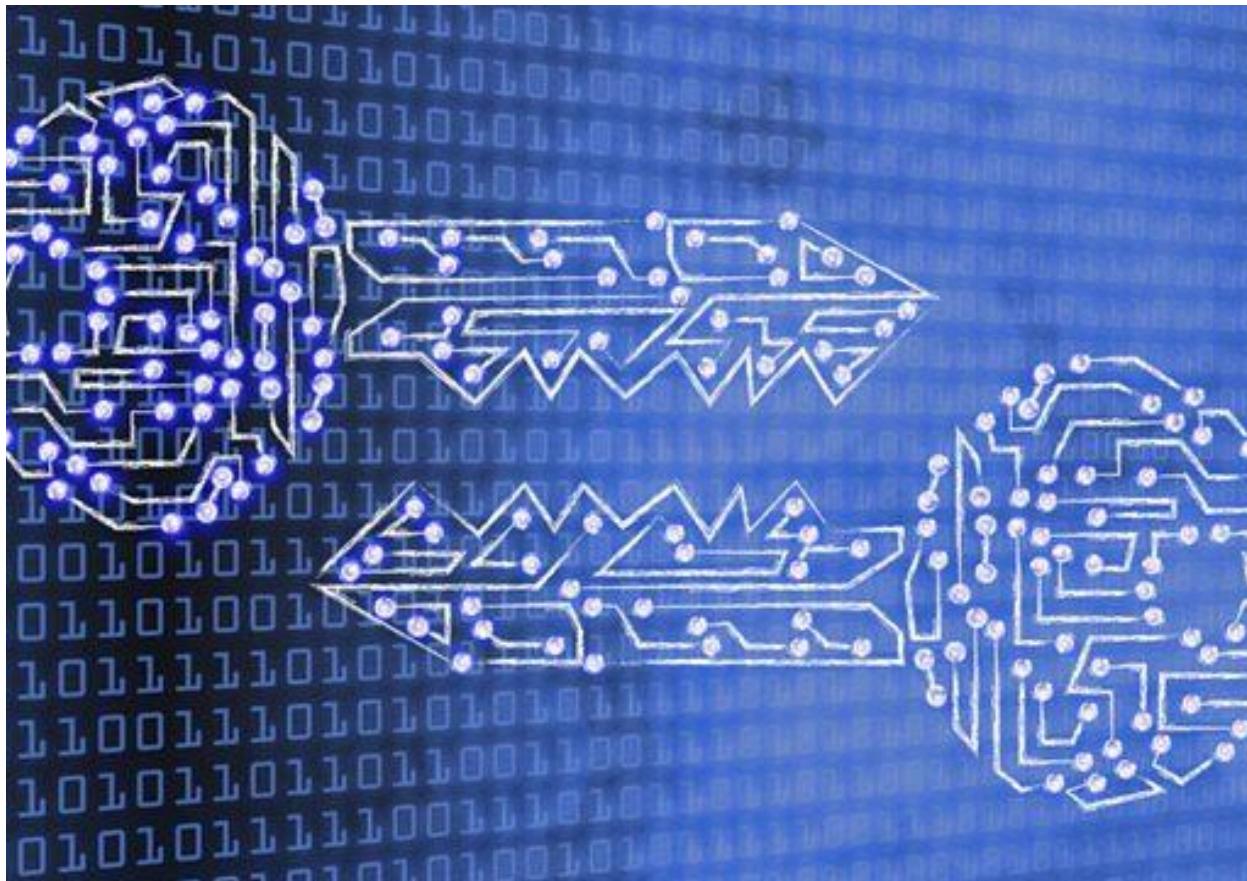


Cryptography

Defensive Security 1 - Assignment 3



by

Uthaya Krishnan

Sheridan College

The Basics:

Public-key cryptography :

It uses a pair of keys - a public key and a private key. The public key can be shared openly, while the private key is kept secret.

In email encryption, the sender uses the recipient's public key to encrypt the email. Only the recipient, who possesses the corresponding private key, can decrypt the message. This ensures that even if the encrypted email is intercepted, it cannot be read without the private key. This method of encryption provides confidentiality and ensures that only the intended recipient can access the message content .

Private-Key Cryptography in Email Security

Private-key cryptography:

It is also known as symmetric cryptography. It uses a single key for both encryption and decryption. This key must be shared between the sender and recipient securely. While it is efficient for encrypting and decrypting large amounts of data, its main drawback is the challenge of securely sharing the key.

In email communications, private-key cryptography is less common due to the difficulties in key distribution and management compared to public-key cryptography .

Public Key Infrastructure (PKI) in Email Encryption

Public Key Infrastructure (PKI) :

It is a framework that manages public-key encryption and digital certificates.

In email encryption, PKI facilitates the creation, distribution, and management of public and private keys. PKI involves a Certificate Authority (CA) that issues digital certificates verifying the ownership of public keys. When sending an encrypted email, the sender can obtain the recipient's public key from their digital certificate, ensuring the key's authenticity. PKI thus provides a scalable and secure way to manage keys and authenticate users in email encryption

Selecting Tools

1. GPG (GNU Privacy Guard)

GPG is an open-source implementation of the OpenPGP standard for encrypting and signing data. Choosing this tool because of following features:

Security: GPG provides robust encryption using RSA and other algorithms, ensuring the confidentiality and integrity of emails.

Open Source: Being open-source, GPG is freely available, regularly audited, and maintained by a strong community, ensuring transparency and security.

Compatibility: GPG is compatible with various operating systems and integrates with numerous applications, making it versatile for different use cases.

Proven Track Record: Widely trusted and used by security professionals worldwide, GPG has a long history of reliability and effectiveness in secure communications.

2. Thunderbird with OpenPGP

Thunderbird is a free, open-source, cross-platform email client developed by Mozilla.

OpenPGP is a standard for encrypting and signing data used by GPG and other encryption tools.

User-Friendly: Thunderbird offers an intuitive interface, making it easy for users to manage their emails and integrate encryption tools.

Customizable: With a wide range of add-ons and extensions like Enigmail, Thunderbird can be customized to enhance its functionality, including encryption.

Secure: Thunderbird provides robust security features, including spam filters and phishing protection, and supports integration with GPG for encrypted communications.

Open Source: As an open-source application, Thunderbird benefits from community contributions and regular updates, ensuring it remains secure and up-to-date.

Standardization: OpenPGP is an established standard for encryption and digital signatures, ensuring interoperability between different applications and systems.

Security: It provides strong encryption and authentication mechanisms to protect data integrity and confidentiality.

Generating Keys

Generate a pair of public and private keys using the selected encryption tool.

