# Implementacja mechanizmu AACS (Advanced Access Control System)

Protokoły kryptograficzne

Marcin Jędrzejewski

prowadzący: mgr inż. Albert Sitek

Semestr 15Z

# Spis treści

1.	. Wpr	owadzenie	3	
		acowanie teoretyczne		
	2.1.	AACS	3	
	2.2.	Indeks pojęć	4	
3.	. Kon	cepcja rozwiązania	4	
4.	. Wyr	niki projektu	5	
	4.1.	Zakres realizacji	5	
	4.2.	Interfejs aplikacji	6	
Administrator  Enkryptor		6		
		yptor	8	
	Dek	ryptor	9	
В	Bibliografia			

## 1. Wprowadzenie

Przedmiotem projektu jest implementacja mechanizmu AACS. Jest to technologia mająca na celu ochronę nośników HD DVD i Blu-ray przed odtwarzaniem ich treści na nieautoryzowanych urządzeniach oraz przed nieautoryzowanym kopiowaniem. Zasada działania mechanizmu została szerzej omówiona w dalszej części dokumentu.

## 2. Opracowanie teoretyczne

#### 2.1. AACS

AACS opiera się na szyfrowaniu zawartości płyty kluczami unikalnymi dla znajdujących się na nośniku utworów (**Title Keys**). Klucze te są zaszyfrowane z użyciem klucza unikalnego dla danego nośnika (**Volume Unique Key**), który jest uzyskiwany za pomocą algorytmu **AES-G** z identyfikatora nośnika (**Volume ID**) i klucza medium (**Media Key**). *Media Key* jest wynikiem poprawnego procesowania struktury **MKB** (**Media Key Block**) przy pomocy zestawu kluczy urządzenia (**Device Keys Set**). Zarządzaniem i dystrybucją kluczy zwartych w *MKB* oraz unikalnych kluczy urządzeń zarządza instytucja AACS License Administrator (AACS LA).

MKB ma postać drzewa **subset-difference tree.** Jest to rozległe drzewo binarne, w którym istotną rolę odgrywają jego poddrzewa (większe), które to z kolei posiadają wykluczone jedno mniejsze poddrzewo - stąd nazwa **subset-difference**. W korzeniu każdego większego poddrzewa ze zdefiniowanego klucza początkowego zostają wygenerowane za pomocą algorytmu AES-3G trzy kolejne klucze - pierwszy i trzeci posłużą do przeprowadzenia tej samej operacji odpowiednio w lewym i prawym węźle potomnym, natomiast drugi stanowi **Processing Key** zawarty w danym węźle. Procedurę kontynuuje się, aż do liści większego poddrzewa, lecz z pominięciem mniejszego poddrzewa wykluczonego. W korzeniu tego mniejszego poddrzewa wykorzystuje się jedynie wygenerowany w wyżej opisany sposób *Processing Key*, którym zaszyfrowany zostaje *Media Key* i w tej postaci zostaje on zapisany w bloku *MKB*.

Każde autoryzowane urządzenie otrzymuje zestaw kluczy *Device Keys*. Są to klucze, które w każdym węźle na ścieżce od korzenia do jednego z liści większego poddrzewa (określonego dla każdego urządzenia) były generowane dla węzła potomnego, który nie leży na tej ścieżce. Tym sposobem, każde urządzenie jest w stanie uzyskać *Processing Key* leżący w dowolnym węźle większego poddrzewa, za wyjątkiem węzłów leżących na ścieżce od korzenia do liścia określonego dla danego urządzenia. Dzięki temu, aby uniemożliwić danemu urządzeniu odczytanie klucza *Media Key*, wystarczy zaszyfrować go z użyciem *Processing Key* leżącym w jego liściu. Można też użyć *Processing* 

Key leżącego w wyżej położonym węźle, lecz możliwość dekrypcji Media Key stracą wtedy wszystkie urządzenia, dla których określone zostały liście stanowiące węzły potomne wybranego węzła.

