Struktura 2D kódu
pro jízdní doklady ODIS

Verze 1.6

Historie změn:

Verze	*Datum*	*Jméno*	*Důvod vydání*	*Status*
1.0	14. 4. 2020	Matocha (ODP)	vznik dokumentu	N
1.1	24. 4. 2020	Nenka	Doplnění printsceenu aplikace	N
1.2	14. 8. 2020	Matocha (ODP)	Aktualizace, doplnění výpočtu dynamického proužku	N
1.3	15. 3. 2021	Matocha (ODP)	Doplnění postupu ověření dynamické hodnoty v QR	N
1.4	23. 3. 2021	Nenka	Odstranění neexistujících odkazů	N
1.5	30. 3. 2021	Nenka	Doplnění, že zastávky budou v CIS	A
1.6	20. 9. 2021	Matocha (ODP)	Úprava položky TicketType pro celosíťovou jízdenku	A

Status : A – Aktuálně platný, R – Revize, N – Neplatný

Obsah

Úvod

Tento dokument popisuje 2D kód používaný v mobilní aplikaci Můj ODIS.

1 Struktura QR kódu

1.1 QR kód jízdenky ODIS

Pro uložení jízdního dokladu je použit QR kód. S ohledem na velikost displeje pro zobrazení QR kódu a citlivost odbavovacích zařízení je jako maximální limit pro velikost jednoho kódu stanoveno 1091 B (QR kód verze 23). Jsou-li přenášená data menší než 1091 B, je automaticky použit QR kód nižší verze.

Může dojít k situaci, kdy je potřeba přenášet větší data, tedy je nutné zobrazit více než jeden QR kód. V tom případě je použit tzv. „cyklický" („dynamicky se měnící") QR kód, tedy více kódů limitní velikosti, které se na displeji zařízení pravidelně střídají.

1.2 MAP struktura dat ODIS

Z pohledu dodavatelů prodejních a odbavovacích zařízení je datová věta, uložená do 2D kódu, definována jako struktura REC_MAPPHONE_DATA v dokumentu MAP Karta – Datové struktury (verze 13), kterou mají k dispozici.

Přenášená data se skládají z následujících částí:

- Metadata o přenášené struktuře
- Jízdenky a jejich podpisy
- Průkazy a jejich podpisy (pro ODIS nevyužito).
- Identity Pack a podpis (pro ODIS nevyužito).
- Dodatečná data definovaná zákazníkem (bez podpisu) (pro ODIS nevyužito).

Tato data jsou rozdělena do úseků o velikosti až **1089 B**, každý úsek bude opatřen hlavičkou, tzv. řídícími byty, které mají hexadecimální strukturu 0xCCXY, kde:

- CC je indikátor cyklického kódu, má hodnotu 0xCC,
- X je pořadové číslo QR kódu z intervalu [0, 15],
- Y je celkový počet QR kódů.

Výsledných až 1091 bytů je přeneseno v jednom QR kódu, jednotlivé úseky jsou sestaveny v paměti odbavovacího zařízení.

Položka	***Popis***	***Počet bytů***	***Poznámka***
Verze struktury	Konstantně hodnota „1"	1	Povinné
Metadata	Info o struktuře (vizte níže)	1	Povinné
Jízdenky (Opakuje se dle počtu jízdenek**)**	REC_TICKET_HEADER	12	Pro ODIS zatím vždy jeden nebo více výskytů

