Шифр: Кактус

Назва роботи: «Системи електронного голосування на основі блокчейн. Проблеми, атаки, пропозиції»

Харків 2021

ЗМІСТ

ВСТУП 3
РОЗДІЛ 1. ОГЛЯД ТЕХНОЛОГІЙ БЛОКЧЕЙН 5
РОЗДІЛ 2. ЗАГРОЗИ МЕРЕЖ ЗАСНОВАНИХ НА БЛОКЧЕЙНІ 8
2.1 Атаки на користувачів 8
2.2 Атаки на мережеву взаємодію 9
2.3 Атаки на блокчейн 11
2.4 Атаки на протоколи консенсусу 12
РОЗДІЛ 3. ПРОЄКТУВАННЯ СИСТЕМИ ЕЛЕКТРОННОГО ГОЛОСУВАННЯ 18
3.1 Чинні системи електронного голосування 18
3.2 Гомоморфне шифрування 21
3.3 Система електронного голосування з використанням технології блокчейн 27
3.4 Прозорість голосування 32
ВИСНОВКИ 35
ПЕРЕЛІК ПОСИЛАНЬ 37
ДОДАТОК А 38
ДОДАТОК Б 41

ВСТУП

Голосування є однією з найважливіших аспектів суспільного життя громадян демократичної держави. Саме в такий спосіб у більшості країн розв'язувати важливі питання, що стосуються не тільки внутрішнього політичного строю, але і ситуації у світі в цілому. Референдуми, що проводять кардинально змінюють вектори розвитку країн, екологічне становище, тощо. Право на волевиявлення гарантується абсолютною більшістю конституцій світу. Голосувати мають право працездатні повнолітні громадяни. Згідно з Конституцією України вибори є вільними, прямими, добровільними, особистими, прозорими та з таємним волевиявленням. Дотримання кожної з цих вимог є безумовно важливим та у разі порушення хоча б однієї може свідчити про порушення прав громадян України. Наразі голосування на виборах у президенти України відбувається кожні п’ять років.

Більшість країн використовує метод голосування що є незмінним протягом декількох тисячоліть - виборець (громадянин, що відповідає певним критеріям) повинен віддати свій “голос” (деякий унікальний предмет або підпис) за претендента на посаду, рішення референдуму, тощо. Фізична присутність виборця необхідна для підтвердження згоди на передачу голосу або безпосередньої фізичної передачі його до так званих “урн” - місце збереження.

З широким розповсюдженням електронних приладів з’явилась можливість реалізувати вибори з використанням електронного голосування. Електронне голосування (ЕГ) може бути як офлайн (потребує фізичної присутності виборця на виборчій дільниці), так і онлайн (може бути проведено дистанційно)[1].

Кожен з цих видів поділяється на підвиди:

Офлайн:

- повністю електронне;
- з електронним підрахунком голосів.

Онлайн:

- за допомогою Інтернет-ресурсу;
- за допомогою мобільного телефону.

Електронне офлайн голосування хоч і має певні переваги щодо підрахунку голосів, прозорості та швидкості, проте спадкує проблеми звичайного голосування – необхідність присутності, можливість саботажу, порушення тих чи інших норм голосування, тощо.

Докладніше ми розглянемо електронне голосування за допомогою Інтернет-ресурсів, бо цей варіант є найперспективнішим та цікавішим з точки зору інформаційної безпеки. Електронне голосування через інтернет ще називають інтернет-голосуванням. Існують різні варіанти реалізації систем електронного голосування, кожний з яких створювався під спеціалізовані потреби та конкретні випадки.

У світі вже є достатньо прецедентів проведення електронного голосування, тому спираючись на опит інших країн, можливо зробити висновки стосовно ефективності, надійності, зручності та прозорості виборів. Також проаналізувавши атаки та вразливості інших систем, існує можливість заздалегідь знайти методи вирішення та способи протидії.

Таким чином, використовуючи новітні технології, криптографію та досвід заснований на інших системах, ми маємо змогу розробити модель електронного голосування, що буде відповідати високим стандартам надійності, прозорості та достовірності.

РОЗДІЛ 1. ОГЛЯД ТЕХНОЛОГІЙ БЛОКЧЕЙН

Блокчейн – систематизований ланцюг блоків (зв'язаний список), розподілений між усіма вузлами мережі. Кожний блок містить в собі інформацію про усі транзакції, що були оброблені в мережі. Таким чином історія мережі зберігається в кожному з вузлів, що робить фальсифікацію транзакцій значно складнішою. Система у такому разі є прозорою і кожний з вузлів може по бажанню проводити аудит.

Важливо зазначити інші технології, які роблять блокчейн систему більш надійним способом зберігання інформації. Геш-функції у блокчейн слугують своєрідним підписом, перевірка якого може слугувати підтвердженням відповідності. В такому разі криптостійкість системи напряму залежить від криптостійкості геш-функції. Своєчасне оновлення крипто стандартів є важливим для забезпечення надійності усієї мережі

Зрозуміло, що використання тільки геш-сум не є достатнім для забезпечення необхідного рівня безпечності, тому використовують так зване дерево Меркла. Дерево Меркла (рисунок 1.1) це спеціалізована структура даних яка пов'язує геші попередніх блоків з наступними блоками. Виконана ця структура у вигляді двійкового дерева. Спочатку ми беремо пару геш-значень обчислених для певних блоків, на які поділено деякий масив даних та обчислюємо їх загальну геш-суму. В тому разі, якщо пари в блоку немає, він потрапляє на наступний рівень без змін. Обчислення геш-сум пар виконується до тих пір, доки не залишиться лише одне геш-значення. Дерево Меркла гарантує збереження цілісності та достовірності збереженої інформації.

Верхнє геш-значення (*root hash*) потрібно для безпосереднього порівняння. Перш, ніж підтверджувати трансакцію, необхідно завантажити верхній геш з надійного джерела, яке не обов'язково повинно бути звичайним вузлом мережі.

Для цього підійде Інтернет-ресурс з надійною та захищеною базою даних. Далі порівнюють отримані геш-значення з перевіреним. Якщо геш-значення не збігаються, це означає що геш-значення було пошкоджено або скомпрометовано.

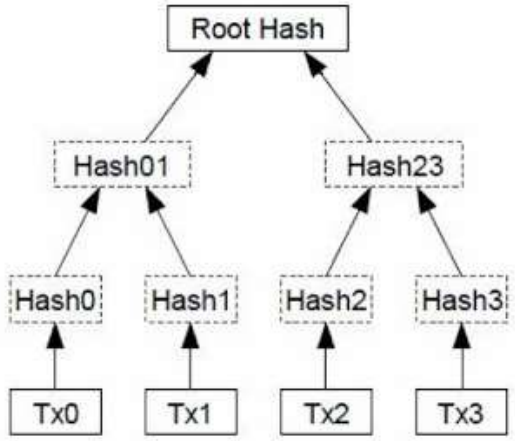

Рисунок 1.1 - Структура дерева Меркла

У такому разі потрібно обчислювати геш-значення на іншому вузлі та продовжувати процедуру доки співвідношення не буде знайдене. Ця структура даних створена таким чином, щоб мінімізувати можливість модифікації даних у мережі.

У блокчейн мережі криптографічні геш-функції використовують для:

- створення адреси користувача;
- створення унікальних ідентифікаторів;
- захисту цілісності даних блоку;
- захисту заголовка блоку.

Геш функції повинні бути незворотними - тобто з геш-значення неможливо отримати вхідне значення, детермінованими - завжди, при будь-яких обставинах одному і тому самому вхідному значенню відповідатиме те саме геш-значення, унікальними - одному вхідному значенню відповідає лише одне геш-значення, мати невелику кількість колізій (збігів), всі геш значення мають однакову бітову довжину. Проте іноді навіть цих критеріїв недостатньо для гарантування відповідного рівня захисту, тому блокчейн мережа повинна бути достатньо адаптивною, щоб при необхідності була змога змінити алгоритм знаходження підписів.

Розглянемо структуру блоку на прикладі блокчейн мережі Біткоїну.

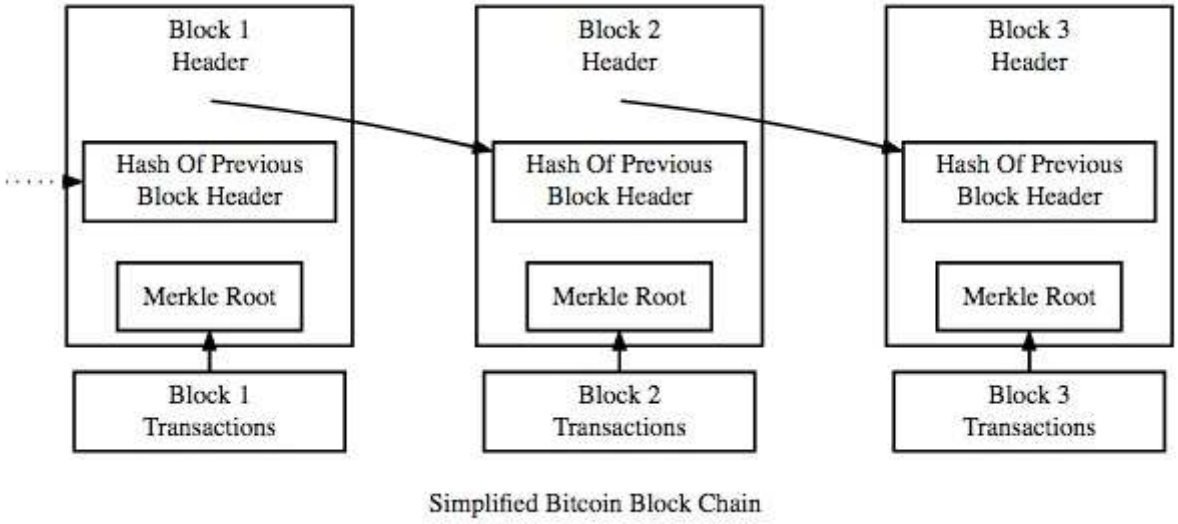

Рисунок 1.2 - Структура блоку блокчейн

Як бачимо, кожний блок містить в собі посилання на попередній блок. Це посилання виконано у вигляді геш-значення попереднього блоку. Процес перевірки попереднього блоку у такому разі зводиться до знаходження геш-значення заголовку імовірного блока-пращура. Далі знаходиться сегмент зі значенням деревом меркла [4]. Перша частина несе в собі організаційні та функції безпеки. Саме ці поля відповідають за безпечність мережі.

