CMSC389R

Cryptography II

COMPUTER SCIENCE
UNIVERSITY OF MARYLAND

homework 9 recap

- Hash cracking
- Hash crack and submit challenge
- Questions?

agenda

- Some theory
  - Private key vs public key cryptography
- Heavy emphasis on applications
  - Focus on PGP
  - Message signing/verification

hashing

- Last week, we covered popular hashing algorithms
  - What they look like
  - How they can be cracked
  - Their weaknesses
- How does this tie into the course?

encryption

- Private key cryptography
  - Also known as symmetric cryptography
  - One key for encryption/decryption
  - Need to keep this *private*
- Public key cryptography
  - Also known as asymmetric cryptography
  - Public key -> encrypt, known to all
  - Private key -> decrypt, kept secret

symmetric key cryptography

- Meet your two new best friends:
  - Alice and Bob
- Alice wants to send Bob a message
  - Expectation: channel in which message is sent is actively wiretapped
  - Goal: send Bob a secret message without eavesdropper (Eve) recovering the message

symmetric key cryptography

- Alice generates a private key K and meets with Bob in person
  - Alice assures that Bob is *really* Bob
  - Alice give Bob K
- Alice and Bob part ways
  - Alice and Bob encrypt/decrypt messages with the *shared* key K.

symmetric key cryptography

- Examples:
  - Vigenere Cipher
  - One-time pad (OTP)
  - Data Encryption Standard (DES)
  - Advanced Encryption Standard (AES)

symmetric key cryptography

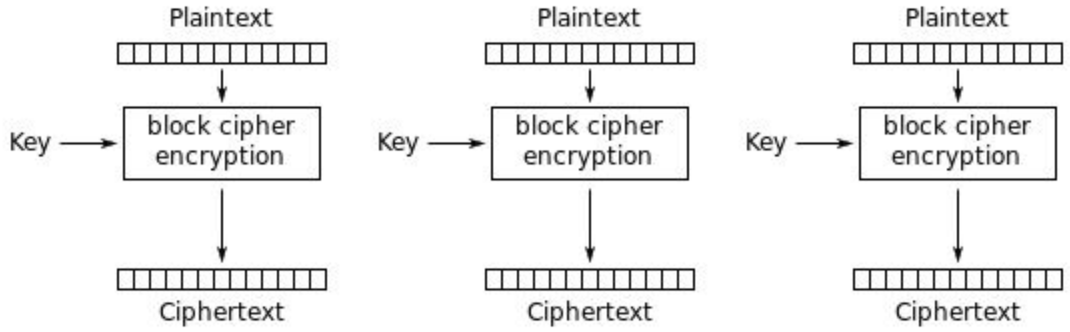
- Advantages
  - Very secure with small key length
  - Relatively fast (vs asymmetric)
- Disadvantages
  - Difficult to share key
  - Both parties vulnerable if key is compromised

symmetric key cryptography

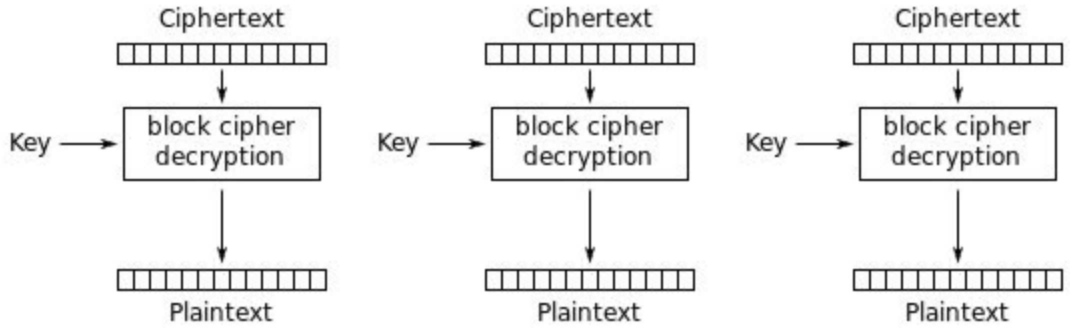
- Stream ciphers
  - Generate stream of random-enough as long as plaintext to encrypt it
  - Perfect secrecy: len(key) >= len(plaintext)
- Block ciphers
  - Work on individual chunks or “blocks” of plaintext
  - Ease of use with shorter keys and longer plaintexts (i.e. large files)

