

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

Właściciel dokumentu	<i>Ministerstwo Finansów</i>
Wersja dokumentu	<i>4.0.1</i>
Status dokumentu	<i>dokumentacja</i>
Data utworzenia	<i>2017-12-05</i>
Data modyfikacji	<i>2021-11-09</i>
Nazwa pliku	<i>Opis techniczny protokołu komunikacyjnego kasa – Centralne Repozytorium Kas – Standardy kryptograficzne_v.4.0.1.docx</i>



## 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
2020-01-20	DI	Dostosowanie specyfikacji do kas w postaci oprogramowania	3.0.0
2020-04-20	DI	Uzupełnienie przykładów dla kas w postaci oprogramowania	3.0.1
2020-05-05	DI	Uzupełnienie przykładów dla kas w postaci oprogramowania	3.0.2
2021-02-18	CIRF	Dostosowanie specyfikacji do kas rejestrujących online	4.0.0
2021-11-09	CIRF	Dodanie opisu formatu QR kodu loteryjnego – punkt A.10.7	4.0.1



## Spis treści

1	Słownik pojęć używanych w dokumencie .....	7
1.1	Wykaz specyfikacji technicznych użytych w dokumencie .....	9
2	Zabezpieczenie kanału komunikacyjnego .....	10
3	Certyfikaty kasy fiskalnej .....	11
3.1	Kasy rejestrujące w postaci urządzenia .....	11
3.2	Kasy rejestrujące w postaci oprogramowania .....	12
4	Algorytmy kryptograficzne .....	13
4.1	Podpisywanie .....	13
4.2	Szyfrowanie symetryczne .....	13
4.3	Szyfrowanie klucza szyfrującego .....	14
4.4	Algorytmy kryptograficzne w kasach rejestrujących w postaci oprogramowania oraz kasach rejestrujących online .....	14
5	Szyfrowanie komend oraz zbiorów danych i dokumentów w postaci elektronicznej .....	17
5.1	Podpisywanie i szyfrowanie komend .....	18
5.2	Podpisywanie i szyfrowanie zbiorów danych oraz dokumentów w postaci elektronicznej .....	19
	Załącznik A .....	21
	A.1 Funkcje użyte w opisach .....	21
	A.2 Podpisywanie komend .....	22
	A.3 Szyfrowanie komend .....	24
	A.4 Podpisywanie danych .....	26
	A.5 Szyfrowanie danych .....	28
	A.6 Wysyłanie danych .....	30
	A.7 Odebranie komendy .....	31
	A.8 Odszyfrowanie komendy .....	32
	A.9 Weryfikacja podpisu komendy .....	34
	A.10 Algorytmy kryptograficzne w kasach rejestrujących w postaci oprogramowania oraz kasach rejestrujących online .....	35
	Załącznik B .....	53
	B.1 Przykładowe certyfikaty środowiska testowego .....	53
	B.2 Przykładowe dane procesu podpisywania komendy w środowisku testowym .....	53
	B.3 Przykładowe dane procesu szyfrowania komendy w środowisku testowym .....	56
	B.4 Przykładowe dane procesu podpisywania danych w środowisku testowym .....	63
	B.5 Przykładowe dane procesu szyfrowania danych w środowisku testowym .....	65
	Załącznik C .....	71
	C.1 Przykładowe certyfikaty środowiska testowego kas w postaci oprogramowania .....	71
	C.2 Przykładowe dane procesu podpisywania danych kas w postaci oprogramowania .....	72
	C.3 Przykładowe dane procesu podpisywania dokumentów w postaci elektronicznej .....	86



## 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](#).
- Kasa rejestrująca - kasa rejestrująca spełniająca kryteria i warunki techniczne określone w rozporządzeniu Ministra Przedsiębiorczości i Technologii z dnia 28 maja 2018 r. w sprawie kryteriów i warunków technicznych, którym muszą odpowiadać kasy rejestrujące.
- Kasa rejestrująca online - kasa rejestrująca spełniająca wymagania techniczne określone w rozporządzeniu Ministra Rozwoju, Pracy i Technologii z dnia 12 września 2021 r. w sprawie wymagań technicznych dla kas rejestrujących.
- Kasa rejestrująca w postaci oprogramowania - kasa rejestrująca w postaci oprogramowania spełniająca wymagania techniczne określone w projekcie rozporządzenie Ministra Finansów, Funduszy i Polityki regionalnej z dnia 7 września 2021 r. zmieniające rozporządzenie w sprawie kas rejestrujących mających postać oprogramowania.
- Kasa rejestrująca w postaci urządzenia - kasa rejestrująca lub kasa rejestrująca online.
- Kasa, kasa fiskalna – kasa rejestrująca, kasa rejestrująca online lub kasa rejestrująca w postaci oprogramowania.
- Dane, zbiór danych, dane przesyłane z kasy, dane przesyłane do repozytorium – ustrukturyzowany zestaw dokumentów fiskalnych i нефiskalnych oraz innych danych wygenerowanych przez kasę przesyłany do repozytorium w strukturze umożliwiającej wysłanie wielu dokumentów. Pojęcia danych, danych przesyłanych do repozytorium i zbioru danych są używane zamiennie.
- Dokument w postaci elektronicznej – ustrukturyzowany pojedynczy dokument fiskalny wytworzony przez kasę rejestrującą online lub kasę rejestrującą w postaci oprogramowania opatrzone podpisem kasy przesyłany do repozytorium i na życzenie do klienta – paragon fiskalny w postaci elektronicznej.
- Repozytorium - system teleinformatyczny, prowadzony przez ministra właściwego do spraw finansów publicznych, którego zasadniczym celem jest odbieranie i gromadzenie danych przekazywanych przez kasy rejestrujące i komunikacja z kasami rejestrującymi w zakresie niezbędnym do ich konfiguracji oraz realizowanie innych zadań dla celów kontrolnych i

analitycznych. Technicznie Repozytorium zostało zorganizowane jako zespół współpracujących ze sobą komponentów i składa się z publicznej chmury Azure (Azure WebApi i Azure Event Hub) oraz Serwera CPD i innych komponentów zlokalizowanych w Centrum Informatyki Resortu Finansów.

- Serwer CPD – serwer zlokalizowany w Centrum Informatyki Resortu Finansów, który realizuje wymianę poleceń pomiędzy kasą i repozytorium w szczególności realizuje fiskalizację kasy.
- Azure - chmura publiczna Azure, przez którą przechodzi główny ruch sieciowy pomiędzy kasami a repozytorium. Składa się z dwóch elementów: Azure WebApi oraz Azure Event Hub.
- Azure WebApi - wydzielona logicznie część chmury Azure, z której kasa otrzymuje polecenia np. zmiana harmonogramu komunikacji, zmiana ustawień, żądanie wysłania dodatkowych dokumentów, itp.
- Azure Event Hub - wydzielona logicznie część publicznej chmury Azure, za pośrednictwem której kasa wysyła dane do repozytorium.
- JPKID - niepowtarzalny w ramach pamięci chronionej numer przypisany w kasie do wytworzonego dokumentu, a w kasach mających postać oprogramowania to unikalny kolejny numer wytworzonego dokumentu w ramach numeru pamięci chronionej. Numer pamięci chronionej to numer przypisany fizycznej pamięci chronionej powiązanej z pamięcią fiskalną, ma postać numeryczną liczoną od jeden i maksymalnie trzy cyfry, a w kasach w postaci oprogramowania ma wartość równą jeden. Para wartości 'JPKID' oraz 'pamiecChr' zawartych w strukturze JSON jednoznacznie identyfikują wytworzony przez kasę dokument. Przez wytworzony dokument należy rozumieć wszystkie dokumenty przesyłane przez kasę do repozytorium czyli dokumenty fiskalne, dokumenty нефiskalne oraz zdarzenia. Identyfikator JPKID może mieć postać numeryczną liczoną od jeden i maksymalnie piętnaście cyfr a wraz z numerem pamięci chronionej jednoznacznie identyfikuje dokument wytworzony w kasie i przesłany do repozytorium. Identyfikator JPKID może mieć również postać osiemnastu cyfr otrzymanych przez połączenie znakowo numeru pamięci chronionej "pamiecChr" oraz identyfikatora dokumentu "JPKID" dopełniając obie wartości zerami odpowiednio do trzech i do piętnastu miejsc tak aby wynik składał się z osiemnastu cyfr.



## 1.1 Wykaz specyfikacji technicznych użytych w dokumencie

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

## 2 Zabezpieczenie kanału komunikacyjnego

W komunikacji urządzenia fiskalnego z repozytorium do zabezpieczania połączenia sieciowego stosowany jest standard TLSv1.2. Zalecany algorytm 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 kasy fiskalnej 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 kas 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 wygenerowanego dla każdej z kas z określoną ważnością w usłudze Azure WebApi. Certyfikaty repozytorium oraz usług przyjmowania danych do chmury publicznej przekazywane są do urządzenie fiskalnego podczas procesu fiskalizacji zgodnie z dokumentem „Opis techniczny protokołu komunikacyjnego kasa – Centralne Repozytorium Kas – Specyfikacja komend”.

### 3 Certyfikaty kasy fiskalnej

Kasa fiskalna 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](#).

#### 3.1 Kasy rejestrujące w postaci urządzenia

W kasach rejestrujących w postaci urządzenia wymagane jest umieszczenie w nazwie podmiotu (commonName) tylko numeru unikatowego kasy. Ważność certyfikatu kasy rejestrującej w postaci urządzenia nie może przekroczyć dwudziestu lat (zalecany okres ważności to pięć lat), a data ważności certyfikatu kasy nie może wykraczać poza datę ważności certyfikatu producenta. Poszczególne certyfikaty kas rejestrujących w postaci urządzenia 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) rsads(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)}

### 3.2 Kasy rejestrujące w postaci oprogramowania

W kasach rejestrujących w postaci oprogramowania wymagane jest umieszczenie atrybucie nazwa powszechna (commonName) pola podmiot tylko numeru unikatowego kasy oraz w atrybucie numer seryjny (serialNumber) pola podmiot tylko Numeru Identyfikacji Podatkowej (NIP) podatnika używającego kasy poprzedzonego prefiksem „VATPL-”. Ważność certyfikatu kasy rejestrującej w postaci oprogramowania nie może być krótsza niż pięć lat i nie może przekroczyć dziesięciu lat, a data ważności certyfikatu kasy nie może wykraczać poza datę ważności certyfikatu producenta. Poszczególne certyfikaty kas rejestrujących postaci oprogramowania 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 pola podmiot certyfikatów kas – wymagania szczegółowe:

- commonName [CN] = **wymagany**

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

- serialNumber = **wymagany**

OID description: [2.5.4.5] {joint-iso-itu-t(2) ds(5) attributeType(4) serialNumber(5)}

- 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 komend, zbioru danych i dokumentów elektronicznych przesyłanych pomiędzy kasą i repozytorium 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 SHA-256 opisaną w dokumencie [FIPS 180-4](#). W nagłówku JOSE obiektu JWS podpisanych danych JPK w atrybucie „alg” symbol algorytmu przyjmie wartość RS256 („alg”: „RS256”).

#### 4.1.1 Podpisywanie zbioru danych

W celu weryfikacji po stronie repozytorium autentyczności danych przesyłanych przez kasy w atrybucie „jpkcertificate” należy zamieścić certyfikat z kluczem publicznym kasy fiskalnej użytym do podpisania danych, w formacie binarnym DER zakodowanym Base64 bez znacznika początku i końca certyfikatu oraz bez znaków końca linii.

#### 4.1.2 Podpisywanie dokumentu w postaci elektronicznej

W celu weryfikacji po stronie repozytorium autentyczności dokumentów w postaci elektronicznej przesyłanych przez kasy rejestrujące online lub kasy rejestrujące w postaci oprogramowania certyfikat z kluczem publicznym kasy należy zamieścić jako jednoelementową tablicę w atrybucie „x5c”, 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 komend, zbiorów danych i dokumentów w postaci elektronicznej przesyłanych pomiędzy kasą i repozytorium 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	SHA-256

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 składającej się tylko z pola 'CN - commonName'.

### 4.4 Algorytmy kryptograficzne w kasach rejestrujących w postaci oprogramowania oraz kasach rejestrujących online

W kasach rejestrujących online oraz w kasach rejestrujących w postaci oprogramowania wyróżniamy dwa sposoby tworzenia i przesyłania danych do repozytorium. Pierwszy sposób to podobnie jak w kasach rejestrujących zbiory danych zawierające zestaw wielu dokumentów różnego typu. Drugi sposób to pojedyncze dokumenty w postaci elektronicznej. Podział ten jest ściśle określony poprzez opublikowane schematy struktur JSON. W kasach rejestrujących w postaci oprogramowania oraz kasach rejestrujących online występują następujące rodzaje schematów:

- o struktura zbioru danych,
- o struktura dokumentu w postaci elektronicznej.

W związku z wyodrębnieniem dwóch sposobów tworzenia danych wyróżniamy również dwa podejścia tworzenia podpisu i sumy kontrolnej (skrótów SHA2) poszczególnych dokumentów.

#### 4.4.1 Podpisywanie dokumentów przesyłanych do repozytorium w zbiorach danych

W celu wyliczenia podpisywanego skrótu dokumentu należy połączyć znakowo wyszczególnione dla danego typu dokumentu dane usuwając występujące na początku i na końcu białe znaki oraz przekształcając ciągi znaków do wielkich liter. Elementy niewystępujące należy pominąć i dla powstałego ciągu znaków wyliczyć wartość funkcji skrótu następnie zaszyfrować kluczem prywatnym kasy w sposób opisany w punkcie [4.1](#). Otrzymaną wartość bajtową należy zamieścić w strukturze JSON w postaci szesnastkowej. Poniżej wyszczególnienie pozycji w odpowiedniej kolejności, które należy użyć do wyliczenia skrótu dla poszczególnych typów dokumentów:

- |  |  |
|--|--|
| <ul style="list-style-type: none"><li>• Raport fiskalny fiskalizacji:<ul style="list-style-type: none"><li>- NIP</li><li>- nrUnik</li><li>- dataFisk</li><li>- sumaZm</li><li>- serwID</li></ul></li></ul>   | <ul style="list-style-type: none"><li>• Paragon anulowany<ul style="list-style-type: none"><li>- NIP</li><li>- nrUnik</li><li>- nrDok</li><li>- sumaBrutto</li><li>- zakSprzed</li></ul></li></ul> |
| <ul style="list-style-type: none"><li>• Raport fiskalny dobowy:<ul style="list-style-type: none"><li>- NIP</li><li>- nrUnik</li><li>- sprzedBrutto</li><li>- podatekNal</li><li>- zakRap</li></ul></li></ul> | <ul style="list-style-type: none"><li>• Dokument niefiskalny<ul style="list-style-type: none"><li>- NIP</li><li>- nrUnik</li><li>- nrDok</li><li>- zak</li></ul></li></ul>                         |

Dodatkowo wyszczególnienie pozycji w odpowiedniej kolejności, które należy użyć do wyliczenia skrótu dla poszczególnych typów dokumentów wyłącznie dla kas rejestrujących online:

- |              |                     |
|--------------|---------------------|
| • Faktura:   | • Faktura anulowana |
| - NIP        | - NIP               |
| - nrUnik     | - nrUnik            |
| - nrDok      | - nrDok             |
| - sumaBrutto | - sumaBrutto        |
| - zakSprzed  | - zakSprzed         |

Podpisywanie dokumentów w zbiorach danych zaprezentowane jest w załączniku A punkt [A.10.1](#).

#### 4.4.2 Wyznaczanie wartości funkcji skrótu dokumentów przesyłanych w zbiorach danych

W celu wyznaczenia wartości funkcji skrótu dokumentu należy użyć funkcji skrótu SHA-256. Wyznaczenie wartości funkcji skrótu dokumentu przesyłanego w zbiorach danych polega na obliczeniu skrótu SHA2 ze skrótu poprzedniego dokumentu i podpisu bieżącego dokumentu.

Wyznaczanie skrótu dokumentów przesyłanych w zbiorach danych opisane jest w punkcie [A.10.2](#).

#### 4.4.3 Podpisywanie dokumentów w postaci elektronicznej

Podpisywanie dokumentów w postaci elektronicznej realizowane jest z wykorzystaniem obiektów JWS zgodnie ze standardem tworzenia podpisów cyfrowych dla dokumentów JSON opisanym w dokumencie [RFC 7515](#). W sposób szczegółowo opisany w punkcie [5.2](#).

Podpisywanie dokumentów w postaci elektronicznej szczegółowo opisane jest w punkcie [A.10.3](#).

#### 4.4.4 Wyznaczanie wartości funkcji skrótu dokumentu w postaci elektronicznej

W celu wyznaczenia wartości funkcji skrótu dokumentu w postaci elektronicznej należy użyć funkcji skrótu (haszowania) SHA-256. Wyznaczenie wartości funkcji skrótu dokumentu w postaci obiektu JWS (**paragon fiskalny w postaci elektronicznej**) polega na obliczeniu wyniku funkcji SHA-256 dla całości dokumentu zakodowanego Base64URL obejmującego wszystkie trzy elementy, czyli nagłówek, zawartość i podpis dokumentu. Należy zaznaczyć iż nagłówek JWS zawiera wartość funkcji skrótu poprzedniego dokumentu o ile taki istniał.

Wyznaczanie skrótu dokumentów w postaci elektronicznej opisane jest w punkcie [A.10.4](#).

#### 4.4.5 Weryfikacja ciągłości łańcucha dokumentów

Badanie ciągłości łańcucha dokumentów oprócz podpisu bieżącego dokumentu wymaga wskazania poprzedniego dokumentu danego typu celem uzyskania jego skrótu SHA2. Dlatego wymagane jest umieszczenie w bieżącym dokumencie identyfikatora poprzedniego dokumentu, który składa się z osiemnastu cyfr, a jeśli jest to pierwszy dokument danego typu to z osiemnastu zer.

W celu określenia identyfikatora poprzedniego dokumentu należy połączyć znakowo identyfikator pamięci chronionej poprzedniego dokumentu "pamiecChr", który dla kas w postaci oprogramowania zawsze ma wartość równą jeden oraz identyfikatora dokumentu "JPKID" dopełniając

obie wartości zerami odpowiednio do trzech i do piętnastu miejsc tak aby wynik składał się z osiemnastu cyfr.

Mechanizm weryfikacji ciągłości łańcucha dokumentów opisany jest w punkcie [A.10.5](#).

#### 4.4.6 Tworzenie kodu weryfikującego dokument w postaci elektronicznej

Do utworzenia kodu weryfikującego dokument należy użyć bajtowo połączonych danych:

- wartość funkcji skrótu dokumentu,
- numer unikatowy kasy,
- numer kolejny dokumentu,
- znacznik czasu odzwierciedlający datę i czas zakończenia sprzedaży,
- kod autoryzacyjny.

Utworzenie kodu autoryzacyjnego opiera się na wykorzystaniu metody tworzenia kodu uwierzytelnienia wiadomości (MAC - Message Authentication Code) z wykorzystaniem funkcji skrótu (haszowania) SHA-256. Parametrami wejściowymi funkcji HMAC-256 jest tablica bajtowa utworzona z połączonych bajtowo wymienionych powyżej atrybutów: wartość funkcji skrótu SHA-256 dokumentu obliczona zgodnie z punktem [4.4.4](#) (32 bajty), numer unikatowy kasy tablica bajtowa znaków ASCII (13 bajtów), numer kolejny dokumentu (JPKID wraz z numerem pamięci chronionej) (8 bajtów), znacznik czasu odzwierciedlający datę i czas zakończenia sprzedaży w postaci numerycznej - UNIX timestamp w milisekundach (8 bajtów). Kluczem funkcji haszującej jest ważny na dzień wystawienia dokumentu klucz współdzielony pobrany z serwera CPD.

Utworzenie kodu weryfikującego polega na połączeniu bajtowym kolejno atrybutów: wartości funkcji skrótu SHA-256 dokumentu (32 bajty), numeru unikatowego kasy (13 bajtów), numeru kolejnego dokumentu (JPKID wraz z numerem pamięci chronionej) (8 bajtów), znacznika czasu w postaci numerycznej - UNIX timestamp w milisekundach (8 bajtów) oraz wyliczonego kodu autoryzacyjnego (32 bajty).

Szczegółowy opis usługi pobierania kluczy współdzielonych zawiera dokument „Opis techniczny protokołu komunikacyjnego kasa – Centralne Repozytorium Kas – Specyfikacja komend”.

Natomiast sposób weryfikacji kodu autoryzacyjnego na podstawie przesłanego kodu weryfikującego opisany jest w dokumencie „Centralne Repozytorium Kas – Opis usługi sprawdzającej kod weryfikujący paragon”.

Szczegółowy opis mechanizmu tworzenia kodu przedstawiono w punkcie [A.10.6](#).



## 5 Szyfrowanie komend oraz zbiorów danych i dokumentów w postaci elektronicznej

Wszystkie dane (dokumenty, komendy, odpowiedzi, raporty itp.) przechodzące przez publiczną chmurę są podpisywane i szyfrowane zarówno przez kasę fiskalną 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. Kasa fiskalna powinna 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 kasie fiskalnej. 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 jednej kasie i przechowywany w zasobach ministerstwa.

### Nazwy atrybutów kluczy kasy fiskalnej

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

Repozytorium operuje na trzech parach unikalnych kluczy:

- do dwustronnej komunikacji TLS kasy fiskalnej z serwerem CPD,
- do dwustronnej komunikacji TLS kasy z chmurą publiczną,
- do podpisywania i szyfrowania przesyłanych komend.

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 kas fiskalnych.

### Nazwy tagów kluczy Repozytorium

Nazwa tagu	Opis
-digitalCertificateWebApiTLS	Certyfikat WebAPI do komunikacji TLS z kasą fiskalną
-privateKeyWebApiTLS	Klucz prywatny WebAPI do komunikacji TLS z kasą fiskalną
-digitalCertificateCPDServerTLS	Certyfikat serwera CPD do komunikacji TLS z kasą fiskalną
-privateKeyCPDServerTLS	Klucz prywatny serwera CPD do komunikacji TLS z kasą fiskalną
-digitalCertificateRepositoryEncrypt	Certyfikat serwera CPD do szyfrowania komend przesyłanych do kasy
-privateKeyRepositoryEncrypt	Klucz prywatny serwera CPD do podpisywania komend przesyłanych do kasy
-digitalCertificateAzureEventHubTLS	Certyfikaty usługi EventHub na chmurze Azure do komunikacji TLS

## 5.1 Podpisywanie i szyfrowanie komend

Podpisywanie oraz szyfrowanie komend realizowane jest przez repozytorium 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 „Opis techniczny protokołu komunikacyjnego kasa – Centralne Repozytorium Kas – Specyfikacja komend”,
- 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 klucza 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 zbiorów danych oraz dokumentów w postaci elektronicznej

Podpisywanie oraz szyfrowanie zbioru danych oraz dokumentów w postaci elektronicznej 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, dostępne następujące rodzaje schematów:
  - struktura zbioru danych dla kas rejestrujących,
  - struktura zbioru danych dla kas rejestrujących online,
  - struktura zbioru danych dla kas rejestrujących w postaci oprogramowania,
  - struktura dokumentu w postaci elektronicznej dla kas rejestrujących online,
  - struktura dokumentu w postaci elektronicznej dla kas rejestrujących w postaci oprogramowania,
- 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:

dla zbioru danych:

- *"alg"* zawierającego symbol użytego algorytmu podpisu (RS256),
- *"jpkmetadata"* zawierającego zakodowany w Base64 obiekt JSON składający się z opcjonalnych 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,

dla dokumentów w postaci elektronicznej:

- *"alg"* zawierającego symbol użytego algorytmu podpisu (RS256),
- *"jpkmetadata"* zawierającego zakodowany w Base64 obiekt JSON składający się z opcjonalnych parametrów opisujących numer korelacyjny komendy oraz metodę kompresowania,
- *"x5c"* zawierającego zgodnie z opisem w [sekcji 4.1.6 dokumentu RFC 7515](#) jednoelementową tablicę JSON z certyfikatem klucza publicznego użytym do podpisu, w formacie DER zakodowanym Base64 bez znacznika początku i końca certyfikatu oraz bez znaków końca linii,
- *"eParagon.mf.gov.pl"* zawierającego zakodowany w Base64 obiekt JSON składający się z parametrów: identyfikatora bieżącego dokumentu *"JPKID"* (wraz z numerem pamięci chronionej), wersji schematu użytej struktury JSON *"wersja"*, daty i czasu wystawienia dokumentu *"dataJPK"* oraz elementu *"JPKREF"* zawierającego identyfikator *"JPKID"* (wraz z numerem pamięci chronionej) i wartość funkcji skrótu *"SHA256"* poprzedniego dokumentu,

- 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 1 MB - 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 danych przedstawiono 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 SHA-256.
- 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 SHA-256.
- RS256Verify - funkcja weryfikująca podpis algorytmem RSA z wykorzystaniem SHA-256.
- HS256 - funkcja generująca MAC algorytmem SHA-256 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 kasy fiskalnej, 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 ... 0NCJ2zprYt8XrNO7281jyA=="]}
```

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 kasie fiskalnej połączenie się z serwerem CPD i wywołanie wskazanej usługi (KFD – wykonanie komendy CMD08: Wyślij certyfikaty kasy fiskalnej).

### 4. Przygotowane dane należy zakodować w Base64URL:

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

Postać przykładowych danych w formacie Base64URL:

```
JWS_DATA_URL => eyJhdHRyaWJldGVzIjpb7ImNwZFNlcnZpY2VOYW11Ijois0ZEIn19
```

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 ... cnZpY2VOYWllIjoiS0ZEIn19
```

Pełna postać przykładowych danych do podpisu zakodowanych Base64URL przedstawiona jest w punkcie [B.2.4](#).

Wartość funkcji skrótu SHA-256 w formacie szesnastkowym obliczona na danych do podpisu z punktu [B.2.4](#):

```
68e24de4af1da3859f0e8658b229f8aa01950d56cfa1bbf8c4cb3c55d49683e9
```

6. Tworzenie podpisu z wykorzystaniem algorytmu RSA z funkcją skrótu SHA-256 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 kasy fiskalnej, 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 kasy fiskalnej 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. Zaszzyfrowanie 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 kasy fiskalnej 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, na przykład identyfikator harmonogramu przesyłania danych (TFD),
  - *compression* - metoda kompresowania przesyłanych danych:
    - DEFLATE - kompresja algorytmem opisanym w dokumencie [RFC 1951](#),
    - NONE – przesłanie nieskompresowanych danych,
    - brak parametru *"compression"* oznacza brak kompresji danych.

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 kasy fiskalnej 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. Zaszzyfrowanie 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 „Opis techniczny protokołu komunikacyjnego kasa – Centralne Repozytorium Kas – Specyfikacja komend” i umieszczenie otrzymanej wartości w parametrze "*commandId*". Natomiast zaszyfrowane dane należy umieścić w parametrze "*encryptedData*". Jeżeli rozmiar tworzonej paczki przekracza wielkość 1 MB 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 przestać 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 „Opis techniczny protokołu komunikacyjnego kasa – Centralne Repozytorium Kas – Specyfikacja komend” opierającym się na komunikacji z serwerem CPD kasa przełącza się na komunikację z usługą WebApi umieszczoną w publicznej chmurze Azure. Kasa fiskalna 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 „Opis techniczny protokołu komunikacyjnego kasa – Centralne Repozytorium Kas – Specyfikacja komend” 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 podpisu 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.

## A.10 Algorytmy kryptograficzne w kasach rejestrujących w postaci oprogramowania oraz kasach rejestrujących online

### A.10.1 Podpisywanie dokumentów przesyłanych do repozytorium w zbiorach danych:

Poniżej przedstawiono sposób podpisywania poszczególnych dokumentów przesyłanych w zbiorach danych. W zależności od typu dokumentu wyszczególniono zestaw pól struktury JSON użytych do obliczenia wartości skrótu SHA2 funkcją SHA-256 i podpisaniem algorytmem RSA.

#### A.10.1a Podpisywanie raportu fiskalnego fiskalizacji:

1. Przygotowanie danych do podpisu polegające na połączeniu wyspecyfikowanych danych pobranych ze struktury JSON w jeden łańcuch:

```
RAPFISK_SIGNING_INPUT => NIP|nrUnik|dataFisk|sumaZm|serwID
```

Skrócona postać przykładowych danych do podpisu oparta na przykładzie z punktu [C.2.1a](#):

```
NIP      = 6970000802
nrUnik   = WTE2001000009
dataFisk = 2020-04-10T01:23:45.678Z
sumaZm   = 9DD845A0C9C285DC2E13F3253352E1836DC603C0
serwID   = KW123
```

```
RAPFISK_SIGNING_INPUT => 6970 ... 10T01 ... W123
```

Pełna postać przykładowych danych do podpisu przedstawiona jest w punkcie [C.2.3a](#).

Wartość funkcji skrótu SHA-256 w formacie szesnastkowym obliczona dla pełnych danych do podpisu dokumentu:

```
feffe767b2d728cf6ce5fc00f83ccc7be82748a86b1b7be59ec67b3bf1aa39c7
```

2. Tworzenie podpisu z wykorzystaniem algorytmu RSA z funkcją skrótu SHA-256 oraz klucza prywatnego kasy fiskalnej:

```
RAPFISK_SIGN => RS256(RAPFISK_SIGNING_INPUT,RSA_PRIVATE_KEY)
```

Otrzymany podpis zapisywany jest w postaci szesnastkowej:

```
RAPFISK_SIGN_HEX => HEX(RAPFISK_SIGN)
```

Skrócona postać podpisu w postaci szesnastkowej:

```
RAPFISK_SIGN_HEX => 3019505879124143c ... 11fe22a622de55bb
```

Pełna postać przykładowego podpisu w postaci szesnastkowej przedstawiona jest w punkcie [C.2.4a](#).

Pełną postać podpisu w postaci szesnastkowej należy zapisać w strukturze JSON w polu "RSA":

```
rapFisk/podpis/RSA => RAPFISK_SIGN_HEX
```

### A.10.1b Podpisywanie raportu fiskalnego dobowego:

1. Przygotowanie danych do podpisu polegające na połączeniu wyspecyfikowanych danych pobranych ze struktury JSON w jeden łańcuch:

```
RAPDOB_SIGNING_INPUT => NIP||nrUnik||sprzedBrutto||podatekNal||zakRap
```

Skrócona postać przykładowych danych do podpisu oparta na przykładzie z punktu [C.2.1a](#):

```
NIP          = 6970000802
nrUnik       = WTE2001000009
sprzedBrutto = 49060
podatekNal   = 5060
zakRap       = 2020-04-10T23:23:45.678Z
```

```
RAPDOB_SIGNING_INPUT => 6970 ... 10T23 ... 678Z
```

Pełna postać przykładowych danych do podpisu przedstawiona jest w punkcie [C.2.3b](#).

Wartość funkcji skrótu SHA-256 w formacie szesnastkowym obliczona dla pełnych danych do podpisu dokumentu:

```
6629368888063bbcc9cf80349d1a2b33b8d20f3ee6a0b4aabf15f51bc53d976e
```

2. Tworzenie podpisu z wykorzystaniem algorytmu RSA z funkcją skrótu SHA-256 oraz klucza prywatnego kasy fiskalnej:

```
RAPDOB_SIGN => RS256(RAPDOB_SIGNING_INPUT, RSA_PRIVATE_KEY)
```

Otrzymany podpis zapisywany jest w postaci szesnastkowej:

```
RAPDOB_SIGN_HEX => HEX(RAPDOB_SIGN)
```

Skrócona postać podpisu w postaci szesnastkowej:

```
RAPDOB_SIGN_HEX => 21184e6e94b7c822 ... b8a257d9467f4c1ce
```

Pełna postać przykładowego podpisu w postaci szesnastkowej przedstawiona jest w punkcie [C.2.4b](#).

Pełną postać podpisu w postaci szesnastkowej należy zapisać w strukturze JSON w polu "RSA":

```
rapDob/podpis/RSA => RAPDOB_SIGN_HEX
```

### A.10.1c Podpisywanie dokumentu нефiskalnego:

1. Przygotowanie danych do podpisu polegające na połączeniu wyspecyfikowanych danych pobranych ze struktury JSON w jeden łańcuch:

```
NIEFISK_SIGNING_INPUT => NIP||nrUnik||nrDok||zak
```

Skrócona postać przykładowych danych do podpisu oparta na przykładzie z punktu [C.2.1a](#):

```
NIP      = 6970000802
nrUnik   = WTE2001000009
```

```
nrDok    = 2
zak      = 2020-04-10T03:23:45.678Z

NIEFISK_SIGNING_INPUT => 6970 ... 10T03 ... 678Z
```

Pełna postać przykładowych danych do podpisu przedstawiona jest w punkcie [C.2.3c](#).

Wartość funkcji skrótu SHA-256 w formacie szesnastkowym obliczona dla pełnych danych do podpisu dokumentu:

```
1b46eb0df68a6a72766885366ef68ea42548491fc2608a75d8aedfa4cc720f6a
```

## 2. Tworzenie podpisu z wykorzystaniem algorytmu RSA z funkcją skrótu SHA-256 oraz klucza prywatnego kasy fiskalnej:

```
NIEFISK_SIGN => RS256(NIEFISK_SIGNING_INPUT, RSA_PRIVATE_KEY)
```

Otrzymany podpis zapisywany jest w postaci szesnastkowej:

```
NIEFISK_SIGN_HEX => HEX(NIEFISK_SIGN)
```

Skrócona postać podpisu w postaci szesnastkowej:

```
NIEFISK_SIGN_HEX => 282f2b77102bbb60 ... fc5e1cbba9c79fb1
```

Pełna postać przykładowego podpisu w postaci szesnastkowej przedstawiona jest w punkcie [C.2.4c](#).

Pełną postać podpisu w postaci szesnastkowej należy zapisać w strukturze JSON w polu "RSA":

```
wydrNiefisk/podpis/RSA => NIEFISK_SIGN_HEX
```

### A.10.1d Podpisywanie paragonu anulowanego:

#### 1. Przygotowanie danych do podpisu polegające na połączeniu wyspecyfikowanych danych pobranych ze struktury JSON w jeden łańcuch:

```
PARAGANUL_SIGNING_INPUT => NIP||nrUnik||nrDok||sumaBrutto||zakSprzed
```

Skrócona postać przykładowych danych do podpisu oparta na przykładzie z punktu [C.2.1a](#):

```
NIP          = 6970000802
nrUnik       = WTE2001000009
nrDok        = 5
sumaBrutto   = 1000000
zakSprzed    = 2020-04-10T06:23:45.678Z

PARAGANUL_SIGNING_INPUT => 6970 ... 10T06 ... 678Z
```

Pełna postać przykładowych danych do podpisu przedstawiona jest w punkcie [C.2.3d](#).

Wartość funkcji skrótu SHA-256 w formacie szesnastkowym obliczona dla pełnych danych do podpisu dokumentu:

```
9ebecd28bdcc0166b0967d67e048c1943a673215f014ce71e9424c0e82451e0f
```

2. Tworzenie podpisu z wykorzystaniem algorytmu RSA z funkcją skrótu SHA-256 oraz klucza prywatnego kasy fiskalnej:

```
PARAGANUL_SIGN => RS256(PARAGANUL_SIGNING_INPUT, RSA_PRIVATE_KEY)
```

Otrzymany podpis zapisywany jest w postaci szesnastkowej:

```
PARAGANUL_SIGN_HEX => HEX(PARAGANUL_SIGN)
```

Skrócona postać podpisu w postaci szesnastkowej:

```
PARAGANUL_SIGN_HEX => 9e61a538a24334c0 ... 03e51453caa1c8f3
```

Pełna postać przykładowego podpisu w postaci szesnastkowej przedstawiona jest w punkcie [C.2.4d](#).

Pełną postać podpisu w postaci szesnastkowej należy zapisać w strukturze JSON w polu "RSA":

```
paragAnul/podpis/RSA => PARAGANUL_SIGN_HEX
```

#### A.10.1e Podpisywanie faktury wyłącznie w kasach rejestrujących online:

1. Przygotowanie danych do podpisu polegające na połączeniu wyspecyfikowanych danych pobranych ze struktury JSON w jeden łańcuch:

```
FAKTURA_SIGNING_INPUT => NIP||nrUnik||nrDok||sumaBrutto||zakSprzed
```

Skrócona postać przykładowych danych do podpisu:

```
NIP          = 6970000802
nrUnik       = WTE2001000009
nrDok        = 5
sumaBrutto   = 1000000
zakSprzed    = 2020-04-10T06:23:45.678Z
```

```
FAKTURA_SIGNING_INPUT => 6970 ... 10T06 ... 678Z
```

2. Tworzenie podpisu z wykorzystaniem algorytmu RSA z funkcją skrótu SHA-256 oraz klucza prywatnego kasy fiskalnej:

```
FAKTURA_SIGN => RS256(FAKTURA_SIGNING_INPUT, RSA_PRIVATE_KEY)
```

Otrzymany podpis zapisywany jest w postaci szesnastkowej:

```
FAKTURA_SIGN_HEX => HEX(FAKTURA_SIGN)
```

Skrócona postać podpisu w postaci szesnastkowej:

```
FAKTURA_SIGN_HEX => 9e61a538a24334c0 ... 03e51453caa1c8f3
```

Pełną postać podpisu w postaci szesnastkowej należy zapisać w strukturze JSON w polu "RSA":

```
faktura/podpis/RSA => FAKTURA_SIGN_HEX
```

#### A.10.1f Podpisywanie faktury anulowanej wyłącznie w kasach rejestrujących online:

1. Przygotowanie danych do podpisu polegające na połączeniu wyspecyfikowanych danych pobranych ze struktury JSON w jeden łańcuch:

```
FAANUL_SIGNING_INPUT => NIP||nrUnik||nrDok||sumaBrutto||zakSprzed
```

Skrócona postać przykładowych danych do podpisu:

```
NIP          = 6970000802
nrUnik       = WTE2001000009
nrDok        = 5
sumaBrutto   = 1000000
zakSprzed    = 2020-04-10T06:23:45.678Z
```

```
FAANUL_SIGNING_INPUT => 6970 ... 10T06 ... 678Z
```

2. Tworzenie podpisu z wykorzystaniem algorytmu RSA z funkcją skrótu SHA-256 oraz klucza prywatnego kasy fiskalnej:

```
FAANUL_SIGN => RS256(FAANUL_SIGNING_INPUT,RSA_PRIVATE_KEY)
```

Otrzymany podpis zapisywany jest w postaci szesnastkowej:

```
FAANUL_SIGN_HEX => HEX(FAANUL_SIGN)
```

Skrócona postać podpisu w postaci szesnastkowej:

```
FAANUL_SIGN_HEX => 9e61a538a24334c0 ... 03e51453caa1c8f3
```

Pełną postać podpisu w postaci szesnastkowej należy zapisać w strukturze JSON w polu "RSA":

```
faAnul/podpis/RSA => FAANUL_SIGN_HEX
```

#### A.10.2 Wyznaczanie wartości funkcji skrótu dokumentów przesyłanych w zbiorach danych:

Poniżej przedstawiono sposób wyliczania wartości skrótu SHA2 funkcją SHA-256 bieżącego dokumentu na podstawie skrótu poprzedniego dokumentu oraz podpisu bieżącego dokumentu. Wyliczane skróty służą do weryfikowania ciągłości łańcucha dokumentów tego samego typu przesyłanych w zbiorach danych, czyli paragonów fiskalnych anulowanych oraz raportów fiskalnych dobowych. Poniżej przedstawiono przykłady oparte na paragonach anulowanych, a identyczna zasada dotyczy raportów fiskalnych dobowych.

