**УДК 004.056.55** **DOI:10.30837/rt.2019.3.198.12**

*І.Д. ГОРБЕНКО, д-р техн. наук, О.В. ПОТІЙ, д-р техн. наук,*
*Ю.І. ГОРБЕНКО, канд. техн. наук, А.І. ПУШКАРЬОВ, М. В. ЄСІНА, канд. техн. наук*

# ПРИНЦИПИ ПОБУДУВАННЯ ТА АНАЛІЗУ ІНФРАСТРУКТУР ВІДКРИТОГО КЛЮЧА НА ОСНОВІ ЗАСТОСУВАННЯ ТЕХНОЛОГІЇ БЛОКЧЕЙН

## Вступ

Результати теоретичних та практичних досліджень асистемних підходів та досвіду застосування нових інформаційних технологій (ІТ), що наведені у [1 – 9] дозволяють зробити висновки про можливості інтенсивного розроблення, активного розповсюдження та застосування децентралізованих інформаційних технологій. Для досягнення вказаного в тій чи іншій мірі необхідно виконати наступні вимоги [10, 1 – 9].

1. Застосування принципу децентралізації існуючих ІТ повинне поліпшити хоча б один із важливих для цільового застосування параметр: вартість, складність (часова та просторова), швидкість, прибутковість, безпека (загальна та інформаційна), анонімність, прозорість, гнучкість тощо. Причому важливо, щоби цільовий параметр, за яким здійснюється оцінка, пропонувався самим замовником (користувачами).

2. Покращення ІТ, що досягається, має бути суттєвим, причому децентралізація повинна покращувати хоча б один важливий для користувачів параметр, як мінімум в 2-3 рази.

3. Нова, в даному випадку технологія блокчейн (ТБЧ), не повинна істотно програвати існуючим ІТ за іншими важливими параметрами. Наприклад [10], якщо ТБЧ працює в три рази швидше, але, якщо вона при цьому в два рази дорожче і в 1,5 рази складніше – вона, скоріше за все, не буде застосовуватись. Вважається, що у цілому нова ІТ повинна бути в чомусь краще в 2-3 рази, а за всіма іншими параметрами програвати не більше, ніж у 1,5 рази. Тобто, покращення повинне не тільки давати суттєві переваги, але ще й компенсувати побічні ефекти програшу.

Таким чином, «кращість» і перевага нової ІТ носить суб'єктивний характер, вони повинні формуватись скоріше всього користувачами, і в меншій мірі розробниками. У більшості випадків кращість може визначатись рівнем продаж таких ІТ. Відсутність певних продаж показує, що явної переваги цільовою аудиторії відносно нової чи удосконаленої ІТ не визнано.

Аналіз показав, що серед нових технологій суттєвий розвиток та «кращість» досягають ІТ, що розробляються чи удосконалюються на основі використання ТБЧ, що ґрунтується на децентралізації. Таким чином, існує проблема, сутність якої в тому, що децентралізація, в тому числі у вигляді ТБЧ, є «начебто» перспективною інформаційною технологією. В той же час, де саме та як її краще використовувати, де вона буде, в порівнянні з існуючими технологіями типу «Клієнт–сервер», кращою, на наш погляд, є не вирішеним питанням.

Одним із важливих та необхідних додатків ТБЧ є, по суті, удосконалення інфраструктури відкритого ключа (ІВК) на основі використання при її побудови принципів децентралізації та прозорості тощо. У явному вигляді проблема вже викладена та обговорюється в значному числі робіт [1 – 15].

Мета статті:

- обґрунтування можливостей та необхідності створення ІВК на основі ТБЧ;
- розробка структури ІВК на основі БЧ та оцінка складності впровадження;
- аналіз удосконаленої моделі ІВК з прозорістю сертифікатів на основі БЧ;
- аналіз основних проблемних питань перспективних ІВК на базі БЧ;
- загальна оцінка стійкості ІВК на основі БЧ до відомих атак.

Автори розуміють, що стаття носить характер первинного загальносистемного аналітичного огляду та відображає погляди авторів на можливості та необхідність створення ІВК з використанням ТБЧ. Ця наша впевненість ґрунтується на тому, що діюча ІВК України роз-

роблена в суттєвій мірі за участі авторів цієї статті, в тому числі практично реалізована та підтримується при експлуатації [16 – 18].

**Загальні положення щодо побудування та аналізу ІВК на основі застосування БЧ**

Особливістю асиметричних криптографічних перетворень є те, що при їх виконанні використовується одна або декілька асиметричних пар ключів. Наприклад, для ЕП та АСШ використовуються дві різні асиметричні ключові пари [16 – 19]. Наприклад, для криптографічних перетворень у групі точок еліптичної кривої [16, 23] кожна асиметрична ключова пара ($d_A$, $Q_A$), де $1 \le d_A < n$ є випадкове число – особистий (закритий) ключ, а $Q_A$ – точка на еліптичній кривій – відкритий ключ, що обчислюється способом використання скалярного множення:

$$Q_A = d_A \cdot G (\mathrm{mod}\, q),$$

де $G$ – базова точка на еліптичній кривій порядку $n$, $q$ – модуль перетворення.

Згідно з вимогами до таких асиметричних криптосистем щодо застосування особистих ключів повинні безумовно бути виконаними вимоги забезпечення їх конфіденційності, цілісності, справжності, доступності та неспростовності. Вказані вимоги можуть бути забезпеченими кожним із користувачів, оскільки особистий ключ доступний тільки його власнику і він повинен і може зберігати його в таємниці. В той же час відкритий ключ повинен бути доступним, як мінімум, усім користувачам домену, а то і усьому цифровому світу. При застосуванні відкритих ключів з високим рівнем гарантій повинні бути забезпеченими щодо них послуги цілісності, справжності, доступності та неспростовності, незалежно від математичного методу, що використовується для побудови (генерування) асиметричної пари. Вказані асиметричні пари ключів застосовуються для електронного підпису (ЕП), асиметричного шифрування (АСШ) та різних криптографічних протоколів (КРП).

В системах БЧ ЕП є основним механізмом забезпечення цілісності, справжності та неспростовності транзакцій. У [16 – 23] наведено основні методи асиметричних криптоперетворень, стандартизованих щодо ЕП, які на нинішній час знайшли широке застосування, в тому числі для захисту транзакцій. Причому відкриті ключі перевірки ЕП повинні бути доступними всім користувачам, що виконують, наприклад, перевірку підписаних електронних документів, даних тощо. За таких умов необхідно забезпечити їх цілісність, справжність і доступність та їх неспростовність для надання користувачам електронних довірчих послуг [19, 22].

**Факти та приклади здійснення атак на існуючі ІВК**

Нині ІВК є третьою довіреною стороною, яка надає, в тому числі, послуги автентифікації та ідентифікації особистостей в Інтернеті та інших мережах. У загальному випадку ІВК визначає політику та процедури, що необхідні для видачі, управління, перевірки та розповсюдження цифрових сертифікатів для безпечного використання підпису (ЕП), асиметричного шифрування (АСШ) та різних криптографічних протоколів (КРП) [22, 24]. Зазвичай управління ІВК ґрунтується на стандарті сертифіката [24] ДСТУ ITU-TRec.X.509|ISO/IEC 9594-8:2006 (2015), який забезпечує перевірку права власності на особистий ключ деяким зовнішнім об'єктом. Сертифікат X.509 визначає структуру даних, яка пов'язує значення відкритого ключа з суб'єктами (наприклад, доменними іменами) тощо. Причому прив'язка користувачів затверджується центром сертифікації ключів (ЦСК), які з використанням особистого ключа підписують кожен відкритий ключ, виготовляючи таким чином сертифікат відповідного відкритого ключа.

У процесі широкого застосування ІВК виявлено суттєвий недолік, що пов’язаний з можливою компрометацією особистих ключів чи особистого ключа ЦСК. У цьому випадку ЦСК є точками відмови в ІВК, оскільки це приводить до компрометації ключів усіх користувачів, що обслуговуються цим ЦСК [36 – 39]. Нижче наводяться приклади таких компрометацій.

Їх критичність в тому, що компрометація особистого ключа ЦСК приводить до компрометації усіх користувачів. Тому потрібно провести повне відновлення засобом блокування компрометованих особистих ключів як ЦСК, так і користувачів, що є надскладною проблемою [16 – 18].

У ряді джерел наведено приклади таких компрометацій, в тому числі [25]. Так ряд таких веб-додатків, як інтернет-банкінг, розсилка повідомлень, електронна торгівля тощо стали невід'ємною частиною нинішнього життя. У світі як стандарт де-факто, для забезпечення послуг автентичності, цілісності та конфіденційності у вказаних додатках використовуються сертифікати SSL/TLS. Ці сертифікати видаються ЦСК, які вважаються третіми довіреними організаціями. Зокрема, очікується, що ЦСК діють згідно з деякими правилами, які позначені як документи про сертифікаційну політику (Certificate Policy – CP) та Заяви про практики сертифікатів (Cate Practice Statement – CPS). У такій моделі довіри ЦСК мають абсолютну відповідальність за видачу дійсних сертифікатів кожному суб'єкту (користувачу). Проте ЦСК можуть бути скомпрометовані та підроблені, причому чинні сертифікати можуть бути видані через неналежну практику безпеки або невідповідність CP та CPS. Протягом останнього десятиліття відбулися серйозні інциденти через вищезгадані причини, які коротко наводяться нижче [25, 26].

1) Шкідливе програмне забезпечення Stuxnet підписано особистими ключами двох скомпрометованих тайванських ЦСК, мета яких контролювати специфічну промислову систему, яка, ймовірно, є в Ірані, наприклад, газопровід або електростанція.

2) Comodo CA (ЦСК), що має велику частку на ринку SSL, зламано в березні 2011 р. [53]. Один з центрів реєстрації (ЦР) піддався атаці, щоб видати 9 сертифікатів, де зловмисник простежується назад до Ірану.

3) Голландський CA DigiNotar започатковано в липні 2011 р., було видано 531 зловмисний сертифікат для цінних доменів, таких як *.google.com, *windowsupdate.com та *.mozilla.com. Ці сертифікати можуть бути легко використані для поширення зловмисних оновлень Windows або плагінів Firefox без привернення уваги. Щонайменше 300000 унікальних IP-адрес виявлено за допомогою служб Google через ці сертифікати, 99 % трафіку яких надходить з Ірану.

