Integrity and Authentication with an HMAC

Tom Ginader

Basics

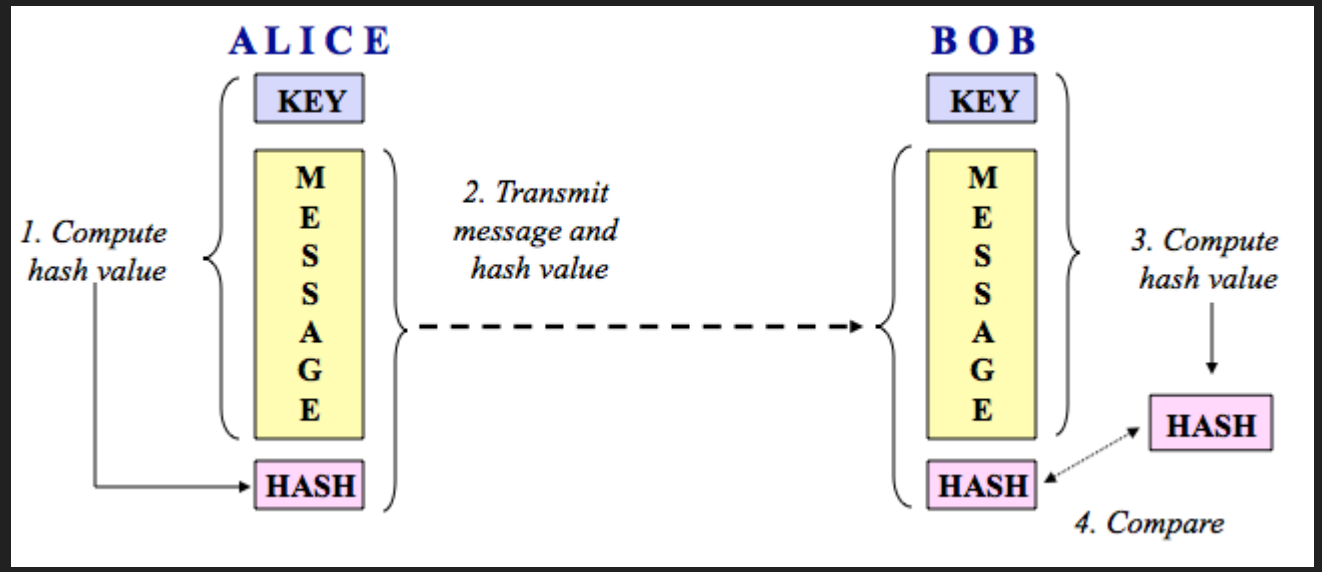
- CIA of security
  - Confidentiality
    - Keeping information secret
    - Encryption
  - **Integrity**
    - Making sure information can't be changed
    - Hashing algorithms
  - Availability
    - Backups, uptime, and disaster recovery
    - RAID