symmetric key cryptography

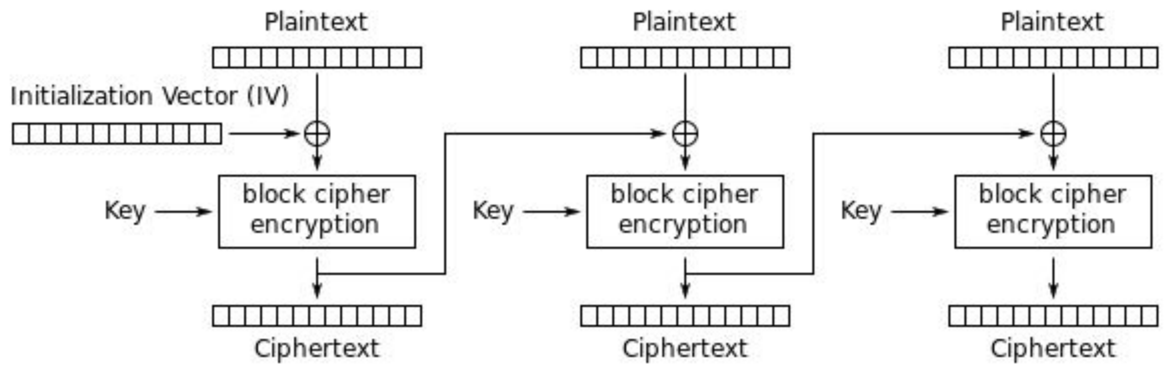
- Block cipher mode of operation
  - Cipher only dictates individual block algorithm
  - Mode of operation dictates how blocks interact or depend on each other



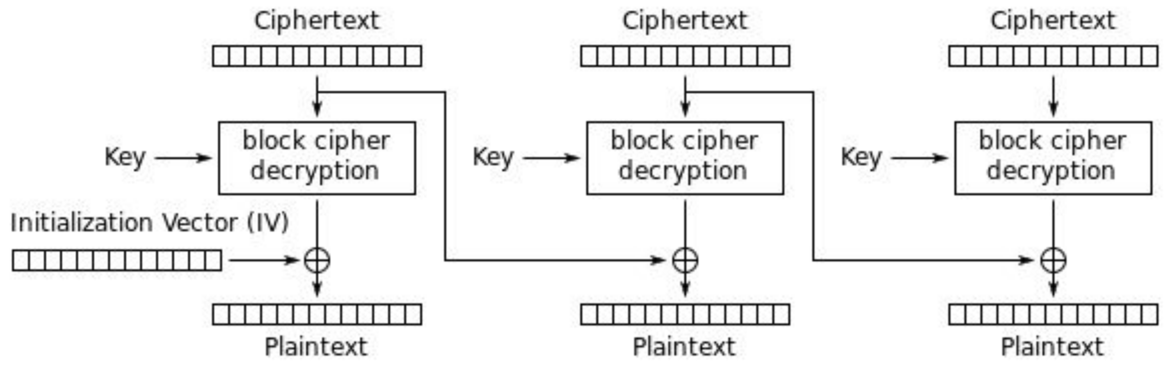
Electronic Codebook (ECB) mode encryption



