ການເຂົ້າ ແລະ ຖອດລະຫັດຂໍ້ມູນຂ່າວສານ

ເກດຕິສັກ ໄຊຊະນະດາສີ*, ບຸນມີ ຄຳອິນຊຸ, ປິ່ນຄຳ ທານະວັນ ແລະ ໄພລິນ ໂພສີປອງ

ພາກວິຊາຄະນິດສາດ ແລະ ສະຖິຕິ, ຄະນະວິທະຍາສາດທຳມະຊາດ, ມະຫາວິທະຍາໄລແຫ່ງຊາດ

***ຕິດຕໍ່ພົວພັນ:** ເກດຕິສັກ ໄຊຊະນະດາສີ, ພາກວິຊາຄະນິດສາດ ແລະ ສະຖິຕິ, ຄະນະວິທະຍາສາດທຳມະຊາດ, ມະຫາວິທະຍາໄລແຫ່ງຊາດ*
ເບີໂທ: +856 20 55556446, +856 20 59977998, ອີເມວ:
k.xayxanadasy@nuol.edu.la; kedtysack@gmail.com

ຂໍ້ມູນບົດຄວາມ:
ວັນທີສະໜັກ: 10 ພະຈິກ 2020
ປັບປຸງສຳເລັດ: 04 ທັນວາ 2020
ຕອບ-ຮັບ: 18 ມັງກອນ 2021

ບົດຄັດຫຍໍ້

ການສຶກສາ ການເຂົ້າ ແລະ ຖອດລະຫັດຂໍ້ມູນຂ່າວສານ, ມີຈຸດປະສົງເພື່ອນຳໃຊ້ເຕັກນິກຕ່າງໆໃນທາງຄະນິດສາດໃນການເຂົ້າ ແລະ ຖອດລະຫັດຂໍ້ມູນຂອງການສື່ສານ ໃນຮູບແບບຕ່າງໆ ແລະ ເພື່ອຂຽນ script ດ້ວຍພາສາ Python ເຂົ້າກັບ algorithm ຂອງແຕ່ລະວິທີໃນການເຂົ້າ ແລະ ຖອດລະຫັດຂໍ້ມູນຂ່າວສານ. ຜົນຂອງການສຶກສາເຫັນວ່າ ການເຂົ້າ ແລະ ຖອດລະຫັດຂໍ້ມູນຂ່າວສານ ແມ່ນມີຄວາມຈຳເປັນຫຼາຍຕໍ່ການສື່ສານທາງດ້ານເຄືອຂ່າຍອີເລັກໂທຣນິກ ໂດຍສະເພາະແມ່ນ ທາງດ້ານທຸລະກິດ. ເພື່ອເປັນການປ້ອງກັນ ບໍ່ໃຫ້ຂໍ້ມູນຮົ່ວໄຫຼ ຫຼື ຖືກລັກລອບຈາກກຸ່ມຄົນທີ່ບໍ່ຫວັງດີ (Hacker). ໃນບົດຄວາມນີ້ ຈະໄດ້ສຶກສາການເຂົ້າ ແລະ ຖອດລະຫັດຂໍ້ມູນຂ່າວສານຊຶ່ງປະກອບມີ 3 ປະເພດຄື: ວິທີຂອງ Caesar cipher, Affine cipher ແລະ RSA cipher.

ຄຳສຳຄັນ:
ການເຂົ້າ - ຖອດລະຫັດຂໍ້ມູນ, Key, Crack, Protocol, Hacker

Cryptography for Information

Kedtysack Xayxanadasy*, Bounmy Khaminsou, Pinkham Thanavanh and Phailin Phosipong

Department of Mathematics and Statistics, Faculty of Natural Sciences, NUOL, Lao PDR

Corresponding author:
Kedtysack Xayxanadasy, Department of Mathematics and Statistics, Faculty of Natural Sciences, NUOL, Lao PDR.
Tel : +856 20 55556446, +856 20 59977998
E-mail : k.xayxanadasy@nuol.edu.la; kedtysack@gmail.com

Abstract

The aim of this study is to apply mathematics methods into the encryption and decryption for communication such as text message, email and so on. Furthermore, this includes how to write scripts with algorithm by Python programming of each method of the encryption and decryption. The result indicates that the encryption and decryption are essential for electronic communication especially business areas for protecting the data and information from hacking or hackers. In this article, We will study Three types of encryption and decryption such as Caesar cipher, Affine cipher and RSA cipher are examined in this study.

