Biuletyn WAT
Vol. LXVI, Nr 4, 2017

PDF Encryption bazujący na certyfikatach X.509 — teoretyczny poziom bezpieczeństwa

JOANNA DMITRUK[1], MICHAŁ GLET

[1] Szkoła Główna Gospodarstwa Wiejskiego, Wydział Nauk Ekonomicznych,
Katedra Ekonomiki Edukacji, Komunikowania i Doradztwa,
ul. Nowoursynowska 166, 02-787 Warszawa, joanna_dmitruk@sggw.pl
Wojskowa Akademia Techniczna, Wydział Cybernetyki,
Instytut Matematyki i Kryptologii, ul. Gen. W. Urbanowicza 2,
00-908 Warszawa, michal.glet@wat.edu.pl

Streszczenie. PDF Encryption bazujący na certyfikatach X.509 to mechanizm mający na celu zapewnienie bezpieczeństwa zawartości plików PDF oraz umożliwiający przypisanie jej odpowiednich uprawnień. W artykule opisano sposób działania mechanizmu oraz przeanalizowano maksymalny teoretyczny poziom bezpieczeństwa, jaki może być zapewniony zabezpieczonym plikom PDF. Dla obliczonego poziomu bezpieczeństwa wskazane zostały zestawy algorytmów, z jakich należy korzystać, aby zapewnić danym możliwie największy poziom bezpieczeństwa. Na koniec wskazano kilka rekomendacji, którymi w łatwy sposób można podnieść maksymalny teoretyczny poziom bezpieczeństwa plików PDF zabezpieczonych mechanizmem PDF Encryption bazującym na certyfikatach X.509.
Słowa kluczowe: DRM, kryptografia, poziom bezpieczeństwa, PDF Encryption, Adobe, X.509
DOI: 10.5604/01.3001.0010.8221

1. Wstęp

Niniejszy artykuł rozpoczyna serię opracowań związanych z mechanizmami zabezpieczania plików PDF. W pierwszym artykule z serii opisany został mechanizm PDF Encryption w wersji umożliwiającej szyfrowanie zawartości plików PDF z wykorzystaniem certyfikatów X.509. Dodatkowo przeanalizowany został teoretyczny poziom zapewnianego przez ten mechanizm bezpieczeństwa. W kolejnych artykułach z serii opisane zostaną inne wersje mechanizmu oraz ataki na nie.

Mechanizm PDF Encryption jest jednym ze sposobów zabezpieczania zawartości pliku PDF oraz nadawania uprawnień do operacji na nim. Zawartość jest zabezpieczana poprzez jej szyfrowanie algorytmami symetrycznymi (RC4 lub AES). Uprawnienia są nadawane za pomocą określenia odpowiedniej maski bitowej. O ile zaszyfrowanej zawartości nie da się (teoretycznie) odczytać bez posiadania klucza szyfrującego, o tyle ochrona przed nieuprawnionymi operacjami zależy tylko i wyłączenie od oprogramowania, w którym otwierany jest dany dokument — sam mechanizm PDF Encryption nie zapewnia mechanizmów kontroli i ograniczania uprawnień do operacji na dokumencie (np. ograniczenia drukowania).

Mechanizm PDF Encryption umożliwia zabezpieczenie i przypisanie uprawnień do zawartości pliku PDF, więc można go uznać częściowo za mechanizm typu DRM (*Digital Rights Management*). PDF Encryption sam z siebie nie zapewnia wszystkich funkcjonalności systemu DRM, ale w połączeniu z odpowiednim oprogramowaniem umożliwiającym działanie na zawartości plików PDF może taki mechanizm tworzyć.

2. Mechanizm PDF Encryption bazujący na certyfikatach X.509

Mechanizm PDF Encryption bazujący na certyfikatach X.509 umożliwia zabezpieczenie zawartości plików PDF przed nieautoryzowanym dostępem. Mechanizm korzystający z certyfikatów został wprowadzony w wersji 1.4 specyfikacji formatu PDF. W chwili obecnej obowiązującym standardem jest wersja 2.0.

Mechanizm PDF Encryption jest mechanizmem szyfrującym zawartość plików PDF. Dodatkowo umożliwia określenie dopuszczalnych akcji, jakie można wykonać na zabezpieczonym dokumencie. W ramach dostępnych akcji istnieje możliwość określenia uprawnień do:

— drukowania,
— modyfikacji,
— kopiowania zawartości,
— kopiowania zawartości ze względu na lepszą dostępność,
— wydzielania stron,
— komentowania,
— wypełniania pól formularza,
— podpisywania,
— tworzenia stron szablonowych.

Dla każdej z powyższych akcji możliwe jest przypisanie jednej z dwóch wartości:

— dopuszczone,
— niedopuszczalne.

