Examen - Attaques sur carte à puce 2024-2025

Durée : 54 minutes

6 points

Nicolas Debande
nicolas.debande@idemia.com

Exercice 1. Attaques par canaux auxiliaires (side-channel).

1. Décrire le principe d'une attaque DPA mono-bit. [1 pt]
2. Quels facteurs influent sur la réussite d'une telle attaque ? [0.5 pt]
3. Citer quatre contre-mesures pour se défendre contre cette attaque. Pour chacune d'entre elles, expliquer en quoi la contre-mesure protège. [1 pt]
4. Est-il possible de se protéger complètement contre les attaques par side-channel ? Justifier [0.5 pt]

Exercice 2. Attaques par faute sur AES .

L'algorithme 1 décrit l'algorithme d'AES.

1. D'une manière générale, comment fonctionne une attaque par injection de fautes ? [0.5 pt]
2. Expliquer l'attaque DFA sur AES. [1.5 pt]
3. Un AES protégé contre les side-channel par une technique de masquage est-elle sensible à la DFA ? Justifier. [0.5 pt]
4. Citer une contre-mesure qui permet spécifiquement de contrer cette attaque. Justifier. [0.5 pt]

Algorithme 1 : AES

Data : M
Result : C

```
1  C = AddRoundKey(M, K_0);
2  for i ← 1 to 9 do
3  |   C = SubBytes(C);
4  |   C = ShiftRows(C);
5  |   C = MixColumns(C);
6  |   C = AddRoundKey(C, K_i);
7  end
8  C = SubBytes(C);
9  C = ShiftRows(C);
10 C = AddRoundKey(C, K_10);
11 return C;
```

Partie I.Tobor

10 points

Décembre 2024

1 Questions de cours (2 points)

Reliez/associez les concepts/termes de la colonne gauche avec ceux de la colonne droite. Justifiez rapidement vos choix.

Concept 1	Concept 2
Cold Reset	Java Card
GlobalPlatform	ATR
GPO	APDU
BAC	ATC
T=1	Certificat
PKI	PACE

2 EMVCo et authentification (3 points)

Pour concevoir les spécifications de la carte bancaire EMVCo est parti de quelques exigences de base. Le système entier (carte, utilisateur, terminal de paiement, réseau informatique bancaire, etc) doit assurer, entre autres, des fonctionnalités suivantes:

1. Déléguer à la carte une possibilité de decision définitive ou partielle de valider la transaction (sans être connecté aux serveurs bancaires)
2. Identifier et authentifier le propriétaire de la carte (par la carte)
3. Vérifier l'authenticité de la carte (par le terminal de paiement)
4. Fabriquer une preuve unique et infalsifiable de la transaction passée (par la carte et à destination de tous les acteurs concernés)
5. Authentifier (par la carte) la banque emmetrice dans le cas où le terminal est obligé de confirmer la transaction directement avec le système banquaire

Sachant que le protocole complet et les commandes échangées sont les suivantes:

- `SELECT`
- `GET PROCESSING OPTIONS`
- plusieurs `READ BINARY` et `GET DATA`
- `VERIFY PIN`
- `INTERNAL AUTHENTICATE`
- `GENERATE AUTHENTICATION CRYPTOGRAM 1`
- `EXTERNAL AUTHENTICATE`
- `GENERATE AUTHENTICATION CRYPTOGRAM 2`

Expliquez quel propriété/fonctionnalité est assuré par quelle commande et résumez le principe.

3 Problème: "Digital Tachograph" (5 points)

3.1 Contexte

"Digital Tachograph" est une série de spécifications d'Union Européenne (*Regulation EU 2016/799*) pour le transport routier professionnel visant à remplacer et à compléter les fonctionnalités du Chronotachygraphe: le disque papier dans les camions et/ou les bus qui servait à enregistrer et suivre l'activité de chauffeur (vitesse, temps de repos, etc). Cet exercice propose d'analyser la gestion de certificats et de la structure PKI (Public Key Infrastructure) utilisées par le tachygraphe moderne et plus sécurisé.

Pour l'information: Pour les besoins de cet exercice les spécifications et les exemples viennent directement de la *Regulation EU 2016/799* et de données officielles de campagne standard de tests de conformité de Digital Tachograph après qqs simplifications et adaptations.

