МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Дніпровський національний університет залізничного транспорту імені академіка В. Лазаряна

Кафедра «Електронні обчислювальні машини»

«ДО ЗАХИСТУ»

Завідувач кафедри

_____ _____
(підпис) (ПІБ)
«____» _____ 20____р.

ДИПЛОМНИЙ ПРОЕКТ

на здобуття ОКР «сцеціаліст» / «магістр»

Галузь знань_____12_____Інформаційні технології_____
(шифр) (назва)

Спеціальність _____125_____Кібербезпека_____
(код) (повна назва)

Тема_Розробка_додатку_до_системи_Lider_для_відпрацювання_лабораторних_робіт_за_курсом_«Прикладна_криптологія»_____
_____
_____

Керівник дипломного проекту _____ _____ _____
(посада) (підпис) (ПІБ)

Консультант розділу з БЖД _____ _____ _____
(посада) (підпис) (ПІБ)

Нормоконтролер _____ _____ _____
(посада) (підпис) (ПІБ)

Студент групи _____ _____ _____
(група) (підпис) (ПІБ)

Student _____
(family name)

Дніпро
2020

Дніпровський національний університет залізничного транспорту імені академіка В. Лазаряна

Факультет комп'ютерних технологій і систем кафедра ЕОМ

Спеціальність Кібербезпека

«ЗАТВЕРДЖУЮ»

Завідувач кафедри

_____

(підпис)

«___»_____2020р.

ЗАВДАННЯ

до дипломної роботи на здобуття ОКР магістр

студента групи КБ1921 Бєляєва О.І.

(номер групи) (ПІБ)

1 Тема дипломної роботи Розробка додатку до системи Lider для відпрацювання лабораторних робіт за курсом «Прикладна криптологія»

затверджена наказом по університету від «16» 12 2019 р. № 945 ст.

2 Термін подання студентом закінченої роботи 10.12.2020_____

3 Вихідні дані до дипломної роботи Опис системи Lider. Перелік лабораторних робіт: Комплекс робіт по вивченню сучасних квантових криптографічних протоколів

4 Зміст пояснювальної записки (перелік питань до розробки) Вступ, огляд проблематики, огляд алгоритмів квантової криптографії, постановка задачі, розробка веб-додатку, інструкція до використання додатку на практичних та

лабораторних заняттях, охорона праці та безпека в надзвичайних ситуаціях, висновки, список використаних джерел_____

5 Перелік креслень (демонстраційного матеріалу) Схеми квантових криптографічних протоколів, демонстрації сторінок навчальної програми

6 Розділи та консультанти

Розділ	Консультант	Підпис, дата	
		завдання видав	завдання прийняв
Основний	Проф. Жуковицький І.В.		
Охорона праці та безпека в надзвичайних ситуаціях			

КАЛЕНДАРНИЙ ПЛАН

Назва розділу	Термін виконання	Обсяг розділу, %
Вступ		1%
Огляд проблематики		5%
Огляд алгоритмів квантової криптографії		15%
Постановка задачі		5%
Розробка веб-додатку		40%
Інструкція до використання додатку на практичних та лабораторних заняттях		10%
Охорона праці та безпека в надзвичайних ситуаціях		5%
Висновки		1%
Оформлення пояснювальної записки		15%
Підготовка презентації		3%

Дата видачі завдання: «___»_____ 20___ р.

Керівник дипломного проекту (роботи) _____ _____
(підпис) (ПІБ)

Завдання прийняв до виконання _____ _____
(підпис) (ПІБ)

Зміст

РЕФЕРАТ

Темою магістерської дипломної роботи є розробка веб-додатку для дослідження протоколів і технологій квантової криптографії.

Робота містить 77 сторінок, зокрема 30 ілюстрацій, 3 таблиці та 23 джерела інформації.

Тема магістерської дипломної роботи є актуальною, оскільки вивчення технологій квантової криптографії є важливою складовою галузі інформаційної безпеки, а отже має здійснюватись в максимально ефективний спосіб.

Мета роботи полягає в розробці веб-додатку, який би дозволяв вивчати принципи роботи алгоритмів протоколів квантової криптографії та дозволяв би напрацьовувати практичні навички у їх використанні.

Об’єктом дослідження є основні протоколи квантової криптографії, що існують і знайшли застосування на сьогодні.

Предметом дослідження є розробка практичного засобу для вивчення протоколів квантової криптографії.

При виконанні роботи проводився аналіз протоколів квантового розподілу секретного ключа між абонентами.

У роботі був розроблений і запропонований потужний інструмент для вивчення технологій квантової криптографії.

ABSTRACT

The theme of the master's thesis is developing web-application for study protocols and technologies of quantum cryptography.

The work contains 77 pages, including 30 illustrations, 3 tables and 23 sources of information.

The topic of the master's thesis is relevant, because study quantum technologies is an important part of information security and must be doing in the best way.

The purpose of the work is to develop a web application that would allow to study the principles of algorithms of quantum cryptography protocols and would allow to develop practical skills in their use.

The object of research is the basic protocols of quantum cryptography that exist and are used today.

The subject of research is the development of a practical tool for studying the protocols of quantum cryptography.

During the work the analysis of protocols of quantum distribution of a secret key between subscribers was carried out.

The paper developed and proposed a powerful tool for studying quantum cryptography technologies.

ПЕРЕЛІК СКОРОЧЕНЬ

JS – JavaScript

BB84 – Bennet-Bassard 84

B92 – Bennet-92

E91 – Ekkert-91

SARG-04 – Scarini-Acin-Gisin

KMB-09 – Khan-Murphy-Beige-09

QKD – Quantum Key Distribution

ВСТУП

Квантова криптографія — напрямок криптології, що вивчає можливості захисту інформації за допомогою властивостей квантових часток. Ідейними засновниками даної галузі наприкінці XX століття стали Чарльз Беннет і Жиль Бассард, які у 1984 році, розробили криптографічний протокол BB84, що базувався на дослідженнях Стівена Візнера і ґрунтувався на законах квантової фізики, а не на математичних законах, як це було до того.

Основною ідеєю нового підходу стало використання фотонів як носіїв інформації. Це гарантувало цілісність і недоторканість передаваних даних, адже швидкість світла дозволяла виключити можливість зміни даних в каналах зв'язку.

Розквіт електроніки дав в свою чергу серйозний поштовх до подальшої розвитку галузі квантової криптографії, що тепер орієнтувалась на появу і поширення квантових комп'ютерів і захисту даних від їх обчислювальних можливостей.

Так, за останні 35 років з'явились численні протоколи квантової криптографії. Найбільшого поширення з них набули зокрема протоколи розподілу секретного ключа, серед яких: BB84, E91, B92, SARG-04, KMB-09, COW, S09, S13, DPS, AK-15 та деякі інші.

Ефективність кожного з них різниться в залежності від ідей, на яких вони були побудовані та складності побудови самих протоколів. Однак вже нині є цілком очевидним, що в перспективі дані протоколи та їх ідейні нащадки разом з деякими елементами класичної криптографії стануть основою захисту обміну інформацією.

ОГЛЯД ПРОБЛЕМАТИКИ

Розвиток можливостей інформаційно-обчислювальної техніки, що увійшов до своєї активної стадії наприкінці минулого століття дозволив вивести промисловість, науку та електроніку на досі небачений рівень інтеграції з інформаційними технологіями. Нині практично будь-яка сфера людської діяльності, зокрема якщо казати про країни першого та другого світу, в тому чи іншому ступені зазнала автоматизації та комп'ютеризації. Серед очевидних переваг такого роду модернізації можна виділити наприклад: пришвидшення бізнес-процесів, створення принципово нових способів ведення підприємницької діяльності, можливість миттєвого обміну інформацією, тощо. Відносно низька вартість, доступність і багатофункціональність сучасної інформаційно-обчислювальної техніки в свою чергу тільки сприяють процесу її подальшої інтеграції до найрізноманітніших сфер життя людини.

Втім, поруч із постійно зростаючими перевагами непомітно, але цілком реально зростають і ризики. Ризики, що виливаються у політичні, економічні, репутаційні втрати.

Головним фактором появи ризиків у сфері інформаційних технологій була й донині залишається безпека даних, що полягає в дотриманні секретності, цілісності й доступності інформації.

Виконання цих умов потребує використання численних апаратних і програмних засобів, що були б побудовані на базі не скомпрометованих (на момент їх застосування) технологій. Або принаймні забезпечували такий рівень інформаційного захисту даних, аби витрати на захист не перевищували потенційної шкоди, що її можуть заподіяти зловмисники.

В цьому плані одним з найефективніших рішень, як в сенсі збереження секретності, так і в сенсі забезпечення доступності інформації, є криптографічне зашифровування даних. Програмна реалізація шифрування

дозволяє відносно швидку зміну технологій без необхідності заміни серверного обладнання або принципової перебудови мереж зв’язку. Водночас швидкості виконання сучасними комп’ютерами криптографічних операцій дозволяють зберігати оперативність при роботі з шифрованою інформацією.

Зворотнім боком медалі та критичним недоліком криптографічного шифрування даних є той важливий момент, що з постійним нарощуванням обчислювальних потужностей сучасних комп’ютерів скорочуються фінансові, матеріально-технічні та часові витрати на дискредитацію криптографічних ключів та протоколів. Так, донедавна цілком безпечні протоколи та розміри ключів до них, стають вразливими перед дедалі новішими способами криптоаналізу та звичайного методу перебору (“brute force” — від англ. “груба сила”). Збільшення розміру ключів при цьому може бути радше тимчасовим, та аж ніяк не остаточним рішенням проблеми, адже це, по-перше, не зупиняє розвитку технологій, а відповідно потенційних можливостей імовірних зловмисників; по-друге, уповільнює роботу самих інформаційних систем, спричиняючи тим самим як фінансових, так і репутаційних збитків.

Яскравим прикладом на підтвердження цьому факту є експеримент французької групи дослідників, на чолі з Еммануелем Томе (фран. “Emmanuel Thomé”) з французького Національного інституту досліджень в сфері комп’ютерних наук і автоматизації (фран. “National Institute for Research in Computer Science and Automation in France”) у 2019-ому році, під час якого групі науковців вдалось зламати ключ до криптоалгоритму RSA-240 [7], довжиною 795 біт. Час, витрачений на злам склав близько 8 мільйонів годин в переводі на роботу одного комп’ютеру. Попередній експеримент даної дослідницької групи зі зламу ключа довжиною 768 біт того ж таки RSA-240 [8], який відбувся у 2010-ому році, зайняв близько 35 мільйонів годин. Попри те, що наразі найбільш широко вживаною довжиною ключа даного криптоалгоритму є 2048

бітів, результати двох експериментів наочно вказують, що у найближчі 25 років даний рубіж буде впевнено подоланий, а відповідно зникне гарантія цілковитої захищеності як персональних даних і заощаджень мільярдів громадян, так і засекречених державних даних.

Не останню роль в цьому процесі відіграють квантові технології, зокрема так звані квантові комп'ютери, які за прогнозами дослідників зможуть зламувати ключі до сучасних криптоалгоритмів за лічені години.

Інтенсивний характер розвитку технологій не дозволяє надалі дотримуватись екстенсивних підходів у розв'язанні питань, пов'язаних із проблемами інформаційної безпеки. Інакше кажучи, постійне збільшення довжини ключів та рівнів складності систем безпеки без зміни технологій на більш сучасні приречене на поступовий занепад та значно більші втрати з боку держави, підприємств і пересічних громадян.

Саме тому невідворотним є перехід в сфері інформаційних технологій до кардинально нових підходів, зокрема й до алгоритмів і протоколів квантової криптографії.

Найбільш поширеним і реалізованим на практиці наразі є сімейство протоколів розподілу секретного ключа в квантовій криптографії. Питання розподілу секретного ключа між абонентами є одним з найбільш проблематичних в цілій криптографії. Адже успішне його розв'язання дозволило б використання абсолютно стійких шифрів. Простіше кажучи, це могло б серйозно ускладнити будь-які спроби зловмисників порушити секретність чи цілісність інформації.

Розробка подібного роду протоколів, заснованих на принципах квантової криптографії почалась наприкінці минулого століття і за понад 35 років досліджень створила цілий ряд продуктів та їх модифікації, що нині вже використовуються у великих міжнародних компаніях, але наразі ще не готові до

масового вжитку. Використання квантових технологій потребує необхідного обладнання, яке продовжують вдосконалювати численні дослідницькі центри та великі компанії, зокрема: MagiQ Technologies (QPN Security Gateway), ID Quantique (Clavis2, Cerberis), Toshiba Research Europe Ltd (Quantum Key Server), Institute for Quantum Optics and Quantum Information, Northwestern University, SmartQuantum, BBN Technologies of Cambridge, TREL, NEC, Mitsubishi Electric, ARS Seibersdorf Research, Los Alamos National Laboratory, QinetiQ (Quantum Net), тощо.

Попри це, цілком очевидно, що питання дослідження і розробки протоколів квантової криптографії є цілком дозрілим і має здобувати якнайширшого кола дослідників. В тому числі й в Україні.

На превеликий жаль, наразі в Україні існує вкрай убоге підґрунтя для ефективного проведення досліджень в галузі квантової криптографії. На сьогодні університети й дослідницькі центри практично не мають необхідного обладнання, яке дозволило б проводити практичні експерименти з передачі даних квантовими каналами зв’язку. Причиною цьому є вкрай висока вартість подібного роду обладнання та загальна нерозвиненість даного напрямку досліджень, що в свою чергу не дозволяє отримати швидких фінансових вигод від потенційного інвестування у дослідження.

Крім очевидних матеріально-технічних обмежень існує також проблема нестачі доступних методичних матеріалів з даної тематики. Дану проблему вповні можна висловити в трьох пунктах: мовний бар’єр (абсолютна більшість публікацій та досліджень здійснюється англійською мовою), доступність існуючих публікацій (значна частина досліджень публікується в системах, до яких важко отримати доступ для ознайомлення) та нестача джерел інформації з доступним для розуміння студентами викладом проблематики.

Перший аспект з вищеперелічених є цілком природнім і може бути подоланий шляхом технічно-програмних засобів перекладу.

Другий є дещо більш проблематичним. Складна система доступу до міжнародних академічних інформаційних систем і наукових журналів, на кшталт “Physical review journals” [4] є серйозною завадою при ознайомленні з новітніми науковими дослідженнями. Складність доступу з одного боку полягає в платній основі користування подібного роду ресурсами, а з іншого у складності процесу верифікації користувачів в системі, що в свою чергу є наслідком не надто високої ступені інтеграції національного наукового співтовариства в систему міжнародних досліджень.

Частковим рішенням може слугувати використання альтернативних інформаційних систем та баз даних, які б не потребували особливої аутентифікації користувачів і надавали б доступ до наукових статей безпосередньо.

До таких баз даних можна віднести систему, що функціонує на базі Корнуельського університету (Cornell University), що у США. Система arXiv.org [3] дозволяє користувачам отримувати прямий доступ через мережу Internet до великої кількості наукових статей від провідних дослідників з усього світу. Зокрема, мова йде про творців сучасної квантової криптографії: Валеріо Скарані (Valerio Scarani) — розробник криптографічного протоколу розподілу секретного ключа SARG-04, Ніколас Ґісін (Nicolas Gisin) — співрозробник протоколу SARG-04 та автор протоколу COW (протокол когерентного одностороннього шифрування), тощо.

На жаль дана система є виключно англомовною, а також не зовсім повною, адже, попри те, що вона є цілком задовільною з точки зору самостійних наукових досліджень, відповідно є прийнятною для вчених, та не може бути вповні використана для навчання широкого загалу студентства. Тому

здебільшого може бути лише прикладом і джерелом матеріалів для створення альтернативних національних академічних баз даних.

Третій же є, мабуть, найбільш складним, адже вимагає розробки, поширення і постійного доповнення грамотних і доступних для розуміння методичних матеріалів, які б дозволяли ознайомитись з історією, проблематикою, основними принципами і протоколами квантової в криптографії як звичайним студентам університетів, які проходять вивчення даної технології в рамках курсу криптології, так і дослідникам з числа студентів, магістрантів, аспірантів і викладачів, які бажають провадити свою наукову діяльність в галузі квантової криптографії.

Певним прикладом розв'язання даної проблеми можна вважати систему, що застосовується в університеті Святого Андрія (St Andrews University) [1], що у Великобританії. На сайті університету існує цифровий симулятор роботи деяких протоколів квантової криптографії з досить детальним описом принципів та подробиць роботи цих алгоритмів. Зокрема мова йде про протокол QKD [2] (відомий також як B92), а також протокол BB84. Окремо розглядаються загальні принципи роботи криптографічних протоколів, як явища.

Втім, даний приклад, попри свої незаперечні переваги не позбавлений попередніх двох аспектів основної проблеми, а саме мовного бар'єру та прямої прив'язки до наукових статей.

Серед іншого варто відзначити ще й той момент, що система університету Святого Андрія (хоч і має багато інших симуляцій на різного роду тематику) обмежена симуляцією роботи виключно двох протоколів квантової критографії (BB84, B92), які є далеко не найновішими технологіями, розробленими в даній галузі.

Крім того, хоч дана система є у вільному доступі й може бути використана для навчання будь-яким користувачем мережі Internet, та вона є досить

малознаною (зокрема в колах українських студентів і викладачів), через що не знаходить широкого застосування в університетському середовищі.