4) Компанія Trustwave Center Authority продала сертифікат для підлеглого CA. Цей під-CA випустив зловмисні сертифікати TLS, якими вони користувалися у внутрішньому трафіку TLS.

5) Турецька компанія CA Turktrust помилково видала сертифікати ЦСК замість сертифікатів TLS у грудні 2012 р. Ці сертифікати використовувались для створення сертифікатів TLS для внутрішнього трафіку. Google визначив зловмисний сертифікат Google через Chrome.

6) Під-ЦСК китайської компанії CNNIC, що знаходиться в Єгипті, видала зловмисні сертифікати TLS для інспекції дорожнього руху в березні 2015 р. Пізніше визначено, що CNNIC експлуатується без документованого CPS.

7) Lenovo Superfish розгорнув місцеві ЦС у своїх продуктах у 2015 р. Цей сертифікат використовується для вставки реклам у веб-сайти, які захищені TLS. Оскільки закриті ключі ЦС знаходяться в оперативній пам'яті комп'ютера, вони можуть бути легко використані для внутрішнього трафіку.

8) У вересні 2015 р. компанія Symantec видала неавторизовані сертифікати для доменів Google. Пізніше Symantec стверджував, що ці сертифікати виготовляються з метою тестування.

9) Symantec придбав Blue Coat у травні 2016 р. Blue Coat має пристрої для перехоплення зашифрованого Інтернет-трафіку. Blue Coat став під-ЦС при Symantec. Ця уніфікація посилила скептицизм.

10) Малайзійський ЦСК DigiCert Sdn. Bhd. помилково випустив 22 слабкі SSL-сертифікати, які можна було використовувати для видавання веб-сайтів і підписання шкідли-

вого програмного забезпечення. У результаті, основним браузерам довелося відкликати свою довіру до всіх сертифікатів, виданих DigiCert Sdn. Bhd.

Існували також проблеми з порушеннями сертифікату TrustWave, великим американським центром сертифікації. TrustWave визнав, що він видав підпорядковані кореневі сертифікати одному з клієнтів, а клієнт зміг контролювати трафік на їх внутрішній мережі. Загроза цього в тому, що підпорядковані кореневі сертифікати дозволяють їх власникам створювати SSL-сертифікати для майже будь-якого домену в Інтернеті. Хоча TrustWave скасував сертифікат і заявив, що він більше не буде видавати підпорядковані кореневі сертифікати клієнтам, він показує, наскільки легко ЦСК може робити помилки і наскільки серйозними можуть бути наслідки цих помилок.

Наведені фатальні випадки призводять до того, що багато досліджень розподіляють абсолютну довіру до ЦС на декілька органів. Для виявлення підроблених, але дійсних сертифікатів TLS [25, 26], застосовується закріплення ключа, краудсорсингу та доведення до браузерів інформації про відкликання тощо. Вказані початкові рішення, які частково реалізовані, на жаль зазнали невдачі через проблеми масштабування.

**Основні підходи класичного вирішення проблеми захисту ІВК**

У процесі досліджень та удосконалення ІВК було запропоновано два підходи класичного вирішення проблеми ІВК SSL/TLS. Це удосконалення існуючої ІВК на основі системного журналу [26] та застосування децентралізованої мережі однорангової сертифікації, яка отримала назву Мережі довіри (Web of Trust, WoT) [26].

*Підхід щодо ІВК, заснований на журналах (лог)* [26]. Такий підхід був запропонований в якості нового вирішення проблеми удосконалення традиційних ІВК – його ЦСК. Ідея підходу полягає у використанні загальнодоступних серверів журналів, які контролюють та публікують сертифікати, що видані ЦСК. Такі загальнодоступні журнали забезпечують прозорість, гарантуючи, що лише загальнодоступні сертифікати приймаються та довіряються кінцевим клієнтам. Отже, будь-яка неправильна поведінка ЦСК буде виявлена користувачами та серверами. Прикладом реалізації такого підходу є сертифікати Google Transparency [26], що є найбільш поширенішою ІВК на основі журналу. Зараз такий підхід доступний як у системах Chrome, так і у Firefox. Також відомо багато пропозицій, що дозволяють розширити можливості удосконалених ІВК на основі журналу, в основному це може бути досягнуто за рахунок підтримки відкликання та обробки помилок. Але, на жаль, незважаючи на ці переваги такого ІВК, вони все ще мають кілька проблем, що пов'язані, наприклад, з відкликанням сертифікатів, як пояснено у [25, 26].

*Підхід щодо ІВК на основі мережі довіри (Web of Trust – WoT).* Підхід заснований на основі децентралізації, при його використанні користувачі можуть визнавати як надійних інших підписувачів, підписуючи у них сертифікати відкритих ключів. У даному випадку кожен користувач має сертифікат, що містить його відкритий ключ і електронні (цифрові) підписи від осіб, які вважають його таким, що заслуговує на довіру. Потім сертифікат завіряється третьою довіреною стороною, наприклад ЦСК, якщо можливо перевірити, що сертифікат містить підпис того, кому є довіра. Такий підхід має перевагу над розподіленим характером довіри, оскільки в цьому випадку усувається будь-яка центральна точка відмови ЦСК (компрометації його особистого ключа). Але такий підхід має недолік, що пов'язаний з ускладненням приєднання нових або віддалених користувачів до мережі. Скоріше всього це пов’язане з тим, що деякі існуючі члени WoT зазвичай повинні особисто зустрітися з новим користувачем, щоб вперше підтвердити свою особистість і підписати відкритий ключ. Крім того, на відміну від підходу, заснованого на ЦСК, при застосуванні WoT виникають проблеми з відкликанням компрометованого ключа. Тобто користувач, який обмежений вибором користувача, на довіру якого він спирається, не може відкликати особистий ключ в разі його втрати або компрометації. Практичні можливості такого підходу залежать від можливостей періо-

дичної відправки у браузери списків відкликаних сертифікатів. В такому разі виникає довіра до недійсного (компрометованого) сертифікату [24 – 26].

**Підхід до побудови ІВК, заснованого на БЧ**

Аналіз показав, що застосування ТБЧ для створення захищених ІВК надихнуло багатьох дослідників, внаслідок появилось значне число робіт, в першу чергу [26 – 28]. Основний аргумент при обґрунтуванні застосування БЧ полягає в тому, що рішення, засновані на ТБЧ, можуть об'єднати переваги ІВК засновані на журналах, та підході WoT, а також вирішити деякі проблеми зі звичайною системою ІВК. Так, з одного боку, БЧ усуває потенційні точки відмови підходу заснованого на ІВК на основі журналу і проблеми розгортання, які розглянемо нижче. З іншого боку, підхід, що заснований на БЧ, пом'якшує потреби WoT у нових власниках сертифікатів, щоб довести достовірність існуючих членів мережі, а також пом'якшує вимоги WoT для нових власників сертифікатів, щоб довести достовірність існуючих членів мережі.

Згідно з [25 – 28] удосконалення може засновуватись на ТБЧ і інфраструктурі ІВК для управління сертифікатами X.509. Вказане може досягатись на розширенні формату стандартного сертифіката X.509, так щоби він був сумісний з підходом ІВК на основі ТБЧ. Це, по суті, досягається завдяки полям розширення X.509, які використовуються для вбудовування метаданих ТБЧ. Також ІВК на основі БЧ забезпечує надійне керування цифровими сертифікатами.

У першу чергу при удосконаленні ІВК на основі БЧ необхідно обґрунтувати ланцюг довіри.

Як уже розглядалось [24, 16], класичні системи ІВК ґрунтуються на основі ЦСК, які виготовляють та обслуговують сертифікати, що відповідають стандарту X.509. Кожен сертифікат засвідчує право власності на відкритий ключ. Наприклад, коли користувач входить в Twitter через веб-браузер, спочатку веб-браузер перевіряє заявлений сертифікат, який містить відкритий ключ Twitter, перевіряючи ЦСК даного сертифікату. Зазвичай веб-браузери попередньо налаштовані на прийом сертифікатів від певних відомих ЦСК. Для того, щоб сертифікат був довіреним, він повинен бути виданий кореневим центром сертифікації, який існує в довіреному сховищі браузера або пристрої користувача, або допоміжним центром сертифікації, якому довіряють за допомогою підпису кореневого ЦСК. Так нині, як правило, продукти Mozilla постачаються з 154 кореневими сертифікатами [29]. Крім того, фірми Apple, Microsoft і Google мають своє власне сховище довірених кореневих сертифікатів, що вбудовані у їхні продукти.

Зв'язок між конкретним даним сертифікатом і кореневим сертифікатом відомий як ланцюг довіри. При цьому важливо, що ланцюг довіри може включати будь-яку кількість сертифікатів підлеглих ЦСК, тобто ЦСК нижчого рівню. Тому між даним сертифікатом і сертифікатом кореневого СА є зв’язок, а X.509v3 [24] має розширення під назвою Основні обмеження, що може обмежити максимальну глибину дійсного ланцюга сертифікатів (ланцюга довіри) [30].

На рис. 1 [26] наведено шлях сертифікації від сертифіката кінцевого суб'єкта до кореневого ЦСК, де починається ланцюг довіри. Отже, якщо сертифікат кінцевого об'єкта не був виданий довіреним ЦСК, веб-браузер потім перевірить, чи був сертифікат ЦСК випущений довіреним ЦСК, і т.д. поки не буде знайдено довірений ЦСК. За цієї умови браузер зазвичай відображає помилку.

Особливості застосування технології БЧ. Розподілений реєстр, тобто БЧ, позитивно розглядається завдяки успіху в застосуванні у Bitcoin. Нині більшість БЧ-платформ використовуються у фінансових додатках, однак починають з'являтися все більше нових додатків для різних сфер. Звичайно це додатки, що вимагають високої надійності і повного усунення ризиків маніпулювання даними. Також БЧ є розподіленим, тому він не має вразливостей, що пов’язані з одиничною точкою відмови.