L'énoncé est long et détaillé pour fournir toutes les informations nécessaires, mais les réponses attendues sont plutôt très courtes !

3.2 Informations et rappel de spécifications

3.2.1 Tachograph

Le Digital Tachograph définit plusieurs dispositifs qui doivent communiquer et collaborer ensemble. Parmi ces éléments on retrouve:

- **VU (Vehicule Unit)**: un ordinateur de bord de camion ou bus qui centralise les informations de différents capteurs de véhicule (vitesse, position GPS, heure, intervention d'atelier, passage des portique de péage), etc et d'autres événements (contrôles routiers, etc). Il est équipé des dispositifs d'entrée / sortie (clavier et/ou qqs touches de contrôle, écran, mini imprimante) et il contient aussi deux lecteurs de cartes à puce qui servent à communiquer avec les éléments suivants ci-dessus :
- 4 types de carte à puce:
  - **Driver** : Identifie/représente le chauffeur. Stocke ses informations personnelles (nom, numéro de permis de conduire) les données relatives à l'activité de conduite (par exemple les temps de conduite, des intervalles de repos obligatoire).
  - **Company** : Identifie la société du transport. Autorise VU à lire, afficher et imprimer les données stockées dans VU et sur les cartes "Driver" et "Workshop".
  - **Workshop** : Identifie l'atelier de maintenance. Stocke, entre autres, les informations relatives de calibration des capteurs.
  - **Control** : Identifie l'autorité de contrôle (police, gendarmerie, douane, ...). Autorise VU à lire, afficher et imprimer les données stockées dans VU est sur d'autres cartes.
- D'autres dispositifs/elements/périphériques non mentionnés ici.

Exemple d'utilisation:

- Pendant la conduite le chauffeur doit mettre sa carte dans VU. De différents paramètres provenant de capteurs du camion (vitesse, temps de pause, etc) y sont enregistrés et horodatés pendant la conduite.
- Lors du contrôle routier, la police met leur carte dans le second lecteur de cartes. Après avoir identifié et authentifié cette carte, le VU peut être utilisé pour afficher et imprimer les données de la carte du chauffeur et d'autres informations relatives au véhicule (par exemple les données de la dernière révision effectuée par l'atelier de maintenance).

Pour éviter la fraude (une modification de données par des personnes non autorisées) et protéger la vie privée (accès aux données personnelles) les dispositifs doivent s'authentifier mutuellement les uns par rapport aux autres (par exemple le VU versus les cartes). Chaque cycle d'échanges commence par la procédure "d'appairage" entre les dispositifs qui consiste en échange et verification des certificats. Une authentification mutuelle basée sur la cryptographie asymétrique est ensuite utilisée

pour les autorisations d'accès et/ou pour la génération de clés temporelles utilisées pour chiffrer la communication.

Le déploiement et la gestion de parc est assurée pas les États Membres d'UE, mais garantie numériquement par UE via le "certificat racine". Bien sûr le dispositif complet doit être interopérable entre les États Membres, par exemple un policier allemand doit pouvoir contrôler un chauffeur slovaque qui travaille pour la société italienne en conduisant le camion immatriculé en Portugal.

3.2.2 PKI et Certificates

Le choix de la cryptographie asymétrique est évident: Vu le nombre de dispositifs qui doivent être interoperable sans être interconnecté ou connecté à un système centralisé, chaque élément du système possède ses propres clés cryptographiques privées et publiques et le PKI garantit leur cohérence et leur authenticité. La racine de ce PKI, le "point de confiance" commun est l'Union Européenne.

Un certificat est un ensemble de données signé par une Autorité de Certification (CA). Ces données peuvent être n'importe quoi, mais le cas particulier est un élément de la chaîne des certificats où un certificat contient un nouveau identifiant et une nouvelle clé publique qui sert comme la "délégation" de privilèges de CA. En résumé on y retrouve :

- L'identifiant de "propriétaire" de certificat (CHR) et de l'autorité qui l'a signé (CAR),
- La clé publique de "propriétaire" (celle qui est certifée),
- Les droits et les autorisations particulières (CHA) que CA a donné/autorisé/délégué à CHR,
- Les dates du début de validité et d'expiration.