	REC_TICKET_BASIC	39	
	REC_TICKET_SELL	15	
	Počet segmentů	1	
	Až 4 segmenty	Max. 112	Proměnlivá délka
	ID účtu	4	Big endian
	ID ECC klíče	1	
	ECC podpis	48	
Průkazy (Opakuje se dle počtu průkazů)	REC_PASSPORT_HEADER	12	Na ODIS nevyužito (hodnota 0, další položky neuloženy)
	REC_PASSPORT_DATA	36	
	ID účtu	4	Big endian
	ID ECC klíče	1	
	ECC podpis	48	
Identity Pack	Info o identity packu (viz níže)	1	Na ODIS nevyužito, tedy zde hodnota 0 a další položky nejsou uloženy
	Vydavatel identity packu (service provider dle MAP číselníku)	3	Big endian
	Formát fotografie	1	
	Délka fotografie	2	Big endian
	Fotografie	Max. 1300	Proměnlivá délka
	REC_CARDHOLDER_BASIC_INFO	18	
	REC_CARDHOLDER_PERSONAL_INFO	64	
	Délka dat specifických pro poskytovatele služeb	2	BigEndian
	Data specifická pro poskytovatele služeb	Max. 32767	
	ID účtu	4	Big endian
	ID ECC klíče	1	
	ECC podpis	48	
Datová část specifická pro zákazníka	Délka dat	2	Pro ODIS vždy 0 nebo 2
	Data	Max. 32767	2 bajty obsahující časové razítko QR kódu (mimo časovou platnost QR kódu nemusí být časové razítko přítomno)
Uživatelské jméno	Délka uživatelského jména	2	Pro ODIS vždy 0 (nevyužito) Big endian
	Data	Max. 32767	

Detailní popis vybraných položek:

Metadata – binární tvar JJJJPPPP, kde

- JJJJ je číslo z intervalu [0, 15] a značí počet jízdenek v přenášených datech (pro ODIS nyní vždy 1 nebo více),
- PPPP je číslo z intervalu [0, 15] a značí počet průkazů v přenášených datech (pro ODIS nyní vždy 0),

Info o identity packu – binární tvar SSDDDDHH, kde SS je číslo z intervalu [0, 3] a značí stav identity packu:

Hodnota	***Význam***
0	Nepřítomný (v tomto případě je „Info o identity packu" jediný přítomný byte z celého identity packu)
1	Potvrzený
2	Nepotvrzený, obsluha zařízení by měla provést verifikaci osobních údajů a fotky
3	Zamítnutý

- DDDD je číslo z intervalu [0, 127] a jedná se o pořadové číslo identity packu v rámci jednoho účtu;
- HH jsou příznaky, zda jsou přítomny struktury nesoucí informace o držiteli: 0 – není přítomna, 1 – je přítomna. Vyšší bit značí REC_CARDHOLDER_BASIC_INFO, nižší REC_CARDHOLDER_PERSONAL_INFO.

ID ECC klíče – identifikátor klíče v rámci číselníku poskytovatele služeb, který vydal produkt. S číselníkem musí být obeznámena odbavovací zařízení a na základě identifikátoru musí být schopna dohledat příslušný veřejný klíč. Podpisový klíč je tajný a musí být držen v bezpečném úložišti (SAM modul, HSM, …). Pro jízdenky ODIS je vždy uvedena hodnota 0x01. Ostré jízdenky lze ověřit pomocí veřejného klíče s hodnotou:

```
04 20 F1 B8 12 A9 F6 1A 6D CC ED 18 9A 0E A7 98 80 1F 35 CC 53 9B 50 FB
29 4D 34 1E 7E C1 9F AC 1D 66 FF 48 89 6B 38 96 C2 12 3A 7F CC B9 E1 3F D9
```

ECC podpis – podpis je kódován jako dvě neznaménková čísla, každé do 24 bytů, použitá eliptická křivka **secp192r1**.

ID účtu – číselný identifikátor účtu, přiřazený serverem v okamžiku založení. Odbavovací zařízení musí zkontrolovat, že se identifikátory ve všech předkládaných produktech rovnají.

