Opis techniczny protokołu komunikacyjnego kasa – Centralne Repozytorium Kas –Standardy kryptograficzne

Właściciel dokumentu	*Ministerstwo Finansów*
Wersja dokumentu	*2.0.0*
Status dokumentu	*dokumentacja*
Data utworzenia	*2017-12-05*
Data modyfikacji	*2018-06-25*
Nazwa pliku	*Opis techniczny protokołu komunikacyjnego kasa – Centralne Repozytorium Kas –Standardy kryptograficzne _v2.0.0.docx*

Historia zmian

Data	Autor	Podsumowanie zmian	Wersja
2017-12-05	DI	Utworzenie dokumentu.	1.7.8
2018-01-31	DI	Określenie formatu parametru „kid” obiektu JWE.	1.7.9
2018-04-05	DI	Uzupełnienie przykładów weryfikacji komend i danych.	1.8.0
2018-05-15	DI	Ujednolicenie zapisów ze specyfikacją komend.	1.8.1
2018-06-25	DI	Publikacja BIP MF	2.0.0

Spis treści

Standardy kryptograficzne dla Centralnego Repozytorium danych o sprzedaży detalicznej..... **Błąd! Nie zdefiniowano zakładki.**

Historia zmian 2

Spis treści 3

1 Słownik pojęć używanych w dokumencie 4

1.1 Wykaz specyfikacji technicznych użytych w dokumencie 4

2 Zabezpieczanie kanału komunikacyjnego 5

3 Certyfikaty urządzenia fiskalnego.....6

4 Algorytmy kryptograficzne 7

4.1 Podpisywanie..... 7

4.2 Szyfrowanie symetryczne 7

4.3 Szyfrowanie klucza szyfrującego..... 7

5 Szyfrowanie komend oraz danych.....8

5.1 Podpisywanie i szyfrowanie komend..... 9

5.2 Podpisywanie i szyfrowanie danych 10

Załącznik A.....11

A.1 Funkcje użyte w opisach 11

A.2 Podpisywanie komend 12

A.3 Szyfrowanie komend 14

A.4 Podpisywanie danych 16

A.5 Szyfrowanie danych..... 18

A.6 Wysyłanie danych..... 20

A.7 Odebranie komendy..... 21

A.8 Odszyfrowanie komendy..... 22

A.9 Weryfikacja podpisu komendy 24

Załącznik B.....25

B.1 Przykładowe certyfikaty środowiska testowego 25

B.2 Przykładowe dane procesu podpisywania komendy w środowisku testowym..... 25

B.3 Przykładowe dane procesu szyfrowania komendy w środowisku testowym..... 28

B.4 Przykładowe dane procesu podpisywania danych w środowisku testowym 35

B.5 Przykładowe dane procesu szyfrowania danych w środowisku testowym 37

1 Słownik pojęć używanych w dokumencie

- TLS 1.2 – Transport Layer Security bezpieczny protokół przesyłania danych warstwy aplikacyjnej w wersji 1.2 opisany w dokumencie RFC 5246.
- JSON – JavaScript Object Notation tekstowy format wymiany danych bazujący na podzbiorze języka JavaScript opisany w dokumencie RFC 7159.
- JWS – JSON Web Signature standard tworzenia podpisów cyfrowych dla dokumentów JSON opisany w dokumencie RFC 7515.
- JWE – JSON Web Encryption standard szyfrowania dokumentów bazujących na strukturze JSON opisany w dokumencie RFC 7516.
- JWK – JSON Web Key standard tworzenia struktury klucza kryptograficznego w formacie JSON opisany w dokumencie RFC 7517.
- JWA – JSON Web Algorithms wykaz algorytmów kryptograficznych używanych w JWE i JWS opisany w dokumencie RFC 7518.
- Base64 – kodowanie danych binarnych przy użyciu podzbioru US-ASCII, opisane w sekcji czwartej dokumentu RFC 4648. Zastosowanie takiego formatu pozwala dane binarne umieścić w strukturach danych tekstowych.
- Base64URL – kodowanie danych binarnych z użyciem znaków dozwolonych w adresacji domenowej URL oraz nazewnictwie plików zdefiniowane w sekcji piątej dokumentu RFC 4648. Dodatkowo usuwa się znak dopełnienia '=' z końca zakodowanych danych oraz wszystkie znaki końca linii, spacje i inne dodatkowe białe znaki. Szczegółowa implementacja jest w załączniku C dokumentu RFC 7515.

1.1 Wykaz specyfikacji technicznych użytych w dokumencie

Kod	Zagadnienie
RFC 1951	DEFLATE Compressed Data Format Specification version 1.3
RFC 4648	The Base16, Base32, and Base64 Data Encodings
RFC 5246	The Transport Layer Security (TLS) Protocol Version 1.2
RFC 7159	The JavaScript Object Notation (JSON) Data Interchange Format
RFC 7515	JSON Web Signature (JWS)
RFC 7516	JSON Web Encryption (JWE)
RFC 7517	JSON Web Key (JWK)
RFC 7518	JSON Web Algorithms (JWA)
RFC 2104	HMAC: Keyed-Hashing for Message Authentication
RFC 3447	Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1
RFC 4492	Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)
RFC 5289	TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode (GCM)
RFC 5280	Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile
RFC 3279	Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile
RFC 4055	Additional Algorithms and Identifiers for RSA Cryptography for use in the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile
FIPS 180-4	Secure Hash Standard (SHS)
JOSE	JSON Object Signing and Encryption (JOSE)

2 Zabezpieczanie kanału komunikacyjnego

W komunikacji urządzania fiskalnego z repozytorium do zabezpieczania połączenia sieciowego stosowany jest standard TLSv1.2. Zalecanym algorytmem szyfrowania kanału komunikacyjnego jest algorytm ECDHE_RSA_WITH_AES_128_CBC_SHA256 (kod heksadecymalny {0xC0,0x27}, dziesiętnie 49191) wskazany w dokumencie RFC 5289. Do komunikacji urządzenia fiskalnego z serwerami opatrzonymi nazwą domenową z sufiksem „.mf.gov.pl" należy użyć uwierzytelniania dwustronnego z wykorzystaniem certyfikatu kasy wystawionego przez zaufanego producenta oraz certyfikatami serwerów wystawionymi przez certyfikat główny ministerstwa. Magazyn certyfikatów kluczy publicznych zaufanych producentów składowany jest w zasobach ministerstwa oddzielnie dla środowiska testowego oraz produkcyjnego. Repozytorium umożliwia zarejestrowanie kilku ważnych certyfikatów danego producenta. W przypadku kompromitacji klucza prywatnego producenta kas certyfikat klucza publicznego skojarzony ze skompromitowanym kluczem prywatnym zostanie usunięty z repozytorium. Klucze urządzeń fiskalnych związane ze skompromitowanym kluczem prywatnym producenta muszą być wymienione. Identyczna sytuacja zaistnieje w przypadku wygaśnięcia ważności certyfikatu klucza publicznego dostarczonego przez producenta.

W komunikacji kasy z usługą EventHub chmury Azure należy użyć uwierzytelniania jednostronnego z wykorzystaniem jednorazowego biletu uwierzytelniającego. Certyfikaty repozytorium oraz usług przyjmowania danych do chmury publicznej przekazywane są do urządzanie fiskalnego podczas procesu fiskalizacji zgodnie z dokumentem „Specyfikacja komend dla Centralnego Repozytorium danych o sprzedaży detalicznej".

3 Certyfikaty urządzenia fiskalnego

Urządzenie fiskalne musi posiadać przyporządkowane dwie pary unikalnych kluczy asymetrycznych. Jedna z par kluczy wykorzystywana jest do komunikacji TLS z serwerem CPD oraz chmurą publiczną. Druga para kluczy wykorzystywana jest do podpisywania i szyfrowania wymienianych danych.

Klucze publiczne o długości 2048 bitów muszą być podpisane certyfikatem CA producenta algorytmem RSA z dopełnieniem PKCS1 w wersji 1.5 z wykorzystaniem funkcji skrótu SHA-256 (sha256WithRSAEncryption) wyszczególnionym w sekcji 5 dokumentu RFC 4055, w postaci certyfikatu X.509 w wersji 3 (X.509v3) opisanym w dokumencie RFC 5280. Wymagane jest umieszczenie w nazwie podmiotu (CN – commonName) tylko numeru unikatowego kasy fiskalnej. Ważność certyfikatu urządzenia fiskalnego nie może przekroczyć 20 lat (zalecany okres ważności to pięć lat), a data ważności certyfikatu kasy nie może wykraczać poza datę ważności certyfikatu producenta. Poszczególne certyfikaty kas muszą charakteryzować się przynajmniej następującymi cechami oznaczonymi jako krytyczne (critical):

- certyfikat do komunikacji TLS:
  - Key Usage: digitalSignature
  - Extended Key Usage: clientAuth (TLS WWW client authentication)
- certyfikat do podpisywania i szyfrowania:
  - Key Usage: digitalSignature, nonRepudiation, keyEncipherment

Zawartość certyfikatów – wymagania szczegółowe:

- commonName [CN] = **wymagany**

OID description: [2.5.4.3] {joint-iso-itu-t(2) ds(5) attributeType(4) commonName(3)}

- countryName [C] = **wymagany**

OID description: [2.5.4.6] {joint-iso-itu-t(2) ds(5) attributeType(4) countryName(6)}

- organizationName [O] = **wymagany**

OID description: [2.5.4.10] {joint-iso-itu-t(2) ds(5) attributeType(4) organizationName(10)}

- localityName [L] = opcjonalny

OID description: [2.5.4.7] {joint-iso-itu-t(2) ds(5) attributeType(4) localityName(7)}

- stateOrProvinceName = opcjonalny

OID description: [2.5.4.8] {joint-iso-itu-t(2) ds(5) attributeType(4) stateOrProvinceName(8)}

- organizationalUnitName [OU] = opcjonalny

OID description: [2.5.4.11] {joint-iso-itu-t(2) ds(5) attributeType(4) organizationalUnitName(11)}

- emailAddress [E] = opcjonalny

OID description: [1.2.840.113549.1.9.1] {iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-9(9) emailAddress(1)}

- organizationIdentifier = opcjonalny

OID description: [2.5.4.97] {joint-iso-itu-t(2) ds(5) attributeType(4) organizationIdentifier(97)}

4 Algorytmy kryptograficzne

Algorytmy kryptograficzne zostały wybrane z listy algorytmów wskazanych w specyfikacji RFC 7518, w której opisano również sposób implementacji danego algorytmu. Za podstawę kryptografii asymetrycznej przyjęto algorytm RSA o długości klucza minimum 2048 bitów, natomiast wykorzystywanym algorytmem symetrycznym jest algorytm AES z blokiem o rozmiarze 128 bitów.

4.1 Podpisywanie

Algorytmem wykorzystywanym do podpisywania danych jest algorytm RSA z dopełnieniem PKCS1 w wersji 1.5 (RSASSA-PKCS1-v1_5) opisany w sekcji 8.2 specyfikacji RFC 3447 oraz w sekcji 3.3 dokumentu RFC 7518 wraz z funkcją skrótu SHA256 opisaną w dokumencie FIPS 180-4. W nagłówku JOSE obiektu JWS podpisanych danych JPK w atrybucie „*alg*" symbol algorytmu przyjmie wartość RS256 ("*alg*":"*RS256*"). Dodatkowo w atrybucie „*jpkcertificate*" należy zamieścić certyfikat z kluczem publicznym urządzenia fiskalnego użytym do podpisania danych w formacie binarnym DER zakodowanym Base64 bez znacznika początku i końca certyfikatu oraz bez znaków końca linii.

4.2 Szyfrowanie symetryczne

Algorytmem wykorzystywanym do szyfrowania podpisanych danych jest algorytm AES z blokiem i kluczem o rozmiarze 128 bitów w trybie CBC z metodą tworzenia kodu uwierzytelnienia wiadomości (MAC - Message Authentication Code) przy użyciu funkcji skrótu (haszowania) SHA-256 opisany w sekcji 5.2.3 dokumentu RFC 7518 (AES_128_CBC_HMAC_SHA_256). W implementacji mechanizmu szyfrowania należy użyć następującej specyfikacji algorytmu AES:

Klucz haszujący	MAC Key Size	16 bytes
Klucz szyfrujący	Encryption Key Size	16 bytes
Tryb szyfru	Cipher Mode	CBC (Chain Block Chaining)
Dopełnienie	Padding	PKCS#7
Rozmiar bloku	Block Size	16 bytes
Wektor inicjujący	Initialization Vector	16 bytes
Kod uwierzytelnienia wiadomości	Message Authentication Code	SHA256

W nagłówku JOSE obiektu JWE zaszyfrowanych danych w atrybucie „*enc*" symbol algorytmu szyfrującego przyjmie wartość A128CBC-HS256 ("*enc*": "*A128CBC-HS256*").

4.3 Szyfrowanie klucza szyfrującego

Algorytmem wykorzystywanym do szyfrowania klucza szyfrującego jest algorytm RSA z dopełnieniem PKCS1 w wersji 1.5 (RSAES-PKCS1-V1_5) opisany w sekcji 7.2 specyfikacji RFC 3447 oraz w sekcji 4.2 dokumentu RFC 7518. Klucz publiczny do szyfrowania klucza szyfrującego o długości 2048 bitów w postaci certyfikatu X.509 podpisany przez certyfikat główny ministerstwa zostanie udostępniony publicznie oddzielnie dla środowiska testowego oraz produkcyjnego. W nagłówku JOSE obiektu JWE zaszyfrowanych danych w atrybucie „*alg*" symbol algorytmu szyfrującego klucz szyfrujący przyjmie wartość RSA1_5 ("*alg*":"*RSA1_5*"). Dodatkowo w atrybucie „*kid*" należy zamieścić numer seryjny oraz wystawcę certyfikatu użytego do zaszyfrowania klucza szyfrującego. Struktura atrybutu "kid" ma postać dwóch wartości rozdzielonych przecinkiem - numeru seryjnego w postaci szesnastkowej (cyfry i wielkie litery) oraz nazwy wystawcy certyfikatu zawierającej tylko 'common name'.

5 Szyfrowanie komend oraz danych

Wszystkie dane (komendy, odpowiedzi, raporty itp.) przechodzące przez publiczną chmurę są podpisywane i szyfrowane zarówno przez urządzenie fiskalne jak i serwer CPD. Klucz publiczny jest przesyłany w formie certyfikatu X.509 podpisanego przez wystawcę w formacie binarnym DER zakodowanym Base64 bez znacznika początku i końca certyfikatu oraz bez znaków końca linii. Urządzenie fiskalne powinno posiadać przyporządkowane dwie pary unikalnych kluczy asymetrycznych. Jedna z par kluczy wykorzystywana jest do dwustronnej komunikacji TLS z chmurą publiczną oraz serwerem CPD. Druga para kluczy wykorzystywana jest do podpisywania i szyfrowania wymienianych danych. Klucze prywatne przechowywane są w urządzeniu fiskalnym. Klucz publiczny używany do szyfrowania danych powinien zostać przesłany do serwera CPD w celu weryfikacji wystawcy oraz późniejszego użycia do komunikacji z kasą. Przesłany klucz publiczny przyporządkowany jest dokładnie jednemu urządzaniu fiskalnemu i przechowywany w zasobach ministerstwa.

Nazwy atrybutów kluczy urządzenia fiskalnego

Nazwa tagu	Opis
-digitalCertificateCashRegisterTLS	Certyfikat urządzenia fiskalnego do komunikacji TLS z serwerem CPD i chmurą Azure.
-privateKeyCashRegisterTLS	Klucz prywatny urządzenia fiskalnego do komunikacji TLS z serwerem CPD i chmurą Azure.
-digitalCertificateCashRegisterEncrypt	Certyfikat urządzenia fiskalnego do szyfrowania wymienianych danych
-privateKeyCashRegisterEncrypt	Klucz prywatny urządzenia fiskalnego do podpisywania wymienianych danych

Repozytorium operuje na trzech parach unikalnych kluczy:

- do dwustronnej komunikacji TLS urządzenia fiskalnego z serwerem CPD,
- do dwustronnej komunikacji TLS kasy z chmurą publiczną,
- do podpisywania i szyfrowania przesyłanych danych.

Klucze prywatne przechowywane są w zasobach ministerstwa, natomiast klucze publiczne repozytorium oraz klucz publiczny CA usługi przyjmowania danych chmury publicznej są przesyłane do kasy podczas procesu fiskalizacji w postaci certyfikatu X.509 podpisanego przez wystawcę w formacie binarnym DER zakodowanym Base64 bez znacznika początku i końca certyfikatu oraz bez znaków końca linii. Dodatkowo repozytorium przechowuje certyfikaty zaufanych producentów kas w celu weryfikacji kluczy publicznych urządzeń fiskalnych.

Nazwy tagów kluczy Repozytorium

Nazwa tagu	Opis
-digitalCertificateWebApiTLS	Certyfikat WebAPI do komunikacji TLS z urządzeniem fiskalnym
-privateKeyWebApiTLS	Klucz prywatny WebAPI do komunikacji TLS z urządzeniem fiskalnym
-digitalCertificateCPDServerTLS	Certyfikat serwera CPD do komunikacji TLS z urządzeniem fiskalnym
-privateKeyCPDServerTLS	Klucz prywatny serwera CPD do komunikacji TLS z urządzeniem fiskalnym
-digitalCertificateRepositoryEncrypt	Certyfikat serwera CPD do szyfrowania wymienianych danych
-privateKeyRepositoryEncrypt	Klucz prywatny serwera CPD do podpisywania wymienianych danych
-digitalCertificateAzureEventHubTLS	Certyfikat CA usługi EventHub na chmurze Azure do komunikacji TLS

5.1 Podpisywanie i szyfrowanie komend

Podpisywanie oraz szyfrowanie komend realizowane jest z wykorzystaniem obiektów JWS oraz JWE struktury JSON w formacie JOSE z wykorzystaniem kodowania Base64URL. W pierwszej kolejności realizowany jest podpis, a następnie szyfrowanie podpisanej komendy.

Etapy przygotowania paczki:

- utworzenie obiektu JSON zgodnie z formatem danej komendy po przez wypełnienie pola "*attributes*" parametrami opisanymi w dokumencie „Specyfikacja komend dla Centralnego Repozytorium danych o sprzedaży detalicznej",
- utworzenie obiektu JWS - podpisanie obiektu JSON zawierającego pole "*attributes*" algorytmem RSA z dopełnieniem PKCS1-v1.5 i funkcją haszującą SHA-256 oraz dodanie parametrów nagłówka JOSE:
    - "*alg*" zawierającego symbol użytego algorytmu podpisu (RS256),
    - "*x5c*" zawierającego zgodnie z opisem w sekcji 4.1.6 dokumentu RFC 7515 jednoelementową tablicę JSON z certyfikatem podpisującym ministerstwa w formacie DER zakodowanym Base64 bez znacznika początku i końca certyfikatu oraz bez znaków końca linii,
- utworzenie obiektu JWE - zaszyfrowanie obiektu JWS algorytmem AES 128 CBC z uwierzytelnieniem wiadomości funkcją skrótu SHA-256 (AES_CBC_HMAC_SHA2) oraz klucza algorytmem RSA z dopełnieniem PKCS1-v1.5 oraz dodanie parametrów nagłówka JOSE:
    - "*alg*" zawierającego symbol użytego algorytmu podpisu (RSA1_5),
    - "*enc*" zawierającego symbol użytego algorytmu szyfrującego (A128CBC-HS256),
    - "*kid*" zawierającego numer seryjny oraz wystawcę certyfikatu klucz publicznego użytego do zaszyfrowania klucza szyfrującego,
- utworzenie paczki z polem "*commandId*" i "*encryptedCommand*" zawierającym obiekt JWE.

Szczegółowy opis podpisywania oraz szyfrowania zaprezentowany jest w załącznikach A.2 oraz A.3.

5.2 Podpisywanie i szyfrowanie danych

Podpisywanie oraz szyfrowanie danych realizowane jest z wykorzystaniem obiektów JWS oraz JWE struktury JSON w formacie JOSE z wykorzystaniem kodowania Base64URL. W pierwszej kolejności realizowany jest podpis, a następnie szyfrowanie podpisanych danych.

Etapy przygotowania paczki:

- utworzenie obiektu JSON zgodnie z aktualnym schematem dokumentu JPK,
- utworzenie obiektu JWS - podpisanie obiektu JSON lub skompresowanych danych algorytmem RSA z dopełnieniem PKCS1-v1.5 i funkcją haszującą SHA-256 oraz dodanie parametrów nagłówka JOSE:
  - ”*alg*” zawierającego symbol użytego algorytmu podpisu (RS256),
  - ”*jpkmetadate*” zawierającego zakodowany w Base64 obiekt JSON składający się z parametrów opisujących numer korelacyjny komendy oraz metodę kompresowania,
  - ”*jpkcertificate*” zawierający certyfikat klucza publicznego użyty do podpisu w formacie DER zakodowanym Base64 bez znacznika początku i końca certyfikatu oraz bez znaków końca linii,
- utworzenie obiektu JWE - zaszyfrowanie obiektu JWS algorytmem AES 128 CBC z uwierzytelnieniem wiadomości funkcją skrótu SHA-256 (AES_CBC_HMAC_SHA2) oraz klucza algorytmem RSA z dopełnieniem PKCS1-v1.5 oraz dodanie parametrów nagłówka JOSE:
  - ”*alg*” zawierającego symbol użytego algorytmu podpisu (RSA1_5),
  - ”*enc*” zawierającego symbol użytego algorytmu szyfrującego (A128CBC-HS256),
  - ”*kid*” zawierającego numer seryjny oraz wystawcę certyfikatu klucza publicznego użytego do zaszyfrowania klucza szyfrującego,
- opcjonalnie podział danych na części nie większe niż 256 kB - limit usługi EventHub,
- utworzenie paczki/paczek z odpowiednimi wartościami atrybutów "*commandId*", "*packageNr*", "*isLast*" oraz polem "*encryptedData*" zawierającym cały obiekt JWE albo podzielone binarnie jego części.

Szczegółowy opis podpisywania oraz szyfrowania zaprezentowany jest w załącznikach A.4 oraz A.5.

Załącznik A

A.1 Funkcje użyte w opisach

- UTF8 - funkcja zapisująca zbiór bajtów w łańcuch znaków w kodowaniu UTF8.
- SHA256 - funkcja skrótu używająca algorytmu SHA256.
- BASE64 - funkcja kodująca dane zgodnie ze specyfikacją RFC 4648.
- DecodeB64 - funkcja dekodująca dane zgodnie ze specyfikacją RFC 4648.
- BASE64URL - funkcja kodująca dane zgodnie z załącznikiem C dokumentu RFC 7515.
- DecodeB64URL - funkcja dekodująca dane zgodnie z załącznikiem C dokumentu RFC 7515.
- RANDOM - funkcja generująca pseudolosowy ciąg bajtów - Strong Random Generator (RNG).
- DEFLATE - funkcja kompresująca dane algorytmem opisanym w dokumencie RFC 1951.
- INFLATE - funkcja dekompresująca dane algorytmem w dokumencie RFC 1951.
- RS256 - funkcja podpisująca algorytmem RSA z wykorzystaniem SHA256.
- RS256Verify - funkcja weryfikująca podpis algorytmem RSA z wykorzystaniem SHA256.
- HS256 - funkcja generująca MAC algorytmem SHA256 zgodnie z RFC 2104, argumenty:
  - dodatkowe dane autoryzujące (AAD),
  - klucz haszujący.
- A128CBC – funkcja szyfrująca dane, argumenty to:
  - jawny tekst
  - klucz szyfrujący
  - wektor inicjujący (IV).
- A128CBCDecrypt – funkcja odszyfrowująca dane, argumenty to:
  - zaszyfrowany tekst
  - klucz szyfrujący
  - wektor inicjujący (IV).
- RSA1_5 - funkcja szyfrująca klucz symetryczny algorytmem RSA.
- RSA1_5Decrypt - funkcja odszyfrowująca klucz symetryczny algorytmem RSA.
- || - operator łączący dwa łańcuchy znaków (np. ‘Hello’ || ‘ world’ => ‘Hello world’).

A.2 Podpisywanie komend

Załącznik przedstawia sposób podpisywania komend wysyłanych z repozytorium do urządzenia fiskalnego, wzorowany na opisie zawartym w załączniku A.2 specyfikacji RFC 7515.

