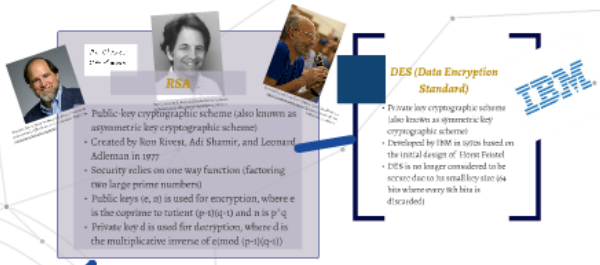
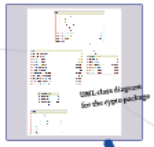
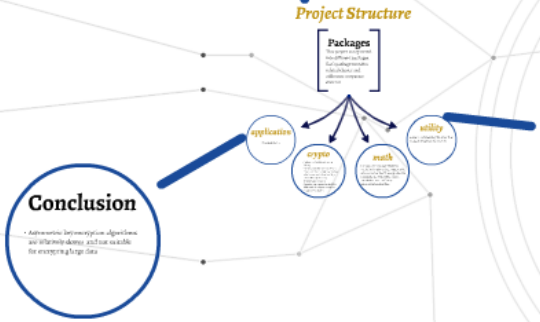
RSA & DES Implementation