Installing gnupg tool

```
(kaliuser㉿kali)-[~]
$ sudo apt-get install gnupg
[sudo] password for kaliuser:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bsd-mailx dirmngr exim4-base exim4-config exim4-daemon-light gnupg-l10n gnupg-utils gnutls-bin gpg gpg-agent
  gpg-wks-client gpg-wks-server gpgconf gpgsm gpgv libgnutls-dane0t64 libgnutls30t64 libhogweed6t64 liblockfile1
  libnettle8t64
Suggested packages:
  exim4-doc-html | exim4-doc-info eximon4 SPF-tools-perl
The following NEW packages will be installed:
  bsd-mailx exim4-base exim4-config exim4-daemon-light liblockfile1
The following packages will be upgraded:
  dirmngr gnupg gnupg-l10n gnupg-utils gnutls-bin gpg gpg-agent gpg-wks-client gpg-wks-server gpgconf gpgsm gpgv
  libgnutls-dane0t64 libgnutls30t64 libhogweed6t64 libnettle8t64
16 upgraded, 5 newly installed, 0 to remove and 701 not upgraded.
Need to get 8800 kB of archives.
After this operation, 69.6 kB disk space will be freed.
Do you want to continue? [Y/n] y
Get:1 http://mirror.0xem.ma/kali kali-rolling/main amd64 gpg-wks-client amd64 2.2.43-7 [100 kB]
Get:11 http://mirror.0xem.ma/kali kali-rolling/main amd64 exim4-config all 4.98-1 [248 kB]
Get:3 http://mirrors.jevincanders.net/kali kali-rolling/main amd64 gnupg-l10n all 2.2.43-7 [701 kB]
Get:4 http://mirror.accuris.ca/kali kali-rolling/main amd64 gpg-wks-server amd64 2.2.43-7 [90.5 kB]
Get:6 http://kali.download/kali kali-rolling/main amd64 gpg-agent amd64 2.2.43-7 [248 kB]
Get:5 http://mirror.accuris.ca/kali kali-rolling/main amd64 gnupg-utils amd64 2.2.43-7 [500 kB]
Get:8 http://kali.download/kali kali-rolling/main amd64 gpgconf amd64 2.2.43-7 [119 kB]
```

Generating key for Sender using GPG

```
(kaliuser㉿kali)-[~]
$ gpg --full-generate-key
gpg (GnuPG) 2.2.43; Copyright (C) 2023 g10 Code GmbH
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

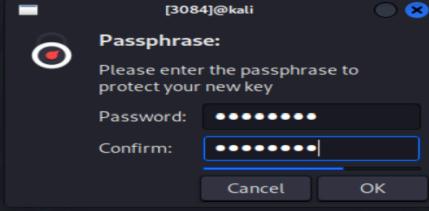
Please select what kind of key you want:
 (1) RSA and RSA (default)
 (2) DSA and Elgamal
 (3) DSA (sign only)
 (4) RSA (sign only)
 (14) Existing key from card
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (3072) 2048
Requested keysize is 2048 bits
Please specify how long the key should be valid.
  0 = key does not expire
  <n> = key expires in n days
  <n>w = key expires in n weeks
  <n>m = key expires in n months
  <n>y = key expires in n years
Key is valid for? (0) 2
Key expires at Thu Aug  8 17:03:30 2024 EDT
Is this correct? (y/N) y

GnuPG needs to construct a user ID to identify your key.

Real name: udhai
Email address: udhayakrishnan2019@gmail.com
Comment: its me
You selected this USER-ID:
 "udhai (its me) <udhayakrishnan2019@gmail.com>"

Change (N)ame, (C)omment, (E)mail or (O)key/(Q)uit? o
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.

```



Exporting public key and private key for sender

```
GnuPG needs to construct a user ID to identify your key.

Real name: udhai
Email address: udhayakrishnan2019@gmail.com
Comment: its me
You selected this USER-ID:
  "udhai (its me) <udhayakrishnan2019@gmail.com>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit?
We need to generate a lot of random bytes. It is a
some other action (type on the keyboard, move the m
disks) during the prime generation; this gives the
generator a better chance to gain enough entropy.
We need to generate a lot of random bytes. It is a
some other action (type on the keyboard, move the m
disks) during the prime generation; this gives the
generator a better chance to gain enough entropy.
gpg: revocation certificate stored as '/home/kaliuser/.gnupg/openpgp-revocs.d/E92AD62CE95BA39C61FE57968D3BA07447AE8D1A.rev'
public and secret key created and signed.

pub    rsa2048 2024-08-06 [SC] [expires: 2024-08-08]
      E92AD62CE95BA39C61FE57968D3BA07447AE8D1A
uid            udhai (its me) <udhayakrishnan2019@gmail.com>
sub    rsa2048 2024-08-06 [E] [expires: 2024-08-08]

(kaliuser㉿kali)-[~]
$ gpg --export -a udhai > publickey.asc

(kaliuser㉿kali)-[~]
$ gpg --export-secret-keys -a udhai > privatekey.asc
```

```
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: revocation certificate stored as '/home/kaliuser/.gnupg/openpgp-revocs.d/E92AD62CE95BA39C61FE57968D3BA07447AE8D1A.rev'
public and secret key created and signed.

pub    rsa2048 2024-08-06 [SC] [expires: 2024-08-08]
      E92AD62CE95BA39C61FE57968D3BA07447AE8D1A
uid            udhai (its me) <udhayakrishnan2019@gmail.com>
sub    rsa2048 2024-08-06 [E] [expires: 2024-08-08]
```

Generating key for Receiver using GPG

```
(14) EXISTING KEY FROM CARD          krish udhai
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (3072) 2048   me
Requested keysize is 2048 bits
Please specify how long the key should be valid.
  0 = key does not expire
  <n> = key expires in n days
  <n>w = key expires in n weeks
  <n>m = key expires in n months
  <n>y = key expires in n years
Key is valid for? (0) 2
Key expires at Thu Aug  8 19:21:14 2024 EDT
Is this correct? (y/N) y
Freelancer.com

GnuPG needs to construct a user ID to identify your key.
  Real name: krish
  Email address: pandiarajan.srinivasan1@gmail.com
  Comment: its me pan
  You selected this USER-ID:
    "krish (its me pan) <pandiarajan.srinivasan1@gmail.com>"  (no subject)

  Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? o
  We need to generate a lot of random bytes. It is a good idea to perform
  some other action (type on the keyboard, move the mouse, utilize the
  disks) during the prime generation; this gives the random number
  generator a better chance to gain enough entropy.
```

```

We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number ...
generator a better chance to gain enough entropy.
gpg: revocation certificate stored as '/home/kaliuser/.gnupg/openpgp-revocs.d/A1FF78DA726149F8C82D3938BFEFBC34ECE9E349.rev'
public and secret key created and signed.

pub rsa2048 2024-08-06 [SC] [expires: 2024-08-08] (no subject)
      A1FF78DA726149F8C82D3938BFEFBC34ECE9E349
uid          krish (its me pan) <pandiarajan.srinivasan1@gmail.com>
sub rsa2048 2024-08-06 [E] [expires: 2024-08-08]

(kaliuser㉿kali)-[~] $ 