Przypisanie wartości „dopuszczone” umożliwia wykonywanie określonej akcji na zabezpieczonym dokumencie (po uzyskaniu do niego dostępu). Przypisanie wartości „niedopuszczalne” uniemożliwia wykonywanie określonej akcji na zabezpieczonym

dokumencie (po uzyskaniu do niego dostępu). Obsługa uprawnień realizowana jest na poziomie oprogramowania, poza funkcjami mechanizmu PDF Encryption.

Mechanizm PDF Encryption szyfruje wszystkie ciągi znaków i strumienie znajdujące się w dokumencie PDF z wyjątkiem:

— wartości wpisu *ID* z nagłówka,
— ciągów znaków w słowniku *Encrypt*,
— ciągów znaków znajdujących się wewnątrz zaszyfrowanych strumieni takich jak ich zawartość lub skompresowane obiekty,
— ciągów zapisanych w postaci szesnastkowej w ramach wpisu *Content* w słowniku *Signature*.

Dodatkowo szyfrowaniu nie podlegają innego typu obiekty takie jak obiekty typu liczbowego (*integer*) czy też logicznego (*boolean*). Obiekty te nie są szyfrowane z uwagi na fakt, że są wykorzystywane do przechowywania informacji o strukturze dokumentu, a nie o jego treści. Widzimy zatem, że mechanizm PDF Encryption służy głównie do szyfrowania treści dokumentu, a nie jego struktury. W związku z tym, tworząc zabezpieczony dokument PDF, trzeba mieć świadomość tego faktu i należy tworzyć dokument w taki sposób, aby jego struktura nie ujawniała żadnych (albo zbyt wielu) informacji o przechowywanej treści.

W zabezpieczonym (zaszyfrowanym) pliku PDF informacje związane z mechanizmem PDF Encryption przechowywane są w tak zwanym słowniku *Encrypt*. Słownik ten znajduje się w nagłówku pliku PDF. Poniższa tabela prezentuje najważniejsze wpisy, jakie mogą znajdować się w słowniku *Encrypt*.

Tabela 1

Główne wpisy słownika *Encrypt*

Nazwa	Typ	Obowiązkowość
Filter	ciąg znaków	Wpis obowiązkowy
SubFilter	ciąg znaków	Wpis opcjonalny
V	liczba	Wpis obowiązkowy
Length	liczba	Wpis opcjonalny
CF	słownik	Wpis opcjonalny
StmF	ciąg znaków	Wpis opcjonalny
StrF	ciąg znaków	Wpis opcjonalny
EFF	ciąg znaków	Wpis opcjonalny

Znaczenia powyższych wpisów opisane zostały w tabeli 2.

Tabela 2

Znaczenia wpisów słownika *Encrypt*

Nazwa wpisu	Znaczenie
Filter	Określa nazwę mechanizmu (*handlera*) bezpieczeństwa wykorzystanego do zabezpieczenia pliku PDF. Przykładowym mechanizmem jest mechanizm szyfrujący bazujący na certyfikatach X.509.
SubFilter	Nazwa dodatkowego mechanizmu bezpieczeństwa, w jednoznaczny sposób wskazująca na format i sposób szyfrowania, który umożliwia odszyfrowanie zawartości dokumentu bez wykorzystania mechanizmu opisanego we wpisie *Filter*.
V	Numer określający algorytm wykorzystany do zaszyfrowania/odszyfrowania zawartości dokumentu PDF. W ramach specyfikacji PDF 2.0 określone zostały następujące numery algorytmów: 0 — algorytm z poza specyfikacji PDF, 1 — algorytm zgodny ze specyfikacją (RC4, AES) z kluczem o długości 40 bitów, 2 — algorytm zgodny ze specyfikacją (RC4, AES) z kluczem o długości większej niż 40 bitów, 3 — algorytm spoza specyfikacji PDF (np. niejawny) o długości klucza od 40 do 128 bitów, 4 — algorytm korzystający z wartości wpisów *CF*, *StmF* oraz *StrF*, zgodny ze specyfikacją (RC4, AES) z kluczem o długości 128 bitów, 5 — algorytm korzystający z wartości wpisów *CF*, *StmF* oraz *StrF*, zgodny ze specyfikacją (AES) z kluczem o długości 256 bitów.
Length	Określa bitową długość klucza szyfrującego.
CF	Słownik słowników — przechowuje informacje o wpisach będących nazwami filtrów kryptograficznych oraz o wartościach będących słownikami zawierającymi dane specyficzne dla filtrów (np. ustawienia).
StmF	Nazwa filtra kryptograficznego, z którego należy skorzystać celem odszyfrowania strumieni zawartych w pliku PDF. Podana nazwa powinna stanowić wpis w słowniku CF.
StrF	Nazwa filtra kryptograficznego, z którego należy skorzystać celem odszyfrowania ciągów znaków zawartych w pliku PDF. Podana nazwa powinna stanowić wpis w słowniku CF.
EFF	Nazwa filtru kryptograficznego, z którego należy skorzystać celem zaszyfrowania dodanej do pliku PDF zawartości danych zewnętrznych — załączników.