Formát fotografie – číselník formátů:

Hodnota	***Význam***
0	Ztrátové WEBP
1	Bezztrátové WEBP
2	JPG
3	PNG
4 – 255	RFU

REC_TICKET_HEADER

Název	*Bitů*	*Typ*	*Hodnota / význam*
TicketStatus	7	Status Code	Status jízdenky (vychází z kapitoly 7.27 EN 1545-1). Na ODIS nyní vždy 7 (platný doklad)
ContractValidityBeginDate	14	Date Stamp	Datum počátku platnosti dokladu
ContractValidityBeginTime	11	Time Stamp	Čas počátku platnosti dokladu
ContractValidityEndDate	14	Date Stamp	Poslední den platnosti dokladu
ContractValidityEndTime	11	Time Stamp	Čas konce platnosti dokladu
ContractNetworkID	12	Network ID	Omezení platnosti jízdenky na vybranou síť (ODIS: 134)
ContractProviderID	24	Provider ID	Omezení platnosti jízdenky pouze na vybraného dopravce (0: nevyužito, platí v celé síti dle contractNetworkID)
VisibleForOtherProviders	1	FLAG	Viditelný pro jiné dopravce? V rámci ODIS vždy 1.
IsInterrupted	1	FLAG	Platnost dokladu může být přerušen? (na ODIS nevyužito)
RFU	1		
Celkem	96 b,	12 B	

REC_TICKET_BASIC

Název	*Bitů*	*Typ*	*Hodnota / význam*
NipSystem	8	ENUM	Způsob práce se souborem kleští (s časovou platností jízdenky). Pro ODIS konstantě 0.
TicketValidityFromNip	16	INTEGER(0..8190)	Pokud není nastaven v hlavičce příznak IsInterrupted, pak platnost jízdenky od posledního označení v minutách. Pokud je příznak IsInterrupted nastaven, pak číslo určuje od kdy na kolik dnů je platnost dokladu přerušena. ODIS konstantě 0 – nevyužito.
LastContractValidityType	2	ENUM	Typ hodnoty LastContractValidity ODIS konstantě 0 – nevyužito.

LastContractValidity	14	INTEGER(0..16383)	Hodnota prolongace dokladu ODIS konstantě 0 – nevyužito.
ContractPriceUnit	4	Pay Unit Map	Měna a násobek ceny jízdního dokladu při prodeji nebo prolongaci ‘0000‘B – CZK v celých korunách ‘1000’B – CZK v haléřích ‘0001‘B – EUR v celých jednotkách ‘1001’B – EUR v centech Pro ODIS CZK v haléřích.
ContractPaymentMeans	4	Payment Means	Typ platby při prodeji nebo prolongaci jízdního dokladu. Pro ODIS 0 (nespecifikováno)
ContractPrice	32	INT4	Cena dokladu v jednotkách dle ContractPriceUnit
ContractID	16	INT2	Číslo aktuálního/budoucího kontraktu (v systémech, které číslo kontraktu používají).
PreviousContractID	16	INT2	Číslo předchozího kontraktu (používá se při prolongaci)
LinkToOriginalTicket	4		Ukazuje na doklad, ke kterému byl tento doklad vydán (například ke kterému je tento doklad doplatkem). Lze využít pouze při více dokladech v jednom 2D kódu.
DocumentType	4	ENUM	Typ dokladu. 0: Jízdenka / místenka / „kupón“ MHD 1: Pokuta 2: Změna jízdenky / rozšíření jízdenky 3: Doplatková jízdenka V ODIS vždy 0: Jízdenka
NumberOfTicketsTotal	5	INTEGER(0..31)	Počet koupených jízdenek. Obvykle 1; má smysl u jízdenek, jejichž platnost