1. Przygotowanie chronionego nagłówka podpisu (JWS Protected Header):
   - wyszczególnienie użytego algorytmu podpisu w parametrze "*alg*",
   - dodanie parametru "*x5c*" zawierającego jednoelementową tablicę z certyfikatem klucza publicznego ministerstwa użytego do podpisu w formacie DER zakodowanym Base64 bez znacznika początku i końca certyfikatu oraz bez znaków końca linii,
     UWAGA: ciąg znaków reprezentujący certyfikat może zostać poszerzony o wstawienie znaku specjalnego '\' poprzedzającego znak '/'.

   Skrócona postać nagłówka w formacie JSON:

   ```
   JWS_PH => {"alg":"RS256","x5c":["MIIFHDCCAwSgAwIBAgITOgAA … 0NCJ2zprYt8XrNO728ljyA=="]}
   ```

   Pełna postać z wykorzystaniem certyfikatu testowego B.1.1 przedstawiona jest w punkcie B.2.1.

2. Przygotowany nagłówek przekształcany jest przez kodowanie Base64URL:

   ```
   JWS_PH_URL => BASE64URL(JWS_PH)
   ```

   Skrócona postać nagłówka w formacie Base64URL:

   ```
   JWS_PH_URL => eyJhbGciOiJSUzI1NiIsIng1 … 0OFhyTk83MjhsanlBPT0iXX0
   ```

   Pełna postać nagłówka zakodowanego Base64URL przedstawiona jest w punkcie B.2.2.

3. Przygotowanie zawartości komendy do podpisu:

   ```
   JWS_DATA => {"attributes":{"cpdServiceName":"KFD"}}
   ```

   W przykładzie użyto komendę CMD01 nakazującą urządzeniu fiskalnemu połączenie się z serwerem CPD i wywołanie wskazanej usługi (KFD – wykonanie komendy CMD08: Wyślij certyfikaty urządzenia fiskalnego).

4. Przygotowane dane należy zakodować w Base64URL:

   ```
   JWS_DATA_URL => BASE64URL(JWS_DATA)
   ```

   Postać przykładowych danych w formacie Base64URL:

   ```
   JWS_DATA_URL => eyJhdHRyaWJ1dGVzIjp7ImNwZFNlcnZpY2VOYW1lIjoiS0ZEIn19
   ```

   Pełna postać danych zakodowanego Base64URL przedstawiona jest w punkcie B.2.3.

5. Przygotowanie danych do popisu polegające na połączeniu nagłówka i danych zakodowanych w Base64URL rozdzielonych kropką:

   ```
   JWS_SIGNING_INPUT => JWS_PH_URL||.||JWS_DATA_URL
   ```

   Skrócona postać przykładowych danych do popisu w formacie Base64URL:

```
JWS_SIGNING_INPUT => eyJhbGciOiJSUzI1NiIsIng1 … cnZpY2VOYW1lIjoiS0ZEIn19
```

Pełna postać przykładowych danych do podpisu zakodowanych Base64URL przedstawiona jest w punkcie B.2.4.

Wartość funkcji skrótu SHA256 w formacie szesnastkowym obliczona na danych do podpisu z punktu B.2.4:

```
68e24de4af1da3859f0e8658b229f8aa01950d56cfa1bbf8c4cb3c55d49683e9
```

6. Tworzenie podpisu z wykorzystaniem algorytmu RSA z funkcja skrótu SHA256 oraz klucza prywatnego ministerstwa i zakodowanie podpisu Base64URL:

```
JWS_SIGN => RS256(JWS_SIGNING_INPUT,RSA_PRIVATE_KEY)
```

Otrzymany podpis zapisywany jest w kodowaniu Base64URL:

```
JWS_SIGN_URL => BASE64URL(JWS_SIGN)
```

Skrócona postać podpisu w formacie Base64URL:

```
JWS_SIGN_URL => HlKhuau2-ZLYoBip8ed2J7Js … 8o7mxe-FFv1RLSP1zoMU-NGHw
```

Pełna postać przykładowego podpisu zakodowanego Base64URL przedstawiona jest w punkcie B.2.5.

7. Przygotowanie obiektu JWS polegające na połączeniu danych do podpisu i otrzymanego podpisu zakodowanych w Base64URL rozdzielonych kropką:

```
JWS => JWS_SIGNING_INPUT||.||JWS_SIGN_URL
```

```
albo
```

```
JWS => JWS_PH_URL||.||JWS_DATA_URL||.||JWS_SIGN_URL
```

Pełna postać przykładowego obiektu JWS z wykorzystaniem certyfikatu B.1.1 przedstawiona jest w punkcie B.2.6.

A.3 Szyfrowanie komend

Załącznik przedstawia sposób szyfrowania komend wysyłanych z repozytorium do urządzenia fiskalnego, wzorowany na opisie zawartym w załączniku A.2 specyfikacji RFC 7516.

1. Przygotowanie chronionego nagłówka szyfrowania (JWE Protected Header):
    - wyszczególnienie algorytmu asymetrycznego szyfrowania klucza w parametrze ”*alg*”,
    - wyszczególnienie algorytmu symetrycznego szyfrowania danych w parametrze ”*enc*”,
    - dodanie parametru ”*kid*” zawierającego numer seryjny w postaci szesnastkowej oraz nazwę wystawcy „common name” certyfikatu klucza publicznego użytego do zaszyfrowania klucza szyfrującego.

    Przykładowa postać nagłówka w formacie JSON:

    ```
    JWE_PH => {"enc":"A128CBC-HS256","alg":"RSA1_5","kid":"0A2B4C6D8E0F,CN=Producent"}
    ```

    Pełna postać z wykorzystaniem certyfikatu testowego B.1.2 przedstawiona jest w punkcie B.3.1.

2. Przygotowany nagłówek przekształcany jest przez kodowanie Base64URL:

    ```
    JWE_PH_URL => BASE64URL(JWE_PH)
    ```

    Pełna postać nagłówka zakodowanego Base64URL przedstawiona jest w punkcie B.3.2.

3. Przygotowanie danych używanych do szyfrowania symetrycznego:
    - wygenerowanie 32 bajtowego losowego klucza algorytmu szyfrującego,
    - wydzielenie pierwszych 16 bajtów klucza algorytmu szyfrującego jako klucz haszujący,
    - wydzielenie ostatnich 16 bajtów klucza algorytmu szyfrującego jako klucz szyfrujący,
    - wygenerowanie 16 bajtowego losowego wektora inicjującego,

    ```
    JWE_AES_CEK => RANDOM(32)

    JWE_MAC_KEY => FIRST 16 BYTES FROM JWE_AES_CEK
    JWE_AES_KEY => LAST 16 BYTES FROM JWE_AES_CEK

    JWE_AES_IV => RANDOM(16)
    ```

    Wartości zastosowane w przykładach przedstawiono w punkcie B.3.3.

4. Zaszyfrowanie klucza algorytmu szyfrującego (Content Encryption Key) składającego się z klucza haszującego i klucza szyfrującego z wykorzystaniem algorytmu asymetrycznego RSA kluczem publicznym urządzenia fiskalnego i zakodowanie w Base64URL:

    ```
    JWE_CEK_URL => BASE64URL(RSA1_5(JWE_AES_CEK,RSA_PUBLIC_KEY))
    ```

    Przykładowa wartość zakodowana w Base64URL z użyciem certyfikatu B.1.2 przedstawiona jest w punkcie B.3.4.

5. Zakodowanie wektora inicjującego w Base64URL:

    ```
    JWE_IV_URL => BASE64URL(JWE_AES_IV)
    ```

    Przykładowa wartość zakodowanego wektora inicjującego w Base64URL przedstawiona jest w punkcie B.3.5.

6. Zaszyfrowanie podpisanej komendy algorytmem symetrycznym:

```
JWE_TXT_URL => BASE64URL(A128CBC(JWS,JWE_AES_KEY,JWE_AES_IV))
```

Przykładowa wartość zakodowanego Base64URL przedstawiona jest w punkcie B.3.6.

7. Przygotowanie dodatkowych danych uwierzytelniających (Additional Authenticated Data) poprzez użycie utworzonego chronionego nagłówka szyfrowania (JWE Protected Header):

```
JWE_AAD_URL => JWE_PH_URL
```

Przykładową wartość dodatkowych danych uwierzytelniających w postaci Base64URL przedstawiono w punkcie B.3.7a.

Obliczenie AL (ADD Length) - liczby bitów dodatkowych danych uwierzytelniających (AAD) oraz przedstawienie tej wartości w postaci 64-bitowej liczby w formacie Big-Endian.

```
JWE_AL => JWE_AAD_URL BITS LENGTH CONVERT TO 64 BIT BIG-ENDIAN VALUE
```

Przykład utworzenia tablicy bajtów odzwierciedlającej długość ADD przedstawiony jest w punkcie B.3.7b.

8. Wyliczenie etykiety uwierzytelniającej (Authentication Tag) z wykorzystaniem funkcji HMAC z funkcją haszującą SHA-256 przy użyciu klucza haszującego i połączonych tablic bajtów dodatkowych danych uwierzytelniających (AAD), wektora inicjującego (IV), zaszyfrowanych danych oraz wektora długości AAD (AL) i użycie pierwszych 16 bajtów wyliczonej wartości:

```
JWE_AT_DATA => JWE_AAD_URL_BYTES||JWE_AES_IV||DecodeB64URL(JWE_TXT_URL)||JWE_AL

JWE_AT_256 => HS256(JWE_AT_DATA, JWE_MAC_KEY)

JWE_AT => FIRST 16 BYTES FROM JWE_AT_256
```

Przykładowa wartość przedstawiona jest w punkcie B.3.8a, a sposób wyliczenia w punkcie B.3.8b.

9. Przygotowanie obiektu JWE polegające na połączeniu chronionego nagłówka szyfrowania (JWE Protected Header), zaszyfrowanego klucza algorytmu szyfrującego (CEK), wektora inicjującego (IV), zaszyfrowanych danych oraz etykiety uwierzytelniającej (Authentication Tag) rozdzielonych kropką:

```
JWE => JWE_PH_URL||'.'||JWE_CEK_URL||'.'||JWE_IV_URL||'.'||JWE_TXT_URL||'.'||JWE_AT_URL
```

Przykładowa wartość zakodowanego Base64URL przedstawiona jest w punkcie B.3.9.

A.4 Podpisywanie danych

Załącznik przedstawia sposób podpisywania danych wysyłanych z urządzenia fiskalnego do repozytorium, wzorowany na opisie zawartym w załączniku A.2 specyfikacji RFC 7515.

1. Przygotowanie chronionego nagłówka podpisu (JWS Protected Header):
   - wyszczególnienie użytego algorytmu podpisu w parametrze "*alg*",
   - dodanie parametru "*jpkcertificate*" zawierającego certyfikat klucza publicznego użytego do podpisu w formacie DER zakodowanym Base64 bez znacznika początku i końca certyfikatu oraz bez znaków końca linii,
     UWAGA: ciąg znaków reprezentujący certyfikat może zostać poszerzony o wstawienie znaku specjalnego '\' poprzedzającego znak '/',
   - dodanie opcjonalnego parametru "*jpkmetadata*" zawierającego zakodowany w Base64 obiekt JSON składający się z opcjonalnych parametrów:
     - correlationId - numer korelacyjny, czyli identyfikator "*commandId*" wykonywanej komendy pobranej z usługi WebApi,
     - compression - metoda kompresowania przesyłanych danych:
       - DEFLATE - kompresja algorytmem opisanym w dokumencie RFC 1951,
       - NONE - brak kompresji,
     - UWAGA: ciąg znaków reprezentujący metadane może zostać poszerzony o wstawienie znaku specjalnego '\' poprzedzającego znak '/'.

   Skrócona postać nagłówka w formacie JSON:

   ```
   JWS_PH => {"jpkcertificate":"MIIC … hZiS","alg":"RS256","jpkmetadata":"eyJj … In0="}
   ```

   Pełna postać z wykorzystaniem certyfikatu testowego przedstawiona jest w punkcie B.4.1.

   Przykładowa postać parametru "*jpkmetadata*" w formacie JSON:

   ```
   {"correlationId":"TFD.ZTE1234567890.2018-01-01T01:00:00.000Z"}
   ```

   ```
   {"correlationId":"TFD.ZTE1234567890.2018-01-01T01:00:00.000Z","compression":"DEFLATE"}
   ```

2. Przygotowany nagłówek przekształcany jest przez kodowanie Base64URL:

   ```
   JWS_PH_URL => BASE64URL(JWS_PH)
   ```

   Pełna postać nagłówka zakodowanego Base64URL przedstawiona jest w punkcie B.4.2.

3. Przygotowanie zawartości danych do podpisu i opcjonalnie skompresowanie:

   ```
   JWS_DATA => { "JPK": { "naglowek": {…}, "podmiot1": {…}, "content": [ … ] } }
   ```

   ```
   opcjonalnie skompresowanie:
   ```

   ```
   JWS_DATA => DEFLATE(JWS_DATA)
   ```

   Pełna postać przykładowych nieskompresowanych danych przedstawiona jest w punkcie B.4.3.

4. Przygotowane dane należy zakodować w Base64URL:

   ```
   JWS_DATA_URL => BASE64URL(JWS_DATA)
   ```

Pełna postać przykładowych danych zakodowanych Base64URL przedstawiona jest w punkcie B.4.4.

5. Przygotowanie zawartości do popisu polegające na połączeniu nagłówka i danych zakodowanych w Base64URL rozdzielonych kropką:

```
JWS_SIGNING_INPUT => JWS_PH_URL||.||JWS_DATA_URL
```

6. Tworzenie podpisu z wykorzystaniem algorytmu RSA oraz klucza prywatnego kasy i zakodowanie podpisu Base64URL:

```
JWS_SIGN_URL => BASE64URL(RS256(JWS_SIGNING_INPUT,RSA_PRIVATE_KEY))
```

7. Przygotowanie obiektu JWS polegające na połączeniu danych do podpisu i otrzymanego podpisu zakodowanych w Base64URL rozdzielonych kropką:

```
JWS => JWS_SIGNING_INPUT||.||JWS_SIGN_URL
```

albo

```
JWS => JWS_PH_URL||.||JWS_DATA_URL||.||JWS_SIGN_URL
```

Pełna postać przykładowego obiektu JWS z wykorzystaniem certyfikatu B.1.2 przedstawiona jest w punkcie B.4.5.

A.5 Szyfrowanie danych

Załącznik przedstawia sposób szyfrowania danych wysyłanych z urządzenia fiskalnego do repozytorium, wzorowany na opisie zawartym w załączniku A.2 specyfikacji RFC 7516.

1. Przygotowanie chronionego nagłówka szyfrowania (JWE Protected Header):
    - wyszczególnienie algorytmu asymetrycznego szyfrowania klucza w parametrze "*alg*",
    - wyszczególnienie algorytmu symetrycznego szyfrowania danych w parametrze "*enc*",
    - dodanie parametru "*kid*" zawierającego numer seryjny w postaci szesnastkowej oraz nazwę „common name" wystawcy certyfikatu klucza publicznego użytego do zaszyfrowania klucza szyfrującego.

    Przykładowa postać nagłówka w formacie JSON:

    ```
    JWE_PH => {"enc":"A128CBC-HS256","alg":"RSA1_5","kid":"0A2B4C6D8E0F, CN=Ministerstwo"}
    ```

    Pełna postać z wykorzystaniem certyfikatu testowego B.1.1 przedstawiona jest w punkcie B.5.1.

2. Przygotowany nagłówek przekształcany jest przez kodowanie Base64URL:

    ```
    JWE_PH_URL => BASE64URL(JWE_PH)
    ```

    Pełna postać nagłówka zakodowanego Base64URL przedstawiona jest w punkcie B.5.2.

3. Przygotowanie danych używanych do szyfrowania symetrycznego:
    - wygenerowanie 32 bajtowego losowego klucza algorytmu szyfrującego,
    - wydzielenie pierwszych 16 bajtów klucza algorytmu szyfrującego jako klucz haszujący,
    - wydzielenie ostatnich 16 bajtów klucza algorytmu szyfrującego jako klucz szyfrujący,
    - wygenerowanie 16 bajtowego losowego wektora inicjującego,

    ```
    JWE_AES_CEK => RANDOM(32)

    JWE_MAC_KEY => FIRST 16 BYTES FROM JWE_AES_CEK
    JWE_AES_KEY => LAST 16 BYTES FROM JWE_AES_CEK

    JWE_AES_IV => RANDOM(16)
    ```

    Wartości zastosowane w przykładach przedstawiono w punkcie B.5.3.

4. Zaszyfrowanie klucza algorytmu szyfrującego (Content Encryption Key) składającego się klucza haszującego i klucza szyfrującego z wykorzystaniem algorytmu asymetrycznego RSA kluczem publicznym ministerstwa i zakodowanie w Base64URL:

    ```
    JWE_CEK_URL => BASE64URL(RSA1_5(JWE_AES_CEK,RSA_PUBLIC_KEY))
    ```

    Przykładowa wartość zakodowana w Base64URL z użyciem certyfikatu B.1.1 przedstawiona jest w punkcie B.5.4.

5. Zakodowanie wektora inicjującego (IV) w Base64URL:

    ```
    JWE_IV_URL => BASE64URL(JWE_AES_IV)
    ```

    Przykładowa wartość zakodowanego wektora inicjującego w Base64URL przedstawiona jest w punkcie B.5.5.

6. Zaszyfrowanie podpisanej komendy algorytmem symetrycznym:

```
JWE_TXT_URL => BASE64URL(A128CBC(JWS,JWE_AES_KEY,JWE_AES_IV))
```

Przykładowa wartość zakodowanego Base64URL przedstawiona jest w punkcie B.5.6.

7. Przygotowanie dodatkowych danych uwierzytelniających (Additional Authenticated Data) poprzez użycie utworzonego chronionego nagłówka szyfrowania (JWE Protected Header):

```
JWE_AAD_URL => JWE_PH_URL
```

Przykładowa wartość przedstawiona jest w punkcie B.5.7a.

Obliczenie AL (ADD Length) - liczby bitów dodatkowych danych uwierzytelniających (AAD) oraz przedstawienie tej wartości w postaci 64-bitowej liczby w formacie Big-Endian.

```
JWE_AL => JWE_AAD_URL BITS LENGTH CONVERT TO 64 BIT BIG-ENDIAN VALUE
```

Przykładowa wartość zakodowanego Base64URL przedstawiona jest w punkcie B.5.7b.

8. Wyliczenie etykiety uwierzytelniającej (Authentication Tag) z wykorzystaniem funkcji HMAC z funkcją haszującą SHA-256 przy użyciu klucza haszującego i połączonych tablic bajtów dodatkowych danych uwierzytelniających (AAD), wektora inicjującego (IV), zaszyfrowanych danych oraz wektora długości AAD (AL) i użycie pierwszych 16 bajtów wyliczonej wartości:

```
JWE_AT_DATA => JWE_AAD_URL_BYTES||JWE_AES_IV||DecodeB64URL(JWE_TXT_URL)||JWE_AL

JWE_AT_256 => HS256(JWE_AT_DATA, JWE_MAC_KEY)

JWE_AT => FIRST 16 BYTES FROM JWE_AT_256
```

Przykładowa wartość przedstawiona jest w punkcie B.5.8a, a sposób wyliczenia w punkcie B.5.8b.

9. Przygotowanie obiektu JWE polegające na połączeniu chronionego nagłówka szyfrowania (JWE Protected Header), zaszyfrowanego klucza algorytmu szyfrującego (CEK), wektora inicjującego (IV), zaszyfrowanych danych oraz etykiety uwierzytelniającej (Authentication Tag) rozdzielonych kropką:

```
JWE => JWE_PH_URL||'.'||JWE_CEK_URL||'.'||JWE_IV_URL||'.'||JWE_TXT_URL||'.'||JWE_AT_URL
```

Przykładowa wartość zakodowanego Base64URL przedstawiona jest w punkcie B.5.9.

A.6 Wysyłanie danych

Przygotowanie paczki zawierającej podpisane i zaszyfrowane dane polega na wygenerowaniu identyfikatora paczki w formacie opisanym w dokumencie „Specyfikacja komend dla Centralnego Repozytorium danych o sprzedaży detalicznej” i umieszczenie otrzymanej wartości w parametrze "*commandId*". Natomiast zaszyfrowane dane należy umieścić w parametrze "*encryptedData*". Jeżeli rozmiar tworzonej paczki przekracza wielkość 256kB to należy utworzyć z tym samym identyfikatorem kilka paczek nie przekraczających wskazany limit (cała paczka wraz z parametrami nie może przekraczać maksymalnej wielkości). Kolejny numer paczki należy zamieścić w parametrze "*packageNr*", aczkolwiek dla pojedynczej paczki musi on mieć wartość równą zero. Parametrem wymaganym do scalenia dokumentu jest parametr "*isLast*", którego wartość równa jeden określa ostatnią paczkę w przeciwnym wypadku powinien mieć wartość zero. Poszczególne paczki z uzyskanymi w wyniku podziału fragmentami należy przesłać jako osobny komunikat do usługi EventHub chmury Azure.

Przykład dla dokumentu składającego się tylko z jednego fragmentu:

```
{ "commandId": "DFD.AAA1234567890.2017-07-01T00:00:00.000Z", "packageNr": 0, "isLast":1,
"encryptedData": JWE }
```

Przykład dla dokumentu składającego się z wielu fragmentów:

```
JWE1||JWE2||JWE3 => JWE

{ "commandId": "DFD.AAA1234567890.2017-07-01T00:00:00.000Z", "packageNr": 1, "isLast":0,
"encryptedData": JWE1 }

{ "commandId": "DFD.AAA1234567890.2017-07-01T00:00:00.000Z", "packageNr": 2, "isLast":0,
"encryptedData": JWE2 }

{ "commandId": "DFD.AAA1234567890.2017-07-01T00:00:00.000Z", "packageNr": 3, "isLast":1,
"encryptedData": JWE3 }
```

A.7 Odebranie komendy

Po poprawnym zakończeniu procesu fiskalizacji opisanym w dokumencie „Specyfikacja komend dla Centralnego Repozytorium danych o sprzedaży detalicznej” opierającym się na komunikacji z serwerem CPD kasa przełącza się na komunikację z usługą WebApi umieszczoną w publicznej chmurze Azure. Urządzenie fiskalne komunikując się z WebApi pobiera przygotowane w formacie JSON paczki zawierające podpisane i zaszyfrowane komendy. Każda paczka zawiera w parametrze "*commandId*" wygenerowany identyfikator komendy w formacie opisanym w dokumencie „Specyfikacja komend dla Centralnego Repozytorium danych o sprzedaży detalicznej” oraz podpisaną i zaszyfrowaną komendę w parametrze "*encryptedCommand*".

Przykład komendy składającego się tylko z jednego fragmentu:

```
{ "commandId": "CCS.ZTE1701000901.2018-03-30T09:56:29.062Z", "encryptedCommand": JWE }
```

A.8 Odszyfrowanie komendy

Odebrana komenda ma format obiektu JWE składającego się z rozdzielonych kropką członów - chronionego nagłówka szyfrowania (JWE Protected Header), zaszyfrowanego klucza algorytmu szyfrującego (CEK), wektora inicjującego (IV), zaszyfrowanych danych oraz etykiety uwierzytelniającej (Authentication Tag):

```
JWE_PH_URL||'.'||JWE_CEK_URL||'.'||JWE_IV_URL||'.'||JWE_TXT_URL||'.'||JWE_AT_URL => JWE
```

Wyodrębnienie poszczególnych części pozwoli na poprawne zweryfikowanie i odszyfrowanie pobranej komendy.

1. Odkodowanie Base64URL chronionego nagłówka szyfrowania (JWE Protected Header) pozwoli na pobranie informacji o zastosowanych algorytmach szyfrowania (parametry „*enc*" oraz „*alg*") oraz zidentyfikowaniu użytego certyfikatu (parametr „*kid*").

   ```
   JWE_PH => DecodeB64URL(JWE_PH_URL)
   ```

2. Następnie należy odszyfrować klucz przy użyciu klucza prywatnego kasy, a z otrzymanej 32 bajtowej wartości wydzielić 16 bajtowy klucz haszujący oraz 16 bajtowy klucz szyfrujący:

   ```
   JWE_AES_CEK => RSA1_5Decrypt(DecodeB64URL(JWE_CEK_URL),RSA_PRIVATE_KEY)

   JWE_MAC_KEY => FIRST 16 BYTES FROM JWE_AES_CEK
   JWE_AES_KEY => LAST 16 BYTES FROM JWE_AES_CEK
   ```

3. Odkodowanie Base64URL wektora inicjującego:

   ```
   JWE_AES_IV => DecodeB64URL(JWE_IV_URL)
   ```

4. Przygotowanie dodatkowych danych uwierzytelniających (Additional Authenticated Data) poprzez użycie wyodrębnionego chronionego nagłówka szyfrowania (JWE Protected Header):

   ```
   JWE_AAD_URL => JWE_PH_URL
   ```

   Obliczenie AL (ADD Length) - liczby bitów dodatkowych danych uwierzytelniających (AAD) oraz przedstawienie tej wartości w postaci 64-bitowej liczby w formacie Big-Endian.