```

Security alert - A new sign-in on Linux udhayakrishnan2019@gmail.com We

Exporting public and private keys for receiver

```

GnuPG needs to construct a user ID to identify your key.
Real name: krish
Email address: pandiarajan.srinivasan1@gmail.com
Comment: its me pan
You selected this USER-ID:
  "krish (its me pan) <pandiarajan.srinivasan1@gmail.com>"

Change (N)ame, (C)oмment, (E)mail or (O)kay/(Q)uit?
We need to generate a lot of random bytes. It is a
some other action (type on the keyboard, move the m
disks) during the prime generation; this gives the
generator a better chance to gain enough entropy.
We need to generate a lot of random bytes. It is a
some other action (type on the keyboard, move the m
disks) during the prime generation; this gives the
generator a better chance to gain enough entropy.
gpg: revocation certificate stored as '/home/kaliuser/.gnupg/openpgp-revocs.d/A1FF78DA726149F8C82D3938BFEFBC34ECE9E349.rev'
public and secret key created and signed.

pub rsa2048 2024-08-06 [SC] [expires: 2024-08-08]
      A1FF78DA726149F8C82D3938BFEFBC34ECE9E349
uid          krish (its me pan) <pandiarajan.srinivasan1@gmail.com>
sub rsa2048 2024-08-06 [E] [expires: 2024-08-08] (no subject)

(kaliuser㉿kali)-[~] $ gpg --export -a krish > publickey.asc
(kaliuser㉿kali)-[~] $ gpg --export-secret-keys -a krish > privatekey.asc

```

Passphrase:

Please enter the passphrase to export the OpenPGP secret key:
 "krish (its me pan) <pandiarajan.srinivasan1@gmail.com>"
 2048-bit RSA key, ID BFEFBC34ECE9E349,
 created 2024-08-06.

Password: |

Cancel OK

Security alert - A new sign-in on Linux udhayakrishnan2019@gmail.com We noticed a new sign-in ...

Pair of public and private keys for sender and receiver

```

(kaliuser㉿kali)-[~] $ gpg --list-keys
gpg: checking the trustdb
gpg: marginals needed: 3  completes needed: 1  trust model: pgp
gpg: depth: 0  valid: 2  signed: 0  trust: 0-, 0q, 0n, 0m, 0f, 2u
gpg: next trustdb check due at 2024-08-08
/home/kaliuser/.gnupg/pubring.kbx

pub    rsa2048 2024-08-06 [SC] [expires: 2024-08-08]
      E92AD62CE95BA39C61FE57968D3BA07447AE8D1A
uid          [ultimate] udhai (its me) <udhayakrishnan2019@gmail.com>
sub    rsa2048 2024-08-06 [E] [expires: 2024-08-08]

pub    rsa2048 2024-08-06 [SC] [expires: 2024-08-08]
      A1FF78DA726149F8C82D3938BFEFBC34ECE9E349
uid          [ultimate] krish (its me pan) <pandiarajan.srinivasan1@gmail.com>
sub    rsa2048 2024-08-06 [E] [expires: 2024-08-08]

(kaliuser㉿kali)-[~] $ 

```

Configuring Email Accounts

Configuring Sender Mail Account in Thunderbird

Set Up Your Existing Email Address

To use your current email address fill in your credentials.
Thunderbird will automatically search for a working and recommended server configuration.

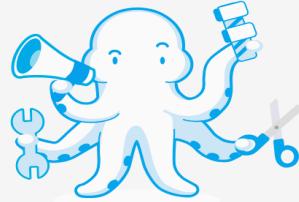
Your full name
udhai krish

Email address
udhayakrishnan2019@gmail.com

Password
••••••••••
 Remember password

Configure manually Cancel Continue

Your credentials will only be stored locally on your computer.



 Account successfully created

You can now use this account with Thunderbird.
You can improve the experience by connecting related services and configuring advanced account settings.

✉ udhai krish udhayakrishnan2019@gmail.com IMAP


Cloud services connected to your account:

-  Account settings
-  End-to-end encryption
-  Add a signature
-  Download dictionaries

Connect your linked services

Thunderbird detected other services linked to your email account.

Not sure about your next steps?
[Getting started](#) - [Support forum](#) - [Privacy policy](#)

Address Books

Thunderbird found one address book linked to your email account.

CARDDAV **Address Book** Connect

Configuring Receiver Mail Account in ThunderBird

Your full name
krish udhai

Email address
pandiarajan.srinivasan1@gmail.com

Password

Remember password

✓ Configuration found in Mozilla ISP database.

Available configurations

IMAP
Keep your folders and emails synced on your server

Incoming IMAP SSL/TLS
imap.gmail.com

Outgoing SMTP SSL/TLS
smtp.gmail.com

Username
pandiarajan.srinivasan1@gmail.com

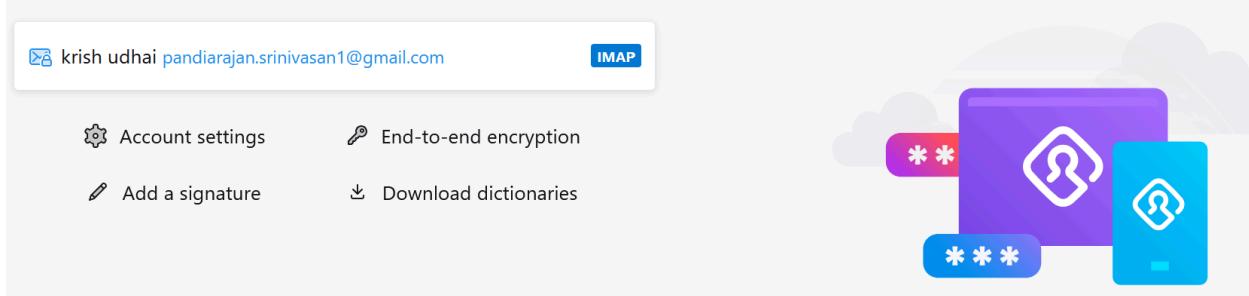
POP3
Keep your folders and emails on your computer

[Configure manually](#) [Cancel](#) [Done](#)