Do zabezpieczania (szyfrowania) zawartości pliku PDF specyfikacja przewiduje możliwość zastosowania jednego z dwóch algorytmów symetrycznych:

— RC4 — szyfr strumieniowy, niezmieniający długości szyfrowanych danych (nie wymaga dopełnienia tekstu jawnego). Specyfikacja w wersji 2.0 oznacza algorytm RC4 jako „deprecated". W związku z tym, w nowych implementacjach i do zabezpieczania nowych dokumentów nie zaleca się jego stosowania. Wynika to ze słabości i ataków kryptoanalitycznych, jakie pojawiają się na algorytm RC4 od wielu lat.

— AES (*Advanced Encryption Standard*) — szyfr blokowy, zmieniający długość zaszyfrowanych danych (tekst jawny wymaga dopełnienia do wielokrotności długości przetwarzanego bloku — do wielokrotności 128 bitów). Sposób dopełnienia danych tekstu jawnego (ciągi znaków, strumienie) musi być zgodny z algorytmem dopełniania danych opisanym w dokumencie RFC-2898. Algorytm AES opisany jest w standardzie FIPS-197.

Oprócz RC4 oraz AES specyfikacja PDF wykorzystuje na potrzeby mechanizmu PDF Encryption również inne prymitywy kryptograficzne. Jednym z nich jest funkcja skrótu MD5. Jest ona wykorzystywana, gdy wpis *V* ma wartość 1, 2, 3 lub 4. MD5 jest wtedy wykorzystywane do wygenerowania klucza dla algorytmu szyfrującego, bazując na haśle wprowadzonym przez użytkownika. Z uwagi na fakt, że funkcja MD5 posiada wiele poważnych podatności kryptograficznych, nie zaleca się jej wykorzystywania.

W związku z powyższym widać, że dla nowych dokumentów i implementacji powinno się korzystać z mechanizmu PDF Encryption tylko i wyłącznie z ustawionym wpisem *V* na wartość 5 — korzystanie z algorytmu AES-256 i niekorzystanie z funkcji MD5.

Tabela 3 opisuje kroki algorytmu szyfrującego mechanizmu PDF Encryption w zależności od wartości wpisu *V*.

W ramach specyfikacji PDF 2.0 określone zostały dwa mechanizmy umożliwiające wykorzystanie mechanizmu PDF Encryption (*security handlers*) — bazujący na haśle oraz bazujący na certyfikatach X.509. W dalszej części rozdziału opisany zostanie mechanizm oparty o certyfikaty X.509.

Mechanizm PDF Encryption bazujący na certyfikatach X.509 do zabezpieczania dokumentów PDF wykorzystuje kryptografię klucza publicznego. Mechanizm ten umożliwia zdefiniowanie listy albo list odbiorców, którzy będą w stanie odszyfrować zawartość dokumentu PDF. Dodatkowo, mechanizm umożliwia przypisanie różnych ograniczeń do różnych list odbiorców. Co więcej, w przeciwieństwie do mechanizmu szyfrowania bazującego na hasłach, dokument może odszyfrować tylko odbiorca znajdujący się na liście odbiorców (osoba posiadająca odpowiedni klucz prywatny).

Mechanizm szyfrowania bazujący na certyfikatach X.509 korzysta ze standardu kryptograficznego tworzenia i zapisywania wiadomości CMS (*Cryptographic Message Syntax*, RFC5652). CMS wykorzystywany jest m.in. do zapisywania/kodowania list odbiorców, kluczy szyfrujących czy też informacji o ograniczeniach.

W momencie zabezpieczania dokumentu, korzystając z mechanizmu szyfrowania bazującego na certyfikatach X.509, osoba tworząca dokument musi posiadać dostęp do certyfikatów X.509 wszystkich odbiorców dokumentu. Dodatkowo należy upewnić się, wykonując odpowiednie walidacje, że posiadane certyfikaty rzeczywiście należą do odbiorców oraz że nie utraciły swojej ważności. Zaleca się korzystanie z zaufanej infrastruktury PKI zbudowanej na bazie co najmniej jednego zaufanego wystawcy certyfikatów i jednego zaufanego dostawcy usług PKI (np. listy CRL, serwer znakowania czasem, OCSP).

W zabezpieczonym (zaszyfrowanym) pliku PDF dodatkowe informacje związane z mechanizmem szyfrowania bazującym na certyfikatach X.509 przechowywane są w słowniku *Encrypt*. Tabela 4 prezentuje dodatkowe, specyficzne dla mechanizmu, wpisy w słowniku *Encrypt*.

