BombAppetit

SIRS - Group 14

ist192525 Miguel Nunes
ist199250 João Vieira
ist193460 João Lima

Introduction

Secure Document

Infrastructure

Secure Channels and Key Distribution

Security Challenge

Conclusion

Secure Document Format - Protect

- MealVoucher **confidentiality**: Encrypt with client public key
- Restaurant data **authenticity**: Digital signature including cryptographic hash and random nonce
- Client and Server keypair RSA encrypted with 2048 bits length

Secure Document Format - Check

- Uses the **server**'s public key to decrypt the cryptographic hash and nonce.
- Computes new hash and checks against received hash
- Checks the decrypted nonce against existing nonces

nonce.txt

```
1   PqUDiKkpcoW7TKtlDgbN0t7gtpdWJZzDI8sYSO7KjmM6rxzL
2   HbRtLQnG5XsZDv7m1YrPwQF8lEk2E3hsz9AWvxMJUaKFUpi3
3   eWjVbhoNtx0NzF96ziYRvxV7jc6Akz8Lbq/WG6kH9iCmJ/hH
4   ywUd0lHdK5KbvsoRcq2Y6pNmReyQPlS8qnb2lQr2hUD5fBLw
5   QgA1md4Rns48rPtRkQajhTvw6swYkWJozWL7B0V0DvE5Qxep
6   VUx7SaeM8Xwlh7Rvl/pfbIzSlQAlbCkcNcWg0T0+ps9kJk7T
7   lC4ymSC2QVPhjFgRFIyYIw3HwBFG8yzTwKGzrPnFp6eFR2rp
8   SzDcKdVpVXme+RQtzLH1wVm6VD8P7Kl5xPYOnq0YQUoRIS2D
9   CnZS1Vcn0p4s9uIfFy7sJTFgP2JXGCBGKkLzNZrYj3q5H5MZ
10  fDZU3QUepsmVj2V2zB7ROAc8J7lw1LsyZSe9Lr7WGLMOyRuU
```

```
"security": {
  "hash": "SmXinHXR6lkkPzNnENBHa7Xdf2DXVKLc
  "nonce": "G57jvVktSe9UNkezEsO2PENL0qnlX5g
}
```

Secure Document Format - Unprotect

- Removes the encryption from mealVoucher and removes the security field from a protected document.

Technologies used:

- Java crypto library
- Google Gson library
- Maven

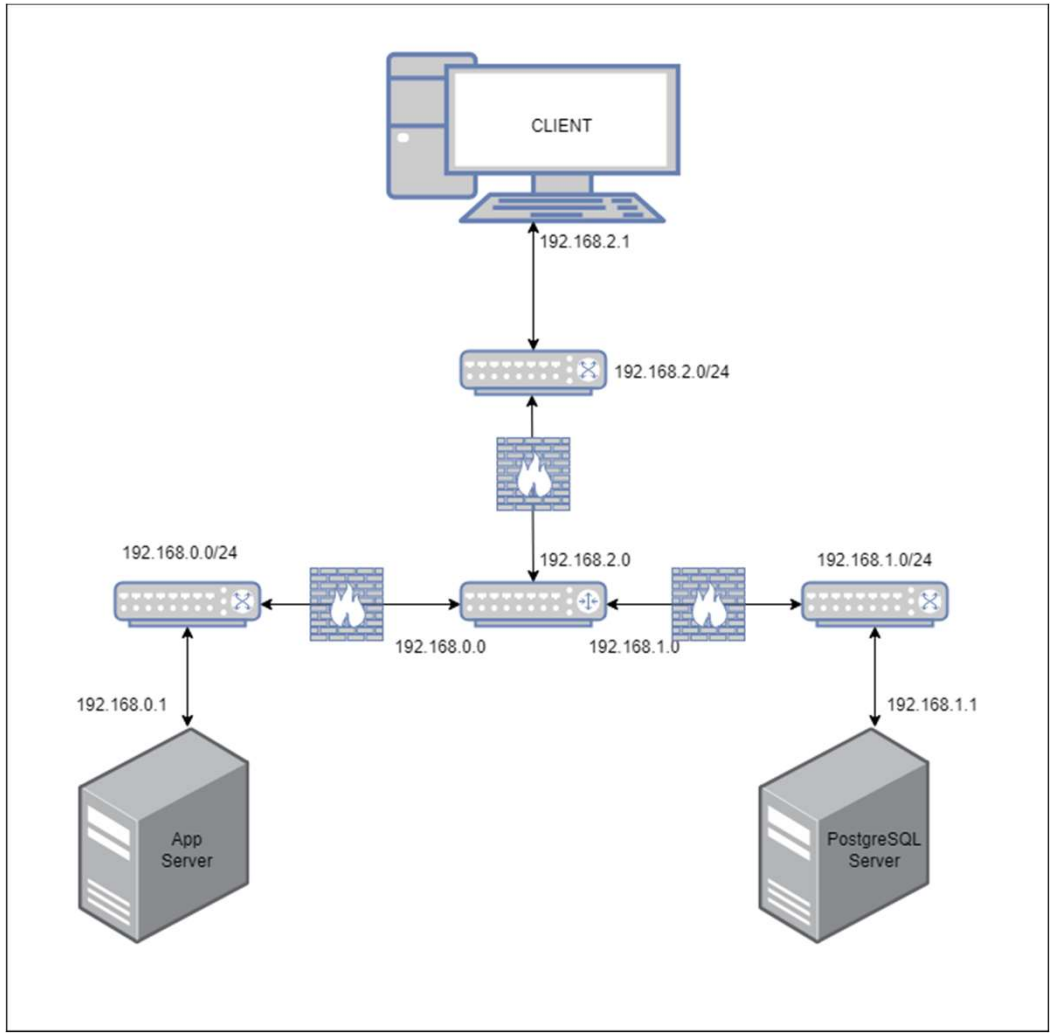
Unprotected Document


```
{
  "restaurantInfo": {
    "owner": "Maria Silva",
    "restaurant": "Dona Maria",
    "address": "Rua da Glória, 22, Lisboa",
    "genre": ["Portuguese", "Traditional"],
    "menu": [
      {
        "itemName": "House Steak",
        "category": "Meat",
        "description": "A succulent sirloin grilled steak.",
        "price": 24.99,
        "currency": "EUR"
      },
      {
        "itemName": "Sardines",
        "category": "Fish",
        "description": "A Portuguese staple, accompanied by potatoes and salad.",
        "price": 21.99,
        "currency": "EUR"
      },
      {
        "itemName": "Mushroom Risotto",
        "category": "Vegetarian",
        "description": "Creamy Arborio rice cooked with assorted mushrooms and Parmesan cheese.",
        "price": 16.99,
        "currency": "EUR"
      }
    ],
    "mealVoucher": {
      "code": "VOUCHER123",
      "description": "Redeem this code for a 20% discount in the meal. Drinks not included."
    }
  }
}
```