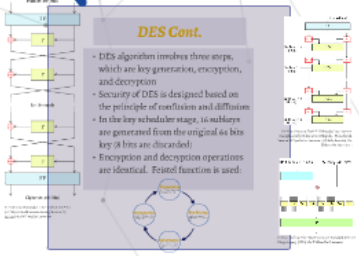
Outline

- RSA & DES algorithms recap
- Program architecture
- Demo
- Conclusion





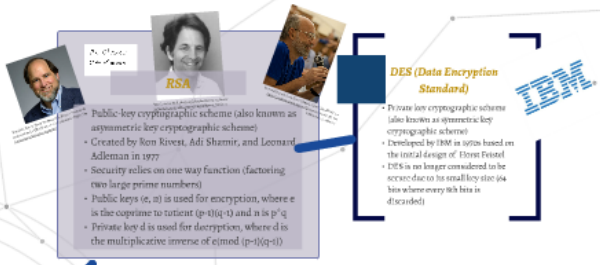


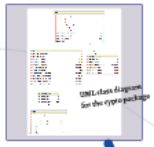


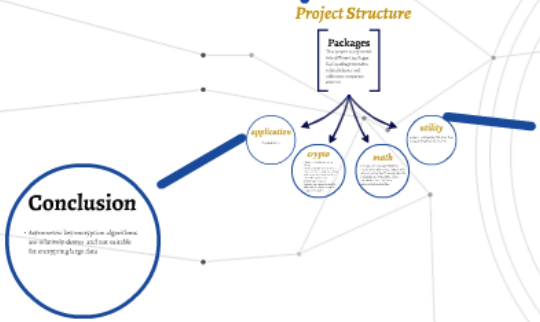
RSA & DES Implementation

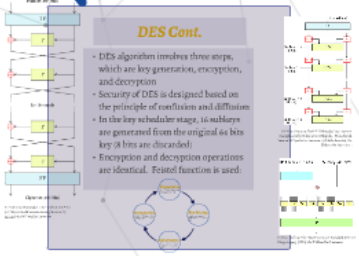
Outline

- RSA & DES algorithms recap
- Program architecture
- Demo
- Conclusion



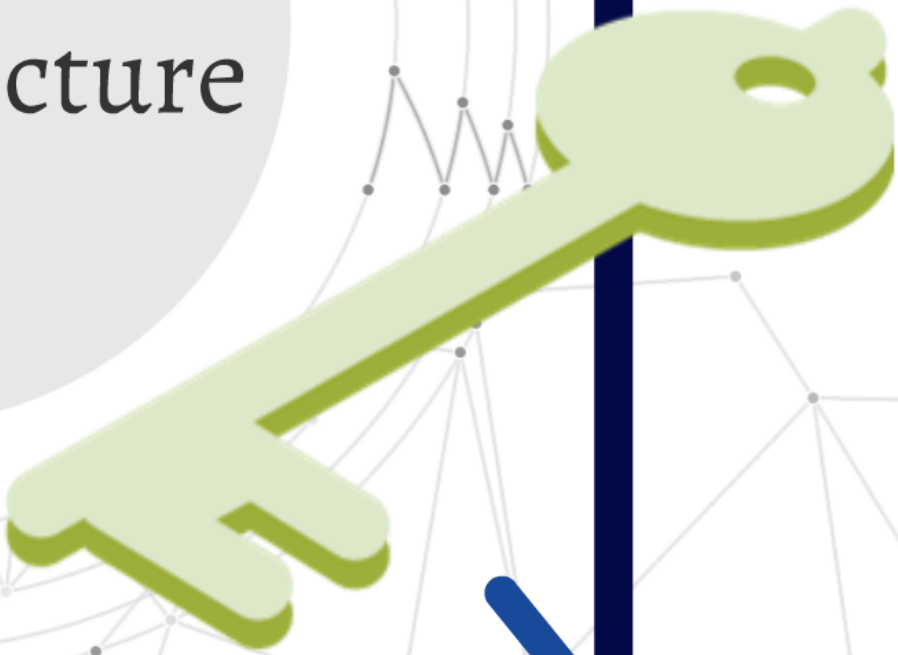






Outline

- RSA & DES algorithms recap
- Program architecture
- Demo
- Conclusion



st photo" by Ronald L. Rivest - Own work. er CC BY-SA 4.0 via Commons - https://
ikimedia.org/wiki/

$$P \leftarrow C^d \bmod n$$
$$C \leftarrow P^e \bmod n$$

http://cs.stanford.edu/people/eroberts/courses/soco/projects/2003-04/dna-computing/adleman_bio.htm

RSA

By Ira Abramov from Even Yehuda, Israel [CC BY-SA 2.0 (http://creativecommons.org/licenses/by-sa/2.0)], via Wikimedia Commons

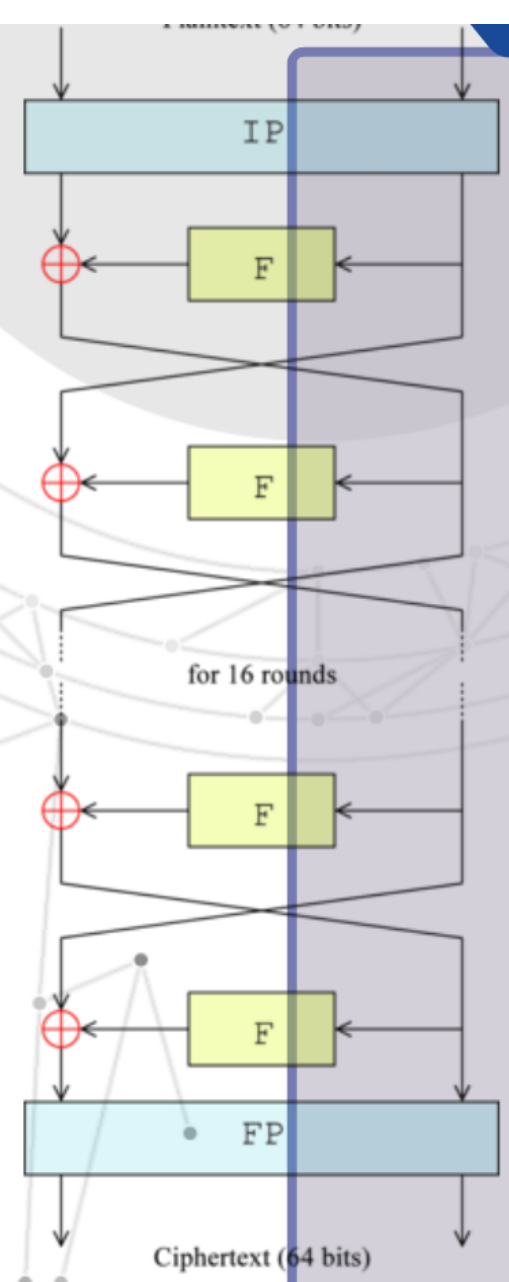
- Public-key cryptographic scheme (also known as asymmetric key cryptographic scheme)
- Created by Ron Rivest, Adi Shamir, and Leonard Adleman in 1977
- Security relies on one way function (factoring two large prime numbers)
- Public keys (e, n) is used for encryption, where e is the coprime to totient (p-1)(q-1) and n is p*q
- Private key d is used for decryption, where d is the multiplicative inverse of e(mod (p-1)(q-1))

DES (Data Encryption Standard)