Basics

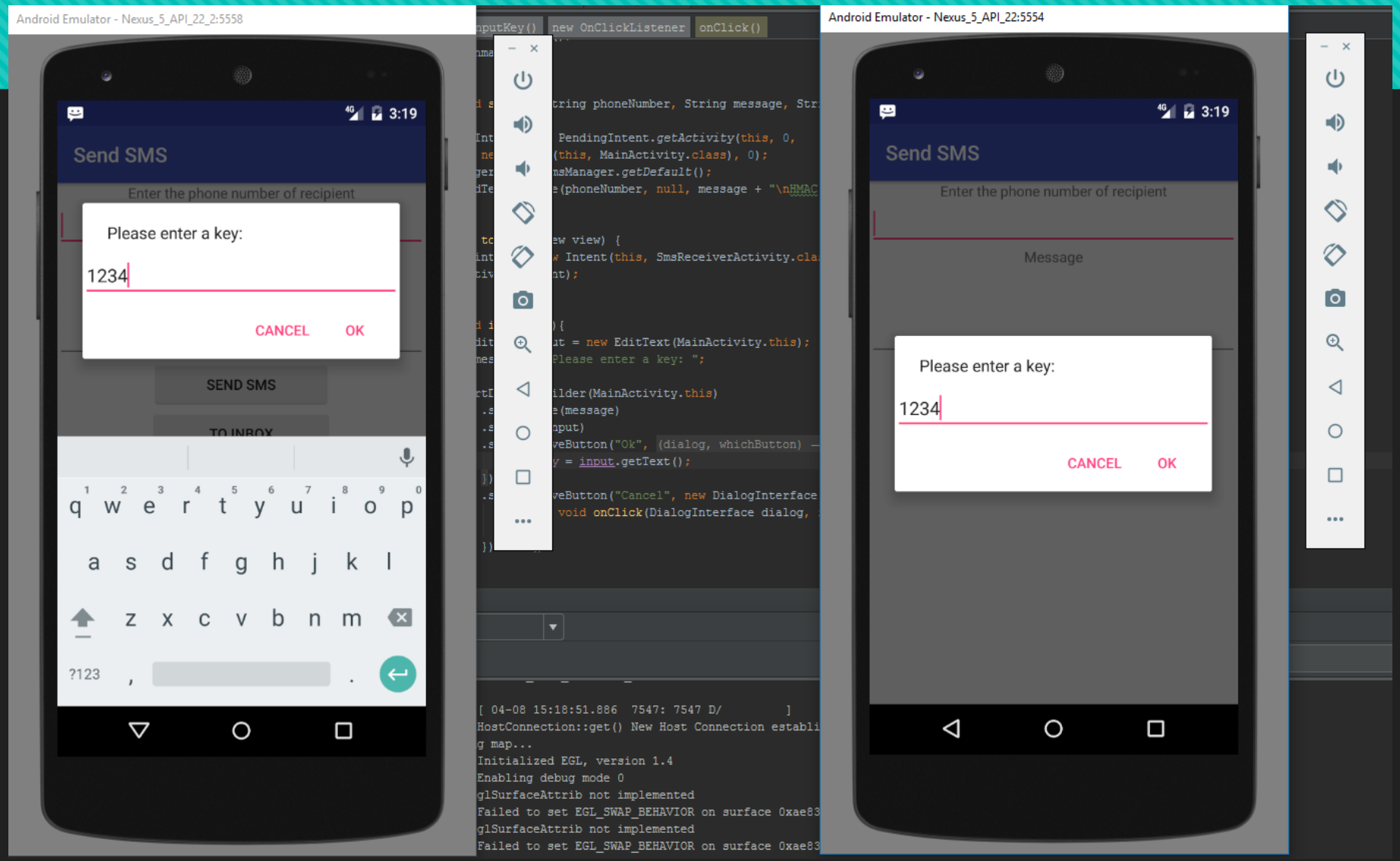
- AAA
  - **Authentication**
    - Proving someone's identity
    - Username and password, shared secret
  - Authorization
    - What someone is allowed to do