			začíná až první validací (ve vozidle, na nástupišti, …).
ContractVehicleClassCodeRestriction	3	ENUM	Povolená vozová třída (v závislosti na dopravním prostředku) 0: bez omezení (nespecifikováno) 1: 1. třída nebo její ekvivalent (Business) 2: 2. třída nebo jejich ekvivalent (Economy/Standard) 3: Premium 4-7: RFU V ODIS nespecifikováno.
TicketNumber	8	INT1	Pořadí jízdenky (pro jednu jízdu může být potřeba více současně platných dokladů – jízdenka + místenka apod.). V ODIS se nepoužívá.
ContractValidityRestrictDay	8	Restrict Days of Week	Omezení platnosti na dny (vhodné např. pro žákovské jízdenky). bity: 0 – 6 = Po až Ne, bit 7 = ‚h' (vizte položku dále). Nastavený bit = doklad platí. Standardně tedy bude vyplněno hodnotou 0x7F (7 bitů)
ContractValidityRestrictCode	8	INT1	Omezení platnosti dle číselníku, uplatňuje se, pokud je nastaven nejvyšší bit ‚h' položky *contractValidityRestrictDays*.
CustomerProfile1	16	INT2	První profil dokladu cestujícího. Tento profil ověřuje aplikační logika při prodlužování.
CustomerTariff1	16	INT1	První tarif cestujícího (doplňující informace v rámci profilu).
ContractPassengers1	8	INT1	Počet cestujících s profilem 1.

ContractTransportMeansRestriction	12	ENUM	Povolené dopravní prostředky. 0 = bez omezení.
SpeciemenFlag	1	FLAG	Příznak, že se jedná o testovací doklad.
ReturnTicketFlag	1	FLAG	Příznak, že se jedná o zpáteční relační doklad (relace je platná pro oba směry; pokud není bit nastaven, je relace platná pouze ve směru Z-Do). Týká se jak jednorázových, tak časových jízdních dokladů.
RFU	2		
ExtendedPassengersFlag	1	FLAG	Pokud je příznak nastaven, první ze segmentů obsahuje rozšířené informace o cestujících (struktura PassengersExtendedInfo)
TicketType	7	ENUM	Udává jak strukturu dat jízdního dokladu, tak počet segmentů, které doklad zabírá nad rámec prvního segmentu. V ODIS připadají v úvahu hodnoty: - 2/18/34/50 pro kilometrický relační tarif - 3/19/35/51 pro zónový tarif - 0 pro celosíťovou jízdenku
RFU	8		
VariantPart	88		Variantní část. Obsah závisí na TicketType. Vybrané hodnoty jsou uvedeny pod tabulkou
Celkem	312 b	39 B	

Dodatečné segmenty mají dynamickou strukturu. Ta závisí na

- nastaveném bitu ExtendedPassengersFlag – pak je obsahem prvního dodatečně alokovaného segmentu datová struktura PassengersExtendedInfo;
- typu struktury dokladu TicketType. Od toho se odvíjí jak počet alokovaných dodatečných segmentů, tak jejich struktura. Ta je popsána v následujících kapitolách.

Datová struktura PassengersExtendedInfo

Název	*Bitů*	*Typ*	*Hodnota / význam*
CustomerProfile2	16	INT2	Typ druhého CP (obvykle sleva)
CustomerTariff2	16	INT1	Rozšiřující informace o druhém CP
ContractPassengers 2	8	INT1	Počet spolucestujících osob s druhým CP
CustomerProfile3	16	INT2	Typ třetího CP (typicky spolucestující věci (zavazadlo, pes))
CustomerTariff3	16	INT1	Rozšiřující informace o typu dokladu spolucestující věci
ContractPassengers 3	8	INT1	Počet spolucestujících osob se třetím CP.
CustomerProfile4	16	INT2	Čtvrtý typ CP dokladu
CustomerTariff4	16	INT1	Rozšiřující informace o čtvrtém CP dokladu
ContractPassengers 4	8	INT1	Počet spolucestujících osob s profilem 4
OverbookingFlag	1	FLAG	Dle UIC 918-3 a UIC 918-2 ANNEX 3a
CorporateFrequent	1	FLAG	Dle UIC 918-3 a UIC 918-2 ANNEX 3a
CustomerFrequent	1	FLAG	Dle UIC 918-3 a UIC 918-2 ANNEX 3a
RFU	37		
CustomerID	64	BCD STRING(8)	Číslo uživatelské karty, ID průkazu či jiná identifikace cestujícího
Celkem	224 b,	28 B	

TicketType

Struktura čísla identifikujícího typ je tvořena dle následujícího pravidla:

b6	*b5*	*b4*	*b3*	*b2*	*b1*	*b0*
X	S	S	T	T	T	T
Rezervováno	Počet dodatečných segmentů		Struktura dat dokladu			

Platí, že pokud informace o směrování zaberou více segmentů, pak identifikace jedné stanice / zóny nemůže „přetéct" mezi více segmenty.