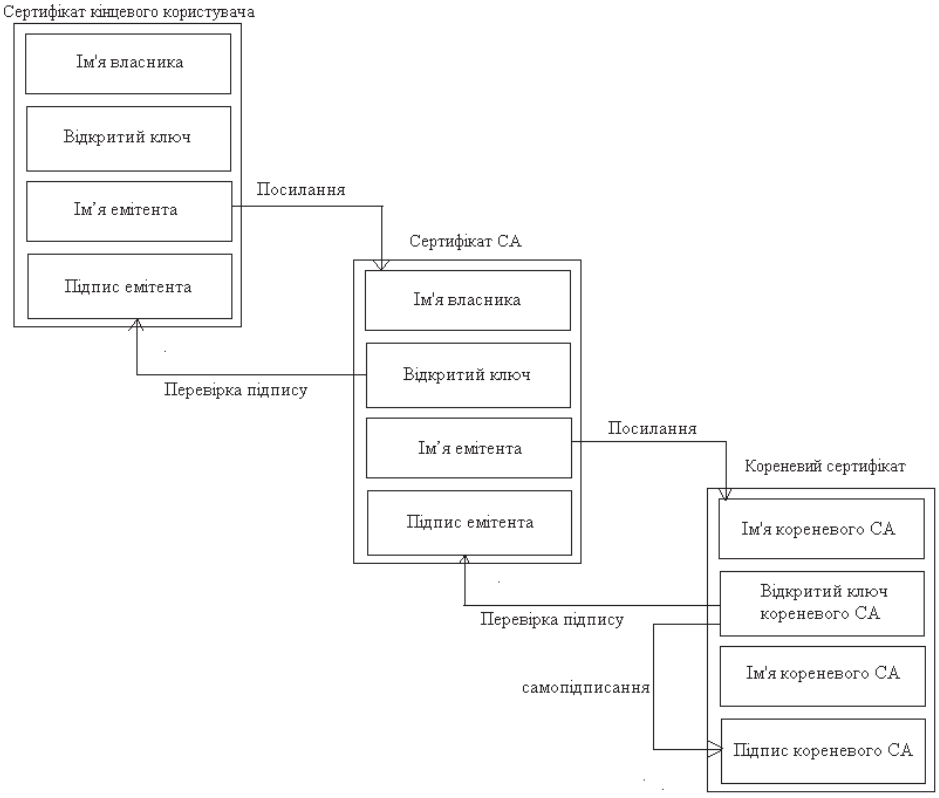


Рис. 1. Ланцюг традиційної довіри

Наразі розроблено БЧ, що дозволяють виконувати довільну логіку, відому як смарт-контракти. У загальному смарт-контракт – це програма, яка виконується поверх БЧ і використовує базовий порядок транзакцій для забезпечення узгодженості результатів виконання смарт-контракту між партнерами (peers) [32]. Так, наприклад, БЧ Ethereum підтримує складну та повну за Тьюрингом мову Solidity (http://solidity.readthedocs.io/en/develop/), яка може бути використана для програмування та визначення широкого кола сценаріїв застосування [32].

Також, щодо ІВК технологія БЧ надає такі важливі засоби безпеки як відкликання сертифікатів, усунення центральних точок збою і надійний запис транзакцій. Наприклад, при швидкому відкликанні сертифікатів ІВК, що засновані на БЧ, можна миттєво ізолювати інфікований ЦСК і відповідні сертифікати без очікування наступного оновлення списків відкликаних сертифікатів (CRL). Крім того, ІВК на основі ТБЧ, як відкритий журнал тільки для додавання (запису), природно надає властивість прозорості сертифікату, що запропоновано в [27]. Тому при подальших дослідженнях необхідно вибрати платформу, наприклад, Ethereum і мову програмування смарт-контрактів, наприклад, Solidity. Це можна пояснити тим, що вони мають велике співтовариство розробників програмного забезпечення з відкритим вихідним кодом. Це робить процес розробки програмного забезпечення набагато більш ефективним.

Основні дослідження щодо реалізації БЧ для побудови системи ІВК. Реалізація БЧ для побудови системи ІВК ретельно вивчалась дослідниками та розробниками. Так, в [33] автори пропонують Blockstack, який використовує для надання системи реєстрації імен, реалізацію БЧ біткойну, де імена пов'язані з відкритими ключами.

Подібно до Blockstark, ІВК на основі БЧ реалізується в Emercoin (https://emercoin.com/en/tech-solutions/) (проект EmerSSH). Emercoin – це загальнодоступний ТБЧ, досить близький до біткойну з точки зору архітектури, що включає в себе гібридний консенсус доказу роботи і доказу ставки (в залежності від доступності майнингових потужностей). Emercoin не має смарт-контрактів і зберігає тільки ґеш-значення сертифікатів у ТБЧ. Сам EmerSSH зберігає тільки ґеш-значення сертифіката для ТБЧ, що на думку авторів, зме-

ншиться ризик «людина посередині». Це досягається тим, що як тільки ґеш-значення сертифіката завантажується в БЧ, встановлюється безпечне з'єднання за допомогою відкритого ключа сертифікату і обмін для кожного з'єднання проводиться абсолютно новими ключами.

Спеціалісти Fromknecht та інші [28] пропонують для реалізації ІВК для зберігання доменів та пов'язаних з ними відкритих ключів використовувати БЧ Certcoin.

Аналіз показав, що усі згадані дослідження пропонують підходи, що засновані виключно на БЧ. У роботі [26] пропонується використовувати загальний стандартний сертифікат X.509v3 з незначним доповненням до полів розширення з інформацією, що пов'язана з ТБЧ [30]. При цьому розширений сертифікат X.509 може бути перевірений класичним ланцюгом довіри на основі ЦСК або з використанням структури ІВК на основі ТБЧ. Вказані автори, на наш погляд, першими запропонували такий гібридний сертифікат.

**Структура ІВК на основі БЧ**

У цьому підрозділі аналізується структура для управління ІВК на платформі БЧ [26].

*Основні можливості захисту*. Структура ІВК на основі БЧ підтримує відкликання сертифікату, що є суттєвою проблемою в традиційних системах ІВК. Також, оскільки неможливо видалити інформацію з БЧ, то тільки батьківський ЦСК може позначати виданий ним сертифікат як відкликаний. Тобто, будь-яка неправильна поведінка ЦСК стосовно відкликання сертифікату буде також простежена і помічена всіма іншими суб'єктами.

*Особливості проектування та застосування (методологія проектування)*. Структура ІВК [26] на основі БЧ базується на гібридних сертифікатах X.509, як це показано на рис. 2. Сертифікат містить певну інформацію про середовище ІВК в полях розширення. Значення полів розширення наступні:

- Ідентифікатор ключа суб'єкта: зберігає особистість власника сертифіката.

- Ім'я БЧ: містить назву платформи ТБЧ. Наразі використовується загальнодоступний ТБЧ Ethereum, але потрібно охопити більше платформ.

- Ідентифікатор ключа ІВК: містить адресу смарт-контракту поточного ЦСК, якщо це сертифікат ЦСК. Для сертифікатів не ЦСК поле порожнє.

- Ідентифікатор ЦСК емітента: має адресу смарт-контракту ЦСК, що видав цей сертифікат. Дозволяє валідатору знайти смарт-контракт батьківського ЦСК в ТБЧ і перевірити, чи сертифікат з відповідним ґеш-значенням був виданий і не був відкликаний.

Для кореневих сертифікатів це поле порожнє.

- Алгоритм ґешування: містить інформацію про алгоритм ґешування, який використовувався при обчисленні ґеш-значення сертифікату, завантаженого в ТБЧ.

Таким чином, згідно [26] структура передбачає три типи сертифікатів: сертифікати кореневого ЦСК (Root-CA), під-ЦСК (Sub-CA) і кінцевого користувача (Enduser). У табл. 1 наведено ієрархію гібридних сертифікатів БЧ. У першому рядку представлений кореневий ЦСК, в якому сертифікат випущений і підписаний ним самим (самопідписаний). ЦСК емітента відсутній, що підтверджено ID емітента ЦСК (в п'ятому стовпці 0x00000000). Сертифікат під-ЦСК подається у другому рядку – він був виданий кореневим ЦСК, а ID емітента ЦСК вказує на кореневий ЦСК. Останній рядок містить сертифікат кінцевого користувача, що виданий під-ЦСК. У наступному розділі ми пояснюємо, як ми реалізували цю структуру на платформі ТБЧ.

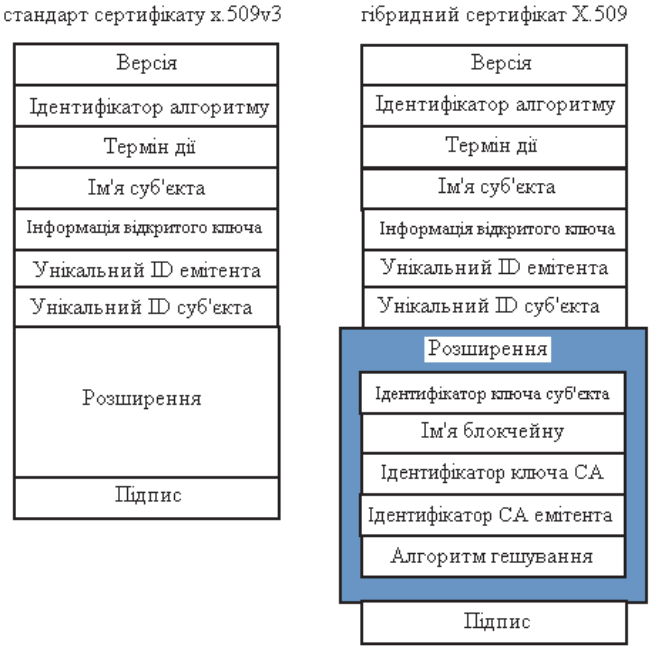


Рис. 2. Стандартний і гібридний сертифікат X.509

Таблиця 1

Ієрархія гібридних сертифікатів ТБЧ

| Cert. | Issued By | Issued To | CA Contract ID | Issuer CA ID |
|---|---|---|---|---|
| RootCA | RootCA | RootCA | 0x1234xxxx | 0x00000000 |
| SubCA | RootCA | SubCA | 0x5631xxxx | 0x1234xxxx |
| EndUser | SubCA | End user | - | 0x5631xxxx |

Архітектура ІВК БЧ. Основна ідея структури згідно [26] полягає в тому, що кожен WCR має спеціальний смарт-контракт, який містить наступну інформацію:

- Масив з ґеш-значеннями виданих сертифікатів, а також може містити дату закінчення терміну дії кожного сертифікату та іншу технічну інформацію.