Окремо варто було б зауважити ще й те, що вказані симуляції носять радше демонстраційний характер, аніж практичний, себто вони лишень дають уявлення про принципи дії алгоритмів, не залишаючи можливості перевірки отриманих знань на практиці.

Зважаючи на це, можемо взяти дану систему за гарний приклад, але очевидно приклад, який не позбавлений недоліків, які необхідно виправляти.

Підхід до кожного з аспектів означеної вище загальної проблеми окремо від інших був би дуже половинчастим і не дозволяв би вповні вирішити проблему.

Саме тому метою даного дослідження ставимо об’єднання усіх трьох аспектів описаної вище проблеми доступності для дослідників інформації та практичного застосування отриманих знань в єдине питання, розв’язок якого бачимо шляхом розробки веб-додатку, який би містив у собі доступну для пошуку і розуміння інформацію про протоколи розподілу секретного ключа в квантовій криптографії, а саме протоколи: BB84, E91, B92, Sars-04, KMB-09. А також опис їх практичного застосування у системах обміну інформацією.

ОГЛЯД АЛГОРИТМІВ КВАНТОВОЇ КРИПТОГРАФІЇ

3.1. Протокол BB84

Початком і по суті фундаментом для подальшого розвитку прикладного напряму квантової криптографії можна вважати протокол розподілу секретного ключа квантовими каналами зв’язку — BB84.

Концепція даного протоколу була розроблена і вперше опублікована Чарльзом Беннетом (Bennett C. H.) і Жилем Брассардом (Brassard G.) у 1984 році (звідки, власне, й назва протоколу) в статті “Quantum cryptography: Public key distribution and coin tossing” [10].

В основу роботи протоколу був покладений принцип порівняння випадково згенерованих користувачами бітових послідовностей, інформація про які закодовувалась у послідовності фотонів шляхом поляризації кожного з них у відповідності до заздалегідь узгодженого абонентами базису поляризації.

Співпадіння станів поляризованих фотонів в двох абонентів в результаті стає спільним секретним ключем абонентів.

В своїй праці Беннет і Брассард описують алгоритм роботи протоколу як послідовність з чотирьох кроків:

1) Абонент А обирає випадковим чином один базис (припустимо, прямолінійний) та послідовність випадкових бітів (однієї тисячі має бути достатньо). Абонент закодовує свої біти як послідовність фотонів у відповідності до обраного базису, використовуючи ту ж таки схему кодування, що і раніше. Після цього відправляє отриману послідовність поляризованих фотонів абоненту В.

2) Абонент В самостійно і випадково вибирає для кожного фотона послідовність базисів для декодування надісланих абонентом А значень. Відповідно він зчитує фотони, записуючи результати у дві таблиці, одну з прямолінійно отриманих фотонів, а другу з діагонально прийнятих фотонів. Через втрати в його детекторах та в каналі передачі деякі фотони можуть не прийматись взагалі, що призведе до пропусків в його таблицях. В той же час абонент В робить припущення щодо того, яким набором базисів скористовся абонент А, і повідомляє про це йому. Якщо припущення було вірним, спільний секретний ключ збільшується на одне значення (біт).

3) Абонент А повідомляє абоненту В, чи правильними були надіслані ним до того припущення, розповідаючи, в яку послідовність базисів він насправді використав. Це здійснюється класичними каналами зв’язку.

4) Абонент В перевіряє, що жодного обману не сталось, порівнюючи послідовність абонента А зі своїми таблицями. Для підтвердження даних необхідно, аби було повне співпадіння із таблицею, що відповідає базисам абонента А, і відсутністю співвідношення з іншою таблицею.

Alice's bit string	1	0	1	0	0	1	1	1	0	1	0	1	1	0	0
Alice's random basis								Rectilinear							
Photons Alice sends	↕	↔	↕	↔	↔	↕	↕	↕	↔	↕	↔	↕	↕	↔	↔
Bob's random bases	R	D	D	D	R	R	D	R	R	D	R	R	D	D	R
Bob's rectilinear table	1					1					0				0
Bob's diagonal table		0		1						1			0		
Bob's guess								'Rectilinear'							
Alice's reply								'You win'							
Alice sends her original bit string to certify	'1	0	1	0	0	1	1	1	0	1	0	1	1	0	0'
Bob's rectilinear table	1					1					0				0
Bob's diagonal table		0		1						1			0		

Рис 3.1.1. - Приклад роботи алгоритму BB84

Протокол BB84 завдяки своїй простоті володіє однією з найбільших ефективностей в плані генерації секретного ключа.

3.2. Протокол B92

У 1992 році Беннет пропонує протокол для розподілу квантового ключа на основі двох неортогональних квантових станів, що отримав назву B92 або протокол двох станів (Quantum Key Distribution) [11].

Протокол B92 — це протокол розподілу квантових ключів (QKD), який використовує поляризовані фотони як носії інформації. В цьому плані він є ідейним наступником протоколу BB84, з тією відміною, що використовує при роботі лише два стани замість чотирьох. Як і попередній протокол, B92 також базується на принципі невизначеності Гейзенберга.

Експериментально доведено, що протокол B92 є безумовно безпечним. Чудовий доказ безумовної безпеки протоколу B92 навів у своїй статті “Unconditionally secure key distribution based on two non orthogonal states” дослідник Кійоші Тамакі (Kiyoshi Tamaki). У ній він обґрунтував безпеку B92

навіть за умови присутність будь-якого ворога, який може виконати будь-якого роду операцію, дозволену квантовою фізикою. Таким чином безпека протоколу не може бути скомпрометована майбутнім розвитком в квантових обчислень. Втім, інші елементи даного протоколу, пов'язані з безпекою продовжують глибоко аналізуватись і обговорюватись науковою спільнотою [13].

Використання квантового каналу, який не був би повністю підконтрольний зловмиснику (абоненту С) без ризику бути виявленим, дозволяє генерувати секретний ключ з безумовним рівнем безпеки, який ґрунтувався б на законах квантової фізики. Присутність абоненту С стає очевидною для користувачів каналу через надзвичайно високий рівень помилок.

Протокол B92 припускає, що двоє законних користувачів, обмінюються інформацією за двома конкретними каналами, до яких також має доступ і зловмисник:

- класичний канал, який може бути публічним; абонент С може його прослуховувати пасивно (без виявлення);
- квантовий канал, який (за своєю природою) абонент С не може слухати без втручання у процес передачі даних.

Загалом же робота алгоритму в протоколі поділяється на дві фази.

Перша фаза B92 включає передачу даних квантовим каналом зв'язку, тоді як друга фаза відбувається через класичний канал передачі інформації.

У протоколі B92 необхідно виконати кілька налаштувань:

1) Перша фаза (квантові передачі)

а) Абонент А обирає випадковим чином послідовність бітів $A \in \{0,1\}^n$, $n>N$ (N — довжина результуючого ключа). Якщо $A_i = 0$, абонент А надсилає абоненту В стан 0 за квантовим каналом, і якщо $A_i = 1$, він надсилає йому стан "+" для всіх $i \in \{0,1,..., n\}$.

б) Абонент В у свою чергу створює випадковий вектор бітів В є $\{0,1\}^n$, n>N. Якщо $B_i = 0$, абонент В обирає базис “+” і якщо $B_i = 1$, абонент В вибирає базис “х”, для всіх і є {0,1,..., n}.

в) Абонент В вимірює відповідно кожен квантовий стан надісланий абонентом А (0 або 1) у вибраній основі (“+” або “х”).

г) Абонент В будує векторний тест Т є $\{0,1\}^n$, n>N дотримуючись наступного правила: якщо вимірювання абонента В дає 0 або “+”, $T_i = 0$, а якщо він дає 1 або “-”, $T_i = 1$, для всіх і є {0,1,..., n}.

2) Другий етап (публічний обмін даними)

а) Абонент В надсилає класичним каналом вектор Т абоненту А.

б) Абоненти зберігають лише ті біти векторів А і В, для яких $T_i = 1$. У такому випадку а за відсутності зловмисника маємо: $A_i = 1 - B_i$ і спільний необроблений ключ сформований як A_i (або і $1 - B_i$).

в) Абонент А надсилає зразок необробленого ключа абоненту В за класичним каналом зв’язку. Якщо існує *і* таке що $A \neq 1 - B_i$, тоді у мережі присутній зловмисник і спілкування переривається.

г) Спільний секретний ключ К є $\{0,1\}^n$ утворюється за допомогою необробленого ключа після порівняння зразків на кроці 2с).

Поетапний приклад роботи алгоритму протоколу B92 був наведений автором в оригінальній статті від 1992 року.

Bits chosen by Alice	$A_i = 0$				$A_i = 1$			
States sent by Alice	$\|0\rangle$				$\|+\rangle$			
Bits chosen by Bob	$B_i = 0$		$B_i = 1$		$B_i = 0$		$B_i = 1$	
Basis chosen by Bob	$\oplus$		$\otimes$		$\oplus$		$\otimes$	
Results of the measures of Bob	$\|0\rangle$	$\|1\rangle$	$\|+\rangle$	$\|-\rangle$	$\|0\rangle$	$\|1\rangle$	$\|+\rangle$	$\|-$
Probability to measure the state	1	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	0
The value of the test	**0**	**-**	**0**	**1**	**0**	**1**	**0**	**-**

Рис. 3.2.1 - Приклад виконання алгоритму B92, наведений Чарльзом Беннетом в статті “Quantum Cryprography Using Any Two Nonorthogonal States”

3.3. Протокол SARG-04

Протокол SARG-04 був розроблений групою дослідників з університету прикладної фізики у Женеві у 2004 році. Ідейними натхненниками проекту стали вчені Валеріо Скаріні (Valerio Scarani), Антоніо Асін (Antonio Acın), Грегорі Ріборді (Gregoire Ribordy) та Ніколас Ґісін (Nicolas Gisin) [14].

Протокол SARG-04 є свого роду продовженням та природнім розвитком протоколу BB84. Це стає очевидно, коли проаналізувати квантовий етап роботи протоколу (етап надсилання послідовностей поляризованих фотонів).

Головною ідеєю і мотивацією до створення протоколу стали дослідження з протидії PNS-атакам.

PNS-атаки (photon-number-splitting attack — від англ. “атака з поділу кількості фотонів”) — методика нападу на квантові мережі передачі даних, основною ідеєю яких є розділ числа поляризованих фотонів. Це дозволяє в теорії нівелювати переваги квантових каналів зв’язку, а простіше — організувати прослуховування каналу зв’язку.

Подібна небезпека виникає внаслідок використання не одиночних фотонів для передачі (умовно) одного біту інформації, а декількох, що дозволяє зменшити затухання сигналу, збільшивши тим самим відстань передачі даних.

Головна ідея протоколу полягає в публічному анонсуванні абонентом А однієї з чотирьох пар неортогональних станів, в якій би використовувались обидва базиси.

Слід відзначити, що підвищена стійкість протоколу по відношенню до BB84 може гарантуватись виключно у разі, якщо зловмисник не володіє здатністю блокувати усі посилки, що містять один або два фотони. А для посилок, складених з трьох фотонів може вимірювати два з них в різних базисах.

Може здійснюватися за тією ж технологією, що і BB84, і має на меті

No.	1	2		3	4		5		6	7		8
Sender's Random Bits	0	0		1	1		1		1	0		0
Sender's Random Basis	+	+		+	+		×		×	×		×
Sender's States	↑	↑		→	→		↘		↘	↗		↗
Receiver's Random Basis	+	×		+	×		+		×	+		×
Receiver's Possible Measurement	↑	↘	↗	→	↘	↗	↑	→	↘	↑	→	↗
Receiver's Result	0	1	0	1	1	0	0	1	1	0	1	0
Sender Announcement states	↑,↘	↑,↗	↑,↘	↑,↗	↑,↘	↑,↗	→,↘	→,↗	→,↘	→,↗	→,↘	→,↗	↑,↘	→,↘	↑,↘	→,↘	↑,↘	→,↘	↑,↗	→,↗	↑,↗	→,↗	↑,↗	→,↗
Discovered States			↑	↑				→	→				↘	↘				↗	↗			
The sifted Key			0	0				1	1				1	1				0	0			
Correct Error Discard	Discard	Discard	Discard	Correct	Correct	Discard	Discard	Discard	Discard	Correct	Correct	Discard	Discard	Correct	Correct	Discard	Discard	Discard	Discard	Discard	Discard	Correct	Correct	Discard	Discard	Discard

уникнути PNS-атак [15].

Рис. 3.3.1. - Алгоритм роботи протоколу SARG-04

Таким чином протокол SARG-04 зменшує ризики PNS-атак до мінімуму, залишаючи можливість для їх реалізації лише в декількох чітко визначених випадках.

3.4. Протокол E91

Протокол E91 був розроблений Артуром Екертом (Arthur Ekert) в 1991 році. Іншою назвою протоколу — є абревіатура EPR (Einstein-Podolsky-Rosenberg), сформована з імен авторів на парадоксу Ейнштейна-Подольського-Розенберга.

На відміну від двох попередніх, даний протокол не є прямим наступником BB84, використовуючи при своїй роботі дещо відмінний принцип.

Робота протоколу передбачає використання пари фотонів, які утворюються в антисиметричних поляризаційних станах. Перехоплення одного з них не може дати потенційному зловмиснику жодної інформації, у той час як для абонентів A і B є сигналом про те, що лінія зв'язку була дискредитована.

Ефект EPR виникає, коли симетричний атом випромінює два фотони в сторону двох спостерігачів (абонентів). Фотони, що при цьому випромінюються, мають невизначену поляризацію, однак в силу симетрії їх поляризації завжди протилежні.

Вкрай важливим в даному явищі є те, що поляризація фотонів стає очевидною тільки під час вимірювання.

На основі ефекту EPR Екертом був запропонував протокол, який гарантує безпечний розподіл секретного ключа між абонентами зв'язку.

Протокол Екерта передбачав окрім наявності стандартних абонентів існування нейтрального генератора, який би й здійснював генерацію протилежно поляризованих фотонів.

Загалом алгоритм виконання протоколу має наступний вигляд:

а) Генератор поляризованих фотонів одночасно відправляє абонентам A і B послідовності протилежно поляризованих фотонів

б) Абоненти випадковим чином обирають послідовність поляризаторів, якою відбуватиметься "зчитування" інформації з отриманих від генератора фотонів.

в) Провівши поляризацію отриманої послідовності, абоненти утворюють власний вектор значень, що складатиметься з порядкових номерів поляризаторів, в яких успішно відбулось “зчитування” даних від генератора.

г) Отриманими векторами абоненти обмінюються через класичні канали зв’язку. Спільні значення у векторах стають спільним секретним секретним ключем абонентів.

Ефективність даного протоколу є значно нижчою в порівнянні з попередніми, однак є значно стійкішим до спроб дискредитації.

3.5. Протокол KMB-09

В оригінальному протоколі BB84 Беннета і Брассарда, прослуховувач може бути виявлений, оскільки його спроби перехопити інформацію призводять до квантової бітової помилки (QBER — Quantum Bit Error Rate), імовірність якої є не меншою за 25%. Для того, щоб подолати цю вразливість у 2009 році групою дослідників був розроблений альтернативний протокол розподілу квантових ключів, у якому абоненти A і B використовують дві взаємно неупереджені основи, однією яких кодується "0", а іншою "1" [16].

Безпека розробленого протоколу, що отримав назву KMB-09 (Khan, Murphy, Beige), обумовлена мінімальним показником похибки передачі індексу (ITER — Index Transmission Error Rate), якій створюється наявністю підслуховувача і значно зростає для фотонних станів більш високих розмірів.

Це дозволяє отримати більше шуму в лінії електропередачі, збільшуючи тим самим можливу відстань між абонентами A і B без потреби в проміжних вузлах.

Перед початком роботи протоку повинно відбутись налаштування самої системи, в якій для обох абонентів мають бути визначені значення, якими

закодовуватиметься інформація у відповідності до анонсованого абонентом А

Index announced by Alice	States measured by Bob							
	$\|e_1\rangle$	$\|e_2\rangle$	$\|e_3\rangle$	$\|e_4\rangle$	$\|f_1\rangle$	$\|f_2\rangle$	$\|f_3\rangle$	$\|f_4\rangle$
1	×	1	1	1	×	0	0	0
2	1	×	1	1	0	×	0	0
3	1	1	×	1	0	0	×	0
4	1	1	1	×	0	0	0	×

індексу.

Рис. 3.5.1. - анонсовані абонентом А індекси та їх означення в процесі кодування-декодування (поляризації) фотонів

Послідовність індексів, що відповідають поляризації безпосередньо кожного сигналу (фотону або групки фотонів) відправляється абонентом А незахищеним каналом зв’язку.

Паралельно із цим квантовим каналом зв’язку надсилається послідовність поляризованих сигналів, які декодуються абонентом В безпосередньо в спільний секретний ключ обох абонентів.