Infrastructure

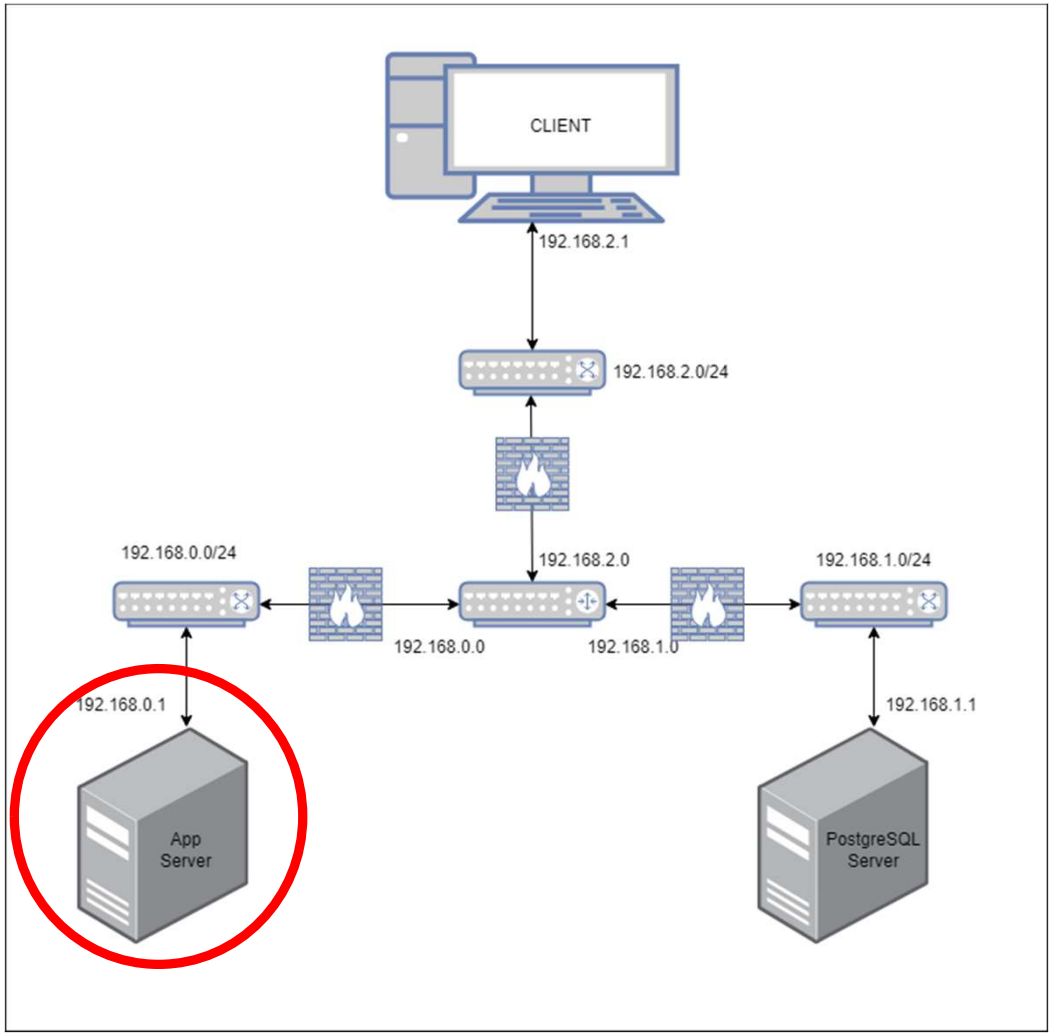
- 4 Machines
  - Application Server
  - Database Server
  - Client
  - Router

- 3 Subnets
  - Application ⇔ Router (192.168.0.0/24)
  - Database ⇔ Router (192.168.1.0/24)
  - Client ⇔ Router (192.168.2.0/24)