##### A.10.2a Wyliczenie wartości skrótu pierwszego paragonu anulowanego:

1. Przygotowanie danych do wyliczenia skrótu w przypadku pierwszego dokumentu polega na pobraniu podpisu bieżącego dokumentu:

```
PARAGANUL_SHA_INPUT => PARAGANUL_SIGN
```

Skrócona postać podpisu bieżącego dokumentu w postaci szesnastkowej:

PARAGANUL\_SIGN\_HEX => 9e61a538a24334c0 ... 03e51453caa1c8f3

Pełna postać podpisu bieżącego dokumentu w postaci szesnastkowej przedstawiona jest w punkcie [C.2.4d](#).

Skrócona postać danych wejściowych w postaci szesnastkowej:

PARAGANUL\_SHA\_INPUT => 9e61a538a24334c0 ... 03e51453caa1c8f3

Pełna postać przykładowych danych wejściowych w postaci szesnastkowej przedstawiona jest w punkcie [C.2.5a](#).

## 2. Wyliczenie wartości skrótu SHA2 funkcją skrótu SHA-256 dla przygotowanych danych:

PARAGANUL\_SHA => SHA256(PARAGANUL\_SHA\_INPUT)

Otrzymany wynik zapisywany jest w postaci szesnastkowej:

PARAGANUL\_SHA\_HEX => HEX(PARAGANUL\_SHA)

Wartość funkcji skrótu SHA-256 w formacie szesnastkowym obliczona dla pełnych danych wejściowych:

eee0e3068482a34527630f4c73c05d0d3523899f5fa25d022b9415ff8057d3be

Otrzymany skrót w postaci szesnastkowej należy zapisać w strukturze JSON w polu "SHA":

paragAnul/podpis/SHA => PARAGANUL\_SHA\_HEX

Dodatkowo w strukturze JSON w polu "JPK" należy wpisać identyfikator poprzedniego dokumentu, a jeżeli jest to pierwszy dokument danego typu to wpisujemy osiemnaście zer:

paragAnul/podpis/JPK => 000000000000000000

### A.10.2b Wyliczenie wartości skrótu drugiego paragonu anulowanego:

#### 1. Przygotowanie danych do wyliczenia skrótu polegające na bajtowym połączeniu skrótu poprzedniego dokumentu i podpisu bieżącego:

PARAGANUL\_SHA\_INPUT => PARAGANUL\_SHA\_PREV || PARAGANUL\_SIGN

Opierając się na przykładowych danych z punktu [C.2.1a](#) skrót poprzedniego dokumentu, wyliczony w punkcie [A.10.2a](#) wynosi:

PARAGANUL\_SHA\_PREV => eee0e3068482a34527630f4c73c05d0d3523899f5fa25d022b9415ff8057d3be

Skrócona postać podpisu bieżącego dokumentu w postaci szesnastkowej:

PARAGANUL\_SIGN\_HEX => ab6c66f6f3cbc34d ... cdb694b9afb0ebf3

Pełna postać podpisu bieżącego dokumentu w postaci szesnastkowej przedstawiona jest w punkcie [C.2.4e](#).

Skrócona postać danych wejściowych w postaci szesnastkowej:



```
PARAGANUL_SHA_INPUT => eee0e3068482a345 ... cdb694b9afb0ebf3
```

Pełna postać przykładowych danych wejściowych w postaci szesnastkowej przedstawiona jest w punkcie [C.2.5b](#).

## 2. Wyliczenie wartości skrótu SHA2 funkcją skrótu SHA-256 dla przygotowanych danych:

```
PARAGANUL_SHA => SHA256(PARAGANUL_SHA_INPUT)
```

Otrzymany wynik zapisywany jest w postaci szesnastkowej:

```
PARAGANUL_SHA_HEX => HEX(PARAGANUL_SHA)
```

Wartość funkcji skrótu SHA-256 w formacie szesnastkowym obliczona dla pełnych danych wejściowych:

```
a64a913986f2a18d4db5ecbe02309fc96d19c683830622140a2d4310425f911c
```

Otrzymany skrót w postaci szesnastkowej należy zapisać w strukturze JSON w polu "SHA":

```
paragAnul/podpis/SHA => PARAGANUL_SHA_HEX
```

Dodatkowo w strukturze JSON w polu "JPK" należy wpisać identyfikator poprzedniego dokumentu. Opierając się na przykładowych danych z punktu [C.2.1a](#), identyfikator pamięci chronionej poprzedniego dokumentu pamiecChr = 1, a identyfikator pierwszego paragonu anulowanego JPKID = 6, dlatego identyfikator poprzedniego dokumentu wynosi:

```
paragAnul/podpis/JPK => 001000000000000006
```

### A.10.2c Wyliczenie wartości skrótu trzeciego paragonu anulowanego:

## 1. Przygotowanie danych do wyliczenia skrótu polegające na bajtowym połączeniu skrótu poprzedniego dokumentu i podpisu bieżącego:

```
PARAGANUL_SHA_INPUT => PARAGANUL_SHA_PREV || PARAGANUL_SIGN
```

Opierając się na przykładowych danych z punktu [C.2.1a](#) skrót poprzedniego dokumentu, wyliczony w punkcie [A.10.2b](#) wynosi:

```
PARAGANUL_SHA_PREV => a64a913986f2a18d4db5ecbe02309fc96d19c683830622140a2d4310425f911c
```

Skrócona postać podpisu bieżącego dokumentu w postaci szesnastkowej:

```
PARAGANUL_SIGN_HEX => 54ce7bad1079bc8a ... 73a937ec2f1ca93f
```

Pełna postać podpisu bieżącego dokumentu w postaci szesnastkowej przedstawiona jest w punkcie [C.2.4f](#).

Skrócona postać danych wejściowych w postaci szesnastkowej:

```
PARAGANUL_SHA_INPUT => a64a913986f2a18d ... 73a937ec2f1ca93f
```

Pełna postać przykładowych danych wejściowych w postaci szesnastkowej przedstawiona jest w punkcie [C.2.5c](#).

## 2. Wyliczenie wartości skrótu SHA2 funkcją skrótu SHA-256 dla przygotowanych danych:

```
PARAGANUL_SHA => SHA256 (PARAGANUL_SHA_INPUT)
```

Otrzymany wynik zapisywany jest w postaci szesnastkowej:

```
PARAGANUL_SHA_HEX => HEX (PARAGANUL_SHA)
```

Wartość funkcji skrótu SHA-256 w formacie szesnastkowym obliczona dla pełnych danych wejściowych:

```
ac6450833bef0f47888dea1272d179e8a197c8b6968c16782eeaff4f2769430d
```

Otrzymany skrót w postaci szesnastkowej należy zapisać w strukturze JSON w polu "SHA":

```
paragAnul/podpis/SHA => PARAGANUL_SHA_HEX
```

Dodatkowo w strukturze JSON w polu "JPK" należy wpisać identyfikator poprzedniego dokumentu. Opierając się na przykładowych danych z punktu [C.2.1a](#), identyfikator pamięci chronionej poprzedniego dokumentu pamiecChr = 1, a identyfikator drugiego paragonu anulowanego JPKID = 9, dlatego identyfikator poprzedniego dokumentu wynosi:

```
paragAnul/podpis/JPK => 001000000000000009
```

### A.10.3 Podpisywanie dokumentów w postaci elektronicznej:

Poniżej przedstawiono sposób podpisywania dokumentów w postaci elektronicznej wysyłanych z kasy rejestrującej online lub kasy rejestrującej w postaci oprogramowania do repozytorium oraz na życzenie do klienta, równoważny opisowi podpisywania danych przesyłanych do repozytorium [A.4](#), a 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 zgodnie z opisem w [sekcji 4.1.6 dokumentu RFC 7515](#) jednoelementową tablicę JSON z certyfikatem klucza publicznego użytym 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 parametru "eParagon.mf.gov.pl" zawierającego zakodowany w Base64 obiekt JSON składający się z parametrów:
  - JPKID - identyfikatora bieżącego dokumentu (wraz z numerem pamięci chronionej),
  - wersja - wersji schematu użytej struktury JSON,
  - dataJPK - daty i czasu wystawienia dokumentu,
  - JPKREF - elementu zawierającego:
    - JPKID - identyfikator poprzedniego dokumentu (wraz z numerem pamięci chronionej),
    - SHA256 - wartość funkcji skrótu poprzedniego dokumentu,
- 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 – przesłanie nieskompresowanych danych,
  - brak parametru "compression" oznacza brak kompresji danych.

UWAGA: ciąg znaków reprezentujący metadane może zostać poszerzony o wstawienie znaku specjalnego '\' poprzedzającego znak '/'.

Skrócona postać nagłówka w formacie JSON:

```
JWS_PH => {"alg": "RS256", "x5c": ["MIID ... Rw=="], "eParagon.mf.gov.pl": "eyJ3 ... In19",
"jpkmetadata": "eyJj ... In0="}
```

Pełna postać z wykorzystaniem certyfikatu testowego przedstawiona jest w punkcie [C.3.1](#).

Zgodnie z punktem [4.1](#) parametr "alg" przyjmuje wartość "RS256", ponieważ do podpisania dokumentu należy użyć algorytmu asymetrycznego RSA z dopełnieniem PKCS1 w wersji 1.5.

W parametrze "x5c" zamieszczono certyfikat klucza publicznego (punkt [C.1.2](#)) testowej kasy w postaci oprogramowania w formie jednoelementowej tablicy w formacie DER zakodowanym Base64 bez znacznika początku i końca certyfikatu oraz bez znaków końca.

W parametrach "eParagon.mf.gov.pl" oraz "jpkmetadata" zamieszczono zakodowane w Base64 odpowiednie struktury JSON.

Przykładowa postać parametru "eParagon.mf.gov.pl" w formacie JSON:

```
a) paragon pierwszy
{
  "wersja": "JPK_KASA_PARAGON_v1-0",
  "JPKID": "001000000000000004",
  "dataJPK": "2020-04-10T04:23:45.678Z",
  "JPKREF": {
    "SHA256": "0000000000000000000000000000000000000000000000000000000000000000",
    "JPKID": "000000000000000000"
  }
}

b) paragon drugi
{
  "wersja": "JPK_KASA_PARAGON_v1-0",
  "JPKID": "001000000000000005",
  "dataJPK": "2020-04-10T05:23:45.678Z",
  "JPKREF": {
    "SHA256": "F88E6E5AD956195072FC437954AB3BA05F0F3FF677F6A0129CB9EFC83538BA0",
    "JPKID": "001000000000000004"
  }
}

c) paragon trzeci
{
  "wersja": "JPK_KASA_PARAGON_v1-0",
  "JPKID": "001000000000000007",
  "dataJPK": "2020-04-10T07:23:45.678Z",
  "JPKREF": {
    "SHA256": "7769C389BB33C7F34B06D1CB6C0DA13F9AD30585BFE47B7DB085A2A26890E101",
    "JPKID": "001000000000000005"
  }
}
```

```

}

d) paragon czwarty
{
  "wersja": "JPK_KASA_PARAGON_v1-0",
  "JPKID": "001000000000000008",
  "dataJPK": "2020-04-10T08:23:45.678Z",
  "JPKREF": {
    "SHA256": "BBF43DA8F1BB31AE6F3149905732C50AD7A88BC6A2CE2A9360B45CBC1CDCF0F1",
    "JPKID": "001000000000000007"
  }
}

```

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

```

{"correlationId":"TFD.WTE2001000009.2020-04-03T11:50:00.322Z"}

{"correlationId":"TFD.WTE2001000009.2020-04-03T11:50:00.322Z","compression":"NONE"}

{"correlationId":"TFD.WTE2001000009.2020-04-03T11:50:00.322Z","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 [C.3.2](#).

## 3. Przygotowanie zawartości danych do podpisu i opcjonalnie skompresowanie:

```

JWS_DATA => { "dokument": { "naglowek": {...}, "podmiot1": {...}, "paragon": { ... } } }

opcjonalnie skompresowanie:

JWS_DATA => DEFLATE(JWS_DATA)

```

Pełna postać przykładowych ustrukturyzowanych danych przedstawiona jest w punkcie [C.3.3](#).

Pełna postać przykładowych nieskompresowanych danych przedstawiona jest w punkcie [C.3.4](#).

## 4. Przygotowane dane należy zakodować w Base64URL:

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

Pełna postać przykładowych danych zakodowanych Base64URL przedstawiona jest w punkcie [C.3.5](#).

## 5. Przygotowanie zawartości do podpisu 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 [C.1.2](#) przedstawiona jest w punkcie [C.3.6](#).

#### A.10.4 Wyznaczanie wartości funkcji skrótu dokumentów w postaci elektronicznej:

Poniżej przedstawiono sposób wyliczania wartości skrótu SHA2 funkcją SHA-256 bieżącego dokumentu na podstawie skrótu poprzedniego dokumentu oraz podpisu bieżącego dokumentu. Wyliczane skróty służą do weryfikowania ciągłości łańcucha dokumentów tego samego typu.

Przygotowanie danych do wyliczenia skrótu polega na pobraniu całego obiektu JWS ponieważ zawiera on podpis bieżącego dokumentu jak również skrót poprzedniego:

```
PARAGON_SHA_INPUT => PARAGON_JWS
```

Skrócona postać dokumentu w postaci elektronicznej:

```
PARAGON_SHA_INPUT => eyJlUGFyYWdvbi5t ... aXwz4zui_fOrXxsg
```

Pełna postać dokumentu w postaci elektronicznej przedstawiona jest w punkcie [C.3.6](#).

Wyliczenie wartości skrótu SHA2 funkcją skrótu SHA-256 dla przygotowanych danych:

```
PARAGON_SHA => SHA256(PARAGON_SHA_INPUT)
```

Otrzymany wynik zapisany w postaci szesnastkowej:

```
PARAGON_SHA_HEX => HEX(PARAGON_SHA)
```

a) wartość funkcji skrótu SHA-256 w formacie szesnastkowym obliczona dla dokumentu elektronicznego z punktu [C.3.6a](#):

```
f88e6e5ad956195072fc437954ab3ba05f0f3ff677f6a0129cb9effc83538ba0
```

b) wartość funkcji skrótu SHA-256 w formacie szesnastkowym obliczona dla dokumentu elektronicznego z punktu [C.3.6b](#):

```
7769c389bb33c7f34b06d1cb6c0da13f9ad30585bfe47b7db085a2a26890e101
```

c) wartość funkcji skrótu SHA-256 w formacie szesnastkowym obliczona dla dokumentu elektronicznego z punktu [C.3.6c](#):

```
bbf43da8f1bb31ae6f3149905732c50ad7a88bc6a2ce2a9360b45cbc1cdcf0f1
```

d) wartość funkcji skrótu SHA-256 w formacie szesnastkowym obliczona dla dokumentu elektronicznego z punktu [C.3.6d](#):

```
ebc3974cde25a119e123f6661e5c8d51670c735d4c10b123b2d4e57befe4021c
```

Otrzymany skrót w postaci szesnastkowej należy zapisać w nagłówku następnego dokumentu danego typu w postaci elektronicznej. Zapisu należy dokonać w nagłówku obiektu JWS w parametrze "eParagon.mf.gov.pl" w polu "SHA256":

```
JPKREF/SHA256 => PARAGON_SHA_HEX
```

W punkcie [A.10.3](#) przedstawiono przykładowe postaci parametru *"eParagon.mf.gov.pl"*, w których odpowiednio zawarto obliczone powyżej skróty SHA2. Mianowicie skrót pierwszego paragonu zawarto w nagłówku drugiego paragonu, skrót drugiego paragonu zawarto w nagłówku trzeciego paragonu i skrót trzeciego paragonu zawarto w nagłówku czwartego paragonu. Nagłówek pierwszego paragonu zawiera parametr *"eParagon.mf.gov.pl"*, którego pola "JPKREF/SHA256" i "JPKREF/JPKID" zawierają same zera odpowiednio sześćdziesiąt cztery oraz osiemnaście wskazujące, że jest to pierwszy dokument danego typu i nie posiada poprzednika.

#### A.10.5 Weryfikacja ciągłości łańcucha dokumentów w kasach w postaci oprogramowania:

Poniżej przedstawiono sposób weryfikacji ciągłości łańcucha zarówno dokumentów w postaci elektronicznej, czyli paragonów fiskalnych jak i dokumentów przesyłanych w zbiorach danych (paragonów fiskalnych anulowanych i raportów fiskalnych dobowych).

Mechanizm weryfikacji ciągłości łańcucha paragonów fiskalnych opiera się na porównaniu obliczonej wartości skrótu SHA2 funkcją SHA-256 poprzednika z wartością zawartą w bieżącym dokumencie. Natomiast weryfikacja ciągłości łańcucha paragonów fiskalnych anulowanych i raportów fiskalnych dobowych polega na wyliczeniu wartości skrótu SHA2 funkcją SHA-256 poprzednika, a następnie wyliczeniu wartości skrótu SHA2 funkcją SHA-256 bieżącego dokumentu na podstawie skrótu poprzedniego dokumentu oraz podpisu bieżącego dokumentu i porównaniu otrzymanego wyniku ze skrótem SHA2 bieżącego dokumentu zapisanym w strukturze JSON.

##### 1. Weryfikacja łańcucha dokumentów w postaci elektronicznej - paragonów fiskalnych:

W załączniku C w punkcie [C.3.6](#) zamieszczono przykład sekwencji czterech paragonów fiskalnych z możliwością weryfikacji łańcucha dokumentów. Rozpoczynając od ostatniego (czwartego) paragonu należy obliczyć skrót poprzednika o ile istnieje i porównać z wartością zapisaną w nagłówku bieżącego dokumentu.

W paragonie czwartym w nagłówku JWS parametr *"eParagon.mf.gov.pl"* ma postać:

```
{
  "wersja": "JPK_KASA_PARAGON_v1-0",
  "JPKID": "001000000000000008",
  "dataJPK": "2020-04-10T08:23:45.678Z",
  "JPKREF": {
    "SHA256": "BBF43DA8F1BB31AE6F3149905732C50AD7A88BC6A2CE2A9360B45CBC1CDCF0F1",
    "JPKID": "001000000000000007"
  }
}
```

Z powyższego należy wywnioskować iż poprzednik będzie charakteryzował się wartościami:

```
PARAGON_REF_SHA_HEX => BBF43DA8F1BB31AE6F3149905732C50AD7A88BC6A2CE2A9360B45CBC1CDCF0F1
PARAGON_REF_PAMIEC_CHR => 001 => 1
PARAGON_REF_JPKID => 0000000000000007 => 7
```

Opierając się na przykładowych danych z punktu [C.3.3](#), można zidentyfikować iż poprzednim paragonem był paragon trzeci zawierający identyfikator dokumentu JPKID równy 7, punkt [C.3.3c](#):

```
/dokument/paragon/pamiecChr == 1
/dokument/paragon/JPKID == 7
```

W związku z tym dla jego postaci elektronicznej (całego obiektu JWS) należy wyliczyć skrót SHA2:

```
PREV_PARAGON_SHA => SHA256(PREV_PARAGON_JWS)
```

Otrzymany wynik w postaci szesnastkowej:

```
PREV_PARAGON_SHA_HEX => HEX(PREV_PARAGON_SHA)
```

Wartość funkcji skrótu SHA-256 w formacie szesnastkowym obliczona dla dokumentu z punktu [C.3.6c](#):

```
bbf43da8f1bb31ae6f3149905732c50ad7a88bc6a2ce2a9360b45cbc1cdcf0f1
```

```
PARAGON_REF_SHA_HEX - BBF43DA8F1BB31AE6F3149905732C50AD7A88BC6A2CE2A9360B45CBC1CDCF0F1
PREV_PARAGON_SHA_HEX - bbf43da8f1bb31ae6f3149905732c50ad7a88bc6a2ce2a9360b45cbc1cdcf0f1
```

Porównując wyliczoną wartość ze skrótem zawartym w nagłówku dokonuje się weryfikacji sekwencji i w przypadku poprawnej powtarza się opisany algorytm traktując poprzednika jako bieżący dokument. Iterację można przeprowadzać dla wybranej liczby dokumentów lub do osiągnięcia pierwszego dokumentu, który nie posiada poprzednika. Wystąpienie rozbieżności przy porównywaniu skrótów SHA2 świadczy jednoznacznie o niepoprawnym utworzeniu łańcucha dokumentów.

## 2. Weryfikacja łańcucha dokumentów przesyłanych w zbiorach danych:

W załączniku C w punktach [C.2.2d](#), [C.2.2e](#) oraz [C.2.2f](#) zamieszczono przykład sekwencji trzech paragonów fiskalnych anulowanych z możliwością weryfikacji łańcucha dokumentów. Rozpoczynając od ostatniego ([C.2.2f](#)) paragonu anulowanego należy uzyskać skrót poprzednika o ile istnieje i połączyć bajtowo z podpisem bieżącego paragonu anulowanego, a z otrzymanej tablicy bajtów wyliczyć skrót SHA-256 i porównać z wartością zapisaną w bieżącym dokumencie.

W strukturze JSON paragonu anulowanego element *"podpis"* zawiera odpowiednie dane:

Fragment struktury JSON poprzedniego paragonu anulowanego [C.2.2f](#):

```
"paragAnul": {
  "JPKID": 10,
  "pamiecChr": 1,
  ...
  "podpis": {
    "RSA": "54ce7bad ... 2f1ca93f",
    "SHA": "ac6450833bef0f47888dea1272d179e8a197c8b6968c16782eeaff4f2769430d",
    "JPK": "001000000000000009"
  }
}
```

Z powyższego należy pobrać weryfikowany skrót SHA oraz podpis potrzebny do weryfikacji:

```
PARAGANUL_SHA_HEX => ac6450833bef0f47888dea1272d179e8a197c8b6968c16782eeaff4f2769430d
PARAGANUL_RSA_HEX => 54ce7bad ... 2f1ca93f
```

Dodatkowo należy wywnioskować iż poprzednik będzie charakteryzował się wartościami:

```
PARAGANUL_REF_PAMIEC_CHR => 001 => 1
PARAGANUL_REF_JPKID => 0000000000000009 => 9
```

Opierając się na powyższych danych można zidentyfikować iż poprzednim paragonem anulowanym był paragon zawierający identyfikator dokumentu JPKID równy 9, punkt [C.2.2e](#):

```
//paragAnul/pamiecChr == 1
//paragAnul/JPKID == 9
```

Fragment struktury JSON poprzedniego paragonu anulowanego [C.2.2e](#):

```
"paragAnul": {
  "JPKID": 9,
  "pamiecChr": 1,
  ...
  "podpis": {
    "RSA": "ab6c66f6 ... afb0ebf3",
    "SHA": "a64a913986f2a18d4db5ecbe02309fc96d19c683830622140a2d4310425f911c",
    "JPK": "001000000000000006"
  }
}
```

Ze wskazanego dokumentu należy pobrać skrót SHA:

```
PREV_PARAGANUL_SHA => a64a913986f2a18d4db5ecbe02309fc96d19c683830622140a2d4310425f911c
```

Następnie połączyć bajtowo z podpisem bieżącego dokumentu:

```
PARAGANUL_INPUT => PREV_PARAGANUL_SHA || PARAGANUL_RSA_HEX
```

W celu wyliczenia i weryfikacji skrótu SHA analizowanego dokumentu:

```
PARAGANUL_INPUT_SHA => SHA256(PARAGANUL_INPUT)
```

Otrzymany wynik w postaci szesnastkowej:

```
PARAGANUL_INPUT_SHA_HEX => HEX(PARAGANUL_INPUT_SHA)
```

Wartość funkcji skrótu SHA-256 w formacie szesnastkowym obliczona dla dokumentu z punktu [C.2.5c](#):

```
ac6450833bef0f47888dea1272d179e8a197c8b6968c16782eeaff4f2769430d
```

```
PARAGANUL_SHA_HEX          - ac6450833bef0f47888dea1272d179e8a197c8b6968c16782eeaff4f2769430d
PARAGANUL_INPUT_SHA_HEX    - ac6450833bef0f47888dea1272d179e8a197c8b6968c16782eeaff4f2769430d
```

Porównując wyliczoną wartość ze skrótem zawartym w strukturze JSON bieżącego dokumentu w elemencie "podpis" dokonuje się weryfikacji sekwencji i w przypadku poprawnej powtarza się opisany algorytm traktując poprzednika jako bieżący dokument. Iterację można przeprowadzać dla wybranej liczby dokumentów lub do osiągnięcia pierwszego dokumentu, który nie posiada poprzednika. Wystąpienie rozbieżności przy porównywaniu skrótów SHA2 świadczy jednoznacznie o niepoprawnym utworzeniu łańcucha dokumentów.

#### A.10.6 Tworzenie kodu weryfikującego dokument kasy w postaci oprogramowania:

Poniżej przedstawiono sposób tworzenia kodu weryfikującego paragon fiskalny na podstawie klucza współdzielonego oraz odpowiednich danych jednoznacznie identyfikujących dokument.



1. Przygotowanie danych identyfikujących paragon fiskalny, czyli wartość funkcji skrótu dokumentu, numer unikatowy kasy, numer kolejny dokumentu (JPKID wraz z numerem pamięci chronionej), znacznik czasu odzwierciedlający datę i czas zakończenia sprzedaży.

Wyliczenie skrótu paragonu fiskalnego (całego obiektu JWS) opisane w punkcie [A.10.4](#):

```
PARAGON_SHA_INPUT => PARAGON_JWS
```

Skrócona postać dokumentu w postaci elektronicznej:

```
PARAGON_SHA_INPUT => eyJlUGFyYWdvbi5t ... aXwz4zui_fOrXxsg
```

Pełna postać dokumentu w postaci elektronicznej przedstawiona jest w punkcie [C.3.6a](#).

Wyliczenie wartości skrótu SHA2 funkcją skrótu SHA-256 dla przygotowanych danych:

```
PARAGON_SHA => SHA256(PARAGON_SHA_INPUT)
```

Otrzymany wynik zapisany w postaci szesnastkowej:

```
PARAGON_SHA_HEX => HEX(PARAGON_SHA)
```

Wartość funkcji skrótu SHA-256 w formacie szesnastkowym obliczona dla dokumentu elektronicznego z punktu [C.3.6a](#):

```
PARAGON_SHA_HEX => f88e6e5ad956195072fc437954ab3ba05f0f3ff677f6a0129cb9effc83538ba0
```

Numer unikatowy kasy przydzielony jest każdej kasie i zapisany w bazie danych kasy oraz wysyłany w strukturach danych w formacie JSON. Opierając się na przykładowych danych z punktu [C.3.3a](#) kasy testowej numer unikatowy przyjmuje wartość:

```
PARAGON_NR_UNIK => /dokument/podmiot1/nrUnik => WTE2001000009
```

Zamieniając powyższą wartość w tablicę bajtową otrzymamy poniższą postać szesnastkową:

```
PARAGON_NR_UNIK_HEX => 5754453230303130303030303039
```

Pozostałe dane zawarte są zarówno w dokumencie jak również w nagłówku JWS w parametrze *"eParagon.mf.gov.pl"*, który dla paragonu pierwszego z punktu [C.3.6a](#) ma postać:

```
{
  "wersja": "JPK_KASA_PARAGON_v1-0",
  "JPKID": "00100000000000000004",
  "dataJPK": "2020-04-10T04:23:45.678Z",
  "JPKREF": {
    "SHA256": "0000000000000000000000000000000000000000000000000000000000000000",
    "JPKID": "000000000000000000"
  }
}
```

W związku z tym numer kolejny dokumentu (JPKID wraz z numerem pamięci chronionej) oraz znacznik czasu odzwierciedlający datę i czas zakończenia sprzedaży przyjmą wartości:

```
PARAGON_JPKID => 00100000000000000004
PARAGON_DATA_JPK => 2020-04-10T04:23:45.678Z
```

Powyższe wartości w postaci liczby całkowitej (data jako UNIX timestamp w milisekundach):

```
PARAGON_JPKID => 1000000000000004
PARAGON_DATA_JPK => 1586492625678
```

Obie wartości należy przedstawić w postaci ośmiobajtowej tablicy - postać szesnastkowa:

```
PARAGON_JPKID_HEX => 00038d7ea4c68004
PARAGON_DATA_JPK_HEX => 000001716254530e
```

Przykładowy klucz współdzielony w środowisku testowym wygenerowany dla kasy testowej:

```
PARAGON_SHARED_KEY_B64 => gZ5ff8DkMdgdLJGqC54Qh3PONDboF2Fp6D0VVLwd3oY=
PARAGON_SHARED_KEY_HEX => 819e5f7fc0e431d81d2c91aa0b9e108773ce3436ce176169e83d1554bc1dde86
```

2. Wyliczenia kodu autoryzacyjnego należy dokonać bazując na połączeniu bajtowym powyższych danych i kluczu współdzielonym w funkcji haszującej HS256:

```
PARAGON_SHA_HEX => f88e6e5ad956195072fc437954ab3ba05f0f3ff677f6a0129cb9effc83538ba0
PARAGON_NR_UNIK_HEX => 57544532303031303030303039
PARAGON_JPKID_HEX => 00038d7ea4c68004
PARAGON_DATA_JPK_HEX => 000001716254530e
```

```
PARAGON_VERIFY_DATA_HEX =>
PARAGON_SHA_HEX|PARAGON_NR_UNIK_HEX|PARAGON_JPKID_HEX|PARAGON_DATA_JPK_HEX
```

```
PARAGON_VERIFY_DATA_HEX =>
f88e6e5ad956195072fc437954ab3ba05f0f3ff677f6a0129cb9effc83538ba0575445323030313030303039
00038d7ea4c68004000001716254530e
```

Wynik użycia powyższego ciągu bajtów oraz klucza współdzielonego w funkcji haszującej HS256 w wyniku zwróci kod autoryzacyjny o wartości w postaci szesnastkowej:

```
PARAGON_VERIFY_CODE_HEX =>
db7d31eda33af45020f8a388b09737a5a836c6dee1be16e23d9e2c7441539a93
```

3. Utworzenie kodu weryfikującego polega na połączeniu bajtowo danych weryfikujących i kodu autoryzacyjnego:

```
PARAGON_QR_CODE_HEX =>
f88e6e5ad956195072fc437954ab3ba05f0f3ff677f6a0129cb9effc83538ba0575445323030313030303039
00038d7ea4c68004000001716254530edb7d31eda33af45020f8a388b09737a5a836c6dee1be16e23d9e2c7441
539a93
```

po przekodowaniu do formatu Base64URL:

```
PARAGON_QR_CODE_URL =>
-
I5uWtlWGVBBy_EN5VKs7oF8PP_Z39qASnLnv_INTi6BXVEUyMDAxMDAwMDA5AAONfqTGgAQAAAFxYlRTDtt9Me2jOvR
QIPijiLCXN6WoNsbe4b4W4j2eLHRBU5qT
```

Poniżej przykład wizualizacji kodu weryfikującego w postaci QR Code w wersji 6 (41x41 modułów):



#### A.10.7 Tworzenie QR kodu loteryjnego:

Poniżej przedstawiono opis formatu kodu loteryjnego wraz ze strukturą poszczególnych pozycji oddzielanych średnikiem:

a) numer unikatowy kasy	trzynastoznakowy numer unikatowy kasy fiskalnej składający się z trzech wielkich liter i dziesięciu cyfr, całość zapisana bez separatorów
b) numer identyfikacji podatkowej (NIP) podatnika	dziesięcioznakowy numer identyfikacji podatkowej podatnika składający się z samych cyfr
c) data sprzedaży	data w formacie yyyy-mm-dd, gdzie: dd - to dzień w postaci dwucyfrowej z wiodącym zerem mm - to miesiąc w postaci dwucyfrowej z wiodącym zerem yyyy – to rok w postaci czterocyfrowej
d) numer kolejny paragonu	ciąg znaków wskazujący kolejny numer paragonu, identyczny z wydrukiem na paragonie, jeśli na wydruku początek numeru zawiera zera wiodące to należy je uwzględnić, numer kolejny paragonu należy zapisać w cudzysłowie
e) łączna wartość sprzedaży brutto	kwota w złotych wraz z groszami oddzielonymi kropką
f) łączna wysokość podatku należnego	kwota w złotych wraz z groszami oddzielonymi kropką
g) typ formy płatności	liczba od 1 do 10 określająca jedną z form "Gotówka", "Karta", "Waluta obca", "Bon", "Czek", "Kredyt", "Przelew", "Voucher", "Mobilna" lub "Inna" wyliczonych w definicji „TTypFormy” w schemacie dokumentu JSON, podawany jest tylko jeden typ formy płatności, dla płatności mieszanej należy podać przeważającą formę, jeśli przeważających form płatności jest więcej to należy podać pierwszą w numeracji,
h) nazwa formy płatności	opcjonalny zestaw znaków alfanumerycznych opisujący nazwę formy płatności, nazwę formy płatności należy zapisać w cudzysłowie, znaki zakodowane w UTF-8 należy zapisać w postaci znaku zapytania

### Przykłady:

```
ABC0123456789;9999999999;2021-07-15;"F0001/1";123.00;23.00;1;  
ABC0123456789;9999999999;2021-07-15;"00001";123.00;23.00;2;"VISA"  
ABC0123456789;9999999999;2021-07-15;"1";123.00;23.00;3;"EUR"  
ABC0123456789;9999999999;2021-07-15;"F000001";123.00;23.00;1;"P?atno??  
gotówkowa w z?otych polskich"
```

Poniżej przykład wizualizacji kodu loteryjnego w postaci QR Code w wersji 5 (37x37 modułów):



## 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 kasy fiskalnej do repozytorium:

```
-----BEGIN CERTIFICATE-----
MIIFHDCCAwSgAwIBAgITOGAAAAjMj1WBXU6mOgABAAAACDANBgkqhkiG9w0BAQ0F
ADAWMRQwEgYDVQQDEw1S2FzeS1TdWJDQTAwFw0xNzA4MjIwNjMzMTNaGA8yMDky
MDcxMDEyMjcyOVowcgcxZCZAJBgNVBAYTA1BMMRQwEgYDVQQIEw1NYXpvd21lY2tp
ZTERMA8GA1UEBxMIV2Fyc3phd2ExHzAdBgNVBAoMFk1pbmlzdGVyc3R3byBGaW5h
bnPDs3cxIzAhBgNVBAStGKRlcGFydGFTZW50IEluZm9ybWFOeXphY2ppMR4wHAYD
VQQDExV0ZXN0LWuta2FzeS5tZi5nb3YucGwxKjAoBgkqhkiG9w0BCQEWG2luZm8u
ZS1kZWtsYXJhY2p1QG1mLmdvdi5wbDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCC
AQoCggEBAMvYVGj8Ynhy6P28bKj9MleA7+QXKCTPJZ4M6MIxIaqA4lodd9No+Ws
gRETVzEPiB8raL9n3uM+RBFwK2A4VvuAWuGzx2drkfMznpsVFLosQnadB1rjBCY5
G/pMX6eI7Bltx4XFYK/1cY1U+mFVc94Ryfyxy0ZWSD8IGV9n0AilDpRfIJB0u5a
3oquz8ZZGuWyU95KWBKRAD7SV2bpTlYWX4UHHte323HTYL3rDbKP73HAoyl0bSmS
vmB9MyNzWgBf73UOHmzXPqpURbLFnr+11TA0FA8kOylxtijyGMXpICOai7av2ofG
t65v0GJg5w1JuqWvkXfUYoyGUGUYaQsCAwEAAaOBrtCBqjAdBgNVHQ4EFgQUx7xK
j1TXCorOEa2hY/jdz6NKA0wHwYDVR0jBBgwFoAUBb+Partd6TV4PV1kTUrtJads
SdowWgYIKwYBBQUHAQEETjBMMEoGCCsGAQUFBzACHj5maWx1oi8vLy9zYXAtd21u
LTgyNi9DZXJ0RW5yb2xsL3Nhcc13aW4tODI2X2VLYXN5LVN1YkNBKDEpLmNydDAM
BgNVHRMBAf8EaJAAMA0GCSqGSIb3DQEBAQUAA4ICAQCKdUR2DhgieXUW+y2rgaE6
orWBPYmXveH2IPv0rPGzqdgUFcNH816YzDorEnOAvbRLB8BaoH+Wn/eElAQxqE5+
47VgScIUf4oNHwXnnf1R1XRoYcFZ/fBkIW2nfOK1C8y2vHtZG1QEyyVD/cxv7ubg
O1jfoYScsHv5DItStFUBclvg3xrFi2zG5ahblMwqCGrvGPKOxR9+mXGD+eoThBHE
P6aJF3Zu41mVwT/4cbSr5m3c77deEQ2CpQPGL874PiHy9omkjev9F5yoBzI7ypha
lyEIdbASU0UiUErjbs+hnnwOREv1bQQzQfS7qiKMBZTM4pzOv/Ro6f+0cBf7c16X
tHrEg1i/aNagKo34nFhUscQcUTCh3MsCKuVSZU3dbCdSLIvdOJIS5FLP+gr8LbQW
9uR/NgWJhYr/w06k6AOF+TaJw8eakv5ELDOuzhipqB63BuMSCGFZcUQ2bDhdc5gc
V9G1NgVEXmTee3fn89OQT7GrCwFzNxmAM6gJOMARYWl5Hmgr/pOb1MX5Vehgao
HpjoveMAacONbtiOwFMUyhPdCJmnLP67lokvGq7PDJ/DUBesApQVm91TM6QbWjda
nKGB6kYJ+7H5ESI8sp/nzjHXDZxeIPO7lOTItKdRW82kRcBR9TNDSS6rt5sI16LW
ONCJ2zprYt8XrNO7281jYA==
-----END CERTIFICATE-----
```

#### B.1.2 Certyfikat klucza publicznego kasy fiskalnej do podpisywania danych wytworzonych przez kasę oraz szyfrowania klucza szyfrującego przesyłanych komend wysyłanych do kasy:

```
-----BEGIN CERTIFICATE-----
MIICfjCCAeegAwIBAgIQzILUrkd2iqNCxC1ZrG7UBDANBgkqhkiG9w0BAQsFADAU
MRIWEAYDVQQDEw1NRiB1LUthc3kwHhcNMTCxMTE1MDk0MDAzWhcNMTCxMTE1MDk0
MDAYWjAYMRYwFAYDVQQDEw1aVEUxNzAxMDAwOTAxMIIIBIjANBgkqhkiG9w0BAQEF
AAOCAQ8AMIIBCgKCAQEAwy3Kc3oTipU451OIX6K3rruFY83vMqYJqwoWzrRVmZn8
5cNHSwoa/f96AW0/akADbJ3uo7U8oWhTF/rj8xIds4uimrN1YiPFmbXAMaeRQDdbB
a/qvI5SRQtK9Bmse7KyspIFXVfEWP17OkDiBEZ/n+NC6ERzzKkxA3gMRQFGpHUSQ
2+EOi7kykPGi1f8Yh/2czd+FBvyrp8oSjyX951DdCsQG+rIwlz9p8PeoFwUggwhb
2tM148U3nD9gZGXLUgOMBZ3nJ9U3fHvdi7XCpvn0PqvTSZLNL45yqXETu6bAZWB5
Ab4q5EVV41unrjnJqc3fPD2OLXpINIssg6uqyTVCZQIDAQABO0kwrZBFBgNVHQEE
PjA8gBDGD6f6PwMBTV/bee5Lr1c9oRYwFDESMBAGA1UEAxMJTUUYgZS1LYXN5ghCz
wtV5n24/mUCWe9d7xth6MAOGCSqGSIb3DQEBCwUAA4GBAKRTZFPJY5ObY4VVGpJL
14Xb2JNtWpNXdwPs3N8I2r1iGc0dxyq8R4C9X125G0LGXXXTMDtnE1k+xmCk0aU
6bj2xpfezLhW6i1+mmHTB/2+JhsKp5oRRTXg8SpH5G1vwQI9ek9B/bYvn72nKrUa
Tp3PZsmCNqmlD0VayfTrhZiS
-----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": [ "MII FHDCCAwSgAwIBAgITogAAAAj m j 1WBXU6mOgABAAAACDANBgkqhkiG9w0BAQ0FADAWMRQwEgYDVQQDEwt1S2FzeS1TdWJDQTAgFw0xNzA4MjIwNjMzMTNaGA8yMDkyMDcxMDEyMjcyOVowcgxwCzAJBgNVBAYTA1BMMRQwEgYDVQQIEwtNYXpvd21lY2tpZTERMA8GA1UEBxMIV2Fyc3phd2ExH2AdBgNVBAoMFk1pbmlzdGVyc3R3byBGaW5hbnpD s3cxIzAhBgNVBAsTGRlrcGFydGFtZW50IEluZm9ybWFOeXphY2ppMR4wHAYDVQQDExV0ZXN0LWUta2FzeS5tZi5nb3YucGwxKjAoBgkqhkiG9w0BCQEWG2luZm8uZS1kZWtsYXJhY2plQG1mLmdvdi5wbDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCAQoCggEBAMvyVXGj8YNhy6P28bKj9MleA7+QXKCTpJZ4M6MIxiaqA4lodd9No+WsgRETVzEPiB8raL9n3um+RBFwK2A4VvuAWuGZx2drkfMZNpSVFLOsQnadB1rjBCY5G/pMX6eI7Bltx4XFYK/1cy1U+mFVc94Ryfyxyy0ZWSd8IGV9n0AilDpRfIJBOu5a3oquz8ZZGuWyU95KWBKRAD7SV2bpTlYWX4UHHte323HTYL3rDbKP73HaoylObSmSvmB9MyNzWgBf73UOHmzXPpPuRbLfnR+llTA0FA8kOylxtijygMXpICOai7av2ofGt65v0GJg5w1JuqWvkQXFUYoyGGUYaQsCAWEAAa0BrTCBqjAdBgNVHQ4EFgQUx7xKj1TXCorOEsa2hy/jdz6NkA0wHwYDVR0jBBgwFoAUBb+Partd6TV4PV1kTurtJadsSdowWgYIKwYBBQUHAQEETjBMMEoGCCsGAQUFBzAchj5maWx1oi8vLy9zYXAtd2luLTgyNi9DZXJ0RW5yb2xsL3Nhcc13aW4tODI2X2VLYXN5LVN1YknBKDEpLmNydDAMBGNVHRMBAF8EAJAAMA0GCSqGSIb3DQEEDQUAA4ICAQCkdUR2DhgieXUW+y2rgaE6orWBPyMxveH2IPv0rPGzqdgUFcNH816YzDorEnOAvbRLB8BaoH+Wn/eE1AQxqE5+47VgSciUF4oNHwXnnf1R1XRoYcFZ/fBkIW2nfOK1C8y2vHtZG1QEyyVD/cxv7ubgOiJfOYScsHv5DItStFUBclvg3xrFi2zG5ahb1MwqCGrvgPKOxR9+mXGD+eoThBHEP6aJF3Zu41mVwT/4cbSr5m3c77deEQ2CpQPL874PiHy9omkjev9F5yoBzI7yphalyEIdbASU0UiUErjbS+hnnWERrVlBQQzQfS7qiKMBZTM4pzOv/Ro6f+OcbF7c16XtHrEg1/aNagKo34nFhUscQcUTCh3MsCKuVSZU3dbCdSLIvdOJIS5FLP+gr8LbQW9uR/NgwJhYr/w06k6AOF+TaJw8eakv5ELDOuzhipqB63BuMSCGFZcUQ2bDhdc5gcV9G1NgVEXmToee3fn89QQTc7GrCwFzNxmAM6gJMARyW15Hmgr/pOb1MX5VehgaoHppjoveMAacONbtioWfMUyhpDcJmnlP671okvGq7PDJ/DUBespAqVm91TM6QBWjdanKGB6kJY+7H5ESI8sp/nzjHXDZxeIPO71OTItKdRW82kRcBR9TNDSS6rt5sI16LW0NCJ2zprYt8XrNO7281jyA==" ] }
```

## B.2.2 Pełna postać chronionego nagłówka podpisu komend przesyłanych do kasy fiskalnej zakodowana w Base64URL:

```
eyJhbGciOiJSUzI1NiIsIng1YyI6WyJNSU1GSERDQ0F3U2dBd01CQWdJVE9nQUFBQWptaajFXQ1hVNm1PZ0FCQUFBQUNEQU5CZ2txaGtpRz13MEJUBTBGQURBV01SUx dFZ11EVlFRREV3dGxTMkZ6ZVMxVGRXSkrRREVEFnRncweE56QTRNaK13TmPNek1UTmFHQTTh5TURreU1EY3hNREv5TWpj eU9Wb3dnY2d4Q3pBSkJnTlZCQVlUQWxCTU1SUx dFZ11EVlFRSUV3dE5ZW5Wb2ZJsbFkydHBAVEVSTUE4R0ExVUVCeE1JVjJGeWMzcGhkMkV4SHpBZEJnTlZCQW9NRmsxcGJtbHpkR1Z5YzNSM2J5QkdHvZVoYm5QRHMzY3hJekFoQmdOVk1JbC1RHa1JsY0dGeWRHRnRaVzUwSUVsdVptOX1iV0YwZVhwaFkyChBNUjR3SEFZRFZRUURFeFYWw1hOMExXVXRhMkZ6ZVM1dFppNW5iM1l1Y0d3eEtqQW9CZ2txaGtpRz13MEJDUUVXRzJsdVptOHVaUzFrWld0c1lYSmhZMnBsUUCxbUxtZH2kaTV3YkRDQ0FTSxdEUV1KS29aSW h2Y05BUUVVCQ1FBRGdnRVBBRENDQVFvQ2dnRUJBTXZ5VlhHajhZTmh5N1AyOGJLa1JnbnBVBnYtRWETdDfBKWjRNNk1JeG1hcUE0bG9kZD10bytXc2dSRVRWekVQaUI4cmFMOW4zdU0rUk1JGd0syQTRWdnVBV3VHwngyZHRjZk1abnBTvkZMT3NRbmFkQjFyakJDWTVHL3BNWDZ1StdCbHR4NFhGWUsvMWNZMVUrbUZWYzZk0UnlmeXh5eTBaV1NEOE1HVj1uMEFpbERwUmZJSkJPdTVhM29xdXo4W1pHdVd5VTtk1S1dCS1JBRDdTVjJicfRswVdYnFYIaFR1MzIzSFRZTDNjYRGJLUDczSEFveWxPY1NtU3ZtQj1NeU56V2dCZjczVU9IbXpYUHFwVJiTEZuUitsbFRBMEZBOGTPeWx4dG1qewdNRWHBJQ09haTObdljJvZkd0NjV2MedKZzV3bEp1cVd2a1FYRLV5b31HR1VZYVVFzQ0F3RUFBYU9C1RDQnFqQWRCZ05WSFE0RUZnUUV4N3hLa1JFUWENvck9FeGEyaFkvamR6Nk5rQTB3SHdZRFZSMGpCQmd3Rm9BVUJiK1BhcnRkN1RWnFBWMWtUVXJ0SmFkc1Nkb3dXZ11JS3dZQk1RVUhbUUVFVGpCTU1Fb0dDQ3NHQVFRkKJ6QUNoajVtYVd4bE9pOHZMeT16WVhBdGQybhVMVGD5Tmk5RfPySjBSVzV5YjJ4c0wzTmhjQzEzYVc0dE9ESTYm1ZMwVhONuXWTjFZa05CS0RfcExtTn1kREFNQmdOVkhSTUJBZjhFQWpBQU1BMEdDU3FHU01iM0RRRUJEUUVVBQTRJQ0FRQ2tKVVIyRGhnaWV4VVcreTJyZ2FFNm9yV0JQeU14dmVIMk1QdjByUed6cWRnVUZjTkg4MTZzekRvckVuT0F2Y1JMqjhcYw9IK1duL2VFbEFReHFFNs0N1ZnU2NjVUY0b05Id1hubmZsUjFYUm9Zy0ZaL2ZCa0lXMM5mT0sxQzh5MnZIdFpHMFVFeXlWRC9jeHY3dWJnT21KZk9ZU2NzSHY1RE1U03RGVUJjbHbZmM3hyRmkyekc1YWhibE13cUNHcnZnUETPeFi5K21YR0QrZW9UaeJIRVA2YUpGM1p1NDftVndULRzY1NynW0zYzc3ZGVFTUTJdCFFQR0w4NzRQaUh5OW9ta2pldj1GNXlvQnpJN31waGFseUVJZGJBU1UwVWlVRXJqY1MraG53MFJFc1YxY1FRelFmUzdxaUtNqlpUTTRwek92L1JvNmYrT2NCZjdjMTZyDEhyRWdsas9hTmFnS28zNG5GaFVzY1FjVVRDaDNNc0NLdVzTW1UzZGJDZFNMSXZkb0pJUZVGTFARZ3I4TGRJRVz11U9OZ3dKaF1yL3cwNms2QU9GK1RhSnc4ZWFRdjVFTERPdXpoaXBxQjYzYzQnVNU0NHR1pjVVEYyKRoZGM1Z2NwOudsTmdWRVhtVG91ZTNmbjg5T1FUQzdHckN3RnpOeHdBTTZnSjBNQVJ5V2w1SG1nci9wT2IxTVg1VmVoZ2FvSHBqb3Z1TUFhY090YnRpt3dmTVV5aFbkQ0ptbXkxQNjdsb2t2R3E3UERKL0RVYmVzcEFxVm05MVRNN1FiV2pkYW5LR0I2a0pZKzdINUVtSThzcC9uempIWGrAeGVJUE83be9USXRLZfJXODJrUmNCUj1UTkRTUzZydDVzSTE2TFcwTknKNmpwcl10OFhyTk83Mjhsan1BPT0iXX0
```

## B.2.3 Pełna postać przykładowych danych komend przesyłanych do kasy fiskalnej zakodowana w Base64URL:

```
eyJhdHRyaWw1dGVzIjpw7ImNwZFN1cnZpY2VOYW11Ijois0ZEIn19
```

## B.2.4 Pełna postać przykładowych danych do podpisu komendy przesyłanych do kasy fiskalnej zakodowana w Base64URL:

```
eyJhbGciOiJSUzI1NiIsIng1YyI6WyJNSU1GSERDQ0F3U2dBd01CQWdJVE9nQUFBQWptaajFXQ1hVNm1PZ0FCQUFBQUNEQU5CZ2txaGtpRz13MEJUBTBGQURBV01SUx dFZ11EVlFRREV3dGxTMkZ6ZVMxVGRXSkrRREVEFnRncweE56QTRNaK13TmPNek1UTmFHQTTh5TURreU1EY3hNREv5TWpj eU9Wb3dnY2d4Q3pBSkJnTlZCQVlUQWxCTU1SUx dFZ11EVlFRSUV3dE5ZW5Wb2ZJsbFkydHBAVEVSTUE4R0ExVUVCeE1JVjJGeWMzcGhkMkV4SHpBZEJnTlZCQW9NRmsxcGJtbHpkR1Z5YzNSM2J5QkdHvZVoYm5QRHMzY3hJekFoQmdOVk1JbC1RHa1JsY0dGeWRHRnRaVzUwSUVsdVptOX1iV0YwZVhwaFkyChBNUjR3SEFZRFZRUURFeFYWw1hOMExXVXRhMkZ6ZVM1dFppNW5iM1l1Y0d3eEtqQW9CZ2txaGtpRz13MEJDUUVXRzJsdVptOHVaUzFrWld0c1lYSmhZMnBs
```

UUcxbUxtZHZkaTV3YkRDQ0FTSXdEUv1KS29aSWh2Y05BUUVCQ1FBRGdnRVBBRENDQVFvQ2dnRUJBTXZ5V1hHajhZTmh5N1  
AyOGJLaJl1NbGVBNytRWETdDfBKWjRNNk1JeG1hcUE0bG9kZD1ObytXc2dSRVRWekVQaUI4cmFMOW4zdU0rUkJGd0syQTRW  
dnVBV3VHWngyZHJrZk1abnBTvkZMT3NRbmFkQjFyakJDWTVHL3BNWDZ1STdCbHR4NFhGWUsvMWNZMVUrbUZWYzk0UnlmeX  
h5eTBaV1NEOE1HVjluMEFpbERwUmZJSkJPdTVhM29xdXo4WlpHdVd5VTk1S1dCS1JBRDdTVjJicFRsWdYNFV1aFR1mZ1z  
SFRZTDNyrGJLUDczSEFveWxPY1NtU3ZtQj1NeU56V2dCZjczVU9IbXpYUHFwdVJiTEZuUitsbFRBMEZBOGTPeWx4dG1qeW  
dNWHBJQ09haTdhdjJvZkd0NjV2MEDKZzV3bEp1cVd2a1FYR1V5b31HR1VZVVFzQ0F3RUFBYU9Cc1RDQnFqQWRRCZ05WSFE0  
RUZnUVV4N3hLaJFUWENvck9FeGEyaFkvamR6Nk5rQTB3SHdZRFZSMGpCQmd3Rm9BVUJiK1BhcnRkN1RWNFBWMWtUVXJ0Sm  
Fkc1Nkb3dXZ11JS3dZQkJRvUhbUUVFVGpCTU1Fb0dDQ3NHQVFVRk1J6QUNoajVtYvd4bE9pOHZMeT16WVhBdGQybHVMVGd5  
Tmk5RFpYsJBSVzV5YjJ4c0wzTmhjQzEzYVc0dE9ESTJYMLZMWVhONuXWTjFZa05CS0RfcExtTnlkREFNQmdOVkhSTUJBZj  
hFQWpBQU1BMEdDU3FHU01iM0RRRUJEUUVBQTRJQ0FRQ2tkvVIyRGhnaWV4VVcreTJyZ2FFNm9yV0JQeU14dmVIMklQdjBy  
UEd6cWRnVUZjTkg4MTZZekRvckVuT0F2Y1JMQjhCYW9IK1duL2VFbEFeReHFFNSs0N1ZnU2NJVUy0b05Id1hubmZsUjFYUm  
9ZY0ZaL2ZCa01XMM5mT0sxQzh5MnZIdFpHMFVFeX1WRC9jeHY3dWJnT21KZk9ZU2NzSHY1RE10U3RGVUJjBhZnM3hyRmky  
ekc1YWhibE13cUNHcnZnUETPeFI5K21YR0QrZW9UaEJIRVA2YUpGM1p1NDFtVndULzRjY1NynNW0zYzc3ZGVFUTJDcFFQR0  
w4NzRQaUh5OW9ta2pldj1GNXlvQnpJN31waGFseUVJZGJBUIUwVWlVRXJqYlMraG53MFJFc1YxYlFRelFmUzdxaUtNqlpU  
TTRwek92L1JvNmYrT2NCZjdjMTZyDEhyRWdsas9hTmFnS28zNG5GaFVzY1FjVVRDaDNNc0NLdVZTW1UzZGJDZFNMSXZkb0  
pJUzVGTFArZ3I4TGRVZl1U19OZ3dKaF1yL3cwNms2QU9GK1RhSnc4ZWFrdjVFTERPdXpoaXBxQjYzQnVNU0NHR1p1jVVEy  
YkRoZGM1Z2NWOUdsTmdWRVhtVG91ZTNmbjg5T1FUQzdHckN3RnpOeHdBTTZnSjBNQVJ5V2w1SG1nci9wT2IxTVg1VmVoZ2  
FvSHBqb3Z1tUFhY090YnRpT3dmTVV5aFBkQ0ptbkxQNjdsb2t2R3E3UERKL0RVYmVzcEFxVm05MVRNN1FiV2pkYW5LR0I2  
a0pZKzdINUVTSThzcC9uempIWGRaeGVJUE83bE9USXRLZfJXODJrUmNCUj1UTkRTUzZydvZSTE2TFcwTknKNmpwcll0OF  
hyTk83Mjhsan1BPT0iXX0.eyJhdHRyaWJldGVzIjpp7ImNwZFN1cnZpY2VOYW11Ijois0ZEIn19

## B.2.5 Pełna postać podpisu przykładowej komendy przesyłanej do kasy fiskalnej:

H1Khuau2-ZLYoBip8ed2J7Js5HHOr1b96v1h-udv0OUjm5wWc0wSPCTqDe4rYe9R1qeG8Z4xXbPJE287o01bYtqe-  
VuYzL5sDNHvLi1RQBZqTTHpWCNOw3mfM6vrKX\_EXJ1wNeGu8aavozYKfVxIHWNZfZq6Ff1dCiAkxxXu35dEyBYGIZHbEz3  
4AJu8KOY-  
O24ZV6qr2tERB\_SUYOgS4ZgigvY9loCdb\_Vuui2sKYTMW55bT\_BBL0gdm8yp7M2RNNCHitrGgEsen5otmpsgzh-  
hQAP7rZYZw0510gLU8xAaN\_2RMIaz1vNluAM4o8o7mxe-FFv1RLSP1zoMU-NGHw

## B.2.6 Pełna postać obiektu JWS przykładowej komendy przesyłanej do kasy fiskalnej:

eyJhbGciOiJSUzI1NiIsIng1YyI6WyJNSU1GSERDQ0F3U2dBU01CQWdJVE9nQUFBQWptaJfFXQ1hVNm1PZ0FCQUFBQUNEQU  
5CZ2txaGtpRz13MEJBUTBGQURBV01SUXdFZ11eV1FRREV3dGxTmkz6ZVMxVGRXSkrRREVEFnRncweE56QTRNak13TmPnek1U  
TmFhQTh5TURreU1EY3hNREv5TWpjEU9Wb3dnY2d4Q3pBSkJnT1ZCQV1UQWxCTU1SUXdFZ11eV1FRSUV3dE5ZWHB2ZDJsbF  
kydHBAVEVSTUE4R0ExVUVCeE1JVjJGeWMzcGhkMkV4SHpBZEJnT1ZCQW9NRmsxcGJtbHpkR1Z5YzNSMJ5QkdhVzVoYm5Q  
RHMzY3hJekFoQmdOVkhJBc1RHa1JsY0dGeWRHRnRaVzUwSUVsdVptOX1iV0YwZVhwaFkyChBNUjR3SEFZRFZRUURFeFYwW1  
hOMExXVXRhMkz6ZVM1dFppNW5iM1l1Y0d3eEtqQW9CZ2txaGtpRz13MEJDUUUVXRzJsdVptOHVaUzFrWld0c1lYSmhZMnBs  
UUcxbUxtZHZkaTV3YkRDQ0FTSXdEUv1KS29aSWh2Y05BUUVCQ1FBRGdnRVBBRENDQVFvQ2dnRUJBTXZ5V1hHajhZTmh5N1  
AyOGJLaJl1NbGVBNytRWETdDfBKWjRNNk1JeG1hcUE0bG9kZD1ObytXc2dSRVRWekVQaUI4cmFMOW4zdU0rUkJGd0syQTRW  
dnVBV3VHWngyZHJrZk1abnBTvkZMT3NRbmFkQjFyakJDWTVHL3BNWDZ1STdCbHR4NFhGWUsvMWNZMVUrbUZWYzk0UnlmeX  
h5eTBaV1NEOE1HVjluMEFpbERwUmZJSkJPdTVhM29xdXo4WlpHdVd5VTk1S1dCS1JBRDdTVjJicFRsWdYNFV1aFR1mZ1z  
SFRZTDNyrGJLUDczSEFveWxPY1NtU3ZtQj1NeU56V2dCZjczVU9IbXpYUHFwdVJiTEZuUitsbFRBMEZBOGTPeWx4dG1qeW  
dNWHBJQ09haTdhdjJvZkd0NjV2MEDKZzV3bEp1cVd2a1FYR1V5b31HR1VZVVFzQ0F3RUFBYU9Cc1RDQnFqQWRRCZ05WSFE0  
RUZnUVV4N3hLaJFUWENvck9FeGEyaFkvamR6Nk5rQTB3SHdZRFZSMGpCQmd3Rm9BVUJiK1BhcnRkN1RWNFBWMWtUVXJ0Sm  
Fkc1Nkb3dXZ11JS3dZQkJRvUhbUUVFVGpCTU1Fb0dDQ3NHQVFVRk1J6QUNoajVtYvd4bE9pOHZMeT16WVhBdGQybHVMVGd5  
Tmk5RFpYsJBSVzV5YjJ4c0wzTmhjQzEzYVc0dE9ESTJYMLZMWVhONuXWTjFZa05CS0RfcExtTnlkREFNQmdOVkhSTUJBZj  
hFQWpBQU1BMEdDU3FHU01iM0RRRUJEUUVBQTRJQ0FRQ2tkvVIyRGhnaWV4VVcreTJyZ2FFNm9yV0JQeU14dmVIMklQdjBy  
UEd6cWRnVUZjTkg4MTZZekRvckVuT0F2Y1JMQjhCYW9IK1duL2VFbEFeReHFFNSs0N1ZnU2NJVUy0b05Id1hubmZsUjFYUm  
9ZY0ZaL2ZCa01XMM5mT0sxQzh5MnZIdFpHMFVFeX1WRC9jeHY3dWJnT21KZk9ZU2NzSHY1RE10U3RGVUJjBhZnM3hyRmky  
ekc1YWhibE13cUNHcnZnUETPeFI5K21YR0QrZW9UaEJIRVA2YUpGM1p1NDFtVndULzRjY1NynNW0zYzc3ZGVFUTJDcFFQR0  
w4NzRQaUh5OW9ta2pldj1GNXlvQnpJN31waGFseUVJZGJBUIUwVWlVRXJqYlMraG53MFJFc1YxYlFRelFmUzdxaUtNqlpU  
TTRwek92L1JvNmYrT2NCZjdjMTZyDEhyRWdsas9hTmFnS28zNG5GaFVzY1FjVVRDaDNNc0NLdVZTW1UzZGJDZFNMSXZkb0  
pJUzVGTFArZ3I4TGRVZl1U19OZ3dKaF1yL3cwNms2QU9GK1RhSnc4ZWFrdjVFTERPdXpoaXBxQjYzQnVNU0NHR1p1jVVEy  
YkRoZGM1Z2NWOUdsTmdWRVhtVG91ZTNmbjg5T1FUQzdHckN3RnpOeHdBTTZnSjBNQVJ5V2w1SG1nci9wT2IxTVg1VmVoZ2  
FvSHBqb3Z1tUFhY090YnRpT3dmTVV5aFBkQ0ptbkxQNjdsb2t2R3E3UERKL0RVYmVzcEFxVm05MVRNN1FiV2pkYW5LR0I2  
a0pZKzdINUVTSThzcC9uempIWGRaeGVJUE83bE9USXRLZfJXODJrUmNCUj1UTkRTUzZydvZSTE2TFcwTknKNmpwcll0OF  
hyTk83Mjhsan1BPT0iXX0.eyJhdHRyaWJldGVzIjpp7ImNwZFN1cnZpY2VOYW11Ijois0ZEIn19.H1Khuau2-  
ZLYoBip8ed2J7Js5HHOr1b96v1h-udv0OUjm5wWc0wSPCTqDe4rYe9R1qeG8Z4xXbPJE287o01bYtqe-  
VuYzL5sDNHvLi1RQBZqTTHpWCNOw3mfM6vrKX\_EXJ1wNeGu8aavozYKfVxIHWNZfZq6Ff1dCiAkxxXu35dEyBYGIZHbEz3  
4AJu8KOY-  
O24ZV6qr2tERB\_SUYOgS4ZgigvY9loCdb\_Vuui2sKYTMW55bT\_BBL0gdm8yp7M2RNNCHitrGgEsen5otmpsgzh-  
hQAP7rZYZw0510gLU8xAaN\_2RMIaz1vNluAM4o8o7mxe-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 kasy fiskalnej zakodowana w Base64URL:

```
eyJlbmMiOiJBMTI4Q0JDLUhTMjU2IiwiaWxnbG9kaWkiOiJBMjU0MmQ0YU0MGY2OGFhMzQyYzQyZDU5YWZ2ZWQ0MDQsQ049TUyZS1LYXN5In0
```

### 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:

```
hsileeNctbhjLR60diiBZ7U6kFPqzU3Pc6DDneWQncoblSZTEv7bsqlAv-QDmR3liTRWuOzcYgduuDwAbmz1xwFI3cVOBVeQBIRAgBtFuX_xEpmBdXAZrMXY37dX1SoCu0Rno0HnBs5bDsnuX4TiZe4jEKNggOHPf88mErGW26BwU1WbvFlyj9HhEfI2H46D-y29djnkXI_7MpM994hTRTq6JGPE-9m2ZjOtU6Yw4Ty1kRmUeK9iWQkdIKJHy6TGHd9qnh03Odcyu_DPHJnbmhbB5AaT5yAD9kGciticMC4PUDju3_qcHgiwPo94s4m8jlRBo7M8jldVJvvOpU28ng
```

### B.3.5 Przykładowa wartość wektora inicjującego zakodowana w Base64URL:

```
1kE2bOFzIkxFKpFObeCIww
```

### B.3.6 Przykładowa wartość zaszyfrowanych danych zakodowana w Base64URL:

```
PoikvBdxUJgfyLYAr9tyYOediKO7oQTdEUAPcW6QSKcmPCule3TKtk-ypKVXtMrJvJd0uRweKpGvx_uYU-IEENg_Dh4o1Q8A7PQ5D2of_qZCW9pXi-DCnSE6wtPAggUCYKbeXWdJj8S-jzDrx6z-nu9ie25f4pw5EnRsA7pGLfK0MIao6oaEDcrmmVm646tf9ezCKWACZX3HyOLpN0Q6PtovXPrigV07V7bQ-MCSsj0AoxKmdjv1YNUGblU-WxAPaSKI-7L54YgkMR3ob9eRUZNAHIs2jNTAPCuOML2zyc04Qz0saa5h4Q8zuG9W00aAKJI0CQqHWH3FRqMSOINoXkbay5QVzrZ-Kbbyf6LusspxI5aibMNPBucrXkodWWszFc3th_jPQ4r6YZB-RzAO_inRNQqFsTJneP00Jjkd_0SfkanIcU5sjsbBH9G6Vry5ECtOr5Me3IPcPIXGcdGqBO3TaklpXCHj1-qOVWeDbZ5YxxcQfhhsVfyOtM5ei4yIkNrXICWTagie-kmlBnF2mtjm0iv1YCuca8a44RcD5BdpaGor1lEeizjAPBYi8QWtXSU5KDNXeC0gN4zUNPuccuzPzPk1QaQ7Mt8h9rgzOB9cDIoOVLFe_hgAq4fn4xG25QKiTVLcOd8PKB8oA4S_f_4nO6cO7zPx_DP4Tu9cBE4M7DJxjZY2d4A21w-HGx4YukTwaX_w9bcfhKSYNKFYKky6aa180Z4JaMW902NHhleG0ER6c3Y7JFVCyWw9CXIJKfT6BvY8q5CdVFrW4_CF38-AIuXKRU-HizqOjM8TrOErY0hOu5443u7KQedT4pyZWYAzVai4MK1Kj5jLUyLOZ4inHPGqTdT3AaTFX5FDwOM_ivUjwbwptwk7MC8euoHTXQj02yB4p-TpdGsX4qO9BHUG3fgFPNbrSFDWn80KNV8C6VmOBaTNZ8MTLkSB3lQhMEC3mYCeB3fMiy_E79NmePsttGKCCcPotwFv4-qW-cGpl1-T_Z5P8_EqN6ss0400w4PlAq7Q0mJdmpaf0Kqnt-gf7KcwYG5Vq3_5DJCXBu0DSigydQyBU15E4OUH8X02sSNGetmiLuAmkrOV6dysu0bJVlu81OeyN1bQv7cN_qiL1DQbt_gsquhspXI8h1lMHYxQ2ygIrDm-UsMWdj120jBiY7z1FOVHJxbdna_fZN1dA1Tkg1L-vuHdJprEqcBpnjoOkE8wWpGwalreKi4DHOf3nOm_JEfpdxh07DfPpG6fx-krxkkf2_ybyg773NSATjBBOeRbZo1GrI8Ic7fdHIpfYyxhcCU9zLRJlWg0pjjzHJIxo-prwhW90BHqjFYpnoYnMEvs6SXCES4OPFS2mXw8PFr51PkRkYfkSewraTnWwnw3WFqfiOBjCKEdpfbF5axbhT-rg5BTHUi-VvPT3MrPInRy-p0BijKLB9y8Z0R8DLX07ZoWajJOS55XWdZ8hdPWCoc509odTRcUUEvi-uqWQJMQcmZCiRwhsU9v051T2t-aaWhLL9Q-e58M_pbM0HfLU_ea8jf5Vaf_bXht18BgZi2dvt-vddwy09kVB-jQkzJmvLe2_hcWlRkQ2X7-EmDlLt7iad7Nn20861a21DKdQjbVDXbXZVF2cp9jdtg_Ls5rMHgcTU2ORJG4jzchCJi4qRVUsa9t4znWEbtgLS64T64qScxgcNl8gRzFwLxCGG_6UqFbUgqIkHfWj1zw572rA_WJGB0UVBPIrXltRzHKC2bhc85KTKsv00qIHVklDCrF47rTz-Dx54IzGhsW_BRc5az1cshoNaABAuW2V_hS-
```



DoMsLLpli5qfiRBw\_WrhBbZjSFUn1AILl2CeeJss9pZ9ZNe2aPWjd75vTknxURE9mT8-jh1v3DjTs8\_eBQWBT9-BdEy0AEPfKtZgZ4QHxLj\_v3VlR3F99Fu2dj977LJlSiOnXMJlZLhpyPrTnwjFuPavNe8HmsvjgysyMgHt\_Lu\_bx4Oqk0P-MrfwbOXsSjjJivvyjS-P5g2p194L5sGonvh2Jr15Q\_aBHgKIiyakYxey3zbYcJfS1RoV5MSg2P\_MJYoJI1Ffm\_AAqUCRBKldYQLhTIilQzjOb5\_TZ8\_3za2pjH7xLWC91egX6d8tFqpyXlXaLchQUObnKJgqubHejatfBZU1BM82BmyigS8PseeDPC2Cj6tg34BKra7q3I20H5XQrDFbyho7H01Y55jb5UvLsSmHdIAgoFOh7iSnki4iYrkQrPVCWlvbb2KvhnHotCeYFsdxfLa43y21StuYBoNP5Buu-cz4jovMwQyMZSgU7ojx8ae2RkrGzJxD7p943U9w21mTNJDDbPIdUg\_QaPEi9PaxH-7mB72wazfDtNEB5S-MBQpIdIazD6QB4skD3b2h49rXggBYjyT59yxPrHkhDy\_bo352BMqRxokfjZlKdirVHS4cLy0SXty06E3TCUt5\_tfJ-0bo-kT-Dj4wR7k04Yzd6thPKi6AQJKPZz1ARZEguGu2-9p46AVZd1mCWDTCNlEDsL\_gvb7GJFIknZt215voxI6WMjzyLio-jyGGMBHgoQSsd246LQY-LNJCKJa8a18MbNqpRRBZBdmPnvWUoK8eTQ6Nkt3mZQNQcrqvrRtrQ\_OECAScwcDoSct4d6C-\_w-I77E\_FNVEvdnVDbWMKJMc9\_F99qfBTYPL9AL\_Cx89SyOJOReWiWLC1GoPjw91MV3K5fxX\_1EXapNyrfWW3jo90jeVKr42heWacLTFMCCdNXyMuwVT8NttBRpkJwee-dae0vqomTHQqal\_gyqc01J95pAbATYkjAGKYfnImS6c4fEP2VD6EAro7WQzmM3IzEvFF9dWMD5RnuwMKB2HECCxvSKkc03fQC\_xMu0EgqliqkE\_QyT2aJpPTyf7pB-zSy2C6gYt5DVlnc8ea6i1ElPSwdaS9uzMVHE0Hqrfq6Px23uZxcBVYcNiSXCguUaAyLmOf6YsXK7hHHx\_uoVLv1B3SDME4GDI8bp0PisW\_LTViAlvjktSRm0021PqCu58mHgEP9h0jM-s582vcw9orSdi63q2OvK6dBojawORWZ\_t-MWTc7r0-y\_766bekLrktYVHJkfbUFM4keDOEP9I4M09mKHnCLGCMFok5z1JDxtN71M\_HXXpwJJ4hrCJIUOT2Z00glIRGE1YhMqILfhDJNELhhlytsqSy32RTOz1qbIoglTilpQOwxBGPUVazwcXTXOFtHrd6074gpGHQkhWurIcYxLcm4rD8gja97iLm2J2keD5bHbgCryyjb8BotGZTA\_oj1VihMe9P6k69QwUCOo-MgQSVYgq6IXS3EbfdOBbWobjRtytaY46YXY9-4OREk6B1Hes8Xn5bG-O7w0xJ4OKyKAES6iyRyFktr\_RP6rBMOfpOLz90G68hX8ZwOVGWUIE8e8J5gB3XbpPYbX51Wi2kx-7CYXRDbYUii361338tHks-aXcZ8Hb7h-D8bGxObateUSBR3Ka-6gCDMYHdSYijxHhhBxTTyrAcgzrLKD7CvV4PwQ5cUVVx1qE7VeosUMIS9BotMXQmr12C0k1ulAHW8JspE43Mpced0NuUUAxib9EVuNkxK1dJLXwtv8OMsugQ4\_bOWQMMCPz-15o5ciPGht0YkDAZJoODYMDpmZvMJLYb3QBzBbt2kgag8Si0a\_9NZrktTLOuRM

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

eyJlbmMiOiJBMTI4Q0JDLUhTMjU2IiwiaWxnxIjoilUlNBmV8lIiwiaWl2IjoilY2M4MmQ0YWU0MGY2OGFhMzQyYzYyZDU5YWZ2ZWQ0MDQsQ049TU9yZS1LYXN5In0

odzwierciedlenie AAD w postaci szesnastkowej:

65794a6c626d4d694f694a424d54493451304a444c5568544d6a5532496977695957786e496a6f69556c4e424d5638314969776961326c6b496a6f6959324d344d6d5130595755304d4759324f4746684d7a5179597a51795a44553559574d325a5751304d44517351303439545559675a53314c59584e35496e30

### 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:

```
eyJlbmMiOiJBMTI4Q0JDLUhTMjU2IiwiaWYwbnIjoUlNBmV81Iiwia2lkIjoiy2M4MmQ0YWU0MGY2OGFhMzQyYzQyZDU5YW  
M2ZWQ0MDQsQ049TUyGzS1LYXN5In0
```

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

**65794a6c626d4d694f694a424d54493451304a444c5568544d6a5532496977695957786e496a6f69556c4e424d5638  
314969776961326c6b496a6f6959324d344d6d5130595755304d4759324f4746684d7a5179597a51795a4455355957  
4d325a5751304d44517351303439545559675a53314c59584e35496e30**

użyty 16 bajtowy wektor inicjujący IV [B.3.5](#):

postać Base64URL:

```
lkE2b0FzIkxFKpFObeCIww
```

postać szesnastkowa wykorzystana w funkcji HMAC:

**9641366ce173224c452a914e6de088c3**

zaszyfrowane dane użyte do wyliczenia etykiety uwierzytelniającej [B.3.6](#):

postać Base64URL:

```
PoikvBdxUJgflYAr9tyYOediK07oQTdEUAPcW6QSKcmPCule3TKtk-ypKVXtMrJvJd0uRweKpGvx_uyU-  
IEENg_Dh4o1Q8A7PQ5D2of_qZCW9pXi-DCnSE6wtPAggUCYKbeXWdJj8S-jzDrx6z-  
nu9ie25f4pw5EnRsA7pGLfk0MIao6oaEDcrmVm646tf9ezCKwACZX3HyOLpN0Q6PtovXPriGV07V7bQ-  
MCSsj0AoxKMdjv1YNUgblU-WxAPaSKI-  
7L54YgkMR3ob9eRUZNAHIsz2jNTAPCuOML2zycO4Qz0saa5h4Q8zuG9W00aAKJi0CQqHWh3FRqMSOInOxkbay5QVzrZ-  
Kbbyf6LusspxI5aibMNPBucrXkodWWszFc3th_jPQ4r6YZB-  
RzAO_inRNQqFsTJneP00Jjkd_0SfkanIcU5sjxsBbH9G6Vry5ECtOr5Me3IPcPIXGcdGqB03TaklpXCHj1-  
qOVWeDbZY5xxcQfhhsVYfOtM5ei4yIkNrXICwTagie-  
kmlBnF2mtjm0iv1YCUca8a44Rcd5BdpaGor1lEeizjAPBYi8QWtXSU5KDNXeC0gN4zUNPuccuzPzPk1QaQ7Mt8h9rgzOB9  
cDioOVLFe_hqAq4fn4xG25QKiTVLcOd8PKB8oA4S_f_4nO6cO7zPx_DP4Tu9cBE4M7DJxjZY2d4A21w-  
HGx4YukTwaX_w9bcfhKSYNkFYKky6aal80Z4JaMW902NHhleG0ER6c3Y7JFVCyWw9CXIJKfT6BVY8q5CdVFrW4_CF38-  
AIuXKRu-  
HizqOjM8TrOErY0hou5443u7KQedT4pyZWyAzVai4MK1Kj5jLUyLOZ4inHPgqTdT3AaTFX5FDwOM_ivUjwbwptwk7MC8eu  
oHTXQj02yB4p-  
TpdGsx4q09BHUG3fgfPNbrSFDWn80KNV8C6VmObATNZ8MTLkSB3lQhMEC3mYCeB3fMiy_E79NmePsttGKCCPotwFv4-  
qW-cGpl1-T_Z5P8_EqN6ss0400w4PlAq7Q0mJdmpaf0KqnT-  
gf7KcwYG5Vq3_5DJXCbu0DSigydQyBUl5E4OUH8X02sSNGetmiLuAmkrOV6dysu0bJVlu81OeyNlbQv7cN_qiL1DQbt_gs  
quhspxI8h1lMHYxQ2ygrIdm-UsMWdj120jBIy7z1FOVHJxbdna_fZN1dA1Tkgl1L-  
vuHdJprEqcBpnjo0Ke8wWpGwalreKi4DHOf3nOm_JEfpxh07DfPpG6fx-  
krxkkf2_ybyg773NSATjBBOeRbZolGrI8Ic7fdHIpfYyxhcCU9zLRJlWg0pjzHJIxoX-  
prwhW90BHqjFYpnOynMEvs6SXCES4OPFS2mXw8PFR51PkRkYfkSewraTnWwnw3WFqfiOBjCKeDpfbf5axbht-rg5BTHUi-  
VvPT3MrPInRy-p0BiJkLB9y8Z0R8DLX07ZoWajJOS55XWdZ8hdPWCoc509odTRcUUEvi-  
uqWQJMQCmZCiRwhsU9vO51T2t-aaWhLL9Q-e58M_pbm0HfLU_ea8jf5Vaf_bXht18BgZi2dvt-vddwY09kVB-  
jQkzJmvLe2_hcWlRkQ2X7-  
EmDlLt7iad7Nn20861a21DKdQjbVDXbXZVF2cp9jdtg_Ls5rMHgcTU2ORJG4jzcHCJi4qRVU9a9t4znWEbtgLS64T64qSc  
xqgcNL8gRzFwLxCCG_6UqFbUgqIkHfWj1zw572rA_WJGB0UVBfiRxltrZHKC2bhc85KTKsv00qIHVklDCrF47rTZ-  
Dx54IzGhsW_BRC5az1cshoNaABAuW2v_hs-  
DoMsLLpl15qfiRBW_WrhBbZjSFUn1AIL12CeeJss9pZ9ZNe2aPWjd75vTknxURE9mT8-jhlv3DjTs8_eBQWBT9-  
BdEy0AEPfKtZgZ4QHxLj_v3Vlr3F99Fu2dj977LJ1SiOnXMJlZLhyPrTnWjFuPavNe8HmsvjgysyMgHt_Lu_bx4OQk0P-  
MrfwbOxsSjjJivjjs-  
P5g2p194L5sGonvh2Jr15Q_aBHgKIiyakYxey3zbYcJfS1RoV5MSg2P_MJYoJI1Ffm_AAqUCRBK1dYQLhTiilQzjOb5_TZ  
8_3za2pjH7xLWC9legX6d8tFqpyX1XaLchQUObnKJgqubHejatfBZU1BM82BmyigS8PseeDPC2Cj6tg34BKra7q3I20H5X  
QeDFbyho7H01Y55jb5UvLsSmHdiAgoFoh7iSnki4iYrkQrPVCWlVbb2KvnhHotCeYFsdxfLa43y21STuYBoNP5Buu-  
cz4jovMwQyMZSgU7ojx8ae2RkrGzJx7Dp943U9w21mTNJDDbIdUg_QaPEi9PaxH-7mb72waZfdtNEB5S-  
MBQpIdIazD6QB4skD3b2h49rXggBYjyT59yxPrHkhDy_bo352BMqRxoKfjZlKdirVHS4cLy0SXty06E3TCut5_tFJ-0bo-  
kT-Dj4wR7k04Yzd6thPKi6AQJKPZz1ARZEguGu2-9p46AVZd1mCWDTCNlEDsL_gvb7GJFIknZt215voxI6WMjzyLio-  
jyGMBHgoQSsd246LQY-LNJCKJa8a18MbNqRRBZBdmPnvWUoK8eTQ6Nkt3mZQNQcrqvrRtrQ_OECAScwcDoSct4d6C-  
_w-
```

I77E\_FNVEvdnVDbWMKJMc9\_F99qfBTYPL9AL\_Cx89SyOJOrBWiWLClGoPjw91MV3K5fxX\_1EXapNyrfWW3jo90jeVKr42h  
eWacLTFMCCdNXyMuvWT8NttBRpkJwee-  
dae0vqomtHQQa1\_gyqc01J95pAbATYkjAGKYfnImS6c4fEP2VD6EAro7WQzmM3IzEvFF9dWMD5RnuwMKB2HECCxvSKkc03  
fQC\_xMuOEg1iqkE\_QyT2aJpPTyf7pB-  
zSy2C6gYt5DVlnc8ea6i1ElPSwdaS9uzMVHE0Hqrfq6Px23uZXCbVycNiSXCguUaAyLmOf6YsXK7hHHx\_uoVLVlB3SDME4  
GDI8bpOPisW\_LTViAlVjktSRm0021PqCu58mHgEP9h0jM-s582vcw9orSdi63q2OvK6dBoJawORWZ\_t-MWTc7r0-  
y\_766bekLrktyVHJkfbUFM4keDOEP9I4MO9mKHnCLGCMFok5z1JDxtN71M\_HXXpwJJ4hrCJIUOT2Z00glIRGE1YhMqILfh  
DJNELhhlytsqSy32RTOz1qbIog1TilpQOwxGPuvVazwcXTXOfHrd6074gpGHQkhWurIcYxLcm4rD8gja97iLm2J2keD5b  
HbgCryyJb8BotGZTA\_oj1VihMe9P6k69QwUCOo-MgQSVYgq6IXS3EbfDObWobjRtytaY46YXY9-4OREk6BlHes8Xn5bG-  
O7w0xJ4OKyKAES6iyRyFktr\_RP6rBMOfpOLZ90G68hX8ZwOVGWUIE8e8J5gB3XbpPYbX51Wi2kx-  
7CYXRDbYui361338tHks-aXcZ8Hb7h-D8bGxObateUSBR3Ka-  
6gCDMYHdSYijxHhhBxTTyrAcgzrLKD7CvV4PwQ5cUVx1qE7VeosUMIS9BotMXQmr12C0k1ulAHW8JspE43Mpced0NuUUaX  
ib9EVuNkXk1DjLXwtv8OMsugQ4\_bOWQMMCPz-  
15o5ciPGht0YkDAZJoODYMDpmZvMJLYb3QZbBbt2kgAG8Si0a\_9NZrkpTLOuRM

postać szesnastkowa wykorzystana w funkcji HMAC:

3e88a4bc177150981f2d802bf6dc9839e7622eeee84137445003dc5ba41229c98f0aed5edd32ad93eca92955ed32b2  
6f25dd2e47078aa46bf1feec94f88104360fc3878a3543c03b3d04e43da87ffa99096f695e2f830a7484eb0b4f02a81  
409829b79759d263f12a3cc3af1eb3fa7bbd89edb97f8a70e449d1b00ee918b7e4d0c21aa3aa1a0372b9959bae3a  
b5ff5ecc22b0002657dc7c8e2e937443a3eda2f5cfcae2195d3b57b6d0f8c092b23d00a3128c763bf560d5066e553e5  
b100f692922fbb2f9e18824311de86fd7915193401c8cecd3a35300f0ae38c2f6cf270ee10cf4b1a6b987843ccee1b  
d58e39a00a248d0242a1d6877151a8c48e20da1791b6b2e50573ad9f8a6dbc9fe8bbacb29c48e5a89b30d3c1b9cad7  
9287565acc5737b61fe33d0e2be98641f91cc03bf8a744d42a16c4c99de3f43898e477fd127e46a721c539b23c6c6  
c11fd1ba56bcb9102b4eaf931edc83dc3c85c671d1aa04edd36a49695c21e3d7ea8e5567836d9639c717107e1852bd  
87ceb4ce5e8b8c8890dad721c5936a089efa49a50671769ad8e6d22bf5602b9c6bc6b8e11703e417696863abd6511e  
8b38c03c1622f105ad5d2539283357782d20378cd434fb9c72eccfccf93541a43b32df21f6b833381f5c0c8a0e54b1  
5efe1a80ab87e7e311b6e502a24d52dc39df0f281f280384bf7ffe273ba70eef33f1fc33f84eef5c044e0cec32718d  
9636778036d70f871b1e18ba44f0697ff0f5b71f84a4983641582a4cba69a97cd19e0968c5bdd36347865786d0447a  
73763b245542c96c3d09720929f4fa05563cab909d545ad6e3f085dfcf8022e5ca454f878b3a8e8ccf13ace12b6348  
4ebb9e38deeca41e753e29c995b20335a8b830ad4a8f98cb5322ce6788a71cf82a4dd4f701a4c55f9143c0e33f8a  
f5236f0a29b7093b302f1eba81d35d08f4db2078a7e4e9746b31e2a3bd047520ddf8053cd6eb4850d69fcd0a355f02  
e9598e05a4cd67c3132e4481de54213040b799809e0777cc8b2f2c4efd36678fb2db462827023e8b7016fe3ea96f9c1  
a9d75f93fd9e4ff3f12a37ab2cd383b4c383e502aed0d2625d9a969fd0aaa74fe805eca730606e55ab7ff90c909706  
ed034a2832750c81525e44e0e507f17d36b123467ad9a22ee00c92b395e9dcacbb46c9565bbcd4e7b23756d0bf70d  
fea88bd43401b74832caae86cd7123c87594c1f2c50d2808ac39be52c316763976d23048cbce514e5472716dd9daf  
df64d95d0354e48352fcbefee1dd269ac4a9c0699e3a0e29ef305a91b06a5ade2a2e031ce7f79ce9b2f4479e77184eec  
37cfa46e9fc7e92bc64905dbfc9bca0efbdcd4804e304139e45b668d46ac8f0873b7dd1c8a45632c6170253dcccb449  
956834a63cc7248c685fea6bc215bdd011ea8c56299ceca7304bece925c2112e0e3c54b6997c3c3c5af9d4f9119187  
e449ec2b6939d6c27c37585a9f88e06308a783a5f6c5e5ac5b853fab8390531d48be56f3d3dccacf227472fa9d018a  
328b07dcb67447c0cb5f4ed9a1602324e4b9e5759d67c85d3d60a80b93bda1d4d1714504be2faea964093100a6642  
891c21b14f6f3b9d53dadf9a69684b2fd43e7b9f0cfe96cc077cb53f79af237f955a7ff6d786dd7c060662d9dbedf  
af75dc18d3d91507e8d0933266bcb7b6fe171696b290d97efe1260e52edee269dec9f6d3ceb56b694329d4236d50d  
76d7655176729f6376d83f2ece6b30781c4d4d8e4491b88f37070898b8a91554b1af6de339d611bb602d2eb84fae2a  
49cc6a81c34bf204731702f10821bfe94a856d482a2241df5a3973c39ef6ac0fd624607451504f891c65b51cc7282d  
9b85cf392932acbcded207564d430ab178eeb4d9f83c79e08cc686c5bf0517396b395cb21a0d680040530d95fe14b  
e0e832c2cba658b9a9f891070fd6ae105b663485527d4020bd7609e789b2cf6967d64d7b668f5a377be6f4e49f1511  
13d993f3e8e1d6fdc38d3b3cfde0505814fdf81744cb40043df29366078407c4b8ffbf754baf717df45bb6763f7be  
cb2754a23a75cc26564b872a4fad39f08c5b8f6af35ef079acbe3832b323201edfcbbbb6f1e0e424d0ff8cadfc1b39  
7b128e3262bf28d2f8fe60da9d7de0be6c1a89ef87626b97943f6811e02888b26a4631118df36d87097d2d51a15e4c  
4a0d8ffcc258a0923515f9bf000a9471104ad5d6102e14c88a54338ce6f9fd367cfff7cdada98c7ef12d60bdd5e817e  
9df2d16aa725f55da2c85050e6e728982ab9b1de8dab5f0593504cf36066ca2812f0fb1e7833c22d828fab60df80a  
aadaeeadcd8db41f95d0ac315bca1a3b1ce958e798db54bcb912987748020a053a1ee24a79232262b910ac65425a5  
bdb6f62af84d1e8b4279816c7717cb6b8df2db5493b9806834fe41baef9ccf88e8bccc10c8c652814ee88f1f1a7b64  
64ac6cc9c43ee9f78dd4f70db59933490c306921d520fd068f122f4f6b11feee607bdb06997c3b4d101e52f8c050a4  
87486b30fa401e2c903ddbd1e3dad78200588f24f9f72c4fac79210f2fdb37e7604ca91c6891f8d9d4a762ad51d2  
e1c2f2d125edcb4e84dd3094b79fed149fb46e8fa44fe0e3e3047b934e18cddead84f2a2e8040928f673d40459120b  
86bb6fbd8a78e805597759825834c2365103b0bfe0bdbc62452249d9b76d79b8c48e96323cf22e2a3e8f21863011e  
0a104ac776e3a2d063e2cd2422896bc6b5f0c6cdaa945105905d98f9ef594a0af1e4d0e8d913de665035072baafad1  
b6b43f38408049cc1c0e849cb7877a0beff0f88efb13f14d544bdd9d50db58c28931cf7f17df6a7c14d83cbf402ff0  
b1f3d4b2a093ab0568962c2946a0f8f0f75315dcae5fc57ff51176a9372adf596de3a3d3a37952abe3685e59a70b4c  
530209d35760cbb0553f0db6d051a6427079ef9d69ed2faa89ad1d041ad7f832a9cd3527de901b0136248c018a61f  
9c8992e9cel1f10fd950fa100ae8ed643398cdc8c4bc517d756303e519ec0c281d871020b1bd24734dd402ff13  
2e0d48358aa904fd0c93d9a2693d3c9fee907ec2d2cb60ba818b790d596773c79aea2d4494f4b075a4bddd33151c4d0  
7aab7eae8fc76dee65770155870d8925c282e51a0322e639fe98b172bb8471f1feea152d5941dd20cc13818323c6e9  
d0f8ac5bf2d356200bbe392db2b9b4d36d4fa82bb9f261ea10fff61d2333eb39f36bdcc3da2b49d8badead8ebcae9d0  
688dac0e45667fb7e3164dceebd3ecbfefae9b7a42eb92dc951c991f6d414ce247833843fd23830ef662879c22c60a  
6168939ce5243c6d9fbd4cfc75d7a70249e21ac224850e4f664ed2094844613562132a20b7e10c93442e1875cadb2a  
4b2df64533b3d6a6c8a20d538a5a503b0c5b18f5156b3c1c5d35ce16d1eb77a3bbe20a461d09215aeac8718c4b726e  
2b0fc8236bdee22e6d89da4783e5b1db802af2ca307c068b4665303fa23d558a131ef4fea4ebd4305023a8f8c81049  
5620aba2174b711b7dd3816d639b8d1b72b5a638e985d8f7ee0e44493a0651deb3c5e7e5b1be3bbc34c49e0e2b2280  
12cea2c91c8592daff44feab04c39fa4e2f3f741baf215fc67039519650813c7bc279801dd76e93d86d7e655a2da4c

7eec261744375b6148b7ea5df7f72d1e4b3e697719f076fb87e0fc6c6c4e6dab5e512051dca6beea00833181dd4988a3c478610714d3cab01c833acb283ec2bd5e0fc10e5c515c75a84ed57a8b143084bd068b4c5d09abd760b4935ba50075bc26ca44e3732971e77436e51469789bf4456a34ac4ad438cb5f0b6ff0e32cba0438fdb39640c302a73fa5e68e5c88f1a1b746240c0649a0e0d8303a6666f3092d86f7401cc16edda481a1bc4a2d1affd359ae4a532ceb913

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

65794a6c626d4d694f694a424d54493451304a444c5568544d6a5532496977695957786e496a6f69556c4e424d5638314969776961326c6b496a6f6959324d344d6d5130595755304d4759324f4746684d7a5179597a51795a44553559574d325a5751304d44517351303439545559675a53314c59584e35496e309641366ce173224c452a914e6de088c33e88a4bc177150981f2d802bf6dc9839e76228eee84137445003dc5ba41229c98f0aed5edd32ad93eca92955ed32b26f25d2e47078aa46bf1fec94f88104360fc3878a3543c03b3d0c39d3e43da87ffa99096f695e2f830a7484eb0b4f02a81409829b79759d263f12fa3cc3af1eb3fa7bbd89edb97f8a70e449d1b00ee918b7e4d0c21aa3aa1a10372b9959bae3ab5fff5ecc22b0002657dc7c8e2e937443a3eda2f5cfae2195d3b57b6d0f8c092b23d00a3128c763bf560d5066e553e5b100f692922fbb2f9e18824311de86fd7915193401c8ceda335300f0ae38c2f6cf270ee10cf4b1a6b987843ccee1bd58e39a00a248d0242a1d6877151a8c48e20da1791b6b2e50573ad9f8a6dbcb29c48e5a89b30d3c1b9cad79287565acc5737b61fe33d0e2be98641f91cc03bf8a744d42a16c4c99de3f43898e477fd127e46a721c539b23c6c6c11fd1ba56bcb9102b4eaf931edc83dc3c85c671dlaa04edd36a49695c21e3d7ea8e5567836d9639c717107e1852bd87ceb4ce5e8b8c8890dad721c5936a089efa49a50671769ad8e6d22bf5602b9c6bc6b8e11703e417696863abd6511e8b38c03c1622f105ad5d2539283357782d20378cd43afb9c72eccfcfc93541a43b32df21f6b833381f5c0c8a0e54b15efe1a80ab87e7e311b6e502a24d52dc39df0f281f280384bf7ffe273ba70eef33f1fc33f84eef5c044e0cec32718d9636778036d70f871b1e18ba44f0697ff0f5b71f84a4983641582a4cba69a97cd19e0968c5bdd36347865786d0447a73763b245542c96c3d09720929f4fa05563cab909d545ad6e3f085dfcf8022e5ca454f878b3a8e8ccf13ace12b63484eb9e38deeeeca41e753e29c995b203355a8b830ad4a8f98cb5322ce6788a71cf82a4dd4f701a4c55f9143c0e33f8af5236f0a29b7093b302f1eba81d35d08f4db2078a7e4e9746b31e2a3b0d47520ddf8053cd6eb4850d69fcd0a355f02e9598e05a4cd67c3132e4481de54213040b799809e0777cc8b2fc4efd36678fb2db462827023e8b7016fe3ea96f9c1a9d75f93fd9e4ff3f12a37ab2cd383b4c383e502aed0d2625d9a969fd0aaa74fe805eca730606e55ab7ff90c909706ed034a2832750c81525e44e0507f17d36b123467ad9a22ee00c92b395e9dcacbb46c9565bbcd4e7b23756dbfb70df8e88bd43401b7f82caae86ca7123c87594c1f2c50db2808ac39be52c316763976d23048cbbce514e5472716dd9dafdf64d95d0354e48352febeeldd269ac4a9c0699e3a0e29ef305a91b06a5ade2a2e031ce7f79ce9bf2447e977184e0c37cf4a46e9fc7e92bc64905dbfbc9bca0efbdc4804e304139e45b668d46ac8f0873b7dd1c8a45632c6170253dcc449956834a63cc7248c685fea6bc215bdd011ea8c56299ceca7304bece925c2112e0e3c54b6997c3c3c5af9d4f9119187e449ec2b6939d6c27c37585a9f88e06308a783a5f6c5e5ac5b853fab8390531d48be56f3d3dccacf227472fa9d018a328b07dcbc67447c0cb5f4ed9a1602324e4b9e5759d67c85d3d60a80b93bda1d4d1714504be2faea964093100a6642891c21b14f6f3b9d53dadf9a69684b2fd43e7b9f0cfe96ccd077cb53f79af237f955a7ff6d786dd7c060662d9dbedfaf75dc18d3d91507e8d0933266bcb7b6fe171696b290d977efe1260e52edee269dec9f6d3ceb56b694329d4236d50d76d7655176729f6376d83f2ece6b30781c4d4d8e4491b88f37070898b8a91554b1af6de339d611bb602d2eb84fae2a49cc6a81c34bf204731702f10821bfe94a856d482a2241df5a3973c39ef6ac0fd624607451504f891c65b51cc7282d9b85cf392932acbc2a207564d430ab178eeb4d9f83c79e08cc686c5bf0517396b395cb21a0d680040530d95fe14be0e832c2cba658b9a9f891070fd6ae105b663485527d4020bd7609e789b2cfc6967d64d7b668f5a377be6f4e49f151113d993f3e8e1d6fcd38d3b3cfd0505814fdf81744cb40043df293660678407c4b8ffbf754baf717df45bb6763f7becb2754a23a75cc26564b872a4fad39f08c5b8f6af35ef079acbe3832b323201edfcbbbb6f1e0e424d0ff8cadfc1b397b128f3262bf28d2f8fe60da9d7de0be6c1a89ef87626b97943f6811e02888b26a4631118df36d87097d2d51a5e4c4a0d8efcc258a0923515f9bf000a9471104ad56d102e14c88a54338ce6f9fd367cfff7cdada98c7ef12d60bdd5e817e9df2d16aa725f55da2dc85050e6e728982ab9b1de8dab5f05953504cf36066ca2812f0fble7833c2d828fab60df804aadaeeadc8db41f95d0ac315bca1a3b1ce958e798dbe54bcbb12987748020a053alee24a7922e2262b910acf5425a5bdb6f62af84d1e8b4279816c7717cb6b8df2db5493b9806834fe41baef9ccf88e8bccc10c8c652814ee88f1f1a7b6464ac6cc9c43ee9f78dd4f70db59933490c306921d520fd068f122f4f6b11feee607bdb06997c3b4d101e52f8c050a487486b30fa01e2c903ddbdale3dad78200588f24f9f72c4fac79210f2fdb37e7604ca91c6891f8d9d4a762ad51d2e1c2f2d125edcb4e84dd3094b79fed149fb46e8fa44fe0e3e3047b934e18cddead84f2a2e8040928f673d40459120b86bb6fbda78e805597759825834c2365103b0bfe0bdbec62452249d9b76d79be8c48e96323cf22e2a3e8f21863011e0a104ac776e3a2d063e2cd2422896bcb5f0c6cdaa945105905d98f9ef594a0af1e4d0e8d913de665035072baafad1b6b43f38408049cc1c0e849cb877a0befff0f88efb13f14d544bdd9d50db58c28931cf7f17df6a7c14d83cbf402ff0b1f3d4b2a093ab0568962c2946a0f8f0f75315dcae5fc57ff51176a9372adf596de3a3d3a37952abe3685e59a70c530209d35760cbb0553f0db6d051a6427079ef9d69ed2faa89ad1d041ad7f832a9cd3527de6901b0136248c018a61f9c8992e9ce1f10fd950fa100ae8ed643398cdc8cc4bc517d756303e519e0c281d871020b1bd22a4734dddf402ff132ed048358aa904fd0c93d9a2693d3c9fee907ecd2cb60ba818b790d596773c79aea2d4494f4b075a4bdbbb33151c4d07aab7eae8fc76dee65770155870d8925c282e51a0322e639fe98b172bb8471f1feea152d5941dd20cc13818323c6e9d0f8ac5bf2d356200bbe392db2b9b4d36d4fa82bb9f261ea10ff61d2333eb39f36bdcc3da2b49d8badead8ebcae9d0688dac0e45667fb7e3164dceebd3ecbfefae9b7a42eb92dc951c991f6d414ce247833843fd23830ef662879c22c60a6168939ce5243c6d9fbd4cfc75d7a70249e21ac224850e4f664ed2094844613562132a20b7e10c93442e1875cadb2a4b2df64533b3d6a6c8a20d538a5a503b0c5b18f5156b3c1c5d35ce16d1eb77a3bbe20a461d09215aeac8718c4b726e2b0fc8236bdee22e6d89da4783e5b1db802af2ca307c068b4665303fa23d558a131ef4fea4ebd4305023a8f8c8e10495620

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 kasy fiskalnej:

eyJlbmMiOiJBMTI4Q0JDLUhTMjU2IiwiaWxnbG9kaXkiOiJY2M4MmQ0YWU0MGY2OGFhMzQyYzQyZDU5YW  
M2ZWQ0MDQsQ049TUyGZS1LYXN5In0.hsilcenCtbhjLR60diiBZ7U6kFPqzU3Pc6DDnewQncob1sZTEv7bsqlAv-  
QDmR31iTRWuOzcYgduuDuAbmz1xwFI3cVObVeQBiRAgBtFuX\_xEpmBdXAZxRMXY37dX1SoCu0Rno0HnBs5bDsnuX4Tize4  
jEKNggOhpf88mErGW26BwU1WbvFlyj9HhEfI2H46D-y29djnkXI\_7MpM994hTRTq6JGPE-  
9m2ZjOtU6Yw4Ty1kRmUeK9iWQkdIKJHy6TGHd9qn03Odcyu\_DPHJnbmhHb5AaT5yAD9kGciticMC4PUDju3\_qcHgiwPo94  
s4m8j1RBo7M8jldVJvvOpU28ng.lkE2bOFzIkxFKpFObeCIww.PoikvBdxUJgfLYAr9tyYOediK07oQTdEUAPcW6QSKcmP  
Cu1e3TKtk-yPKVXTMrJvJdOuRweKpGvx\_uyU-IEENg\_Dh4o1Q8A7PQ5D2of\_qZCW9pXi-  
DCnSE6wtPAqgUCYKbeXWdJj8S-jzDrx6z-  
nu9ie25f4pw5EnRsA7pGLfk0Miao6oaEDcrmVm646tf9ezCKwACZX3HyOLpN0Q6PtovXPriGV07V7bQ-  
mCSsj0AoxKMdjv1YNUGblU-WxAPaSki-  
7L54YgkMR3ob9eRUZNAHizs2jNTAPCuOML2zyco4Qz0saa5h4Q8zuG9WOOaAKJIOCQqHWh3FRqMSOInoXkbay5QVzrZ-  
Kbbyf6LusspxI5aibMNPBucrXkodWWszFc3th\_jPQ4r6YZB-  
RzAO\_inRNQqFsTJneP00Jjkd\_0SfkanIcU5sjxsBbH9G6Vry5ECtOr5Me3IPcPIXGcdGqB03TaklpXCHj1-  
qOVWeDbZy5xxcQfhSvYfOtM5ei4yIkNrXICWTagie-  
km1BnF2mtjm01v1YCuca8a44Rcd5BdpaGOr11EeizjAPBYi8QWtXSU5KDNXeC0gN4zUNPuccuzPzPk1QaQ7Mt8h9rgzOB9  
cDIoOVLFe\_hqAq4fn4xG25QKiTVLcOd8PKB8oA4S\_f\_4nO6cO7zPx\_DP4Tu9cBE4M7DJxjZY2d4A21w-  
HGx4YukTwaX\_w9bcfhKSYNKyFYKky6aal80Z4JaMW902NHhleG0ER6c3Y7JFVCyWw9CXIJKft6BVY8q5CdVFRW4\_CF38-  
AIuXKRU-  
HizqOjM8TroErY0hou5443u7KQedT4pyZWYAzVai4MK1Kj5jLUyLOZ4inHPgqTdT3AaTFX5FDwOM\_ivUjwbwoptwk7MC8eu  
oHTXQj02yB4p-  
TpdGs4q09BHUG3fgFPNbrSFDWn80KNV8C6VmOBaTNZ8MTLkSB3lQhMEC3mYCeB3fMiy\_E79NmePsttGKCCPotwFv4-  
qW-cGp11-T\_Z5P8\_EqN6ss0400w4PlAq7Q0mJdmpaf0Kqnt-  
gF7KcwYG5Vq3\_5DJCXBu0DSigydyBu15E4OUH8X02sSNGetmiLuAMkrOV6dysu0bJVlu81OeyN1bQv7cN\_qiL1DQbt\_gs  
quhspxI8h11MHyxQ2ygIrDm-UsMWdj120jBiY7z1FOVHJxbdna\_fZn1dA1TkglL-  
vuHdJprEgcBpnj0Oke8wWpGwalreKi4DHO3nOm\_JEfpxh07DfPpG6fx-  
krxkkF2\_ybyg773NSATjBB0eRbZol1GrI8Ic7fdHlPfyxhccU9zLRJLWg0pjzHJIxoX-  
prwhW90BHqjFYpnOynMEvs6SXCES4OPFS2mXw8PFR51PkRkYfkSewraTnWwnw3WFqfiOBjCKEdpfbF5axbht-rg5BTHUi-  
VvPT3MrPInRy-p0BijKLB9y8Z0R8DLX07ZoWAjJOS55XWdZ8hdPWCoC5O9odTRcUUEvi-  
uqWQJMQCmZCiRwhsU9v051T2t-aaWhLL9Q-e58M\_pbM0HfLU\_ea8jf5Vaf\_bXht18BgZi2dvt-vddwy09kVB-  
jQkzJmvLe2\_hcWlrKQ2X7-  
EmD1Lt7iad7Nn20861a21dKdQjbVDXbXZVF2cp9jdtg\_Ls5rMHgcTU2ORJG4jzcHCJi4qRVU5a9t4znWEbtgLS64T64qSc  
xqgcNL8gRzFwLxCCG\_6UqFbUgqIkHfWj1zw572rA\_WJGB0UUBPiRxltrZHKC2bhc85KTKsv00qIHVklDCrF47rTZ-  
Dx54IzGhsW\_BRC5azlcsHoNaABAUw2V\_hs-  
DoMsLLpli5qfiRBw\_WrhBbZjSFUn1AIL12CeeJss9pZ9ZNe2aPwjd75vTknxURE9mT8-jh1v3DjTs8\_eBQWBT9-  
BdEy0AEPfKtZgZ4QHxLj\_v3VLR3F99Fu2dj977LJ1SiOnXMJLZLhypPrTnwjFuPavNe8HmsvjgysyMgHt\_Lu\_bx4OQk0P-  
MrfwbOXsSjjJivvyS-  
P5g2p194L5sGonvh2Jr15Q\_aBHgKIiyakYxYX3zbYcJfS1R0V5MSg2P\_MJYoJI1Ffm\_AAqUcRBK1dYQLhTiilQzjOb5\_TZ  
8\_3za2pjH7xLWC91egX6d8tFqpyX1XaLchQUObnKJgqubHejatfBZU1BM82BmyigS8PseeDPC2Cj6tg34BKra7q3I20H5X  
QrDFbyho7H01Y55jb5UvLsSmHdIagoFOh7iSnki4iYrkQrPVCWlvbb2KvhnHotCeYFsdxfLa43y21StuYBoNP5Buu-  
cz4jovMwQyMZSgU7ojx8ae2RkrGzJxD7p943U9w21mTNJDDbIdUg\_QaPEi9PaxH-7mB72waZfDtNEB5S-  
MBQpIdIazD6QB4skD3b2h49rXggBYjYt59yxPrHkhDy\_bo352BMqRxoKfjZ1KdirVHS4cLy0SXty06E3TCut5\_tfJ-0bo-  
kt-Dj4wr7k04Yzd6thPKi6AQJKPZz1ARZEguGu2-9p46AVZd1mCWDTCN1EDSL\_gvb7GJFIknZt215voxI6WMjzyLio-  
jyGGMBHgoQSsd246LQY-LNJCKJa8a18MbNqpRRBZBdmPnvWUoK8eTQ6Nkt3mZQNQcrqvrRtrQ\_OECAScwcDoSct4d6C-  
\_w-  
I77E\_FNVEvdnVDBWMKJMc9\_F99qfBTYPL9AL\_Cx89SyoJOrBWiWLC1GoPjw91MV3K5fxX\_1EXapNyrfWW3jo90jeVKr42h

eWacLTFMCCdNXYMuwVT8NttBRpkJwee-  
dae0vqomtHQQa1\_gyqc01J95pAbATYkjAGKYfnImS6c4fEP2VD6EAro7WQzmM3IzEvFF9dWMD5RnuwMKB2HECCxvSKkc03  
fQC\_xMu0EglikE\_QyT2aJpPTyf7pB-  
zSy2C6gYt5DVlnc8ea6i1E1PSwdaS9uzMVHE0Hqrfq6Px23uZXcBVYcNiSXCguUaAyLmOf6YsXK7hHHx\_uoVLV1B3SDME4  
GDI8bp0PisW\_LTViALvjktserm0021PqCu58mHqEP9h0jM-s582vcw9orSdi63q2OvK6dBojawORWZ\_t-MWTc7r0-  
y\_766bekLrktyVHJkfbUFM4keDOEP9I4M09mKHnCLGCMFok5z1JDxtn71M\_HXXpwJJ4hrCJIUOT2ZO0glIRGE1YhMqILfh  
DJNELhhlytsqSy32RTOz1qbIog1TilpQOwxGPGUVazwcXTXOFtHrd6O74gpGHQkhWurIcYxLcm4rD8gja97iLm2J2keD5b  
HbgCryyjb8BotGZTA\_oj1VihMe9P6k69QwUCOo-MgQSVYgq6IXS3EbfDOBbWObjRtytaY46YXY9-4OREk6BlHes8Xn5bG-  
O7w0xJ4OKyKAES6iyRyFktr\_RP6rBMOfpOLz90G68hX8ZwOVGWUIE8e8J5gB3XbpPYbX51Wi2kx-  
7CYXRDbYU1361338tHks-aXcZ8Hb7h-D8bGxObateUSBR3Ka-  
6gCDMYHdSYijxHhhBxTTyrAcgzrLKD7CvV4PwQ5cUVx1qE7VeosUMIS9BotMXQmr12C0k1ulAHW8JspE43Mpced0NuUUaX  
ib9EVuNKxK1DjLXwtv8OMsugQ4\_bOWQMMcpz-  
15o5ciPGht0YkDAZJoODYMDpmZvMJLYb3QBzBbt2kgaG8Si0a\_9NZrkpTLOuRM.ct35FpBFiOdWQ7aw8Jk45A

```
{
  "JPK": {
    "naglowek": {
      "wersja": "JPK_KASA_v0-92",
      "dataJPK": "2018-03-30T17:25:26.027Z"
    },
    "podmiot1": {
      "nazwaPod": "Zażółć gęślą jaźń",
      "nrFabr": "ZTE-FAB-0123456789",
      "NIP": "111111111",
      "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:

eyJKUEsiOnsibmFnbG93ZWsionSid2Vyc2phIjoiSlBLX0tBU0FfdjAtOTIiLCJkYXRhS1BLIjoiMjAxOC0wMy0zMfQxNz  
oyNToyNj4wMjdaIn0sInBvZG1pb3QxIjpw7Im5hendhUG9kIjoiWmHFVMOzxyLEhyBnxJnFm2zEhSBqYw6xYqILCJuckZh  
YnIoiJaVeUtrkFCLTAXmJmONTy3QDkILCJOSVaiOiIXMTEmXTEmXTEiIwiYWRyZXNjb2QiOisJmXpY2EioiIjVbG1IjYs  
IsImlpZWpzeWY1I6klpZWpzeWY293b8WbXicILCJuckxvay7I6kYsTG9rIiwicG9jerhXNqOjUoU9jdenRhIwbnJeb211jJoi  
TnJeb211Iiwia29kUG9jenQioiIwM0CwMDAifSwibnJvbm1rIjoiWlRfMTIzNDU2Nzg5MCI5Im5yRXdpZCI6IjIwMTgVMT  
IzNDU2Nzg5In0sImNvbmlRbnIjoiQ1o7I7npkYXJ6ZW5pZS16eyJwYW1pZWNDaHioIjEsIkpQs01EIjoxoSwizGF0YUN6YXMi  
OiIyMDE4LTZlLTMwVDE3OjI1IjE2LjA5Nk0sIml0LjC0eXAxIisMDEiOnsiaW5mbyI6I6f3YXJ5am51IHplcm93YV5pZSBwYW  
1pxJlJaSBvZGVyYWN5am5laioj9fX19XN19