Čísla zastávek (výchozí a cílové stanice) budou uváděna vždy podle číselníku CIS.

TicketType může nabývat následujících hodnot:

ticketType	*Význam*

0	Doklad nemá trasu (síťová jízdenka). Nepotřebuje žádné dodatečné segmenty.
1	Jednoduchý relační doklad Z, Do a volitelně Přes bez nároků na dodatečný segment. Určen pro systémy se zónově-relačním tarifem. Nepočítá s variantou jízdenky pro více PTO na jednom dokladu.
17	Shodně jako T*icketType* = 1 v případě, že je potřeba uložit větší počet nácestných zón. Ty jsou uloženy do jednoho segmentu.
33	Shodně jako T*icketType* = 1 v případě, že je potřeba uložit větší počet nácestných zón. Ty jsou uloženy do dvou segmentů.
49	Shodně jako T*icketType* = 1 v případě, že je potřeba uložit větší počet nácestných zón. Ty jsou uloženy do tří segmentů.
2	Kilometrický doklad Z, Do a volitelně Přes bez nároků na dodatečný segment. Určený pro autobusové a železniční jízdenky. Nepočítá s variantou jízdenky pro více PTO na jednom dokladu.
18	Shodně jako T*icketType* = 2 v případě, že je potřeba uložit větší počet nácestných stanic. Ty jsou uloženy do jednoho.
34	Shodně jako T*icketType* = 2 v případě, že je potřeba uložit větší počet nácestných stanic. Ty jsou uloženy do dvou segmentů.
50	Shodně jako T*icketType* = 2 v případě, že je potřeba uložit větší počet nácestných stanic. Ty jsou uloženy do tří segmentů.
3	Doklad je dán výčtem zón. Je určen typicky pro jedno až třízónové jízdenky systémů integrované dopravy, které určují trasu výčtem zón.
19	Shodně jako TicketType = 3 pro případy, kdy je potřeba uložit větší počet zón. Zabírají 1 dodatečný segment.
35	Shodně jako TicketType = 3 pro případy, kdy je potřeba uložit větší počet zón. Zabírají 2 dodatečné segmenty.
51	Shodně jako TicketType = 3 pro případy, kdy je potřeba uložit větší počet zón. Zabírají 3 dodatečné segmenty.
4	Místenka, bez alokace dodatečného segmentu (zejm. pro potřeby ČD, nyní nedefinováno)
20, 36, 52	Místenka, alokován jeden, dva, resp. tři dodatečné segmenty (nyní nevyužito)
5	Kilometrický doklad Z, Do a Přes bez nároků na dodatečný segment. Určený pro autobusové a železniční jízdenky pro více PTO na jednom dokladu.
21, 37, 53	Shodně jako TicketType = 5, je alokován jeden, dva, resp. 3 dodatečné segmenty.

6	Bezešvá jízdenka (platná pro jednoho nebo více PTO v rámci jednoho dokladu), bez dodatečného segmentu.
22, 38, 54	Shodně jako TicketType = 6, je alokován jeden, dva, resp. 3 dodatečné segmenty.
15, 31, 47, 63	Datový obsah variantní části je zcela v režii PTO (žádný, jeden, dva nebo tři dodatečné segmenty).