    - Access controls, admin privileges
  - Accounting
    - Keeping track of what a user does
    - Logging

Basics

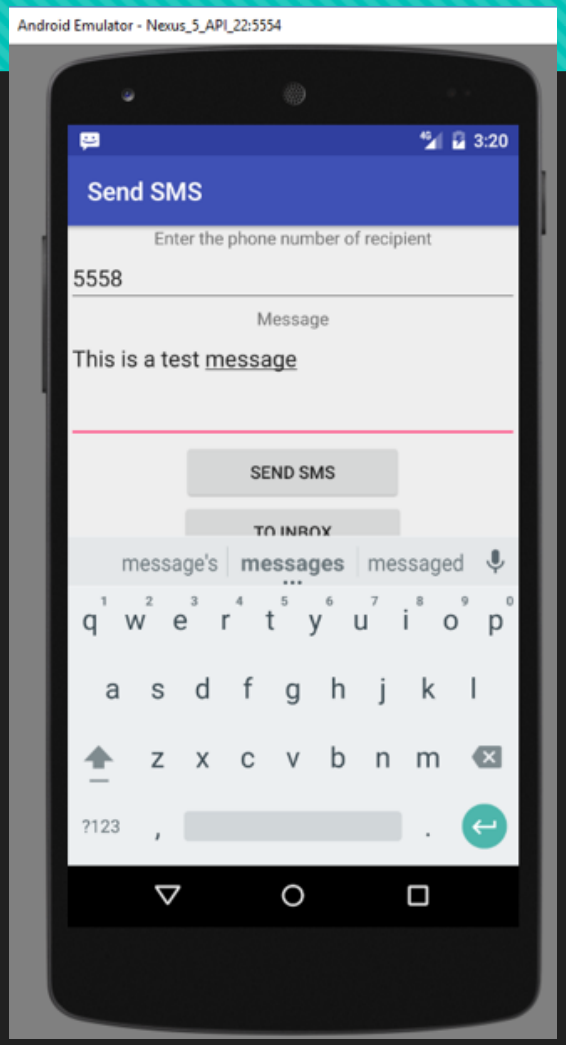
- HMAC
  - Keyed-hash message authentication code