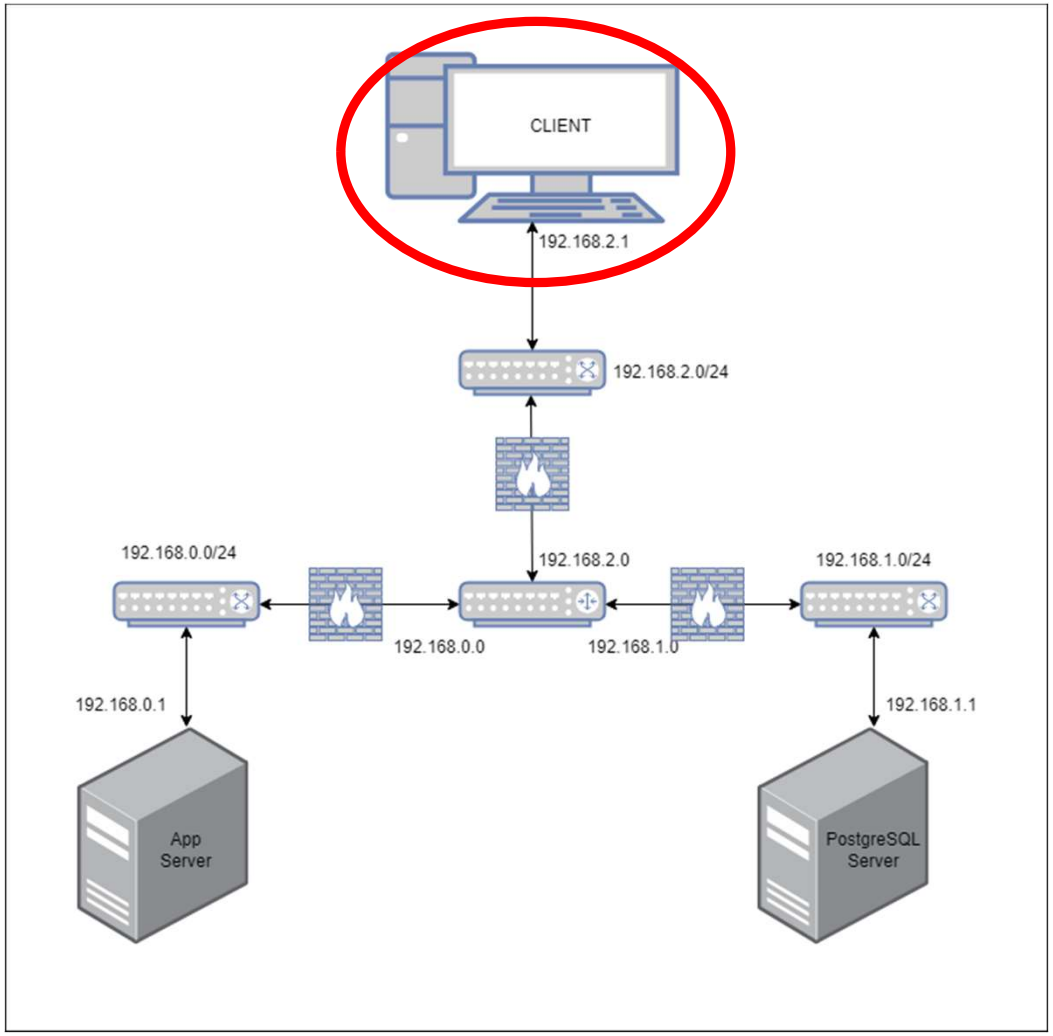
Application Server

- IP 192.168.0.1
- Exposed in port 8443
- Using the Spring framework in Java



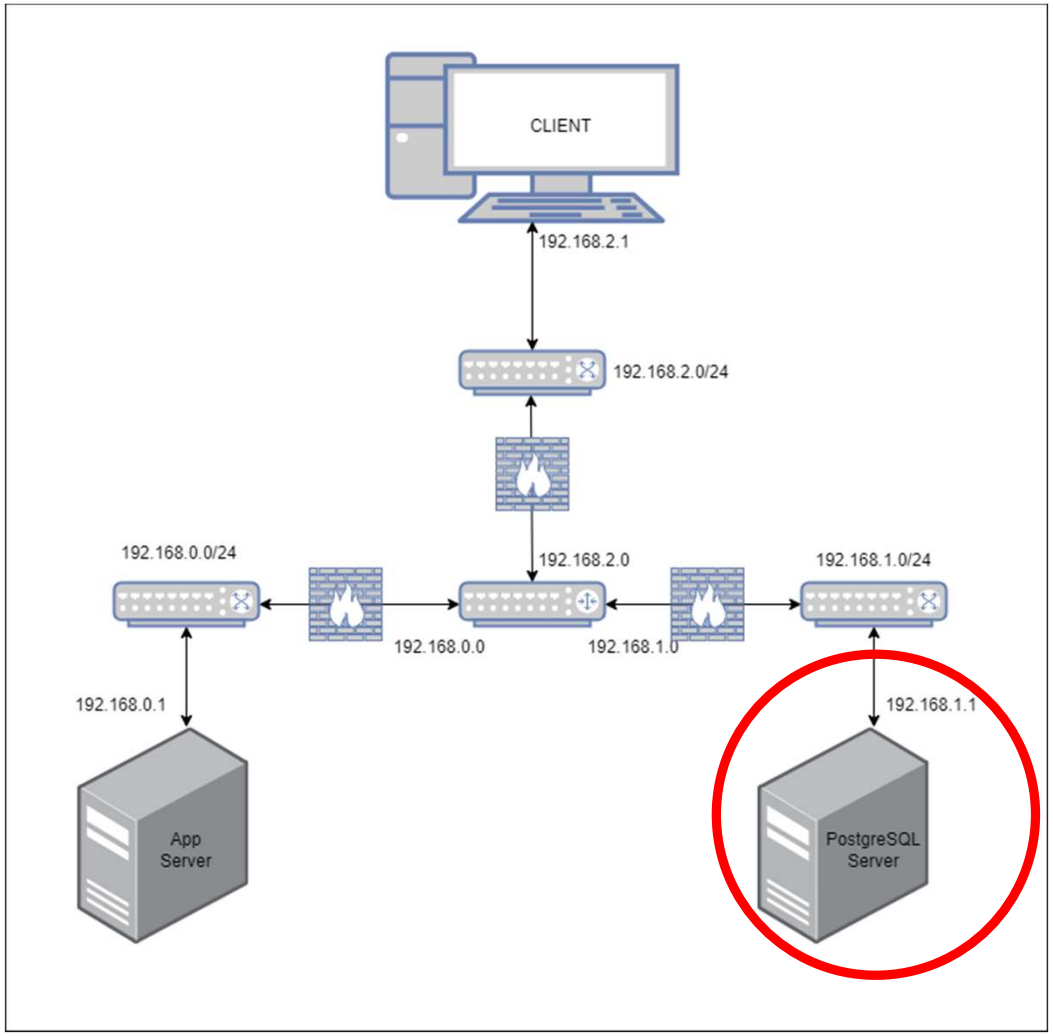
Client

- IP 192.168.2.1
- Python command line application