TicketType = 2, Kilometrická relační jízdenka

VariantPart má následující strukturu:

Název	***Bitů***	***Typ***	***Hodnota / význam***
ContractJourneyElemSize	5	INTEGER(0..31)	Velikost reprezentace jedné stanice zmenšená o jednu v bitech
ContractJourneyViaCount	5	INTEGER(0..31)	Počet stanic přes
ContractJourneyLength	10	INTEGER(0..1024)	Délka trasy v kilometrech / tarifních jednicích
ContractJourneyFrom	*Dle contractJourneyElemSize, max 32 bitů*		Kód výchozí stanice
ContractJourneyTo	*Dle contractJourneyElemSize, max 32 bitů*		Kód cílové stanice
ContractJourney	*Dle contractJourneyElemSize,*		Výčet nácestných stanic, každá o velikosti dle *contractJourneyElemSize*.
Celkem	88 b,	11 B	

Směrování se musí vejít do 78 bitů. Pro 24 bitové kódy (např. železničních) stanic postačuje pro uložení výchozí, cílové a jedné nácestné stanice. Pokud se směrování do této struktury nevejde, použije se TicketType 18, 34 nebo 50 (podle počtu nácestných stanic) a je alokován jeden až tři dodatečné segmenty. Platí, že nácestné stanice jsou v tomto segmentu uloženy všechny (není potřeba dodatečný segment), nebo žádná (a všechny jsou v dodatečných segmentech).

Při ticketType = 1 není doplňující segment alokován.

TicketType = 18 / 34 / 50, Kilometrická relační jízdenka

VariantPart má následující strukturu:

Název	*Bitů*	*Typ*	*Hodnota / význam*
ContractJourneyElemSize	5	INTEGER(0..31)	Velikost reprezentace jedné stanice zmenšená o jednu v bitech
ContractJourneyViaCount	5	INTEGER(0..7)	Počet stanic přes.
ContractJourneyLength	10	INTEGER(0..1024)	Délka trasy v kilometrech / tarifních jednicích
ContractJourneyFrom	*Dle contractJourneyElemSize*		Kód výchozí stanice
ContractJourneyTo	*Dle contractJourneyElemSize*		Kód cílové stanice
Celkem	88 b,	11 B	

Každý alokovaný dodatečný segment má strukturu:

Název	*Bitů*	*Typ*	*Hodnota / význam*
ContractJourney	224	OCTET STRING(28)	Pole nácestných stanic. Velikost reprezentace jedné dle *contractJourneyElemSize*,tedy nejvíce 7 stanic při contractJourneyElemSize = 31
Celkem	224 b,	28 B	

TicketType = 3, Jednoduchá zónová jízdenka

VariantPart má následující strukturu:

Název	*Bitů*	*Typ*	*Hodnota / význam*
ContractJourneyElemSize	5	INTEGER(0..31)	Velikost reprezentace jedné zóny zmenšená o jednu v bitech.
ContractJourneyViaCount	5	INTEGER(0..31)	Počet zón
ContractJourneyZones	78		Výčet zón
Celkem	88 b,	11 B	

U tohoto typu jízdenky platí všechno nebo nic, buď se tedy všechny zóny vejdou do 78 bitů a není potřeba dodatečný segment (tj. dodatečný segment není alokován), nebo je tento blok volný a zóny jsou pouze v dodatečném segmentu/segmentech.

TicketType = 19 / 35 / 51, Zónová jízdenka s dodatečnými segmenty

VariantPart má následující strukturu:

Název	*Bitů*	*Typ*	*Hodnota / význam*
ContractJourneyElemSize	5	INTEGER(0..31)	Velikost reprezentace jedné zóny v bitech zmenšená o jednu.
ContractJourneyViaCount	5	INTEGER(0..31)	Počet zón
RFU	78		
Celkem	88 b,	11 B	

Je alokován příslušný počet dodatečných segmentů, každý se strukturou:

Název	*Bitů*	*Typ*	*Hodnota / význam*
ContractJourneyZones	224	OCTET STRING(28)	Seznam zón
Celkem	224 b,	28 B	

REC_TICKET_SELL

Struktura obsahuje informace o prodeji dokladu.

