Secrets Chats Protocol

Nie istnieją obecnie uniwersalne mechanizmy ochrony danych o odpowiedniej jakości, które mogłyby łatwo zostać zintegrowane przez twórców aplikacji mobilnych. Opracowane rozwiązania są specyficzne dla danego systemu operacyjnego i najczęściej zależne bezpośrednio od jego wersji. Naszym celem było wytworzenie łatwego w implementacji, uniwersalnego systemu ochrony sekretów aplikacji składowanych w ramach jednego wspólnego kontenera.

Michał Glet

absolwent Wydziału Cybernetyki Wojskowej Akademii Technicznej (kierunek Informatyka, specjalność Kryptologia). Asystent badawczo-dydaktyczny w Instytucie Matematyki i Kryptologii WAT. Autor i współautor kilkunastu publikacji naukowych z zakresu kryptologii, cyberbezpieczeństwa oraz złośliwego oprogramowania. Twórca i współtwórca algorytmów i rozwiązań kryptograficznych oraz steganograficznych wyróżnianych na międzynarodowych wystawach wynalazczości. Ekspert w zakresie tworzenia i eksploatacji systemów informatycznych oraz rozwiązań z zakresu bezpieczeństwa, dostępności oraz poufności danych.

Kamil Kaczyński

absolwent Wydziału Cybernetyki Wojskowej Akademii Technicznej (kierunek Informatyka, specjalność Kryptologia). Asystent badawczo-dydaktyczny w Instytucie Matematyki i Kryptologii WAT, aktywny członek International Association for Cryptologic Research. Autor i współautor kilkunastu publikacji naukowych z zakresu kryptologii i steganografii. Twórca i współtwórca algorytmów i rozwiązań kryptograficznych oraz steganograficznych wyróżnianych na międzynarodowych wystawach wynalazczości. Ekspert w zakresie tworzenia i eksploatacji systemów zapewniających integrację z technologiami blockchain i mechanizmami kryptograficznymi pozwalającymi na zapewnienie poufności, integralności i dostępności danych.

Obecnie 3,5 mld użytkowników smartfonów korzysta z aplikacji, z których każda przechowuje potencjalnie wrażliwe dane. Aplikacje, które przetwarzają dane medyczne, finansowe lub osobiste (np. komunikatory mobilne) należą do grupy szczególnie zagrożonych i powinny być chronione z wykorzystaniem odpowiednio efektywnych metod. Tymczasem przykładowo niemalże 40% spraw rozwodowych, które mają miejsce we Włoszech, zawiera w materiałach dowodowych wiadomości wymieniane przez niewiernych małżonków za pośrednictwem aplikacji Whatsapp: http://www.thetimes.co.uk/tto/news/world/europe/article4262527.ece.

Wykorzystywany przez WhatsApp mechanizm ochrony danych nie jest więc wystarczająco bezpieczny, bo umożliwia łatwe odzyskanie historii prowadzonej komunikacji.

Podobne bolączki dotyczą aplikacji, których twórcy deklarują pełne skupienie na prywatności i bezpieczeństwie danych. Przed trzema laty wskazywaliśmy na luki bezpieczeństwa mechanizmu składowania danych Signal[1], który wykorzystuje jedynie mechanizmy systemu operacyjnego – w tym przypadku Android Keystore do przechowywania kluczy chroniących bazę danych. Ataki na mechanizm Keystore także były przedmiotem wielu publikacji naukowych ukazujących wpływ wykorzystania wybranych schematów kryptograficznych na bezpieczeństwo całego rozwiązania, w tym brak zachowania integralności szyfrogramu, co pozwala na zredukowanie długości wykorzystywanego klucza symetrycznego.