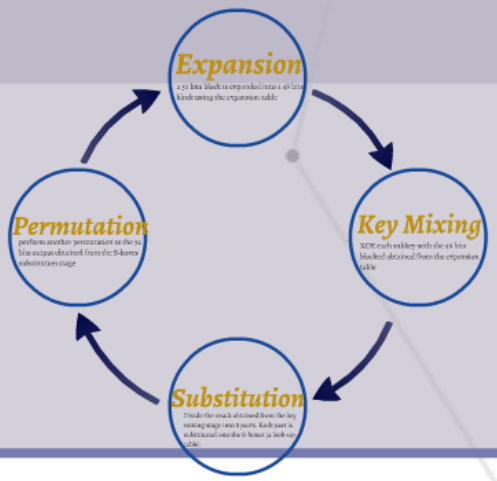
- Private key cryptographic scheme (also known as symmetric key cryptographic scheme)
- Developed by IBM in 1970s based on the initial design of Horst Feistel
- DES is no longer considered to be secure due to its small key size (64 bits where every 8th bits is discarded)

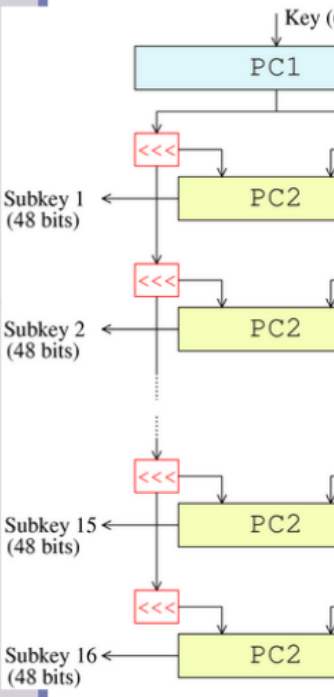
DES Cont.

- DES algorithm involves three steps, which are key generation, encryption, and decryption
- Security of DES is designed based on the principle of confusion and diffusion
- In the key scheduler stage, 16 subkeys are generated from the original 64 bits key (8 bits are discarded)
- Encryption and decryption operations are identical. Feistel function is used:

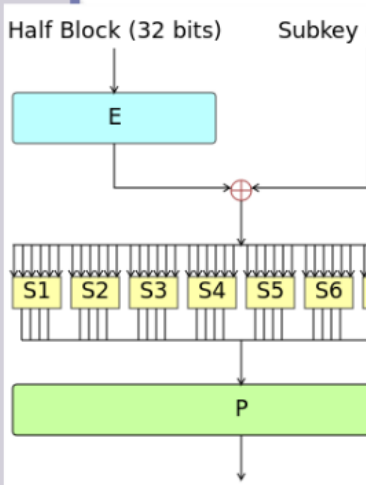


By Francisco Menendez (Own work) [CC BY-SA 3.0 (http://creativecommons.org/licenses/by-sa/3.0)], via Wikimedia Commons

