YURIY FEDKOVYCH CHERNIVTSI NATIONAL UNIVERSITY
in cooperation with
National Academy of Sciences of Ukraine
Institute of Cybernetics NAS Ukraine
Taras Shevchenko National University of Kyiv
National Technical University of Ukraine
«Igor Sikorsky Kyiv Polytechnic Institute»

Proceedings of the Eleventh International Conference on

«INFORMATICS AND COMPUTER TECHNICS PROBLEMS»

(PICT – 2022)

10 – 13 November, 2022, Chernivtsi, UKRAINE

Інститут кібернетики імені В.М. Глушкова НАН України
Київський національний університет імені Тараса Шевченка
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Чернівецький національний університет імені Юрія Федьковича

«ПРОБЛЕМИ ІНФОРМАТИКИ ТА КОМП'ЮТЕРНОЇ ТЕХНІКИ» (ПІКТ – 2022)

Праці XI-ї Міжнародної науково-практичної конференції

ЧЕРНІВЦІ
10 – 13 ЛИСТОПАДА, 2022

УДК 004.25

ДОБРЄЦОВА О.А., РУСНАК М.А
Чернівецький національний університет імені Юрія Федьковича
(Україна)

ГЕНЕРАТОР ОДНОРАЗОВИХ ЗАХИЩЕНИХ ЧАТІВ

Запропоновано метод генерації одноразових захищених чатів та його реалізація мовою JavaScript.

Постановка задачі. Розробити генератор одноразових захищених чатів за допомогою мови програмування JavaScript.

Чат — засіб обміну повідомленнями через комп'ютерну мережу в режимі реального часу, а також програмне забезпечення, яке дозволяє організовувати таке спілкування.

Захищений чат це програма, у якій повідомлення захищені шифруванням та не зберігаються не серверах. Через захищені чати можна надсилати файли, картинки, посилання та відео, здійснювати аудіо та відео дзвінки.

Одноразовий захищений чат – це тимчасово створена розмова двох користувачів, у якій кожне повідомлення зашифроване. Після завершення сеанс видаляється автоматично. Такі відомі месенджери як Telegram та Viber мають подібний функціонал, але в нього обмежені можливості і доступний він лише на смартфонах.

Захищеність забезпечують не тільки шифрування повідомлень, а й генерація унікального ключа сесії. Ключ втрачає актуальність в момент завершення сеансу чата. Генерація здійснюється генератором псевдовипадкових чисел, якому задано зерно та достатня ентропія, що забезпечують криптографічну стійкість та продуктивність. В якості джерела ентропії можуть використовуватися шуми в комп’ютері, послідовність яких достатньо важко передбачити. До того ж подібний «шум» може використовуватись не лише для визначення початкового «зерна», а й використовуватись для вибору «зерна» на подальших етапах генерації. Це дозволяє унеможливити передбачення наступних згенерованих чисел, навіть знаючи попередні числа і за рахунок цього значно збільшується період неповторюваної послідовності чисел.

До популярних математичних методів генерації псевдовипадкових чисел є лінійний конгруентний метод генерації. Лінійний конгруентний метод працює за формулою

$$x_{n+1} = (a * x_n + C) \pmod m, \mathrm{C} > 0 \quad (1)$$

де x_{n+1} та x_i - наступне і попереднє числа, a, c, m - константи, mod – операція знаходження залишку від ділення. В даному методі період повторення згенерованої послідовності дорівнює числу m, тобто при використанні даного методу задача зводиться до вибору такого числа m, при використанні якого період повторення чисел в згенерованій послідовності

задовольняв би наші вимоги до послідовності.