Наступний сегмент – корисна інформація. У такому випадку – інформація про транзакцію. Саме ця частина блоку є метою його формування. Ця інформація додаючись у блокчейн та стає незмінною. Інформація про транзакцію з моменту появи у блокчейні зберігається на кожному з нодів для аудиту та забезпечення безпечності усієї мережі.

РОЗДІЛ 2. ЗАГРОЗИ МЕРЕЖ ЗАСНОВАНИХ НА БЛОКЧЕЙНІ

Додавання нового блоку в мережу блокчейн потребує отримання консенсусу від заздалегідь обраної групи. Консенсус, по суті являє собою згоду вузлів на створення нового блоку. Через те, що вже додані блоки в блокчейні не можуть бути змінними, процедурі додавання нових блоків повинно привертатися багато уваги. Алгоритм консенсусу в блокчейні настільки ж важливий як обрана криптографія, бо атаки, що передбачають маніпулювання алгоритмом консенсусу зустрічаються набагато частіше ніж атаки наприклад на геш-функції.

Як правило, в протоколах застосовують такі алгоритми консенсусу - Proof-of-Work (PoW) - доказ роботи, Proof-of-Stake (PoS) - доказ власності. Звичайно існують і інші, проте вони досить екзотичні і рідко використовуються, проте їх ми теж розглянемо.

Атаки в області блокчейн-технологій можна поділити на кілька основних напрямів. Перше — це атаки на окремих користувачів криптовалют, або інших активних суб'єктів блокчейну. Друга область — це мережна взаємодія між нодами (комп'ютерами мережі). Третє — це безпосередньо сам блокчейн, і обраний консенсус [5]. Розглянемо кожний з векторів детальніше.

2.1 Атаки на користувачів

Мета зловмисника у такому разі заволодіти закритими ключами користувача, щоб вкрасти його монети або якісь інші права, які представляють йому ці закриті ключі. Засоби можуть використовуватися різні - це віруси, соціальна інженерія, злам, тощо. Всі ці способи не нові та мало відрізняється від крадіжки коштів у традиційних електронних платіжних системах. Ключова відмінність — повернення коштів тут неможливо, тому що скасувати транзакцію в блокчейні в принципі неможливо. Протидіяти цим атакам можна підвищенням рівня комп'ютерної грамотності всіх користувачів.

Крім того, існують більш екзотичні варіанти атак спрямованих на окремих користувачів. Наприклад, існує різновид атак, коли у блокчейн додається якийсь

контент, що збігається з сигнатурами різних вірусів. Якщо у користувача стоїть на комп'ютері антивірус, він може розпізнати цю сигнатуру і видаляти файли блокчейну, і нода таким чином перестане працювати. Другий варіант — це записувати в блокчейні якусь нелегальну інформацію, розкриваючи користувача якимось юридичним ризикам. Третій варіант — базується на тому, що велика доля блокчейн проєктів використовує відкрите програмне забезпечення, розробники якого, в теорії, можуть впровадити бекдор. Крім того, зломавши репозиторій, можна підмінити код на заражений. Мало хто завжди читає безпосередній вихідний код перед тим як його компілювати, проте така проста міра могла б зберегти користувачів від цих потенційних атак. Протидія цим атакам складається в уважному ставленню до ПО та його роботи.

2.2 Атаки на мережеву взаємодію

2.2.1 DoS-атаки

Атака, яка може бути застосована завжди, коли ми маємо справу з мережевою взаємодію, це Denial of Service - відмова в обслуговуванні. Мета — це вивести з роботи окремо взяту ноду або навіть частину мережі так, щоб вона не могла нормально функціонувати. Досягається це надісланням великої кількості запитів до окремих адресатів. Засоби цієї атаки — стандартні засоби DoS-атак, нічим не відрізняються від протидії їм у звичайних мережах, тому протидія стандартна: мережеві фільтри, фаєрволи, хардварні, програмні. Мережі достатньо підтримувати велику кількість нод, щоб економічної було невигідно атакувати якісь окремо вузли. З ростом кількості нод зростає і ціна атаки.

2.2.2 Атака Сибілли (Eclipse attack)

Розглянемо атаку на прикладі Bitcoin. В середньому нода підключається до 8 інших нод для передачі інформації. Можна спробувати зробити так, щоб певна нода, з якої була спроба створити транзакцію, приєднатися до 8 нод і всі 8 нод належали б зловмиснику, який не дасть нам провести транзакцію або дасть нам неправдиву інформацію про поточний статус блокчейну.

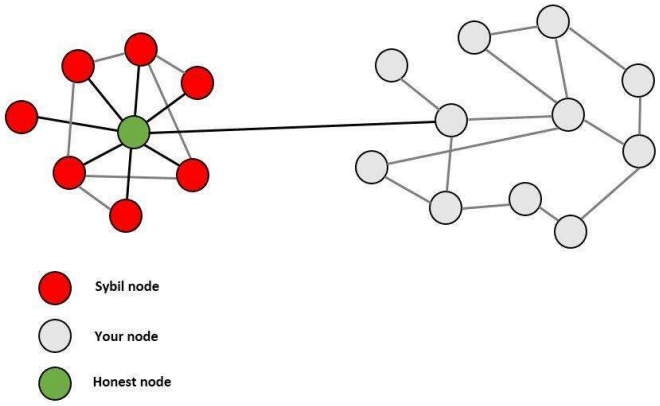

Рисунок 2.1 - Ілюстрація атаки Сибілли

Станом на 2012 рік, дані показали, що масштабні атаки Sybil можуть бути здійснені дуже дешево та ефективно в чинних системах, таких як наприклад BitTorrent Mainline DHT. Чим більше ідентифікаторів належить зловмиснику, тим більше шансів, що наступний сеанс деякого користувача з p2p-мережею виявиться замкнутий на цих вузлах-псевдонімах. Через відсутність надійного центру, в одноранговій мережі є 2 способи визнати новий ідентифікатор: або отримати гарантії його сумлінності від інших вузлів, або самостійно перевірити його яким-небудь чином.

При прямій перевірці:

- Навіть якщо ресурси обмежені, зловмисник все одно може контролювати якесь число ідентифікаторів.
- Зловмисник може створювати ідентифікатори-псевдоніми знову і знову, якщо він не зобов'язаний підтверджувати володіння всіма ними одночасно.

При непрямій перевірці:

- Досить велике число підконтрольних ідентифікаторів дозволяє підробляти необмежену кількість нових.
- Зловмисник завжди зможе контролювати велику кількість ідентифікаторів, якщо він не зобов'язаний безперервно їх підтверджувати.

Зі зростанням децентралізованої мережі зростає і кількість ідентифікаторів-псевдонімів. Стає недоцільно вимагати в кожного користувача підтверджувати володіння своїми ідентифікаторами одночасно і безперервно, оскільки це суттєво заважає масштабованості мережі. Але протидія цій атаці досить легко. Якщо наша нода приєднається хоча б до однієї чесної ноди, припустимо, атака вже не буде успішною через те, що якщо ми підроблюємо інший статус блокчейна з короткшим ланцюгом, нода зможе зрівнявши ланцюги точно зрозуміти який з них є чинним і, відповідно, піде за правильної гілкою консенсусу і наша нода автоматично заблокує всіх зловмисників. Тобто щоб захиститися від цієї атаки, досить підключатися до якомога більшій кількості нод. На сьогодні 8 нод є достатньою кількістю для забезпечення необхідного рівню захищеності.

2.3 Атаки на блокчейн

2.3.1 Атаки на криптографію

Атакується використаний криптографічні алгоритми. У разі знаходження критичної вразливості в криптографії на основі еліптичних кривих, руйнується вся ідея з використанням асиметричного шифрування та закритими-відкритими ключами, що використовуються сьогодні. Зловмисник може починати витрачати кошти інших користувачів, та зламувати усю систему загалом. Проте, поки що, жодної уразливості знайдено не було. Біткоин, і всі інші криптовалюти використовують нові та актуальні криптографічні алгоритми і сумнівно, що в найближчому майбутньому будуть знайдено якісь критичні уразливості.

2.3.2 DoS-атака на блокчейн

Мета атаки у цьому випадку відрізняється від звичайної DoS-атаки. Зловмисник намагається завалити ноди валідними операціями, які, наприклад, становлять великі затрати при обробці. Біткоин дозволяє створювати складні конструкції, складні умови витрат, скрипти, розумні контракти, які можуть, дуже довго обробляється окремими нодами. Протидія цьому досить проста: обмежувати розмір кожної окремої транзакції, розмір скриптів в транзакції,

кількість складних криптографічних операцій. Також ми можемо зробити обмеження на кількість збережених транзакцій в пам’яті кожної окремої ноди, для того, щоб мінімізувати ризик виходу з роботи через відсутність вільної оперативної пам'яті. При цьому це не є правилом усього консенсусу, це є правилом кожної окремої ноди, налаштованим персонально.

2.3.3 Спам-атака

Атака, що дійсно використовувалося, і досі часто використовується. Метою цієї атаки є підвищення комісії в Мережі. Транзакції користувачів у результаті відбуваються з дуже великими затримками або їм доводиться платити дуже великі комісії. У разі, коли блоки в мережі є обмежені, припустимо, одним мегабайтом, блок вміщує 100 до 3000 транзакції. Майнер, як зрозуміло, обирає транзакції, за які платять найбільші комісії. Якщо користувач створює транзакцію з комісією розміром n, нападник створює транзакції з комісією >n, що призводить до того, що наша транзакція не додається до блоку і залишається чекати більш сприятливих умов. Засоби протидії цієї атаці дуже прості - користувач створює велику кількість транзакцій самому собі, проте така протидія все ж таки потребує також оплати комісії майнерам.

2.4 Атаки на протоколи консенсусу

Proof-of-Work

PoW базує довіру до вузлів згідно з виконаним завданням. Зазвичай завдання це обчислення складної математичної задачі, обчислення якої займає багато часу та ресурсів, а перевірка отриманого результату робиться досить швидко. Механізм PoW забезпечує здатність вузла мережі перевірити, що є вузол, який додає новий блок в блокчейн фактично виконав розрахунки. Даний процес є спробою знайти хеш заголовку блоку (частина блокчейну, яка містить посилання на попередній блок і підсумкове значення транзакцій). Це стримує атаки відмови у наданні послуг та інші порушення послуг, такі як спам у мережі, дозволяючи запитувачу послуг провести деякі дослідження, для яких зазвичай потрібен час обробки пристроєм. Ключовим аспектом цих схем є їх асиметрія:

для запитувача задача становить набагато більшу складність аніж на стороні контролера. Для забезпечення безпеки в мережах з таким алгоритмом консенсусу штучно обмежують пропускну здатність - швидкість додавання нових блоків у мережу. Наприклад, новий блок у мережі Bitcoin формується лише кожні 10 хвилин. Це потрібно для того, щоб уся мережа встигла оновити інформацію щодо стану блокчейну. Такий підхід хоч і працює, проте має суттєві недоліки.