Article Info:
Submitted: Nov 10, 2020.
Revised: Dec 04, 2020.
Accepted: Jan 18, 2021

Keywords: *Cryptography, Key, Crack, Protocol, Hacker.*

1. ພາກສະເໜີ

ການເຂົ້າລະຫັດ Cryptology ໝາຍເຖິງ ການນຳໃຊ້ເຕັກນິກ ຫຼື ວິທີທາງຄະນິດສາດເຂົ້າມາປ່ຽນຂໍ້ມູນໃຫ້ກາຍເປັນລະຫັດ ດ້ວຍກົດລັບ ຫຼື ຫຼັກເກນໃດໜຶ່ງ (ໂດຍທົ່ວໄປແລ້ວເອີ້ນວ່າ: Key) ເພື່ອເຮັດໃຫ້ຂໍ້ມູນເປັນຄວາມລັບ (ສີສະເວງສັບ, 2015). ນອກນັ້ນມັນຍັງເປັນການສຶກສາກ່ຽວກັບ Cryptography ແລະ Cryptanalysis, ຊຶ່ງວ່າ Cryptography ເປັນການສຶກສາເຖິງຂະບວນການສື່ສານທີ່ມີຄວາມປອດໄພ ໂດຍຜ່ານການໃຊ້ງານກຸນແຈ (Key) ເພື່ອເຂົ້າລະຫັດ ແລະ Cryptanalysis ເປັນການສຶກສາເຖິງຂະບວນການຂອງການຖອດລະຫັດຂໍ້ມູນ (Crack) ທີ່ຜ່ານຂະບວນການ Cryptology (ສີສະເວງສັບ, 2015). ໃນອາດີດ Cryptology ໄດ້ຮັບຄວາມສົນໃຈເປັນພິເສດ, ໂດຍສະ ເພາະໃນຊ່ວງສົງຄາມ ເມື່ອແຕ່ລະປະເທດຈຳເປັນຕ້ອງໃຊ້ລະຫັດລັບ ເພື່ອສື່ສານກັບກອງທະຫານຂອງຕົນ. ໃນຂະນະດຽວກັນນັ້ນ, ກໍ່ຕ້ອງພະຍາຍາມ ຖອດລະຫັດສື່ ສານຂອງຝ່າຍສັດຕູໃຫ້ໄດ້ ເພື່ອຊອກຮູ້ ຄວາມລັບທາງທະຫານຂອງຝ່າຍສັດຕູ (Bunchman, 1997).

Applications ທີ່ເຄີຍໃຊ້ໃນສົງຄາມ ກໍ່ຍັງມີການນຳມາໃຊ້ຢູ່ໃນປະຈຸບັນ, ແຕ່ການໃຊ້ງານ Cryptography ໃນສັງຄົມຍຸກໃໝ່ນີ້ກຳລັງເປັນທີ່ນິຍົມໃນທົ່ວໂລກຂອງສື່ສັງຄົມອອນລາຍ (Internet), ການ Shift Data ໃນເຄືອຂ່າຍ ກຳລັງເພີ່ມຄວາມນິຍົມຢ່າງວ່ອງໄວ ຈົນກາຍມາເປັນພື້ນຖານຂອງການສົ່ງຂໍ້ມູນໃນເຄືອຂ່າຍນັ້ນ ມີໂອກາດຖືກດັກຈັບຂໍ້ມູນໄດ້ສະເໝີ (Bunchman, 1997). ລະຫັດຜ່ານ, ໝາຍເລກບັດ ເຄຼດິດ, ຂໍ້ມູນສ່ວນຕົວ ແລະ ອື່ນໆ. ທັງໝົດສາມາດຖືກລັກລອບເອົາຂໍ້ມູນໄປໄດ້, ຖ້າວ່າການສື່ສານນັ້ນໃຊ້ຮູບແບບ Protocol ທີ່ບໍ່ມີການເຂົ້າລະຫັດ. Protocol ທີ່ມີການເຂົ້າລະຫັດ ໄດ້ຊ່ວຍໃຫ້ ການຕິດຕໍ່ພົວພັນກັນໃນທາງທຸລະກິດໂດຍນຳໃຊ້ອິນເຕີເນັດນັ້ນ ມີຄວາມເປັນສ່ວນຕົວຫຼາຍຂຶ້ນ ແລະ ຖ້າວ່າປາສະຈາກ Secure Socket Layer (SSL) ແລ້ວ, ໝາຍເລກບັດເຄຼດິດ ແລະ Transection ທຸລະກິດອື່ນໆ ອາດກາຍເປັນສິ່ງທີ່ບໍ່ສະດວກທີ່ຈະມີການຮັບສົ່ງກັນ ຫຼື ບໍ່ມີຄວາມປອດໄພໃນໂລກຂອງສັງຄົມອອນລາຍ (Bunchman, 1997).