   ```
   JWE_AL => JWE_AAD_URL BITS LENGTH CONVERT TO 64 BIT BIG-ENDIAN VALUE
   ```

5. Wyliczenie etykiety uwierzytelniającej (Authentication Tag) z wykorzystaniem funkcji HMAC z funkcją haszującą SHA-256 przy użyciu odszyfrowanego klucza haszującego i połączonych tablic bajtów wyodrębnionych części - dodatkowych danych uwierzytelniających (AAD), wektora inicjującego (IV), zaszyfrowanych danych oraz wektora długości AAD (AL) oraz porównanie pierwszych 16 bajtów wyliczonej wartości z odebraną etykietą uwierzytelniającą (AT):

   ```
   JWE_AT_DATA => JWE_AAD_URL_BYTES||JWE_AES_IV||DecodeB64URL(JWE_TXT_URL)||JWE_AL

   JWE_AT_256 => HS256(JWE_AT_DATA, JWE_MAC_KEY)

   JWE_AT => FIRST 16 BYTES FROM JWE_AT_256
   ```

Pozytywny wynik porównania odebranej i wyliczonej etykiety uwierzytelniającej zapewnia kasie operowanie na wiarygodnych i integralnych danych.

Przykład wyliczenia etykiety uwierzytelniającej przedstawiony jest w punkcie B.3.8b.

10. Odszyfrowanie komendy algorytmem symetrycznym:

```
JWS => A128CBCDecrypt(DecodeB64URL(JWE_TXT_URL),JWE_AES_KEY,JWE_AES_IV)
```

W wyniku poprawnego odszyfrowania danych uzyskany zostanie obiekt JWS, czyli podpisana kluczem publicznym ministerstwa komenda.

A.9 Weryfikacja podpisu komendy

Odszyfrowana komenda ma format obiektu JWS składającego się z rozdzielonych kropką członów - chronionego nagłówka podpisu (JWS Protected Header), zakodowanych danych oraz podpisu:

```
JWS_PH_URL||.||JWS_DATA_URL||.||JWS_SIGN_URL => JWS
```

Weryfikowanie podpisu komendy:

1. Przygotowanie danych do weryfikacji popisu polegające na połączeniu nagłówka i danych zakodowanych w Base64URL rozdzielonych kropką:

   ```
   JWS_SIGNING_INPUT => JWS_PH_URL||.||JWS_DATA_URL
   ```

2. Odkodowanie chronionego nagłówka podpisu (JWS Protected Header) oraz pobranie informacji o zastosowanym algorytmie podpisu (parametr „*alg*”) i użytego certyfikatu (parametr „*x5c*”):

   ```
   JWS_PH => DecodeB64URL(JWS_PH_URL)
   ```

3. Weryfikacja przesłanego podpisu komendy z wykorzystaniem algorytmu asymetrycznego oraz klucza publicznego ministerstwa pobranego z parametru „*x5c*” chronionego nagłówka podpisu:

   ```
   RS256Verify(JWS_SIGNING_INPUT,JWS_SIGN,RSA_PUBLIC_KEY)
   ```

4. Weryfikacja certyfikatu klucza publicznego ministerstwa pobranego z nagłówka chronionego z certyfikatem pobranym w trakcie procesy fiskalizacji kasy.

Załącznik B

B.1 Przykładowe certyfikaty środowiska testowego

B.1.1 Certyfikat klucza publicznego ministerstwa do podpisywania komend oraz szyfrowania klucza szyfrującego przesyłanych danych z urządzenia fiskalnego do repozytorium:

```
-----BEGIN CERTIFICATE-----
MIIFHDCCAwSgAwIBAgITOgAAAAjmj1WBXU6mOgABAAAACDANBgkqhkiG9w0BAQ0F
ADAWMRQwEgYDVQQDEwtlS2FzeS1TdWJDQTAgFw0xNzA4MjIwNjMzMTNaGA8yMDky
MDcxMDEyMjcyOVowgcgxCzAJBgNVBAYTAlBMMRQwEgYDVQQIEwtNYXpvd2llY2tp
ZTERMA8GA1UEBxMIV2Fyc3phd2ExHzAdBgNVBAoMFk1pbmlzdGVyc3R3byBGaW5h
bnPDs3cxIzAhBgNVBAsTGkRlcGFydGFtZW50IEluZm9ybWF0eXphY2ppMR4wHAYD
VQQDExV0ZXN0LWUta2FzeS5tZi5nb3YucGwxKjAoBgkqhkiG9w0BCQEWG2luZm8u
ZS1kZWtsYXJhY2plQG1mLmdvdi5wbDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCC
AQoCggEBAMvyVXGj8YNhy6P28bKj9MleA7+QXKCtPJZ4M6MIxiaqA4lodd9No+Ws
gRETVzEPiB8raL9n3uM+RBFwK2A4VvuAWuGZx2drkfMZnpSVFLOsQnadB1rjBCY5
G/pMX6eI7Bltx4XFYK/1cY1U+mFVc94Ryfyxyy0ZWSD8IGV9n0AilDpRfIJBOu5a
3oquz8ZZGuWyU95KWBKRAD7SV2bpTlYWX4UHhTe323HTYL3rDbKP73HAoylObSmS
vmB9MyNzWgBf73UOHmzXPqpuRbLFnR+llTA0FA8kOylxtijygMXpICOai7av2ofG
t65v0GJg5wlJuqWvkQXFUyoyGGUYaQsCAwEAAaOBrTCBqjAdBgNVHQ4EFgQUx7xK
j1TXCorOExa2hY/jdz6NkA0wHwYDVR0jBBgwFoAUBb+Partd6TV4PV1kTUrtJads
SdowWgYIKwYBBQUHAQEETjBMMEoGCCsGAQUFBzAChj5maWxlOi8vLy9zYXAtd2lu
LTgyNi9DZXJ0RW5yb2xsL3NhcC13aW4tODI2X2VLYXN5LVN1YkNBKDEpLmNydDAM
BgNVHRMBAf8EAjAAMA0GCSqGSIb3DQEBDQUAA4ICAQCkdUR2DhgiexUW+y2rgaE6
orWBPyMxveH2IPv0rPGzqdgUFcNH816YzDorEnOAvbRLB8BaoH+Wn/eElAQxqE5+
47VgScIUF4oNHwXnnflR1XRoYcFZ/fBkIW2nfOK1C8y2vHtZG1QEyyVD/cxv7ubg
OiJfOYScsHv5DItStFUBclvg3xrFi2zG5ahblMwqCGrvgPKOxR9+mXGD+eoThBHE
P6aJF3Zu41mVwT/4cbSr5m3c77deEQ2CpQPGL874PiHy9omkjev9F5yoBzI7ypha
lyEIdbASU0UiUErjbS+hnw0RErV1bQQzQfS7qiKMBZTM4pzOv/Ro6f+OcBf7c16X
tHrEgli/aNagKo34nFhUscQcUTCh3MsCKuVSZU3dbCdSLIvdoJIS5FLP+gr8LbQW
9uR/NgwJhYr/w06k6AOF+TaJw8eakv5ELDOuzhipqB63BuMSCGFZcUQ2bDhdc5gc
V9GlNgVEXmToee3fn89OQTC7GrCwFzNxwAM6gJ0MARyWl5Hmgr/pOb1MX5Vehgao
HpjoveMAacONbtiOwfMUyhPdCJmnLP67lokvGq7PDJ/DUbespAqVm91TM6QbWjda
nKGB6kJY+7H5ESI8sp/nzjHXdZxeIPO7lOTItKdRW82kRcBR9TNDSS6rt5sI16LW
0NCJ2zprYt8XrNO728ljyA==
-----END CERTIFICATE-----
```

B.1.2 Certyfikat klucza publicznego urządzenia fiskalnego do podpisywania danych wytworzonych przez kasę oraz szyfrowania klucza szyfrującego przesyłanych komend wysyłanych do urządzenia fiskalnego:

```
-----BEGIN CERTIFICATE-----
MIICfjCCAeegAwIBAgIQzILUrkD2iqNCxC1ZrG7UBDANBgkqhkiG9w0BAQsFADAU
MRIwEAYDVQQDEwlNRiBlLUthc3kwHhcNMTcxMTE1MDk0MDAzWhcNMTkxMTE1MDk0
MDAyWjAYMRYwFAYDVQQDEw1aVEUxNzAxMDAwOTAxMIIBIjANBgkqhkiG9w0BAQEF
AAOCAQ8AMIIBCgKCAQEAwy3Kc3oTipU45lOIX6K3rruFY83vMqYJqwoWzrRVmZn8
5cNHSwoa/f96AW0/akADbJ3uo7U8oWhTF/rj8xIds4uimrN1YiPFmbXAMaeRQDbB
a/qvI5SRQtK9Bmse7KyspIFXVfEWP17OkDiBEZ/n+NC6ERzzKkxA3gMRQFGpHUSQ
2+EOi7kykPGi1f8Yh/2czd+FBvyrp8oSjyX951DdCsqG+rIwlz9p8PeoFwUggwhb
2tMl48U3nD9gZGXLuGOMBZ3nJ9U3fHVdi7XCpvn0PqvTSZLNL45yqXETu6bAZWB5
Ab4q5EVvI4unrjnJqc3fPD2OLXpINIssg6uqyTVCZQIDAQABo0kwRzBFBgNVHQEE
PjA8gBDGD6f6PWMBTV/bee5Lrlc9oRYwFDESMBAGA1UEAxMJTUYgZS1LYXN5ghCz
wtV5n24/mUCWe9d7xth6MA0GCSqGSIb3DQEBCwUAA4GBAKRTZFPJY5ObY4VVGpJL
l4Xb2JNtWpNXdwPs3N8I2rliGc0dxyq8R4C9X125G0GLgXXXTMDtnE1k+xmCk0aU
6bj2xpfezLhW6i1+mmHTB/2+JhsKp5oRRTXg8SpH5G1vwQI9ek9B/bYvn72nKrUa
Tp3PZsmCNqm1D0VayfTRhZiS
-----END CERTIFICATE-----
```

B.2 Przykładowe dane procesu podpisywania komendy w środowisku testowym

B.2.1 Pełna postać chronionego nagłówka podpisu komend przesyłanych do kasy:

{"alg":"RS256","x5c":["MIIFHDCCAwSgAwIBAgITOgAAAAjmj1WBXU6mOgABAAAACDANBgkqhkiG9w0BAQ0FADAWMRQ
wEgYDVQQDEwtlS2FzeS1TdWJDQTAgFw0xNzA4MjIwNjMzMTNaGA8yMDkyMDcxMDEyMjcyOVowgcgxCzAJBgNVBAYTAlBMM
RQwEgYDVQQIEwtNYXpvd2llY2tpZTERMA8GA1UEBxMIV2Fyc3phd2ExHzAdBgNVBAoMFk1pbmlzdGVyc3R3byBGaW5hbnP
Ds3cxIzAhBgNVBAsTGkRlcGFydGFtZW50IEluZm9ybWF0eXphY2ppMR4wHAYDVQQDExV0ZXN0LWUta2FzeS5tZi5nb3Yuc
GwxKjAoBgkqhkiG9w0BCQEWG2luZm8uZS1kZWtsYXJhY2plQG1mLmdvdi5wbDCCASIwDQYJKoZIhvcNAQEBBQADggEPADC
CAQoCggEBAMvyVXGj8YNhy6P28bKj9MleA7+QXKCtPJZ4M6MIxiaqA4lodd9No+WsgRETVzEPiB8raL9n3uM+RBFwK2A4V
vuAWuGZx2drkfMZnpSVFLOsQnadB1rjBCY5G/pMX6eI7Bltx4XFYK/1cY1U+mFVc94Ryfyxyy0ZWSD8IGV9n0AilDpRfIJ
BOu5a3oquz8ZZGuWyU95KWBKRAD7SV2bpTlYWX4UHhTe323HTYL3rDbKP73HAoylObSmSvmB9MyNzWgBf73UOHmzXPqpuR
bLFnR+llTA0FA8kOylxtijygMXpICOai7av2ofGt65v0GJg5wlJuqWvkQXFUyoyGGUYaQsCAwEAAaOBrTCBqjAdBgNVHQ4
EFgQUx7xKj1TXCorOExa2hY/jdz6NkA0wHwYDVR0jBBgwFoAUBb+Partd6TV4PV1kTUrtJadsSdowWgYIKwYBBQUHAQEET
jBMMEoGCCsGAQUFBzAChj5maWxlOi8vLy9zYXAtd2luLTgyNi9DZXJ0RW5yb2xsL3NhcC13aW4tODI2X2VLYXN5LVN1YkN
BKDEpLmNydDAMBgNVHRMBAf8EAjAAMA0GCSqGSIb3DQEBDQUAA4ICAQCkdUR2DhgiexUW+y2rgaE6orWBPyMxveH2IPv0r
PGzqdgUFcNH816YzDorEnOAvbRLB8BaoH+Wn/eElAQxqE5+47VgScIUF4oNHwXnnflR1XRoYcFZ/fBkIW2nfOK1C8y2vHt
ZG1QEyyVD/cxv7ubgOiJfOYScsHv5DItStFUBclvg3xrFi2zG5ahblMwqCGrvgPKOxR9+mXGD+eoThBHEP6aJF3Zu41mVw
T/4cbSr5m3c77deEQ2CpQPGL874PiHy9omkjev9F5yoBzI7yphalyEIdbASU0UiUErjbS+hnw0RErV1bQQzQfS7qiKMBZT
M4pzOv/Ro6f+OcBf7c16XtHrEgli/aNagKo34nFhUscQcUTCh3MsCKuVSZU3dbCdSLIvdoJIS5FLP+gr8LbQW9uR/NgwJh
Yr/w06k6AOF+TaJw8eakv5ELDOuzhipqB63BuMSCGFZcUQ2bDhdc5gcV9GlNgVEXmToee3fn89OQTC7GrCwFzNxwAM6gJ0
MARyWl5Hmgr/pOb1MX5VehgaoHpjoveMAacONbtiOwfMUyhPdCJmnLP67lokvGq7PDJ/DUbespAqVm91TM6QbWjdanKGB6
kJY+7H5ESI8sp/nzjHXdZxeIPO7lOTItKdRW82kRcBR9TNDSS6rt5sI16LW0NCJ2zprYt8XrNO728ljyA=="]}

B.2.2 Pełna postać chronionego nagłówka podpisu komend przesyłanych do urządzenia fiskalnego zakodowana w Base64URL:

eyJhbGciOiJSUzI1NiIsIng1YyI6WyJNSUlGSERDQ0F3U2dBd0lCQWdJVE9nQUFBQWptajFXQlhVNm1PZ0FCQUFBQUNEQU
5CZ2txaGtpRzl3MEJBUTBGQURBV01SUXdFZ1lEVlFRREV3dGxTMkZ6ZVMxVGRXSkRRVEFnRncweE56QTRNakl3TmpNek1U
TmFHQTh5TURreU1EY3hNREV5TWpjeU9Wb3dnY2d4Q3pBSkJnTlZCQVlUQWxCTU1SUXdFZ1lEVlFRSUV3dE5ZWHB2ZDJsbF
kydHBaVEVSTUE4R0ExVUVCeE1JVjJGeWMzcGhkMkV4SHpBZEJnTlZCQW9NRmsxcGJtbHpkR1Z5YzNSM2J5QkdhVzVoYm5Q
RHMzY3hJekFoQmdOVkJBc1RHa1JsY0dGeWRHRnRaVzUwSUVsdVptOXliV0YwZVhwaFkycHBNUjR3SEFZRFZRUURFeFYwWl
hOMExXVXRhMkZ6ZVM1dFppNW5iM111Y0d3eEtqQW9CZ2txaGtpRzl3MEJDUUVXRzJsdVptOHVaUzFrWld0c1lYSmhZMnBs
UUcxbUxtZHZkaTV3YkRDQ0FTSXdEUVlKS29aSWh2Y05BUUVCQlFBRGdnRVBBRENDQVFvQ2dnRUJBTXZ5VlhHajhZTmh5Nl
AyOGJLajlNbGVBNytRWEtDdFBKWjRNNk1JeGlhcUE0bG9kZDlObytXc2dSRVRWekVQaUI4cmFMOW4zdU0rUkJGd0syQTRW
dnVBV3VHWngyZHJrZk1abnBTVkZMT3NRbmFkQjFyakJDWTVHL3BNWDZlSTdCbHR4NFhGWUsvMWNZMVUrbUZWYzk0UnlmeX
h5eTBaV1NEOElHVjluMEFpbERwUmZJSkJPdTVhM29xdXo4WlpHdVd5VTk1S1dCS1JBRDdTVjJicFRsWVdYNFVIaFRlMzIz
SFRZTDNyRGJLUDczSEFveWxPYlNtU3ZtQjlNeU56V2dCZjczVU9IbXpYUHFwdVJiTEZuUitsbFRBMEZBOGtPeWx4dGlqeW
dNWHBJQ09haTdhdjJvZkd0NjV2MEdKZzV3bEp1cVd2a1FYRlV5b3lHR1VZYVFzQ0F3RUFBYU9CclRDQnFqQWRCZ05WSFE0
RUZnUVV4N3hLajFUWENvck9FeGEyaFkvamR6Nk5rQTB3SHdZRFZSMGpCQmd3Rm9BVUJiK1BhcnRkNlRWNFBWMWtUVXJ0Sm
Fkc1Nkb3dXZ1lJS3dZQkJRVUhBUUVFVGpCTU1Fb0dDQ3NHQVFFRkJ6QUNoajVtYVd4bE9pOHZMeTl6WVhBdGQybHVMVGd5
Tmk5RFpYSjBSVzV5YjJ4c0wzTmhjQzEzYVc0dE9ESTJYMlZMWVhONUxWTjFZa05CS0RFcExtTnlkREFNQmdOVkhSTUJBZj
hFQWpBQU1BMEdDU3FHU0liM0RRRUJEUVVBQTRJQ0FRQ2tkVVIyRGhnaWV4VVcreTJyZ2FFNm9yV0JQeU14dmVIMklQdjBy
UEd6cWRnVUZjTkg4MTZZekRvckVuT0F2YlJMQjhCYW9IK1dzL2VFbEFReHFFNSs0N1ZnU2NJVUY0b05Id1hubmZsUjFYUm
9ZY0ZaL2ZCa0lXMm5mT0sxQzh5MnZEdFpHMVFFeXlWRC9jeHY3dWJnT2lKZk9ZU2NzSHY1REl0U3RGVUJjbHZnM3hyRmky
ekc1YWhibE13cUNHcnZnUEtPeFI5K21YR0QrZW9UaEJIRVA2YUpGM1p1NDFtVndULzRjYlNyNW0zYzc3ZGVFUTJDcFFQR0
w4NzRQaUh5OW9ta2pldjlGNXlvQnpJN3lwaGFseUVJZGJBU1UwVWlVRXJqYlMraG53MFJFclYxYlFRelFmUzdxaUtNQlpU
TTRwek92L1JvNmYrT2NCZjdjMTZYdEhyRWdsaS9hTmFnS28zNG5GaFVzY1FjVVRDaDNNc0NLdVZTWlUzZGJDZFNMSXZkb0
pJUzVGTFArZ3I4TGJRVzl1Ui9OZ3dKaFlyL3cwNms2QU9GK1RhSnc4ZWFrdjVFTERPdXpoaXBxQjYzQnVNU0NHRlpjVVEy
YkRoZGM1Z2NWOUdsTmdWRVhtVG9lZTNmbjg5T1FUQzdHckN3RnpOeHdBTTZnSjBNQVJ5V2w1SG1nci9wT2IxTVg1VmVoZ2
FvSHBqb3ZlTUFhY09OYnRpT3dmTVV5aFBkQ0ptbkxQNjdsb2t2R3E3UERKL0RVYmVzcEFxVm05MVRNNlFiV2pkYW5LR0I2
a0pZKzdINUVTSThzcC9uempIWGRaeGVJUE83bE9USXRLZFJXODJrUmNCUjlUTkRTUzZydDVzSTE2TFcwTkNKMnpwcll0OF
hyTk83MjhsanlBPT0iXX0

B.2.3 Pełna postać przykładowych danych komendy przesyłanych do urządzenia fiskalnego zakodowana w Base64URL:

eyJhdHRyaWJ1dGVzIjp7ImNwZFNlcnZpY2VOYW1lIjoiS0ZEIn19

B.2.4 Pełna postać przykładowych danych do podpisu komendy przesyłanych do urządzenia fiskalnego zakodowana w Base64URL:

eyJhbGciOiJSUzI1NiIsIng1YyI6WyJNSUlGSERDQ0F3U2dBd0lCQWdJVE9nQUFBQWptajFXQlhVNm1PZ0FCQUFBQUNEQU
5CZ2txaGtpRzl3MEJBUTBGQURBV01SUXdFZ1lEVlFRREV3dGxTMkZ6ZVMxVGRXSkRRVEFnRncweE56QTRNakl3TmpNek1U
TmFHQTh5TURreU1EY3hNREV5TWpjeU9Wb3dnY2d4Q3pBSkJnTlZCQVlUQWxCTU1SUXdFZ1lEVlFRSUV3dE5ZWHB2ZDJsbF
kydHBaVEVSTUE4R0ExVUVCeE1JVjJGeWMzcGhkMkV4SHpBZEJnTlZCQW9NRmsxcGJtbHpkR1Z5YzNSM2J5QkdhVzVoYm5Q
RHMzY3hJekFoQmdOVkJBc1RHa1JsY0dGeWRHRnRaVzUwSUVsdVptOXliV0YwZVhwaFkycHBNUjR3SEFZRFZRUURFeFYwWl
hOMExXVXRhMkZ6ZVM1dFppNW5iM111Y0d3eEtqQW9CZ2txaGtpRzl3MEJDUUVXRzJsdVptOHVaUzFrWld0c1lYSmhZMnBs

UUcxbUxtZHZkaTV3YkRDQ0FTSXdEUVlKS29aSWh2Y05BUUVCQlFBRGdnRVBBRENDQVFvQ2dnRUJBTXZ5VlhHajhZTmh5Nl
AyOGJLajlNbGVBNytRWEtDdFBKWjRNNk1JeGlhcUE0bG9kZDlObytXc2dSRVRWekVQaUI4cmFMOW4zdU0rUkJGd0syQTRW
dnVBV3VHWngyZHJrZk1abnBTVkZMT3NRbmFkQjFyakJDWTVHL3BNWDZlSTdCbHR4NFhGWUsvMWNZMVUrbUZWYzk0UnlmeX
h5eTBaV1NEOElHVjluMEFpbERwUmZJSkJPdTVhM29xdXo4WlpHdVd5VTk1S1dCS1JBRDdTVjJicFRsWVdYNFVIaFRlMzIz
SFRZTDNyRGJLUDczSEFveWxPYlNtU3ZtQjlNeU56V2dCZjczVU9IbXpYUHFwdVJiTEZuUitsbFRBMEZBOGtPeWx4dGlqeW
dNWHBJQ09haTdhdjJvZkd0NjV2MEdKZzV3bEp1cVd2a1FYRlV5b3lHR1VZYVFzQ0F3RUFBYU9CclRDQnFqQWRCZ05WSFE0
RUZnUVV4N3hLajFUWENvck9FeGEyaFkvamR6Nk5rQTB3SHdZRFZSMGpCQmd3Rm9BVUJiK1BhcnRkNlRWNFBWMWtUVXJ0Sm
Fkc1Nkb3dXZ1lJS3dZQkJRVUhBUUVFVGpCTU1Fb0dDQ3NHQVFVRkJ6QUNoajVtYVd4bE9pOHZMeTl6WVhBdGQybHVMVGd5
Tmk5RFpYSjBSVzV5YjJ4c0wzTmhjQzEzYVc0dE9ESTJYMlZMWVhONUxWTjFZa05CS0RFcExtTnlkREFNQmdOVkhSTUJBZj
hFQWpBQU1BMEdDU3FHU0liM0RRRUJEUVVBQTRJQ0FRQ2tkVVIyRGhnaWV4VVcreTJyZ2FFNm9yV0JQeU14dmVIMklQdjBy
UEd6cWRnVUZjTkg4MTZZekRvckVuT0F2YlJMQjhCYW9IK1duL2VFbEFReHFFNSs0N1ZnU2NJVUY0b05Id1hubmZsUjFYUm
9ZY0ZaL2ZCa0lXMm5mT0sxQzh5MnZIdFpHMVFFeXlWRC9jeHY3dWJnT2lKZk9ZU2NzSHY1REl0U3RGVUJjbHZnM3hyRmky
ekc1YWhibE13cUNHcnZnUEtPeFI5K21YR0QrZW9UaEJIRVA2YUpGM1p1NDFtVndULzRjYlNyNW0zYzc3ZGVFUTJDcFFQR0
w4NzRQaUh5OW9ta2pldjlGNXlvQnpJN3lwaGFseUVJZGJBU1UwVWlVRXJqYlMraG53MFJFclYxYlFRelFmUzdxaUtNQlpU
TTRwek92L1JvNmYrT2NCZjdjMTZYdEhyRWdsaS9hTmFnS28zNG5GaFVzY1FjVVRDaDNNc0NLdVZTWlUzZGJDZFNMSXZkb0
pJUzVGTFArZ3I4TGJRVzl1Ui9OZ3dKaFlyL3cwNms2QU9GK1RhSnc4ZWFrdjVFTERPdXpoaXBxQjYzQnVNU0NHRlpjVVEy
YkRoZGM1Z2NWOUdsTmdWRVhtVG9lZTNmbjg5T1FUQzdHckN3RnpOeHdBTTZnSjBNQVJ5V2w1SG1nci9wT2IxTVg1VmVoZ2
FvSHBqb3ZlTUFhY09OYnRpT3dmTVV5aFBkQ0ptbkxQNjdsb2t2R3E3UERKL0RVYmVzcEFxVm05MVRNNlFiV2pkYW5LR0I2
a0pZKzdINUVTSThzcC9uempIWGRaeGVJUE83bE9USXRLZFJXODJrUmNCUjlUTkRTUzZydDVzSTE2TFcwTkNKMnpwcll0OF
hyTk83MjhsanlBPT0iXX0.eyJhdHRyaWJ1dGVzIjp7ImNwZFNlcnZpY2VOYW1lIjoiS0ZEIn19