> *Należy także zwrócić uwagę, że mechanizmy systemu operacyjnego wiążą klucz główny urządzenia z wprowadzonym przez użytkownika sekretem – wzorem blokady, hasłem, kodem PIN, biometrią.*

Proces odzyskiwania sekretu może być przeprowadzony z wykorzystaniem oprogramowania śledczego, takiego jak Cellebrite UFED Ultimate, tym samym znacząco redukując poziom skomplikowania procesu odzyskiwania danych. W opracowaniu „Analysis of secure key storage solutions on Android"[2] autorzy dokonali analizy różnych metod bezpiecznego przechowywania kluczy, wskazując przy tym, iż na dzień opublikowania pracy żadna z badanych metod nie gwarantowała odpowiedniego poziomu ochrony. Większość z nich spełniała dwa z trzech wymagań – powiązanie z aplikacją, powiązanie z urządzeniem lub wymaganie świadomej zgody użytkownika na dostęp do danych. Powiązanie z aplikacją oznacza, iż dany sekret jest dostępny tylko dla wybranej instancji aplikacji, powiązanie z urządzeniem oznacza, iż sekret może być odczytany tylko przez dane urządzenie. Ostatnie wymaganie – świadomość użytkownika oznacza, iż klucz może zostać udostępniony jedynie wtedy, kiedy użytkownik wykona akcję potwierdzającą udostępnienie przechowywanego klucza kryptograficznego. W tej pracy zaproponowana została metoda, która pozwala na połączenie wszystkich trzech wymagań, tym samym stanowiąc rozwiązanie efektywne, także w przypadku występowania luk bezpieczeństwa mechanizmów systemowych.

Nasza propozycja

W celu ochrony sekretu aplikacji składowanych w ramach jednego wspólnego kontenera postanowiliśmy wykorzystać m.in. schemat podziału sekretu. Rozwiązanie takie jest zupełnie niezależne od wykorzystywanego systemu operacyjnego czy też zainstalowanych rozwiązań sprzętowych. Może być zatem z powodzeniem wykorzystywane na starszych urządzeniach. Stworzyliśmy uniwersalny mechanizm, który na bazie dostarczonego przez użytkownika hasła tworzy bezpieczny kontener dla przechowywanych danych. Rozwiązanie to może być wykorzystywane np. do ograniczania dostępu do zbiorów danych przechowywanych przez aplikację i z powodzeniem zastępować mechanizmy logiczne.

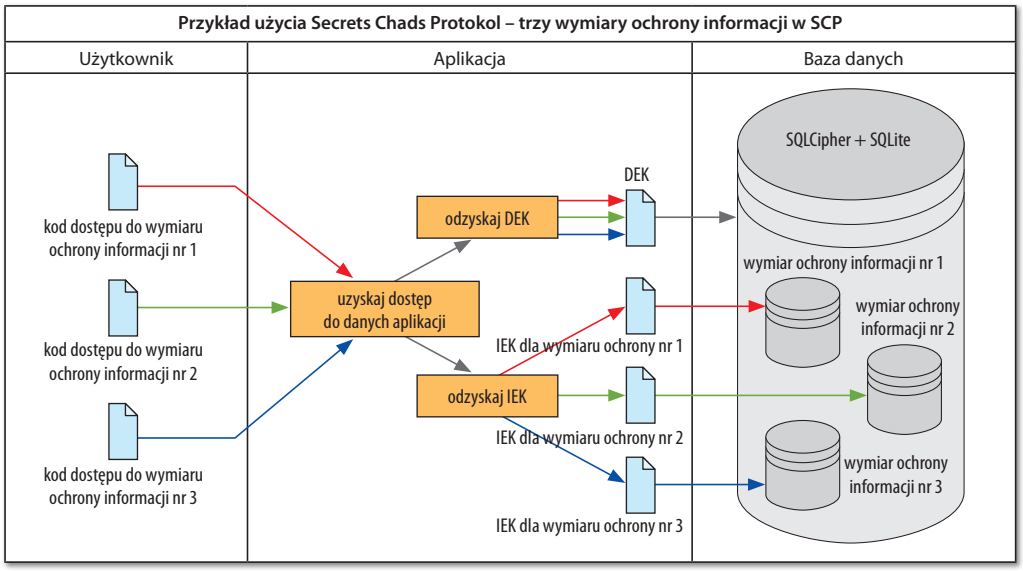
Na potrzeby opisu Secret Chats Protocol (SCP) wprowadzone zostały poniższe pojęcia:

- **Wymiar ochrony informacji** – zbiór danych, które podlegać będą ochronie i do których dostęp zostanie uzyskany za pomocą pojedynczego kodu bezpieczeństwa.
- **Database Encryption Key (DEK)** – sekret wspólny dla wszystkich wymiarów ochrony informacji.
- **Information Encryption Key (IEK)** – sekret unikalny dla każdego z wymiarów ochrony informacji.

1 K. Kaczyński, Security analysis of Signal Android database protection mechanisms. *International Journal on Information Technologies and Security*, Vol. 11, No 4 (2019), pp. 63-70.

M. Glet, Security analysis of Signal data storage mechanisms in iOS version. *International Journal on Information Technologies and Security*, Vol. 11, No 4 (2019), pp. 71-88.

2 T. Cooijmans, J. de Ruiter, E. Poll, Analysis of secure key storage solutions on Android. SPSM 2014: Proceedings of the 4th ACM Workshop on Security and Privacy in Smartphones and Mobile Devices, pp. 11-20.



Architektura zakłada wykorzystanie hasła o wysokiej entropii, które jest krytyczne dla odpowiedniego poziomu bezpieczeństwa rozwiązania.

Kluczową częścią naszego unikatowego mechanizmu jest możliwość tworzenia wielu wymiarów ochrony informacji. Z punktu widzenia użytkownika, każdy wymiar wygląda jak oddzielna baza danych, bez możliwości dostępu do postaci jawnej danych innych wymiarów. Dostęp do danych danego wymiaru jest możliwy jedynie poprzez podanie poprawnego hasła zapewniającego dostęp do tego wymiaru – każdy wymiar posiada inne hasło dostępowe.

Idea jest bardzo prosta – rozwiązanie SCP ma umożliwić użytkownikom przechowywanie danych aplikacji w wielu wymiarach ochrony informacji. W danym momencie użytkownik uzyskuje dostęp do danych z pojedynczego wymiaru ochrony informacji. Co istotne, SCP nie ujawnia liczby wymiarów bezpieczeństwa, które zostały utworzone przez użytkownika.

W protokole SCP zakładamy, że jeden udział posiada zawsze aplikacja. Własności algorytmu Shamira sprawiają, że uzyskanie przez atakującego dostępu do tego udziału (np. poprzez analizę wsteczną kodu oraz danych aplikacji) nie wpływa na bezpieczeństwo sekretu. Drugi udział, niezbędny do odtworzenia wartości DEK, odzyskiwany jest z kodu dostępu wprowadzonego przez użytkownika. Mechanizm odzyskiwania DEK w SCP od klasycznej wersji algorytmu podziału sekretu Shamira odróżniają m.in. możliwości:

- stosowania dowolnych, wybranych przez użytkownika kodów dostępu;
- dynamicznego dodawania kolejnych udziałów (wymiarów ochrony informacji) bez konieczności modyfikacji już istniejących.

SCP wykorzystuje algorytm podziału sekretu Shamira[3] typu (2,k) (oznaczenie *SSS(2,k)*). Algorytm ten pozwala na ukrycie sekretu w *k* udziałach oraz odzyskanie go przy posiadaniu dowolnych *2* z nich. Pomysł Shamira bazuje na interpolacji wielomianowej oraz fakcie, że posiadając dowolne dwa punkty z przestrzeni R^2 (płaszczyzny euklidesowej) $(x_1,y_1),(x_2,y_2)$, takie, że $x_1 \neq x_2$, można skonstruować tylko i wyłącznie jeden wielomian $f(x) \in R[x]$ stopnia 1 taki, że $f(x_1)=y_1$ oraz $f(x_2)=y_2$.