Table II
WORKING OF KMB09

Alice's random bits	1	0	1	1	1	1	0	1	0	1	1	0	1	1	0	0	1	0	1	1
Alice's random bases	f_1	e_1	f_2	f_1	f_1	f_1	e_2	f_1	e_2	f_2	f_2	e_2	f_2	e_1	e_2	e_1	f_1	f_2	e_2	e_1
Alice's index '*i*'	1	1	2	1	1	1	2	1	2	2	2	2	2	1	2	1	1	2	2	1
Bob's random bases	f_1	f_2	f_2	f_2	e_1	e_1	e_1	e_2	f_1	e_2	f_2	e_1	f_2	f_1	e_2	f_2	f_1	e_2	e_1	f_1
Bob now verify his index's with the one announced by Alice using table II																				
Alice announces her indices	1	1	2	1	1	1	2	1	2	2	2	2	2	1	2	1	1	2	2	1
Bob's Inter-pretations	×	0	×	0	×	×	1	1	0	×	×	1	×	×	×	0	×	×	1	×
Bits resulting to ITER				ITER			ITER					ITER								
Bit received correctly		0						1	0							0			1	

Рис. 3.5.2. - алгоритм роботи протоколу KMB-09

Протокол KMB-09 не є простим узагальненням класичного протоколу BB84. Він вирішує одразу декілька серйозних проблем, що постають перед квантовою криптографією, а саме:

- мінімізує можливість прослуховування мережі при використанні не однофотонних сигналів;
- позбавляє необхідності в створенні додаткових вузлів у мережі, які б збільшували дальність дії протоколу;
- захищає мережу від PNS-атак;

Таким чином можемо бачити, що даний протокол є одним з найбільш ефективних та затребуваних рішень в сучасній квантовій криптографії.

3.6. Висновки з розділу

В даному розділі були розглянуті найбільш відомі й найліпше досліджені на даний момент протоколи квантової криптографії. Кожен з них володіє цілком очевидними перевагами та недоліками, які пов'язані як з технічними (проблема

відстаней при використанні одиничного фотону в якості сигналу), так і з безпековими (імовірність прослуховування та PNS-атак з боку зловмисників) питаннями.

Спроби розв'язання останніх призводять до частинних результатів, однак одночасно і до деяких інших наслідків, зокрема ускладнення роботи самих протоколів та зниження ефективності генерації ключів [17].

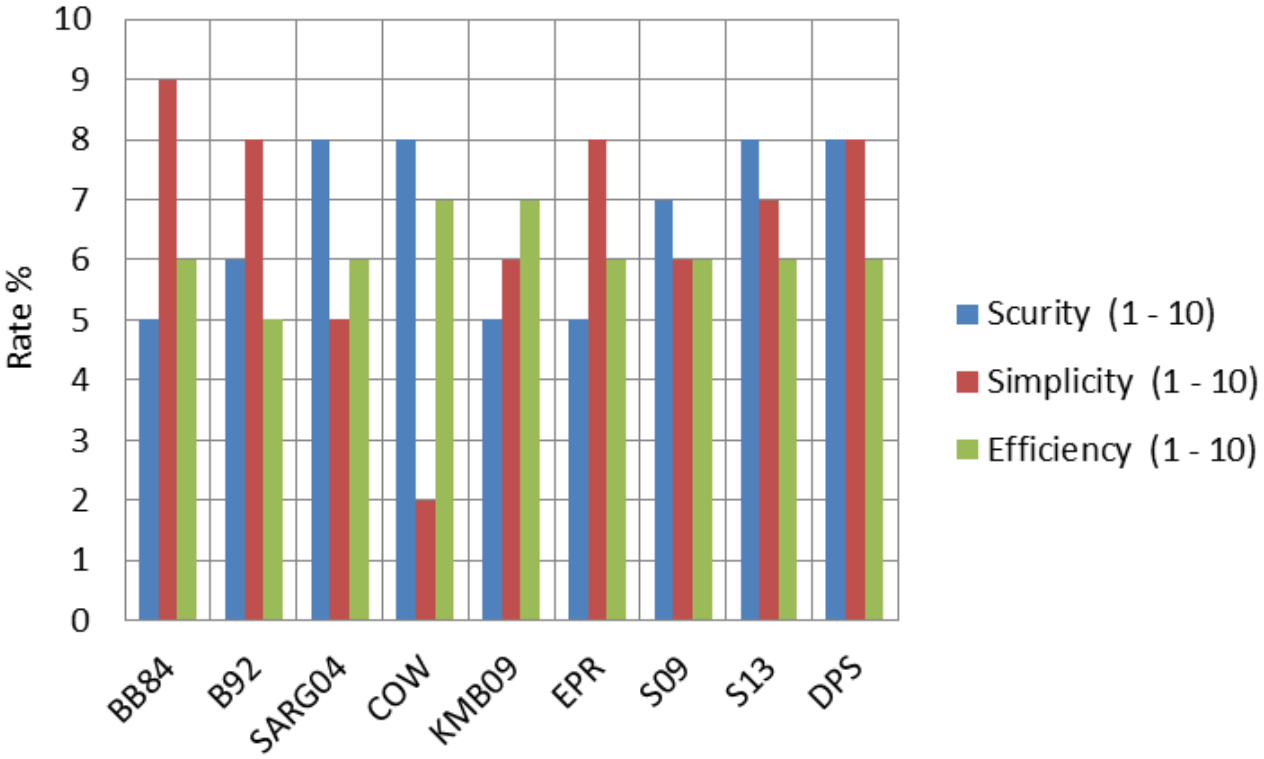


Рис. 3.6.1 - порівняння найпоширеніших протоколів квантового розподілу секретного ключа

Перший наслідок є цілком закономірним, враховуючи, що по суті більшість сучасних протоколів є розвитком протоколу Беннета-Брасарда, а отже не можуть бути простішими як для розуміння, так і для технічної реалізації продуктами.

Другий наслідок є направду дещо суперечливим, адже питання ефективності генерації секретного ключа — отримання довшого ключа від послідовності передаваних сигналів — виглядає наразі не надто актуальним.

Щонайголовніше через те, що швидкість світла дозволяє проводити практично миттєві операції з найбільшою кількістю сигналів. Це знаходить своє підтвердження в статистиці новітніх протоколів, розробники яких поступаються швидкодією (ефективністю та простотою) заради підвищення рівня секретності нових протоколів.

ПОСТАНОВКА ЗАДАЧІ

Розглянувши проблематику вивчення квантової криптографії в сучасних умовах та основні протоколи, що були розроблені й знайшли своє застосування в добі останніх тридцяти п’яти років, розглянемо основні характеристики, що повинно мати програмне рішення для повноцінного вирішення поставленого ряду питань і викликів.

1) Доступність

Одним з основних питань, що вже були підняті у зв’язку з означеною проблематикою є питання доступності як навчально-методичних матеріалів, так і демонстраційно-тренувальних комплексів для студентів навчальних закладів та незалежних дослідників. Цілком очевидно, що це є питання в тому числі й поширення продукту в зацікавленому середовищі. При чому поширення максимального.

Для цього цілком доречним є розміщення розроблюваного додатку (програмного рішення) в мережі Internet, зокрема на серверах університетів (наприклад, сервері кафедри “ЕОМ” ДНУЗТ) або веб-хостингах, які б відповідали технічним вимогам для повноцінного функціювання розміщеної системи (наприклад сервіс GitHub).

Це дозволило б використовувати навчальну систему в будь-якій точці України та загалом земної кулі, у якій наявний доступ до мережі Internet.

Іншим важливим моментом, який безперечно входить до сфери доступності, є мова веб-додатку і супутніх із ним методичних матеріалів, які дозволили б вільно використовувати систему в навчальних цілях. Тому в якості основної мови системи буде використана українська. При подальшому розвитку додатку, зокрема для його використання іноземними студентами, можливе додавання англійської як додаткової мови для відображення інтерфейсу. Додавання інших мов було б цілком зайвим і геть невиправданим з точки зору ефективності роботи додатку кроком, який би лише збільшував об’єми коду і забирав би значну частину часу для написання та своєї імплементації в продукт не додаючи до системи якихось критично необхідних новацій чи функцій.

2) Повнота і актуальність

Розгляд одного з криптографічних протоколів для повноти знання має бути процесом всестороннім і комплексним. Повинен включати в себе, як ознайомлення з теоретичними матеріалами, оригінальними статтями науковців-

дослідників, так і практичну частину, яка б передбачала візуалізацію роботи алгоритмів.

Для цієї мети доречним виглядає поєднання безпосередньо самого візуального ряду, який способом анімації відображав би виконання етапів алгоритмів, та текстового опису, котрий слугував би поясненням до того чи іншого етапу виконання алгоритму. Подібного роду методичні матеріали виконуватимуть функцію наочного посібника та спрощеного (без нагромадження науковою термінологією) переказу основних ідей, принципів і прийомів, які були покладені в основу протоколів.

Водночас для реалізації повноти і актуальності знання необхідно врахувати, що знання не може бути повноцінним, обмежуватись розумінням єдиного протоколу (наприклад, BB84) без ознайомлення з новішими алгоритмами. Це означає, що вкрай необхідним є розробити візуалізації та практичні завдання до ряду протоколів, як основних і базових (на кшталт BB84 чи E91), так і до новіших, які є нині використовуваними і актуальними. Це потенційно може дати хоча б мінімальне бачення розвитку самої галузі квантової криптографії, а отже є цілком доречним і органічним в межах цілі, описаної в даному пункті.

3) Практичність

Практичність знань і навичок нині є однією з тих рис, що цінуються найбільше. Саме тому сухе знання, навіть підкріплене візуальним рядом, не може вважатись повноцінним і цілісним без вміння його застосовувати на практиці.

Цей факт робить цілком очевидною необхідність розробки серед іншого практичних субдодатків, які б перевіряли (а на ділі формували) знання і навички студента в умовах роботи протоколів квантової криптографії,

наближених до умов реального часу, себто так, як вони працюють безпосередньо на рівні прийому-передачі та обробки сигналів.

Такі практичні тренажери необхідно розробити для кожного окремо розглянутого протоколу квантової криптографії.

Підсумки

Метою даного дипломного проекту є розробка веб-додатку, який мав би містити:

- опис п’яти найбільш актуальних протоколів квантової криптографії на момент 2020 року;
- візуалізацію роботи кожного з них з доступними поясненнями кожного етапу виконання алгоритмів;
- посилання на дослідження і детальний опис кожного з протоколів квантової криптографії, описаних в даному дипломному проекті;
- практичні завдання, які б перевіряли знання і навички, здобути в ході ознайомлення з принципами роботи протоколів квантової криптографії.

РОЗРОБКА ВЕБ-ДОДАТКУ

5.1. Загальна структура веб-додатку

Розробка веб-додатку, який би повністю відповідав поставленій задачі вимагає попередньо створення мапи чи загальної схеми, яка в спрощеній формі показувала б основні функціональні й практичні конструкти, які обов’язково мають бути присутні для реалізації основних завдань.

В найзагальнішій формі окреслимо цю схему як основний додаток, який включатиме в себе основні симуляції, описи алгоритмів і практичні завдання, та набір практичних завдань, які входять до першої згаданої частини, але існують як незалежні від неї частини, основною метою яких є напрацювання (тренування) практичних навичок по роботі з алгоритмами протоколів.

Умовно можна сказати, що друга частина частково дублює функції першої, однак робить це виключно з міркувань практичної зручності.

Відповідно щодо фактичної структури додатку цілком доречно використати наступну ієрархію:

- Основний додаток (включає в себе всі теоретичні, демонстраційні та практичні матеріали)
- Практична реалізація протоколу BB84
- Практична реалізація протоколу B92
- Практична реалізація протоколу E91
- Практична реалізація протоколу SARG-04
- Практична реалізація протоколу KMB-09

Таким чином ми бачимо, що всі основні блоки додатку (всі html-файли) є присутні на одному рівні, себто ієрархія є однорівнева.

Говорячи про функціональну ієрархію схема є дещо складнішою.

1. Основний додаток
    - Головний екран (меню з посиланнями на всі функціональні блоки - сторінки)
    - Вступ (короткий опис основних принципів квантової криптографії)
    - Опис і демонстрація роботи алгоритму протоколу BB84 (є прописані в файлі основної програми)

- Опис і демонстрація роботи алгоритму протоколу B92 (є прописані в файлі основної програми)
- Опис і демонстрація роботи алгоритму протоколу E91 (є прописані в файлі основної програми)
- Опис і демонстрація роботи алгоритму протоколу SARG-04 (є прописані в файлі основної програми)
- Опис і демонстрація роботи алгоритму протоколу KMB-09 (є прописані в файлі основної програми)

2. Практична реалізація протоколу BB84 (реалізовано в окремому файлі)

3. Практична реалізація протоколу B92 (реалізовано в окремому файлі)

4. Практична реалізація протоколу E91 (реалізовано в окремому файлі)

5. Практична реалізація протоколу SARG-04 (реалізовано в окремому файлі)

5. Практична реалізація протоколу KMB-09 (реалізовано в окремому файлі)

Бачимо, що функціональна ієрархія ускладнюється і стає дворівневою, адже функції, що є реалізовані в тілі основної програми, дублюються в окремих модулях для того, аби користувач мав змогу використовувати їх незалежно від решти, тим самим отримуючи максимально доречний для своїх цілей матеріал.

Відповідно до цього робимо висновок щодо того, з яких програмних модулів повинен складатись додаток: з 6 html-файлів, в одному з яких буде прописана основна програма з посиланнями на інші 5 модулів, що реалізовуватимуть функції симуляції роботи алгоритмів або виконуватимуть альтернативні завдання.

Крім того на сервері (у репозиторії) додатку розмістимо css-файл, що міститиме опис усіх стилів, які застосовуватимуться в тілі програмних модулів. Враховуючи те, що реалізації візуальних стилів мають бути ідентичними для

всіх сторінок, зробимо css-файл спільним для всіх файлів (єдиним для всіх), що дозволить зменшити об’єм необхідної для додатку пам’яті — по суті зменшить розмір самого додатку.

Те саме зробимо з javascript-кодом, залишивши мінімальні його фрагменти всередині самих файлів для практичних потреб.

Підводячи підсумки, можемо побачити, що загалом проект додатку складатиметься з 13 файлів, серед яких:

- index.html — файл тіла основної програми
- BB84.html – файл практичної реалізації протоколу BB84
- B92.html – файл практичної реалізації протоколу B92
- E91.html – файл практичної реалізації протоколу E91
- SARG-04.html – файл практичної реалізації протоколу SARG-04
- KMB-08.html – файл практичної реалізації протоколу KMB-08
- styles.css – файл з описом стилів в проекті
- index.js — файл із описом функціональної частини сторінки index.html
- BB84.js – файл із описом функціональної частини сторінки BB84
- B92.js – файл із описом функціональної частини сторінки B92
- E91.js – файл із описом функціональної частини сторінки E91
- SARG-04.js – файл із описом функціональної частини сторінки SARG-04
- KMB-08.js – файл із описом функціональної частини сторінки KMB-08

5.2. Технології для розробки

Перш ніж приступити до безпосередньої розробки веб-додатку оберемо технології, що їх ми застосовуватимемо в процесі написання програми.

Основними і безальтернативними мовами програмування при розробці даного веб-додатку вважатимемо мову розмітки HTML (Hyper Text Marker

Language), мову стилів CSS (Cascading Style Sheets) та мову сценаріїв JS (JavaScript). З усіх версій обраних мов оберемо CSS3, HTML5 і ES5.