B.2.5 Pełna postać podpisu przykładowej komendy przesyłanej do urządzenia fiskalnego:

HlKhuau2-ZLYoBip8ed2J7Js5HHOr1b96vlh-udv0OUjm5wWc0wSPCTqDe4rYe9RlqeG8Z4xXbPJE287o01bYtqE-
VuYZL5sDNHvLi1RQBZqTTHpWCNOw3mfM6vrKX_EXJ1wNeGu8aavozYKfVxIHWNZfQz6Ff1dCiAkxxXu35dEyBYGIZHbEz3
4AJu8KOY-
O24ZV6qr2tERB_SUYOgS4ZgigvY9loCDb_Vuui2sKYTMW55bT_BBL0gdm8yp7M2RNNcHitrGgEsen5otmpsgzh-
hQAP7rZYZw05l0gLU8xAaN_2RMIaz1vN1uAM4o8o7mxe-FFv1RLSP1zoMU-NGHw

B.2.6 Pełna postać obiektu JWS przykładowej komendy przesyłanej do urządzenia fiskalnego:

eyJhbGciOiJSUzI1NiIsIng1YyI6WyJNSUlGSERDQ0F3U2dBd0lCQWdJVE9nQUFBQWptajFXQlhVNm1PZ0FCQUFBQUNEQU
5CZ2txaGtpRzl3MEJBUTBGQURBV01SUXdFZ1lEVlFREV3dGxTMkZ6ZVMxVGRXSkRRVEFnRncweE56QTRNakl3TmpNek1U
TmFHQTh5TURreU1EY3hNREV5TWpjeU9Wb3dnY2d4Q3pBSkJnTlZCQVlUQWxCTU1SUXdFZ1lEVlFSUV3dE5ZWHB2ZDJsbF
kydHBaVEVSTUE4R0ExVUVCeE1JVjJGeWMzcGhkMkV4SHpBZEJnTlZCQW9NRmsxcGJtbHpkR1Z5YzNSM2J5QkdhVzVoYm5Q
RHMzY3hJekFoQmdOVkJBc1RHa1JsY0dGeWRHRnRaVzUwSUVsdVptOXliV0YwZVhwaFkycHBNUjR3SEFZRFZRUURFeFYwWl
hOMExXVXRhMkZ6ZVM1dFppNW5iM1l1Y0d3eEtqQW9CZ2txaGtpRzl3MEJDUUVXRzJsdVptOHVaUzFrWld0c1lYSmhZMnBs
UUcxbUxtZHZkaTV3YkRDQ0FTSXdEUVlKS29aSWh2Y05BUUVCQlFBRGdnRVBBRENDQVFvQ2dnRUJBTXZ5VlhHajhZTmh5Nl
AyOGJLajlNbGVBNytRWEtDdFBKWjRNNk1JeGlhcUE0bG9kZDlObytXc2dSRVRWekVQaUI4cmFMOW4zdU0rUkJGd0syQTRW
dnVBV3VHWngyZHJrZk1abnBTVkZMT3NRbmFkQjFyakJDWTVHL3BNWDZlSTdCbHR4NFhGWUsvMWNZMVUrbUZWYzk0UnlmeX
h5eTBaV1NEOElHVjluMEFpbERwUmZJSkJPdTVhM29xdXo4WlpHdVd5VTk1S1dCS1JBRDdTVjJicFRsWVdYNFVIaFRlMzIz
SFRZTDNyRGJLUDczSEFveWxPYlNtU3ZtQjlNeU56V2dCZjczVU9IbXpYUHFwdVJiTEZuUitsbFRBMEZBOGtPeWx4dGlqeW
dNWHBJQ09haTdhdjJvZkd0NjV2MEdKZzV3bEp1cVd2a1FYRlV5b3lHR1VZYVFzQ0F3RUFBYU9CclRDQnFqQWRCZ05WSFE0
RUZnUVV4N3hLajFUWENvck9FeGEyaFkvamR6Nk5rQTB3SHdZRFZSMGpCQmd3Rm9BVUJiK1BhcnRkNlRWNFBWMWtUVXJ0Sm
Fkc1Nkb3dXZ1lJS3dZQkJRVUhBUUVFVGpCTU1Fb0dDQ3NHQVFVRkJ6QUNoajVtYVd4bE9pOHZMeTl6WVhBdGQybHVMVGd5
Tmk5RFpYSjBSVzV5YjJ4c0wzTmhjQzEzYVc0dE9ESTJYMlZMWVhONUxWTjFZa05CS0RFcExtTnlkREFNQmdOVkhSTUJBZj
hFQWpBQU1BMEdDU3FHU0liM0RRRUJEUVVBQTRJQ0FRQ2tkVVIyRGhnaWV4VVcreTJyZ2FFNm9yV0JQeU14dmVIMklQdjBy
UEd6cWRnVUZjTkg4MTZZekRvckVuT0F2YlJMQjhCYW9IK1duL2VFbEFReHFFNSs0N1ZnU2NJVUY0b05Id1hubmZsUjFYUm
9ZY0ZaL2ZCa0lXMm5mT0sxQzh5MnZIdFpHMVFFeXlWRC9jeHY3dWJnT2lKZk9ZU2NzSHY1REl0U3RGVUJjbHZnM3hyRmky
ekc1YWhibE13cUNHcnZnUEtPeFI5K21YR0QrZW9UaEJIRVA2YUpGM1p1NDFtVndULzRjYlNyNW0zYzc3ZGVFUTJDcFFQR0
w4NzRQaUh5OW9ta2pldjlGNXlvQnpJN3lwaGFseUVJZGJBU1UwVWlVRXJqYlMraG53MFJFclYxYlFRelFmUzdxaUtNQlpU
TTRwek92L1JvNmYrT2NCZjdjMTZYdEhyRWdsaS9hTmFnS28zNG5GaFVzY1FjVVRDaDNNc0NLdVZTWlUzZGJDZFNMSXZkb0
pJUzVGTFArZ3I4TGJRVzl1Ui9OZ3dKaFlyL3cwNms2QU9GK1RhSnc4ZWFrdjVFTERPdXpoaXBxQjYzQnVNU0NHRlpjVVEy
YkRoZGM1Z2NWOUdsTmdWRVhtVG9lZTNmbjg5T1FUQzdHckN3RnpOeHdBTTZnSjBNQVJ5V2w1SG1nci9wT2IxTVg1VmVoZ2
FvSHBqb3ZlTUFhY09OYnRpT3dmTVV5aFBkQ0ptbkxQNjdsb2t2R3E3UERKL0RVYmVzcEFxVm05MVRNNlFiV2pkYW5LR0I2
a0pZKzdINUVTSThzcC9uempIWGRaeGVJUE83bE9USXRLZFJXODJrUmNCUjlUTkRTUzZydDVzSTE2TFcwTkNKMnpwcll0OF
hyTk83MjhsanlBPT0iXX0.eyJhdHRyaWJ1dGVzIjp7ImNwZFNlcnZpY2VOYW1lIjoiS0ZEIn19.HlKhuau2-
ZLYoBip8ed2J7Js5HHOr1b96vlh-udv0OUjm5wWc0wSPCTqDe4rYe9RlqeG8Z4xXbPJE287o01bYtqE-
VuYZL5sDNHvLi1RQBZqTTHpWCNOw3mfM6vrKX_EXJ1wNeGu8aavozYKfVxIHWNZfQz6Ff1dCiAkxxXu35dEyBYGIZHbEz3
4AJu8KOY-
O24ZV6qr2tERB_SUYOgS4ZgigvY9loCDb_Vuui2sKYTMW55bT_BBL0gdm8yp7M2RNNcHitrGgEsen5otmpsgzh-
hQAP7rZYZw05l0gLU8xAaN_2RMIaz1vN1uAM4o8o7mxe-FFv1RLSP1zoMU-NGHw

B.3 Przykładowe dane procesu szyfrowania komendy w środowisku testowym

B.3.1 Pełna postać chronionego nagłówka obiektu JWE komend przesyłanych do kasy:

{"enc":"A128CBC-HS256","alg":"RSA1_5","kid":"cc82d4ae40f68aa342c42d59ac6ed404,CN=MF e-Kasy"}

B.3.2 Pełna postać chronionego nagłówka obiektu JWE komend przesyłanych do urządzenia fiskalnego zakodowana w Base64URL:

eyJlbmMiOiJBMTI4Q0JDLUhTMjU2IiwiYWxnIjoiUlNBMV81Iiwia2lkIjoiY2M4MmQ0YWU0MGY2OGFhMzQyYzQyZDU5YWM2ZWQ0MDQsQ049TUYgZS1LYXN5In0

B.3.3 Wartości przykładowych danych użytych do szyfrowania zakodowanych szesnastkowo:

JWE_AES_CEK => 852cde285e375dac45ff7c44ee6d12e306b4e7086a2e0f3e0dbc1e3e1e3a0e68

JWE_MAC_KEY => 852cde285e375dac45ff7c44ee6d12e3
JWE_AES_KEY => 06b4e7086a2e0f3e0dbc1e3e1e3a0e68

JWE_AES_IV => 9641366ce173224c452a914e6de088c3

B.3.4 Zaszyfrowana wartość przykładowego klucza algorytmu szyfrującego algorytmem asymetrycznym RSA z wykorzystaniem klucza publicznego kasy zakodowana w Base64URL:

hsileeNCtbhjLR60diiBZ7U6kFPqzU3Pc6DDneWQncob1sZTEv7bsqlAv-
QDmR3liTRWuOzcYgduuDwAbmz1xwFI3cVObVeQBiRAgBtFuX_xEpmBdXAzXrMXY37dX1SoCu0Rno0HnBs5bDsnuX4TiZe4
jEKNggOhpf88mErGW26BwU1WbvFlyj9HhEfI2H46D-y29djnkXI_7MpM994hTRTq6JGPE-
9m2ZjOtU6Yw4Ty1kRmUeK9iWQkdIKJHy6TGHD9qnh03Odcyu_DPHJnbmhhB5AaT5yAD9kGcitcMC4PUDju3_qcHgiwPo94
s4m8j1RBo7M8jldVJvvOpU28ng

B.3.5 Przykładowa wartość wektora inicjującego zakodowana w Base64URL:

lkE2bOFzIkxFKpFObeCIww

B.3.6 Przykładowa wartość zaszyfrowanych danych zakodowana w Base64URL:

PoikvBdxUJgfLYAr9tyYOediKO7oQTdEUAPcW6QSKcmPCu1e3TKtk-ypKVXtMrJvJd0uRweKpGvx_uyU-
IEENg_Dh4o1Q8A7PQ5D2of_qZCW9pXi-DCnSE6wtPAqgUCYKbeXWdJj8S-jzDrx6z-
nu9ie25f4pw5EnRsA7pGLfk0MIao6oaEDcrmVm646tf9ezCKwACZX3HyOLpN0Q6PtovXPriGV07V7bQ-
MCSsj0AoxKMdjv1YNUGblU-WxAPaSki-
7L54YgkMR3ob9eRUZNAHIzs2jNTAPCuOML2zycO4Qz0saa5h4Q8zuG9WOOaAKJI0CQqHWh3FRqMSOINoXkbay5QVzrZ-
Kbbyf6LusspxI5aibMNPBucrXkodWWszFc3th_jPQ4r6YZB-
RzAO_inRNQqFsTJneP0OJjkd_0SfkanIcU5sjxsbBH9G6Vry5ECtOr5Me3IPcPIXGcdGqBO3TaklpXCHj1-
qOVWeDbZY5xxcQfhhSvYfOtM5ei4yIkNrXIcWTagie-
kmlBnF2mtjm0iv1YCuca8a44RcD5BdpaGOr1lEeizjAPBYi8QWtXSU5KDNXeC0gN4zUNPuccuzPzPk1QaQ7Mt8h9rgzOB9
cDIoOVLFe_hqAq4fn4xG25QKiTVLcOd8PKB8oA4S_f_4nO6cO7zPx_DP4Tu9cBE4M7DJxjZY2d4A21w-
HGx4YukTwaX_w9bcfhKSYNkFYKky6aal80Z4JaMW902NHhleG0ER6c3Y7JFVCyWw9CXIJKfT6BVY8q5CdVFrW4_CF38-
AIuXKRU-
HizqOjM8TrOErY0hOu5443u7KQedT4pyZWyAzVai4MK1Kj5jLUyLOZ4inHPgqTdT3AaTFX5FDwOM_ivUjbwoptwk7MC8eu
oHTXQj02yB4p-
TpdGsx4qO9BHUg3fgFPNbrSFDWn80KNV8C6VmOBaTNZ8MTLkSB3lQhMEC3mYCeB3fMiy_E79NmePsttGKCcCPotwFv4-
qW-cGp11-T_Z5P8_EqN6ss04O0w4PlAq7Q0mJdmpaf0KqnT-
gF7KcwYG5Vq3_5DJCXBu0DSigydQyBUl5E4OUH8X02sSNGetmiLuAMkrOV6dysu0bJVlu81OeyN1bQv7cN_qiL1DQBt_gs
quhspxI8h1lMHyxQ2ygIrDm-UsMWdjl20jBIy7zlFOVHJxbdna_fZNldA1Tkg1L-
vuHdJprEqcBpnjoOKe8wWpGwalreKi4DHOf3nOm_JEfpdxhO7DfPpG6fx-
krxkkF2_ybyg773NSATjBBOeRbZo1GrI8Ic7fdHIpFYyxhcCU9zLRJlWg0pjzHJIxoX-
prwhW90BHqjFYpnOynMEvs6SXCES4OPFS2mXw8PFr51PkRkYfkSewraTnWwnw3WFqfiOBjCKeDpfbF5axbhT-rg5BTHUi-
VvPT3MrPInRy-p0BijKLB9y8Z0R8DLX07ZoWAjJOS55XWdZ8hdPWCoC5O9odTRcUUEvi-
uqWQJMQCmZCiRwhsU9vO51T2t-aaWhLL9Q-e58M_pbM0HfLU_ea8jf5Vaf_bXht18BgZi2dvt-vddwY09kVB-
jQkzJmvLe2_hcWlrKQ2X7-
EmDlLt7iad7Nn20861a2lDKdQjbVDXbXZVF2cp9jdtg_Ls5rMHgcTU2ORJG4jzcHCJi4qRVUsa9t4znWEbtgLS64T64qSc
xqgcNL8gRzFwLxCCG_6UqFbUgqIkHfWjlzw572rA_WJGB0UVBPiRxltRzHKC2bhc85KTKsvO0qIHVk1DCrF47rTZ-
Dx54IzGhsW_BRc5azlcshoNaABAUw2V_hS-

```
DoMsLLpli5qfiRBw_WrhBbZjSFUn1AIL12CeeJss9pZ9ZNe2aPWjd75vTknxURE9mT8-jh1v3DjTs8_eBQWBT9-
BdEy0AEPfKTZgZ4QHxLj_v3VLr3F99Fu2dj977LJ1SiOnXMJlZLhypPrTnwjFuPavNe8HmsvjgysyMgHt_Lu_bx4OQk0P-
MrfwbOXsSjjJivyjS-
P5g2p194L5sGonvh2Jrl5Q_aBHgKIiyakYxEY3zbYcJfS1RoV5MSg2P_MJYoJI1Ffm_AAqUcRBK1dYQLhTIilQzjOb5_TZ
8_3za2pjH7xLWC91egX6d8tFqpyX1XaLchQUObnKJgqubHejatfBZU1BM82BmyigS8PseeDPC2Cj6tg34BKra7q3I20H5X
QrDFbyho7HOlY55jb5UvLsSmHdIAgoFOh7iSnki4iYrkQrPVCWlvbb2KvhNHotCeYFsdxfLa43y21STuYBoNP5Buu-
cz4jovMwQyMZSgU7ojx8ae2RkrGzJxD7p943U9w21mTNJDDBpIdUg_QaPEi9PaxH-7mB72waZfDtNEB5S-
MBQpIdIazD6QB4skD3b2h49rXggBYjyT59yxPrHkhDy_bo352BMqRxokfjZ1KdirVHS4cLy0SXty06E3TCUt5_tFJ-0bo-
kT-Dj4wR7k04Yzd6thPKi6AQJKPZz1ARZEguGu2-9p46AVZd1mCWDTCNlEDsL_gvb7GJFIknZt215voxI6WMjzyLio-
jyGGMBHgoQSsd246LQY-LNJCKJa8a18MbNqpRRBZBdmPnvWUoK8eTQ6NkT3mZQNQcrqvrRtrQ_OECAScwcDoSct4d6C-
_w-
I77E_FNVEvdnVDbWMKJMc9_F99qfBTYPL9AL_Cx89SyoJOrBWiWLClGoPjw91MV3K5fxX_1EXapNyrfWW3jo9OjeVKr42h
eWacLTFMCCdNXYMuwVT8NttBRpkJwee-
dae0vqomtHQQa1_gyqc01J95pAbATYkjAGKYfnImS6c4fEP2VD6EAro7WQzmM3IzEvFF9dWMD5RnuwMKB2HECCxvSKkc03
fQC_xMu0EgliqkE_QyT2aJpPTyf7pB-
zSy2C6gYt5DVlnc8ea6i1ElPSwdaS9uzMVHE0Hqrfq6Px23uZXcBVYcNiSXCguUaAyLmOf6YsXK7hHHx_uoVLVlB3SDME4
GDI8bp0PisW_LTViALvjktsrm0021PqCu58mHqEP9h0jM-s582vcw9orSdi63q2OvK6dBojawORWZ_t-MWTc7r0-
y_766bekLrktyVHJkfbUFM4keDOEP9I4MO9mKHnCLGCmFok5zlJDxtn71M_HXXpwJJ4hrCJIUOT2ZO0glIRGE1YhMqILfh
DJNELhh1ytsqSy32RTOz1qbIog1TilpQOwxbGPUVazwcXTXOFtHrd6O74gpGHQkhWurIcYxLcm4rD8gja97iLm2J2keD5b
HbgCryyjB8BotGZTA_oj1VihMe9P6k69QwUCOo-MgQSVYgq6IXS3EbfdOBbWObjRtytaY46YXY9-4OREk6BlHes8Xn5bG-
O7w0xJ4OKyKAEs6iyRyFktr_RP6rBMOfpOLz90G68hX8ZwOVGWUIE8e8J5gB3XbpPYbX5lWi2kx-
7CYXRDdbYUi36l338tHks-aXcZ8Hb7h-D8bGxObateUSBR3Ka-
6gCDMYHdSYijxHhhBxTTyrAcgzrLKD7CvV4PwQ5cUVx1qE7VeosUMIS9BotMXQmr12C0k1ulAHW8JspE43Mpced0NuUUaX
ib9EVuNKxK1DjLXwtv8OMsugQ4_bOWQMMCpz-
l5o5ciPGht0YkDAZJoODYMDpmZvMJLYb3QBzBbt2kgaG8Si0a_9NZrkpTLOuRM
```

B.3.7a Przykładowa wartość dodatkowych danych uwierzytelniających - nagłówek JWE:

```
eyJlbmMiOiJBMTI4Q0JDLUhTMjU2IiwiYWxnIjoiUlNBMV81Iiwia2lkIjoiY2M4MmQ0YWU0MGY2OGFhMzQyYzQyZDU5YW
M2ZWQ0MDQsQ049TUYgZS1LYXN5In0
```

odzwierciedlenie AAD w postaci szesnastkowej:

```
65794a6c626d4d694f694a424d54493451304a444c5568544d6a5532496977695957786e496a6f69556c4e424d5638
314969776961326c6b496a6f6959324d344d6d5130595755304d4759324f4746684d7a5179597a51795a4455355957
4d325a5751304d4451735130343954555967 5a53314c59584e35496e30
```

B.3.7b Przykład wyliczenia reprezentacji wartości długości dodatkowych danych uwierzytelniających zakodowana w Base64URL:

```
JWE_AAD_URL BYTES LENGTH -> 123

JWE_AAD_URL BITS LENGTH -> 123 * 8 = 984

JWE_AT = [0, 0, 0, 0, 0, 0, 3, 216]
```

odzwierciedlenie w postaci szesnastkowej:

```
00000000000003d8
```

B.3.8a Przykładowa wartość etykiety uwierzytelniającej zakodowana w Base64URL:

```
ct35FpBFiOdWQ7aw8Jk45A
```

odzwierciedlenie w postaci szesnastkowej:

```
72ddf916904588e75643b6b0f09938e4
```

B.3.8b Przykład wyliczenia wartości etykiety uwierzytelniającej przykładowych danych:

Postać szesnastkowa odszyfrowanego 32 bajtowego klucza CEK (Content Encryption Key) B.3.3:

```
852cde285e375dac45ff7c44ee6d12e306b4e7086a2e0f3e0dbc1e3e1e3a0e68
```

pierwsze 16 bajtów wykorzystywane jako klucz w funkcji HMAC (JWE_MAC_KEY):

postać szesnastkowa wykorzystana w funkcji HMAC:
852cde285e375dac45ff7c44ee6d12e3

użyte dane autoryzujące AAD (nagłówek JWE) B.3.7a:

postać ASCII:
{"enc":"A128CBC-HS256","alg":"RSA1_5","kid":"cc82d4ae40f68aa342c42d59ac6ed404,CN=MF e-Kasy"}

postać Base64URL:
eyJlbmMiOiJBMTI4Q0JDLUhTMjU2IiwiYWxnIjoiUlNBMV81Iiwia2lkIjoiY2M4MmQ0YWU0MGY2OGFhMzQyYzQyZDU5YWM2ZWQ0MDQsQ049TUYgZS1LYXN5In0

postać szesnastkowa (bajty postaci Base64URL) wykorzystana w funkcji HMAC:
65794a6c626d4d694f694a424d54493451304a444c5568544d6a5532496977695957786e496a6f69556c4e424d5638314969776961326c6b496a6f6959324d344d6d5130595755304d4759324f4746684d7a5179597a51795a4455355957 4d325a5751304d4451735130343954555967 5a53314c59584e35496e30

użyty 16 bajtowy wektor inicjujący IV B.3.5:

postać Base64URL:
lkE2bOFzIkxFKpFObeCIww

postać szesnastkowa wykorzystana w funkcji HMAC:
9641366ce173224c452a914e6de088c3

zaszyfrowane dane użyte do wyliczenia etykiety uwierzytelniającej B.3.6:

postać Base64URL:
PoikvBdxUJgfLYAr9tyYOediKO7oQTdEUAPcW6QSKcmPCu1e3TKtk-ypKVXtMrJvJd0uRweKpGvx_uyU-IEENg_Dh4o1Q8A7PQ5D2of_qZCW9pXi-DCnSE6wtPAqgUCYKbeXWdJj8S-jzDrx6z-nu9ie25f4pw5EnRsA7pGLfk0MIao6oaEDcrmVm646tf9ezCKwACZX3HyOLpN0Q6PtovXPriGV07V7bQ-MCSsj0AoxKMdjv1YNUGblU-WxAPaSki-7L54YgkMR3ob9eRUZNAHIzs2jNTAPCuOML2zycO4Qz0saa5h4Q8zuG9WOOaAKJI0CQqHWh3FRqMSOINoXkbay5QVzrZ-Kbbyf6LusspxI5aibMNPBucrXkodWWszFc3th_jPQ4r6YZB-RzAO_inRNQqFsTJneP0OJjkd_0SfkanIcU5sjxsbBH9G6Vry5ECtOr5Me3IPcPIXGcdGqBO3TaklpXCHj1-qOVWeDbZY5xxcQfhhSvYfOtM5ei4yIkNrXIcWTagie-kmlBnF2mtjm0iv1YCuca8a44RcD5BdpaGOr1lEeizjAPBYi8QWtXSU5KDNXeC0gN4zUNPuccuzPzPk1QaQ7Mt8h9rgzOB9cDIoOVLFe_hqAq4fn4xG25QKiTVLcOd8PKB8oA4S_f_4nO6cO7zPx_DP4Tu9cBE4M7DJxjZY2d4A21w-HGx4YukTwaX_w9bcfhKSYNkFYKky6aal80Z4JaMW902NHhleG0ER6c3Y7JFVCyWw9CXIJKfT6BVY8q5CdVFrW4_CF38-AIuXKRU-HizqOjM8TrOErY0hOu5443u7KQedT4pyZWyAzVai4MK1Kj5jLUyLOZ4inHPgqTdT3AaTFX5FDwOM_ivUjbwoptwk7MC8euoHTXQj02yB4p-TpdGsx4qO9BHUg3fgFPNbrSFDWn80KNV8C6VmOBaTNZ8MTLkSB3lQhMEC3mYCeB3fMiy_E79NmePsttGKCcCPotwFv4-qW-cGp11-T_Z5P8_EqN6ss04O0w4PlAq7Q0mJdmpaf0KqnT-gF7KcwYG5Vq3_5DJCXBu0DSigydQyBUl5E4OUH8X02sSNGetmiLuAMkrOV6dysu0bJVlu81OeyN1bQv7cN_qiL1DQBt_gsquhspxI8h1lMHyxQ2ygIrDm-UsMWdjl20jBIy7zlFOVHJxbdna_fZNldA1Tkg1L-vuHdJprEqcBpnjoOKe8wWpGwalreKi4DHOf3nOm_JEfpdxhO7DfPpG6fx-krxkkF2_ybyg773NSATjBBOeRbZo1GrI8Ic7fdHIpFYyxhcCU9zLRJlWg0pjzHJIxoX-prwhW90BHqjFYpnOynMEvs6SXCES4OPFS2mXw8PFr51PkRkYfkSewraTnWwnw3WFqfiOBjCKeDpfbF5axbhT-rg5BTHUi-VvPT3MrPInRy-p0BijKLB9y8Z0R8DLX07ZoWAjJOS55XWdZ8hdPWCoC5O9odTRcUUEvi-uqWQJMQCmZCiRwhsU9vO51T2t-aaWhLL9Q-e58M_pbM0HfLU_ea8jf5Vaf_bXht18BgZi2dvt-vddwY09kVB-jQkzJmvLe2_hcWlrKQ2X7-EmDlLt7iad7Nn20861a2lDKdQjbVDXbXZVF2cp9jdtg_Ls5rMHgcTU2ORJG4jzcHCJi4qRVUsa9t4znWEbtgLS64T64qScxqgcNL8gRzFwLxCCG_6UqFbUgqIkHfWjlzw572rA_WJGB0UVBPiRxltRzHKC2bhc85KTKsvO0qIHVk1DCrF47rTZ-Dx54IzGhsW_BRc5azlcshoNaABAUw2V_hS-DoMsLLpli5qfiRBw_WrhBbZjSFUn1AIL12CeeJss9pZ9ZNe2aPWjd75vTknxURE9mT8-jh1v3DjTs8_eBQWBT9-BdEy0AEPfKTZgZ4QHxLj_v3VLr3F99Fu2dj977LJ1SiOnXMJlZLhypPrTnwjFuPavNe8HmsvjgysyMgHt_Lu_bx4OQk0P-MrfwbOXsSjjJivyjS-P5g2p194L5sGonvh2Jrl5Q_aBHgKIiyakYxEY3zbYcJfS1RoV5MSg2P_MJYoJI1Ffm_AAqUcRBK1dYQLhTIilQzjOb5_TZ8_3za2pjH7xLWC91egX6d8tFqpyX1XaLchQUObnKJgqubHejatfBZU1BM82BmyigS8PseeDPC2Cj6tg34BKra7q3I20H5XQrDFbyho7HOlY55jb5UvLsSmHdIAgoFOh7iSnki4iYrkQrPVCWlvbb2KvhNHotCeYFsdxfLa43y21STuYBoNP5Buu-cz4jovMwQyMZSgU7ojx8ae2RkrGzJxD7p943U9w21mTNJDDBpIdUg_QaPEi9PaxH-7mB72waZfDtNEB5S-MBQpIdIazD6QB4skD3b2h49rXggBYjyT59yxPrHkhDy_bo352BMqRxokfjZ1KdirVHS4cLy0SXty06E3TCUt5_tFJ-0bokT-Dj4wR7k04Yzd6thPKi6AQJKPZz1ARZEguGu2-9p46AVZd1mCWDTCNlEDsL_gvb7GJFIknZt215voxI6WMjzyLiojyGGMBHgoQSsd246LQY-LNJCKJa8a18MbNqpRRBZBdmPnvWUoK8eTQ6NkT3mZQNQcrqvrRtrQ_OECAScwcDoSct4d6C-_w-

I77E_FNVEvdnVDbWMKJMc9_F99qfBTYPL9AL_Cx89SyoJOrBWiWLClGoPjw91MV3K5fxX_1EXapNyrfWW3jo9OjeVKr42h
eWacLTFMCCdNXYMuwVT8NttBRpkJwee-
dae0vqomtHQQa1_gyqc01J95pAbATYkjAGKYfnImS6c4fEP2VD6EAro7WQzmM3IzEvFF9dWMD5RnuwMKB2HECCxvSKkc03
fQC_xMu0Eg1iqkE_QyT2aJpPTyf7pB-
zSy2C6gYt5DVlnc8ea6i1ElPSwdaS9uzMVHE0Hqrfq6Px23uZXcBVYcNiSXCguUaAyLmOf6YsXK7hHHx_uoVLVlB3SDME4
GDI8bp0PisW_LTViALvjktsrm0021PqCu58mHqEP9h0jM-s582vcw9orSdi63q2OvK6dBojawORWZ_t-MWTc7r0-
y_766bekLrktyVHJkfbUFM4keDOEP9I4MO9mKHnCLGCmFok5zlJDxtn71M_HXXpwJJ4hrCJIUOT2ZO0glIRGE1YhMqILfh
DJNELhh1ytsqSy32RTOz1qbIog1TilpQOwxbGPUVazwcXTXOFtHrd6O74gpGHQkhWurIcYxLcm4rD8gja97iLm2J2keD5b
HbgCryyjB8BotGZTA_oj1VihMe9P6k69QwUCOo-MgQSVYgq6IXS3EbfdOBbWObjRtytaY46YXY9-4OREk6BlHes8Xn5bG-
O7w0xJ4OKyKAEs6iyRyFktr_RP6rBMOfpOLz90G68hX8ZwOVGWUIE8e8J5gB3XbpPYbX5lWi2kx-
7CYXRDdbYUi361338tHks-aXcZ8Hb7h-D8bGxObateUSBR3Ka-
6gCDMYHdSYijxHhhBxTTyrAcgzrLKD7CvV4PwQ5cUVx1qE7VeosUMIS9BotMXQmr12C0k1ulAHW8JspE43Mpced0NuUUaX
ib9EVuNKxK1DjLXwtv8OMsugQ4_bOWQMMCpz-
l5o5ciPGht0YkDAZJoODYMDpmZvMJLYb3QBzBbt2kgaG8Si0a_9NZrkpTLOuRM

postać szesnastkowa wykorzystana w funkcji HMAC:
**3e88a4bc177150981f2d802bf6dc9839e76228eee84137445003dc5ba41229c98f0aed5edd32ad93eca92955ed32b2
6f25dd2e47078aa46bf1feec94f88104360fc3878a3543c03b3d0e43da87ffa99096f695e2f830a7484eb0b4f02a81
409829b79759d263f12fa3cc3af1eb3fa7bbd89edb97f8a70e449d1b00ee918b7e4d0c21aa3aa1a10372b9959bae3a
b5ff5ecc22b0002657dc7c8e2e937443a3eda2f5cfae2195d3b57b6d0f8c092b23d00a3128c763bf560d5066e553e5
b100f692922fbb2f9e18824311de86fd7915193401c8cecda335300f0ae38c2f6cf270ee10cf4b1a6b987843ccee1b
d58e39a00a248d0242a1d6877151a8c48e20da1791b6b2e50573ad9f8a6dbc9fe8bbacb29c48e5a89b30d3c1b9cad7
9287565accc5737b61fe33d0e2be98641f91cc03bf8a744d42a16c4c99de3f43898e477fd127e46a721c539b23c6c6
c11fd1ba56bcb9102b4eaf931edc83dc3c85c671d1aa04edd36a49695c21e3d7ea8e5567836d9639c717107e1852bd
87ceb4ce5e8b8c8890dad721c5936a089efa49a50671769ad8e6d22bf5602b9c6bc6b8e11703e417696863abd6511e
8b38c03c1622f105ad5d2539283357782d20378cd434fb9c72eccfccf93541a43b32df21f6b833381f5c0c8a0e54b1
5efe1a80ab87e7e311b6e502a24d52dc39df0f281f280384bf7ffe273ba70eef33f1fc33f84eef5c044e0cec32718d
9636778036d70f871b1e18ba44f0697ff0f5b71f84a4983641582a4cba69a97cd19e0968c5bdd36347865786d0447a
73763b245542c96c3d09720929f4fa05563cab909d545ad6e3f085dfcf8022e5ca454f878b3a8e8ccf13ace12b6348
4ebb9e38deeeca41e753e29c995b203355a8b830ad4a8f98cb5322ce6788a71cf82a4dd4f701a4c55f9143c0e33f8a
f5236f0a29b7093b302f1eba81d35d08f4db2078a7e4e9746b31e2a3bd047520ddf8053cd6eb4850d69fcd0a355f02
e9598e05a4cd67c3132e4481de54213040b799809e0777cc8b2fc4efd36678fb2db462827023e8b7016fe3ea96f9c1
a9d75f93fd9e4ff3f12a37ab2cd383b4c383e502aed0d2625d9a969fd0aaa74fe805eca730606e55ab7ff90c909706
ed034a2832750c81525e44e0e507f17d36b123467ad9a22ee00c92b395e9dcacbb46c9565bbcd4e7b23756d0bfb70d
fea88bd43401b7f82caae86ca7123c87594c1f2c50db2808ac39be52c316763976d23048cbbce514e5472716dd9daf
df64d95d0354e48352febee1dd269ac4a9c0699e3a0e29ef305a91b06a5ade2a2e031ce7f79ce9bf2447e977184eec
37cfa46e9fc7e92bc64905dbfc9bca0efbdcd4804e304139e45b668d46ac8f0873b7dd1c8a45632c6170253dccb449
956834a63cc7248c685fea6bc215bdd011ea8c56299ceca7304bece925c2112e0e3c54b6997c3c3c5af9d4f9119187
e449ec2b6939d6c27c37585a9f88e06308a783a5f6c5e5ac5b853fab8390531d48be56f3d3dccacf227472fa9d018a
328b07dcbc67447c0cb5f4ed9a1602324e4b9e5759d67c85d3d60a80b93bda1d4d1714504be2faea964093100a6642
891c21b14f6f3b9d53dadf9a69684b2fd43e7b9f0cfe96ccd077cb53f79af237f955a7ff6d786dd7c060662d9dbedf
af75dc18d3d91507e8d0933266bcb7b6fe171696b290d97efe1260e52edee269decd9f6d3ceb56b694329d4236d50d
76d7655176729f6376d83f2ece6b30781c4d4d8e4491b88f37070898b8a91554b1af6de339d611bb602d2eb84fae2a
49cc6a81c34bf204731702f10821bfe94a856d482a2241df5a3973c39ef6ac0fd624607451504f891c65b51cc7282d
9b85cf392932acbced2a207564d430ab178eeb4d9f83c79e08cc686c5bf0517396b395cb21a0d680040530d95fe14b
e0e832c2cba658b9a9f891070fd6ae105b663485527d4020bd7609e789b2cf6967d64d7b668f5a377be6f4e49f1511
13d993f3e8e1d6fdc38d3b3cfde0505814fdf81744cb40043df293660678407c4b8ffbf754baf717df45bb6763f7be
cb2754a23a75cc26564b872a4fad39f08c5b8f6af35ef079acbe3832b323201edfcbbbf6f1e0e424d0ff8cadfc1b39
7b128e3262bf28d2f8fe60da9d7de0be6c1a89ef87626b97943f6811e02888b26a4631118df36d87097d2d51a15e4c
4a0d8ffcc258a0923515f9bf000a9471104ad5d6102e14c88a54338ce6f9fd367cff7cdada98c7ef12d60bdd5e817e
9df2d16aa725f55da2dc85050e6e728982ab9b1de8dab5f05953504cf36066ca2812f0fb1e7833c2d828fab60df804
aadaeeadc8db41f95d0ac315bca1a3b1ce958e798dbe54bcbb12987748020a053a1ee24a7922e2262b910acf5425a5
bdb6f62af84d1e8b4279816c7717cb6b8df2db5493b9806834fe41baef9ccf88e8bccc10c8c652814ee88f1f1a7b64
64ac6cc9c43ee9f78dd4f70db59933490c306921d520fd068f122f4f6b11feee607bdb06997c3b4d101e52f8c050a4
87486b30fa401e2c903ddbda1e3dad78200588f24f9f72c4fac79210f2fdba37e7604ca91c6891f8d9d4a762ad51d2
e1c2f2d125edcb4e84dd3094b79fed149fb46e8fa44fe0e3e3047b934e18cddead84f2a2e8040928f673d40459120b
86bb6fbda78e805597759825834c2365103b0bfe0bdbec62452249d9b76d79be8c48e96323cf22e2a3e8f21863011e
0a104ac776e3a2d063e2cd2422896bc6b5f0c6cdaa945105905d98f9ef594a0af1e4d0e8d913de665035072baafad1
b6b43f38408049cc1c0e849cb7877a0beff0f88efb13f14d544bdd9d50db58c28931cf7f17df6a7c14d83cbf402ff0
b1f3d4b2a093ab0568962c2946a0f8f0f75315dcae5fc57ff51176a9372adf596de3a3d3a37952abe3685e59a70b4c
530209d35760cbb0553f0db6d051a6427079ef9d69ed2faa89ad1d041ad7f832a9cd3527de6901b0136248c018a61f
9c8992e9ce1f10fd950fa100ae8ed643398cdc8cc4bc517d756303e519eec0c281d871020b1bd22a4734ddf402ff13
2ed048358aa904fd0c93d9a2693d3c9fee907ecd2cb60ba818b790d596773c79aea2d4494f4b075a4bdbb33151c4d0
7aab7eae8fc76dee65770155870d8925c282e51a0322e639fe98b172bb8471f1feea152d5941dd20cc13818323c6e9
d0f8ac5bf2d356200bbe392db2b9b4d36d4fa82bb9f261ea10ff61d2333eb39f36bdcc3da2b49d8badead8ebcae9d0
688dac0e45667fb7e3164dceebd3ecbfefae9b7a42eb92dc951c991f6d414ce247833843fd23830ef662879c22c60a
6168939ce5243c6d9fbd4cfc75d7a70249e21ac224850e4f664ed20948446135621 32a20b7e10c93442e1875cadb2a
4b2df64533b3d6a6c8a20d538a5a503b0c5b18f5156b3c1c5d35ce16d1eb77a3bbe20a461d09215aeac8718c4b726e
2b0fc8236bdee22e6d89da4783e5b1db802af2ca307c068b4665303fa23d558a131ef4fea4ebd4305023a8f8c81049
5620aba2174b711b7dd3816d639b8d1b72b5a638e985d8f7ee0e44493a0651deb3c5e7e5b1be3bbc34c49e0e2b2280
12cea2c91c8592daff44feab04c39fa4e2f3f741baf215fc67039519650813c7bc279801dd76e93d86d7e655a2da4c**

7eec261744375b6148b7ea5df7f2d1e4b3e697719f076fb87e0fc6c6c4e6dab5e512051dca6beea00833181dd4988a3c478610714d3cab01c833acb283ec2bd5e0fc10e5c515c75a84ed57a8b143084bd068b4c5d09abd760b4935ba50075bc26ca44e3732971e77436e51469789bf4456e34ac4ad438cb5f0b6ff0e32cba0438fdb39640c302a73fa5e68e5c88f1a1b746240c0649a0e0d8303a6666f3092d86f7401cc16edda481a1bc4a2d1affd359ae4a532ceb913

długość dodatkowych danych autoryzujących w formacie Base64URL wynosi 123 bajty, czyli 984 bity i w formacie Big-Endian w postaci szesnastkowej ma wartość B.3.7b:

00000000000003d8

Dokonując konkatenacji bajtowej AAD, IV oraz danych zaszyfrowanych i długości AAD do wyliczenia MAC będzie wykorzystany ciąg bajtów w postaci szesnastkowej:

65794a6c626d4d694f694a424d54493451304a444c5568544d6a5532496977695957786e496a6f69556c4e424d5638
314969776961326c6b496a6f6959324d344d6d5130595755304d4759324f4746684d7a5179597a51795a4455355957
4d325a5751304d4451735130343954555967 5a53314c59584e35496e309641366ce173224c452a914e6de088c33e88
a4bc177150981f2d802bf6dc9839e76228eee84137445003dc5ba41229c98f0aed5edd32ad93eca92955ed32b26f25
dd2e47078aa46bf1feec94f88104360fc3878a3543c03b3d0e43da87ffa99096f695e2f830a7484eb0b4f02a814098
29b79759d263f12fa3cc3af1eb3fa7bbd89edb97f8a70e449d1b00ee918b7e4d0c21aa3aa1a10372b9959bae3ab5ff
5ecc22b0002657dc7c8e2e937443a3eda2f5cfae2195d3b57b6d0f8c092b23d00a3128c763bf560d5066e553e5b100
f692922fbb2f9e18824311de86fd7915193401c8cecda335300f0ae38c2f6cf270ee10cf4b1a6b987843ccee1bd58e
39a00a248d0242a1d6877151a8c48e20da1791b6b2e50573ad9f8a6dbc9fe8bbacb29c48e5a89b30d3c1b9cad79287
565accc5737b61fe33d0e2be98641f91cc03bf8a744d42a16c4c99de3f43898e477fd127e46a721c539b23c6c6c11f
d1ba56bcb9102b4eaf931edc83dc3c85c671d1aa04edd36a49695c21e3d7ea8e5567836d9639c717107e1852bd87ce
b4ce5e8b8c8890dad721c5936a089efa49a50671769ad8e6d22bf5602b9c6bc6b8e11703e417696863abd6511e8b38
c03c1622f105ad5d2539283357782d20378cd434fb9c72eccfccf93541a43b32df21f6b833381f5c0c8a0e54b15efe
1a80ab87e7e311b6e502a24d52dc39df0f281f280384bf7ffe273ba70eef33f1fc33f84eef5c044e0cec32718d9636
778036d70f871b1e18ba44f0697ff0f5b71f84a4983641582a4cba69a97cd19e0968c5bdd36347865786d0447a7376
3b245542c96c3d09720929f4fa05563cab909d545ad6e3f085dfcf8022e5ca454f878b3a8e8ccf13ace12b63484ebb
9e38deeeca41e753e29c995b203355a8b830ad4a8f98cb5322ce6788a71cf82a4dd4f701a4c55f9143c0e33f8af523
6f0a29b7093b302f1eba81d35d08f4db2078a7e4e9746b31e2a3bd047520ddf8053cd6eb4850d69fcd0a355f02e959
8e05a4cd67c3132e4481de54213040b799809e0777cc8b2fc4efd36678fb2db462827023e8b7016fe3ea96f9c1a9d7
5f93fd9e4ff3f12a37ab2cd383b4c383e502aed0d2625d9a969fd0aaa74fe805eca730606e55ab7ff90c909706ed03
4a2832750c81525e44e0e507f17d36b123467ad9a22ee00c92b395e9dcacbb46c9565bbcd4e7b23756d0bfb70dfea8
8bd43401b7f82caae86ca7123c87594c1f2c50db2808ac39be52c316763976d23048cbbce514e5472716dd9dafdf64
d95d0354e48352febee1dd269ac4a9c0699e3a0e29ef305a91b06a5ade2a2e031ce7f79ce9bf2447e977184eec37cf
a46e9fc7e92bc64905dbfc9bca0efbdcd4804e304139e45b668d46ac8f0873b7dd1c8a45632c6170253dccb4499568
34a63cc7248c685fea6bc215bdd011ea8c56299ceca7304bece925c2112e0e3c54b6997c3c3c5af9d4f9119187e449
ec2b6939d6c27c37585a9f88e06308a783a5f6c5e5ac5b853fab8390531d48be56f3d3dccacf227472fa9d018a328b
07dcbc67447c0cb5f4ed9a1602324e4b9e5759d67c85d3d60a80b93bda1d4d1714504be2faea964093100a6642891c
21b14f6f3b9d53dadf9a69684b2fd43e7b9f0cfe96ccd077cb53f79af237f955a7ff6d786dd7c060662d9dbedfaf75
dc18d3d91507e8d0933266bcb7b6fe171696b290d97efe1260e52edee269decd9f6d3ceb56b694329d4236d50d76d7
655176729f6376d83f2ece6b30781c4d4d8e4491b88f37070898b8a91554b1af6de339d611bb602d2eb84fae2a49cc
6a81c34bf204731702f10821bfe94a856d482a2241df5a3973c39ef6ac0fd624607451504f891c65b51cc7282d9b85
cf392932acbced2a207564d430ab178eeb4d9f83c79e08cc686c5bf0517396b395cb21a0d680040530d95fe14be0e8
32c2cba658b9a9f891070fd6ae105b663485527d4020bd7609e789b2cf6967d64d7b668f5a377be6f4e49f151113d9
93f3e8e1d6fdc38d3b3cfde0505814fdf81744cb40043df293660678407c4b8ffbf754baf717df45bb6763f7becb27
54a23a75cc26564b872a4fad39f08c5b8f6af35ef079acbe3832b323201edfcbbbf6f1e0e424d0ff8cadfc1b397b12
8e3262bf28d2f8fe60da9d7de0be6c1a89ef87626b97943f6811e02888b26a4631118df36d87097d2d51a15e4c4a0d
8ffcc258a0923515f9bf000a9471104ad5d6102e14c88a54338ce6f9fd367cff7cdada98c7ef12d60bdd5e817e9df2
d16aa725f55da2dc85050e6e728982ab9b1de8dab5f05953504cf36066ca2812f0fb1e7833c2d828fab60df804aada
eeadc8db41f95d0ac315bca1a3b1ce958e798dbe54bcbb12987748020a053a1ee24a7922e2262b910acf5425a5bdb6
f62af84d1e8b4279816c7717cb6b8df2db5493b9806834fe41baef9ccf88e8bccc10c8c652814ee88f1f1a7b6464ac
6cc9c43ee9f78dd4f70db59933490c306921d520fd068f122f4f6b11feee607bdb06997c3b4d101e52f8c050a48748
6b30fa401e2c903ddbda1e3dad78200588f24f9f72c4fac79210f2fdba37e7604ca91c6891f8d9d4a762ad51d2e1c2
f2d125edcb4e84dd3094b79fed149fb46e8fa44fe0e3e3047b934e18cddead84f2a2e8040928f673d40459120b86bb
6fbda78e8055977598258 34c2365103b0bfe0bdbec62452249d9b76d79be8c48e96323cf22e2a3e8f21863011e0a10
4ac776e3a2d063e2cd2422896bc6b5f0c6cdaa945105905d98f9ef594a0af1e4d0e8d913de665035072baafad1b6b4
3f38408049cc1c0e849cb7877a0beff0f88efb13f14d544bdd9d50db58c28931cf7f17df6a7c14d83cbf402ff0b1f3
d4b2a093ab0568962c2946a0f8f0f75315dcae5fc57ff51176a9372adf596de3a3d3a37952abe3685e59a70b4c5302
09d35760cbb0553f0db6d051a6427079ef9d69ed2faa89ad1d041ad7f832a9cd3527de6901b0136248c018a61f9c89
92e9ce1f10fd950fa100ae8ed643398cdc8cc4bc517d756303e519eec0c281d871020b1bd22a4734ddf402ff132ed0
48358aa904fd0c93d9a2693d3c9fee907ecd2cb60ba818b790d596773c79aea2d4494f4b075a4bdbb33151c4d07aab
7eae8fc76dee65770155870d8925c282e51a0322e639fe98b172bb8471f1feea152d5941dd20cc13818323c6e9d0f8
ac5bf2d356200bbe392db2b9b4d36d4fa82bb9f261ea10ff61d2333eb39f36bdcc3da2b49d8badead8ebcae9d0688d
ac0e45667fb7e3164dceebd3ecbfefae9b7a42eb92dc951c991f6d414ce247833843fd23830ef662879c22c60a6168
939ce5243c6d9fbd4cfc75d7a70249e21ac224850e4f664ed2094844613562132a20b7e10c93442e1875cadb2a4b2d
f64533b3d6a6c8a20d538a5a503b0c5b18f5156b3c1c5d35ce16d1eb77a3bbe20a461d09215aeac8718c4b726e2b0f
c8236bdee22e6d89da4783e5b1db802af2ca307c068b4665303fa23d558a131ef4fea4ebd4305023a8f8c810495620

aba2174b711b7dd3816d639b8d1b72b5a638e985d8f7ee0e44493a0651deb3c5e7e5b1be3bbc34c49e0e2b228012ce
a2c91c8592daff44feab04c39fa4e2f3f741baf215fc67039519650813c7bc279801dd76e93d86d7e655a2da4c7eec
261744375b6148b7ea5df7f2d1e4b3e697719f076fb87e0fc6c6c4e6dab5e512051dca6beea00833181dd4988a3c47
8610714d3cab01c833acb283ec2bd5e0fc10e5c515c75a84ed57a8b143084bd068b4c5d09abd760b4935ba50075bc2
6ca44e3732971e77436e51469789bf4456e34ac4ad438cb5f0b6ff0e32cba0438fdb39640c302a73fa5e68e5c88f1a
1b746240c0649a0e0d8303a6666f3092d86f7401cc16edda481a1bc4a2d1affd359ae4a532ceb91300000000000003
d8

Wynik użycia powyższego ciągu bajtów oraz klucza JWE_MAC_KEY 852cde285e375dac45ff7c44ee6d12e3
w funkcji haszującej HS256 zwraca 32 bajtową wartość w postaci szesnastkowej:

72ddf916904588e75643b6b0f09938e4cef03e71f0cac0a8e9cd1ac9dd9db985

wydzielając pierwsze 16 bajtów otrzymanego wyniku:

72ddf916904588e75643b6b0f09938e4

po przekodowaniu do formatu Base64URL:

ct35FpBFiOdWQ7aw8Jk45A

otrzymujemy wyliczoną etykietę uwierzytelniającą identyczną z B.3.8a.