Tabela 3

Kroki algorytmu szyfrującego w zależności od wartości wpisu *V*

Wartość *V*	Kroki algorytmu szyfrującego mechanizmu PDF Encryption
1 2 3 4	1. Pobierany jest numer obiektu i numer generacji z identyfikatora obiektu ciągu znaków lub strumienia, który ma zostać zaszyfrowany. 2. Odczytane w kroku numer 1 numery zapisywane są w formacie binarnym i doklejane są do *n*-bajtowego klucza szyfrującego. Uzyskane w ten sposób dane mają długość *n* + 5 bajtów, gdyż z numeru obiektu brane są trzy mniej znaczące bajty, a z numeru generacji dwa. W przypadku, w którym wykorzystywany jest algorytm AES, klucz należy dodatkowo rozszerzyć o 4 bajty o wartości ASCII „sAIT" (0×73, 0×41, 0×6C, 0×54). Gdy *V* = 1, wtedy *n* = 5. W pozostałych przypadkach *n* = wartość wpisu *Length*/8. 3. Do funkcji MD5 przekazywany jest wynik kroku numer 2. 4. Jako klucz szyfrujący wykorzystywana jest odpowiednia liczba bitów z danych wyjściowych funkcji skrótu MD5 (z kroku 3). Pobierana liczba bitów określona jest w wartości wpisu *Length*. 5. Szyfrowanie: 5.1. W przypadku algorytmu RC4 generowany jest odpowiedniej (w zależności od długości danych do zaszyfrowania) długości ciąg bajtów, który następnie poddawany jest operacji XOR z danymi tekstu jawnego (danymi do zaszyfrowania). Uzyskany w ten sposób ciąg bajtów jest szyfrogramem. 5.2. W przypadku algorytmu AES wykorzystywany jest tryb pracy CBC (*Cipher Block Chaining*) i losowo wygenerowany wektor inicjalizujący (16 bajtów). Wygenerowany wektor inicjalizujący doklejany jest (na początku) do uzyskanego szyfrogramu.
5	1. Do wygenerowania klucza szyfrującego wykorzystywany jest 32-bajtowy klucz szyfrowania pliku. Generowany jest klucz o długości 256 bitów. 2. Do szyfrowania używa się algorytmu AES. Wykorzystywany jest tryb pracy CBC (*Cipher Block Chaining*) i losowo wygenerowany wektor inicjalizujący (16 bajtów). Wygenerowany wektor inicjalizujący doklejany jest (na początku) do uzyskanego szyfrogramu.

Tabela 4

Dodatkowe wpisy słownika *Encrypt*

Nazwa	Typ	Obowiązkowość
Recipients	tablica	Wpis obowiązkowy
P	liczba	Wpis obowiązkowy

Znaczenia konkretnych wpisów opisane zostały w tabeli 5.

Tabela 5

Znaczenia dodatkowych wpisów słownika *Encrypt*

Nazwa wpisu	Znaczenie
Recipients	Tablica zawierająca ciągi bajtów. Każdy ciąg bajtów jest obiektem typu CMS zawierającym odbiorców dokumentu. Każdy ciąg bajtów zawiera odbiorców z pojedynczej listy odbiorców. Dodatkowo obiekt CMS zawiera klucze umożliwiające odszyfrowanie zawartości dokumentu przez odbiorców danej listy, jak również zakres ograniczeń do dokumentu. Ograniczenia przypisywane są do listy (każda lista może posiadać inny zestaw ograniczeń).
P	Zestaw flag (zbiór 32-bitowych masek) określających, jakie ograniczenia obowiązują na dokumencie w momencie otwarcia dokumentu z uprawnieniami użytkownika (tożsame z hasłem użytkownika z mechanizmu opartego o hasła). Ustawienie bitu numer 2 na wartość 1 oznacza, że wszystkie ograniczenia powinny zostać zignorowane. Zgodnie ze standardem PDF 2.0 zdefiniowane zostały następujące ograniczenia (notacja Little-Endian dla pojedynczej maski bitowej): • bit numer 2 — jeżeli ustawiony na 1, wszystkie ograniczenia są ignorowane, • bit numer 3 — ograniczenie drukowania, • bit numer 4 — ograniczenie modyfikacji zawartości, • bit numer 5 — ograniczenie kopiowania zawartości, • bit numer 6 — ograniczenie możliwości dodawania komentarzy i adnotacji, • bit numer 9 — ograniczenie możliwości wypełniania pól formularzy, • bit numer 10 — ograniczenie kopiowania zawartości ze względu na lepszą dostępność, • bit numer 11 — ograniczenie możliwości wydzielania stron, zmiany struktury dokumentu, • bit numer 12 — ograniczenie tworzenia stron szablonowych, drukowania w wysokiej jakości.

Przechowywane w obiekcie CMS dane zawierają informacje związane z kluczem deszyfrującym (klucz symetryczny dla algorytmu RC4 albo AES) umożliwiającym odszyfrowanie zawartości dokumentu. Informacje te są osobno szyfrowane dla każdego odbiorcy z listy odbiorców z wykorzystaniem jego klucza publicznego pobranego z certyfikatu X.509. Odbiorca po otrzymaniu zaszyfrowanego dokumentu znajduje w obiekcie CMS wpis przeznaczony dla niego. Z tego wpisu pobiera zaszyfrowane informacje i odszyfrowuje je, korzystając z posiadanego przez siebie klucza prywatnego (związanego z certyfikatem X.509, którym posługiwał się twórca dokumentu). Po odszyfrowaniu odbiorca uzyskuje klucz symetryczny, za pomocą którego może odszyfrować zawartość typu „enveloped data" wiadomości CMS. Odszyfrowana zawartość składa się z:

— 20-bajtowej wartości początkowej, za pomocą której możliwe jest wygenerowanie klucza szyfrującego, którym zaszyfrowana została zawartość (ciągi znaków i strumienie) zabezpieczonego dokumentu PDF (zgodnie z algorytmem opisanym w „Kroki algorytmu szyfrującego w zależności od wartości wpisu *V*");

— 4-bajtowej maski bitowej opisującej ograniczenia przypisane do dokumentu dla danego odbiorcy.

