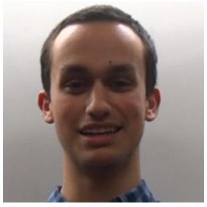
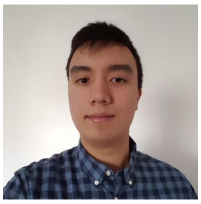
SIRS Alameda Group 12

Smartphone as a security token

Tomás Cunha 81201



Guilherme Santos 81209



Nuno Santos 81703

-Problem (Given the chosen scenario, why is security necessary? What is the main problem being solved? Use around 200 words)

After pairing the phone with the computer, keys are generated and certain files on the computer can be encrypted when the phone is not connected to it. A phone is connected when near to the computer through the exchange of a token. We need to ensure that:
- the paired computer files can only be decrypted with the paired phone's presence, so attackers can't fake the phone's presence.
- the phone only sends the token to the paired computer, so attackers can't fake the computer's presence. Otherwise the fake computer can get the token and use a fake phone to decrypt the files.
- the key exchange (token) is secure because if the attacker is listening and gets the exchanged message by the phone he cannot get the keys to decrypt the files.
- token needs to be unique to avoid replay attacks.
- the key storage needs to be secure so that if the device keeping the keys is compromised the attacker cannot get the keys to decrypt the files.

Problem being solved:
Only those with access to both the paired phone and computer can access the encrypted files.

-Requirements (Which security requirements were identified for the solution? Present as list)

- Confidentiality (only the paired computer can read the token)
- Authentication of origin (the paired computer only accepts tokens sent by the paired phone)
- Fault tolerance (if the computer crashes when the computer is connected, the files cannot be left unprotected)

-Proposed solution (overview with diagram and explanation with around 200 words or less. *Describe basic, intermediate, and advanced versions of the solution*. Be explicit about keys and how they will be distributed.)

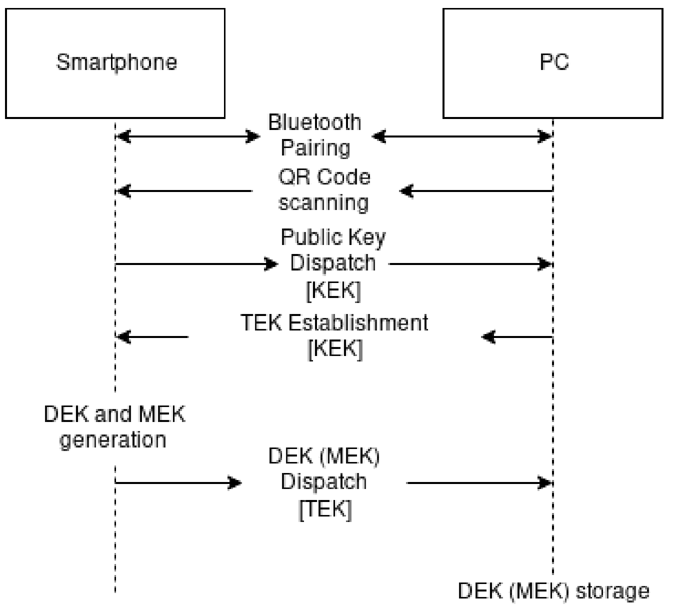
During the initial pairing, the PC creates and displays its RSA public key in a QR code on the screen. The phone scans this key, stores it, and sends its own generated RSA public key encrypted with the PC's public key, which is stored by the PC to be used as a Key Encryption Key. This serves as a protection from man-in-the-middle attacks.
After this verification, the PC generates and shares an AES-256 session key. From this point on, this key is used to secure the communication channel. The phone generates two other AES-256 keys, one to be used for file encryption, and one to encrypt that key before sending it to the PC. This key is stored, and the pairing process is completed.
After pairing, the phone can connect to the PC, who (after the initial session key exchange) sends the encrypted File Encryption key, which is securely stored on the PC. The phone decrypts it and sends it back to the PC, who uses it to decrypt the files. The phone then has to keep sending heartbeats (which include a nonce to avoid replay attacks) so that the PC can know the phone is in range. After the phone is disconnected, since the PC no longer receives heartbeats, it encrypts the files again and deletes the FE key from storage, keeping only its encrypted copy. The PC maintains a log of the operations it performs. In case of a crash, after recovery it will encrypt all the files with the decrypted FE key, then deletes the key and waits for the phone to be reconnected.