B.3.9 Pełna postać obiektu JWE przykładowej komendy przesyłanej do urządzenia fiskalnego:

eyJlbmMiOiJBMTI4Q0JDLUhTMjU2IiwiYWxnIjoiUlNBMV81Iiwia2lkIjoiY2M4MmQ0YWU0MGY2OGFhMzQyYzQyZDU5YW
M2ZWQ0MDQsQ049TUYgZS1LYXN5In0.hsileeNCtbhjLR60diiBZ7U6kFPqzU3Pc6DDneWQncob1sZTEv7bsqlAv-
QDmR3liTRWuOzcYgduuDwAbmz1xwFI3cVObVeQBiRAgBtFuX_xEpmBdXAzXrMXY37dX1SoCu0Rno0HnBs5bDsnuX4TiZe4
jEKNggOhpf88mErGW26BwU1WbvFlyj9HhEfI2H46D-y29djnkXI_7MpM994hTRTq6JGPE-
9m2ZjOtU6Yw4Ty1kRmUeK9iWQkdIKJHy6TGHD9qnh03Odcyu_DPHJnbmhhB5AaT5yAD9kGcitcMC4PUDju3_qcHgiwPo94
s4m8j1RBo7M8jldVJvvOpU28ng.lkE2bOFzIkxFKpFObeCIww.PoikvBdxUJgfLYAr9tyYOediKO7oQTdEUAPcW6QSKcmP
Cu1e3TKtk-ypKVXtMrJvJd0uRweKpGvx_uyU-IEENg_Dh4o1Q8A7PQ5D2of_qZCW9pXi-
DCnSE6wtPAqgUCYKbeXWdJj8S-jzDrx6z-
nu9ie25f4pw5EnRsA7pGLfk0MIao6oaEDcrmVm646tf9ezCKwACZX3HyOLpN0Q6PtovXPriGV07V7bQ-
MCSsj0AoxKMdjv1YNUGblU-WxAPaSki-
7L54YgkMR3ob9eRUZNAHIzs2jNTAPCuOML2zycO4Qz0saa5h4Q8zuG9WOOaAKJI0CQqHWh3FRqMSOINoXkbay5QVzrZ-
Kbbyf6LusspxI5aibMNPBucrXkodWWszFc3th_jPQ4r6YZB-
RzAO_inRNQqFsTJneP0OJjkd_0SfkanIcU5sjxsbBH9G6Vry5ECtOr5Me3IPcPIXGcdGqBO3TaklpXCHj1-
qOVWeDbZY5xxcQfhhSvYfOtM5ei4yIkNrXIcWTagie-
kmlBnF2mtjm0iv1YCuca8a44RcD5BdpaGOr1lEeizjAPBYi8QWtXSU5KDNXeC0gN4zUNPuccuzPzPk1QaQ7Mt8h9rgzOB9
cDIoOVLFe_hqAq4fn4xG25QKiTVLcOd8PKB8oA4S_f_4nO6cO7zPx_DP4Tu9cBE4M7DJxjZY2d4A21w-
HGx4YukTwaX_w9bcfhKSYNkFYKky6aal80Z4JaMW902NHhleG0ER6c3Y7JFVCyWw9CXIJKfT6BVY8q5CdVFrW4_CF38-
AIuXKRU-
HizqOjM8TrOErY0hOu5443u7KQedT4pyZWyAzVai4MK1Kj5jLUyLOZ4inHPgqTdT3AaTFX5FDwOM_ivUjbwoptwk7MC8eu
oHTXQj02yB4p-
TpdGsx4qO9BHUg3fgFPNbrSFDWn80KNV8C6VmOBaTNZ8MTLkSB3lQhMEC3mYCeB3fMiy_E79NmePsttGKCcCPotwFv4-
qW-cGp11-T_Z5P8_EqN6ss04O0w4PlAq7Q0mJdmpaf0KqnT-
gF7KcwYG5Vq3_5DJCXBu0DSigydQyBUl5E4OUH8X02sSNGetmiLuAMkrOV6dysu0bJVlu81OeyN1bQv7cN_qiL1DQBt_gs
quhspxI8h1lMHyxQ2ygIrDm-UsMWdjl20jBIy7zlFOVHJxbdna_fZNldA1Tkg1L-
vuHdJprEqcBpnjoOKe8wWpGwalreKi4DHOf3nOm_JEfpdxhO7DfPpG6fx-
krxkkF2_ybyg773NSATjBBOeRbZo1GrI8Ic7fdHIpFYyxhcCU9zLRJlWg0pjzHJIxoX-
prwhW90BHqjFYpnOynMEvs6SXCES4OPFS2mXw8PFr51PkRkYfkSewraTnWwnw3WFqfiOBjCKeDpfbF5axbhT-rg5BTHUi-
VvPT3MrPInRy-p0BijKLB9y8Z0R8DLX07ZoWAjJOS55XWdZ8hdPWCoC5O9odTRcUUEvi-
uqWQJMQCmZCiRwhsU9vO51T2t-aaWhLL9Q-e58M_pbM0HfLU_ea8jf5Vaf_bXht18BgZi2dvt-vddwY09kVB-
jQkzJmvLe2_hcWlrKQ2X7-
EmDlLt7iad7Nn20861a2lDKdQjbVDXbXZVF2cp9jdtg_Ls5rMHgcTU2ORJG4jzcHCJi4qRVUsa9t4znWEbtgLS64T64qSc
xqgcNL8gRzFwLxCCG_6UqFbUgqIkHfWjlzw572rA_WJGB0UVBPiRxltRzHKC2bhc85KTKsvO0qIHVk1DCrF47rTZ-
Dx54IzGhsW_BRc5azlcshoNaABAUw2V_hS-
DoMsLLpli5qfiRBw_WrhBbZjSFUn1AIL12CeeJss9pZ9ZNe2aPWjd75vTknxURE9mT8-jh1v3DjTs8_eBQWBT9-
BdEy0AEPfKTZgZ4QHxLj_v3VLr3F99Fu2dj977LJ1SiOnXMJlZLhypPrTnwjFuPavNe8HmsvjgysyMgHt_Lu_bx4OQk0P-
MrfwbOXsSjjJivyjS-
P5g2p194L5sGonvh2Jrl5Q_aBHgKIiyakYxEY3zbYcJfS1RoV5MSg2P_MJYoJI1Ffm_AAqUcRBK1dYQLhTIilQzjOb5_TZ
8_3za2pjH7xLWC91egX6d8tFqpyX1XaLchQUObnKJgqubHejatfBZU1BM82BmyigS8PseeDPC2Cj6tg34BKra7q3I20H5X
QrDFbyho7HOlY55jb5UvLsSmHdIAgoFOh7iSnki4iYrkQrPVCWlvbb2KvhNHotCeYFsdxfLa43y21STuYBoNP5Buu-
cz4jovMwQyMZSgU7ojx8ae2RkrGzJxD7p943U9w21mTNJDDBpIdUg_QaPEi9PaxH-7mB72waZfDtNEB5S-
MBQpIdIazD6QB4skD3b2h49rXggBYjyT59yxPrHkhDy_bo352BMqRxokfjZ1KdirVHS4cLy0SXty06E3TCUt5_tFJ-0bo-
kT-Dj4wR7k04Yzd6thPKi6AQJKPZz1ARZEguGu2-9p46AVZd1mCWDTCNlEDsL_gvb7GJFIknZt215voxI6WMjzyLio-
jyGGMBHgoQSsd246LQY-LNJCKJa8a18MbNqpRRBZBdmPnvWUoK8eTQ6NkT3mZQNQcrqvrRtrQ_OECAScwcDoSct4d6C-
_w-
I77E_FNVEvdnVDbWMKJMc9_F99qfBTYPL9AL_Cx89SyoJOrBWiWLClGoPjw91MV3K5fxX_1EXapNyrfWW3jo9OjeVKr42h

eWacLTFMCCdNXYMuwVT8NttBRpkJwee-
dae0vqomtHQQa1_gyqc01J95pAbATYkjAGKYfnImS6c4fEP2VD6EAro7WQzmM3IzEvFF9dWMD5RnuwMKB2HECCxvSKkc03
fQC_xMu0Eg1iqkE_QyT2aJpPTyf7pB-
zSy2C6gYt5DVlnc8ea6i1ElPSwdaS9uzMVHE0Hqrfq6Px23uZXcBVYcNiSXCguUaAyLmOf6YsXK7hHHx_uoVLVlB3SDME4
GDI8bp0PisW_LTViALvjktsrm0021PqCu58mHqEP9h0jM-s582vcw9orSdi63q2OvK6dBojawORWZ_t-MWTc7r0-
y_766bekLrktyVHJkfbUFM4keDOEP9I4MO9mKHnCLGCmFok5zlJDxtn71M_HXXpwJJ4hrCJIUOT2ZO0glIRGE1YhMqILfh
DJNELhh1ytsqSy32RTOz1qbIog1TilpQOwxbGPUVazwcXTXOFtHrd6O74gpGHQkhWurIcYxLcm4rD8gja97iLm2J2keD5b
HbgCryyjB8BotGZTA_oj1VihMe9P6k69QwUCOo-MgQSVYgq6IXS3EbfdOBbWObjRtytaY46YXY9-4OREk6BlHes8Xn5bG-
O7w0xJ4OKyKAEs6iyRyFktr_RP6rBMOfpOLz90G68hX8ZwOVGWUIE8e8J5gB3XbpPYbX5lWi2kx-
7CYXRDdbYUi36l338tHks-aXcZ8Hb7h-D8bGxObateUSBR3Ka-
6gCDMYHdSYijxHhhBxTTyrAcgzrLKD7CvV4PwQ5cUVx1qE7VeosUMIS9BotMXQmr12C0k1ulAHW8JspE43Mpced0NuUUaX
ib9EVuNKxK1DjLXwtv8OMsugQ4_bOWQMMCpz-
l5o5ciPGht0YkDAZJoODYMDpmZvMJLYb3QBzBbt2kgaG8Si0a_9NZrkpTLOuRM.ct35FpBFiOdWQ7aw8Jk45A

B.4 Przykładowe dane procesu podpisywania danych w środowisku testowym

B.4.1 Pełna postać chronionego nagłówka podpisu danych przesyłanych do repozytorium:


```
{"jpkcertificate":"MIICfjCCAeegAwIBAgIQzILUrkD2iqNCxC1ZrG7UBDANBgkqhkiG9w0BAQsFADAUMRIwEAYDVQQ
DEwlNRiBlLUthc3kwHhcNMTcxMTE1MDk0MDAzWhcNMTkxMTE1MDk0MDAyWjAYMRYwFAYDVQQDEw1aVEUxNzAxMDAwOTAxM
IIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAwy3Kc3oTipU45lOIX6K3rruFY83vMqYJqwoWzrRVmZn85cNHSwo
a\/f96AW0\/akADbJ3uo7U8oWhTF\/rj8xIds4uimrN1YiPFmbXAMaeRQDbBa\/qvI5SRQtK9Bmse7KyspIFXVfEWP17Ok
DiBEZ\/n+NC6ERzzKkxA3gMRQFGpHUSQ2+EOi7kykPGi1f8Yh\/2czd+FBvyrp8oSjyX951DdCsqG+rIwlz9p8PeoFwUgg
whb2tMl48U3nD9gZGXLuGOMBZ3nJ9U3fHVdi7XCpvn0PqvTSZLNL45yqXETu6bAZWB5Ab4q5EVvI4unrjnJqc3fPD2OLXp
INIssg6uqyTVCZQIDAQABo0kwRzBFBgNVHQEEPjA8gBDGD6f6PWMBTV\/bee5Lrlc9oRYwFDESMBAGA1UEAxMJTUYgZS1L
YXN5ghCzwtV5n24\/mUCWe9d7xth6MA0GCSqGSIb3DQEBCwUAA4GBAKRTZFPJY5ObY4VVGpJLl4Xb2JNtWpNXdwPs3N8I2
rliGc0dxyq8R4C9X125G0GLgXXXTMDtnE1k+xmCk0aU6bj2xpfezLhW6i1+mmHTB\/2+JhsKp5oRRTXg8SpH5G1vwQI9ek
9B\/bYvn72nKrUaTp3PZsmCNqm1D0VayfTRhZiS","alg":"RS256","jpkmetadata":"eyJjb3JyZWxhdGlvbklkIjoi
VEZELlpURTEyMzQ1Njc4OTAuMjAxOC0wMS0wMVQwMTowMDowMC4wMDBaIn0="}
```


B.4.2 Pełna postać chronionego nagłówka podpisu danych przesyłanych do repozytorium zakodowana w Base64URL:


```
eyJqcGtjZXJ0aWZpY2F0ZSI6Ik1JSUNmakNDQWVlZ0F3SUJBZ0lReklMVXJrRDJpcU5DeEMxWnJHN1VCREFOQmdrcWhraU
c5dzBCQVFzRkFEQVVNUkl3RUFZRFZRUURFd2xOUmlCbExVdGhjM2t3SGhjTk1UY3hNVEUxTURrME1EQXpXaGNOTVRreE1U
RTFNRGswTURBeVdqQVlNUll3RkFZRFZRUURFdzFhVkVVeE56QXhNREF3T1RBeE1JSUJakFOQmdrcWhraUc5dzBCQVFFRk
FBT0NBUThBTUlJQkNnS0NBUUVBd3kzS2Mzb1RpcFU0NWxPSVg2SzNycnVGWTgzdk1xWUpxd29XenJSVm1abjg1Y05IU3dv
YVwvZjk2QVcwXC9ha0FEYkozdW83VThvV2hURlwvcmo4eElkczR1aW1yTjFZaVBGbWJYQU1hZVJRRGJCYVwvcXZJNVNSUX
RLOUJtc2U3S3lzcElGWFZmRVdQMTdPa0RpQkVaXC9uK05DNkVSenpLa3hBM2dNUlFGR3BIVVNRMitFT2k3a3lrUEdpMWY4
WWhcLzJjemQrRkJ2eXJwOG9TanlYOTUxRGRDc3FHK3JJd2x6OXA4UGVvRndVZ2d3aGIydE1sNDhVM25EOWdaR1hMdUdPTU
JaM25KOVUzZkhWZGk3WENwdm4wUHF2VFNaTE5MNDV5cVhFVHU2YkFaV0I1QWI0cTVFVnZJNHVucmpuSnFjM2ZQRDJPTFhw
SU5Jc3NnNnVxeVRWQ1pRSURBUUFCbzBrd1J6QkZCZ05WSFFFRVBqQThnQkRHRDZmNlBXTUJUVlwvYmVlNUxybGM5b1JZd0
ZERVNNQkFHQTFVRUF4TUpUVVlnWlMxTFlYTjVnaEN6d3RWNW4yNFwvbVVDV2U5ZDd4dGg2TUEwR0NTcUdTSWIzRFFFQkN3
VUFBNEdCQUtSVFpGUEpZNU9iWTRWVkdwSkxsNFhiMkpOdFdwTlhkd1BzM044STJybGlHYzBkeHlxOFI0QzlYMTI1RzBHTG
dYWFhUTUR0bkUxayt4bUNrMGFVNmJqMnhwZmV6TGhXNmkxK21tSFRCXC8yK0poc0twNW9SUlRYZzhTcEg1RzF2d1FJOWVr
OUJcL2JZdm43Mm5LclVhVHAzUFpzbUNOcW0xRDBWYXlmVFJoWmlTIiwiYWxnIjoiUlMyNTYiLCJqcGttZXRhZGF0YSI6Im
V5SmpiM0p5Wld4aGRHbHZia2xrSWpvaVZFWkVMbHBVUlRFeU16UTFOamM0T1RBdU1qQXhPQzB3TVMwd01WUXdNVG93TURv
d01DNHdNREJhSW4wPSJ9
```


B.4.3 Przykładowa postać nieskompresowanych danych przesyłanych do repozytorium:


```
{
    "JPK": {
        "naglowek": {
            "wersja": "JPK_KASA_v0-92",
            "dataJPK": "2018-03-30T17:25:26.027Z"
        },
        "podmiot1": {
            "nazwaPod": "Zażółć gęślą jaźń",
            "nrFabr": "ZTE-FAB-0123456789",
            "NIP": "1111111111",
            "adresPod": {
                "ulica": "Ulica",
                "miejsc": "Miejscowość",
                "nrLok": "NrLok",
                "poczta": "Poczta",
                "nrDomu": "NrDomu",
                "kodPoczt": "00-000"
            },
            "nrUnik": "ZTE1234567890",
            "nrEwid": "2018/123456789"
        },
        "content": [
            {
                "zdarzenie": {
                    "pamiecChr": 1,
                    "JPKID": 19,
                    "dataCzas": "2018-03-30T17:25:16.017Z",
                    "typ": {
                        "01": {
```

```
                            "info": "Awaryjne zerowanie pamięci operacyjnej"
                        }
                    }
                }
            }
        ]
    }
}
```

B.4.4 Przykładowa postać danych przesyłanych do repozytorium zakodowana w Base64URL:

eyJKUEsiOnsibmFnbG93ZWsiOnsid2Vyc2phIjoiSlBLX0tBU0FfdjAtOTIiLCJkYXRhSlBLIjoiMjAxOC0wMy0zMFQxNzoyNToyNi4wMjdaIn0sInBvZG1pb3QxIjp7Im5hendhUG9kIjoiWmHFvMOzxYLEhyBnxJnFm2zEhSBqYcW6xYQiLCJuckZhYnIiOiJaVEUtRkFCLTAxMjM0NTY3ODkiLCJOSVAiOiIxMTExMTExMTExIiwiYWRyZXNQb2QiOnsidWxpY2EiOiJVbGljYSIsIm1pZWpzYyI6Ik1pZWpzY293b8WbxIciLCJuckxvayI6Ik5yTG9rIiwicG9jenRhIjoiUG9jenRhIiwibnJEb211Ijoi TnJEb211Iiwia29kUG9jenQiOiIwMC0wMDAifSwibnJVbmlrIjoiWlRFMTIzNDU2Nzg5MCIsIm5yRXdpZCI6IjIwMTgvMT IzNDU2Nzg5In0sImNvbnRlbnQiOlt7InpkYXJ6ZW5pZSI6eyJwYW1pZWNDaHIiOjEsIkpQS0lEIjoxOSwiZGF0YUN6YXMi OiIyMDE4LTAzLTMwVDE3OjI1OjE2LjAxN1oiLCJ0eXAiOnsiMDEiOnsiaW5mbyI6IkF3YXJ5am5lIHplcm93YW5pZSBwYW 1pxJljaSBvcGVyYWN5am5laiJ9fX19XX19