2.4.1 Атака типу Перегони

У разі, якщо транзакція не має підтверджень, отримувач транзакції (наприклад магазин), може зазнавати так званої "атаки перегонів". Працює це наступним чином - дві транзакції створюються з одних і тих же фондів і потім надсилаються різним адресатам. У цьому випадку кошти отримають лише один із цих адресатів. Через те, що підтвердження цієї транзакції з’явилося першим в блокчейні, інші признаються невалідними.

Користувачі можуть вживати численних запобіжних заходів для зменшення цього типу нападів, але важливо розуміти, що будяка транзакція повинна бути підтверджена в блокчейні перед тим як вважати її здійсненою.

2.4.2 Атака Фінні

Суть в тому, що зловмисник контролює певний обсяг хешрейта мережі.

Припустимо, нападник відправив певну кількість монет зі своєї адреси A на свою іншу адресу B. Зловмисник спробує знайти блок з цією транзакцією і, як тільки знайде його, зробить нову транзакцію на ті ж монети. Якщо продавець не дочекається підтвердження транзакції і відвантажить товар, то зловмисник, при наявності достатніх потужностей, вибере блок з власною операцією і підтвердить її. А та, що була відправлена продавцеві, залишиться не валідною. Саме тому завжди необхідно чекати не менше 6 підтверджень транзакції майнерами.

2.4.3 Атака Vector76

Також відома як "атака з підтвердженням". Це поєднання двох вищезгаданих атак, що дає злочинцю можливість витратити кошти двічі просто з підтвердженням.

2.4.4 Brute force атака

Ця атака можлива, навіть якщо магазин чи послуга очікують декількох підтверджень транзакцій. Атака вимагає, щоб зловмисник володів відносно високопродуктивним обладнанням (хеш-частотою). Злочинець відправляє транзакцію в магазин, оплачуючи товар / послугу, і водночас продовжує шукати з'єднання в блокчейн (blockchain fork), який розпізнає цю транзакцію. Після певної кількості підтверджень магазин відправляє товар. Якщо злочинець у цей момент виявив більше n блоків, він розбиває виделку на блокчейні та повертає свої гроші, але якщо злочинцю не вдалося це зробити, напад може вважатись провалом, а кошти надсилаються в магазин, як це має бути.

Успіх цієї атаки залежить від швидкості (хеш-частоти) зловмисника та кількості підтверджень для магазину / послуги. Наприклад, якщо зловмисник володіє 10% розрахункової потужності мережі Bitcoin і магазин очікує 6 підтверджень на успішну транзакцію, ймовірність успіху такої атаки складе 0,1%.

2.4.5 Атака 51%

Атака, при якій в розпорядженні зловмисника повинні знаходитися більш ніж половина потужності всієї мережі загалом, так званий «контрольний пакет» потужностей блокчейн мережі. Криптовалюти, що базуються на технології блокчейн та використовують PoW (наприклад Bitcoin) на ранніх етапах свого існування найбільш підвернені цій атаці. Використання розподіленого реєстру, транзакції якого підтверджують майнери видає грошову винагороду у разі підтвердження. Чим більше обчислювальної потужності зосереджено в руках майнера, тим більше шансів знайти правильне рішення першим, отримати право створити новий блок і отримати відповідну винагороду. В такому разі навіть якщо майнер не намагається провести ці атаку, він зацікавлений в зосередженні ресурсів, що може потенційно впливати негативно на всю систему загалом. В гіршому випадку якщо в руках майнера або пулу (група майнерів, що об'єдналася) виявиться 51 % обчислювальної потужності всієї системи, то саме

цей учасник гарантовано зможе контролювати всі операції по системі, генерувати блоки, підтверджувати або блокувати транзакції, що суперечить основній ідеї криптовалют — децентралізації. Володіння 51 % потужності всієї мережі не є атакою, поки учасник діє відповідно до правил і не перешкоджає природній роботі системи. Хоч це і несе шкоду іншим майнерам і робить видобуток невигідним, але якщо транзакції підтверджуються вірно, то користувач не завдає шкоди системі. Атака починається там, де учасник використовує свою перевагу для нечесного видобутку. Ця атака призводить до створення альтернативного ланцюгу блоків, саботування підтвердження блоків інших майнерів та зупинення добутку монет на “дійсному” ланцюгу. В мережі Bitcoin затвердження транзакції вимагає сформувати 6 наступних блоків (це займає годину, бо кожен блок формується приблизно 10 хвилин), відповідно зловмисник повинен сформувати 6 наступних блоків у своєму прихованому ланцюгу. Далі зловмисник будь-яким способом витрачає або переводить свої активи та відкриває свій прихований ланцюг блоків. В такому разі формально обидва ланцюги є підтвердженими та валідним, що призводить до конфлікту в системі. Цю атаку називають атакою подвійної трати.

В роботі професора Індійського Інституту Технологій Саравананa Вижаякумарана присвяченій безпечності протоколів, використаних у Блокчейні він розглядає атаку 51% на Біткоїн та приходить до висновку, що атака не є рентабельною для зловмисника. За розрахунками активи, необхідні для реалізації атаки набагато більші за потенційну отриману винагороду.

Proof-of-stake

Ідея полягає у використанні «частки» (stake) як ресурсу, який визначає, яка саме нода отримує право видобутку наступного блоку. У підході Proof-of-Stake Ноди також намагаються гешувати дані в пошуках результату менше певного значення, але складність в цьому випадку розподіляється пропорційно і відповідно до балансу даного вузла. Іншими словами - відповідно до кількості монет (токенів) на рахунку користувача. Таким чином, більше шансів

згенерувати наступний блок має вузол з великим балансом. Схема виглядає досить привабливо насамперед через невеликі вимоги до обчислювальних ресурсів, а також тому, що не стоїть питання «витрачені даремно» потужностей.

Як правило, на користь використання PoS наводяться наступні аргументи: для проведення атаки потрібні значні кошти, що робить її недоцільною з фінансової точки зору, одночасно з цим, якщо в розпорядженні нападника є велика кількість токенів, він сам постраждає від атаки, оскільки це порушить стійкість криптовалюти. Проте є також деякі недоліки. PoS дає додаткову мотивацію до накопичення коштів в одних руках, що може негативно позначитися на децентралізації мережі. Якщо утворюється невелика група, яка збере у себе чималі кошти, вона зможе нав'язувати свої правила роботи мережі іншим учасникам.

2.4.6 Атака Nothing-at-Stake (порожній стек)

Нападник може спробувати зробити форк блокчейну, створити довший альтернативний ланцюжок за допомогою витрачання «уявних» ресурсів. Більш того, його можуть підтримати інші майнери, оскільки вони також не витрачають «справжні» ресурси. За допомогою форку нападник може відхиляти певні транзакції і здійснювати атаку «подвійної витрати».

2.4.7 Інші варіанти протоколів консенсусу

Proof of Activity (доказ активності) - стандартна гібридна схема, що поєднує PoW і PoS;

Delegated Proof of Stake (делеговане підтвердження частки) - загальний термін, що описує еволюцію базових консенсус протоколів на основі підтвердження частки. DPoS використовується в BitShares, EOS і Tezos;

Proof of Burn (доказ спалювання) - «спалювання» відбувається шляхом посилання монет на таку адресу, з якого гарантовано не можна їх витратити. Позбавляючись таким чином від своїх монет, користувач отримує

право на довічний Майнінг, який також влаштований як лотерея серед всіх власників спалених монет;

Proof of Capacity (доказ ресурсів) - реалізація популярної ідеї «мегабайти як ресурси». Необхідно виділити суттєвий обсяг дискового простору, щоб включитися в Майнінг;

Proof of Storage (доказ зберігання) - схожа на попередню концепція, при якій виділене місце використовується всіма учасниками як спільне хмарне сховище.

Кожна блокчейн мережа обирає протокол консенсусу, зважаючи уваги на свою специфіку. Крім того, немає жодного обмеження на створення нових протоколів консенсусу. Головне, щоб потенційні користувачі мали змогу упевнитися в безпечності протоколу.

Для створення блокчейну, що буде використано для інтернет-голосування, на жаль, жодний з представлених варіантів не підходить. Proof-of-work потребує великих обчислювальних потужностей, що для учасників голосування може бути проблемою. Так само Proof of Capacity, що спирається на обладнання користувачів мережі теж не є ефективним. Через те, що розподіл активів мережею буде рівномірний proof-of-stake не буде працювати. Proof of Burn – так само як і попередній варіант не може бути реалізованим.

РОЗДІЛ 3. ПРОЄКТУВАННЯ СИСТЕМИ ЕЛЕКТРОННОГО ГОЛОСУВАННЯ

3.1 Чинні системи електронного голосування

Розпочнемо проєктування з аналізу чинних систем електронного голосування. У світі розрізняють такі види електронного голосування:

1) E-voting (electronic voting) – це звичайні вибори, на певному етапі яких використовують електронне обладнання. Це можуть бути як електронні урни для голосів, що автоматично підраховують голоси, так і термінали, які реалізують роботу з електронними бюлетенями. Більшість країн використовує саме такий варіант голосування. Наприклад, у США починаючи з 2002 року, коли був прийнятий Закон «Допоможи Америці голосувати» був сформований спеціалізований фонд та розроблена комп`ютерна система голосування. Причиною цих змін стали неточності та розбіжності, що мали місце під час Президентських виборів у 2000. В США навіть на території одного і того ж штату може бути декілька різних систем голосування, проте найпоширенішими і досі є вибори з використанням електронного підрахунку голосів.

Інший варіант голосування – використання «електронних урн» має приклад у Бразилії, де таке голосування вводилось в декілька етапів, що дозволило декілька разів перевірити його надійність. У 1996 урни були використані на етапі муніципальних виборах лише в одному регіоні. У 2000 «електронними урнами» користувались при муніципальних виборах по всій території країни. В 2002 представники країн США, Мексики, Японії, Венесуели пильно слідкували за ходом перших електронних національних виборів президента. Жодного порушення не було виявлено[2].