Název	*Bitů*	*Typ*	*Hodnota / význam*
ActionDate	14	Date Stamp	Datum prodeje nebo prolongace dokladu.
ActionTime	11	Time Stamp	Čas prodeje nebo prolongace dokladu.
RFU	7		RFU + zarovnání na celé B
ActionProvider	24	Provider ID	Kód dopravce/výdejce, který provedl poslední prodejní operaci
ContractTransaction	64	OCTET STRING(8)	Jedinečný identifikátor prodejní transakce v systému výdejce.
Celkem	120 b	15 B	

1.3 Ověření bezpečnostního proužku a časového razítka na odbavovacím zařízení

Odbavovací zjistí ze serveru (viz kapitola 3) aktuální serverový čas T a zobrazí na displeji dva grafické prvky: pro čas T – 15 sekund a pro čas T + 15 sekund. Obsluha považuje údaje na displeji mobilního telefonu za platné, pokud se grafický prvek v mobilním telefonu shoduje s jedním z grafických prvků zobrazených v odbavovacím zařízení.

1.1.1 Výpočet barev grafického prvku

Odbavovací zařízení získá dvojici barev pro grafický prvek pro daný vstupní čas následujícím postupem:

1. Odbavovací zařízení stáhne ze serveru nebo načte z lokálního úložiště dříve stažené tajné hodnoty SC0, SC1, SC2, SC3 pro vstupní serverový čas (viz kapitola 2.2) posunutý o 15 vteřin do minulosti nebo do budoucnosti.
2. Vypočítá X = posunutý serverový čas v milisekundách od půlnoci 1. ledna 1970 GMT.

3. Vypočítá T = X / **30000**. Číslo T dále interpretuje jako 4 byty T0, T1, T2 a T3. T0 značí nejnižší byte a T3 nejvyšší byte čísla T ve formátu big endian.
4. Vypočítá byty G1, B1, R2, G2 jako:
    a. G1 = ((T0 XOR T1 XOR T2 XOR SC0) krát LC0) & 0xFF
    b. B1 = ((T0 XOR T2 XOR T3 XOR SC1) krát LC1) & 0xFF
    c. R2 = ((T0 XOR T1 XOR T2 XOR T3 XOR SC2) krát LC2) & 0xFF
    d. G2 = ((T0 XOR T1 XOR T3 XOR SC3) krát LC3) & 0xFF,

    kde hodnoty LC0, LC1, LC2 a LC3 jsou tajné konstanty bezpečně předané dodavateli odbavovacího zařízení.
5. Získá první barvu definovanou pomocí RGB, kde R = 0, G = G1, B = B1.
6. Získá druhou barvu definovanou pomocí RGB, kde R = R2, G = G2, B = 0.
7. Zobrazí obě barvy na displeji, první barvu vlevo, druhou barvu vpravo.

1.1.1.1 Příklad kódu

Následující kód zachycuje výše uvedený postup v jazyce Java:

```
Date serverTime = getServerTime(); //ziskani serveroveho casu
long X = serverTime.getTime();
int T = (int) (X / 30000);

byte[] sc = getServerValues(serverTime); //ziskani tajných hodnot ze serveru pro dany cas
int[] lc = getLocalConstants(); //nacteni tajnych konstant

int firstGreen = ((xorAll((byte) T, (byte) (T >> 8), (byte) (T >> 16), sc[0]) * lc[0])& 0xFF);
int firstBlue = ((xorAll((byte) T, (byte) (T >> 16), (byte) (T >> 24), sc[1]) * lc[1]) &
0xFF);
int secondRed = ((xorAll((byte) T, (byte) (T >> 8), (byte) (T >> 16), (byte) (ltInt >> 24),
sc[2]) * lc[2]) & 0xFF);
int secondGreen = ((xorAll((byte) T, (byte) (T >> 8), (byte) (T >> 24), sc[3]) * lc[3]) &
0xFF);

Integer firstColor = Color.argb(255, 0, firstGreen, firstBlue);
Integer secondColor = Color.argb(255, secondRed, secondGreen, 0);

return new Pair<>(firstColor, secondColor);
```

1.1.2 Výpočet alfanumerických znaků grafického prvku a časového razítka QR

Pro dvojici barev (a jejich složky G1, B1, R2, G2 získané postupem popsaným v předchozí kapitole) získá odbavovací zařízení alfanumerické znaky následujícím postupem:

1. Spočítá hash SHA-512 ze vstupního pole [12, 119, 3, 6, 255, 0, G1, B1, 255, R2, G2, 0]
2. První dva bajty výsledného hashe zobrazí jako 4 hexadecimální znaky (0-9A-F) na displeji.

