKI).

Головним недоліком криптографії з відкритим ключем є низька швидкість шифрування. Це також вимагає значно більшої обчислювальної потужності. Для звичайних комп’ютерних систем це перестало бути суттєвою

проблемою. Проте говорячи про меш мережі, де енергоефективність вузлів, низька вартість виробництва, надійність та простота займають ключову роль. Враховуючи те, що системні характеристики більшості пристроїв, що використовуються в меж мережах мають доволі низькі показники.

Так як критична інфраструктура будується на довгий час, та її модифікація займає тривалий час, а важливість важко переоцінити потрібно звернути на довгострокову перспективу та проблеми, що виникають.

Наразі вся криптографія з відкритим ключем, яка використовується в реальному світі, базується на використанні наступних

обчислювально складних задачах:

– факторинг;
– дискретні логарифмічні прості числа;
– еліптичної криві.

Усе це можна ефективно вирішити за допомогою квантових комп’ютерів, що означає більш безпечні заміни які будуть потрібні в найближчі роки, оскільки технологія квантових обчислень вдосконалюється.

На даний момент це не короткострокова проблема, але дослідники активно працюють над квантовими обчисленнями, які дозволять комп’ютерам виконувати цю роботу за допомогою грубої сили. Ці машини обіцяють розв’язувати великі математичні проблеми, перевіряючи кожну ітерацію задачі одночасно, а не послідовно.

Дерево Merkle — це нелінійна двійкова хеш-деревоподібна структура даних. Кожен листовий вузол дерева зберігає хеш-значення елемента даних, тоді як середній вузол зберігає хеш хешів двох відповідних дочірніх вузлів. Основна перевага використання дерева Merkle полягає в тому, що кілька важливих фрагментів інформації можна перевірити щодо окремого елемента даних або набору даних у цілому без необхідності мати доступ до повного набору даних. Наприклад, можна перевірити, чи є окремий елемент даних частиною даного набору даних, або довести, що елемент даних насправді

є частиною більшого набору даних без необхідності зберігати та аналізувати повний набір даних. Саме завдяки цим практичним застосуванням дерева Merkle зазвичай використовуються в таких галузях, як Blockchain, які фундаментально базуються на мережах P2P, які часто включають сценарії, коли дані вибираються з джерела, достовірність якого не гарантована, і, таким чином, дані вибираються та перевіряються. одночасно. Введення дерев Меркла в рівняння може допомогти запобігти таким проблемам, як синхронізація повного набору даних лише для усвідомлення того, що його неможливо перевірити, таким чином заощаджуючи багато часу та пропускної здатності.

Мережева автентифікація Merkle на основі дерева — це протокол, який перевіряє, що перевіряльник і верифікатор володіють однаковими даними. Таким чином, на відміну від публічної верифікації, передбачається, що верифікатор має деяку таємну (тобто не загальнодоступну) інформацію про дані, які підлягають перевірці.

Виходячи з твердого припущення про те, що неможливо знайти прообраз заданого значення хеш-функції за обчислювально прийнятний час, можна гарантувати, що лише об’єкти, які володіють однаковими даними, можуть отримати те саме дерево Меркла. Коротше кажучи, безпека автентифікації на основі дерева Merkle базується на безпеці використовуваної хеш-функції. Тому верифікатор зберігає лише значення кореневого вузла дерева та видаляє решту метаданих після створення дерева.

Алгоритми з відкритим ключем використовують дорогу модульну арифметику, експоненціальні операції і тому не підходять для mesh-клієнтів. Альтернативою використанню ресурсоємної автентифікації відкритого ключа з уразливим квантовим комп’ютером є система на основі дерев Меркла. Загальновідомо, що алгоритми на основі хешування, такі як MD5 і SHA-2, обчислювально дешевші, ніж алгоритми з симетричним ключем, які, у свою чергу, обчислювально менші, ніж алгоритми з відкритим ключем. Популярні криптографічні хеш-функції, такі як SHA-1 або MD5, працюють так само, як

блокові шифри. Тобто вони беруть звичайні тексти та розбивають їх на блоки фіксованого розміру, а потім ітерують за допомогою функції протягом деякої кількості раундів. Вони вважаються безпечними, якщо зіткнень не виявлено; SHA-1 було зламано близько 10 років тому. Вони повинні бути швидкими та мати ефект, що невеликі зміни вхідних даних призведуть до великих змін у вихідних даних. Дерева Merkle пропонують недорогу автентифікацію для mesh-клієнтів.

Криптографічні хеш-функції додають функції безпеки до типових хеш-функцій, що ускладнює виявлення вмісту повідомлення або інформації про одержувачів і відправників.

Властивості хеш-функцій

Щоб бути ефективним криптографічним інструментом, хеш-функція повинна володіти такими властивостями;

Pre-Image Resistance:

- ця властивість означає, що обчислювально важко повернути хеш-функцію назад.
- іншими словами, якщо хеш-функція h створила хеш-значення z, тоді важко знайти будь-яке вхідне значення x, яке хешує z.
- ця властивість захищає від зловмисника, який має лише хеш-значення та намагається знайти вхідні дані.

Second Pre-Image Resistance:

- ця властивість означає, що, враховуючи вхідні дані та їх хеш, буде важко знайти інші вхідні дані з тим самим хешем.
- іншими словами, якщо хеш-функція h для вхідного параметра x створює хеш-значення h(x), тоді буде важко знайти будь-яке інше вхідне значення y, таке, що h(y) = h(x).
- ця властивість хеш-функції захищає від зловмисника, який має вхідне значення та його хеш і хоче замінити інше значення як законне значення замість вихідного вхідного значення.

Collision Resistance:

– ця властивість означає, що буде важко знайти два різні вхідні дані будь-якої довжини, які призведуть до того самого хешу. Ця властивість також називається хеш-функцією без колізій.

– іншими словами, для хеш-функції h важко знайти будь-які два різні входи x і y, щоб h(x) = h(y).

– оскільки хеш-функція є функцією стиснення з фіксованою довжиною хешу, неможливо, щоб хеш-функція не мала колізій. Ця властивість відсутності колізій лише підтверджує, що ці колізії має бути важко знайти.

– завдяки цій властивості зловмиснику дуже важко знайти два вхідних значення з однаковим хешем.

Сила схеми автентифікації дерева Merkle полягає в наявності безпечної хеш-функції, і практичні криптографічні хеш-функції існують. Метою хеш-функції є створення «відбитка» повідомлення, тобто хеш-функція s() застосовується до файлу M і створює s(M), який ідентифікує M, але набагато менший.

2.1.1 Аналіз складності RSA

Складність будь-якого криптографічного алгоритму може вимірюватися з точки зору часу, простору або енергії, необхідної для того щоб шифрувати та розшифровувати повідомлення. Отже, складність описує обчислювальні зусилля, необхідні для криптосистеми для шифрування та дешифрування даних.

Ідея RSA базується на тому, що велике ціле число важко розкласти на множники. Відкритий ключ складається з двох чисел, де одне число є множенням двох великих простих чисел. І закритий ключ також походить від тих самих двох простих чисел. Отже, якщо хтось може розкласти велике число на множники, приватний ключ буде зламано. Тому міцність шифрування повністю залежить від розміру ключа, і якщо ми подвоюємо або потроїмо

розмір ключа, міцність шифрування зростає експонентно. Ключі RSA зазвичай можуть мати довжину 1024 або 2048 біт, але експерти вважають, що 1024-бітні ключі можуть бути зламані найближчим часом.

Проте із розвитком квантових систем обчислення, які можуть виконувати задачі одночасно, задачі, що базуються на складності факторизації великих чисел не зможуть надавати потрібного рівня захисту.

– складність обчислювання часу:

Публічний ключ: N,e;

Закритий ключ d;

$$\lg e = O(1), \lg d \leq \beta \; and \lg N \leq \beta \qquad (2.1)$$

Тоді застосування відкритого ключа вимагає O(1) модульних множень і використовує O(ß2) бітові операції. Застосування секретного ключа вимагає O(ß) модульне множення з використанням O(ß3).

Порівняльні характеристики використання RSA з різною довжиною ключа зображено в таблиці 2.1

Таблиця 2.1 – Порівняльна характеристика часу виконання ключів

Час виконання для генерації ключів і шифрування				
Key size RSA	1024 bit	2048 bit	3078 bit	7680 bit
Час вимірювання	102.89 3 мл	127.83 5 мл	149.27 2 мл	164.51 5 мл

– складність пам’яті:

Споживання пам’яті в RSA порівнюючи з деревами Merkle, має значну перевагу так як немає потреби в зберіганні великого дерева.

Для кожного вузла необхідно зберігати тільки свій закритий ключ.

Також значною перевагою є те, що один набір ключів може аутентифіковувати необмежену кількість пристроїв.

– складність повідомлень:

В розробці для проведення емуляції буде використано обмежену довжину повідомлення в 32 біта, про те в реальній складності повідомлення буде використано довжину як мінімум в 1024 біта, а максимально ефективний розмір повідомлення враховуючи обмежені системні характеристики в довжину в 2048 біт. Таким чином складність повідомлення буде значити O(ß).

2.1.1 Аналіз складності дерев Меркла

Дерево Merkle – це деревовидна структура, де кожен листовий вузол є криптографічним хешем базових даних, а кожен нелистовий вузол є хешем своїх прямих нащадків. Як правило, дерева Merkle мають коефіцієнт розгалуження, що дорівнює двом, тобто кожен вузол має до двох дочірніх елементів. У верхній частині кожного дерева знаходиться кореневий хеш, який змінюється кожного разу, коли до дерева додається новий листковий вузол.

Перевагою використання дерев Merkle є змога протистояти загрозам, що можуть виникнути у зв’язку з розвитком технологій квантових обчислень.

Про те до основних недоліків можна віднести потребу знати кількість пристроїв, що будуть працювати в мережі.

Також одним з основних недоліків використання дерев Merkle є потреба в більшій кількості даних, що повинні зберігатися на пристрої.

Враховуючи те, що дерева Merkle вирішують проблеми, що виникають в інших подібних алгоритмах, саме ця розробка дозволила використовувати дерева Merkle в технологіях blockchain.

Основною перевагою, що надає використання дерев Merkle є відсутність потреби в зберіганні всього хешу повідомлення для відтворення.

– складність обчислювання часу:

Так як дерево Merkle є повним бінарним деревом, тому кількість вузлів на висоті h буде дорівнювати 2*h.

Висота дерева з n листя буде дорівнювати $log_2 n$.

Кількість внутрішніх вузлів розраховуються за формулою (2h-1).

Верхньою межею є O(β).

– складність пам’яті:

Використання пам’яті можна описати за значенням розміру ключа та розміру дерева.

$$2^k + 1 * \beta = 0(\beta * 2^k) \quad (2.2)$$

– складність повідомлень:

Ми маємо верхню межу O(h ß) для складності повідомлення.

2.2 Експериментальне середовище

Існує велика кількість різноманітних симуляторі та емуляторів мережі. Мережеві стимулятори використовують як системними адміністраторами при створенні нових мереж так і для моделювання та створення нових версій та розробок з використанням нових технологій. Використання емуляторів мережі дозволяє створювати нові мережі або модифікувати існуючі не ризикуючи існуючою мережею.

Існує два основних підходи: симуляція це комплекс програмних апаратних засобів або їх поєднання призначені для копіювання функції однієї обчислювальної системи на інший. Відмінність симуляції не ставить за мету точне відтворення поведінки однієї системи, а концентрується на відтворенні моделі системи і певних ключових особливостей або параметрів.

До можливостей емуляторів слід віднести створення різних мережевих топологій, створення нових каналів зв'язку з різними характеристиками, регуляція трафіку у віртуальній мережі. Використовуючи різні протоколи та надає можливість підключення різних мережевих пристроїв.

Натомість ключовою перевагою використання стимуляторів є можливість розглядати задачі які неможливо відтворити маючи наявне

обладнання.