Tabela 6 przedstawia główne kroki algorytmu generowania klucza szyfrującego plik na podstawie 20-bajtowej wartości początkowej.

Tabela 6

Algorytm generowania klucza szyfrującego plik na podstawie wartości początkowej

Nazwa algorytmu	Główne kroki algorytmu
Algorytm generowania klucza szyfrującego plik na podstawie wartości początkowej — dla klucza o długości nie większej niż 128 bitów	1. Do funkcji skrótu SHA-1 przekazywana jest odszyfrowana 20-bajtowa wartość początkowa. 2. Do funkcji skrótu SHA-1 przekazywane są bajty każdego obiektu CMS z wartości wpisu *Recipients* w kolejności, w jakiej występują w dokumencie. 3. Do funkcji skrótu SHA-1 przekazywane są 4 bajty — każdy o wartości 0×FF. 4. Pierwsze n bajtów z wyjścia funkcji SHA-1 stanowi klucz szyfrujący plik, gdzie n jest wartością wpisu *Length* podzieloną przez 8.
Algorytm generowania klucza szyfrującego plik na podstawie wartości początkowej — dla klucza o długości równej 256 bitów	1. Do funkcji skrótu SHA-256 przekazywana jest odszyfrowana 20-bajtowa wartość początkowa. 2. Do funkcji skrótu SHA-256 przekazywane są bajty każdego obiektu CMS z wartości wpisu *Recipients* w kolejności, w jakiej występują w dokumencie. 3. Do funkcji skrótu SHA-256 przekazywane są 4 bajty — każdy o wartości 0×FF. 4. Pierwsze *n* bajtów z wyjścia funkcji SHA-256 stanowi klucz szyfrujący plik, gdzie *n* jest wartością wpisu *Length* podzieloną przez 8.

Mechanizm szyfrowania na podstawie certyfikatów X.509 umożliwia wykorzystanie jednego z poniższych algorytmów do zaszyfrowania zawartości wiadomości CMS (pola *enveloped data*):

— RC2 (klucz do 128 bitów),
— RC4 (klucz do 256 bitów),
— DES (klucz 64 bity),
— Triple DES (klucz do 128 bitów),
— AES w trybie CBC (klucz 128, 192 i 256 bitów).

Z uwagi na powszechnie znane słabości pierwszych pięciu algorytmów, dla nowych dokumentów i implementacji zaleca się korzystanie jedynie z algorytmu AES.

Mechanizm szyfrowania bazujący na certyfikatach X.509 wprowadzony został w specyfikacji PDF w wersji 1.4. Tabela 7 prezentuje szczegóły techniczne mechanizmów szyfrowania określonych w poszczególnych wersjach specyfikacji formatu PDF.

Tabela 7

Mechanizmy szyfrowania w poszczególnych wersjach standardu PDF

Wersja PDF	Algorytm klucza publicznego	Algorytm szyfrowania wiadomości CMS	Algorytm szyfrowania dokumentu PDF
PDF 1.0	brak	nie dotyczy	nie dotyczy
PDF 1.1	brak	nie dotyczy	nie dotyczy
PDF 1.2	brak	nie dotyczy	nie dotyczy
PDF 1.3	brak	nie dotyczy	nie dotyczy
PDF 1.4	RSA do 2048 bitów	RC2 (klucz do 128 bitów) RC4 (klucz do 256 bitów) DES (klucz 64 bity) Triple DES (klucz do 128 bitów) AES w trybie CBC (klucz 128, 192 i 256 bitów)	RC4 (klucz 128 bitów)
PDF 1.5	RSA do 2048 bitów		RC4 (klucz 128 bitów)
PDF 1.6	RSA do 8192 bitów		AES-128
PDF 1.7	RSA do 8192 bitów		AES-128
PDF 1.7 Level 3	RSA do 8192 bitów; ECC na krzywych nazwanych P-256, P-384, P-521		AES-256
PDF 1.7 Level 8	RSA do 8192 bitów; ECC na krzywych nazwanych P-256, P-384, P-521		AES-256
PDF 2.0	RSA do 8192 bitów; ECC na krzywych nazwanych P-256, P-384, P-521		AES-256

3. Analiza bezpieczeństwa

W rozdziale tym dokonana zostanie analiza teoretycznego poziomu bezpieczeństwa zapewnianego przez mechanizm PDF Encryption korzystający z certyfikatów X.509. Dokonana analiza abstrahuje od skutecznych (o złożoności mniejszej niż pełne przeszukanie przestrzeni klucza) ataków na wykorzystane prymitywy kryptograficzne. Z uwagi na fakt, że w omawianym mechanizmie występują prymitywy zarówno z kryptografii klucza publicznego, jak również kryptografii symetrycznej, do oceny teoretycznego poziomu bezpieczeństwa wykorzystane zostały rekomendacje zawarte w dokumencie NIST SP 800-57.