Ten algorytm w SCP jest podstawą mechanizmu do odzyskiwania wartości DEK. W mechanizmie tym wykorzystujemy operacje w ciele GF(p) oraz wielomian postaci $f(x)=(a_0+a \cdot x) \pmod p$, gdzie *p* jest liczbą pierwszą. Wartość DEK jest ukrywana w wartości współczynnika $a_0=\tau(DEK)$. Secret Chats Protocol dzieli sekret na k części poprzez wyznaczenie współrzędnych punktów $(x_1,f(x_1))$, $(x_2,f(x_2))$,…, $(x_{(k-1)}, f(x_{(k-1)}))$, $(x_k,f(x_k))$. Posiadając dowolne dwa punkty, można odzyskać wielomian f(x) oraz wartość współczynnika a_0 i tym samym odzyskać wartość DEK.

Secret Chats Protocol dla każdego wymiaru ochrony informacji tworzy nowy udział w rozumieniu algorytmu podziału sekretu Shamira. W związku z tym instancja aplikacji posiadającej m wymiarów ochrony informacji wykorzystuje m+1 udziałów – jeden dla aplikacji oraz po jednym dla każdego wymiaru ochrony informacji. Dzięki temu, niezależnie od tego, która para udziałów zostanie wykorzystana (który kod dostępu zostanie użyty) do odtworzenia DEK, jego wartość zawsze będzie taka sama.

3 A. Shamir, How to share a secret. Communications of the ACM 22.11 (1979), pp. 612-613.

DEK

Zgodnie z tym, co zostało już przedstawione, wartość DEK (Database Encryption Key) jest taka sama dla każdego użytego kodu dostępu, czyli dla każdego wymiaru ochrony informacji. Jednym z założeń podczas tworzenia SCP było zapewnienie łatwej integracji z aktualnie istniejącymi aplikacjami. Część z nich, na potrzeby zabezpieczania składowanych danych, korzysta z szyfrowanych baz danych oraz rozwiązań typu SQLCipher. W tym przypadku wartość DEK może zostać wykorzystana do odzyskania klucza szyfrującego bazę danych. Rzeczywiste wykorzystanie DEK zależy tak naprawdę od twórców aplikacji. Prezentowany protokół SCP daje im sekret wspólny dla wszystkich wymiarów ochrony informacji, czyli dla wszystkich kodów dostępu zdefiniowanych np. przez użytkownika aplikacji.

Proces odzyskiwania DEK bazuje na algorytmie SSS(2,k) oraz autorskim algorytmie Common Container Algorithm (CCA). Wartość DEK ukrywana jest z wykorzystaniem algorytmu *SSS(2,k)*, natomiast algorytm *CCA* jest wykorzystywany m.in. do:

1. Zapewnienia użytkownikom możliwości stosowania własnych, dowolnych, kodów dostępu.
2. Przechowywania wszystkich danych związanych z *SSS(2,k)* w jednym miejscu.
3. Zapewniania możliwości łatwego składowania w istniejących aplikacjach.
4. Zapewnienia możliwości łatwego dodawania kolejnych wymiarów ochrony informacji, np. bez konieczności modyfikacji istniejących już wymiarów oraz kodów dostępu.

Szczegółowy opis kolejnych kroków algorytmu CCA oraz całego protokołu SCP można znaleźć w naszym artykule „Secret Sharing Scheme for Creating Multiple Secure Storage Dimensions for Mobile Applications”[4].

Co osiągnęliśmy?

Zaproponowany mechanizm ochrony danych pozwala na zwiększenie poziomu bezpieczeństwa danych aplikacji przechowywanych lokalnie. Pozwala on twórcom aplikacji na wykorzystanie zarówno mechanizmów dostarczanych przez system operacyjny, bezpiecznych modułów sprzętowych, jak i sekretu znanego tylko użytkownikowi. Dzięki tym założeniom użytkownik zawsze będzie musiał wyrazić zgodę na wykorzystanie kluczy kryptograficznych, co utrudnia wykonanie skutecznego ataku na dane aplikacji.