Програми симулятори можна класифікувати за такими параметрами як низько функціональні, середньо функціональні та високо функціональні. За принципом роботи класифікуються: в реальному часі або дискретна-подієві Дискретно-подієві відрізняються від програм що працюють в реальному часі швидкістю моделювання яка першочергово залежить від потужності комп'ютера на якому відтворюються моделювання.

UNetLab – безкоштовний емулятор з допомогою якого можна створювати різноманітні інженерні рішення. UNetLab має зручний графічний інтерфейс який надає змогу додавати нові зображення, що будуть використовуватися на графічному стенді до прикладу з Microsoft Visio.

Емулятор підтримує значну кількість мережевих пристроїв як маршрутизатори так і пристрої безпеки та дозволяє використовувати деякі технології з хмарних рішень.

OMNeT++ – модульна бібліотека C++ першочергово створена для розробки мережевих стимуляторів.

Використовується для моделювання в науковому співтоваристві в промисловості, забезпечуючи свою роботу на різноманітних операційних системах Windows, Linux, Mac OS. Також за останні роки було добавлено велику кількість різноманітних інтернет протоколів що дозволяють зручно створювати власні модулі.

Cisco Packet Tracer – симулятор мережі що випускається компанією Cisco і надає змогу моделювати мережі, налаштувати пристрої, маршрутизатори комутатори та інші які випускає компанія Cisco. Симулятор дозволяє відтворювати налаштування пристроїв, що відбувається через термінал або командну строку. Містить велику кількість графічних інтерфейсів та діалогових вікон що значно полегшують доступ користувачів, а також дозволяють переглядати усі потрібні дані які є в Cisco Packet Tracer.

Використовує велику кількість пристроїв різного призначення та зв'язків

що дозволяє проектувати проекти будь-якого рівня .

GNS3 – графічний симулятор мережі який дозволяє створювати різні мережеві топології, що найчастіше використовується для перевірки технології або ж схему мережі, перед уведенням її в дію. Основною перевагою є те що симулятор підтримує найбільшу кількість різноманітних пристроїв від різних виробників мережевого обладнання, на відміну від Cisco Packet Tracer, який підтримує переважно тільки обладнання від компанії Cisco. Також GNS3 дозволяє використовувати необмежену кількість пристроїв, а основною перевагою є те що система при моделюванні дозволяє використовувати реальний комп'ютер який можна підключити через віртуальну машину і таким чином можна перевірити чи працює підключення до серверів автентифікації, перевіряти налаштування Firewall.

Відповідний функціонал симулятора потребу значних ресурсів. Про те якщо задачею є навчання, або ж побудова простих моделей, ресурси які потребує симулятор є незначними.

GNS3 є повністю у вільнім доступі а також має відкритий вихідний код що дозволяє вносити свої зміни при потребі.

Mininet – емулятор комп'ютерних мереж який дозволяє створювати віртуальні пристрої, такі як комутатори, контролювали маршрутизатори мережі. Основними перевагами Mininet є комплексне тестування топології, а також можливість одночасної розробки декількох спеціалістів однієї топології.

NS 3 – симулятор мережі з дискретними подіями націлений для проведення досліджень та освіту.

Основними перевагами і програмного забезпечення NS 3 є відкритий вихідний код, можливість симуляції великих комп'ютерних мереж, обробки трафіку різних мережевих пристроїв.

Вихідний код є відкритим, що дозволяє створювати свої моделі в NS 3

Був створений для проведення експериментів які важко виконувати в

реальному середовищі, NS 3 переважно використовується в таких операційних системах як Linux та Mac OS. NS 3 не має власного графічного інтерфейсу, але для відображення моделювання можна використовувати інші графічні інструменти. Для налаштування систем, а також для написання нових модулів необхідно використовувати мови програмування C++ або Python. Робота з симулятором відбувається через команду строку. NS 3 реалізований як набір відкритих бібліотек в яких зібрана велика кількість протоколів передачі інформації, які можна об'єднувати разом. Також підтримується об'єднання з іншими зовнішніми бібліотеками.

Для виконання поставленої задачі було досліджено функціонал вище описаного програмного забезпечення. Більшість симуляторів та емуляторів не надають змогу безпосередньо вносити зміни в існуючі протоколи. Або ж для внесення змін необхідно переписувати значну частину функціоналу пристрої, що використовують в моделюванні.

Враховуючи відсутність вибору протоколів автентифікації в пристроях у більшості симуляторів та емуляторів основним критерієм для вибору середовища моделювання, є можливість додавати нові функції.

Тим паче в моделюванні яке необхідно провести не має потреби у використанні існуючих пристрїв, а моделюється тільки алгоритм автентифікації.

Вибір середовище моделювання враховуючи всі потреби було обрано NS 3.

NS 3 не моделює окремі пристроїв системи проте про те NS 3 дозволяє створювати окремі вузли в системі які можуть спілкуватися між собою через найпоширеніші протоколи до прикладу tcp або udp, також мережевий симулятор надає змогу створювати меш мережі. NS 3 містить велику кількість уже написаних бібліотек до прикладу модулі wi-fi протоколи udp tcp різноманітні топології мереж та навіть при потребі надає змогу працювати з енергетичними моделями пристрою.

Більше частина бібліотека написана на мові C++ тому найкращим вибором для написання нових модулів є C++ хоча також додано можливість використовувати мову Python. Проте на вибір мови слід брати до уваги наявність стандартних бібліотек які будуть використовуватись при написанні модуля. Більшість написаний бібліотек представляють одну із моделей або з технологій до прикладу є написані модулі які відповідають за надсилання пакетів або ж модуль application який відповідає за виконання тих чи інших функцій які будуть застосовані у вузлах.

Перевагою використання симулятора мережі NS 3 є те що він в поставляється з низкою різних типів мереж які вже є налаштовані. Зокрема налаштований меш мережі. Проте в існуючих мережам не передбачено жодних функцій для автентифікації вузлів як і в більшості інших симуляторів

Тому функцію автентифікації необхідно створити та модифікувати існуючу меш мережу для потреб симуляції.

Приклад найпростішої існуючої мережі, що було передбачено в мережевім симуляторі NS 3, (див. рис. 2.2).

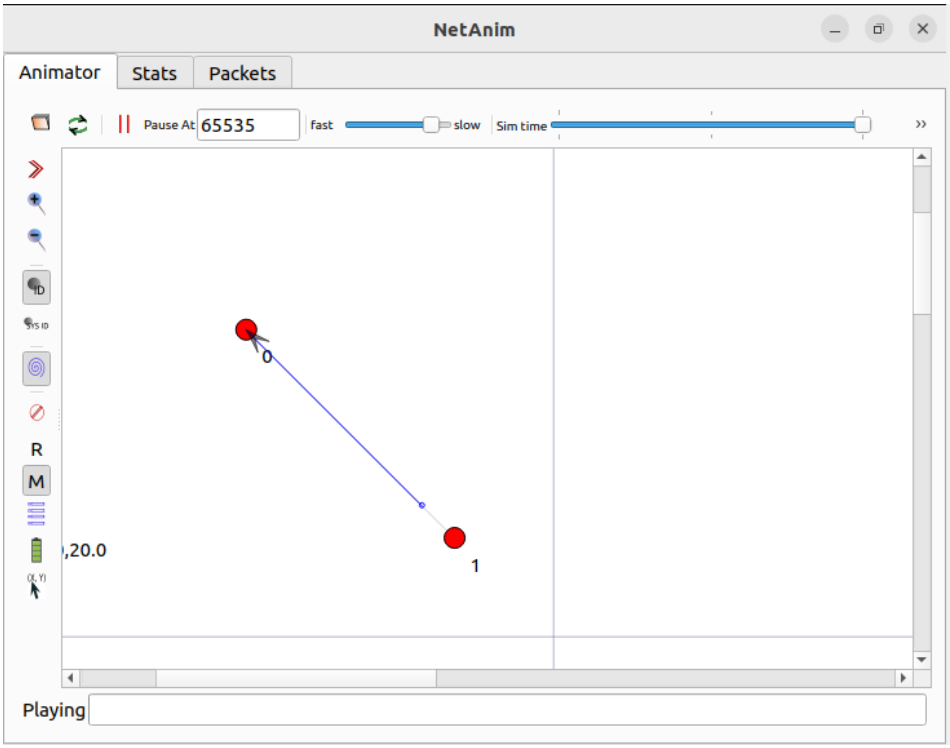

Рисунок 2.1 – Приклад існуючої мережі в NS 3

У NS 3 основна абстракція обчислювального пристрою називається вузлом. Ця абстракція представлена в C++ класом Node. Клас Node надає методи для керування представлення обчислювальних пристроїв у моделюванні.

Базовий вузол в симуляторі NS 3 можна представити як оболонку комп'ютера до якої можна додати внутрішні компоненти включаючи різноманітні протоколи та програми. Створюючи вузол він має порожню оболонку в якому є унікальний ідентифікатор в системі, що використовується для розподілення під час моделювання.

У NS 3 основною абстракцією для програми користувача, яка генерує певну діяльність для моделювання, є Application. Ця абстракція представлена в C++ класом Application. Клас Application надає методи для керування представленнями нашої версії програм рівня користувача в симуляції.

В результаті виконаного дослідження для реалізації поставленої задачі було вибрано мережевий симулятор NS 3. Розглянуто існуючі реалізовані мережі в симуляторі. Вибрано існуючу меш мережу. При оцінці мережевого симулятора було поставлено задачі які необхідно виконати для проведення моделювання, яке визначає швидкість автентифікації використовуючи алгоритми RSA та дерев Merkle.

2.3 Інформаційна модель в середовищі ns-3

В мережевім симуляторі NS 3, є створеною модель меш мережі розміром в 9 пристроїв, концептуальна модель зображена на рисунку 2.2.

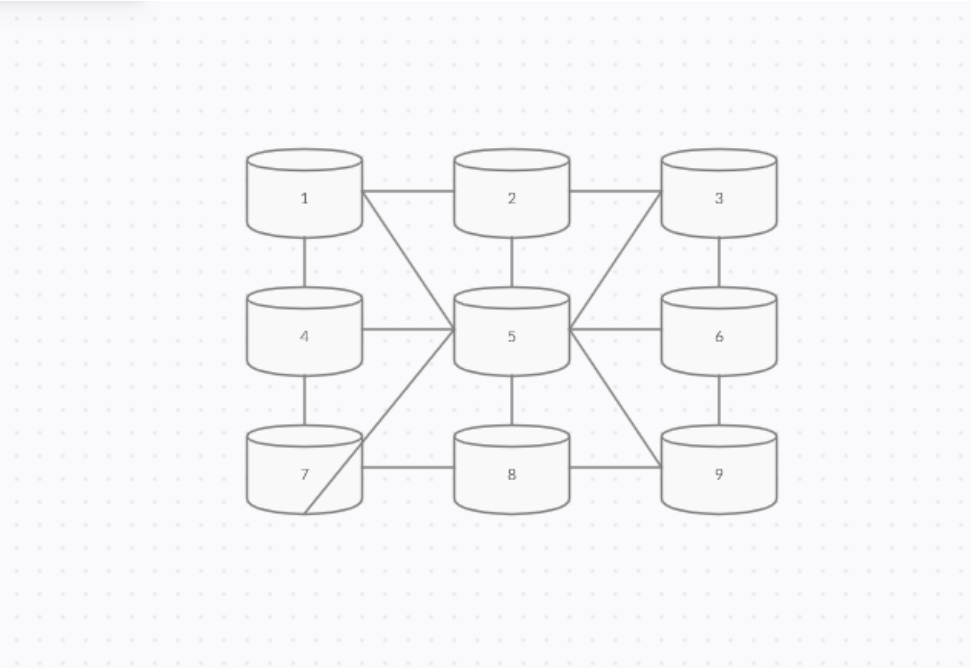

Рисунок 2.2 – Концептуальна модель меш мережі

З моделі видно, що кожен із пристроїв об’єднаний з сусіднім.

