

УДК 004

Я.Якубенко, магістр гр. КН-22МЗ,

Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ СИНТЕЗУ ТА АНАЛІЗУ QR-КОДІВ НА МОБІЛЬНИХ ПРИСТРОЯХ ПІД ОС ANDROID

У статті розроблено програмне забезпечення, яке призначено для системи синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android. Метою розробки є дослідження та програмна реалізація системи синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android. Об'єктом дослідження є процес синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android. Предметом дослідження є методи синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android. Методи дослідження базуються на методах теорії кодування, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Постановка проблеми. QR-код – квадратна картинка в яку закодована якась інформація.

Є багато речей, які можна сказати про QR-коди, а саме:

- кілька режимів кодування;
- режим кодування ECI;
- кодування символів кандзі;
- оптимізація даних;
- мікро QR-коди;
- читання QR-кодів у браузері.

Все це розглянемо далі у даній роботі.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-20] було виявлено певні прогалини у забезпеченні системи синтезу та аналізу qr-кодів на мобільних пристроях під ОС Android.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

- Огляд існуючих систем синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android.
- Дослідження системи синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android.
- Програмна реалізація системи синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android.

Об'єктом дослідження є процес синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android.

Предметом дослідження є методи синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android.

Методи дослідження базуються на методах теорії кодування, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу. Розглянемо існуючі програми для зчитування QR-кодів.

Безпечний зчитувач QR-кодів TrendMicro

Якщо ви шукаєте програму для сканування QR-коду, яка надає пріоритет безпеці, Safe QR Reader від TrendMicro стане чудовим вибором. Доступний як для iOS, так і для Android, це чудовий інструмент для пристроїв, керованих через рішення iOS MDM.

Оскільки ця програма розроблена провідною компанією з кібербезпеки, ви отримуйте такий самий захист під час сканування QR-коду, як і для інших веб-дій. Ви можете фільтрувати рівень безпеки для веб-загроз і повідомляти про підозрілий веб-сайт із програми.

У додатку для iOS сканер QR-коду зручно розміщено у верхньому правому куті, що є зручним вибором, оскільки програма не є спеціальним сканером QR-коду. Якщо натиснути це вперше, запитується дозвіл камери, без перерв для сканування після цього.

Його версія Google Play – це спеціальний сканер QR-кодів без інших антивірусних функцій, які захаращують додаток iOS. Ви також отримуйте історію сканування та можливість сканувати зображення QR-коду з галереї, чого немає в програмі iOS.

Додаток може не мати багатьох додаткових функцій, таких як штрих-код, PDF, інше сканування, заміна браузерів, історія сканування тощо. Але коли йдеться про основну роботу – забезпечення безпечного сканування QR-коду – він добре блокує шкідливі QR-коди.

Ось кілька причин, чому ми вважаємо TrendMicro однією з найбезпечніших програм для сканування QR-кодів для Android та iOS:

1. Пропонує перевірки безпеки, які гарантують, що QR-код не веде до небезпечного посилання чи шкідливого вмісту – як на iOS, так і на Android
2. Схематичні веб-сайти блокуються та повідомляються миттєво. Він в основному діє як антивірус для QR-кодів!
3. Він зберігає історію всіх сканувань QR-кодів, зроблених у програмі (Android).
4. У ньому немає надокучливої реклами та не продається реклама третім сторонам, що робить його ідеальним додатком для людей, які піклуються про дані

Програма доступна в Apple App Store як «Trend Micro Mobile Security» і в Google Play як «QR Scanner».

Сканер QR і штрих-кодів від Gamma Play

Ця програма від Gamma Play, яка має понад 500 мільйонів завантажень (у Google Play) і 2,66 мільйонів відгуків, є одним із найпопулярніших сканерів QR-кодів, доступних для iOS та Android.

Це справжній, простий додаток для сканування QR-коду, який ми можемо завантажити (на iPhone), запустити та відразу відсканувати QR-код! Те саме стосується програми для Android.

Однією з особливостей програми для Android, яка мене вразила, є те, що вона щоразу робить зображення QR-коду, який ви скануєте. Програма для iOS не має цієї функції.

Ось деякі інші функції, які я помітив під час гри з програмою в обох ОС:

1. Він зберігає історію всіх минулих сканувань. Корисно, якщо ви хочете повторно переглянути посилання, яке ви сканували з продукту
2. Після кожного сканування програма автоматично визначає відповідні дії та пропонує швидкі клацання, щоб перейти за URL-адресою, зберегти контактну інформацію або зателефонувати за номером телефону VoIP
3. Ви можете сканувати зображення QR-коду, які зберігаються в галереї вашого телефону
4. Деякі користувачі високо оцінили програму за швидкість сканування QR-коду (менше однієї секунди)

Додаток для iOS має менше функцій, орієнтованих на взаємодію з користувачем, таких як вібрація під час сканування та дотик для фокусування. Люди Android отримують їх поширення.