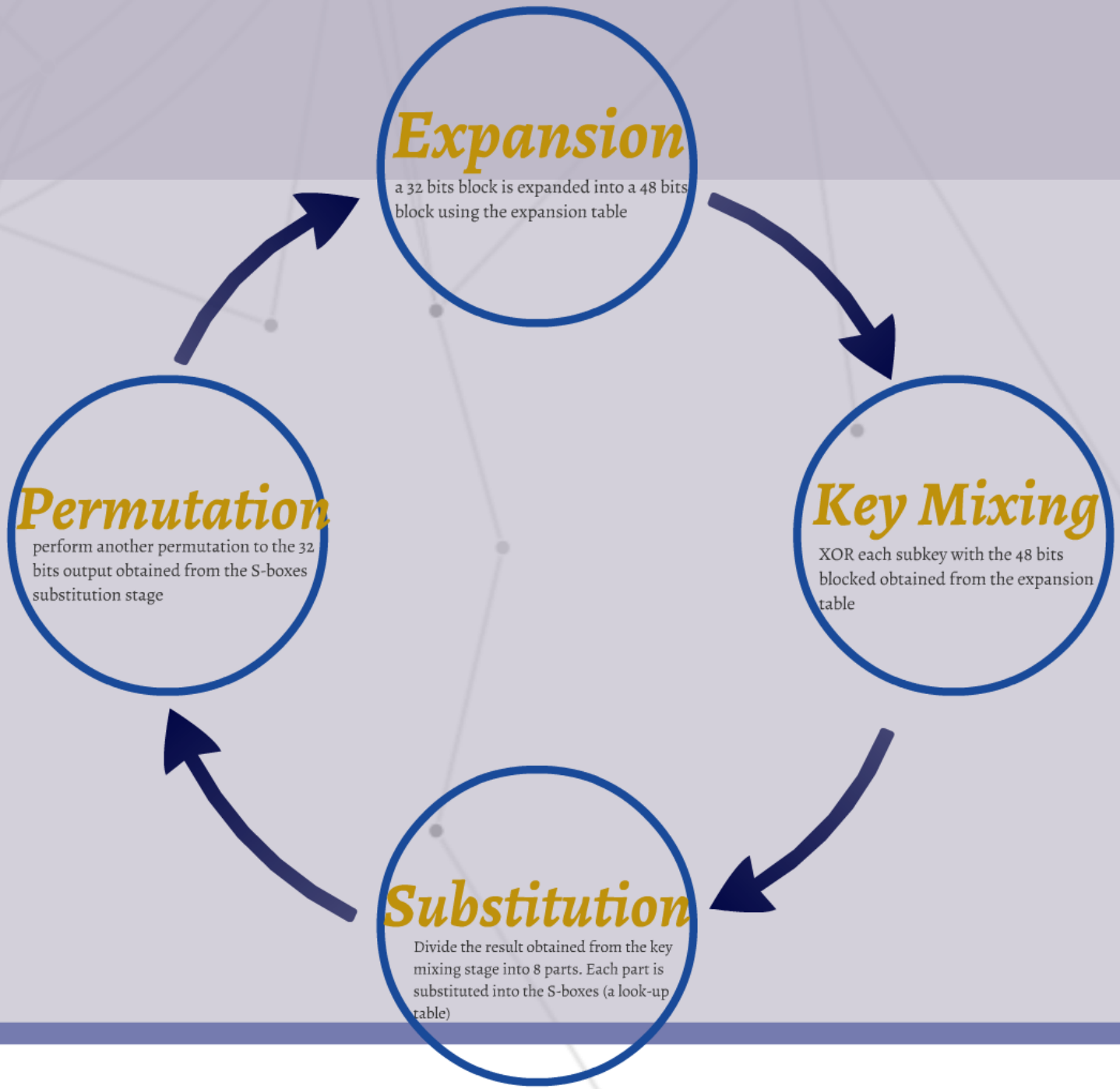




By Matt Crypto at English Wikipedia La
was uploaded by Zebez at en.wikipedia. (T
from en.wikipedia to Commons.) [Public d
Wikimedia



By User:Hellisp (File:Data Encryption Stan
Diagram.png) [CC0], via Wikimedia Comm


Expansion
a 32 bits block is expanded into a 48 bits block using the expansion table
Key Mixing
XOR each subkey with the 48 bits blocked obtained from the expansion table
Substitution
Divide the result obtained from the key mixing stage into 8 parts. Each part is substituted into the S-boxes (a look-up table)
Permutation
perform another permutation to the 32 bits output obtained from the S-boxes substitution stage

Expansion

a 32 bits block is expanded into a 48 bits block using the expansion table

Key Mixing

XOR each subkey with the 48 bits blocked obtained from the expansion table

Substitution

Divide the result obtained from the key mixing stage into 8 parts. Each part is substituted into the S-boxes (a look-up table)

Permutation

perform another permutation to the 32 bits output obtained from the S-boxes substitution stage

Project Structure

Packages

This project is organized into different packages. Each package contains related classes and addresses a separate concern.

application

UI related classes

crypto

- Package for building the crypto library.
- Includes two abstract base classes SymmetricKey and AsymmetricKey which serve as the basic building blocks for implementing cryptographic schemes.
- Includes implementation for RSA, DES, and block cipher operation modes ECB and CFB.

math

Contains common math functions needed for implementing cryptographic schemes such as Euclidean algorithm for computing the GCD, Miller Rabin's probabilistic test, and binary exponentiation algorithm.

utility

contains useful auxiliary functions that are used throughout the program.

Packages

This project is organized into different packages. Each package contains related classes and addresses a separate concern.

Packages

This project is organized into different packages. Each package contains related classes and addresses a separate concern.

application

UI related classes

crypto

- Package for building the crypto library.
- Includes two abstract base classes SymmetricKey and AsymmetricKey which serve as the basic building blocks for implementing cryptographic schemes.
- Includes implementation for RSA, DES, and block cipher operation modes ECB and CFB.

math

Contains common math functions needed for implementing cryptographic schemes such as Euclidean algorithm for computing the GCD, Miller Rabin's probabilistic test, and binary exponentiation algorithm.

utility

contains useful auxiliary functions that are used throughout the program.