Враховуючи розмір моделі розраховуємо, що дерево Merkle має розмір глибинною в 2 з 4 листками, які можна використовувати для автентифікації вузлів. Модель дерева Merkle, зображено на рисунку 2.3

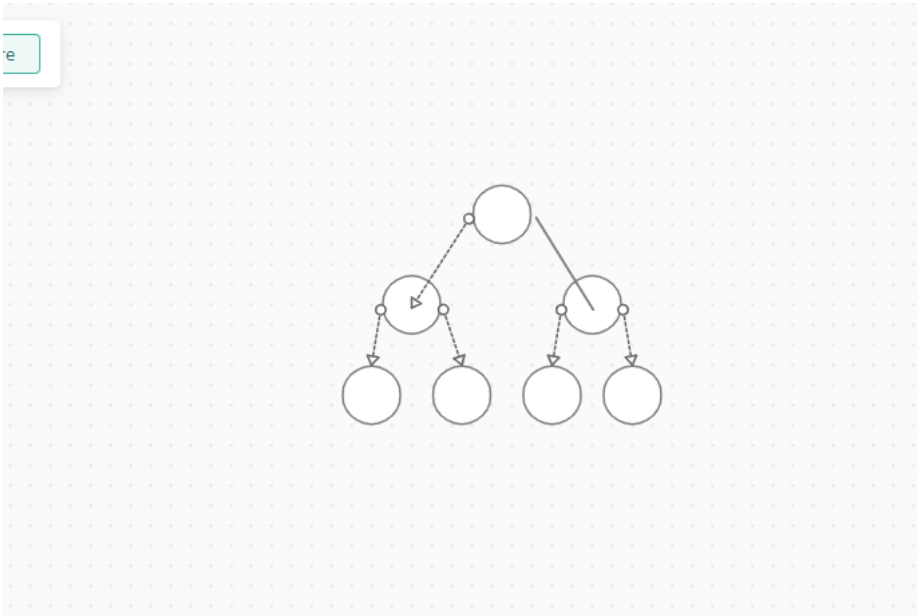

Рисунок 2.3 – Представлення дерева Merkle

Сьогодні в сфері промисловості ми починаємо спостерігати використання WMN для систем керування освітленням. Зараз ці мережі обмежені окремими ділянками будівель, а не цілими будівлями, і часто обмежуються не більше ніж 64 пристроями. З цієї причини ми визначили цей експеримент як мережу з 64 вузлів. Мережі з 64 вузлів має бути достатньо для проведення експерименту, щоб отримати чітку різницю в часі виконання завдань.

РОЗДІЛ 3. МОДЕЛЮВАННЯ ПРОГРАМНОГО ЗАХИСТУ ІНФОРМАЦІЇ НА ОБ'ЄКТАХ КРИТИЧНОЇ ІНФРАСТРУКТУРИ

Моделювання в симуляторі ns-3 не дозволяє задіювати моделі існуючих пристроїв, а саме тому важливо перед проведенням моделювання звернути увагу на основі загрози які були знайдені для пристроїв, що уже використовуються в Smart Grid.

Під час симуляції ми оцінюємо тільки швидкодію роботи алгоритмів, проте в реальному пристрої необхідно враховувати навіть дані, що генеруються.

Інтелектуальні лічильники є частиною так званої розширеної інфраструктури вимірювання (AMI) Smart Grid і на даний момент вони вже розгорнуті в багатьох країнах. У 2010 році були виявлені недоліки безпеки в наборі радіочіпів, який використовується в деяких інтелектуальних лічильниках. Це забезпечує потребу у реалізації прикладного дослідження вразливостей, з якими зіткнеться Smart Grid.

Розумні лічильники, як правило, бездротові, і в цьому випадку вони використовували стандарт ZigBee для свого зв'язку. Недоліки були виявлені в радіочіпсеті Texas Instruments CC2530. CC2530 — це система на чіпі з тактовою частотою 2,4 ГГц, оснащена радіочастотним трансивером, сумісним зі стандартом IEEE 802.15.4, який підходить для програм ZigBee. CC2530 транслює в діапазоні частот 2400-2483,5, який використовується в усьому світі і має до 16 каналів. Частота 2,4 ГГц не є рідкістю для багатьох бездротових пристроїв, таких як мережі WiFi, бездротові телефони, пристрої Bluetooth тощо. Проблеми виникли в CC2530 Z-Stack (стек протоколів ZigBee) версії 2.2.2-2.3.0. Стан специфікації CC2530: Генератор випадкових чисел використовує 16-бітний LFSR для генерації псевдовипадкового числа, яке може зчитуватися ЦП або використовуватися безпосередньо строб-процесором команд. Він може бути засіяний випадковими даними з шуму в радіо АЦП.

Першою проблемою було використання 16-розрядного регістра зсуву з лінійним зворотним зв'язком (linear feedback shift register - LFSR) для генерації випадкових чисел. З 16-бітами існує 2^{16} або 65531 внутрішніх можливих станів, тобто випадкових чисел. Ці випадкові числа використовувалися для шифрування даних. Вимога до криптографічно випадкових чисел полягає в тому, що вони повинні бути статистично випадковими та непередбачуваними. З 65531 внутрішніми станами це не відповідає вимогам, необхідним для підтримки безпечної системи.

Саме генерація випадкових чисел є надважливою загрозою в пристроях Smart Grid, так обмеженість в ресурсах не дозволяє створювати велику множину унікальних значень, що будуть використовуватися у ключах.

Компанія Texas Instruments виправила недолік 16-бітного LFSR у версії 2.3.0 мікропрограми Z-stack. Незрозуміло, скільки лічильників увійшло в поле з цим недоліком і скільки з них було виправлено. Pacific Gas & Electric повідомляє, що тільки в їхньому домені встановлено 5 мільйонів розумних лічильників. Розумні лічильники не обійшлися без суперечок. Здебільшого це сталося через вищі рахунки за електроенергію, занепокоєння радіочастотними сигналами, але це також включало ідею конфіденційності. Тепер будуть доступні дані, які фіксуватимуть дуже детальні та особисті звіти про спосіб життя, звички та іншу інформацію, якою можуть бути зацікавлені зловмисники.

Проте однією з найсерйозніших перешкод для витоку інформації є створення надійною системи автентифікації.

3.1 Параметри експерименту

Експеримент проводився на Acer Aspire під керування операційної ситеми Ubuntu 22.10, Inter Core i5 2.3 ГГц і 8 ГБ пам’яті DDR4 2133 МГц.

В ns-3 надається певна кількість стандартних моделей які уже є готовими для використання.

Проте мережа mesh, що надається в ns-3, потребувала модифікацій, щоб імітувати процедуру автентифікації. Існуюча модель мережі не передбачає використання автентифікації та використання криптографічних протоколів.

Стандартна модель меш мережі, що надається в ns-3 є зображеною на рисунку 3.1

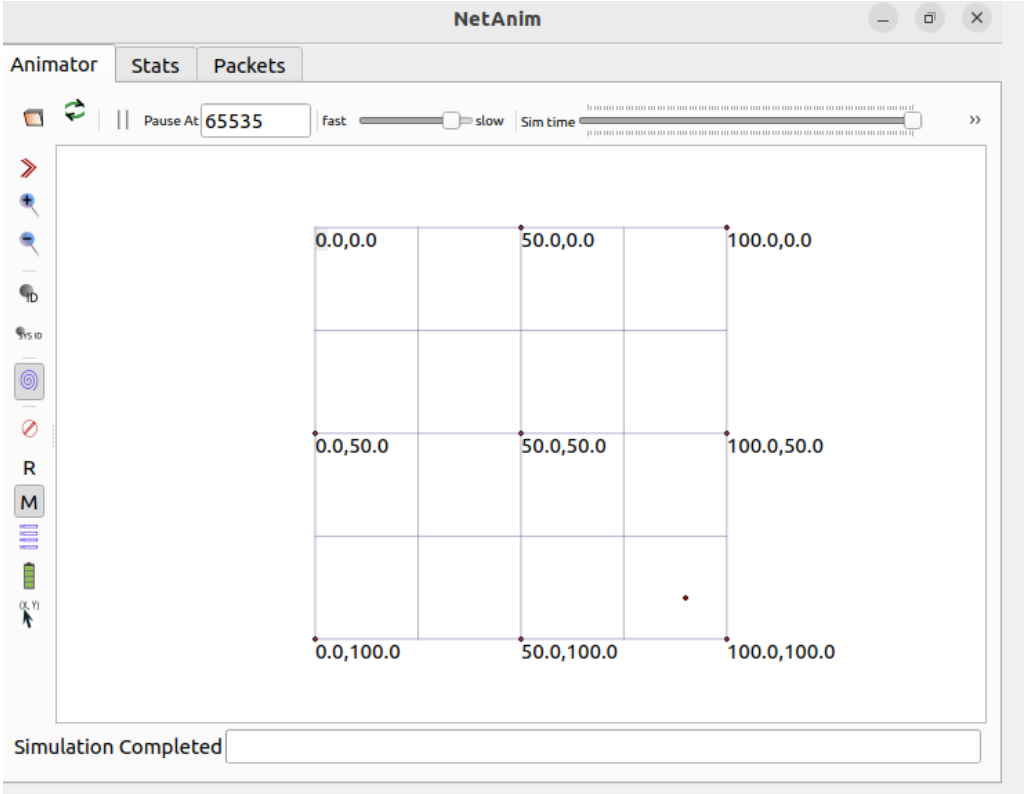

Рисунок 3.1 – Модель стандартної меш мережі

Меш мережа, що надається в ns-3 містить меш мережу розміром 3 на 3, що складається з 9 вузлів, які імітують передачу пакетів. Кожен вузол меш мережі для спілкується між своїми сусідами. Відповідно кожен вузол має трафік із 4 сусідами, що знаходяться найближче.

Таким чином стандартна модель відповідає потребам для реалізації моделювання функції автентифікації. Проте автентифікація при збільшенні меш мережі до 64 пристроїв потребує збільшення кількості сусідніх елементів які будуть брати участь в автентифікації.

Обмеження кількості вузлів в 64 пристрої спричинено актуальністю використання меш мереж в промисловості саме такою кількістю пристроїв.

Аналізуючи Smart Grid було виявлено, що меш мережа переважно має покривати домогосподарства, невелику вулицю, приватний житловий будинок, або ж навіть окремі торгові точки, що знаходяться в однім торговім центрі, в залежності від ефективності, що оцінюється.

Таким чином меш мережі є розділеними та спілкування між ними відбувається з допомогою інших пристроїв. Тому використання при моделювання меш мережі розміром в 64 вузла є доцільним.

Найоптимальнішим варіантом є збільшення пристроїв, що спілкуються для автентифікації вузла в меш мережі до 16 сусідніх вузлів. Тому для виконання моделювання необхідно збільшити крок за яким вузол визначає чи елементи є сусідніми чи ні.

Для моделювання використовується розмір повідомлення в 32 біт. Така обмежена довжина ключа спричинена обмеженими технічними ресурсами, проте при масштабуванні тестування до довжини ключа, що вважається безпечним, а саме від 1024 до 2048 біт, тенденція отриманих результатів повинна залишатися незмінною. Ключі розміром більше 2048 біт, хоч і є безпечнішими через обмеженість ресурсів вузлів меш мереж не доцільні у використанні. При збільшенні ж технічних характеристик меш мережі, значно зросте і вартість на її реалізацію.

Також у виконані роботи необхідно звернути увагу на швидкість існуючої моделі для більш вдалого порівняння швидкодії функцій автентифікації.

Тестування існуючої моделі дозволить краще зрозуміти на скільки сильно розмір мережі з використанням автентифікації впливає на швидкодію роботи.

А також наддасть данні про відхилення під час моделювання, що в свою чергу дозволяє визначати допустимі межі відхилення при моделюванні функцій автентивікації.

Так як результат моделювання не є однаковим необхідно провести тестування існуючі меш мережі для визначення відсотку відхилення. Для кращої вибірки тестування проводиться п’ять разів, що зображено в таблиці 3.1.

Також слід уточнити, що перед моделюванням для тесту було зроблено декілька тестів, щоб дані, що постійно використовуються програмою для моделювання були завантажені в кеш пам’ять. В іншому випадку ж результати суттєво відрізняються від кількості проведених тестувань.

