

# EOC Vysočina

## Tokenizační algoritmus



## Obsah

|  |          |
|--|----------|
| <b>1. Úvod.....</b>                                  | <b>3</b> |
| <b>2. Popis algoritmu .....</b>                      | <b>4</b> |
| 2.1 Popis algoritmu pro bankovní karty.....          | 4        |
| 2.2 Popis algoritmu pro ostatní identifikátory ..... | 4        |
| <b>3. Identifikátory.....</b>                        | <b>5</b> |
| 3.1 Bankovní karta.....                              | 5        |
| 3.2 In Karta 2. generace.....                        | 5        |
| 3.3 Jihlavská karta .....                            | 5        |
| 3.4 Mobilní aplikace .....                           | 6        |
| <b>4. Principy bezpečnosti.....</b>                  | <b>7</b> |
| <b>5. Příklad výpočtu.....</b>                       | <b>8</b> |
| 5.1 Příklad výpočtu pro bankovní kartu .....         | 8        |
| 5.2 Příklad výpočtu pro In Kartu.....                | 8        |

Tento dokument a veškerý jeho obsah je chráněn autorským právem a dokument i veškerý jeho obsah je, s eventuální výjimkou explicitně uvedených částí, vlastnictvím společnosti ODP–software, spol. s r. o., IČO 61683809, DIČ CZ61683809, sídlem Perneroва 2819/2a v Praze 3, PSČ 130 00, zapsané v obchodním rejstříku vedeném u Městského soudu v Praze, sp. zn. C 37829. Tento dokument nesmí být reprodukován ani citován, ať z části nebo v celku, bez předchozího písemného souhlasu jeho vlastníka nebo Kraje Vysočina.

© Copyright ODP–software, spol. s r. o. 2021.

## 1. Úvod

Tokenizací se pro účely tohoto dokumentu rozumí kryptografická operace, která bezpečně převádí hodnotu zvoleného identifikátoru (například bezkontaktní čipové karty) na hodnotu zástupnou, tzv. token. Samotný postup výpočtu, pomocí něhož je z hodnoty identifikátoru za pomoci tajného klíče token odvozen, nazýváme tokenizačním algoritmem.

Cílem tokenizace je utajení původní citlivé hodnoty identifikátoru, například čísla bankovní karty, pro použití v dopravních a zúčtovacích systémech. Na rozdíl od citlivého údaje, kterým je snadno zneužitelné číslo bankovní karty, může být token předdáván i do potenciálně nedůvěryhodných prostředí, aniž by došlo ke kompromitaci původní citlivé hodnoty.

Základními požadavky na tokenizační algoritmus jsou:

- Ireversibilita – ze znalosti tokenu není možné odvodit původní číslo identifikátoru
- Unikátnost výstupu – pro každé dvě rozdílné vstupní hodnoty identifikátorů je výstupem algoritmu rozdílný token (za předpokladu omezené délky hodnoty identifikátoru).
- Důvěrnost – útočník není schopen pro vstupní hodnotu identifikátoru odvodit jeho token bez znalosti tajného klíče
- Dostupnost – komponenta, která token počítá, musí být schopna dojít k výsledku v přiměřeném čase a za využití přiměřených prostředků. Tokenizační algoritmus bude počítán i při odbavení v autobusové linkové dopravě, jeho výpočet nesmí ohrozit plynulý nástup cestujících.
- Nezávislost na fyzickém nosiči – Token by měl být pro ty identifikátory, u kterých je to možné, počítán pouze z údajů dostupných cestujícímu, ideálně tedy vytištěných na nosiči samotném v lidsky čitelné podobě. Cílem je, aby cestující mohl provést registraci identifikátoru přes webový prohlížeč a serverová část mohla provést prvotní tokenizaci identifikátoru bez fyzické přítomnosti nosiče.

## 2. Popis algoritmu

### 2.1 Popis algoritmu pro bankovní karty

Tokenizační algoritmus EOC Kraje Vysočina pro bankovní karty je definován předpisem:

**TOKEN (DATA, KEY1, VERSION) := VERSION || SHA256 (2K3DES (KEY1, IV, DATA || PADDING\_FF) )**

s následujícím významem jednotlivých pojmů/zkratk:

| Název      | Význam   | Poznámka  |
|------------|--|---|
| TOKEN      | Výstupní pole bajtů pevné délky 33 B.  |   |
| DATA       | Vstupní pole bajtů, získané z čísla a expirace bankovní karty postupem popsaným v kapitole <b>Identifikátory</b> .   |   |
| KEY1       | Tajný symetrický klíč šifry 2K3DES ve vlastnictví poskytovatele tokenizace bankovních karet, předaný bezpečnou cestou Kraji Vysočina nebo přímo nahraný do platebního terminálu. | Délka 16 B.   |
| VERSION    | Verzový byte   | Konstanta 0x11 (algoritmus verze 1, klíč č. 1)  |
| SHA256     | Hashovací funkce SHA256.   | <a href="https://cs.wikipedia.org/wiki/Secure_Hash_Algorithm">https://cs.wikipedia.org/wiki/Secure_Hash_Algorithm</a> |
| 2K3DES     | Symetrická bloková šifra 2K3DES s délkou klíče 16 B, mód CBC   | <a href="https://en.wikipedia.org/wiki/Triple_DES">https://en.wikipedia.org/wiki/Triple_DES</a>                       |
| IV         | Inicializační vektor   | Pro účely tokenizace v EOC Kraje Vysočina bude vždy použit <b>nulový</b> inicializační vektor.                        |
| PADDING_FF | Zarovnání na délku bloku šifry byty s hodnotou 0xFF  |   |