Basic: Initial implementation without fault tolerance and secure channels.

Intermediate: Implementation of confidentiality with secure channels.

Advanced: Implementation of fault tolerance.

Pairing



KEK - Key Encryption Keys - Asymmetric Keys

TEK - Traffic Encryption Key - Session Key

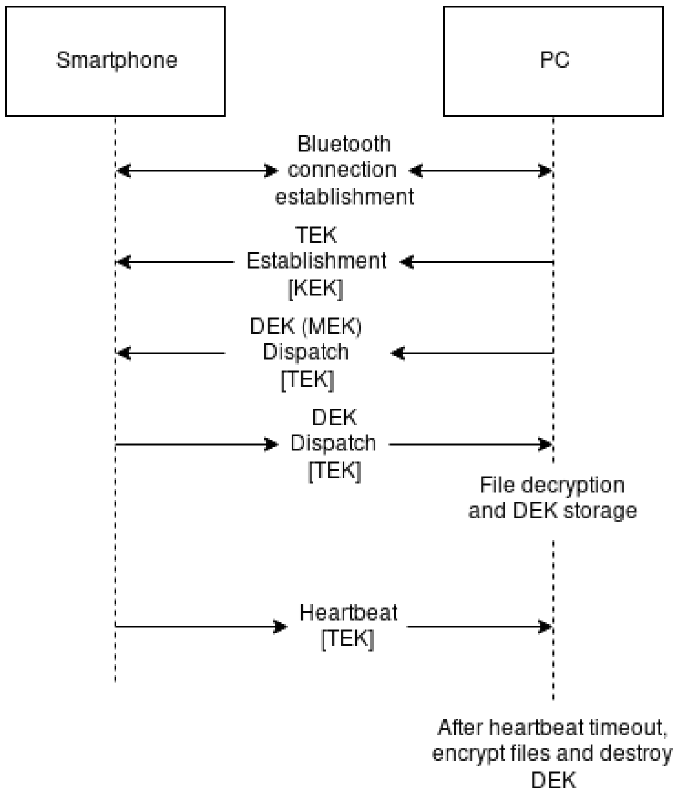
DEK - Data Encryption Key - Used to encrypt/decrypt the files on the computer

MEK - Master Encryption Key - Used to encrypt the DEK

<message> [*key initials*] - This message is encrypted using *key*

<file> (*key initials*) - This file is encrypted using *key*

Connection


Smartphone
PC
Bluetooth connection establishment
TEK Establishment [KEK]
DEK (MEK) Dispatch [TEK]
DEK Dispatch [TEK]
File decryption and DEK storage
Heartbeat [TEK]
After heartbeat timeout, encrypt files and destroy DEK

-Tool references (libraries, etc. that will be used in the project. State if tool has been found/installed/tested/well-tested at the time of proposal)

Well-tested: Git, Java, Android Studio, Pycharm, Python 3, Java Crypto, Bouncy Castle, python-qrcode
Installed: pycrypto, pybluez, Android Bluetooth, zxing

-Work plan (table containing one row per week until the submission date; and one column per group member with expected activities for the given week; some cells may be blank because of other courses. State clearly when basic, intermediate and advanced versions are expected to be achieved)

	Tomás	Guilherme	Nuno
30 oct – 5 nov	Start Python App	Start Android App	Start Android App
6 nov – 12 nov	Android Bluetooth	Android Bluetooth	Python Bluetooth
13 nov – 19 nov	Complete Basic	Complete Basic	Complete Basic
20 nov – 26 nov	Python Cipher	Python Cipher	Android Cipher
27 nov – 3 dec	Complete intermediate and start fault tolerance	Complete intermediate and start fault tolerance	Complete intermediate and start fault tolerance
4 dec – 7 dec	Complete advanced	Complete advanced	Complete advanced

Basic :	19/11/2017
Intermediate :	29/11/2017
Advanced :	07/12/2017