Secure Payments Using Smartphones

Group T 07

Motivation

Smartphones are part of everyone’s life:

- In 2014 there were 1.5 billion smartphone users. By the end of 2016 this number is expected to be almost 2.1 billion and will still be growing for more years.

The best way to bring a service to the population is by making a small phone app that people of all ages can use easily and without mistakes.

Requirements

- ➔ **Integrity**
- ➔ **Confidentiality**
- ➔ **Authenticity**
- ➔ **Non-repudiation**
- ➔ **Double Spending**

Furthermore, all communication between the server and the database is **protected against SQL injection** (using jdbc with prepared statements).

Solution (1/2)

- UDP packets do not grant proper delivery and order we stand before an unreliable channel of communication.
- Built a security system to secure this channel against attacks and network failures.

Solution (2/2)

Basic Version

On this stage we will be creating a simple UDP message transmission protocol with max length of **120 bytes** to simulate SMS messages and implement bank transfers. Each message will have:

- (Destination + Origin) IBANs - 27 bytes
- Transaction value - 4 bytes
- Challenge-response to confirm the payment order

Intermediate Version

We will be implementing:

- Message digest truncated SHA2 (to assure integrity) - 128 bit (16 bytes)
- Nounce UUID v4 (freshness) - 128 bit (16 bytes)

Advanced Version

- AES ciphered communication
- Money transfer limit within a certain time frame to prevent abusive usage

Communication

Phone Number | IV | Hash | TID | Operation | Parameters

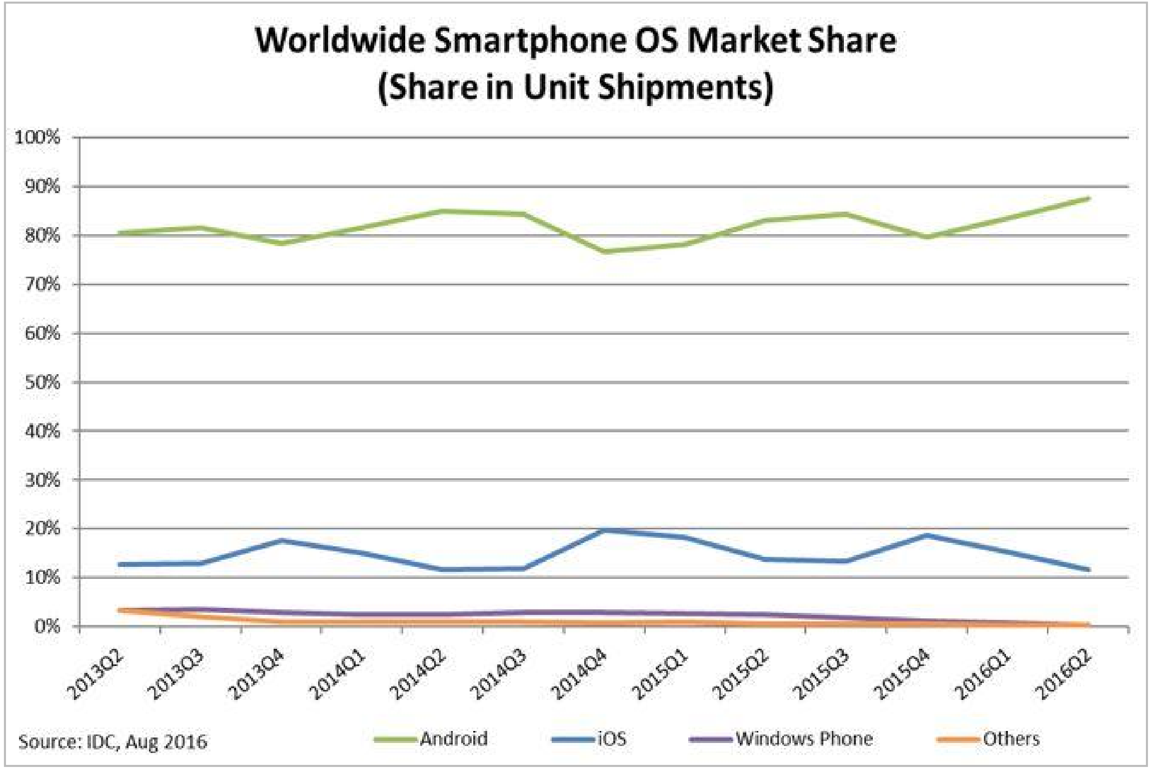
IV | Hash | TID | Operation | State | Parameters