QR Code Reader від TeaCapps

QR Code Reader від TeaCapps є одним із найвищих рейтингів програм у Apple App Store (4,7/13 000 відгуків) і Google Play (4,4/2,8 млн відгуків). Інтерфейс користувача простий. Є регульована рамка камери, яку можна стиснути, щоб зменшити або збільшити зону фокусування сканера – це корисно під час сканування QR-кодів у людних місцях.

Основні функції безкоштовні, але ви можете заплатити своїм часом, щоб розблокувати деякі розширені функції, як-от підтвердження сканування вручну, за допомогою перегляду реклами – іноді багатьох із них.

Ось деякі функції, які мені сподобалися під час випробування:

1. Консервально ставиться до дозволів додатків і вимагає лише камери та доступу до подальших URL-адрес із QR-кодів, що є хорошою ознакою
2. Пропонує необмежену історію сканування та допомагає легко керувати нею
3. Ви можете експортувати історію сканування як файл CSV або зберегти її у своєму хмарному сховищі
4. Існує функція масштабування, яка допомагає зосередитися на скануванні здалеку

Під час тестування я помітив, що програма має елегантний і простий дизайн UI/UX і працює точно так, як рекламується, за винятком випадкової реклами. Звичайно, ви можете спробувати версію без реклами для зручнішого використання.

QR & Barcode Reader від QR SCAN Team

Простий і функціональний, команда QR SCAN створила безкоштовний додаток без будь-яких умов. У програмі все безкоштовно, від копіювання в буфер обміну до розширених функцій, таких як пакетне сканування та спеціальні дії.

Це ідеальний додаток для додаткової конфіденційності даних, оскільки розробники не надають даних для продажу реклами.

Додаток містить необхідні функції для сканування, перемикання ліхтарика, повороту камери, перегляду історії тощо прямо під інтерфейсом камери. Як правило, параметри Android приховані в меню гамбургера, і для доступу до них потрібне додаткове клацання, що є бажаним кроком.

На основі моїх спостережень, ось деякі з функцій, які я вважаю корисними:

1. Ви можете вимкнути миттєве переспрямування після сканування, запобігаючи автоматичному потраплянню на веб-сторінку (безкоштовно)
2. Простий у використанні з елегантним дизайном для телефону та планшета (багато користувачів також сказали це у відгуках)
3. Історія сканування детальна, сканування класифіковані відповідно до типу вмісту, доступ до якого отримано після сканування
4. Додаток ефективно використовує акумулятор телефону
5. Ви також можете читати зображення JPEG за допомогою сканера

Simple QR-сканер

Програма для сканування QR-кодів від Simple Design має приємний інтерфейс користувача та чудову анімацію сканування. Опція для перемикання ліхтарика зручно розташована у верхньому правому куті, а лівий займає кнопка доступу до галереї. Однак це майже все.

Ви отримуєте деякі параметри для покращення роботи зі скануванням, як-от налаштування історії та програм, як-от вибір пошукової системи за умовчанням. Ця програма не має додаткових функцій, таких як експорт, зображення відсканованого QR-коду, пакетне сканування та ручне підтвердження переспрямування.

Зосередження на основних функціях також допомагає програмі мати невеликий обсяг пам'яті. Ось деякі інші спостереження:

1. Ви отримувате масштабування, щоб ловити невеликі або віддалені QR-коди без зусиль
2. Має параметр користувацького пошуку, який допомагає додавати веб-сайти до пошуку за QR-кодом
3. Ви можете вибрати зображення з QR-кодом у своїй галереї, щоб отримати дані в програмі
4. Ви можете включити ліхтарик для безперервного сканування QR-коду в темряві

QR Reader для iPhone від TapMedia

Якщо вам потрібна програма для сканування QR-кодів, спеціально створена для iPhone, це саме те. Він також має високу оцінку 4,7 (4500 відгуків).

Після оновлення ця програма поширюється на PDF-файли, візитні картки та сканери головоломок (усі з швидкими жєстами гортання).

Інтерфейс простий із параметрами доступу до програми «Фотографії», ліхтарика та камери, розташованих у верхній частині інтерфейсу для легкого доступу. Ви також можете переглянути історію сканування, клацнувши спадне меню.

Меню налаштувань є вичерпним, що дає вам детальний контроль над сценаріями сканування та постсканування. Наприклад, ви можете вмикати/вимикати сканування NFC-тегів (якщо ваш iPhone це підтримує), масштабування камери (якщо ви хочете автоматично), чудову лазерну анімацію під час сканування, вібрацію для вказівки на успішне сканування тощо.

Окрім оптимізації для iOS, я помітив ще кілька ключових функцій:

1. Він може ефективно сканувати різні типи QR-кодів і штрих-кодів
2. Він має функцію сканування бази даних для надсилання QR-кодів і штрих-кодів на сервер або API
3. Він оснащений стандартними функціями безпеки для забезпечення більшої безпеки.

Однак якщо ви завантажуєте програму, шукаючи базовий сканер QR-коду для iPhone, вичерпні параметри можуть заплутати вас або просто залишитися невикористаними, якщо ви заплатите за це.

