Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)
Факультет комп'ютерно-інформаційних систем і програмної інженерії
(назва факультету)
Кафедра кібербезпеки
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр
(освітній рівень)

на тему: *«Розроблення методів ключового хешування з використанням арифметики еліптичних кривих»*

Виконав: студент (ка)
Спеціальності:
125 «Кібербезпека»
(шифр і назва напряму підготовки, спеціальності)
Кушнір В.П.
підпис (прізвище та ініціали)

Керівник *Александер Марек Богуслав*
підпис (прізвище та ініціали)

Нормоконтроль
підпис (прізвище та ініціали)

Завідувач кафедри *Загородна Н.В.*
підпис (прізвище та ініціали)

Рецензент
підпис (прізвище та ініціали)

м. Тернопіль – 20*21*

АНОТАЦІЯ

Розробка методів ключового хешування з використанням арифметики еліптичних кривих // Кваліфікаційна робота ОР «Бакалавр» //Кушнір Володимир Петрович// Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра кібербезпеки, група СБс-42 // Тернопіль, 2021 // С. , рис. – , табл. – , кресл. – , додат. – .

Ключові слова: ІНТЕРНЕТ-ПЛАТІЖНА СИСТЕМА, ЦІЛІСНІСТЬ ІНФОРМАЦІЇ, АВТЕНТИФІКАЦІЯ ПОВІДОМЛЕНЯ, ЕЛЕКТРОННИЙ ЦИФРОВИЙ ПІДПИС, ХЕШ-ФУНКЦІЯ.

Кваліфікаційна робота присвячена ключового хешування для забезпечення цілісності та автентичності інформації в Інтернет-платіжних системах. У результаті в роботі проведено аналіз існуючих механізмів забезпечення цілісності та автентичності інформації в Інтернет-платіжних системах. проведена оцінка стійкості сучасних алгоритмів ключового та без ключового хешування за допомогою методики тестування NIST STS національного інституту стандартів та технологій США. Проведений аналіз побудови хеш-функцій MASH-1 та MASH-2 та визначені їх часові характеристики. З використанням положень теорії еліптичних кривих розроблені методи вдосконалення алгоритмів ключового хешування MASH-1 та MASH-2. Обгрунтовані рекомендації, щодо їх використання для забезпечення цілісності та автентичності транзакцій в Інтернет-платіжних системах. Створений програмний продукт імітаційної моделі ключевого алгоритму хешування з використанням арифметики в групі точок еліптичної кривої.

ANNOTATION

Development of key hashing methods using elliptic curves theory// Thesis of educational level "Bachelor" // Kushnir Volodymyr Petrovych // Ternopil National Technical University named after Ivan Pulyuy, Faculty of Computer Information Systems and software engineering, Department of Cybersecurity, СБс-42 group // Ternopil, 2021 // P. , fig. -, table. - , chair. - , added. -.

Keywords: INTERNET PAYMENT SYSTEM, INTEGRITY OF INFORMATION, AUTHENTICATION MESSAGES, DIGITAL SIGNATURE, HASH FUNCTION.

The qualification thesis is devoted to modeling of hash key to ensure the integrity and authenticity of information in the payment systems of Internet. The result of work is the analysis of existent mechanisms which provides integrity and authentication of information in the payment systems of Internet. Conducted estimation of firmness of modern algorithms key and without the key randomising by the method of testing NIST STS national institute of standards and technologies of the USA. Conducted analysis of construction of hash function MASH-1 and MASH-2 and certain them sentinel descriptions. With the use of positions of theory of elliptic curves the developed methods The software product of simulation model of keyed hashing algorithm is created with the use of arithmetic in the group of points of elliptic curve.

ЗМІСТ

ВСТУП

Розвиток високорентабельної економіки неможливий без впровадження сучасної безготівкової системи грошового обігу та використання ефективних платіжних інструментів, таких як Інтернет-платіжна система. Швидкий ріст кількості транзакцій у сучасних Інтернет-платіжних системах, поява нових форм електронних послуг, стрімкий розвиток обчислювальної техніки висувають нові вимоги до надійності та забезпечення цілісності та автентичності в Інтернет-платіжних системах – складних багаторівневих системах централізованого керування, що забезпечують якісний важливий канал проведення фінансових транзакцій. Проведений аналіз [1 – 4; 12; 29] показав, що сучасні механізми забезпечення цілісності та автентичності даних в Інтернет-платіжних системах не забезпечують зростаючі потреби до стійкості. Виникає протиріччя, коли існуючі на практиці механізми забезпечення цілісності та автентичності даних не володіють властивостями, необхідними для задоволення вимогам безпеки, що до них висунуті. Дане наукове дослідження дозволить вирішити виявлене протиріччя. Таким чином, дослідження та аналіз існуючих методів ключового хешування для забезпечення цілісності та автентичності даних в Інтернет-платіжних системах та подальше їх вдосконалення є актуальним.

Слід зазначити, що на поточний період Україна не має Національного стандарту хешування інформації, для забезпечення цілісності та автентичності даних. В Україні використаються алгоритми, визначені міжнародними та Російськими стандартами, а саме ДСТУ 34.310-95, ДСТУ 34.311-95 і ДСТУ 28147-89, ДСТУ 4145-02 [1-4]. Таким чином, розробка і дослідження перспективних методів та алгоритмів ключового хеширования є актуальним напрямком подальших досліджень.

Дослідження в роботі проводилися у відповідності з наступними нормативними актами.

1. Концепція розвитку зв'язку України до 2010 року, затверджена постановою Кабінету Міністрів України “Про Концепцію розвитку зв'язку

України до 2010 року" від 9 грудня 1999 р. № 2238;

2. Концепція Національної програми інформатизації схваленої Законом України "Про Концепцію Національної програми інформатизації" від 4 лютого 1998 р. № 75/98-ВР;

3. Державна науково-технічна програма "Створення перспективних телекомунікаційних систем і технологій";

4. Науково-дослідницькі розробки: "Дослідження перспективних методів і механізмів забезпечення цілісності і автентичності даних, що циркулюють у системі комерційного банка" (ХНЕУ, м. Харків) номер державної реєстрації 0108U009227.

Мета роботи – забезпечення цілісності та автентичності транзакцій в Інтернет-платіжних системах на основі методів ключового хешування з використанням арифметики еліптичних кривих.

Об'єкт дослідження – процес забезпечення цілісності та автентичності в Інтернет-платіжних системах.

Предмет дослідження – методи ключового хешування з використанням арифметики еліптичних кривих.

Для досягнення поставленої мети необхідно вирішити наступні задачі:

– визначення особливостей структури Інтернет-платіжної системи.

– аналіз механізмів забезпечення цілісності та автентичності інформації, що циркулює в Інтернет-платіжній системі.

– розроблення методу ключового хешування з використанням еліптичних кривих;

– програмна реалізація методу та обґрунтування рекомендацій щодо його використання.

РОЗДІЛ 1 СУЧАСНІ ПРОБЛЕМИ ЗАБЕЗПЕЧЕННЯ ЦІЛІСНОСТІ ТААВТЕНТИЧНОСТІ ТРАНЗАКЦІЙ В ІНТЕРНЕТ-ПЛАТІЖНИХ СИСТЕМАХ

Розглядаються особливості структури Інтернет-платіжної системи. Визначаються сучасні проблеми забезпечення цілісності та автентичності транзакцій в Інтернет-платіжних системах. Досліджуються алгоритми цифрового підпису, які використовуються в Інтернет-платіжних системах.

1.1. Структура Інтернет-платіжної системи та основні погрози безпеки Інтернет-платіжної системи

Розвиток економіки будь-якої держави сьогодні неможливий без високоефективної системи грошового обігу та використання сучасних платіжних механізмів. Одним із самих прогресуючих напрямків розвитку платіжних систем є Інтернет-платіжні системи, які дозволяють робити миттєві і безготівкові транзакції, використовуючи віртуальні рахунки та електронні гроші [1-3]. Процес розвитку електронної комерції вимагає наявності відповідних платіжних інструментів, що дозволяють здійснювати on-line розрахунки відповідно до загальноприйнятих світових стандартів. У цьому зв'язку на перший план виходять надійність, безпека, а також терміновість здійснення платежів в Інтернет-платіжних системах.

Інтернет-платіжна система – складна багаторівнева система децентралізованого керування, що забезпечує якісний канал проведення фінансових транзакцій у середовищі Інтернет.

Інтернет-платіжна система – система проведення розрахунків між фінансовими, бізнес-організаціями та Інтернет-користувачами в процесі покупки/продажу товарів і послуг через Інтернет. Саме платіжна система і можливість оплати замовлення на сайті дозволяє стверджувати, що даний сайт є

повноцінним магазином, а не службою по обробці замовлень або електронною вітриною [4 – 10].

Системи електронної комерції повинні задовольняти ряд умов щодо оплати платежів:

1. Конфіденційність. Це означає, що покупець, оплачуючи через Інтернет хоче бути певним, що дані його картки та деталі його покупки залишаться в секреті для всіх сторонніх осіб, окрім тих, хто має на це законне право.

2. Цілісність інформації. Повинно бути неможливим для зробити змінив інформацію про покупки.

3. Автентифікація. Всі сторони торгової угоди (покупці та продавці) повинні бути впевнені, що всі є тими, за кого вони себе видають.

4. Засоби оплати. Повинна бути можливість оплатити шшироким асортиментом платіжних засобів.

5. Авторизація. Транзакція буде авторизована, якщо вона схвалена платіжною системою і навпаки – неавторизована транзакція означає відхилення платіжно системою. Ця процедура дозволяє так перевірити наявність необхідних коштів у покупця.

6. Гарантії ризиків продавця. Продаючи товар в інтернеті, продавець зустрічається з безліччю ризиків, пов'язаних з відмовами від товару та несумлінністю покупця. Величина ризиків повинна бути погоджена із провайдером платіжної системи та інших організацій, включеними в торговельні ланцюжки, з допомогою додаткових угод.

7. Мінімізація плати за транзакцію. Плата за обробку транзакцій замовлення та оплати товарів, природно, входить у їхню вартість, тому зниження ціни транзакції буде впливати на конкурентоспроможність. [1,9,18,26].

Всі зазначені вище умови повинні бути власивими справжній Інтернет платіжній системі. Таким чином, всі платіжні системи за наявною схемою платежів можна розділити на:

– дебетові (працюючі з електронними чеками та цифровою готівкою)
– кредитні (працюючі із кредитними картками).

Дебетові схеми платежів побудовані за аналогією до їх оффлайн прототипів: чековим та звичайним грошовим.

До схеми розрахунків (див. рис. 1.1.) залучені дві незалежні сторони: емітенти та користувачі. Емісійна організація (емітент) випускає електронні гроші, які потім купують користувачі, оплачують із їхньою допомогою товари та послуги, а далі продавець погашає їх в емітента (обмінює електронні гроші на реальні). При емісії кожна грошова одиниця засвідчується електронним підписом емітента, який перевіряється перед погашенням.

Прикладами дебетових систем є: PayCash, Interplat, WebMoney та ін.[12].

Інтернет-кредитні системи (див. рис. 1.2.) є аналогами звичайних систем, що працюють із кредитними картами. Відмінність складається в проведенні всіх транзакцій через Інтернет, і як наслідок, у необхідності додаткових засобів безпеки та автентичності ("віртуальні" платіжні системи: WebMoney, Рапіда, E-port, Кредитпілот, банківські – FakturaPAY та CyberCheck).

Загальна схема платежів у кредитній онлайновій платіжній системі (рис. 1.2) полягає в наступному[9, 21, 26, 16].

1. Покупець на Web-сайті торговця (не обов'язково магазина) формує кошик товарів або послуг і вибирає спосіб оплати “кредитна карта”.

2. Далі параметри кредитної картки (номер, ім'я власника, дата закінчення дії та ін.) повинні надійти в оналайн платіжну систему для наступної авторизації. Це можна зробити двома шляхами:

– через магазин, це означає, що дані картки вводяться безпосередньо на сайті магазина, після чого їх отримає платіжна Інтернет-система Інтернет (2а);

– на сервері платіжної системи (2б).

3. Наступний крок залежить від того, чи веде банк-емітент онлайно-вую базу даних (БД) рахунків.

4. При наявності БД процесінговий центр передає банку-емітенту запит на авторизацію картки (4б) і потім одержує результат (4а). Якщо ж такої бази немає, то процесінговий центр сам зберігає відомості про стан рахунків

власників карток, стоп-лист та виконує запити на авторизацію. Ці відомості регулярно обновляються банками-емітентами.

5. Результат авторизації після цього подається в платіжну систему Інтернету.

6. Магазин одержує результат авторизації.

7. Покупцеві передається результат авторизації

– через магазин (7а);

– безпосередньо від платіжної системи Інтернету (7б).

8. Якщо результат авторизації – позитивний, то:

– магазин робить послугу, або відвантажує товар (8а);

– процесінговий центр передає в розрахунковий банк відомості про зроблену транзакцію (8б). Гроші з рахунку покупця в банку-емітенті перераховуються через розрахунковий банк на рахунок магазина в банку-екваєрі.

Існуючі на даний момент електронні платіжні системи по типі доступу до електронного рахунку можна розділити на дві великі групи:

– які потребують установки на комп'ютер користувача додаткового програмного забезпечення;

– платіжні системи що мають Web-інтерфейс.

Основними перевагами електронних платіжних систем є [9]:

– доступність – будь-який користувач має можливість відкрити власний електронний рахунок;

– мобільність – незалежно від місця свого знаходження користувач може здійснювати будь-які фінансові операції зі своїм рахунком;

– простота використання – для відкриття та використання електронного рахунку не потрібні спеціальні знання;

– оперативність – переклад грошових коштів з рахунку на рахунок відбувається за лічені хвилини.

Перелік основних платіжних систем, що використовуються в Україні та їх характеристики представлено в табл. 1.1.

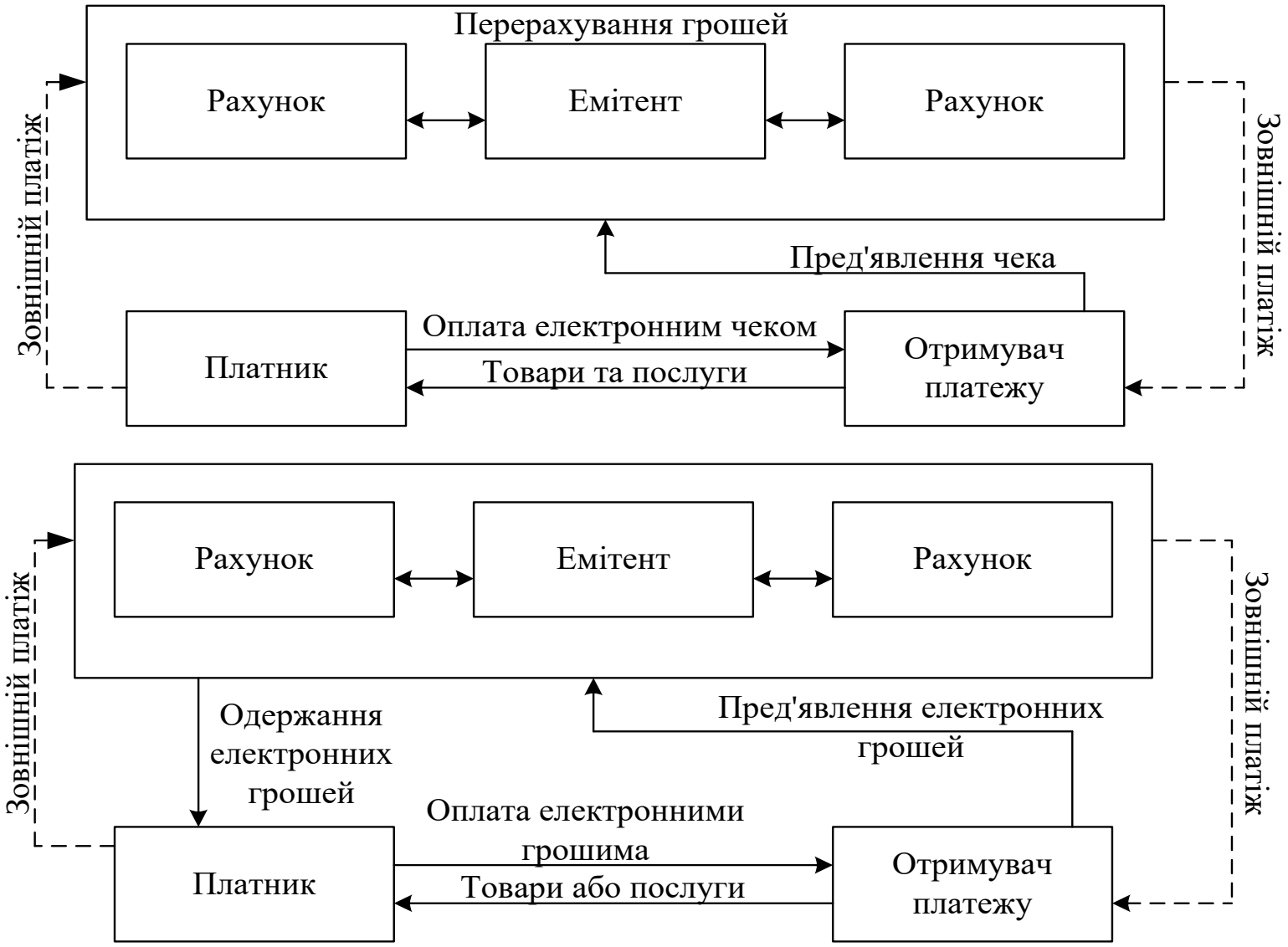


Рисунок 1.1 - Дебетові схеми платежів за допомогою електронних чеків та грошей

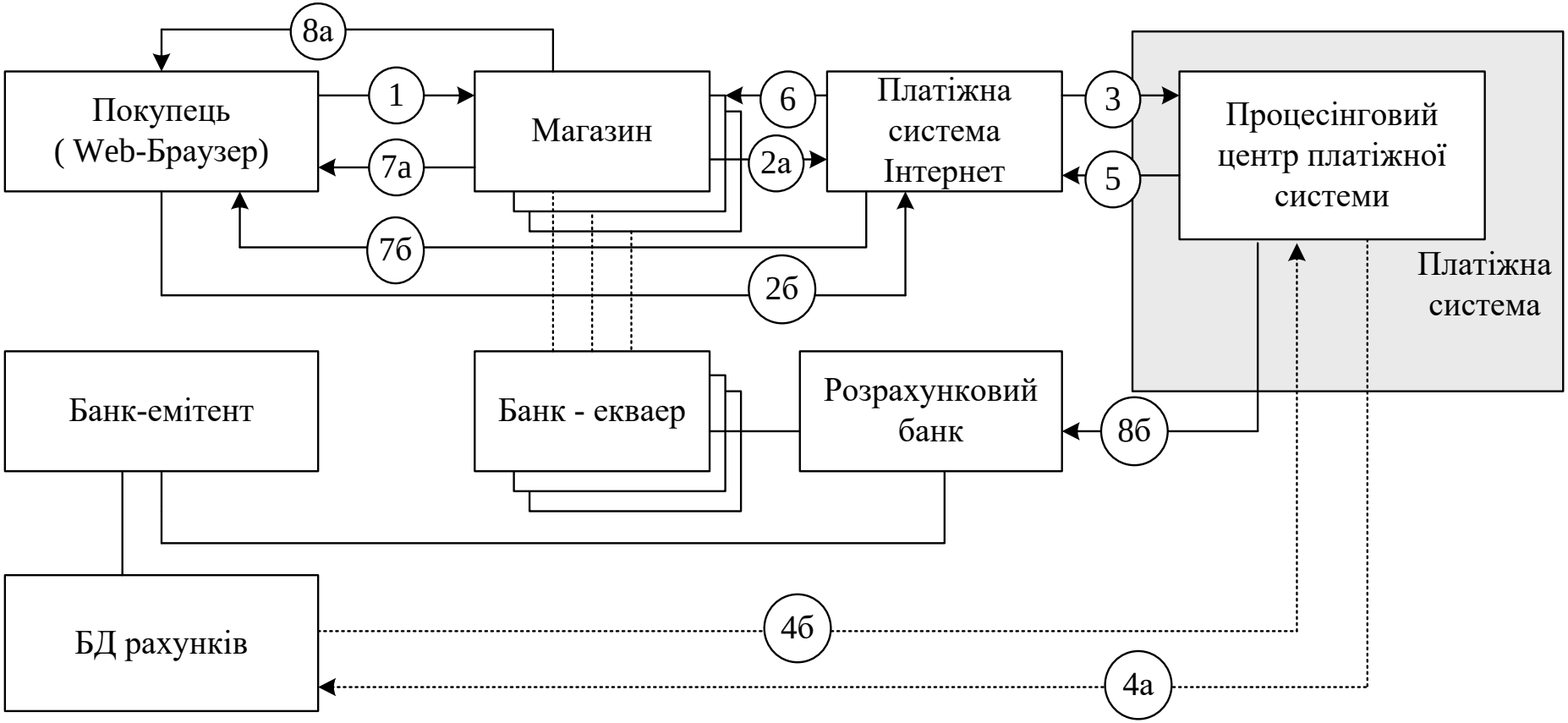


Рисунок 1.2 - Загальна схема платежів в Інтернет-кредитній системі

Таблиця 1.1 -Перелік Інтернет-платіжних систем , які використовуються в Україні

Найменування	Схема платежу	Властивості системи та її особливості
PayCash (www.imoney.com.ua)	дебетова	Анонімні платежі з використанням цифрової готівки, що зберігаються на жорсткому диску користувача; спеціальне програмне забезпечення інсталюється на ПК користувача. Платежі С2С дозволяються.
Interplat (www.interplat.com.ua)	дебетова	Предоплачена смарт-карта НСМЕП + карт-зчитувач у платника. Платежі С2С не дозволяються.
WebMoney (www.uawm.com)	дебетова	Анонімні платежі з використанням цифрової готівки, що зберігається на “жорсткому” диску користувача. Дозволяються платежі С2С. Сто тисяч користувачів в Україні.
Portmone (www.portmone.com)	кредитна	Електронна доставка та оплата рахунків з використанням платіжних карт Visa, MasterCard.
UkrMoney (www.ukrmoney.com)	кредитна	Ця система працює в співробітництві із КБ "Приватбанк", хоча, по суті, є небанківською. Електронний рахунок в UkrMoney – онлайн-відображення реального безособового рахунку.

Аналіз показав, що найбільшу частку на ринку Інтернет-платежів України займає ІПС Portmone.

Аналіз зростання Інтернет-платежів в Україні на прикладі ІПС Portmone (за даними компанії Воля-кабель Portmone на ринку Інтернет-платежів в Україні займає 93,2%) показав, що оборот системи в 2008 р. склав 206 млн. грн., що майже в 2 рази перевищує оборот 2007 року (111 млн. грн.). Кількість успішних транзакцій за 2008 рік склало 3,7млн. грн. Оборот за перших 9 місяців цього року склав 137,8 млн. грн. [9]. Тенденція росту кількості платежів представлене на рис.1.3., вона обумовлена тим, що кількість українців, що мають постійний доступ в Інтернет, збільшується з кожним днем, а останнім часом особливий внесок у це зробили мобільні оператори, надаючи послугу широкополосного мобільного Інтернету.

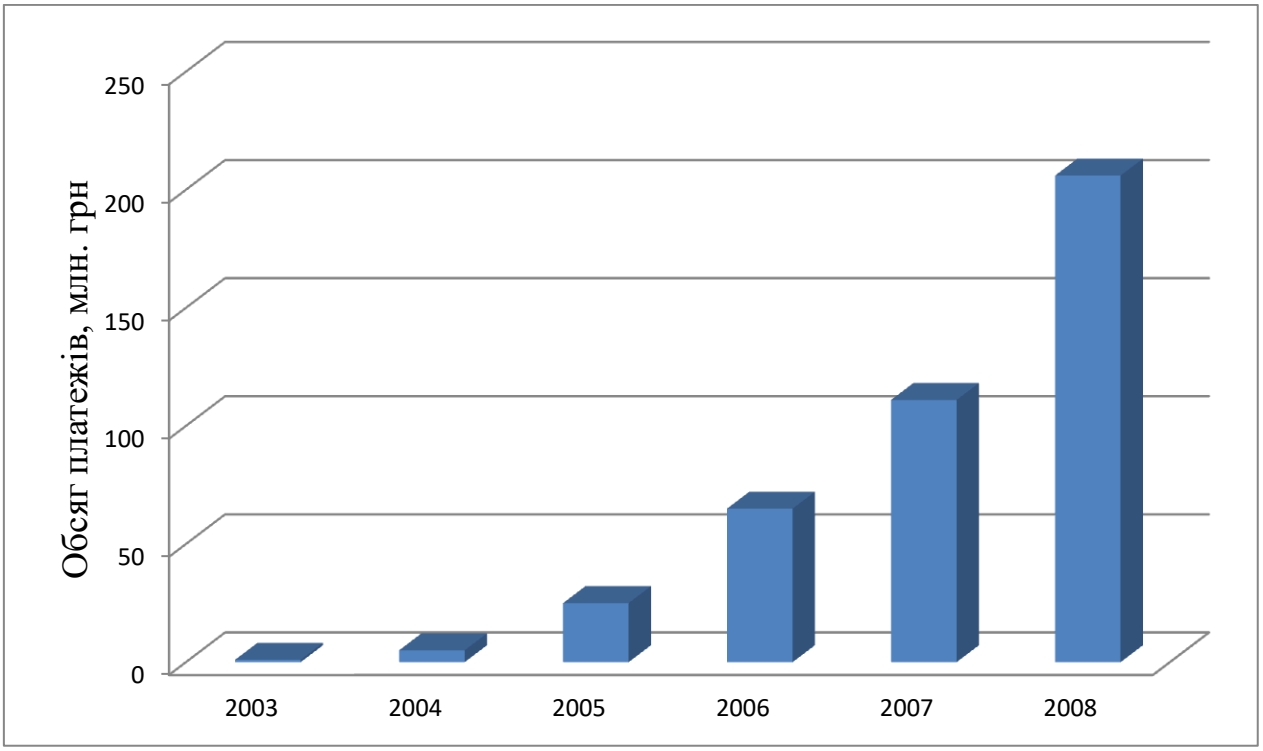


Рисунок 1.3 - Тенденція зростання обсягів платежів в ІПС Portmone

Аналіз структури Інтернет-платіжної системи дозволив зробити наступні висновки Інтернет платіжна система являє собою розподілену информаційно-розрахункову систему, призначену для обробки платежів за товари та послуги

виконуваних із застосуванням платіжних карт або з використанням електронної готівки в середовищі глобальної комп'ютерної мережі Інтернет.

Аналіз показує, що останнім часом загальний обсяг інформації, яка обробляється та передається в ІПС зріс багаторазово і загальні тенденції свідчать про те, що така динаміка зберігається. Разом з тим, стає гострішою проблема захисту інформації в електронній комерції, до якої призводять наступниі чинники:

– високі темпи зростання комп’ютерних обчислювальних можливостей, розширенням областей використання інформаційних технологій;

– велика кількість учасників інформаційної взаємодії (людей і організацій);

– ставлення до інформації, як до товару, переходу до ринкових відносин, з властивою їм конкуренцією і промисловим шпигунством;

– накопичення величезних обсягів інформації різного призначення на електронних носіях;

– інтенсифікація обміну інформацією між учасниками цього процесу;

– вдосконалення та доступність механізмів доступу користувачів до інформаційних ресурсів;

– зростання кваліфікованих користувачів в Інтернет-платіжних системах.