- Відображення даних про відкликання, на які посилається ґеш-значення сертифікату. Якщо сертифікат відкликано, ЦСК, який видав цей сертифікат, додає дані відкликаного сертифікату.

Крім того, якщо сертифікат є сертифікатом ЦСК, то він також завантажується у відповідний смарт-контракт ЦСК. Далі, оскільки сертифікат містить адресу смарт-контрактів батьківського CA, то він дозволяє перевіряти все дерево центру сертифікації, тобто ланцюг довіри від користувача до кореневого сертифіката.

Реалізація ІВК заснованого на БЧ. Запропонована структура ІВК, заснована на БЧ, включає три основні частини – для тестування в основному використовувалися служба звільнення, перевірки сертифікатів та веб-інтерфейс користувача. Їх призначення у наступному [26].

1) Служба звільнення (Restful service). Розроблена разом з Golang як окремий веб-сервер, що забезпечує доступ до загальнодоступного БЧ Ethereum. Вона надає всі можливості видачі, відкликання та перевірки сертифікатів. Важливо відзначити, що перевірка проводиться «безкоштовно» з точки зору витрат криптовалюти загальнодоступного БЧ, оскільки перевірка не додає або не змінює дані в БЧ.

Служба звільнення (Restful service) надає такі основні функції:

- Реєстрація користувача. Додає ґеш-значення сертифікату до смарт-контракту даного ЦСК. У якості альтернативи надання ґеш-значення як параметру, сертифікат може бути

завантажений в службу Restful. У цьому випадку ґеш-значення обчислюється на основі завантаженого сертифіката.

- Чорний список користувачів: відкликає сертифікат, тобто переміщує сертифікат (звичайний або ЦСК) з білого списку до чорного списку. Технічно це досягається шляхом додавання посилання на ґеш-значення сертифікату до відображення відкликання у смарт-контракті.

- Створення контракту. Створює порожній контракт для нового ЦСК. Викликається батьківським ЦСК при видачі сертифіката для його під-ЦСК.

- Заповнення договору. Після створення порожнього смарт-контракту для під-ЦСК батьківський ЦСК повинен завантажити до нього сертифікат під-ЦСК, що містить адресу батьківського смарт-контракту та адресу смарт-контракту під-ЦСК у полях розширення. Після заповнення смарт-контракту під-ЦСК з сертифікатом його батьківського ЦСК, адреса облікового запису Ethereum під-ЦСК записується в змінну власника смарт-контракту, забезпечуючи таким чином доступ на запис тільки для під-СА.

- Перевірка-cert (перегляд/постійна функція). Перевірка сертифікату з листа до кореня дерева ЦСК. Важливо, що перевірка сертифікатів може бути проведена службою Restful, яка базується на коді Golang або на основі виклику окремого смарт-контракту перевірки. При цьому перевірка проводиться при нульовій вартості.

Перші чотири функції функціональності служби Restful передбачають авторизацію користувача Ethereum, що відповідає батьківському ЦСК. Важливою особливістю функціональності служби Restful є навмисне багатоступеневе ініціювання реєстрації під-ЦСК (додавання сертифіката під-ЦСК до списку затверджених сертифікатів в смарт-контракті батьківського ЦСК). На відміну від традиційного сертифіката кінцевого користувача, який ініціюється тільки за допомогою функції «Зареєструвати користувача служби Restful», сертифікат під-ЦСК запускається за допомогою наступних кроків.

- Батьківський ЦСК повинен створити порожній смарт-контракт для під-ЦСК з функцією Створити-контракт. Тепер батьківський ЦСК може генерувати гібридний сертифікат, що вводить нову адресу смарт-контракту у відповідне поле розширень сертифікату.

- Батьківський ЦСК заповнює новий смарт-контракт під-ЦСК згенерованим сертифікатом, використовуючи функцію «Заповнити контракт служби Restful». Після виконання функції «Заповнити контракт» права на написання нового смарт-контракту передаються виключно до під-ЦСК з заповненням адреси під-ЦСК Ethereum у поле власника нового смарт-контракту.

- Ґеш-значення смарт-контракту фіксується в білому списку батьківського ЦСК з функцією «Зареєструвати користувача».

2) Перевірка: Модуль перевірки містить смарт-контракт, який дозволяє перевірити ланцюг довіри для даного сертифікату (шлях від листа або сертифікату кінцевого об'єкта до кореня в дереві ЦСК. Важливо відзначити, що перевірка смарт-контракту не залежить від перевірки коду Golang у службі Restful, альтернативний підхід до перевірки сертифікатів також може бути реалізовано у такій структурі. Причому, обидва підходи до перевірки не передбачають жодних виплат криптовалюти, оскільки ТБЧ не змінюється.

3) Веб-інтерфейс користувача: Інтерфейс користувача дозволяє клієнтам перевіряти весь додаток – додавати сертифікати ЦСК та сертифікати кінцевого користувача на різних рівнях дерева ЦСК під різними обліковими записами ЦСК, відкликати сертифікати тощо. Очевидно, що веб-інтерфейс може розглядатися як оболонка для згаданої вище служби Restful. Варто зазначити, що тестовий веб-інтерфейс має свій власний смарт-контракт, який зберігає деякі дані, включаючи посилання від батьківського ЦСК до смарт-контрактів його під-ЦСК. Це дозволяє переходити від кореня до листів (сертифікатів кінцевого об'єкта) дерева сертифікатів, але за умов, що даний ланцюг довіри був завантажений з веб-інтерфейсом.

*Переваги ІВК на основі БЧ.* Аналіз показав, що ІВК на основі БЧ має перед традиційною ІВК наступні переваги:

- перевірка сертифікату і його ланцюга сертифікатів ЦСК проста і швидка;

- ІВК на основі ТБЧ вирішує давню проблему традиційних ІВК, не вимагаючи використання сервісу, який видає списки відкликання сертифікатів (CRL). Це здійснюється завдяки синхронізації ТБЧ між вузлами мережі, де будь-яка модифікація стану сертифіката буде миттєво повідомлена на всі вузли [34].

Іншим важливим аспектом в контексті безпеки Інтернету є те, що ІВК, що базується на БЧ, надає гнучкий захист від атак "людина посередині" (MITM). Традиційно MITM розглядається як серйозний ризик для безпеки, що передбачає, що зловмисник може захопити з'єднання веб-браузера для певних веб-сайтів, представивши дійсний сертифікат (тобто підроблений відкритий ключ) для цього домену. Для користувачів і веб-браузерів важко визначити заміну сертифіката у випадку, якщо зв'язаний ЦСК був зламаний зловмисником [35]. Підхід ІВК, що заснований на БЧ, робить атаки MITM практично неможливими. Це пояснюється тим, що коли ЦСК публікує або відкликає відкритий ключ веб-сайту/домену на БЧ, інформація буде розподілена по тисячам вузлів. Таким чином, порушення відкритого ключа буде (теоретично) поза питанням [36]. Традиційна ІВК усуває ризики MITM шляхом вбудовування сертифікатів кореневого ЦСК в інсталяцію браузера, таким чином штучно розширюючи бар'єри входу ЦСК і збільшуючи час, необхідний для відкликання сертифіката кореневого ЦСК.

**Оцінка та експериментальні результати**

Нижче подаються результати, що стосуються оцінки продуктивності, складності (вартості) функцій, заснованих на БЧ та обмеження щодо платформи ІВК на основі ТБЧ [26].

Продуктивність. Щоб визначити ефективність ІВК на основі смарт-контрактів Ethereum, в [26] проведено ряд експериментів на відкритому Ethereum Testnet (Rinkeby) (https://www.rinkeby.io). Сутність його в перевірці сертифікатів ЦСК по повному шляху від листа (даний сертифікат ЦСК) до кореня (ланцюг довіри). У результаті експерименту зроблено порівняння продуктивності між перевіркою на основі смарт-контракту та перевіркою коду Golang служби Restful.

Перевірка служби Restfu. Спочатку була зроблена перевірка сертифікату, хоча повний ланцюг довіри був заснований на службі Restful. Вона отримує сертифікат для кожного ЦСК з БЧ, аналізує сертифікат з бібліотеками Golang для вилучення адреси смарт-контракту батьківського ЦСК і потім перевіряє дійсність сертифікату на основі відповідного ґеш-значення, що зберігається у смарт-контракті батьківського CA.

Перевірка на основі смарт-контракту. Альтернативний підхід, що застосований, виявляється більш значно ефективнішим. Ідея його полягає в тому, що спеціальний смарт-контракт читає і аналізує сертифікати ЦСК, що зберігаються в БЧ виключно за допомогою коду Solidity та компілятора для смарт-контракту Ethereum. Зокрема, оскільки смарт-контракт не змінює ТБЧ, перевірка здійснюється «безкоштовно».

Із рис. 3 видно, що хоча продуктивність смарт-контрактів може бути менш вражаючою порівняно з криптографічними бібліотеками Golang, заснованими на відносно коротких ланцюгах довіри (менше 400 під-ЦСК), починаючи від ланцюга довіри довжиною, що перевищує 500 під-ЦСК, продуктивність перевірки на основі смарт-контракту вище, ніж у коду Golang.

Наприклад, для ланцюга довіри з довжиною близько 1100 під-ЦСК перевірка на основі смарт-контракту тривала приблизно 7 с, тоді як для коду Golang з використанням криптографічних бібліотек Golang було потрібно майже 15 с. У [26] для експериментів була використана стандартна робоча станція DELL з процесором Intel Core i7, але з оперативною пам'яттю 32 Гб.