- Possible operations
  - Get Restaurant Information
  - Create Review
  - Transfer Vouchers



Database Server

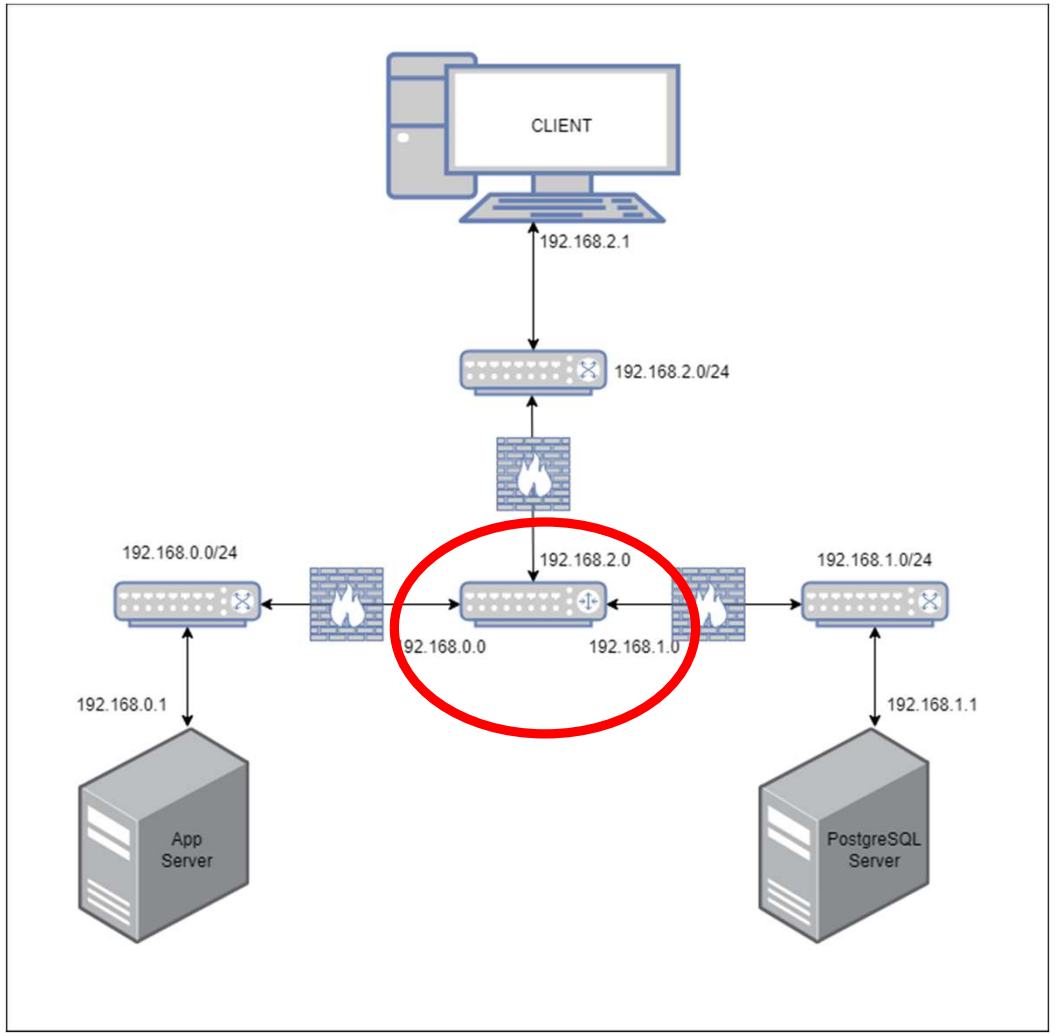
- IP 192.168.1.1
- Exposed in port 5432
- Using PostgreSQL as the DB