Ці чинники обумовили зростання кількості загроз та поширення злочинів в сфері електронної комерції, так званих “кіберзлочинів”.

Як загрозу для ІПС можна розглядати конкретну фізичну особу або подію, що представляє небезпеку для електронних платежів і приводить до порушення їхньої конфіденційності, цілісності, доступності та законного використання.

Загрози інформації в ІПС можна класифікувати наступним чином [2, 8, 26, 11]:

– об’єктивні, що характеризуються природнім впливом на об’єкт захисту, які не залежать від людини;

– суб’єктивні, пов’язані з діяльністю людини.

Серед останніх можливо виділити:

– ненавмисні, викликані помилковими діями користувачів чи співробітниками;

– навмисні, що є результатом навмисних дій порушників.

Навмисні погрози є найбільш чисельними у класифікації видів погроз. Їх структура представлена на рисунку 1.5.

Ці погрози пов'язані зі схильністю інформації фізичному перекручуванню або знищенню, можливістю випадкової або навмисної модифікації, а також небезпекою несанкціонованого одержання інформації особами, для яких вона не призначена. Діаграма розподілу погроз в ІПС представлена на рис. 1.4.[26]

Лідируючу позицію займають порушення конфіденційності даних, що приводять до витоку закритої інформації.

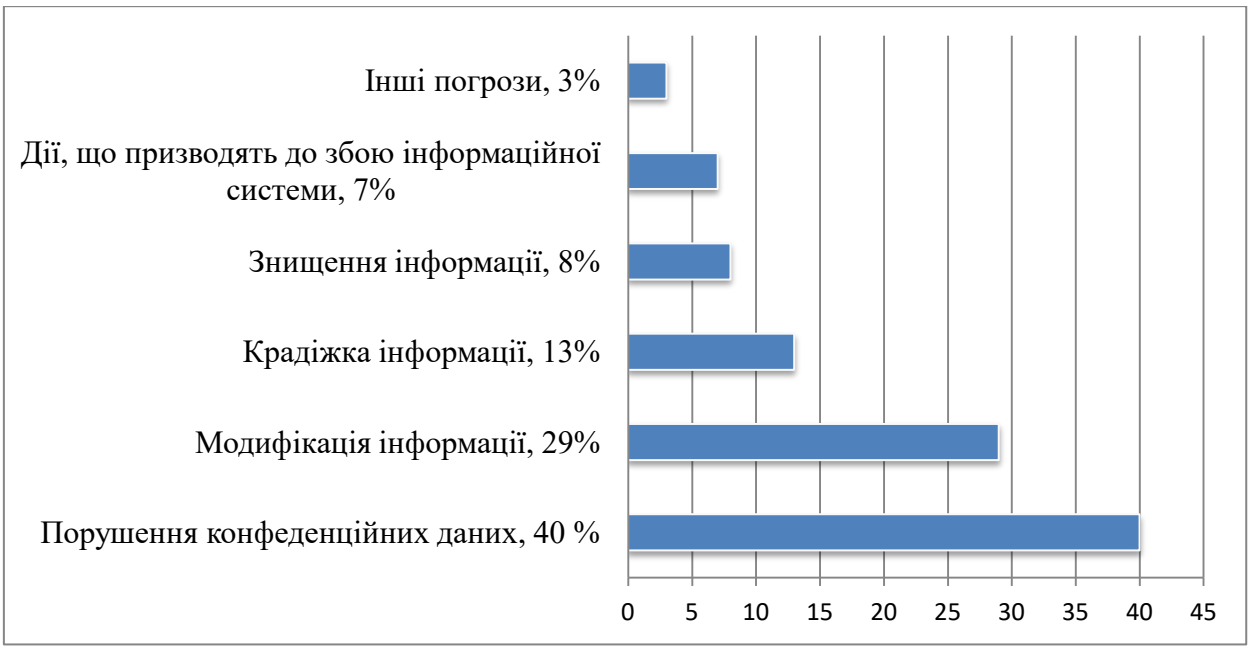


Рисунок 1.4 - Діаграма розподілення погроз в ІПС

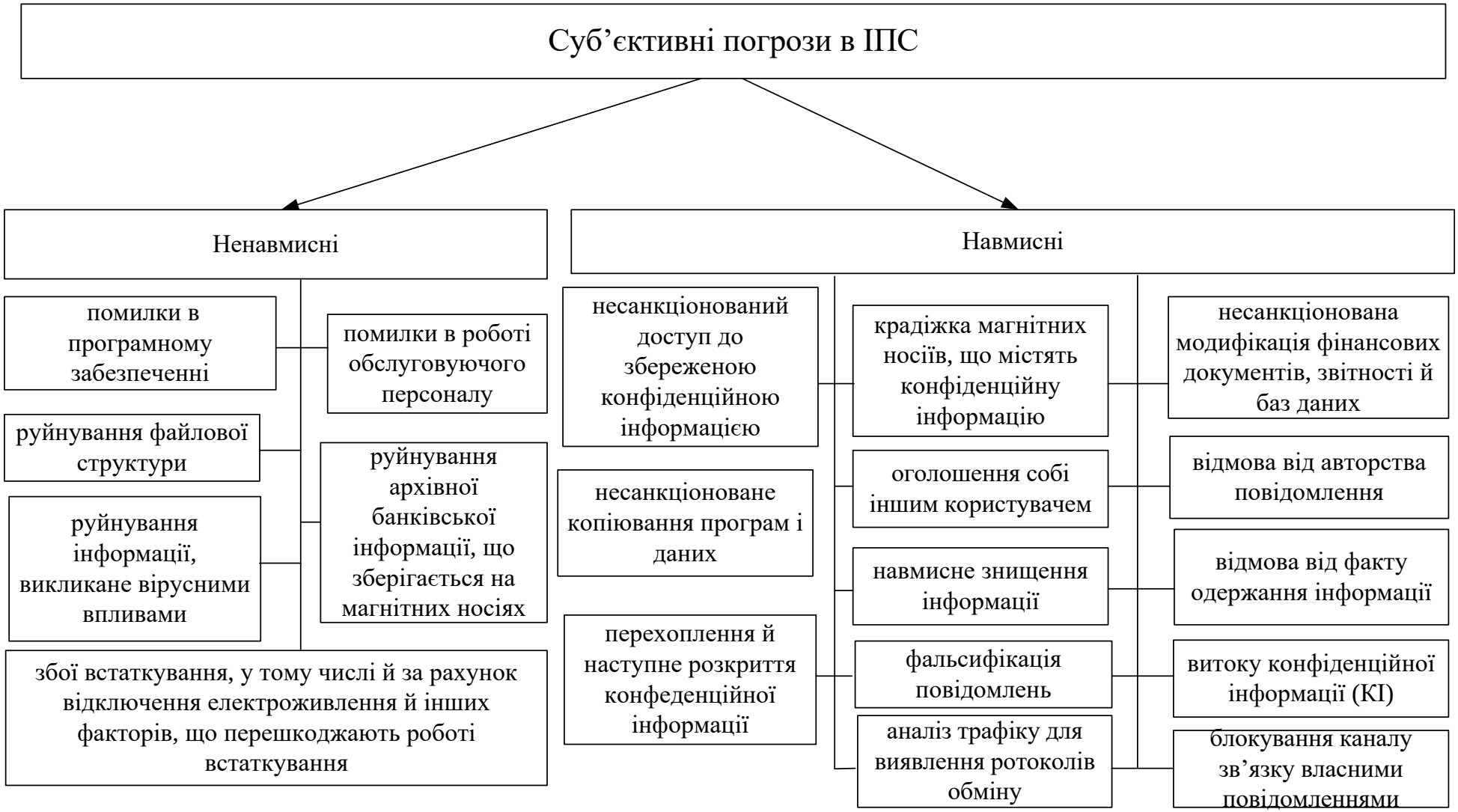


Рисунок 1.4 - Основні типи погроз інформаційних ресурсів

Таким чином, проведені дослідження показали, що найпоширенішими атаками на ІПС, що можуть заподіяти найбільший економічний збиток, є неправомірні дії, спрямовані на зниження конфіденційності, цілісності та автентичності інформації. Недооцінка проблем, пов'язаних з безпекою інформації в ІПС, може привести до величезних фінансових втрат.

Основними вимогами з інформаційної безпеки є [2, 8, 26, 11]:

– відсутність можливості списання грошових коштів з рахунку платника іншими особами;

– можливість підтвердження платником перед третіми особами (наприклад, в суді) факту здійснення платежу, його отримання і призначення даного платежу (наприклад, покупка товару належної якості);

– можливість підтвердження одержувачем грошей перед третіми особами факту одержання платежу і його призначення;

– можливість підтвердження емітентом факту проведення всіх авторизованих транзакцій за даним рахунком дійсним власником цього рахунку;

– гарантія того, що списана з рахунку сума не буде втрачена в момент передачі та потрапить точно та винятково за призначенням;

– виключення можливості підробки квитанцій емітента користувачами;

– забезпечення дозволу всіх спірних питань між емітентом і користувачами винятково електронним чином за допомогою повідомлень із цифровим підписом;

– забезпечення можливості дозволу спірних питань між користувачами без участі емітента; система в цілому повинна бути стійкою до шахрайських дій, у тому числі − у випадку форс-мажорних обставин.

Будь-які інтернет-платежі повинні бути конфіденційними. Тому необхідно, щоб платіжна система гарантувала конфіденційність, а надання

розширеної та додаткової інформації третім особом було прерогативою користувача.

Реалізація платіжної системи повинна бути простою, мати інтуїтивий інтерфейс та забезпечувати надійну роботу, оскільки порушення доступності сервісу може привести до великих фінансових втрат сторін.

Крім викладених вище вимог, будь-яка онлайн платіжна система повинна задовольняти вимоги по гнучкості, масштабованості та ефективності.

Для забезпечення безпеки в Інтернет-платіжних системах використовується технологія протоколу SSL і електронний цифровий підпис на основі криптографічного алгоритму RSA з довжиною ключа в 1024 біта, які забезпечують механізми цілісності та автентичності даних [1, 9, 26].

Разом з тим, домінування в Інтернеті традиційних платіжних систем (зі своїми стандартами захисту електронних транзакцій) не тільки перешкоджає подальшому розвитку електронної комерції, але і загрожує повністю зупинити ведення певних видів е-бизнесу. Результати досліджень, інформаційні повідомлення підтверджують цю думку [9, 11, 21].

Кількість онлайнових підроблених транзакцій, здійснюваних за допомогою кредитних/дебетових карт, в 12 разів вище, ніж в оффлайні. Ця різниця виливається в додаткові витрати, які наносять відчутну втрату прибутковості електронної комерції. У результаті дослідження, об'єктами якого стали 165 звичайних, Інтернет- та змішаних магазинів, було виявлено, що 1,15% всіх онлайнових покупок є підробленими в порівнянні з 0,06 – 0,09% в оффлайні. При цьому 64% від всіх chargebacks (операція повернення незаконно списаних коштів на картковий рахунок клієнта), пов'язаних з онлайновим бізнесом, є наслідком підроблених транзакцій, у той час як в оффлайні тільки 44% від всіх chargebacks пов'язані з підробкою.

Недовіра покупців до онлайнових платіжних систем є основною перешкодою для розвитку електронної комерції в різних країнах. Майже 40% всіх Інтернет-користувачів заявили, що острах “засвітити” свою кредитну/дебетову карту в Інтернеті є основним стримуючим від здійснення онлайнових покупок

фактором. Таким чином, підвищення безпеки платіжних систем є ключовим моментом для залучення клієнтів і підвищення довіри покупців до електронної комерції. У цілому, 28% користувачів Інтернет уважають, що основним способом зниження ризику є здійснення комерційних операцій на відомих сайтах, а 22% зволіли б купити товар, що сподобався, в оффлайновому магазині.

В 2003 році випадки шахрайства із кредитними/дебетовими картами складуть 14% від загального числа електронних транзакцій. Безпека електронних платежів з використанням кредитних карток стала основним бар'єром у розвитку онлайнових покупок для 79% користувачів, 73% споживачів стурбовані небезпекою Інтернет-торгівлі, а 83% не зважуються проводити онлайнові транзакції. Ускладнення процесу автентифікації електронних розрахунків, сумніви із приводу доставки та її висока вартість змушують багатьох потенційних покупців закривати сторінки із бланками замовлення.

Електронний бізнес не одержить достатнього розвитку доти, поки Інтернет-продавці не нададуть своїм клієнтам безпечні канали для проведення платіжних операцій. Англійське агентство по оцінці кредитоспроможності Experian провело дослідження, що показало, що 57% з 800 компаній, що займаються комерційною діяльністю в Інтернет, не повідомляють владу про випадки фінансового шахрайства, а половина з них уважають, що поліція взагалі не цікавиться такими ситуаціями. До судового розгляду доходить лише в 9% випадків, і в такий спосіб 9 з 10 онлайнових злочинів залишаються не тільки безкарними, але і непоміченими. 20% Інтернет-продавців заявляють, що незаконні операції становлять 1% від загального обсягу продажів, у той час, як деякі називають цифру в 10%. Майже половина всіх респондентів заявила, що вони не можуть бути до кінця впевнені в тому, що їх клієнт дійсно є тим, за кого він себе видає, і лише 15% компаній використовують у своїй діяльності автоматичні системи, що перевіряють дійсність карти [11]. Таким чином основною проблемою, з якої сьогодні зштовхнулася електронна комерція, є безпека платежів. І ситуація, у якій користувачі хочуть робити покупки, але побоюють-

ся шахрайства з боку продавця, буде тривати, поки довіра до систем електронних розрахунків не виросте.

Проведені дослідження показали, що Інтернет одночасно є і надзвичайно ефективним комунікативним засобом і середовищем, що викликає досить велику недовіру у користувачів, безпека електронних платежів є досить серйозним критерієм успіху конкретної системи і електронного бізнесу, що її використовує. У цьому зв'язку на перший план виходять надійність та безпека, здійснення платежів в Інтернет-платіжних системах.

Система захисту інформації повинна забезпечувати високий рівень інформаційної безпеки ІПС на кожному етапі підготовки, обробки та виконання транзакцій за рахунок покладеного в її основу комплексного підходу до проблеми забезпечення захисту.

Створення захищеного середовища обробки інформації реалізується на декількох рівнях [8 – 12]:

– перший − рівень захищеного операційного середовища, що забезпечує авторизований доступ до файлів, каталогів і програм окремо на читання, зміну та запуск і задовольняє загальновизнаному у світі рівню безпеки;

– другий − рівень система управління базами даних (СУБД), що забезпечує авторизований доступ до інформації в базі даних окремо на читання, додавання та модифікацію, а також автоматичне ведення протокольних журналів роботи користувачів;

– третій − рівень прикладного програмного забезпечення ІПС, на якому реалізовані підсистеми: внутрішнього аудиту; розмежування доступу до інформації та керування правами користувачі;

– четвертий − рівень засобів криптографічного захисту інформації, який реалізує послуги безпеки з використанням відповідних механізмів.

Проаналізуємо існуючі послуги та механізми захисту інформації, розглянемо можливі підходи до забезпечення цілісності та автентичності інформації в ІПС.

Послуга безпеки є абстрактним поняттям, яку може бути використане для характеристик вимог безпеки.

Відповідно до міжнародних стандартів ISO 7498, ISO/IEC 10181 визначені п'ять базових загальноприйнятих послуг безпеки [1 – 4, 8, 9, 16]:

1. Автентифікація забезпечує гарантію надійної ідентифікації джерела повідомлення або електронного документа, а так само того, що джерело не є підробленим.

2. Управління доступом забезпечує можливість контролю доступу до інформаційних ресурсів або самою системою, що володіє ресурсами, або системою, який ці ресурси надаються.

3. Конфіденційність даних забезпечує захист від несанкціонованого одержання інформації.

4. Цілісність даних забезпечує гарантію модифікації інформації, яка пересилається по каналам, тільки тими суб'єктами, які мають на це право.

5. Приналежність забезпечує доказ приналежності з підтвердженням дійсності джерела повідомлень і доказ приналежності з підтвердженням доставки.

Для реалізації послуг безпеки в ІПС використовуються наступні механізми:

– механізм цифрового підпису;

– безключові хеш-функції, що представляють собою механізми забезпечення захисту цілісності даних, які включають криптографічні контрольні функції;

– ключові хеш-функції, що представляють собою механізми аутентифікації;

Взаємозв'язок послуг і механізмів безпеки наведений в таблиці 1.2.

Таблиця 1.2 -Взаємозв'язок послуг і механізмів безпеки в ІПС

Послуги безпеки	Механізми безпеки		
	безключові хеш-функції	ключові хеш-функції	цифровий підпис
автентифікація абонентів;	ні	так	так
неможливість відмови від авторства повідомлення;	ні	так	так
контроль цілісності повідомлення;	так	так	так
забезпечення конфіденційності повідомлення;	ні	ні	так
реєстрація послідовності повідомлень;	так	так	так
контроль цілісності послідовності повідомлень;	так	так	так
забезпечення конфіденційності потоку повідомлень.	ні	ні	так

Стандарти ISO 7498-2 та ISO/IEC 10181 кажуть, що одним із найкращих способів забезпечення автентичності даних і джерел повідомлень а також цілісності, є механізм цифрового підпису (ЦП) [23, 35]. Розглянемо алгоритми ЦП, які використовуються для забезпечення автентичності в Інтернет-платіжних системах.

1.2 Аналіз алгоритмів цифрового підпису в Інтернет-платіжних системах

Цифровий підпис представляє собою рядок даних, що формується на основі деякого секретного ключа особи, що підписує, і залежить від змісту повідомлення, що підписується, представленого в цифровому виді [3 - 8].

Таким чином, цифровий підпис повідомлення – це блок даних невеликого розміру, отриманий у результаті криптографічного перетворення повідомлення довільної довжини з використанням особистого ключа відправника [8], що зв'язує повідомлення з деяким що породжує або підписує його об'єктом.

Система ЦП включає дві процедури: 1) накладання підпису; 2) перевірки дійсності підпису (рис. 1.6). Відправник підписує повідомлення своїм за-

критим ключем, а отримувач перевіряє дійсність підпису відкритим ключем асиметричної системи. У пепроцедурі генерації підпису використається секретний ключ відправника повідомлення, у процедурі верифікації підпису – відкритий ключ відправника. Секретний ключ зберігається абонентом у таємниці та використається їм для формування ЦП. Відкритий ключ відомий всім користувачам мережі [1, 2].

Оскільки використовувані на практиці схеми електронного підпису не пристосовані для підписання повідомлень довільної довжини, а процедура, що складається в розбивці повідомлення на блоки та у генерації підпису для кожного блоку окремо, вкрай неефективна, тому схеми підпису використовуються до хеш-коду повідомлення.

У такий спосіб при формуванні ЦП відправник обчислює хеш-функцію h(M) тексту, що підписує, M, призначену для стиску документа, що підписує, M до декількох десятків або сотень біт (фіксованої довжини). Потім кількість m шифрується на особистому ключі відправника. Одержувані при цьому пари чисел являє собою ЦП для даного тексту M (рисунок 1.6).

При верифікації ЦП одержувач повідомлення знову обчислює хеш-функцію h* = h(M') прийнятого по каналі вихідного тексту M (можливо зміненого), після чого за допомогою відкритого ключа відправника перевіряє, чи відповідає отриманий підпис обчисленому значенню хеш-функції m' = m'' [1, 3].

В ІПС використовуються наступні алгоритми цифрового підпису: RSA, DSA, ЦП ДСТУ-4145 і т.д.

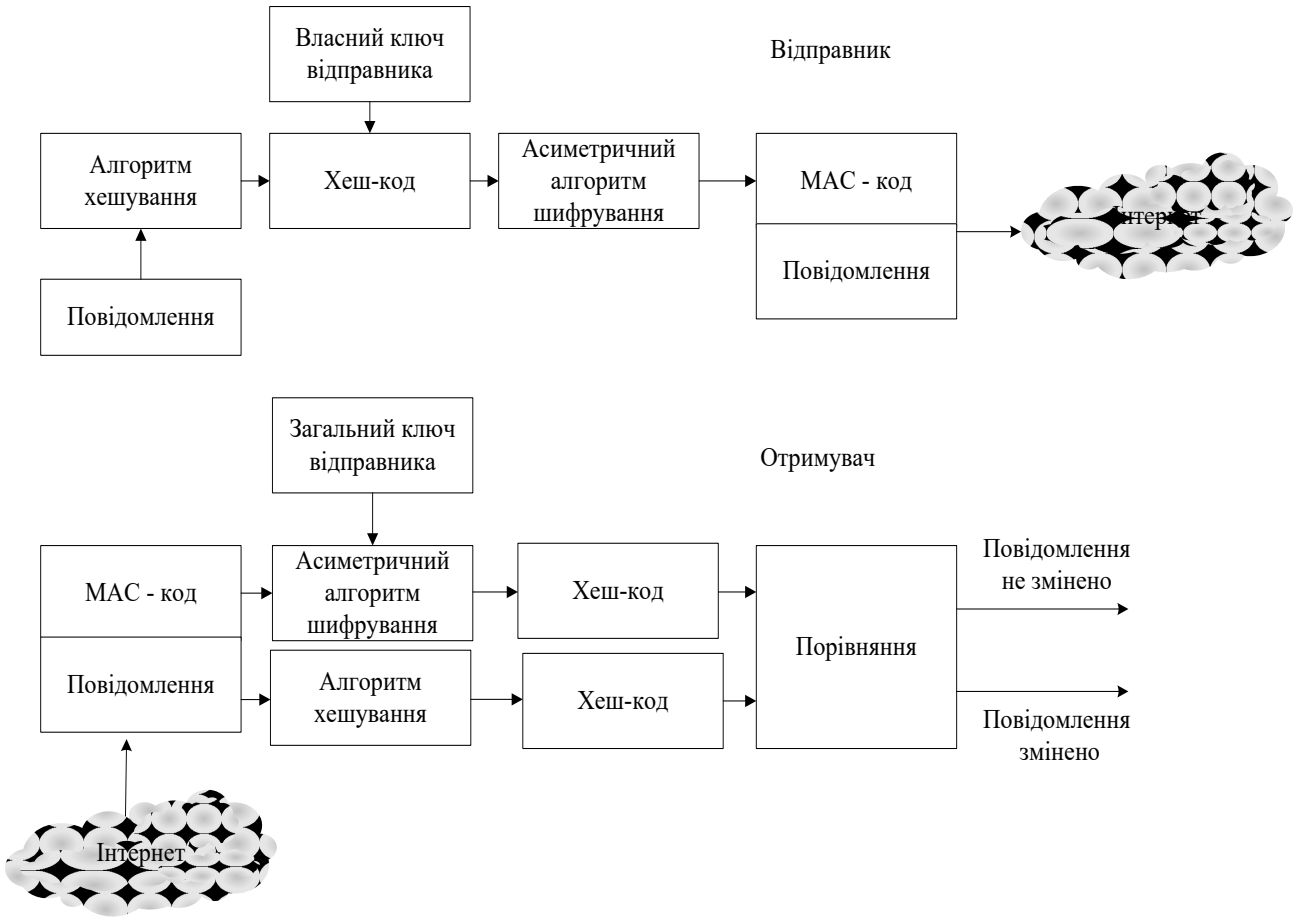


Рисунок 1.6 - Реалізація процесів створення та верифікації цифрового підпису

Алгоритм RSA [2 - 5, 37] ґрунтується на NP-повній задачі знаходження факторизації (розкладання цілого параметра n на добуток двох різних простих чисел приблизно рівних один по одному величини). Зазвичай в системах цифрового підпису на основі алгоритму RSA застосовують довгі цілі числа n (звичайно від 75 до 128 байт).

Крім того, при генерації і обчисленні ключів у системі RSA потрібно, щоб прості числа p і q задовольняли ряд додаткових вимогСхема формування ЦП RSA представлена на рис.1.7 [15, 16, 19].

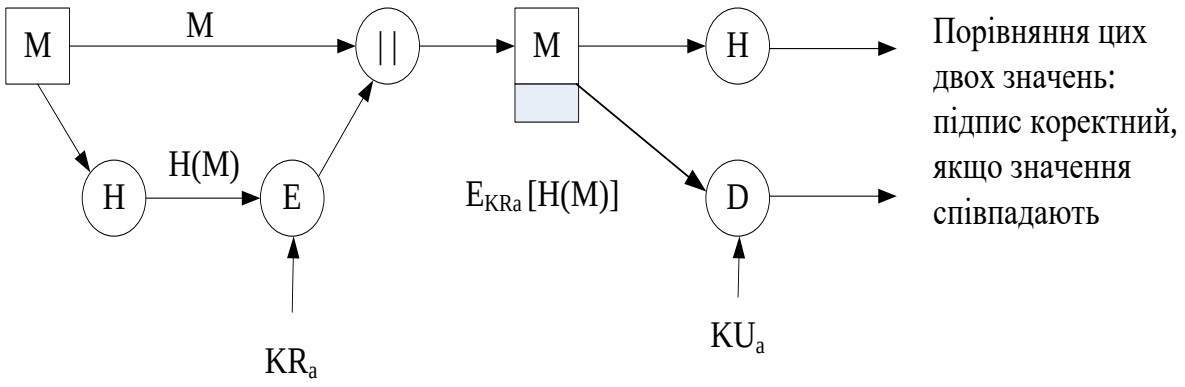


Рисунок 1.7 - Створення і перевірка підпису за допомогою алгоритму RSA

Алгоритм DSA [26, 27, 30]. Алгоритм DSA, що став надалі основою національного стандарту США на цифровий підпис має в порівнянні з алгоритмом RSA цілий ряд переваг:

– при заданому рівні стійкості цифрового підпису цілі числа, з якими доводиться проводити обчислення, є на 20% меншими, що відповідно зменшує складність обчислень не менш, ніж на 70% і дозволяє помітно скоротити об'єм використовуваної пам'яті;

– при виборі параметрів досить перевірити всього три умови;

– процедура підписування за допомогою цього методу не дозволяє обчислювати (як це можливо в RSA) цифрові підписи під новими повідомленнями без знання секретного ключа.

У порівнянні з оригінальним алгоритмом Ель Гамаля метод DSA має одну важливу перевагу – при заданому в стандарті рівні стійкості, числа, що беруть участь в обчисленні підпису, мають довжину 20 байт кожне, скорочуючи загальну довжину підпису до 40 байт.

Оскільки більшість операцій при обчисленні підпису і її перевірці також виробляється по модулю з 20 байт, скорочується час обчислення підпису та об'єм використовуваної пам'яті.

В алгоритмі Ель Гамаля довжина підпису при такому рівні стійкості дорівнювала б 128 байт.