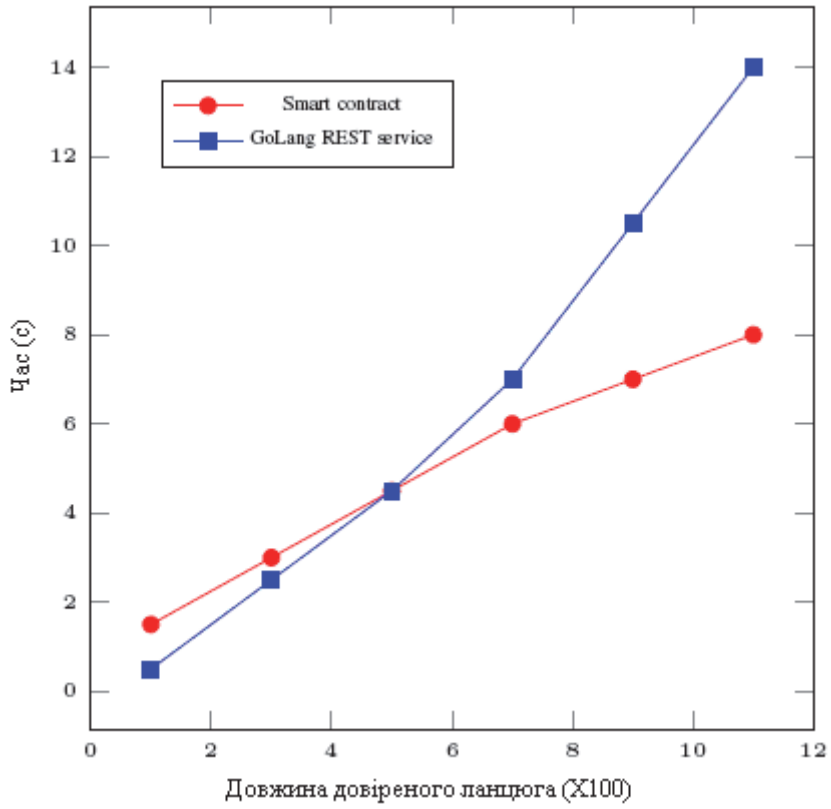


Рис. 3. Перевірка ланцюга довіри сертифіката CA

**Витрати на запуск нового ЦСК в Ethereum**

*Витрати на запуск*. Як показано в [26], витрати на підтримку ІВК є, можливо, важливою перевагою рішень ІВК на основі ТБЧ. Якщо припустити, що кінцеві користувачі все одно будуть використовувати локальну копію загальнодоступного БЧ, то усі витрати в основному складаються з плати майнерам, які підтверджують запис даних в БЧ.

Виходячи з експериментів з тестовим загальнодоступним ТБЧ Rinkeby [26], запуск ЦСК, (включаючи створення порожнього смарт-контракту, завантаження сертифікату в цей смарт-контракт, запис ґеш-значення цього сертифіката в смарт-контракт батьківського CA тощо) коштує 0,07 Ethers, що при нинішній вартості Ethers близько 700 USD за Ether переводиться у 50 USD за сертифікат ЦСК.

Можливо ініціювання звичайного сертифіката кінцевого суб'єкта, що передбачає лише записування його ґеш-значення у смарт-контракт батьківського ЦСК, призводить до набагато менших витрат у розмірі 7-10 USD за сертифікат. З огляду на поточну ціну на вихідний річний сертифікат кінцевого суб'єкта в розмірі декількох сотень доларів, витрати на сертифікат ТБЧ, схоже, не перевищують витрати, витрачені на існуючу інфраструктуру ЦСК.

*Обмеження та проблеми створення та застосування.* По-перше, загальнодоступні БЧ характеризуються значним збільшенням розміру БЧ, що повторюється на всі вузли, які беруть участь в системі. Особливо це стосується Ethereum та інших подібних платформ з підтримкою смарт-контрактів, які важливі для організації ефективної ІВК. Наприклад, у грудні 2017 р. розмір Ethereum ChainData з FAST Sync досяг 38,89 Гб порівняно з 20,46 Гб у вересні 2017 р. (https://etherscan.io/chart2/chaindatasizefast) [62].

По-друге, велика волатильність криптовалют призводить до певної невизначеності витрат на завантаження/оновлення сертифікатів як в довгостроковій, так і в короткостроковій перспективі. Інакше кажучи, вартість роботи БЧ безпосередньо пов'язана з ціною відповідних криптовалют, таких як Ether. Наприклад, у травні 2017 р. ціна Ефіру склала 85,43 доларів (https://bitcoinmagazine.com/price/), тоді як у грудні 2017 р. ціна досягає 729,01 доларів, що має на увазі зростання вартості операцій з ТБЧ в вісім разів за сім місяців.

По-третє, якщо для перевірки сертифіката використовується синтаксичний аналіз смарт-контракту з кодом Solidity, а не зовнішнім модулем Golang (тобто служби Restful), то виникають обмеження у використанні ґеш-функцій і асиметричних криптографічних функцій, доступних для смарт-контрактів. Наприклад, для Ethereum без попередньої обробки даних можна використовувати тільки SHA256 як ґеш-функцію і підписи ECDSA на основі криптографії еліптичних кривих.

Нарешті, оскільки права доступу для модифікації даних сертифікату засновані на системі облікових записів платформи БЧ, то втрачений пароль призводить до безповоротно втраченого доступу до облікового запису. Деяким рішенням проблем ЦСК, які втрачають свій пароль доступу до ТБЧ, може бути організація порожнього смарт-контракту і копіювання всіх даних зі старого смарт-контракту в новий, так як теоретично будь-який смарт-контракт завжди доступний для читання будь-кому. Очевидно, що створення нового смарт-контракту ЦСК може призвести до перевидання сертифіката ЦСК (принаймні в поточній реалізації), оскільки сертифікати ЦСК можуть містити посилання на відповідний смарт-контракт.

**Нова модель ІВК з прозорістю сертифікатів на основі БЧ**

*Загальні положення та стан.* У традиційних ІВК ЦСК вважаються повністю довіреними. Однак на практиці абсолютна відповідальність ЦС за забезпечення надійності викликала серйозні проблеми безпеки інформації та кібербезпеки. Щоб запобігти подібним проблемам, компанія Google у 2013 р. впровадила концепцію прозорості сертифікатів (ПС). Пізніше для зниження рівня довіри до ЦС запропоновано кілька нових моделей ІВК. Наприклад, підзвітна інфраструктура ключа (ПІК), стійка до атак інфраструктура відкритого ключа (CAIBK) та розподілена прозора інфраструктура ключа (РПІК). Проте, всі ці пропозиції все ще є вразливими до атак розділеного цифрового світу, якщо порушник здатний реалізувати різні плани загроз на журнал. У роботі [25] щоб усунути атаки розділеного світу та забезпечити ідеальну прозорість сертифікації та відкликання сертифікатів пропонується нова архітектура ІВК з прозорістю сертифікатів, заснована на БЧ, яка була названа CertLedger. Всі сертифікати TLS (Transport Layer Security), їх статус відкликання, весь процес відкликання та довірене управління ЦСК здійснюються в CertLedger. CertLedger надає унікальний, ефективний і надійний процес валідації сертифікату, що виключає звичайні непридатні та несумісні процеси сертифікації, що реалізуються різними постачальниками програмного забезпечення. Клієнти TLS в CertLedger також більше не вимагають перевірку достовірності та зберігання довірених сертифікатів ЦСК.

Як стандарт де-факто, сертифікати SSL/TLS використовуються для забезпечення послуг автентичності, цілісності та конфіденційності для цих існуючих додатків. Ці сертифікати видаються ЦСК, які вважаються довіреними організаціями у звичайних системах ІВК. Зокрема, очікується, що ЦСК діють згідно з деякими правилами, які позначені як документи про сертифікаційну політику (Certificate Policy – CP) та заява про практики сертифікатів (Cate Practice Statement – CPS). У поточній моделі довіри ЦСК мають абсолютну відповідальність за видачу правильних сертифікатів для визначеного суб'єкта. Проте ЦСК все ще можуть бути скомпрометовані та підроблені, або чинні сертифікати можуть бути видані через неналежну практику безпеки інформації або невідповідність CP та CPS. Необхідно відмітити, що протягом останнього десятиліття відбулися серйозні інциденти, які наведені вище в розділі 2 цієї статті. Вказані фатальні випадки призводять до того, що багато досліджень здійснюють абсолютну довіру до ЦСК на декілька органів. Для виявлення [25] підроблених, але дійсних сертифікатів TLS, закріплення ключа, краудсорсингу, донесення до браузерів інформації про відкликання є початковими рішеннями, які частково реалізовані, але зазнали невдачі через проблеми масштабування.

*Розподілена прозора інфраструктура ключа (РПІК)* [25]. Інфраструктура РПІК визначає загальнодоступну архітектуру управління сертифікатами, яка зменшує вразливості та, як наслідок, запобігає використанню підроблених сертифікатів та виявляється стійкою, навіть,

якщо всі постачальники послуг діють у домовленості [37]. Здійснюється обслуговування журналу сертифікатів (ОЖС) та обслуговування журналу відображення (ОЖВ) – двох нових об'єктів, що введені у РПІК. Причому ОЖС зберігають всі дійсні, відкликані та не чинні сертифікати для набору доменів і надають докази щодо їх існування або відсутності. Також ОЖВ підтримує зв'язок між набором доменних імен та ОЖС, які підтримують журнали для них. «Дзеркала» підтримують повну копію даних, що зберігаються як в ОЖС, так і ОЖВ. Причому ЦСК здійснюють перевірку ідентифікаторів та видають сертифікати, але вони не є єдиними суб'єктами, що забезпечує довіру при підключенні до домену. Якщо взяти концепцію «незалежного ключа», то домен володіє двома типами сертифікатів – сертифікатом TLS і майстер-сертифікатом, який використовується для запиту нового сертифіката TLS від ЦСК, і реєстрації його в ОЖС. Користувачі або, зокрема, браузери спершу роблять запит до ОЖВ для того, щоб знайти правильний ОЖС для конкретного домену. Для прийняття рішення про підключення перші доведення, отримані від ОЖВ, перевіряються, далі робиться запит до ОЖС для того, щоб отримати докази для сертифікату TLS домену.

