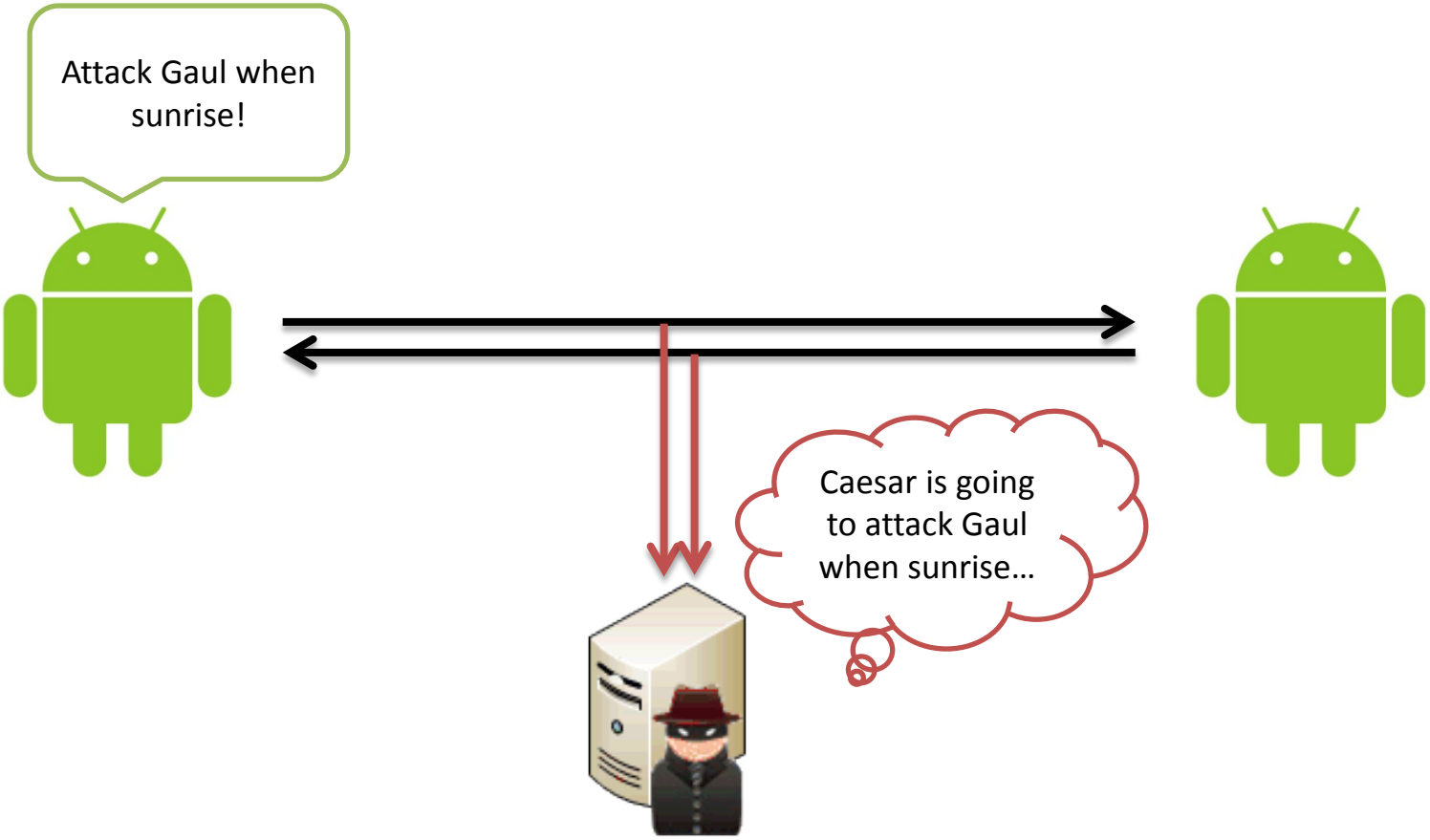
CSCI3310 Course Project

Secure Instant Messaging on Android

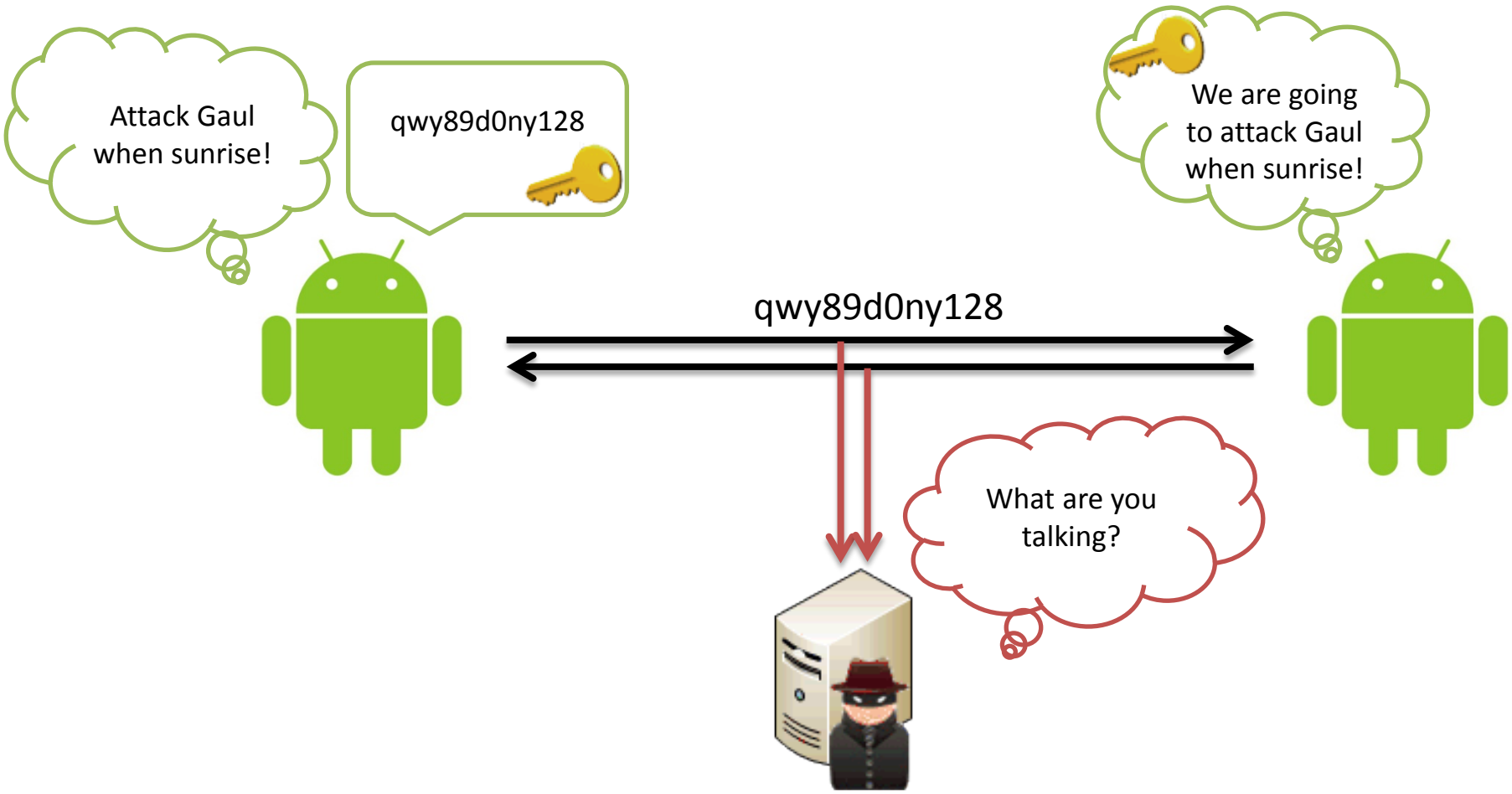
Student name: Lo Yu Ho