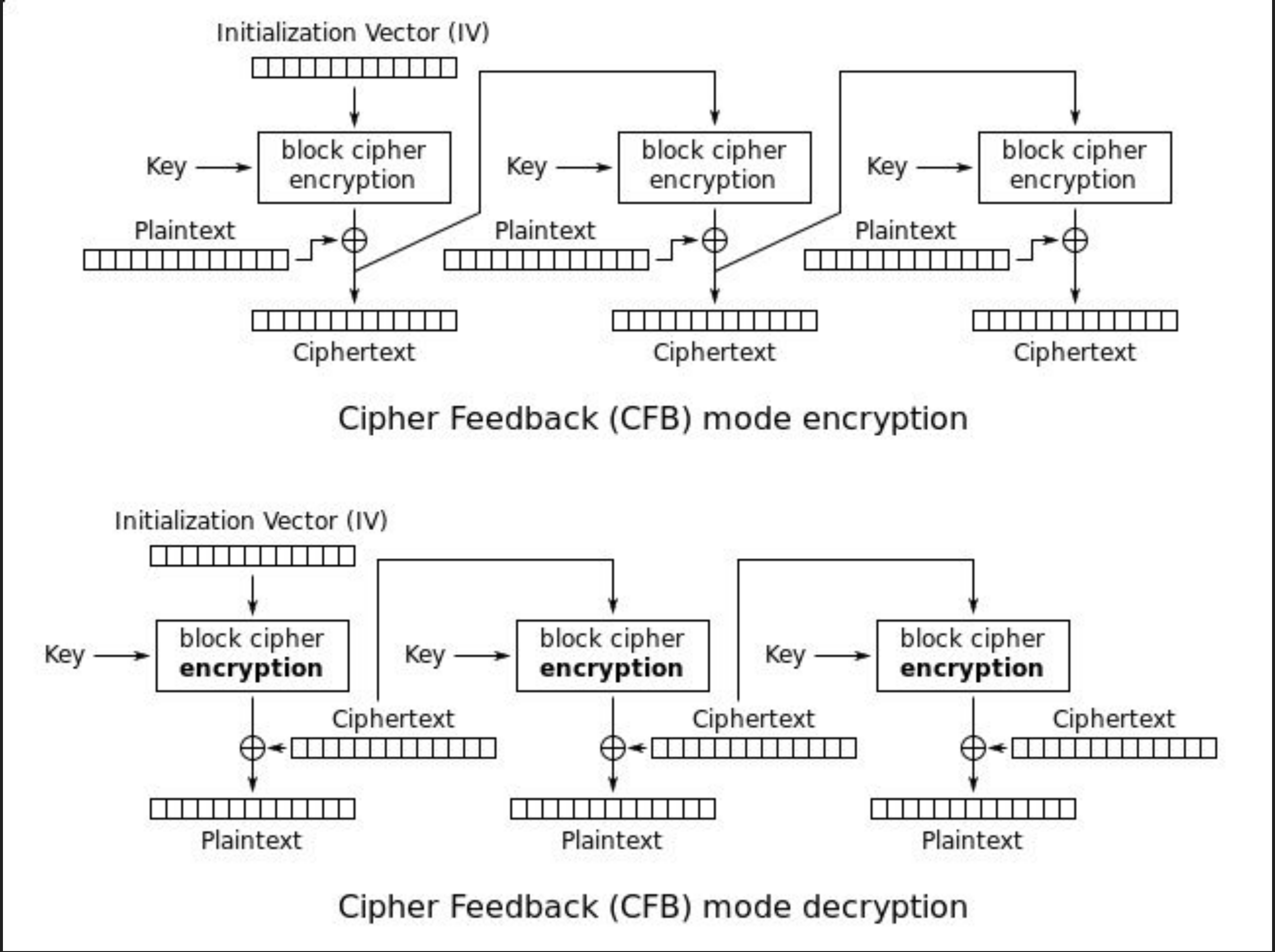
Electronic Codebook (ECB) mode decryption