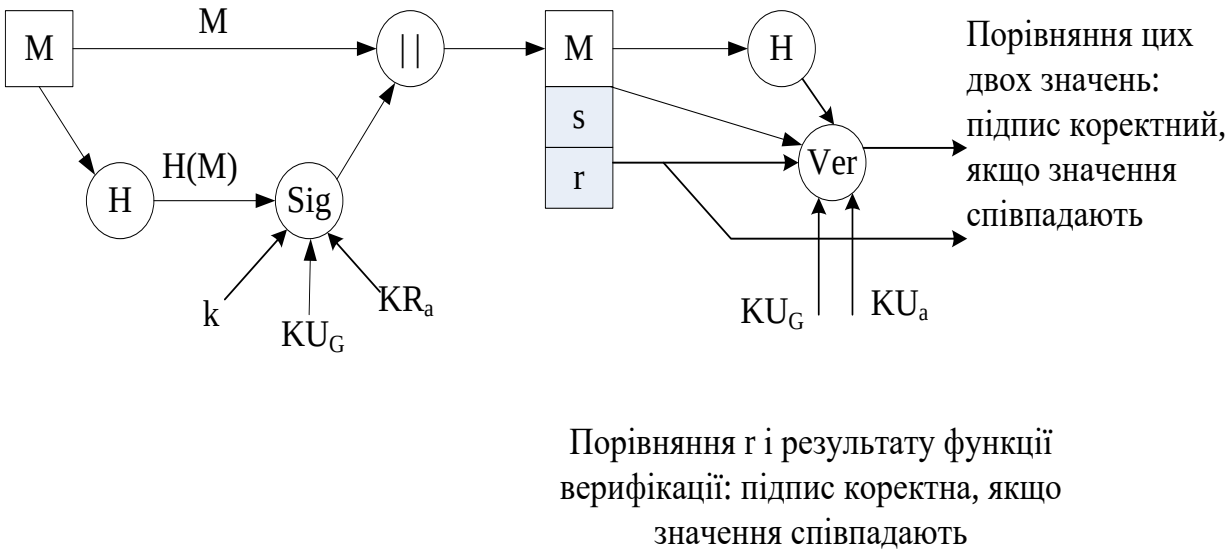


Рисунок 1.8 - Створення і перевірка підпису за допомогою алгоритму DSA

Алгоритм ЦП ДСТУ-4145. Алгоритм обчислення і перевірки ЦП заснований на властивостях груп точок еліптичної кривої над полями GF (2m). Стійкість алгоритму заснована на NP-повній задачі знаходження дискретного логарифма еліптичної кривої [8] (знаходження значення k по базовій точці P та розташованої на кривій точці k). При цьому алгоритми на основі еліптичної кривої використають ключі малих розмірів, що знижує вимоги до обчислювальних потужностей у порівнянні з вимогами до алгоритмів на основі RSA.

Розглянемо вплив процесів забезпечення цілісності та автентичності інформації на ефективність ІПС. Один з варіантів здійснення платежів з використанням електронних грошей на базі стандарту електронних платежів EMV (Europay, MasterCard, VISA) [6, 8] представлений на рис. 1.9.

Згідно рис. 1.9 клієнт А, що бажає оплатити товарі або послуги послуги в електронному магазині звертається до Інтернет-платіжній системі (1). Потім через процессинговый центр А посилає в банк-емітент, у якому є його рахунок, електронну купюру – документ (номінал і номер купюри) (2). Документ підписаний А на його ключі ksecret. Маючи в наявності ключ kpublicA,банк-емітент перевіряє підпис А.

У випадку її дійсності знімає з рахунки А цю суму та підписує документ своїм підписом (4). Одержавши електронну купюру банку А може розплатитися з В, з умовою, що всі троє перебувають на зв'язку (5). Для цього А підписує купюру (7), а В, перевіривши дійсність підпису, видаляє її та ставить свою (8), після чого через розрахунковий банк (10) відправляє в банк-екваер. Банк перевіряє наявність купюри в списку використаних і якщо її не було, то переводить суму рівну номіналу на рахунок В (11).

Процес обслуговування транзакції у ІПС, наприклад, при покупці якого-небудь товару, можна представити наступними етапами:

1) запит клієнта через термінал (банкомат, електронний магазин і ін.);

2) формування контрольної послідовності (підпис) клієнтом А для ідентифікації та автентифікації в банку-емітенті;

3) передача даних у розрахунковий процесінговий центр (РПЦ1);

4) перевірка контрольної послідовності (підпису) у відповідному РПЦ1;

5) формування контрольної послідовності (підпису) РПЦ1 (банком-емітентом);

6) передача даних від РПЦ1 у ГПЦ;

7) перевірка контрольної послідовності підпису в ГПЦ;

8) формування контрольної послідовності (підпису) у головний процесінговий центр (ГПЦ) і передача в РПЦ2 (банк-екваер);

9) перевірка підпису РПЦ2 (банком-екваером).

Проведемо оцінку залежності часу обслуговування транзакцій ІПС від обчислювальної складності криптоперетворень. Складність базових RSA-Перетворень із довжиною ключа 512 біт складе порядку 768 примітивних операцій на 32-разрадяній платформі. Типова ЕОМ з тактовою частотою процесора − 1 ГГц потенційно дозволяє проводити додавання 32-розрядних чисел за 10-9 с. Час множення більших чисел, для обчислення RSA-модуля з довжиною ключа 512 біт складе 0,768 мс, 768 біт − 1,15 мс, 1024 біта − 1,536 мс [13, 63, 65]. У середньому час передачі транзакцій довжиною 1, 5−2 Кбіт по ви-

користовуваним сьогодні каналах між терміналом і РПЦ зі швидкістю 16 − 32 кбіт/с складе порядку 0,025 − 0,05 с.

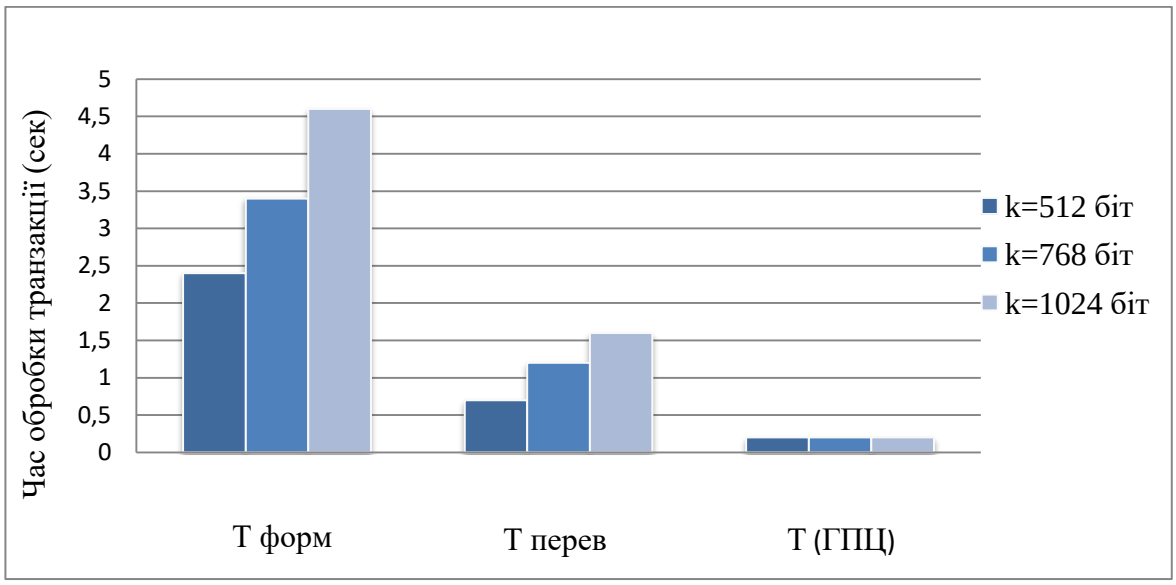
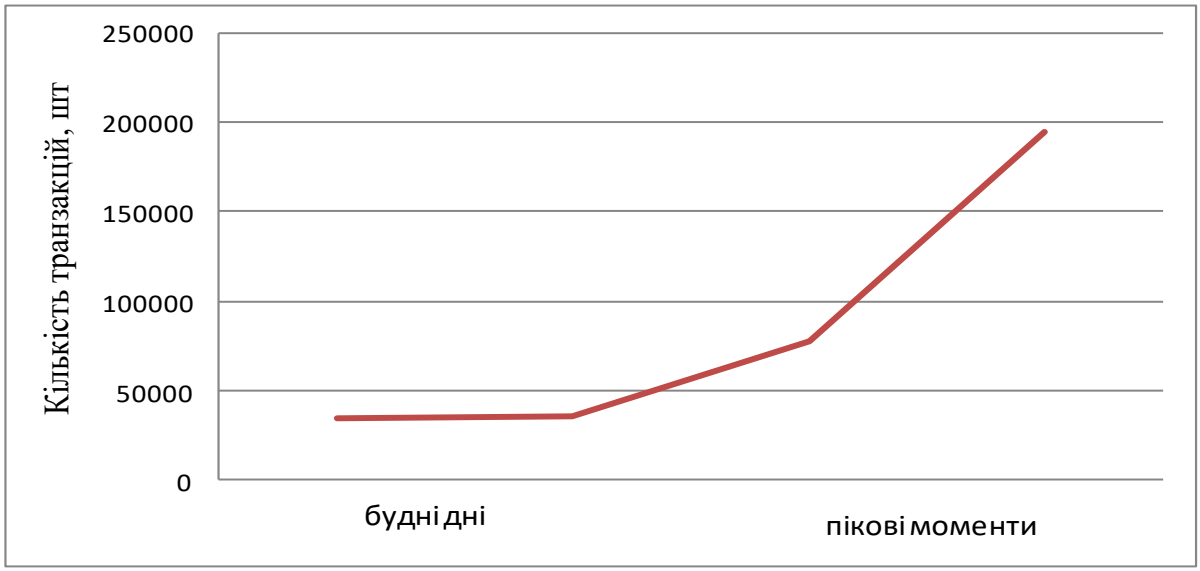


Рисунок 1.9 - Вплив довжини ключа на час обробки транзакції в ІПС

Загальна чисельність користувачів ІПС Portmone досягає 300 000 чоловік, з них 120 000 є активними (користувачі, які хоча б раз в три місяці користуються послугами системи). Згідно цих даних були проведені розрахунки та побудована крива зростання кількості транзакцій в пікові моменти, яка наведена на рис. 1.10.



Риунок 1.10 - Зростання кількості транзакцій в пікові моменти

При збільшенні довжини ключа RSA-подібної схеми автентифікації з 512 біт до рекомендованої довжини 1024 біт інтенсивність обслуговування Інтернет-платіжної системи знижується на 27%, що в пікові моменти з урахуванням збільшення транзакцій у кілька разів призведе до збою в системі.

Таким чином виникає протиріччя між зростанням кількості транзакцій між користувачами ІПС та банківськими установами, які підлягають криптографічним перетворенням, та лімітом часу для їх обробки підсистемами захисту інформації в автоматизованих банківських системах. Таким чином, актуальною задачею є розробка метода ключового хешування з використанням арифметики еліптичних кривих для забезпечення цілісності та автентичності інформації, яка циркулює в Інтернет-платіжних системах.

1.3 Висновки до першого розділу:

1. Інтернет платіжна система відноситься до складних багаторівневих систем управління критичного застосування, у яких передача інформації вимагає контролю безпеки на кожному рівні. Зростає кількість транзакцій, а з ними і обсяг інформації, що циркулює в ІПС.

2. Проведені дослідження показали, що найпоширенішими атаками на ІПС, що можуть заподіяти найбільший фінансовий збиток, є неправомірні дії, спрямовані на зниження конфіденційності, цілісності та автентичності інформації. Недооцінка проблем, пов'язаних з безпекою інформації в ІПС, може привести до величезних фінансових втрат.

3. Таким чином виникає протиріччя між зростанням кількості транзакцій між користувачами ІПС та банківськими установами, які підлягають криптографічним перетворенням, та лімітом часу для їх обробки підсистемами захисту інформації в Інтернет-платіжних системах.

РОЗДІЛ 2 УДОСКОНАЛЕННЯ МЕТОДУ КЛЮЧОВОГО ХЕШУВАННЯ З ВИКОРИСТАННЯМ АРИФМЕТИКИ В ГРУПІ ТОЧОК ЕЛЕПТИЧНОЇ КРИВОЇ

Проведено аналіз перспективних методів забезпечення цілісності та автентичності інформації. Представлена класифікація хеш-функцій та порівняльний аналіз ключових і безключових хеш-функцій. Досліджується алгоритм побудови ключової функції хешування з використанням модулярної арифметики.

Проводитися аналіз характеристик ключових хеш-функцій побудованих на основі модулярної арифметики. Розробляються рекомендації до подальшого вдосконалення.

2.1. Аналіз перспективних методів забезпечення цілісності та автентичності інформації

Криптографічні хеш-функції відіграють фундаментальну роль у криптографічному захисту інформації, яка циркулює в ІПС. Особливо широко вони використовуються при забезпеченні цілісності даних і автентифікації повідомлень. Широко поширені наступні методи забезпечення дійсності повідомлення[1 – 4, 26]:

– додавання до повідомлення коду дійсності повідомлення (код автентифікації повідомлення) (message authentification code, MAC-код) або зашифрованої контрольної суми;

– введення цифрових підписів.

Одним зі складових елементів механізмів безпеки реалізуючих функції цілісності, аутентифікації і причетності саме і є хеш-функції.

Хеш-функція бере на вхід повідомлення і породжує на виході деякий образ цього повідомлення, що називається хеш-кодом, хеш-результатом, хеш-значенням або просто хеш. Або більш точно, хеш-функція h відображає двійко-

вий рядок довільної кінцевої довжини m у двійковий рядок фіксованої довжини, скажемо n. У криптографії використовується саме ця ідея, тобто коли хеш-код виступає в ролі компактного представлення (образа) деякого вхідного рядка, за яким можна точно ідентифікувати вихідне повідомлення.

Хеш-функції можна розділити на два класи [42 – 50]: безключові хеш-функції, тобто хеш-функції на вхід яких подається тільки повідомлення і ключові хеш-функції, тобто хеш-функції на вхід яких подається повідомлення і секретний ключ.

Для подальших міркувань наведемо наступне визначення хеш-функції.

Хеш-функція [16], у самому загальному змісті, є функція h(x), яка як мінімум має такі дві властивості:

– стиск – тобто функція h відображає вхідний рядок x кінцевої довільної довжини у вихідний рядок y = h(x) фіксованої довжини n;

– легкість обчислення – при відомій h і вхідному рядку x легко обчислити h(x).

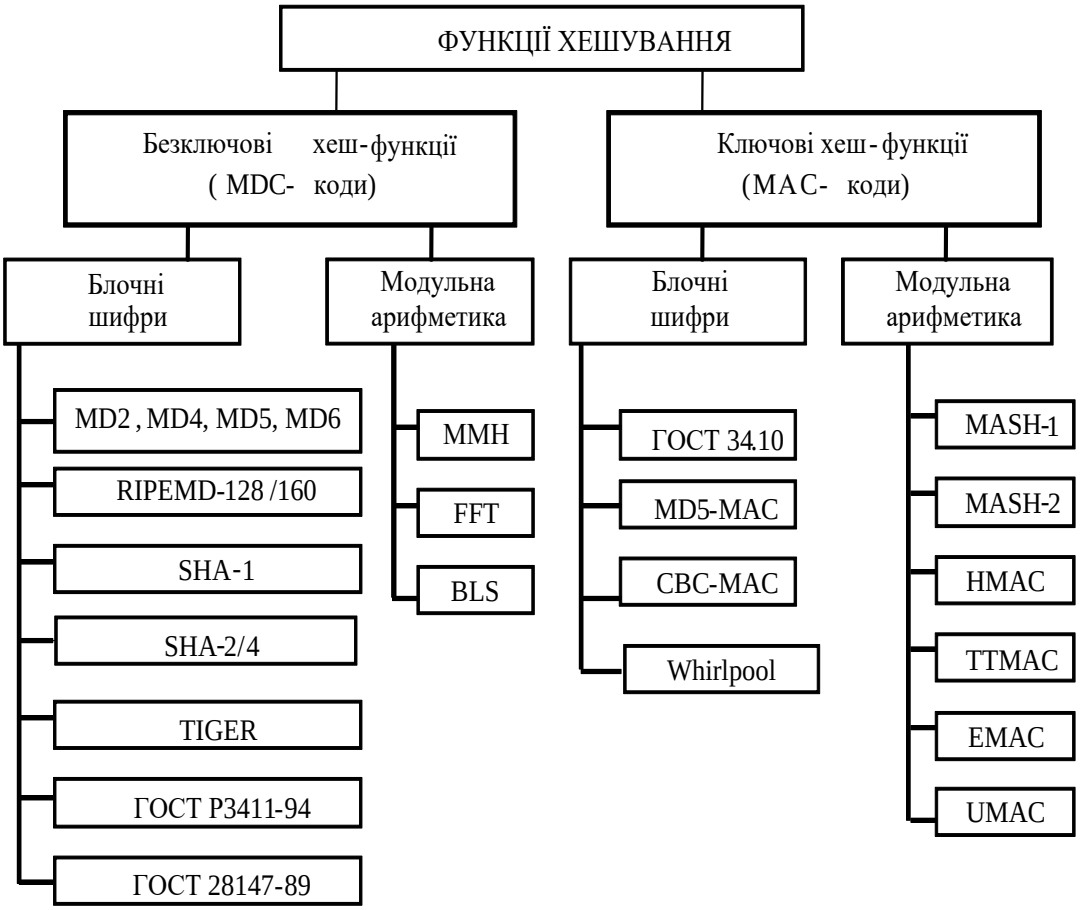


Рисунок 2.1 - Класифікація способів хешування

До безключових хеш-функцій відносяться коди виявлення змін повідомлення (MDC-код, modification detection code), також відомі як коди виявлення маніпуляцій над чи повідомленнями коди цілісності повідомлень. MDC-коди призначені для формування стиснутого образу або хеш-кода повідомлення, що задовольняє спеціальним властивостям. В остаточному підсумку MDC-коди забезпечують, разом з іншими механізмами, цілісність даних. У свою чергу MDC-коди можуть бути розбиті на однобічні хеш-функції, для яких складно знайти вхідне значення за відомим хеш-кодом і стійкі до зіткненням хеш-функції, для яких складно знайти два вхідних значення, що мають той самий хеш-код. Безключові хеш-функції є одним зі складових елементів цифрових підписів.

Розглянемо найбільш поширені безключові функції хешування SHA-1, SHA-2, MD4, MD5, MD6, RIPEMD-128, RIPEMD-160 та ГОСТ-8147 [42 - 54]. Їх характеристики наведені в табл. 2.1.

Таблиця 2.1 - Порівняльна характеристика безключових хеш-функцій

Характеристика	MD5	MD6	SHA-1	SHA-2 (256/512)	ГОСТ 28147-89	RIPEMD-128	RIPEMD-160
Довжина дайджесту, біт	128	512	160	256/512	256	128	160
Розмір блоку обробки, біт	512	512	512	512/1024	512	512	512
Кількість ітерацій	64	168	80	64/80	32	128	160
Кількість елементарних логічних функцій	4	4	3	6/6	8	5	5
Кількість додаткових констант	64		4	64/64	-	4	4
Швидкість роботи на Pentium Ш 1000 MHz, Мбіт/с	574,64		344,43	135,5/68,7	315,27	63.8	39.8

Однією з найважливіших властивостей хеш-функцій є їх захищеність від лобової атаки, яка залежить від довжини дайджесту. Аналіз показав, що найбільшу довжину дайджесту мають алгоритми SHA-2 та MD6, які є вдосконаленими алгоритмами ранніх версій. Збільшення довжини дайджесту веде до збільшення не тільки кількості операцій (в MD6 у порівнянні з MD5 кількість ітерацій збільшено в 2,5 рази) чи разом з зміною блоку обробки вхідної послідовності (в SHA-2 при зміні довжини дайджесту з 256 до 512 біт довжина блоку обробки змінилась з 512 до 1024 та кількість операцій зросла в 1,25 рази).

Не менш важливим параметром є швидкість роботи алгоритму. Вона залежить від складності та кількості операцій необхідних для вироблення хеш-коду, а також від розміру обробляє мого блоку. Проведень аналіз безключових функцій дозволивши виявити наступну тенденцію при збільшенні довжини дайджесту в 2 рази швидкість формування хеш-коду знижується також приблизно в 2 рази.

До ключових хеш-функцій відносяться MAC-коди [16]. Вони призначені для забезпечення цілісності даних і автентифікації повідомлень на основі вико-

ристання механізмів симетричної криптографії. MAC-код - хеш-функція з двома вхідними параметрами (повідомленням і секретним ключем). На виході такого алгоритму формується хеш. Якщо людина не знає ключа, то неможливо отримати такий самий хеш для цього ж повідомлення

Для вивчення основних характеристик MAC-кодів і проведення їх порівняльної оцінки розглянемо основні практичні алгоритми MAC-кодів:

– Two-Track-MAC: K.U.Leuven, Бельгія та Debis AG, Німеччина;

– UMAC: розробка корпорації Intel , Університету з штату Невада в Рено, Науково-Дослідної лабораторії IBM, Університету з Каліфорнії (США) і Technion, (Ізраїль);

– CBC-MAC (ISO/IEC 9797-1);

– HMAC (ISO/IEC 9797-1);

– MASH-1 та MASH-2 (ISO/IEC 10118-4).

UMAC алгоритм відомий у модифікаціях UMAC(1999) [3] і UMAC(2000) [4, 41], забезпечує надзвичайно високу швидкість обчислень. Розробники UMAC переслідували дві головні цілі: швидкість обчислень; доказову таємність. Рішення цих завдань виявилося можливим на основі застосування композиційної схеми з багаторазовим універсальним хешуванням і криптографічним обчисленням тегу аутентификации.

UMAC (1999) обчислює MAC код шляхом попереднього стиску повідомлення із установленим відношенням, використовуючи NH універсальне сімейство хеш-функцій. NH хешування обчислюється для повідомлення, попередньо розбитого на блоки фіксованої довжини (за винятком останнього блоку, що може бути коротше). Для блоку з 1024 слів по 32 біта, можна одержати значення хеша в 64 біт, що визначає коефіцієнт стиску 512. Показник новизни разом з хеш-значеннями всіх блоків і інформацією про довжину поєднуються в один рядок. Результуючий рядок обробляється псевдовипадковою функцією PRF для одержання автентифікованого тегу. У якості PRF функції застосовується одна із криптографічних хеш-функції в режимі CBCMAC або HMAC.

Two-Track-MAC (також відомий як TTMAC) – код аутентификации повідомлення. Алгоритм працює на блоках 512 біт, розділених на слова по 32 біт, використовує секретний ключ 160 біт, і робить вихід до 160 біт.

Проект Two-Track-MAC заснувань на хеш-функції RIPEMD-160 з модифікаціями. Спочатку, повідомлення, що буде завірено доповнюється 1 бітною частиною, і потім 0-бітами, поки його довжина не стані рівної 448 по модулі 512. Двійкове подання довжини первісного повідомлення (mod 2^{64}) додається в кінець, так що довжина повідомлення стає кратної 512. Кожний блок з 512 бітами розбивається на шістнадцять слів по 32 біта $W_0,\ldots,W_{15}$.... . Секретний ключ ця безліч із п'яти слів з 32 бітами $K_0,\ldots,K_4$ [18, 27, 42 - 54]

У табл. 2.2 представлені основні результати по оцінці швидкодії Two-Track-MAC алгоритму для різних операційних платформ [5, 12]. Швидкість обчислень визначається кількістю циклів процесора, затрачуваних на один байт оброблюваного повідомлення.

Таблиця 2.2 -Швидкодія алгоритмів формування MAC кодів

Алгоритм	Дов-жина MAC коду (біт)	Довжина ключа (біт)	Тип ПЕОМ				
			Pentium 2	PIII/Linux	Pentium 4	Xeon	AMD
Ttmac	160	160	21	21	40	37	21
Umac-16	64	128	6.1	6.0	6.2	6.1	6.2
Umac-32	64	128	2.5	2.9	6.7	6.6	1.9
HMAC-Whirlpool	512	512	86	72	98	103	100
HMAC-MD4	128	512	4.7	4.7	6.4	6.4	4.7
HMAC-MD5	128	512	7.2	7.3	9.4	9.4	7.4
HMAC-RIPE-MD	160	512	23	18	27	26	21
HMAC-SHA-0	160	512	16	15	23	23	13
HMAC-SHA-1	160	512	16	15	25	24	12
HMAC-SHA-2	256 384 512	512	40 84 84	39 84 84	40 124 124	39 132 132	33 72 72
HMAC-Tiger	192	512	24	21	28	26	20
CBCMAC-Rijndael	128	128	24	26	26	27	31
CBCMAC-DES	64	56	62	61	72	69	54
CBCMAC-Shacal	512	160	31	31	67	74	29
MASH-1	1024	1024	19	19	35	33	19
MASH-2	1024	1024	20	20	37	35	20

Аналіз табл.2.2 показує, що схеми ключового хешування MASH-1, MASH-2 побудовані на основі модулярної арифметики дозволяють одержати високі показники стійкості. Таким чином перспективним напрямком подальших досліджень є розробка методів ключового хешування, дослідження особливостей їх реалізації для забезпечення цілісності та автентичності транзакцій в ІПС.

2.2. Аналіз алгоритмів ключевого хешування побудованих на модулярній арифметиці MASH-1 і MASH-2

Для забезпечення автентифікації повідомлень в ІПС саме МАС-коди підходять найкраще. Через те, що в ІПС передається велика кількість повідомлень малого розміру (1−2 Кбіт), потрібно забезпечувати автентифікацію кожного повідомлення окремо. Для цього підходять надійні ключові хеш-функції побудовані на модулярній арифметиці.

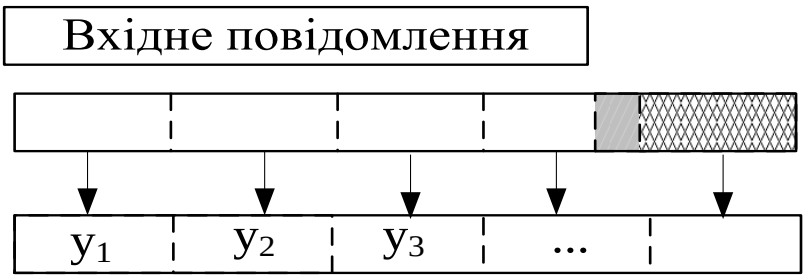
Основною ідеєю хеш-функцій заснованих на модулярной арифметиці є використання у якості циклової функції ітеративної функції, що використовує модулярну арифметику. Причинами стандартизації та застосування таких хеш-функцій є забезпечення необхідного рівня стійкості. Основним недоліком функцій є низька швидкість формування хеш-кода [42].

Хеш-Функція MASH-1 (Modular Arithmetic Secure Hash - 1) використовує RSA подібні модулі N, довжина яких забезпечує необхідну стійкість. У якості вхідної послідовності використовується двійковий рядок x довжиною $0 \le b \le 2^{n/2}$, де n розрядність хеш-кода.

Алгоритм MASH-1складається з наступних етапів [45]:

1. Системні установки та визначення констант. Установити RSA-подібний модуль N = pq довжиною m біт, де p і q випадкові прості числа. Визначити двійкову довжину n хеш-коду як найбільший добуток числа 16 і довжини хеш-коду, що задовольняє нерівності 16×n < m.