Mechanizm PDF Encryption bazujący na certyfikatach X.509 pozwala na wykorzystanie algorytmu RSA (w wersji z kluczem do 8192 bitów), algorytmów opartych o krzywe eliptyczne P-256, P-384 lub P521, szyfru blokowego AES (z kluczem 128 lub 256 bitów) oraz szyfru strumieniowego RC4 (z kluczem do 128 bitów).

Przeprowadzona dalej analiza dotyczy maksymalnego teoretycznego poziomu bezpieczeństwa, który mechanizm PDF Encryption może zapewnić.

Tabela 8

Teoretyczne poziomy bezpieczeństwa zgodne z NIST SP 800-57

Poziom bezpieczeństwa	Algorytm symetryczny	Minimalna liczba bitów klucza w algorytmie RSA	Liczba bitów krzywej eliptycznej
128	AES-128	3072	256-383
192	AES-192	7680	384-511
256	AES-256	15360	od 512

Z zawartości tabeli 7 oraz tabeli 8 wynika, że maksymalny teoretyczny poziom bezpieczeństwa zapewniany przez analizowany mechanizm może wynosić 256 bitów — w przypadku wykorzystania:

— algorytmu opartego o krzywą eliptyczną P-521,
— algorytmu AES-256 na etapie szyfrowania danych w wiadomości CMS,
— algorytmu AES-256 na etapie szyfrowania zawartości pliku PDF.

Teoretyczny poziom bezpieczeństwa zależy od poziomu bezpieczeństwa najsłabszego wykorzystywanego prymitywu kryptograficznego. Kilka przykładowych poziomów bezpieczeństwa w zależności od wykorzystywanych algorytmów prezentuje tabela 9.

Tabela 9

Teoretyczne poziomy w zależności od wykorzystanych algorytmów

Teoretyczny poziom bezpieczeństwa	Algorytm klucza publicznego	Algorytm symetryczny (na potrzeby CMS)	Algorytm symetryczny (szyfrowanie PDF)
256	Na krzywej P-521	AES-256	AES-256
128	Na krzywej P-521	AES-128, AES-256	AES-128
192	RSA z kluczem 8192-bitowym	AES-256	AES-256
128	RSA z kluczem 3072-bitowym	AES-128, AES-256	AES-128, AES-256
80	RSA z kluczem 1024-bitowym	AES-128, AES-256	AES-128, AES-256
112	RSA z kluczem 2048-bitowym	AES-128, AES-256	AES-128, AES-256
192	Na krzywej P-384	AES-256	AES-256
128	Na krzywej P-256	AES-128, AES-256	AES-128, AES-256

Analizując zawartość tabeli 9, widać, że nie wystarczy stosować algorytm AES-256, żeby zapewnić odpowiednio wysoki poziom bezpieczeństwa. W szczególnym przypadku (wykorzystania algorytmu RSA z kluczem 1024-bitowym) poziom bezpieczeństwa zamiast 256 bitów może spaść w okolice 80 bitów. Należy być świadomym tego faktu podczas wyboru algorytmów i ich parametrów w mechanizmie PDF Encryption.

Powyżej przeprowadzona analiza abstrahuje od przebiegu algorytmu, w którym wykorzystywane są analizowane prymitywy kryptograficzne. Główne kroki (kryptograficzne) mechanizmu PDF Encryption bazującego na certyfikatach X.509 wyglądają następująco:

1. Tworzony jest klucz do szyfrowania pliku PDF — zgodnie z algorytmem „Algorytm generowania klucza szyfrującego plik na podstawie wartości początkowej — dla klucza o długości równej 256 bitów".
2. Dane niezbędne do wygenerowania klucza z kroku 1 są opakowywane w zaszyfrowaną wiadomość CMS.
3. Klucz niezbędny do odszyfrowania danych z kroku 2 szyfrowany jest kluczem publicznym odbiorcy pliku PDF.