Le garant commun du système "Digital Tachograph" est donc UE à travers l'organisme **ERCA** "European Root CA". Il est identifié par un certain `ERCA.ID` et possède la clé privée `ERCA.SK` et la clé publique `ERCA.PK`.

Les garants nationaux sont authentifiés par des certificats `MSCA.xx.CERT` délivrés par des **MSCA** "Member State CA" (un certificat par le Pays Membre d'UE). Ils sont signés par des clés privées de **ERCA**. Ils forment le 2eme niveau de la chaîne de certificats. Chaque Pays Membre possède aussi sa paire de clés `MSCA.xx.SK` et `MSCA.xx.PK`

Les certificats associés à des équipements immatriculés dans chacun de pays membres d'UE sont, quant à eux, signés par des clés privées de **MSCA** appropriées. Ils forment donc le 3eme niveau de la chaîne de certificats.

Tous les dispositifs DIS (VU et les cartes) possèdent `DIS.ID` et `DIS.PK` publiques et `DIS.SK` qui est gardé secret et privée. Ils connaissent obligatoirement:

- `ERCA.ID` et `ERCA.PK` (ces données sont publiques mais comme le système peut être utilisé sans l'accès à l'internet, elles sont stockées localement dans chaque VU et chaque carte)

Ils connaissent aussi et ils peuvent fournir à la demande:

- `MSCA.xx.CERT` le certificat de "son" Pays Membre qui atteste `DIS`
- `DIS.CERT` qui garantit `DIS.PK`

`ERCA.PK` a une durée de vie limitée de 17 ans. Pour pouvoir le modifier, **ERCA** fournit, quand c'est nécessaire et dans la période de validité transitoire, un certificat spécial (*link certificate*) avec la nouvelle PK et auto signé.

Les certificats contiennent également les dates de validité et d'expiration qui doivent être cohérentes (incluses dedans) avec les dates de certificat "père".

Un certificat au format *CVCert* (Card Verifiable Certificate) contient les champs suivants regroupés dans une structure TLV hiérarchique:

Tag				Longueur	Description
7f21				*var*	CC - Cardholder Certificate
	7f4e			*var*	CB - Certificate Body - les données dont on certifie l'authenticité
		5f29		1	CPI - Certificate Profile Identifier - version du format - doit être 01
		42		*var*	CAR - Certification Authority Reference - identifiant de l'autorité de certification
		5f4c		7	CHA - Certificate Holder Authorization - droit d'accès (très simplifié, doit être `ff534d524454xx`)
		7f49		*var*	PK - Public Key
			06	*var*	DP - Domain Parameter - identifiant de la courbe elliptique standard
			86	*var*	PP - Public Point - clé publique (point sur la courbe)
		5f20		*var*	CHR - Certificate Holder Reference - identifiant du propriétaire de certificat
		5f25		4	CED - Certificate Effective Date
		5f24		4	CEX - Certificate Expiration Date
	5f37			*var*	SIG - Certificate Signature

Public Point: La clé publique ECC est codée en forme spéciale "point non compressée" i.e. formatée avec: `04 | valeur de la coordonnée X | valeur d'Y`

Par exemple: une clé de 256 bits == 32 (0x20) octets, le codage complet TLV correspondant sera donc `86 41 04 XX XX XX ... XX YY YY ... YY`

DP: Courbes elliptiques utilisables dans Digital Tachograph: (le nombre dans le nom de la courbe indique la taille de module en bits)

Courbe	Codage de Domain Parameter
NIST P-256 (secp256r1)	`2a 86 48 ce 3d 03 01 07`
NIST P-384 (secp384r1)	`2b 81 04 00 22`
brainpoolP256t1	`09 2b 24 03 03 02 08`
brainpoolP384t1	`09 2b 24 03 03 02 0c`
brainpoolP512t1	`09 2b 24 03 03 02 0e`

Dates: Sur 4 octets. Pour les besoin de cet exercice on va simplifier le format de la date et on va considérer qu'il est `YY YY MM DD` en BCD.

CHA: en forme `ff 53 4d 52 44 54 xx`. Composé de 6 octets fixes et le 7eme définissant le type de l'équipement pour lequel ce certificat est désigné.

Code	01	02	03	04	05	06	...	09	...
Type	Driver Card	Workshop Card	Control Card	Company Card	Motion Sensor	Vehicle Unit	...	Communication Module	..