Зчитувач QR-кодів від Mixerbox

Ще один сканер QR-кодів для iPhone створений для того, що він називається. Єдиним доповненням є те, що він дуже швидкий для сканування, як він стверджує. Навіть якщо QR-код видно десь поблизу інтерфейсу камери, він сканується (але не перенаправляється, якщо ви не зробите це вручну).

Інтерфейс простий, лише п'ять опцій: доступ до галереї, ліхтарика, історії та налаштувань програми. Натискання налаштувань програми також не заплутає вас; він має лише чотири варіанти, з яких ви, ймовірно, скористаєтесь поширеними запитаннями та статусом підписки. Це майже все про сканер QR-кодів Mixer Box.

Ось деякі інші спостереження:

1. Підтримує понад 15 типів форматів штрих-кодів 1D і 2D з високою швидкістю сканування
2. Чудові функції озвучення, які підходять для щоденного використання

Ми рекомендуємо використовувати Secure QR Reader від TrendMicro для Android і Trend Micro Mobile Security на iPhone як найкращі програми QR Scanner у 2024 році для iOS та Android. Згідно з нашим оглядом, ця програма надає перевагу безпеці над додатковими функціями, водночас забезпечуючи мінімум, необхідний для якісного сканування QR-коду.

Найкращим міжплатформним додатком для сканування QR-кодів із найкращим набором доданих функцій є QR & Barcode Scanner від Gamma Play.

Який найкращий додаток для сканування QR-кодів для iPhone та Android?

Серед найкращих безкоштовних програм для сканування QR-кодів для iPhone та Android:

1. Захищений зчитувач QR-кодів від TrendMicro.

2. QR і зчитувач штрих-кодів від Gamma Play.
3. Зчитувач QR-кодів від TeaCapps.
4. QR і зчитувач штрих-кодів від QR SCAN Team.
5. Простий дизайн QR-сканера.
6. QR Reader для iPhone від TapMedia.
7. Зчитувач QR-кодів від MixerBox.

Чи можна відсканувати QR-код без програми?

Так, ви можете, якщо на вашому мобільному пристрої встановлено Android версії 9 і новішої або iOS версії 11 і новішої. Однак, якщо у вас немає мобільного пристрою, ось кілька способів сканування QR-коду без програми:

1. Онлайн-декодер: якщо вам потрібно відсканувати QR-код на екрані комп'ютера, але під рукою немає мобільного пристрою, скористайтеся онлайн-декодером. ZXing Decoder Online є одним із найкращих доступних.

2. Декодер QR-коду веб-камери: QR-коди можна сканувати за допомогою веб-камери, щоб розшифрувати їх. Спробуйте такі програми, як WebQR, щоб почати сканування.

3. Оптичні сканери QR-кодів: ці сканери є альтернативою додаткам QR-кодів. Це портативні або стаціонарні оптичні сканери. Ви можете використовувати їх для масового сканування QR-кодів, наприклад тих, що використовуються для продажу квитків і виставлення рахунків, які потрібно підключити до комп'ютерного програмного забезпечення для запуску.

Чи є в моєму телефоні QR-сканер?

Для сканування QR-коду потрібен мобільний пристрій. Більшість мобільних пристроїв мають вбудований сканер QR-коду в камеру.

Щоб перевірити, чи є у вашому телефоні вбудований сканер QR-коду, відкрийте камеру та наведіть її на QR-код. Якщо QR-код спрямовує вас на веб-сторінку, що містить деталі QR-коду, або відображає банер, у вашому телефоні є сканер QR-коду.

Якщо ні, вам потрібно встановити програму для сканування QR-коду.

Як я можу сканувати QR-код за допомогою свого телефону?

Відкрийте камеру на своєму мобільному пристрої та розмістіть телефон так, щоб QR-код з'явився у цифровому видошукачі. Натисніть підказку, яка з'явиться на екрані, щоб запустити QR-код.

Ви повинні встановити програму для сканування QR-коду, якщо ваш телефон не підтримує сканування QR-коду. Ми рекомендуємо Safe QR Reader від TrendMicro, QR Reader від TeaCapps і QR & Barcode Scanner від Gamma Play.

Чи потрібна програма для сканування QR-коду?

Вам не потрібно завантажувати програму стороннього розробника, якщо на вашому мобільному пристрої встановлено iOS 11 і новіших версій або Android 9 і новіших версій. Замість цього ви можете використовувати QR-сканер, вбудований у рідну камеру мобільного пристрою.

Якщо ваш мобільний пристрій не підтримує операційну систему, інсталюйте програму для сканування QR-кодів, наприклад Safe QR Reader від TrendMicro або сканер QR і штрих-кодів від Gamma Play на телефонах iOS і Android.

Як відсканувати QR-код зі скріншота або зображення на телефоні?

Якщо на вашому мобільному пристрої встановлено найновішу операційну систему, ви можете відсканувати QR-код зі знімка екрана або зображення на телефоні за допомогою Google Lens.

Щоб відсканувати QR-код за допомогою Google Lens, запустіть опцію Lens на своєму мобільному пристрої, щоб отримати доступ до вмісту QR-коду. Дізнайтеся більше про це тут.