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

[illegible]



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

### B.5.1 Pełna postać chronionej 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 kasy fiskalnej zakodowana w Base64URL:

```
eyJraWQiOiIzQTAwMDAwMDA4RTY4RjU1ODE1RDRFQTYzQTAwMDEwMDAwMDAwOCwgQ049ZUthc3ktU3ViQ0EiLCJlbmMiOiJBMTI4Q0JDLUhTMjU2IiwiaWwXnIjoilU1NBmV81In0
```

### B.5.3 Wartości przykładowych danych użytych do szyfrowania:

```
JWE_AES_CEK => 0fae0202c8aa5d39bac1b9f58a9f440c4b700abdf7e661ac4c609e288529be
```

```
JWE_MAC_KEY => 0fae0202c8aa5d39bac1b9f58a9f440c
```

```
JWE_AES_KEY => 4b700abdf7e661ac4c609e288529be
```

```
JWE_AES_IV => c773654a97535031619a525f4285f3dd
```

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

```
giC4C064EjKUDWmh0TmoQVuxBxByz5yWO53Hxda3JelgRchNmnq0s38RXyJt9L1e3SDe5hnZuVtgPKufUBgEblItprrium  
GYnqyhuzIbD0m8akTq9JyJHQ7SERZlGYIzAgQbIn7NJAKswtzhpP56PxSnMRwegdx0PoW-  
ZlTx2dYSpHRobWYvpHjz4t25H_poYZh2nAmmzC4nWOGnlshNI0qxk21E64_Tb-  
4ACpoqvk6WlPGygUYT4M1hoAN4w_P0fsTyQxcmsXt3RpfiCbCROY2oLyQYGVYnTodbpLp7T4kVKN3XnzT6szBFzMCJsQ9B  
0Ug26mQARKh0cx7FWOtVCg
```

### B.5.5 Przykładowa wartość wektora inicjującego zakodowana w Base64URL:

```
x3NlSpdTUDFhmlJfQoXz3Q
```

### B.5.6 Przykładowa wartość zaszyfrowanych danych zakodowana w Base64URL:

```
jnyxL4MrKbWmgY5ZeUrL_etV-  
byYTeEwp8fh1j5q8ii54kYWQPUBErh3APg7KL18ZjY644HdbcZgy168x0cACpVhh5MqA93K5CX10e-  
4VUH97iBcHo_VdzUW_igX3HPyfl1WLrt2PgIiqFOuzjapVYiVFzmXwhKsoWx4oYzIJfUJYM1QFCHnPVWPON1TnpDV6VRss  
km3oC4sCn9DW0TJ-  
h8HHzvXT18n6F1SHLrqcVgfb4cpz5w6X_k8TBPXaJNgOvEgcbUTXNkKkUWs2xTKNdJAinKTzXR9WFsPajd6R5brwVIEpRg  
sHy6icr4RyxGO7fPS8JmKsbEY2CqedVO62ZtYREH1K2giji51XPEpLPa7kFHdJ4uknjOjbgK-_Kbb9BUVdhdfO6-  
bNpT67fiReZ--JLYu-cAYVnoTMjOGqSXOjNLiPzBVGLQ3Q02yw875kPf0nrlPihkNoCGgdBgP66mvF0CdP-  
0hoZiMYKh9qOzeAtuiDb0kQL9lmpexz_YZ1GQs-UrkYP85Fnv-vpvPkgBp9kjl9x1SiB4qvj86Ryaf-  
ic8P_FvOy9bPv9wRtGIAdTcJBuK0vaIb7W7FCYiBKyKv6oR0vsoXghHS8NrH7ARm0VIC-  
q9v_S8exfy5Rik6znN51Qwsdz5qG0b-rQU5GslhLFckc8Ra1wDMvyfv6qSE6-  
dbpmbIsw9rCh2tvNzwM2nN4cHMZQErfa8-  
bTuGcRehZVwrjmfYkV9kcWqbSVajJfWAzeZLmjmcU3hytB6Ke7Yo2HyIjtQNqfWYWy4JmpgqSEVexk6MxebQr35s2LKZ8H  
ouO7jqqPV1RHh_oFIxNcaleisxh4nuEm3T8OfXU8UHvdjRHu2V_HjLZFFtuzT51iydzCHCUVK020mNqF92mMg3w3p6VTe6  
xBtteuIfG5uORdfw9Onulu3e-PVe7P3H27iuPvLdS-  
hG0ywN8aissvnDrIe2gKL_Na7Cuf_9XGv2EHXEEFqYJ67GUIdMYEK1M3JVM0QiQ0zFL11lPwyN9nHdXRYfneWo4931krUe  
Yh-yxAtmgeH2lKt67pQ6jwZrqXS9WVBpBdN5JzMS8RjJShZY2BEotJp5MqMdz-  
IfZwnx07JBjuMobskg_lyPmtCWQXoAb-MA574jiv40EEKq9HkS27T59wOS-  
XdOi32KJHs3a9b1pSgmZqiWSUpwLaOuuZYjV9H7_sz5i6w8oCBYxrcOLcjYlT2cp0HIAHiM1LnHrDvmAN0WehlrthK2I4E  
sy9yHf11RjFwzoUmSz5sYdnx2jqKOEsKaiaqc45IMTEdFIV8pWJcFsaxMx1CjUeumu_IGom0aiMDikvAyrs-  
BmFwIOVSRzIsyезngHHS6K5H3R9isuiTA64OkomnqKicC2TFjmx97EAjMGG3CBqc7fzdPMDcCYRoTgrqJ8JF-  
gjYED4RBAZ8cZYBjx77neOpZ-  
Kdy_8h3zT70GmJzmzjshrqDnPV3CdT_T7JCPWPoE0cQk2oUiVcCnx3DsH6x_1tti6FkzfdqAcGj7VftQN5XAwVcmMBZQwh  
ChFS945C1sord1SLWqWfaxG0CBqu5rVfWODOZsV4WAxX0XjrE6CbDbeLuCrzcjTto3rcRnrsVAwb-  
fnC97FEsevl4Lfg7S7iE8YRCR1QW3M40aI-46SHMBov2u92RdPvzLpYjIf3N29YmijYkFMcYnfVlt7-
```

U2Ou9YpNNhF5h2veEHZWUS07gyy-psOtkjLXwp2MpKQucf3jyXJWyuArU4qVAaN-E-  
Wzk0l0nI\_5zFb3W26AbEoumrhL8FpY8P5KGYAgH1B3uVNs2TQzdb3cAg41Ie9PBeeNtMi2-m505-  
r1ljD0uLhmaZeidaCYdr4V50AiC3y4ggd6R4GE0u6QXttDcgwvRAHUUBtzT4R0Ne4-KzcZVGK3Is2zGQL-  
HrTii\_oOgmDBpdTsnWTX8wFBITUw\_tKf7YOZbwedLk8QVLYKji9ZdlPOek-w-  
zRXaYDTQotleP4oRdMy4wmPnlmHsdF8RJ0UCQ8YrRF1Fz7ogp9C8ftU3YpGo9QxfKyaw46TsiGlQrEBCQ3r8gRNOTXHCs  
O41-3iB05u4pDhZzz-7WbHltXGuDqFik7BWnq-  
V0wRhX7KXeI3ikGSsp5eCrigbHWsci\_QdCmj5YjR5FV7Av29VqkZScCKZQ4\_f4SHiYPuM7hurB56qLmpBlhkvxUZHyvp0u  
PCa-vDgSn0FQ23dAVvmUI6epkad\_-YwUtOthCJvfphZeIH8rvAOrlp1YM71EEkW-  
u5xrCrdHtBy7NQpGCVYs\_Cm6g\_UIfguOrgOBmvgEKQlKapuLLi\_rGwHsSN0lnMA9yx3sC4PGTeIAHcalwtOp1sFQwUxcgE  
QRV8TY86G17qlrQIs-v9-  
uDwSEI7aZpqGfZmZKaQOWCno23i4XbDQnnnjCdF1fmMxACorg5g\_Wjcpgs91fPiePv9\_1YYEnXRPduqhQbVb1J2txdOoS  
CrG9RZesYhV0KBvLOHBbhFBCD-  
5IAPXXIJ3It2aU2HJUTD77B60U2d0LZ6RzQ2OvmNgJ8oUSozKplTiYaDLKE3mgMO4o2IksZjvKISi69MmsJy059vQXkPBG  
AJ3hDlHsCwrxCg43ZvbcV2T55466HhsBv8WAtTl01psY00IBAp2QgbFfkZkYEaP5XgLmdtAGBJwPjSSgH4dM3lMgbgRXyd  
yo9qoEFUJj3-bzYMGOP5mSIdKQguzYv7\_fuClBzAeq\_iYHdqLaXK-tSq0\_mHw2jLgVPGYk-  
02UZwUEUINqqXnfD04YPQ04XgmtrieqJzTaQH3xGpD2N2u-Z8VqaJkYD-  
k99\_Qftu0WUYOWvWaw0vgTATN\_Vu2EgPB9uOSC5U17jmiieGAf5O7yTlivMdU2hMETPrsa-  
FXomu56nzu0kxE0rx7h\_Ckf2fDI3QatBSSnyndmK3jWiEtte65U\_8-wJSU8Js\_mE5-  
MtXDBk9qobRfMcJpfzAgatdD7RWRyuIV85rERPOqKfaoqe\_xtliuW8zNhy634zLx6kv\_aKI6d8jE1QgO0qZIdgRcKqpPlz  
SPmC2rf1Z7ReOCTH\_pc97Nj\_0pXQBzftI3oWhnF6gpTtROKHn6-kbvne249vASSPM70UO9y8PXyMK-  
kQGT8GGsmreAlWh1SFOMgiN-cfRrlGxHTAn0w

### B.5.7a Przykładowa wartość dodatkowych danych uwierzytelniających:

eyJraWQiOiIzQTAWMDAwMDA4RTY4RjU1ODElRDRFQTYzQTAWMDEwMDAwMDAwOCwgQ049ZUthc3ktU3ViQ0EiLCJlbmMiOi  
JBMTI4Q0JDLUhTMjU2IiwiaWxnIjoiaUlNBMV81In0

### 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](#):

0fae0202c8aa5d39bac1b9f58a9f440c4b700abdfdd7e661ac4c609e288529be

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:  
eyJraWQiOiIzQTAwMDAwMDA4RTY4RjU1ODE1RDRFQTYzQTAwMDEwMDAwMDAwOCwgQ049ZUthc3ktU3ViQ0EiLCJlbmMiOiJBMTI4Q0JDLUhTMjU2IiwiaWxnbG9kaWkiOiJlbnBMV81In0

postać szesnastkowa (bajty postaci Base64URL) wykorzystana w funkcji HMAC:  
**65794a72615751694f69497a515441774d4441774d44413452545934526a55314f444531524452465154597a515441774d4445774d4441774d4441774f437767513034395a55746863336b7455335669513045694c434a6c626d4d694f694a424d54493451304a444c5568544d6a5532496977695957786e496a6f69556c4e424d563831496e30**

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-  
byYTeEwp8fH1j5q8ii54kYWQPUBErh3APg7KLl8ZjY644HdbCZgy168x0cACpVhh5MqA93K5CX10e-  
4VUH97iBcHo\_VdzUw\_iqX3HPyfl1WLrt2PgIiqFOuzjapVYiVFzmXwhKsoWx4oYzIJfUJYM1QFcHnPVWFPONlTnpDV6VRss  
km3oC4sCn9DWOtTJ-  
h8HHzvXTl8n6F1SHLrqcVgfb4cpz5w6X\_k8TBPXaJNgOvEgcbUTXNkKkUws2xTKNdJAinKTzXR9WFsPajd6R5brwVIEpRg  
sHy6icr4RyxGO7fPS8JmKsbEY2CqedVO62ZtYREH1K2giji5lXPEpLPa7kFHdJ4uknjOjbgK-\_Kbb9BUVDhdFO6-  
bNpT67fiReZ--JLyu-cAYVnoTMjOGqSXOjNLiPZbVGiQ3Q02yw875kPf0nrlPihkNoCGgdBgP66mvF0CdP-  
0hoZiMYKh9qOzeAtuiDb0kQL9lmpexz\_YZlGQS-UrkYP85FNV-vpVPkgBp9kjl9x1SiB4qvj86Ryaf-  
ic8P\_FvOy9bPv9wRtGIADTCJBuK0vaIb7W7FCYiBKyKv6oR0vsoXghHS8NrH7ARm0VIC-  
q9v\_S8exfy5Rik6znN5lQwsdz5qG0b-rQU5GslhLfckc8RalwDMvyfv6qSE6-  
dbpmbIsw9rCh2tvNzwM2nN4cHMZQErfpa8-  
bTuGcRehZVwrjmfYkV9kcWqbSVajJfWAzeZLmjmcU3hytB6Ke7Yo2HyIjtQNqfWyWY4JmpggSEVexk6MxeBQR35s2LKZ8H  
ouO7jqPv1RHh\_oFIxNcaleisxh4nuEm3T8OfXU8UHvdjRhu2V\_HjLZFFftuzT5liydzcHCUVK020mNqF92mMg3w3p6VTe6  
xBtteuWfG5uORdfw9Onulu3e-PVe7P3H27iuPvLdS-  
hG0ywn8aissvnDrIe2gKL\_Na7Cuf\_9XGv2EHXEEFqYJ67GUIdMYEK1M3JVM0QiQ0zFL1l1PwyN9nHdXRyfnWo4931krUe  
UYh-yxAtmgeH2lkt67pQ6jwZrQXS9WVBpBdN5JzMS8rJShZY2BEotJp5MqMdz-  
IFZwnx07JBjuMobskg\_lyPmtCWQXoAb-MA574jiV40EEKq9HkS27T59wOS-  
XdOi32KJHs3a9b1pSgmZqiWSUpwLa0uuYzYV9H7\_sz5i6w8oCBYxrcOLcjYlT2cp0HIAhiMlLnHrDvmAN0WehlrthK2I4E  
sy9yHfllRjFwzoUmSz5sYdnx2jqKOEskAiaqc45IMTEdFIV8pWJcFsaxMxlCjUeumu\_IGom0aiMDikvAysr-  
BmFwIOVSRzIsyzeznghHS6K5H3R9isuiTA64OkomnqKICc2TFjmx97EAjMGG3CBqc7fzdPMDcCYRoTgrq8JF-  
gjYED4RBAZ8cZyBJx77neOpZ-  
Kdy\_8h3zT70GmJzmzjshrqDnPV3CdT\_T7JCPWPoE0cQk2oUivKcNx3DsH6x\_1tti6FkzfdqAcGj7VftQN5AXAwVcmMBZQwh  
ChFS945C1sORdlSLWgWfaxG0CBqu5rVfwoDOZsV4WAxX0XjrE6CbDbeLuCrzczjTto3rcRnrsVAWb-  
fnC97FEseVL4Lfg7S7iE8YRCRlQW3M40aI-46SHMBov2u92RdPvzLpYjIf3N29YmijYkFMcYnfVlt7-  
U2Ou9YpNNhF5h2veEHZwUS07ggy-pSotkjLXwp2MpKQucf3jyXJWyuArU4qVAaE-E-  
WZk01oNI\_5zFb3W26AbEoumrhL8FpY8P5KGYAgH1B3uVNs2TQzdb3cAg4lIe9PBeeNtMi2-m505-  
rlljd0uLhmaZeidaCYdr4V50AiC3y4ggd6R4GE0u6QXttdCgwwRAHUUBtzT4RONE4-KzcZVGK3Is2zGQL-  
HrTii\_oOgmDBpdTsnWTX8wFBITUw\_tKf7YOZbwedLk8QVLYKji9ZdlPOek-w-  
zRXaYDTQotlep4oRdMy4wmPnlmHsdF8RjUUCQ8YrRf1Fz7ogp9C8ftU3YpGo9QxfKyaw46TsiGlQrEBCQ3r8gRNOTXHCs  
O4l-3iB05u4pDhZzz-7WbHlTXGuDqFik7BWnq-  
VOWRhX7KXeI3ikGSsp5eCrigbHWsci\_QdCmj5YjR5FV7Av29VqkZScCKZQ4\_f4SHiYPuM7hurB56qLmpBlhkvxUZHyvp0u  
PCa-vDgSn0FQ23dAvvmUI6epkad\_-YwUtOthCjvfpHzeiH8rvAOr1p1YM7lEEkw-  
u5xrCrdHtBy7NQpGCVYs\_Cm6g\_UIfguOrgOBmvgEKQlKapuLLi\_rGwHsSN0lnMA9yx3sC4PGTeIAHcalwtOp1sFQwUxcgE  
QRV8TY86G17qlrQIs-v9-  
uDwSEI7aZpqGfZmZKaQOWCno23i4XbDQnnnjCdFlfmMxACorg5g\_Wjcpgs9lfpiePv9\_1YYEnXRPduqhQbVb1J2txdOoSM  
CrG9RZesYhV0KBvLOHBbhFBCD-  
5IAPXXIJ3It2aU2HJUTd77B60U2d0lZ6RzQ2OvmNgJ8oUsoZKplTiYaDLKE3mgMO4o2IksZjvKISi69MmSJy059vQXkPBG  
AJ3hDlHsCwrxCg43ZvbcV2T55466HhsBv8WAtTl01psY00IBAp2QgbFfkZkYEaP5XgLmdtAGBJwPjSSgH4dM3lMgbgRXyd  
yo9qoEFUJj3-bzYMGOP5mSIdKQguzYv7\_fuClBzAeq\_iYHdqLaXK-tSq0\_mHw2jLgVPGYk-  
02UZwUEUINqqXnfd04YPQ04XgmtrieqJzTaQH3xGpD2N2u-Z8VqaJkYD-  
k99\_Qftu0WUYOWvWaw0vgTATN\_Vu2EgPB9uOSC5U17jmiieGAf5O7yTlivMdU2hMETPrsa-  
FXomu56nzu0kxE0rx7h\_Ckf2fDI3QatBSSnyndmK3jwiEtte65U\_8-wJSU8Js\_mE5-  
MtXDBk9qobRfMcJpfzAgatdD7RWRyuIV85rERPOqKfaoqe\_xtliuW8zNhy634zLx6kv\_aKI6d8jE1QgO0qZIdgRcKqpP1z  
SPmC2rf1Z7ReOCtH\_pc97Nj\_0pXQBzftI3oWhnF6gpTtROKHn6-kbvne249vASSPM70U09y8PXyMK-  
kQGT8GGsmreAlWhlSFOMgiN-cfRrlGxHTAn0w

postać szesnastkowa wykorzystana w funkcji HMAC:  
**8e76312f832b29b5a6818e59794acbfdeb55f9bc984de130a7c7c7d63e6af228b9e2461640f50112b87700f83b28b97c66363ae381dd6dc660cb5ebcc747000a956187932a03ddcae425e5d1efb85541fdee205c1e8fd5773516fe2a97dc73f27e5d562ebb763e0222a853aece36a9558895173997c212aca16c78a18cc825f50960cd5015c1e73d558f38d953**

9e90d5e9546cb249b7a02e2c0a7f435b44c9fa1f071f3bd74f5f27e85d521cbaea71581f6f8729cf9c3a5fff93c4c13  
d76893603af12071b5135cd90a9145acd14ca35d2408a7293cd747d585b0f6a377a4796ebc15204a5182c1f2ea272  
be11cb118eedf3d2f0998ab1b118d82a9e7553bad99b584441f52b68228e2e655cf1292cf6bb9051dd278ba49e33a3  
6e02befca6dbf415157617453baf9b3694faedf891799f9b24bcaef9c018567a13323d06a925ce8cd2e23d96d51b54  
37434db2c3cef990f7f49eb94f8a190da021a074180feba9af17409d3fed21a1988c60a87da8ecde02dba20dbd2440  
bf7598f7b1cff619d4642cf94ae460ff39167bfebe9b9cf920069f64265f71d52881e2abe3f3a47269ffa273c3ff16f  
3b2f5b3eff7046d1886834dc241b8ad2f6886fb5bb14262204ac8abfaa11d2fb285e08474bc36b1fb0119b45480bea  
bdbff4bc7b17f2e518a4eb39cde75430b1dcf9a86d1bfab414e46b3584b15c91cf116b5c0332fc9fbfaa9213af9d6e  
999b22cc3dac2876b6f373c0cda7378707319404adfa5af3e6d3b867117a1655c2b8e67d8915f64716a9b4956a325f  
580cde64b9a399c537872b41e8a7bb628d87c888ed40da9f5b2598e099a982a48455ec64e8cc5e050af7e6cd8b299f  
07a2e3bb8eaa8f57544787fa0523135c6a57a2b318789ee126dd3f0e7d753c507bdd8d11eed95fc78cb64516dbb34f  
9d62c9dcdc1c25152b4db498da85f7698c837c37a7a5537bac41b6d7aec087c6e6e39175fc3d3a7bb5bb77be3d57bb  
3f71f6ee2b8fbc752fa11b4cb037c6a2b2cbe70eb21eda028bfcd6bb0ae7fff571afd841d710416a609ebb1948833  
1810ad4cdc954cd10890d3314bd6594fc3237d9c77574727e7796a38f77d64ad4794621fb2c40b6681e1f69644faee  
943a8f066ba974bd59506905d379273312f118c94a1658d81128b49a7932a31dcfe21f6709f1d3b2418ee3286ec920  
ff5c8f9ad096417a006fe300e7be23895e341042aaf47912dbb4f9f70392f9774e8b7d8a247b376bd6f5a5282666a8  
96494a702da3aeb9962357d1fbf6ccf98bac3ca02072c6b70e2dc8d8d53d9ca741c868788cd4b9c7ac3be600dd167a  
1d6bb612b623812ccbd877e5d518c5c33a14992cf9b18767c768ea28e12c90089aa9ce3920c4c4745215f29589705  
blac4cc750a351eba6bbf206a26d1a88c0e292f032aecf81985c08395491cc8b327b39e08474ba2b91f747d8acba24  
c0eb83a4a269ea288702d931639b1f7b1008cc186dc206a73b7f374f303702611a1382bab245fa98d8103e11104067  
c719601271efb9de3a967e29dcbff21df34fbd06989ce6ce3b21aea0e73d5dc2753fd3ec908f58fa04d1c424da8508  
bca0a7c770ec1fac7fd6db62e859337dda807068fb55fb503795c0c15726301650c210a1152f78e42d6ca110f548b5  
aa59f6b11b4081aaee6b55fc0e0ce66c578580c57d178eb13a09b0db78bb82af37234eda37adc467aec54059bf9f9c  
2f7b144b1ebcbe0b7e0ed2ee213c611091d505b733839a23ee3a487301a2fdae7645d3efccba588c87f7376f589a2  
8d890531c6277d596defe5363aef58a4d361179876bde107cd6512d3b832cbea523ad932d7c29d8ca4a42e71fde3c  
97256cae02b538a9501a37e13e59990ed68348ff9cc56f75b6e806c4a2e9ab84bf05a58f0fe4a1980201f5077b9536  
cd9343375bddc020e2521ef4f05e78db4c8b6fa6e74e7eaf5963774b8b8666997a275a09876be15e740220b7cb8820  
77a478184d2ee905edb5d0a0c2f45600751406dcd3e1138d7b8f8acdc65518adc8b36cc640bf87ad38a2fe83a09830  
69753b275935fcc050484d4c3fb4a7fb60e65bc1e74b93c4152d82a38bd65d94f39e93ec3ecd15da6034d0a2d95e3f  
8a1174ccb8c263e7d661ec745f1127450243c62b445d45cfba20a7d0bc7ed5376291a8f50c5f2b26b0e3a4ec886950  
ac4042437afc81134e4d71dc0ac3b897ede2074e6ee290e1673cfeed66c796d5c6b83a858a4ec15a7abe574c11857e  
ca5de8b78a4192b29e5e0ab8a06c75ac722fd07429a3e588d1e4557b02fdbd56a91949c08a650e3f7f84878983ee33  
b86eac1e7aa8b9a9065864bf15191f2be9d2e3c26bebc38129f4150db774056f99423a7a991a77ff98c14b4eb61089  
bdfa6165e887f2bb0c3abd69d5833bd441245bebb9c6b0ab747b41cbb350a4609562cfc29ba83f5087e0b8eae03819  
af80429094a6a9b8b2e2feb1b01ec48d3a59cc03adb1dec0b83c64de2001dc6b5c2d3a9d6c150c14c5c80441157c4d  
8f3a1b5eea96b408b3ebfdfae0f048423b699a6a19f66664a6903960a7a36de2e176c34279e78c2745d5f98cc400a8  
ae0e60fd68dca60b3dd5f3e278fbfdff56181275d13ddbaa8506d56f5276b7174ea12302ac6f5165eb18855d0a06f2  
ce1c16e1141083fb9200a575c827722dd9a5361c952d0fbec1eb453674e959e91cd0d8ebe636027ca144a8ccaa654e  
261a0cb284de680c3b8a36224b198ef2884a2ebd326489cb4e7dbd05e43c1180277843947b02c2bc42838dd9b9db715  
d93e79e3ae8786c06ff1602d4e5d35a6c60ed08040a764206c57e466460468fe5780b99db401812703e3492807e1d3  
37d4c81b8115f2772a3bdaa81055098f7f9bcb8306a0f6664a274a420bb362feff7ee0a507301eabf8981dda8b6972b  
eb52ab4fe61f0da32e054f19893ed36519c1411420daa5e77ddd3860f434e17826b6b89ea89cd36901f7c46a43d8d  
daef99f15a9a264603fa4f7dfd07edbb459460e5af59ac34be04c04cdfd5bb61203c1f6e3920b9525ee39a289e1807  
f93bbc93962bcc754da13044cfaec6be157a26bb9ea7ceed24c44d2bc7b87f0a47f67c323741ab414929f29dd98ade  
35a212db44eb953ff3ec09494f09b3f984e7e32d5c3064f6aa1b45f31c2697f30206ad743ed1591cae215f39ac444f  
3aa29f6a8a9eff1b758ae5bccd872eb7e332f1ea4bfff68a23a77c8c4d5080ed2a64876045c2aaa4fd7348f982db7  
f567b45e382b47fe973dec8dfdd295d00737ed237a1686717a8294ed44e2879fafa46ef9dedb8f6f01248f33bd143b  
dc3cd3d7c8c2be910193f061ac9ab78095687548538c82237e71f46b946c474c09f4c

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

65794a72615751694f69497a515441774d4441774d44413452545934526a55314f444531524452465154597a515441  
774d4445774d4441774d4441774f437767513034395a55746863336b7455335669513045694c434a6c626d4d694f69  
4a424d54493451304a444c5568544d6a5532496977695957786e496a6f69556c4e424d563831496e30c773654a9753  
5031619a525f4285f3dd8e76312f832b29b5a6818e59794acbfddeb55f9bc984de130a7c7c7d63e6af228b9e2461640  
f50112b87700f83b28b97c66363ae381dd6dc660cb5ebcc747000a956187932a03ddcae425e5d1efb85541fdee205c  
1e8fd5773516fe2a97dc73f27e5d562ebb763e022a853aece36a9558895173997c212aca16c78a18cc825f50960cd  
5015c1e73d558f38d9539e90d5e9546cb249b7a02e2c0a7f435b44c9fa1f071f3bd74f5f27e85d521cbaea71581f6f  
8729cf9c3a5fff93c4c13d76893603af12071b5135cd90a9145acd14ca35d2408a7293cd747d585b0f6a377a4796eb  
c15204a5182c1f2ea272be11cb118eedf3d2f0998ab1b118d82a9e7553bad99b584441f52b68228e2e655cf1292cf6  
bb9051dd278ba49e33a36e02befca6dbf415157617453baf9b3694faedf891799f9b24bcaef9c018567a13323d06a9  
25ce8cd2e23d96d51b5437434db2c3cef990f7f49eb94f8a190da021a074180feba9af17409d3fed21a1988c60a87d  
a8ecde02dba20dbd2440bf7598f7b1cff619d4642cf94ae460ff39167bfebe9b9cf920069f64265f71d52881e2abe3f

3a47269ffa273c3ff16f3b2f5b3eff7046d1886834dc241b8ad2f6886fb5bb14262204ac8abfaa11d2fb285e08474b  
c36b1fb0119b45480beabdbff4bc7b17f2e518a4eb39cde75430b1dcf9a86d1bfab414e46b3584b15c91cf116b5c03  
32fc9fbbfaa9213af9d6e999b22cc3dac2876b6f373c0cda7378707319404adfa5af3e6d3b867117a1655c2b8e67d89  
15f64716a9b4956a325f580cde64b9a399c537872b41e8a7bb628d87c888ed40da9f5b2598e099a982a48455ec64e8  
cc5e050af7e6cd8b299f07a2e3bb8eaa8f57544787fa0523135c6a57a2b318789ee126dd3f0e7d753c507bdd8d1lee  
d95fc78cb64516dbb34f9d62c9dc1c25152b4db498da85f7698c837c37a7a5537bac41b6d7aec087c6e6e39175fc  
3d3a7bb5bb77be3d57bb3f71f6ee2b8fbcb752fa11b4cb037c6a2b2cbe70eb21eda028bfcdbbb0ae7fff571afdf841d  
710416a609ebb19488331810ad4cdc954cd10890d3314bd6594fc3237d9c77574727e7796a38f77d64ad4794621fb2  
c40b6681e1f69644faee943a8f066ba974bd59506905d379273312f118c94a1658d81128b49a7932a31dcfe21f6709  
f1d3b2418ee3286ec920ff5c8f9ad096417a006fe300e7be23895e341042aaf47912dbb4f9f70392f9774e8b7d8a24  
7b376bd6f5a5282666a896494a702da3aeb9962357d1fbf6eccf98bac3ca02072c6b70e2dc8d8d53d9ca741c868788c  
d4b9c7ac3be600dd167a1d6bb612b623812ccbbdc877e5d518c5c33a14992cf9b18767c768ea28e12c90089aa9ce392  
0c4c4745215f29589705b1ac4cc750a351eba6bbf206a26d1a88c0e292f032aecf81985c08395491cc8b327b39e084  
74ba2b91f747d8acba24c0eb83a4a269ea288702d931639b1f7b1008cc186dc206a73b7f374f303702611a1382bab  
245fa08d8103e1104067c719601271efb9de3a967e29dcbbf21df34fbd06989ce6ce3b21aea0e73d5dc2753fd3ec90  
8f58fa04d1c424da8508bca0a7c770ec1fac7fd6db62e859337dda807068fb55fb503795c0c15726301650c210a115  
2f78e42d6ca110f548b5aa59f6b11b4081aaee6b55fc0e0ce66c578580c57d178eb13a09b0db78bb82af37234eda37  
adc467aec54059bf9f9c2f7b144b1ebcbe0b7e0ed2ee213c611091d505b733839a23ee3a487301a2fdaef7645d3efc  
cba588c87f7376f589a28d890531c6277d596defe5363aef58a4d361179876bde107cd6512d3b832cbea523ad9232d  
7c29d8ca4a42e71fde3c97256cae02b538a9501a37e13e59990ed68348ff9cc56f75b6e806c4a2e9ab84bf05a58f0f  
e4a1980201f5077b9536cd9343375bddc020e2521ef4f05e78db4c8b6fa6e74e7eaf5963774b8b8666997a275a0987  
6be15e740220b7cb882077a478184d2ee905edbd5d0a0c2f45600751406dcd3e1138d7b8f8acdc65518adc8b36cc640  
bf87ad38a2fe83a0983069753b275935fcc050484d4c3fb4a7fb60e65bc1e74b93c4152d82a38bd65d94f39e93ec3e  
cd15da6034d0a2d95e3f8a1174ccb8c263e7d661ec745f1127450243c62b445d45cfba20a7d0bc7ed5376291a8f50c  
5f2b26b0e3a4ec886950ac4042437afc81134e4d71dc0ac3b897ede2074e6ee290e1673cfeed66c796d5c6b83a858a  
4ec15a7abe574c11857eca5de8b78a4192b29e5e0ab8a06c75ac722fd07429a3e588d1e4557b02fdbd56a91949c08a  
650e3f7f84878983ee33b86eac1e7aa8b9a9065864bf15191f2be9d2e3c26bebc38129f4150db774056f99423a7a99  
1a77ff98c14b4eb61089bdfa6165e887f2bbcc03abd69d5833bd441245bebb9c6b0ab747b41cbb350a4609562cfc29b  
a83f5087e0b8eae03819af80429094a6a9b8b2e2feb1b01ec48d3a59cc03dcb1dec0b83c64de2001dc6b5c2d3a9d6c  
150c14c5c80441157c4d8f3alb5eea96b408b3ebfdfae0f048423b699a6a19f66664a6903960a7a36de2e176c34279  
e78c2745d5f98cc400a8ae0e60fd68dca60b3dd5f3e278fbfdff56181275d13ddbaa8506d56f5276b7174ea12302ac  
6f5165eb18855d0a06f2cclc16e1141083fb9200a575c827722dd9a5361c952d0fbec1eb453674e959e91cd0d8ebe6  
36027ca144a8ccaa654e261a0cb284de680c3b8a36224b198ef2884a2ebd326489cb4e7dbd05e43c1180277843947b  
02c2bc42838dd9bdb715d93e79e3ae8786c06ff1602d4e5d35a6c60ed08040a764206c57e466460468fe5780b99db4  
01812703e3492807e1d337d4c81b8115f2772a3daa81055098f7f9bcd8306a0fe664a274a420bb362feff7ee0a5073  
01eabf8981dda8b6972beb52ab4fe61f0da32e054f19893ed36519c1411420daaa5e77ddd3860f434e17826b6b89ea  
89cd36901f7c46a43d8ddae99f15a9a264603fa4f7dfd07edbb459460e5af59ac34be04c04cdfd5bb61203c1f6e39  
20b9525ee39a289e1807f93bbcc93962bcc754da13044cfaec6be157a26bb9ea7ceed24c44d2bc7b87f0a47f67c3237  
41ab414929f29dd98ade35a212db44eb953ff3ec09494f09b3f984e7e32d5c3064f6aa1b45f31c2697f30206ad743e  
d1591cae215f39ac444f3aa29f6a8a9eff1b758ae5bcccd872eb7e332f1ea4bff68a23a77c8c4d5080ed2a64876045  
c2aaa4fd7348f982dab7f567b45e382b47fe973dec8ffdd295d00737ed237a1686717a8294ed44e2879fafa46ef9de  
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 z kasy fiskalnej:

eyJraWQiOiIzQTAwMDAwMDA4RTY4RjU1ODE1RDRFQTYzQTAwMDEwMDAwMDAwOCwgQ049ZUthc3ktU3ViQ0EiLCJlbmMiOi  
JBMTIiLQ0JDLUhTMjU2IiwiaWYwXnIjoilU1NBMV81In0uZiC4C064EjKUDWmh0TmoQVUxXbXByZ5yW053Hxda3Je1gRchNmnq0  
s38RXyJt9L1e3SDe5hnZuVtPgKufUBgEblItpryumGYnqyhuzIbD0m8akTq9JyJHQ7SERZLIGYIzAgQbIn7NJAKswtzhpP  
56PxSnMRwegdx0PoW-Z1Tx2dYSphRobWYvpHjz4t25H\_poYZh2nAmmzC4nWOGnlshNI0qxk21E64\_Tb-  
4ACpoqv6k6WlPGygUYT4M1hoAN4w\_P0fsTyQxcmSxt3RpficbCROY2oLyQYGVYnTodbpLp7T4kVKN3XnzT6szBFzMCJsQ9B  
0Ug26mQARkH0cx7FWOtVCg.x3NlSpdTUDFhmlJfQoXz3Q.jnYxL4MrKbWmgY5ZeUrL\_etV-  
byYTeEwp8fH1j5q8ii54kYWQPUBERh3APg7KL18ZjY644HdbcZgy168x0cACpVhh5MqA93K5CX10e-  
4VUH97iBcHo\_VdzUW\_iqX3HPYf1lWLRt2PgIiqFOuzjapVYiVfzmXwhKsoWx4oYzIjFUJYM1QfCHnPVWPON1TnpDV6VRss

