3. Шинкаренко К. В., Кориков А. М. Применение неслучайных стирающих кодов в IP телевещании // Томськ: «Известия Томского политехнического университета», 2008. – № 5. – С. 42–46.

Микосовський В.І.
студент,
Науковий керівник: **Полторак В.П.**
Кандидат технічних наук, доцент,
Національний технічний університет України
«Київський політехнічний інститут»

КРИПТОГРАФІЯ НА ЕЛІПТИЧНИХ КРИВИХ

Еліптична крива – набір точок, описаних рівнянням Вейерштрассе [1]:

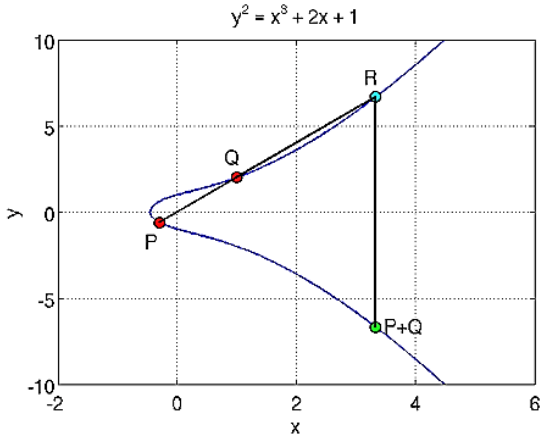
$$y^2 = x^3 + ax + b$$

Еліптичні криві поділяються на гладкі та сингулярні. Для гладких еліптичних кривих виконується наступна нерівність:

$$4a^3 + 27b^2 \neq 0$$

Тоді як для сингулярних кривих ця умова не виконується. В системах шифрування сингулярні криві зазвичай не використовуються, так як це значно знижує стійкість системи.

Арифметичні операції в еліптичній криптографії проводяться над точками кривої. Основною операцією являється «складання». Складання двох точок легко зобразити графічно:



Як видно з рисунка, для складання точок P і Q, необхідно провести між ними пряму лінію, котра обов'язково перетне криву в якій-небудь тертій точці R. Відобразимо точку R відносно горизонтальної осі координат і отримаємо шукану точку $P + Q$.

Алгебраїчне представлення «складання» запишемо складання двох точок в виді формули: $P + Q = -R$

Нехай координатам точки P будуть (x_p, y_p), а координатами точки Q відповідно (x_q, y_q). Обчислимо:

$$\alpha = \frac{y_q - y_p}{x_q - x_p}$$

І тоді координати точки $P + Q$ будуть рівні:

$$x_{p+q} = \alpha^2 - x_p - x_q$$

$$y_{p+q} = -y_p + \alpha\,(x_p - x_r)$$

Всі розглянуті вище криві відносяться до еліптичних кривих над дійсними числами. І це приводить до проблеми округлення. Тобто, використовуючи криві над дійсними числами, неможливо отримати бієкцію між вихідним текстом і зашифрованими даними. Щоб вирішити цю проблему, в криптографії використовуються тільки криві над скінченними полями. Це означає, що під еліптичною кривою підрозумовують набір точок, координати яких належать скінченному полю.

В криптографії розглядають два види еліптичних кривих: над скінченним полем Z_p – кільце різниць по модулю просто числа. І над полем $GF\,(2^m)$ – бінарне скінченне поле.

У еліптичних кривих над полем $GF\,(2^m)$ є одна важлива перевага, елементи поля $GF\,(2^m)$ можуть бути легко представлені в виді n-бітних кодових слів, а це дозволяє збільшити швидкість апаратної реалізації еліптичних алгоритмів. Всі математичні операції на еліптичних кривих над скінченним полем проводяться по законах скінченного поля, над яким побудована еліптична крива. Тобто для обчислення, наприклад, суми двох точок кривої E над кільцем різниць Z_p, всі операції проводяться по модулю числа p.

Втім, тут є певні проблеми. Якщо ми складемо два одинакових елемента із бінарного скінченного поля, то отримаємо в результаті 0, так як складання відбувається по подулю 2. Це означає, що характеристика такого поля рівна 2. Але еліптична крива виду $y^2 = x^3 + ax + b$ описана над полем характеристики 2 чи 3 стає сингулярною, а як уже зазначалось, такі криві застосовувати не рекомендується.

Тому над бінарним скінченним полем використовуються криві виду:

$$y^2 + xy = x^3 + ax^2 + b, b \neq 0$$

Ще одним важливим поняттям еліптичної криптографії являється порядок еліптичної кривої, який показує кількість точок кривої над скінченним полем.

Теорема Хассе стверджує [2, с. 105], що якщо N – кількість точок кривої, визначеної над полем Z_q з q елементами, тоді справедлива рівність:

$$|N - (q + 1)| \leq 2\sqrt{q}$$

Так як бінарне скінченне поле $GF\,(2^n)$ складається з 2^n елементів, ми можемо сказати, що порядок кривої $E_{2n}\,(a, b)$ рівний $2^n + 1 - t$, де $|t| \leq \sqrt{2^n}$. З числом t пов'язане наступне визначення: еліптична крива над бінарним скінченним полем називається суперсингулярною, якщо t ділиться на характеристику поля (в випадку бінарного поля характеристика рівна 2) без

остачі. Використовувати суперсингулярні криві в криптографії, також не можна, як і сингулярні.