2. Попередня обробка . Доповнити, якщо необхідно, рядок x нульовими бітами, для того, щоб одержати двійковий рядок довжиною t × n/2 для найменшого t ≥ 1. Розділити доповнений текст на n/ 2-розрядні блоки x_1, ..., x_t і додати останній блок x_{t+1}, що містить n/ 2-розрядне подання числа b. Даний етап алгоритму представлений на рис. 2.2.



Риунок. 2.2 - Етап алгоритма-попередня обробув повідомлення

3. Розширення. Виконати розширення блоку x_i в n-розрядний блок y_i шляхом вставки між 4-розрядними напівбайтами блоку x_i комбінації із чотирьох одиниць (1111). Останній блок y_{t+1} формується дещо по-іншому.

4. Циклова функція. Для всіх $1 < i \le t + 1$ склеїти два n-розрядних вхідних блоки (H_{i-1} , y_i) в один n-розрядний блок згідно з формулою

$$H_i = \left(\left(\left((H_{i-1} \oplus y_i) \vee A\right)^2 \bmod N\right) \perp n\right) \oplus H_{i-1},$$

де $\vee$ – операція побітового логічного АБО;

$\oplus$ - додавання по модулі два (XOR);

$\perp$ – збереження молодших n-розрядів m-розрядного результату.

Загальний вид циклової функції представлений на рис.2.3.

5. Закінчення. У якості хеш-коду приймається n-розрядний блок H_{t+1}.

Алгоритм MASH-2 відрізняється від алгоритму MASH-1 тільки показником ступеня в цикловій функції, що має такий вигляд

$$H_i = \left(\left(\left((H_{i-1} \oplus y_i) \vee A\right)^{2^8+1} \bmod N\right) \perp n\right) \oplus H_{i-1}$$

Загальний вигляд циклової функції представлений на рис. 2.4.

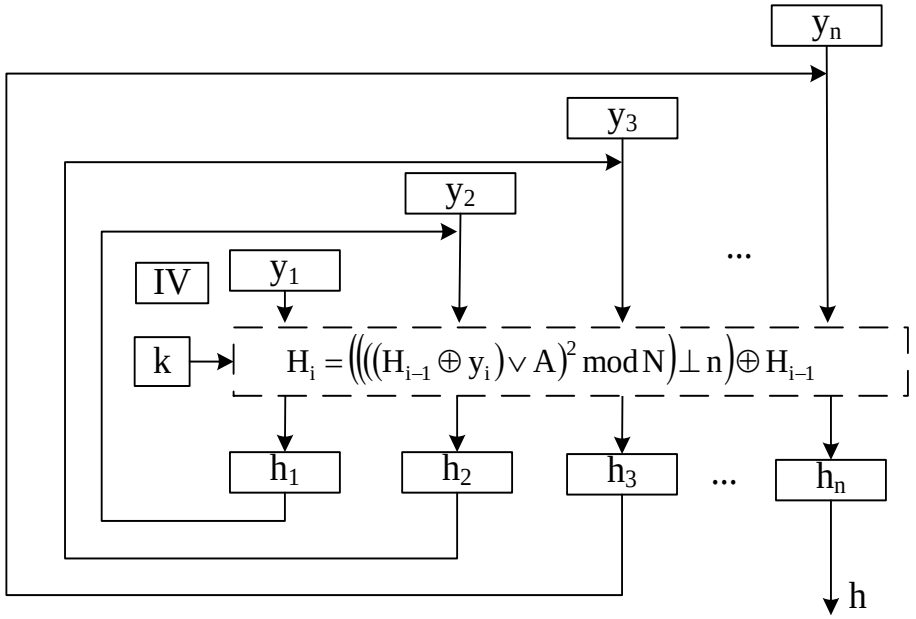


Рисунок 2.3 - Загальний вид циклової функції MASH-1

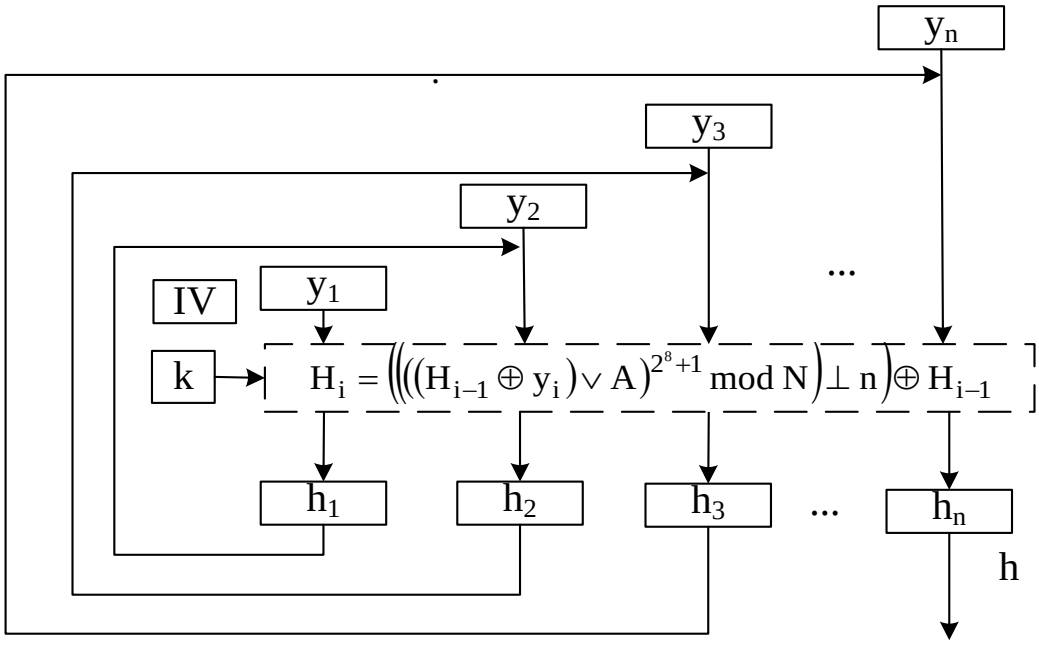


Рисунок 2.4 - Загальний вид циклової функції MASH-2

Пізніша версія стандарту MASH-2 відрізняється від MASH-1 показником ступеня, що забезпечує велику нелінійність цикловой функції, і зниження вірогідності колізій з одного боку, а з іншою викликає зниження швидкості хешування MASH-2. Таким чином актуальною задачею є визначення часових характеристик отриманих експериментально.

2.3. Оцінка часових показників ключових алгоритмів MASH-1, MASH-2

Оцінимо показники часу формування хеш-коду при невеликих розмірах вхідних повідомлень від 100 біт до 2 Кбіт. На рис. 2.5. представлені залежності швидкості формування хеш-коду від довжини ключа при розмірі повідомлення до 2 Кбіт, на рис. 2.6. залежності при використанні розміру вхідної послідовності до 40 Кбайт для алгоритму MASH-1.

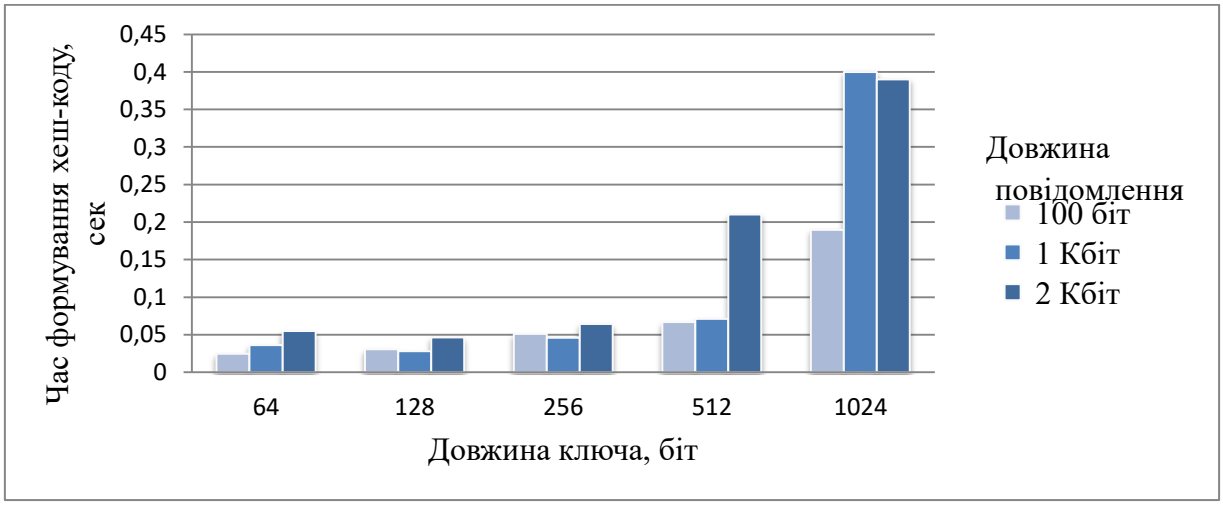


Рисунок 2.5 - Залежність швидкості формування хеш-коду від довжини ключа для алгоритму MASH-1 при довжині повідомлення від 100 біт до 2 Кбіт

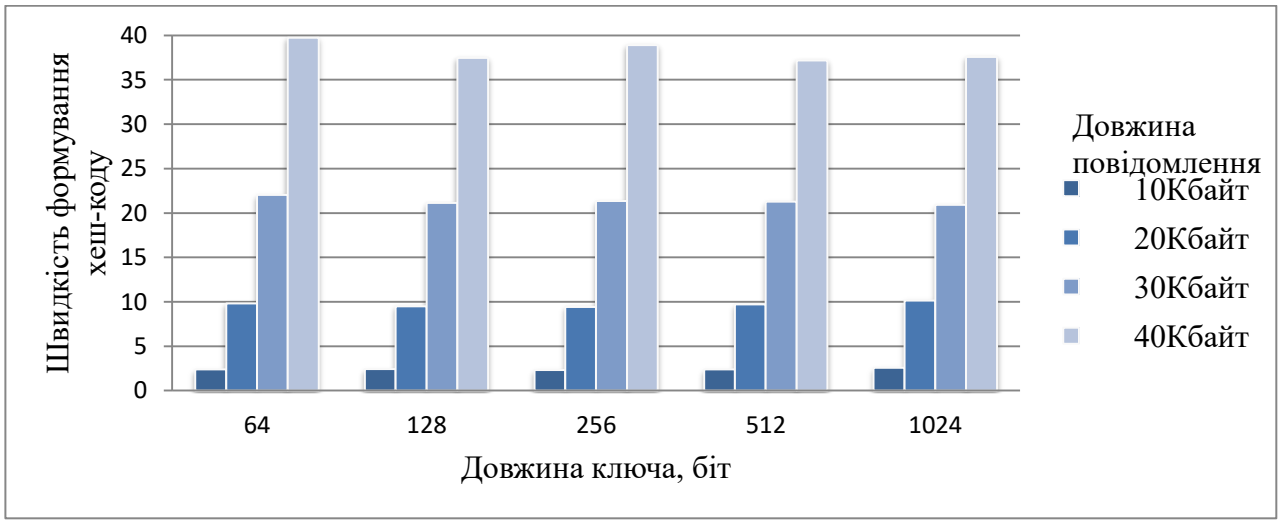


Рисунок 2.6- Залежність швидкості формування хеш-коду від довжини ключа для алгоритму MASH-1 при довжині повідомлення від 10─40 Кбайт

На рис. 2.7. приведена діаграма швидкості формування хеш-коду від довжини ключа при розмірі повідомлення до 2 Кбіт, на рис. 2.8 залежності при використанні розміру вхідної послідовності до 40 Кбайт для алгоритму MASH-2.

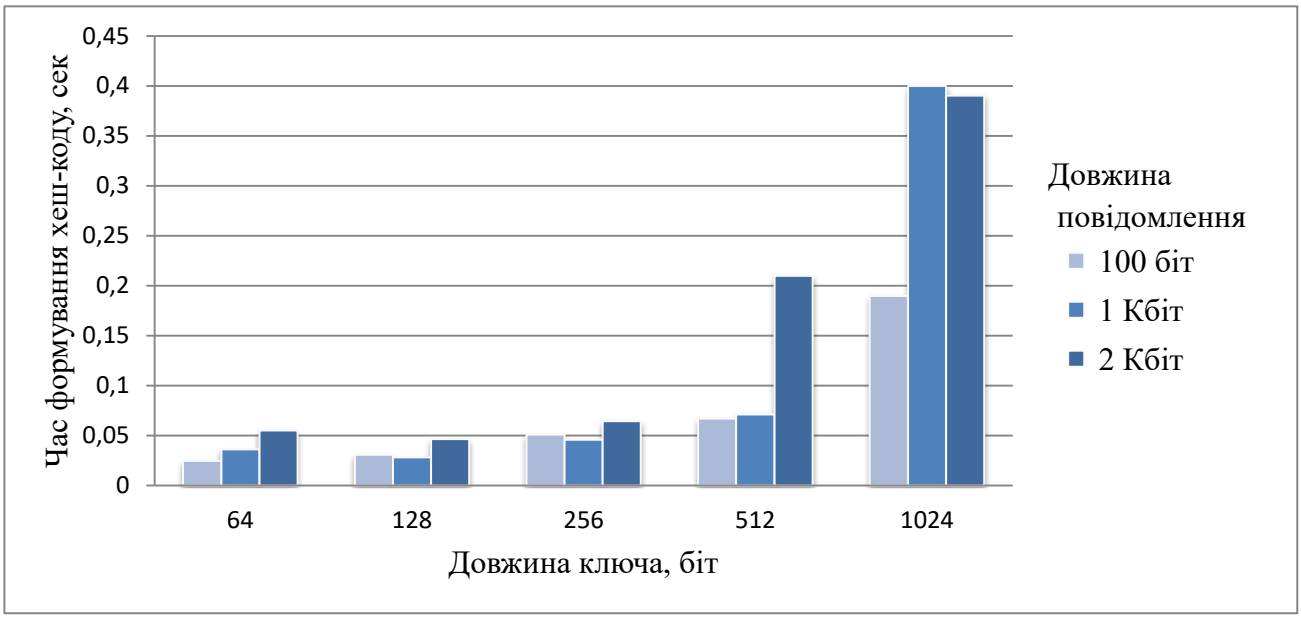


Рисунок 2.7 - Залежність швидкості формування хеш-коду від довжини ключа для алгоритму MASH-2 при довжині повідомлення від 100 біт до 2 Кбіт

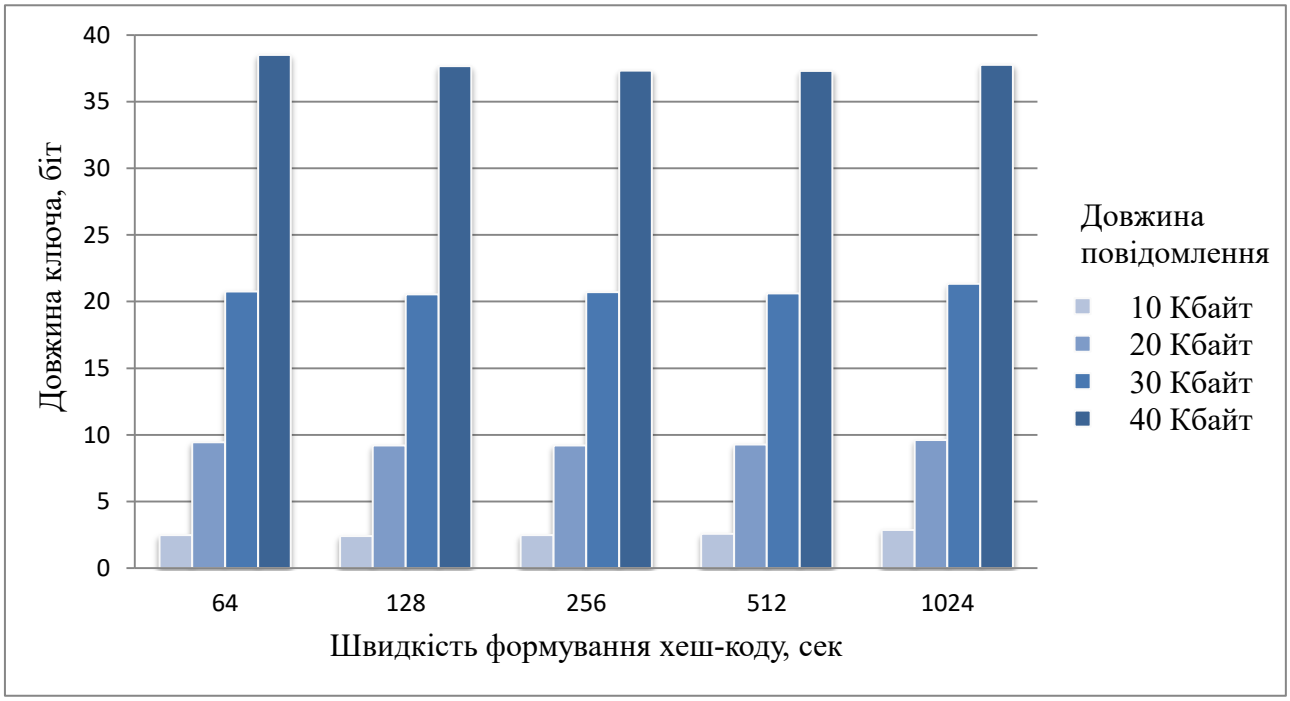


Рисунок 2.8 - Залежність швидкості формування хеш-коду від довжини ключа для алгоритму MASH-2 при довжині повідомлення від 10–40 Кбайт

Представлені на Рис.2.5−2.8 залежності показують, що при більшій довжині ключів і розмірності повідомлень, час формування хеш-коду закономірно збільшується.

Проведемо дослідження співвідношення загального часу формування хеш-коду до часу генерації простих чисел. Нехай довжина повідомлення рівна 10 Кбайт, проведемо вимір часу формування простих чисел і модулярних перетворень, результат вимірів представлений на Рисунок 2.9. Зафіксуємо довжину повідомлення в 40 Кбайт, проведемо вимір часу формування простих чисел і модулярних перетворень, результат вимірів представлений на рис. 2.10.

Аналіз співвідношення загального часу формування хеш-коду до часу генерації простих чисел показує, що процедура пошуку простих чисел необхідної довжини вимагає більших витрат часу. При невеликій довжині послідовності, яка хешується, швидкість формування хеш-коду знижує, що приводить до зниження ефективності алгоритму.

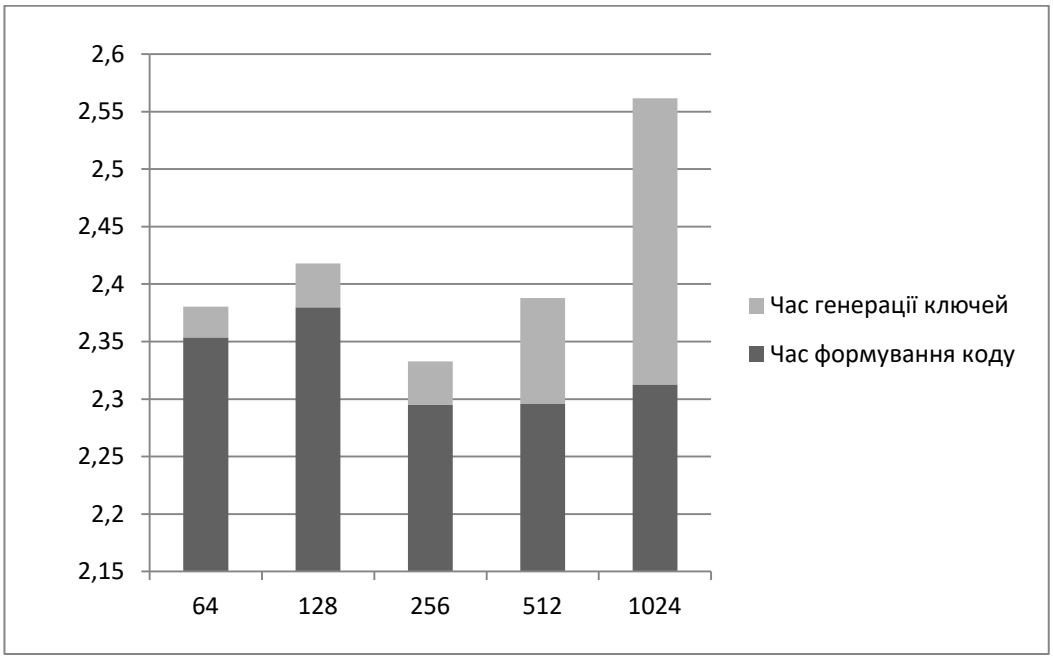


Рисунок 2.9 - Співвідношення загального часу формування хеш-коду до часу генерації простих чисел при довжині повідомлення 10 Кбайт в алгоритмі MASH-1

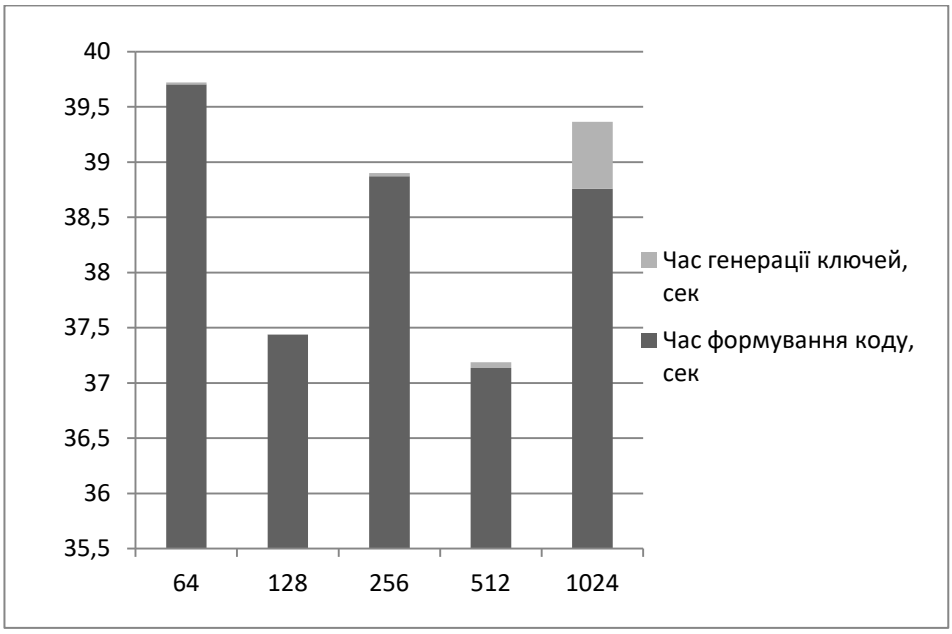


Рисунок 2.10 - Співвідношення загального часу формування хеш-коду до часу генерації простих чисел при довжині повідомлення 40Кбайт в алгоритмі MASH-1

За результатами проведеного аналізу стану інформаційної безпеки в сучасних ІПС для рішення сформованого протиріччя необхідно розробити новий метод автентифікації даних на основі ключового хешування з використанням арифметики еліптичних кривих, що задовольняє сучасним ймовірносно-тимчасовим вимогам, що пред'являються до ІПС.

2.4. Удосконаленння алгоритмів MASH-1 та MASH-2 за допомогою використання арифметики еліптичних кривих

У якості основного криптографічного примітива в несиметричних криптосистемах побудованих над групою точок еліптичної кривої використовують групові операції додавання та подвоєння точок. Розглянемо загальні положення криптографії на еліптичних кривих[8, 14, 19, 23, 24, 31, 34, 36, 42].

Еліптична крива (EC) над полем GF(q) – множина точок проективного простору (X, Y, Z), що задовольняє загальному рівнянню Вейерштрасса [69-72]:

$$Y^2Z + a_1XYZ + a_3YZ^2 = X^3 + a_2X^2Z + a_4XZ^2 + a_5Z^3,$$

де $a_i \in K, i = \overline{1, 5}$, (K - фіксоване алгебраїчне замикання GF(q)).

Точно одна точка кривої з Z = 0, так, що O(0, 1, 0) - точка на нескінченності. За допомогою заміни $x = X/Z$, $y = Y/Z$, разом із точкою O, маємо:

$$EC: y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_5.$$

Для еліптичної кривої над полем $GF(2^m)$ маємо два випадки:

суперсингулярна крива EC_1:

$$y^2 + a_3y = x^3 + a_4x + a_5,$$

несуперсингулярна крива EC_2:

$$y^2 + a_1xy = x^3 + a_2x^2 + a_5.$$

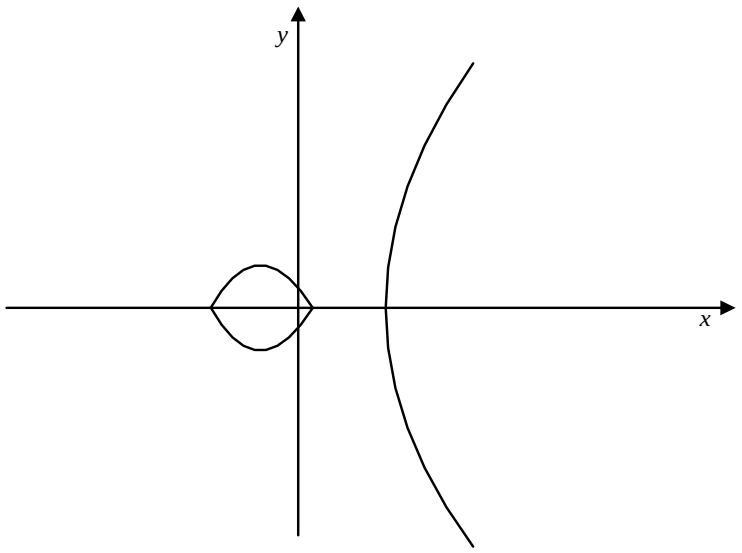


Рисунок 2.11 - Несуперсингулярна крива EC_1

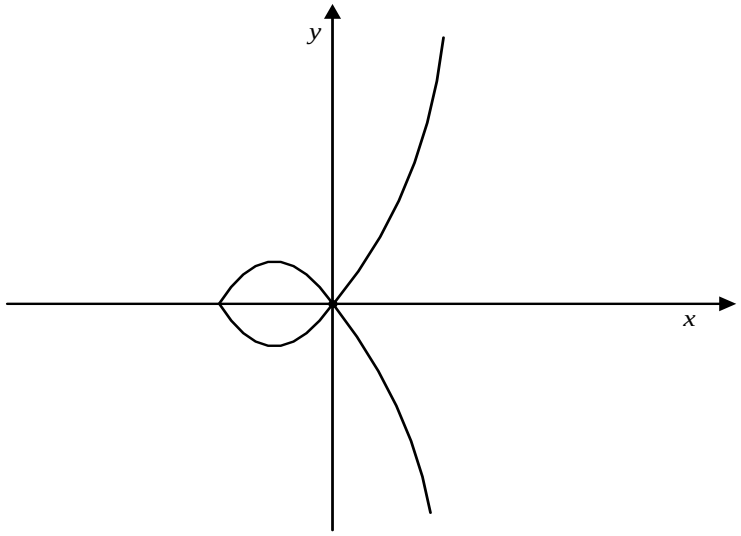


Рисунок 2.12 -Супермингулярна крива EC_2

Нехай задана супернесингулярна елептична крива над полем $GF(2^m)$ в аффінному поданні

$$y^2 + xy = x^3 + ax^2 + b(\mathrm{mod}\, f(x), 2),$$

де a і b - параметри ЕК.