km3oC4sCn9DW0TJ-  
h8HHzvXTl8n6FlSHLrqcVgfb4cpz5w6X\_k8TBPXaJNgOvEgcbUTXNkKkUWs2xTKNdJAinKTzXR9WfsPajd6R5brwVIEpRg  
sHy6icr4RyxGO7fPS8JmKsbEY2CqedVO62ZtYREH1K2giji5lXPEpLPa7kFHdJ4uknjOjbgK-\_Kbb9BUVdhdfO6-  
bNpT67fiReZ--JLYu-cAYVnoTMjOGqSXOjNLiPZbVG1Q3Q02yw875kPf0nrlPihkNoCGgdBgP66mvF0CdP-  
0hoZiMYKh9qOzeAtuiDb0kQL91mPexz\_YZ1GQs-UrkYP85Fnv-vpvPkgBp9kJl9x1SiB4qvj86Ryaf-  
ic8P\_FvOy9bPv9wRtGIaDTcJBuK0vaIb7W7FCYiBKyKv6oR0vsoXghHS8NrH7ARmOVIC-  
q9v\_S8exfy5Rik6znN51Qwsdz5qG0b-rQU5GslhLfckc8RalwDMvyfv6qSE6-  
dbpmbIsw9rCh2tvNzwM2nN4cHMZQEr fpa8-  
bTuGcRehZVwrjmfYkV9kcWqbSVajJfWAzeZLmjmcU3hytB6Ke7Yo2HyIjtQNqfWyWY4JmpgqSEVexk6MxeBQr35s2LKZ8H  
ouO7jqQPV1RHh\_oFIxNcaleisxh4nuEm3T8OfXU8UHvdjRHu2V\_HjLZFFtuzT51iydzcHCUVK020mNqF92mMg3w3p6VTe6  
xBtteuWIfG5uORdfw9Onulu3e-PVe7P3H27iuPvLdS-  
hg0ywN8aissvnDrIe2gKL\_Na7Cuf\_9XGv2EHXEfYJ67GUIdMYEK1M3JVM0QiQ0zFL111PwyN9nHdXRyfnWo4931krUe  
UYh-yxAtmgeH21kT67pQ6jwZrQXS9WVBpBdN5JzMS8RjJShZY2BEotJp5MqMdz-  
IfZwnx07JBjuMobskg\_lyPmtCWQXoAb-MA574jiV40EEKq9HkS27T59wOS-  
XdOi32KJHs3a9b1pSgmZqiWSUpwLaOuuZyYV9H7\_sz5i6w8oCBYxrcOLcjYlT2cp0HIaHiM1LnHrDvmAN0WehlrthK2I4E  
sy9yHf11RjFwzUmSz5sYdnx2jqKOEskAiaqc45IMTEdFIV8pWJcFsaxMxlCjUeumu\_IGom0aiMDikvAyrS-  
BmFWiOVSrZIsyiezngHHS6K5H3R9isuiTA64OkomnqKIcC2TFjmx97EAjMGG3CBqc7fzdPMDcCYRoTgrq8JF-  
gjYED4RBAZ8cZYBJx77neOpZ-  
Kdy\_8h3zT70GmJzmzjshrqDnPV3CdT\_T7JCPWPoE0cQk2oUIvKcNx3DsH6x\_1tti6FkzfdqAcGj7VftQN5XAwVcmMBZQwh  
ChFS945C1soRD1SLWqWfaxG0CBqu5rVfwODOZsV4WAxX0XjrE6CbDbeLuCrzcjTto3rcRnrsVAWb-  
fnC97FEseVl4Lfg7S7iE8YRCR1QW3M4OaI-46SHMBov2u92RdPvzLpYjIf3N29YmijYkFMcYnfVlt7-  
U2Ou9YpNNhF5h2veEHZwUS07gyy-pSotkjlXwp2MpKQucf3jyXJWyuArU4qVAaA-E-  
WZk01oNI\_5zFb3W26AbEoumrhL8FpY8P5KGYAgH1B3uVNs2TQzdb3cAg41ie9PBeenTmi2-m505-  
r11jd0uLhmaZeidaCYdr4V50AiC3y4ggd6R4GE0u6QXttCdGwvRWAHUUBtzT4RONE4-KzcZVGK3Is2zGQL-  
HrTii\_oOgmDBpdTsnWTX8wFBITUw\_tKf7YOZbwedLk8QVLYKji9ZdlPOek-w-  
zRXaYDTQotleP4oRdMy4wmPnlmHsdF8RJ0UCQ8YrRF1Fz7ogp9C8ftU3YpGo9QxfKyaw46TsiGlQrEBCQ3r8gRNOTXHCs  
O41-3iB05u4pDhZzz-7WbHltXGuDqFik7BWnq-  
V0wRhX7KXeI3ikGSSp5eCrigbHWsci\_QdCmj5YjR5FV7Av29VqkZScCKZQ4\_f4SHiYPuM7hurB56qLmpBlhkvxUZHyvp0u  
PCa-vDgSn0FQ23dAVvmUI6epkad\_-YwUtOthCJvfphZeiH8rvAOrlp1YM71EEkW-  
u5xrCrdHtBy7NQpGCVys\_Cm6g\_UIfguOrgOBmvgEKQlKapuLLi\_rGwHsSN0lnMA9yx3sC4PGTeIAHcalwtOp1sFQwUxcgE  
QRV8TY86G17qlrQIs-v9-  
uDwSEI7aZpgGfZmZKaQOWCno23i4XbDQnnnjCdFlfmMxACorg5g\_Wjcpgs91fPiePv9\_1YYenXRPduqhQbVb1J2txdOoSM  
CrG9RZesYhV0KBvLOHBbhFBCD-  
5IApXXIj3It2aU2HJUTd77B60U2d0lZ6RzQ2OvmNgJ8oUSozKplTiYaDLKE3mgMO4o2IksZjvKISi69MmSJy059vQXkPBG  
AJ3hDlHsCwrxCg43ZvbcV2T55466HhsBv8WAtTl01psY00IBAp2QgbFfkZkYEaP5XgLmdtAGBJwPjSSgH4dM31MgbgRXyd  
yo9qoEFUJj3-bzYMGOP5mSIdKQguzYv7\_fuClBzAeq\_iYHdqLaXK-tSq0\_mHw2jLgVPGYk-  
02UZwUEUINqqXnfd04YPQ04XgmtrieqJzTaQH3xGpD2N2u-Z8VqaJkYD-  
k99\_Qftu0WUYOWwWaw0vgTATN\_Vu2EgPB9uOSC5U17jmiieGAf5O7yTlivMdU2hMETPrsa-  
FXomu56nzu0kxE0rx7h\_Ckf2fDI3QatBSSnyndmK3jWiEtte65U\_8-wJSU8Js\_mE5-  
MtXDBk9qobRfMcJpfzAgatdD7RWRyuIV85rERPOqKfaoqe\_xtliuW8zNhy634zLx6kv\_aKI6d8jE1QgO0qZIdgRcKqpP1z  
SPmC2rf1Z7ReOCTH\_pc97Nj\_0pXQBzftI3oWhnF6gpTtROKHn6-kbvne249vASSPM70U09y8PXyMK-  
kQGT8GGsmreAlWh1SFOMgiN-cfRrlGxHTAn0w.-x\_YGNfKRdmlmIdWG7qxEA



```
-----BEGIN CERTIFICATE-----
MIIDKjCCApOgAwIBAgIQ27uolIpDHKdAJJLSAHD4+zANBgkqhkiG9w0BAQsFADAU
MRIwEAYDVQQDEwlhNRIbLlUthc3kwHhcnMjAwMwE2MTA2NTU5WWhcnMjUwMwE2MTA2
NTU4WjCBrrEjMCEGAlUECXMARGVWYXJ0YWI1bnQgSW5mb3JtYXR5emFjamkxHjAC
BgNVBAMoTFU1bmlzdGVyY3R3bGyBAW5hbnnVdzE2MBcGA1UEBRMjVkfFUVUWY2Njk3
MDANBgMdgMjFjZWMBQGA1UEAxMNVN1RFMjA1WjAwTAWMAWDAOTERMAAGA1UEBxMUE2Fw3ph
d2ExFDBASBgNVBAGTC01hem93aWVja2llMQswCQYDVQQGEWJQTDCCASiWdQYJKoZI
hvcNAQEBBQADggEPADCCAQoCggEBALtSH7AwSOT4tQ7863BAWbt2DvGqEgDNV/g0d
tWK81C5vK677mb9Y7U0PccrJG4ht7B+kmRvFs092YgnjmqPEqAJj4GyCicGXa1+6
f6xeUQknwvpaHvpmPGvZjlsAmHyScLdGqOXzQgrxv/WE64jJgendcucsGu6AZHRC
w7rxGMAXq2YsGepNZbQq8hphl7sAjjoPHZCJWU5PP1nsNzddSOEJndJYHNbbrom
62FLbjuzXQMk/gWpFACgY9vhoHio07BCSiJ473FC4tpmhxOPqmdZDYwInSU0MgaYP3
p803z2f16g1kg1T21dt4GroubChiffw4RumFwuiPZSe8DKD834PQ8CAwEAAANeMfww
EwYDVR0lBAwwCgYIKwYBBQUHAwIwRQYDVDR0BBD4wPIAoxg+n+jlJAU1f23nuS65X
PaEWMBQXejAQBgNVBAMTCU1GIGUtS2FzeYIQs8LVE29uP51AlnvXe8bYeJANBgkq
hkiG9w0BAQsFAAOBgQBZfK/KxEy/3z1fP0r4Nt5Dg5LbJ737kRf1ZRhu3PHmMthA
ac2kTULtaaaWC+iL70nm8t/J9kRD9UyK83vLmLbPGVEbcOlY+50o5XYetubSdLfNl
m4kifiFDh5NP4AKZQdTpCp4jbaAOWRFJCbrrZzBv6M0XytAE/mlrTrml9K313MxRw==
-----END CERTIFICATE-----
```

## C.2 Przykładowe dane procesu podpisywania danych kas w postaci oprogramowania

### C.2.1 Przykładowe zbiory danych

#### C.2.1a Pełna postać przykładowego zbioru danych:

```
{ "JPK": { "naglowek": { "wersja": "JPK_KASA_v2-0", "wysylka": "Z", "dataJPK": "2020-04-10T10:23:45.678Z", "podmiot1": { "nazwaPod": "Nazwa podmiotu", "nrFabr": "WTE2001000009", "NIP": "6970000802", "adresPod": { "ulica": "Ulica", "miejsc": "Miejscowosc", "nrLok": "NrLok", "poczta": "Poczta", "nrDomu": "NrDomu", "kodPoczt": "00-000", "nrUnik": "WTE2001000009", "nrEwid": "2020/000001949", "content": { "rapFisk": { "zakRap": "2020-04-10T01:23:45.789Z", "firmwareId": "01", "sumaZm": "9DB845A0C9C285DC2E13F3253352E1836DC603C0", "typWl asn": "1", "kodUS": "3014", "serwID": "KW123", "pamiecChr": 1, "sposobUzytk": "4", "licznikParag": 1, "JPK ID": 1, "dataFisk": "2020-04-10T01:23:45.678Z", "kasjer": "Kasjer", "stPTU": [ { "id": "A", "wart": 2300 }, { "id": "B", "wart": 800 }, { "id": "C", "wart": 500 }, { "id": "D", "wart": 0 }, { "id": "E", "wart": 0 }, { "id": "F", "wart": 0 }, { "id": "G", "wart": 0 }, { "id": "Z", "wart": 0 } ], "model": "PREFIX DLA TESTÓW KW", "waluta": "PLN", "katKasy": [ "01" ], "nrDok": 1, "nrKasy": "001", "podpis": { "RSA": "3019505879124143c92b60d5e57ebc08ba8fb7c9d2celbae02ba6b77a9c14b7794f7f8acbddff2c2e382fbb6820cd0e0a87fa988f2c67f9b34ae0571cfc0b23b727b7912c9d30b567c56e710cbfa88f09937bbfd1c54065df893eafb7455823716c292d548dc98dfc4ad29c81332450b84bea1a4bbf11e0c4dcc8d21f835476b87ff5a68be7d514fe15d3ebb76c2559bfe93a8cdfd09bf9bab7b5df918225c2f46778f59fb9deef898eb4e932f405a1403d9e0e78772c25675f14edc34d457f42aeb0a719a7a9638654bae7823641b1f95981b37fbd2cff3fda256e2e8f2e5aa8aca729e2d9a9b32893d6664b223da5c1879afcec64fea8b11fe22a622de55bb" } } }, { "zdarzenie": { "pamiecChr": 1, "JPKID": 2, "dataCzas": "2020-04-10T02:23:45.678Z", "typ": { "21": { "atrybuty": { "checkFreqWebApi": 100, "sendFreqEventHub": 100, "shipmentType": "RF", "info": "Zapis harmonogramu transmisji danych w bazie danych kasy" } } } }, { "wydrNiefisk": { "pamiecChr": 1, "zak": "2020-04-10T03:23:45.678Z", "JPKID": 3, "kasjer": "Kasjer", "grafika": 1, "nrDok": 2, "zawart": [ { "kodGraficzny": "kodGraficzny0", "kodKreskowy": "kodKreskowy0", "tekst": "tekst0" }, { "kodGraficzny": "kodGraficzny1", "kodGraficzny2": "kodKreskowy": [ "kodKreskowy1", "kodKreskowy2" ], "tekst": [ "WYDRUK NIEFISKALNY", "TEST" ] } ], "nrKasy": "001", "podpis": { "RSA": "282f2b77102bbb60717093d6ad9c2db648201be698baa316c03e7598446fc7915042e9d4a19d5a846ad5998c7467eceddlf545dc6619d6c0a17e55f9380d01e584dff a84971bc9e91f922e6f2f4fe243c1382b8fa067809451241992622faa9e7b83feb4a3d6d8f742278c1e5174570a63b113495384a346ad6f1e0f5072d91c9227ea68abbbbc9a902f0463a126fbb43ef1380dac8421cd8d45cb797ab75a7c3b08988709e64bc174d9aeaad5e9eala44d01e0c3e0a7ecb97baec560f34f4a6ce34d27749d5b220384ce2d9add085e77a947be2b532c55ce0f5a92b9266cc262580ce7d56c528792d5a908b8a503b4188e3e77f3230cb53fc5e1cbb9c79fbl" } } }, { "paragAnul": { "pamiecChr": 1, "total": { "zap1Zwrot": 1000000 }, "zakSprzed": "2020-04-10T06:23:45.678Z", "JPKID": 6, "nrParag": 3, "kasjer": "Kasjer", "stPTU": [ { "id": "G", "wart": "ZW" } ], "p dsum": { "waluta": "PLN", "sumaNetto": [ { "brutto": 1000000, "vat": 0, "idStPTU": "G" } ], "sumaBrutto": 1000000, "sumaPod": 1000000, "nrDok": 5, "pozycja": [ { "towar": { "brutto": 1000000, "ilosc": "1", "oper": false, "cena": 1000000, "nazwa": "Nazwa towaru", "idStPTU": "G" } } ], "nrKasy": "001", "podpis": { "RSA": "9e61a538a24334c0af624987d5967c11d89286f02bcd1c9123ff284c460e3dcf2ae7cca47c1a5079010b201d2a6ace3c1d619377c57501a3773e7faaa3ac3c2ea7e b26409b3ff852d715cf1c16cc1b490e7876bc23af51720174c729a2c8aea7d4a58cf3d99b54582698764e58922af469d011cbc5100bb57bbbcf3f373bf629a33257eec641662c3f44cbc9d39c7d7f885a1b8fcd6da954e918077f7ff3279e2f84ea9aa944ab9123dd10f711870a8d1d6856ecaa6a6df0a7bce1703fd79376bec82cec37ea4ba48a64063ebcdf4 d99f9e18c37d0777bd4437ca819d19b2a8b2f73cfff076cb240ee0580d90eaba0c7aad3dff10af58e64c03e51453caa1c8f3", "JPK": "000000000000000000", "SHA": "eee0e3068482a34527630f4c73c05d0d3523899f5fa25d022b9415ff8057d3be" } } }, { "paragAnul": { "pamiecChr": 1, "total": { "zap1Zwrot": 1000000 }, "zakSprzed": "2020-04-10T09:23:45.678Z", "JPKID": 9, "nrParag": 6, "kasjer": "Kasjer", "stPTU": [ { "id": "G", "wart": "ZW" } ], "p dsum": { "waluta": "PLN", "sumaNetto": [ { "brutto": 1000000, "vat": 0, "idStPTU": "G" } ], "sumaBrutto": 1000000, "sumaPod": 1000000, "nrDok": 8, "pozycja": [ { "towar": { "brutto": 1000000, "ilosc": "1", "oper": false, "cena": 1000000, "nazwa": "Nazwa towaru", "idStPTU": "G" } } ], "nrKasy": "001", "podpis": { "RSA": "ab6c66f6f3c3c34dde1e8f3b3080bee64555a b5f3886bae21f4be9139a11f6c09bc8765f229bfd99bece7fb7dbb5cf18fffc17e7cfbd56bc8b637d4a821e483d8b6c 5cc947e356a543b5eba1020a499fe7506d49c5d3fbd61dbd428124783fe0ee6532f57b7600d8266435eda9e5fead4 7c4d31b18961fe21ad824ccc6b20d51961a6221bcef4554322f3127c9e488aaf6a54d0498096484840668372b5cf6b 22a73a26334249a74fb4c87568dbde4693dfeed4ec44f473dcf4921c3da287f3e8b1085521a833b416c7a269fa6d03 ed23ea1fb9e9577c78ae60179b4303629674f1fdfbacaed55706a7b466957b93c083d1e3f4af231f503cdb694b9afb 0ebf3", "JPK": "0010000000000000006", "SHA": "a64a913986f2a18d4db5ecbe02309fc96d19c683830622140a2d4 310425f911c" } } }, { "paragAnul": { "pamiecChr": 1, "total": { "zap1Zwrot": 1000000 }, "zakSprzed": "2020-04-10T10:23:45.678Z", "JPKID": 10, "nrParag": 7, "kasjer": "Kasjer", "stPTU": [ { "id": "G", "wart": "ZW" } ], "p odsum": { "waluta": "PLN", "sumaNetto": [ { "brutto": 100000000, "vat": 0, "idStPTU": "G" } ], "sumaBrutto": 100000000, "sumaPod": 100000000, "nrDok": 9, "pozycja": [ { "towar": { "brutto": 100000000, "ilosc": "1", "o
```



```

per":false,"cena":100000000,"nazwa":"Nazwa
towaru","idStPTU":"G"}]],{"nrKasy":"001","podpis":{"RSA":"54ce7bad1079bc8a7a69908766aa6e420edf6
11442f28f28cfbd51c3c181ad545055ff81465ab6dbbd93fcdf0fb78b83ef56304e83fb302bf0e07c2114b7a947ad20
c012d204efef1cd53aaff0469830aabe01a8a5828c798872bd6dbb7d8767ab832d892a40f81c07d7246e34f8716d5cd
0fd206da3bbb89aae831010f3e2278a6ce76c9206b022e6a246a0b889cbe571c062bf214a3dd924168f4913bde545d
ba76f42875bb1c7060fd06ded94f9c9ab7e9f2d04f276b9ccfaebdbdac48cc92b5b4f629504daca05d887a1cd8e035
e50d7b07af374609f3bd6080e4bed80330d62924987d440f0b06f7e99a3ec980df0e1951e073854e20173a937ec2f1
ca93f","JPK":"001000000000000009","SHA":"ac6450833bef0f47888dea1272d179e8a197c8b6968c16782eeaf
f4f2769430d"}]],{"rapDob":{"sprzedBrutto":49060,"zakSprzed":"2020-04-
10T22:23:45.678Z","liczbaParAnul":3,"zakRap":"2020-04-
10T23:23:45.678Z","sprzedPar":{"wartWgPTU":[{"netto":22000,"vat":5060,"idStPTU":"A"},{"netto":
20200,"vat":0,"idStPTU":"G"}],"sumaBrutto":49060,"sumaPod":5060},"podatekNal":5060,"dokNiefisk
":1,"liczbaPar":4,"pamiecChr":1,"JPKID":11,"kasjer":"Kasjer","stPTU":[{"id":"A","wart":2300},{
"id":"G","wart":"ZW"}],"waluta":"PLN","wartParAnul":111000000,"rozpSprzed":"2020-04-
10T04:23:45.678Z","nrDok":10,"nrRap":1,"nrKasy":"001","podpis":{"RSA":"21184e6e94b7c822223c90b
3769567188c285c262d10be052c6820242368a5f655af1c77138930fe51873f2e2ee6c6a7666f3f8cf9f1776dad23
ce32afb687f6b5057bdf38ce3bb732ff27d95b1682c541b43410af8cad7262bc93dc7466b09588311242b4c297251a
e54afd8911607a482854a8bb326b68ae7c8d1295266488ee635a483e90157f8c943a3add99329edbc2ca18d9954e97
6f32f4bda9d74eb99527f411436c8857d4b8e804e72b0c0e6a58e47a6975749b8421874f095041cc5429d9d18b3f44
6f0c834964e2150c2f56b6de9e72f62cfa5f194aa5650a55b64f5f903d52b9435e84183860810835e5a266aef281b9
2ab8a257d9467f4c1ce","JPK":"000000000000000000","SHA":"7d68d863163c76b46d26e543d44d0f7d373beef
55d055f3ff6929484d9e374f4"}]]}}}}

```

## C.2.1b Ustrukturyzowana postać przykładowego zbioru danych:

```

{
  "JPK": {
    "naglowek": {
      "wersja": "JPK_KASA_v2-0",
      "dataJPK": "2020-04-10T10:23:45.678Z",
      "wysylka": "Z"
    },
    "podmiot1": {
      "nazwaPod": "Nazwa podmiotu",
      "nrFabr": "WTE2001000009",
      "NIP": "6970000802",
      "adresPod": {
        "ulica": "Ulica",
        "miejsc": "Miejscowosc",
        "nrLok": "NrLok",
        "poczta": "Poczta",
        "nrDomu": "NrDomu",
        "kodPoczt": "00-000"
      },
      "nrUnik": "WTE2001000009",
      "nrEwid": "2020/000001949"
    },
    "content": [
      {
        "rapFisk": {
          "JPKID": 1,
          "pamiecChr": 1,
          "nrDok": 1,
          "dataFisk": "2020-04-10T01:23:45.678Z",
          "stPTU": [
            {
              "id": "A",
              "wart": 2300
            },
            {
              "id": "B",
              "wart": 800
            },
            {
              "id": "C",
              "wart": 500
            },
            {
              "id": "D",
              "wart": 0
            }
          ]
        }
      ]
    ]
  }
}

```

```

    },
    {
        "id": "E",
        "wart": 0
    },
    {
        "id": "F",
        "wart": 0
    },
    {
        "id": "G",
        "wart": "ZW"
    }
],
"katKasy": [
    "01"
],
"licznikParag": 1,
"model": "PREFIX DLA TESTÓW KW",
"typWlasn": "1",
"sposobUzytk": "4",
"firmwareId": "01",
"sumaZm": "9DD845A0C9C285DC2E13F3253352E1836DC603C0",
"kodUS": "3014",
"serwID": "KW123",
"waluta": "PLN",
"nrKasy": "001",
"kasjer": "Kasjer",
"zakRap": "2020-04-10T01:23:45.789Z",
"podpis": {
    "RSA":
"3019505879124143c92b60d5e57ebc08ba8fb7c9d2ce1bae02ba6b77a9c14b7794f7f8acbddff2c2e382fbb6820cd
0e0a87fa988f2c67f9b34ae0571cfc0b23b727b7912c9d30b567c56e710cbfa88f09937bbfd1c54065df893eafb745
5823716c292d548dc98dfc4ad29c81332450b84beala4bbf11e0c4dcc8d21f835476b87ff5a68be7d514fe15d3ebb7
6c2559bfe93a8cdfd09bf9bab7b5df918225c2f46778f59fb9deef898eb4e932f405a1403d9e0e78772c25675f14ed
c34d457f42aeb0a719a7a9638654bae7823641b1f95981b37fbd2cff3fda256e2e8f2e5aa8aca729e2d9a9b32893d6
664b223da5c1879afceec64fea8b11fe22a622de55bb"
}
}
},
{
    "zdarzenie": {
        "JPKID": 2,
        "pamiecChr": 1,
        "dataCzas": "2020-04-10T02:23:45.678Z",
        "typ": {
            "21": {
                "info": "Zapis harmonogramu transmisji danych w bazie danych
kasy",
                "atrybuty": {
                    "sendFreqEventHub": 100,
                    "checkFreqWebApi": 100,
                    "shipmentType": "RF"
                }
            }
        }
    }
},
{
    "wydrNiefisk": {
        "JPKID": 3,
        "pamiecChr": 1,
        "grafika": 1,
        "nrDok": 2,
        "zawart": [
            {
                "kodGraficzny": "kodGraficzny0",
                "kodKreskowy": "kodKreskowy0",
                "tekst": "tekst0"
            },
            {

```

```

        "tekst": [
            "WYDRUK NIEFISKALNY",
            "TEST"
        ],
        "kodKreskowy": [
            "kodKreskowy1",
            "kodKreskowy2"
        ],
        "kodGraficzny": [
            "kodGraficzny1",
            "kodGraficzny2"
        ]
    },
    "zak": "2020-04-10T03:23:45.678Z",
    "nrKasy": "001",
    "kasjer": "Kasjer",
    "podpis": {
        "RSA":
"282f2b77102bbb60717093d6ad9c2db648201be698baa316c03e7598446fc7915042e9d4a19d5a846ad5998c7467e
cedd1f545dc6619d6c0a17e55f9380d01e584dffa84971bc9e91f922e6f2f4fe243c1382b8fa067809451241992622
faa9e7b83feb4a3d6d8f742278c1e5174570a63b113495384a346ad6f1e0f5072d91c9227ea68abbbc9a902f0463a1
26fbb43ef1380dac8421cd8d45cb797ab75a7c3b08988709e64bc174d9aeead5e9eala44d01e0c3e0a7ecb97baec56
0f34f4a6ce34d27749d5b220384ce2d9add085e77a947be2b532c555ce0f5a92b9266cc262580ce7d56c528792d5a9
08b8a503b4188e3e77f3230cb53fc5e1cbba9c79fb1"
    }
},
{
    "paragAnul": {
        "JPKID": 6,
        "pamiecChr": 1,
        "nrDok": 5,
        "pozycja": [
            {
                "towar": {
                    "brutto": 1000000,
                    "cena": 1000000,
                    "idStPTU": "G",
                    "ilosc": "1",
                    "nazwa": "Nazwa towaru",
                    "oper": false
                }
            }
        ],
        "stPTU": [
            {
                "id": "G",
                "wart": "ZW"
            }
        ],
        "podsum": {
            "sumaNetto": [
                {
                    "idStPTU": "G",
                    "brutto": 1000000,
                    "vat": 0
                }
            ],
            "sumaPod": 1000000,
            "sumaBrutto": 1000000,
            "waluta": "PLN"
        },
        "total": {
            "zaplZwrot": 1000000
        },
        "nrParag": 3,
        "nrKasy": "001",
        "zakSprzed": "2020-04-10T06:23:45.678Z",
        "kasjer": "Kasjer",
        "podpis": {

```

```
        "RSA":  
        "9e61a538a24334c0af624987d5967c11d89286f02bcd1c9123ff284c460e3dcf2ae7cca47c1a5079010b201d2a6ac  
e3c1d619377c57501a3773e7faaa3ac3c2ea7eb26409b3ff852d715cf1c16cc1b490e7876bc23af51720174c729a2c  
8aea7d4a58cf3d99b54582698764e58922af469d011cbc5100bb57bbcf3f373bf629a33257eec641662c3f44cbc9d  
39c7d7f885a1b8fcd6da954e918077f7ff3279e2f84ea9aa944ab9123dd10f711870a8d1d6856ecaa6a6df0a7bce17  
03fd79376bec82cec37ea4ba48a64063ebcdf4d99f9e18c37d0777bd4437ca819d19b2a8b2f73cff076cb240ee0580  
d90eaba0c7aad3dff10af58e64c03e51453caa1c8f3",  
        "SHA":  
        "eee0e3068482a34527630f4c73c05d0d3523899f5fa25d022b9415ff8057d3be",  
        "JPK": "000000000000000000"  
    }  
},  
{  
    "paragAnul": {  
        "JPKID": 9,  
        "pamiecChr": 1,  
        "nrDok": 8,  
        "pozycja": [  
            {  
                "towar": {  
                    "brutto": 10000000,  
                    "cena": 10000000,  
                    "idStPTU": "G",  
                    "ilosc": "1",  
                    "nazwa": "Nazwa towaru",  
                    "oper": false  
                }  
            }  
        ],  
        "stPTU": [  
            {  
                "id": "G",  
                "wart": "ZW"  
            }  
        ],  
        "podsum": {  
            "sumaNetto": [  
                {  
                    "idStPTU": "G",  
                    "brutto": 10000000,  
                    "vat": 0  
                }  
            ],  
            "sumaPod": 10000000,  
            "sumaBrutto": 10000000,  
            "waluta": "PLN"  
        },  
        "total": {  
            "zaplZwrot": 10000000  
        },  
        "nrParag": 6,  
        "nrKasy": "001",  
        "zakSprzed": "2020-04-10T09:23:45.678Z",  
        "kasjer": "Kasjer",  
        "podpis": {  
            "RSA":  
            "ab6c66f6f3cbc34dde1e8f3b3080bee64555ab5f3886bae21f4be9139a11f6c09bc8765f229bfd99bece7fb7dbb5c  
f18ffc17e7cfbd56bc8b637d4a821e483d8b6c5cc947e356a543b5eba1020a499fe7506d49c5d3fbd61dbd4281247  
83fe0ee6532f57b7600d8266435eda9e5fead47c4d31b18961fe21ad824ccc6b20d51961a6221bcef4554322f3127c  
9e488aaf6a54d0498096484840668372b5cf6b22a73a26334249a74fb4c87568dbde4693dfeed4ec44f473dcf4921c  
3da287f3e8b1085521a833b416c7a269fa6d03e1d23ea1fde9577c78ae60179b4303629674f1fdfbacaed55706a7b4  
66957b93c083d1e3f4af231f503cdb694b9afb0ebf3",  
            "SHA":  
            "a64a913986f2a18d4db5ecbe02309fc96d19c683830622140a2d4310425f911c",  
            "JPK": "0010000000000000006"  
        }  
    }  
},  
{  
    "paragAnul": {
```

```

        "JPKID": 10,
        "pamiecChr": 1,
        "nrDok": 9,
        "pozycja": [
            {
                "towar": {
                    "brutto": 1000000000,
                    "cena": 1000000000,
                    "idStPTU": "G",
                    "ilosc": "1",
                    "nazwa": "Nazwa towaru",
                    "oper": false
                }
            }
        ],
        "stPTU": [
            {
                "id": "G",
                "wart": "ZW"
            }
        ],
        "podsum": {
            "sumaNetto": [
                {
                    "idStPTU": "G",
                    "brutto": 1000000000,
                    "vat": 0
                }
            ],
            "sumaPod": 1000000000,
            "sumaBrutto": 1000000000,
            "waluta": "PLN"
        },
        "total": {
            "zaplwrot": 1000000000
        },
        "nrParag": 7,
        "nrKasy": "001",
        "zakSprzed": "2020-04-10T10:23:45.678Z",
        "kasjer": "Kasjer",
        "podpis": {
            "RSA":
"54ce7bad1079bc8a7a69908766aa6e420edf611442f28f28cfbd51c3c181ad54505ff81465ab6dbbd93fcaf0fb78b
83ef56304e83fb302bf0e07c2114b7a947ad20c012d204efe1cd53aaff0469830aabe01a8a5828c798872bd6dbb7d8
767ab832d892a40f81c07d7246e34f8716d5cd0fd206da3bbb89aae831010f3e2278a6ce76c9206b022e6a246a0b88
9cbe571c062bf214a3dd924168f4913bde545dba76f42875bb1c7060fd06ded94f9c9ab7e9f2d04f276b9ccfaebdbd
ac48cc92b5b4f629504daca05d887a1cd8e035e50d7b07af374609f3bd6080e4bed80330d62924987d440f0b06f7e9
9a3ec980df0e1951e073854e20173a937ec2f1ca93f",
            "SHA":
"ac6450833bef0f47888dea1272d179e8a197c8b6968c16782eeaff4f2769430d",
            "JPK": "0010000000000000009"
        }
    },
    {
        "rapDob": {
            "JPKID": 11,
            "pamiecChr": 1,
            "nrDok": 10,
            "nrRap": 1,
            "rozpSprzed": "2020-04-10T04:23:45.678Z",
            "zakSprzed": "2020-04-10T22:23:45.678Z",
            "stPTU": [
                {
                    "id": "A",
                    "wart": 2300
                },
                {
                    "id": "G",
                    "wart": "ZW"
                }
            ]
        }
    }

```

```

    ],
    "sprzedPar": {
      "sumaBrutto": 49060,
      "sumaPod": 5060,
      "wartWgPTU": [
        {
          "idStPTU": "A",
          "netto": 22000,
          "vat": 5060
        },
        {
          "idStPTU": "G",
          "netto": 20200,
          "vat": 0
        }
      ]
    },
    "podatekNal": 5060,
    "sprzedBrutto": 49060,
    "waluta": "PLN",
    "liczbaPar": 4,
    "liczbaParAnul": 3,
    "wartParAnul": 111000000,
    "dokNiefisk": 1,
    "zakRap": "2020-04-10T23:23:45.678Z",
    "nrKasy": "001",
    "kasjer": "Kasjer",
    "podpis": {
      "RSA":
"21184e6e94b7c822223c90b3769567188c285c262d10be052c6820242368a5f655af1c77138930fe51873f2e2eea6
c6a7666f3f8cf9f1776dad23ce32afb687f6b5057bdf38ce3bb732ff27d95b1682c541b43410af8cad7262bc93dc74
66b09588311242b4c297251ae54afd8911607a482854a8bb326b68ae7c8d1295266488ee635a483e90157f8c943a3a
dd99329edbc2ca18d9954e976f32f4bda9d74eb99527f411436c8857d4b8e804e72b0c0e6a58e47a6975749b842187
4f095041cc5429d9d18b3f446f0c834964e2150c2f56b6de9e72f62cfa5f194aa5650a55b64f5f903d52b9435e8418
3860810835e5a266aef281b92ab8a257d9467f4c1ce",
      "SHA":
"7d68d863163c76b46d26e543d44d0f7d373beef55d055f3fffb929484d9e374f4",
      "JPK": "000000000000000000"
    }
  }
}
]
}
}

```

## C.2.2 Przykłady ustrukturyzowanych dokumentów przesyłanych w zbiorach danych:

### C.2.2a Ustrukturyzowana postać przykładowego raportu fiskalnego fiskalizacji:

```

{
  "rapFisk": {
    "JPKID": 1,
    "pamiecChr": 1,
    "nrDok": 1,
    "dataFisk": "2020-04-10T01:23:45.678Z",
    "stPTU": [
      {
        "id": "A",
        "wart": 2300
      },
      {
        "id": "B",
        "wart": 800
      },
      {
        "id": "C",
        "wart": 500
      }
    ]
  }
}

```

```

        {
            "id": "D",
            "wart": 0
        },
        {
            "id": "E",
            "wart": 0
        },
        {
            "id": "F",
            "wart": 0
        },
        {
            "id": "G",
            "wart": "ZW"
        }
    ],
    "katKasy": [
        "01"
    ],
    "licznikParag": 1,
    "model": "PREFIX DLA TESTÓW KW",
    "typWlasn": "1",
    "sposobUzytk": "4",
    "firmwareId": "01",
    "sumaZm": "9DD845A0C9C285DC2E13F3253352E1836DC603C0",
    "kodUS": "3014",
    "serwID": "KW123",
    "waluta": "PLN",
    "nrKasy": "001",
    "kasjer": "Kasjer",
    "zakRap": "2020-04-10T01:23:45.789Z",
    "podpis": {
        "RSA":
"3019505879124143c92b60d5e57ebc08ba8fb7c9d2ce1bae02ba6b77a9c14b7794f7f8acbddff2c2e382fbb6820cd
0e0a87fa988f2c67f9b34ae0571cfc0b23b727b7912c9d30b567c56e710cbfa88f09937bbfd1c54065df893eafb745
5823716c292d548dc98dfc4ad29c81332450b84beala4bbf11e0c4dcc8d21f835476b87ff5a68be7d514fe15d3ebb7
6c2559bfe93a8cdfd09bf9bab7b5df918225c2f46778f59fb9deef898eb4e932f405a1403d9e0e78772c25675f14ed
c34d457f42aeb0a719a7a9638654bae7823641b1f95981b37fbd2cff3fda256e2e8f2e5aa8aca729e2d9a9b32893d6
664b223da5c1879afceec64fea8b11fe22a622de55bb"
    }
}
}

```

## C.2.2b Ustrukturyzowana postać przykładowego raportu fiskalnego dobowego:

```

{
    "rapDob": {
        "JPKID": 11,
        "pamiecChr": 1,
        "nrDok": 10,
        "nrRap": 1,
        "rozpSprzed": "2020-04-10T04:23:45.678Z",
        "zakSprzed": "2020-04-10T22:23:45.678Z",
        "stPTU": [
            {
                "id": "A",
                "wart": 2300
            },
            {
                "id": "G",
                "wart": "ZW"
            }
        ],
        "sprzedPar": {
            "sumaBrutto": 49060,
            "sumaPod": 5060,
            "wartWgPTU": [
                {
                    "idStPTU": "A",

```

```

        "netto": 22000,
        "vat": 5060
    },
    {
        "idStPTU": "G",
        "netto": 20200,
        "vat": 0
    }
]
},
"podatekNal": 5060,
"sprzedBrutto": 49060,
"waluta": "PLN",
"liczbaPar": 4,
"liczbaParAnul": 3,
"wartParAnul": 111000000,
"dokNiefisk": 1,
"zakRap": "2020-04-10T23:23:45.678Z",
"nrKasy": "001",
"kasjer": "Kasjer",
"podpis": {
    "RSA":
"21184e6e94b7c822223c90b3769567188c285c262d10be052c6820242368a5f655af1c77138930fe51873f2e2eea6
c6a7666f3f8cf9f1776dad23ce32afb687f6b5057bdf38ce3bb732ff27d95b1682c541b43410af8cad7262bc93dc74
66b09588311242b4c297251ae54afd8911607a482854a8bb326b68ae7c8d1295266488ee635a483e90157f8c943a3a
dd99329edbc2ca18d9954e976f32f4bda9d74eb99527f411436c8857d4b8e804e72b0c0e6a58e47a6975749b842187
4f095041cc5429d9d18b3f446f0c834964e2150c2f56b6de9e72f62cfa5f194aa5650a55b64f5f903d52b9435e8418
3860810835e5a266aef281b92ab8a257d9467f4c1ce",
    "SHA": "7d68d863163c76b46d26e543d44d0f7d373beef55d055f3ffb929484d9e374f4",
    "JPK": "000000000000000000"
}
}
}

