Name: Omar Mohamed Mohamed AbdElkareem Sec: 2 B.N: 8

Security Project 01 – Sending Secure E-mails

Project Modules:

- **Send module**:
  - **getPublicKey**: generate public & private key for the sender & receiver if there isn't already
  - **generateKey**: generate random 56-bit session key
  - **encryptKey**: encrypt the session key with the public key using RSA
  - **encryptMessage**: encrypt the text message with the session key using DES-CBC
  - **sendEmail**: establish connection to gmail to login to the sender email address and send the message (composed of the length of the encrypted session key + the integer value of it + the length of the encrypted message + the integer value of it) to the receiver email address with the subject *"Security Project 01 - Omar AbdElkareem"*
  - **main:** get the emails and the sender password through console or a file, then call the corresponding functions.
- **Receive module**:
  - **getPrivateKey**: generate public & private key for the sender & receiver if there isn't already
  - **decryptKey**: decrypt the session key with the public key using RSA
  - **decryptMessage**: decrypt the text message with the decrypted session key using DES-CBC
  - **extractMessage**: construct the encrypted message and the encrypted session key from the message text
  - **receiveEmail**: establish connection to login to the receiver mail address and get the latest message with the subject *"Security Project 01 - Omar AbdElkareem"*

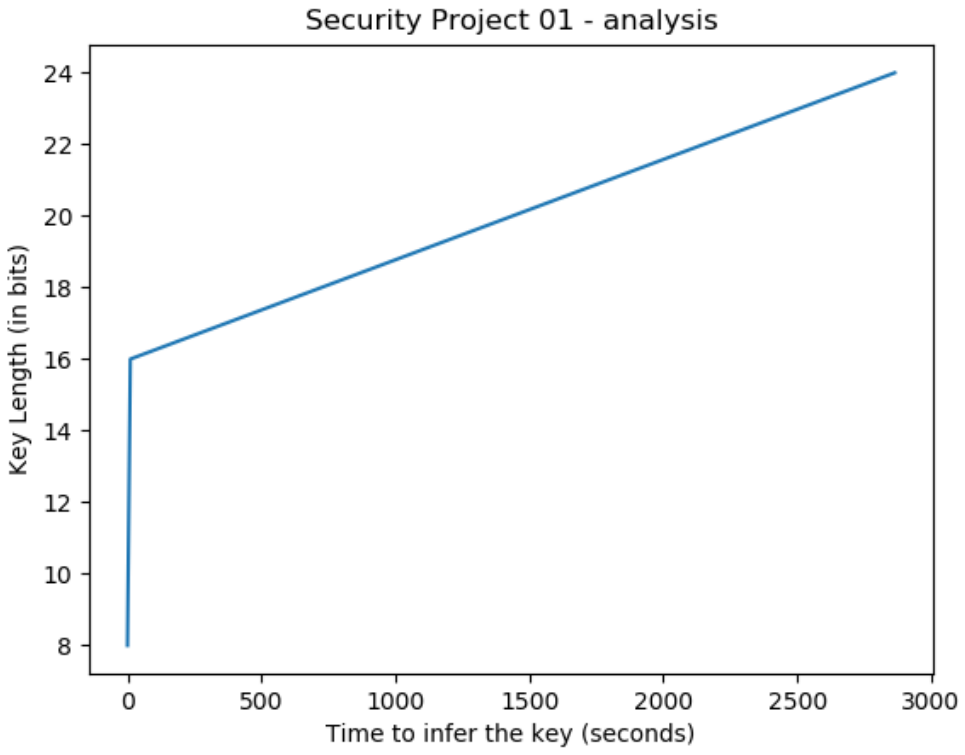
Name: Omar Mohamed Mohamed AbdElkareem Sec: 2 B.N: 8

Security Project 01 – Sending Secure E-mails

  - **main**: get the emails and the sender password through console or a file, then call the corresponding functions and print the decrypted message.
- **Analysis module**:
  - **encryptMessage**: the same function as in Send module
  - **attack**: brute force attack on a given plain text and cipher text
  - **main**: generating random session keys with sizes = 8, 16, 24, … 56 bits, then using it to encrypt the plain text *"Security Project 01 - analysis - Omar AbdElkareem"* to test the brute force attack, calculate the taken time to break the key, plot a plot indicating the relation between key length in bits vs taken time to break the key in seconds.

Analysis results:

- The time to infer the key is growing exponentially with the key length
- The maximum key length I could break is 24 bits (in 2863 seconds)
- The plot:



Name: Omar Mohamed Mohamed AbdElkareem **Sec: 2** **B.N: 8**

Security Project 01 – Sending Secure E-mails

Important Notes:

- To be able to establish a connection with gmail account, you must first follow the instructions in this link: Python Quickstart - Google Drive API
- For Gmail API to be able to access the gmail accounts , you must allow this less secure app access option from here: Less Secure apps access