ຂໍ້ມູນສ່ວນຕົວທັງໝົດຂອງຜູ້ໃຊ້ ເປັນຄວາມລັບຈະໄດ້ຮັບການປົກປ້ອງດ້ວຍ Algorithms ໃນການເຂົ້າລະຫັດ (ບາງຄັ້ງກໍ່ອາດຈະບໍ່ປອດໄພພໍ) (Coutin -ho , 1999). ໃນປັດຈຸບັນ, ລະບົບການເຂົ້າລະຫັດທີ່ໄດ້ຮັບການພິສູດ ແລະ ຍອມຮັບແລ້ວວ່າປອດໄພແນ່ ນອນນັ້ນ ແມ່ນຍັງຂ້ອນຂ້າງຫ່າງໄກຫຼາຍຈາກຄວາມເປັນຈິງ ວ່າຈະນຳມາໃຊ້ໃນທາງປະຕິບັດໄດ້ ຫຼື ບໍ່ (Coutinho, 1999). ຄວາມທ້າທາຍໃນການຖອດລະຫັດຂໍ້ມູນກໍ່ເປັນສິ່ງທີ່ໜ້າຄົ້ນຫາສຳລັບ Hacker ມືໃໝ່, ແຕ່ຍິ່ງໄປກວ່ານັ້ນ ຂໍ້ມູນອັນມີຄ່າໃນທາງທຸລະກິດ ທີ່ຖືກເຂົ້າລະຫັດໄວ້ ກໍ່ຍິ່ງເປັນສິ່ງທີ່ໜ້າສົນໃຈຫຼາຍຂຶ້ນໄປອີກສຳລັບ Hacker ມືອາຊີບ (Erickson, 2017). ໃນກໍລະນີນີ້ ການປ້ອງກັນດ້ວຍການເຂົ້າລະຫັດທີ່ປອດໄພ ແລະ ໝາແໜ້ນພໍ ຈຶ່ງເປັນຫົນທາງດຽວທີ່ສາມາດລີກລ້ຽງຈາກການລັກລອບຂໍ້ມູນ. ໃນທຳນອງດຽວກັນ ລະບົບກວດສອບການບຸກລຸກດັ່ງກ່າວມັກຈະມີລາຄາແພງ, ລະບົບການສື່ສານຜ່ານທາງສັງຄົມອອນລາຍ ທີ່ມີການເຂົ້າລະຫັດ ແລະ ໃຫ້ຄວາມປອດໄພແກ່ຜູ້ໃຊ້ນັ້ນ ກໍ່ອາດເຮັດໄດ້ຍາກຂຶ້ນ ເພື່ອຈະກວດສອບຫາພຶດຕິກຳ ຫຼື ການເຮັດວຽກຂອງ Hacker ທີ່ເຂົ້າມາໃນຊ່ອງທາງການເຂົ້າລະຫັດ (Erickson, 2017). ດັ່ງນັ້ນ, ໃນທາງປະຕິບັດລະບົບການເຂົ້າ ແລະ ຖອດລະຫັດຂໍ້ມູນ ໂດຍນຳໃຊ້ເຕັກນິກ ຫຼື ວິທີການທາງຄະນິດ ສາດເຂົ້າມາປ່ຽນຂໍ້ມູນໃຫ້ກາຍ ເປັນລະຫັດນັ້ນ ໄດ້ຮັບການພິສູດ ໃນທາງຄະນິດສາດແລ້ວວ່າ Practically Secure (ປອດໄພ ແລະ ພຽງພໍທີ່ຈະໃຊ້ໃນທາງປະຕິບັດ)(Rosen.k.h., 2012) ຊຶ່ງໝາຍເຖິງ ລະບົບການເຂົ້າລະຫັດມີຄວາມປອດໄພພໍສົມຄວນ ແລະ ເຮັດວຽກໄດ້ບໍ່ຊ້າເກີນໄປ, ໃຊ້ເວລາໃນການປະ ມວນຜົນບໍ່ຫຼາຍ ແລະ ມີຄວາມສະຖຽນສູງ ໃນການຮັບສົ່ງຂໍ້ມູນປົກກະຕິ. ໃນອາດີດຜ່ານມາ ເຄີຍມີບາງລະບົບການເຂົ້າລະຫັດທີ່ບໍ່ປອດໄພຊຶ່ງອາດມີສາເຫດມາຈາກການ Implement ບໍ່ດີພໍ, ຂະໜາດຂອງກຸນແຈ ຫຼື key ສັ້ນ