Опис QR-кодів

QR-коди містять дані, це чесно сказати. Тип даних можна вибрати, але, звичайно, це визначає максимальний обсяг інформації, яку можна зберегти:

- номери (до 7089);
- алфавітно-цифровий (цифри, великі літери, купа символів: ~65% дорожче цифр);
- байти (лише 8 біт символів у кодуванні Latin-1, ~140% дорожче);
- кандзі (~290% дорожче).

Не дивно, що кандзі є одним із основних наборів символів, оскільки QR-коди були розроблені Denso Wave, японською компанією з автоматизації.

Насправді в останніх версіях є й інші режими кодування, але, як згадувалося раніше, поки що ми зосередимося на 8-бітних байтах. І, зрештою, QR-код – це набір бітів, тож якщо ви хочете закодувати свою інформацію як забажаєте, ви можете.

Крім того, QR-коди також можуть *перемикатися* в інший режим кодування в середині своїх даних, але ми поки не будемо розглядати цей випадок.

Розміри

QR-коди завжди квадратні, але їх розміри різняться. Розмір визначається незвичайним терміном «версія», так що версія 1 має розмір 21×21 пікселів, тоді як версія 40 (найбільша) має розмір 177×177 пікселів. QR-код на 1 версію більший на 4 пікселі ширший і вищий, тому розмір становить $(17 + \text{версія} * 4)$ пікселів.

Крім того, ми повинні називати їх не пікселями, а скоріше «модулями» (знову незвично, але, можливо, щось було втрачено в перекладі з японської).

Оскільки більші QR-коди складніше декодувати (і обчислювально дорожче), мета полягає в тому, щоб використовувати найменшу можливу «версію» для того обсягу даних, який ми хочемо зберегти.

Більші QR-коди розділяють свої дані на кілька блоків (до 81).

Виправлення помилок

Кожен QR-код містить «модулі» виправлення помилок – і ні, ми не можемо видалити їх, щоб збільшити доступний простір. Але ми можемо вибрати один з 4 рівнів виправлення помилок.

Таблиця 1 – Рівні виправлення помилок

Рівень	лист	Відновлення даних
Низький	Л	~7%
Середній	М	~15%
Квартиль	Q	~25%
Високий	Х	~30%

Вищим виправленням помилок можна зловживати для створення QR-кодів, частково покритих логотипами та зображеннями, але їх можна прочитати завдяки виправленню помилок.

Фіксовані шаблони

Багато хто з нас може з першого погляду розпізнати, що таке QR-код, і це завдяки деяким загальним характеристикам:

- це квадратні зображення;
- вони чорно-білі або принаймні двох кольорів, дуже віддалених у спектрі яскравості (тому замість цього ми називатимемо їх «темними» та «світлими»);
- вони складаються з сітки квадратних точок;
- вони мають деякі легко впізнавані візерунки навколо фотографій.

Щодо останнього, сенс легкого розпізнавання саме тому, чому вони були розроблені таким чином: давайте пам'ятаємо, що QR (= Швидка відповідь) коди були розроблені для використання промисловими автоматизованими машинами.

Ці моделі:

- **шаблони шукача:** 7×7 квадратів, розміщених у верхньому лівому, верхньому правому та нижньому лівому кутах, розділених лінією порожніх модулів;
- **шаблони вирівнювання:** квадрати розміром 5×5 , розташовані на кутах і перетинах сітки $n \times n$ (якщо вони не зайняті шаблонами шукача); n коливається між 2 і 6, тому їх $n^2 - 3$, за винятком версії 1, яка не має шаблону вирівнювання;
- **часові візерунки:** горизонтальна і вертикальна лінії чергування темних і світлих модулів, що з'єднують шаблони шукача (ви помітили це, лише якщо глибоко вивчили деякі QR-коди);
- **темний модуль:** просто модуль, який завжди темний, розміщений у 9-му стовпці та $(4 * \text{версія}) + 10$ -му рядку (б'юся об заклад, ви ніколи цього не помічали!).

Крім того, у великих QR-кодах (від версії 7 і вище) кілька областей зарезервовано для даних формату.

Ємність

Залежно від версії, режиму кодування та рівня виправлення помилок визначається *ємність QR-коду*. Доступний простір, який не зайнятий фіксованими шаблонами чи зарезервованими областями, поділено на групи з 8 модулів, які називаються «кодовими словами»: уявіть їх як класичні 8-бітні байти.

Тому загальна кількість доступних кодових слів є фіксованою для кожної версії: 26 для версії 1, 44 для версії 2 і так далі, до 3706 для версії 40.

Для кожної версії також визначаються кодові слова, зарезервовані для виправлення помилок.

Без зайвих слів, давайте почнемо створювати невеликий QR-код із кодуванням ISO-8859-1 байт!

ISO-8859-1

Так, QR-коди використовують ISO-8859-1 (також відомий як Latin-1) для кодування рядків байтів. Сьогодні UTF-8 більш поширений, але деякий час тому цього не було.