У випадку нелінійних конгруентних методів генерації псевдовидакових чисел попереднє x_i підводиться до ступеню n, що відповідає наступному співвідношенню для квадратного генератора (n=2)

$$x_{i+1} = (a * x_i^n + C) \bmod m \quad (2)$$

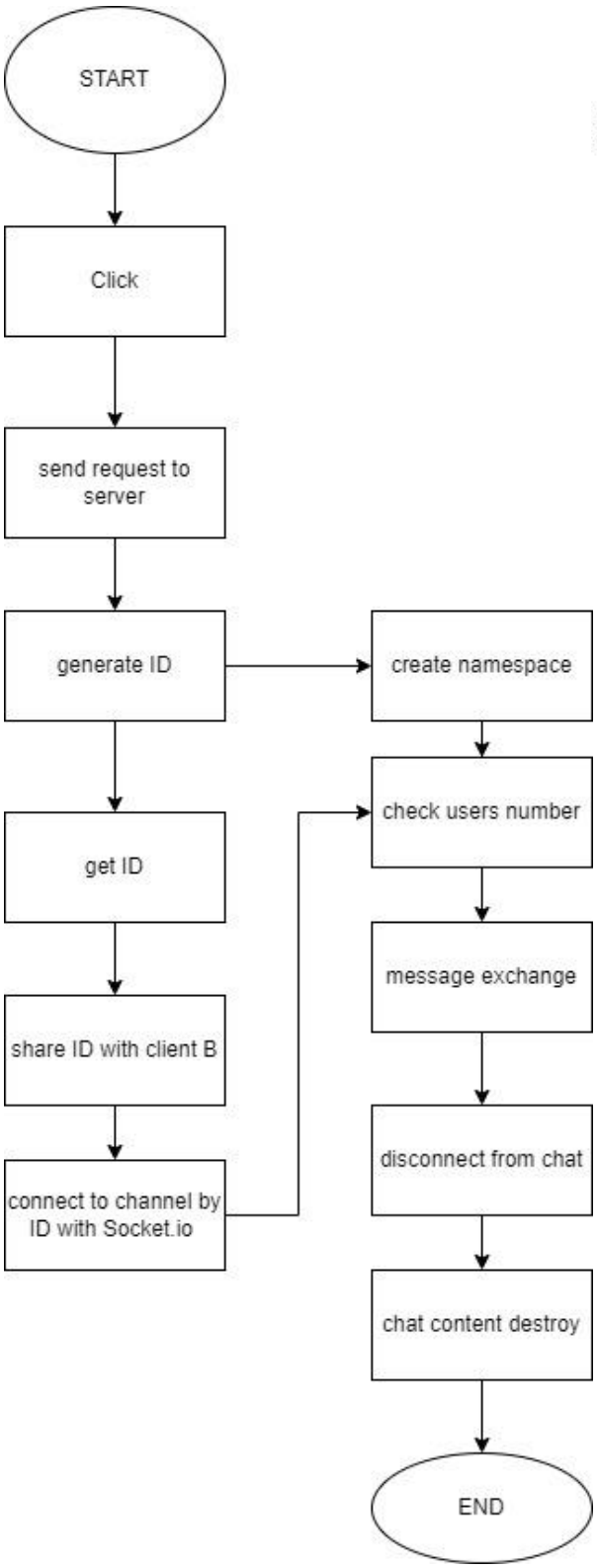
Використання подібних методів збільшує період повторення чисел в згенерованій послідовності, а для значного збільшення періоду часто використовують суперпозицію нелінійних конгруентних генераторів. Саме використання таких методів генерації чисел дозволяє збільшити криптостійкість усієї системи, що потребує надійності захисту від сторонніх посягань.

Основний результат та приклад. У роботі описується розробка, пристосована для використання на ПК. Опишемо структури даних, які використовує розробка, та функції, що вона виконує.

Алгоритм роботи: користувач повинен ввести ім’я, встановлюється зв’язок із сервером, генерується та повертається id-ключ кімнати. Створюється так звана «кімната» для спілкування, в якій буде відбуватися обмін контентом. Користувач А ділиться id з користувачем В. З’єднання з, так званим, каналом взаємодії користувачів через встановлюється через id кімнати. Йде перевірка кількості під’єднаних користувачів, якщо все в порядку, то відбувається обмін контентом між користувачами, якщо ж кількість користувачів перевищена, то зайві підключення «відрізаються». Після від’єднання зв’язок знищується та сесії дані видаляються. (Рис.1)