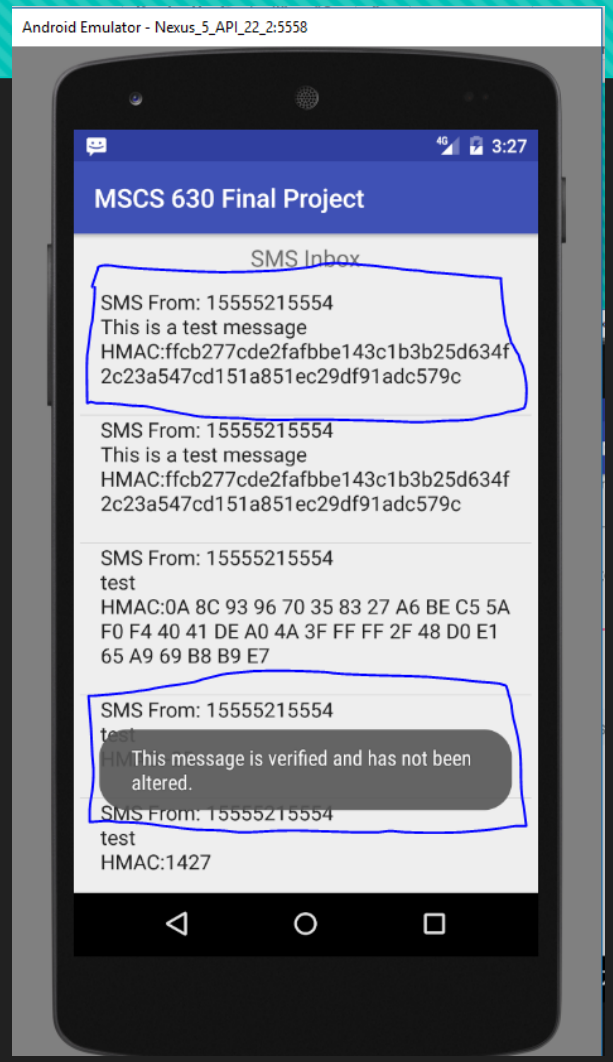
Experiments



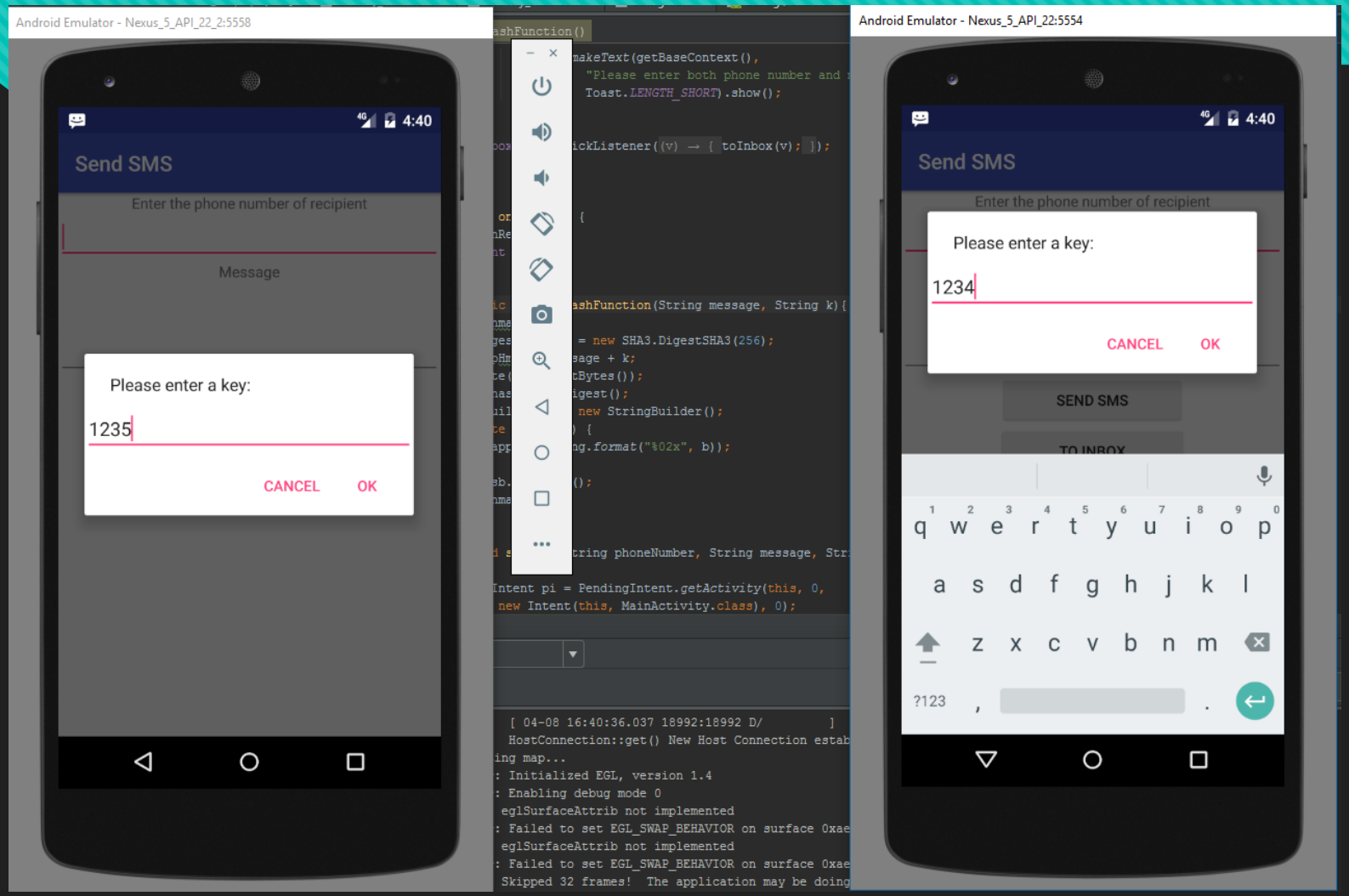
Experiments