B.4.5 Pełna postać obiektu JWS przykładowych danych przesyłanych do repozytorium:

eyJqcGtjZXJ0aWZpY2F0ZSI6Ik1JSUNmakNDQWVlZ0F3SUJBZ0lRekIMVXJrRDJpcU5DeEMxWnJHN1VCREFOQmdrcWhraUc5dzBCQVFzRkFEQVVNUkl3RUFZRFZRUURFd2xOUmlCbExVdGhjM2t3SGhjTk1UY3hNVEUxTURrME1EQXpXaGNOTVRreE1URTFNRGswTURBeVdqQVlNUll3RkFZRFZRUURFdzFhVkVVeE56QXhNREF3T1RBeE1JSUJakFOQmdrcWhraUc5dzBCQVFFRkFBT0NBUThBTUlJQkNnS0NBUUVBd3kzS2Mzb1RpcFU0NWxPSVg2SzNycnVGWTgzdk1xWUpxd29XenJSVm1abjg1Y05IU3dvYVwvZjk2QVcwXC9ha0FEYkozdW83VThvV2hURlwvcmo4eElkczR1aW1yTjFZaVBGbWJYQU1hZVJRRGJCYVwvcXZJNVNSUXRLOUJtc2U3S3lzcElGWFZmRVdQMTdPa0RpQkVaXC9uK05DNkVSenpLa3hBM2dNUlFGR3BIVVNRMitFT2k3a3lrUEdpMWY4WWhcLzJjemQrRkJ2eXJwOG9TanlYOTUxRGRDc3FHK3JJd2x6OXA4UGVvRndVZ2d3aGlydE1sNDhVM25EOWdaR1hMdUdPTUJaM25KOVUzZkhWZGk3WENwdm4wUHF2VFNaTE5MNDV5cVhFVHU2YkFaV0I1QWI0cTVFVnZJNHVucmpuSnFjM2ZQRDJPTFhwSU5Jc3NnNnVxeVRWQ1pRSURBUUFCbzBrd1J6QkZCZ05WSFFFRVBqQThnQkRHRDZmNlBXTUJUVlwvYmVlNUxybGM5b1JZd0ZERVNNQkFHQTFVRUF4TUpUVVlnWlMxTFlYTjVnaEN6d3RWNW4yNFwvbVVDV2U5ZDd4dGg2TUEwR0NTcUdTSWIzRFFFQkN3VUFBNEdCQUtSVFpGUEpZNU9iWTRWVkdwSkxsNFhiMkpOdFdwTlhkd1BzM044STJybGlHYzBkeHlxOFI0QzlYMTI1RzBHTGdYWFhUTUR0bkUxayt4bUNrMGFVNmJqMnhwZmV6TGhXNmkxK21tSFRCXC8yK0poc0twNW9SUlRYZzhTcEg1RzF2d1FJOWVrOUJcL2JZdm43Mm5LclVhVHAzUFpzbUNOcW0xRDBWYXlmVFJoWmlTliwiYWxnIjoiUlMyNTYiLCJqcGttZXRhZGF0YSI6ImV5SmpiM0p5Wld4aGRHbHZia2xrSWpvaVZFWkVMbHBVUlRFeU16UTFOamM0T1RBdU1qQXhPQzB3TVMwd01WUXdNVG93TURvd01DNHdNREJhSW4wPSJ9.eyJKUEsiOnsibmFnbG93ZWsiOnsid2Vyc2phIjoiSlBLX0tBU0FfdjAtOTIiLCJkYXRhSlBLIjoiMjAxOC0wMy0zMFQxNzoyNToyNi4wMjdaIn0sInBvZG1pb3QxIjp7Im5hendhUG9kIjoiWmHFvMOzxYLEhyBnxJnFm2zEhSBqYcW6xYQiLCJuckZhYnIiOiJaVEUtRkFCLTAxMjM0NTY3ODkiLCJOSVAiOiIxMTExMTExMTExIiwiYWRyZXNQb2QiOnsidWxpY2EiOiJVbGljYSIsIm1pZWpzYyI6Ik1pZWpzY293b8WbxIciLCJuckxvayI6Ik5yTG9rIiwicG9jenRhIjoiUG9jenRhIiwibnJEb211IjoiTnJEb211Iiwia29kUG9jenQiOiIwMC0wMDAifSwibnJVbmlrIjoiWlRFMTIzNDU2Nzg5MCIsIm5yRXdpZCI6IjIwMTgvMTIzNDU2Nzg5In0sImNvbnRlbnQiOlt7InpkYXJ6ZW5pZSI6eyJwYW1pZWNDaHIiOjEsIkpQS0lEIjoxOSwiZGF0YUN6YXMiOiIyMDE4LTAzLTMwVDE3OjI1OjE2LjAxN1oiLCJ0eXAiOnsiMDEiOnsiaW5mbyI6IkF3YXJ5am5laiJ9fX19XX19.upliPTXaTwHTJrtHMFJ5reFugWpY595RLwIruU9mto3XitQzTvgQqCL-VUm0-XobYzpjk7r3p45H2Dux0KuxEzwZqpE9RdU2bnY6QOCiBrmwdZGwEhXln6Hxr-rTOy2Wn24fVOry3MtqzMhtdjA9fdfyXTbbooh9Yp1ckN-YX2uoSlqB6BbkVfwopDLoPdSZFvGMaub4mWox3dSlgf7DXaajvemFvHARtkzzj0hlg28M2pJleJP77x6d5b3RPNw-ZqGFwaJZZtXqf_hnE4e1bIx-pwXIC2C1hgCY7vDXLCXYM9nePCevL8yZ0Wc1rjVsiCiqzoEyzyMyMS2VTIUZeQ

B.5 Przykładowe dane procesu szyfrowania danych w środowisku testowym

B.5.1 Pełna postać chronionego nagłówka obiektu JWE danych przesyłanych z kasy:

{"kid":"3A00000008E68F55815D4EA63A000100000008, CN=eKasy-SubCA","enc":"A128CBC-HS256","alg":"RSA1_5"}

B.5.2 Pełna postać chronionego nagłówka obiektu JWE danych przesyłanych z urządzenia fiskalnego zakodowana w Base64URL:

eyJraWQiOiIzQTAwMDAwMDA4RTY4RjU1ODE1RDRFQTYzQTAwMDEwMDAwMDAwOCwgQ049ZUthc3ktU3ViQ0EiLCJlbmMiOiJBMTI4Q0JDLUhTMjU2IiwiYWxnIjoiUlNBMV81In0

B.5.3 Wartości przykładowych danych użytych do szyfrowania:

```
JWE_AES_CEK => 0fae0202c8aa5d39bac1b9f58a9f440c4b700abdffd7e661ac4c609e288529be

JWE_MAC_KEY => 0fae0202c8aa5d39bac1b9f58a9f440c
JWE_AES_KEY => 4b700abdffd7e661ac4c609e288529be

JWE_AES_IV => c773654a97535031619a525f4285f3dd
```

B.5.4 Zaszyfrowana wartość przykładowego klucza algorytmu szyfrującego algorytmem asymetrycznym RSA z wykorzystaniem klucza publicznego ministerstwa zakodowana w Base64URL:

giC4C064EjKUdWmh0TmoQVuXbxByz5yWO53Hxda3Je1gRchNmnq0s38RXyJt9L1e3SDe5hnZuVtgPKufUBgEblItprryumGYnqyhuzIbD0m8akTq9JyJHQ7SERZlGYIzAgQbIn7NJAKswtzhpP56PxSnMRwegdx0PoW-ZlTx2dYSpHRobWYvpHjz4t25H_poYZh2nAmmzC4nWOGnlshNI0qxk21E64_Tb-4ACpoqvk6WlPGygUYT4M1hoAN4w_P0fsTyQxcmsXt3RpfiCbCROY2oLyQYGVYnTodbpLp7T4kVKN3XnzT6szBFzMCJsQ9B0Ug26mQARkH0cx7FWOtVCg

B.5.5 Przykładowa wartość wektora inicjującego zakodowana w Base64URL:

x3NlSpdTUDFhmlJfQoXz3Q

B.5.6 Przykładowa wartość zaszyfrowanych danych zakodowana w Base64URL:

jnYxL4MrKbWmgY5ZeUrL_etV-byYTeEwp8fH1j5q8ii54kYWQPUBErh3APg7KLl8ZjY644HdbcZgy168x0cACpVhh5MqA93K5CXl0e-4VUH97iBcHo_VdzUW_iqX3HPyfl1WLrt2PgIiqFOuzjapVYiVFzmXwhKsoWx4oYzIJfUJYM1QFcHnPVWPONlTnpDV6VRsskm3oC4sCn9DW0TJ-h8HHzvXT18n6F1SHLrqcVgfb4cpz5w6X_k8TBPXaJNgOvEgcbUTXNkKkUWs2xTKNdJAinKTzXR9WFsPajd6R5brwVIEpRgsHy6icr4RyxGO7fPS8JmKsbEY2CqedVO62ZtYREH1K2giji5lXPEpLPa7kFHdJ4uknjOjbgK-_Kbb9BUVdhdFO6-bNpT67fiReZ--JLyu-cAYVnoTMj0GqSXOjNLiPZbVG1Q3Q02yw875kPf0nrlPihkNoCGgdBgP66mvF0CdP-0hoZiMYKh9qOzeAtuiDb0kQL91mPexz_YZ1GQs-UrkYP85Fnv-vpvPkgBp9kJl9x1SiB4qvj86Ryaf-ic8P_FvOy9bPv9wRtGIaDTcJBuK0vaIb7W7FCYiBKyKv6oR0vsoXghHS8NrH7ARm0VIC-q9v_S8exfy5Rik6znN51Qwsdz5qG0b-rQU5Gs1hLFckc8Ra1wDMvyfv6qSE6-dbpmbIsw9rCh2tvNzwM2nN4cHMZQErfpa8-bTuGcRehZVwrjmfYkV9kcWqbSVajJfWAzeZLmjmcU3hytB6Ke7Yo2HyIjtQNqfWyWY4JmpgqSEVexk6MxeBQr35s2LKZ8HouO7jqqPV1RHh_oFIxNcaleisxh4nuEm3T8OfXU8UHvdjRHu2V_HjLZFFtuzT51iydzcHCUVK020mNqF92mMg3w3p6VTe6xBtteuwIfG5uORdfw9Onu1u3e-PVe7P3H27iuPvLdS-hG0ywN8aissvnDrIe2gKL_Na7Cuf_9XGv2EHXEEFqYJ67GUiDMYEK1M3JVM0QiQ0zFL1llPwyN9nHdXRyfneWo4931krUeUYh-yxAtmgeH2lkT67pQ6jwZrqXS9WVBpBdN5JzMS8RjJShZY2BEotJp5MqMdz-IfZwnx07JBjuMobskg_1yPmtCWQXoAb-MA574jiV40EEKq9HkS27T59wOS-XdOi32KJHs3a9b1pSgmZqiWSUpwLaOuuZYjV9H7_sz5i6w8oCByxrcOLcjY1T2cp0HIaHiM1LnHrDvmAN0Weh1rthK2I4Esy9yHfl1RjFwzoUmSz5sYdnx2jqKOEskAiaqc45IMTEdFIV8pWJcFsaxMx1CjUeumu_IGom0aiMDikvAyrs-BmFwIOVSRzIsyezngh HS6K5H3R9isuiTA64OkomnqKIcC2TFjmx97EAjMGG3CBqc7fzdPMDcCYRoTgrq8JF-gjYED4RBAZ8cZYBJx77neOpZ-Kdy_8h3zT70GmJzmzjshrqDnPV3CdT_T7JCPWPoE0cQk2oUIvKCnx3DsH6x_1tti6FkzfdqAcGj7VftQN5XAwVcmMBZQwhChFS945C1soRD1SLWqWfaxG0CBqu5rVfwODOZsV4WAxX0XjrE6CbDbeLuCrzcjTto3rcRnrsVAWb-fnC97FEsevL4Lfg7S7iE8YRCR1QW3M4OaI-46SHMBov2u92RdPvzLpYjIf3N29YmijYkFMcYnfVlt7-

```
U2Ou9YpNNhF5h2veEHzWUS07gyy-pSOtkjLXwp2MpKQucf3jyXJWyuArU4qVAaN-E-
WZkO1oNI_5zFb3W26AbEoumrhL8FpY8P5KGYAgH1B3uVNs2TQzdb3cAg4lIe9PBeeNtMi2-m505-
r1ljd0uLhmaZeidaCYdr4V50AiC3y4ggd6R4GE0u6QXttdCgwvRWAHUUBtzT4RONe4-KzcZVGK3Is2zGQL-
HrTii_oOgmDBpdTsnWTX8wFBITUw_tKf7YOZbwedLk8QVLYKji9ZdlPOek-w-
zRXaYDTQotleP4oRdMy4wmPn1mHsdF8RJ0UCQ8YrRF1Fz7ogp9C8ftU3YpGo9QxfKyaw46TsiGlQrEBCQ3r8gRNOTXHcCs
O4l-3iB05u4pDhZzz-7WbHltXGuDqFik7BWnq-
V0wRhX7KXei3ikGSsp5eCrigbHWsci_QdCmj5YjR5FV7Av29VqkZScCKZQ4_f4SHiYPuM7hurB56qLmpBlhkvxUZHyvp0u
PCa-vDgSn0FQ23dAVvmUI6epkad_-YwUtOthCJvfphZeiH8rvAOr1p1YM71EEkW-
u5xrCrdHtBy7NQpGCVYs_Cm6g_UIfguOrgOBmvgEKQlKapuLLi_rGwHsSNOlnMA9yx3sC4PGTeIAHca1wtOp1sFQwUxcgE
QRV8TY86G17qlrQIs-v9-
uDwSEI7aZpqGfZmZKaQOWCno23i4XbDQnnnjCdF1fmMxACorg5g_Wjcpgs91fPiePv9_1YYEnXRPduqhQbVb1J2txdOoSM
CrG9RZesYhV0KBvLOHBbhFBCD-
5IApXXIJ3It2aU2HJUtD77B60U2dOlZ6RzQ2OvmNgJ8oUSozKplTiYaDLKE3mgMO4o2IksZjvKISi69MmSJy059vQXkPBG
AJ3hDlHsCwrxCg43ZvbcV2T55466HhsBv8WAtTl01psYO0IBAp2QgbFfkZkYEaP5XgLmdtAGBJwPjSSgH4dM31MgbgRXyd
yo9qoEFUJj3-bzYMGoP5mSidKQguzYv7_fuClBzAeq_iYHdqLaXK-tSq0_mHw2jLgVPGYk-
02UZwUEUINqqXnfd04YPQ04XgmtrieqJzTaQH3xGpD2N2u-Z8VqaJkYD-
k99_Qftu0WUYOWvWaw0vgTATN_Vu2EgPB9uOSC5Ul7jmiieGAf5O7yTlivMdU2hMETPrsa-
FXomu56nzu0kxE0rx7h_Ckf2fDI3QatBSSnyndmK3jWiEttE65U_8-wJSU8Js_mE5-
MtXDBk9qobRfMcJpfzAgatdD7RWRyuIV85rERPOqKfaoqe_xt1iuW8zNhy634zLx6kv_aKI6d8jE1QgO0qZIdgRcKqpP1z
SPmC2rf1Z7ReOCtH_pc97Nj_0pXQBzftI3oWhnF6gpTtROKHn6-kbvne249vASSPM70UO9y8PXyMK-
kQGT8GGsmreAlWh1SFOMgiN-cfRrlGxHTAn0w
```

B.5.7a Przykładowa wartość dodatkowych danych uwierzytelniających:

```
eyJraWQiOiIzQTAwMDAwMDA4RTY4RjU1ODE1RDRFQTYzQTAwMDEwMDAwMDAwOCwgQ049ZUthc3ktU3ViQ0EiLCJlbmMiOi
JBMTI4Q0JDLUhTMjU2IiwiYWxnIjoiUlNBMV81In0
```

B.5.7b Przykład wyliczenia reprezentacji wartości długości dodatkowych danych uwierzytelniających zakodowana w Base64URL:

```
JWE_AAD_URL BYTES LENGTH -> 135

JWE_AAD_URL BITS LENGTH -> 135 * 8 = 1080

JWE_AT = [0, 0, 0, 0, 0, 0, 4, 56]
```

odzwierciedlenie w postaci szesnastkowej:

```
0000000000000438
```

B.5.8a Przykładowa wartość etykiety uwierzytelniającej zakodowana w Base64URL:

```
-x_YGNfKRdmlmIdWG7qxEA
```

odzwierciedlenie w postaci szesnastkowej:

```
fb1fd818d7ca45d9a59887561bbab110
```

B.5.8b Przykład wyliczenia wartości etykiety uwierzytelniającej przykładowych danych:

Postać szesnastkowa odszyfrowanego 32 bajtowego klucza CEK (Content Encryption Key) B.5.3:

```
0fae0202c8aa5d39bac1b9f58a9f440c4b700abdffd7e661ac4c609e288529be
```

pierwsze 16 bajtów wykorzystywane jako klucz w funkcji HMAC (JWE_MAC_KEY):

postać szesnastkowa wykorzystana w funkcji HMAC:
0fae0202c8aa5d39bac1b9f58a9f440c

użyte dane autoryzujące AAD (nagłówek JWE) B.5.7a:

postać ASCII:

```
{"kid":"3A00000008E68F55815D4EA63A000100000008, CN=eKasy-SubCA","enc":"A128CBC-
HS256","alg":"RSA1_5"}
```

postać Base64URL:
eyJraWQiOiIzQTAwMDAwMDA4RTY4RjU1ODE1RDRFQTYzQTAwMDEwMDAwMDAwOCwgQ049ZUthc3ktU3ViQ0EiLCJlbmMiOiJBMTI4Q0JDLUhTMjU2IiwiYWxnIjoiUlNBMV81In0

postać szesnastkowa (bajty postaci Base64URL) wykorzystana w funkcji HMAC:
65794a72615751694f69497a515441774d4441774d4441345254593452655154597a515441774d4445774d4441774d4441774f437767513034395a55746863336b7455335669513045694c434a6c626d4d694f694a424d54493451304a444c5568544d6a5532496977695957786e496a6f69556c4e424d563831496e30

użyty 16 bajtowy wektor inicjujący IV B.5.5:

postać Base64URL:
x3NlSpdTUDFhmlJfQoXz3Q

postać szesnastkowa wykorzystana w funkcji HMAC:
c773654a97535031619a525f4285f3dd

zaszyfrowane dane użyte do wyliczenia etykiety uwierzytelniającej B.5.6:

postać Base64URL:
jnYxL4MrKbWmgY5ZeUrL_etV-
byYTeEwp8fH1j5q8ii54kYWQPUBErh3APg7KLl8ZjY644HdbcZgy168x0cACpVhh5MqA93K5CXl0e-
4VUH97iBcHo_VdzUW_iqX3HPyfl1WLrt2PgIiqFOuzjapVYiVFzmXwhKsoWx4oYzIJfUJYM1QFcHnPVWPONlTnpDV6VRss
km3oC4sCn9DW0TJ-
h8HHzvXT18n6F1SHLrqcVgfb4cpz5w6X_k8TBPXaJNgOvEgcbUTXNkKkUWs2xTKNdJAinKTzXR9WFsPajd6R5brwVIEpRg
sHy6icr4RyxGO7fPS8JmKsbEY2CqedVO62ZtYREH1K2giji5lXPEpLPa7kFHdJ4uknjOjbgK-_Kbb9BUVdhdFO6-
bNpT67fiReZ--JLyu-cAYVnoTMj0GqSXOjNLiPZbVG1Q3Q02yw875kPf0nrlPihkNoCGgdBgP66mvF0CdP-
0hoZiMYKh9qOzeAtuiDb0kQL91mPexz_YZ1GQs-UrkYP85Fnv-vpvPkgBp9kJl9x1SiB4qvj86Ryaf-
ic8P_FvOy9bPv9wRtGIaDTcJBuK0vaIb7W7FCYiBKyKv6oR0vsoXghHS8NrH7ARm0VIC-
q9v_S8exfy5Rik6znN51Qwsdz5qG0b-rQU5Gs1hLFckc8Ra1wDMvyfv6qSE6-
dbpmbIsw9rCh2tvNzwM2nN4cHMZQErfpa8-
bTuGcRehZVwrjmfYkV9kcWqbSVajJfWAzeZLmjmcU3hytB6Ke7Yo2HyIjtQNqfWyWY4JmpgqSEVexk6MxeBQr35s2LKZ8H
ouO7jqqPV1RHh_oFIxNcaleisxh4nuEm3T8OfXU8UHvdjRHu2V_HjLZFFtuzT51iydzcHCUVK020mNqF92mMg3w3p6VTe6
xBtteuwIfG5uORdfw9Onu1u3e-PVe7P3H27iuPvLdS-
hG0ywN8aissvnDrIe2gKL_Na7Cuf_9XGv2EHXEEFqYJ67GUiDMYEK1M3JVM0QiQ0zFL1llPwyN9nHdXRyfneWo4931krUe
UYh-yxAtmgeH2lkT67pQ6jwZrqXS9WVBpBdN5JzMS8RjJShZY2BEotJp5MqMdz-
IfZwnx07JBjuMobskg_1yPmtCWQXoAb-MA574jiV40EEKq9HkS27T59wOS-
XdOi32KJHs3a9b1pSgmZqiWSUpwLaOuuZYjV9H7_sz5i6w8oCByxrcOLcjY1T2cp0HIaHiM1LnHrDvmAN0Weh1rthK2I4E
sy9yHfl1RjFwzoUmSz5sYdnx2jqKOEskAiaqc45IMTEdFIV8pWJcFsaxMx1CjUeumu_IGom0aiMDikvAyrs-
BmFwIOVSRzIsyeznghHS6K5H3R9isuiTA64OkomnqKIcC2TFjmx97EAjMGG3CBqc7fzdPMDcCYRoTgrq8JF-
gjYED4RBAZ8cZYBJx77neOpZ-
Kdy_8h3zT70GmJzmzjshrqDnPV3CdT_T7JCPWPoE0cQk2oUIvKCnx3DsH6x_1tti6FkzfdqAcGj7VftQN5XAwVcmMBZQwh
ChFS945C1soRD1SLWqWfaxG0CBqu5rVfwODOZsV4WAxX0XjrE6CbDbeLuCrzcjTto3rcRnrsVAWb-
fnC97FEsevL4Lfg7S7iE8YRCR1QW3M4OaI-46SHMBov2u92RdPvzLpYjIf3N29YmijYkFMcYnfVlt7-
U2Ou9YpNNhF5h2veEHzWUS07gyy-pSOtkjLXwp2MpKQucf3jyXJWyuArU4qVAaN-E-
WZkO1oNI_5zFb3W26AbEoumrhL8FpY8P5KGYAgH1B3uVNs2TQzdb3cAg4lIe9PBeeNtMi2-m505-
r1ljd0uLhmaZeidaCYdr4V50AiC3y4ggd6R4GE0u6QXttdCgwvRWAHUUBtzT4RONe4-KzcZVGK3Is2zGQL-
HrTii_oOgmDBpdTsnWTX8wFBITUw_tKf7YOZbwedLk8QVLYKji9ZdlPOek-w-
zRXaYDTQotleP4oRdMy4wmPn1mHsdF8RJ0UCQ8YrRF1Fz7ogp9C8ftU3YpGo9QxfKyaw46TsiGlQrEBCQ3r8gRNOTXHcCs
O4l-3iB05u4pDhZzz-7WbHltXGuDqFik7BWnq-
V0wRhX7KXei3ikGSsp5eCrigbHWsci_QdCmj5YjR5FV7Av29VqkZScCKZQ4_f4SHiYPuM7hurB56qLmpBlhkvxUZHyvp0u
PCa-vDgSn0FQ23dAVvmUI6epkad_-YwUtOthCJvfphZeiH8rvAOr1p1YM71EEkW-
u5xrCrdHtBy7NQpGCVYs_Cm6g_UIfguOrgOBmvgEKQlKapuLLi_rGwHsSNOlnMA9yx3sC4PGTeIAHca1wtOp1sFQwUxcgE
QRV8TY86G17qlrQIs-v9-
uDwSEI7aZpqGfZmZKaQOWCno23i4XbDQnnnjCdF1fmMxACorg5g_Wjcpgs91fPiePv9_1YYEnXRPduqhQbVb1J2txdOoSM
CrG9RZesYhV0KBvLOHBbhFBCD-
5IApXXIJ3It2aU2HJUtD77B60U2dOlZ6RzQ2OvmNgJ8oUSozKplTiYaDLKE3mgMO4o2IksZjvKISi69MmSJy059vQXkPBG
AJ3hDlHsCwrxCg43ZvbcV2T55466HhsBv8WAtTl01psYO0IBAp2QgbFfkZkYEaP5XgLmdtAGBJwPjSSgH4dM31MgbgRXyd
yo9qoEFUJj3-bzYMGoP5mSidKQguzYv7_fuClBzAeq_iYHdqLaXK-tSq0_mHw2jLgVPGYk-
02UZwUEUINqqXnfd04YPQ04Xgmtrieqj zTaQH3xGpD2N2u-Z8VqaJkYD-
k99_Qftu0WUYOWvWaw0vgTATN_Vu2EgPB9uOSC5Ul7jmiieGAf5O7yTlivMdU2hMETPrsa-
FXomu56nzu0kxE0rx7h_Ckf2fDI3QatBSSnyndmK3jWiEttE65U_8-wJSU8Js_mE5-
MtXDBk9qobRfMcJpfzAgatdD7RWRyuIV85rERPOqKfaoqe_xt1iuW8zNhy634zLx6kv_aKI6d8jE1QgO0qZIdgRcKqpP1z
SPmC2rf1Z7ReOCtH_pc97Nj_0pXQBzftI3oWhnF6gpTtROKHn6-kbvne249vASSPM70UO9y8PXyMK-
kQGT8GGsmreAlWh1SFOMgiN-cfRrlGxHTAn0w

postać szesnastkowa wykorzystana w funkcji HMAC:
8e76312f832b29b5a6818e59794acbfdeb55f9bc984de130a7c7c7d63e6af228b9e2461640f50112b87700f83b28b97c66363ae381dd6dc660cb5ebcc747000a956187932a03ddcae425e5d1efb85541fdee205c1e8fd5773516fe2a97dc73f27e5d562ebb763e0222a853aece36a9558895173997c212aca16c78a18cc825f50960cd5015c1e73d558f38d953