Похибка в такім випадку становить більше десяти відсотків, що є критичним для проведення тестування подібних моделей.

Пристрої, що використовуються в меш мережах, зазвичай є обмеженими в ресурсах. Найчастіше використовують до 1 ГБ оперативної пам’яті та процесори з частотою близько 1 ГГц. Звісно для пристроїв, що плануються використовуватися для Smart Grid, будуть вноситися зміни із врахуванням потреб. Так як в даний момент усі існуючі мережі Smart Grid, перебувають у стані безперервної модифікації та оцінці ефективності роботи.

Таблиця 3.1 – Результати тестування меш мережі розміром в 9 вузлів без проведення автентифікації

Значення в мілісекундах	Експери мент 1	Експери мент 2	Експерим ент 3	Експер имент 4	Експери мент 5	Середнє значення
Експеримент и	796.279 мл	742.066 мл	746.34 мл	860.949 мл	802.47 мл	789.62 мл
Відхилення	6.659 мл	47.554 мл	43.28 мл	71.329 мл	12.85 мл	36.334 мл

Продовження таблиці 3.1

Відсоток відхилення	Відсоток середнього відхилення дорівнює 4,6 %

При проведенні моделювання було проведено 5 експериментів, та розраховано середнє відхилення.

Середнє відхилення дорівнює 4,6 відсотків. Такий розмір відхилення результатів уже є значним, про те ще дозволяє використовувати модель для тестування функцій автентифікації.

Графік середньої швидкості взаємодії вузлів в меш мережі розміром 3 на 3 тобто в 9 вузлів, зображено на рисунку 3.2.

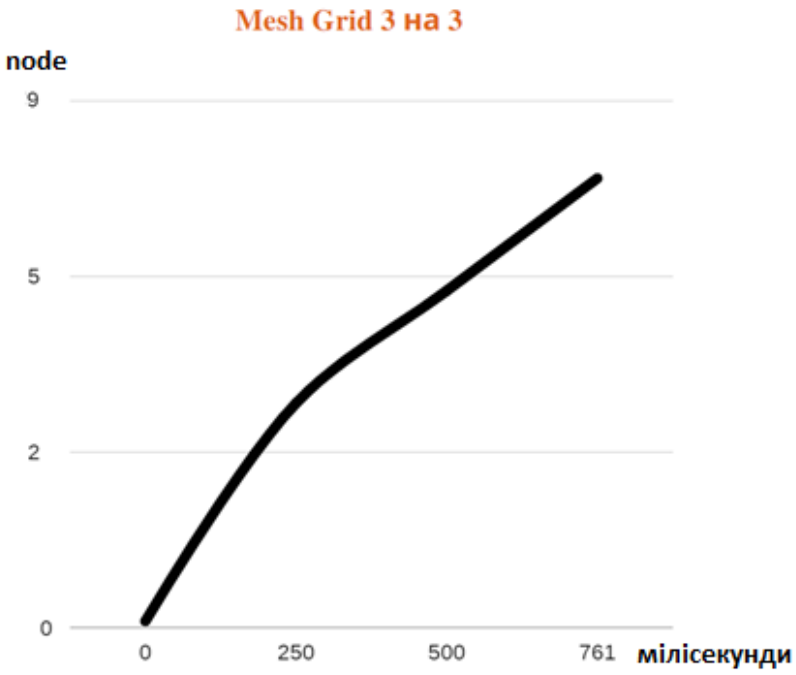

Рисунок 3.2 – Час взаємодії вузлів

Отримані дані, щодо часу дозволяють оцінити швидкість спілкування пристроїв меш мережі між собою.

Для проведення експерименту було розширено існуючу меш мережу до 64 вузлів. Вузли, є статичними, тому що пристрої Smart Grid, теж є статичними.

Кожен пристрій спілкується водночас тільки з одним сусідом. Це зроблено тому що невідомо які технічні характеристики має вузол меш мережі. Та чи має змогу він спілкуватися з кількома пристроями в меш мережі водночас.

Звісно використання двох каналів зв’язку значно б пришвидшило швидкодію мережі.

Також збільшено довжину кроку, для того щоб автентичність вузла підтверджували інші шістнадцять вузлів. Таким способом досягається найбільша безпека від загрози 51%.

До атаки 51 % вразливі всі технології, в яких використовується алгоритм доказу виконання роботи PoW.

Tree Merkle саме і дозволяє заощаджувати пам’ять пристроїв та виконувати доказ роботи Pow.

Так як зловмисник може отримати доступ безпосередньо до самого вузла, то необхідно буде також реалізовувати фізичних пристрою. До прикладу як міцний корпус, до якого неможливо підключитися не маючи відповідного дозволу. Захисний кожух використовується в багатьох пристроях, найпростішим прикладом є лічильники, що знаходяться в кожного вдома.

Проте щоб захистити меш мережу від підключення пристрою в середині системи, та модифікації, або ж порушення конфіденційності якраз і використовується збільшення пристроїв, що необхідні вузлу для автентифікації.

Вважається, що функція автентифікації та фізичних захист пристроїв, повинен ускладнити задачу зловмиснику і не допустити можливість підміни великої кількості пристроїв меш мережі.

Результат модифікації існуючої моделі зображено на рисунку 3.3.

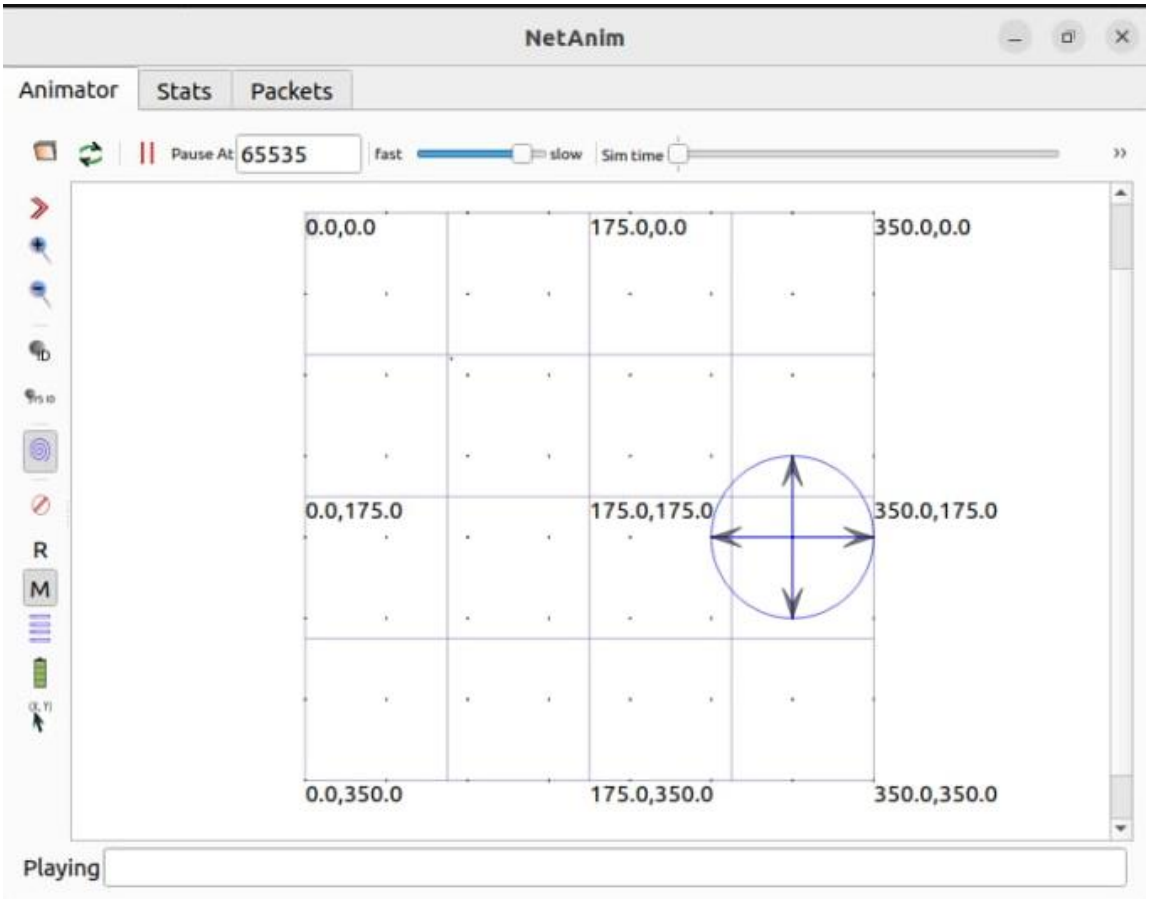

Рисунок 3.3 – Меш мережа розміром в 64 вузла

Відображення на рисунку 3.3 дозволяє побачити пошук сусідніх вузлів, що будуть використовуватися для автентифікації.

В залежності від розміру існуючої мережі для її безпеки можна використовувати різну кількість вузлів для автентифікації. Враховуючи, що переважно меш мережі можуть набувати різних розмірів.

До прикладу меш мережа складається з 16 вузлів і кожен вузол є добре захищеним фізично, а також немає можливості прослуховувати мережу. Такі мережі можуть використовуватися на підприємствах які містять периметр безпеки, але при цім можуть потребувати значно частішого оновлення даних.

За таких умов можна обмежитися автентифікацією із чотирьома сусідніми пристроями.

Створення вузлів в меш мережі та їх ключових параметрів таких як ip адреса, mac адреса та порядковий номер, зображено на рисунку 3.4.

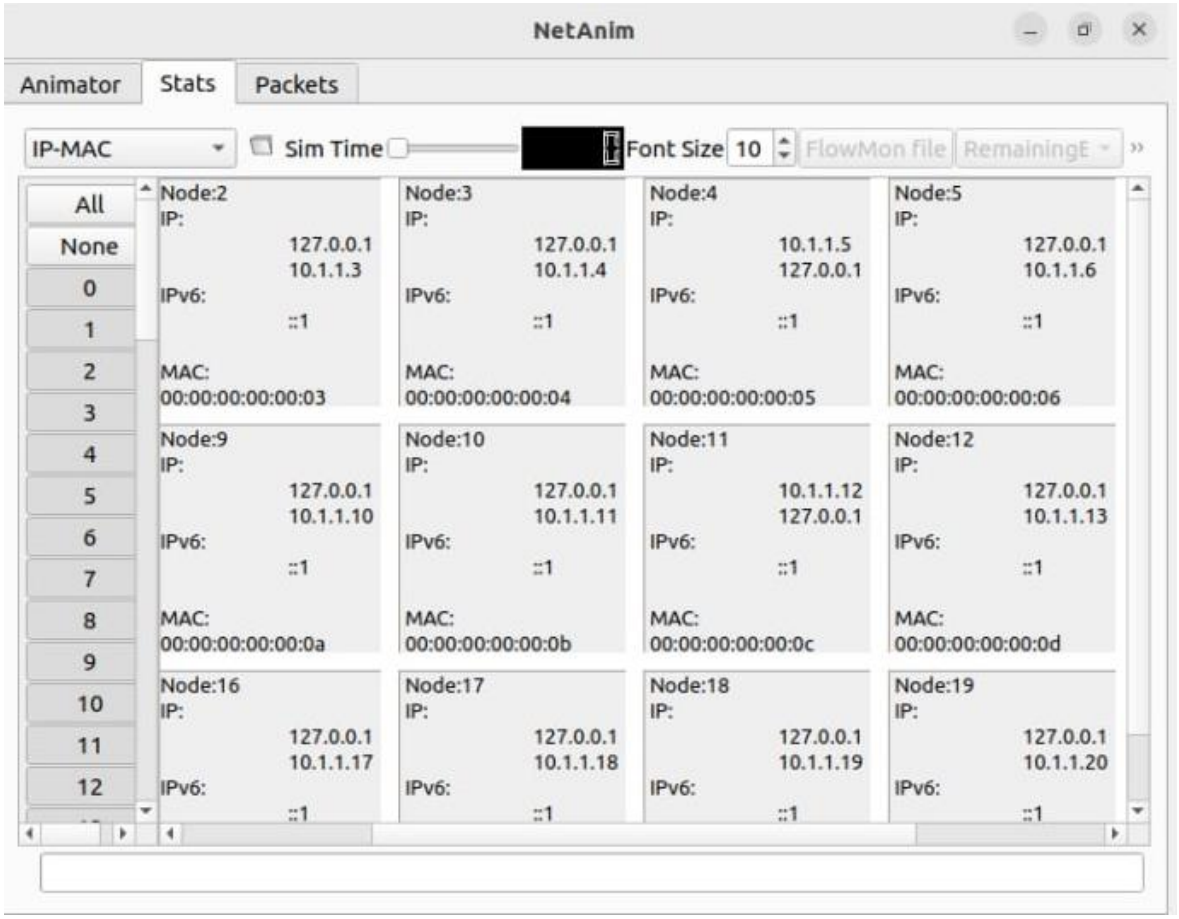

Рисунок 3.4 – Таблиця вузлів в меш мережі

З наведених даних на рисунку 3.4 видно, що вузли меш мережі є правильно сформованими. Та діють в одному діапазоні IP адрес.