Вона визначена множиною точок $(x, y) \in GF(2^m) \times GF(2^m)$, a і $b \in GF(2^m)$, $b \neq 0(\mathrm{mod}\, f(x), 2)$. Точки на ЕС, включаючи точку нескінченності O, утворять групу з операцією додавання.

Якщо точка $P_1 = (x_1, y_1)$ й $P_2 = (x_2, y_2)$ належать еліптичній кривій, тобто $P_i \in E(GF(2^m))$, то для кожної з них існує зворотна точка, відповідно $-P_1 = (x_1, x_1 + y_1)$ й $-P_2 = (x_2, x_2 + y_2)$, а також точка $P_3 = (x_3, y_3)$, така що $P_1 + P_2 = P_3$. Координати точки $P_3 = (x_3, y_3)$ визначаються з використанням співвідношень

$$x_3 = \lambda^2 + \lambda + x_1 + x_2 + a \ (\mathrm{mod}\, f(x), 2); \qquad (2.1)$$

$$y_3 = \lambda(x_1 + x_3) + x_3 + y_1 \pmod{f(x), 2}; \tag{2.2}$$

$$\lambda = \begin{cases} \dfrac{y_1 + y_2}{x_1 + x_2} \pmod{f(x), 2}, & \text{якщо} \quad P_1 \neq P_2; \\ \dfrac{x_1^2 + y_1}{x_1} \pmod{f(x), 2}, & \text{якщо} \quad P_1 = P_2, \end{cases} \tag{2.3}$$

скалярне множення визначається для декількох точок $G \in E\left(GF\left(2^m\right)\right)$ як

$$d \cdot G \pmod{f(x), 2} = \underbrace{G + G + G + \ldots + G}_{d \text{ раз}} \pmod{f(x), 2}. \tag{2.4}$$

Операція (2.4) реалізується за рахунок застосування операцій додавання $(P_1 \neq P_2)$ або подвоєння $(P_1 = P_2)$.

Точка G має порядок n на ЕК, якщо

$$n \cdot G \pmod{f(x), 2} = O; \tag{2.5}$$

де O є точка на нескінченності (нуль).

Найважливішою задачею при виконанні операцій (2.1) – (2.5) є мінімізація складності. У такий спосіб використання несуперсингулярної кривої вимагає високої обчислювальної складності та накладає додаткові обмеження.

Застосування раціональних кривих, дозволить знизити обчислювальну складність реалізації групових операцій додавання й подвоєння точок [14, 23, 24, 30].

Розглянемо плоску алгебраїчну криву, задану рівнянням f(x, y) = 0, тобто множину пар (x, y) обертаючих у нуль рівняння кривої. Якщо f(x, y) = 0 – багаточлен з коефіцієнтами з кінцевого поля GF(q), алгебраїчна крива задається множиною рішень (x, y) над GF(q). Алгебраїчна крива, задана множиною рішень рівняння f(x, y) = 0 є раціональної, якщо координати її точок можуть бути виражені через раціональні функції від одного параметра, тобто якщо існують

дві такі раціональні функції φ(t) і ψ(t), хоча б одна з яких не постійна [68] і виконується рівність:

$$f(\varphi(t), \psi(t)) = 0,$$

те φ(t) і ψ(t) тотожні відносно t. Якщо $t = t_0$ – значення параметра, відмінне від кінцевого числа значень, що обертають у нуль знаменники функцій φ(t) і ψ(t) то точка (φ(t), ψ(t)) належить кривій. Установлюване в такий спосіб відповідність між значенням параметра t і точками кривої є однозначним (якщо виключити кінцеві підмножини як зі значень параметра, так і із множини точок). При цьому параметр t може бути виражений як раціональна функція від x і y.

Таким чином, якщо відомо, що крива f(x, y) = 0 раціональна, а коефіцієнти в аналітичних вираженнях функцій φ(t) і ψ(t) належать полю раціональних чисел, то, коли t пробігає всі раціональні значення параметризація x = φ(t), y = ψ(t) дає всі точки кривої (за винятком, можливо, кінцевого їхнього числа).

Розглянемо раціональну криву f(x, y) = 0 третього ступеня. Нехай $P_1(X_1, Y_1)$ і $P_2(X_2, Y_2)$ дві прості (не кратні) точки кривої. Проведемо через них пряму (січну). Справедливо наступне твердження, що пряма, що проходить через дві прості точки раціональної кривої третього ступеня перетинає цю криву не більш ніж в одній простій точці [17].

Знаходження третьої простої точки перетинання прямої і раціональної кривої третього ступеня покладемо в основу операції додавання точок (Рисунок2.12). Відсутності третьої точки перетинання відповідає випадок проходження прямої паралельно осі OY (Рисунок2.13). Задамо, таким чином, операцію інвертування точок раціональної кривої третього ступеня.

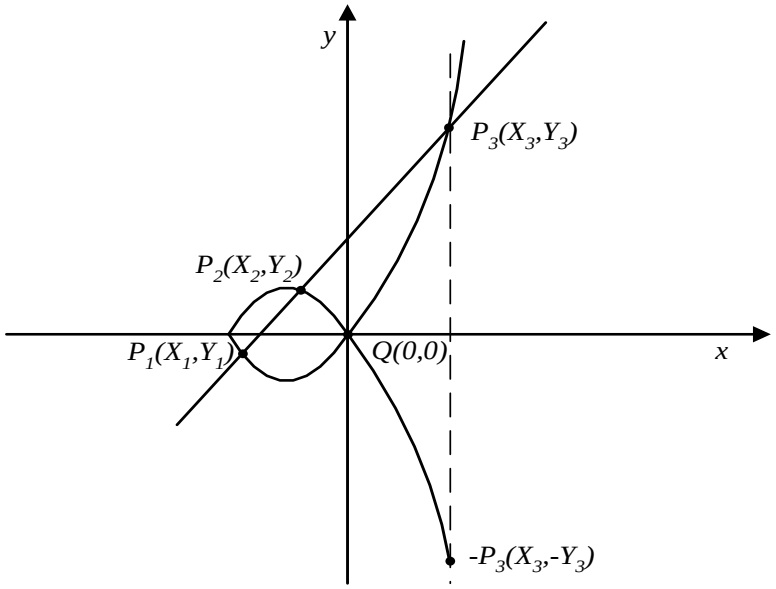


Рисунок 2.12 -Додавання точок раціональної кривої третього ступеня

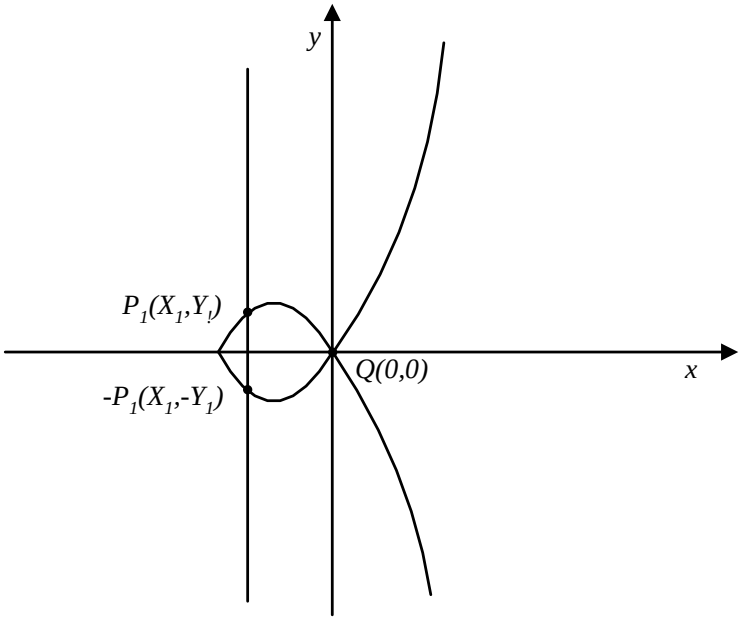


Рисунок2.13 - Інвертування (заперечення) точок раціональної кривої третього ступеня

Розглянемо подвоєння точки раціональної кривої (Рисунок2.14). Нехай, $P_1(X_1, Y_1)$, $P_2(X_2, Y_2)$ дві прості точки кривої, але при цьому $P_1(X_1, Y_1) = P_2(X_2, Y_2)$. Проведемо через точку $P_1(X_1, Y_1)$ дотичну до раціональної кривої $f(x, y) = 0$ третього ступеня. Справедливо наступне твердження.

Дотична, що проходить через просту точку на раціональній кривій третього ступеню, перетинає цю криву не більш ніж в одній простій точці.

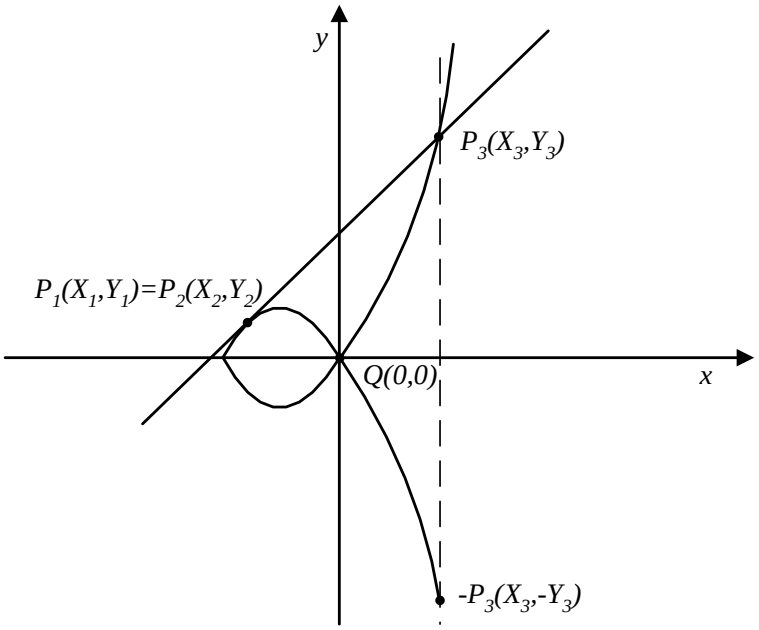


Рисунок 2.14 - Подвоєння точок раціональної кривої третього ступеня

Знаходження третьої простої точки перетинання дотичної і раціональної кривої третього ступеня покладемо в основу операції подвоєння точок. Відсутності третьої точки перетинання відповідає випадок проходження прямої паралельно осі OY. Це відповідає інвертуванню точки $P_1(X_1, Y_1)$ у точку $P_2(X_2, Y_2)$, тобто в саму себе.

Розглянемо раціональну криву

$$y^2 + xy = x^3 + x^2 \qquad (2.6)$$

над $GF(2^m)$.

Крива містить одну особливу точку Q(0, 0). Кратність особливої точки дорівнює двом. Рід кривої в цьому випадку дорівнює:

$$g = \frac{(n-1)(n-2)}{2} - \sum_{X} \frac{m_p(m_p-1)}{2} = 0,$$

тобто крива раціональна.

Розглянемо рівняння прямої (січної), що проходить через довільні дві прості точки кривої $P_1(X_1, Y_1)$ і $P_2(X_2, Y_2)$:

$$y = ax + b,$$

$$\text{де } a = \frac{Y_2 + Y_1}{X_2 + X_1},\ b = aX_1 + Y_1.$$

Підставивши рівняння прямої (січної), що проходить через довільні дві прості точки кривої $P_1(X_1, Y_1)$ і $P_2(X_2, Y_2)$ у вираження (2.6), одержимо

$$(ax + b)^2 + (ax + b)x = x^3 + a_2x^2.$$

Розкриємо дужки та приведемо подібні, одержимо

$$x^3 + (a_2 - a^2 - a)x^2 - bx - b^2 = 0,$$

отже,

$$X_1 + X_2 + X_3 = a^2 + a - a_2.$$

Останнє вираження дозволяє в явному виді записати координати третьої точки перетинання січної і кривої (3.6):

$$X_3 = a^2 + a - a_2 - X_1 - X_2,$$
$$Y_3 = a_3 + b.$$

Пряма, що проходить через особливу точку на раціональній кривій третього ступеню, перетинає цю криву ще не більш ніж в одній простій точці (Рисунок2.15).

Відсутності третьої точки перетинання відповідає випадок проходження прямої паралельно осі OY. При цьому значенні кутового коефіцієнта t пряма перетинається із кривої в нескінченно вилученій точці.

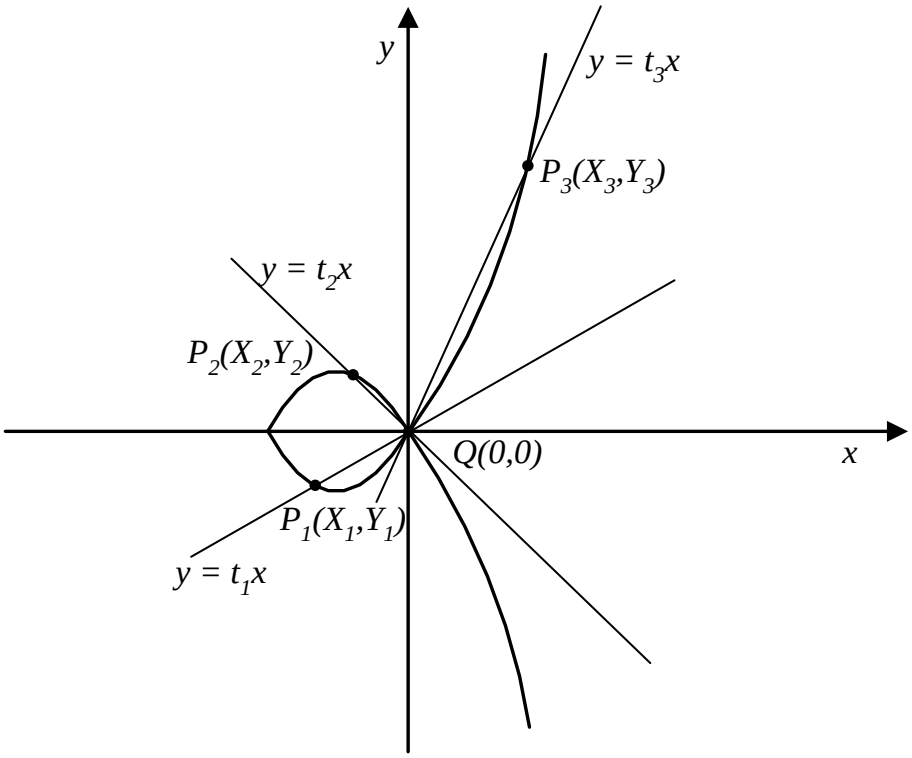


Рисунок 2.15 -Параметризація раціональної кривої третього ступеня

Виразимо координати точок раціональної кривої через раціональні функції від параметра t, тобто через дві такі раціональні функції φ(t) і ψ(t), одна з яких не постійна, при $f(\varphi(t),\psi(t))=0$ тотожно відносно t. Коли t пробігає всі значення GF(q) параметризація x = φ(t), y = ψ(t) дає всі точки кривої (за винятком, можливо, кінцевої їх кількості).

Розглянемо плоску раціональну криву f(x, y) = 0, задану множиною рішень рівняння (2.6) над $GF(2^m)$. Проведемо через особливу точку Q(0, 0) січним, заданим рівнянням y = tx, де t - кутовий коефіцієнт січної.

По визначенню січна перетне криву (2.6) ще в одній точці. Підставивши рівняння січної в (2.6) одержимо

$$t^2x^2 + tx^2 = x^3 + x^2.$$

Звідки одержимо координати шуканої точки перетинання січної із кривої:

$$X = t^2 + t + 1,$$
$$Y = t = t^3 + t^2 + t.$$

Таким чином, коли t пробігає всі значення $GF(2^m)$ параметризація $x = \varphi(t)$, $y = \psi(t)$ дає всі точки кривій (за винятком, можливо, кінцевого їхнього числа).

Групова операція додавання точок раціональної кривої третього ступеня, заданої рівнянням (3.6), у параметричному виді тотожна вираженню:

$$t_3 = \frac{t_1 t_2}{t_1 + t_2 + 1},$$

де t_1, t_2 і t_3 – локальні параметри точок $P_1(X_1, Y_1)$, $P_2(X_2, Y_2)$ і $P_3(X_3, Y_3)$ відповідно.

Розглянемо групову операцію подвоєння точок раціональної кривої. Групова операція подвоєння точок раціональної кривої третього ступеня, заданої рівнянням (2.6), у параметричному виді тотожна вираженню:

$$t_3 = t_1^2,$$

де $t_1 = t_2$ і t_3 – локальні параметри точок $P_1(X_1, Y_1) = P_2(X_2, Y_2)$ і $P_3(X_3, Y_3)$ відповідно.

Розглянуті перетворення в групі точок еліптичних кривих широко використовуються при побудові криптосистем з відкритим ключем

[17,21]. У їхній основі лежить теоретична складність завдання дискретного логарифмування.

Скористаємося поняттям дискретного логарифма, уведеного в [17]. Нехай H – кінцева група, g і y – елементи цієї групи. Будь-яке ціле x, таке, що $g^x = y$ називається дискретним логарифмом y за основою g. Кожний елемент y (H має дискретний логарифм за основою g тоді і тільки тоді, коли H є циклічною групою з утворюючої g. У загальному випадку відомі алгоритми для обчислення дискретних логарифмів у групах порядку m мають приблизно однакову складність відносно m як для алгоритмів факторизації m [17, 21].

Стосовно до групи точок еліптичної кривої використовують наступне поняття дискретного логарифма на кривій [21]. Нехай H_{EC} – кінцева група точок еліптичній кривій, P_i і P_j – елементи цієї групи. Будь-яке ціле x, таке, що $xP_i = P_j$ називається дискретним логарифмом на еліптичній кривій. Криптостійкість алгоритмів, побудованих на еліптичних кривих заснована на трудності узяття дискретного логарифма та складається у визначенні x по відомим P_i і P_j.

Таким чином реалізація криптосистеми на раціональних кривих третього ступеня дозволить одержати найбільший показник стійкості криптосистеми при фіксованій обчислювальній складності .

2.5. Удосконалення методу ключового хешування на основі застосування перетворень у групі точок еліптичної кривої

Алгоритм ефективного способу швидкого криптографічного перетворення інформації, в основі якого лежить теоретико-складна проблема узяття дискретного логарифма в групі точок еліптичної кривої реалізується з використанням раціональної кривої. При цьому текст m_i, де i = 0; n, подається координатою *X* точки еліптичної кривої EC, після обчислюється координата Y. Отримаємо точку кривої $P_1(x_1, y_1)$. Точка кривої умножається на таємний скаляр k (особистий ключ користувача), отримаємо точку P_k. Цикловою функцією алгоритму хешування є скалярне добуток:

$$P_k = k * P_1 .$$

Наступна точка, якою подається наступний блок вихідного тексту, умножається на результат модифікації скаляра *k.*

$$k_i = k_{i-1} \oplus X[P_{i-1}] .$$

Результатом хешування є координати точки кривої, яка отримується на останньому раунді.

$$h(M) = X[P_n] \oplus Y[P_n]$$

Загальний вигляд хеш-функції:

$$h_i = X[k_i * P_i] = X[(k_{i-1} \oplus X[P_{i-1}]) * P_i] \oplus Y[(k_{i-1} \oplus X[P_{i-1}]) * P_i],$$

де $P_i = \varphi_{EC}(M_i)$ - результат відтворення з блоку вихідного повідомлення M_i точки кривої.

Алгоритм ключового хешування наведений на Рисунок2.2. На першому кроці необхідно встановити ключові параметри хеш-функції. Після розбиття повідомлення на блоки (крок 3) потрібно доповнити блок відкритого тексту M_b Наступним кроком є присвоєння раундовому ключу значення ключової послідовності. Після присвоєння значення і-го блоку координаті *X* еквівалентної точки кривої (крок 5) знаходиться координата Y (крок 6) та створюється точка (крок 7). Потім здійснюється скалярний добуток (крок 8) та встановлюється ключ наступного раунду (крок 9). Таким чином, алгоритм з 5 по 8 крок виконується n разів, стільки скільки блоків відкритого тексту. Останнім кроком виконується додавання за модулем 2 координат X та Y. На виході хеш-функції

отримаємо послідовність довжиною l_k. Останній крок дозволяє позбавитись випадку коли функція може приймати нульове значення, якщо використовувати в якості хеш-кода значення однієї з координат.

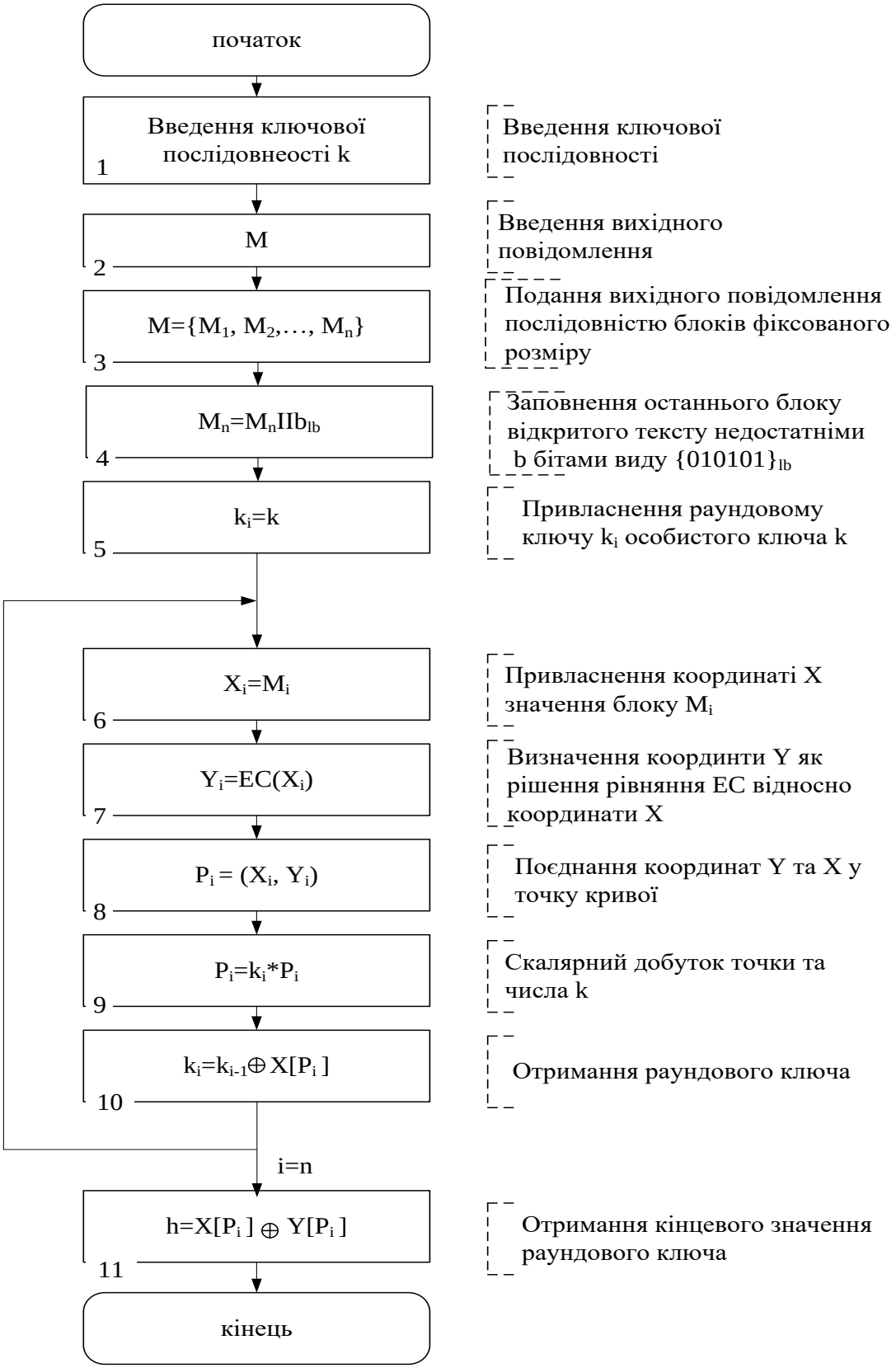
Для зниження обчислювальної складності основного кроку може бути використанний інший спосіб ключового хешування.

Аналогічним чином представляється текст точками кривої $m_i \to X_i$, по координаті X визначається точка кривої, потім точки кривої складаються та множаться на секретний скаляр, а результат хешування представляється координатою точки кривої:

$$h = X[k * (P_1 + P_2 + ... + P_n)]$$

Алгоритм ключового хешування з використанням другого методу представлений на Рисунок2.17.

Аналіз алгоритмів показує, що при використанні алгоритму формування хеш-коду за допомогою першого методу (множення кожного блоку повідомлення, що перетворений в точку еліптичної кривої, на скаляр (секретний ключ)) значно знижує його швидкість, що не дозволяє його використовувати при передачі повідомлень вільної довжини, але стійкість хеш-коду значно вища, тому що основується на обчислювально-складної задачі знаходження дискретного алгоритму в групі точок еліптичної кривої.