Основна проблема полягає в тому, що, хоча UTF-8 може охоплювати мільйони символів (або «кодових точок»), Latin-1 має лише 255 символів. Це воно. Ні емодзі, ні інших алфавітів. Якщо ви хочете перевірити, чи рядок дійсний для Latin-1, перевірка проста:

```
const LATIN1_RE = /^[x00-\xff]*$/;
function isLatin1(string) {
  return LATIN1_RE.test(string);
}
```

Якщо деякі символи *знаходяться* за межами ISO-8859-1, добре... ви або відкидаєте їх, або використовуєте режим ECI. Крім того, деякі зчитувачі автоматично розпізнають, якщо натомість використовується UTF-8, але це може бути ненадійним вибором для публічних QR-кодів.

Режим кодування

Перш за все, нам потрібно знайти правильний режим кодування. Кожен режим має відповідне значення згідно з наступною таблицею.

Таблиця 2 – Режим кодування

Режим кодування	Значення бітів
Числовий	0001 (1)
Буквено-цифровий	0010 (2)
Байт	0100 (4)
Кандзі	1000 (8)
ECI	0111 (7)

Визначення оптимального режиму кодування легко зробити, просто перевіривши, які символи входять до рядка. Єдина проблема полягає в правильному виявленні символів

кандзі. У жодному разі я не експерт у кандзі, тому я просто покладаюся на нову підтримку Unicode ES2018 RegExpy JavaScript.

Насправді я не знаю, чи він ідеально підходить для режиму кандзі, тому, якщо хтось знає, просто пишiть у коментарях! (Ймовірно, я придумаю краще рішення пізніше в серії.)

Зрештою, маємо `getEncodingMode('https://www.qrcode.com/') === 4`.

Версія

Давайте прагнути до найменшої можливої версії: оскільки вона складається з 23 символів, ми можемо перевірити в різних таблицях (тут, наприклад), що нам знадобиться принаймні код версії 2. Крім того, оскільки ми там, ми можемо отримати найвищий можливий рівень корекції – середній, у нашому випадку.

Також ця інша таблиця говорить нам, що версія 2 може містити 28 кодових слів даних для середньої корекції: ці 2 запасних кодових слова будуть використовуватися для інформації про дані.

Якщо ми хотіли отримати вищий рівень виправлення помилок, ми повинні були вибрати більшу версію.

Біти даних

Перші 4 біти нашої послідовності даних – це 0100, наш режим кодування.

Тоді ми скажемо, якої довжини буде наш рядок. Для цього знову потрібна таблиця, оскільки кількість бітів, зарезервованих для цього значення, є змінною.

Таблиця 3 – Біти даних

Режим кодування	Версія 1-9	Версія 10-26	Версія 27-40
Числовий	10	12	14
Буквено-цифровий	9	11	13
Байт	8	16	16
Кандзі	8	10	12

Оскільки `getLengthBits(4, 2) === 8` нам знадобиться 8 бітів, а 23 (довжина нашого рядка) у двійковій системі дорівнює 10111, наші перші біти:

01000001 0111....

Далі *реальні* дані. Все, що нам потрібно зробити, це отримати код символів рядка в ISO-8859-1:

`h t t p s : / / w w w . q r c o d e . c o m /`

104 116 116 112 115 58 47 47 119 119 46 113 114 99 111 100 101 46 99 111 109 47

Тепер перетворіть усе в двійковий і об'єднайте в попередню послідовність:

01000001 01110110 10000111 01000111 01000111

00000111 00110011 10100010 11110010 11110111

01110111 01110111 01110010 11100111 00010111

00100110 00110110 11110110 01000110 01010010

11100110 00110110 11110110 11010010 1111....

Тепер ми *повинні* поставити *блок завершення*, який рівно 4 нулі, тому останні кодові слова будуть 11110000. Ми все ще маємо заповнити 3 із 28 доступних кодових слів.

Залишилося місце

Ми заповнили всі 8 бітів останнього кодового слова, інакше нам довелося б заповнити решту бітів нулями (це завжди так у байтовому режимі).

За допомогою решти кодових слів ми можемо зробити дві речі:

– замість 4-бітового блоку завершення ми можемо поставити інший блок режиму кодування (можливо, інший) і почати іншу послідовність – але лише з трьома кодовими словами ми не можемо зробити багато;

– заповніть решту кодових слів послідовностями 11101100 00010001 (що перетворюється на 236 і 17 у десятковій системі), доки не буде досягнуто обмеження.

Чому 236 і 17? Я поняття не маю, але я припускаю, що вони (Denso Wave?) зробили багато спроб і перевірили, що ці дві послідовності створюють найлегше розпізнавані коди.

У підсумку маємо:

65 118 135 71 71 7 51 162 242 247 119 119 114 231 23 38 54 246 70 82 230 54 246 210
240 236 17 236

Або в двійковому вигляді:

```
01000001 01110110 10000111 01000111 01000111
00000111 00110011 10100010 11110010 11110111
01110111 01110111 01110010 11100111 00010111
00100110 00110110 11110110 01000110 01010010
11100110 00110110 11110110 11010010 11110000
11101100 00010001 11101100
```

Переведення в код

Для нашої функції `getBytes` знадобляться три речі:

- звичайно, вміст, який буде розділено на кодові слова;
- скільки бітів потрібно для визначення довжини вмісту: як ми бачили, це залежить від режиму кодування (у цьому випадку байт) і версії (у нашому випадку це 8);
- кількість кодових слів, які потрібно заповнити: це знову залежить від версії QR-коду та рівня виправлення помилок (у нашому випадку це 28).

