Politechnika Krakowska

Wydział Informatyki i Telekomunikacji

Informatyka Cyberbezpieczeństwo

Sprawozdanie z przedmiotu:

Bezpieczeństwo systemów informatycznych

Projekt 5

Semestr: letni(II), 2022/2023

Kierunek: Informatyka II stopień, niestacjonarne

Rok: I

Grupa: CY1

Wykonał:

Artur Hamernik 148942

1. Bouncy Castle

Bouncy Castle ma szeroki wybór algorytmów i łatwe w użyciu API, ale wymaga pewnego wysiłku w zrozumieniu i może być nieco mniej wydajny w niektórych przypadkach. Zależy to od indywidualnych potrzeb i preferencji programisty.

Plusy:

- Oferuje szeroki zakres algorytmów szyfrowania, podpisów cyfrowych, certyfikatów i innych narzędzi związanych z bezpieczeństwem. Daje to programistom dużą elastyczność i możliwość wyboru odpowiednich algorytmów dla swoich konkretnych potrzeb.
- Biblioteka ma intuicyjne API, które ułatwia programistom korzystanie z różnych funkcji kryptograficznych. Dzięki temu można szybko i efektywnie implementować bezpieczeństwo w aplikacjach.
- Bouncy Castle ma duże wsparcie społeczności, co oznacza, że jest regularnie aktualizowany i udostępnia najnowsze funkcje i poprawki błędów. Istnieje również dostęp do forum dyskusyjnego i innych zasobów, które pomagają programistom w rozwiązywaniu problemów i wymianie informacji.

Minusy:

- Pomimo że Bouncy Castle oferuje obszerną dokumentację, niektóre jej części mogą być trudne do zrozumienia dla początkujących użytkowników. Niektóre aspekty biblioteki mogą wymagać więcej czasu i wysiłku w celu pełnego zrozumienia i skutecznego korzystania z nich.
- W niektórych przypadkach Bouncy Castle może być nieco mniej wydajny w porównaniu do innych bibliotek kryptograficznych. Odpowiedni dobór algorytmów i optymalne wykorzystanie funkcji mogą być ważne dla osiągnięcia dobrej wydajności.
- Bouncy Castle jest rozbudowaną biblioteką, co oznacza, że ma wiele zależności i może wpłynąć na złożoność i ilość kodu. To może mieć znaczenie dla projektów o ograniczonym rozmiarze lub zależnych od szybkiego ładowania.

2. Tink

Tink oferuje wysoki poziom bezpieczeństwa, prostotę użycia i modularność, ale może mieć mniejszą liczbę dostępnych algorytmów w porównaniu do niektórych innych bibliotek. Wybór Tink zależy od konkretnych wymagań projektu i preferencji programisty.

Plusy:

- Tink został stworzony przez zespół Google, który jest znany z dbałości o bezpieczeństwo. Biblioteka została zaprojektowana z myślą o minimalizowaniu błędów programistycznych i potencjalnych luk w zabezpieczeniach. Oferuje silne algorytmy kryptograficzne i solidne metody implementacji.
- Tink został zaprojektowany w taki sposób, aby ukryć złożoność kryptografii i dostarczyć prosty i spójny interfejs programistyczny. Dzięki temu programiści mogą łatwo korzystać z różnych funkcji kryptograficznych, nawet jeśli nie mają głębokiej wiedzy na temat kryptografii.
- Tink jest modułową biblioteką, co oznacza, że można wybrać tylko te komponenty, które są potrzebne dla danego projektu. To pozwala na bardziej elastyczną integrację i minimalizację nadmiaru kodu. Biblioteka również oferuje wsparcie dla różnych platform i języków programowania.

Minusy:

- Mimo że Tink oferuje wiele algorytmów kryptograficznych, nie jest tak rozbudowany jak niektóre inne biblioteki. Niektóre bardziej zaawansowane lub mniej popularne algorytmy mogą nie być dostępne w Tink.
- W niektórych przypadkach nowe wersje Tink mogą nie być w pełni kompatybilne z wcześniejszymi wersjami. To oznacza, że aktualizacja biblioteki może wymagać pewnych zmian w kodzie istniejących aplikacji.
- Chociaż Tink oferuje wsparcie dla wielu popularnych platform i języków programowania, nie jest dostępny na każdej z nich. Może to być ograniczeniem dla projektów działających na mniej popularnych platformach.

3. Java Security

Java Security oferuje wbudowane mechanizmy bezpieczeństwa, silne wsparcie społeczności i dobrą dokumentację, ale może być złożone i mniej elastyczne w porównaniu do niektórych zewnętrznych bibliotek. Wybór Java Security zależy od specyficznych potrzeb projektu i doświadczenia programisty.

Plusy:

- Java Security jest częścią platformy Java, co oznacza, że ma wbudowane mechanizmy bezpieczeństwa. Oferuje funkcje takie jak uwierzytelnianie, autoryzacja, kontrola dostępu, szyfrowanie i zarządzanie kluczami, które mogą być wykorzystane w aplikacjach Java.
- Java jest jednym z najpopularniejszych języków programowania, co oznacza, że Java Security ma szerokie wsparcie społeczności. Istnieje wiele zasobów, forum dyskusyjnych i gotowych rozwiązań, które mogą pomóc programistom w rozwiązywaniu problemów związanych z bezpieczeństwem i wymianie informacji.
- Java Security jest dobrze udokumentowane, co ułatwia programistom zrozumienie i korzystanie z różnych funkcji biblioteki. Istnieje również dostęp do narzędzi i bibliotek zewnętrznych, które mogą ułatwić implementację bezpieczeństwa w aplikacjach Java.

Minusy:

- Java Security może być złożonym tematem, zwłaszcza dla początkujących programistów. Implementacja i konfiguracja odpowiednich mechanizmów bezpieczeństwa w aplikacji Java może wymagać pewnego wysiłku i wiedzy.
- W niektórych przypadkach Java Security może być mniej elastyczne niż niektóre zewnętrzne biblioteki. Może to prowadzić do pewnych ograniczeń w dostępnych funkcjach i konfiguracji.
- Java Security jest częścią platformy Java, co oznacza, że ma pewien rozmiar i może wpływać na wydajność aplikacji. W przypadku projektów o ograniczonym rozmiarze lub wymagających optymalnej wydajności, może to być istotne.

Biblioteka i algorytm	Czas wykonania [MS]
BouncyCastle - AES-CBC	65
BouncyCastle - Blowfish-CBC	2
BouncyCastle - RSA	655
BouncyCastle - DSA	329
BouncyCastle – SHA256	6
Tink - AES	291
Java Security - RSA	118
Tink - ECDSA_P256	89
Tink - SHA256	10