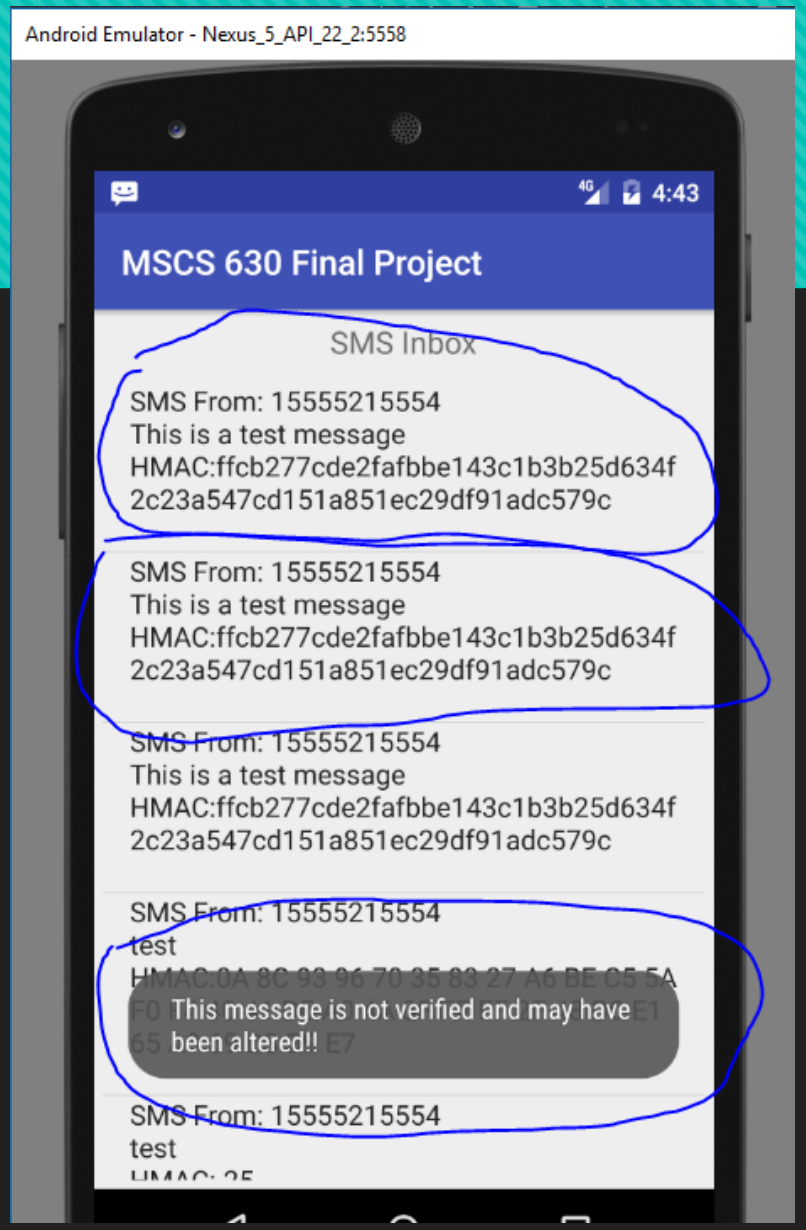
Experiments



Experiments

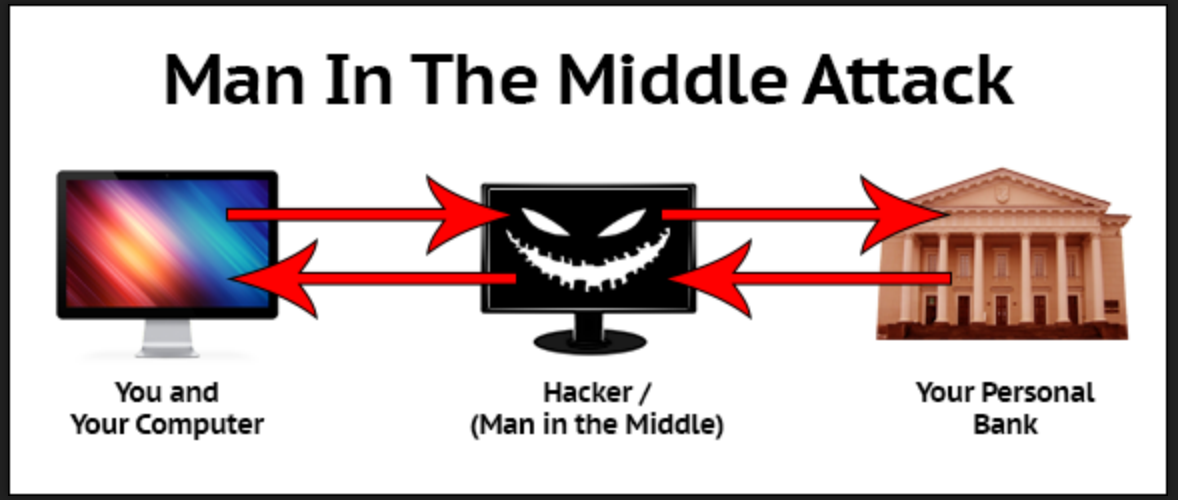


Experiments