Router

- An interface for each subnet
- Firewall Rules

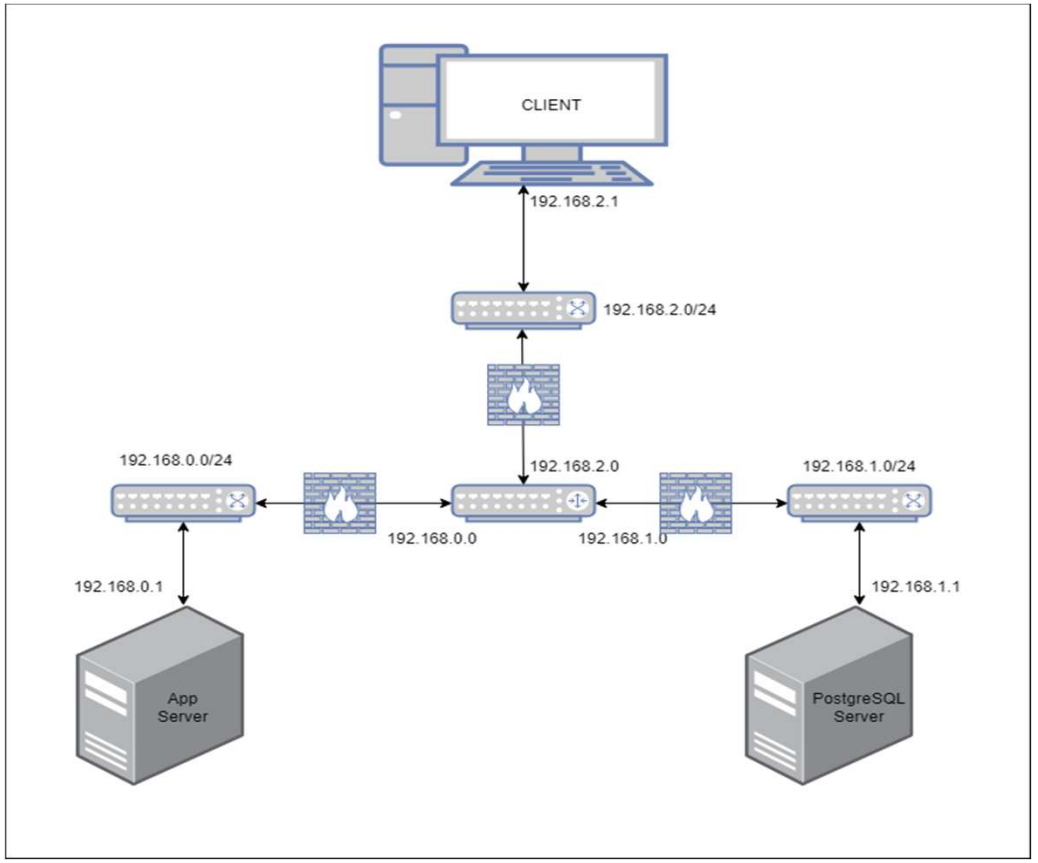
```
Chain FORWARD (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  192.168.1.0/24       192.168.0.0/24
ACCEPT     all  --  192.168.0.0/24       192.168.1.0/24
ACCEPT     all  --  192.168.2.0/24       192.168.0.0/24
ACCEPT     all  --  192.168.0.0/24       192.168.2.0/24
DROP       all  --  192.168.2.0/24       192.168.1.0/24
DROP       all  --  192.168.1.0/24       192.168.2.0/24
```



Secure channels and key distribution

- TLS: Client ⇔ Server
- No TLS: Server ⇔ DB

- Keys-Pairs already created with Public Keys already exchanged.