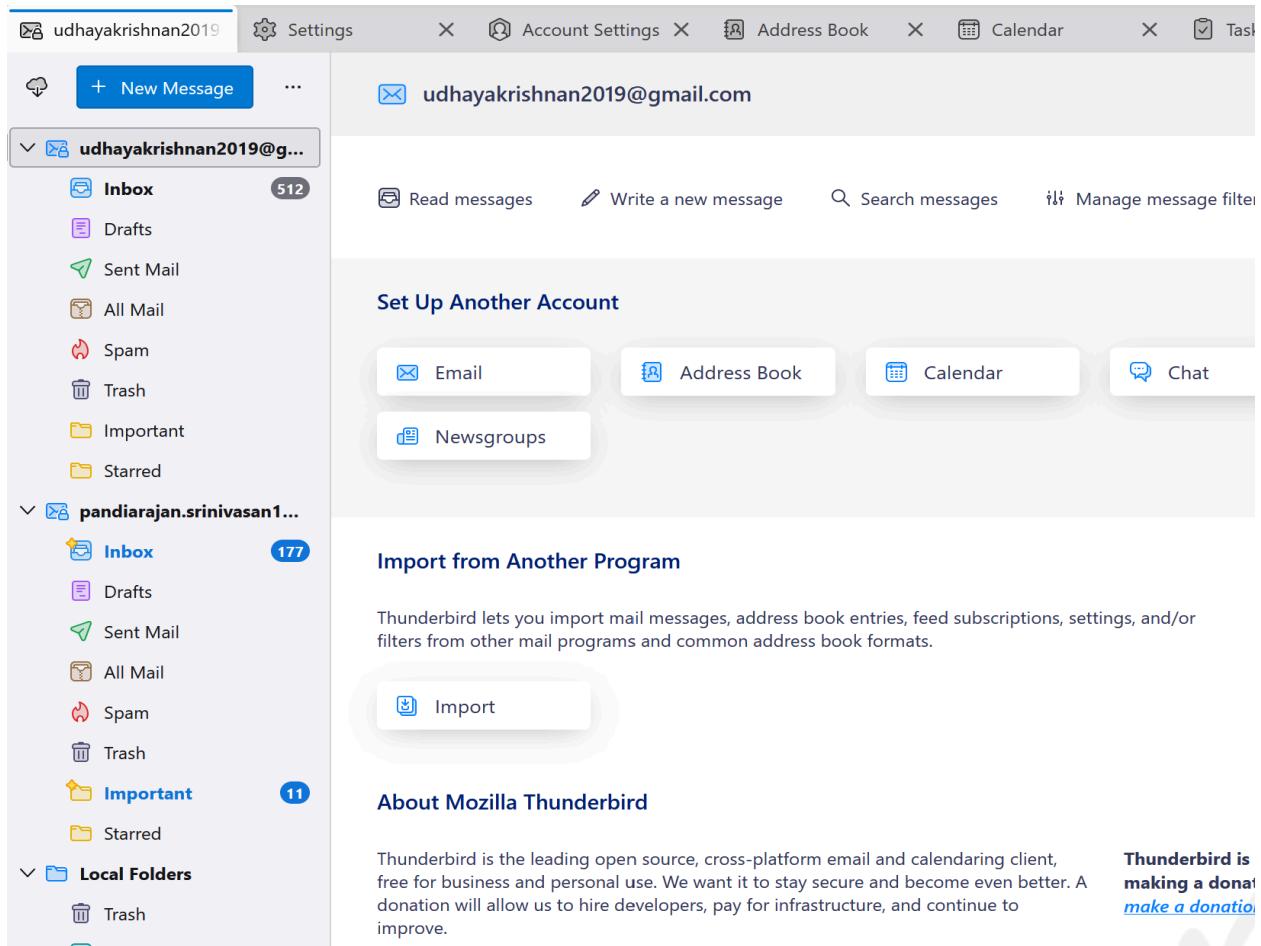
✓ Account successfully created

You can now use this account with Thunderbird.

You can improve the experience by connecting related services and configuring advanced account settings.



Both Accounts Configured Successfully

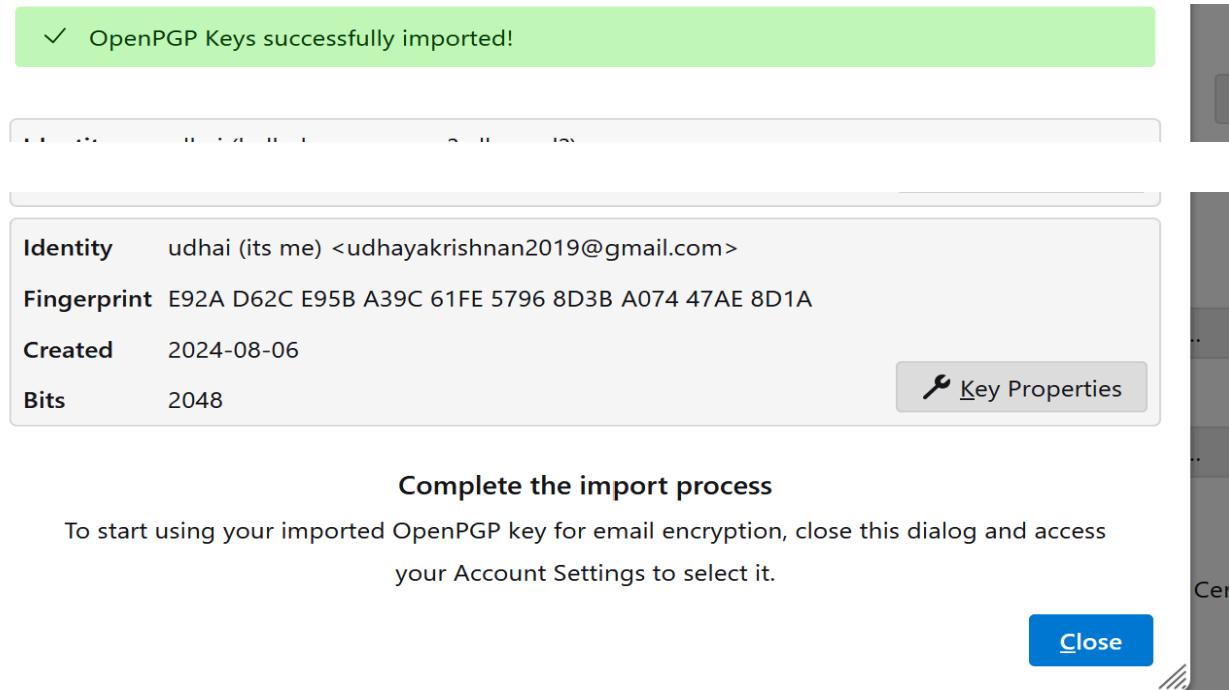


Exchanging Public Keys

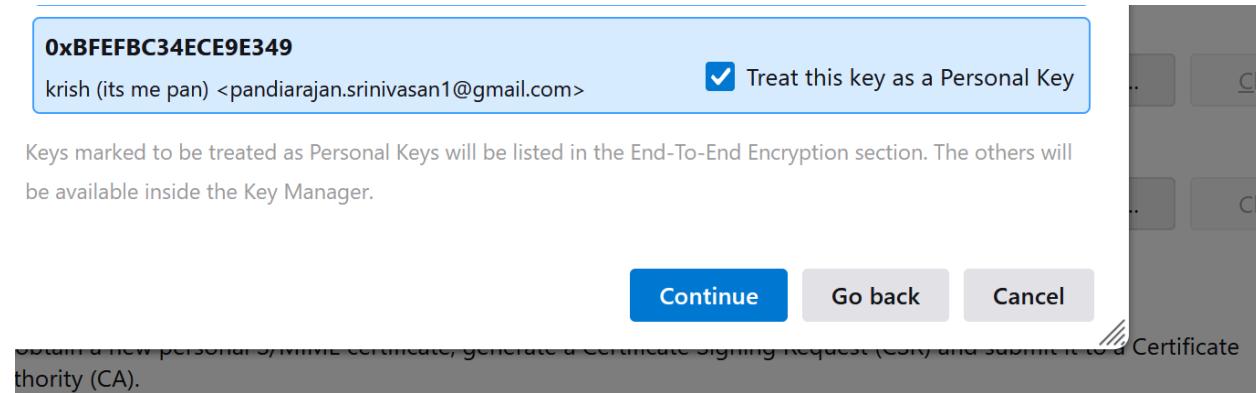
Exchanging public keys between two email accounts is a critical step in setting up secure email communication using encryption. The process ensures that each party can encrypt messages to be sent securely to the other party, and only the corresponding recipient can decrypt and read them.