Програма передбачає архітектуру клієнт-сервер. Компонент клієнт – комп’ютер на стороні користувача, відправляє запит до сервера для надання інформації або виконання певних дій. Сервер – компонент, що призначений, в свою чергу, для вирішення певних завдань з виконання програмних кодів, надання доступу до певних ресурсів, виконання сервісних функцій за запитом клієнтів. [1]

Коли користувач натискає кнопку «Отримати id кімнати», на сервер відправляється запит, в момент обробки запиту викликається генератор ключа, який повертає id із відповіддю на запит. В момент генерації створюється кімната, в якій вже є користувач А, але ще немає користувача В. (Рис.2)



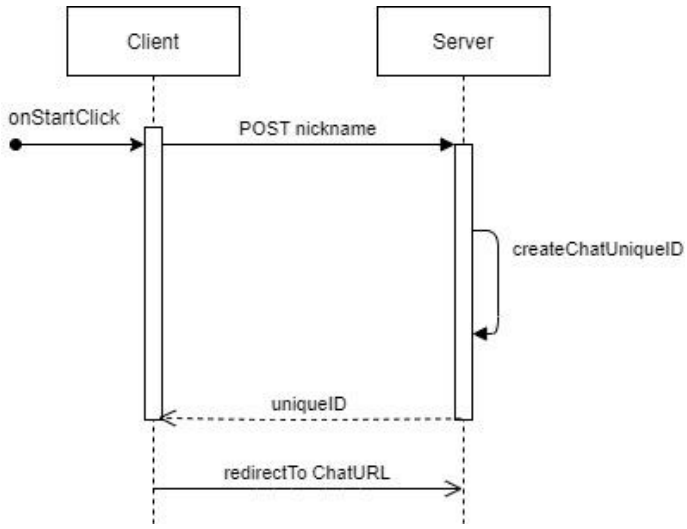


Рис. 1. Алгоритм роботи Рис. 2. Процес генерування чату

В якості генератора виступає функція crypto.getRandomBytes, яка підтримується модулем crypto, що містить NodeJS. Функція генерує криптостійкі псевдовипадкові дані, які переводяться у шістнадцятирічний рядок.

Для забезпечення підвищеної захищеності, чат повинен відбуватися в реальному часі. Ми використовуємо Socket.io бібліотеку, яка є подієво-

орієнтовною. При першому запуску користувачу пропонується ввести своє ім’я (псевдонім). За допомогою методів Socket.io встановлюється зв’язок з сервером, створюється унікальний ідентифікатор для кожного сокета та запитує ідентифікатор кожного разу, коли користувач відвідує веб-сторінку. Якщо ви оновлюєте або закриваєте веб-сторінку, сокет запускає подію відключення, яка показує, що користувач від’єднався від сокета.[2] З’єднання з чатом встановлюється через id кімнати саме за допомогою даної бібліотеки. Аналогічні дії відбуваються для користувача B, що вводить вже існуючий id.

Обмін контентом між користувачами забезпечується сокетами. Коли користувач натискає кнопку «Надіслати повідомлення», відправляється запит на сервер, який проходить обробку через Socket.io і повертає обидвом клієнтам дані, після чого вже відбувається рендеринг контенту у клієнтській частині. (Рис. 3).

Nodemon — це інструмент Node.js, який автоматично перезапускає сервер після виявлення змін у файлі, а Socket.io дозволяє нам налаштувати з’єднання в реальному часі на сервері.

Express.js — Node.js-фреймворк, що передбачає розробку веб-серверів і є основою проекту. CORS — це пакет Node.js, який дозволяє безпечно спілкуватися між користувачами з різних доменів. Іншими словами, CORS — це функція безпеки браузера, яка обмежує перехресні HTTP-запити з іншими серверами та визначає, які домени отримують доступ до ваших ресурсів. За допомогою цього інструменту здійснюється встановлення зв’язку між користувачами.