Group number: 21

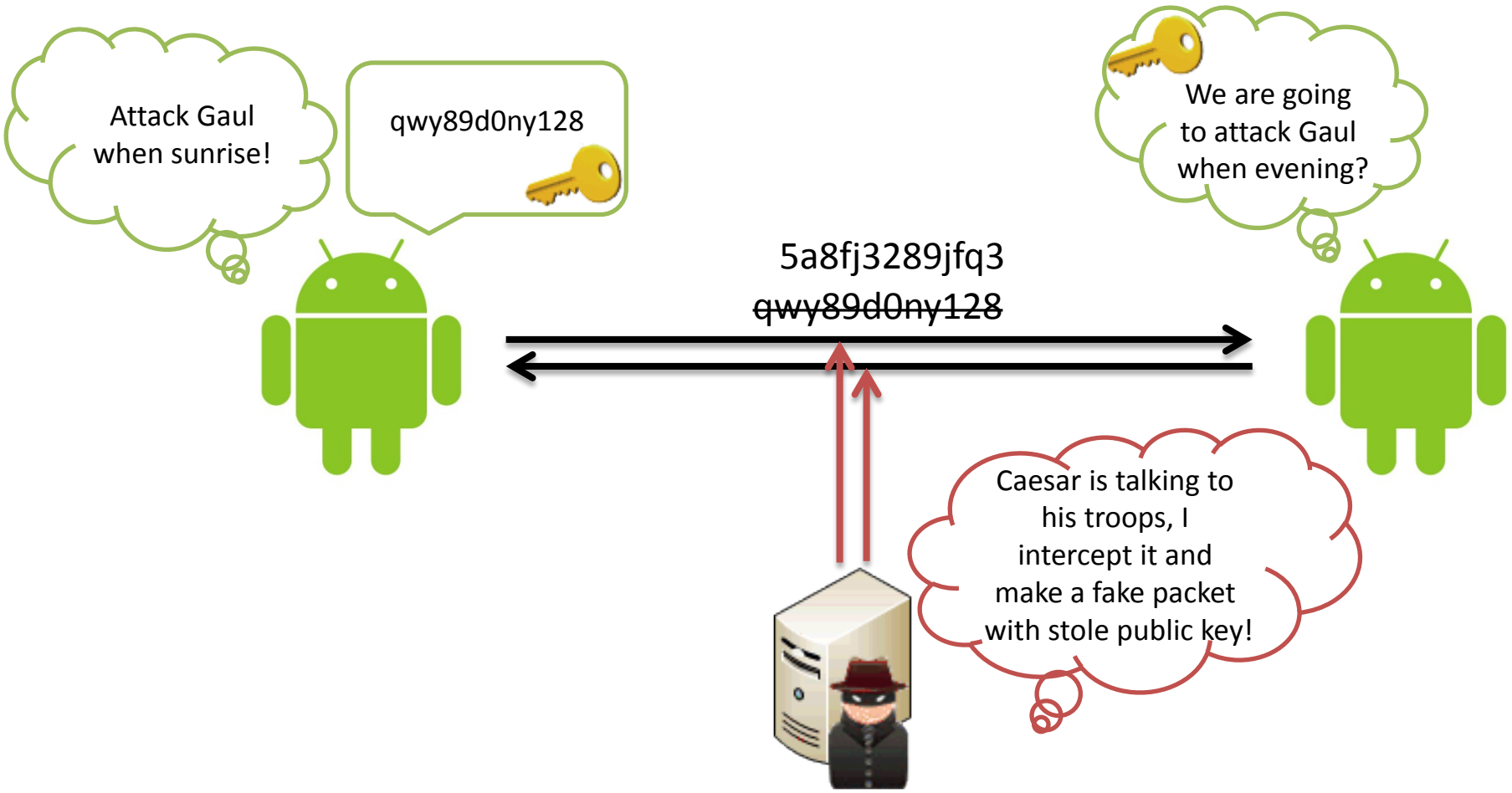
Motivation


Attack Gaul when sunrise!
Caesar is going to attack Gaul when sunrise…

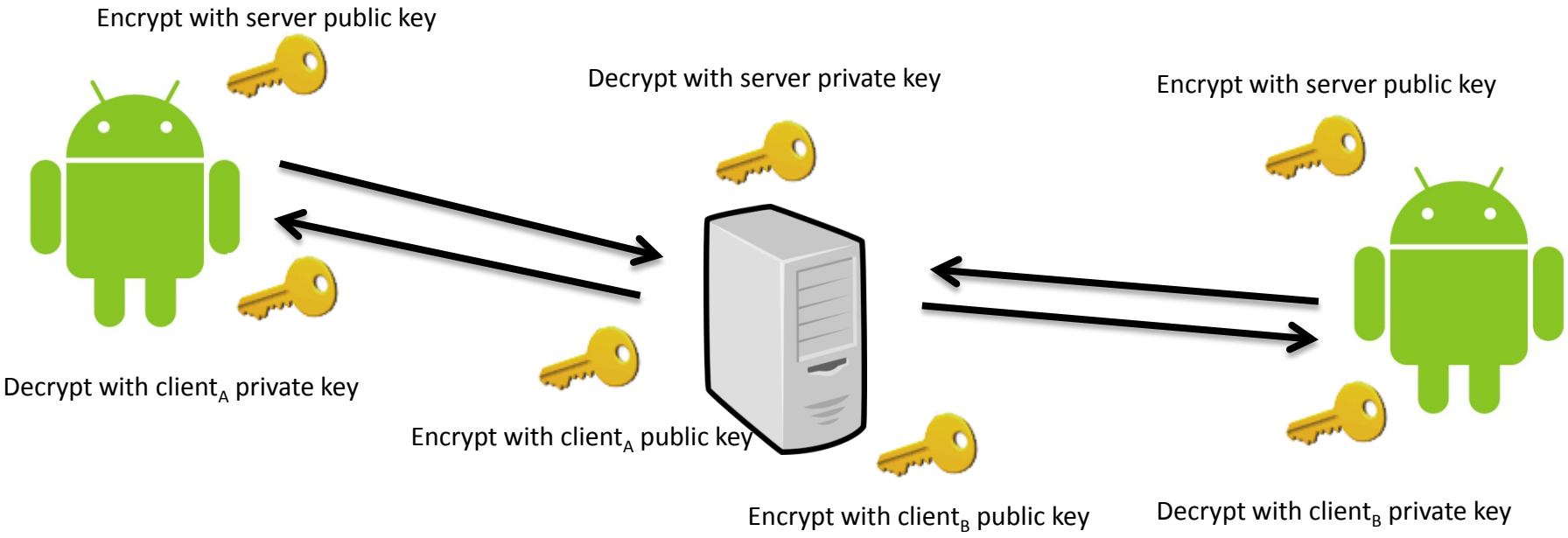
Encryption


Attack Gaul when sunrise!
qwy89d0ny128
We are going to attack Gaul when sunrise!
qwy89d0ny128
What are you talking?

Problem


Attack Gaul when sunrise!
qwy89d0ny128
We are going to attack Gaul when evening?
5a8fj3289jfq3
~~qwy89d0ny128~~
Caesar is talking to his troops, I intercept it and make a fake packet with stole public key!