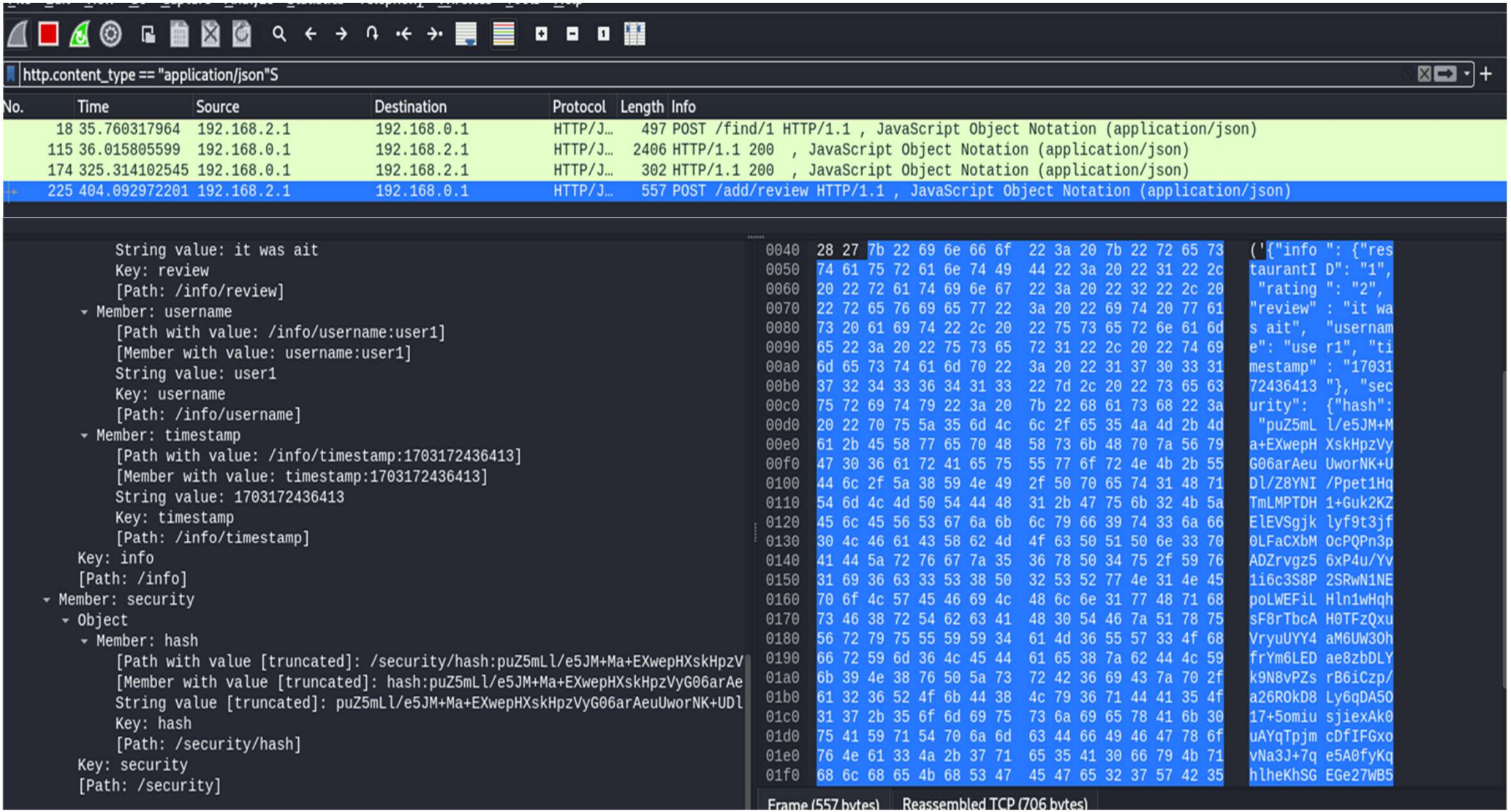
Security Considerations

- Assumptions
  - Communication channel between the Application and the Database can’t be compromised
  - The Application and the Database are fully trusted

- Possible Limitations
  - No authentication service leads to possible impersonation

Security Challenge - Reviews

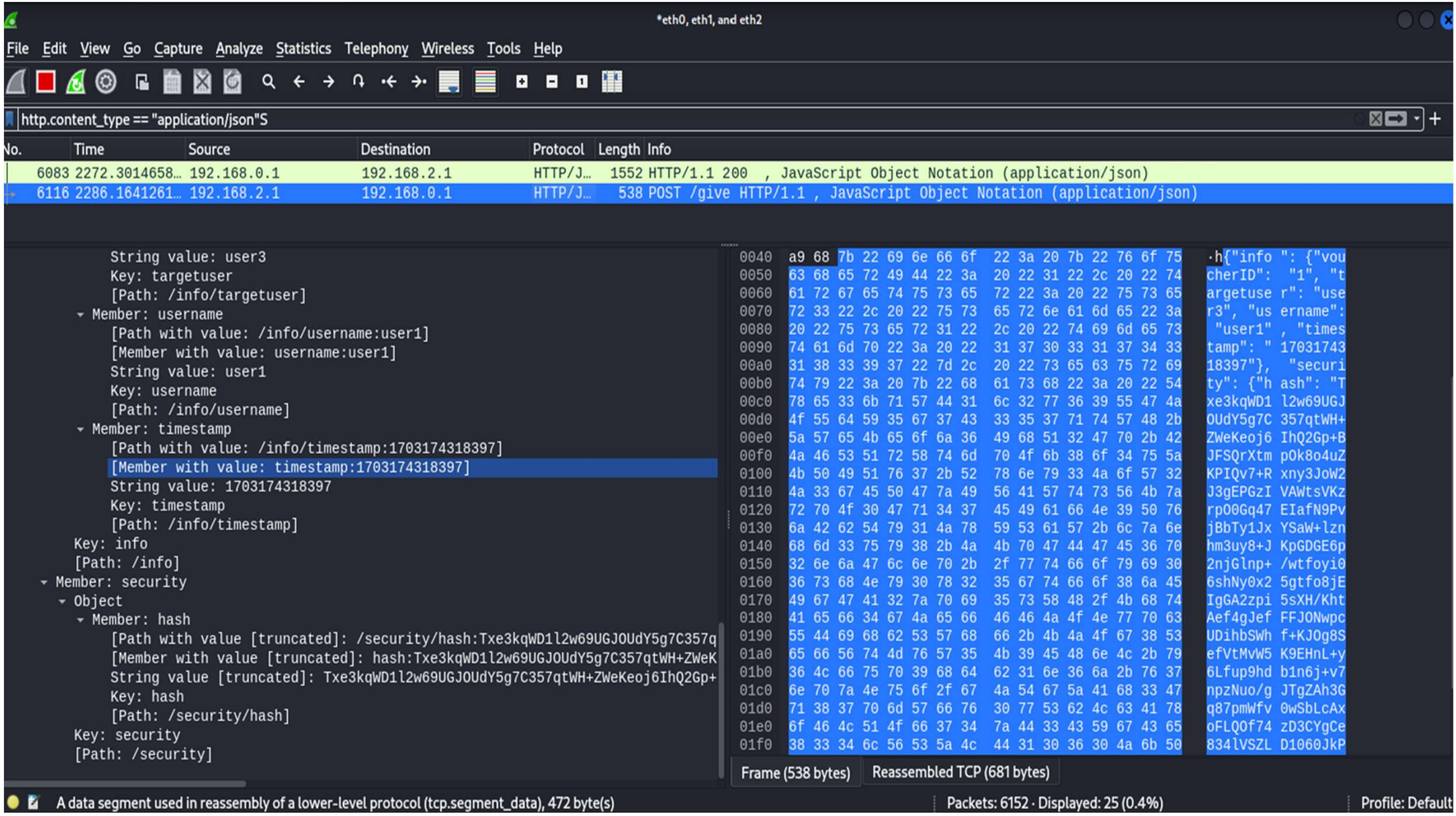
- instead of a random nonce to guarantee freshness, we use a timestamp.
- **Authenticity** is guaranteed by the digital signature
- Reviews are necessarily **non-repudiable** since the author's identity and message integrity are guaranteed