Ці види голосування хоч і мають свої переваги, проте спадкує недоліки і навіть додає нових.

Фізичне голосування використовувалося століттями. За цей час майже усі методи шахрайства були апробовані та до кожного з них було знайдено протидію. Основною перевагою безпеки фізичних виборів є те, що атаки, як

правило, дуже погано масштабуються. Крім того для шахрайства у фізичних виборах потрібно мати досить велику кількість людей, і навіть однієї людини буде достатньо для того, щоб розкрити шахрайство.

Використання електронного підрахунку або урн призводить до того, що замість центрального органу влади, що веде підрахунок ми маємо довіряти пристроям, валідність ПЗ на яких перевірити дуже важко. Зазвичай один з варіантів підвищення довіри до програмного забезпечення це використання відкритого ПЗ. У такому разі будь-який експерт має змогу перевірити надійність коду та вказати на недоліки та потенційні слабкі місця. Проте на жодному з зазначених вище прикладах не було використане відкрите ПЗ. Інша проблема перевірити відповідність ПЗ встановленого на обладнанні до заявленого. Так, існують порівняння хеш сум, проте у такому разі відповідальність переноситься на довіру до програм валідаторів.

Далі, після проведення голосування, результати з кожної виборчої дільниці повинні бути передані у ЦВК. Це робиться декількома способами. Перший – фізичне переміщення урни або носія інформації до ЦВК. У такому разі валідність носія та інформації може бути скомпрометована. Саме такий спосіб було використано у Бразилії. Другий – передача даних через мережу інтернет. На жаль є приклади, коли ці данні передавалися по мережі без шифрування, завадостійкого кодування та будь-яких валідаторів. У такому разі майже будь-яка атака може скомпрометувати вибори.

На конференції DEFCON, що проводиться для спеціалістів з комп’ютерної безпеки, були змагання, метою яких був злам «електронних урн». Виявилось що урни можуть бути зламані менш ніж за хвилину, та на них може бути встановлене модифіковане ПЗ. Крім того, через те, що урни увесь час підтримають зв’язок один з одним існує можливість передачі шкідливого коду від одної урни усім іншим.

Сукупність цих факторів не дозволяє вважати голосування з електронним підрахунком та електронними урнами надійним.

2) I-voting (internet-voting) – голосування з використанням мережі інтернет.

Цей вид голосування є більш рідким та на повну використовується лише в Естонії. У кожного громадянина Естонії є електронна ID-картка, що підтримує стандарт X.509. Використання цього стандарту дозволяє використовувати цю картку для автентифікації громадянина у всіх державних та банківських структурах.

Перші вибори з можливістю голосування через інтернет були проведені у 2005 році. Лише 2% громадян скористалися голосуванням через інтернет. Це обумовлено складністю голосування. На той час голосування через інтернет складалося з декількох етапів. Спочатку потрібно було отримати код місцевої влади. Потім придбати пристрій, що підключався через USB порт до комп'ютера та сканував ID-картку. Лише після усіх цих махінації виборець мав змогу віддати свій голос. Голосування тривало цілодобово та відбувалося безпосередньо за 6 днів до дня виборів і закінчилося за 4 дні до дня виборів.

Зараз голосування в Естонії стало простішим. Необхідно завантажити на смартфон додаток, або відвідати необхідний сайт у мережі інтернет, там ввести данні ID-картки і одразу перейти до голосування. Ці спрощення позитивно вплинули на кількість виборців, що беруть участь у голосуванні саме через інтернет. Важливою перевагою інтернет-голосування в Естонії є той факт, що виборець може голосувати скільки завгодно разів. Враховується тільки останній голос. Чому це важливо? Через те, що голосування проходить дистанційно, виборча комісія не може гарантувати що на виборця не буде впливу. Крім того, можливість змінити голос скільки завгодно разів робить покупку голосів на виборах – неефективним способом шахрайства.

Електронне голосування в Естонії не тільки економить кошти і час на підрахунок, але і підвищило кількість учасників голосування. Важливо зрозуміти, що не зважаючи увагу на те, що Естонія продовжує модернізувати систему вона має певні недоліки. Наприклад, не гарантується безпека учасника

голосування, також немає можливості підтвердити що конкретний голос виборця був врахований.

3.2 Гомоморфне шифрування

Голосування у своїй сутності містить вимоги, що є майже протилежними. З одного боку - вимога анонімності, з іншого – вимога прозорості та довіри. На практиці це означає що з одного боку кожен учасник голосування повинен мати змогу перевірити чи було враховано його голос, а з іншого – не повинно бути способу дізнатися який саме голос віддав учасник голосування. Це зроблено з метою унеможливлення будь-якого впливу на учасників голосування через підкуп або погрози чи переслідування через голосування. Тобто з одного боку ми повинні мати перелік усіх голосів та можливість перевірки, з іншого – неможливість встановити який саме голос було віддано. Це виглядає як парадокс, вирішення якого є неможливим, проте криптографія, а саме гомоморфе шифрування може вдовольнити обидві ці потреби [3].

Звичні системи шифрування залежать від спільного використання ключа (відкритого або особистого) між вузлами, які беруть участь в обміні зашифрованим повідомленням. Однак, цей підхід створює проблеми конфіденційності. Користувачі або постачальники послуг з ключем мають ексклюзивні права на дані. Особливо у випадку популярних хмарних сервісів контроль над конфіденційністю конфіденційних даних втрачається. Навіть коли ключі не передаються, зашифрований матеріал передається третій стороні, якій необов'язково потрібен доступ до контенту. Більш того, ненадійні сервери, провайдери та хмарні оператори можуть продовжувати ідентифікувати елементи користувачів протягом довгого часу після того, як користувачі припиняють відносини зі службами [8]. Гомоморфне шифрування - особливий тип схем шифрування, може розв'язувати ці проблеми, оскільки він дозволяє будь-якій третій стороні працювати з зашифрованими даними, не розшифровуючи їх заздалегідь. Схема повністю гомоморфного шифрування ПГШ (або FHE - Fully Homomrphic Encryption), яка дозволяє виконувати будь-яку обчислювальну

функцію для зашифрованих даних, була представлена Крейгом Джентрі у 2009 році [6]. Попри те, що це було проривним досягненням, різні реалізації досі демонстрували, що ПГШ все ще необхідно значно поліпшити, щоб мати змогу ефективно функціонувати практичним на кожній платформі.

Для реалізації у системі електронного голосування, зважаючи на вище зазначені факти розумним буде використання ЧГШ – частково гомоморфного шифрування.

Розглянемо гомоморфне шифрування детальніше.

Нехай k – ключ шифрування, m – деяке повідомлення що необхідно зашифрувати, тоді E(k, m) – функція шифрування. Функція E називається гомоморфной щодо операції «*» (додавання або множення) над відкритими повідомленнями m_1 і m_2, якщо існує ефективний алгоритм F який, отримавши на вхід будь-яку пару шифрованих повідомлень виду $E(k, m_1)$і $E(k, m_2)$ повертає шифрований текст $C = F(E(k, m_1), E(k, m_2))$ який при розшифруванні буде співпадати з результатом операції «*», виконаної над відкритим текстом. Тобто $D(F(E(k, m_1), E(k, m_2))) = m_1 * m_2$, де функція D є функцією розшифрування[9].

Будь-яка стандартна система шифрування характеризується трьома операціями: операцією генерація ключів (key generation), операцією шифрування (encypt) і операцією розшифрування (decrypt).

Для опису гомоморфной системи шифрування крім трьох перерахованих вище операцій потрібно описати операцію обчислень (eval). Гомоморфне шифрування використовує послідовності з чотирьох операцій: генерації ключів, шифрування, обчислення, розшифрування:

1. Генерація ключів. Клієнтом генерується пара ключів – відкритий та особистий p, q для розшифрування та шифрування відкритого тексту.
2. Шифрування. Клієнт шифрує відкритий текст – повідомлення m, використовуючи секретний ключ q. Далі, шифроване повідомлення C = E(q, m) та відкритий ключ p передаються на сервер.

3. Обчислення. Сервер отримує функцію F, використовує F та p для обчислення результату операції над шифрованим текстом, далі сервер відправляє результат клієнту;
4. Розшифрування. Клієнт розшифровує отримане від сервера значення, використовуючи q.

Нехай E - функція шифрування; D - функція розшифрування; t_1 і t_2 - відкриті тексти; символи «⊗» і «⊕» позначають операції множення і додавання над шифрованими текстами, відповідні операціям множення і складання над відкритими текстами [10].

Система шифрування вважається гомоморфною відносно операції множення, якщо $D(E(t_1) \otimes E(t_2)) = t_1 \times t_2$.

Система шифрування вважається гомоморфною відносно операції додавання, якщо $D(E(t_1) \oplus E(t_2)) = t_1 + t_2$.

Якщо система є гомоморфною відносно як операції множення так і додавання, таку систему вважають повністю гомоморфною.

Якщо криптосистема з такими властивостями зможе зашифрувати два біта, то, оскільки операції додавання і множення формують над бітами повний по Тьюрингу базис, стає можливим обчислити будь-яку булеву функцію, а отже, і будь-яку іншу обчислювальну функцію.

Перейдемо до конкретних прикладів систем частково гомоморфного шифрування.

RSA

Одним з перших прикладів систем ЧГШ є RSA, запропонована у 1978 році Рівестом, Шаміром та Адлеманом. Безпечність криптосистеми ґрунтується на складності факторингу добутку двох великих простих чисел.

1. Генерація ключів. Для двох великих простих чисел p, q обчислюємо $n = p*q$ та $\varphi = (p-1)*(q-1)$. Далі обираємо e взаємно просте з φ. Обчислюємо число d таке, що $e*d \equiv 1 \bmod(\varphi)$. Причому e - відкрита експонента, d - закрита експонента, n - модуль.

Відкритий ключ – (n, e)

Закритий ключ – (n, d).

2. Шифрування. Повідомлення M перетворюється таким чином, щоб $1<m<n$. Далі проводиться шифрування по формулі $c = m^e \bmod n$.
3. Розшифрування. Відновлення повідомлення з шифротексту відбувається з використанням пари закритих ключів.

   $m = c^d \bmod n$.