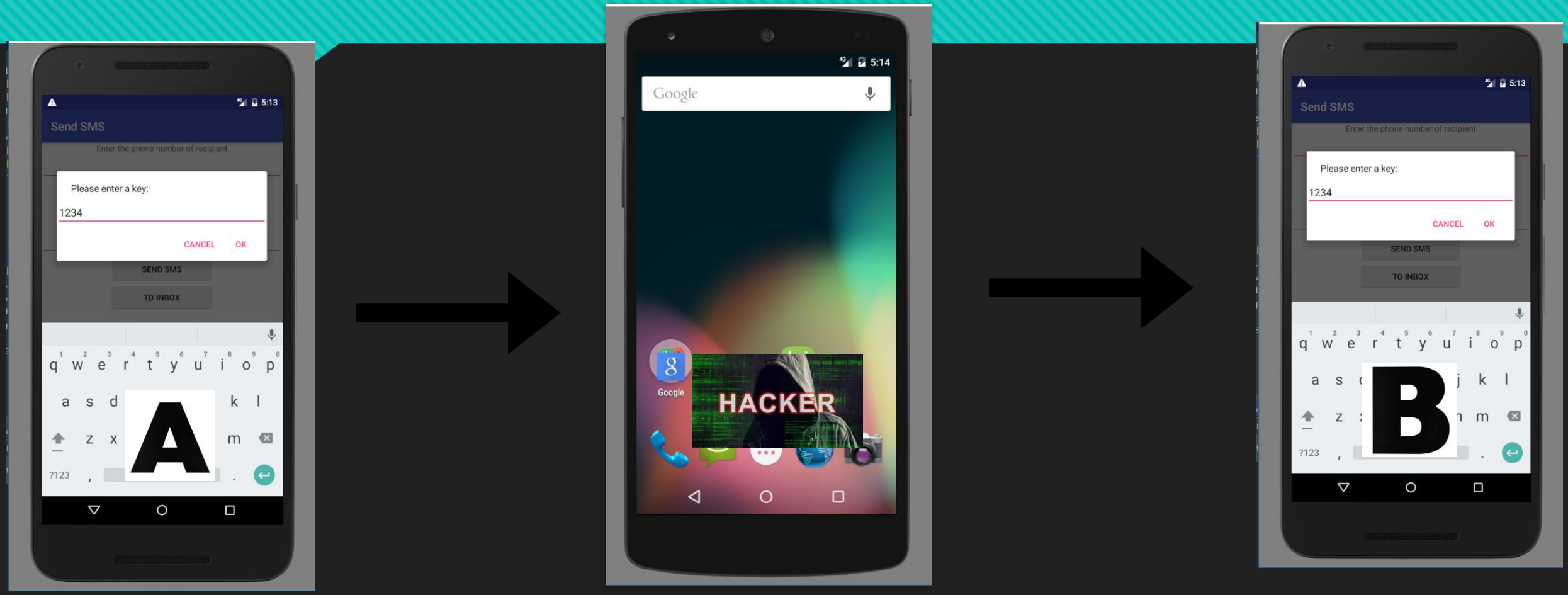


Man in the Middle Attack

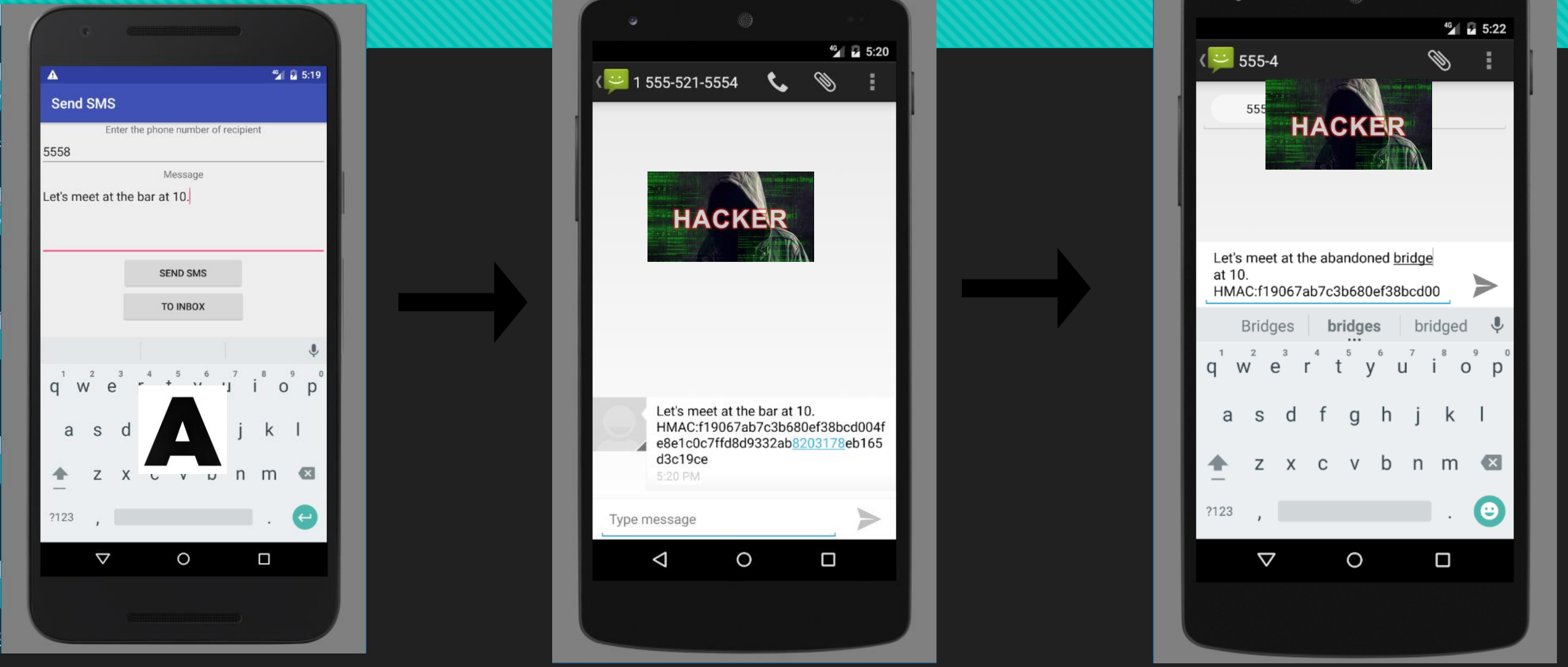
- When two parties or a client/server try to communicate with one another in a legitimate way, but all of the traffic is unknowingly going through a 3rd party