CAR et CHR Dans le cas de certificats **ERCA** et **MSCA** les octets 2 à 4 codent le pays membre (3 lettres ASCII), propriétaire de certificat. Signification d'autres octets n'est pas spécifiée et elle est laissée au libre choix de CA.

Il n'y a pas de dispositions spécifiques pour le nommage de certificats de niveau 3.

3.3 Autres utilitaires

Table ASCII

	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
...																
20		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z					

3.4 Certificats

On dispose des certificats suivants:

(1)

```
7f 21 81 a9 7f 4e 81 82  5f 29 01 01 42 07 5f 41  52 43 2e 30 33 5f 4c 07
ff 53 4d 52 44 54 03 7f  49 4d 06 08 2a 86 48 ce  3d 03 01 07 86 41 04 13
98 d2 59 67 f9 b2 ea 0d  d7 d6 32 0d 3f 35 c0 3a  18 cb 1b a8 35 b5 25 60
ee e4 bd 05 a8 f7 d8 85  6a ce b4 0a 80 9c ac f5  c8 c1 cb d7 82 8d 56 3d
a8 e4 77 08 c6 1c 0f d1  62 ad 39 d1 98 b1 59 5f  20 0a 21 41 52 43 2e 31
32 2e 32 33 5f 25 04 20  12 05 01 5f 24 04 20 13  04 30 5f 37 20 02 fd 9c
9f 71 10 df 2f b1 ab 55  1c 04 76 b6 d4 80 ce 1f  4d 7a 23 1f 0d 01 3f 5a
9a aa 3f 58 3c
```

(2)

```
7f 21 81 82 7f 4e 5c 5f  29 01 01 42 06 5f 45 55  20 2e 31 5f 4c 07 ff 53
4d 52 44 54 00 7f 49 2c  06 08 2a 86 48 ce 3d 03  01 07 86 20 d3 77 70 4b
b8 80 15 dd 75 ab 2d f4  74 df cb 22 34 2c 50 da  64 fc b2 8c 58 ad c9 d8
48 5f 2d bc 5f 20 06 5f  45 55 20 2e 31 5f 25 04  20 04 01 01 5f 24 04 20
20 01 31 5f 37 20 f0 35  fb fa 83 39 d1 07 06 8c  81 aa 45 28 d8 07 14 00
19 db be 37 15 0b 0b 7b  ab 61 2e ce 83 d3
```

(3)

```
7f 21 81 93 7f 4e 6d 5f  29 01 00 7f 49 51 06 09  04 00 7f a4 00 00 01 02
01 81 10 3b 09 fd b4 18  ad be 18 ed 61 43 31 c7  d9 1d 3f 82 10 59 d9 3a
17 84 41 35 ac af 39 e3  c1 b0 53 c1 35 86 20 d6  68 00 f7 8e ef 88 6a 19
e2 5f 2c 62 55 86 b5 14  7d 7d 0c e0 fb d8 bd 86  bb 45 0a 6c b8 7a 27 5f
20 12 46 52 5f 54 45 53  54 5f 43 56 43 41 5f 30  30 30 30 36 5f 37 20 43
3c a3 6f 2d ae d3 aa 63  22 73 4e 71 e4 be 57 9a  e1 26 d1 48 50 b1 c8 87
3f 7b 8f aa d7 1a b0
```

(4)

```
7f 21 81 a5 7f 4e 7f 5f  29 01 01 42 06 5f 45 55  20 2e 31 5f 4c 07 ff 53
4d 52 44 54 00 7f 49 4d  06 08 2a 86 48 ce 3d 03  01 07 86 41 04 e8 be e2
0a d5 09 f3 e4 93 38 64  4f d6 c5 64 f0 84 f1 9b  d9 b4 ac b4 20 3b 67 cf
3c 96 9d a3 e9 da de 33  99 cf 9b c2 79 33 a6 1c  3a ca 2c a6 e1 cf 63 b8
```

```
2f 97 01 2e 1b b7 8a 05  40 41 55 aa 47 5f 20 08  2b 55 54 4f 2e 30 30 35
5f 25 04 20 10 01 01 5f  24 04 20 14 01 01 5f 37  20 16 9c 0b 50 ef d7 c1
18 40 54 d7 cf a6 26 56  9f b2 73 be b7 7b f2 33  69 c3 dd ec 47 96 eb 80
91
```