Mechanizm AACS rozpoznaje w urządzeniu zarówno **napęd** (*Drive*) jak i **host** (*Host*) – czyli oprogramowanie z niego korzystającego. Zarówno napęd jak i host muszą posiadać certyfikaty podpisane przez AACS LA, by móc przystąpić do dekrypcji danych z nośnika. Utrudnia to nieautoryzowane kopiowanie zawartości nośnika poprzez wykorzystanie tzw. *analogowej dziury* (ang. *analog hole*) – nieautoryzowane oprogramowanie do odtwarzania mogłoby posłużyć do skopiowania danych. W bloku MKB znajdują się listy odwołanych certyfikatów napędów (*Drive Revocation List*) i hostów (*Host Revocation List*). Zarówno napędy jak i hosty zobowiązane są do aktualizowania listy odwołanych certyfikatów do wersji najbardziej aktualnej spośród "widzianych" na procesowanych nośnikach.

Wspomniany wcześniej *Volume ID* jest identyfikatorem nośnika zapisanym nań w sposób uniemożliwiający jego skopiowanie – poprzez zapisanie w obszarze pamięci niedostępnym do zapisu w urządzeniach użytkowych – a tym samym kopiowanie nośników "bit po bicie" i uzyskanie dwóch identycznych kopii jednego nośnika. Przy próbie skopiowania nośnika w taki sposób, różna od oryginalnej wartość *Volume ID* uniemożliwiałby zdekodowanie *Volume Unique Key* koniecznego do odczytu zawartości dysku.

#### 2.2. Indeks pojęć

- Title Keys klucze potrzebne bezpośrednio do (de)szyfrowania zawartości nośnika
- Volume Unique Key klucz potrzebny do (de)szyfrowania Title Keys
- Volume ID unikalny identyfikator nośnika, potrzebny do uzyskania Volume Unique Key
- Media Key klucz, z którego użyciem przy udziale Volume ID powstaje Volume Unique Key
- MKB Media Key Block struktura używana do propagacji Media Keys i realizacji autoryzacji według mechanizmu AACS
- Device Keys Set zestaw kluczy unikalnych urządzenia odtwarzającego
- Subset-difference tree struktura drzewiasta MKB zawierająca Media Keys
- Napęd (Drive) fizyczny napęd czytający nośniki
- Host oprogramowanie używane do odtwarzania nośników

# 3. Koncepcja rozwiązania

Wynikiem projektu ma być program, składający się z trzech modułów:

- Administrator moduł pełniący role AACS LA jest odpowiedzialny za generowanie MKB i kluczy dla symulowanych urządzeń deszyfrujących; podpisywanie certyfikatów napędów i hostów; rewokację kluczy urządzeń, certyfikatów napędów i certyfikatów hostów
- Enkryptor moduł odpowiedzialny za enkrypcję nośników
- Dekryptor moduł odpowiedzialny za symulację urządzenia do odtwarzania nośników przy pomocy napędu i hosta

Program zostanie wytworzony z użyciem języka JAVA.

## 4. Wyniki projektu

#### 4.1. Zakres realizacji

W ramach projektu zaimplementowałem istotę mechanizmu AACS, tj. algorytm **subset- difference tree**.