```

## C.2.2c Ustrukturyzowana postać przykładowego dokumentu niefiskalnego:

```

{
    "wydrNiefisk": {
        "JPKID": 3,
        "pamiecChr": 1,
        "grafika": 1,
        "nrDok": 2,
        "zawart": [
            {
                "kodGraficzny": "kodGraficzny0",
                "kodKreskowy": "kodKreskowy0",
                "tekst": "tekst0"
            },
            {
                "tekst": [
                    "WYDRUK NIEFISKALNY",
                    "TEST"
                ],
                "kodKreskowy": [
                    "kodKreskowy1",
                    "kodKreskowy2"
                ],
                "kodGraficzny": [
                    "kodGraficzny1",
                    "kodGraficzny2"
                ]
            }
        ],
        "zak": "2020-04-10T03:23:45.678Z",
        "nrKasy": "001",
        "kasjer": "Kasjer",
        "podpis": {
            "RSA":
"282f2b77102bbb60717093d6ad9c2db648201be698baa316c03e7598446fc7915042e9d4a19d5a846ad5998c7467e
cedd1f545dc6619d6c0a17e55f9380d01e584dffa84971bc9e91f922e6f2f4fe243c1382b8fa067809451241992622

```



```

faa9e7b83feb4a3d6d8f742278c1e5174570a63b113495384a346ad6f1e0f5072d91c9227ea68abbbc9a902f0463a1
26fbb43ef1380dac8421cd8d45cb797ab75a7c3b08988709e64bc174d9aeead5e9eala44d01e0c3e0a7ecb97baec56
0f34f4a6ce34d27749d5b220384ce2d9add085e77a947be2b532c555ce0f5a92b9266cc262580ce7d56c528792d5a9
08b8a503b4188e3e77f3230cb53fc5e1cbba9c79fb1"
    }
  }
}

```

## C.2.2d Ustrukturyzowana postać przykładowego paragonu fiskalnego anulowanego:

```

{
  "paragAnul": {
    "JPKID": 6,
    "pamiecChr": 1,
    "nrDok": 5,
    "pozycja": [
      {
        "towar": {
          "brutto": 1000000,
          "cena": 1000000,
          "idStPTU": "G",
          "ilosc": "1",
          "nazwa": "Nazwa towaru",
          "oper": false
        }
      }
    ],
    "stPTU": [
      {
        "id": "G",
        "wart": "ZW"
      }
    ],
    "podsum": {
      "sumaNetto": [
        {
          "idStPTU": "G",
          "brutto": 1000000,
          "vat": 0
        }
      ],
      "sumaPod": 1000000,
      "sumaBrutto": 1000000,
      "waluta": "PLN"
    },
    "total": {
      "zaplZwrot": 1000000
    },
    "nrParag": 3,
    "nrKasy": "001",
    "zakSprzed": "2020-04-10T06:23:45.678Z",
    "kasjer": "Kasjer",
    "podpis": {
      "RSA":
"9e61a538a24334c0af624987d5967c11d89286f02bcd1c9123ff284c460e3dcf2ae7cca47c1a5079010b201d2a6ac
e3c1d619377c57501a3773e7faaa3ac3c2ea7eb26409b3ff852d715cf1c16cc1b490e7876bc23af51720174c729a2c
8aea7d4a58cf3d99b54582698764e58922af469d011cbc5100bb57bbbcf3f373bf629a33257eec641662c3f44cbc9d
39c7d7f885alb8fcd6da954e918077f7ff3279e2f84ea9aa944ab9123dd10f711870a8d1d6856ecaa6a6df0a7bce17
03fd79376bec82cec37ea4ba48a64063ebcdf4d99f9e18c37d0777bd4437ca819d19b2a8b2f73cff076cb240ee0580
d90eaba0c7aad3dff10af58e64c03e51453caa1c8f3",
      "SHA": "eee0e3068482a34527630f4c73c05d0d3523899f5fa25d022b9415ff8057d3be",
      "JPK": "00000000000000000000"
    }
  }
}

```

## C.2.2e Ustrukturyzowana postać przykładowego paragonu fiskalnego anulowanego:

```

{

```

```

"paragAnul": {
  "JPKID": 9,
  "pamiecChr": 1,
  "nrDok": 8,
  "pozycja": [
    {
      "towar": {
        "brutto": 10000000,
        "cena": 10000000,
        "idStPTU": "G",
        "ilosc": "1",
        "nazwa": "Nazwa towaru",
        "oper": false
      }
    }
  ],
  "stPTU": [
    {
      "id": "G",
      "wart": "ZW"
    }
  ],
  "podsum": {
    "sumaNetto": [
      {
        "idStPTU": "G",
        "brutto": 10000000,
        "vat": 0
      }
    ],
    "sumaPod": 10000000,
    "sumaBrutto": 10000000,
    "waluta": "PLN"
  },
  "total": {
    "zaplZwrot": 10000000
  },
  "nrParag": 6,
  "nrKasy": "001",
  "zakSprzed": "2020-04-10T09:23:45.678Z",
  "kasjer": "Kasjer",
  "podpis": {
    "RSA":
"ab6c66f6f3cbc34dde1e8f3b3080bee64555ab5f3886bae21f4be9139a11f6c09bc8765f229bfd99bece7fb7dbb5c
f18ffc17e7cfbd56bc8b637d4a821e483d8b6c5cc947e356a543b5eba1020a499fe7506d49c5d3fbd61dbd4281247
83fe0ee6532f57b7600d8266435eda9e5fead47c4d31b18961fe21ad824ccc6b20d51961a6221bcef4554322f3127c
9e488aaf6a54d0498096484840668372b5cf6b22a73a26334249a74fb4c87568dbde4693dfeed4ec44f473dcf4921c
3da287f3e8b1085521a833b416c7a269fa6d03e1d23ea1fde9577c78ae60179b4303629674f1fdfbacaed55706a7b4
66957b93c083d1e3f4af231f503cdb694b9afb0ebf3",
    "SHA": "a64a913986f2a18d4db5ecbe02309fc96d19c683830622140a2d4310425f911c",
    "JPK": "0010000000000000006"
  }
}
}

```

## C.2.2f Ustrukturyzowana postać przykładowego paragonu fiskalnego anulowanego:

```

{
  "paragAnul": {
    "JPKID": 10,
    "pamiecChr": 1,
    "nrDok": 9,
    "pozycja": [
      {
        "towar": {
          "brutto": 100000000,
          "cena": 100000000,
          "idStPTU": "G",
          "ilosc": "1",
          "nazwa": "Nazwa towaru",

```

```

        "oper": false
    }
},
"stPTU": [
    {
        "id": "G",
        "wart": "ZW"
    }
],
"podsum": {
    "sumaNetto": [
        {
            "idStPTU": "G",
            "brutto": 100000000,
            "vat": 0
        }
    ],
    "sumaPod": 100000000,
    "sumaBrutto": 100000000,
    "waluta": "PLN"
},
"total": {
    "zaplZwrot": 100000000
},
"nrParag": 7,
"nrKasy": "001",
"zakSprzed": "2020-04-10T10:23:45.678Z",
"kasjer": "Kasjer",
"podpis": {
    "RSA":
"54ce7bad1079bc8a7a69908766aa6e420edf611442f28f28cfbd51c3c181ad54505ff81465ab6dbbd93fcaf0fb78b
83ef56304e83fb302bf0e07c2114b7a947ad20c012d204efe1cd53aaff0469830aabe01a8a5828c798872bd6dbb7d8
767ab832d892a40f81c07d7246e34f8716d5cd0fd206da3bbb89aae831010f3e2278a6ce76c9206b022e6a246a0b88
9cbe571c062bf214a3dd924168f4913bde545dba76f42875bb1c7060fd06ded94f9c9ab7e9f2d04f276b9ccfaebdbd
ac48cc92b5b4f629504daca05d887a1cd8e035e50d7b07af374609f3bd6080e4bed80330d62924987d440f0b06f7e9
9a3ec980df0e1951e073854e20173a937ec2f1ca93f",
    "SHA": "ac6450833bef0f47888dea1272d179e8a197c8b6968c16782eeaff4f2769430d",
    "JPK": "001000000000000009"
}
}
}

```

### C.2.3 Przykłady danych do podpisu dokumentów przesyłanych w zbiorach danych

#### C.2.3a Pełna postać przykładowych danych do podpisu raportu fiskalnego fiskalizacji:

6970000802WTE20010000092020-04-10T01:23:45.678Z9DD845A0C9C285DC2E13F3253352E1836DC603C0KW123

#### C.2.3b Pełna postać przykładowych danych do podpisu raportu fiskalnego dobowego:

6970000802WTE20010000094906050602020-04-10T23:23:45.678Z

#### C.2.3c Pełna postać przykładowych danych do podpisu dokumentu нефiskalnego:

6970000802WTE200100000922020-04-10T03:23:45.678Z

#### C.2.3d Pełna postać przykładowych danych do podpisu paragonu anulowanego:

6970000802WTE2001000009510000002020-04-10T06:23:45.678Z

## C.2.4 Przykładowe podpisy danych dokumentów przesyłanych w zbiorach danych

### C.2.4a Pełna postać podpisu danych raportu fiskalnego fiskalizacji:

3019505879124143c92b60d5e57ebc08ba8fb7c9d2ce1bae02ba6b77a9c14b7794f7f8acbddff2c2e382fbb6820cd0e0a87fa988f2c67f9b34ae0571cfc0b23b727b7912c9d30b567c56e710cbfa88f09937bbfd1c54065df893eafb7455823716c292d548dc98dfc4ad29c81332450b84bea1a4bbf11e0c4dcc8d21f835476b87ff5a68be7d514fe15d3ebb76c2559bfe93a8cdafd09bf9bab7b5df918225c2f46778f59fb9deef898eb4e932f405a1403d9e0e78772c25675f14edc34d457f42aeb0a719a7a9638654bae7823641b1f95981b37fbd2cff3fda256e2e8f2e5aa8aca729e2d9a9b32893d6664b223da5c1879afceec64fea8b11fe22a622de55bb

### C.2.4b Pełna postać podpisu danych raportu fiskalnego dobowego:

21184e6e94b7c822223c90b3769567188c285c262d10be052c6820242368a5f655af1c77138930fe51873f2e2eea6c6a7666f3f8cf9f1776dad23ce32afb687f6b5057bdf38ce3bb732ff27d95b1682c541b43410af8cad7262bc93dc7466b09588311242b4c297251ae54afd8911607a482854a8bb326b68ae7c8d1295266488ee635a483e90157f8c943a3add99329edbc2ca18d9954e976f32f4bda9d74eb99527f411436c8857d4b8e804e72b0c0e6a58e47a6975749b8421874f095041cc5429d9d18b3f446f0c834964e2150c2f56b6de9e72f62cfa5f194aa5650a55b64f5f903d52b9435e84183860810835e5a266aef281b92ab8a257d9467f4c1ce

### C.2.4c Pełna postać podpisu danych dokumentu niefiskalnego:

282f2b77102bbb60717093d6ad9c2db648201be698baa316c03e7598446fc7915042e9d4a19d5a846ad5998c7467ecedd1f545dc6619d6c0a17e55f9380d01e584df8a84971bc9e91f922e6f2f4fe243c1382b8fa067809451241992622faa9e7b83feb4a3d6d8f742278c1e5174570a63b113495384a346ad6f1e0f5072d91c9227ea68abbbcb9a902f0463a126fbb43ef1380dac8421cd8d45cb797ab75a7c3b08988709e64bc174d9aeead5e9ea1a44d01e0c3e0a7ecb97baec560f34f4a6ce34d27749d5b220384ce2d9add085e77a947be2b532c555ce0f5a92b9266cc262580ce7d56c528792d5a908b8a503b4188e3e77f3230cb53fc5e1cbb9c79fb1

### C.2.4d Pełna postać podpisu danych paragonu anulowanego (pierwszego):

9e61a538a24334c0af624987d5967c11d89286f02bcd1c9123ff284c460e3dcf2ae7cca47c1a5079010b201d2a6ace3c1d619377c57501a3773e7faaa3ac3c2ea7eb26409b3ff852d715cf1c16cc1b490e7876bc23af51720174c729a2c8aea7d4a58cf3d99b54582698764e58922af469d011cbc5100bb57bbbcf3f373bf629a33257eec641662c3f44cbc9d39c7d7f885a1b8fcd6da954e918077f7ff3279e2f84ea9aa944ab9123dd10f711870a8d1d6856ecaa6a6df0a7bce1703fd79376bec82cec37ea4ba48a64063ebcdf4d99f9e18c37d0777bd4437ca819d19b2a8b2f73cff076cb240ee0580d90eaba0c7aad3dff10af58e64c03e51453caal8f3

### C.2.4e Pełna postać podpisu danych paragonu anulowanego (drugiego):

ab6c66f6f3cbc34dde1e8f3b3080bee64555ab5f3886bae21f4be9139a11f6c09bc8765f229bfd99bece7fb7dbb5cf18fffc17e7cfbd56bc8b637d4a821e483d8b6c5cc947e356a543b5eba1020a499fe7506d49c5d3fbd6b1dbd428124783fe0ee6532f57b7600d8266435eda9e5fead47c4d31b18961fe21ad824ccc6b20d51961a6221bcef4554322f3127c9e488aaf6a54d0498096484840668372b5cf6b22a73a26334249a74fb4c87568dbde4693dfeed4ec44f473dcf4921c3da287f3e8b1085521a833b416c7a269fa6d03e1d23ea1fde9577c78ae60179b4303629674f1fdfbacaed55706a7b466957b93c083d1e3f4af231f503cdb694b9afb0ebf3

### C.2.4f Pełna postać podpisu danych paragonu anulowanego (trzeciego):

54ce7bad1079bc8a7a69908766aa6e420edf611442f28f28cfbd51c3c181ad54505ff81465ab6dbbd93fcaf0fb78b83ef56304e83fb302bf0e07c2114b7a947ad20c012d204efe1cd53aaff0469830aabe01a8a5828c798872bd6dbb7d8767ab832d892a40f81c07d7246e34f8716d5cd0fd206da3bbb89aae831010f3e2278a6ce76c9206b022e6a246a0b889cbe571c062bf214a3dd924168f4913bde545dba76f42875bb1c7060fd06ded94f9c9ab7e9f2d04f276b9ccfaebdbdac48cc92b5b4f629504daca05d887a1cd8e035e50d7b07af374609f3bd6080e4bed80330d62924987d440f0b06f7e99a3ec980df0e1951e073854e20173a937ec2f1ca93f

## C.2.5 Dane wejściowe służące do wyliczenia skrótu SHA2

### C.2.5a Pełna postać danych wejściowych służących do wyliczenia skrótu SHA2 pierwszego paragonu anulowanego:

```
9e61a538a24334c0af624987d5967c11d89286f02bcd1c9123ff284c460e3dcf2ae7cca47c1a5079010b201d2a6ace
3c1d619377c57501a3773e7faaa3ac3c2ea7eb26409b3ff852d715cf1c16cc1b490e7876bc23af51720174c729a2c8
aea7d4a58cf3d99b54582698764e58922af469d011cbc5100bb57bbbcf3f373bf629a33257eec641662c3f44cbc9d3
9c7d7f885a1b8fcd6da954e918077f7ff3279e2f84ea9aa944ab9123dd10f711870a8d1d6856ecaa6a6df0a7bce170
3fd79376bec82cec37ea4ba48a64063ebcdf4d99f9e18c37d0777bd4437ca819d19b2a8b2f73cff076cb240ee0580d
90eaba0c7aad3dff10af58e64c03e51453caa1c8f3
```

### C.2.5b Pełna postać danych wejściowych służących do wyliczenia skrótu SHA2 drugiego paragonu anulowanego:

```
eee0e3068482a34527630f4c73c05d0d3523899f5fa25d022b9415fff8057d3beab6c66f6f3cbc34dde1e8f3b3080be
e64555ab5f3886bae21f4be9139a11f6c09bc8765f229bfd99bece7fb7dbb5cf18ffc17e7cfbd56bc8b637d4a821e4
83d8b6c5cc947e356a543b5eba1020a499fe7506d49c5d3fbbdb61dbd428124783fe0ee6532f57b7600d8266435eda9
e5fead47c4d31b18961fe21ad824ccc6b20d51961a6221bcef4554322f3127c9e488aaf6a54d049809648484066837
2b5cf6b22a73a26334249a74fb4c87568dbde4693dfeed4ec44f473dcf4921c3da287f3e8b1085521a833b416c7a26
9fa6d03e1d23ea1fde9577c78ae60179b4303629674f1fdfbacaed55706a7b466957b93c083d1e3f4af231f503cddb6
94b9afb0ebf3
```

### C.2.5c Pełna postać danych wejściowych służących do wyliczenia skrótu SHA2 trzeciego paragonu anulowanego:

```
a64a913986f2a18d4db5ecbe02309fc96d19c683830622140a2d4310425f911c54ce7bad1079bc8a7a69908766aa6e
420edf611442f28f28cfbd51c3c181ad54505ff81465ab6dbbd93fcaf0fb78b83ef56304e83fb302bf0e07c2114b7a
947ad20c012d204efe1cd53aaff0469830aabe01a8a5828c798872bd6dbb7d8767ab832d892a40f81c07d7246e34f8
716d5cd0fd206da3bbb89aae831010f3e2278a6ce76c9206b022e6a246a0b889cbe571c062bf214a3dd924168f4913
bde545dba76f42875bb1c7060fd06ded94f9c9ab7e9f2d04f276b9ccfaebdbdac48cc92b5b4f629504daca05d887a1
cd8e035e50d7b07af374609f3bd6080e4bed80330d62924987d440f0b06f7e99a3ec980df0e1951e073854e20173a9
37ec2f1ca93f
```

```
{ "eParagon.mf.gov.pl": "eyJ3ZXJzameIoiJKUetfs0FTQV9QQvJBRO9OX3YxLTAiLCJKUetJRci6IjAwMTAwMDAwMDAwMDAwMDAwNyIsImRhdGFKUESioiIyMDIwLTA0LTEwVDA3OjIzOjQ1LjY3OFoicLKJKUetSRUYiOnsiU0hBMjU2IjoineiNzc2OUmZODlCQjMzQzdGMzRCMDZEMUNCnKmwREEXm0Y5QUqZMDU4NUJGRtQ3QjdEQjaA4NUeyQTITODkwRTewMSIs1kpQS01EIjYiMDAxMDAwMDAwMDAwMDAwMDAIIn19", "alg": "RS256", "jpkmetadataata": "eyJjb2lwcmVzc2lvbi6IKFRKRkksBVEUIfq==", "x5c": [ "MIIDKjCCApOgAwIBAQIQT7uoIIPDHkdAJLSAHd4+zaNBgkhgi9EwCBQAUFADUMRIEAYDVQDEwNlRiBlLUthc3kwHhcNMjAwMzE2MTAzNTU5WhcnMjUwMzE2MTAzNTU4WjBrJjEjMCEGA1UEBxCsMarGVwYXN0YW1lbncQGSW5mb3JtYXR5emFjamkxHjAcBgNVBAoTFUlpbm1zdGVyc3R3byBGaw5hbnnvdzEZMBcGA1UEBRMQvkFUUEwtNjk3MDAwMDgwMjEWMBQGA1UEAAmXNVlRFmjAwMTAwMDAwOTERMA8GA1UEBXMIV2Fyc3phd2ExFDASBgNVBAGTC01hem93aWVja2llMQswCQYDVQVQKGewJQTDCCA1VDQYJKoZIhvcNAQEUBBQAdggEPADCCAQOCgGEBA1LTSh7ASot4qt863BAwb2TDdvGgEqEDNV\ /g0dtWK81C5vK677m977U0PccRGJ4ht7B+kmRvfS092YgnjmPQeqAJJ4GYcICGXai+6f6xeUQknwAvhpvmPgVzjlAmHySLnKgQXQgrxV\ /WE64jJgenducsuGa6AZHRcw7rxGMAxGa2YSpEpNZBq8qhpl17saJJ0PHZCJwU5PP1nsNzdSOEJndJYHnbbrcom6ZFtbju2QMk\ /gwpFACgyV9hoHiOo7BCSiZ3fPC4tpmhxOPqmdZDYwInsUOMgaYP3p803z2f16lgkg2Tldt4GroubKiffw4RumFwuIP2Ze8DKD834PQ8CAwEAANeMFwwEwYDVR01BAwwCgYIKwYBBQUHAwIwRQYDVROBBD4wPIAQxg+n+j1JAu1f23nuS65XPaeWBmqxEjAQBGBNVBAMTJCIGUTS2FzeYIqs8LVez9uP51AlnvXe8bYeAnJBgkqhkiG9w0BAQsFAAOBgQBZfk\ /KxEy\ /3zIFP0r4nt5dg5LbJ737krFIzRh3PFhmThaac2kTLtaamXCw1L70nm8t\ /J9krD9uyk83vLmLbPGVEbcOlY+50o5XYetubSdLFnlm4kiFUDh5NP4AKzQdTpc4jbaOWRFJCbrZzbv6MOxytAE\ /mlrTmlr9K313MxRw==" ] }
```

```
{ "eParagon.mf.gov.pl": "eyJ3ZXZjamEioiJKUetfs0FTQV9QQVJBR09OX3YxLTaiLCJKUETJRCI6IjAwMTAwMDAwMDAwMDAwMDAwOCIsImRhdkGFKUESIoIyMDIwLTA0LTEwVDAA0jIzOqJ1LjY3OFoiLCJKUETSUYiOnsiU0hBMjU2IjoIQkGJNDNEQThGMUJCmZFBRTZGMzEO0TkwNtCzmKMIMEFEN0E4OEJDnKEYQ0UyQTkeznJBcNDVDQkMxQRDRjBGMSSisIkpQS0lEtJoIMDAwMDAwMDAwMDAwMDAwMDA3In1r", "alg": "RS256", "x5c": ["MIIDKjCCApOGawIBAgIT27uolPdhKAJJLSAHd4+ZANBgqhkiG9w0BAQsFAADARIEAYDVQDEwlNRilLUtheC3kwhHemAjWmeZEMTAZNTU5WhcnUmjUwMeZMTAZNTU4WjCBrjEjMCEGA1UECzMARGVwYXJ0YW1lbGQs5mb3JtYXR5emFnamxhXjAcBgNVBAAwTFU1pbml2dGVycCw3Z2byBGAw5hbGlnvndEZMBcGA1UEBRMQVkFUUEwtNjk3MDAwMDgwMjEWMBCGGA1UEAxMNv1RFmjAwMTAwMDAwOTERMA8GA1UEBXMIV2Fyc3phd2ExFDASBgNVBAGTC01hem93aWVya21lMQswCQYDVQQGEWJTDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBALtSH7AWsOT4qt863BAwbT2DvgGEGDNV\g0dtWK81C5vk677mb9Y7U0PccRJG4ht7b+kmRvfcsO92YgnjpmQEAgJ4GyCicGXaI+6f6xeUQknvwAPhvmpGvjSlsAmHyScLdGQXozGqrXV\WE64jJgendcucsG6AZHRchrxGMAXPC2ysGpNZBq8hpl7sAjjoPHZCWJPu5PPlnsNdzdSOBJndJYHNbbromf2FLbjju2QMk\g/wPfACgvY9hoHiOo7BCSI23fPC4tpmhxoPmqdzDywlnSU0MgaYP3p803z2f16glkg2Tldt4GroubKiFFw4RumFuWiPZZZe8DKD834PQ8CAwEEAAaNefFwwEWYDVR01BAwwCGYIKwYBBQUHAwIEWRQYDVR0BBDB4wPIAQxg+n+jJAU1f23nuS65XPaeEWWBQxEJAQBGNVBAMTCU1IGIGUTS2FezEIYS8LVez9uP51AlnvXe8bYeJANBgqhkiG9w0BAQsFAAOBGQBZfk\KxEy\3ziFP0r4nt5Dg5LbJ737KRfiZRhu3PHmMthAac2KTltaaaWC+iL70nm8t\J9KrD89yk83vLwmLBPGVEbcOIY+5oo5XYetubSdLfNLm4kiFUDh5NF4AKzQdTpcp4jbAOWRFJCbrZzbv6M0XytAE\m1rTrml9K313MrRx==" ] }
```

C.3.2a Pełna postać chronionego nagłówka paragonu fiskalnego (pierwszego) zakodowana w Base64URL:

C.3.2b Pełna postać chronionego nagłówka paragonu fiskalnego (drugiego) zakodowana w Base64URL:

eyJlUGFYyWdvbi5tZi5nb3YucGwiOiJleUozWlhKemFtRWlPaUPLVUV0ZlMwRlRRVjlRUVZKQlIWou9YM1l4TFRBaUxDSk  
tVRXRKUkNjNklqQXdNVEF3TURBd0lEQXdNREF3TURBd05TSXNJbVJoZEdGS1VFc2lPaU15TURJd0xUQTBMVEV3VkrBMU9q  
SXpPaLExTGPzZM09Gb2lMQ0pLVUV0U1JVWW1PbnNpVTBQk1qVTJJam9pUmpnNFJWWkZOVUZFT1RVMK1UazFNRGN5UmtNME  
l6YzVOVFJCUWpQoQlFUQTFSakJHTTBar05YzNsAlpCTURFJe9VTkmpNVVZHUmtNME16VXpRPUCTURJc0l1rCFFTMGXFSWpva  
uA1EQXhNREF3TURBd0lEQXdNREF3TURBMElWtKiLCJhbgGciOiJSUz2lNiIsImpwa2l1dGfkyXRhIjoziZl1KamIzSnlaE3  
hozEdsdmJrbGtJam9pVkvVarUxSfVSVSEl3TURFd0lEQXdNRGt1TWpBeU1DMHdOQzB3TTFReU1UbzFNRG93TUM0ek1qSmFJ  
aXdpWT15dGNISmxjM05wYjI0aU9pSkVSVVpNUVZSRklMD0iLCJ4NWMI0IsiTu1JREtqQ0NBcE9nQXdxJQkFnsVEYn3VvbE  
lwREhLZEFKakxTQhENc26QU5CZ2taGtpRmZl3MEJBUXNQUKURBVU1SSXJqFVQ1VlVFRREv3be5SaUsJTFV0aGMza3dIaGNO  
TWpBd0l6RTJNVEF6T1RVNVd0Y05NalV3TxpPzK1TQXpOVFU0V2pDqnJqRWpNQ0VHQTFVRUN4TWFSRJ23WVhKMLXMXyb1  
FnU1clbWJzSnRZFw1lZW1GamFa3htakFjQmdOVXkjbB1RGVTFwY1semRHNhVjM1ZyZl1CR2FNSWh1bk5Z2HpfFWK1c0DAB  
MVVFQlJNUVZrR1VVRXd0TmprM0lEQXdNRGd3TWpFV01CUudBMVVFQXhNT1YxUkZNakF3TVRBd0lEQXdPVEVSTUE4R0ExVU  
VCeElJVjJGGeWZcGhKMKV4RkRBU0JnTlZCQWdUQzAXaGvOTNhV1ZgYtJsbe1Rc3dDUVlEVlFRF0V3S1FURENDQVNJdORR  
WpULb1pJahZjTkJRRUJCUFUEZ2dFUEFEQ0NBW9Dn2dFQkFmDFNIN0F3U0UNHFOUDYzQkF3YnQYQhZJhUcVnRE5WXC9nMG  
R0V0s4MUM1dks2NdtYzJl1Zl1WUGNjUpKHNGh0N0Ira2lSdmZzTzkyWUduam1UUVXUpQpneQ5R2lR1hhsSS2zj3Z4ZVVr  
a253dnBBSHwzbVBhd1pqbHNBhU5U2NMZEdxT1h6UWdyeFZcl1dFNjRqSmdlbmRjdwNzR3U2QVpIUKNXN3J4R01BeZEdBMn



1TZ0VwTlpCUXE4aHBobDdzQWpqMFBiWkNKd1U1UFAxbnN0emRkU09FSm5kS1lITmJicm9tNlpgGTGJqdTJRTWtcL2dXcEZBQ2dZVjlob0hpT283QkNTaVozZ1BDNHRwbWh4T1BxbWRaRfL3SW5TVtBNZ2FZUDNwODAzeJjMmTZNmWtnMlRsZHQ0R3JvdWJLaWZmVzRSdW1Gd3VJUfPaZThES0Q4MzRQUthDQXdFQUFhTmVNRnd3RXdzRFZSMGxCQXd3Q2dZSUt3WUJCUVVlQXdJd1JRWURWUjBCQkQ0d1BJQVF4ZytUk2oxakFVMWYyM251UzY1WFBhRVdNQ1F4RWpBUUJnTlZCQU1UQ1UxR01HVXRTMkZ6ZV1JUXM4TFZlWjl1UDVsQWxudlhlOGJZZWpBtkJna3Foa2lHOXcwQkFRc0ZBQU9CZ1FCWmZrXC9LeEV5XC8zeklMUDByNE50NURnNUxiSjczN2tSZklaUmh1M1BtBU10aEFhYzJrVEx0YWFhV0MraUw3MG5tOHRcL0o5a1JEOVV5azgzdkxtTGJQR1ZFymNPbFkrNU9vNVhZZXR1Y1NkTEZuTG00a2lGVURoNU5QNEFle1FkVHBDcDRqYkFPV1JGSkNic1paYlY2TBTBYeXRBRVwvbWxyVHJtbd1LMzEzTxhSdz09I119

### C.3.2c Pełna postać chronionego nagłówka paragonu fiskalnego (trzeciego) zakodowana w Base64URL:

eyJlUGFyYWdvbi5tZi5nb3YucGwiOiJleUozWlhKemFtRWlPaUpLVUV0ZlMwRlRRVjlRUVZKQlIwOU9YM1l4TFRBaUxDsktVRXRKUKnJNk1qQXdNVEF3TURBd01EQXdNREF3TURBd055SXNJbVJoZEdGS1VFc2lPaU15TURJd0xUQTBMVEV3VkrBM09qSXpPalExTGpZM09Gb2lMQ0pLVUV0U1JVWw1PbnNpVTBoQk1qVTJJam9pTnpjMk9VTXpPRGxDUWpNelF6ZEdNelJDTURaRU1VTkNoa013UkVFeE0wWTVRVVF6TURVNE5VSksdSVFEzUWpkRVFqQTR0VUV5UVRJmK9Ea3dSVEV3TVNJc0lrcFFTMGxFSWpvaU1EQXhNREF3TURBd01EQXdNREF3TURBMUluMTkiLCJhbGciOiJSUzI1NiIsImpwa2l1dGFkYXRhIjoiZXlKamIyMXdjYVZ6YzJsdmJpSTZJa1JGUmt4QlZlFVWlMUT09IiwieDVjIjpbIk1JSURLakNDQXBPZ0F3SUJBZ01RMjd1b2xJcERIS2RSBmpMUFIrDQrekFOqmdrcWhraUc5dzBCQVZzRkFEQVNVUk13RUZFRFZRUURFd2x0UmlCbExVdGhjM2t3SGhjTklqQXdNekUyTVRBek5UVTVXaGNOTWpVd016RTJNVEF6T1lRVNfdqQ0JyYkVqTUNFR0ExVUVDeE1hUkdWd1lYsJbZVZsZm5RZlNXNW1iM0p0VWhSNWVtRmphaWt4SGpBY0JnTlZCQW9URlUxcGJtbHpkR1Z5YzNSM2J5QkdhVzVoYm50dmR6RVpNqMNHQTfVRUJUSTVFWa0ZVVUV3dE5qazNNREF3TURnd01qRVdNQ1FHQTFVRUF4TU5WMVJGTWpBd01UQXdNREF3T1lRFUk1BOEdBMVVFQnhNSVYyRnljM3BoZDJFeEZEQVNCZ05WQkFnVEMwMWhlbTtkzYVdWamEyYkVNUXN3Q1FZFRZRUUdFd0pRVERDQ0FTSXdEUVlKS29aSWWh2Y05BUUVCQ1FBRGdnRVBBRENDQVFvQ2dnRUJBTHTRTSDdBd1NPVDRxVDG2M0JBd2J0MkR2R3FFZ0ROVlwwZzBkdFdlODFBNXZLNjc3bW15WTdvMFBjY1JkRzRodDdCK2ttUnZmc085M1lnbmptcFFFcUFKajRHeUNpY0dYUkrNmY2eGVVUWtud3ZwQUh2cG1QR3ZaamxzQW1IeVnJTGhRcU9Ye1FncnhWXC9XRTY0akpnZW5kY3Vjc0d1NkFaSFJDVzdyeEdnQXhHQJTJ5U2dFcE5aQ1FxoGhwaGw3c0FgaJbQSFpDSndVNVBQMw5zTnpkZFNPRUpuZEpZSE5iYnJvbtZaRkxianUyU1rXC9nV3BGQUUNhWVY5aG9IaU9vN0JDU2laM2ZQZzR0cG1oeE9QcW1kWkRZd0luU1UwTWdhWVAzcdGwM3oyZjE2ZzFrZzJUbGR0NEdyb3ViS2lmZlcl0UnVtRnd1SVBaWmU4REtEODM0UFE4Q0F3RUFBYU51TUZ3d0V3WURWUjBsQkF3d0NnWU1Ld1lCQ1FVSEF3SXdsUVlEVlIwQkJENHdQSUFReGerb1tqMwPbVTfMmJNudVM2NVhQYUUVXUJREtEVqQVFCZ05WQkFNVENVMUdJR1V0UzJGEmVZSVFzOEExWZVo5dVA1bEFsbntZYZThiWVWqQU5CZ2txaGtpRz13MEJBUXNGQUFPQmdRQlpma1wvS3hFeVwvM3pJZlAwcjRodDVEZzVMYko3MzdrUmZJW1JodTNQSG1NdGhBYWMya1RMDGFhYVdDK2lMNzBubTh0XC9KOWtSRDLVwS4M3ZMbUxiUEdWRWJjT2xZKzVPbzVYWWV0dWJTZEExGbkxtNgtpRlVEaDVOUDRBS3prZFRwQ3A0amJBt1dSRkpdYnJaWmJWNk0wWHl0QUVcL2lsc1RybWw5SzMxM014Unc9PSJdfQ

### C.3.2d Pełna postać chronionego nagłówka paragonu fiskalnego (czwartego) zakodowana w Base64URL:

eyJlUGFyYWdvbi5tZi5nb3YucGwiOiJleUozWlhKemFtRWlPaUpLVUV0ZlMwRlRRVjlRUVZKQlIwOU9YM1l4TFRBaUxDsktVRXRKUKnJNk1qQXdNVEF3TURBd01EQXdNREF3TURBd09DSXNJbVJoZEdGS1VFc2lPaU15TURJd0xUQTBMVEV3VkrBNE9qSXpPalExTGpZM09Gb2lMQ0pLVUV0U1JVWw1PbnNpVTBoQk1qVTJJam9pUWtKR05ETkVRVGhHTVVKQ016RkJSVFpHTXpFME9Ua3d0VGN6TWtNMU1FRkVOMEU0T0VKRE5rRXlRMFV5UVRrek5qQkNORFZEUWtNeFEwUkRSakJHTVNJc0lrcFFTMGxFSWpvaU1EQXhNREF3TURBd01EQXdNREF3TURBM0luMTkiLCJhbGciOiJSUzI1NiIsIng1YyI6WyJNSU1ES2pdQ0FwT2dBd01CQWdJUTI3dW9sSXBESEtkQUppqTFNBSEQ0K3pBTkJna3Foa2lHOXcwQkFRc0ZBREFFVTVJJd0VBWURWUVFERXdsTlJpQmxMVXR0YzNrd0hoY05NakF3TXpFMk1UQXpOVFU1V2hjTklqVXdNekUyTVRBek5UVTRXakNCmpFak1DRUdBMVVFQ3hNYVJHVndZWEowWVcxGJuUWdTVzVtYjNkZFlYUjVlbUZZqYW1reEhqQWNCZ05WQkFvVEZVMXBibWx6ZEdWeWMzUjNieUJHYVclagJUTnZkeKvaTUJjR0ExVUVUCUk1RVmtGVVVFd3ROamszTURBd01EZ3dNakVXTUJRR0ExVUVBeE1OVjFSRk1qQXdNVEF3TURBd09URVJNQThHQTFVRUJ4TU1WMkZ5YzNwaGQyRXhGREFTQmdOVk1Jb2lRDMDFoZW05M2FXVmhmMmxsTVFzd0NRWURWUVFHRXdkUUVREQ0NBu013RFFZSktvWk1odmN0QVFFQkJRQURnZ0VQURDQ0FRb0NnZ0VCQUx0U0g3QXdTT1Q0cVQ4NjNCQXdiDjEJedkdxRWdETlZcl2cwZHRXSzgxQzV2SzY3N2liOVk3VTBQY2NSSkc0aHQ3QitrbVJ2ZnNpOTJZZ25qbXBRRXFBsm0R3lDaWNHwGFJKzZmNnh1VVFrbnd2cEFIdnBtUEd2Wmpsc0FtSH1TY0xkR3FPWHPzRZJ4V1wvV0U2NGpKZ2VuZGN1Y3NHdTBZWkhsSQ1c3cnhHTUF4R0EYeVnNRXBOWkJRcThocGhsN3NBamowUEhaQ0p3VTVQUdFuc056ZGRtT0VKbmRKWUhoYmJyb202WkZMYmp1M1FNalwvZ1dwRkFDZ1lWOWhvSG1PbzdCQ1NpWjNmUEM0dHBtaHhPUHftZfPpEWXdJb1NVME1nYVlQM3A4MDN6MmYxNmcxa2cyVGxkdDRHcm91YktpZmZlXNFJlbUz3dU1QWlp1OERLRDgzNFBROENBd0VBQWFOZU1Gd3dFd11EVlIwbEJBd3dDZ1lJS3dZQkJRvUhBd013U1FZRFZSMJEJCRR3UE1BUXhnK24rajFqQVUxZjIzbnVTNjVYUGFFV01CUXhFakFRQmdOVk1JBTVRDVTfHsUdVdFMyRnplWU1RczhMVMVaOXVQNWxBbG52WGU4Y1l1lakFOQmdrcWhraUc5dzBCQVZzRkFBT0JnUUAZmtcL0t4RX1cLzN6SWZQMH10TnQ1RGclTGJKNzM3a1JmSVpSaHUzUEhtTXRoQWFjMmtUTHRhYWFXYQytpTDcwbm04dFwvSj1rUkQ5VX1rODN2TG1MY1BHVKviY09sWSs1T281WF1ldHViu2RMRm5MbTRraUZVRGg1T1A0Qut6UWRUCeNwNgpiQU9XUkZKQ2JyW1piVjZNMfh5dEFFXC9tbhJUcm1sOUszMTNNeFJ3PT0iXX0