Для більш ефективної розробки мовою JavaScript використаємо декілька додаткових бібліотек, які будуть динамічно підключатись в процесі завантаження додатку в браузері. Зокрема для скорочення об'ємів коду і спрощення його написання використаємо бібліотеку jQuery, для підключення якої в програмі необхідно буде прописати посилання на розміщення файлу бібліотеки на сайті розробника в описовій частині файлу (в тегу "header"):

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
</script>
```

Для кращої візуалізації даних, а саме для створення анімацій всередині слайдів, підключимо бібліотеку AnimeJS:

```
<script src="https://animejs.com/lib/anime.min.js"></script>
```

А також спеціальну бібліотеку для створення анімованих графіків на системі координат ChartJS:

```
<script src="https://www.chartjs.org/dist/2.9.3/Chart.min.js"></script>
<script src="https://www.chartjs.org/samples/latest/utils.js"></script>
```

Серед інших технологій, що будуть необхідні в процесі розробки веб-додатку, варто відзначити хостинг GitHub, на якому буде розміщений додаток, завдяки чому буде доступний практично в будь-якій точці світу, де наявний інтернет-зв'язок. Опісля закінчення процесу розробки додаток буде розміщений в репозиторії на GitHub, де буде доступний у відкритому вигляді його код і безпосередньо сам працюючий додаток за адресою:

https://olexandrbelyaev.github.io/

5.3. Основна програма

Розглянемо структуру основної програми (сторінки index.html). Для цього слід попередньо визначити, як і з яких частин має складатись сама програма, а вірніше підпрограма.

Найлогічнішим в даному питанні буде розділити підпрограму на три не вірні між собою частини:

- вступ, що складатиметься з головного екрану, на якому буде короткий опис додатку, посилання на симуляції роботи протоколів квантової криптографії та кнопка для початку загального уроку (основної частини підпрограми);
- основна частина складатиметься з попереднього екрану, на якому будуть розписані поняття квантової криптографії та деяких її принципів, екранів описів протоколів квантової криптографії з демонстраціями прикладів їх роботи на практиці, та з практичних завдань, що йтимуть після екранів опису і демонстрацій;
- підсумкова сторінка міститиме результати п’яти практичних завдань і посилання на початковий екран, наукові статті з описами принципів роботи протоколів квантової криптографії та посилання на окремі модулі з практичними реалізаціями протоколів (симуляторами).

Подібна структура почергової зміни екранів у вікні веб-браузера по суті є структурою додатку типу SinglePage – себто демонстрація нових сторінок має відбуватись без відкриття нових сторінок в браузері чи перезавантаження старих.

Найбільш оптимальною реалізацією цього стандарту в нашому випадку вважатимемо створення структури, що реалізувала б формат презентації, яка складається з послідовності слайдів, з яких лише один може бути відображений на екрані у той час як усі інші залишаються невидимими.

Перехід же до іншого (наступного) слайду зробимо можливим за умови відкриття усіх матеріалів на слайді (ознайомлення з усією необхідною інформацією). Зробимо це через механізм зникаючої кнопки, яка зникатиме щоразу, як на неї клацнуть, а з'являтиметься щоразу як на екрані буде повністю показана вся необхідна інформація або практичне завдання буде коректно виконане.

Для того, щоб реалізувати такого роду слайдову конструкцію пропишемо блочну структуру, яка являтиме собою контейнер для усіх екранів і перемикатиме їх у відповідності до описаних вище умов.

Приклад :

```
<div id="main" class="flex-container">
    <div id="0">
    </div>
</div>
```

Прописуємо в класі "flex-container" властивості контейнеру, які дозволили б повністю заполонити йому майже увесь простір екрану, окрім невеликої смуги в самому низу для того, щоб розмістити там кнопку переключання слайдів. Робимо це визначивши висоту контейнеру, як "height:87vh;". Визначаємо спосіб відображення контейнерного елементу на екрані як "display: flex", а всі відступи навколо нього прирівнює до нуля "margin:0".

За замовчуванням перший (нульовий) слайд програми робимо видимим, у той час як решту аналогічно написаних слайдів невидимими, прописуючи у їх властивостях значення "display: none". Таким чином, разом із нульовим слайдом створюємо ще 12 слайдів.

Приклад :

```
  <div id="1" style="display: none;">
```

```
</div>
```

Заповнюємо нульовий слайд інформацією про додаток та додаємо всередину нього спеціальний контейнерний елемент, всередині якого зберігатимуться блоки з посиланнями на симуляції роботи протоколів квантової криптографії. Задаємо блоку властивість відображення на екрані як “display:flex” і базову ширину кожній комірці (по суті кожному посиланню) як “flex-basis:15vw;”, що еквівалентно 15% від ширини екрану монітора, на якому промальовується картинка слайду.

Додаємо на слайд кнопку переходу до початку роботи основної програми, а вірніше до наступного слайду. В сценарії роботи кнопки при її натисканні прописуємо наступний код:

```
function start(){
  $(document).ready(function(){
      $("#0").hide();
      $("#1").show(1000);
      $("#add").show(1000);
  });}
```

При цьому використовуємо бібліотеку jQuery для мінімізації коду всередині js-файлів.

Варто зауважити, що цю кнопку додаємо виключно для того, щоб користувачу могли бути показані слайд, який має ідентифікатор “1” та кнопка переходу до наступних слайдів. Сама ж натиснути кнопка зникає і в подальшій роботі програми участі вже не бере.

Додаємо візуальні стилі до основних елементів слайду, зокрема до посилань на інші сторінки проекту, кнопку переходу і наявний на слайді текст.

В результаті маємо перший слайд чи то пак меню додатку.

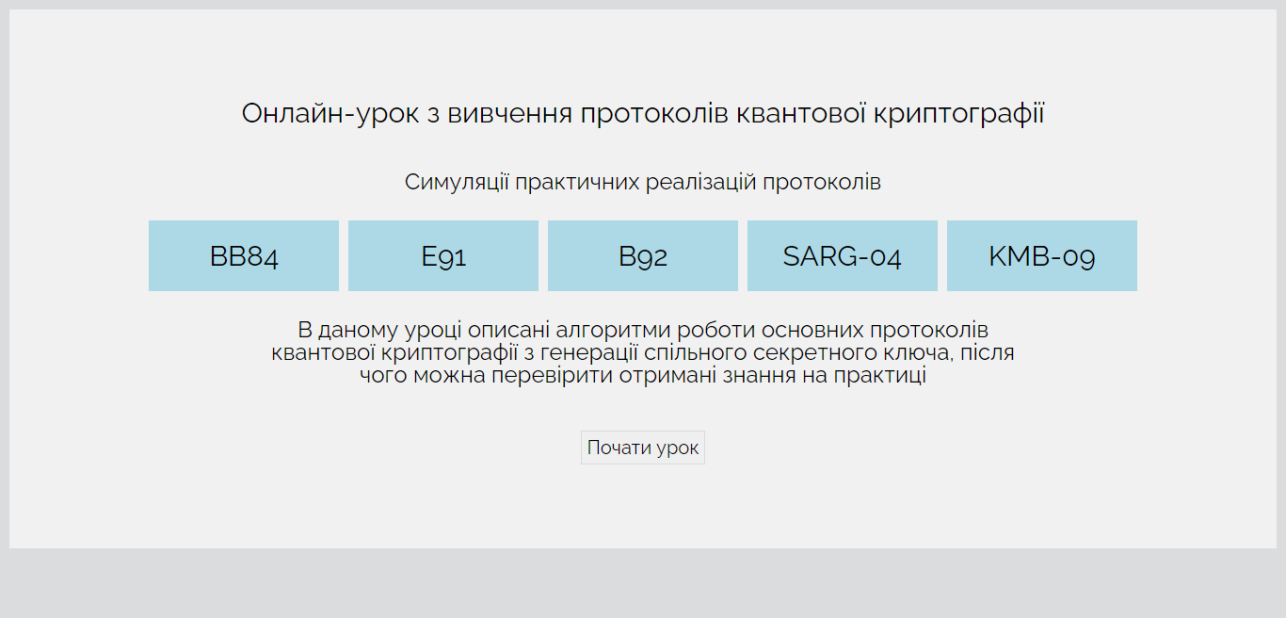


Рис. 6.1.1. - зовнішній вигляд початкового екрану додатку

5.3.1. Сторінка прологу

Другим слайдом додатку є так звана сторінка прологу. Заносимо у неї відомості про ідеї та принципи роботи протоколів квантової криптографії. Робимо це, додаючи стоковий тег <p> і прописуючи стилі всередині класу “text”.

Сторінка прологу є першим слайдом, попід яким з’являється кнопка переходу до наступних слайдів. Пропишемо сценарій зникнення даної кнопки при її натисканні та загалом сценарій для переходу від одного слайду до іншого.

Попередньо створюємо змінну “f” значення якої за замовчанням буде дорівнювати нулю. Значення цієї змінної позначатиме те, який слайд нині є видимим, що відповідно означатиме, що всі інші мають бути невидимими.

```
var f=1;
function add(){
    $(document).ready(function(){
    if (f!=12){
        $("#"+f).css("display","none");
```