Рисуно к2.16 - Алгоритм ключового хешування

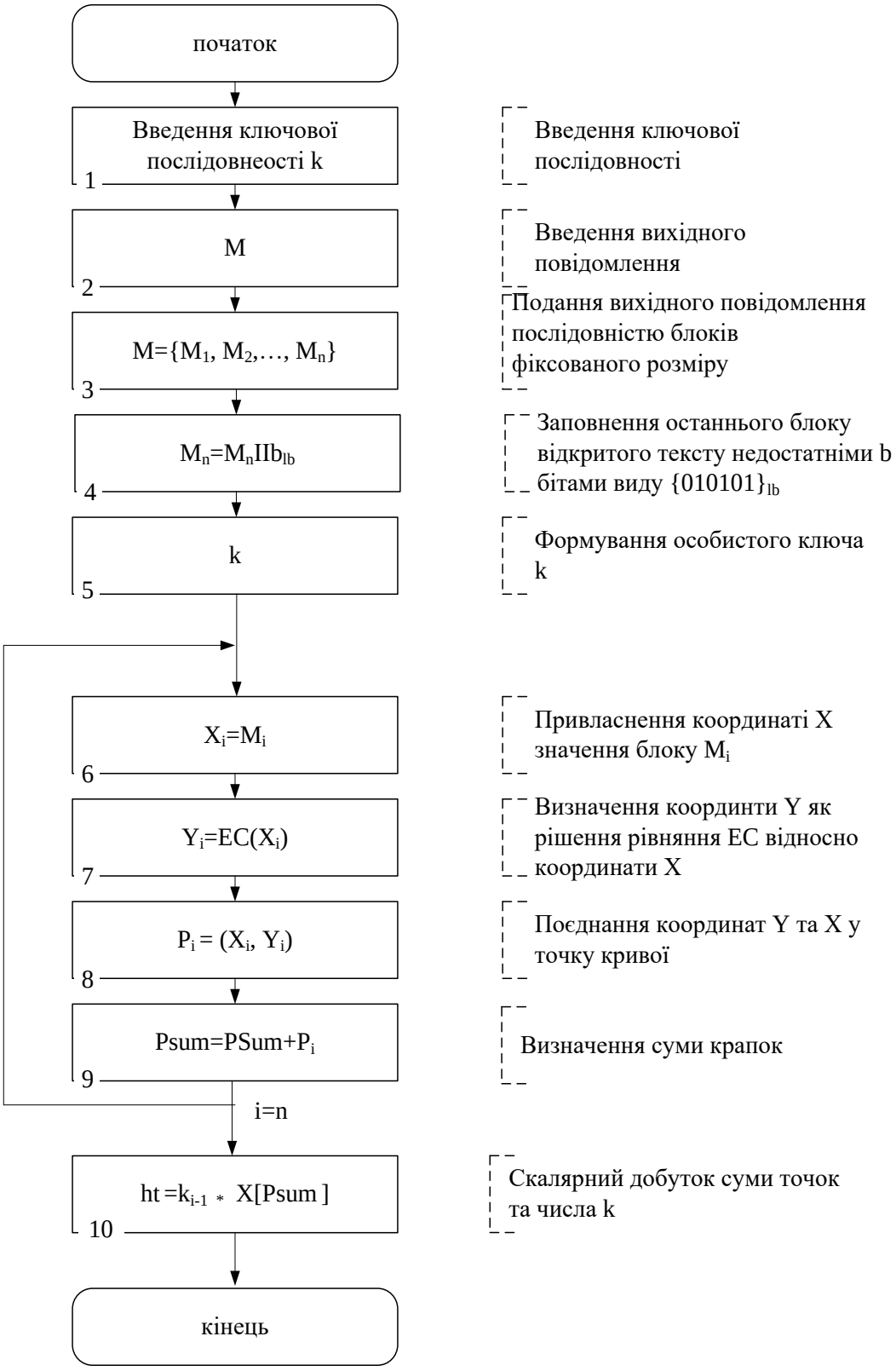


Рисунок 2.17 - Алгоритм ключового хешування

При використанні другого алгоритму (множення суми точок еліптичної кривої, які є представленням блоків повідомлення, на скаляр (секретний ключ)) який дозволяє формувати хеш-код вільної довжини за час, який задовольняє вимогам до сучасних ЦП (час формування ЦП на еліптичних кривих(ДСТУ-4145)).

Таким чином перший алгоритм пропонується використовувати при передачі транзакцій фіксованої довжини до 1,5–2 Кбіта, другий алгоритм пропонується використовувати при передачі повідомлення вільної довжини.

2.6 Висновки до другого розділу

1. Для забезпечення процедури хешування в цифрових підписах використовуються безключові хеш-функції. Однак, проведень аналіз показавши, що у випадку використання MDC-кодів по даному вхідному повідомленню хеш-код може обчислити будь-який суб'єкт, а при використанні MAC-коду обчислити хеш-код по даному вхідному повідомленню може тільки суб'єкт, що володіє секретним ключем.

2. Проведений аналіз стійкості алгоритмів ключового хешування дозволив визначити, що найбільш ефективними алгоритмами для забезпечення цілісності та автентичності транзакцій краще використовувати ключові функції хешування, наприклад алгоритми переможці міжнародного криптографічного конкурсу MASH-1 і MASH-2.

4. З використанням основних положень теорії еліптичних кривих теоретично обґрунтована можливість побудови ключових хеш-функцій з використанням арифметики в групі точок еліптичної кривої. Запропоновані два методи удосконалення алгоритмів ключового хешування.

РОЗДІЛ 3 РОЗРОБКА ПРАКТИЧНИХ РЕКОМЕНДАЦІЙ З ВИКОРИСТАННЯ АЛГОРИТМУ КЛЮЧОВОГО ХЕШУВАННЯПОБУДОВАНОГО НА АРИФМЕТИЦІ ЕЛІПТИЧНИХ КРИВИХ

3.1 Розробка програмного пакета, що реалізує досліджуваний алгоритм ключового хешування на основі модулярної арифметики (MASH-1, MASH-2)

Для реалізації криптографічної обробки інформації в ІПС розроблений програмний пакет, що реалізує алгоритм ключового хешування побудований на основі модулярної арифметиці. Він реалізований мовою високого рівня Java у середовищі NetBeans IDE 6.1 і призначений для демонстрації процесу хешування інформації в дослідницьких цілях.

На Рисунок3.1 − 3.4 представлений інтерфейс програмного модуля, що реалізує алгоритм ключового хешування з використанням модулярної арифметики.

У верхньому полі “Вихідне повідомлення” уводиться із клавіатури або завантажується з файлу значення повідомлення. Вибір файлу здійснюється за допомогою вікна відкриття файлу, що викликається кнопкою “Відкрити”. Вікно вибору файлу представлене на Рисунок3.1. Результат завантаження повідомлення з файла наведений на Рисунок3.2. Повідомлення відображається в текстовому полі, його перегляд здійснюється за допомогою смуги прокрутки.

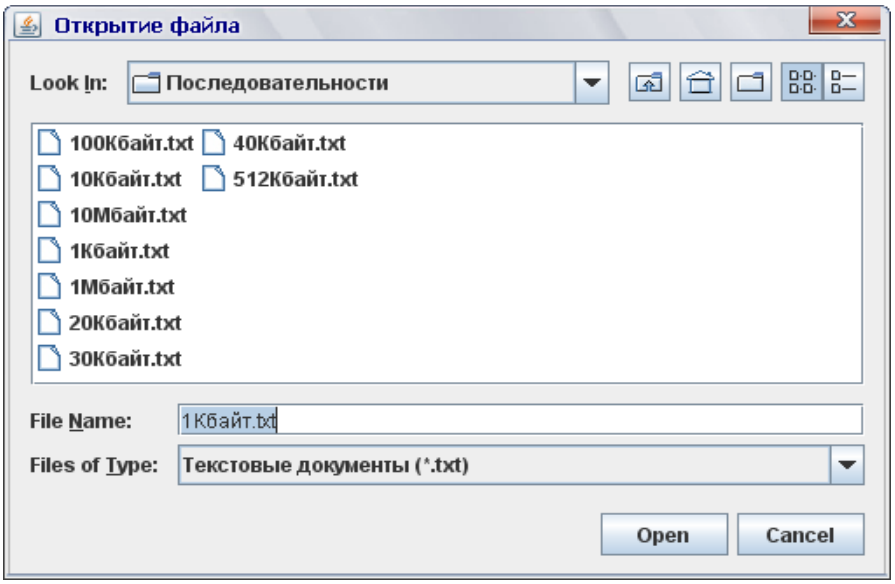


Рисунок 3.1 - Інтерфейс програмного пакета ─ відкриття файлу

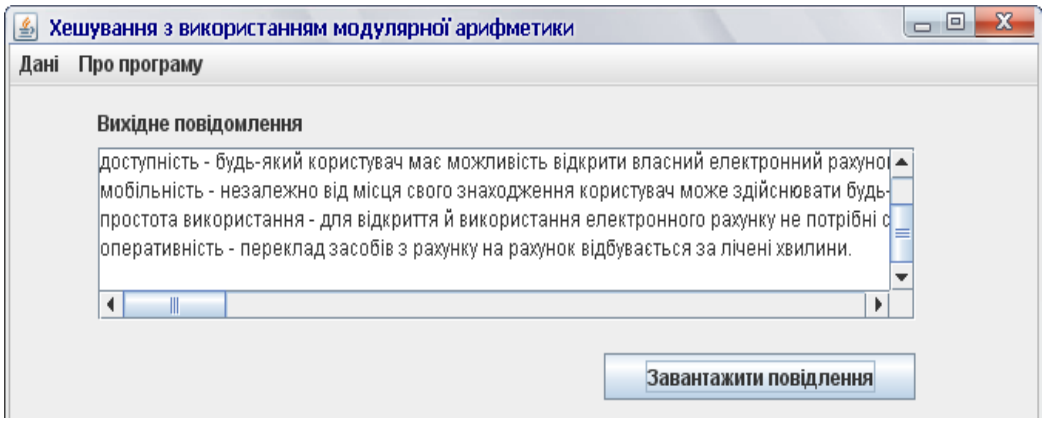


Рисунок 3.2 - Інтерфейс програмного пакета ─ уведення повідомлення

На Рисунок3.3 представлений етап роботи пакета, що складається у формуванні простих чисел. У текстове поле “Уведіть довжину модуля в бітах” уводиться довжина модуля, що формується як добуток двох простих чисел p і q. Після натискання кнопки “Згенерувати” у текстових полях “p=” і “q=” будуть відображені значення простих чисел у десятковій системі рахування, які далі будуть використані при формуванні модуля.

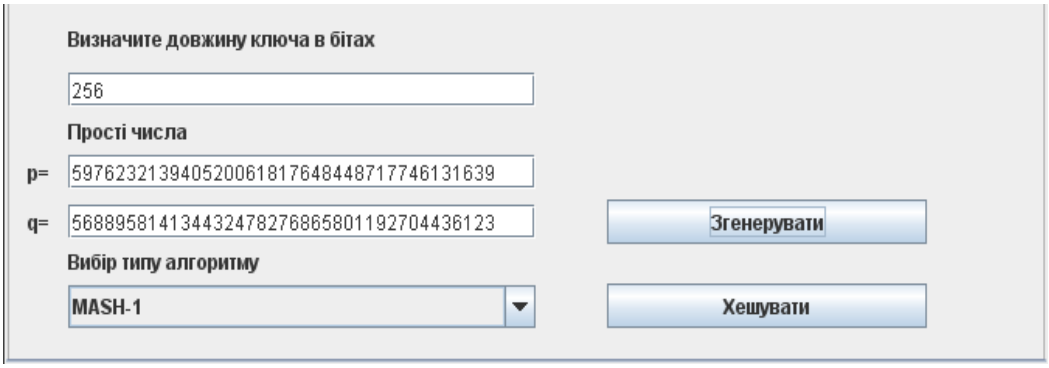


Рисунок 3.3 - Інтерфейс програмного пакета ─ генерація простих чисел

На рис 3.4 представлений етап формування хеш-коду. Спочатку завантажується повідомлення, далі генеруються ключі вказаної довжини. Процес хешування повідомлення ініціюється натисканням кнопки “Хешувати”. Після натискання на кнопку “Хешувати” у текстовому полі “хеш-код повідомлення” виводитися результат хешування в шестнадцтирічній системі рахування, який можно зберегти у текстовий файл за допомогою кнопки “Зберегти хеш-код у файл”. Лістинг вихідного коду програмного продукту наведений у додатку А.

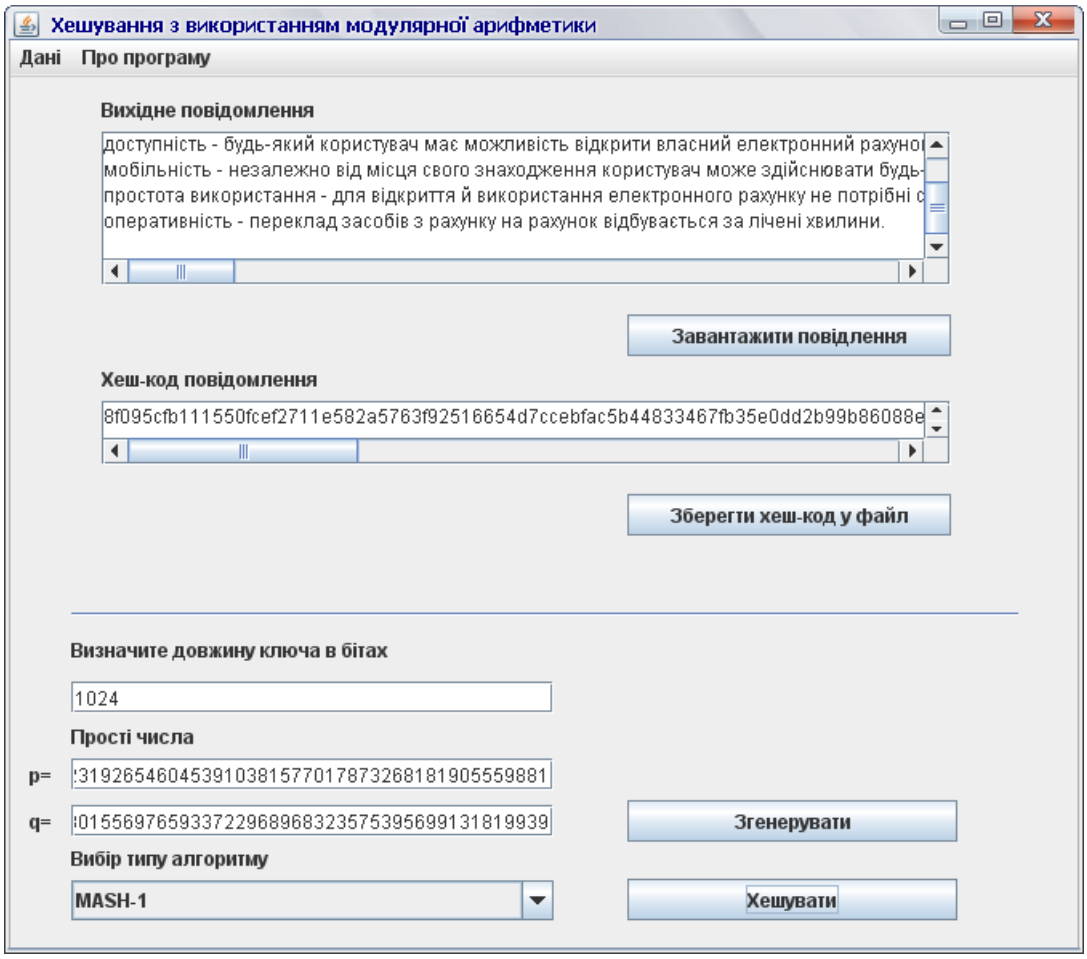


Рисунок 3.4 - Інтерфейс програмного пакета ─ формування хеш-коду

3.2 Розробка пропозицій по апаратній реалізації пристроїв ключового хешуванняз використанням арифметики еліптичних кривих

Для забезпечення необхідної цілісності та автентичності передачі транзакцій в ІПС із використанням алгоритмів ключового хешування побудованого на арифметиці еліптичних кривих розроблені апаратні засоби формування хеш-кодів. Структурна схема відповідного пристрою, заснованого на використанні арифметики еліптичних кривих, представлена на Рисунок3.5.

Пристрій функціонує в такий спосіб. Обробка інформації в схемі визначається наступними операціями. Ключові дані (КД) надходять на вхід пристрою уведення ключових даних (ПУКД). Уведені КД перетворяться в коефіцієнти a_1, …, a_6 однорідного багаточлена, що задає вид еліптичної кривої над GF(q),

$\forall a_i \in GF(q)$, який генерується за допомогою пристрою генерації виду еліптичної кривої.

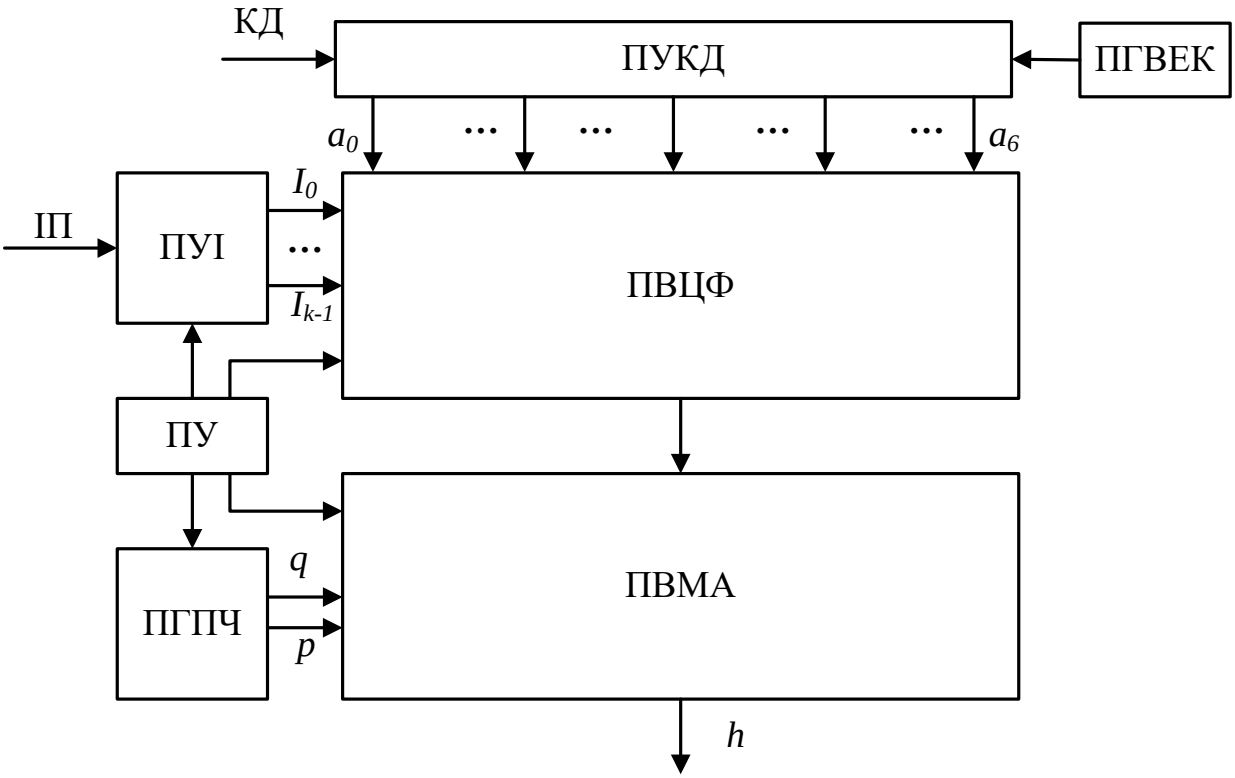


Рисунок 3.5 - Структурна схема пристрою формування хеш-коду із із використанням арифметики еліптичної кривої

Формування хеш-коду здійснюється в такий спосіб. Інформаційна послідовність (ІП) надходить на вхід пристрою уведення інформації (ПУІ) де розбивається на блоки по *k* символів з GF(q). Інформаційні блоки I_0, …, I_{k-1} надходять на вхід пристрою виконання циклів хеш-функції (ПВЦФ) $P_k = k * P_i$. З ПВЦФ інформаційна послідовність у стислому виді надходить на пристрій виконання модулярної арифметики (ПВМА). На цей блок також надходять значення двох простих чисел, які генеруються у пристрої генерації простих чисел (ПГПЧ).

3.3 Розробка програмного пакета, що реалізує запропоновані способи ключового хешування з використанням арифметики еліптичних кривих

Для реалізації криптографічної обробки інформації в ІПС розроблений програмний пакет, що реалізує алгоритм ключового хешування з використанням арифметики еліптичної кривої. Він реалізований мовою високого рівня Java у середовищі NetBeans IDE 6.1 і призначений для демонстрації процесу хешування інформації в дослідницьких цілях.

На Рисунок2.3 – 2.6 представлений інтерфейс програмного модуля, що реалізує процедуру хешування з використанням арифметики еліптичної кривої.

У текстовому полі “Вихідне повідомлення” уводиться із клавіатури (Рисунок3.6) або завантажується з файлу значення повідомлення. Вибір файлу здійснюється за допомогою вікна відкриття файлу, що викликається кнопкою “Відкрити”. Вікно вибору файлу представлене на Рисунок3.7.

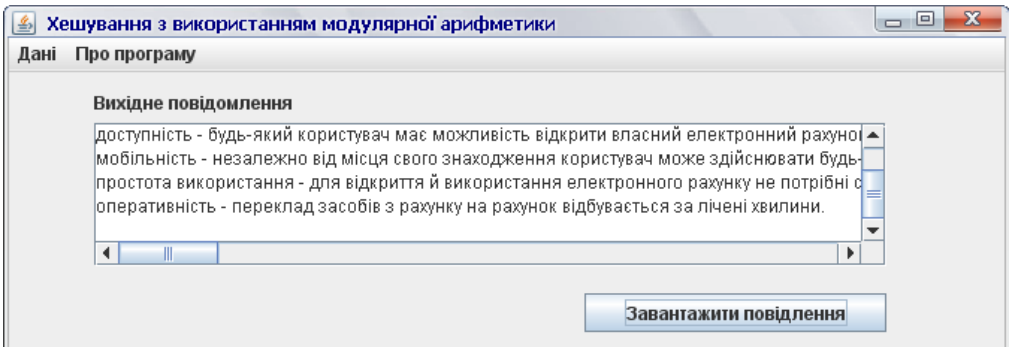


Рисунок 3.6 - Інтерфейс програмного пакета - уведення повідомлення

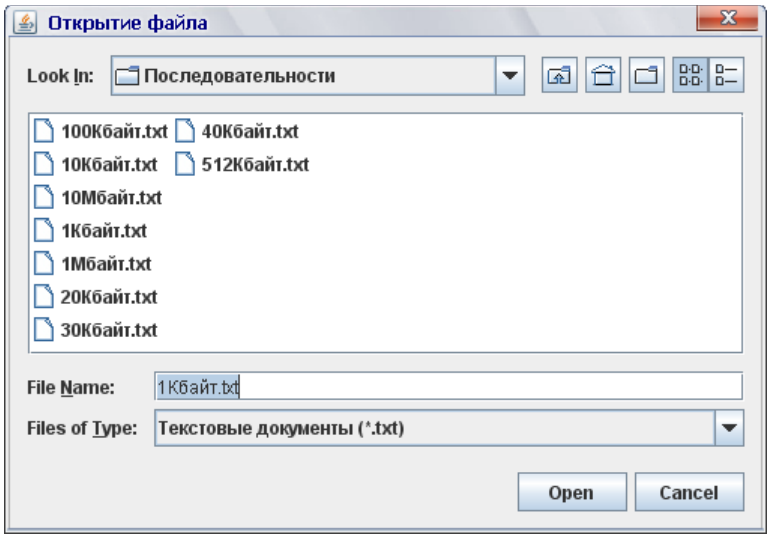


Рисунок 3.7- Інтерфейс програмного пакета - відкриття файлу

На Рисунок3.8 представлений етап роботи пакета, який складається у формуванні простих чисел. У вікно уведення “Уведіть довжину модуля в бітах” уводиться довжина модуля, що формується як добуток двох простих чисел p і q. Після натискання кнопки “Згенерувати” у текстових полях “p=” і “q=” будуть відображені значення простих чисел у десятковій системі вирахування, які далі будуть використані при формуванні модуля. Їх значення в процесі хешування інших повідомлень можуть бути незмінними.

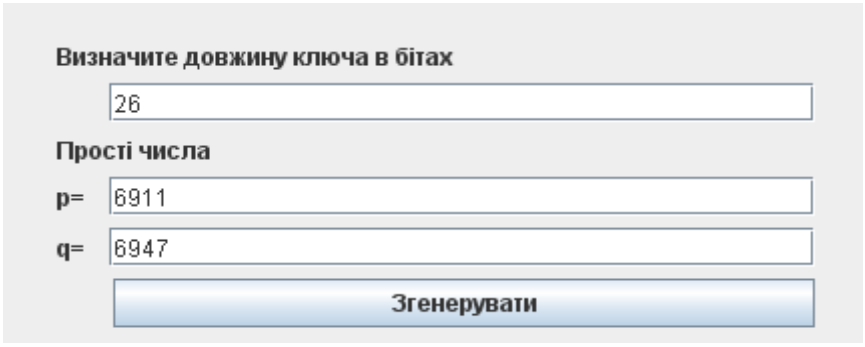


Рисунок 3.8- Інтерфейс програмного пакета - генерація простих чисел

На Рис. 3.9. представлений етап роботи пакета визначення виду еліптичної кривої та її параметрів, що складається в завданні, розширеного поля . Вид кривої зафіксований та представлений у полі “Вид кривої”. Інтерфейс додатку дозволяє визначити поле, в якому будуть розташовані точки кривої. Для цього необхідно в списку, що випадає, “Потужність поля GF(2^m)=” вибрати значення m і таким чином визначити розширене поле для еліптичної кривої.

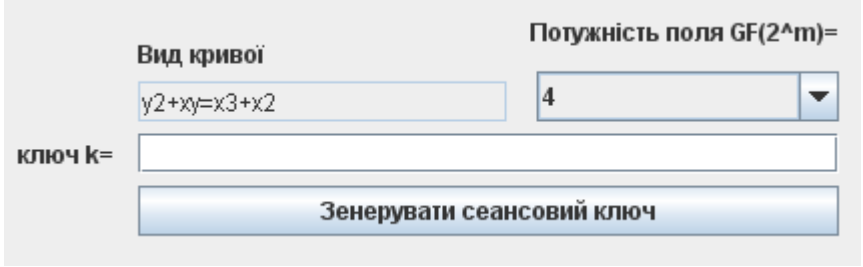


Рисунок 3.9 - Інтерфейс програмного пакета – визначення параметрів

еліптичної кривої

Генерація сеансового ключа реализуеться з використанням кнопки “Згенерувати сеансовий ключ” , після натискання якої значення ключа відобразиться в полі “ключ к=”.

Процес хешування повідомлення ініціюється натисканням кнопки “Хешувати”.

Після натискання на кнопку “Хешувати” у текстовому полі “хеш-код повідомлення” виводитися результат хешування в шестнадцатирічній системі рахування.

Лістинг вихідного коду програмного засобу наведений у додатку Б.