Таким чином вдосконалена модель є готовою для імітації використання протоколів автентифікації використовуючи RSA та Merkle Tree.

Моделювання моделі з включеними протоколами відбувається послідовно, а не водночас, тому що при моделюванні може виконуватися тільки одна симуляція.

Враховуючи можливі відхилення при тестуванні базової моделі, було проведено серію тестувань для визначення середнього значення, та відсотку відхилення в кожній з моделей. Прогнозуючи швидкодію алгоритмів та їх

різницю відхилення до п’яти відсотків є допустимими.

3.2 Модель захисту з використанням дерева Меркла

Для хешу була використана функція hash(), доступна з tr1/функціональною бібліотекою C++.

Була додана нова функція для вузлів, яка при створені нового вузла, створює нове дерево Merkle, яке зберігається у вузлі. Пізніше, коли вузли об’єднуються в мережу, ми додаємо функцію автентифікації вузлів.

Tree Merkle з 16 листками (глибиною 4) було б достатньо. Якщо ми припустимо мережу з 64 пристроїв, то якщо кожен вузол може автентифікуватися з 16 іншими вузлами навколо нього, цього має бути достатньо для створення надійної системи. Крім того, оскільки велике дерево споживає більше ресурсів, є перевагою мати маленьке дерево.

При моделюванні враховуємо час генерації та час автентифікації разом.

Тестування проводилося п’ять разів. В середньому генерація та автентифікація триває близько 20000 мілісекунд.

Враховуючи, що довжина ключа тільки 32 біта, то час автентифікації можна тривалим. Проте те потрібно враховувати, що кожен вузол підтверджує автентичність з іншими шістнадцятьма вузлами. Тобто під час симуляції відбувається 1024 автентифікацій. Для мереж Smart Grid, враховуючи навіть збільшення довжини ключа, результат є цілком допустимим, так як вузли мережі передаватимуть дані не постійно, а з певним інтервалом.

Середнє відхилення під час тестування моделі складає 4.1 %.

Враховуючи що тестування стандартної моделі без функцій автентифікації складає 4.6 %. Тому середнє відхилення не перевищує допустимої межі.

Результат проведеного тестування та визначення середнього відхилення зображено в таблиці 3.2.

Таблиця 3.2 – Результати тестування меш мережі розміром в 64 вузла з проведення автентифікації Tree Merkle

Значення в мілісекундах	Експеримент 1	Експеримент 2	Експеримент 3	Експеримент 4	Експеримент 5	Середнє значення
Експерименти	19206.7 мл	18897.5 мл	18907.3 мл	20189.89 мл	21218.4 мл	19683.9 мл
Відхилення	477.258 мл	786.458 мл	776.658 мл	505.932 мл	1534.442 мл	816.149 мл
Відсоток відхилення	Відсоток середнього відхилення дорівнює 4,1 %					

Швидкодія проведення автентифікації використовуючи алгоритм Tree Merkle, зображена на рисунку 3.5.

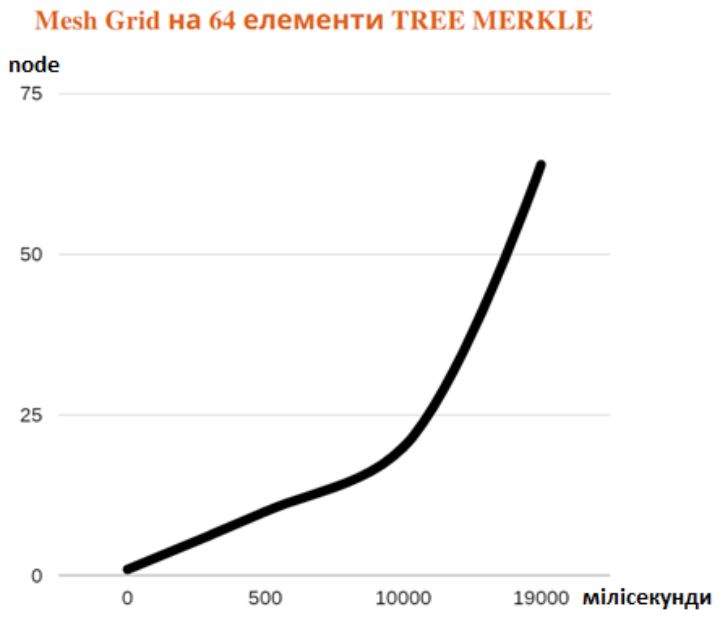

Рисунок 3.5 – Час автентифікації вузлів

В результаті виконання тестування було розраховано похибку та середній час автентифікації Tree Merkle, з довжиною ключа в 32 біт.

Використання Tree Merkle, показує задовільний результат швидкодії, про проблемою використання цього алгоритму є необхідність мати визначену кількість вузлів в меш мережі.

Слід додати, що при використанні більшого ключа та алгоритму хешування, час реалізації автентифікації може суттєво зрости. Про те навіть при зростанні у два рази, алгоритм Tree Merkle, залишається значно швидшим, та менш ресурсоємним.

3.3 Модель захисту з використанням RSA

Модель RSA будується за схожою схемою, що і Tree Merkle, функції реалізовуються також у вузлах. Однак замість дерева в кожному вузлі RSA зберігає закритий ключ. Відкритий ключ немає необхідності зберігати його у вузлі.

RSA може використовувати пару відкритий та приватний ключ для автентифікації з будь-якою кількістю інших вузлів. Це є найбільшою перевагою в порівнянні з Tree Merkle. Саме з допомогою RSA є можливість розбудовувати меш мережі в Smart Grid при потребі.

Також тестування проводилося п’ять разів.

В середньому автентифікація відбувається, за 56670 мілісекунд, такий час автентифікації, з урахування того факту, що розмір ключа є всього 32 біт, є значно гіршим чим в Tree Merkle. Навіть з урахуванням, що автентифікація відбувається в загальному 1024 рази, результати є не задовільними.

Відсоток відхилення становить 1.2%, що не виходить за межі відхилення стандартної моделі, відповідно тестування можна вважати правильними.

Результати тестування зображено в таблиці 3.3.

Якщо збільшити довжину ключа, процес автентифікації буде тривати значно довше. Та для того щоб прискорити в декілька раз процес автентифікації з використанням алгоритму RSA, можна за допомогою одночасного спілкування вузлів між собою.

Таблиця 3.3 – Результати тестування меш мережі розміром в 64 вузла з проведення автентифікації RSA

Значення в мілісекундах	Експеримент 1	Експеримент 2	Експеримент 3	Експеримент 4	Експеримент 5	Середнє значення
Експерименти	56691.2 мл	56721.3 мл	57618.5 мл	57756.7 мл	55364.4 мл	56670.42 мл
Відхилення	20.78 мл	50.88 мл	948.08 мл	1086.28 мл	1306.02 мл	682.408 мл
Відсоток відхилення	Відсоток середнього відхилення дорівнює 1,2 %					

Також існують різні реалізації алгоритму RSA, прикладом є реалізація на основі еліптичних кривих.

Проте досягнути значно кращих результатів по швидкодію навіть з усіма покращеннями не можливо. Враховуючи перевагу алгоритму у відсутності фіксованої кількості пристроїв в меш мережі. Можна допустити використання алгоритму в невеликих меш мережах, які будуть мати захищену зону та обмежену кількість пристроїв.

За таких умов довжина ключа, та кількість вузлів які необхідні для верифікації одного вузла буде зменшено, а відповідно час на виконання операції автентифікації буде прийнятним.

Графік автентифікації вузлів в меш мережі за допомогою алогоритму RSA, зображено на рисунку 3.6.

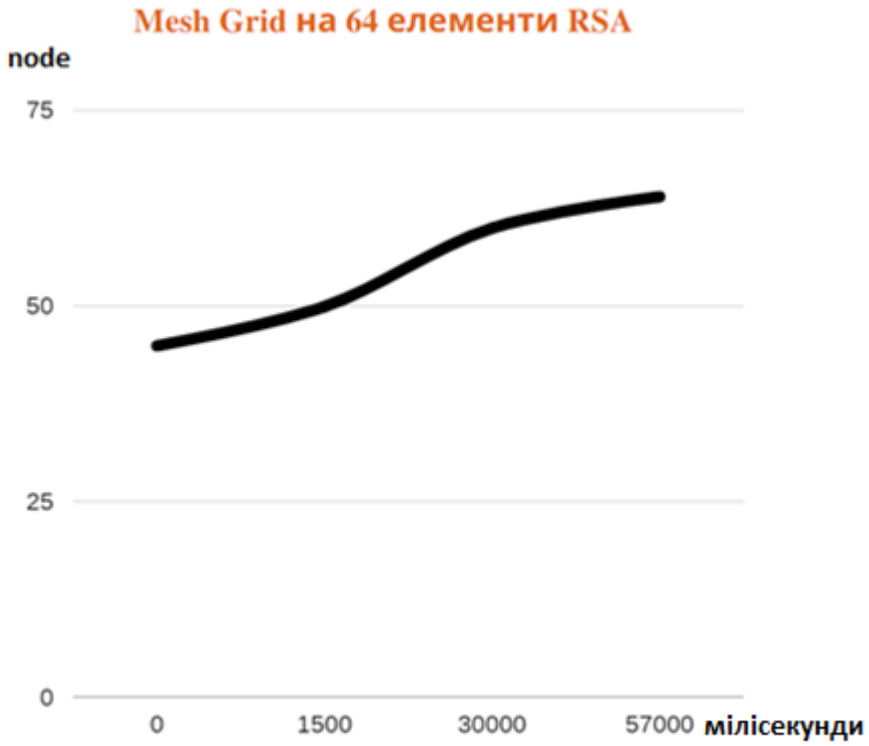

Рисунок 3.6 - Час автентифікації вузлів

Також слід не забувати, що при умові значного розвитку квантових технологій, криптографічні алгоритми засновані на складності факторизації стають повністю вразливими.

Smart Grid є модифікацією електричної інфраструктури, а відповідно технології, що використовуються повинні бути надійними, довговічними та захищеними від уже існуючих так і від майбутніх загроз.

Перехоплення даних, що пов’язані з електромережами можуть мати негативні наслідки не тільки для окремої людини, а для всього механізму.

Тому використання RSA алгоритму є допустим для автентифікації в меш мережах, про те не для пристроїв Smart Grid.

Над розробкою квантовостійких криптографічних алгоритмів ведеться активна робота, тому цілком можна припустити, що такі алгоритми як RSA, протягом наступних років будуть модифікуватися або відходити у минуле.

3.4 Результати експерименту

Наші перші тести показали, що схема Merkle була набагато швидшою за схему RSA. Очевидно, якби Merkle використовував SHA-1 або іншу криптографічну хеш-функцію, це трохи сповільнилося б, але все одно мало б легко перемогти RSA. Однак, якщо ми вважаємо, що RSA не має обмежень щодо кількості пристроїв, які він може автентифікувати, тоді RSA має перевагу перед деревом Merkle у цьому сенсі.