У РПІК передбачається, що всі майстер-сертифікати є справжніми, і випуск підроблених сертифікатів не є вірогідним, оскільки ЦСК працюють на підприємствах, які не можуть втратити репутацію. Однак це не є дійсним аргументом, оскільки більшість підроблених сертифікатів генеруються через відсутність належного контролю або процесів безпеки. А саме, якщо ЦСК і ОЖС піддаються компрометації, РПІК не зможе запобігти випуску підроблених майстер та TLS сертифікатів. З цієї точки зору, порушник, який контролює ОЖС та здатний здійснювати підробки, використовуючи дійсні майстер або TLS сертифікати, може зробити атаку розділеного цифрового світу. На жаль, цю атаку не можна виявити, оскільки в РПІК не має процесу моніторингу через припущення про справжні сертифікати.

*Прозорість сертифікату та відкликання на основі ТБЧ*. Ванг та інші запропонували прозорість сертифікації та відкликання на основі БЧ для зберігання сертифікатів TLS та їх статусу відкликання [38]. Коротше кажучи, у цій схемі веб-сервери публікують свої сертифікати TLS у БЧ, використовуючи їхні пари ключів публікації, які використовуються для підписання операцій. Ці пари ключів видачі відрізняються від пари ключів у сертифікаті та спочатку засвідчуються певним набором веб-серверів, які вже існують в БЧ. У цій схемі транзакції мають термін дії, тому сертифікати TLS та їх статус відкликання додаються до БЧ періодично протягом свого терміну дії. Під час рукостискання TLS веб-сервер посилає транзакцію сертифікату та свій шлях аудиту Мерклі до клієнта TLS, який перевіряє його дійсність через свої локально збережені заголовки синхронізованих блоків.

Однак ця пропозиція має такі недоліки. Вона має ненадійну основу для забезпечення надійності ключів публікації. А саме, «сильний порушник», який може отримати підроблені, але дійсні сертифікати TLS від пошкоджених ЦСК, може заздалегідь створити деякі фіктивні домени (тобто веб-сервери) і може використовувати їх для створення дійсного підпису транзакції пари ключів публікації. Ця проблема виникає через довіру до веб-серверів. Автори [25] пропонують вирішити цю проблему, створивши більше публічно-довірених ЦСК, щоб визнати недійсними підроблені транзакції. Однак тоді постає питання рівня довіри, яке явно не уточнюється. Причому, прозорість відкликання покладається на ЦСК, щоб опублікувати дані про анулювання сертифікатів TLS на ТБЧ. Однак скомпрометовані або непрацюючі ЦСК не можуть видавати СВС або давати відповідь клієнту в зазначений час.

Щодо атак людина-посередині, коли порушник здатен переконати клієнта у незавершеній транзакції відкликаного TLS сертифікату, то під час рукостискання TLS веб-сервери передають клієнту TLS транзакцію сертифіката для підтвердження сертифікату TLS. Клієнт TLS приймає цю транзакцію, якщо термін її дії не закінчився, і додає її до затвердженого блоку. Однак відкликаний або оновлений TLS сертифікат також може мати незавершену транзакцію у ТБЧ. Тому, як тільки порушник надсилає цю незавершену транзакцію сертифіката зі своїми доказами Мерклі клієнту TLS, він приймається під час рукостискання TLS.

Клієнти TLS не можуть виявити остаточний стан сертифікату, оскільки клієнти лише перевіряють наявність транзакції у відповідному блоці.

Така пропозиція, з точки зору витрат на зберігання, також є неефективною. По-перше сертифікат TLS додається до ТБЧ періодично протягом його терміну дії, по-друге CBC може бути доданий до ТБЧ для кожного відкликаного сертифікату (тобто кількість вставок CBC до ТБЧ дорівнює кількості відкликаних сертифікатів), а по-третє пара ключів публікації додається до ТБЧ періодично. Крім того, він має заголовки великого розміру, які містять імена DNS, існуючі в транзакціях блоку.

Як публічний ТБЧ вирішує проблеми ІВК. По суті БЧ – це спільний, незмінний, децентралізований відкритий журнал, що містить постійно зростаючий список блоків. Блок – це структура даних, яка складається з заголовку і списку транзакцій. Кожен блок пов'язаний з попереднім блоком за рахунок криптографічного ґеш-значення, тому блоки по своїй суті захищені від підробки та перегляду. Мережа ТБЧ – це децентралізована однорангова (P2P) мережа, що складається з повних вузлів і легких вузлів. Повні вузли зберігають копію ТБЧ, перевіряють і розповсюджують нові транзакції і блоки по всій мережі, тоді як легкі вузли зберігають тільки заголовки блоків. Всі вузли можуть створювати транзакції для зміни стану ТБЧ. Нові блоки транзакцій колективно перевіряються і додаються до існуючого ланцюга відповідно до розділеного механізму консенсусу.

У [25, 39] показано, що БЧ вирішує такі проблеми ІВК:

1) атаку розділеного світу;

2) проблеми з відкликанням та перевіркою сертифікатів;

3) проблеми управління довіреними сертифікатами/сховищами ключів.

Характеристики ТБЧ для ІВК. Використовуючи послідовність прийняття рішень Wust і Gervais [25], можна визначити тип БЧ для керування журналом сертифікації. Будемо вважати, що записувач – це сутність, яка здатна накопичувати нові транзакції в новий блок і додавати його в ТБЧ.

Для повноти викладення необхідно дати відповіді га такі питання:

1. Чи потрібно зберігати стан сертифікатів?

Сертифікати TLS постійно оновлюються через закінчення терміну дії або скасування. Стан сертифікатів повинен зберігатися та оновлюватися тоді, коли це необхідно.

2. Чи існують кілька авторів?

Сертифікати TLS, створені довіреними ЦСК, додаються до журналу. Окремий підроблений записувач може додавати підроблений, але дійсний сертифікат до журналу, затримувати або ігнорувати додавання справжніх. Тому збільшення децентралізації під час запису до журналу зменшить ризик підробок через те, що широка участь записувачів призведе до більш надійного і стійкого журналу.

3. Чи можете ви використовувати третю довірену сторону (ТДС), яка завжди онлайн?

Сильний супротивник може керувати будь-якою ТДС, що може призвести до єдиної точки відмови. Прийняття онлайн-ТДС є основним джерелом вразливості.

4. Чи відомі всі записувачі?

Записувачі можуть бути відомі або невідомі. Проте, якщо вони відомі, їх слід вибирати та розганяти по всьому світу таким чином, щоб їх зловмисна співпраця та маніпуляції не могли бути можливими.

5. Чи усі записувачі є довіреними?

Незважаючи на те, що всі записувачі, здається, є довіреними, деякі з них можуть контролюватися сильним зловмисником.

6. Чи потрібна громадська перевірка?

Статус існування та дійсності всіх сертифікатів TLS повинен бути перевірений громадськістю для досягнення повної прозорості. Таким чином, блок-схема рішень призводить до ТБЧ без дозволу або публічного ТБЧ з дозволом для керування журналами сертифікації. Однак ТБЧ вимагає наступні додаткові можливості. Перш за все, він повинен містити інфра-

структуру смарт-контрактів для реалізації необхідних правил перевірки переходу станів. По-друге, базовий механізм консенсусу не повинен призводити до тимчасових розгалужень, оскільки деякі клієнти TLS можуть перевірити неправильний стан сертифіката TLS до того, як блоки будуть повністю підтверджені. По-третє, час, необхідний для підтвердження нового блоку в механізмі консенсусу, не повинен бути високим, щоб транзакції могли змінювати стан сертифікатів TLS за прийнятний період часу. Нарешті, архітектура БЧ повинна дозволяти клієнтам TLS перевірити остаточний стан сертифікатів TLS і ефективно генерувати докази Мерклі. Відмітимо, що стан дерев Мерклі в основному підтримується для ефективного вироблення підтвердження та слідкування за кінцевими станами активів. Корінь Мерклі цього дерева зберігається в заголовках блоків, тому цілісність дерева і створені з нього докази стану можна перевірити [16 – 18].

Необхідно відмітити, що CertLedger можна розгорнути на існуючій архітектурі БЧ, що задовольняє вимогам [25], як у Ethereum, Neo та Ontology. У цих архітектурах можна обрати будь-який механізм консенсусу, який не призведе до тимчасових розгалужень, таких як PBFT та DBFT [25].

Призначення та сутність CertLedger. CertLedger – це архітектура ІВК для перевірки, зберігання та анулювання сертифікатів TLS та керування довіреними сертифікатами ЦСК у публічному БЧ. Вона спрямована на те, щоб забезпечити більш прозорий життєвий цикл видачі та відкликання сертифікатів та усунути будь-які види атак «суб'єкта посередині». Більш того, він також має на меті уніфікувати процес валідації сертифікатів для всіх клієнтів TLS, оскільки реалізації різних клієнтів TLS не є узгодженими та відповідними.

CertLedger управляє функціями ІВК через об'єкти стану. Об'єкт стану – це цифровий документ, який складається з даних і незмінного коду смарт-контракту для управління ним. Кожен об'єкт стану має унікальну адресу в БЧ. Державні зміни активів ініціюються операціями та відстежуються через державні об'єкти. CertLedger, як правило, містить наступні об'єкти стану.

*Об'єкт доменного стану* зберігає та управляє станами всіх сертифікатів TLS та їх статусом відкликання. Цей об'єкт стану містить необхідний код для перевірки сертифіката TLS відповідно до міжнародних стандартів, таких як RFC 5280 [40]. Він використовує Об'єкт Стану Довіреного ЦСК, при побудові надійного шляху для сертифікату TLS. Крім того, він також містить необхідний код для зміни статусу сертифіката TLS на «відкликаний». Його смарт-контракт перевіряє сертифікат TLS, додаючи до CertLedger у наступній послідовності перевірки:

- чи сертифікат вже додано;
- чи є діючим сертифікат;
- чи сертифікат відповідає профілю сертифіката TLS;
- чи підписаний сертифікат одним із сертифікатів ЦСК в Довіреному ЦСК;
- чи зберігати новий сертифікат TLS у об'єкті стану домену;
- чи встановлено його статус скасування як «не відкликано».

**Проблемні питання сучасних та перспективних ІВК на базі БЧ**

*Проблеми сучасних ІВК.* Проведений аналіз показав, що стосовно сучасних ІВК з ЦСК існують наступні проблеми:

1) Єдина точка збою. ЦСК несуть повну відповідальність за відкликання сертифікатів та надання послуг по їх скасуванню. Якщо ЦСК несправний або скомпрометований, то вся система стає під загрозу, а по суті компрометується.