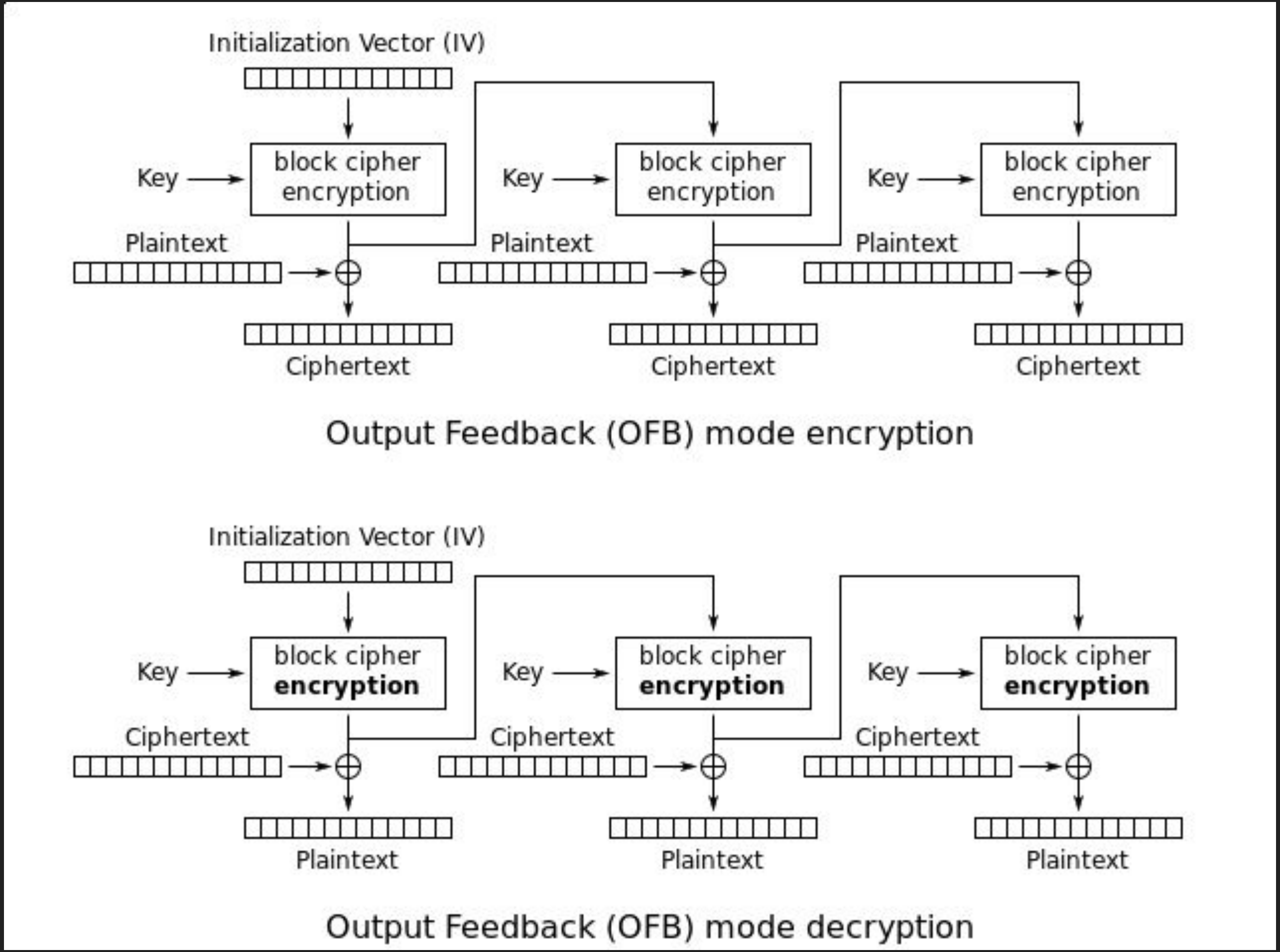


Cipher Block Chaining (CBC) mode encryption

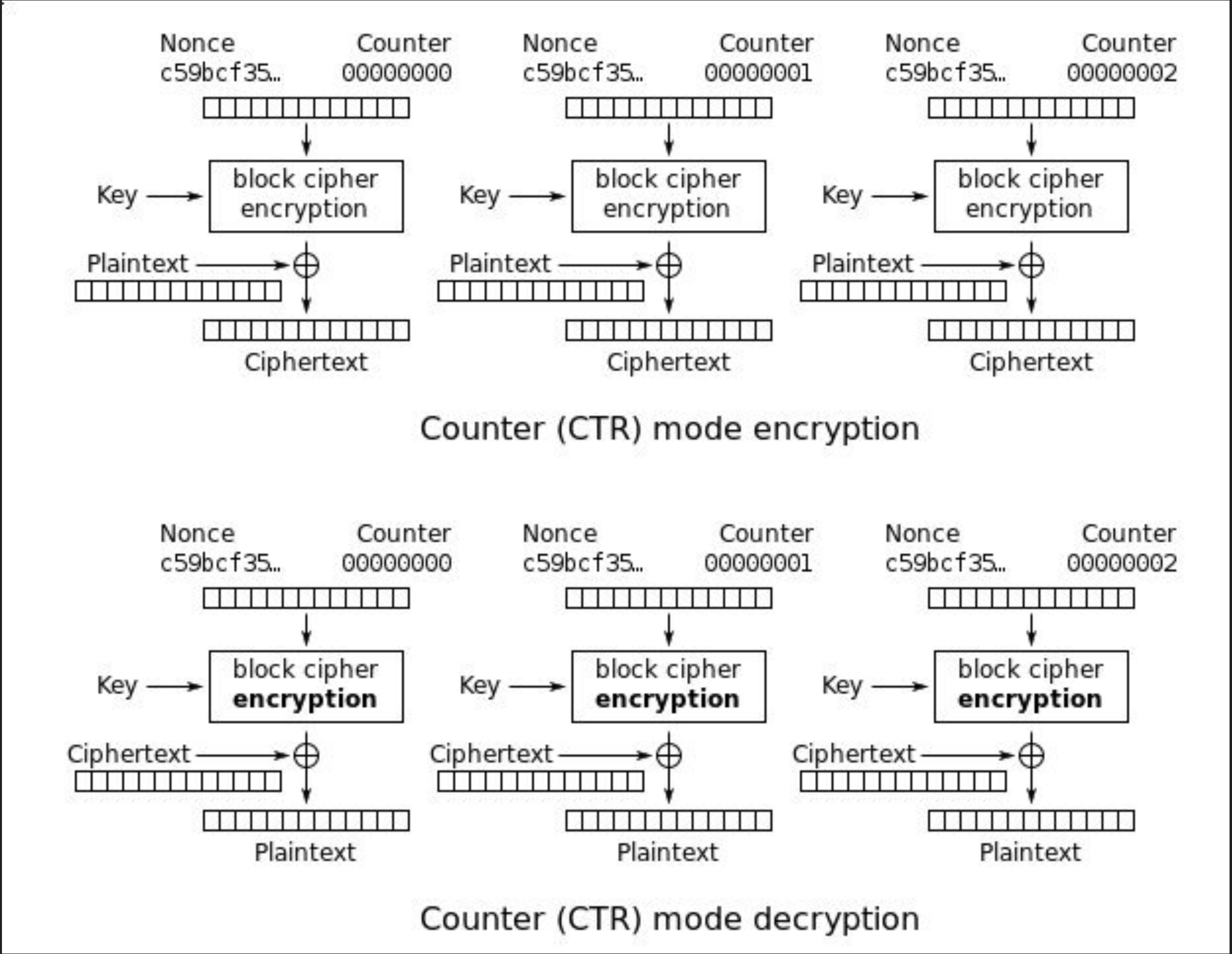


Cipher Block Chaining (CBC) mode decryption

- 
Initialization Vector (IV)
Key
block cipher encryption
Plaintext
Ciphertext
Cipher Feedback (CFB) mode encryption
Initialization Vector (IV)
Key
block cipher encryption