До переваг алгоритму RSA слід віднести:

– необмежена кількість пристроїв, що може використовуватися в Smart Grid;

– зберігання в кожному пристрої тільки одного ключа, ефективніше використання пам’яті пристрою.

До недоліків алгоритму RSA слід віднести:

– низька швидкість автентифікації;

– із збільшення довжини ключа швидкість автентифікації потребує великої кількості ресурсів;

– криптосистема є нестійкої проти загрози квантових обчислень.

До переваг Tree Merkle слід віднести:

– висока швидкість автентифікації;

– стійкість проти загрози квантових обчислень;

До недоліків Tree Merkle слід виділити:

– обмежена кількість пристроїв в меш мережі;

– значно більша потреба в об’ємі пам’яті пристрою в порівнянні з RSA;

– пам’яті потрібно кратно більше із кожним пристроєм, що потребує новий рівень. Проте це також є перевагою, так як для дрібних меш мереж немає потреби будувати велике дерево. А стандартний розмір меш мереж на даним момент становить 64 вузли.

Результати проведених моделювань зображено в таблиці 3.4

Порівнюючи результати прослідковується статистика за якою алгоритм Tree Merkle є майже у три рази швидшим за алгоритм RSA.

Використання автентифікації в найкращім в декілька разів сповільнює зв’язок між вузлами меш мережі.

Таблиця 3.4 – Порівняльні характеристики моделювання

Значення в мілісекундах	1 ЕКСПЕРЕМЕНТ	2 ЕКСПЕРЕМЕНТ	3 ЕКСПЕРЕМЕНТ	Середнє значення
MESH GRID 3 x 3	796.279 мл	742.066 мл	746.34 мл	761.561 мл
TREE MERKLE	19206.7 мл	18897.5 мл	18907.3 мл	19003.8 мл
RSA	56691.2 мл	56721.3 мл	57618.5 мл	57010.3 мл

Середнє відхилення під час проведення дослідження не перебільшує 5%, а отримані результати є достатньо значними, що дозволяє не брати до уваги присутні відхилення. Відхилення у кожній з досліджених моделей зображено в таблиці 3.5.

Таблиця 3.5 – Середнє відхилення

Моделі	MESH GRID 3 x 3	TREE MERKLE	RSA
Відхилення	4.6 %	4.1 %	1.2 %

Графічне порівняння швидкодії RSA та Tree Merkle алгоритмів, зображено на рисунку.

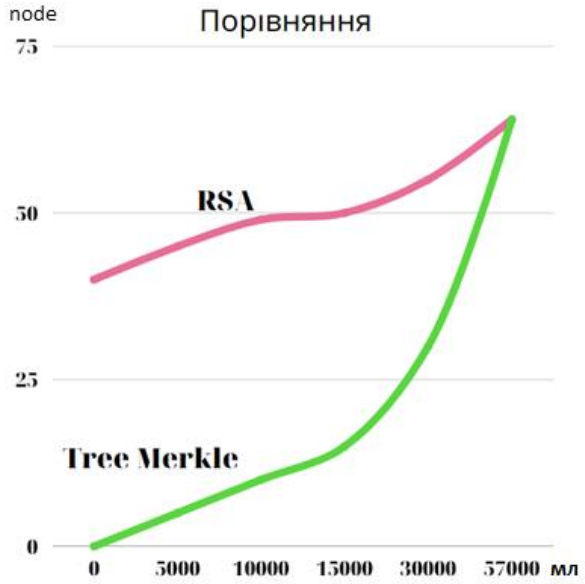

Рисунок 3.7 – Порівняння часу виконання алгоритмів RSA та Tree Merkle

З отриманого графіку видно, що швидкість Tree Merkle є значно більшою.

Відповідно для автентифікації пристроїв в Smart Grid, доцільним буде використовувати Tree Merkle.

Незначна потреба в ресурсах. Для пристроїв, що оперують даними електромереж пам’ять яка потрібна для збереження дерева теж є не значною, а також стійкість до квантових обчислень роблять Tree Merkle, перспективним алгоритмом в Smart Grid.

РОЗДІЛ 4. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Охорона праці

Тема кваліфікаційної роботи магістра присвячена розробці моделей захисту для меш мережі в Smart Grid, що є частиною критичної енерго інфраструктури. Враховуючи що технологія Smart Grid призначена для ефективного розподілу електроенергії між користувачами, то тема передбачає роботу з електронними приладами, то важливим є дотримання вимог з охорони праці і техніки безпеки.

Наведено нижче правила необхідно застосувати під модифікації електромереж.

Проаналізуємо основні правила і норми які необхідно дотримуватися при експлуатації електронних приладів та роботі з електромережами.

Працівник в процесі роботи зобов`язаний проходити:

- повторні інструктажі - не рідше одного разу в квартал;
- перевірку знань інструкції з охорони праці та діючої інструкції з надання першої допомоги потерпілим від нещасних випадків на виробництві один раз на рік;
- медичний огляд - один раз в два роки;
- перевірку знань правил для працівників, що мають право підготовки робочого місця, допуску, право бути виконавцем робіт, наглядачами або членом бригади, - один раз на рік.

При порушенні Правил охорони праці, в залежності від характеру порушень, проводиться позаплановий інструктаж або позачергова перевірка знань.

Виконавець робіт, зайнятий випробуваннями електрообладнання, а також працівники, які проводять випробування одноосібно з використанням стаціонарних випробувальних установок, повинні пройти місячне стажування

під контролем досвідченого працівника. Кожен працівник повинен знати, де перебуває аптечки і вміти нею користуватися. Працівник, який бере участь в проведенні вимірювань і випробувань електрообладнання, повинен працювати в спецодязі і застосовувати засоби захисту, що видаються відповідно до діючих галузевими нормами.

Для забезпечення захисту від ураження при випадковому дотику до струмоведучих частин діючої електроустановки або частин, що знаходяться під вимірювальним або випробувальним напругою, необхідні наступні способи і засоби захисту:

- захисні огородження;
- безпечне розташування струмоведучих частин;
- захисне відключення;
- ізоляція струмоведучих частин;
- ізоляція робочого місця;
- попереджувальна сигналізація, блокування, знаки безпеки.

Для забезпечення безпеки робіт при вимірах і випробуваннях зі зняттям напруги в електроустановці слід виконувати:

- відключення електроустановки від джерела живлення;
- механічне замикання приводів комутаційних апаратів;
- зняття запобіжників;
- від`єднання кінців живильних ліній та інші заходи, що виключають можливість помилкової подачі напруги на робоче місце;
- перевірку відсутності напруги;
- заземлення відключених струмопровідних частин (накладення переносних заземлень, ввімкнені заземлювальні ножі);
- огородження робочого місця або залишаються під напругою струмоведучих частин, до яких в процесі роботи можна доторкнутися або наблизитися на неприпустиме відстань.

Для іншого персоналу який займається налаштуванням пристрої,

працює з комп’ютерними системами теж необхідно проходити інструктаж.

У своїй роботі працівники повинні використовувати тільки ті електроприлади, до яких вони допущені посадовими інструкціями. При користуванні складними електроприладами дотримуватись спеціальних інструкцій. Перед увімкненням електроприладу необхідно перевіряти справність розетки мережі, а також, вилку і мережевий шнур, чи не порушена ізоляція. Уникати перегрівання, переохолодження, а також попадання вологи та пилу всередину електроприладу. Не загороджувати вентиляційні отвори, вони потрібні для запобігання перегріванню.

Працівникам що не мають відповідного дозволу забороняється:

— торкатися до клем та електропроводів, до розеток мережі, до арматури освітлення;

— відкривати електрощити;

— розбирати та робити самостійно ремонт (самого устаткування, дротів і т. ін.);

— самостійно проводити ремонт мереж, електророзеток, вимикачів.

Дотримання правил безпеки при користуванні електроприладами, а також при правильні експлуатації електричних мереж убезпечує від отримання значної кількості виробничих травм.

4.2 Фактори виробничого середовища і їх вплив на життєдіяльність людей

Нормальна життєдіяльність людини вагомо залежить від умов зовнішнього середовища, зокрема виробничого. Адже в процесі трудової діяльності на організм людини чиниться своєрідний “тиск” несприятливими виробничими факторами, що прямо чи опосередковано впливають на її здоров’я та працездатність. Серед виробничих факторів прийнято розрізняти небезпечні та шкідливі.

Небезпечний виробничий фактор – виробничий фактор, дія якого за певних умов може призвести до травм або іншого раптового погіршення здоров’я працівника.

Шкідливий виробничий фактор – виробничий фактор, вплив якого може призвести до погіршення стану здоров’я, зниження працездатності працівника.

Небезпечні та шкідливі виробничі фактори за природою дії поділяються на такі групи: фізичні, хімічні, біологічні та психофізіологічні.

До фізичних небезпечних та шкідливих виробничих факторів належать: рухомі машини та механізми; пересувні частини виробничого устаткування; підвищена запиленість та загазованість повітря робочої зони; підвищена чи понижена температура поверхонь устаткування, матеріалів чи повітря робочої зони; підвищений рівень шуму, вібрацій, інфразвукових коливань, ультразвуку, іонізуючих випромінювань, статичної електрики, електромагнітних випромінювань, ультрафіолетової чи інфрачервоної радіації; підвищені чи понижені барометричний тиск, вологість, іонізація та рухомість повітря; небезпечне значення напруги в електричному колі; підвищена напруженість електричного чи магнітного полів; відсутність чи нестача природного світла; недостатня освітленість робочої зони; підвищена яскравість світла; пряме та відбите випромінювання, що створює засліплюючу дію.

До хімічних небезпечних та шкідливих виробничих факторів належать хімічні речовини, які за характером дії на організм людини поділяються на:

– загальнотоксичні, що викликають отруєння всього організму;

– подразнюючі, що викликають подразнення дихального тракту та слизових оболонок;

– сенсибілізуючі, що діють як алергени;

– канцерогенні, що викликають ракові захворювання;

– мутагенні, що призводять до змін наслідкової інформації;

– такі, що впливають на репродуктивну (дітонароджувальну) функцію.

До біологічних небезпечних та шкідливих виробничих факторів належать патогенні мікроорганізми (бактерії, віруси, мікроскопічні грибки та ін.) та продукти їх життєдіяльності, а також макроорганізми (рослини та тварини).

До психофізіологічних небезпечних та шкідливих виробничих факторів належать фізичні (статичні та динамічні) і нервово-психічні перевантаження (розумове перенапруження, перенапруження органів чуття, монотонність праці, емоційні перевантаження).

Один і той же небезпечний і шкідливий виробничий фактор за природою своєї дії може належати одночасно до різних груп.

Залежно від наслідків впливу на працюючих шкідливих та небезпечних виробничих факторів розрізняють виробничі травми, професійні захворювання та професійні отруєння, внаслідок яких може відбутись зниження або втрата працездатності (тимчасова чи постійна, повна чи часткова), можливий і фатальний кінець.

Виробнича травма – порушення анатомічної цілісності організму людини або його функцій внаслідок дії виробничих факторів.

Професійне захворювання – патологічний стан людини, обумовлений роботою і пов’язаний з надмірним напруженням організму або несприятливою дією шкідливих виробничих факторів.

Професійне отруєння – це порушення стану здоров’я в результаті дії шкідливих речовин при їх проникненні в організм людини у виробничих умовах. Довготривалий вплив незначних доз шкідливих речовин призводить до хронічних отруєнь. Проникнення в організм великої кількості шкідливих речовин за короткий час (не більше доби) спричинює гострі отруєння.

Суттєвий вплив на стан організму працівника, його працездатність здійснює мікроклімат (метеорологічні умови) виробничого приміщення, під яким розуміють клімат внутрішнього середовища цього приміщення, який

визначається температурою, відносною вологістю, рухом повітря та тепловим випромінюванням нагрітих поверхонь, що в сукупності впливають на тепловий стан організму людини.

В процесі трудової діяльності людина перебуває у постійній тепловій взаємодії з виробничим середовищем. За нормальних мікрокліматичних умов в організмі працівника, завдяки терморегуляції, підтримується постійна температура тіла (36,6 °С).