application

UI related classes

crypto

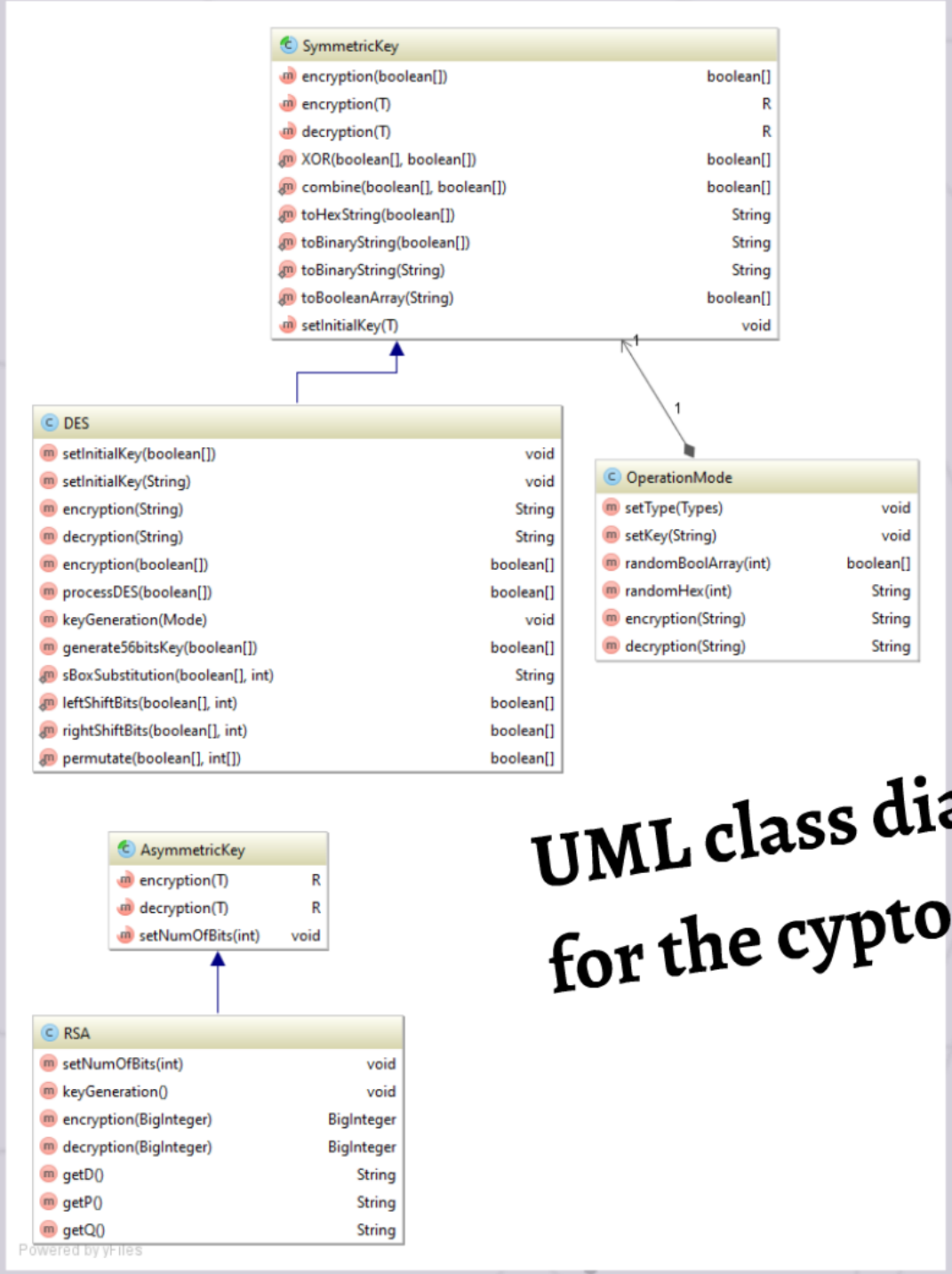
- Package for building the crypto library.
- Includes two abstract base classes SymmetricKey and AsymmetricKey which serve as the basic building blocks for implementing cryptographic schemes.
- Includes implementation for RSA, DES, and block cipher operation modes ECB and CFB.

math

Contains common math functions needed for implementing cryptographic schemes such as Euclidean algorithm for computing the GCD, Miller Rabin's probabilistic test, and binary exponentiation algorithm.

utility

contains useful auxiliary functions that are used throughout the program.



UML class diagram for the cypto package

Conclusion

- Asymmetric key encryption algorithms are relatively slower and not suitable for encrypting large data

RSA & DES Implementation

Outline

- RSA & DES algorithms recap
- Program architecture
- Demo
- Conclusion