Sending the keys over the mail securely and importing the same in Thunderbird to configure with the sender and receiver accounts

Importing the key-pair for the sender



Importing the key-pair for the receiver



OpenPGP

Thunderbird found 1 personal OpenPGP key associated with
 **pandiarajan.srinivasan1@gmail.com**

 Add Key...

✓ Your current configuration uses key ID **0xBFEFBC34ECE9E349** [Learn more](#)

None

Do not use OpenPGP for this identity.

0xBFEFBC34ECE9E349

 Expires on: 2024-08-08 [Change Expiration Date](#)

Publishing the public key on a keyserver allows others to discover it.

[Publish](#)



Use the OpenPGP Key Manager to view and manage public keys of your correspondents and all other keys not listed above.

Both key pairs were configured successfully.

Identity udhai (its me) <udhayakrishnan2019@gmail.com>

Fingerprint E92A D62C E95B A39C 61FE 5796 8D3B A074 47AE 8D1A

Created 2024-08-06

Bits 2048

 [Key Properties](#)

Identity krish (its me pan) <pandiarajan.srinivasan1@gmail.com>

Fingerprint A1FF 78DA 7261 49F8 C82D 3938 BFEF BC34 ECE9 E349

Created 2024-08-06

Bits 2048

 [Key Properties](#)

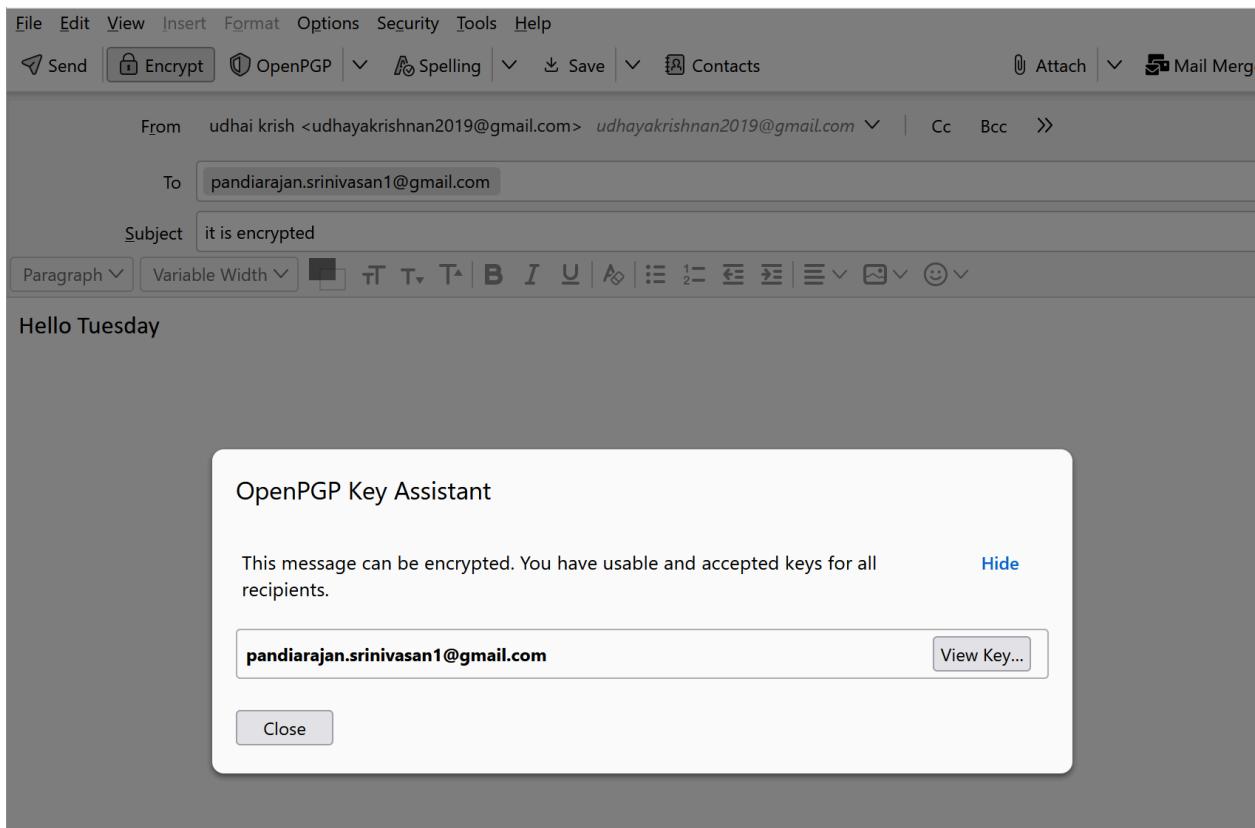
Complete the import process

To start using your imported OpenPGP key for email encryption, close this dialog and access
your Account Settings to select it.

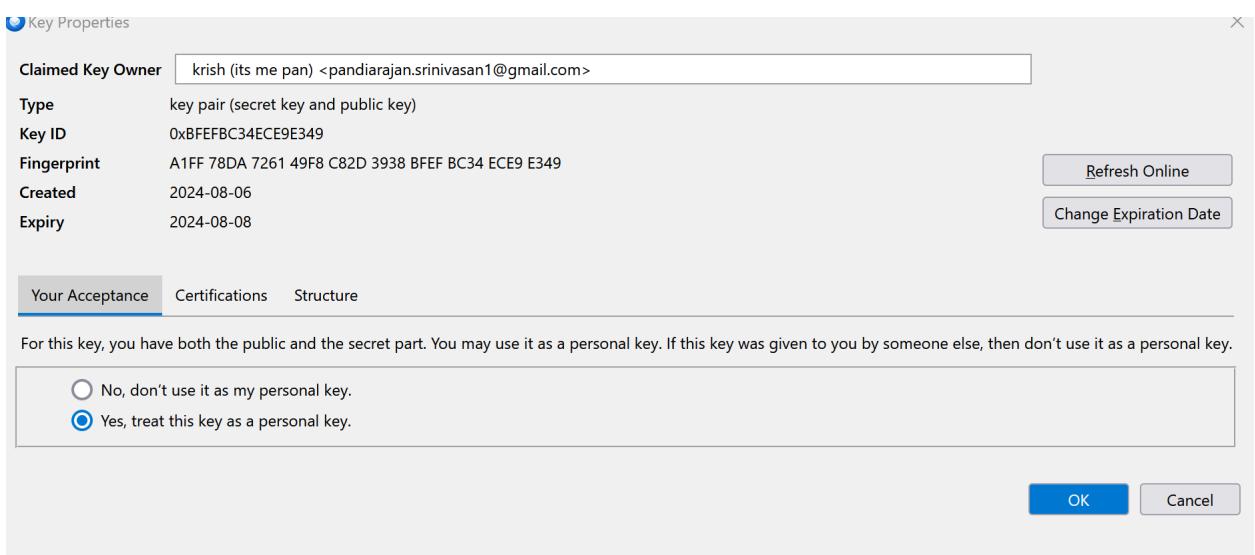
[Close](#)