2) Необхідність застосування третьої довіреної сторони. Причому користувачі систем повинні довіряти ЦСК, адже він відповідає за генерацію та управління відкритими ключами користувачів. В разі компрометації ЦСК існує високий ризик безпеки системи, що використовує сертифікати відкритих ключів.

3) Висока вартість та неефективність управління ключами при великій кількості значно розподілених додатків з багатьма користувачами.

4) Проблемою для ІВК з Мережею довіри є те, що існує бар'єр для додавання нових користувачів, так як: нові користувачі повинні мати довіру у вже зареєстрованих користувачів. Обидва типи ІВК мають недолік – не можна сховати ідентичність або відкритий ключ зареєстрованої особи [25].

Переваги ІВК на основі ТБЧ. ІВК на основі ТБЧ у порівнянні з традиційними PKI мають такі переваги:

1) Швидкість та простота перевірки сертифікату та ланцюга сертифікатів ЦСК. Сертифікати не потрібно підписувати, це означає, що вони коротші та потрібно менше часу на передачу сертифікату повернутого ланцюгом сертифікатів ЦСК.

2) Немає потреби у сервісі, який видає списки відкликаних сертифікатів (Certificate Revocation Lists, CRL), що є відомою з перших років застосування проблемою традиційних ІВК. CRL можуть бути дуже великими, та повинні зберігатися верифікатором та постійно оновлюватися по всій мережі. Таке спрощення здійснюється завдяки ТБЧ-синхронізації між вузлами мережі, де про будь-яку зміну стану сертифіката негайно повідомляється всім вузлам [62].

3) Немає потреби відповідати на запити протоколу онлайн статусу сертифікатів (online certificate status protocol, OCSP). Перевірки OCSP додають затримки в мережі для підтвердження сертифікатів та викривають інформацію про те, що суб'єкт подає верифікатору сертифікат. При цьому відбувається спостереження з криптографічними можливостями (правами) [64].

4) ІВК на БЧ можна використовувати для резервування простих сертифікатів БЧ так само, як і дорогих сертифікатів БЧ, і обидва випадки виграють від вищезазначених переваг.

5) Ще одним важливим аспектом в контексті безпеки в Інтернеті є те, що ІВК на основі БЧ забезпечує гнучкий захист від атак «людина по середині» (MITM).

*Проблемні питання ІВК на базі БЧ*. Технологія ТБЧ має ряд недоліків, які маються також внесені, у тому числі у ІВК на базі БЧ. Серед проблемних питань можна відзначити наступні:

1) Розмір відкритих БЧ постійно зростає, що розповсюджується на всі вузли, які беруть участь в системі. Особливо це актуально для платформ БЧ з підтримкою смарт-контрактів, які є важливими для ефективної організації ІВК.

2) Вартість ТБЧ-операцій напряму залежить від ціни відповідної криптовалюти.

3) Існує таке обмеження. Якщо для перевірки сертифікату використовується аналіз смарт-контракту з кодом Solidity, а не зовнішній модуль Golang, то може з'явитись обмеження на використання ґеш-функцій та асиметричних криптографічних функцій, доступних для смарт-контрактів. Наприклад, Ethereum без попередньої обробки даних може використовувати в якості ґеш-функції лише SHA256 та ЕП з використанням ECDSA на основі криптографії на еліптичних кривих.

4) Втрата паролю призводить до безповоротної втрати доступу до облікового запису. Рішенням проблем втрати паролю доступу до ЦС може стати створення порожніх смарт-контрактів та копіювання до них усіх даних до нового смарт-контракту.

**Стійкість ІВК на основі БЧ до атак**

1) Атака розділеного цифрового світу

У ІВК на основі публічних журналів «сильний» порушник, який має можливість контролювати довірені об'єкти типу ЦСК, та оператор журналу, може застосовувати атаки розділеного цифрового світу, надаючи різний вигляд журналів цільовим жертвам [24]. Хоча деякі з цих пропозицій не можуть виявити цю атаку, інші пропонують використовувати швидкий моніторинг для виявлення атак. А саме, протоколи домовленості для її виявлення, шляхом почергового перегляду журналу для клієнтів TLS та серверів [25, 26].

Тим не менш, ця атака може бути виявлена лише в тому випадку, якщо:

- є достатня кількість клієнтів і серверів TLS у домовленості взаємодії;
- принаймні деякі з них можуть переглядати справжній журнал і вимагати від журналу підтвердження узгодженості.

2) Суб'єкт (людина) посередині (MITM). Зазвичай MITM вважається головним ризиком для безпеки, коли зловмисник перешкоджає підключенню веб-браузера до певних веб-сайтів, надаючи дійсний сертифікат (тобто підроблений відкритий ключ) для цього домену. Для користувачів та веб-браузерів важко визначити заміну сертифіката у випадку, якщо зловмисник зламав відповідний ЦСК [25, 26].

Аналіз показав, що атака MITM є можливою коли порушник здатен переконати клієнта у незавершеній транзакції відкликаного TLS сертифікату. Більш конкретно, під час рукостискання TLS веб-сервери передають клієнту TLS транзакцію сертифіката для підтвердження сертифікату TLS. Клієнт TLS приймає цю транзакцію, якщо термін її дії не закінчився, і додає її до затвердженого блоку. Однак відкликаний або оновлений TLS сертифікат також може мати незавершену транзакцію у БЧ. Як тільки порушник надсилає цю незавершену транзакцію сертифіката зі своїми доказами Мерклі клієнту TLS, він приймається під час рукостискання TLS. Клієнти TLS не можуть виявити остаточний стан сертифікату, оскільки клієнти лише перевіряють наявність транзакції у відповідному блоці.

Підхід до побудови ІВК на основі БЧ робить атаки MITM практично неможливими, оскільки коли ЦСК публікує або робить відкликання відкритого ключа веб-сайту/домену в БЧ, інформація буде поширюватися на тисячі вузлів, тому підробка відкритого ключа (теоретично) не можлива [16, 17]. Традиційна ІВК вирішує ризики MITM, додаючи кореневі сертифікати ЦС до налаштування браузера, тим самим штучно розширюючи вхідні бар'єри ЦС та збільшуючи час, необхідний для відкликання кореневого сертифіката ЦСК [16, 17].

**Висновки**

1. При застосуванні принципу децентралізації стосовно існуючих ІВК необхідно поліпшити хоча б одну із важливих для цільового застосування таких характеристик як вартість, складність (часова та просторова), швидкість, прибутковість, безпека (загальна та інформаційна), анонімність, прозорість, гнучкість тощо.

2. Цільовий параметр, за яким здійснюється покращення ІВК має бути суттєвим, причому децентралізація повинна покращувати хоча б один важливий для користувачів параметр, як мінімум в 2-3 рази.

3. Нова технологія, в даному випадку технологія ІВК з використанням ТБЧ, не повинна істотно програвати існуючим ІТ за іншими важливими параметрами. Вважається, що у цілому нова ІВК повинна бути в чомусь краще в 2-3 рази, а за всіма іншими параметрами програвати не більше, ніж в 1,5 рази.

4. «Кращість» і перевага нової ІВК на основі ТБЧ мають суб'єктивний характер, вимоги до них повинні формуватись скоріше всього користувачами, і в меншій мірі розробниками.

5. У більшості випадків кращість може визначатись рівнем продаж таких ІТ. Відсутність певних продаж показує, що явної переваги цільової аудиторії відносно нової чи удосконаленої ІВК не визнано.

6. Одним із важливих та необхідних додатків ТБЧ є, по суті, удосконалення інфраструктури відкритого ключа (ІВК) на основі використання при її побудові принципів децентралізації та прозорості тощо.

7. Відомий ряд фатальних випадків, які приводили до того, що багато досліджень розподіляють абсолютну довіру до ЦС на декілька органів. Для виявлення підроблених, але дійсних сертифікатів TLS можна, застосовувати закріплення ключа, краудсорсингу та доведення до браузерів інформації про відкликання тощо. Вказані початкові рішення, які частково реалізовані, на жаль зазнали невдачі через проблеми масштабування.

8. Основним аргументом обґрунтування застосування БЧ при побудуванні ІВК є те, що рішення, засновані на ТБЧ, можуть об'єднати переваги ІВК, засновані на журналах, та підходи WoT, а також вирішити деякі проблеми зі звичайної системи ІВК. Так, БЧ усуває потенційні точки відмови підходу, заснованого на ІВК на основі журналу, і проблеми розгортання. З іншого боку, підхід, що заснований на БЧ, пом'якшує потреби WoT у нових власниках сертифікатів, щоб довести достовірність існуючих членів мережі, а також пом'якшує вимоги WoT для нових власників сертифікатів, щоб довести достовірність існуючих членів мережі.

9. Структура ІВК на основі БЧ підтримує відкликання сертифікату, що є суттєвою проблемою в традиційних системах ІВК. Також, оскільки неможливо видалити інформацію з БЧ, то тільки «батьківський» ЦСК може позначати виданий ним сертифікат як відкликаний. Тобто, будь-яка неправильна поведінка ЦСК стосовно відкликання сертифікату буде також простежена і помічена всіма іншими суб'єктами.

10. ІВК на основі БЧ перед традиційною ІВК має наступні переваги:

- перевірка сертифікату і його ланцюга сертифікатів ЦСК проста і швидка;
- ІВК на основі ТБЧ вирішує давню проблему традиційних ІВК, не вимагаючи використання сервісу, який видає списки відкликання сертифікатів (CRL).

11. Перевірка ланцюга довіри сертифіката на основі смарт-контракту виявляється більш значно ефективнішою. Ідея його полягає в тому, що спеціальний смарт-контракт читає і аналізує сертифікати ЦСК, що зберігаються в БЧ виключно за допомогою коду Solidity та компілятора для смарт-контракту Ethereum.