Доказ гомоморфності.

$$E(m_1) * E(m_2) = (m_1^e \bmod n) * (m_2^e \bmod n) = (m_1 * m_2)^e \; mod \; n = E(m_1 * m_2)$$

Як бачимо система гомоморфна лише щодо операції множення.

Криптосистема Гольдвассера – Мікалі.

Ця система ЧГШ була винайдена у 1982. Система базується на проблемі квадратичних залишків[7].

1. Генерація ключів. Для двох великих простих чисел p, q обчислюємо

   $n = p*q$. У – випадкове ціле число, що вдовольняє вимогу

   $(y/p) = (y/q) = -1$ (символи Якобі).

   Відкритий ключ – (n, y)

   Закритий ключ – (p, q).
2. Шифрування. По-перше, повідомлення (m) перетворюється в рядок біт. Потім для кожного біта повідомлення, що утворюється взаємно проста з n квадратична не резидуальна величина. Потім кожен біт шифрується наступним чином:

   $C_i = E(m_i) = y^2_i x^{mi} \bmod n$.
3. Розшифрування. Для розшифрування кожного з бітів шифротексту потрібно перевіряти чи являється біт модулем квадратичного залишку n, чи ні. Якщо являється - повертаємо 0, інакше - 1.

Доказ гомоморфності.

$E(m_1)*E(m_2) = (y^2_1 x^{m1} \bmod n)*(y^2_2 x^{m2} \bmod n) = (y_1 * y_2)^2 x^{m1+m2} \bmod n = E(m_1+m_2)$.

Криптосистема ГМ є гомоморфною відносно операції додавання і обчислюється побітно[12].

Криптосистема Ель – Гамаля.

Криптосистема була винайдена у 1985. Являє собою покращену версію алгоритму Діффі-Галмана.

1. Генерація ключів. Генерується випадкове просте число p. Обирається примітивний елемент g поля Z_p. Далі обираємо випадкове ціле число x в діапазоні від 1 до p-1. Обчислюємо y за формулою $y = g^x \bmod p$.

   Відкритий ключ – (p, g, y)

   Закритий ключ – x.
2. Шифрування. Спочатку обираємо сесійний ключ k у проміжку від 1 до p-1. Далі обчислюємо $a = g^k \bmod p$ та $b = y^k M \bmod p$. Де M – повідомлення. Пара (a, b) – шифротекст.
3. Розшифрування.

   $M = b(a^x)^{-1} \bmod p$, або

   $M = b*a^{(p-1-x)}$.

Доказ гомоморфності.

$E(m_1) * E(m_2) = (g^{x_1}, m_1 h^{x_1}) * (g^{x_2}, m_2 h^{x_2}) = (g^{x_1+x_2}, m_1 * m_2 h^{x_1+x_2}) = E(m_1 * m_2$

Як бачемо, система гомоморфна відносно операції множення.

Криптосистема Бенало.

По своїй суті ця система є подальшим розвитком ідей криптосистеми Гольдвассера - Мікалі. Замість побітного шифрування тут використовується шифрування цілим блоком.

1. Генерація ключів. Обираємо розмір блоку r та два великих простих числа p, q, що відповідають потребам

   r, (p-1)/r є взаємно простими.

   r, q-1 r є взаємно простими.

   Обчислюємо n = p*q, φ = (p-1)*(q-1).

   Обираємо $y \in Z_n^*$, $y^{\varphi/r} \neq 1 \bmod n$.

$x = y^{\varphi/r}$

Відкритий ключ – (y, r, n)

Закритий ключ – (φ, x).

2. Шифрування. Для початку необхідно модифікувати повідомлення, щоб воно було менше розміру блоку. Далі обираємо довільне U є Z_n^* .

   $E(m) = y^m u^r \bmod n$.

3. Розшифрування.

   $a = c^{\varphi/r} \bmod n$

   $m = \log_x(a)$, або $x^m \equiv a \bmod n$.

Доказ гомоморфності

$$E(m_1) * E(m_2) = (y^{m_1} u_1^r \bmod n) * (y^{m_2} u_2^r \bmod n) = ym_1 + m_2(u_1 * u_2)r \bmod n$$
$$= E(m_1 + m_2 \bmod n)$$

Стійкість криптосистеми Бенало базується на складності обчислення вичетів високих ступенів. ЧГШ Бенало є гомоморфним відносно операції додавання.

Кожна з зазначених криптосистем може бути використана з максимальною перевагою для певних задач. У нашому випадку криптосистема Бенало є найбільш придатною для використання. Це забезпечується гомоморфізмом по операції додавання, бо результати електронного голосування повинні бути складені. Крім цього той факт, що шифрування проходить над усім повідомленням одразу забезпечує меншу кількість виконаних операції.

Надалі з розвитком досліджень та вдосконаленням існуючих криптосистем може бути знайдена більш ефективна криптосистема, що могла б бути повністю гомоморфною, проте саме для задач поставлених в сфері електронних виборів критичнішим є швидкість обробки та надійність системи а не повнота операції за Тьюрінгом.

3.3 Система електронного голосування з використанням технології блокчейн

Система електронного голосування на основі блокчейну, що може бути використана для реалізації волевиявлення на рівні держави повинна відповідати високим стандартам та дотримуватися усіх норм.

Зрозуміло, що така інновація як електронні вибори повинна впроваджуватися поступово та в декілька кроків. На початкових етапах повністю припиняти звичайні вибори – погана ідея. Навіть попередні апробовування на менших масштабах можуть неповністю виявити усі недоліки. Крім того, навіть у разі підтвердження безпечності та правильності функціонування усієї системи немає гарантій що певні категорії громадян зможуть або забажають брати участь у голосуванні.

Зважаючи ці фактори електронне голосування спочатку повинно бути реалізовано як допоміжний спосіб голосування. Та проведене до дати фізичного голосування для уважнішого слідкування та аудиту.

Процес реєстрації для участі у виборах у розробленій системі повинен бути наступним.

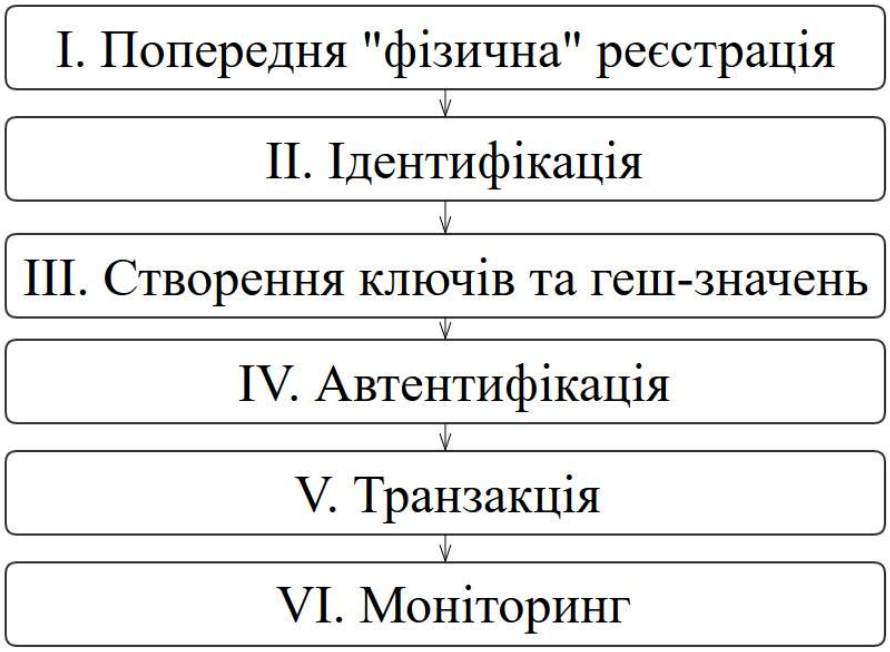

Рисунок 3.1 - Етапи електронного голосування

1. Попередня фізична реєстрація.

На цьому етапі виборець, що бажає проголосувати через інтернет повинен прийти до відділу попередньої реєстрації з необхідними документами: паспорт, Id-картку, чи будь-який інший документ ідентифікації, що має відповідний захист. Ці відділи можуть бути розташовані у відділах банку, відділах пошти, тощо. Відкриття відділів попередньої реєстрації відбувається за тиждень до дня проведення виборів.

2. Ідентифікація. Учасник голосування проходить перевірки, чи має він право голосувати. У разі відповідності усім критеріям вважається що учасник голосування буде голосувати онлайн, а тому у списках реєстрації на фізичні вибори ставиться відповідна помітка для того, щоб голос виборця був зарахований лише один раз.

3. Створення ключів. Пройшовши ідентифікацію, учасник голосування отримує два значення – свій номер у системі та додатковий ідентифікатор. Ці данні є хеш-значеннями. Надалі у системі голосуючий буде представлений саме цією парою значень, без прив'язки до ім'я чи інших демаскуючих факторів.

4. Автентифікація. Автентифікація учасника голосування буде проведена безпосередньо перед голосуванням. Онлайн голосування розпочнеться за 3 дні до початку фізичних виборів та скінчиться за добу до їх початку.

5. Транзакція. У цьому випадку транзакцією називають передачу голосу відповідному кандидату. Аналогом цієї операції в звичайному голосуванні є передача бюлетеню до урни.

6. Моніторинг. Цей етап необхідний для фінального підрахунку та перевірки прозорості виборів. ЦВК – центральна виборча комісія на цьому етапі перевіряє усі транзакції та підбиває підсумки.

Далі ЦВК переходить до проведення фізичного голосування.

Після проведення обох раундів виборів ЦВК повинна ретельно перевірити проходження обох та визначити, чи можна вважати вибори валідними. Якщо усе пройшло як треба, ЦВК оголошує результати. У разі виникнення проблем або

правопорушення буде об’явлено про анулювання виборів та буде проведено ще один раунд для остаточного результату.

Наступним кроком розглянемо структуру наших електронних виборів.

Розроблено два варіанти виконання внутрішньої структури системи електронного голосування, кожна з яких має свою переваги.