Poznámka: Tytéž dva bajty jsou zároveň uloženy v QR kódu v položce **„Datová část specifická pro zákazníka"**. Zařízení po načtení QR kódu ověří, zda uložené bajty odpovídají bajtům získaným popisovaným postupem.

1.2 Příklad výpočtu:

Vstupní hodnoty:
X = 1556541913447

SC0 = 5
SC1 = 27
SC2 = 12
SC3 = 19
LC0 = 101
LC1 = 57
LC2 = 67
LC3 = 31

Výstupy:

G1 = **66**
B1 = **77**
R2 = **48**
G2 = **232**

Alfanumerický řetězec: **EE 93**

1.4 Příklad jízdenky ODIS

2. Komunikace se serverem

2.1 Aktuální serverový čas

MapPhoneServer sděluje aktuální serverový čas pomocí webové služby:

GET https://mphs.kodis.cz/mapphoneserverodis/MapPhoneServerWS.svc/time

Odpověď

200 OK

```
{
    "Status": "OK",
    "Timestamp": "20190222083428"
}
```

kde Timestamp značí aktuální čas serveru ve formátu „YYYYMMDDHHmmss".

Odbavovací zařízení musí pravidelně zjišťovat a ukládat odchylku svého systémového času od serverového času MapPhoneServeru tak, aby bylo následně schopno zobrazit grafický prvek i v případě nedostupného spojení se serverem.

2.2 Tajné hodnoty

MapPhoneServer sděluje tajné hodnoty pro grafický prvek na základě předaného uživatelského jména a hesla:

POST https://mphs.kodis.cz/mapphoneserverodis/MapPhoneServerWS.svc/getVisualInspectionKeys

```
{
        "User":"prihlasovaci_jmeno",
        "Password":"heslo",
        "BeginDateTime":"20180816091300",
        "EndDateTime":"20180823091300"
}
```

Kde:

- User … přihlašovací jméno
- Password … heslo
- BeginDateTime … počátek intervalu, pro který mají být navráceny tajné hodnoty, ve formátu „YYYYMMDDHHmmss"
- EndDateTime … konec intervalu, pro který mají být navráceny tajné hodnoty, ve formátu „YYYYMMDDHHmmss". EndDateTime nesmí být větší než aktuální serverový čas navýšený o hodnotu stanovenou konfigurací serveru (zpravidla 15 dnů).

Odpověď

200 OK


```
{
   "Status": "OK",
   "VIS": [
      {
         "ID": "896",
         "S": "BHCrZg==",
         "ValidFrom": "201808160900",
         "ValidTo": "201808161000"
      },
      {
         "ID": "897",
         "S": "OOrGaw==",
         "ValidFrom": "201808161000",
         "ValidTo": "201808161100"
      },
      …
 ]
}
```


Sady tajných hodnot jsou v odpovědi uvedeny v poli VIS. Každý záznam v poli VIS obsahuje informace:

- ID … pořadové číslo jedné sady tajných hodnot
- S … 4 bajty SC0 až SC3 kódované jako Base64
- ValidFrom … počátek platnosti sady ve formátu YYYYMMDDHHmmss (včetně)
- ValidTo … konec platnosti sady ve formátu YYYYMMDDHHmmss

Odbavovací zařízení musí zajistit pravidelné stahování tajných hodnot tak, aby bylo následně schopno zobrazit grafický prvek i v případě nedostupného spojení se serverem.