IEK

Secret Chats Protocol nie określa kroków ani procedur odtwarzania wartości IEK (Information Encryption Key). Wszystko zatem zależy od twórców aplikacji. SCP dostarcza aplikacji różne kody dostępu dla różnych wymiarów ochrony informacji. Na tej podstawie aplikacja powinna odzyskać np. klucze szyfrujące umożliwiające uzyskanie dostępu do danych przechowywanych w wybranym wymiarze ochrony informacji. W tym celu z powodzeniem można skorzystać np. z algorytmów opisanych w RFC-8018[5].

SCP definiuje także zupełnie nową jakość w dziedzinie ochrony danych przechowywanych we wspólnym kontenerze – pozwala na wykorzystanie wspólnej bazy danych, wewnątrz której istnieje możliwość wydzielenia dostępu do danych dla użytkowników znających sekret. Obecnie tego rodzaju ochrona jest realizowana z wykorzystaniem logiki aplikacji – dobrym przykładem jest tutaj komunikator Viber oraz jego funkcjonalność ukrytych czatów. Przy wykorzystaniu zaproponowanego rozwiązania możliwe jest tworzenie dowolnej liczby wymiarów ochrony informacji, bez ryzyka ujawnienia tej wartości np. poprzez analizę ustawień aplikacji. Rozwiązanie to może być szczególnie przydatne, gdy jedynie część historii prowadzonej komunikacji ma być prezentowana.

Jak SCP chroni przed Pegasusem?

W ostatnim czasie, m.in. w wyniku różnych medialnych doniesień, rośnie świadomość użytkowników związana z bezpieczeństwem oraz poufnością danych przechowywanych na urządzeniach mobilnych. Najczęściej omawianą oraz analizowaną publicznie aplikacją zagrażającą naszym danym oraz naszej prywatności jest Pegasus (P).

Pokusiliśmy się o estymację skuteczności SCP przy pewnych założeniach związanych ze sposobem działania oprogramowania szpiegującego – jej wyniki przedstawiamy w tabeli. Należy jednak pamiętać, że wykorzystanie potencjału Secrets Chats Protocol zależy od deweloperów oraz konkretnych aplikacji.

4 M. Glet, K. Kaczyński, Secret Sharing Scheme for Creating Multiple Secure Storage Dimensions for Mobile Applications. *International Journal on Information Technologies and Security*, Vol. 12, No 4 (2020), pp. 83-102.

5 *Moriarty, Kathleen, Burt Kaliski, and Andreas Rusch*. Pkcs# 5: Password-based cryptography specification version 2.1. Internet Engineering Task Force (IETF) (2017).

Pomysł na Secrets Chats Protocol opublikowaliśmy w 2020 r. Został doceniony m.in. przez społeczność badawczo-naukową, co przyniosło nam medale na targach wynalazczości:

1. **Złoty medal Prix Eiffel 2021 za „Secret Chats Protocol", mgr inż. Michał Glet, mgr inż. Kamil Kaczyński, Lyon, Francja;**
2. **Złoty medal Inova Croatia 2020 za „Secret Chats Protocol", nagroda specjalna MIIA, mgr inż. Michał Glet, mgr inż. Kamil Kaczyński, Zagrzeb, Chorwacja;**
3. **Złoty medal Intarg 2021 za „Secret Chats Protocol", mgr inż. Michał Glet, mgr inż. Kamil Kaczyński, Katowice, Polska;**
4. **Złoty medal Euroinvent 2021 za „Secret Chats Protocol", mgr inż. Michał Glet, mgr inż. Kamil Kaczyński, Bukareszt, Rumunia.**

Załóżmy zatem, że:
1. Secrets Chats Protocol zaimplementowany został w aplikacji A w sposób poprawny, a jego użycie jest zgodne z proponowanym przez nas podejściem (w pełni szyfrowana baza danych z wykorzystaniem DEK, szyfrowane dane w wymiarach ochrony informacji z wykorzystaniem IEK).
2. Na urządzeniu mobilnym zainstalowane zostało oprogramowanie szpiegujące P.