```
            $("#"+(f+1)).css("display","block");
            $("#add").hide();
            f++;}
        });
    }
```

Завдяки описаному сценарію ми можемо робити невидимим активний слайд, майже одночасно з цим виводячи на екран наступний, просто змінюючи їх властивості “display” цих елементів. При цьому робимо кнопку переходу між слайдами невидимою, аби користувач не міг пропустити інформацію чи завдання на наступному слайді.

Бібліотека jQuery мови JavaScript дозволяє робити зникнення та появу елементів більш плавними, додаючи анімацію при виконанні коду сценарію. Це дозволяє робити додаток візуально привабливішим і менш статичним, що є важливим для сприйняття користувачами.

5.3.2. Сторінка ознайомлення з протоколом BB84

Даний слайд, що йде у програмі під ідентифікатором “2” є першим слайдом, що виконує функцію демонстрації роботи алгоритму протоколу. В цьому випадку протоколу BB84.

Для реалізації цієї функції розіб’ємо роботу зі слайдом на декілька етапів, кожен з яких розповідатиме про певну стадію виконання протоколу BB84. Це дозволить дозовано доносити інформацію до користувача, що є кращим для її розуміння та засвоєння.

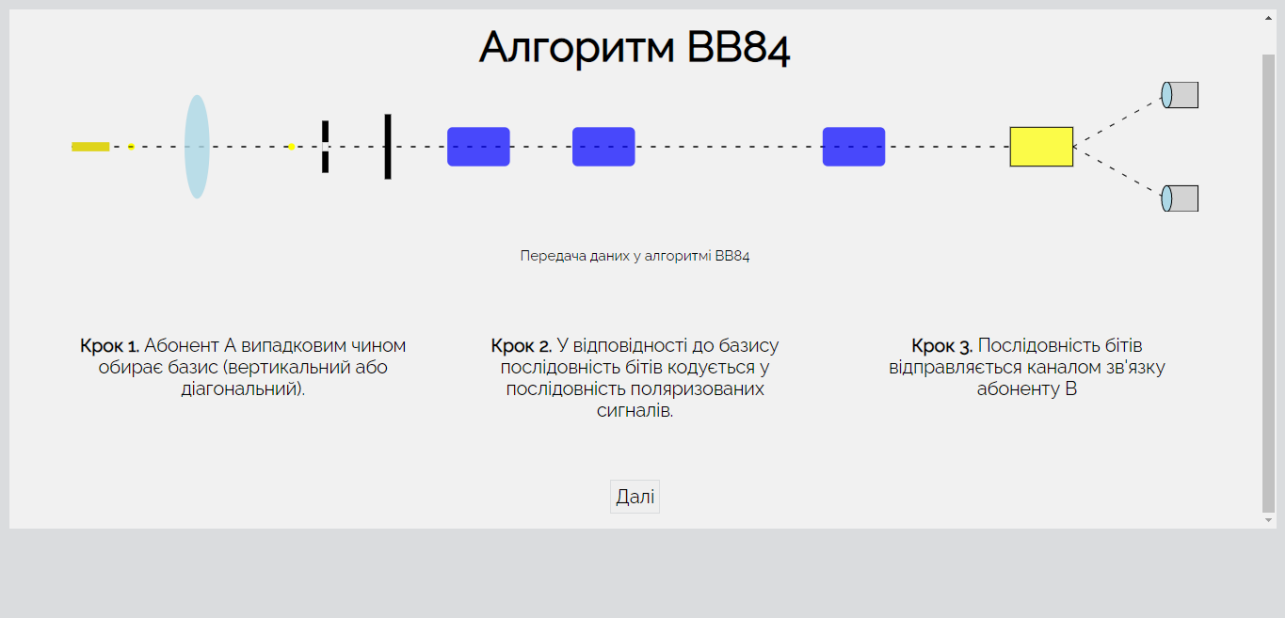
Усього слайд міститиме 4 етапи.

Перший етап розпочинається з переходу від слайду прологу до безпосередньо даного. Він містить анімовану умовну візуалізацію роботи протоколу на обладнанні. Анімацію створюємо за допомогою підключеної

заздалегідь бібліотеки AnimeJS. Одразу під елементом з анімацією розміщуємо опис перших трьох дій, що реалізуються при роботі протоколу. Опис кожного кроку вміщуємо в одному реченні і додаємо в горизонтальний контейнерний елемент.

Під цим елементом розміщуємо кнопку “Далі”, що її натискання завершує перший етап роботи зі слайдом і розпочинає другий.

Друга стадія роботи зі слайдом додає перед кнопкою “Далі” два набори по вісім елементів вводу <textarea>, перший з яких заповнений обраними системою в довільном порядку значеннями “1” або “0”. Це є імітацією генерації абонентом А випадкової послідовності бітів для їх подальшої поляризації та відправлення абоненту В. Це демонструється в другому наборі елементів вводу, в якому відображаються значення поляризованих у відповідності до набору



бітів фотонів.

Рис. 6.5.1. - зовнішній вигляд першого етапу 3-го слайду веб-додатку

Опісля наборів елементів вводу розміщуємо опис наступних трьох кроків реалізації алгоритму протоколу BB84. Робимо це в стилі опису попередніх трьох кроків.

Візуалізувавши другий етап слайду перед кнопкою “Далі”, ми зберегли її положення внизу слайду, що дозволило зберегти інтуїтивний інтерфейс сторінки і з наступним натисканням на неї завершити другий етап, перейшовши до третього.

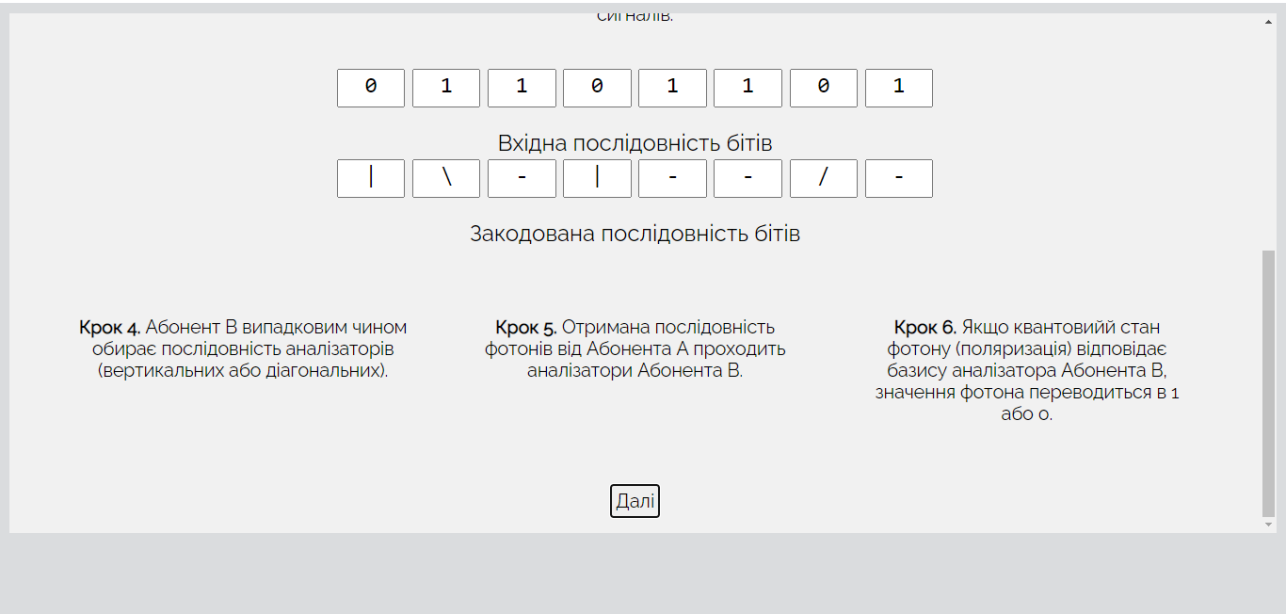


Рис. 6.5.2. - зовнішній вигляд другого етапу 3-го слайду веб-додатку

Третій етап реалізує описані раніше кроки з четвертого по шостий, візуалізуючи в наборі елементів вводу послідовність випадкових базисів, обраних абонентом В та результати вимірювання з їх допомогою отриманої послідовності поляризованих фотонів від абоненту А. Послідовність декодується і абонент В отримує секретний ключ.

На цьому третій етап виконання слайду завершується і з натисканням на кнопку “Далі” починається четвертий.

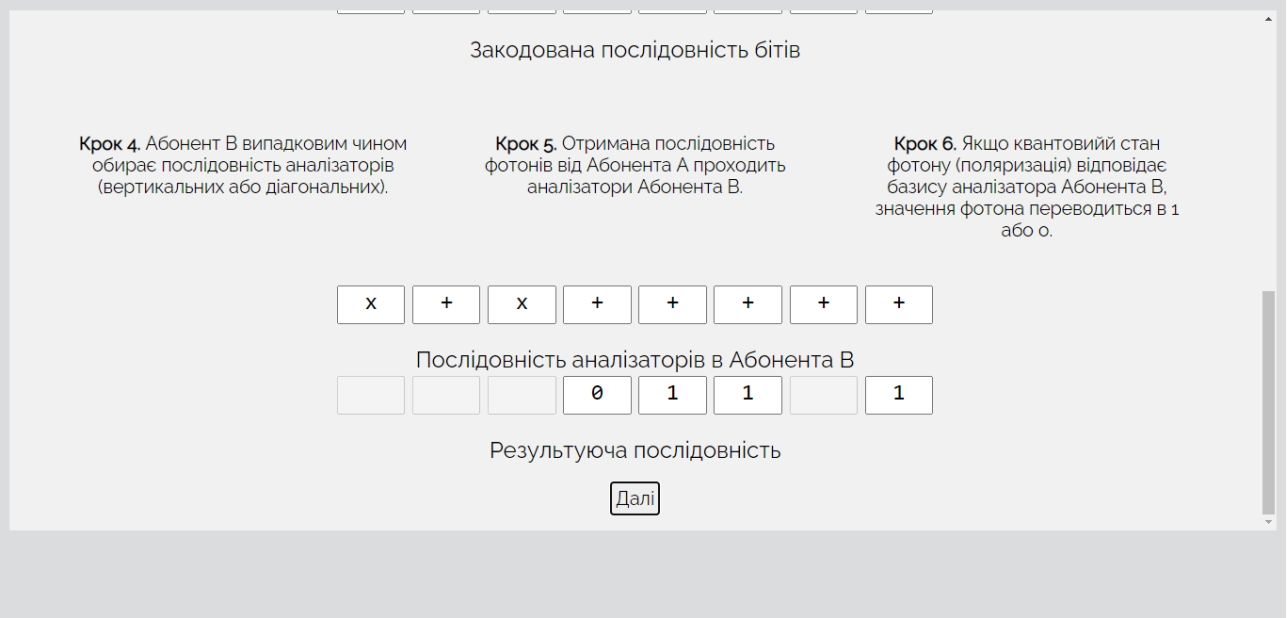


Рис. 6.5.3. - зовнішній вигляд третього етапу 3-го слайду веб-додатку

Четвертий етап повідомляє користувача, що абонент В повідомляє відкритим каналом зв’язку про номери успішно виміряних фотонів в послідовності, таким чином формуючи секретний ключ і на стороні абонента А. При цьому кнопка “Далі” зникає і з’являється кнопка переходу до наступного слайду основної програми.

Кожне наявне в даному слайді значення, яке має обиратись абонентом, обирається браузером випадково, але відповідно до заданих умов, через що кожного разу, як відкриватиметься даний слайд, користувач бачитиме новий варіант результатів роботи протоколу. Це дозволяє цілісно ознайомити користувача з роботою протоколу, не зациклюючи його на одному, заздалегідь прописаному варіанті розвитку подій.

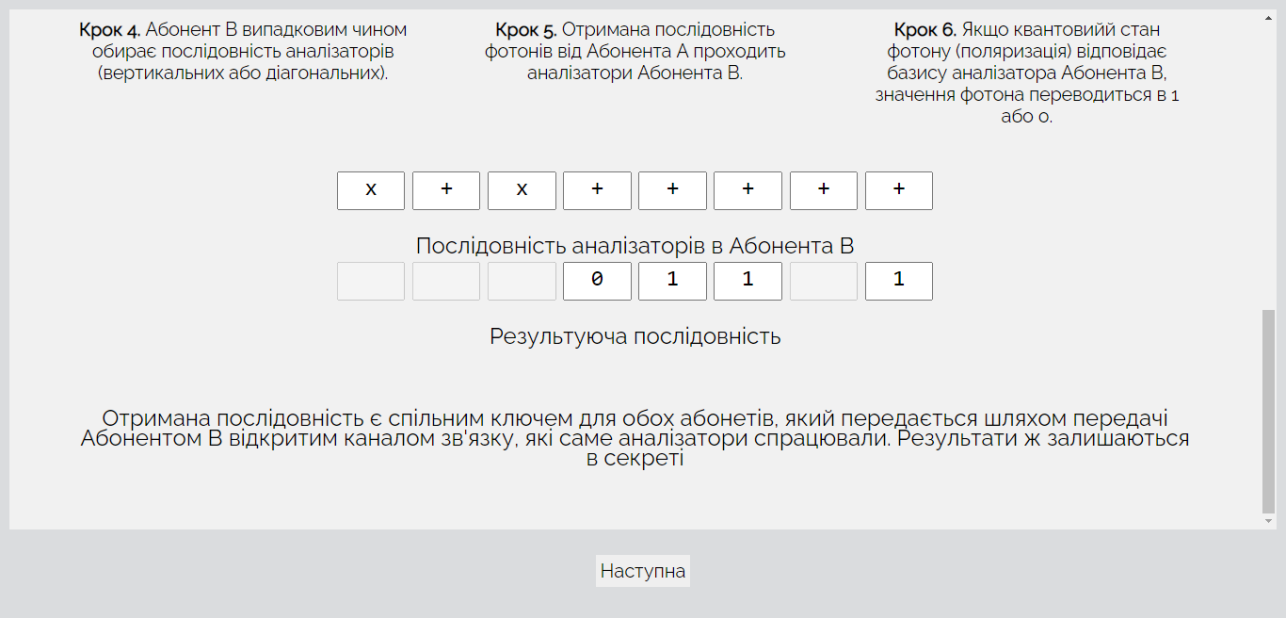


Рис. 6.5.4. - зовнішній вигляд четвертого етапу 3-го слайду веб-додатку

5.3.3. Практичне завдання №1

Сторінка практичного завдання №1 реалізує генерацію спільного секретного ключа між абонентами в ручному режимі, який має здійснюватись з безпосередньою взаємодією з користувачем, який має вручну закодовувати бітову послідовність у відповідності до умов, згенерованих браузером та прописаних у завданні.

Робота даного слайду, як і попереднього, розбита на декілька етапів.

Першим етапом є введення користувачем вручну значень поляризованих у відповідності до наведених браузером базисів. Усього можливі 4 варіанти вводу: “-”, “/”, “\”, “|”.

По натисканні кнопки “Далі” відповіді, записані в послідовність елементів вводу, перевіряються у відповідності до алгоритму роботи протоколу BB84 і підсвічуються зеленим або червоним кольором у відповідності до того правильно було закодоване значення чи ні. Неправильно закодовані значення відкидаються та більше не беруть участі в завданні. Натомість з’являються послідовність базисів поляризації абонента В і нові елементи вводу, до яких

слід занести “T” (чи “T”) або “F” (чи “f”) як результат аналізу того, чи вдало відбувся аналіз отриманої послідовності фотонів у відповідності до базисів абонента B.

Результати за натисканням кнопки “Далі” знову перевіряються і теж підсвічуються відповідним кольором в разі правильної або хибної відповіді. По наступному натисканні кнопки “Далі” з’являється поле вводу, до якого необхідно записати декодований результат роботи протоколу — отриманий секретний ключ і натиснути кнопку “Далі”.

Усі результати, отримані в ході практичного завдання збережуться і будуть виведені на останньому слайді опісля проходження всієї основної програми.

Даний слайд є програмною основою для модуля симуляції роботи протоколу BB84, до якого можна перейти зі слайду меню для напрацювання практичних навичок і кращого розуміння конкретно цього протоколу квантової

криптографії.

Рис. 6.6.1. - приклад роботи зі слайдом практичної роботи №1

Рис. 6.6.2. - приклад роботи зі слайдом практичної роботи №1

5.3.4. Сторінка ознайомлення з протоколом B92

Наступний слайд є демонстрацією роботи алгоритму протоколу B92. Враховуючи, що він багато в чому є повторенням протоколу BB84 з тією лише різницею, що він використовує додатково неортогоналні, кругові базиси поляризації, цілком надмірним буде розробляти демонстраційний механізм, аналогічний до того, що був наведений на слайді ознайомлення з протоколом BB84. Замість цього обмежимось статичною демонстрацією з восьми бітами, які відповідають усім можливим варіантам поляризації в протоколі B92.

Тим не менш і тут ми використовуємо принцип дозованої подачі інформації та, як і в попередніх слайдах, розбиваємо слайд на чотири етапи виконання.

Перший етап стандартно розпочинається з переходом до самого слайду і демонструє опис історії появи протоколу, таблицю з базисами, що використовуються в даному протоколі й початок процесу формування секретного ключа між абонентами. При чому візуально розділяємо сторінку на

дві частини, аби, застосовуючи принципи інтуїтивного інтерфейсу, користувач мав змогу відстежувати дії на стороні кожного з абонентів.

Перший крок, який, власне, є частиною першого етапу виконання слайду містить таблицю на стороні абонента А, в якій прописані випадкова послідовність бітів (чотири нулі та чотири одиниці), випадкова послідовність базисів (чотири перпендикулярних і чотири кругових) і результати поляризації у відповідності до обраних базисів.

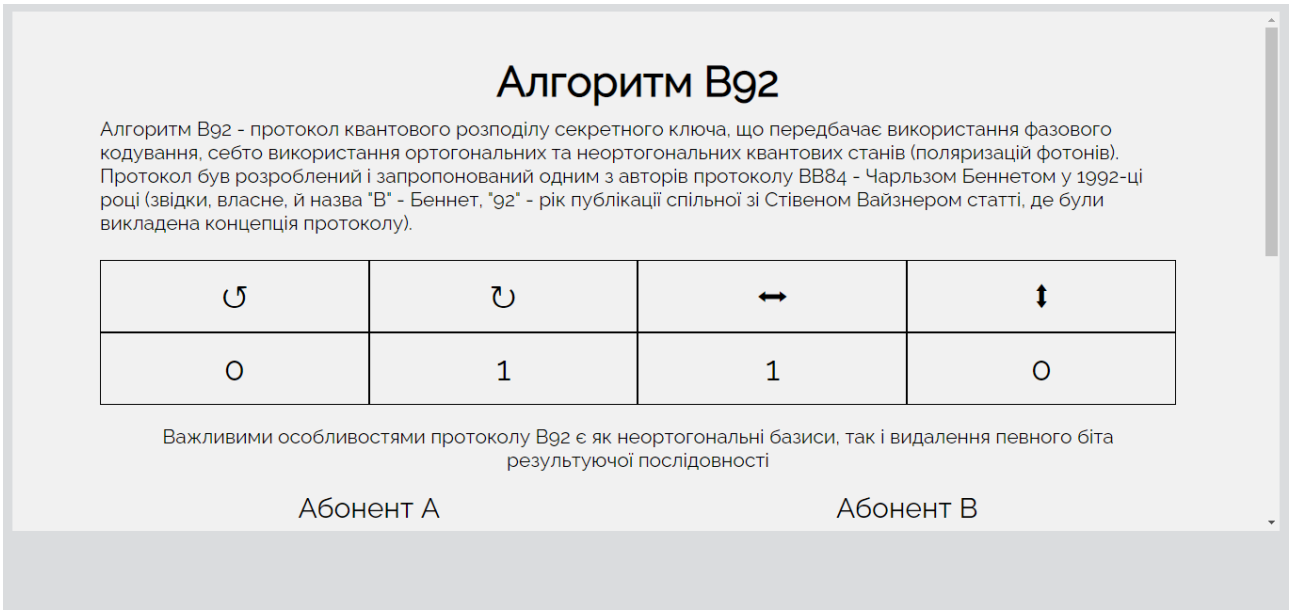


Рис. 6.7.1. - перший етап роботи демонстраційного слайду протоколу B92

Другий етап розпочинається з натиснення кнопки “Далі” і додає на екран опис другого кроку, який реалізовується на боці абонента В і таблиця з набором базисів абонента В та результатами вимірювання отриманої від абонента А послідовності за тими ж базисами.

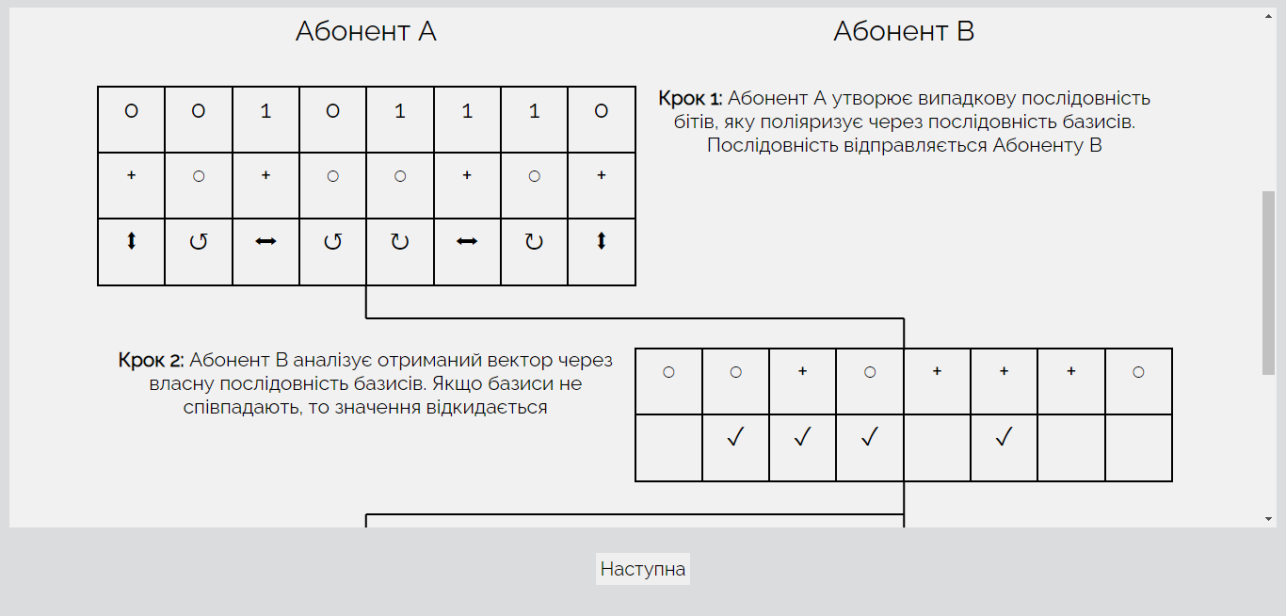


Рис. 6.7.2. - перший і другий етапи роботи демонстраційного слайду протоколу B92

Третій етап аналогічний другому, тільки на стороні абонента А відображаються значення поляризованих фотонів, які декодуються в біти.

Рис. 6.7.3. - третій і четвертий етапи роботи демонстраційного слайду протоколу B92

Останній етап даного слайду виводить на екран результат декодування успішно виміряних фотонів без одного, заздалегідь обумовленого, значення. Це

може бути перший або останній біт отриманої послідовності. Отриманий бітовий вектор і є спільним секретним ключем для абонентів А і В, згенерований за протоколом B92.

5.3.5. Практична робота №2

Протокол B92, як вже зазначалось, є продовженням чи то пак модифікацією протоколу BB84. Він має цілком ідентичний алгоритм реалізації, через що повторне використання конструкції, що використовувалась для першого практичного завдання є не раціональним з точки зору якнайкращого засвоєння матеріалу.

Саме тому використаємо інший підхід, який був би більш вдалим для даного протоколу, який є дещо менш ефективним в плані генерації ключів ніж його попередник.

Тому в основу практичного завдання поставимо аналіз і декодування отриманих від умовного абонента А сигналів до того моменту, доки не буде згенерований ключ необхідної довжини.

Для цього додаємо на слайд 3 набори елементів вводу, два з яких заповнюватимуться браузером послідовностями отриманих сигналів та власних для абонента В базисів.

Користувачу залишимо третій набір елементів, у який він має вводити проаналізовані і декодовані значення. Введені дані будуть проаналізовані браузером і якщо відповідь була правильною набори автоматично згенерованих даних поміняються і смужка заповнення ключа виросте. Якщо ні, то нічого не зміниться.