Sending Encrypted Email and Verification of Encrypted Email

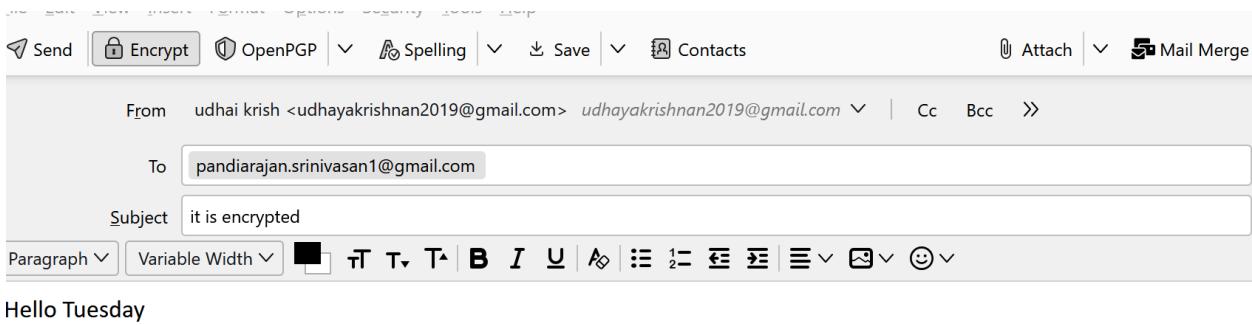
Composing Mail



Select the key from OpenPGP option

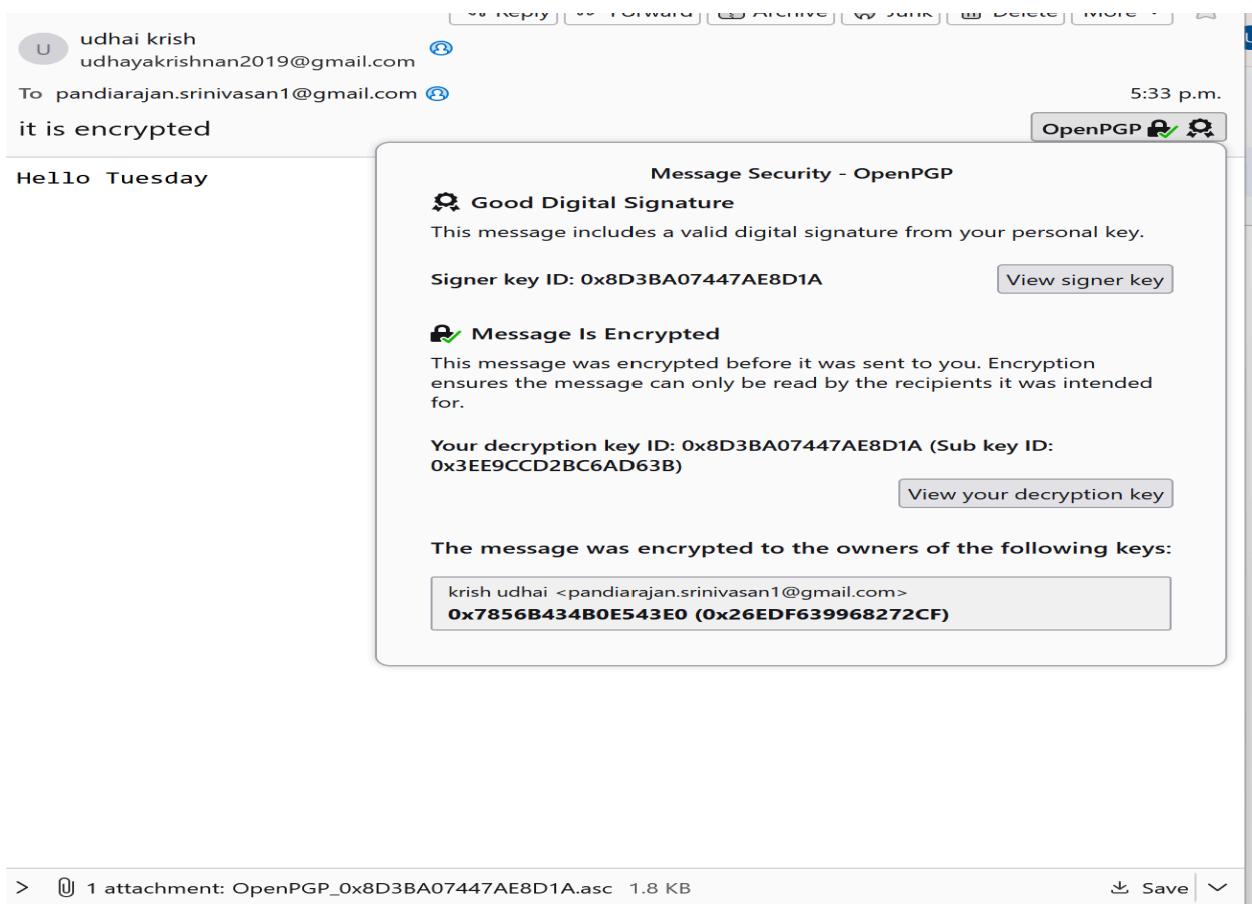


Sending Encrypted Message with the key-pair selected



Hello Tuesday

The Encrypted mail received with and Decrypted with the keys shared



Reply Forward Archive Junk Delete More ▾

U udhai krish
udhayakrishnan2019@gmail.com

To pandiarajan.srinivasan1@gmail.com

2024-08-06, 5:33 p.m.

it is encrypted

Hello Tuesday

Source of: imap://pandiarajan.srinivasan1%40gmail.com@imap.gmail.com:993/fetch%3E... — X

File Edit View Help

Tue, 06 Aug 2024 14:33:42 -0700 (PDT)