Rozpatrzmy następujące założenia co do korzystania z aplikacji A:
A1. Użytkownik nie korzystał z aplikacji A od momentu infekcji oprogramowaniem P.
A2. Użytkownik korzystał z niektórych wymiarów ochrony informacji w aplikacji A od momentu infekcji oprogramowaniem P.
A3. Użytkownik korzystał ze wszystkich wymiarów ochrony informacji w aplikacji A od momentu infekcji oprogramowaniem P.

Rozpatrzmy następujące założenia w kontekście oprogramowania szpiegującego P:
P1. Oprogramowanie P rejestruje w trybie ciągłym zawartość ekranu urządzenia mobilnego oraz aktywności interfejsu wejściowego (np. klawiatury ekranowej).
P2. Oprogramowanie P umożliwia utworzenie rzutu zawartości pamięci operacyjnej wszystkich procesów związanych z aplikacją A.
P3. Oprogramowanie P posiada dedykowaną funkcjonalność do analizy i monitorowania aktywności aplikacji A.

	A1	A2	A3
P1	Brak możliwości uzyskania dostępu do danych	Możliwość uzyskania dostępu do danych z niektórych wymiarów ochrony informacji, aczkolwiek wymaga (manualnej?) analizy zawartości nagranego ekranu oraz wprowadzanych danych	Możliwość uzyskania dostępu do danych ze wszystkich wymiarów ochrony informacji, aczkolwiek wymaga (manualnej?) analizy zawartości nagranego ekranu oraz wprowadzanych danych
P2	Brak możliwości uzyskania dostępu do danych	Możliwość uzyskania dostępu do danych z niektórych wymiarów ochrony informacji, aczkolwiek wymaga (manualnej?) analizy zawartości utworzonego zrzutu pamięci	Możliwość uzyskania dostępu do danych ze wszystkich wymiarów ochrony informacji, aczkolwiek wymaga (manualnej?) analizy zawartości utworzonego zrzutu pamięci
P3	Brak możliwości uzyskania dostępu do danych	Łatwa możliwość uzyskania dostępu do danych z niektórych wymiarów ochrony informacji	Łatwa możliwość uzyskania dostępu do danych ze wszystkich wymiarów ochrony informacji

Należy zaznaczyć, że:
1. „Brak możliwości uzyskania dostępu do danych" nie oznacza całkowitego bezpieczeństwa naszych danych. Nasze dane są tak bezpieczne jak bezpieczne są np. dane uwierzytelniające dostęp do nich, czyli wymiary ochrony informacji są tak bezpieczne jak bezpieczne są używane kody dostępu do nich (odpowiednia długość, zestaw używanych znaków itp.).
2. „Możliwość uzyskania dostępu do danych (...) aczkolwiek wymaga (manualnej?) analizy zawartości nagranego ekranu oraz wprowadzanych danych" oznacza, że operator oprogramowania P będzie musiał wykonać czasochłonną analizę uzyskanych danych (np. wideo z ekranem, logi z klawiatury), aby uzyskać dostęp do danych składowanych w aplikacji.
3. „Możliwość uzyskania dostępu do danych (...) aczkolwiek wymaga (manualnej?) analizy zawartości utworzonego zrzutu pamięci" oznacza, że operator oprogramowania P będzie musiał wykonać czasochłonną analizę uzyskanych danych (zrzuty pamięci), aby uzyskać dostęp do danych składowanych w różnych wymiarach ochrony informacji.

Warto podkreślić, że wszystkie czynności dotyczące oprogramowania P oraz aplikacji A, w których konieczna jest manualna aktywność operatora oprogramowania P, nie są odpowiednie do przeprowadzania inwigilacji na masową skalę.