Kroki numer 2 i 3 są krokami standardowymi w przypadku tworzenia zaszyfrowanych wiadomości w formacie CMS — zgodne ze standardem CMS. Ich teoretyczny poziom bezpieczeństwa zależy od wykorzystanych prymitywów kryptograficznych. W związku z tym dalszej analizie poddany zostanie krok 1, czyli opisany w tabeli 6 „Algorytm generowania klucza szyfrującego plik na podstawie wartości początkowej — dla klucza o długości równej 256 bitów". Algorytm ten generuje 256-bitowy klucz, który następnie może zostać wykorzystany do zaszyfrowania zawartości pliku PDF, korzystając z algorytmu AES-256. Algorytm ten do wygenerowania klucza wykorzystuje funkcję skrótu SHA-256. Teoretyczny poziom bezpieczeństwa funkcji skrótu, zgodnie z rekomendacją NIST SP 800-57, zależy od jej wykorzystania. W schematach podpisów cyfrowych i aplikacjach typu „hash-only" wynosi on 128 bitów. W funkcjach HMAC, funkcjach generowania klucza (KDF) oraz w generatorach liczb pseudolosowych poziom bezpieczeństwa wynosi maksymalnie 256 bitów i zależy od poziomu entropii danych wejściowych. Sposób wykorzystania funkcji SHA-256 w analizowanym algorytmie jest inny (dużo prostszy) niż w klasycznych funkcjach generowania klucza (zgodnych z KDF). Niemniej jednak, na obecnym etapie analizy przyjmujemy, że teoretyczny poziom bezpieczeństwa funkcji SHA-256 wynosi maksymalnie 256 bitów i zależy od poziomu entropii danych wejściowych. Danymi wejściowymi w omawianym algorytmie generowania klucza są:

1. 20-bajtowy ciąg danych pseudolosowych,
2. obiekty CMS z odbiorcami pliku PDF,
3. cztery bajty, każdy o wartości $0 \times FF$.

Z powyższych danych jedynie dane z punktu 1 stanowią tajemnicę i podlegają szyfrowaniu w wiadomości CMS. Pozostałe dane są jawne dla odbiorcy i dla atakującego. W związku z tym poziom entropii przekazywanej do funkcji generującej klucz zależy

od poziomu entropii danych z punktu 1. Poziom ten, z uwagi na ograniczenie liczby bajtów do 20, wynosi maksymalnie 160 bitów. Zatem maksymalny teoretyczny poziom bezpieczeństwa, przy założeniu poziomu bezpieczeństwa funkcji SHA-256 na 256 bitów, dla algorytmu „Algorytm generowania klucza szyfrującego plik na podstawie wartości początkowej — dla klucza o długości równej 256 bitów" wynosi 160 bitów. Algorytm ten generuje klucze dla funkcji AES-256. W związku z tym maksymalny teoretyczny poziom bezpieczeństwa funkcji AES-256 w opisywanym przez nas mechanizmie wynosi 160 bitów. Z tego też powodu maksymalny teoretyczny poziom bezpieczeństwa całego mechanizmu PDF Encryption bazującego na certyfikatach X.509 wynosi 160 bitów. Poniższa tabela prezentuje przykładowe algorytmy, które można wykorzystać w mechanizmie PDF Encryption, aby zachować poziom bezpieczeństwa wynoszący 160 bitów.

Tabela 10

Algorytmy zapewniające najwyższy teoretyczny poziom bezpieczeństwa w mechanizmie PDF Encryption (przy założeniu 256 bitów bezpieczeństwa SHA-256)

Teoretyczny poziom bezpieczeństwa	Algorytm klucza publicznego	Algorytm symetryczny (na potrzeby CMS)	Algorytm symetryczny (szyfrowanie PDF)
160	Na krzywej P-521	AES-256	AES-256
160	Na krzywej P-384	AES-256	AES-256
160	RSA z kluczem 8192-bitowym	AES-256	AES-256

Dodatkowo należy zwrócić uwagę na założenie, jakie zostało przyjęte na maksymalny teoretyczny poziom bezpieczeństwa funkcji skrótu SHA-256. Jeżeli uznać, że poziom ten wynosi 128 bitów, z uwagi na bardzo prosty algorytm generowania klucza (pojedyncze wywołanie funkcji SHA-256), teoretyczny maksymalny poziom bezpieczeństwa całego mechanizmu spada do 128 bitów. Poniższa tabela prezentuje przykładowe algorytmy, które można wykorzystać w mechanizmie PDF Encryption, aby zachować poziom bezpieczeństwa wynoszący 128 bitów.

Tabela 11

Algorytmy zapewniające najwyższy teoretyczny poziom bezpieczeństwa w mechanizmie PDF Encryption (przy założeniu 128 bitów bezpieczeństwa SHA-256)

Teoretyczny poziom bezpieczeństwa	Algorytm klucza publicznego	Algorytm symetryczny (na potrzeby CMS)	Algorytm symetryczny (szyfrowanie PDF)
128	Na krzywej P-521	AES-128, AES-256	AES-128, AES-256
128	RSA z kluczem 8192-bitowym	AES-128, AES-256	AES-128, AES-256
128	RSA z kluczem 3072-bitowym	AES-128, AES-256	AES-128, AES-256
128	Na krzywej P-384	AES-128, AES-256	AES-128, AES-256
128	Na krzywej P-256	AES-128, AES-256	AES-128, AES-256

4. Podsumowanie

Poprawne określenie teoretycznego poziomu bezpieczeństwa zapewnianego przez dany mechanizm/protokół jest niezwykle istotne, gdyż pozwala określić złożoność podstawowego ataku kryptograficznego, który polega przeważnie na przeszukaniu pełnej przestrzeni dostępnych kluczy celem złamania mechanizmu. Poziom ten stanowi również poziom odniesienia pozwalający ocenić skuteczności ataków na mechanizm — jeżeli złożoność ataku jest mniejsza niż złożoność ataku podstawowego, atak uznaje się za skuteczny.