Кількість тепла, що утворюється в організмі, залежить від фізичного навантаження працівника, а рівень тепловіддачі – від мікрокліматичних умов виробничого приміщення. Віддача тепла організмом людини здійснюється, в основному, за рахунок випромінювання і випаровування вологи з поверхні шкіри.

Можливості організму пристосовуватись до метеорологічних умов значні, однак небезмежні. Верхньою межею терморегуляції людини, що знаходиться у стані спокою, прийнято вважати 30–31 °С при відносній вологості 85% чи 40 °С при відносній вологості 30%. При виконанні фізичної роботи ця межа значно нижча. Так, при виконанні важкої роботи теплова рівновага зберігається при температурі повітря 12–14 °С.

Серед чинників зовнішнього середовища, що впливають на організм людини в процесі праці, світлу відводиться одне із чільних місць. Адже відомо, що майже 90% всієї інформації про довкілля людина отримує через органи зору.

Вплив світла на життєдіяльність людини вивчений досить добре. Воно впливає не лише на функцію зору, а й на діяльність організму в цілому: посилюється обмін речовин, збільшується поглинання кисню і виділення вуглекислого газу. Відомий сприятливий вплив природного освітлення на скелетну мускулатуру.

Недостатня або надмірна освітленість, нерівномірність освітлення в полі зору втомлює очі, призводить до зниження продуктивності праці; при цьому

зростає потенційна небезпека помилкових дій і нещасних випадків. Надмірна яскравість джерел світла може спричинити головний біль, різь в очах, розлад гостроти зору; світлові відблиски – тимчасове засліплення.

Для створення сприятливих умов зорової роботи, які б виключали швидку втомлюваність очей, виникнення професійних захворювань, нещасних випадків і сприяли підвищенню продуктивності праці та якості продукції, виробниче освітлення повинно відповідати таким вимогам:

– створювати на робочій поверхні освітленість, що відповідає характеру зорової роботи і не є нижчою за встановлені норми;

– не повинно бути засліплюючої дії як від самих джерел освітлення, так і від інших предметів, що знаходяться в полі зору;

– забезпечити достатню рівномірність та постійність рівня освітленості у виробничих приміщеннях, щоб уникнути частої переадаптації органів зору;

– не створювати на робочій поверхні різких та глибоких тіней (особливо рухомих);

– повинен бути достатній, для розрізнення деталей, контраст поверхонь, що освітлюються;

Залежно від джерела світла виробниче освітлення може бути природним, штучним і суміщеним, при якому недостатнє за нормами природне освітлення доповнюється штучним.

Природне освітлення має важливе фізіолого-гігієнічне значення для працюючих. Воно сприятливо впливає на органи зору, стимулює фізіологічні процеси, підвищує обмін речовин та покращує розвиток організму в цілому. Сонячне випромінювання зігріває та знезаражує повітря, очищуючи його від збудників багатьох хвороб (наприклад, вірусу грипу). Окрім того, природне світло має і психологічну дію, створюючи в приміщенні для працівників відчуття безпосереднього зв’язку з довкіллям.

Природному освітленню властиві і недоліки: воно непостійне в різні періоди доби та року, в різну погоду; нерівномірно розподіляється по площі

виробничого приміщення; при незадовільній його організації може викликати засліплення органів зору.

Штучне освітлення може бути загальним та комбінованим. Загальним називають освітлення, при якому світильники розміщуються у верхній зоні приміщення (не нижче 2,5 м над підлогою) рівномірно (загальне рівномірне освітлення) або з врахуванням розташування робочих місць (загальне локалізоване освітлення). Комбіноване освітлення складається із загального та місцевого. Його доцільно застосовувати при роботах високої точності, а також, якщо необхідно створити певний або змінний, в процесі роботи, напрямок світла. Місцеве освітлення створюється світильниками, що концентрують світловий потік безпосередньо на робочих місцях. Застосування лише місцевого освітлення не допускається з огляду на небезпеку виробничого травматизму та професійних захворювань.

Штучне освітлення передбачається у всіх виробничих та побутових приміщеннях, де недостатньо природного світла, а також для освітлення приміщень в темний період доби. При організації штучного освітлення необхідно забезпечити сприятливі гігієнічні умови для зорової роботи і одночасно враховувати економічні показники.

Також до факторів виробничого середовища відносяться: віброакустичні коливання, виробничі випромінювання (іонізуючі, ультрафіолетові і лазерні), електромагнітні поля. Проте до основних виробничих факторів які є при усіх видах трудової діяльності є мікроклімат та чинники зовнішнього освітлення.

ВИСНОВКИ

У кваліфікаційній роботі магістра було досліджено технологію Smart Grid, що забезпечує ефективне розподілення енергоресурсів та використовується в критичній інфраструктурі електроенергетики. Якщо електромережа використовує Smart Grid, то працездатність електромережі перекладається на Smart Grid.

Досліджено методи, що використовуються для забезпечення безпеки в меш мережах.

Було проаналізовано алгоритми автентифікації в меш мережах, досліджено основні вимоги, що необхідні для забезпечення захисту від існуючих загроз та загроз, що можуть з’явитися із розвитком квантових обчислень.

Вдосконалено модель меш мережі в симуляторі ns-3, а також створено моделі захисту з використанням автентифікації використовуючи алгоритми RSA та Merkle Tree.

Проведено моделювання функцій автентифікації в меш мережах. Результатами моделювання стали: аналіз швидкодії алгоритмів, дослідження переваг та недоліків кожного з алгоритмів.

В процесі дослідження було обґрунтовано доцільність використання алгоритму Tree Merkle, для автентифікації в критичній інфраструктурі, що використовує технологію Smart Grid.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. NIST 7628, “Guidelines for Smart Grid Cyber Security”, p. 3, September 2010.

2. Що таке ISO 27001?. [Електронний ресурс]. URL: https://ims-cert.com/mezhdunarodnaya-sertifikacziya-ua/iso/iec-27001-ua.html.

3. Про затвердження Порядку проведення державної експертизи в сфері технічного захисту інформації в Управлінні державної охорони України. [Електронний ресурс]. URL: https://zakon.rada.gov.ua/laws/show/z0728-18#Text

4. D.Pharkkavi, Dr. D. Maruthanayagam (2014) Space Complexity Analysis of RSA and ECC Based Security Algorithms in Cloud Data:1–198

5. Anubhav Ratha, (2012) ESI Bulletin on Energy Trends and Development in Cloud Data:1–198

6. The NTRU Project. [Електронний ресурс]. URL: http://tbuktu.github.io/ntru/.

7. Парадигма розподілу Spoke-Hub. [Електронний ресурс]. URL: https://hmn.wiki/ru/Hub_and_spoke.

8. NaCl: Networking and Cryptography library. URL: https://nacl.cr.yp.to/hash.html

9. The ZigBee Protocol. [Електронний ресурс]. URL: https://www.netguru.com/blog/the-zigbee-protocol

10. NS3-for-authentication-schemes. [Електронний ресурс]. URL: https://github.com/soumyaxyz/NS3-for-authentication-schemes

ДОДАТОК А – АПРОБАЦІЯ НАУКОВИХ РОБІТ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ІВАНА ПУЛЮЯ

М А Т Е Р І А Л И

X НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

«ІНФОРМАЦІЙНІ МОДЕЛІ, СИСТЕМИ ТА ТЕХНОЛОГІЇ»

7–8 грудня 2022 року

ТЕРНОПІЛЬ
2022

УДК 004.056
С. Сербичанський
(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

ДОСЛІДЖЕННЯ ВИМОГ ДО ФІЗИЧНОГО ТА ПРОГРАМНОГО ЗАХИСТУ ІНФОРМАЦІЇ НА ОБ'ЄКТАХ КРИТИЧНОЇ ІНФРАСТРУКТУРИ В УМОВАХ ЗАГРОЗ І ОБМЕЖЕНЬ

UDC 004.056
S. Serbychanskyi

STUDY OF REQUIREMENTS FOR PHYSICAL AND SOFTWARE PROTECTION OF INFORMATION AT CRITICAL INFRASTRUCTURE OBJECTS UNDER THE CONDITIONS OF THREATS AND LIMITATIONS

Електромережі є основою в сучасному світі, засоби зв'язку, банківська ж система, медична система та більшість інших важливих систем які не можна увити без систем електропостачання. Передбачається, що до 2050 року світове електроспоживання зросте втричі.

Переважно атомні електростанції покривають базову потребу використання (тобто використання електроенергії яке є постійним). Гідроелектростанції покривають в першу чергу перепади в потребах живлення до прикладу час пік.

Теплові електростанції покривають постійні та незначні перепади електроенергії. Звісно існують різні конфігурації мереж в залежності від потреб. Про те швидке інформування місць генерації електроенергії є досить складною задачею. Погіршує проблему саме використання зеленої електроенергетики. Таким чином ж інформування місць генерації та швидка взаємодія набуває основного значення.

Тому різними державами активно виділяється значна кількість коштів на розробку нової системи, що повинна пов'язати місця виробництва електроенергії та їх використання. Нові мережі мають децентралізовану цифрову інфраструктуру, що називаються smart grid.

Smart Grid об'єднує кілька добре відомих, але відмінних галузей, а саме електроенергетику, інформаційні технології та зв'язок.

Кожен пристрій у новій сітці, ймовірно, матиме власну IP-адресу та використовуватиме для зв'язку такі протоколи, як TCP/IP. Таким чином, вони будуть уразливі до подібних загроз безпеці, з якими стикаються сучасні комунікаційні мережі, однак ставки набагато вищі.

Для автентифікації пристрою використовуються алгоритми з відкритим ключем які не є досконалими та мають ряд недоліків при використанні в мережах smart grid.

Альтернативою використанню ресурсоємної автентифікації відкритого ключа є система на основі дерев Меркла. Дерева Merkle пропонують недорогу автентифікацію для mesh-клієнтів. Порівняно з відкритим ключем, вони легкі та швидкі для генерації та пропонують такий самий, а у деяких випадках, кращий захист.