Для QR-кодів від версії 10 і вище нам знадобиться **16** біт для вираження довжини нашого вмісту, тому фактичні дані починатимуться з третього кодового слова.



Інші режими кодування

Наразі ми не будемо вдаватися в подробиці, але для числового режиму нам потрібно розділити число на групи з 3 цифр і закодувати кожен групу 10 бітами ($2^{10} = 1024$, тому втрачений простір мінімальний).

Натомість буквено-цифровий режим містить 45 символів, тому рядок має бути розбитий на групи по 2 символи. Кожен символ має значення (спочатку цифри, потім великі латинські літери, потім пробіл і символи \$, %, *, +, \, -, ., /, :), тому кожен пару символів можна перевести в число в діапазоні від 0 до 2024 ($= 45^2 - 1$). Тому нам потрібно 11 біт на кожен два буквено-цифрових символи ($2^{11} = 2048$).

Для режиму кандзі... о боже. По-перше, нам потрібно отримати код Shift JIS піктограми, і найкращий спосіб зробити це – використовувати бібліотеку на кшталт `iconv-lite` або, якщо ви хочете зробити це самостійно, використати її таблицю символів. Крім того, можна використовувати не всі символи, а лише ті, що знаходяться в діапазонах від `0x8140` до `0x9FFC` і від `0xE040` до `0xEBBF`. Зрештою, символ кандзі займе 13 біт.

Розробка структурної схеми

Скажу чітко: насправді показ QR-коду – це лише проблема візуалізації. Ми можемо зробити це за допомогою SVG, а `<canvas>`, групи квадратів `s` або навіть цих двох емоذجі:  . Це не дуже важливо й не складно для тих, хто має мінімальний досвід у відтворенні матеріалів у мережі.

Важливо отримати матрицю бітів, яка дозволить нам створити таку фігуру.

Почнемо з фактичного зберігання даних. Знову ж таки, для зручності ми можемо використовувати лише масив масивів, тобто матрицю, щоб записати, чи є модуль світлим (0) чи темним (1). Але для рядків ми `Uint8Array` знову можемо використовувати `s`, оскільки вони швидші за звичайні масиви, а також для `.set()` методу, який стане в нагоді. Почнемо з простого:

Другий аргумент – `Array.from` це в основному марфункція, яка дозволяє нам використовувати *новий* типований масив для кожного рядка (тобто `new Array(length).fill(new Uint8Array(length))` використовуватиме *той самий* масив для кожного рядка).

Тепер нам потрібна функція, яка заповнює спроектовану область одиницями або нулями, оскільки це буде корисно для фіксованих шаблонів:

На цьому етапі нам потрібна послідовність модулів, які ми повинні заповнити нашими кодовими словами. Наша стратегія буде:

- почати з порожньої матриці;
- позначте одиницями зарезервовані зони;
- застосуйте 7-крокову ітерацію вище - або подібну.

Ми внесли деякі зміни у функцію вище. Перш за все, ми використовуємо `rowStep` для відстеження того, чи рухаємося ми вгору чи вниз у матриці. Потім ми використовуємо `index` його парність, щоб визначити, чи потрібно нам йти ліворуч чи рухатися по діагоналі.

Для нашого QR-коду версії 2 ми маємо отримати ось що:

```
getModuleSequence(2)
```

```
// Uint8Array(359) [[24, 24], [24, 23], [23, 24],..., [16, 0]]
```

Нарешті настав час розмістити наші дані (як модулі повідомлень, так і модулі виправлення помилок)!

Ми отримуємо прото-QR-код версії 2. Під "прото" я маю на увазі, що його не було перетворено останньою дією: маскуванням. Він полягає в виконанні XOR всіх модулів за допомогою одного з 8 попередньо визначених шаблонів. А *навіщо* нам це робити, можете запитати ви?