http.content_type == "application/json"S
No. Time Source Destination Protocol Length Info
18 35.760317964 192.168.2.1 192.168.0.1 HTTP/J… 497 POST /find/1 HTTP/1.1 , JavaScript Object Notation (application/json)
115 36.015805599 192.168.0.1 192.168.2.1 HTTP/J… 2406 HTTP/1.1 200 , JavaScript Object Notation (application/json)
174 325.314102545 192.168.0.1 192.168.2.1 HTTP/J… 302 HTTP/1.1 200 , JavaScript Object Notation (application/json)
225 404.092972201 192.168.2.1 192.168.0.1 HTTP/J… 557 POST /add/review HTTP/1.1 , JavaScript Object Notation (application/json)
String value: it was ait
Key: review
[Path: /info/review]
Member: username
[Path with value: /info/username:user1]
[Member with value: username:user1]
String value: user1
Key: username
[Path: /info/username]
Member: timestamp
[Path with value: /info/timestamp:1703172436413]
[Member with value: timestamp:1703172436413]
String value: 1703172436413
Key: timestamp
[Path: /info/timestamp]
Key: info
[Path: /info]
Member: security
Object
Member: hash
[Path with value [truncated]: /security/hash:puZ5mLl/e5JM+Ma+EXwepHXskHpzV
[Member with value [truncated]: hash:puZ5mLl/e5JM+Ma+EXwepHXskHpzVyG06arAe
String value [truncated]: puZ5mLl/e5JM+Ma+EXwepHXskHpzVyG06arAeuUworNK+UDl
Key: hash
[Path: /security/hash]
Key: security
[Path: /security]
0040 28 27 7b 22 69 6e 66 6f 22 3a 20 7b 22 72 65 73 ('{"info ": {"res
0050 74 61 75 72 61 6e 74 49 44 22 3a 20 22 31 22 2c taurantI D": "1",
0060 20 22 72 61 74 69 6e 67 22 3a 20 22 32 22 2c 20 "rating ": "2",
0070 22 72 65 76 69 65 77 22 3a 20 22 69 74 20 77 61 "review" : "it wa
0080 73 20 61 69 74 22 2c 20 22 75 73 65 72 6e 61 6d s ait", "usernam
0090 65 22 3a 20 22 75 73 65 72 31 22 2c 20 22 74 69 e": "use r1", "ti
00a0 6d 65 73 74 61 6d 70 22 3a 20 22 31 37 30 33 31 mestamp" : "17031
00b0 37 32 34 33 36 34 31 33 22 7d 2c 20 22 73 65 63 72436413 "}, "sec
00c0 75 72 69 74 79 22 3a 20 7b 22 68 61 73 68 22 3a urity": {"hash":
00d0 20 22 70 75 5a 35 6d 4c 6c 2f 65 35 4a 4d 2b 4d "puZ5mL l/e5JM+M
00e0 61 2b 45 58 77 65 70 48 58 73 6b 48 70 7a 56 79 a+EXwepH XskHpzVy
00f0 47 30 36 61 72 41 65 75 55 77 6f 72 4e 4b 2b 55 G06arAeu UworNK+U
0100 44 6c 2f 5a 38 59 4e 49 2f 50 70 65 74 31 48 71 Dl/Z8YNI /Ppet1Hq
0110 54 6d 4c 4d 50 54 44 48 31 2b 47 75 6b 32 4b 5a TmLMPTDH 1+Guk2KZ
0120 45 6c 45 56 53 67 6a 6b 6c 79 66 39 74 33 6a 66 ElEVSgjk lyf9t3jf
0130 30 4c 46 61 43 58 62 4d 4f 63 50 51 50 6e 33 70 0LFaCXbM OcPQPn3p
0140 41 44 5a 72 76 67 7a 35 36 78 50 34 75 2f 59 76 ADZrvgz5 6xP4u/Yv
0150 31 69 36 63 33 53 38 50 32 53 52 77 4e 31 4e 45 1i6c3S8P 2SRwN1NE
0160 70 6f 4c 57 45 46 69 4c 48 6c 6e 31 77 48 71 68 poLWEFiL Hln1wHqh
0170 73 46 38 72 54 62 63 41 48 30 54 46 7a 51 78 75 sF8rTbcA H0TFzQxu
0180 56 72 79 75 55 59 59 34 61 4d 36 55 57 33 4f 68 VryuUYY4 aM6UW3Oh
0190 66 72 59 6d 36 4c 45 44 61 65 38 7a 62 44 4c 59 frYm6LED ae8zbDLY
01a0 6b 39 4e 38 76 50 5a 73 72 42 36 69 43 7a 70 2f k9N8vPZs rB6iCzp/
01b0 61 32 36 52 4f 6b 44 38 4c 79 36 71 44 41 35 4f a26ROkD8 Ly6qDA5O
01c0 31 37 2b 35 6f 6d 69 75 73 6a 69 65 78 41 6b 30 17+5omiu sjiexAk0
01d0 75 41 59 71 54 70 6a 6d 63 44 66 49 46 47 78 6f uAYqTpjm cDfIFGxo
01e0 76 4e 61 33 4a 2b 37 71 65 35 41 30 66 79 4b 71 vNa3J+7q e5A0fyKq
01f0 68 6c 68 65 4b 68 53 47 45 47 65 32 37 57 42 35 hlheKhSG EGe27WB5
Frame (557 bytes)
Reassembled TCP (706 bytes)

Security Challenge - Voucher Transfer

- Since we assume the **database** is secure, our solution for the transfer of vouchers between clients is handled **server** side.

- There is no need to create dynamic key distribution between **clients** since the only **client** involved in the process is the originator.