Message-ID: <e3fe6e92-a93e-4f3b-b4ca-cfa952d78b01@gmail.com>

Date: Tue, 6 Aug 2024 17:33:43 -0400

MIME-Version: 1.0

User-Agent: Mozilla Thunderbird

Content-Language: en-US

To: pandiarajan.srinivasan1@gmail.com

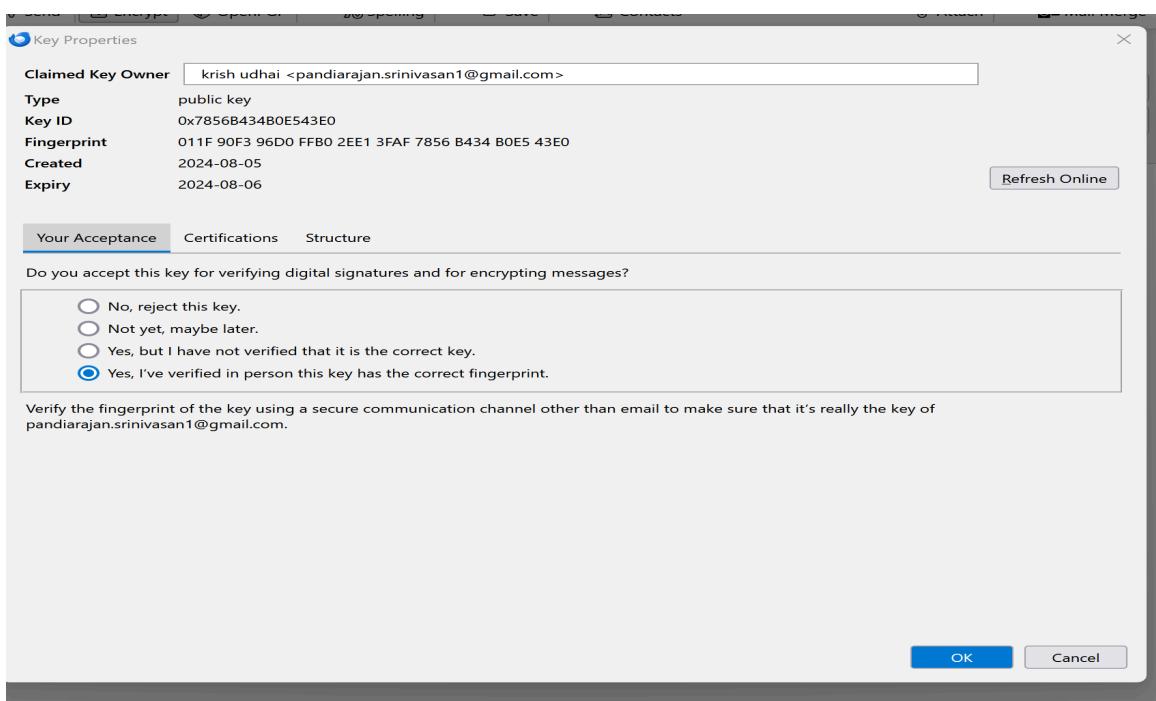
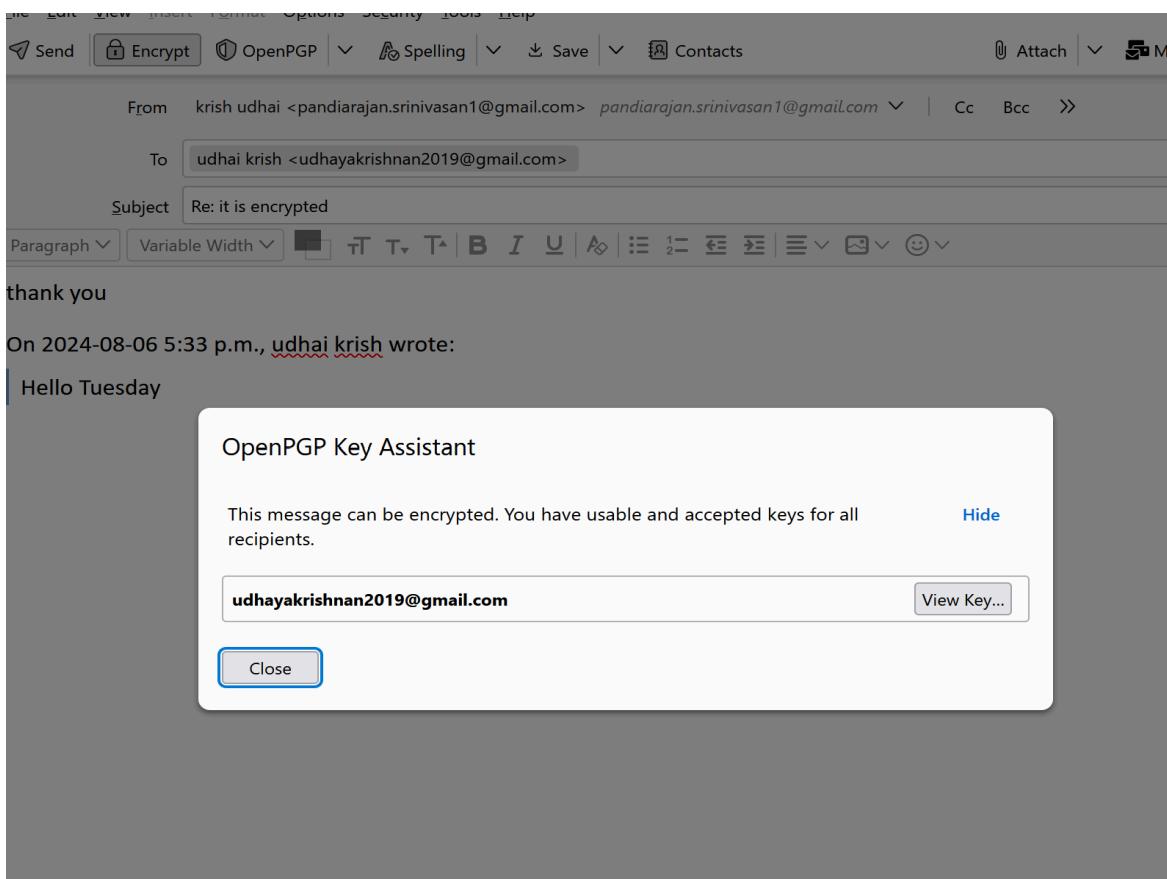
From: udhai krish <udhayakrishnan2019@gmail.com>

Autocrypt: addr=udhayakrishnan2019@gmail.com; keydata=

```
xsBNBGayj8cBCACPr9jHgF4bfI9ZnMUaEthJhYiueN6JW3CZkpRdfDOSEC4IxCREpZ1nw69
4hmmQX3GGk6k74DU48N8fSXLFXwouAVUverSfwowhM+XNpCSXaI8qzJVLCMdLWCJ9M/pKJxN
Y4nX1osbPxyi3ASubsmf72SzPK0iW4hK07nQiCowgfhooZA6NsKLR0txwIjFH+BzISZnc/tS
1nwGdZcF/JRleKexf1PGP/bHeWG4m9RJQqoi/EBde9V8meoOerr2fwavP2V51DxituVex0zb
55MN/TQHBWHWQs0n/72CUjV6iKNHD8digel1CwZQKCwIo1Fi6dgFar1K048AwLDqLsz1ABEB
AAHNLXVkaGFpTChpdHMgbwUpIDx1ZGhheWFrcmlzaG5hbjIwMT1AZ21halwuY29tPsLAlAQ
AQAoAPhYhB0kq1izpW60cYf5X1o07oHRHro0aBQJmso/HAhsDBQkAAqMABQsJCAccBhUKCQgL
AgQWAgnMBAh4BAheAAAoJEI07oHRHro0ay5kIAIcdzccc2NWVG3EY56CyeFA/fWcedYFm9rJH
cW0Sr0uAty/z020v/zJRcQZVvzmIgdcUfg66CwtTJTpR/GzH8ALp0FFzarbk0ijpGneNS8vv
5z16000n4QeUWiHjA0mcZQAo1eqS10WUpxk5yvUTvrNYPId+1Far/ntRtSYPCMuxNJI3dI7
BborPpsFjibV1+GwynVZfRbt04T4aWFtVoeZWVdetTx/1hJH7XkizBE4xREof35hGniWqce1
UrItQijseuXlnzNbMFi1DMwiQKUqwb20dL7GWJs/rLX+2jnmg/VZFvQnr0+YF9u25E1m6FKv
zQnRLqdy56WjZRLSHmn0wE0EzrKPxwEIaK9C1p0P60qVMtIL81Q7Z5KBQk1P1ttv4pUg7tPx
kP0b0AnFWsk01eTwkQ9rTFIg1TsrIN0WiiIa5PfMjtN3/VwtLLbaB281A1FUoAVpL711GuI5
A1Y9XhALT3tRTVV7BUYkaj1b82NuXLBkcASMZBdidTzp9NdZnokp0ieY5ASICqGdGWrcq6e4
VPPtPJtlig+I4SQSEp2fB07kGR/LVAeHhRdXK8nbN9IPJCtvEVoDqty3sgEt/k1Hw3vk1e4U
Ty1zgCE9uYu95TxmEq1aKe6g1Ncd09ycDTWJKLggeIVJc2vIfpYeQ07UNZartyy3xbxJ8r2I
sd9UbdMJoeuNtrEAEQEAAcLAfaQYAQoAJhYhB0kq1izpW60cYf5X1o07oHRHro0aBQJmso/H
AhsMBQkAAqMAAAoJEI07oHRHro0aLwMH/0iygGSFoGJR3GUg1RZ0v34sy9AzUB+LuW0EN8V1
```

