be.ugent.informationsecurity

server

BillingModule

+processBill(Bill): void

BookingModule

-bookings: List<Booking>

-book(int, int, Date, Date): Booking
+book(byte[], byte[]): Booking

CarCommunicationModule

-carUIDs: int[]
-cars: Map<Integer, Car>
-carKeys: Map<Integer, SecretKey>
-carMessageKeys: Map<Integer, SecretKey>

+getCars(): Map<Integer, Car>
+sendBookingInfo(Booking): void
-sendAESKey(Car): SecretKey
+receiveBillingInfo(int, byte[]): void
+receiveAESKey(byte[], byte[]): boolean

Server

-server: Server
+users: Map<Integer, PublicKey>
-privateKey: PrivateKey
-publicKey: PublicKey
-bm: BookingModule
-cm: CarCommunicationModule
-billingModule: BillingModule

+getInstance(): Server
+getPublicKey(): PublicKey
+getPrivateKey(): PrivateKey
+getUserPublicKey(String): PublicKey
+getBookingModule(): BookingModule
+getCarCommunicationModule(): CarCommunicationModule
+getBillingModule(): BillingModule
+register(byte[], byte[]): String
+requestUID(byte[]): byte[]
+executeAESKeyExchange(byte[], byte[]): void

models

Bill

-appUid: int
-bookingId: int
-km: double
-amount: double
-startDate: Date
-endDate: Date

+fromJSON(String): Bill
+toJSON(): String
+getAppUid(): int
+setAppUid(int): void
+getBookingId(): int
+setBookingId(int): void
+getKm(): double
+setKm(double): void
+getAmount(): double
+setAmount(double): void
+getStartDate(): Date
+setStartDate(Date): void
+getEndDate(): Date
+setEndDate(Date): void

Booking

-bookingUid: int
-uid: int
-start: Date
-stop: Date
-appUid: int
-pubAppKey: PublicKey
-pubAppKeyBytes: String
-car: Car

+fromJSON(String): Booking
+toJSON(): String
+getUid(): int
+setUid(): void
+getStart(): Date
+setStart(Date): void
+getStop(): Date
+setStop(Date): void
+getCar(): Car
+setCar(Car): void
+getAppPubKey(): PublicKey
+setAppPubKey(PublicKey): void
+setAppPubKeyFromJSON(): void
+getAppUid(): int
+setAppUid(int): void

utils

AES

+generateKey(): SecretKey
+encrypt(byte[], byte[]}: byte[]
+encrypt(byte[], SecretKey): byte[]
+decrypt(byte[], byte[]): String
+decrypt(byte[], SecretKey): String

PublicPrivateKey

+algorithm: String

+encrypt(String, Key): byte[]
+encrypt(byte[], Key): byte[]
+decrypt(byte[], Key): byte[]
+sign(byte[], PrivateKey): byte[]
+verify(byte[], PublicKey, byte[]): boolean

Transform

#hexArray: char[]

+bytesToHex(byte[]): String
+hexToBytes(String): byte[]
+getString(byte[]): String
+getInt(byte[]): int
+getLong(byte[]): long

1

Car

-uid: int
-privateKey: PrivateKey
-publicKey: PublicKey
-tempKey: SecretKeySpec
-tempAppKey: SecretKeySpec
-bookings: Map<Integer, Booking>
-distance: double
-initTime: Date
-SEND_DELAY: int

+getPublicKey(): PublicKey
+getUID(): int
+receiveAESKey(byte[], byte[]): void
+receiveBookingInfo(byte[]): viod
+execute(byte[]): void
-decrypt(byte[], Key, Mode, byte[]): void
-execute(long, int, byte[]): void
-sendAESkey(): SecretKey

<<Enumeration>>
Command

-AES_APP
-AES_CCM
-COMMAND
-CCM

<<Enumeration>>
Command

-UNLOCK
-LOCK
-START
-STOP

+getCommand(): String
+toString(): String

Main

-app: SmartphoneApp
-COMMANDS_TEXT: String

-processInput(): void
-processCommand(String): void
+main(String[]): void

1

SmartphoneApp

-uid: int
-privateKey: PrivateKey
-publicKey: PublicKey
-booking: Booking
-server: Server

-executeCommand(Command, int): void
-sendAESKey(): SecretKey
-encryptAES(byte[], SecretKey, long, String): byte[]
-encryptMsg(byte[], Key, String): byte[]
-print(String): void
+unlockCar(int): void
+lockCar(int): void
+startCar(int): void
+stopCar(int): void
+bookCar(int): void
+register(): void