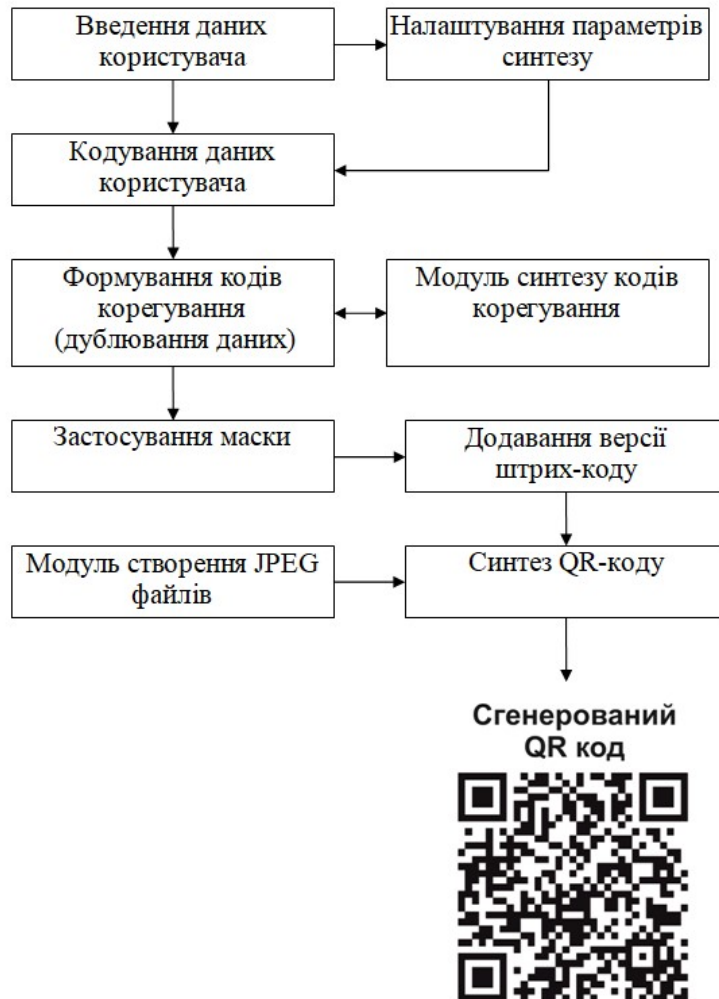


Рисунок 1 – Структурна схема системи

Що ж, цього разу це *має* сенс. Якщо ви подивитесь на наш прото-QR-код, то там є великі області, рівномірно заповнені темними або світлими візерунками, і сканери зазвичай не люблять їх, оскільки вони можуть не відповідати сітці або неправильно підраховувати рядки чи стовпці. Тому нам доведеться застосувати маску, щоб мінімізувати цю проблему.

На рисунку 1 зображено розроблену структурну схему забезпечення системи синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android.

Розглянемо схему згори до низу. Спочатку проходить введення даних користувача, а саме даних які будуть вбудовані у QR код. При необхідності проводиться зміна налаштувань генерації QR коду по замовчанню.

Далі проводиться кодування даних користувача з використання існуючих налаштувань кодування. Після цього проводиться формування кодів корегування тобто дублювання даних для поміхо стійкості з використанням модулю генерації кодів корегування.

Після цих дій проводиться застосування маски та додавання версії штрих-коду. Версії штрих-коду дозволяє при розпізнаванні QR коду користувачем більш точно провести декодування.

Та на останньому етапі проводиться генерація двомірного матричного штрих-коду QR з використанням модуля створення Jpg файлів, що у кінцевому результаті дає повноцінний згенерований QR код.

Висновки. У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android. Рішення даного завдання полягало є у вирішенні наступних задач: Був проведений огляд існуючих систем синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android; Досліджена система синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android; На основі отриманих результатів досліджень створена програмна реалізація системи синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android. Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання синтезу та аналізу QR-кодів на мобільних пристроях під ОС Android. Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Список літератури

1. О.А.Смірнов, Т.В.Смірнова, Л.І. Поліщук, К.О. Буравченко, А.О.Макевнін, «Дослідження хмарних технологій як сервісів», Кібербезпека: освіта, наука, техніка. № 3(7). С. 43-62. 2020.
2. Смірнов О.А., Дреєва Г.М., Дреєв О.М., Смірнова Т.В. «Фрактальний аналіз генератора самоподібного трафіку на основі ланцюга Маркова». Центральноукраїнський науковий вісник. Технічні науки. № 2(33). с. 161-172, 2019.
3. Smirnov, O., Kuznetsov, A., Kuznetsova., K. Synthesis of Discrete Signals with Improved Correlation Properties. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 281-299. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).
4. Смірнова Т.В., Солових Є.К., Смірнов О.А., Дреєв О.М. Побудова хмарних інформаційних технологій оптимізації технологічного процесу відновлення та зміцнення поверхонь деталей. Центральноукраїнський науковий вісник. Технічні науки. № 1(32). с. 184-194, 2019.
5. Смірнов О.А., Котелянець В.В. Стійкі до колізій стохастичні моделі функціонування безпроводових сенсорних мереж. Вісник інженерної академії України, №3, с. 145-152, 2018
6. O. Smirnov, O. Kovalenko, A. Kovalenko, S. Smirnov, V. Vialkova. The mathematical model of the testing technology for Dom Xss vulnerabilities. Scientific & practical cyber security journal (SPCSJ) Vol 2 Issue 1, 22-28 pp. [Електронний Журнал]. Georgia. Tbilisi: SCSA – 2018.
7. Oleksii Smirnov, Oleksandr Kovalenko, Jamil Al-Azzeh, Anna Kovalenko, Serhii Smirnov. Qualitative risk analysis of software development. Asian Journal of Information Technology. – Volume 17(3). – Medwell Journals. – 2018. – P. 218-230.
8. Смірнов О.А., Коваленко О.В., Коваленко А.С., Смірнов С.А. Розробка методу передтестової компіляції й розподілу доступу. Збірник наукових праць III міжнародної науково-практичної конференції “Інформаційна безпека та комп’ютерні технології”, м. Кропивницький. 19-20 квітня 2018р. – Кропивницький: ЦНТУ. – 2018. – С. 214-215
9. Smirnov Oleksii, Kovalenko Oleksandr, Kovalenko Anna, Smirnov Serhii. Method of testing the dom xss vulnerability. International Conference «information technologies, systems and networks ITS-2017». Chisinau,