(5)

```
7f 21 81 c4 7f 4e 81 9d  5f 29 01 01 42 06 5f 45  55 20 2e 31 5f 4c 07 ff
53 4d 52 44 54 00 7f 49  6c 06 07 09 2b 24 03 03  02 0c 86 61 04 11 ca cd
5c 26 97 2e 1e 5b 48 8c  1c d8 d9 27 92 7a 94 6f  40 c1 d5 71 fa 78 81 5f
bf e7 b6 b4 e6 93 53 3a  55 90 60 a2 b7 f7 36 30  76 c3 8f 9b 40 59 d9 3a
17 84 41 35 ac af 39 e3  c1 b0 53 c1 35 3b 09 fd  b4 18 ad be 18 ed 61 43
31 c7 d9 1d 3f 8d 35 6f  b1 be f3 d4 4a 8f f3 0d  a7 b4 dc 2e c8 5f 20 07
5f 41 52 43 2e 30 33 5f  25 04 20 10 01 01 5f 24  04 20 14 01 01 5f 37 20
d6 68 00 f7 8e ef 88 6a  19 e2 5f 2c 62 55 86 b5  14 7d 7d 0c e0 fb d8 bd
86 bb 45 0a 6c b8 7a 27
```

(6)

```
7f 21 81 a9 7f 4e 81 82  5f 29 01 01 42 07 2b 55  54 4f 2e 30 35 5f 4c 07
ff 53 4d 52 44 54 06 7f  49 4d 06 08 2a 86 48 ce  3d 03 01 07 86 41 04 43
3c a3 6f 2d ae d3 aa 63  22 73 4e 71 e4 be 57 9a  e1 26 d1 48 50 b1 c8 87
3f 7b 8f aa d7 1a b0 ab  81 ef 66 db 3a 06 d0 82  a1 d4 80 e8 91 d5 02 16
22 9a bf 03 61 c0 7a 1b  eb 97 37 24 bd 98 94 5f  20 0a 44 45 56 49 43 45
2e 38 2e 33 5f 25 04 20  12 05 01 5f 24 04 20 13  04 30 5f 37 20 02 31 36
d2 67 b6 38 8b a8 d6 cf  07 db a0 0c f3 d8 f4 a9  d8 f0 8f 1c 3b 5f 12 c1
44 ce ed 12 6c
```

(7)

```
7f 21 81 aa 7f 4e 81 83  5f 29 01 01 42 07 2b 55  54 4f 2e 30 35 5f 4c 07
ff 53 4d 52 44 54 01 7f  49 4d 06 08 2a 86 48 ce  3d 03 01 07 86 41 04 2e
d1 d6 91 9a 81 18 09 c6  06 8f a7 de 6c cd cf 2e  05 91 e4 80 94 04 2c 73
a6 09 f2 67 51 53 0b e5  87 6e f9 1f 63 23 c4 ae  fc 48 39 47 ef 63 c6 dc
5b 1d 36 ad b3 e7 3f e7  4b 80 69 af 75 52 d5 5f  20 0b 44 45 56 49 43 45
2e 31 2e 31 31 5f 25 04  20 12 05 01 5f 24 04 20  13 04 30 5f 37 20 4f 2a
1f e5 c2 85 35 a4 74 fc  22 fb 02 67 cf a9 62 dc  a7 b0 ec da ab 5b d8 71
09 33 61 66 b3 57
```

3.5 Questions

1. Quel(s) est (sont) le(s) certificat(s) **ERCA** ?
2. Quel(s) est (sont) le(s) certificat(s) **MSCA** ?
3. Le jeu de tests standards d'interopérabilité utilise deux pays imaginaires. Quels sont ces deux pays (leur codes sur 3 lettres) ?
4. Peut-on dire si la carte **Driver** et le **VU** (les deux identifiés par des certificats ci-dessus) viennent du même pays ? Comment ?
5. La signature d'un des certificats est incorrecte. (On peut le savoir sans la vérifier.) Quel certificat est concerné ?
6. L'appairage initial entre le **VU** (identifié par un certificat approprié ci-dessus) et la carte **Control** (identifiée par un des certificats ci-dessus) est plus longue que entre **VU** et la carte **Driver** (idem). Pourquoi ?