Ciphertext
Plaintext
Cipher Feedback (CFB) mode decryption

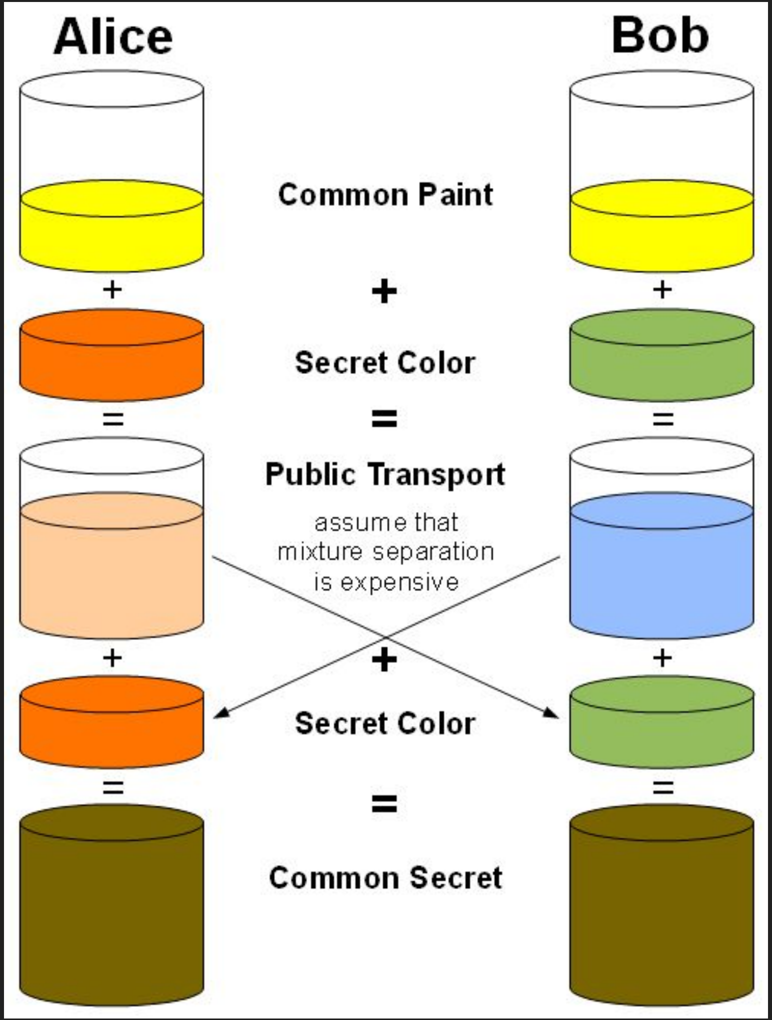
- 
Initialization Vector (IV)
Key
block cipher encryption
Plaintext
Ciphertext
Output Feedback (OFB) mode encryption
Initialization Vector (IV)
Key
block cipher encryption
Ciphertext
Plaintext
Output Feedback (OFB) mode decryption

