CRYPS

PROJECT PROPOSAL

HSIU-YU HUANG	hh2360@columbia.edu
MINITA SHAH	mjs2225@columbia.edu
SAKET GOEL	sg2679@columbia.edu
SARFRAZ NAWAZ	sn2355@columbia.edu

CRYPS? What is it?

CRYPS is a language designed to help regular users as well as programmers perform cryptography as well as related string manipulations with ease. It has inbuilt support for numerous existing cryptographic algorithms, e.g. AES; and some predefined binary data operations, e.g. bitwise shifting.

CRYPS is intended to be used by diverse backgrounds of people. Internet users can use it to encipher messages before exchanging confidential information over mail or instant messengers. Also, it could be used for E-commerce and banking transactions requiring web authentication. Enterprises with intranet systems can encrypt employee credentials with CRYPS. Most importantly, it can be used as a tool to develop new cryptographic algorithms.

Why CRYPS?

Everyone has secrets. And people just do not want to share them with strangers! But how do you share information without letting the rest of the world know? Encode your data! There are thousands of ways to encode information; CRYPS provides the freedom and ease for users to manipulate data for his/her own purpose. This language has certain features that can assist users encipher data easily without having to learn another application software just to do some simple encryption. The data processed by CRYPS can be as small as a word or as large as a file – image files, text files and media files.

So let us get down to see more in-depths of CRYPS…

FEATURES OF CRYPS

PRIMITIVE DATA TYPES

Datatype	**Description**
String	Stores English words or phrases
File Formats	Text, Image and Media
Numbers	Integer or floating point numbers
Date	MM-DD-YYYY
Time	HHMMSS
Alphanum	Stores Hexadecimal data
Boolean	True or False
Buffer	Stores text data that cannot be held in String datatype

OPERATIONS

1. Encryption and Decryption functions
2. Creating Signature, checksum verification.
3. Bit wise operations - Left Shift, Right Shift, Ex-OR
4. Input/Output (File and text) – Open, Read, Close, Append,
5. String Manipulation – Compare, Concatenate, Append, Copy, Reversal.

SAMPLE PROGRAM

```
lib algorithms.lb

# Main function begins

main()
{
string key!                                          # Variable declaration
string[] pin, result!
number n = 10
key <= "ABCDHRYE"!                                   # Assigning a value to variable
key

for number i = 0 to 9                                # For loop begins
{
        oput "Please enter the pin :"!
        iput pin[i]?!                                # Asking the user for PIN inputs

        if pin <> " "
        then
        {
                result[i] = encryptor(pin, key[i])!  # Function call
        }

        i++
        oput "Encrypted Result :" @result[i]!        # Printing the result
}

string encryptor (a,b)
{
        a >> 2! # binary right shift
        return aesEncrypt(a,b)!             }
```

The sample code encrypts a list of PINs accepted from the user with a KEY initialized to a value in the program. The encryption of each PIN value is done in the **encryptor()** function, where the PIN is right shifted with the '>>' operator and then actually encrypted with AES algorithm, by calling the inbuilt aesEncrypt() function. The program also consists of a FOR loop construct with an IF THEN clause executed in every iteration of the loop.