ເກີນໄປ ຫຼື ເກີດມີຊ່ອງວ່າງໃນຂະບວນການເຂົ້າລະຫັດເອງ ໂດຍກົງ (Sweigart.A., 2013).

ຕໍ່ກັບຄວາມສຳຄັນຂອງບັນຫາດັ່ງກ່າວ ຈຶ່ງມີຄວາມສົນໃຈຢາກນຳໃຊ້ເຕັກນິກ ຫຼື ວິທີການທາງຄະນິດສາດ ໃນການເຂົ້າ ແລະ ຖອດລະຫັດຂໍ້ມູນ, ເພື່ອສຶກສາການເຂົ້າ ແລະ ຖອດລະຫັດຂໍ້ມູນດ້ວຍວິທີຂອງ Caesar cipher, Affine cipher ແລະ RSA cipher ແລະ ເພື່ອຂຽນ script ດ້ວຍພາສາ Python ເຂົ້າກັບ algorithm ຂອງແຕ່ລະວິທີໃນການເຂົ້າ ແລະ ຖອດລະຫັດຂໍ້ມູນ.

1. ອຸປະກອນ ແລະ ວິທີການ

ການຄົ້ນຄວ້າໃນຄັ້ງນີ້, ທີມງານຂອງພວກຂ້າພະເຈົ້າ ໄດ້ນຳໃຊ້ວິທີການທາງຄະນິດສາດ ໂດຍສະເພາະແມ່ນທິດສະດີຈຳນວນ (ນິຍາມ ແລະ ຫຼັກເກນ ການຫານຂາດ, ອຸປະຄູນລວມໃຫຍ່ສຸດ (greatest common divisor: gcd), ທະວີຄູນລວມນ້ອຍສຸດ (least common multiple: lcm), ຈຳນວນມູນ, ຄອນກຣູເອນສ໌ ແລະ ຕົວປັ້ນຂອງຄອນກຣູເອນສ໌) ມາຊ່ວຍໃນການເຂົ້າ ແລະ ຖອດລະຫັດຂໍ້ມູນ (ຊົງວິໄລ, 2014) ຊຶ່ງໄດ້ດຳເນີນຕາມຂັ້ນຕອນດັ່ງນີ້:

2.1 ຮູບແບບການສຶກສາ

ສ້າງ Lab ຈຳລອງຂອງການເຂົ້າ ແລະ ຖອດລະຫັດຂໍ້ມູນຕົວຈິງໃນລະບົບຄອມພິວເຕີ, ນຳໃຊ້ທິດສະດີຈຳນວນ ຊຶ່ງມີບົດບາດສຳຄັນຕໍ່ການເຂົ້າ ແລະ ຖອດລະ ຫັດຂໍ້ມູນ (ພະນະລາສີ, 2008), ເຂົ້າລະຫັດຂໍ້ມູນ (ປ່ຽນຂໍ້ມູນໃຫ້ເປັນຕົວເລກ, ເລື່ອນຕຳແໜ່ງຕົວອັກ ສອນແຕ່ລະຕົວ ໂດຍມີການກຳນົດຕຳແໜ່ງທີ່ເປັນຕົວເລກໃຫ້ກັບຕົວອັກສອນແຕ່ລະຕົວ) ແລະ ຖອດລະຫັດ (ເລື່ອນຕຳແໜ່ງຕົວອັກສອນແຕ່ລະຕົວ ກັບຄືນ, ປ່ຽນຕົວເລກໃຫ້ເປັນຕົວອັກສອນ).

