TCSS 487 Project Part 2 Report
Douglas Johnston

Program Instructions

Command Line Mode:
To use this mode, run the program without any arguments. Then follow the instructions printed to the console.

File Mode:
To use this mode, run the program with an argument.

These are the acceptable flags: -KP, -PKF, -EE, -SF, -V

-KP:
Command structure: {input file path with name} -KP {passphrase}
Description: Generates an elliptic key pair from a given passphrase and writes the public and private keys to a file.

-PKF:
Command structure: {input file path with name} -PKF {public key file path with name}
Description: Encrypts a data file under a given elliptic public key file and writes the ciphertext to a file.
PUBLIC KEY FILE REQUIREMENTS: This command assumes that the public key is a point, represented by point a point (x, y). X should be on the first line in the public key file and y should be on the second line.
Public key file example:
132132dasf1231 (This line is X)
1325adsf113asdf (This line is Y)

-EE:
Command structure: {input file path with name} -EE {passphrase}
Description: Decrypts a given elliptic-encrypted file from a given password and writes the decrypted data to a file.
INPUT FILE REQUIREMENTS: This command assumes that the input file is a symmetric cryptogram (Z, c, t) where Z is a point (x, y). The first line should be Z_x, the second line should be Z_y, the third line should be c, and the fourth line should be t.
Input file example
132adsf1132 (This line is Z_x)
Asdff31121a (This line is Z_y)
Asdfadsfasfdafa (This line is c)
Adsfasfafafadsfasfdasfasdfafddasf (This line is t)

-SF:
Command structure: {input file path with name} -SF {password}
Description: Signs a given file from a given password and writes the signature to a file.

-V:
Command structure:
{input file path with name} -V {signature file path with name} {public key file path with name}
Description: Verifies a given data file and its signature file under a given public key file and writes VERIFIED or UNVERIFIED to a file.
SIGNATURE FILE REQUIREMENTS: This command assumes that the signature file is a Signature (h, z). The first line of the file should be h, and the second line should be z.
Signature file example:
Asdfa4564456465 (This line is h)
Asdfadsfad44654 (This line is z)
PUBLIC KEY FILE REQUIREMENTS: This command assumes that the public key is a point, represented by point a point (x, y). X should be on the first line in the public key file and y should be on the second line.
Public key file example:
132132dasf1231 (This line is X)
1325adsf113asdf (This line is Y)

Solution Description
Se TCSS_487_Project_Part_1_Report.pdf for explanations of functions and classes for that part

Point.java
A class that represents a Point (x, y) on the Ed448-Goldilocks curve where x and y are BigIntegers. It contains constants for the modulus p (which defines F_p), d, and the neutral element of addition: Point (0, 1).

equals
Compares two points using .equals on each point's x and y.

add
Uses the Edwards point addition formula to add two points together and returns the resulting point.

scalarMultiply
Preforms scalar multiplication of the point by scalar s, using the exponentiation algorithm

fromLeastSignificantBit
Given an x, returns a point (x, y) where y is the result of the equation:
y = sqrt((1-x^2)/(1+39081x^2)mod p)
And uses the sqrt method provided in the project description to calculate the square root in the equation

sqrt
The sqrt method provided in the project description

opposite
Given the current Point (x, y), this method returns it's opposite (-x, y)

negative
Given the current Point (x, y), this method returns its negation (p - x, y)

Sha_3.java
I added all the necessary functions for DHIES here because it made it more convenient to use the functions I already created. Also added a constant G which is the special point (x_0, y_0) mentioned in the project description.

dhiesKeyPair
Given a password, it generates a KeyPair using the method outlined in the project description.

dhiesEncrypt
Given a string and a public key represented by a point Point, it encrypts the string using the method outlined in the project description.

dhiesDecrypt
Given a SymmetricCryptogram and a password, it decrypts the SymmetricCryptogram using the method outlined in the project description.

dhiesSign
Given a String and a password, it signs the string using the method outlined in the project description.

dhiesVerify
Given a Signature, a string, and a point, it verifies the string using the method outlined in the project description.

KeyPair.java
A class representing a key pair (s, V) where s is the private key represented by a BigInteger, and V is the public key represented by a Point

Signature.java
A class representing a signature (h, z) where both h and z are BigIntegers

SymmectricCryptogram.java
Added a constructor for Z to be a Point instead of a byte array