Tools

JDBC

Easy and efficient API to establish the Server-Database communication between the

Android

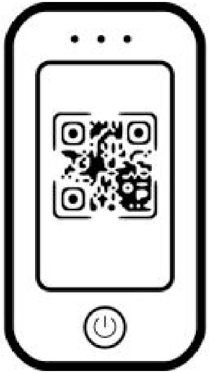


Source: IDC, Aug 2016

The most used mobile Operating System in the world

QRCode

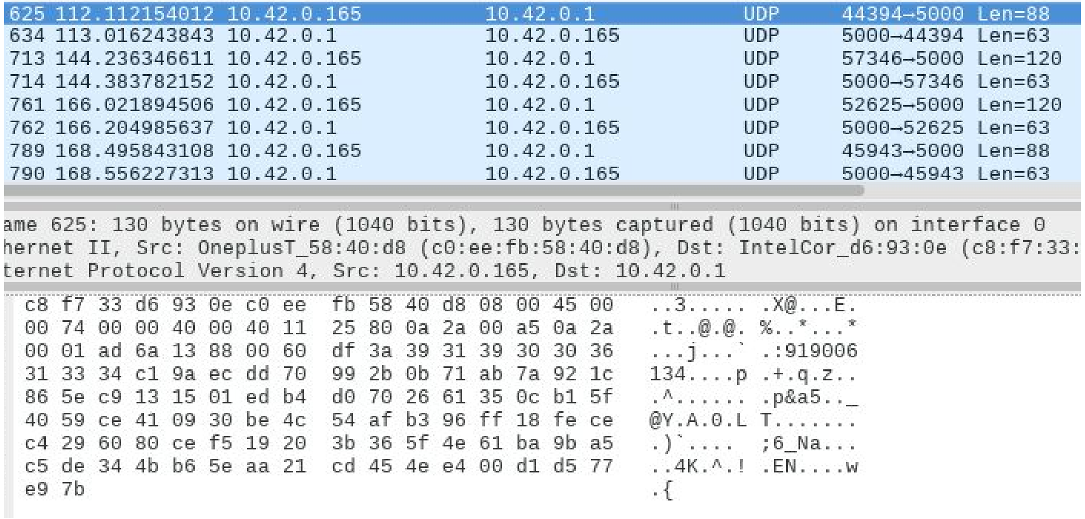
Easiest way to share information between two users



Testing

Wireshark

The best tool to sniff the network



```
625 112.112154012 10.42.0.165          10.42.0.1            UDP       44394→5000 Len=88
634 113.016243843 10.42.0.1            10.42.0.165          UDP       5000→44394 Len=63
713 144.236346611 10.42.0.165          10.42.0.1            UDP       57346→5000 Len=120
714 144.383782152 10.42.0.1            10.42.0.165          UDP       5000→57346 Len=63
761 166.021894506 10.42.0.165          10.42.0.1            UDP       52625→5000 Len=120
762 166.204985637 10.42.0.1            10.42.0.165          UDP       5000→52625 Len=63
789 168.495843108 10.42.0.165          10.42.0.1            UDP       45943→5000 Len=88
790 168.556227313 10.42.0.1            10.42.0.165          UDP       5000→45943 Len=63
```