Maksymalny teoretyczny poziom bezpieczeństwa mechanizmu PDF Encryption w oparciu o certyfikaty X.509 wynosi 128 lub 160 bitów (w zależności od sposobu interpretacji poziomu bezpieczeństwa funkcji SHA-256). Zgodnie z naszą interpretacją wynosi on 160 bitów, gdyż pomimo bardzo prostego schematu wykorzystania SHA-256, analizując bezpieczeństwo, patrzymy na odporność na znalezienie przeciwobrazu, a nie na odporność na kolizje (która w przypadku SHA-256 wynosi 128 bitów — zgodnie z paradoksem dnia urodzin).

Poziom bezpieczeństwa 160 bitów to dużo mniej niż mogą zapewnić algorytmy określone w specyfikacji PDF 2.0 — 256 bitów. Aby zapewnić poziom bezpieczeństwa zgodny z poziomem zapewnianym przez prymitywy kryptograficzne, wystarczyłoby zwiększyć liczbę bajtów sekretu używanego do generowania klucza szyfrującego dla pliku PDF — np. z 20 do 32 bajtów. Analizując opisany mechanizm, kolejną oczywistą rekomendacją wydaje się być zastąpienie algorytmu „Algorytm generowania klucza szyfrującego plik na podstawie wartości początkowej — dla klucza o długości równej 256 bitów" funkcją generowania klucza zgodną np. ze standardem RFC5869 lub NIST 800-56.

W niniejszej pracy staraliśmy się przeanalizować maksymalny teoretyczny poziom bezpieczeństwa zapewniany przez mechanizm PDF Encryption bazujący na certyfikatach X.509. Dodatkowo wskazaliśmy zestawy algorytmów, które zapewniają mechanizmowi poziom bezpieczeństwa wynoszący 160 bitów. Wybór innych algorytmów niż opisywane przez nas w tabeli 10 obniża poziom bezpieczeństwa zaszyfrowanego pliku PDF. Należy o tym pamiętać, tworząc kolejne zaszyfrowane pliki PDF.

Źródło finansowania pracy — środki własne autorów.

Artykuł wpłynął do redakcji 6.11.2017 r. Zweryfikowaną wersję po recenzjach otrzymano 22.12.2017 r.

LITERATURA

[1] NIST: SP 800-57 Part 1 Rev. 4 Recommendation for Key Management, Part 1: General (dostęp: 26.10.2017), https://csrc.nist.gov/publications/detail/sp/800-57-part-1/rev-4/final

[2] NIST: SP 800-56C Recommendation for Key Derivation through Extraction-then-Expansion (dostęp: 2.11.2017), https://csrc.nist.gov/publications/detail/sp/800-56c/final

[3] NIST: FIPS PUB 197 Announcing the ADVANCED ENCRYPTION STANDARD (AES) (dostęp: 29.10.2017), https://csrc.nist.gov/publications/detail/sp/800-56c/final

[4] The Internet Engineering Task Force: RFC-5652 Cryptographic Message Syntax (CMS) (dostęp: 29.11.2017), https://tools.ietf.org/html/rfc5652

[5] The Internet Engineering Task Force: RFC-2898 PKCS #5: Password-Based Cryptography Specification Version 2.0 (dostęp: 2.11.2017), https://tools.ietf.org/html/rfc2898

[6] The Internet Engineering Task Force: RFC-5869 HMAC-based Extract-and-Expand Key Derivation Function (HKDF) (dostęp: 02.11.2017), https://tools.ietf.org/html/rfc5869

[7] ISO/DIS 32000-2.3:2016(E): Document management — Portable document format, Part 2: PDF 2.0 (draft) (dostęp: 20.10.2017), https://www.pdfa.org/wp-content/until2016_uploads/2016/04/2016-02-15_ISO-32000-2_DIS_3.pdf

J. DMITRUK, M. GLET

Analysis of theoretical security level of PDF Encryption mechanism based on X.509 certificates

Abstract. PDF Encryption is a content security mechanism developed and used by Adobe in their products. In this paper, we have checked a theoretical security level of a variant that uses public key infrastructure and X.509 certificates. We have described a basis of this mechanism and we have performed a simple security analysis. Then, we have showed possible tweaks and security improvements. At the end, we have given some recommendations that can improve security of a content secured with PDF Encryption based on X.509 certificates.

Keywords: DRM, cryptography, security level, PDF Encryption, Adobe, X.509
DOI: 10.5604/01.3001.0010.8221