*eth0, eth1, and eth2
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http.content_type == "application/json"S
No. Time Source Destination Protocol Length Info
6083 2272.3014658… 192.168.0.1 192.168.2.1 HTTP/J… 1552 HTTP/1.1 200 , JavaScript Object Notation (application/json)
6116 2286.1641261… 192.168.2.1 192.168.0.1 HTTP/J… 538 POST /give HTTP/1.1 , JavaScript Object Notation (application/json)
String value: user3
Key: targetuser
[Path: /info/targetuser]
Member: username
[Path with value: /info/username:user1]
[Member with value: username:user1]
String value: user1
Key: username
[Path: /info/username]
Member: timestamp
[Path with value: /info/timestamp:1703174318397]
[Member with value: timestamp:1703174318397]
String value: 1703174318397
Key: timestamp
[Path: /info/timestamp]
Key: info
[Path: /info]
Member: security
Object
Member: hash
[Path with value [truncated]: /security/hash:Txe3kqWD1l2w69UGJOUdY5g7C357q
[Member with value [truncated]: hash:Txe3kqWD1l2w69UGJOUdY5g7C357qtWH+ZWeK
String value [truncated]: Txe3kqWD1l2w69UGJOUdY5g7C357qtWH+ZWeKeoj6IhQ2Gp+
Key: hash
[Path: /security/hash]
Key: security
[Path: /security]
0040 a9 68 7b 22 69 6e 66 6f 22 3a 20 7b 22 76 6f 75 ·h{"info ": {"vou
0050 63 68 65 72 49 44 22 3a 20 22 31 22 2c 20 22 74 cherID": "1", "t
0060 61 72 67 65 74 75 73 65 72 22 3a 20 22 75 73 65 argetuse r": "use
0070 72 33 22 2c 20 22 75 73 65 72 6e 61 6d 65 22 3a r3", "us ername":
0080 20 22 75 73 65 72 31 22 2c 20 22 74 69 6d 65 73 "user1" , "times
0090 74 61 6d 70 22 3a 20 22 31 37 30 33 31 37 34 33 tamp": " 17031743
00a0 31 38 33 39 37 22 7d 2c 20 22 73 65 63 75 72 69 18397"}, "securi
00b0 74 79 22 3a 20 7b 22 68 61 73 68 22 3a 20 22 54 ty": {"h ash": "T
00c0 78 65 33 6b 71 57 44 31 6c 32 77 36 39 55 47 4a xe3kqWD1 l2w69UGJ
00d0 4f 55 64 59 35 67 37 43 33 35 37 71 74 57 48 2b OUdY5g7C 357qtWH+
00e0 5a 57 65 4b 65 6f 6a 36 49 68 51 32 47 70 2b 42 ZWeKeoj6 IhQ2Gp+B
00f0 4a 46 53 51 72 58 74 6d 70 4f 6b 38 6f 34 75 5a JFSQrXtm pOk8o4uZ
0100 4b 50 49 51 76 37 2b 52 78 6e 79 33 4a 6f 57 32 KPIQv7+R xny3JoW2
0110 4a 33 67 45 50 47 7a 49 56 41 57 74 73 56 4b 7a J3gEPGzI VAWtsVKz
0120 72 70 4f 30 47 71 34 37 45 49 61 66 4e 39 50 76 rpO0Gq47 EIafN9Pv
0130 6a 42 62 54 79 31 4a 78 59 53 61 57 2b 6c 7a 6e jBbTy1Jx YSaW+lzn
0140 68 6d 33 75 79 38 2b 4a 4b 70 47 44 47 45 36 70 hm3uy8+J KpGDGE6p
0150 32 6e 6a 47 6c 6e 70 2b 2f 77 74 66 6f 79 69 30 2njGlnp+ /wtfoyi0
0160 36 73 68 4e 79 30 78 32 35 67 74 66 6f 38 6a 45 6shNy0x2 5gtfo8jE
0170 49 67 47 41 32 7a 70 69 35 73 58 48 2f 4b 68 74 IgGA2zpi 5sXH/Kht
0180 41 65 66 34 67 4a 65 66 46 46 4a 4f 4e 77 70 63 Aef4gJef FFJONwpc
0190 55 44 69 68 62 53 57 68 66 2b 4b 4a 4f 67 38 53 UDihbSWh f+KJOg8S
01a0 65 66 56 74 4d 76 57 35 4b 39 45 48 6e 4c 2b 79 efVtMvW5 K9EHnL+y
01b0 36 4c 66 75 70 39 68 64 62 31 6e 36 6a 2b 76 37 6Lfup9hd b1n6j+v7
01c0 6e 70 7a 4e 75 6f 2f 67 4a 54 67 5a 41 68 33 47 npzNuo/g JTgZAh3G
01d0 71 38 37 70 6d 57 66 76 30 77 53 62 4c 63 41 78 q87pmWfv 0wSbLcAx
01e0 6f 46 4c 51 4f 66 37 34 7a 44 33 43 59 67 43 65 oFLQOf74 zD3CYgCe
01f0 38 33 34 6c 56 53 5a 4c 44 31 30 36 30 4a 6b 50 834lVSZL D1060JkP
Frame (538 bytes)
Reassembled TCP (681 bytes)
A data segment used in reassembly of a lower-level protocol (tcp.segment_data), 472 byte(s)
Packets: 6152 · Displayed: 25 (0.4%)
Profile: Default

Conclusion

- We developed a secure app that is able to send and receive encrypted data, while also verifying its integrity.
- We are able to:
  - Send and receive reviews
  - Send voucher to other users
  - Protect and unprotect documents
  - Check authenticity and freshness of documents

- Future work:
  - Add Gui interface to the user application
  - Add Authentication Service
  - Add TLS between Application Server and Database
  - Add the ability for more restaurants to register

Live Demonstration

The End