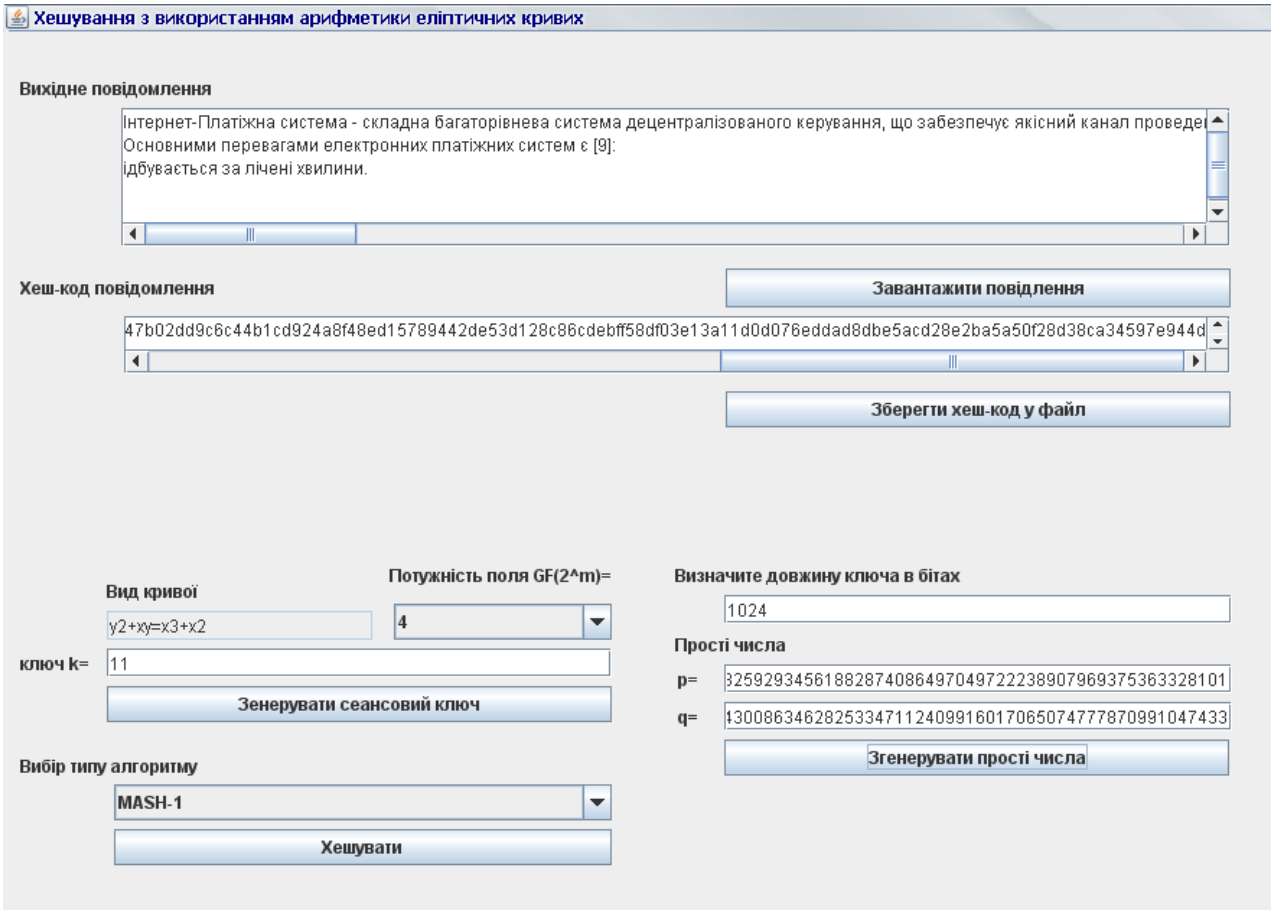


Рисунок 3.10 - Інтерфейс програмного пакета – формування хеш-коду

3.4. Оцінка обчислювальної складності розробленого способу ключового хешування

З використанням розроблених схем ключового хешування з використанням арифметики еліптичних кривих були отримані алгоритми формування хеш-кодів і програмно реалізований розроблений спосіб ключового хешування, що відрізняються від існуючих застосуванням арифметики еліптичних кривих для формування ключових хеш-функцій.

Для оцінки обчислювальної складності будемо використовувати 32-розрядну платформу, із процесором Celeron з тактовою частотою - 1 Ггц. Розрахуємо обчислювальну складність одного раунду роботи циклової функції для розробленого алгоритму. Припустимо, що система виконує одну операцію додавання або зрушення 32-розрядних поліномів за один такт.

Зафіксуємо криву $y^2 + xy = x^3 + x^2 + 1$. Складність операцій додавання, подвоєння та скалярного множення для обраної кривої визначається наступними виразами.

$$I_{слож} = I_{inv} + 3I_{mul} + 9I_{add} = (O\log(m) + 3)I_{mul} + 9I_{add},$$

$$I_{удв} = I_{inv} + 3I_{mul} + 7I_{add} = (O\log(m) + 3)I_{mul} + 7I_{add},$$

$$I_{скалТ} = m\left[0.5I_{слож} + I_{удв}\right] = m\left[(O\log(m) + 3)I_{mull} + 16I_{add}\right].$$

Обчислювальна складність операцій додавання, множення та інверсії поліномів становить:

$$I_{x\,mod} = \frac{m}{w} \cdot I^{w}_{shift} + 0{,}5 \cdot \frac{m}{w} \cdot I^{w}_{add},$$

$$I_{add} = \frac{m}{w} \cdot I^{w}_{add},$$

$$I_{mull} = 2 \cdot \frac{m}{w} \cdot m \cdot (I_{shift}^{w} + 0{,}5 \cdot I_{add}^{w}),$$

$$I_{inv} = O\log(m) I_{mull}.$$

Приклад 4. Зафіксуємо криву $y^2 + xy = x^3 + x^2 + 1$. Вираження для оцінки обчислювальної складності мають вигляд:

$$I_{слож} = I_{inv} + 3I_{mul} + 9I_{add} = (O\log(m) + 3) I_{mull} + 9I_{add},$$

$$I_{удв} = I_{inv} + 3I_{mul} + 7I_{add} = (O\log(m) + 3) I_{mull} + 7I_{add},$$

$$I_{скалТ} = m\left[0{,}5I_{слож} + I_{удв}\right] = m\left[(O\log(m) + 3) I_{mull} + 16I_{add}\right].$$

З урахуванням отриманих виразів оцінимо швидкодію розробленої схеми формування хеш-кодів (Рисунок2.3).

Аналіз Рисунок3.11 показує, що для довжин ключової послідовності (рекомендуються: 64, 96, 128, 160 біт) запропонований спосіб ключового хешування дозволяє одержати наступні значення швидкості хешування: 500, 100, 80, 50 Кбіт/с. Використання для хешування перетворень у групі точок еліптичної кривої дозволяє одержати високу швидкість хешування в порівнянні зі стандартами MASH-1, MASH-2 та криптографічна стійкість запропонованої схеми ключового хешування істотно перевершує їх стійкість.

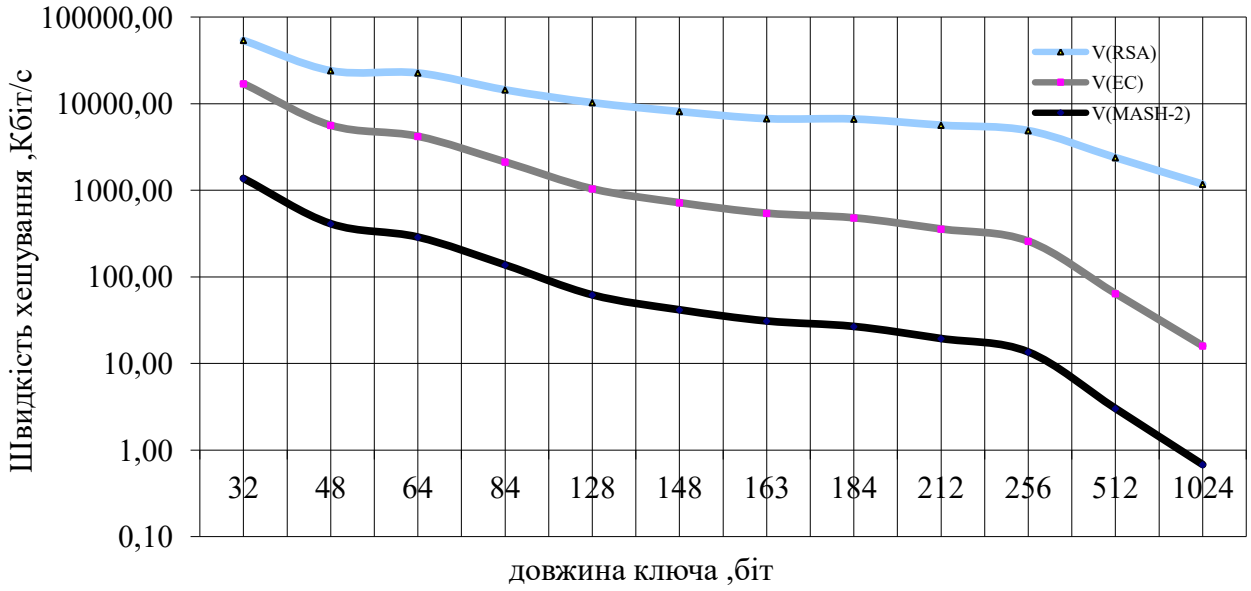


Рисунок 3.11 -Швидкість формування хеш-кодів RSA, MASH-2, EC

Таким чином, у результаті досліджень був отриманий новий спосіб ключового хешування на базі арифметики в групі точок еліптичної кривої, що дозволяє одержати новий підхід до побудови ключових хеш-функцій з використанням арифметики еліптичних кривих. Розроблений спосіб дозволяє при однократному використанні ключа для кожного повідомлення забезпечити безумовну стійкість схем автентифікації, а при многократному доказову стійкість, що є безсумнівним достоїнством у порівнянні з існуючими методами та способами ключового хешування. У випадку багаторазової схеми ключового хешування-запропонований спосіб дозволяє забезпечити стиск і потенційно підвищити автентичність даних.

3.5. Експериментальні дослідження статистичної безпеки ключових хеш-функцій

Важливим питанням практичного використання ключових хеш-функцій для забезпечення цілісності та автентичності транзакцій в Інтернет-платіжних системах є дослідження їх безпеки.

Дослідження безпеки ключових хеш-функцій проводилися відповідно до методики тестування NIST SP 800-22, рекомендованої Національним інститутом по стандартизації і технологіям США.

Для проведення тестування були взяті наступні параметри:

- довжина тестуємої послідовності n = 106 біт;
- кількість тестуємих послідовностей m = 100;
- рівень значимості $\alpha = 0{,}01$.

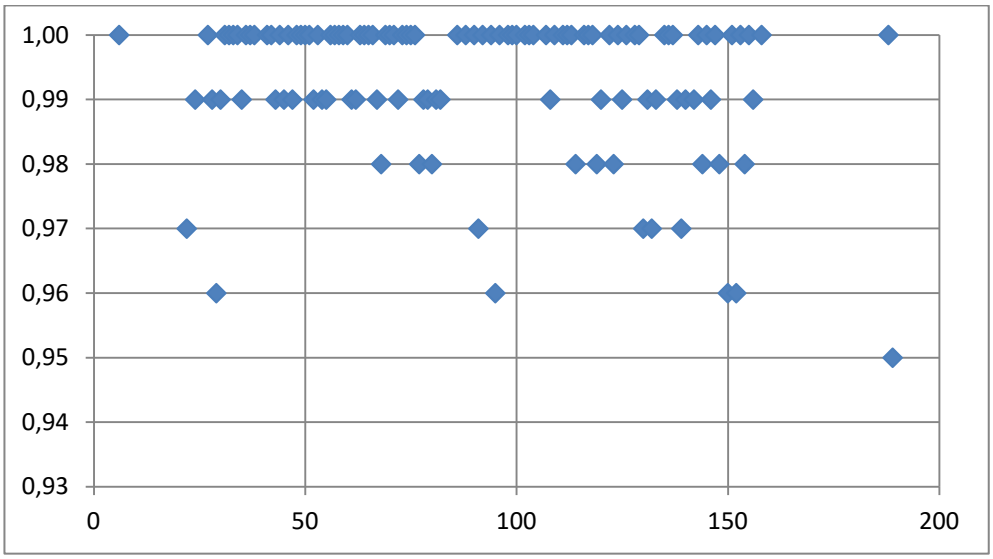
Результати тестування ключових хеш-функцій зведені в табл. 3.1.

В табл. 3.1 наведені результати дослідження алгоритмів MASH-1, MASH-2 та MASH реалізований з використанням арифметики еліптичних кривих (MASH(EC)), HMAC-SHA-256, MACTripleDES, RIPEMD-160. Алгоритми HMAC-SHA-256, MACTripleDES, RIPEMD-160 були реалізовані за допомогою классів, які визначені в мові програмування C#. Лістинг програми наведений у додатку В.

Таблиця 3.1 -Результати тестування ключових хеш-функцій

Статистичні дані	MASH-1	MASH-2	MASH (EC)	HMAC-SHA-256	MAC Triple DES	RIPEMD-160
Кількість тестів, у яких тестування пройшло 99% послідовностей	101	126	142	134	138	129
Кількість тестів, у яких тестування пройшло 96% послідовностей	147	189	189	187	189	189
Кількість тестів, у яких значення ймовірності P <=0,01	4	0	1	3	1	20
Кількість тестів, у яких значення ймовірності P ≤0,001	16	0	1	10	3	1
Кількість тестів, у яких значення ймовірності P ≤0,05	4	8	4	5	6	6
Припустиме значення частки проходження тесту для вибірки розміром 100 двійкових послідовностей.	0,96015					
Припустиме значення частки проходження тесту для вибірки розміром 71 двійкових послідовностей для тесту Random-Excursion	0,954575					

На Рис.3.12. представлено статистичний портрет програмної реалізації ключової хеш-функції MASH-1. Статистичний портрет представляє із себе діаграму ймовірностей проходження відповідних статистичних тестів. Із представленого рисунка видно, що статистичний портрет програмної реалізації хеш-функції MASH-1 відповідає пропонованим вимогам – по всіх тестах позитивно пройшло більше 99 % послідовностей, одна атака на даний алгоритм дозволяє отримати позитивний результат.



Рисуно к3.12. -Статистичний портрет програмної реалізації ключових хеш-функцій, побудованих на основі MASH-1

На Рисунок 3.13. представлено статистичний портрет програмної реалізації ключової хеш-функції MASH-2. З представленого рисунка видно, що статистичний портрет програмної реалізації ключової хеш-функції MASH-2 відповідає пропонованим вимогам – по всіх тестах позитивно пройшло більше 100% послідовностей, що вказує на їх статистичну стійкість.

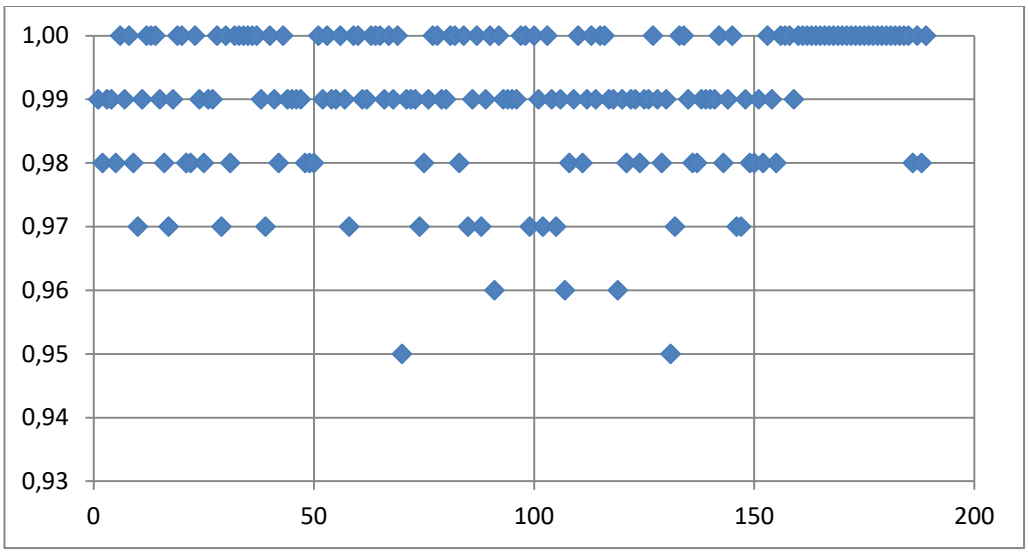


Рисунок 3.13- Статистичний портрет програмної реалізації MASH-2

Для приклада на Рисунок3.14. представлено статистичний портрет програмної реалізації алгоритму блокового симетричного шифрування MASH (EC) у режимі лічильника.

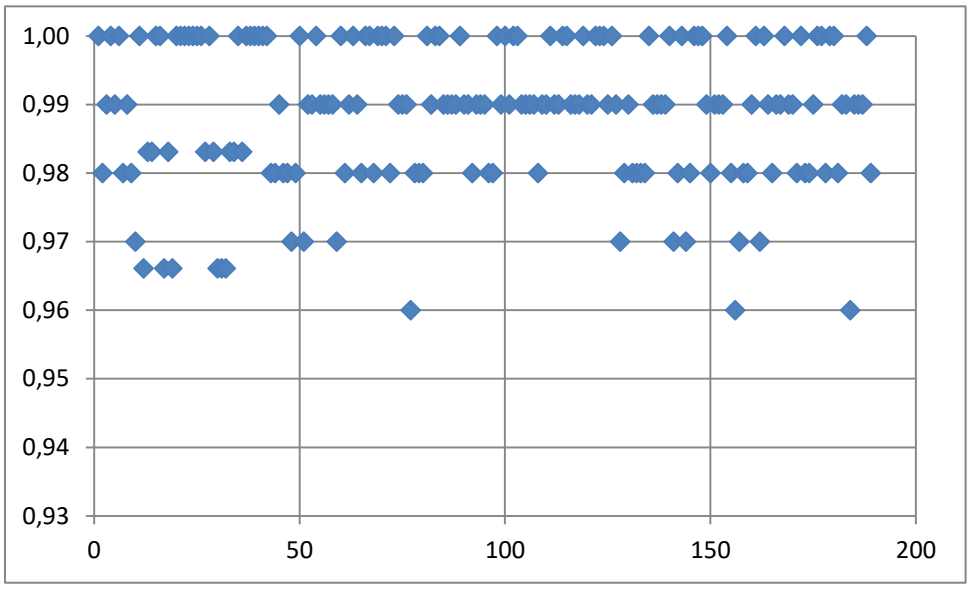


Рисунок 3.14 - Статистичний портрет програмної реалізації MASH (EC)

У табл. 3.2. зведені остаточні результати тестування програмної реалізації ключових хеш-функцій MASH-1, MASH-2, MASH(EC), HMAC-SHA-256, MACTripleDES та безключової RIPEMD-160 Генератори на MASH(EC) і HMAC-SHA-256 мають гарні статистичні властивості. Дійсно, за результатами дослідження статистичної безпеки видно, що ключові хеш-функції забезпечують проходження тестів з більшою ймовірністю, чим алгоритмі безключових хеш-функцій.

Таблиця 3.2 -Остаточні результати тестування

Генератор	Кількість тестів, у яких тестування пройшло більше 99% послідовностей	Кількість тестів, у яких тестування пройшло більше 96% послідовностей
MASH-1	101 (53%)	147 (75%)
MASH-2	126 (67%)	189 (100%)
MASH(EC)	141 (74%)	189 (100%)

Закінчення табл. 3.2

Генератор	Кількість тестів, у яких тестування пройшло більше 99% послідовностей	Кількість тестів, у яких тестування пройшло більше 96% послідовностей
HMAC-SHA-256	134 (71%)	187 (98%)
MACTripleDES	138 (73%)	189 (100%)
RIPEMD-160	129 (68%)	189 (100%)

3.6 Висновки до третього розділу

1. Проведені дослідження показали, що застосування еліптичних кривих у алгоритмах ключового хешування дозволяє забезпечити необхідну стійкість та автентичність передачі даних в ІПС. Вироблено пропозиції за апаратною і програмною реалізацією розроблених алгоритмів ключового хешування з використанням арифметики еліптичних кривих, які дозволяють ефективно реалізувати алгоритми формування MAC-кодів для забезпечення транзакцій вільної довжини.

2. Проведені експериментальні дослідження статистичної безпеки алгоритмів MASH-1 і MASH-2 та запропонованих алгоритмів MASH-1 і MASH-2 на еліптичних кривих показали, що всі 189 статистичних тестів (відповідно до методики тестування NIST SP 800-22) ключові алгоритми побудовані на еліптичних кривих успішно пройшли із критерієм $P_i > 0{,}96015$, що відповідає вимогам даного тестування. Крім того 74% тестів успішно пройшли за критерієм $P_i > 0{,}99$, що є кращим результатом, у порівнянні з алгоритмами HMAC-SHA-256, RIPEMD-160.

3. Проведені дослідження особливостей обміну транзакцій в ІПС дозволили виробити практичні рекомендації з використання ключових алгоритмів побудованих на еліптичних кривих для забезпечення вимог до механізмів цілісності та автентичності в ІПС, що дозволило розв’язати протиріччя і, таким чином, вирішити поставлене наукове завдання.

4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ХОРОНИ ПРАЦІ

4.1 Методи боротьби з монотонністю праці на виробництві

Реалізація творчих здібностей особистості, підвищення мотивації до праці за рахунок так званого «збагачення» праці набувають все більшого значення в розвитку виробництва на сучасному етапі.

Обгрунтування системи заходів по запобіганню монотонності і її негативних наслідків базується на вченні І. П. Павлова і І. М. Сєченова про необхідність розширення поля коркової активності в процесі праці і виключення «довбання» в одну клітину.

Найрадикальнішим заходом є проектування раціональних трудових процесів і операцій на основі оптимального поділу праці. Завдання полягає в тому, щоб кожну операцію зробити змістовною, яка сприяла б розвитку у працівника творчого мислення. Основним принципом проектування раціонального трудового процесу (операції) є принцип збереження певної логічної завершеності і структурної цілісності виконуваної операції. Навіть в умовах глибокої диференціації технологічного процесу необхідно встановлювати таку кількість елементів операції і послідовність їх виконання, яка сприймалася б працівником як логічно завершена одиниця.

Другим важливим принципом проектування трудового процесу є забезпечення достатнього енергетичного рівня операції. Спеціальними дослідженнями встановлено, що негативні психічні стани більшою мірою виявляються при виконанні тих робіт, які через незначну енергетичну вартість не стимулюють функціональної активності організму. Якщо монотонна робота досить інтенсивна за затратами енергії, то нудьга, сонливість, психічне перенасичення можуть не виникати. Доведено, що при фізичній роботі для підтримання активного тонусу кори затрати енергії не повинні бути меншими за 2,5 ккал/хв (150 ккал/год).

Запобіганню монотонності і підвищенню змістовності праці сприяє укрупнення трудових операцій. Завдяки укрупненню операцій у працівника форму-

ється більш складний стереотип трудових дій, що позитивно позначається на стані психофізіологічних функцій. Досвід показує, що операція повинна складатися не менш як з 5—6 елементів за умови збереження цільового змісту.

Важливим засобом боротьби з монотонністю є чергування операцій, кожна з яких є монотонною. Науковою основою чергування операцій є ефект Сєченова, суть якого в тому, що при зміні діяльності активізується інша група нервових центрів, а в раніше працюючих ефективно відбувається «заправка» енергією. Отже, принцип чергування операцій полягає в заміщенні і компенсації психофізіологічних функцій, активізації інших м'язових груп, нервових центрів, зменшенні надмірного напруження працюючих м'язів. Значення чергування операцій, таким чином, полягає в ліквідації негативного впливу однобоких навантажень. На практиці застосовується декілька варіантів чергування операцій: через кожну годину, через 2,5 год, один раз протягом зміни, через день. Відносно зняття фактора монотонності найбільш ефективне чергування операцій один раз протягом зміни, хоча в конкретних виробничих умовах це питання вирішується по-різному. Враховуються умови праці, структура операцій, майстерність працівників.

Чергування операцій пов'язане з суміщенням професій і трудових функцій. Зазначимо, що оволодіння працівником другими і суміжними професіями, крім подолання монотонності і підвищення привабливості праці, підвищує конкурентоспроможність працівника на ринку праці і мобільність на самому підприємстві.

Для зняття монотонності необхідно, щоб операції відрізнялися за характером навантажень, але в той же час були позбавлені інтерферентних елементів.

Основні умови суміщення професій і трудових функцій, які забезпечують зменшення монотонності:

- суміщувані професії повинні змінювати рівень завантаженості різних органів і систем;
- суміщувана операція повинна бути легшою, ніж основна. При легкій монотонній роботі ефективна зміна на більш важку;

- більш монотонну роботу необхідно суміщувати з менш монотонною;
- суміщувані трудові комплекси повинні забезпечувати роботу за участю м'язів-антагоністів, а також зміну робочих поз;
- статичні навантаження повинні компенсуватися помірними динамічними навантаженнями.

При організації монотонних робіт важливе значення має вибір темпу роботи. Темп може бути вільним або примусовим. Кожний з них має переваги і недоліки. Тому при виборі темпу роботи слід виходити зі специфіки конкретного виробництва. В одних випадках доцільним є оптимальний заданий темп з регулюванням швидкості конвеєра у відповідності з кривою працездатності. Варіація швидкості не повинна перевищувати 10—15%. В інших випадках ефективне самостійне регулювання робочого темпу. Останнє застосовується на автономних конвеєрах, що забезпечує не лише свободу ритму, а й регулювання змісту роботи.

Ефективним засобом боротьби з монотонністю є бригадно-групова форма організації потоку. Суть її в тому, що бригада виконує операції всього циклу по виготовленню більш-менш закінченого продукта (вузла). Процеси виготовлення кожного вузла виділяються в самостійні виробничі секції. Робітники працюють у вільному ритмі, а вузли з'єднуються в монтажній секції. В цьому випадку трудовий процес менше розчленований і тісніше кооперований.

Зменшенню негативного впливу монотонних робіт на психічний стан працівників і показники їхньої праці сприяють такі заходи:

- раціоналізація режимів праці і відпочинку;
- естетизація виробничого середовища;
- застосування функціональної музики.

До факторів зменшення монотонності відносяться також психологічні заходи, покликані посилити внутрішні мотиви діяльності. Це, зокрема, психологічна стимуляція трудової діяльності за рахунок постановки проміжних виробничих цілей, забезпечення працівників поточною інформацією щодо виконання

роботи. Особливе значення мають залучення робітників до управління і розв’язання виробничих проблем, а також сприятливий соціально-психологічний клімат, створення умов для спілкування в процесі праці, якщо це можливо. Все це формує позитивні емоційні стани у працівників, посилює їх монотоностійкість.

4.2 Управління та нагляд за безпекою життєдіяльності в Україні

Основою управління безпекою є система організаційно-розпорядчих заходів з профілактики і протидії негативним факторам - недоліки, проблеми, кризові ситуації, які порушують стійке функціонування та розвиток держави, регіона, погрожуючи небезпекою окремій людині.

Управління - це завжди система взаємопов'язаних елементів. Вона складається з чотирьох основних структур. Перша - суб'єкти управління - органи, що відповідають за управління безпекою. Друга - об'єктивна система принципів, правил, певних обмежень, які формують структуру управління. Третя - сукупність інформаційних процесів, необхідних для регулювання стану об'єкта, його контроля. Четверта - кваліфікований, структурований персонал, здатний контролювати ситуацію, приймати рішення. Ця структура багатопланова, складна. Її потрібно розглядати на загальнодержавному, регіональному, місцевому та локальному рівнях.

Вся державна система - Кабінет Міністрів України, Національна Рада з питань безпеки життєдіяльності населення, Комітет з нагляду за охороною праці, структури Міністерства безпечної життєдіяльності, Служба безпеки України, Міністерство внутрішніх справ, органи державного пожежного нагляду, місцеві державні адміністрації та Ради депутатів, їх Виконавчі комітети.

Важливим державним органом є Національна рада з питань безпечної життєдіяльності населення, яка створена відповідно до Закону України "Про охорону праці". Основне її призначення розробка та реалізація державної полі-

тики в галузі охорони життя людей на виробництві та профілактики побутового травматизму, створення системи державного управління цією галуззю.

Національна рада у своїй діяльності керується Конституцією і законами України, постановами Верховної Ради України, указами і розпорядженнями Президента України, декретами, постановами і розпорядженнями Кабінету Міністрів України, а також Положенням про Національну раду з питань безпеки. Вона розробляє та здійснює заходи щодо створення цілісної системи державного управління охороною життя людей на виробництві та профілактики побутового травматизму, вносить на розгляд Кабінету Міністрів України пропозиції про вдосконалення цієї системи; організує і забезпечує контроль за виконанням законодавчих актів і рішень Уряду України.