Реалізовуватимемо даний концепт за допомогою кнопки “Наступний раунд”, якій пропишемо функцію аналізу відповідей та аналізу отриманого

ключа. Якщо довжина ключа відповідатиме поставленому в задачі мінімуму, то завдання закінчиться, зникнуть всі його елементи і з’явиться кнопка переходу до наступного слайду.

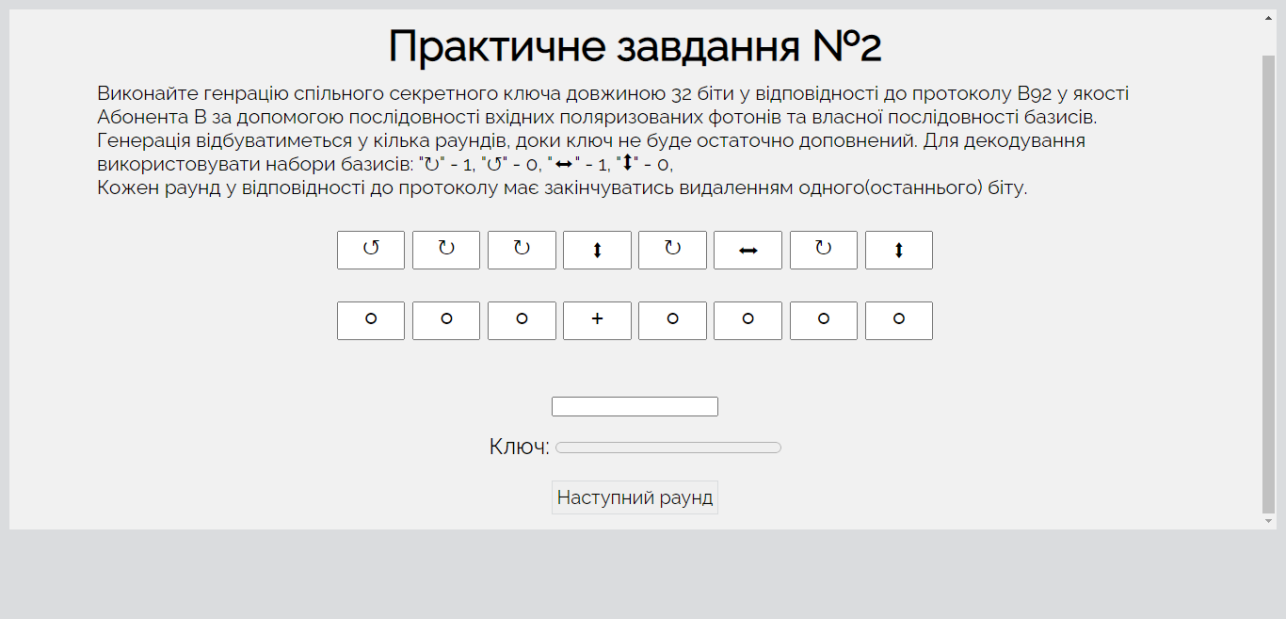


Рис. 6.8.1. - приклад роботи зі слайдом практичної роботи №2

5.3.6. Сторінка ознайомлення з протоколом E91

Для візуалізації алгоритму роботи протоколу E91 скористаємось підходом розділення екрану для паралельного відображення даних і процесів, що відбуваються на стороні абонента A і абонента B.

Для цього створимо за допомогою бібліотеки AnimeJS анімацію, яка символізуватиме одночану генерацію протилежно поляризованих фотонів. При цьому напрям руху фотонів зобразимо як симетричні одна до одної лінії, що йдуть на праву та ліву половини екрану, даючи тим самим зрозуміти, як поляризовані фотони потрапляють до абонентів.

На сторонах абонентів робимо два текстові елементи (по одному на кожному боці) і прописуємо для них сценарій, який би заміняв текст всередині

них на значення поляризації отриманих ними фотонів. Це робимо, аби наочно показати, що абонентам приходять саме протилежно поляризовані фотони.

По обидвох сторонах створюємо елементи для поетапного, як і в попередніх подібних слайдах, викладання інформації.

Перший етап розпочинаємо з натискання кнопки “Далі”, яка додає на екран вищеописану анімацію та послідовності отриманих абонентами фотонів. Під лівою послідовністю описуємо перший крок роботи алгоритму протоколу, а під правою — другий.

Рис. 6.9.1. - перші 2 етапи роботи демонстраційного слайду протоколу E91

Другий та третій етапи організовуємо аналогічно. З тією лише різницею, що в кожного абонента буде власний унікальний набір базисів, якими вимірюватиметься послідовність з першого етапу, а отже відповідно результати в абонентів будуть відрізнятись. Під новими наборами елементів з проаналізованими даними прописуємо кроки з третього по шостий в роботі алгоритму.

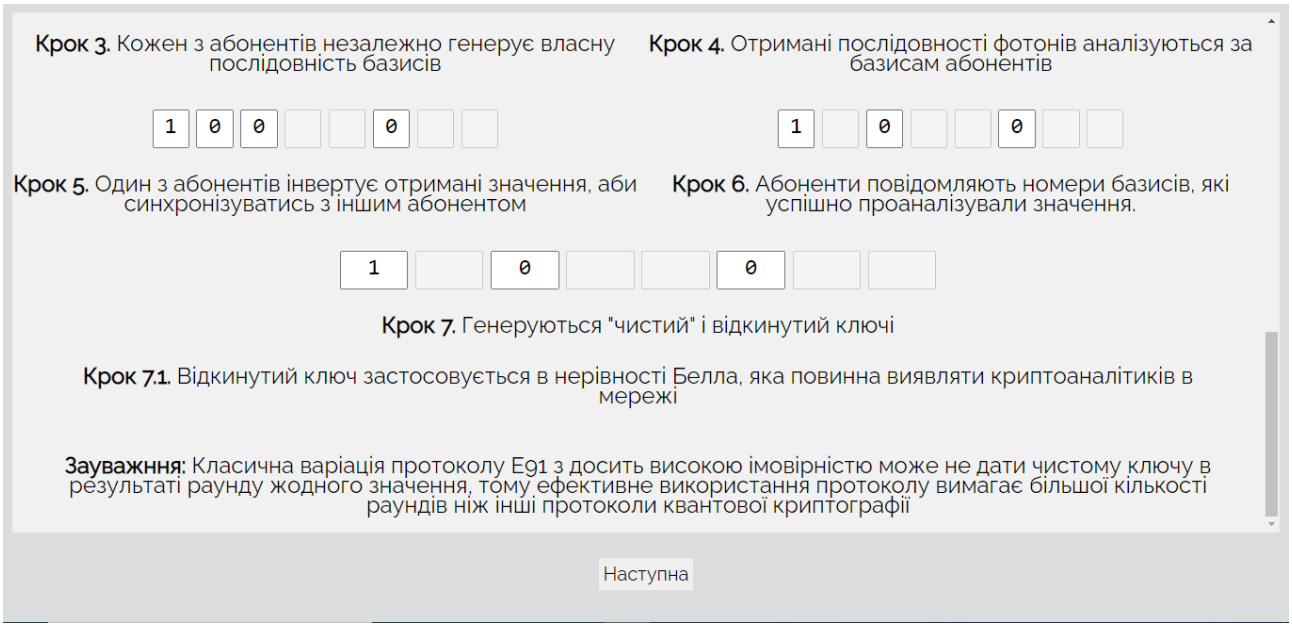


Рис. 6.9.2. - останні 2 етапи роботи демонстраційного слайду протоколу E91

Останній етап виводить на середину екрану результуючий спільний для обох абонентів секретний ключ та пояснення щодо того, що ефективність протоколу при генерації секретного ключа є досить низькою, через що він потребує більшої кількості ітерацій ніж інші аналогічні протоколи.

5.3.7. Практичне завдання №3

Хоч протокол E91 має досить суттєву відмінність від попередніх двох протоколів, та за великим рахунком він працює за тими ж принципами, що й BB84 і B92. Зокрема це полягає в тому, що основною ідеєю в них усіх є співпадіння послідовностей випадкових базисів, які генеруються абонентом.

Через це робити практичне завдання №3 аналогічним до двох попередніх вважатимемо надлишковим.

Натомість ліпше буде краще розкрити питання, що підіймалось в одному з кроків алгоритму на попередньому слайді — щодо проблем ефективності генерації довших ключів протоколом E91.

Для цього опишемо на слайді подробиці даної проблематики і зобразимо їх у вигляді графіків функції, які побудуємо за допомогою бібліотеки ChartsJS.

Самим же практичним завданням зробимо задачу з підрахунку йомвірності генерації ключа необхідної довжини протоколом E91 за умовлену кількість ітерацій.

Опишемо у слайді формули з математичної теорії імовірності і додамо елемент вводу, до якого необхідно буде ввести результат у вигляді цілої частини обчисленого відсотку.

Рис. 6.10.1. - приклад роботи зі слайдом практичної роботи №3

Умови практичного завдання, а саме необхідна довжина повідомлення (в бітах) та загальна кількість фотонів для генерації (спроб або ітерацій) генеруватимуться автоматично, аби щоразу поставала нова задача. Імовірність співпадіння власного базису із базисом генератора фотонів буде сталою і дорівнюватиме 1/3. Імовірність співпадіння базисів абонентів теж буде дорівнювати 1/3.

Таким чином для обрахунку правильної відповіді необхідно буде скористатись формулою:

$$P = n * P(A)^{n1} * P(B)^{n-n1}$$

Прописуємо в сценарії кнопки “Результат” порівняння відповіді з реальним значенням.

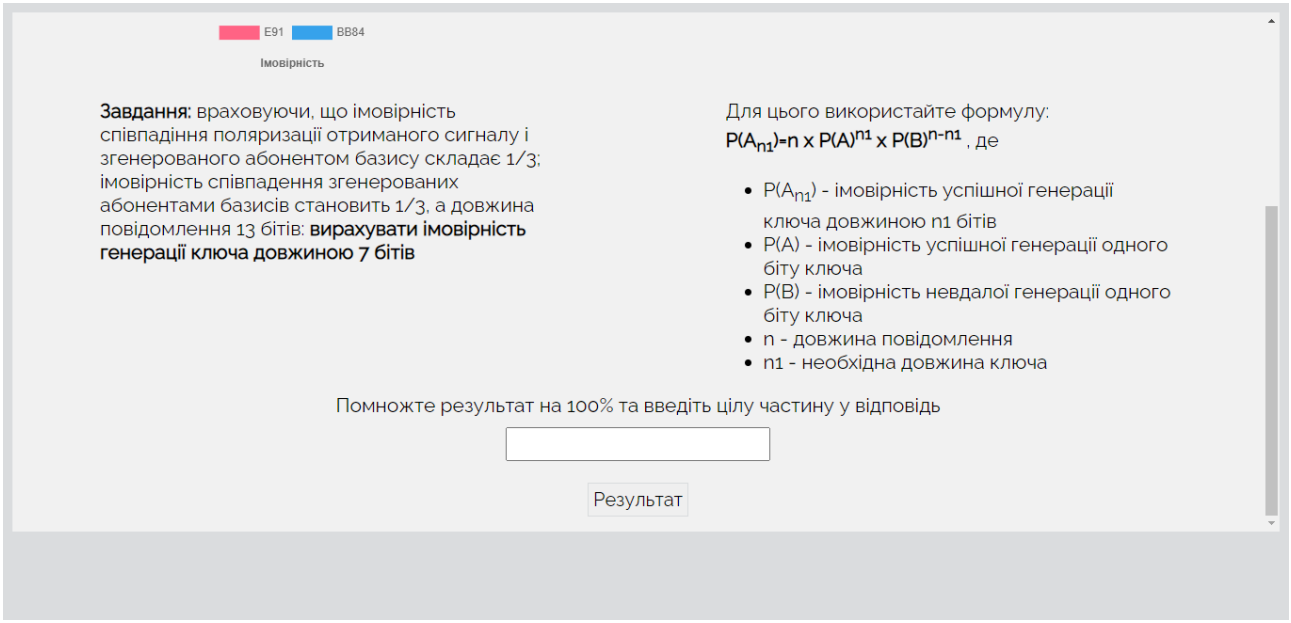


Рис. 6.10.2. - приклад роботи зі слайдом практичної роботи №3

Виконання практичного завдання №4 візуалізує кнопку переходу до наступних слайдів.

5.3.8. Сторінка ознайомлення з протоколом SARG-04

Подібно до того, як ми то робили в попередніх випадках, розіб’ємо сторінку демонстрації алгоритму роботу протоколу SARG-04 на ряд етапів, кожен з яких відповідав би за певний крок виконання алгоритму.

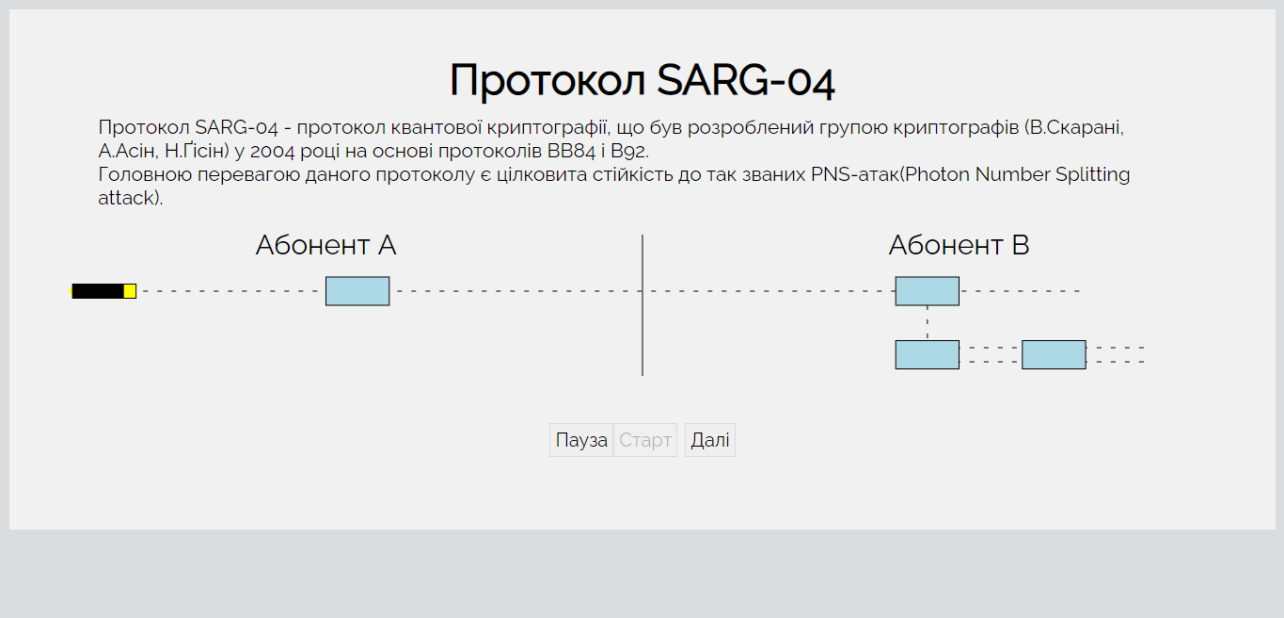


Рис. 6.11.1. - короткий опис і анімація протоколу SARG-04

Розмістимо на слайді одразу після опису протоколу анімовану візуалізацію передачі фотонів квантовими каналами зв’язку у в відповідності до того, як це має відбуватись в протоколі SARG-04. Додамо кнопки, які б відповідали за призупинення анімації та відновлення. Крім того додамо кнопку “Далі”, натискання на яку розпочне перший етап на слайді.

Початковий етап, як і у випадку з протоколом E91 розбиває екран на дві половини, які умовно позначають абонентів A і B, при цьому виводить на екран випадкові бітові послідовності та послідовності базисів, якими абонент A поляризуватиме фотони. Праворуч додаємо опис до цього першого кроку.

Другий етап демонструє дії на стороні абонента B, який вимірює отримані фотони за допомогою власної послідовності випадкових базисів. Відповідно ліворуч словесно описуємо це як крок №2.

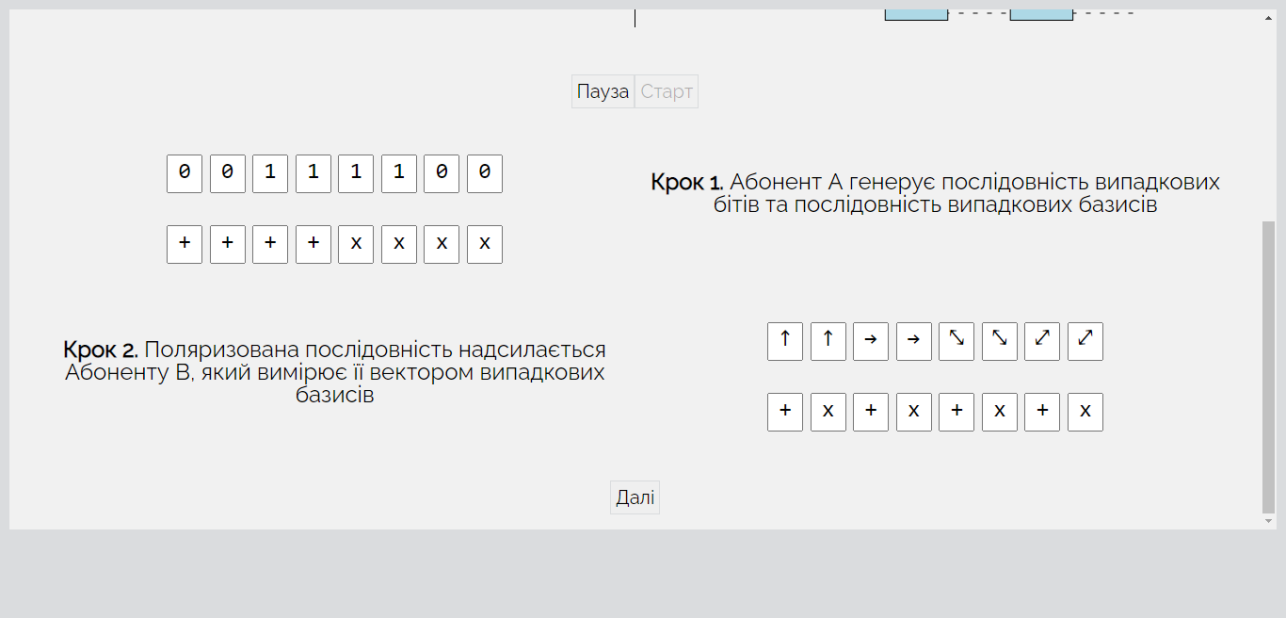


Рис. 6.11.2 - виконання 1-2 етапів роботи слайду протоколу SARG-04

Третім етапом демонструємо, знову ж таки на стороні абонента А, як той передає інформацію про базис, яким слід декодувати отриману раніше інформацію. Описуємо цей крок у правому стовпчику.

Четвертий етап демонструє результати декодування даних, яке полягає в пошуку ортогональних базисів у векторі фотонів та домовленому базисі. Якщо такий базис присутній, то значення декодується іншим, неортогональним базисом, що є у домовленому наборі. Останнє можна також порівняти з процесом інверсії бітів при знаходженні ортогонального базису.

П’ятий етап є етапом повідомлення відкритими каналами зв’язку про результати вимірювань, а отже розміщуємо опис цього кроку на всю ширину екрану і візуалізовуємо кнопку переходу до наступних слайдів.

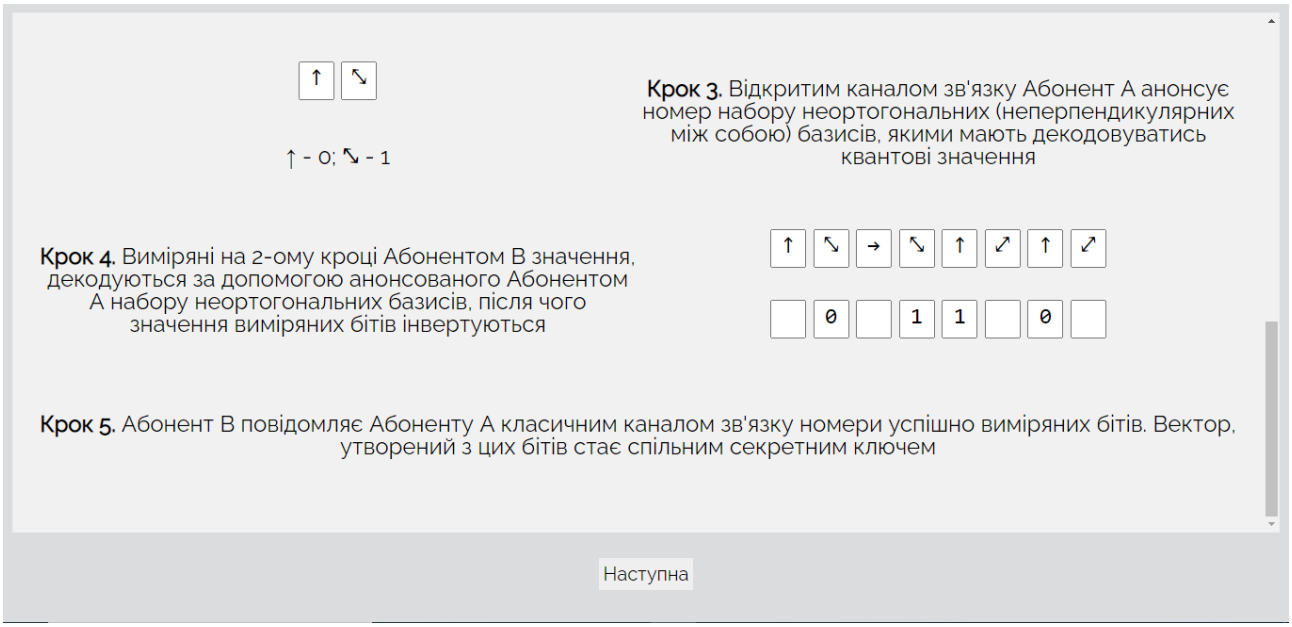


Рис. 6.11.3. - виконання 3-5 етапів роботи слайду протоколу SARG-04

В даному прикладі ми не використали динамічної генерації даних, адже з великою долею ймовірності більшість ключів були б порожніми, що було б недоречним в якості демонстративного матеріалу, тому, як і у випадку з протоколом B92, хоч і з інших причин, використаємо статичний приклад.

5.3.9. Практичне завдання №4

Реалізацію симуляції протоколу SARG-04 в практичному завданні №4 здійснимо за прикладом подібної симуляції в практичному завданні №2. Подібний повтор вважатимемо доречним, адже, враховуючи різницю в декодуванні в цих двох протоколах, завдання не є точною копією до того, що було представлено раніше.

Так, додаємо на екран два набори елементів вводу, з одним з яких працюватиме користувач, вводячи відповіді, а другий слугуватиме контейнером для відображення поляризації отриманих сигналів.

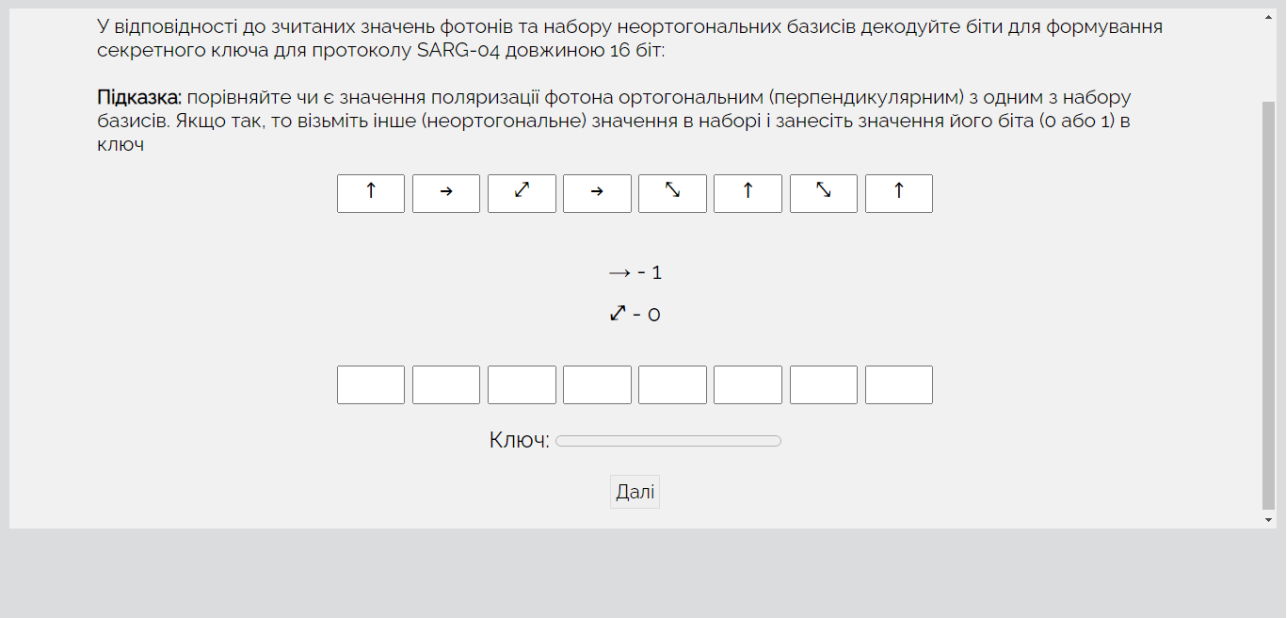


Рис. 6.12.1. - приклад роботи зі слайдом практичної роботи №4

Крім того додаємо на слайд текстові елементи, які динамічно змінюватимемо разом з вхідними послідовностями після закінчення кожного з раундів.

Принцип дії практичного завдання залишаємо незмінним з практичного завдання №2, а саме: згенерувати ключ необхідної довжини. Успішне виконання завдання дозволить відобразити кнопку переходу до інших слайдів.

Як і всі практичні завдання, це є основою для модуля, який існує незалежно від основної програми і мінімальним чином відрізняється від цього слайду.

5.3.10. Сторінка ознайомлення з протоколом KMB-09

Розробимо слайд з демонстрацією розподілу ключа з протоколом KMB-09. Перш за все врахуємо важливу особливість, що вирізняє даний протокол серед аналогів, а саме те, що декодування поляризованих фотонів (чи то пак пучків фотонів) відбувається за двома параметрами: базисом, визначеним абонентом B, та індексом базису, надісланим абонентом A.

Цілком раціонально в цьому випадку використати для пояснення принципів кодування-декодування таблицю, яка б містила ці два параметри і могла наочно показати, як відбувається даний процес.

Тому одразу після опису протоколу додаємо на слайд таблицю з довх наборів базисів для поляризації (кодування) сигналів. В самому тексті опису пояснюємо, що поляризація будь-якого значення відбувається за будь-яким базисом, знищуючи по суті тим самим значення біту, але декодування відбувається на перетині вертикалі й горизонталі таблиці по лініям обраного абонентом В базису і надісланого абонентом А індексу. Якщо ж отримати значення не виходить, то його відкидають.

Рис. 6.13.1. - підготовчий етап роботи зі слайдом протоколу KMB-09

Окремо слід описати виявлення ITER-помилки (індексної помилки). Яка являє собою неправильне декодування занчення (замість закодованого “0” отримують “1” і навпаки). Дана помилка є унікальною особливістю протоколу і використовується як засіб боротьби проти PNS-атак, адже збільшуючи розмір сигналу (пучок фотонів) збільшується і шум в лінії передачі, що дозволяє захистити дані від прослуховування.

Опісля додавання таблиці розташовуємо поетапно посередині екрану кроки з реалізації протоколу KMB-09. Робимо це тому, що для опису кроків в даному протоколі необхідно дещо більше місця і підхід з розділом екрану на 2 або тим паче три частини, як це було до того, є нераціональним.

Створюємо перший етап додаючи вже стандартні елементи зі згенерованими значеннями бітових послідовностей та неортогональних базисів, обраних абонентом А. Додаємо під ними текст опису першого кроку.