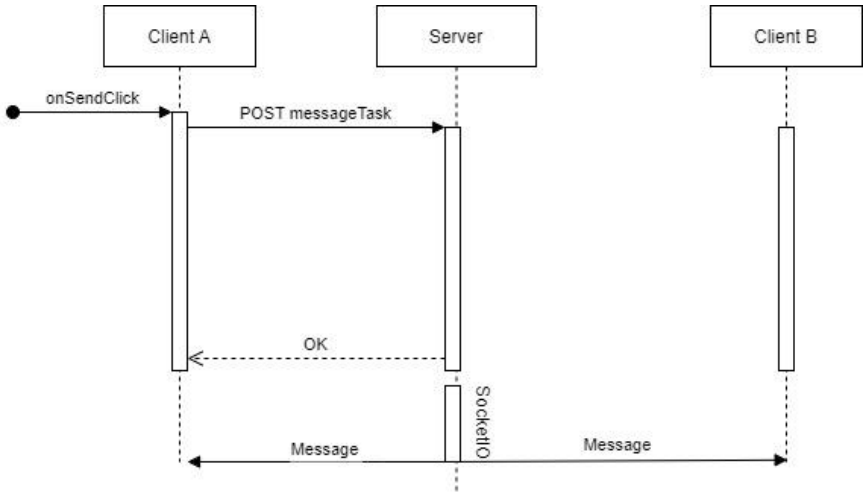


Рис. 3. Відправлення повідомлення

Клієнтська частина розроблена за допомогою потужної бібліотеки React.js, заснованої на інкапсульованих компонентах, що керують власним станом. Завдяки декларативності, створення інтерфейсів спрощується. Компоненти можуть бути як функціональні, так і класові[3]. Варто підкреслити, що необхідно та достатньо описати вигляд різних частин інтерфейсів у кожному стані застосунку для передбачуваної поведінки.

Основними компонентами виступають: компонент вводу імені,

компонент кімнати, компонент повідомлення, список користувачів, список повідомлень, а також поле вводу повідомлень.

Структура повідомлення включає інформацію про ідентифікатор та тип повідомлення, текст повідомлення або шлях до файлу, ідентифікатор та ім’я користувача, дата і час створення та оновлення повідомлення.

Висновок: за допомогою означених інструментів та розробленого алгоритму можна побудувати одноразовий захищений чат для роботи в режимі реального часу.

ПЕРЕЛІК ЛІТЕРАТУРИ

1. Мартін Клепман «Сильно завантажені програми. Програмування, масштабування, підтримка» 2018р. 138ст. (дата звернення 07.09.2022)

2. Джесі Джеймс Гарретт «Ajax: Новий підхід до веб-програми. AdaptivePath» 44ст. (дата звернення 09.09.2022)

3.Документація ReactJS [Електронний ресурс] URL: https://uk.reactjs.org/docs (дата звернення 12.09.2022)

УДК 519.172.1

ІВАНЕШКІН О.І.
МННЦ ІТ і С НАН та МОН (Україна)

НОВА ІНФОРМАЦІЙНА SOFTWARE-ТЕХНОЛОГІЯ ДЛЯ РОБОТИ З НЕОРІЄНТОВАНИМИ ЗМІШАНИМИ ЛІСАМИ $MF(T_I;S_J)$ У ВИРІШЕННІ ПИТАННЯ СЕЛЕКТИВНОГО ПАКУВАННЯ ЇХНЬОЇ СТРУКТУРИ

Створено та реалізовано мовою MS VISUAL C++ 6 нову високоефективну інформаційну software-технологію для роботи з довільного вигляду неорієнтованими змішаними лісами $MF(T_I;S_J)$. За потреби всього 4 байти інформації на кожну вершину, вона дозволяє працювати з об'єктами, що містять до 65536 вершин, здатних бути розташованими на 6400 віртуальних X-рівнях екрану монітора. Наведено приклади практичного використання технології при вирішенні комплексу взаємопов'язаних питань: різні види пакування структури лісів шляхом селективного видалення з неї цілого ряду елементів (вершин, саджанців, гілок та крон дерев), збереження зв’язності компонентів модифікованого лісу, мінімізація витрат часових та технічних ресурсів при моделюванні.

Мета роботи. Створити нову інформаційну технологію, наділену максимальними функціональними можливостями роботи на найбільш загального (ніж традиційні зв'язані дерева) виду об'єктів та розширення сфери її практичного застосування.

Основна частина. Результати, що наводяться, отримані при реалізації