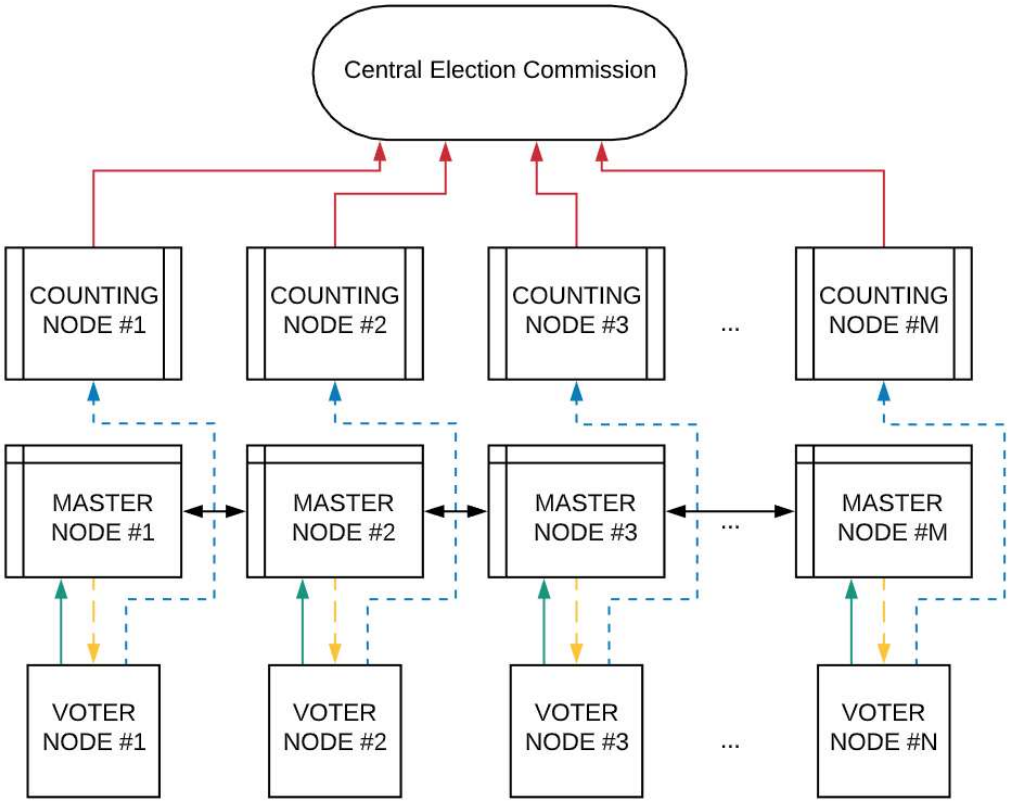

Рисунок 3.2 – Внутрішня структура системи електронного голосування (а)

В системі кожен з учасників голосування для автентифікації повинен ввести парольну пару, яку він отримав на етапі фізичної реєстрації. На цьому етапі учасник голосування стає частиною блокчейн мережі. На рисунку він буде позначений як нода виборця #i – Voter node #i.

Наступним етапом він робить запит до майстер ноди - Master node на отримання бюлетеню. Ці ноди по суті є серверами. Майстер нода комбінує у собі дві різні функції - з одного боку це захищені бази даних, а з іншого це контролюючі вузли, які вирішують чи необхідно видати ноді голосуючого бюлетень для голосування. Майстер ноди – це частина мережі, що була створена заздалегідь і налаштована для протидії атакам. Вона стійка до DDoS атак, що

найбільш імовірні на ці типи вузлів. Кількість майстер нод – фіксована та визначається зважаючи на кількість зареєстрованих користувачів, процесорними можливостями обладнання. Для моделювання пропоную вважати їх кількість відповідною кількості областей в Україні - 24. Таким чином голосуючий з певної області буде закріплений за певним майтсер нодом.

У другому варіанті усі роботи по підрахунку виконуються централізовано.

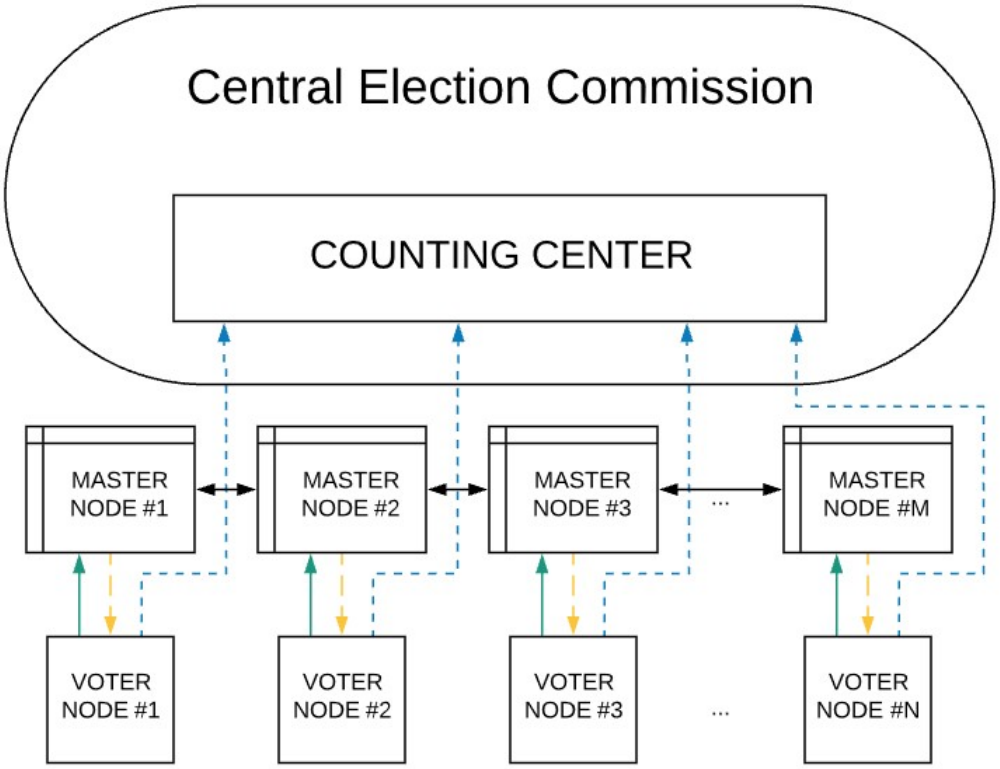

Рисунок 3.3 – Внутрішня структура системи електронного голосування (б)

Порядок проведення виборів між цими варіантами не відрізняється. Проте деякі розбіжності все ж присутні. Так наприклад у першому варіанті процесорні потужності розділені по мережі біль рівномірно, ніж у другому варіанті, що може позитивно сказатися на вартості проведення виборів та на необхідному обладнанні. У другому варіанті все обробляється централізовано, це значить що концентрація потужностей мережі буде саме у ЦВК.

У другому варіанті ми маємо меншу кількість транзакцій в мережі, що позитивно впливає на захищеність усієї системи, проте потребує від ЦВК серверу, що може приймати купу мережевих запитів одночасно.

Бази даних, що зберігає майстер нода складається з 3 полів.

ІДЕНТИФІКАТОР МАЙСТЕР НОДИ	ОСОБИСТИЙ ІДЕНТИФІКАТОР	СТАН ГОЛОСУВАННЯ
3FD83361D45EC5CE2EA5	4D10C03C293B4DA3C5D0EAEF092F5A8C	0
5C52A078AE3F4560090B	819BBC8C71F211F389618AF2A6EBF58D	0
	...	
A0FCBD5113B4358621B5	A9CA26BC3A64C487D8BE6FC3C6E8B399	0

Рисунок 3.4 – Внутрішня структура бази даних майстер ноди

Перше поле – це ідентифікатор майстер ноди. Він потрібен для того щоб відправити бюлетень голосуючому з відповідної майтсер ноди.

Друге поле – особистий ідентифікатор. Цей рядок необхідна для ідентифікації голосуючого та збереження анонімності. Крім того вона допомагає реалізувати функцію аудиту.

Анонімність користувача зберігається завдяки тому, що парольна пара являє собою два хеши. Унікальність пари забезпечується стійкістю хеш-функцій та низькім показником колізій.

Трете поле – поле поточного статусу голосуючого. Це поле необхідно для моніторингу голосу користувача. 0 – не проголосував, 1 – проголосував.

Коли нода учасника голосування робить запит до майстер ноди, майстер нода обробляє запит та перевіряє чи є користувач у її базі даних. Теж саме роблять і інші майстер ноди.

Дана блокчейн мережа, зважаючи на особливості та потреби, не може використовувати жодний з популярних протоколів консенсусу, тому у цій мережі розроблено свій. Для отримання консенсусу кожна з майстер нод аналізує запит та видає своє рішення усім іншим. У разі якщо хоча – б в одній ноді немає запису про учасника голосування, або стоїть помітка про те, що він вже

проголосував, система не видає учаснику голосування токен, тут - аналог бюлетеню.

Бюлетень являє по суті одномірний вектор, розмір якого відповідає кількості кандидатів, що беруть участь у виборах. Він використовує гомоморфне шифрування та реалізує необхідні фактори захисту.

Далі, коли учасники голосування отримує права на голосування та бюлетень, він віддає свій голос за свого кандидата. Далі використовуючи гомоморфне шифрування шифрується та відправляється на адресу ноди підрахунку. Counting node – нода підрахунку, як і майстер нода прикріплена за своєю областю. Ці ноди реалізують підрахунок усіх голосів. Нода підрахунку не може вивести токени та використовується аналогічно процедурі спалювання токенів.

Підрахунки шифруються та відправляються на ноду ЦВК. Саме тут додаються результати з усіх нод підрахунку та реалізується аудит усієї мережі. Ця нода не має великих обчислювальних потужностей та не містить великих об’ємів пам’яті. Її задача бути захищеною та надійною.

Історія транзакцій у вигляді переліку особистих ідентифікаторів, голоси яких були враховано публікується для можливості перевірки враховування.

Підрахунки розшифровуються та публікуються на фінальному етапі голосування.

3.4 Прозорість голосування

Основною проблемою чинних систем голосування є нездатність вдовольняти одразу і необхідність анонімності голосів і прозорість усієї системи. На жаль, навіть системи електронного голосування не передбачають здатність перевірки правильності голосу.

Це погано з тієї точки зору що потенційне підтасування все ще може мати місце. Приведемо приклад. Учасник голосування Боб голосує в електронній системі голосування за кандидата №1. Коли голосування закінчилося Боб хоче перевірити чи дійсно його голос було враховано в результатах. Все що він бачить

– наявність відповідного запису з його ідентифікаційним номером у системі, при цьому не існує можливості перевірити за кого саме проголосував Боб.

Це є перевагою лише з точки зору непримусовості голосування. Тобто не маючи змогу перевірити кому саме було віддано голос Боб не може бути купленим як виборець, або не може бути під впливом[11].