Другим етапом виносимо на екран вектор з індексами, які надсилає абонент А абоненту В і описуємо це текстовим блоком нижче.

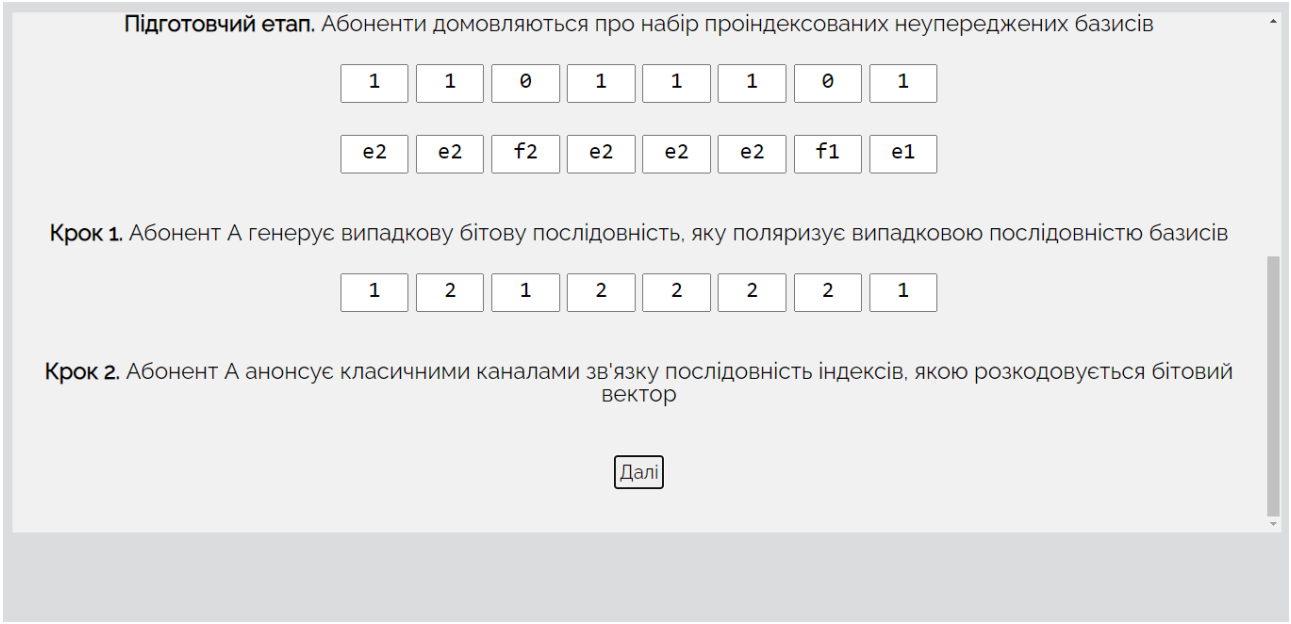


Рис. 6.13.2. - перший етап роботи зі слайдом протоколу KMB-09

Далі відображаємо згенеровану випадковим чином послідовність базисів абонента В, якими він буде аналізувати і декодувати сигнали. Описуємо це і переходимо до останнього етапу.

Четвертою стадією виводимо результуючу послідовність з проаналізованих вхідних даних. В комірках при цьому можуть відображатись як одиниці та нулі, так і символи відкинутих значень (“x”) та символи ITER-помилок (“I”). Ця

послідовність і є спільним секретним ключем для абонентів, що і описуємо в останньому — четвертому кроці.

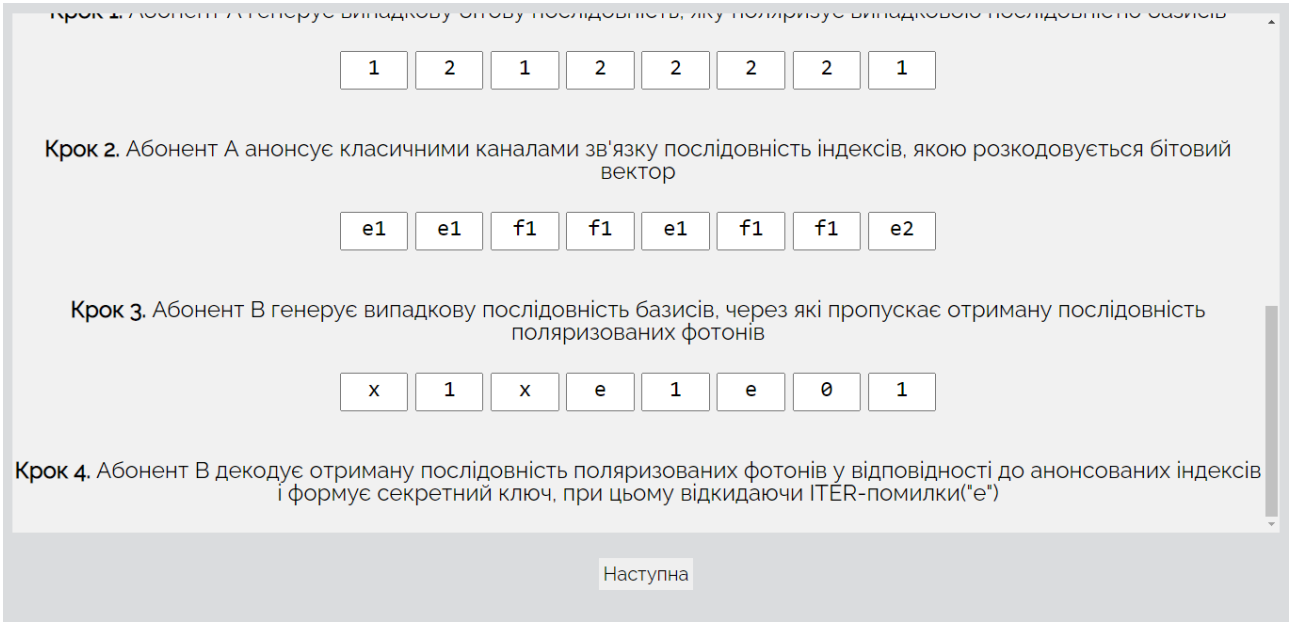


Рис. 6.13.3. - другий етап роботи зі слайдом протоколу KMB-09

Враховуючи, що даний протокол є більш ефективним, аніж попередні, ми можемо використовувати динамічні дані без великих ризиків отримати порожній ключ в результаті.

5.3.11. Практичне завдання №5

Реалізацію практичного завдання за протоколом KMB-09 реалізуємо за приблизно тією ж схемою, що й протоколи SARG-04 та B92 з тією лише різницею, що додатково додамо таблицю для декодування даних та декілька додатковий набір елементів, в яких зберігатиметься інформація про отримані індекси на додачу до базисів абонента B.

Рис. 6.14.1. - приклад роботи зі слайдом практичної роботи №5

Додаємо елементи для реалізації динамічних даних та оновлення їх при кожному новому раунді. Для цього пропишемо сценарій, подібний до тих, що були реалізовані в попередніх практичних завданнях і з урахуванням особливостей самого протоколу КМВ-09.

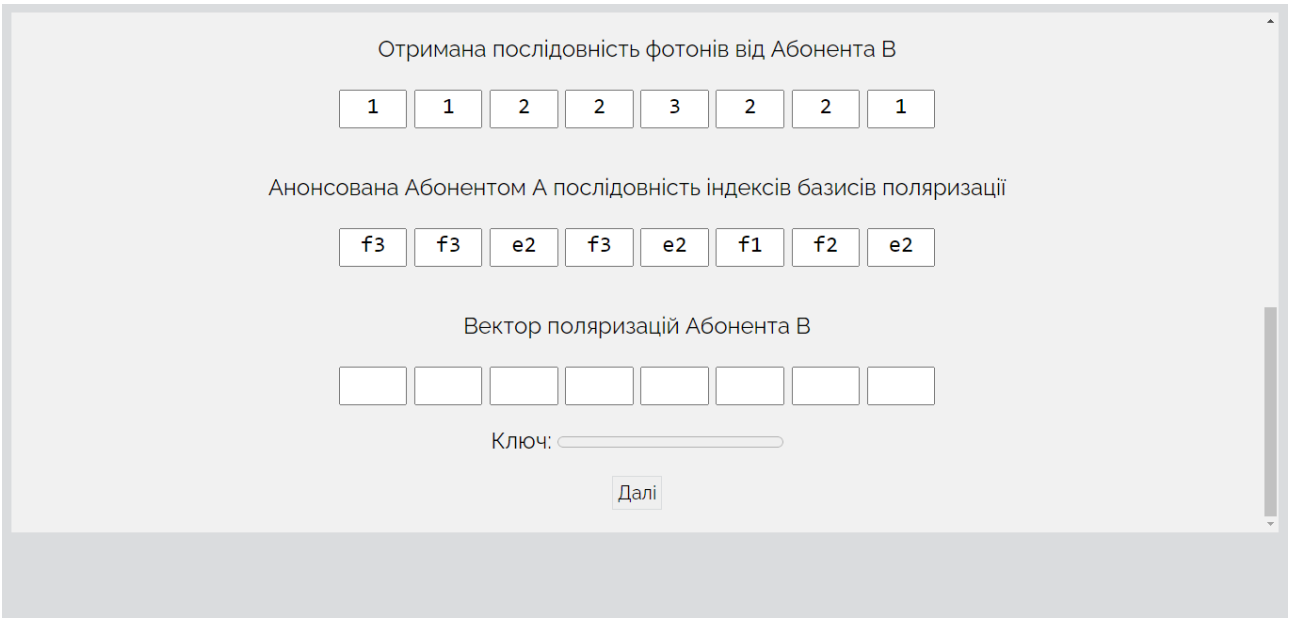


Рис. 6.14.2. - приклад роботи зі слайдом практичної роботи №5

Прописуємо необхідну довжину ключа як 24 біти, після досягнення чого з’явиться кнопка переходу до останнього слайду.

5.3.12. Підсумкова сторінка

Підсумковий слайд виконує роль свого роду табелю, в якому демонструватимуться успіхи, що їх досягатиме користувач в ході проходження навчання в тілі основної програми.

Для цього розмістимо на слайді опис усієї програми, а також додамо таблицю, до якої опісля виконання кожного практичного завдання додаватимуться результати виконання даного завдання та оцінка за проходження даної свого роду лабораторної роботи вцілому.

В перспективі, при подальшому розвитку можливостей додатку, цілком реальним є додавання до даного слайду можливості передачі результатів навчання на сервери спеціалізованих систем, наприклад: Lider, Moodle, абощо.

5.4. Модулі симуляції

До модулів симуляції відносимо усі практичні завдання, які увійшли до основної програми з тією лише різницею, що існуватимуть окремо від тіла головної програми у вигляді ряду менших додатків.

Головною метою створення модулів симуляції є потреба, що її матиме користувач у напрацюванні практичних навичок поводження із конкретним протоколом квантової криптографії та без проходження через усе тіло основної програми.

До числа таких модулів відносимо файли:

- BB84.html
- B92.html
- E91.html
- SARG-04.html
- KMB-09.html

Слід зауважити при цьому, що файл E91.html не є повноцінним симулятором. Радше є наочним прикладом ефективності даного протоколу.

Решта файлів разом із файлами сценаріїв є повноцінними симуляторами, чия мета створити певну базу для розуміння практичної роботи протоколів квантової криптографії.

ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

6.1. Вимоги безпеки при виконанні робіт на робочому місці

До основних законодавчих актів про охорону праці відноситься Закон України про “Основи законодавства України про охорону здоров’я”[1], що регулює суспільні відносини в цій галузі з метою забезпечення гармонічного розвитку фізичних і духовних сил, високої працездатності і довголітнього

активного життя громадян, усунення чинників, які шкідливо впливають на їхнє здоров’я, попередження і зниження захворюваності, інвалідності та смертності, поліпшення спадкоємності. Даний закон передбачає встановлення єдиних санітарно-гігієнічних вимог до організації виробничих та інших процесів, пов’язаних з діяльністю людей, а також до якості машин, устаткування, будинків та таких об’єктів, що можуть шкідливо впливати на здоров’я людей (стаття 28); вимагає проведення обов’язкових медичних оглядів осіб певних категорій, в тому числі працівників, зайнятих на роботах із шкідливими та небезпечними умовами праці (стаття 31); закладає правові основи медико-соціальної експертизи втрати працездатності (стаття 69).

Основними документами, що визначають головні положення щодо охорони праці для прграміста-розробника є ДНАОП 0.00-8.03-93 “Порядок опрацювання та затвердження власником нормативних актів про охорону праці, що діють на підприємстві” [2], ДНАОП 0.00-4.15-98 “Положення про розробку інструкцій з охорони праці” [3], ДНАОП 0.00-4.12-99 “Типове положення про навчання з питань охорони праці” [4].

Щодо охорони праці програміста під час роботи на робочому місці документ описує вимоги до працівника до початку роботи, безпосередньо під час роботи та опісля закінчення роботи.

Перед початком роботи програміст-розробник зобов’язаний перевірити:

• справність обладнання, інструменту, приладів;

• наявність і справність достатнього освітлення, вентиляції, обладнання тощо;

• перевірити справність рубильників, розеток, штепсельних з’єднань тощо.

У разі виявлення будь-яких порушень, несправностей, пошкоджень чи невідповідностей до технічної документації в обладнанні програміст

зобов’язаний повідомити про це відповідальних за функціювання обладнання і керівництво організації (компанії, лабораторії, університету, тощо).

Під час виконання робіт програміст зобов’язаний:

• Виконувати роботу згідно із своїми посадовими обов’язками.

• Не залишати без нагляду своє робоче місце, коли обладнання підключено до електромережі.

У випадку виявлення будь-яких відхилень, несправностей, пошкоджень негайно повідомити керівництво організації.

По закінченню виконання робіт з необхідним обладнанням програміст повинен:

Перевірити своє робоче місце.

• Відключити від електромережі електрообладнання.

• Закрити вікна.

• Вжити заходів особистої гігієни: старанно вимити руки, при можливості прийняти душ.

• Привести в порядок спеціальний одяг, зняти і прибрати його в окреме місце.

Згідно із Законом України “Про забезпечення санітарного та епідемічного благополуччя населення” [5] забезпечення санітарного благополуччя на підприємстві досягається такими основними заходами:

– гігієнічною регламентацією та контролем (моніторингом) усіх шкідливих і небезпечних факторів навколишнього та виробничого середовища;

– державною санітарно-гігієнічною експертизою проектів, технологічних регламентів, інвестиційних програм та діючих об'єктів;

– включенням вимог безпеки щодо здоров'я та життя людини в державні стандарти та нормативно-технічну документацію усіх сфер діяльності суспільства;

– ліцензуванням видів діяльності, пов'язаних з потенційною небезпекою для здоров'я людей;

– пред'явленням відповідних гігієнічних вимог до проектування, забудови, та експлуатації будівель, споруд, приміщень, територій, розробкою та впровадженням нових технологій і обладнання;

– контролем та аналізом стану здоров'я населення та робітників;

– профілактичними санітарно лікувальними заходами;

– запровадженням санкцій до відповідальних осіб за порушення санітарно-гігієнічних вимог.

Відповідно до цього Закону підприємства, установи і організації зобов'язані розробляти і здійснювати санітарні та протиепідемічні заходи; забезпечувати лабораторний контроль за виконанням санітарних норм стосовно рівнів шкідливих для здоров'я факторів виробничого середовища; інформувати органи та установи державної санепідеміологічної служби про надзвичайні події та ситуації, що становлять небезпеку для здоров'я населення; відшкодувати в установленому порядку працівникам та громадянам збитки, яких завдано їх здоров'ю в результаті порушення санітарного законодавства.

6.2. Шкідливі виробничі фактори на робочому місці

У відповідності із наказом Міністерства охорони здоров’я України №248 від 08.04.2014 [6] шкідливими виробничими факторами вважаються:

1) фізичні фактори:

- неіонізуючі електромагнітні поля та випромінювання: електростатичні поля, постійні магнітні поля, електричні та магнітні поля промислової частоти (50 Гц), електромагнітні випромінювання радіочастотного діапазону,

електромагнітні випромінювання оптичного діапазону, зокрема лазерне та ультрафіолетове;

- іонізуючі випромінювання;
- виробничий шум, ультразвук, інфразвук;
- вібрація (локальна, загальна);
- освітлення: природне (відсутність або недостатність), штучне (недостатня освітленість, прямий і відбитий сліпучий відблиск тощо);
- іонізація повітря;

2) хімічні фактори:

- речовини хімічного походження, деякі речовини біологічної природи, які отримані хімічним синтезом та/або для контролю яких використовуються методи хімічного аналізу, аерозолі фіброгенної дії (пил);

3) біологічні фактори:

- мікроорганізми - продуценти, живі клітини та спори мікроорганізмів, що містяться в бактеріальних препаратах, патогенні мікроорганізми;

4) фактори трудового процесу:

- важкість праці - характеристика трудового процесу, що відображає рівень загальних енергозатрат, переважне навантаження на опорно-руховий апарат, серцево-судинну, дихальну та інші системи.

Серед них безпосереднє відношення до роботи програміста-розробника мають такі фактори, як: мікроклімат, неіонізуючі електромагнітні поля та випромінювання, іонізуючі випромінювання, виробничий шум, ультразвук, інфразвук, освітлення. Нормування цих показників затверджене в НПАОП 0.00-7.15-18 [9].

Для нормального, нешкідливого виконання працівником своїх посадових обов’язків дані показники не повинні перевищувати прийнятих у ДСН норм.