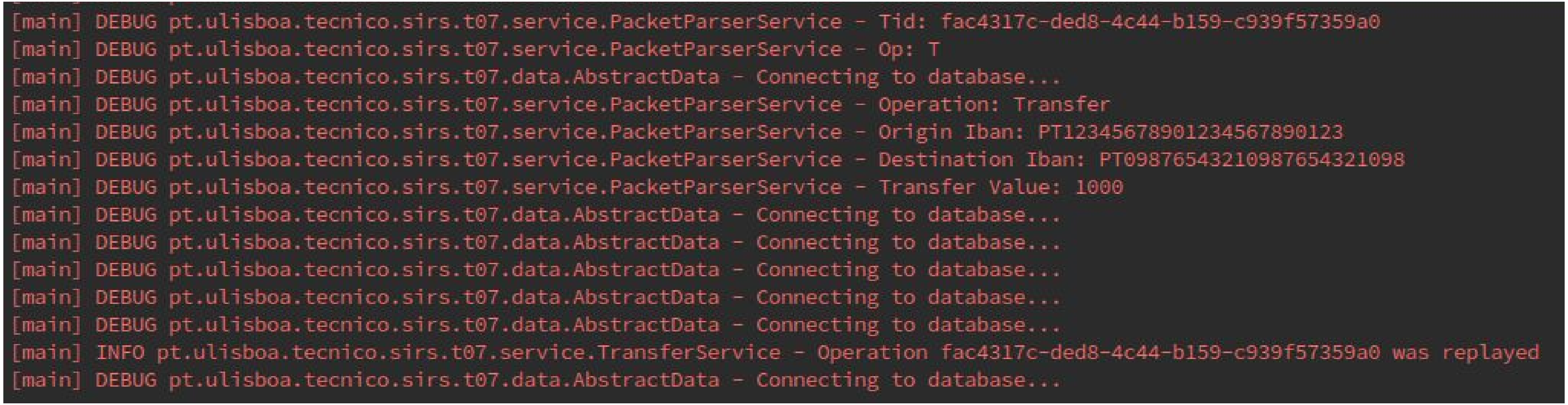
```
ame 625: 130 bytes on wire (1040 bits), 130 bytes captured (1040 bits) on interface 0
hernet II, Src: OneplusT_58:40:d8 (c0:ee:fb:58:40:d8), Dst: IntelCor_d6:93:0e (c8:f7:33:
ternet Protocol Version 4, Src: 10.42.0.165, Dst: 10.42.0.1
```

```
 c8 f7 33 d6 93 0e c0 ee  fb 58 40 d8 08 00 45 00   ..3..... .X@...E.
 00 74 00 00 40 00 40 11  25 80 0a 2a 00 a5 0a 2a   .t..@.@. %..*...*
 00 01 ad 6a 13 88 00 60  df 3a 39 31 39 30 30 36   ...j...` .:919006
 31 33 34 c1 9a ec dd 70  99 2b 0b 71 ab 7a 92 1c   134....p .+.q.z..
 86 5e c9 13 15 01 ed b4  d0 70 26 61 35 0c b1 5f   .^...... .p&a5.._
 40 59 ce 41 09 30 be 4c  54 af b3 96 ff 18 fe ce   @Y.A.0.L T.......
 c4 29 60 80 ce f5 19 20  3b 36 5f 4e 61 ba 9b a5   .)`....  ;6_Na...
 c5 de 34 4b b6 5e aa 21  cd 45 4e e4 00 d1 d5 77   ..4K.^.! .EN....w
 e9 7b                                              .{
```