> 1 attachment: OpenPGP_0x8D3BA07447AE8D1A.asc 1.8 KB Save

Responding the sender back with the shared public key



Decrypted Successfully at the other end

K krish udhai
pandiarajan.srinivasan1@gmail.com

To udhai krish
Re: it is encrypted

2024-08-06, 7:45 p.m.

OpenPGP 

thank you

On 2024-08-06 5:33 p.m., udhai krish wrote:

Hello Tuesday

Source of: imap://udhayakrishnan2019%40gmail.com@imap.gmail.com:993/fetch%3EUI... — □ ×

File Edit View Help

```
for <udhayakrishnan2019@gmail.com>
(version=TLS1_3 cipher=TLS_AES_128_GCM_SHA256 bits=128/128);
Tue, 06 Aug 2024 16:45:01 -0700 (PDT)
Message-ID: <f1361b1a-b28d-405f-9a6b-1414515cdeb9@gmail.com>
Date: Tue, 6 Aug 2024 19:45:03 -0400
MIME-Version: 1.0
User-Agent: Mozilla Thunderbird
To: udhai krish <udhayakrishnan2019@gmail.com>
References: <e3fe6e92-a93e-4f3b-b4ca-cfa952d78b01@gmail.com>
Content-Language: en-US
From: krish udhai <pandiarajan.srinivasan1@gmail.com>
Autocrypt: addr=pandiarajan.srinivasan1@gmail.com; keydata=
xsBNBGayS4BCADCq8Y3GWYhwiqbzEAUJ8MJ04VQPuuK8JEqeP2rNE2BiVvrTFnJVX7s70m
xJd82SfjnoPe5Wl1f0hJbAtzVHB3MOwRBCNrVBBTv+B0pX5dgpUX0SSaDzINFxwTZb+mKkYX
sL6aWQY2A1KQGgvmQNN2n+w3ctncMaHv//FojPFLKDUS13f8iz80Ymgmq+egioNSDKFw2oq
TQOCPDgCtsNFAsyfEJRCmOs1SuKtqM2Y4L0pGxb7zj8hf6iQB0NMx+e7HOIE5N3x12vXIsSZ
is9YBsHvuPfB/cAPbX60VaeEH19L13yXtg7qxSOoq0mBF3uWeXn9oq7zvhEgw3Huaes3ABEB
AAHNNmtyaXNoIChpHMgbWUgcGFuKSA8cGFuZG1hcmFqYW4uc3Jpbml2YXNhbjFAZ21halwu
Y29tPsLA1AQTAQoAPhYhBK/eNpyYUn4yC050L/vvDTs6eNJBQJmsrCeAhsDBQkAAqMABQsJ
CAcCBhUKCQgLAgQWAgnMAh4BAheAAAoJEL/vvDTs6eNjowAH/A1QKscaeT4i4qe8otPHTQHI
UxL216cw1cI1Wz3aMJIZDWPN4g/LimDDS7ySkIIYryRhbx4xVpe5wRgzRp+M47yTDxGvohnoP
F4Yi65AzSMZEd+kntsVIk1pjkg/vq9x31VEAAHq+kCmd7xTW/Wa+CCwyhfa+0zj5Wam6Po2
BDWuBQBmvB07Z70e8xEZGI9g33/7NC9/MeyJFFy5NpLB9WeRNJ6Ub2IT3DMENhardV5GoJTz
0RcxpMD0EH9I0/653zP5aj8EJ1xOs0m+teKKiDDshnr0ka1zzFw8QflbFrvt5ARKJ8SWrwr
Gzymzq2zPm6c4F73FXgZycOSdGVUBvr0wE0EZrKwngEIAJ41vYE09cz1Trt48NZvIrdAoHa
oY1FUksIwmaYjVbmOJyBwW2isyF1ZHsi9qkFaLBkb/q9xtWKR4YdpJg21N+6GBPzvSG9HF+>
```

Save

K krish udhai
pandiarajan.srinivasan1@gmail.com

To udhai krish

Re: it is encrypted

thank you

On 2024-08-06 5:33 p.m.,
Hello Tuesday

7:48 p.m.

OpenPGP 

Message Security - OpenPGP

 **Good Digital Signature**
This message includes a valid digital signature from your personal key.

Signer key ID: 0xBFEFBC34ECE9E349 [View signer key](#)

 **Message Is Encrypted**
This message was encrypted before it was sent to you. Encryption ensures the message can only be read by the recipients it was intended for.

Your decryption key ID: 0xBFEFBC34ECE9E349 (Sub key ID: 0xAAF54E01CB4DE820) [View your decryption key](#)

The message was encrypted to the owners of the following keys:

udhaikrish <udhayakrishnan2019@gmail.com>
0xD274285C0563A35D (0x3651052413EA2B26)

>  1 attachment: OpenPGP_0xBFEFBC34ECE9E349.asc 1.8 KB [Save](#) 

Conclusion

Implementing email encryption using GPG, Thunderbird, and OpenPGP was insightful. It involved understanding key cryptography concepts, selecting appropriate tools, generating keys, configuring email accounts, exchanging public keys, and finally encrypting and sending emails. Each step was crucial in ensuring secure communication.

Importance of Email Encryption in Securing Digital Communications

Email encryption is vital for protecting sensitive information from unauthorized access. It ensures confidentiality, integrity, and authenticity of communications, thereby safeguarding against threats such as eavesdropping, tampering, and phishing attacks. This implementation reinforces the necessity of adopting strong encryption practices to maintain privacy and security in digital communications.