Numéro d'anonymat :

Partie J. Lancia

4 points
Répondre sur le sujet !

Questions de cours

plusieurs réponses possibles pour chaque question
chaque mauvaise réponse retranche des dixièmes de points

1. La machine virtuelle Java Card est …

☐ Une machine à pile
☐ Un programme Java
☐ Un processeur simulé
☐ Un système d'exploitation

2. Cycle de vie d'une applet

☐ Les applets java sont compilées vers un fichier class
☐ Le fichier exp permet d'exporter des fonctions
☐ Le fichier jca est indispensable pour charger une applet
☐ Le fichier cap contient le code des méthodes de l'applet

3. Une applet peut être est chargée …

☐ Dans un Security Domain
☐ Sans authentification cryptographique
☐ Sans être vérifiée par le BCV
☐ Sous forme de fichier jar

4. Quel mécanisme intégré dans la carte permet d'assurer qu'une applet a été validée ?

☐ Le DAP
☐ La vérification de Token
☐ Le BCV
☐ Le firewall

5. L'interface Shareable de l'API Java Card…

☐ Permet le contournement du firewall
☐ Déclare des méthodes partagées
☐ Permet le partage d'objet
☐ Permet les attaques en stack overflow

6. Allocation mémoire

☐ Les objets sont alloués en mémoire persistante
☐ Tous les tableaux sont alloués en mémoire persistante
☐ Les variables locales sont allouées en mémoire transiente
☐ Toutes les références sont allouées en mémoire persistante

7. Pile et frame

☐ Les locales sont stockées dans la pile d'opérande
☐ La frame a une taille constante pour une méthode données
☐ Les arguments d'une fonction sont stockés dans les locales
☐ La pile est systématiquement typée

8. Les écritures mémoires dans une transaction

☐ Sont toutes systématiquement réalisées
☐ Sont réalisées dans leur intégralité ou pas du tout
☐ Peuvent être annulées
☐ Provoquent des débits et des crédits

9. Le contexte d'exécution

☐ Détermine les droits d'accès aux objets
☐ Ne change jamais pendant l'exécution d'une applet
☐ Détermine le possesseur d'un objet lors de sa création
☐ Est identique pour chaque package

10. Les bytecodes

☐ Sont typés
☐ Sont interprétés par le micro-processeur
☐ Agissent sur la pile d'opérande
☐ Agissent sur les locales

Questions pratiques

```
.method private method1(Ljava/lang/Object;)S {
    .stack 1;
    .locals 1;

    .descriptor   Ljava/lang/Object;      1.0;

        L0: sconst_0;
            sstore_2;
            aload_1;
            sstore_2;
            sload_2;
            sreturn;
}

.method private method2()V {
    .stack 1;
    .locals 4;

        L0: sspush 4369;
            sstore_1;
            sspush 8738;
            sstore_2;
            sspush 13107;
            sstore_3;
            sspush 17476;
            sstore 4;
            sspush 4369;
            sspush 4369;
            sstore_1;
            return;
}
```

```
.method private method3()V {
    .stack 1;
    .locals 4;

        L0: sspush 4369;
            sstore_1;
            sspush 8738;
            sstore_2;
            sspush 13107;
            sstore_3;
            sspush 17476;
            sstore 4;
            pop;
            return;
}
```

1. Quelle méthode réalise une attaque de type stack overflow

☐ method1
☐ method2
☐ method3

2. Quelle méthode réalise une attaque de type stack underflow

☐ method1
☐ method2
☐ method3

3. Quelle méthode réalise une attaque de type confusion de type

☐ method1
☐ method2
☐ method3

4. Supprimez une instruction dans la méthode 4 pour produire une confusion de type (barrez l'instruction).

```
.method private method4(Ljava/lang/Object;)[S {
    .stack 1;
    .locals 0;

    .descriptor     Ljava/lang/Object;      1.0;

    L0: sspush 4369;
        sstore_1;
        sspush 8738;
        sstore_2;
        sspush 13107;
        sstore_3;
        sspush 17476;
        sstore 4;
        aload_1;
        checkcast 12 0;
        areturn;
}
```