ДОДАТОК В – ЛІСТИНГ КОДУ

```
NS_LOG_COMPONENT_DEFINE("MeshExample");
uint32_t g_udpTxCount = 0; //!< Rx packet counter.
uint32_t g_udpRxCount = 0; //!< Tx packet counter.
void
TxTrace(Ptr<const Packet> p)
{
    NS_LOG_DEBUG("Sent " << p->GetSize() << " bytes");
    g_udpTxCount++;
}
void
RxTrace(Ptr<const Packet> p)
{
    NS_LOG_DEBUG("Received " << p->GetSize() << " bytes");
    g_udpRxCount++;
}
class MeshTest
{
  public:
        MeshTest();
        void Configure(int argc, char** argv);
        int Run();
  private:
    int m_xSize;
    double m_randomStart;
    double m_totalTime;
    double m_packetInterval;
    uint16_t m_packetSize;
    uint32_t m_nIfaces;
    bool m_chan;
    bool m_pcap;
    bool m_ascii;
    std::string m_stack;
    std::string m_root;
    NodeContainer nodes;
    NetDeviceContainer meshDevices;
    Ipv4InterfaceContainer interfaces;
    MeshHelper mesh;
  private:
    void CreateNodes();
    void InstallInternetStack();
    void InstallApplication();
    void Report();
};
MeshTest::MeshTest()
    : m_xSize(3),
      m_ySize(3),
      m_step(50.0),
      m_randomStart(0.1),
      m_totalTime(100.0),
      m_packetInterval(1),
```

```
      m_packetSize(1024),
      m_nIfaces(1),
      m_chan(true),
      m_pcap(false),
      m_ascii(false),
      m_stack("ns3::Dot11sStack"),
      m_root("ff:ff:ff:ff:ff:ff")
{
}

void
MeshTest::Configure(int argc, char* argv[])
{
    CommandLine cmd(__FILE__);
    cmd.AddValue("x-size", "Number of nodes in a row grid",
m_xSize);
    cmd.AddValue("y-size", "Number of rows in a grid", m_ySize);
    cmd.AddValue("step", "Size of edge in our grid (meters)",
m_step);
    cmd.AddValue("start", "Maximum random start delay for beacon
jitter (sec)", m_randomStart);
    cmd.AddValue("time", "Simulation time (sec)", m_totalTime);
    cmd.AddValue("packet-interval", "Interval between packets in
UDP ping (sec)", m_packetInterval);
    cmd.AddValue("packet-size", "Size of packets in UDP ping
(bytes)", m_packetSize);
    cmd.AddValue("interfaces", "Number of radio interfaces used by
each mesh point", m_nIfaces);
    cmd.AddValue("channels", "Use different frequency channels for
different interfaces", m_chan);
    cmd.AddValue("pcap", "Enable PCAP traces on interfaces",
m_pcap);
    cmd.AddValue("ascii", "Enable Ascii traces on interfaces",
m_ascii);
    cmd.AddValue("stack", "Type of protocol stack.
ns3::Dot11sStack by default", m_stack);
    cmd.AddValue("root", "Mac address of root mesh point in HWMP",
m_root);
    cmd.Parse(argc, argv);
    NS_LOG_DEBUG("Grid:" << m_xSize << "*" << m_ySize);
    NS_LOG_DEBUG("Simulation time: " << m_totalTime << " s");
    if (m_ascii)
    {
        PacketMetadata::Enable();
    }
}
void
MeshTest::CreateNodes()
{
    nodes.Create(m_ySize * m_xSize);
    YansWifiPhyHelper wifiPhy;
    YansWifiChannelHelper wifiChannel =
YansWifiChannelHelper::Default();
```

```
    wifiPhy.SetChannel(wifiChannel.Create());
    mesh = MeshHelper::Default();
    if (!Mac48Address(m_root.c_str()).IsBroadcast())
    {
        mesh.SetStackInstaller(m_stack, "Root",
Mac48AddressValue(Mac48Address(m_root.c_str())));
    }
    else
    {
        mesh.SetStackInstaller(m_stack);
    }
    if (m_chan)
    {

mesh.SetSpreadInterfaceChannels(MeshHelper::SPREAD_CHANNELS);
    }
    else
    {
        mesh.SetSpreadInterfaceChannels(MeshHelper::ZERO_CHANNEL);
    }
    mesh.SetMacType("RandomStart",
TimeValue(Seconds(m_randomStart)));
    mesh.SetNumberOfInterfaces(m_nIfaces);
    meshDevices = mesh.Install(wifiPhy, nodes);
    mesh.AssignStreams(meshDevices, 0);
    MobilityHelper mobility;
    mobility.SetPositionAllocator("ns3::GridPositionAllocator",
                                  "MinX",
                                  DoubleValue(0.0),
                                  "MinY",
                                  DoubleValue(0.0),
                                  "DeltaX",
                                  DoubleValue(m_step),
                                  "DeltaY",
                                  DoubleValue(m_step),
                                  "GridWidth",
                                  UintegerValue(m_xSize),
                                  "LayoutType",
                                  StringValue("RowFirst"));

mobility.SetMobilityModel("ns3::ConstantPositionMobilityModel");
    mobility.Install(nodes);
    if (m_pcap)
    {
        wifiPhy.EnablePcapAll(std::string("mp"));
    }
    if (m_ascii)
    {
        AsciiTraceHelper ascii;
        wifiPhy.EnableAsciiAll(ascii.CreateFileStream("mesh.tr"));
    }
}
void
```

```
MeshTest::InstallInternetStack()
{
    InternetStackHelper internetStack;
    internetStack.Install(nodes);
    Ipv4AddressHelper address;
    address.SetBase("10.1.1.0", "255.255.255.0");
    interfaces = address.Assign(meshDevices);
}
void
MeshTest::InstallApplication()
{
    uint16_t portNumber = 9;
    UdpEchoServerHelper echoServer(portNumber);
    uint16_t sinkNodeId = m_xSize * m_ySize - 1;
    ApplicationContainer serverApps =
echoServer.Install(nodes.Get(sinkNodeId));
    serverApps.Start(Seconds(1.0));
    serverApps.Stop(Seconds(m_totalTime + 1));
    UdpEchoClientHelper
echoClient(interfaces.GetAddress(sinkNodeId), portNumber);
    echoClient.SetAttribute("MaxPackets",
    UintegerValue((uint32_t)(m_totalTime * (1 /
m_packetInterval))));
    echoClient.SetAttribute("Interval",
TimeValue(Seconds(m_packetInterval)));
    echoClient.SetAttribute("PacketSize",
UintegerValue(m_packetSize));
    ApplicationContainer clientApps =
echoClient.Install(nodes.Get(0));
    Ptr<UdpEchoClient> app = clientApps.Get(0)-
>GetObject<UdpEchoClient>();
    app->TraceConnectWithoutContext("Tx", MakeCallback(&TxTrace));
    app->TraceConnectWithoutContext("Rx", MakeCallback(&RxTrace));
    clientApps.Start(Seconds(1.0));
    clientApps.Stop(Seconds(m_totalTime + 1.5));
}
int
MeshTest::Run()
{
    CreateNodes();
    InstallInternetStack();
    InstallApplication();
    Simulator::Schedule(Seconds(m_totalTime), &MeshTest::Report,
this);
    Simulator::Stop(Seconds(m_totalTime + 2));
    Simulator::Run();
    Simulator::Destroy();
    std::cout << "UDP echo packets sent: " << g_udpTxCount << "
received: " << g_udpRxCount
              << std::endl;
    return 0;
}
void
```

```
MeshTest::Report()
{
    unsigned n(0);
    for (NetDeviceContainer::Iterator i = meshDevices.Begin(); i
!= meshDevices.End(); ++i, ++n)
    {
        std::ostringstream os;
        os << "mp-report-" << n << ".xml";
        std::cerr << "Printing mesh point device #" << n << "
diagnostics to " << os.str() << "\n";
        std::ofstream of;
        of.open(os.str().c_str());
        if (!of.is_open())
        {
            std::cerr << "Error: Can't open file " << os.str() <<
"\n";
            return;
        }
        mesh.Report(*i, of);
        of.close();
    }
}
int
main(int argc, char* argv[])
{
    MeshTest t;
    t.Configure(argc, argv);
    return t.Run();
}
int BianaryTransform(int num, int bin_num[])
{
     int i = 0,  mod = 0;
while(num != 0)
     {
          mod = num%2;
          bin_num[i] = mod;
          num = num/2;
          i++;
     }
return i;
}
long long Modular_Exonentiation(long long a, int b, int n)
{
     int c = 0, bin_num[1000];
     long long d = 1;
     int k = BianaryTransform(b, bin_num)-1;

     for(int i = k; i >= 0; i--)
     {
          c = 2*c;
          d = (d*d)%n;
          if(bin_num[i] == 1)
          {
```

```
                c = c + 1;
                d = (d*a)%n;
           }
     }
     return d;
}
int ProducePrimeNumber(int prime[])
{
     int c = 0, vis[1001];
     memset(vis, 0, sizeof(vis));
     for(int i = 2; i <= 1000; i++)if(!vis[i])
     {
          prime[c++] = i;
          for(int j = i*i; j <= 1000; j+=i)
               vis[j] = 1;
     }
     return c;
}
int Exgcd(int m,int n,int &x)
{
     int x1,y1,x0,y0, y;
     x0=1; y0=0;
     x1=0; y1=1;
     x=0; y=1;
     int r=m%n;
     int q=(m-r)/n;
     while(r)
     {
          x=x0-q*x1; y=y0-q*y1;
          x0=x1; y0=y1;
          x1=x; y1=y;
          m=n; n=r; r=m%n;
          q=(m-r)/n;
     }
     return n;
}
void RSA_Initialize()
{
int prime[5000];
     int count_Prime = ProducePrimeNumber(prime);
srand((unsigned)time(NULL));
     int ranNum1 = rand()%count_Prime;
     int ranNum2 = rand()%count_Prime;
     int p = prime[ranNum1], q = prime[ranNum2];

     n = p*q;

     int On = (p-1)*(q-1);

for(int j = 3; j < On; j+=1331)
     {
     int gcd = Exgcd(j, On, d);
     if( gcd == 1 && d > 0)
```

```
          {
               e = j;
               break;
          }
     }
 }
void RSA_Encrypt()
{
     cout<<"Public Key (e, n) : e = "<<e<<" n = "<<n<<'\n';
     cout<<"Private Key (d, n) : d = "<<d<<" n = "<<n<<'\n'<<'\n';

     int i = 0;
     for(i = 0; i < 100; i++)
          Ciphertext[i] = Modular_Exonentiation(Plaintext[i], e,
n);

     cout<<"Use the public key (e, n) to encrypt:"<<'\n';
     for(i = 0; i < 100; i++)
          cout<<Ciphertext[i]<<" ";
     cout<<'\n'<<'\n';

}
void RSA_Decrypt()
{
     int i = 0;
     for(i = 0; i < 100; i++)
          Ciphertext[i] = Modular_Exonentiation(Ciphertext[i], d,
n);

     cout<<"Use private key (d, n) to decrypt:"<<'\n';
     for(i = 0; i < 100; i++)
          cout<<Ciphertext[i]<<" ";
     cout<<'\n'<<'\n';
}
void Initialize()
{
     int i;
     srand((unsigned)time(NULL));
     for(i = 0; i < 100; i++)
          Plaintext[i] = rand()%1000;

     cout<<"Generate 100 random numbers:"<<'\n';
     for(i = 0; i < 100; i++)
          cout<<Plaintext[i]<<" ";
     cout<<'\n'<<'\n';
}
ApplicationContainer verification (ApplicationContainer
appContainer, double time, Ptr<Node> user, Ptr<Node> gateway ,
Ptr<Node> device ){
  if (verbose){
    std::cout<<"user : "<< user->GetObject<Ipv4> ()->GetAddress
(1, 0).GetLocal ();
 ()<<std::endl;
```

```
  }
     appContainer = sendMessage(appContainer, time, node);
     appContainer = sendMessage(appContainer, time, node);
     appContainer = sendMessage(appContainer, time, node);
  return appContainer;
}
MerkleTree::MerkleTree(std::vector<Node*> blocks) {
    std::vector<Node*> nodes;
    while (blocks.size() != 1) {
        printNodeHashes(blocks);
        for (unsigned int l = 0, n = 0; l < blocks.size(); l = l +
2, n++) {
            if (l != blocks.size() - 1) { // checks for adjacent
block
                nodes.push_back(new Node(cryptohash (blocks[l]-
>hash + blocks[l+1]->hash))); // combine and hash adjacent blocks
                nodes[n]->left = blocks[l]; // assign children
                nodes[n]->right = blocks[l+1];
            } else {
                nodes.push_back(blocks[l]);
            }
        }
        std::cout << "\n";
        blocks = nodes;
        nodes.clear();
    }
    this->root = blocks[0];
}
MerkleTree::~MerkleTree() {
    deleteTree(root);
    std::cout << "Tree deleted" << std::endl;
}
void MerkleTree::printTree(Node* n, int indent) {
    if (n) {
        if (n->left) {
            printTree(n->left, indent + 4);
        }
        if (n->right) {
             printTree(n->right, indent + 4);
        }
        if (indent) {
            std::cout << std::setw(indent) << ' ';
        }
        std::cout << n->hash[0] << "\n ";
    }
}
void MerkleTree::deleteTree(Node* n) {
    if (n) {
        deleteTree(n->left);
        deleteTree(n->right);
        n = NULL;
        delete n;
    }
```

```
}
Node::Node(std::string data) {
    this->hash = data;
}
struct Node {
    std::string hash;
    Node *left;
    Node *right;

    Node(std::string data);
};
int main() {
    std::vector<Node*> leaves;
    for (unsigned int i = 0; i < leaves.size(); i++) {
        leaves[i]->left = NULL;
        leaves[i]->right = NULL;
    }
    MerkleTree *hashTree = new MerkleTree(leaves);
    std::cout << hashTree->root->hash << std::endl;
    hashTree->printTree(hashTree->root, 0);
    for (unsigned int k = 0; k < leaves.size(); k++) {
        delete leaves[k];
    }
    delete hashTree;

    return 0;
}
```