Так, згідно з ДСН 3.3.6.037-99 [7] рівень шуму має відповідати стандарту:

Рівні звукового тиску в дБ в октавних смугах з середньогеометричними частотами, Гц								
31,5	63	125	250	500	1000	2000	4000	8000
86	71	61	54	49	45	42	40	38

Таблиця 6.1. - Рівень звукового тиску

Відповідно до ДСН 3.3.6.042-99 [8] мікроклімат при роботі програміста-розробника, яка належить до категорії 1а має бути дотриманий в наступних стандартах:

Температура повітря	Відносна вологість	Швидкість руху повітря
22-24 градуси Цельсію	60-40 г/м^3	0,1 м/сек

Таблиця 6.2. - Показники мікроклімату

Відповідно до додатку №14 наказу №248 Міністерства охорони здоров'я рівень освітлення робочого місця працівника категорії 1а повинен відповідати:

Природнє освітлення	
Коефіцієнт природного освітлення (КПО, %)	≥0,6
Штучне освітлення	
Освітленість, лк	200

Таблиця 6.3. - Показники освітлення робочого місця

6.3. Дії працівників в надзвичайних ситуаціях

Дії працівників підприємств визначаються у відповідності до наказу про "затвердження Типового положення про порядок проведення навчання і перевірки знань з питань охорони праці та Переліку робіт з підвищеною небезпекою" (НПАОП 0.00-4.12-05) [10].

При виявлені небезпечної ситуації (пожежа, землетрус, радіаційна безпека, неполадки в електрогосподарстві, тощо) для власного життя та життя співробітників заспокоїти і заспокоїти оточуючих.

Не усувати самому несправностей електромережі та електрообладнання, а вимкнути загальне електропостачання.

При виявленні пожежі зобов'язаний негайно викликати пожежну частину.

Вжити заходів згідно з планом евакуації на випадок пожежі, виробничих та природних явищ та вивести працівників у безпечне місце. Організувати роботу ДПД щодо збереження майна та цінних паперів.

При появі сторонньої особи, яка застосовує протиправні дії щодо безпеки життєдіяльності оточуючих, викликати міліцію.

У випадку травмування працівників або клієнтів під час роботи підприємства необхідно викликати швидку допомогу або за потреби надати першу долікарську допомогу, за необхідності створити комісію по розслідування нещасного випадку, видати акт встановленого зразка, наказ про підсумки розслідування, повідомлення про наслідки нещасного випадку.

Дії при наданні першої долікарської допомоги. Надання першої медичної допомоги починати з оцінки загального стану потерпілого і на підставі цього скласти думку про характер пошкодження.

У разі різкого порушення або відсутності дихання, зупинки серця негайно зробити штучне дихання та зовнішній масаж серця, викликати швидку медичну допомогу.

Дії при ураженні електричним струмом:

• необхідно звільнити потерпілого від дії електричного струму, відключивши електрообладнання від джерела живлення, а при неможливості відключення – відтягнути його від струмоведучих частин за одяг або застосувавши підручний ізоляційний матеріал;

• за відсутності у потерпілого дихання і пульсу необхідно робити йому штучне дихання і непрямий (зовнішній) масаж серця, звернувши увагу на зіниці. Розширені зіниці свідчать про різке погіршення кровообігу мозку. При такому стані оживлення необхідно починати негайно, після чого викликати швидку медичну допомогу.

Дії при пораненні:

• для надання першої допомоги при пораненні необхідно розкрити індивідуальний пакет, накласти на рану стерильний перев'язувальний матеріал і зав'язати її бинтом;

• якщо індивідуального пакету немає, то для перев'язки необхідно використати чисту носову хустинку, чисту полотняну ганчірку тощо. На те місце ганчірки, що приходиться безпосередньо на рану, бажано накапати декілька капель настойки йоду, щоб одержати пляму розміром більше рани, а після цього накласти ганчірку на рану.

Дії при переломах, вивихах, ударах, розтягненні:

• при переломах і вивихах кінцівок необхідно пошкоджену кінцівку укріпити шиною, фанерною пластинкою, палицею, картоном або іншим подібним предметом. Пошкоджену руку можна також підвісити за допомогою перев'язки або хустки до шиї і прибинтувати до тулуба; • при передбачуваному переломі черепа (несвідомий стан після удару голови, кровотеча з вух або рота) необхідно прикласти до голови холодний предмет (грілку з льодом або снігом, чи холодною водою) або зробити холодну примочку;

• при підозрі перелому хребта необхідно потерпілого покласти на дошку, не піднімаючи його, чи повернути потерпілого на живіт обличчям у низ, наглядаючи при цьому, щоб тулуб не перегинався з метою уникнення ушкодження спинного мозку;

• при переломі ребер, ознакою якого є біль при диханні, кашлю, чханні, рухах необхідно туго забинтувати груди чи стягнути їх рушником під час видиху.

Дії при теплових опіках:

• при опіках вогнем,- парою, гарячими предметами ні в якому разі не можна відкривати пухирі, які утворюються, та перев'язувати опіки бинтом;

• при опіках першого ступеня (почервоніння) обпечене місце обробляють ватою, змоченою етиловим спиртом; при опіках другого ступеня (пухирі) обпечене місце обробляють спиртом, 3 % марганцевим розчином або 4 % розчином таніну;

• при опіках третього ступеня (зруйнування шкіряної тканини) накривають рану стерильною пов'язкою та викликають лікаря.

Дії при кровотечі:

• для того, щоб зупинити кровотечу, необхідно підняти поранену кінцівку вверх, кровоточиву рану закрити перев'язувальним матеріалом (із пакета), складеним у клубочок, придавити її зверху, не торкаючись самої рани, потримати протягом 4 хвилин;

• при сильній кровотечі, яку не можна зупинити пов'язкою, застосовується здавлювання кровоносних судин, які живлять поранену область, за допомогою згинання кінцівок у суглобах, а також пальцями, джгутом або закруткою; при великій кровотечі необхідно термічна обробка рани (обпалення), аби таким чином, оплавивши шкіру, створити природну заслону.

ВИСНОВКИ

Розроблений в даній роботі додаток, є цілком гідним варіантом розв’язку проблематики, що була піднята в перших розділах цього дослідження. Так, попри свою компактність і певну обмеженість функціоналу, даний додаток є унікальним інструментом, який сповнює собою ті методологічні прогалини, що існують в системі сучасної української (і не тільки) освіти.

Зокрема, знаходить своє рішення проблема доступного для розуміння пересічними студентами викладання матеріалів, пов’язаних з такою галуззю сучасної науки, як квантова криптографія, та проблема практичного застосування отриманих під час навчання знань і формування у відповідності до них професійних навичок.

Іншою важливою метою, яка була поставлена (і частково досягнута) є зміна підходів до самого питання методик викладання. Особливо методик віддаленого навчання, що стало особливо актуальним в умовах, що їх сучасний світ. Неповнота успіху в цьому напрямі, варто зазначити, не є хибою самого дослідження чи підходів, які були у ньому розглянуті, а радше системною проблемою, яка має розв’язуватись системно, зокрема через застосування як широкими верствами студентів, так і особливо — викладачами, нових

програмно-технічних засобів, які б піднімали навчальний процес на якісно новий рівень. Розробка і навіть якнайширше застосування одного нового навчального додатку з цілком очевидних причин не може бути при цьому вирішенням проблеми загалом, однак може бути чудовим і вкрай наочним прикладом для подальших спроб і досліджень у цьому напрямку.

Список літератури

1. “Основи законодавства України про охорону здоров’я”: Закон України від 19.11.1992р. №2801-XII

2. НПАОП 0.00-8.03-93 “Порядок опрацювання та затвердження власником нормативних актів про охорону праці, що діють на підприємстві”.

3. НПАОП 0.00-4.15-98 “Положення про розробку інструкцій з охорони праці”.

4. НПАОП 0.00-4.12-99 “Типове положення про навчання з питань охорони праці”.

5. “Про забезпечення санітарного та епідемічного благополуччя населення”: Закон України від 24.02.1994р. №4004-XII. Голос України. 1994. 8 квітня.

6. “Про затвердження Державних санітарних норм та правил «Гігієнічна класифікація праці за показниками шкідливості та небезпечності факторів виробничого середовища, важкості та напруженості трудового процесу»”: Наказ Міністерства охорони здоров’я України від 08.04.2014р. №248.

7. ДСН 3.3.6.037-99 “Санітарні норми виробничого шуму, ультразвуку та інфразвуку”.

8. ДСН 3.3.6-042-99 “Санітарні норми мікроклімату виробничих приміщень”.

9. НПАОП 0.00-7.15-18 “Вимоги щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями”.

10. НПАОП 0.00-4.12-05 “Типове положення про порядок проведення навчання і перевірки знань з питань охорони праці”.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. https://www.st-andrews.ac.uk/
2. https://www.st-andrews.ac.uk/physics/quvis/simulations_html5/sims/cryptography-b92/B92_photons.html
3. https://arxiv.org/
4. https://journals.aps.org/
5. https://www.st-andrews.ac.uk/physics/quvis/simulations_html5/sims/quantum-versus-hv1/quantum-versus-hv1.html
6. https://www.st-andrews.ac.uk/physics/quvis/simulations_html5/sims/cryptography/Quantum_Cryptography.html
7. https://listserv.nodak.edu/cgi-bin/wa.exe?A2=NMBRTHRY;fd743373.1912
8. https://www.popularmechanics.com/science/math/a30149512/longest-encryption-ever-cracked/

9. http://irbis-nbuv.gov.ua/cgi-bin/irbis_nbuv/cgiirbis_64.exe?C21COM=2&I21DBN=UJRN&P21DBN=UJRN&IMAGE_FILE_DOWNLOAD=1&Image_file_name=PDF/sstt_2011_4_9.pdf
10. https://www.sciencedirect.com/science/article/pii/S0304397514004241?via%3Dihub
11. https://www.infoamerica.org/documentos_pdf/bennett1.pdf
12. https://www.researchgate.net/figure/Description-of-the-mechanism-of-B92-protocol_tbl3_228669847
13. https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.90.167904
14. https://arxiv.org/pdf/quant-ph/0211131.pdf
15. https://arxiv.org/pdf/quant-ph/0510025.pdf
16. https://arxiv.org/pdf/0901.3909.pdf
17. https://www.researchgate.net/publication/276409981_Differentiations_of_QKDPs_in_Run-Time_Execution