●


Nonce
c59bcf35...
Counter
00000000
00000001
00000002
Key
block cipher encryption
Plaintext
Ciphertext
Counter (CTR) mode encryption
Counter (CTR) mode decryption

symmetric key cryptography

- Protocols exist for safe generation and sharing of symmetric keys
- e.g. Diffie-Hellman
    - Alice and Bob each pick their own secret
    - Each person transforms their secret based on a publicly agreed upon rule and exchange their transformations
    - Each person applies their secret to the other’s transformation to get the same key



- Diffie-Hellman
- Uses modular arithmetic
- Security provable based on mathematically hard problem
  - Discrete log

$$(g^a \bmod p)^b \bmod p = (g^b \bmod p)^a \bmod p$$

asymmetric key cryptography

- Alice and Bob each generate a public and private key pair
- Each exchange their public keys, keep private keys secret
- Alice applies Bob’s public key to her message and sends to Bob, encrypted
- Bob and ONLY Bob can apply his private key to Alice’s encryption to recover the message

asymmetric key cryptography

- Examples:
  - RSA (*HTTPS*, DRM)
  - PGP (Commonly used in email, ACH)
  - ElGamal (Discrete Logarithm problem)
  - Elliptic-Curve (shorter keys than RSA)

openssl

- Software suite for cryptographic operations
    - Command line tool
        - encryption/decryption
        - Secure sockets (think encrypted netcat)
        - + more
    - Programming libraries in C

openssl

- Encryption and decryption
  - openssl enc
  - -aes-128-MODE for AES128 in various modes
    - other cipher algorithms/modes available
    - cbc, ecb, etc
  - -d decrypt, -e encrypt
  - -K key in hex digits
  - -iv initialization vector (IV) in hex digits
  - -in input, -out output

openssl

- Secure sockets
  - openssl s_client
  - -connect <hostname>:<port>
  - -cipher cipher type
    - type “openssl ciphers” for cipher types

Pretty Good Privacy (PGP)

- Pretty Good Privacy (PGP): Developed in 1991 by Phil Zimmermann. Allows for
  - Encryption/Decryption
  - Signing
- Frequently used in email, files, full disk encryption, etc.

Pretty Good Privacy (PGP)

- Use gpg command line tool to generate public key/private key pair
- Can then share public key with the world
    - MIT PGP Key server
    - Email (ie. enigmail)
    - Keybase
    - …
- Decrypt messages using PGP private key

Pretty Good Privacy (PGP)

- gpg --gen-key
    - Generate key -- ID based on name/email
- gpg --list-secret-keys
- gpg --export --armor you@email.com > pubkey.asc
    - Create a key to send to friends
- gpg --import key.asc
    - Import a friend’s key
- gpg -e -u “Your name” -r “Their name” msg.txt
    - Generates msg.txt.gpg
- gpg --decrypt msg.txt.gpg
    - Display decrypted message

Pretty Good Privacy (PGP)

- Can sign documents too
  - Analogous to “encrypting” with private key
  - Anyone can then use your public key to verify the signature
- gpg --output myfile.sig --sign myfile
  - “Encrypts” file, can’t see message without “decrypting”
- gpg --output myfile --decrypt myfile.sig
- gpg --clearsign myfile
  - Generates myfile.asc
  - Wraps file in signature rather than “encrypting” it
  - Useful for sending/posting publicly as it provides the message/signature in ASCII

Pretty Good Privacy (PGP)

```
-----BEGIN PGP PUBLIC KEY BLOCK-----



mQENBFrHpXsBCADeJrGA5Rwaj4GvAwzGtKt6PFC
oaXj7uJTKl3h2IR2YTFSbyQV2...

-----END PGP PUBLIC KEY BLOCK-----
```

```
-----BEGIN PGP PRIVATE KEY BLOCK-----



02I1BJm5AQ0EWselewEIALahcUsgcJTyUb+yWka
+cN2Tsh3oItAAndhXUR0/zsEN...

-----END PGP PRIVATE KEY BLOCK-----
```

Homework #10

Will be posted soon.

Let us know if you have any questions!

This assignment has 2 parts.