12. Виходячи з експериментів з тестовим загальнодоступним ТБЧ Rinkeby [26], запуск ЦСК, включаючи створення порожнього смарт-контракту, завантаження сертифікату в цей смарт-контракт, запис ґеш-значення цього сертифіката в смарт-контракт батьківського СА тощо, коштує 0,07 Ethers, що при нинішній вартості Ethers близько 700 USD за Ether переводиться у 50 USD за сертифікат ЦСК.

13. Система CertLedger управляє функціями ІВК через об'єкти стану. Об'єкт стану – це цифровий документ, який складається з даних і незмінного коду смарт-контракту для управління ним. Кожен об'єкт стану має унікальну адресу в БЧ. Державні зміни активів ініціюються операціями та відстежуються через державні об'єкти.

14. Стосовно сучасних ІВК з ЦСК існують наступні проблеми: єдина точка збою; необхідність застосування третьої довіреної сторони; висока вартість та неефективність управління ключами при великій кількості значно розподілених додатків з багатьма користувачами; бар'єр для додавання нових користувачів, так як нові користувачі повинні мати довіру у вже зареєстрованих користувачів.

15. ІВК на основі ТБЧ у порівнянні з традиційними PKI мають такі переваги: швидкість та простота перевірки сертифікату та ланцюга сертифікатів ЦСК; немає потреби у сервісі, який видає списки відкликаних сертифікатів; немає потреби відповідати на запити протоколу онлайн статусу сертифікатів (online certificate status protocol, OCSP); ІВК на БЧ можна використовувати для резервування простих сертифікатів БЧ так само, як і специфічних сертифікатів БЧ.

16. ІВК на основі БЧ забезпечує гнучкий захист від атак «людина посередині».

17. Серед проблемних питань можна відзначити такі: розмір відкритих БЧ постійно зростає; якщо для перевірки сертифікату використовується аналіз смарт-контракту з кодом Solidity, а не зовнішній модуль Golang, то може з'явитись обмеження на використання ґеш-функцій та асиметричних криптографічних функцій, доступних для смарт-контрактів; втрата паролю призводить до безповоротної втрати доступу до облікового запису.

18. Ця стаття носить характер первинного загальносистемного аналітичного огляду та відображає погляди авторів на можливості та необхідність створення ІВК з використанням ТБЧ. Ця наша впевненість ґрунтується на тому, що діюча ІВК України практично реалізована та підтримується при експлуатації [16 – 18].

**Список літератури:**


1. Andreas M. Antonopoulos Mastering Bitcoin: Unlocking Digital Cryptocurrencies /Andreas M. Antonopoulos Kyiv : NGITS, 2014. С. 10 – 150.

2. Що таке децентралізований додаток? [Електронний ресурс]. Режим доступу: https://www.coindesk.com/information/what-is-a-decentralized-application-dapp.

3. Don Tapscott, Alex Tapscott Blockchain Revolution: How the Technology Behind Bitcoin is Changing Money, Business, and the World / Don Tapscott, Alex Tapscott Blockchain. Kyiv : Information Systems, 2016. С. 65 – 102.

4. 20 основних застосувань БЧ [Електронний ресурс]. Режим доступу: https://biznesmodeli.ru/blockhain-chto-eto-cases-crypto-top10/.

5. БЧ: атаки, безопасность и криптография [Електронний ресурс]. Режим доступу: https://www.securitylab.ru/blog/personal/ Informacionnaya_bezopasnost_v_detalyah/343072.php.

6. Распределённые реестры и информационная безопасность: от чего защищает БЧ [Електронний ресурс]. Режим доступу: https://habr.com/company/bitfury/blog/341902/.

7. Pavan Duggal Blockchain Contracts and Cyberlaw / Pavan Duggal. Kyiv : Information Systems, 2015. С. 15–39.

8. Tim Harris. Bitcoin: Mastering Bitcoin & Cyptocurrency for Beginners – Bitcoin Basics, Bitcoin Stories, Dogecoin, Reinventing Money & Other Digital Currencies. Kyiv : Economic, 2016. С. 30–47.

9. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008. https://bitcoin.org/bitcoin.pdf.

10. NISTIR 8202 – Blockchain Technology Overview. 2018, 68 p. Access mode: https://doi.org/10.6028/NIST.IR.820210.

11. Возможные атаки на функции хэширования [Електронний ресурс]. Режим доступу: https://studfiles.net/preview/2157418/page:2/.

12. Прикладна криптологія. Теорія. Практика. Застосування : монографія / І.Д. Горбенко, Ю.І. Горбенко. Харків, 2012. С. 340-347.

13. Алгоритмы шифрования – основа работы криптовалют [Електронний ресурс]. Режим доступу: https://tgraph.io/Algoritmy-shifrovaniya – osnova-raboty-kriptovalyut-09-27.

14. Blockchain 3.0 – 5 лучших проектов нового поколения: https://privatfinance.com/blockchain-3-0-5-luchshih-proektov-novogo-pokoleniya.

15. Comparison of cryptographic hash functions [Електронний ресурс]. Режим доступу: https://en.wikipedia.org/wiki/Compaison_of_cryptographic_hash.

16. Горбенко Ю.І., Горбенко І.Д. Інфраструктури відкритих ключів. Системи ЕЦП. Теорія та практика. Харків : Форт. 2010. 593 с.

17. Горбенко І.Д., Горбенко Ю.І. Прикладна криптологія. Харків : ХНУРЕ ; Форт, 2012. 1 та 2-е вид. 868 с.

18. Горбенко Ю.І. Побудування та аналіз систем, протоколів та засобів криптографічного захисту інформації ; за ред. І.Д. Горбенко. Харків : Форт. 2015. 902с.

19. Закон України «Про електронні довірчі послуги» // Відомості Верховної Ради (ВВР). 2017. № 45. ст.400.

20. Закон України «Про захист інформації в інформаційно-телекомунікаційних системах» від 31.05.2005 № 2594-IV.

21. Закон України «Про основні засади забезпечення кібербезпеки України» // Відомості Верховної Ради (ВВР). 2017. № 45, ст.403.

22. Регламент (ЄС) № 910/2014 Європейського Парламенту та Ради від 23 липня 2014 р. «Про електронну ідентифікацію та довірчі послуги для електронних транзакцій у межах внутрішнього ринку та про скасування Директиви 1999/93/ЄС» (1) (COM (2012) 0238-C7-0133/2012 – 2012/0146 (COD)).

23. ДСТУ 4145-2002 «Інформаційні технології. Криптографічний захист інформації. Цифровий підпис, що ґрунтується на еліптичних кривих. Формування та перевіряння».

24. ДСТУ ITU-TRec.X.509|ISO/IEC 9594-8:2015 «Інформаційні технології. Взаємозв'язок відкритих систем. Каталог: Основні положення щодо сертифікації відкритих ключів та сертифікації атрибутів».

25. CertLedger: A New IBK Model with Certificate Transparency Based on Блокчейн. Murat Yasin Kubilay, Mehmet Sabir Kiraz and Haci Ali Mantar. Access mode: https://eprint.iacr.org/2018/1071.pdf.

26. Yakubov Alexander A Blockchain-Based PKI Management Framework / Alexander Yakubov, Wazen M. Shbair, Anders Wallbom, David Sanda, Radu State // Access mode: https://orbilu.uni.lu/bitstream/10993/35468/1/blockchain-based-pki.pdf.

27. L. Axon and M. Goldsmith. PB-PKI: A privacy-aware blockchain-based PKI // Proceedings of the 14th International Joint Conference on e-Business and Telecommunications (ICETE 2017). Volume 4: SECRYPT, Madrid, Spain, July 24-26, 2017, 2017, pp. 311–318. [Electronic resource]. Access mode: https://doi.org/10.5220/0006419203110318.

28. C. Fromknecht, D. Velicanu, and S. Yakoubov. Certcoin: A namecoin based decentralized authentication system // Massachusetts Inst. Technol., Cambridge, MA, USA, Tech. Rep, vol. 6, 2014.

29. Mozilla included CA certificate list, 2017. [Electronic resource]. Access mode: https://www.mozilla.org/en-US/about/governance/policies/security-group/certs/.

30. D. Cooper. Internet x. 509 public key infrastructure certificate and certificate revocation list (crl) profile. 2008.
31. E. Androulaki, C. Cachin, A. D. Caro, A. Sorniotti, and M. Vukolic. Permissioned blockchains and hyperledger fabric // ERCIM News, vol. 2017, no. 110, 2017. [Electronic resource]. Access mode: https://ercim-news.ercim.eu/en110/special/permissioned-blockchains-and-hyperledger-fabric.
32. A. J. Nicholas Stifter and E. Weippl. A holistic approach to smart contract security // ERCIM News, vol. 2017, no. 110, 2017. [Electronic resource]. Access mode: https://ercim-news.ercim.eu/en110/special/a-holistic-approach-to-smart-contract-security/.
33. M. Ali, J. C. Nelson, R. Shea and M. J. Freedman. Blockstack: A global naming and storage system secured by blockchains // USENIX Annual Technical Conference, 2016. pp. 181–194.
34. K. Lewison and F. Corella. Backing rich credentials with a blockchain pki, 2016.
35. M. Alicherry and A. D. Keromytis. Doublecheck: Multi-path verification against man-in-the-middle attacks // Computers and communications, 2009. iscc 2009. ieee symposium on. IEEE, 2009, pp. 557–563.
36. Blockchain & cyber security. let's discuss, 2017. [Electronic resource]. Access mode: https://www2.deloitte.com/content/dam/Deloitte/ie/Documents/Technology/IECBlockchainandCyberPOV 0417.pdf.
37. Jiangshan Yu, Vincent Cheval, and Mark Ryan. DTKI: A new formalized PKI with verifiable trusted parties // The Computer Journal, 59(11):1695-1713, 2016.
38. B. Laurie, A. Langley, and E. Kasper. Certificate transparency. RFC 6962 (experimental), 2013.
39. Scott A Crosby and Dan S Wallach. Efficient data structures for tamper-evident logging // USENIX Security Symposium, pages 317-334, 2009.
40. RFC 5280: Internet X.509 public key infrastructure: certificate and CRL profile.

*Харківський національний*
*університет імені В.Н. Каразіна;*
*АТ «Інститут інформаційних технологій»;*
*Департамент Державної служби спеціального зв'язку*
*та захисту інформації України*

*Надійшла до редколегії 15.09.2019*