### C.3.3 Przykłady ustrukturyzowanych danych dokumentów w postaci elektronicznej:

#### C.3.3a Przykładowa postać ustrukturyzowanych danych paragonu fiskalnego (pierwszego):

```
{
  "dokument": {
    "naglowek": {
      "wersja": "JPK_KASA_PARAGON_v1-0",
      "dataJPK": "2020-04-10T04:23:45.678Z"
    },
    "podmiot1": {
      "nazwaPod": "Nazwa podmiotu",
      "nrFabr": "WTE2001000009",
      "NIP": "6970000802",
      "adresPod": {
        "ulica": "Ulica",
        "miejsc": "Miejscowosc",
        "nrLok": "NrLok",
        "poczta": "Poczta",
        "nrDomu": "NrDomu",
        "kodPoczt": "00-000"
      },
      "nrUnik": "WTE2001000009",
      "nrEwid": "2020/000001612"
    },
    "paragon": {
      "JPKID": 4,
      "pamiecChr": 1,
      "nrDok": 3,
      "pozycja": [
        {
          "towar": {
            "brutto": 1230,
            "cena": 1230,
            "idStPTU": "A",
            "ilosc": "1",
            "nazwa": "Nazwa towaru 1",
            "oper": false
          }
        },
        {
          "towar": {
            "brutto": 1000,
            "cena": 1000,
            "idStPTU": "G",
            "ilosc": "1",
            "nazwa": "Nazwa towaru 2",
            "oper": false
          }
        }
      ],
      "stPTU": [
        {
          "id": "A",
          "wart": 2300
        },
        {
          "id": "G",
          "wart": "ZW"
        }
      ],
      "podsum": {
        "sumaNetto": [
          {
            "idStPTU": "A",
            "brutto": 1230,
            "vat": 230
          },
          {

```

```

        "idStPTU": "G",
        "brutto": 1000,
        "vat": 0
    },
    ],
    "sumaPod": 230,
    "sumaBrutto": 2230,
    "waluta": "PLN"
},
"total": {
    "zaplZwrot": 2230
},
"nrParag": 1,
"nrKasy": "001",
"zakSprzed": "2020-04-10T04:23:45.678Z",
"kasjer": "Kasjer"
}
}
}

```

### C.3.3b Przykładowa postać ustrukturyzowanych danych paragonu fiskalnego (drugiego):

```

{
  "dokument": {
    "naglowek": {
      "wersja": "JPK_KASA_PARAGON_v1-0",
      "dataJPK": "2020-04-10T05:23:45.678Z"
    },
    "podmiot1": {
      "nazwaPod": "Nazwa podmiotu",
      "nrFabr": "WTE2001000009",
      "NIP": "6970000802",
      "adresPod": {
        "ulica": "Ulica",
        "miejsc": "Miejscowosc",
        "nrLok": "NrLok",
        "poczta": "Poczta",
        "nrDomu": "NrDomu",
        "kodPoczt": "00-000"
      },
      "nrUnik": "WTE2001000009",
      "nrEwid": "2020/000001612"
    },
    "paragon": {
      "JPKID": 5,
      "pamiecChr": 1,
      "nrDok": 4,
      "pozycja": [
        {
          "towar": {
            "brutto": 1230,
            "cena": 1230,
            "idStPTU": "A",
            "ilosc": "1",
            "nazwa": "Nazwa towaru 1",
            "oper": false
          }
        },
        {
          "towar": {
            "brutto": 1000,
            "cena": 1000,
            "idStPTU": "G",
            "ilosc": "1",
            "nazwa": "Nazwa towaru 2",
            "oper": false
          }
        }
      ]
    },
    "stPTU": [

```

```

        {
            "id": "A",
            "wart": 2300
        },
        {
            "id": "G",
            "wart": "ZW"
        }
    ],
    "podsum": {
        "sumaNetto": [
            {
                "idStPTU": "A",
                "brutto": 1230,
                "vat": 230
            },
            {
                "idStPTU": "G",
                "brutto": 1000,
                "vat": 0
            }
        ],
        "sumaPod": 230,
        "sumaBrutto": 2230,
        "waluta": "PLN"
    },
    "total": {
        "zaplwrot": 2230
    },
    "nrParag": 2,
    "nrKasy": "001",
    "zakSprzed": "2020-04-10T05:23:45.678Z",
    "kasjer": "Kasjer"
}
}
}

```

### C.3.3c Przykładowa postać ustrukturyzowanych danych paragonu fiskalnego (trzeciego):

```

{
    "dokument": {
        "naglowek": {
            "wersja": "JPK_KASA_PARAGON_v1-0",
            "dataJPK": "2020-04-10T07:23:45.678Z"
        },
        "podmiot1": {
            "nazwaPod": "Nazwa podmiotu",
            "nrFabr": "WTE2001000009",
            "NIP": "6970000802",
            "adresPod": {
                "ulica": "Ulica",
                "miejsc": "Miejscowosc",
                "nrLok": "NrLok",
                "poczta": "Poczta",
                "nrDomu": "NrDomu",
                "kodPoczt": "00-000"
            },
            "nrUnik": "WTE2001000009",
            "nrEwid": "2020/000001612"
        },
        "paragon": {
            "JPKID": 7,
            "pamiecChr": 1,
            "nrDok": 6,
            "pozycja": [
                {
                    "towar": {
                        "brutto": 12300,
                        "cena": 12300,
                        "idStPTU": "A",

```

```

        "ilosc": "1",
        "nazwa": "Nazwa towaru 1",
        "oper": false
    }
},
{
    "towar": {
        "brutto": 10000,
        "cena": 10000,
        "idStPTU": "G",
        "ilosc": "1",
        "nazwa": "Nazwa towaru 2",
        "oper": false
    }
}
],
"stPTU": [
    {
        "id": "A",
        "wart": 2300
    },
    {
        "id": "G",
        "wart": "ZW"
    }
]
},
"podsum": {
    "sumaNetto": [
        {
            "idStPTU": "A",
            "brutto": 12300,
            "vat": 2300
        },
        {
            "idStPTU": "G",
            "brutto": 10000,
            "vat": 0
        }
    ],
    "sumaPod": 2300,
    "sumaBrutto": 22300,
    "waluta": "PLN"
},
"total": {
    "zaplZwrot": 22300
},
"nrParag": 4,
"nrKasy": "001",
"zakSprzed": "2020-04-10T07:23:45.678Z",
"kasjer": "Kasjer"
}
}
}

```

### C.3.3d Przykładowa postać ustrukturyzowanych danych paragonu fiskalnego (czwartego):

```

{
    "dokument": {
        "naglowek": {
            "wersja": "JPK KASA PARAGON v1-0",
            "dataJPK": "2020-04-10T08:23:45.678Z"
        },
        "podmiot1": {
            "nazwaPod": "Nazwa podmiotu",
            "nrFabr": "WTE2001000009",
            "NIP": "6970000802",
            "adresPod": {
                "ulica": "Ulica",
                "miejsc": "Miejscowosc",
                "nrLok": "NrLok",
            }
        }
    }
}

```

```

        "poczta": "Poczta",
        "nrDomu": "NrDomu",
        "kodPoczt": "00-000"
    },
    "nrUnik": "WTE2001000009",
    "nrEwid": "2020/000001612"
},
"paragon": {
    "JPKID": 8,
    "pamiecChr": 1,
    "nrDok": 7,
    "pozycja": [
        {
            "towar": {
                "brutto": 12300,
                "cena": 12300,
                "idStPTU": "A",
                "ilosc": "1",
                "nazwa": "Nazwa towaru 1",
                "oper": false
            }
        },
        {
            "towar": {
                "brutto": 10000,
                "cena": 10000,
                "idStPTU": "G",
                "ilosc": "1",
                "nazwa": "Nazwa towaru 2",
                "oper": false
            }
        }
    ],
    "stPTU": [
        {
            "id": "A",
            "wart": 2300
        },
        {
            "id": "G",
            "wart": "ZW"
        }
    ],
    "podsum": {
        "sumaNetto": [
            {
                "idStPTU": "A",
                "brutto": 12300,
                "vat": 2300
            },
            {
                "idStPTU": "G",
                "brutto": 10000,
                "vat": 0
            }
        ],
        "sumaPod": 2300,
        "sumaBrutto": 22300,
        "waluta": "PLN"
    },
    "total": {
        "zaplZwrot": 22300
    },
    "nrParag": 5,
    "nrKasy": "001",
    "zakSprzed": "2020-04-10T08:23:45.678Z",
    "kasjer": "Kasjer"
}
}

```

### C.3.4 Przykłady nieskompresowanych danych dokumentów w postaci elektronicznej:

#### C.3.4a Przykładowa postać nieskompresowanych danych paragonu fiskalnego (pierwszego):

```
{ "dokument": { "naglowek": { "wersja": "JPK_KASA_PARAGON_v1-0", "dataJPK": "2020-04-10T04:23:45.678Z" }, "paragon": { "pamiecChr": 1, "total": { "zaplwrot": 2230 }, "zakSprzed": "2020-04-10T04:23:45.678Z", "JPKID": 4, "nrParag": 1, "kasjer": "Kasjer", "stPTU": [ { "id": "A", "wart": 2300 }, { "id": "G", "wart": "ZW" } ], "podsum": { "waluta": "PLN", "sumaNetto": [ { "brutto": 1230, "vat": 230, "idStPTU": "A" }, { "brutto": 1000, "vat": 0, "idStPTU": "G" } ], "sumaBrutto": 2230, "sumaPod": 230 }, "nrDok": 3, "pozycja": [ { "towar": { "brutto": 1230, "ilosc": "1", "oper": false, "cena": 1230, "nazwa": "Nazwa towaru 1", "idStPTU": "A" }, { "towar": { "brutto": 1000, "ilosc": "1", "oper": false, "cena": 1000, "nazwa": "Nazwa towaru 2", "idStPTU": "G" } } ], "nrKasy": "001", "podmiot1": { "nazwaPod": "Nazwa podmiotu", "nrFabr": "WTE2001000009", "NIP": "6970000802", "adresPod": { "ulica": "Ulica", "miejsc": "Miejscowosc", "nrLok": "NrLok", "poczta": "Poczta", "nrDomu": "NrDomu", "kodPoczt": "00-000"}, "nrUnik": "WTE2001000009", "nrEwid": "2020/000001612" } } }
```

#### C.3.4b Przykładowa postać nieskompresowanych danych paragonu fiskalnego (drugiego):

```
{ "dokument": { "naglowek": { "wersja": "JPK_KASA_PARAGON_v1-0", "dataJPK": "2020-04-10T05:23:45.678Z" }, "paragon": { "pamiecChr": 1, "total": { "zaplwrot": 2230 }, "zakSprzed": "2020-04-10T05:23:45.678Z", "JPKID": 5, "nrParag": 2, "kasjer": "Kasjer", "stPTU": [ { "id": "A", "wart": 2300 }, { "id": "G", "wart": "ZW" } ], "podsum": { "waluta": "PLN", "sumaNetto": [ { "brutto": 1230, "vat": 230, "idStPTU": "A" }, { "brutto": 1000, "vat": 0, "idStPTU": "G" } ], "sumaBrutto": 2230, "sumaPod": 230 }, "nrDok": 4, "pozycja": [ { "towar": { "brutto": 1230, "ilosc": "1", "oper": false, "cena": 1230, "nazwa": "Nazwa towaru 1", "idStPTU": "A" }, { "towar": { "brutto": 1000, "ilosc": "1", "oper": false, "cena": 1000, "nazwa": "Nazwa towaru 2", "idStPTU": "G" } } ], "nrKasy": "001", "podmiot1": { "nazwaPod": "Nazwa podmiotu", "nrFabr": "WTE2001000009", "NIP": "6970000802", "adresPod": { "ulica": "Ulica", "miejsc": "Miejscowosc", "nrLok": "NrLok", "poczta": "Poczta", "nrDomu": "NrDomu", "kodPoczt": "00-000"}, "nrUnik": "WTE2001000009", "nrEwid": "2020/000001612" } } }
```

#### C.3.4c Przykładowa postać nieskompresowanych danych paragonu fiskalnego (trzeciego):

```
{ "dokument": { "naglowek": { "wersja": "JPK_KASA_PARAGON_v1-0", "dataJPK": "2020-04-10T07:23:45.678Z" }, "paragon": { "pamiecChr": 1, "total": { "zaplwrot": 22300 }, "zakSprzed": "2020-04-10T07:23:45.678Z", "JPKID": 7, "nrParag": 4, "kasjer": "Kasjer", "stPTU": [ { "id": "A", "wart": 2300 }, { "id": "G", "wart": "ZW" } ], "podsum": { "waluta": "PLN", "sumaNetto": [ { "brutto": 12300, "vat": 2300, "idStPTU": "A" }, { "brutto": 10000, "vat": 0, "idStPTU": "G" } ], "sumaBrutto": 22300, "sumaPod": 2300 }, "nrDok": 6, "pozycja": [ { "towar": { "brutto": 12300, "ilosc": "1", "oper": false, "cena": 12300, "nazwa": "Nazwa towaru 1", "idStPTU": "A" }, { "towar": { "brutto": 10000, "ilosc": "1", "oper": false, "cena": 10000, "nazwa": "Nazwa towaru 2", "idStPTU": "G" } } ], "nrKasy": "001", "podmiot1": { "nazwaPod": "Nazwa podmiotu", "nrFabr": "WTE2001000009", "NIP": "6970000802", "adresPod": { "ulica": "Ulica", "miejsc": "Miejscowosc", "nrLok": "NrLok", "poczta": "Poczta", "nrDomu": "NrDomu", "kodPoczt": "00-000"}, "nrUnik": "WTE2001000009", "nrEwid": "2020/000001612" } } }
```

#### C.3.4d Przykładowa postać nieskompresowanych danych paragonu fiskalnego (czwartego):

```
{ "dokument": { "naglowek": { "wersja": "JPK_KASA_PARAGON_v1-0", "dataJPK": "2020-04-10T08:23:45.678Z" }, "paragon": { "pamiecChr": 1, "total": { "zaplwrot": 22300 }, "zakSprzed": "2020-04-10T08:23:45.678Z", "JPKID": 8, "nrParag": 5, "kasjer": "Kasjer", "stPTU": [ { "id": "A", "wart": 2300 }, { "id": "G", "wart": "ZW" } ], "podsum": { "waluta": "PLN", "sumaNetto": [ { "brutto": 12300, "vat": 2300, "idStPTU": "A" }, { "brutto": 10000, "vat": 0, "idStPTU": "G" } ], "sumaBrutto": 22300, "sumaPod": 2300 }, "nrDok": 7, "pozycja": [ { "towar": { "brutto": 12300, "ilosc": "1", "oper": false, "cena": 12300, "nazwa": "Nazwa towaru 1", "idStPTU": "A" }, { "towar": { "brutto": 10000, "ilosc": "1", "oper": false, "cena": 10000, "nazwa": "Nazwa towaru 2", "idStPTU": "G" } } ], "nrKasy": "001", "podmiot1": { "nazwaPod": "Nazwa podmiotu", "nrFabr": "WTE2001000009", "NIP": "6970000802", "adresPod": { "ulica": "Ulica", "miejsc": "Miejscowosc", "nrLok": "NrLok", "poczta": "Poczta", "nrDomu": "NrDomu", "kodPoczt": "00-000"}, "nrUnik": "WTE2001000009", "nrEwid": "2020/000001612" } } }
```

[illegible]

### C.3.6a Pełna postać obiektu JWS paragonu fiskalnego w postaci elektronicznej (pierwszego):

### C.3.6b Pełna postać obiektu JWS paragonu fiskalnego w postaci elektronicznej (drugiego):

eyJ1UGFYWyVdwb5iSt5Zn3YucGwiOiJleUozWlhKemFTRWlPaUplLVUV0ZlMwRlRRVjRlUVZKQllwOU9YM1l4TFRBaUxSdktVRXRkUkNjNklqQXdNVEF3TURBd01EQXdNREF3TURBd05TSXNjbVJoZEdGSjVFc2lPaUl5TURJd0xUQTBMVEV3VkrBMU9qSxSpalExTGpZM09Gb2lMQ0pLVUV0U1JVWWlPbnNpVTBoQk1qVTJJam9pUmpnNFJlUWkZOVUZFT1RVMk1UazFNRGN5UmtNME16YzVOVFJCUWpOQ1FUQTFSakjHTTBaR05qYzNSalpC TURFeU9VTknPPVZHUmtNNE16VXpPRUpCTUNJc0lrcFTTMGxFSWpvaU1EQXhNREF3TURBd01EQXdNREF3TURBMElulMTkiLJcbhGcOiJSUz1NiIsmpwa21ldGFkyXRhijoiZXIkamIzSnlaV3hoZEdsdmJrbGtJam9pvkVaRUxSzfVSVEl3TURD01EQXdNRGt1TWpBeU1DMHdOQzB3TTFReE1UbvFNlGR93TUM0ek1qSmFJaXdpWT5dGNISxmjM05wyj10U9pSkSV5pNUVZSRklumD0lCJ4NWMIoIsiTVU9EtqQ3NBcE9nQXdtQkFsnFvSE1U3VvbELWRghLZEfKakXTUoHEncT6k2pXaGtpR3WjEiBUXNGQURBUV15SXdfQVIEVFRREB3wE5SaUu1FVJ0aGmza3dlaGmzVpBd016RTN1NVEF6TlRVNVNdO5NaIV3TXpFMk1UQXpOVFU0V2pDQnJqRwPnQOVHQTFVRUN4TWFSR1Z3WVhKMfIXMWxibFnU1c1bWlZSnZRWFI1ZW1GamFta3hlakFjQmdOVkBB1RGVTFWYm1semRHVnljM1IzYnlCR2FXNWibhk5Z2HpFwk1CY0dBMVVFQJlUNUVZrRlVRXsd0TmprM01EQXdNRGd3TWpFV01CUUdBMVVFQXhNTlYxUkZNakF3TVRBd01EQXdPVEVSTUE4R0EXVUVCEe1JVjGGeWMzcGhkMkV4RkrBU0JnTIZCQWdUQzAxaGvtOTNhv1ZqYTJsbE1Rc3dDUVIEVFRROV3SIFURENDQVNjdORRWUpLb1pJaHjZTKfRRUJCUUEFZ2dfUEFEQONBUW9DZ2dfQKFmDFNINOF3U09U NHFUODYZqkF3YnYqRHZHcUvNrE5WXC9nMGR0V0s4MUM1dks2NzdtYjZn1UwUGNjUkpHNGh0Nlra21SdmZzTzkyWWduwm1wUUVxQUppqNEd5Q2j1rh5ssS2jZ4ZVnaR253dnBBShZwbVBHdlfpbgHNBbUh5U2NMZEdmT1h6UWdyefZcL1dFnrJqSmdlBmRjgdWnZr3U2QVpIUkNXN3J4R01BeEdBNlT20ZwVlPcUXE4aHobDdzQWppfMpbWlKNkd1U1UFAxbnNoemRkOU09Fsm5Kl3llTmJm9nNlbgTjQdTJRTWtclDx2cEZBQ2dZVjlob0hpT283QkNtaVpOzZlBDNHRwbWh4T218xbWRaRfI3SW5TVTBNZ2FZUDNwODAEzjMlMTZNMTwtnMlRrSZHQ0R3jvdWJLaWzWmZrSdW1Gd3VJUfPaZThES0Q4MzRQUThDQXdFQUFhTmVNRnd3RXdzRFZSMGxCQXd3Q2dZSut3WUJCUVVlQXJdJd1JRURURWUjBCQkQ0d1BJQVF4Zytuk20xakFVMWYyM251UzY1WFBhRvdNQlF4RWpBUUJnTIZCQU1UQ1UxR0lHVXRTMkZ6ZVlUjXUM4TFZlWjl1UDVsQWxudlhIOGJZZWpBTkJna3Foa2lHOXcwQkFrcOZBQU9C21FCWmZrXC9LeEV5XC8zeklmUDByNE50NURnNUxiSjczN2tSZklaUmh1M1BIBU10aEFhYzlrVEx0YVWFhVOMraUw3MG5tOHRcLO5a1JE0VV5azgzdkxtTGJQR1ZFYfMNPbFkrNU9vNVhZZXR1YInkTEZuTG00a2lGVURoNU5QNEFLeIFkVHBDcDRqYkFPV1JGskNiciPaYIY2TTYBeXRBVRwvbWxyVHjThdDlMzEtXhSdz09lI19.hVLbbtNAEP2Vap43sHYubf1mallKirFlokpbKN3S3F8WWu9lRVh-XdmxkmlQAV-2fCzjlz2UnQ8rbUIYNdG5V6Kkync717bZudgga-x4vtilyG2zj8Fs6\_RttbnjRQBQkqchwfm-NKXlZcXiElp4E DiBtD7PLqW0cBNTKqidTEWkTkvkwnH39C2dWBgzhVkbLkTxBdbHprEEFvi-



WWNOrrFnbXqf\_ whYk7O4GgqmAysZEgWACctXsNFLAYjAENC5erSH4voeMEEN0dcoS3Vgi3dE9f3XD5gEOP1C6SZu25FmoonU0i\_g-  
lsC2VJF2zjDoo23Z9BBOWlMagAWilgdiZCSW1zwpT3lnWXMMJewPx0yfcccTm5RRD9TrjcEFTUhf\_5LQilCEM6idpJ6ryQrTJlitoWpT01h-  
qqLRAHJdqVNSpfqOmooovWCK9oIqzjqgFv5m4V7-x8JlB7L48McEaASVxdW94L-UHP-QScvMOG-  
4UKzmaRyBjsEWqO6TeqTNP6xufayV9F1jLqLOTu7viTHISRSIVrdMM4e2iJLSnmaXwF4pTvu5wsbpqPuCP6eBg8RvyQr6YejGlxhNWXLKWzgmZ  
qUo9zMCNmbBN7iussvNnZW97bLT0b9ntzfzJzK4Tc.Byu9xqb7YUehEhJphBj6XrJQ5ak3eAic3xbeGbrPDjjDZdQzOXH-DikjMhRr5gJlQatZZ-  
R5QIRWOUor2sYw0J0lpp1g5L5F\_yGWputFLDOUK5qF8nvZKFknAL\_CD51ie7rumKSevE7Tmvlk736RPD01FIRiu-rSgOKic-  
4qTdTekEI9UbDaOAnHnkMNGhEsam9pv4F4sSpy9ronoDEs4S-SGAParwqzy-AZ8XqmUBl6w5p\_X6N8fNjK-  
lzh3su5VmRDo8JbfaAS3S7sz4VUTr8Sifow2woSjQYXXOeqf1x2ZCJzgCv5AntpR5sHzhMNA4Pb48OgrlPKYmthEU9g

### C.3.6c Pełna postać obiektu JWS paragonu fiskalnego w postaci elektronicznej (trzeciego):

eyJlUGFYWdvb5tZi5nb3YucGwiOiIleUozWlhKemFtRWIPaUpLVUV0ZIMwRIRRVjIRUVZKQIlwOU9YM1I4TFRBAuXdsKtVRXRKUKNjNklqQXdn  
VEF3TURBd01EQXdNREF3TURBd055SXNjbVJoZEdGS1VFc2IPaUI5TURjd0xUQTBMVEV3VkrBNE9qSxpPalExTgPZM09Gb2lMQOpLVUV0U1JV  
WWIPbnNpVTBoQk1qVTJJam9pTnpjMk9VTXpPRGxDUWpNelF6ZEdNelJDTURARU1VTkNOa013UkVFeE0wWTVRVVf6TURVNE5VskdSVFEzU  
WpkRVfQqTROVUV5UVRJMK9Ea3dSEV3TVNjc0lrcFFTMGxFSWpvaU1EQXhNREF3TURBd01EQXdNREF3TURBd01MTkiLCJhbGciOiJSUzI1Ni  
IsImpwa21ldGFkYXRhIjoizXlKamlyMXdjBvZ6YzJsdmJpSTZJa1JGUmt4QlZlVWlmlU09lwiwDVjJlpbk1JSURLakNDQXBPZ0F3SUJlBZ0lRMjd1b2xj  
cERIS2RBSmpMU0FIRDQrekFOQmdrcWhraUc5dzBCQVfZkRkFEQVNVNkl3RUZFRZRUURFd2xOUmlCbExVdGhjM2t3SGhjT1qQXdNekUyTVRB  
ek5UVTvXaGNOTWpVd016RTJNVF6TIRVNFdQJQJyakVqTUNFROExVUVDeE1hUkdWd1YsJbZvZfYs5mRZ1NXNW1iM0p0WVhSNWVtRmPhb  
Wt4SGpBY0JnTlZCQW9URlUxcGJtbHpkR1Z5YzNSM2J5QkdhVzVoYm5OdmR6RVpNqMnHQTFVRUJSTVFwa0ZVVU53dE5qazNNREF3TURnd0  
1qRVdNQLFHQTfVRUF4TU5WMVJGTWpBd01UQXdNREF3T1RFUk1BOEdBMVVFQnhNSVYyRnljM3BoZDJFeEZEQVNCZ05WQkFNVEMwMWhl  
bTkzYVdWamEybGxNUXN3Q1FZRFZRUUdFd0pRVERDQ0FTSXdEUVlKS29aSWH2Y05BUUVCQIFBRGdnRVBBRENDQVfVqZdnRUJBTHTSDdBd  
1NPVDRxVdG2M0JbD2JOMkR2R3FFZROVlwwZzBkdFdLODFDNXLNjc3bWl5WTDVmfBjY1JKRzRodDdCK2ttUnZmc085MllnbmptcFFcUFKajR  
HeUNpY0dYYUkrNmY2eGVVUWtud3ZwQUh2cG1QR3ZaamxzQW1leVNjTGRhC9UYelFncnhWXC9XRTY0akpnZW5kY3Vjc0d1NkFaSjFDVdyE  
dNQXhHQTJ5U2dFce5aQlF0GhwaGw3c0FqajBQSFpDsndVNBQMW5zTnpkZFNPUPuZEpZSE5iYnJvbTZAkxianUyUU1XRc9nV3BGQUINNw  
VY5aG9IaU9vN0JDU2laM2ZQZr0CG1oeE9QcW1kWRZd0luU1UwTWdhWVAzcDgwM3oyZjE2ZzFrZjUzUjBGR0NEdyb3ViS2lmlZlc0UnVtRnd1S  
VBaWmU4REteODM0UFE4Q0F3RUFBU5ITUZ3d0V3WURWUjBsQkF3d0NnWUilD1ICQIFVSEF3SXdSUVIEVlWqKJENHdQUSFReGcrbitqMWp  
BVTfMjMjNudVM2NVhQYUUVXTUJReEVqQVFCZ05WQkFNVENVMUdJR1V0UzJGemVZSVFzOExWZVo5dVA1bEFsbnZY2ThiWVWVQQU5CZ2txaG  
tpRzl3MEJBUXNGQUFPQmdRQlpma1wvS3hFeVwvM3pJZlAwcJR0dDVEZzVMYko3MzdrUmZlWlJodTNQSG1NdGhBYWMya1RMDGfHYVdDK2  
IMNzBubTh0XC9KOWtSRDlVeWs4M3ZMbUxiUEdWRWJt2zKzVPbzVYVWV0dWJTZEgGbkxtNGtpRIVEaDVOUDRBS3pRZFRwQ3A0amJBT1d  
SRkpDYnJaWmJWNk0wWHlOQUVcl21scRybWw5SzMxM014Unc9PSJdfQ.hVJbt5tADP4rIT9ftoNmScs3tnZRI46hJVGITFPkwqjvBw6jqI5Sb\_P  
PmikdtX2CeOznxfB0h13paqshAcoMKHqncq57hTptkjpBAIXu6W4SrxceH3cPetjZ16EwkCURlRb1ThS19O5HTiybWcB\_55MP3wbja\_2MJRQIOG  
H3TFIDWWmUo-  
\_TYQeAKstlhwuse62HZGkwLfp5eSmnrMV7XpVfovcMHKbq4gmAuoTMw8EEwF5NjsFXHAcggENDZebyD4cYCMElUNKdWiYz9GN6cUpDds7OP  
4k7Tpt2tina4vW8jDi24gB2xijZa12oPemdaHHcAlecUQWhLsaqlImTeU6VUp4qQ5QtHc\_DfxxL\_QGJU7FOnyWT4StNa5qxyP4p4UWREqvJAot9J  
SkrdJMqQEFadc3D-YVfowQkqsJTVYV9xx4j\_p45rPaMMW17YYB9\_8wyG\_sczVL3J48OrOfAgKkM7fKJ\_KT13TDotM2294Vap241kBBofW-C-  
z3jPj3C3vvaPl3nIJT1ENzFlZ5dzTlxJlSxUqMbhHKAtoSvbdxAN3r3hn66GldS2Gv-XhQ-X-LCvph-sYgmE9ZetKXEBXqVP36sxMiB3cFjdVlr-  
hszLXXfZ8\_e9d2pt5Pk3l-Ac.coZL-RnBrdTg9Z2fi7heBuQrL8yaKACI7mvUu7ox0h1MJqo9HNKigYupzXz\_q54j8bYN4qm5n6FS-  
qQ6wCMfz4mRf\_HGfxf1pQG3FM7RvB6RAtoGkwywqA77RZuqISP\_a4IemcIjpL2z2uoqBO2VqB8OSAIPzQ7YwBJWBMGD2WS0ztEZGpqcfBpW  
PZhyW1Kw0CvI6JeynHWTgeglGNpj23Z9VslvmSoBKwX7z0kpWd0jD4BtMX5CrNDiDcmh5Q04TOy8P69E8nDrJzJ7HBBKEGzX8LRT03JqQfujXh  
HtDfGb-xZXOmfrV9XcTRrb4mMD2wjJR8NQu3GQasYznoeQLQ

### C.3.6d Pełna postać obiektu JWS paragonu fiskalnego w postaci elektronicznej (czwartego):

eyJlUGFYWdvb5tZi5nb3YucGwiOiIleUozWlhKemFtRWIPaUpLVUV0ZIMwRIRRVjIRUVZKQIlwOU9YM1I4TFRBAuXdsKtVRXRKUKNjNklqQXdn  
VEF3TURBd01EQXdNREF3TURBd09DSXNjbVJoZEdGS1VFc2IPaUI5TURjd0xUQTBMVEV3VkrBNE9qSxpPalExTgPZM09Gb2lMQOpLVUV0U1JV  
WWIPbnNpVTBoQk1qVTJJam9pUwTKR05ETkVRVGhHTVVKQ016RkJSVFpHTXpFME9Ua3dOVGN6TWtNMU1FRkVOMEU0T0VKRE5rRXIRMFV  
5UVRrek5qQkNORFZEUWtNeFwUkrSakJHTVNj0lrcFFTMGxFSWpvaU1EQXhNREF3TURBd01EQXdNREF3TURBd01MTkiLCJhbGciOiJSUzI1Ni  
NiIsIng1YyI6WjYJNSUIES2pDQ0FwT2dBd0lCQWdJUTI3dW9sSXBESEtkUvqTfNBSEQ0K3pBTkJna3Foa2lHOXcwQkFRc0ZBREFTVJJd0VBWUR  
WUVFERXdsTlJpQmxmMVXRoYzNrd0hoY05NakF3TXpFMk1UQXpOVFU1V2hjTkl1qVXdNekUyTVRBek5UVTXakNCmpFak1DRUDBMVVFQ3hN  
YVJHVndZWEowWVxcBGuUWdTVzYtYjNkdFIYUjVlbuZqYw1reEhQqWNCZ05WQkFvVEZVMXBibWx6ZEdWeWmZUjNieUJHYVc1aGJuTnZkek  
VaTUJJR0EXVUVCUK1RVmtGVVVFd3ROamszTURBd01EZ3dNakVXTUJRROExVUVBeE1OVjFSRk1qQXdNVEF3TURBd09URVJNQThHQTfVRUJ4T  
UIWMkZ5YzNwaGQyRXhGREFTQmdOVkJBZ1RDMDFoZW05M2FXVmphMmxsTVFzd0NRWURWUUFHRXkdUUVREQONBUOI3RFFZSkztWklodm  
NOQVFFQkJRQURnZ0VQURDQ0FRb0NnZ0VCQUx0U0g3QXdtT1Q0cVQ4NjNCQXdidDEdKdxRWdETlZlc2cwZHRXSzgzV2S5Y3N2I1oVvk3V  
TBQY2NSSkc0aHQ3QitrbVJ2ZnNPOTJZ25qbXBRRXFBsMo0R3lDaWNHwGfJKzZmNnhVVFrbnd2cEFIdnBtUEd2Wmpsc0ftSHITY0xkr3FPWH  
pRZ3J4VlwwV0U2NGpKZ2VuZGN1Y3NHdTBZWkhSQ1c3cnhHTUF4R0EYevNnRXBOWkJRcThocGhsN3NBamowUEhaQ0p3VTVQUdFuc056ZGR  
T0VQkmbRKWUHOYmJyb202WkZMYmp1MIFNa1wvZ1dwRkFDZ1lWOWhvSGlPbzdCQ1NpWjNmUEM0dBtBaHhPUHFZfPEWXdjblINVME1nY  
VlQM3A4MDND6MmYXmMxc2cyVGxkdDRHcm91YktpZmZNFJ1bUz3dUIQWlplOERLRDgzNFBROENBd0VBQWUJlZU1Gd3dFd1lEVllwbEjBd3d  
DZ1lJS3dZQkJRVUhbD0l3UIFZFSZMEJCRDR3UElBUXhNk24rajfQqVUxZlzbNVTnjYUyGFFV01CUXhFakFRQmdOVkJBTVRDVTfHSUdVdFmYRn  
plWUlrCzhMVMvaOXVQNwXbBg52WGU4YllakFOQmdrcWhraUc5dzBCQVfZkRkFBT0JnUuJaZmtcl0t4RXlClZn6SWZQMHI0TnQ1RGc1TGJKN  
zM3a1JmSVpSaHuZUEhtTXRoQWfjMmtUTHRhYWFxQytpTDcwbm04dFwvSjlrUkQ5VXlR0DN2TG1MYIBHvkViY09sWsS1T281VfildHViU2RM  
Rm5MbTRraUZVRGg1TIAOQUt6UWRUCeNwNgpiQU9XUKZKQ2JyWlpiVjZNMfh5dEFFXC9tbHJUcm1sOUsZMTNNeFJ3PT0iXX0.eyJkb2t1bWVv  
dCl6eyJuYWdsb3dlayl6eyJ3ZXJzameiOiJkUetfS0FTQV9QVjBR09OX3YxLTAlClkYXRhSIBlIjoiajMjYmM0wNC0xMFQwODoyMzo0NS42Nzhln  
OslNBhcmFnb24iOncicGfTaVWjQ2hyjoxLCl0b3RhbCl6eyJ6YXBsWndyb3QiOjlyMzAwfSwiemFrU3ByemVkljoiajMjYmM0wNC0xMFQwODoyMz  
o0NS42NzhaliwiSIBLSUQiojgsIm5yUGFyYWciOjUslmthc2plci16lthc2plci1slnN0UFRVljbeyJpZCl6lkeiLCJ3YXJ0ljoYmZAwfSx7mlkljoYrYslndhc

nQioiAaVyJ9XSwicG9kc3Vtljp7IndhbHV0YSI6IlBMTilsinN1bWFOZXR0byl6W3siYnJ1dHRvJjoxMjMwMwCwidmF0ljoyMzAwLCljPzFN0UFRVljoIQ  
Sj9LHsiYnJ1dHRvJjoxMDAwMCwidmF0ljowLCljPzFN0UFRVljoIRyJ9XSwic3VtYUyjdXR0byl6MjJzMDAsinN1bWFBq2QiOjIzMDB9LCluckKvayl6  
NywicG96eWNqYSi6W3sidG93YXliOnsiYnJ1dHRvJjoxMjMwMwCwiaWxcv2MiOixliwib3Blcil6ZmFsc2UslmNlBmEiOjEyMzAwLClJuYXp3YSi6Ik5h  
endhIHrVd2FydSaxliwaWRTdFBUbVSi6IkEiOseyJ0b3dhcil6eyJicnV0dG8iOjEwMDAwLCljbG9zYyYl6ljiElCjvcGVyYjlpMwYwXzSziW2YvYsi6MTA  
wMDAsim5hendhIjoiTmF6d2EgdG93YXli1DIilCljPzFN0UFRVljoIRyJ9Y9lsm5ys2FzeSi6lJawMSJ9LClwb2RtaW90MSi6eyJyXp3YVbVzCil6Ik5h  
endhIHbVzG6l1pb3R1liwibnJGYWYjlyoiV1RFMjAwMTAwMDAwOSIsIk5JUCi6lJy5NzAwMDA4MDiElCjHJlC1BvZCil6eyJ1bGljYsi6lVsaWNhliwib  
WllanNjlioiTWllanNjb3dvc2MilCJuckxvayl6Ik5yTG9rliwicG9jenRhIjoiUG9jenRhliwibnJeb211ljoITnJeb211liwa29kUG9jenQiOilmwMC0wMDAif  
SwibnJvbmllrjoiV1RFMjAwMTAwMDAwOSIsIm5yRXdpcZCil6lJwMjAwMDAwMDAxNjEynl19Fq.sxTOYlO9QSQpRreOSJM9nLaPlLu-6R5g-L-  
VPGE37SbMASEHmMs4XxmmCc-YjLgimJyL4gRbyl8A5PwdsSZS0H4zhW4xdGftBlPyhnbuiuj4J3rVxq0-  
R1Sdj3mDyzllm1JefllyHdfYl21KcyT3Yuv9d2SjtCAqrOKUGoY-  
EnGhefo5dcaV7LHZxQm7trJL\_xrunK3Pe3S\_anzQW9n14nqpCHBwBKBG7k6b1wPcoh3exUhnJodVGkuGAVCABvuQ\_Zc9wySc3aPtFurPj5vgeg  
1BEu-mt58dzcqW9YfoCrwWBZnh7gBbDeOzrnjKEKhtzv3W0Wa9qd89HdcsTEg