Рада сприяє впровадженню в життя Національної програми з безпечної життєдіяльності та законів, пов'язаних з реалізацією державної політики з питань безпечної життєдіяльності населення. Вона подає Кабінету Міністрів України пропозиції щодо вдосконалення законодавства з цих питань та координує діяльність центральних і місцевих органів державної виконавчої влади в галузі охорони життя людей на виробництві та профілактики побутового травматизму.

Цей орган організує перевірки діяльності центральних і місцевих органів державної виконавчої влади і заслуховує на своїх засіданнях або засіданнях бюро Національної ради звіти керівників з питань, що входять до її компетенції. Її представники беруть участь у міжнародному співробітництві, сприяючи вивченню, узагальненню і поширенню досвіду у галузі охорони життя людей на виробництві та профілактики побутового травматизму, вирішує питання контролю за виконанням укладених договорів і угод у цій галузі.

Рішення Національної ради та її бюро, прийняті в межах їх компетенції, є обов'язковими для виконання центральними та місцевими органами державної виконавчої влади, підприємствами, установами, організаціями та громадянами.

Крім центральних державних органів управління та нагляду за станом безпеки важливе місце займають місцеві органи - обласні, міські, на виробни-

чих об'єктах, які повинні контролювати стан безпеки, охорони праці на кожному підприємстві.

Важливу роботу виконують державні інспектори, які проводять обстеження стану підприємств, фіксують порушення нормативних актів з охорони праці, призупиняють роботу виробництв та об'єктів, на яких виникала пряма загроза здоров'ю, життю працюючих. Велику роль відіграють експертно-технічні центри, які займаються технічною експертизою стану обладнання, промислових об'єктів.

Важливе значення має робота з перегляду нормативно правових актів щодо державної експертизи проектної документації з питань охорони праці, здоров'я робітників, експертизи екологічного стану підприємств, місць їх розташування.

Враховуючи багаторівневість та багатоаспектність системи управління безпекою життєдіяльності, необхідно використовувати методи програмно-цільового та програмно-орієнтовного управління, тобто враховувати специфічні особливості кожної конкретної ситуації, місцевості, об'єкта.

Програмно-цільовий метод вимагає участі структур, які відносяться до різних відомств у рішенні системних задач, орієнтуючи всю систему управління безпекою життєдіяльності на кінцеву ціль - безпечна життєдіяльність всього суспільства.

Проблемно-орієнтовний метод акцентує увагу на прийняття профілактичних заходів, які б попереджували виникнення кризових ситуацій. Разом узяті ці два методи обумовлюють необхідність проведення робіт за двома напрямками. Перший - зумовлює необхідність загальних розробок концептуально-методологічних основ забезпечення безпеки життєдіяльності на державному рівні. Другий - потребує конкретних розробок, які б забезпечили управління безпекою на інших, нижчих рівнях - місцевому, локальному, об'єктному.

ВИСНОВКИ

1. Одним з найбільш ефективних механізмів забезпечення цілісності і автентичності інформації в сучасних ІПС є електроний цифровий підпис. При формуванні ЕЦП відправник обчислює хеш-функцію документа, що підписує, призначену для стиску та перемішування. Інакше кажучи, такі показники ефективності ЕЦП як криптографічна стійкість і обчислювальна складність реалізації безпосередньо визначаються конструктивними особливостями застосовуваної функції хешування. Однак, як показав проведений аналіз, на сьогоднішній день зростання кількості транзакцій, поява нових погроз, розвиток кібертероризму пред'являє нові вимоги до безпеки та надійності при передачі даних в автоматизованих банківських системах і в першу чергу в Інтернет-платіжних системах.

2. Аналіз методів ключового хешування показав, що алгоритми переможці міжнародного криптографічного конкурсу MASH-1 та MASH-2, засновані на модулярній арифметиці і забезпечують максимальну криптостійкість транзакцій за рахунок забезпечення довжини дайджесту хеш-коду у 1024 біта. Однак швидкодія криптоперетворень має низький рівень, що обумовлено використанням процедури пошуку простих чисел великої довжини. Використання арифметики еліптичних кривих дозволить використовувати одне значення модуля, який є добутком простих чисел великої довжини, зменшити довжину вхідної послідовності.

3. Практичне значення результатів досліджень полягає в наступному: написані теоретичні рекомендації використання процедур ключового хешування для забезпечення цілісності та автентичності даних в Інтернет-платіжних системах. Досліджені та порівняні оцінки ймовірносно-часових характеристик алгоритмів ключового хешування для забезпечення цілісності та автентичності транзакцій в Інтернет-платіжних системах. Досліджені алгебраїчні властивості методів ключового хешування в алгоритмах MASH-1 та MASH - 2, обґрунтовані шляхи використання арифметики еліптичних кривих для їх подальшого вдо-

сконалювання, програмно реалізований один із них. Результати досліджень використані у науково-дослідній роботі “Дослідження перспективних методів і механізмів, забезпечення цілісності та автентичності даних, що циркулюють у внутріплатіжній системі комерційного банку”.

4. Наукове значення заключається у проведенні аналізу існуючих методів ключового хешування для забезпечення цілісності та автентичності інформації в Інтернет-платіжних системах, розробленні методів удосконалення алгоритмів ключового хешування, які побудовані з використанням арифметики еліптичних кривих, визначені практичні рекомендації щодо їх використання в Інтернет-платіжних системах.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1 Інформаційні технології. Криптографічний захист інформації. Цифровий підпис, що ґрунтується на еліптичних кривих. Формування та перевірка : ДСТУ 4145-2002. − [Чинний від 2002-01-01]. − К. : Держстандарт України 2002. − 34с. − (Національний стандарт України).

2 Термінологія в галузі захисту інформації в комп'ютерних системах від несанкціонованого доступу: НД СТЗІ 1.1-003-99. − [Чинний від 28.04.1999]. − К.: Держстандарт України 1999. — 24 с. − (Національний стандарт України).

3 ГОСТ 34.310-95. МЕЖГОСУДАРСТВЕННЫЙ СТАНДАРТ. Информационная технология. КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ. Процедура выработки и проверки электронной цифровой подписи на базе асимметричного криптографического алгоритма. Киев. Госстандарт Украины. 1998.

4 ГОСТ 34.311-95. Межгосударственный стандарт. Информационная технология. Криптографическая защита информации. Функция хеширования. Киев. Госстандарт Украины. 1998.

5 ГОСТ Р 34.10-94. Информационная технология. Криптографическая защита информации. Процедуры выработки и проверки электронной цифровой подписи на базе асимметричного криптографического алгоритма.

6 ГОСТ Р34.11-94. Информационная технология. Криптографическая защита информации. Функция хэширования.Криптография в банковском деле. М. И. Анохин, Н. П. Варновский, В. М. Сидельников, В. В. Ященко — М.: МИФИ. 1997. 274с.

7 Закон Україні «Про захист інформації в автоматизованих системах», Відомості Верховної Ради (ВВР), 1994, N 31, ст.286 - Режим доступу до закону: http://zakon1.rada.gov.ua/cgi-bin/laws/main.cgi

8 Інформаційні технології. Криптографічний захист інформації. Цифровий

підпис, що ґрунтується на еліптичних кривих. Формування та перевірка : ДСТУ 4145-2002. ─ [Чинний від 2002-01-01]. ─ К. : Держстандарт України 2002. ─ 34с. ─ (Національний стандарт України).

9 Интернет- платежная система Portmone. – http://portmone.com

10 Термінологія в галузі захисту інформації в комп'ютерних системах від несанкціонованого доступу: НД СТЗІ 1.1-003-99. ─ [Чинний від 28.04.1999]. ─ К.: Держстандарт України 1999. — 24 с. ─ (Національний стандарт України).

11 Аволио Ф. Защита информации на предприятии / Ф.Аволио, Г. Шипли // Сети и системы связи. ─ М., 2000. ─ №8(58). − С. 91−99.

12 Гольдштейн Б.С. Протоколы сети доступа / Б.С. Гольдштейн. ─ М.: Радио и связь, 2001. – 292с.

13 Диффи У., Хеллман М. Защищенность и имитостойкость/ У. Диффи, М.Хеллман / Введение в криптографию. - 1979.- №3 - С. 79-109.

14 Горбенко И.Д. Сложность арифметических операций в группах точек, эллиптических кривых для криптографических операций / И.Д. Горбенко, С.И. Збитнев, А.А. Поляков // Радиотехника. − 2001. − №. 119. − С. 32−37.

15 Долгов В.І. Конспект лекцій з дисципліни «Спеціальні розділи математики. Теорія груп та кілець» / В.І. Долгов, І. В. Лисицька. ─ Х.: ХНУРЕ, 2000.─ 292с.

16 Евсеев С.П. Зашита информации в Интернет-платежных системах / С.П. Евсеев, О.В. Толстолуцкая // Управління розвитком. ─ Х.: ХНЕУ, 2008. ─ №15. ─ с. 45─47

17 Завало С.Т. Алгебра и теория чисел / С.Т. Завало. ─ К.: Вища шк. Головне видавництво, 1980. ─ 402 с.

18 Иванов М.А. Криптографические методы защиты информации в компьютерных системах и сетях / М.А. Иванов. ─ М.: КУДИЦ-ОБРАЗ, 2001. - С.368.

19 Кнэпп Э. Эллиптические кривые / Э. Кнэпп; Пер. с англ. Ф. Ю. Попеленского. – М.: Изд-во «Факториал Пресс», 2004. − 488 с.

20 Конеев И. Р. Информационная безопасность предприятия / И. Р. Конеев, А.В. Беляев. − Спб.: БХВ-Петербург, 2003. − 752 с.

21 Король О.Г. Обеспечение безопасности в Интернет-платежных системах / О.Г. Король, О.В. Толстолуцкая // Проблемы информатики и моделирования: междунар. науч.-техн. конф., 26-28 ноября 2008 г. : тезисы докл. — Х.: НТУ «ХПИ», 2008. ─ с.56

22 Кузнецов А.А. Алгоритмы аутентификации в сетях ISDN / А.А. Кузнецов, В.Е. Чевардин, И.В. Кучерявенко // Системы обработки информации. – 2003. − №5. – С. 140-145.

23 Кузнецов А.А. Быстрые операции сложения точек рациональной кривой // А.А. Кузнецов, В.Е. Чевардин // IPME НАН Украины. – 2004. − №25. - С.109-114.

24 Кузнецов А.А. Метод ключевого хеширования на основе арифметики алгебраических кривых / А.А. Кузнецов, В.Е. Чевардин, В.Н. Лысенко // Системы обработки информации. – №11(39). −С. 101-108.

25 Кульгин М. В. Технологии корпоративных сетей. Энциклопедия / М. В. Кульгин. – СПб: Питер, 2000. ─ 704 с.

26 Методи захисту банківської інформації / В.К. Задірака О.С. Олесюк, Н.О. Недашковський. ─ К.: Вища школа, 1999. ─ 302с.

27 Молдовян А.А. Криптография: скоростные шифры / А.А. Молдовян, Н.А Молдовян., Н.Д. Гуц, Б.В. Изотов. – СПб.: БХВ-Петербург, 2002. - 496 с.

28 Молдовян А.А. Криптография: скоростные шифры. / А.А. Молдовян, Молдовян, Н.Д. Гуц, Б.В. Изотов. – СПб.: БХВ-Петербург, 2002. − 496 с.

29 Олифер В. Г. Новые технологии и оборудование IP-сетей / В. Г. Олифер, Н. А. Олифер. – С.-Пб.: БХВ, 2000. − 512 с.

30 Острик В. В. Алгебраическая геометрия и теория чисел: рациональные и эллиптические кривые / В. В. Острик, М. А. Цфасман. – М.: МЦНМО, 2001.

- С. 496.

31 Ростовцев А. Г. Подпись и шифрование на эллиптической кривой: анализ безопасности и безопасная реализация / А. Г. Ростовце., Е. Б. Маховенко // Проблемы информационной безопасности. Компьютерные системы. ─ СПб., 2003. ─ №1. ─ С. 64−73.

32 Сереченко Д. MPLS и безопасность / Д. Сереченкот // Сети и системы связи. ─ М., 2004. ─ №13 (119). ─ С. 89─91.

33 Скляр Б. К. Цифровая связь. Теоретические основы и практическое применение / Б.К. Скляр. – М.: Вильямс, 2003. – 1104 с.

34 Соловьев Ю.П. Рациональные точки на эллиптических кривых / Ю.П. Соловьев // Соросовский образовательный журнал. – 1997. – №10. – С. 138-143.

35 Толстолуцька О.В. Применение преобразований в группе точек єллиптической несуперсингулярной кривой для формирования ключевіх хеш-функций / О.В. Толстолуцька // Управління розвитком. ─ Х.: ХНЕУ, 2009. ─ №7. ─ с. 78

36 Толстолуцька О.В. Аналіз механізмів забезпечення цілісності та автентичності в Інтернет-платіжних системах / О.В. Толстолуцька // Управління розвитком. ─ Х.: ХНЕУ, 2009. ─ №4. ─ с. 49─51

37 Чевардин В.Е. Метод итерационного хеширования на базе арифметики в группе точек несингулярной эллиптической кривой / В.Е. Чевардин // Радіоелектронні і комп’ютерні системи. – 2005. − № 3. − С. 99-105.

38 Чевардин В.Е. Метод хеширования на базе арифметики суперсингулярных эллиптических кривых / В.Е. Чевардин // ЕЕЈЕТ. – 2005. − №4. – С. 45-47.

39 Шафаревич И.Р. Основы алгебраической геометрии / И.Р. Шафаревич. ─ М.: Наука, 1972.–568с.

40 Шипли Г. Основы безопасности ИТ / Г. Шипли // Сети и системы связи. М., 2003. ─ №4. ─ С. 78-82.

41 Bierbrauer J. On families of hash function via geometric codes and concatenation

/ J. Bierbrauer, T. Johansson, G. Kabatianskii // Advances in Cryptology – CRYPTO 93, Lecture Notes in Computer Science. – 1994 - № 773 - P. 331-342.

42 Carter J. L. Universal classes of hash functions / J.L.Carter, M.N.Wegman // Computer and System Scince – 1979 - №18 - P. 143-154.

43 Chor B. A Knapsack-type public-key cryptosystem based on arithmetic in finite fields / B.Chor /Advances in Cryptology. - NY: Springer-Verlag, 1985 - 54p.

44 Diffi W. The first Ten Years of Public-Key Cryptography/ W. Diffi/ Computer Since – 1988 – №5 – P. 21.

45 ISO 7498-2:1989- Information technology- Open System Interconnection- Basic reference model-Part 2: Security architecture

46 ISO/IEC 10181-(1-7):1996- Information technology- Open System Interconnection- Security framework for open systems.

47 Jakob Jonsson and Burt Kaliski. RSA-PSS. Primitive submitted to NESSIE by RSA,September 2000

48 Krawczyk H. LFSB-based Hashing and Authenticator./ H. Krawczyk/ Proceedings of CRYPTO Notes in Computer Science. – 1994 - №80 - P. 129-139.

49 McElliece R.J. A public-key cryptosystem based on algebraic coding theory/ R.J. McElliece - NY: Springer-Verlag, 1978. – 116p.

50 Pieprzyk J.P. On public-key cryptosystems built using polynomial rings. In Advanced in Cryptography-Eurocrypt/ J.P.Pieprzyk/ - NY: Springer-Verlag, 1985 – 80 p.

51 Preneel B., Biryukov A., «New European Schemes for Signature, Integrity and Encryption» Final report of European project number IST-1999-12324, NESSIE, April 2004. [p. 487- 623]

52 Simmons G.J. Authentication theory/coding theory in Cryptology/ G.J.Simmons/ Computer Science. – 1985 - №96 - P. 411-431.

53 Simons G.J. An impersonation-proof identity verification scheme/ G.J.Simons / Computer Science. – 1988 – №87 – P. 211-215.

54 Smid M.E. The Past and Future / M.E.Smid, D.K.Branstad / Computer Science. –

1988 – №76 – P. 122-132.

ДОДАТКИ

Додаток А
Лістинг програми реалізації ключевої хеш-функції побудованої на модулярній арифметиці

```
package MASH;


import java.lang.*;
import java.math.*;
import java.io.*;


/**
 *
 * @author Olla
 */
public class CryptoMASH {
   private BigInteger exp;
   private BigInteger p;
   private BigInteger q;
   private BigInteger two=BigInteger.valueOf(2);
   private BigInteger ten=BigInteger.valueOf(10);
   private BigInteger fiften=BigInteger.valueOf(15);
   private BigInteger sixteen=BigInteger.valueOf(16);


 public CryptoMASH(int k) {
 if(k==1) this.exp= BigInteger.valueOf(2).pow(8).add(BigInteger.valueOf(1));
 else this.exp=BigInteger.valueOf(2);
 }
 public void SetKey(BigInteger p,BigInteger q){
     this.p=p;
     this.q=q;
 }

 public BigInteger ComputeHash(byte[] msg) {

long startTime = System.nanoTime();

//Конвертируем сообщение в двоичную строку

BigInteger msg_Int=new BigInteger(1,msg);


BigInteger N_Int=this.p.multiply(this.q);

//System.out.println("N="+N_Int);
//b длина двоичной строки
BigInteger b=BigInteger.valueOf(msg_Int.bitLength());
//System.out.println("b="+b);


//Определить двоичную длину n хеш-кода как наибольшее произведение числа 16 и
длины хеш-кода,
//удовлетворяющее неравенству 16*n < m.
int n=N_Int.bitLength()/16*16;//n is largest multiple of 16 not exceeding
bitlength of modulus
System.out.println("exp="+exp);
//System.out.println("n="+n);
//System.out.println("m="+N_Int.bitLength());
```

```
//Проверка соответствует ли длина двоичной строки (b) неравенству  0 <=
b<=2^(n/2)

if (b.compareTo(two.pow(n/2))>0) throw new IllegalArgumentException ("Message is
too large");//Check that msg is not too large for use with MASH2

//Дополнить, если необходимо, строку x нулевыми битами, для того,
//чтобы получить двоичную строку длиной t = n/2 для наименьшего t >= 1.
int amountToShift=msg_Int.bitLength()%(n/2)==0?0:(n/2)-
msg_Int.bitLength()%(n/2);
msg_Int=msg_Int.shiftLeft(amountToShift);

//Define variable for 2 raised to n power
BigInteger twon=two.pow(n);

//В качестве вектора инициализации выбрать H = 0.
BigInteger H=BigInteger.valueOf(0);

//Определить n-битное целое число в качестве константы A =0xf00...00
BigInteger A=BigInteger.valueOf(15).multiply(two.pow(n-4));

//Разделить дополненный текст на n/2-разрядные блоки x1, ..., xt
//и добавить последний блок  , содержащий n/2-разрядное представление числа b.
int t=msg_Int.bitLength()/(n/2);
//System.out.println("msg_Int0000="+msg_Int);
BigInteger prevH;
BigInteger  rem;        for (int i=0;i<t;i++) {
                    prevH=H;
                    H=BigInteger.valueOf(0);

                    for (int j=n/2-4;j>=0;j-=4) { //Каждый байт нач. с 1111B
                               H=H.shiftLeft(4).or(fiften);
                               //Сдвиг вправо
                             rem=msg_Int.shiftRight(j+n/2*(t-1-i)).mod(sixteen);

                               H=H.shiftLeft(4).or(rem);
                                                  }

H=prevH.xor(H).or(A).modPow(exp,N_Int).mod(twon).xor(prevH);
                          }
                    // t+1 block

                    prevH=H;
                    H=BigInteger.valueOf(0); //Process the 4 bit nybbles
BigInteger rem;
                    for (int j=n/2-4;j>=0;j-=4) {
                    //Each byte in last block begins with 1010B
                    H=H.shiftLeft(4).or(ten);
                    rem=b.shiftRight(j).mod(sixteen);
                    //Append this remainder to H
                    H=H.shiftLeft(4).or(rem);

                                              }
H=prevH.xor(H).or(A).modPow(exp,N_Int).mod(twon).xor(prevH);

//Convert to a byte array and return-call helper method getBytes().
int gg=H.bitLength();
long estimatedTime = System.nanoTime() - startTime;
 System.out.println("Время затраченное на формирование хеша=="+estimatedTime);
 return H;
```

```
}}
```

Додаток Б
Лістинг програми реалізації ключевої хеш-функції побудованої на арифметиці еліптичних кривих

```
/*
 * Клас, який реалізує точку еліптичної кривої
 */

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package cryptoec;


import java.lang.*;
import java.math.*;
import java.io.*;
import java.util.Random;

public class ECPoint {


  private BigInteger X;
  private BigInteger Y;
  private BigInteger t;
  private static BigInteger field;


    private  BigInteger  add(BigInteger p1,BigInteger p2){
        return p1.xor(p2);
    }

    private BigInteger multiply(BigInteger p1,BigInteger p2, BigInteger f){
  int size;
  BigInteger s=BigInteger.valueOf(0);

  BigInteger z=BigInteger.valueOf(0);
   for(int i=0;i<p2.bitLength();i++)
  {
     if(p2.testBit(i)){
      z=p1;
      z=z.shiftLeft(i);
      s=s.xor(z);
       }
      }
    return Mod(s,f);
    }

    private  BigInteger Mod(BigInteger a1, BigInteger f_mod){
        if(a1.bitLength()<f_mod.bitLength())return a1;
   int l=f_mod.bitLength();
   BigInteger t=a1;


BigInteger tt=BigInteger.ZERO;
   while(t.bitLength()>=l)
   {
       int k=t.bitLength();
       tt=f_mod.shiftLeft(k-l);
       t=t.xor(tt);
   }   return t;}
```

```
   public  BigInteger scalarPoint(BigInteger n) {
        BigInteger s = t;
        for (int i = n.bitLength() - 1; i > 0; i--) {
            s = doublePoint(s, field);
            if (n.testBit(i - 1)) {
                s = addPoint(s, t, field);
            }
        }
        return s;
    }
       //инверсия
    public BigInteger inv(BigInteger p1, BigInteger f_mod) {
        BigInteger q = BigInteger.valueOf(0);
        q = q.setBit(f_mod.bitLength() - 1);
        BigInteger e = q.subtract(BigInteger.valueOf(2));
        BigInteger x = p1;
        for (int i = e.bitLength() - 1; i > 0; i--) {
            x = multiply(x, x, f_mod);
            if (e.testBit(i-1)) {
                x = multiply(p1, x, f_mod);
            }
        }
        return x;
    }
       public  BigInteger addPoint(BigInteger t1, BigInteger t2, BigInteger
f_mod) {
        BigInteger tt = add(t1, add(t2, BigInteger.valueOf(1)));
        tt = inv(tt, f_mod);
        BigInteger t3 = multiply(t1, t2, f_mod);
        t3 = multiply(t3, tt, f_mod);
        return t3;
    }

    public  BigInteger doublePoint(BigInteger t1, BigInteger f_mod) {
        BigInteger t3 = multiply(t1, t1, f_mod);
        return t3;
    }
 public void dbPoint(){
     this.t=multiply(t,t,field);

 }

 public BigInteger getX(){
     X=multiply(t,t,field).xor(t).xor(BigInteger.valueOf(1));
return X;
 }

 public BigInteger getY(){
 return multiply(X,t,field);
 }
  public void setT( BigInteger t){
 this.t=t;
 }
     public BigInteger getT( ){
 return this.t;
 }
    public void setField( BigInteger field){
 this.field=field;
 }
}
```

```
/*
 * Клас, що реалізує хешування з використанням арифметики еліптичних кривих
 */

package cryptoec;
import java.lang.*;
import java.math.*;
import java.io.*;

public class ECHash {
    public  BigInteger CreateEChash(byte[] msg,BigInteger prev_k,String str,int
n_int){

 BigInteger two=BigInteger.valueOf(2);
//Конвертируем сообщение в двоичную строку
BigInteger msg_Int= new BigInteger(1,msg);
//Дополнить, если необходимо, строку x нулевыми битами, для того,
//чтобы получить двоичную строку длиной t = n/2 для наименьшего t >= 1.
int amountToShift=msg_Int.bitLength()%(n_int)==0?0:(n_int)-
msg_Int.bitLength()%(n_int);
msg_Int=msg_Int.shiftLeft(amountToShift);

int t=msg_Int.bitLength()/(n_int);

BigInteger  rem;
BigInteger bloc=two.pow(n_int);
ECPoint P=new ECPoint();
ECPoint prevP;
BigInteger k=BigInteger.ONE;
P.setField(BigInteger.valueOf(11));
     for (int i=0;i<t;i++) {

                rem=msg_Int.shiftRight(n_int*(t-1-i)).mod(bloc);
                P.setT(rem);
                k=prev_k.xor(P.getX());
                System.out.println("k="+k);
                P.scalarPoint(k);
                prev_k=k;
         }
            System.out.println("t="+ P.getT().toString(10));
        System.out.println("X="+ P.getX().toString(10));
        System.out.println("Y="+ P.getY().toString(10));
        System.out.println("Res="+   P.getX().xor(P.getY()).toString(10));
        return P.getX().xor(P.getY());
    }

}
```

Додаток В

Лістинг програми реалізації ключевих хеш-функції HMAC-SHA-256, MACTripleDES та безключової RIPEMD-160

```
using System;
using System.Linq;
using System.Text;
using System.Security.Cryptography;
using System.IO;


namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
                                            {

const int  initRnd = 77;
Random repeatRnd = new Random(initRnd);

 //////реалізація ключового алгоритму MACTripleDES////////////
Encoding encoding2 = System.Text.Encoding.GetEncoding(1251);


                    for (int i = 0; i < 100000; i++)
                    {
                    KeyedHashAlgorithm MACTripleDES1 = MACTripleDES.Create();
                    byte[] bar = new byte[10];
                    repeatRnd.NextBytes(bar);
                    byte[] hash = MACTripleDES1.ComputeHash(bar);
                    StreamWriter sw = new StreamWriter("C:\\Documents and Set-
tings\\Student\\1.asc", true, encoding2);
                    sw.Write(encoding2.GetString(hash));
                    sw.Close();
                    }

////// реалізація ключового алгоритму HMACSHA256////////////
                    for (int i = 0; i < 100000; i++)
                    {
                    HMAC sha1 = HMACSHA256.Create();
                    byte[] bar = new byte[10];
                    repeatRnd.NextBytes(bar);
                    byte[] hash = sha1.ComputeHash(bar);
                    StreamWriter sw = new StreamWriter("C:\\Documents and Set-
tings\\Student\\2.asc", true, encoding2);
                    sw.Write(encoding2.GetString(hash));
                    sw.Close();
                    }

////// реалізація безключового алгоритму RIPEMD160 ////////////
                    for (int i = 0; i < 100000; i++)
                    {
                    RIPEMD160 RP = RIPEMD160.Create();
                    byte[] bar = new byte[10];
                    repeatRnd.NextBytes(bar);
                    byte[] hash = RP.ComputeHash(bar);
                    StreamWriter sw = new StreamWriter("C:\\Documents and Set-
tings\\Student\\3.asc", true, encoding2);
                    sw.Write(encoding2.GetString(hash));
                    sw.Close();                  }}
```