Такий механізм захисту зменшує вірогідність шахрайства зі сторони учасників голосування, проте не захищає їх від навмисних або ненавмисних махінацій зі сторони системи.

На сьогодні жодного універсального вирішення цієї задачі немає.

Серед запропонованих варіантів найбільш прийнятним є можливість перевірки системи безпосередньо перед посиланням голосу.

Учасник голосування повинен виконати ті ж самі операції, що він робить для голосування, проте обрати опцію перевірки системи замість відправки голосу. У такому разі його голос записується у копію дійсної бюлетені та обробляється безпосередньо на пристрої.

Наведемо приклад. Боб має за мету перевірити систему електронного голосування. Для цього він віддає свій голос за кандидата під номером 3. Далі ця копія бюлетеню проходить гомоморфне шифрування на пристрої та одразу розшифровується, таким чином на пристрої імітується повний цикл проходження голосування. Звіряючи отриманий результат з голосом що віддав Боб можна вирішити чи є система надійною. Перевірка може бути проведена скільки завгодно разів. При цьому кожна перевірка використовує не дійсні бюлетені та не передає голос до рахувального центру. Перевірки також не навантажують систему, бо використовують процесорні потужності обладнання учасника голосування.

Це рішення не є ідеальним бо ми делегуємо проблему достовірності. У разі вище означеної перевірки учасник голосування повинен довіряти системі, що алгоритм що використовується для перевірки системи та безпосередньо для голосування це один і той самий алгоритм.

Також необхідно відзначити що інтернет-ресурс що відповідає за клієнтську частину голосування повинен мати сертифікати відповідності та систему захисту від фішингових атак. Крім того необхідно затвердити механізм перевірки валідності використовного ПЗ при голосуванні. Бо у разі підміни алгоритму уся система електронного голосування втратить сенс.

На жаль на сьогодні проблема прозорості голосування не може бути повністю вирішена.

ВИСНОВКИ

1) В ході дослідження було були розглянуті різні системи електронного голосування. Порівнявши переваги різних варіантів було обрано найперспективніший варіант, що використовує інтернет-голосування та технологію блокчейн. Цей варіант є найбільш надійним, прозорим та швидким. Крім того електронне голосування є дешевшою альтернативою та потребує гроші лише на розробку та підтримку системи.

2) Були розглянуті усі типи атак на блокчейн, приведені приклади та особливу увагу звернено на найбільш імовірні для системи електронного голосування атаки. Цей аналіз вплинув на розробку структури блокчейну та виявив слабкі місця в аналогічних системах.

3) Були проаналізовані інші чинні системи електронного голосування, що використовуються по всьому світі. Також увагу приділено проблемам там недолікам цих систем. Аналіз цих систем допоміг у розробці власної системи голосування. Таким чином спираючись на їх досвід розробка системи базувалась на вирішенні цих проблем.

4) На основі проаналізованої інформації запропоновано два варіанти виконання структури електронного голосування, кожна з яких має свої переваги та може бути застосована на практиці. Також розроблено порядок проведення голосування, що містить 6 пунктів. Серед технологій що підвищать безпечність та прозорість інтернет-голосування запропоновано гомоморфне шифрування. Наведено найбільш популярні схеми частково гомоморфного шифрування. Порівнявши їх характеристики обрано гомоморфну систему Бенало. Вона є гомомомрфною відносно операції додавання, що є важливим для ефективного підрахунку голосів та, крім того, вона здатна шифрувати усе повідомлення одразу, не поділяючи його на біти. Що позитивно впливає на швидкість обробки інформації.

5) Було розроблено програмне забезпечення з графічним інтерфейсом, що реалізує ЧГШ систему Бенало. Це ПЗ показує яким чином голос буде

оброблятися у мережі та реалізує шифрування та підрахування голосів одразу декількох учасників голосування. Це ПЗ може стати основою, яка буде застосовуватися у системі електронного голосування з використанням блокчейн технологій. Програмна реалізація виконана но мові програмування Python, інтерфейс виконано з використанням PyQT.

6) Розроблена система, схеми та методи являють собою комплекс заходів, направлених на реалізацію інтернет-голосування з дотриманням закону України відносно проведення голосування. У роботі відзначена основна проблема усіх систем електронного голосування та приведено одне з розв'язань цієї проблеми. У майбутньому з розвитком цієї області криптографії та появою нових методів шифрування проблема правдивості та прозорості голосування може бути вирішена ефективніше, проте враховуючи сьогоднішні реалії саме наведений спосіб є найбільш надійним.

7) Результати досліджень та розробки доповідалися на НТК. Повний текст доповіді наведено у додатку В.

ПЕРЕЛІК ПОСИЛАНЬ

[1] Готун А. Використання нових інформаційних технологій у виборчому процесі: світовий досвід і практика застосування в Україні, Вісник Київського НУ ім. Шевченка. Філософія і політологія. – 2008. – С. 89-90.

[2] Оніпко О. Шляхи вдосконалення технології виборчого процесу, Вісн. Центр. вибор. коміс. – 2006. – Вип. № 4 (6). – С. 67–70. – Access mode: http://www.cvk.gov.ua/visnyk/pdf/2006_4/visnyk_st_22.pdf.

[3] Homomorphic encryption, Wikipedia – Electronic resource. – Access mode: https://en.wikipedia.org/wiki/Homomorphic_encryption.

[4] Saravanan Vijayakumaran An Introduction to Bitcoin Department of Electrical EngineeringIndian Institute of Technology Bombay - 2017 - С. 59 – 62. Режим доступу: https://www.ee.iitb.ac.in/~sarva/bitcoin/bitcoin-notes-v0.1.pdf

[5] Saad, M.; Spaulding, J.; Njilla, L.; Kamhoua, C.; Shetty, S.; Nyang, D.; Mohaisen, A. Exploring the attack surface of blockchain: A systematic overview. Режим доступу: https://arxiv.org/abs/1904.03487

[6] Полностью гомоморфное шифрование – Electronic resource. – Access mode: http://cryptowiki.net/index.php?title=%D0%9F%D0%BE%D0%BB%D0%BD%D0%BE%D1%81%D1%82%D1%8C%D1%8E_%D0%B3%D0%BE%D0%BC%D0%BE%D0%BC%D0%BE%D1%80%D1%84%D0%BD%D0%BE%D0%B5_%D1%88%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5

[7] Безопасное вычисления с использованием гомоморной криптографии для облачных хранилищ данных. – Electronic resource. – Access mode: https://cyberleninka.ru/article/n/bezopasnye-vychisleniya-s-ispolzovaniem-gomomorfnoy-kriptografii-dlya-oblachnyh-hranilisch-dannyh/viewer

[8] Fully homomorphic encryption: Searching over encrypted cloud data – Electronic resource. – Access mode: https://dl.acm.org/doi/abs/10.1145/3090354.3090364

[9] Practical applications of homomorphic encryption – Electronic resource. – Access mode: https://dl.acm.org/doi/abs/10.1145/2381913.2381924

[10] Can homomorphic encryption be practical? – Electronic resource. – Access mode: https://dl.acm.org/doi/abs/10.1145/2046660.2046682

[11] Blockchain Voting and its effects on Election Transparency and Voter Confidence – Electronic resource. – Access mode: https://dl.acm.org/doi/abs/10.1145/3085228.3085263

[12] A Survey on Homomorphic Encryption Schemes: Theory and Implementation – Electronic resource. – Access mode: https://dl.acm.org/doi/abs/10.1145/3214303

ДОДАТОК А

РЕАЛІЗАЦІЯ СИСТЕМИ ГОМОМОРФНОГО ШИФРУВАННЯ БЕНАЛО

Найбільше переваг при використанні у системі інтернет-голосування надає саме Криптосистема Бенало. Вона реалізує повною мірі гомоморфізм відносно операції додавання та здатна обробляти все повідомлення за один раз.

Перейдемо до реалізації.

Спочатку необхідно знайти просте число r.

```
def get_r(min_r, max_r):
    seed = random.randint(min_r,max_r)
    while True:
        if seed % 2 == 1:
            if is_prime(seed):
                return seed
        seed += 1
```

Як бачимо для цього ми вводимо бажаний діапазон для числа та обираємо його випадково. У разі, якщо в діапазоні не було знайдено простого числа, ми ітеративно знайдемо найближче просте, що буде більше за максимальне значення інтервалу. Це зроблено для більш стабільної роботи програми.

У функції знаходження r є виклик функції перевірки на простоту.

```
def is_prime(val):
    pr = True
    for i in range(2, val // 2 + 1):
        if val % i == 0:
            pr = False
    return pr
```

Результатом виконання функції є булева змінна яка вказує на належність число до множини простих чисел.

Далі ми генеруємо просте число p. Через те, що воно повинне відповідати запиту що r, (p-1)/r є взаємно простими ми будемо генерувати їх одразу такими, щоб вони відповідали запиту. Простоту перевіримо наступною операцією.

```
def get_p(r):
    p = r
    for i in range (1, r-1):
        if is_prime(p*i+1):
            return p*i+1
```

Далі генеруємо q, також починаючи з кінця. Це допомагає швидше знайти необхідні числа.

```
def get_q(r):
    seed = random.randint(r, r * 10)
    while True:
        if seed % 2 == 1:
            if is_prime(seed):
                if seed - 1 % r != 0:
                    return seed
        seed += 1
```

Згенерувавши пару p та q, обчислюємо $n = p * q, \varphi = (p-1) * (q-1)$

```
def get_n(p,q):
    return p*q
def get_fi(p,q):
    return (p-1)*(q-1)
```

Далі необхідно обрати таке значення y що, $y \in Z_n^*$ та $y^{\varphi/r} \neq 1 \bmod n$.

```
def get_y(r, n, fi):
    ys = [i for i in range(2, r-1)]
    while len(ys) >=1:
        i = random.randint(0, len(ys)-1)
        val = ys[i]
        ys.pop(i)
        ost1 = power_mod(val, int(fi/r), n)
        print(" y = {}, ost = {}".format(i, ost1))
        if ost1 != 1 :
            return i
```

У цій функції е виклик функції power_mod. Ця функція відповідає за обчислення результату піднесення числа до заданого ступеня по певному модулю. Для економії процесорних потужностей на кожному ітеративному кроці ми обраховуємо модуль ступеню, що спрощує подальше обчислення.

```
def power_mod(val, pw, md):
    temp = val
    for i in range(1, pw):
        temp = val * temp % md
    return temp
```

Усі вищеозначені функції повинні бути викликані лише раз для ініціалізації системи, далі ми можемо використовувати вже безпосередньо функції шифрування та розшифрування стільки, скільки потрібно.