The payload is encrypted to assure the requirements

Replay Attack

The server is able to detect this types of attacks

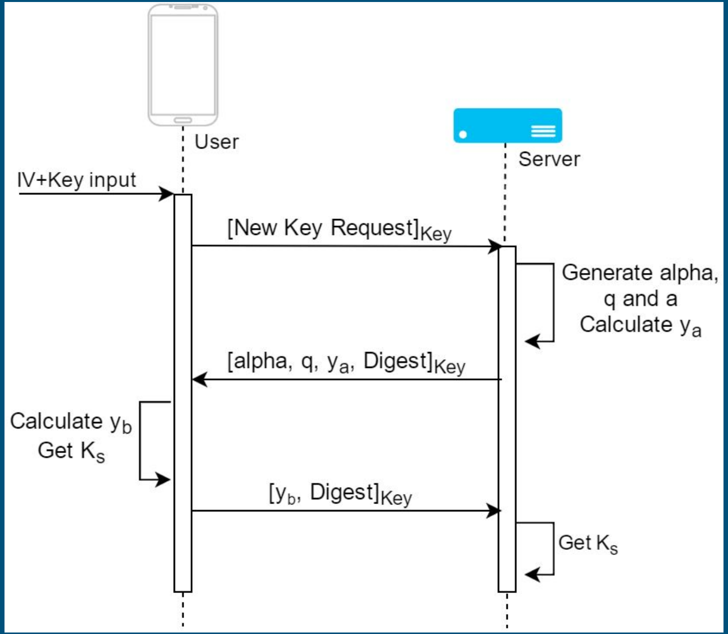


```
[main] DEBUG pt.ulisboa.tecnico.sirs.t07.service.PacketParserService - Tid: fac4317c-ded8-4c44-b159-c939f57359a0
[main] DEBUG pt.ulisboa.tecnico.sirs.t07.service.PacketParserService - Op: T
[main] DEBUG pt.ulisboa.tecnico.sirs.t07.data.AbstractData - Connecting to database...
[main] DEBUG pt.ulisboa.tecnico.sirs.t07.service.PacketParserService - Operation: Transfer
[main] DEBUG pt.ulisboa.tecnico.sirs.t07.service.PacketParserService - Origin Iban: PT12345678901234567890123
[main] DEBUG pt.ulisboa.tecnico.sirs.t07.service.PacketParserService - Destination Iban: PT09876543210987654321098
[main] DEBUG pt.ulisboa.tecnico.sirs.t07.service.PacketParserService - Transfer Value: 1000
[main] DEBUG pt.ulisboa.tecnico.sirs.t07.data.AbstractData - Connecting to database...
[main] DEBUG pt.ulisboa.tecnico.sirs.t07.data.AbstractData - Connecting to database...
[main] DEBUG pt.ulisboa.tecnico.sirs.t07.data.AbstractData - Connecting to database...
[main] DEBUG pt.ulisboa.tecnico.sirs.t07.data.AbstractData - Connecting to database...
[main] DEBUG pt.ulisboa.tecnico.sirs.t07.data.AbstractData - Connecting to database...
[main] INFO pt.ulisboa.tecnico.sirs.t07.service.TransferService - Operation fac4317c-ded8-4c44-b159-c939f57359a0 was replayed
[main] DEBUG pt.ulisboa.tecnico.sirs.t07.data.AbstractData - Connecting to database...
```