Точки еліптичної кривої над скінченним полем являють собою групу. І як було сказано вище, для цієї групи визначена операція складання. Відповідно ми можем представити множення числа k на точку G як $G+G+\cdots+G$ з k доданками. Тепер уявимо, що у нас є повідомлення M представлене в виді цілого числа. Ми можемо зашифрувати його, використовуючи вираз $C=G*M$. Питання в тому, наскільки складно відновити M знаючи параметри кривої $E\ (a,b)$, шифротекст C і точку G.

Дана задача називається дискретним лографимом на еліптичній кривій і не має швидкого рішення. Більш того, вважається, що задача дискретного логарифма на еліптичній кривій є більш складної для рішення, ніж задача дискретного логарифмування в скінченних полях. Найбільш швидкі методи, розроблені для скінченних полів виявляються не робочими в випадку еліптичних кривих. Так, для рішення дискретного логарифма існують досить швидкі алгоритми зі складністю $O\ (exp\ (c\ (\log p \log\log p)^d))$, де c і d – певні константи, а p – розмір поля. Такі алгоритми називаються субекспоненціальними і дозволяють відносно легко вирішувати дискретний логарифм в скінченному полі, якщо розмір поля не вибраний дуже великим, порядку 2^{1024}. В той же час, найбільш швидкі методи рішення дискретного логарифма на еліптичній кривій мають складність $O\ (\sqrt{q})$, де q – кількість точок еліптичної кривої.

Таким чином, для забезпечення рівня стійкості в 2^{80} операцій необхідно, щоб $q=2^{160}$. Для того ж, щоб получити аналогічний рівень стійкості при вичисленні дискретного логарифма в скінченному полі, необхідно поле порядку $q=2^{1024}$.

На основі вищесказаного можна зробити висновки:

- При використанні еліптичної криптографії значно зменшується довжина ключа в порівнянні з «класичною» асиметричною криптографією
- Швидкість роботи еліптичних алгоритмів значно вища, ніж у класичних. Це пояснюється як і розмірами поля, так і застосуванням більш близької для комп'ютерів структури бінарного скінченного поля.
- Через невелику довжину ключа і високу швидкість роботи, алгоритми асиметричної криптографії на еліптичних кривих можуть використовуватися в приладах з обмеженими обчислювальними ресурсами.

Отож, еліптична криптографія – дуже перспективний напрямок, дослідження і впровадження якого дозволить збільшити надійність криптосистем і зменшити при цьому затрати ресурсів.

Список використаних джерел:

1. Эллиптические кривые: новый этап развития современной криптографии [Електронний ресурс]: Режим доступу: http://www.secuteck.ru/articles2/Inf_ security/elept_krivaya_nov_etap_razv_kripto/ – 2014

2. Болотов А., Гашков С., Фролов А., Часовских А. Элементарное введение в эллиптическую криптографию // Москва: КомКнига, 2006. – 328 с.

Поліщук О.В.
студентка,
Національний технічний університет України
«Київський політехнічний інститут»

ПЕРСПЕКТИВНИЙ ПОШУКОВИЙ СЕРВЕР ELASTICSEARCH ЯК НОВИЙ ПІДХІД ПОШУКУ ЗБЕРЕЖЕНИХ ДАНИХ

Збереження даних на сьогодні є звичним процесом. Дану функцію виконують, як реляційні бази даних, так і NoSQL. Останні все більше стають популярним, маючи ряд переваг як у об'ємі збереження даних, так і у структурі збереження інформації. Та вони мають певний недолік, який з іншої ж сторони є й перевагою, а саме структура збереження. Більш того ця структура тягне за собою іншу, не менш важливу проблему – пошук по базі даних. Тому все-більш стають популярними різноманітні пошукові системи, такі як Lucene, ElasticSearch, які дають можливості пошуку по різним полям, додавання інформацію в індекси без додаткової переіндексації, заважтаження даних без додаткового блокування таблиць, зниження навантаження на базу при здійсненні join запитів.

Порівняно з найближчими конкурентами, ElasticSearch дає можливість швидкого пошуку, так як здійснює онлайн індексації, підтримує шардинг, реплікацію, витягує із пошуку оригінальні дані. Розподіленість – тренд сьогодення, все частіше розробники намагаються забезпечити розподіленість тим чи іншим чином, що дає можливість розгортання даних у хмарах.

В ElasticSearch не потрібно вказувати схему даних завчасно, достатньо відправити JSON запит, і сервер самостійно виконає операції для визначення типу. Також розробник може задати типи даних, структуру документу, аналізатор, який використовуватиметься, промапити індекси, тощо.

Звісно для розробників із досвідом зручним буде задати схему документів для збереження інформації й така можливість існує. Більш того пошуком можна керувати, починаючи від задання у якому документі здійснювати пошук, закінчуючи вказанням еквівалентності пошуку тої чи іншої фрази.

Пошук можна здійснювати, використовуючи фасет term або ж аналізатор match. Досить часто помилкою розробників є незнання, того що term не є аналізатором, й пошук здійснюється за непростим алгоритмом. Наприклад, term здійснює пошук, лише якщо слово пишеться у нижньому регістрі (у запиті), у самому ж ElasticSearch текст може бути як у нижньому так й у верхньому регістрі. Term має ряд специфічних функцій, й для пошуку який є очікуваний й бажаний користувачу все ж варто використовувати match query. З іншої