Security Project 1

Sending Secure E-mails

Spring 2019

Team Member:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Email | Section | B.N. |
| Omar Osama | [omarosamasobeih@yahoo.com](mailto:omarosamasobeih@yahoo.com) | 1 | 32 |
| Mina Magdy | mina\_mego5@yahoo.com | 2 | 23 |

# Introduction:

Nowadays, sending secure e-mails has a primary concern due to the extensive usage of e-mails, so they must be sent in a secure way. Sending e-mails requires achieving both confidentiality and authentication since you do not want your inbox messages to be read by other people (confidentiality), and you want to verify the sender’s e-mail address for any e-mail you got (authentication).

There are currently two actively proposed methods for providing these security services Secure/Multipurpose Internet Mail Extension (S/MIME) and Pretty Good Privacy (PGP).

We developed a python application that sends and receives secure encrypted e-mails implementing PGP protocol.

# Project Modules:

1. Sender: responsible for encrypting and sending messages.
2. Receiver: responsible for receiving and decrypting messages.
3. Attacker: responsible for brute force attack to infer the key.

# Sender Module:

1. Generate DES Key:

Randomly generate shared key given key length, maintaining that least significant bit in key’s bytes is zero since it has no effect on the cipher.

1. Encrypt DES key using receiver’s public key using RSA:

Encrypted key = (key ^ e) % n.

Where (e, n) public key pair of the receiver.

1. Convert the key to byte stream with constant defined length and write it in the start of the mail.
2. Encrypt message using DES: (used pyDes library)

Encrypt using the key generated in step 1 the message sent by user.

1. Concatenate the cipher byte stream with the key byte stream.
2. Send the resulting byte stream using mail API.

Receiver Module:

1. Receive mail using mail API.
2. Extract encrypted DES key.
3. Decrypt key using RSA private key:

Key = (encrypted key ^ d) % n, where (d, n) are RSA private key pair.

1. Extract message.
2. Decrypt message using DES key.

Attacker Module:

1. Iterate on all possible DES keys that output different cipher for a given plain text.
2. Encrypt given plain text using each key using DES.
3. Compare generated cipher with the given cipher.
4. If both ciphers match then the key is broken.
5. Repeat same steps on different pairs generated by different key length to analyze brute force attack method.
6. Measure time of breaking the key for each key length.
7. Plot the time with respect to the key length.
8. The time to break key grows exponentially with the key length. 