### 2.2 Popis algoritmu pro ostatní identifikátory

Tokenizační algoritmus EOC Kraje Vysočina pro ostatní identifikátory (kromě bankovních karet) je definován předpisem:

**TOKEN (DATA, KEY2) := SHA256 (AES256 (KEY2, IV, DATA || PADDING\_PKCS7) )**

s následujícím významem jednotlivých pojmů/zkratk:

| Název         | Význam  | Poznámka  |
|---------------|---|---|
| TOKEN         | Výstupní pole bajtů pevné délky 32 B.   |   |
| DATA          | Vstupní pole bajtů, získané z hodnoty identifikátoru postupem popsaným v kapitole <b>Identifikátory</b> . | Nemusí být zarovnáno na délku bloku šifry AES256.   |
| KEY2          | Tajný symetrický klíč šifry AES256 ve vlastnictví Kraje Vysočina.   | Délka 32 B.   |
| SHA256        | Hashovací funkce SHA256.  | <a href="https://cs.wikipedia.org/wiki/Secure_Hash_Algorithm">https://cs.wikipedia.org/wiki/Secure_Hash_Algorithm</a>                                       |
| AES256        | Symetrická bloková šifra AES s délkou klíče 32 B, mód CBC   | <a href="https://cs.wikipedia.org/wiki/Advanced_Encryption_Standard">https://cs.wikipedia.org/wiki/Advanced_Encryption_Standard</a>                         |
| IV            | Inicializační vektor  | Pro účely tokenizace v EOC Kraje Vysočina bude vždy použit <b>nulový</b> inicializační vektor.  |
| PADDING_PKCS7 | Zarovnání na délku bloku šifry dle normy PKCS7  | <a href="https://en.wikipedia.org/wiki/Padding_(cryptography)#PKCS#5_and_PKCS#7">https://en.wikipedia.org/wiki/Padding_(cryptography)#PKCS#5_and_PKCS#7</a> |

### 3. Identifikátory

Identifikátorem v prostředí EOC Kraje Vysočina mohou být následující média:

- Bezkontaktní bankovní karta (VISA, MasterCard)
- In Karta 2. generace Českých drah na platformě MAP s nosičem Mifare DESFireEV1
- Jihlavská karta na nosiči Mifare DESFire
- Mobilní aplikace

Seznam identifikátorů je možné v budoucnu rozšířit o další typy dle požadavků Kraje Vysočina.

#### 3.1 Bankovní karta

Pro bankovní kartu budou jako vstup do tokenizačního algoritmu využity následující údaje vytištěné na těle nosiče:

- Číslo karty
- Expiry karta

Vstupní pole bajtů je pro bankovní kartu definováno jako:

**DATA := BCD (CARD\_NUM || D || EXPIRATION\_YY\_MM || D)**

Pro bankovní kartu s číslem **1234 5678 1234 5678** s expirací **22/09** je tedy vstupními daty pole bajtů (hexadecimálně):  
**1234567812345678D2209D**

#### 3.2 In Karta 2. generace

Pro In Kartu budou jako vstup do tokenizačního algoritmu využity následující údaje vytištěné na těle nosiče:

- Číslo karty (PAN)
- Expiry karta

Vstupní pole bajtů je pro In Kartu definováno jako:

**DATA := ASCII (PAN || EXPIRATION)**

Pro In Kartu s číslem **920311201010002036** s expirací **08/23** je tedy vstupními daty pole bajtů (hexadecimálně):  
**39323033313132303130313030303230333630383233**

#### 3.3 Jihlavská karta

Pro Jihlavskou kartu bude jako vstup do tokenizačního algoritmu využito UID karty Mifare DESFire dle standardu ISO-14443A. Tato hodnota není z karty lidským okem čitelná. Pro případnou registraci Jihlavské karty prostřednictvím webového rozhraní bude nutné vytvořit na backendu převodní službu (tj. cestující zadá číslo Jihlavské karty, backend EOC Kraje Vysočina se dynamicky dotáže backendu Dopravního podniku města Jihlavy na UID zadané karty). Důvodem k tomuto kroku je fakt, že číslo Jihlavské karty není veřejně čitelné z datové struktury karty v okamžiku odbavení bez znalosti čtecích klíčů.