9e90d5e9546cb249b7a02e2c0a7f435b44c9fa1f071f3bd74f5f27e85d521cbaea71581f6f8729cf9c3a5ff93c4c13
d76893603af12071b5135cd90a9145acdb14ca35d2408a7293cd747d585b0f6a377a4796ebc15204a5182c1f2ea272
be11cb118eedf3d2f0998ab1b118d82a9e7553bad99b584441f52b68228e2e655cf1292cf6bb9051dd278ba49e33a3
6e02befca6dbf415157617453baf9b3694faedf891799fbe24bcaef9c018567a13323d06a925ce8cd2e23d96d51b54
37434db2c3cef990f7f49eb94f8a190da021a074180feba9af17409d3fed21a1988c60a87da8ecde02dba20dbd2440
bf7598f7b1cff619d4642cf94ae460ff39167bfebe9bcf920069f64265f71d52881e2abe3f3a47269ffa273c3ff16f
3b2f5b3eff7046d1886834dc241b8ad2f6886fb5bb14262204ac8abfaa11d2fb285e08474bc36b1fb0119b45480bea
bdbff4bc7b17f2e518a4eb39cde75430b1dcf9a86d1bfab414e46b3584b15c91cf116b5c0332fc9fbfaa9213af9d6e
999b22cc3dac2876b6f373c0cda7378707319404adfa5af3e6d3b867117a1655c2b8e67d8915f64716a9b4956a325f
580cde64b9a399c537872b41e8a7bb628d87c888ed40da9f5b2598e099a982a48455ec64e8cc5e050af7e6cd8b299f
07a2e3bb8eaa8f57544787fa0523135c6a57a2b318789ee126dd3f0e7d753c507bdd8d11eed95fc78cb64516dbb34f
9d62c9dcdc1c25152b4db498da85f7698c837c37a7a5537bac41b6d7aec087c6e6e39175fc3d3a7bb5bb77be3d57bb
3f71f6ee2b8fbcb752fa11b4cb037c6a2b2cbe70eb21eda028bfcd6bb0ae7fff571afd841d710416a609ebb1948833
1810ad4cdc954cd10890d3314bd6594fc3237d9c77574727e7796a38f77d64ad4794621fb2c40b6681e1f69644faee
943a8f066ba974bd59506905d379273312f118c94a1658d81128b49a7932a31dcfe21f6709f1d3b2418ee3286ec920
ff5c8f9ad096417a006fe300e7be23895e341042aaf47912dbb4f9f70392f9774e8b7d8a247b376bd6f5a5282666a8
96494a702da3aeb9962357d1fbfeccf98bac3ca02072c6b70e2dc8d8d53d9ca741c868788cd4b9c7ac3be600dd167a
1d6bb612b623812ccbdc877e5d518c5c33a14992cf9b18767c768ea28e12c90089aa9ce3920c4c4745215f29589705
b1ac4cc750a351eba6bbf206a26d1a88c0e292f032aecf81985c08395491cc8b327b39e08474ba2b91f747d8acba24
c0eb83a4a269ea288702d931639b1f7b1008cc186dc206a73b7f374f303702611a1382babc245fa08d8103e1104067
c719601271efb9de3a967e29dcbff21df34fbd06989ce6ce3b21aea0e73d5dc2753fd3ec908f58fa04d1c424da8508
bca0a7c770ec1fac7fd6db62e859337dda807068fb55fb503795c0c15726301650c210a1152f78e42d6ca110f548b5
aa59f6b11b4081aaee6b55fc0e0ce66c578580c57d178eb13a09b0db78bb82af37234eda37adc467aec54059bf9f9c
2f7b144b1ebcbe0b7e0ed2ee213c611091d505b733839a23ee3a487301a2fdaef7645d3efccba588c87f7376f589a2
8d890531c6277d596defe5363aef58a4d361179876bde107cd6512d3b832cbea523ad9232d7c29d8ca4a42e71fde3c
97256cae02b538a9501a37e13e59990ed68348ff9cc56f75b6e806c4a2e9ab84bf05a58f0fe4a1980201f5077b9536
cd9343375bddc020e2521ef4f05e78db4c8b6fa6e74e7eaf5963774b8b8666997a275a09876be15e740220b7cb8820
77a478184d2ee905edb5d0a0c2f45600751406dcd3e1138d7b8f8acdc65518adc8b36cc640bf87ad38a2fe83a09830
69753b275935fcc050484d4c3fb4a7fb60e65bc1e74b93c4152d82a38bd65d94f39e93ec3ecd15da6034d0a2d95e3f
8a1174ccb8c263e7d661ec745f1127450243c62b445d45cfba20a7d0bc7ed5376291a8f50c5f2b26b0e3a4ec886950
ac4042437afc81134e4d71dc0ac3b897ede2074e6ee290e1673cfeed66c796d5c6b83a858a4ec15a7abe574c11857e
ca5de8b78a4192b29e5e0ab8a06c75ac722fd07429a3e588d1e4557b02fdbd56a91949c08a650e3f7f84878983ee33
b86eac1e7aa8b9a9065864bf15191f2be9d2e3c26bebc38129f4150db774056f99423a7a991a77ff98c14b4eb61089
bdfa6165e887f2bbc03abd69d5833bd441245bebb9c6b0ab747b41cbb350a4609562cfc29ba83f5087e0b8eae03819
af80429094a6a9b8b2e2feb1b01ec48d3a59cc03dcb1dec0b83c64de2001dc6b5c2d3a9d6c150c14c5c80441157c4d
8f3a1b5eea96b408b3ebfdfae0f048423b699a6a19f66664a6903960a7a36de2e176c34279e78c2745d5f98cc400a8
ae0e60fd68dca60b3dd5f3e278fbfdff56181275d13ddbaa8506d56f5276b7174ea12302ac6f5165eb18855d0a06f2
ce1c16e1141083fb9200a575c827722dd9a5361c952d0fbec1eb453674e959e91cd0d8ebe636027ca144a8ccaa654e
261a0cb284de680c3b8a36224b198ef2884a2ebd326489cb4e7dbd05e43c1180277843947b02c2bc42838dd9bdb715
d93e79e3ae8786c06ff1602d4e5d35a6c60ed08040a764206c57e466460468fe5780b99db401812703e3492807e1d3
37d4c81b8115f2772a3daa81055098f7f9bcd8306a0fe664a274a420bb362feff7ee0a507301eabf8981dda8b6972b
eb52ab4fe61f0da32e054f19893ed36519c1411420daaa5e77ddd3860f434e17826b6b89ea89cd36901f7c46a43d8d
daef99f15a9a264603fa4f7dfd07edbb459460e5af59ac34be04c04cdfd5bb61203c1f6e3920b9525ee39a289e1807
f93bbc93962bcc754da13044cfaec6be157a26bb9ea7ceed24c44d2bc7b87f0a47f67c323741ab414929f29dd98ade
35a212db44eb953ff3ec09494f09b3f984e7e32d5c3064f6aa1b45f31c2697f30206ad743ed1591cae215f39ac444f
3aa29f6a8a9eff1b758ae5bcccd872eb7e332f1ea4bff68a23a77c8c4d5080ed2a64876045c2aaa4fd7348f982dab7
f567b45e382b47fe973decd8ffd295d00737ed237a1686717a8294ed44e2879fafa46ef9dedb8f6f01248f33bd143b
dcbc3d7c8c2be910193f061ac9ab78095687548538c82237e71f46b946c474c09f4c

długość dodatkowych danych autoryzujących w formacie Base64URL wynosi 123 bajty, czyli 984 bity i w formacie Big-Endian w postaci szesnastkowej ma wartość B.5.7b:

0000000000000438

Dokonując konkatenacji bajtowej AAD, IV oraz danych zaszyfrowanych i długości AAD do wyliczenia MAC będzie wykorzystany ciąg bajtów w postaci szesnastkowej:

65794a72615751694f69497a515441774d4441774d4441345254593452655154597a515441
774d4445774d4441774d4441774f437767513034395a55746863336b7455335669513045694c434a6c626d4d694f69
4a424d54493451304a444c5568544d6a5532496977695957786e496a6f69556c4e424d563831496e30c773654a9753
5031619a525f4285f3dd8e76312f832b29b5a6818e59794acbfdeb55f9bc984de130a7c7c7d63e6af228b9e2461640
f50112b87700f83b28b97c66363ae381dd6dc660cb5ebcc747000a956187932a03ddcae425e5d1efb85541fdee205c
1e8fd5773516fe2a97dc73f27e5d562ebb763e0222a853aece36a9558895173997c212aca16c78a18cc825f50960cd
5015c1e73d558f38d9539e90d5e9546cb249b7a02e2c0a7f435b44c9fa1f071f3bd74f5f27e85d521cbaea71581f6f
8729cf9c3a5ff93c4c13d76893603af12071b5135cd90a9145acdb14ca35d2408a7293cd747d585b0f6a377a4796eb
c15204a5182c1f2ea272be11cb118eedf3d2f0998ab1b118d82a9e7553bad99b584441f52b68228e2e655cf1292cf6
bb9051dd278ba49e33a36e02befca6dbf415157617453baf9b3694faedf891799fbe24bcaef9c018567a13323d06a9
25ce8cd2e23d96d51b5437434db2c3cef990f7f49eb94f8a190da021a074180feba9af17409d3fed21a1988c60a87d
a8ecde02dba20dbd2440bf7598f7b1cff619d4642cf94ae460ff39167bfebe9bcf920069f64265f71d52881e2abe3f

3a47269ffa273c3ff16f3b2f5b3eff7046d1886834dc241b8ad2f6886fb5bb14262204ac8abfaa11d2fb285e08474b
c36b1fb0119b45480beabdbff4bc7b17f2e518a4eb39cde75430b1dcf9a86d1bfab414e46b3584b15c91cf116b5c03
32fc9fbfaa9213af9d6e999b22cc3dac2876b6f373c0cda7378707319404adfa5af3e6d3b867117a1655c2b8e67d89
15f64716a9b4956a325f580cde64b9a399c537872b41e8a7bb628d87c888ed40da9f5b2598e099a982a48455ec64e8
cc5e050af7e6cd8b299f07a2e3bb8eaa8f57544787fa0523135c6a57a2b318789ee126dd3f0e7d753c507bdd8d11ee
d95fc78cb64516dbb34f9d62c9dcdc1c25152b4db498da85f7698c837c37a7a5537bac41b6d7aec087c6e6e39175fc
3d3a7bb5bb77be3d57bb3f71f6ee2b8fbcb752fa11b4cb037c6a2b2cbe70eb21eda028bfcd6bb0ae7fff571afd841d
710416a609ebb19488331810ad4cdc954cd10890d3314bd6594fc3237d9c77574727e7796a38f77d64ad4794621fb2
c40b6681e1f69644faee943a8f066ba974bd59506905d379273312f118c94a1658d81128b49a7932a31dcfe21f6709
f1d3b2418ee3286ec920ff5c8f9ad096417a006fe300e7be23895e341042aaf47912dbb4f9f70392f9774e8b7d8a24
7b376bd6f5a5282666a896494a702da3aeb9962357d1fbfeccf98bac3ca02072c6b70e2dc8d8d53d9ca741c868788c
d4b9c7ac3be600dd167a1d6bb612b623812ccbdc877e5d518c5c33a14992cf9b18767c768ea28e12c90089aa9ce392
0c4c4745215f29589705b1ac4cc750a351eba6bbf206a26d1a88c0e292f032aecf81985c08395491cc8b327b39e084
74ba2b91f747d8acba24c0eb83a4a269ea288702d931639b1f7b1008cc186dc206a73b7f374f303702611a1382babc
245fa08d8103e1104067c719601271efb9de3a967e29dcbff21df34fbd06989ce6ce3b21aea0e73d5dc2753fd3ec90
8f58fa04d1c424da8508bca0a7c770ec1fac7fd6db62e859337dda807068fb55fb503795c0c15726301650c210a115
2f78e42d6ca110f548b5aa59f6b11b4081aaee6b55fc0e0ce66c578580c57d178eb13a09b0db78bb82af37234eda37
adc467aec54059bf9f9c2f7b144b1ebcbe0b7e0ed2ee213c611091d505b733839a23ee3a487301a2fdaef7645d3efc
cba588c87f7376f589a28d890531c6277d596defe5363aef58a4d361179876bde107cd6512d3b832cbea523ad9232d
7c29d8ca4a42e71fde3c97256cae02b538a9501a37e13e59990ed68348ff9cc56f75b6e806c4a2e9ab84bf05a58f0f
e4a1980201f5077b9536cd9343375bddc020e2521ef4f05e78db4c8b6fa6e74e7eaf5963774b8b8666997a275a0987
6be15e740220b7cb882077a478184d2ee905edb5d0a0c2f45600751406dcd3e1138d7b8f8acdc65518adc8b36cc640
bf87ad38a2fe83a0983069753b275935fcc050484d4c3fb4a7fb60e65bc1e74b93c4152d82a38bd65d94f39e93ec3e
cd15da6034d0a2d95e3f8a1174ccb8c263e7d661ec745f1127450243c62b445d45cfba20a7d0bc7ed5376291a8f50c
5f2b26b0e3a4ec886950ac4042437afc81134e4d71dc0ac3b897ede2074e6ee290e1673cfeed66c796d5c6b83a858a
4ec15a7abe574c11857eca5de8b78a4192b29e5e0ab8a06c75ac722fd07429a3e588d1e4557b02fdbd56a91949c08a
650e3f7f84878983ee33b86eac1e7aa8b9a9065864bf15191f2be9d2e3c26bebc38129f4150db774056f99423a7a99
1a77ff98c14b4eb61089bdfa6165e887f2bbc03abd69d5833bd441245bebb9c6b0ab747b41cbb350a4609562cfc29b
a83f5087e0b8eae03819af80429094a6a9b8b2e2feb1b01ec48d3a59cc03dcb1dec0b83c64de2001dc6b5c2d3a9d6c
150c14c5c80441157c4d8f3a1b5eea96b408b3ebfdfae0f048423b699a6a19f66664a6903960a7a36de2e176c34279
e78c2745d5f98cc400a8ae0e60fd68dca60b3dd5f3e278fbfdff56181275d13ddbaa8506d56f5276b7174ea12302ac
6f5165eb18855d0a06f2ce1c16e1141083fb9200a575c827722dd9a5361c952d0fbec1eb453674e959e91cd0d8ebe6
36027ca144a8ccaa654e261a0cb284de680c3b8a36224b198ef2884a2ebd326489cb4e7dbd05e43c1180277843947b
02c2bc42838dd9bdb715d93e79e3ae8786c06ff1602d4e5d35a6c60ed08040a764206c57e466460468fe5780b99db4
01812703e3492807e1d337d4c81b8115f2772a3daa81055098f7f9bcd8306a0fe664a274a420bb362feff7ee0a5073
01eabf8981dda8b6972beb52ab4fe61f0da32e054f19893ed36519c1411420daaa5e77ddd3860f434e17826b6b89ea
89cd36901f7c46a43d8ddaef99f15a9a264603fa4f7dfd07edbb459460e5af59ac34be04c04cdfd5bb61203c1f6e39
20b9525ee39a289e1807f93bbc93962bcc754da13044cfaec6be157a26bb9ea7ceed24c44d2bc7b87f0a47f67c3237
41ab414929f29dd98ade35a212db44eb953ff3ec09494f09b3f984e7e32d5c3064f6aa1b45f31c2697f30206ad743e
d1591cae215f39ac444f3aa29f6a8a9eff1b758ae5bcccd872eb7e332f1ea4bff68a23a77c8c4d5080ed2a64876045
c2aaa4fd7348f982dab7f567b45e382b47fe973decd8ffd295d00737ed237a1686717a8294ed44e2879fafa46ef9de
db8f6f01248f33bd143bdcbc3d7c8c2be910193f061ac9ab78095687548538c82237e71f46b946c474c09f4c000000
0000000438

Wynik użycia powyższego ciągu bajtów oraz klucza JWE_MAC_KEY 0fae0202c8aa5d39bac1b9f58a9f440c
w funkcji haszującej HS256 zwraca 32 bajtową wartość w postaci szesnastkowej:

fb1fd818d7ca45d9a59887561bbab110625d28d63a74d28c90d143724dfa3357

wydzielając pierwsze 16 bajtów otrzymanego wyniku:

fb1fd818d7ca45d9a59887561bbab110

po przekodowaniu do formatu Base64URL:

-x_YGNfKRdmlmIdWG7qxEA

otrzymujemy wyliczoną etykietę uwierzytelniającą identyczną z B.5.8a.

B.5.9 Pełna postać obiektu JWE przykładowej komendy przesyłanej do urządzenia fiskalnego:

eyJraWQiOiIzQTAwMDAwMDA4RTY4RjU1ODE1RDRFQTYzQTAwMDEwMDAwMDAwOCwgQ049ZUthc3ktU3ViQ0EiLCJlbmMiOi
JBMTI4Q0JDLUhTMjU2IiwiYWxnIjoiUlNBMV81In0.giC4C064EjKUdWmh0TmoQVuXbxByz5yWO53Hxda3Je1gRchNmnq0
s38RXyJt9L1e3SDe5hnZuVtgPKufUBgEblItprryumGYnqyhuzIbD0m8akTq9JyJHQ7SERZlGYIzAgQbIn7NJAKswtzhpP
56PxSnMRwegdx0PoW-ZlTx2dYSpHRobWYvpHjz4t25H_poYZh2nAmmzC4nWOGnlshNI0qxk21E64_Tb-
4ACpoqvk6WlPGygUYT4M1hoAN4w_P0fsTyQxcmsXt3RpfiCbCROY2oLyQYGVYnTodbpLp7T4kVKN3XnzT6szBFzMCJsQ9B
0Ug26mQARkH0cx7FWOtVCg.x3NlSpdTUDFhmlJfQoXz3Q.jnYxL4MrKbWmgY5ZeUrL_etV-
byYTeEwp8fH1j5q8ii54kYWQPUBErh3APg7KLl8ZjY644HdbcZgy168x0cACpVhh5MqA93K5CXl0e-
4VUH97iBcHo_VdzUW_iqX3HPyfl1WLrt2PgIiqFOuzjapVYiVFzmXwhKsoWx4oYzIJfUJYM1QFcHnPVWPONlTnpDV6VRss

km3oC4sCn9DW0TJ-
h8HHzvXT18n6F1SHLrqcVgfb4cpz5w6X_k8TBPXaJNgOvEgcbUTXNkKkUWs2xTKNdJAinKTzXR9WFsPajd6R5brwVIEpRg
sHy6icr4RyxGO7fPS8JmKsbEY2CqedVO62ZtYREH1K2giji5lXPEpLPa7kFHdJ4uknjOjbgK-_Kbb9BUVdhdFO6-
bNpT67fiReZ--JLyu-cAYVnoTMj0GqSXOjNLiPZbVG1Q3Q02yw875kPf0nrlPihkNoCGgdBgP66mvF0CdP-
0hoZiMYKh9qOzeAtuiDb0kQL91mPexz_YZ1GQs-UrkYP85Fnv-vpvPkgBp9kJl9x1SiB4qvj86Ryaf-
ic8P_FvOy9bPv9wRtGIaDTcJBuK0vaIb7W7FCYiBKyKv6oR0vsoXghHS8NrH7ARm0VIC-
q9v_S8exfy5Rik6znN51Qwsdz5qG0b-rQU5Gs1hLFckc8Ra1wDMvyfv6qSE6-
dbpmbIsw9rCh2tvNzwM2nN4cHMZQErfpa8-
bTuGcRehZVwrjmfYkV9kcWqbSVajJfWAzeZLmjmcU3hytB6Ke7Yo2HyIjtQNqfWyWY4JmpgqSEVexk6MxeBQr35s2LKZ8H
ouO7jqqPV1RHh_oFIxNcaleisxh4nuEm3T8OfXU8UHvdjRHu2V_HjLZFFtuzT51iydzcHCUVK020mNqF92mMg3w3p6VTe6
xBtteuwIfG5uORdfw9Onu1u3e-PVe7P3H27iuPvLdS-
hG0ywN8aissvnDrIe2gKL_Na7Cuf_9XGv2EHXEEFqYJ67GUiDMYEK1M3JVM0QiQ0zFL1llPwyN9nHdXRyfneWo4931krUe
UYh-yxAtmgeH2lkT67pQ6jwZrqXS9WVBpBdN5JzMS8RjJShZY2BEotJp5MqMdz-
IfZwnx07JBjuMobskg_1yPmtCWQXoAb-MA574jiV40EEKq9HkS27T59wOS-
XdOi32KJHs3a9b1pSgmZqiWSUpwLaOuuZYjV9H7_sz5i6w8oCByxrcOLcjY1T2cp0HIaHiM1LnHrDvmAN0Weh1rthK2I4E
sy9yHfl1RjFwzoUmSz5sYdnx2jqKOEskAiaqc45IMTEdFIV8pWJcFsaxMx1CjUeumu_IGom0aiMDikvAyrs-
BmFwIOVSRzIsyeznghHS6K5H3R9isuiTA64OkomnqKIcC2TFjmx97EAjMGG3CBqc7fzdPMDcCYRoTgrq8JF-
gjYED4RBAZ8cZYBJx77neOpZ-
Kdy_8h3zT70GmJzmzjshrqDnPV3CdT_T7JCPWPoE0cQk2oUIvKCnx3DsH6x_1tti6FkzfdqAcGj7VftQN5XAwVcmMBZQwh
ChFS945C1soRD1SLWqWfaxG0CBqu5rVfwODOZsV4WAxX0XjrE6CbDbeLuCrzcjTto3rcRnrsVAWb-
fnC97FEsevL4Lfg7S7iE8YRCR1QW3M4OaI-46SHMBov2u92RdPvzLpYjIf3N29YmijYkFMcYnfVlt7-
U2Ou9YpNNhF5h2veEHzWUS07gyy-pSOtkjLXwp2MpKQucf3jyXJWyuArU4qVAaN-E-
WZkO1oNI_5zFb3W26AbEoumrhL8FpY8P5KGYAgH1B3uVNs2TQzdb3cAg4lIe9PBeeNtMi2-m505-
r1ljd0uLhmaZeidaCYdr4V50AiC3y4ggd6R4GE0u6QXttdCgwvRWAHUUBtzT4RONe4-KzcZVGK3Is2zGQL-
HrTii_oOgmDBpdTsnWTX8wFBITUw_tKf7YOZbwedLk8QVLYKji9ZdlPOek-w-
zRXaYDTQotleP4oRdMy4wmPn1mHsdF8RJ0UCQ8YrRF1Fz7ogp9C8ftU3YpGo9QxfKyaw46TsiGlQrEBCQ3r8gRNOTXHcCs
O4l-3iB05u4pDhZzz-7WbHltXGuDqFik7BWnq-
V0wRhX7KXei3ikGSsp5eCrigbHWsci_QdCmj5YjR5FV7Av29VqkZScCKZQ4_f4SHiYPuM7hurB56qLmpBlhkvxUZHyvp0u
PCa-vDgSn0FQ23dAVvmUI6epkad_-YwUtOthCJvfphZeiH8rvAOr1p1YM71EEkW-
u5xrCrdHtBy7NQpGCVYs_Cm6g_UIfguOrgOBmvgEKQlKapuLLi_rGwHsSNOlnMA9yx3sC4PGTeIAHca1wtOp1sFQwUxcgE
QRV8TY86G17qlrQIs-v9-
uDwSEI7aZpqGfZmZKaQOWCno23i4XbDQnnnjCdF1fmMxACorg5g_Wjcpgs91fPiePv9_1YYEnXRPduqhQbVb1J2txdOoSM
CrG9RZesYhV0KBvLOHBbhFBCD-
5IApXXIJ3It2aU2HJUtD77B60U2dOlZ6RzQ2OvmNgJ8oUSozKplTiYaDLKE3mgMO4o2IksZjvKISi69MmSJy059vQXkPBG
AJ3hDlHsCwrxCg43ZvbcV2T55466HhsBv8WAtTl01psYO0IBAp2QgbFfkZkYEaP5XgLmdtAGBJwPjSSgH4dM31MgbgRXyd
yo9qoEFUJj3-bzYMGoP5mSidKQguzYv7_fuClBzAeq_iYHdqLaXK-tSq0_mHw2jLgVPGYk-
02UZwUEUINqqXnfd04YPQ04XgmtrieqJzTaQH3xGpD2N2u-Z8VqaJkYD-
k99_Qftu0WUYOWvWaw0vgTATN_Vu2EgPB9uOSC5Ul7jmiieGAf5O7yTlivMdU2hMETPrsa-
FXomu56nzu0kxE0rx7h_Ckf2fDI3QatBSSnyndmK3jWiEttE65U_8-wJSU8Js_mE5-
MtXDBk9qobRfMcJpfzAgatdD7RWRyuIV85rERPOqKfaoqe_xt1iuW8zNhy634zLx6kv_aKI6d8jE1QgO0qZIdgRcKqpP1z
SPmC2rf1Z7ReOCtH_pc97Nj_0pXQBzftI3oWhnF6gpTtROKHn6-kbvne249vASSPM70UO9y8PXyMK-
kQGT8GGsmreAlWh1SFOMgiN-cfRrlGxHTAn0w.-x_YGNfKRdmlmIdWG7qxEA