Функція шифрування. Ми передаємо усі необхідні параметри та отримуємо готовий шифротекст як результат виконання.

```
def encode(y,m,u,r,n):
    return (y**m)*(u**r) % n
```

Функція декодування має в собі ітеративний підхід, це означає шо час виконання розшифрування може змінюватися, проте кількість кроків необхідних для декодування завжди фіксована параметром розміру блоку.

```
def decode(c,fi,r,n,y):
    a = power_mod(c,int(fi/r),n)
    x = power_mod(y,int(fi/r),n)
    for i in range(1, r):
        if x**i % n == a:
            return i
```

Ці функції е основними для криптосистеми Бенало. За необхідністю вхідний текст також може потребувати певних модифікації, проте для системи електронного голосування це не є необхідним.

ДОДАТОК Б

УДК 004.056.55

КАКТУС, КАКТУС

ЕЛЕКТРОННЕ ГОЛОСУВАННЯ НА ОСНОВІ ТЕХНОЛОГІЇ БЛОКЧЕЙН

Вступ

Електронне голосування являє собою подальший розвиток існуючих технологій голосування, що використовуються в Україні та інших демократичних країнах. Такий складний процес як волевиявлення потребує ретельної уваги до кожного з аспектів, які гарантує Конституція та відповідні закони. Використання електронного голосування (у подальшому ЕГ) спрямовано на покращення якості голосування у порівнянні з існуючою системою. ЕГ спрямоване в першу чергу на підвищення рівня прозорості, швидкості, анонімності, достовірності та стійкості до навмисних та ненавмисних завад, атак, тощо.

Загалом розрізняють онлайн та офлайн електронне голосування [1]. Відрізняються вони необхідністю факту фізичної наявності виборця для волевиявлення. У разі необхідності присутності виборця, ЕГ називають офлайн, якщо голосування може бути здійснено дистанційно – онлайн. В свою чергу кожний з цих типів представлений у декількох варіантах:

Офлайн: повністю електронне; з електронним підрахунком голосів.

Онлайн: за допомогою Інтернет-ресурсу; за допомогою мобільного телефону.

Перспективними, спираючись на кількість мобільних телефонів та ступінь розвитку мобільної інфраструктури, виглядають системи онлайн голосування. Легкість доступу до онлайн ресурсів та розвиток криптографії у цій сфері робить системи ЕГ з використанням Інтернет-ресурсу більш бажаними у порівнянні з системами, що використовують канали мобільного зв’язку, і тому ми докладніше розглянемо саме їх.

Голосування онлайн

До переваг онлайн голосування з використанням Інтернет-ресурсів відносяться [2]: можливість дистанційного голосування; забезпечення більш комфортних умов; можливість використання технології для проведення голосування в інших сферах/питаннях; прозорість виборів; вищий рівень залученості молоді; вартість проведення виборів; швидкість та достовірність підрахунку голосів.

Онлайн голосування вирішує деякі фундаментальні проблеми звичайного голосування, в свою чергу успадковуючи деякі недоліки та створюючи деякі нові вразливості. Голосування онлайн є складним для людей, які лише іноді взаємодіють з технологіями або знаходяться у більш-менш ізольованих регіонах. Існує також можливість компрометації бази даних та саботажу виборчих процесів. Через безпосередню роботу з персональними даними, забезпечення збереження їх секретності становить важливу задачу.

Венеціанська Комісія сформувала ряд рекомендації, які є бажаними до виконання у разі проведення демократичних електронних виборів. Кожна з цих рекомендації має за мету покращити систему ЕГ в одному з аспектів, а тому використання лише частини з них не може гарантувати комплексного покращення рівня безпечності голосування. До цих рекомендації відносять:

- електронне голосування може використовуватися лише за умови, що система є безпечною/захищеною і надійною;

- система електронного голосування повинна бути прозорою, тобто надавати можливість перевірки щодо її функціонування;
- виборці повинні мати нагоду одержати підтвердження свого вибору і виправити його у разі допущення помилки;
- для полегшення перерахунку голосів у разі конфліктної ситуації може передбачатися процедура роздрукування голосів.

Блокчейн

Блокчейн – систематизований ланцюг блоків (зв'язаний список), розподілений між усіма вузлами мережі. Кожний блок містить в собі інформацію про усі транзакції, що були оброблені в мережі. Таким чином історія мережі зберігається в кожному з вузлів, що робить фальсифікацію транзакцій значно складнішою.

Важливо відмітити інші технології, які роблять блокчейн систему більш надійним способом зберігання інформації. Геш-функції у блокчейн слугують своєрідним підписом, перевірка якого може слугувати підтвердженням відповідності. В такому разі криптостійкість системи напряму залежить від криптостійкості геш-функції.

Зрозуміло, що використання простих геш-сум не є достатнім для забезпечення необхідного рівня безпечності, тому використовують так зване дерево Мерклі. Дерево Мерклі (рис. 1) це спеціалізована структура даних яка пов'язує геші попередніх блоків з наступними блоками. Виконана ця структура у вигляді двійкового дерева. Спочатку ми беремо пару геш-значень обчислених для певних блоків, на які поділено деякий масив даних та обчислюємо їх загальну геш-суму. В тому разі, якщо пари в блоку немає, він потрапляє на наступний рівень без змін. Обчислення геш-сум пар виконується до тих пір, доки не залишиться лише одне геш-значення. Дерево Мерклі гарантує збереження цілісності та достовірності збереженої інформації.

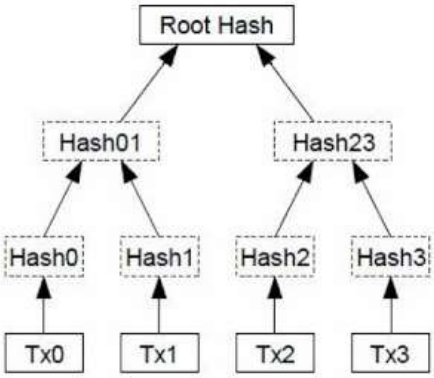

Рис. 1 Структура дерева Мерклі

Верхнє геш-значення (*root hash*) потрібно для безпосереднього порівняння. Перш, ніж підтверджувати трансакцію, необхідно завантажити верхній геш з надійного джерела, яке не обов'язково повинно бути звичайним вузлом мережі. Для цього підійде Інтернет-ресурс з надійною та захищеною базою даних. Далі порівнюють отримані геш-значення з перевіреним. Якщо геш-значення не співпадають, це означає що геш-значення було пошкоджено або скомпрометовано. У такому разі потрібно обчислювати геш-значення на іншому вузлі та продовжувати процедуру доки співвідношення не буде знайдене.

Гомоморфне шифрування

Гомоморфне шифрування – шифрування основною превагою якого є можливість виконувати певні математичні перетворення з шифротекстами, зашифрований результат яких буде відповідати виконанню відповідної операції з відкритим текстом [3].

Системи можуть бути: цілком гомоморфними; частково гомоморфними.

Частково гомоморфними називають системи, які гомоморфні лише для однієї з алгебраїчних операції (множення(1) чи додавання(2)).

Цілком гомоморфні – відповідно ті, що гомоморфні по обом з цих операції (1, 2).

$$D(E(t_1) \otimes E(t_2))=t_1 \times t_2, \quad (1)$$

$$D(E(t_1) \oplus E(t_2))=t_1+t_2, \quad (2)$$

де t – відкритий текст, $E(t)$ – функція шифрування, $D(E(t))$ – функція розшифрування.

Гомоморфне шифрування має великі переваги для її використання у системах електронного голосування. Можливість виконувати операції над зашифрованими даними, зберігаючи анонімними дані виборців.

Існує декілька схем електронного голосування, які використовують гомоморфне шифрування.

Розглянемо криптосистему Бенало, яка є частково гомоморфною за операцією додавання. Кожний з виборців розподіляє свій голос на складові частини, використовуючи відповідну схему гомоморфності по додаванню і відправляє отримані частини представникам обранця. Гоморфність системи гарантує, що при складанні зашифрованих частин результат буде відповідати зашифрованому голосу. Деякий суб'єкт обчислює кінцевий результат голосування, що йому передали представники та отримує дійсний результат голосування.

Саме шифрування у системі Бенало використовує формулу (3)

$$E_r(m)=(y^m u^r) mod\ n, \quad (3),$$

де u – випадкове число Z_r; m – повідомлення; p, q – прості числа, такі, що r, $(p-1)/r$, $(q-1)$ є взаємно простими; y обирається таким чином, що $y^{(p-1)*(q-1)/p(i)} \neq 1$.

Стійкість системи базується на складності обчислення лишків великих ступенів.

Система Бенало базується та модифікує систему Гольдвассера-Мікалі.

Висновки

1. Електронне голосування є наступним кроком розвитку системи голосування в Україні. Перспективність ЕГ допоможе поліпшити діючу виборчу систему та частково усунути існуючі проблеми.

2. Введення ЕГ допоможе збільшити кількості активних виборців серед молоді, забезпечити комфортніші умови голосування, також і для людей у віддалених регіонах та людей з вадами. Система ЕГ є більш прозорою, надійною та швидкою у порівнянні з існуючою.

3. Зрозуміло, що крім переваг існують також недоліки: комп'ютерні збої, хакерські атаки, ціна, необхідна для розробки системи ЕГ, необхідність мати базову комп'ютерну грамотність та необхідні пристрої.

4. Електронна система голосування є альтернативною і для початку вона має бути опцією, що не нав'язує нові технології і не відлякує виборців. У подальшому введення її як основної поліпшить виборчу систему.

ЛІТЕРАТУРА

1. Готун А., Використання нових інформаційних технологій у виборчому процесі: світовий досвід і практика застосування в Україні, Вісник Київського НУ ім. Шевченка. Філософія і політологія. – 2008. – С. 89-90.

2. Оніпко О., Шляхи вдосконалення технології виборчого процесу, Вісн. Центр. вибор. коміс. – 2006. – Вип. № 4 (6). – С. 67–70. – URL: http://www.cvk.gov.ua/visnyk/pdf/2006_4/visnyk_st_22.pdf.

3. Homomorphic encryption – URL: https://en.wikipedia.org/wiki/Homomorphic_encryption.