Vstupní pole bajtů je pro Jihlavskou kartu definováno jako:

**DATA := 0x00 || UID**

Pro Jihlavskou kartu s UID (hexadecimálně) **11223344** je tedy vstupními daty pole bajtů (hexadecimálně):

**0011223344**

Prefix 0x00 bude využit pouze pro Jihlavskou kartu. V případě, že by byl v budoucnu do systému EOC Kraje Vysočina zaveden další identifikátor se stejným principem tokenizace (tedy dle UID), bude mu přiřazen vlastní prefix (0x01, 0x02 atd.).

### 3.4 Mobilní aplikace

Mobilní aplikace bude pro identifikaci účtu používat jednoznačný identifikátor kódovaný do pole bajtů a předávaný v QR kódu v části „Datová část specifická pro zákazníka“, jak je uvedeno v dokumentu „EOC VDV – Analýza a návrh 2D kódu“. Vnitřní strukturu tohoto identifikátoru tento dokument nepopisuje; bude stanovena dodavatelem mobilní aplikace.

Vstupní pole pro tokenizaci je potom pro mobilní aplikaci definováno jako

**DATA := 0xFF || ID**

kde ID značí identifikátor účtu v mobilní aplikaci. Pro identifikátor „11223344“ jsou tedy vstupní data pro tokenizaci

**FF11223344**

## 4. Principy bezpečnosti

Pro zachování maximální bezpečnosti je doporučeno dodržet následující pravidla:

- Tokenizační klíče v držení Kraje Vysočina jsou tajné a mohou být předávány pouze bezpečnou cestou za dodržení všech podmínek bezpečnostní politiky Kraje Vysočina (ideálně tedy přímo v bezpečném HW úložišti – HSM, SAM modulu..., případně zašifrované transportním klíčem a rozdělené pro přenos do několika nezávislých komponent).
- Všechny SW a HW komponenty, které před samotným výpočtem tokenu přijdou do styku s citlivými hodnotami identifikátorů, musí v maximální možné míře dbát na bezpečnost a využít všech dostupných prostředků, aby zabránily úniku citlivých dat.

Tento dokument nespecifikuje pravidla pro nakládání se samotnými identifikátory, nestanovuje žádná omezení pro použití jednotlivých typů ve vztahu k odbavovacím zařízením.

## 5. Příklad výpočtu

### 5.1 Příklad výpočtu pro bankovní kartu

Předpokládejme bankovní kartu s číslem **1234 5678 1234 5678** s expirací **08/23**. V příkladu budeme používat testovací klíč **12ab991756e4aa1890efdb384029b477**

Výpočet tokenu pak pro uvedenou kartu probíhá v následujících krocích:

1. Převod hodnoty identifikátoru na vstupní data  
**1234567812345678 08/23 ->**  
**1234567812345678D2308D**
2. Doplnění vstupních dat paddingem do délky bloku šifry 2K3DES  
**1234567812345678D2308D ->**  
**1234567812345678D2308DFFFFFFFFFFFF**
3. Výpočet 2K3DES  
**1234567812345678D2308DFFFFFFFFFFFF ->**  
**ecb858e7e8b14a13a38381668f59a105**
4. Výpočet SHA256  
**ecb858e7e8b14a13a38381668f59a105 ->**  
**c5bdd02e0f245759c358d9459f6a112130c30e5d37eaa6150970db06c96a87e9**
5. Předřazení verzového bytu  
**c5bdd02e0f245759c358d9459f6a112130c30e5d37eaa6150970db06c96a87e9->**  
**11c5bdd02e0f245759c358d9459f6a112130c30e5d37eaa6150970db06c96a87e9**

### 5.2 Příklad výpočtu pro In Karty

Předpokládejme In Karty s logickým číslem **920311201010002036** a expirací **08/23**. V příkladu budeme používat testovací klíč **7ff8d84d1c3d73a86b1d50347fc1e47174af41192bdf5909b4a968083ce51982**.

Výpočet tokenu pak pro uvedenou kartu probíhá v následujících krocích:

6. Převod hodnoty identifikátoru na vstupní data  
**920311201010002036 08/23 ->**  
**39323033313132303130313030303230333630383233**
7. Doplnění vstupních dat paddingem PKCS7 do délky bloku šifry AES  
**39323033313132303130313030303230333630383233 ->**  
**393230333131323031303130303032303336303832330A0A0A0A0A0A0A0A0A**
8. Výpočet AES 256  
**393230333131323031303130303032303336303832330a0a0a0a0a0a0a0a0a ->**  
**f38ea27e8d4469a35e1dc2fe0cb3b12fea9eac77c104bc4d5f3ce5fa12489dcb**
9. Výpočet SHA256  
**f38ea27e8d4469a35e1dc2fe0cb3b12fea9eac77c104bc4d5f3ce5fa12489dcb ->**  
**359502a49e4935ffeedfc05e1e54bf7ffcae6527e569e81faa84fed661258ae5**

Formát uložení tokenu v back-office EOC Kraje Vysočina nebo ve whitelistu tento dokument nespecifikuje. Doporučuje se spolu s hodnotou tokenu ukládat i ID a verzi použitého klíče, aby mohly být v případě kompromitace klíče efektivně přegenerovány všechny tokeny za pomoci tohoto klíče vytvořené.