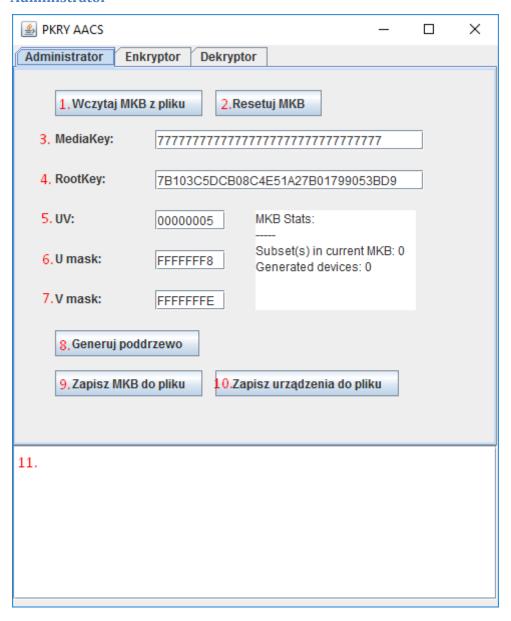
Moduł **Administrator** odpowiada za generowanie struktury MKB zawierającej odpowiedni Media Key i umożliwia zapisanie do osobnych plików tejże struktury oraz danych wymaganych przez autoryzowane urządzenia do odczytania zawartości nośników zaszyfrowanych z jej użyciem.

Moduł **Enkryptor** umożliwia generowanie uproszczonego modelu zaszyfrowanego dysku. Model ten pomija szyfrowanie z użyciem Volume ID i Volume Key (które stanowią jedynie element szyfrowania treści w ten sam sposób kolejnymi kluczami i poza zróżnicowaniem wyników dla różnych nośników nie wprowadzają nic nowego do istoty samego mechanizmu) i szyfruje treść wybranego pliku bezpośrednio za pomocą wskazanego Media Key. Blok MKB nie zostaje zapisany pliku wynikowym.

Moduł **Dekryptor** umożliwia odszyfrowanie pliku (modelu nośnika) zaszyfrowanego z użyciem określonego Media Key - plik ten generowany jest przez moduł *Enkryptor*. Moduł wymaga wczytania pliku struktury MKB zawierającej wymagany Media Key oraz urządzenia zdolnego daną strukturę przetworzyć - pliki te generowane są przez moduł *Administrator*.

#### 4.2. Interfejs aplikacji

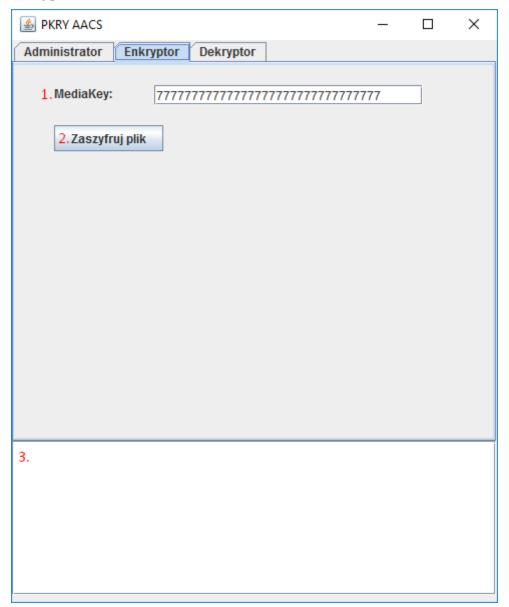
#### **Administrator**



- 1. Przycisk umożliwia wczytanie wcześniej wygenerowanej struktury MKB z pliku.
- 2. Przycisk umożliwia wyczyszczenie aktualnie tworzonej struktury MKB.
- 3. Wartość pola określa klucz Media Key, którym zaszyfrowane zostaną dane.
- 4. Wartość pola klucz, z którego zostaną wygenerowane kolejne trzy klucze w korzeniu większego poddrzewa (wyjaśnione w Opracowaniu teoretycznym)
- 5. Wartość UV określa ścieżkę do korzenia mniejszego poddrzewa wykluczonego. W postaci binarnej patrząc od najbardziej znaczącego bitu każde 0 oznacza wybór lewego potomka a 1 prawego potomka na drodze od korzenia całego drzewa subset-difference. Po dotarciu do