- Republic of Moldova. 17 – 18 October 2017. – Chisinau: Academy of Sciences of Moldova, Military Academy of Armed Forces “Alexandru cel Bun”. 2017. P.7.
10. Смірнов О.А., Смірнов С.А., Коваленко О.В., Коваленко А.С. Технологія тестування DOM XSS уразливості. Науково-практичний журнал кібербезпеки (SPCSJ) № 1. [Електронний журнал]. Грузія. Тбілісі: SCSA - 2017.
 11. Смірнов О.А., Лисенко І.А. Інформаційна технологія проектування тестових наборів з урахуванням вимог до програмного забезпечення. Системи управління, навігації та зв'язку. – Випуск 4 (44). - Полтава: ПолтНТУ. - 2017. - С. 112-115.
 12. Смірнов О.А., Смірнов С.А., Рябой Д.К., Рябая О.В. Модель вузла комутації з відносними пріоритетами, резервуванням ресурсів і обліком реальної надійності обслуговуючих приладів. Збірник тез всеукраїнської науково-практичної інтернет-конференції «Автоматика та комп'ютерно-інтегровані технології у промисловості, телекомунікаціях, енергетиці та транспорті». м. Кропивницький. 16-17 листопада 2017 р. – Кропивницький: ЦНТУ. – 2017. – С. 198-199.
 13. Смірнов О.А., Коваленко О.В. Використання псевдобулевих методів бівалентного програмування для управління ризиками розробки програмного забезпечення. Системи управління, навігації та зв'язку. – Випуск 1 (37). - Полтава: ПолтНТУ. - 2016. - С. 98-103.
 14. Смірнов О.А., Лисенко І.А. Формалізація процесу проектування тестових наборів. Збірник наукових праць Харківського університету Повітряних Сил. Випуск 3 (48). - Харків: ХУПС. - 2016. - С.96-100.
 15. Смірнов О.А., Лисенко І.А. Удосконалення методу перевірки коректності таблиць рішень для подання тестових наборів. Збірник наукових праць "Системи обробки інформації". - Випуск 8 (145). - Х.: ХУПС - 2016. - С. 77-80.
 16. Смірнов О.А., Лисенко І.А. Розробка впорядкованих каскадних таблиць рішень із використанням матриць слідування. Збірник наукових праць "Системи обробки інформації". - Випуск 6 (143). - Х.: ХУПС - 2016. - С. 216-220.
 17. Смірнов О.А., Коваленко О.В., Якименко Н.М., Доренський О.П. Метод кількісної оцінки ризиків розроблення програмного забезпечення. Збірник наукових праць Харківського університету Повітряних Сил. Випуск 2 (47). - Харків: ХУПС. - 2016. - С. 128-133.
 18. Смірнов О.А., Коваленко О.В., Якименко Н.М., Доренський О.П. Метод якісного аналізу ризиків розроблення програмного забезпечення. Наука і техніка Збройних Сил України. – Випуск 2(23). - Харків: ХУПС. - 2016. - С. 150-158.
 19. Смірнов О.А., Коваленко О.В., Якименко Н.М., Доренський О.П. Проблеми аналізу та оцінки ризиків інформаційної діяльності. Збірник наукових праць "Системи обробки інформації". - Випуск 3 (140). - Х.: ХУПС - 2016. - С. 40-42.
 20. Смірнов О.А., Коваленко А.С., Коваленко О.В., Доренський О.П. Удосконалення методу технічного обслуговування об'єктів інтегрованої інформаційної системи. Системи озброєння і військова техніка. – Випуск 2(46) – Х.: ХУПС – 2016. – С. 103-107.
 21. Smirnov A.A., Kovalenko A.V. Kovalenko A.S. Dorensky A.P. Information model and its element for displaying information on technical condition of objects of integrated information system. International Journal of Computational Engineering Research (IJCER). – Volume 6, Issue 1. – India. Delhi. – 2016. – P. 21-27.
 22. Смірнов О.А., Євсєєв С.П., Король О.Г., Коваленко О.В., Коваленко А.С., Смірнов С.А. Архітектура мікропроцесорів та компонентів ЕОМ. Навчальний посібник – Кіровоград: Вид. Лисенко В.Ф., 2015. – 550 с.
 23. Смірнов О.А., Коваленко О.В., Мелешко Є.В., Константинова Л.В., Кожанова А.С. Інженерія програмного забезпечення. Навчальний посібник. За ред. О.А. Смірнова. – Кіровоград: КНТУ 2013. – 409с.