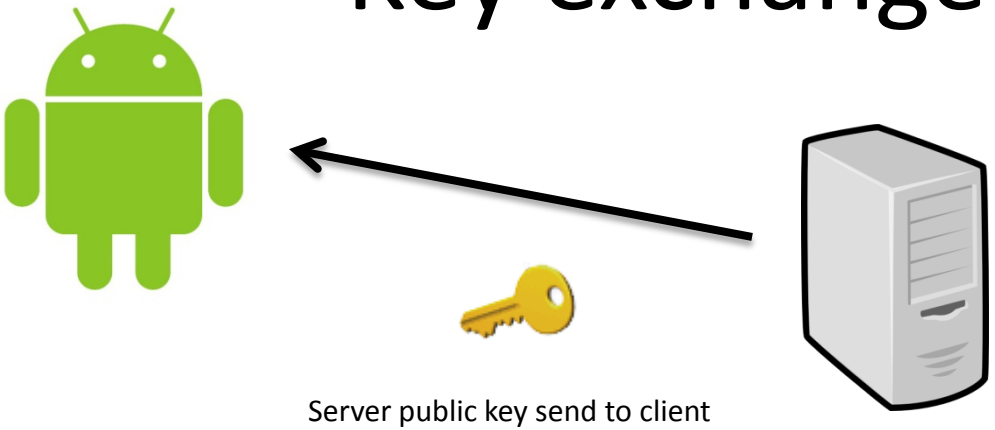
Approach



Each client: 1 public key (given by server), 1 private key (generate by client itself)
Server: N public keys (given by clients), 1 private key (generate by server itself)

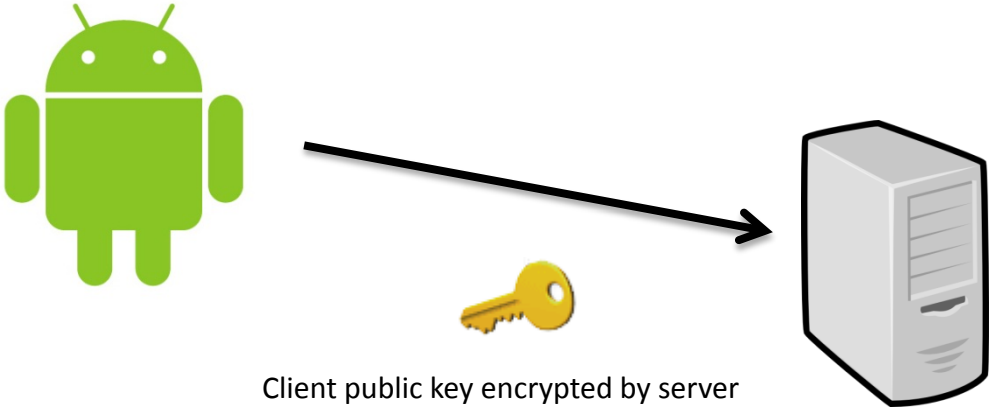
Key exchange

Step 1.



Server public key send to client

Step 2.