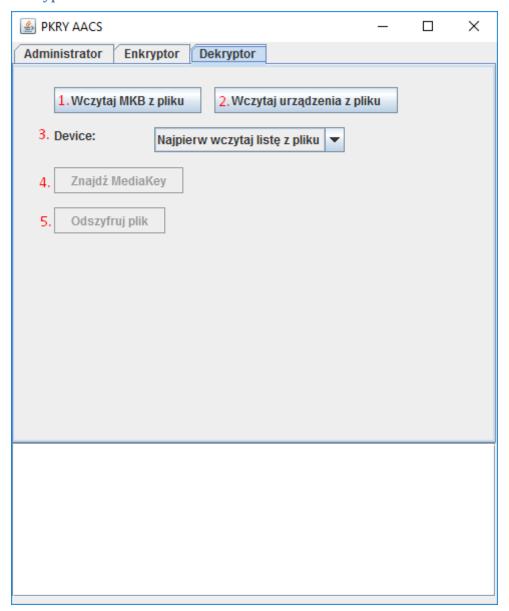
- wykluczonego poddrzewa kolejny bit wartości UV zostaje ustawiony na 1, a wszystkie kolejne na 0.
- 6. Wartość U stanowi maskę dla wartości UV, wynikiem której będzie ścieżka do korzenia większego poddrzewa.
- 7. Wartość V stanowi maskę dla wartości UV, wynikiem której będzie ścieżka do korzenia mniejszego poddrzewa wykluczonego.
- 8. Przycisk rozpoczynający generowanie określonego poddrzewa i dodanie go do struktury MKB.
- 9. Przycisk umożliwia zapisanie struktury MKB do pliku.
- 10. Przycisk umożliwia zapisanie listy danych potrzebnych dla urządzeń zdolnych procesować wygenerowaną strukturę MKB. Każda linijka w wynikowym pliku zawiera zestaw danych dla osobnego urządzenia.
- 11. Okno konsoli.

# **Enkryptor**



- 1. Wartość pola określa klucz Media Key, którym zaszyfrowane zostaną dane.
- 2. Przycisk umożliwia wybór pliku, który zostanie zaszyfrowany z użyciem podanego wyżej klucza. Wynik operacji zostanie zapisany w tym samym pliku. Uwaga nie jest tworzona kopia zapasowa pliku!
- 3. Okno konsoli.

#### **Dekryptor**



- 1. Przycisk umożliwia wczytanie wcześniej wygenerowanej struktury MKB z pliku.
- 2. Przycisk umożliwia wczytanie pliku zawierającego listę danych potrzebnych dla urządzeń zdolnych procesować wygenerowaną strukturę MKB.
- 3. Pole wyboru urządzenia spośród załadowanych z pliku.
- 4. Pole umożliwiaja przetworzenie struktury MKB w celu odnalezienia Media Key.
- 4. Pole umożliwiaja wybór pliku, którego próba odszyfrowania zostanie podjęta przez wybrane urządzenie z wczytaną strukturą MKB. Wynik operacji zostanie zapisany w tym samym pliku. Uwaga nie jest tworzona kopia zapasowa pliku!
- 5. Okno konsoli.

# **Bibliografia**

- 1. http://www.aacsla.com/
- 2. http://www.aacsla.com/specifications/
- 3. Advanced Access Content System (AACS). Introduction and Common Cryptographic Elements http://www.aacsla.com/specifications/specs091/AACS\_Spec\_Common\_0.91.pdf
- 4. *Understanding AACS (including Subset-Difference)* dostęp przez: https://archive.is/NNeRE, źródło: http://forum.doom9.org/showthread.php?t=122363
- 5. https://pl.wikipedia.org/wiki/Advanced\_Access\_Content\_System
- 6. https://en.wikipedia.org/wiki/Advanced\_Access\_Content\_System
- 7. https://en.wikipedia.org/wiki/Security\_of\_Advanced\_Access\_Content\_System
- 8. https://en.wikipedia.org/wiki/AACS\_encryption\_key\_controversy
- 9. https://en.wikipedia.org/wiki/Media\_Key\_Block
- 10. https://en.wikipedia.org/wiki/Device\_Keys
- 11. http://www.videolan.org/developers/libaacs.html