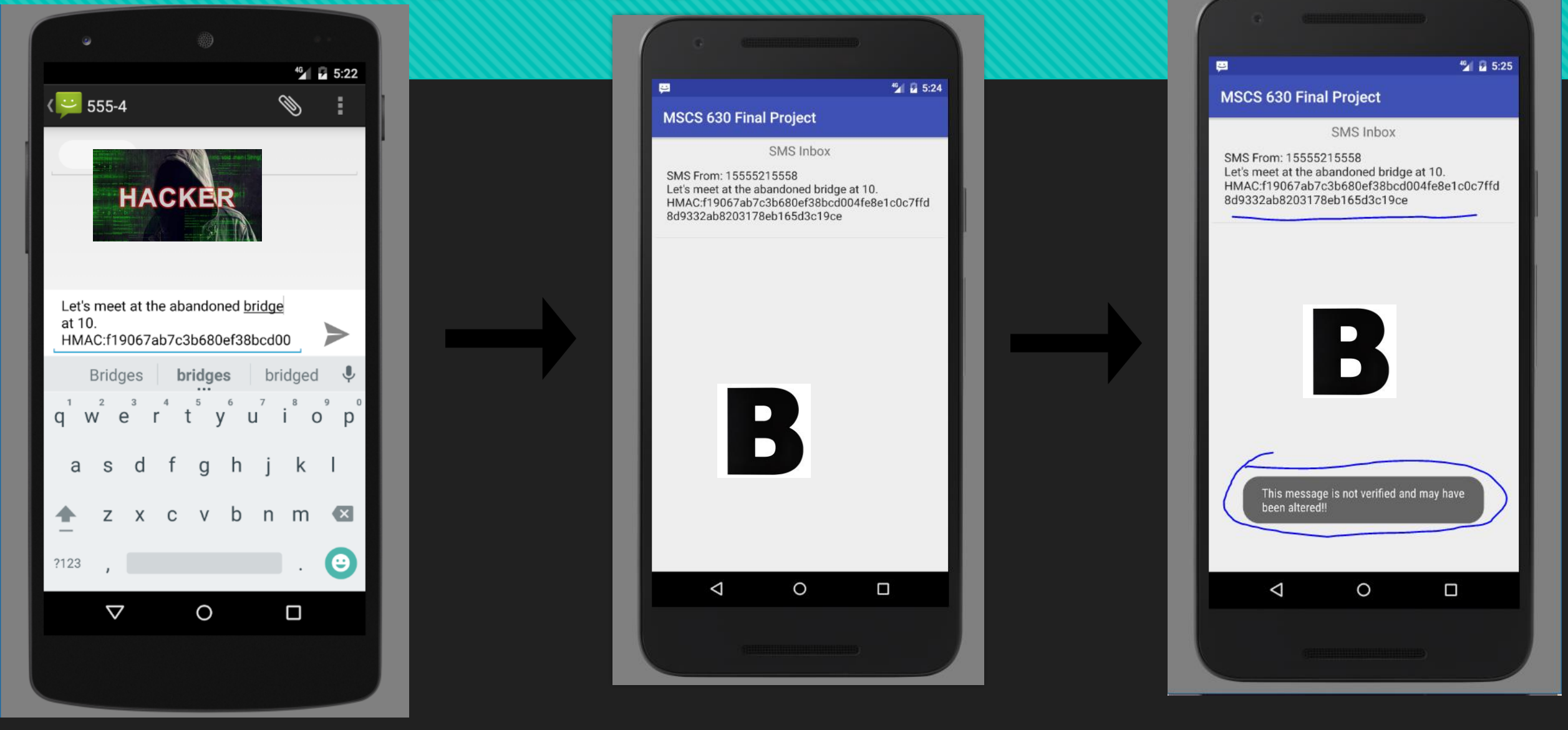
Man in the Middle Attack



Man in the Middle Attack



Man in the Middle Attack



Source Code Algorithm

```
// bouncy castle SHA3
import org.bouncycastle.jcajce.provider.digest.SHA3;
```

```
public static String hashFunction(String message, String k){
    String hmac = "";
    SHA3.DigestSHA3 md = new SHA3.DigestSHA3(256);
    String pHmac = message + k;
    md.update(pHmac.getBytes());
    byte[] hash = md.digest();
    StringBuilder sb = new StringBuilder();
    for (byte b : hash) {
        sb.append(String.format("%02x", b));
    }
    hmac = sb.toString();
    return hmac;
}
```

```
private void sendSMS(String phoneNumber, String message, String hmac)
{
    PendingIntent pi = PendingIntent.getActivity(this, 0,
            new Intent(this, MainActivity.class), 0);
    SmsManager sms = SmsManager.getDefault();
    sms.sendTextMessage(phoneNumber, null, message + "\nHMAC:" + hmac, pi, null);
}
```

```
public void onItemClick(AdapterView<?> parent, View view, int pos, long id) {
    try {
        String[] smsMessages = smsMessagesList.get(pos).split("\n");
        String address = smsMessages[0];
        String smsMessage = "";
        for (int i = 1; i < smsMessages.length; ++i) {
            smsMessage += smsMessages[i];
        }

        String senderHmac = smsMessage.substring(smsMessage.lastIndexOf(":")+1);
        System.out.println(senderHmac);
        String receiverHmac = MainActivity.hashFunction(smsMessage.substring(0,
                                                smsMessage.indexOf(":") - 4),key.toString());
        if(senderHmac.equals(receiverHmac)) {
            Toast.makeText(this, "This message is verified and has not been altered.",
                                                            Toast.LENGTH_SHORT).show();
        }
        else {
            Toast.makeText(this, "This message is not verified and may have been altered!!",
                                                                Toast.LENGTH_SHORT).show();
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
```

Questions?

References

- http://www.garykessler.net/library/crypto.html
- https://www.deepdotweb.com/wp-content/uploads/2016/10/word-image-19.png
- https://www.gohacking.com/wp-content/uploads/2009/01/how-to-become-a-hacker-735x400.jpg