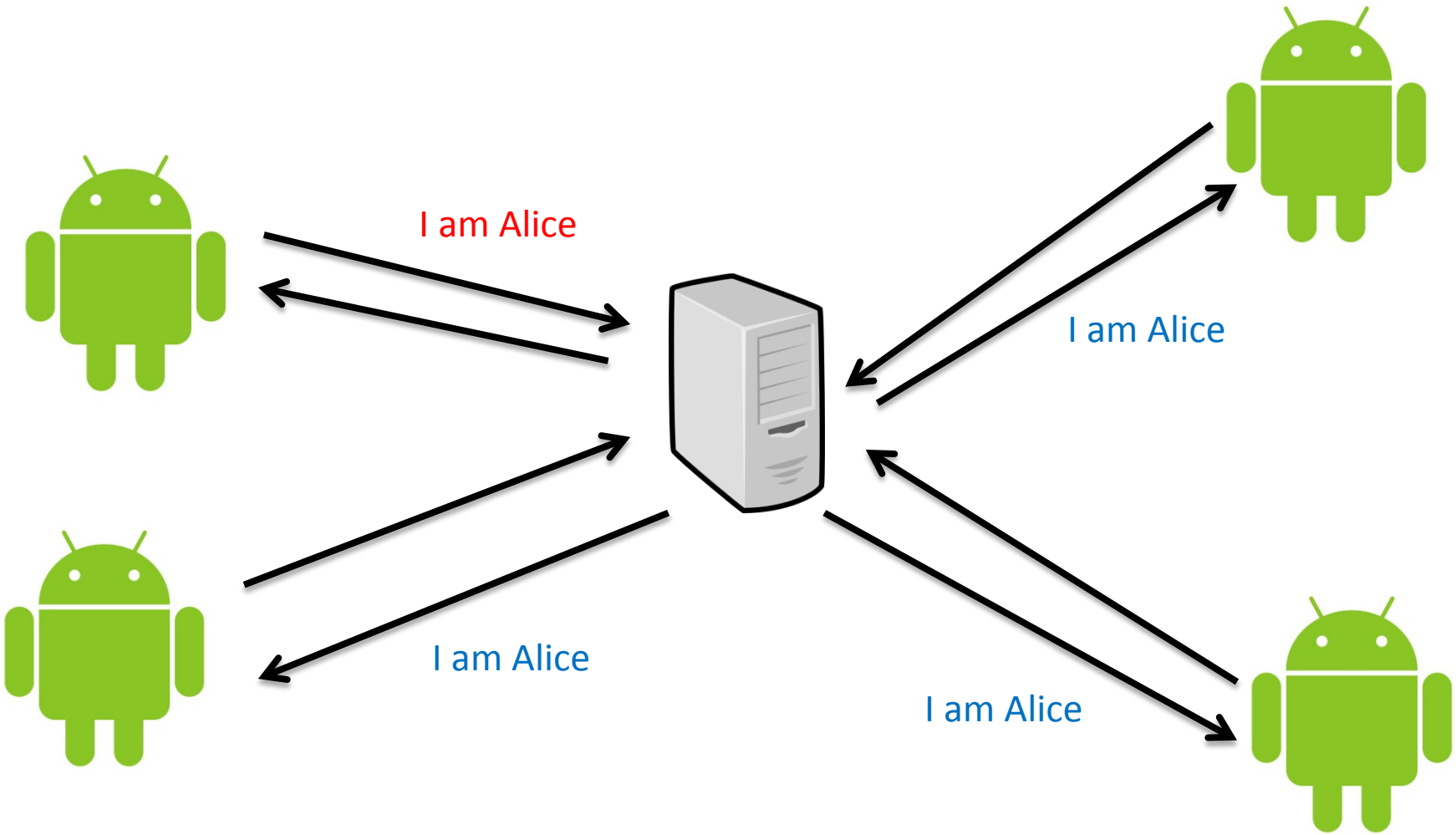
Client public key encrypted by server public key and send to server

Ensure client will not get an “decryptable” packet from any source other than the server

Functions supported

- Login
- Add contact
- Get contact list
- Send message to 1 or many targets
- Receive message
- User id, contact list information store in MySQL, client message wiped every login

Conferencing



Code (total 2178 lines)

- MessageUtil.java (852 lines)
  - Utility tools for both server and client side
- MessageServer.java (445 lines)
  - Server side code
- SecureIM
  - Android project
  - MainActivity.java (150 lines)
  - ContactListMainActivity.java (232 lines)
  - ChatMainActivity.java (191 lines)
  - MessageClient.java (308 lines)

Conclusions (some for answering Dr. Or's questions)

- Performance / Equipment requirements
  - A client holds only one connection even have multiple chatting or conferencing
  - More computation on server than traditional client to client messaging (servers act as trackers)
  - Similar to Facebook, servers do routing jobs, but includes encryption and decryption
  - This model is efficient for client side but heavier workload on server side
  - I think both this efficiency and security provided are people willing to pay for

Conclusions (cont.)

- Major drawbacks
  - User interface is currently over simplified.
  - Need improvement for smoothing user experience
  - Lack of features comparing to instant messenger apps on the market
- To conclude this app
  - A good prototype for demonstrating secure messaging on Android devices