Future Work

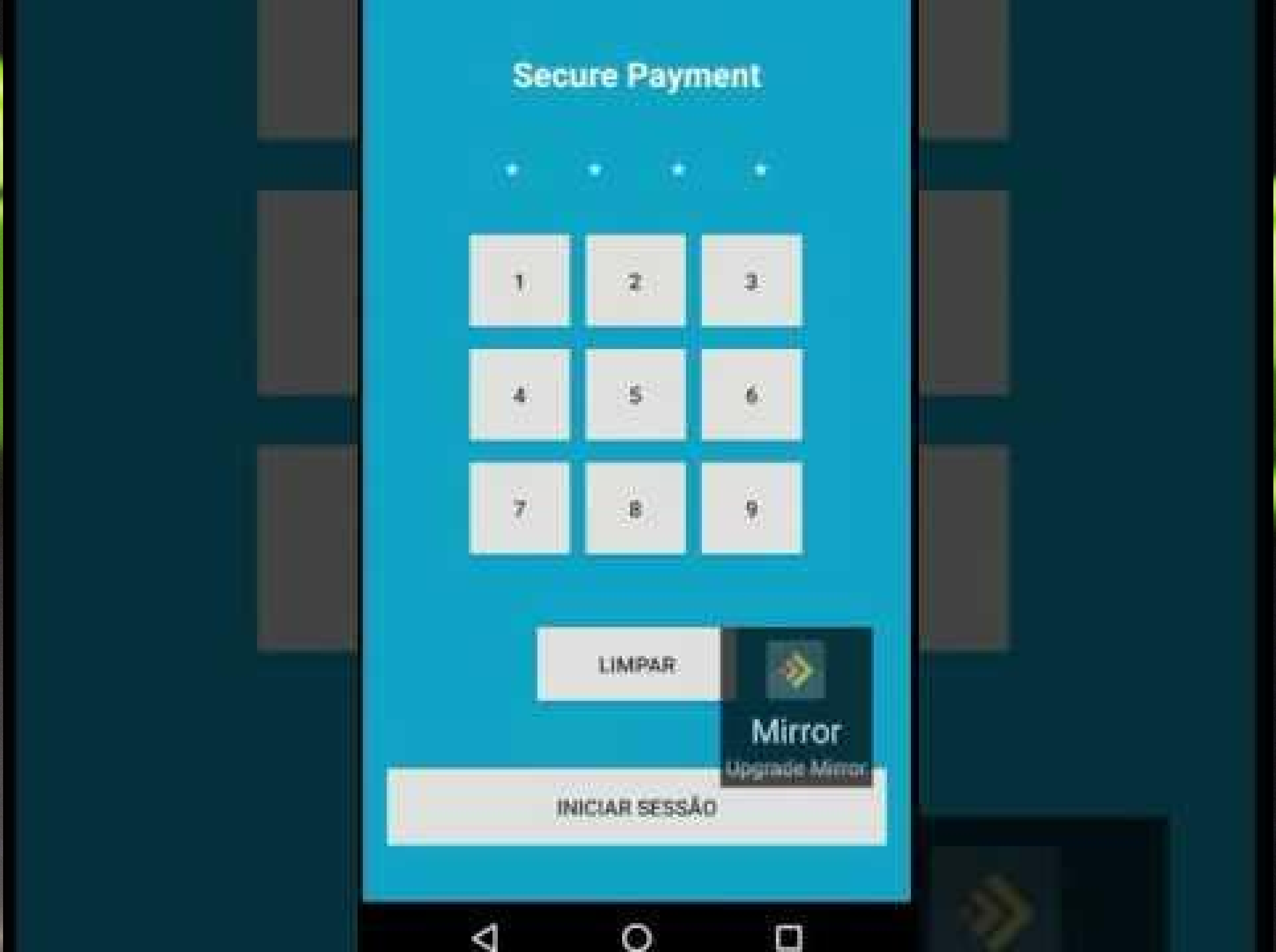
Communication

1. The bank will provide to the client a code containing an **IV** and a **Key**.
2. The app will ask the user for this code and in the end of the setup process the smartphone sends a new key request to the server.
3. The server calculates the **Diffie Hellman** parameters and sends multiple UDP messages to the user's smartphone containing all the information needed to perform the rest of the algorithm.
4. Finally the user's smartphone sends the information that the server needs to conclude the algorithm and both share a new Key.


User
Server
IV+Key input
[New Key Request]$_{Key}$
Generate alpha, q and a Calculate y_a
[alpha, q, y_a, Digest]$_{Key}$
Calculate y_b Get K_s
[y_b, Digest]$_{Key}$
Get K_s

All the messages sent and received are encrypted by the initial key with AES to assure integrity.

Live Demo


Secure Payment
• • • •
1
2
3
4
5
6
7
8
9
LIMPAR
Mirror
Upgrade Mirror
INICIAR SESSÃO

Demo