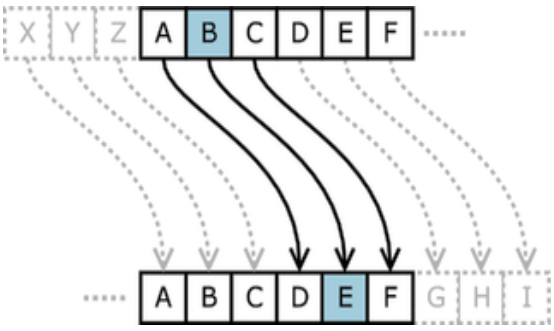
2.2 Algorithms ຂອງການເຂົ້າລະຫັດ

1.) Caesar cipher

Caesar cipher ເປັນການເຂົ້າລະຫັດແບບ Secret Key ຫຼື Symmetric key Cryptography ຄິດຄົ້ນໂດຍ Julius Caesar ເພື່ອສື່ສານກັບທະຫານໃນກອງທັບ ແລະ ປ້ອງກັນບໍ່ໃຫ້ຂໍ້ມູນຂ່າວສານຮົ່ວໄຫຼເຖິງສັດຕູ.

ຫຼັກການຂອງ Caesar cipher ຄືຈະ Shift ຫຼື ເລື່ອນຕົວອັກສອນໄປ 3 ຕຳແໜ່ງຈາກຕຳແໜ່ງເດີມ

ຮູບທີ 1: ຮູບການເລື່ອນລະຫັດຂອງ Caesar cipher



ດັ່ງນັ້ນ, ຕົວອັກສອນປົກກະຕິ ຈະຖືກແທນທີ່ດ້ວຍຕົວອັກສອນ ທີ່ຢູ່ຖັດໄປອີກ 3 ຕົວ, ໂດຍອັກສອນປົກກະຕິ ເຮົາຈະເອີ້ນວ່າ plain text ແລະ ສຳລັບຕົວອັກສອນທີ່ຖືກນຳມາແທນທີ່ ຈະເອີ້ນວ່າ cipher text.

plain text	A	B	C	D	E	F	G	H	I	J	K	L	M
cipher text	D	E	F	G	H	I	J	K	L	M	N	O	P
plain text	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
cipher text	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

ການຖອດລະຫັດກໍ່ຈະໃຊ້ການທຽບຍ້ອນກັບຄືນ 3 ຕຳແໜ່ງລະຫວ່າງ Cipher text ກັບ Plain text

cipher text	D	E	F	G	H	I	J	K	L	M	N	O	P
plain text	A	B	C	D	E	F	G	H	I	J	K	L	M
cipher text	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
plain text	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

ຂໍ້ມູນຕົວອັກສອນທີ່ປ່ຽນເປັນຕົວເລກ

plain text	A	B	C	D	E	F	G	H	I	J	K	L	M
P	0	1	2	3	4	5	6	7	8	9	10	11	12
plain text	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
P	13	14	15	16	17	18	19	20	21	22	23	24	25

ຈຸດສັງເກດຂອງ Caesar cipher ຄື Key ທີ່ໃຊ້ຈະເປັນ Key D ເນື່ອງຈາກວ່າຕົວອັກສອນຕົວທຳອິດຂອງພາສາອັງກິດຄື ຕົວອັກສອນ A, ເມື່ອຜ່ານການເຂົ້າລະຫັດຈະຖືກແທນທີ່ດ້ວຍຕົວອັກສອນ D. ດັ່ງນັ້ນ, ຈະເຫັນວ່າ Cipher ຂອງ Caesar ຈະຂຶ້ນຕົ້ນດ້ວຍຕົວອັກສອນ D, ຊຶ່ງວ່າ ຮູບແບບການເຂົ້າລະຫັດ (algorithm) ໃນທາງຄະນິດສາດແມ່ນ:

$$c = f(p) \equiv (p+k) \bmod 26,$$

$$0 \le p \le 25 \quad \text{ແລະ} \quad k \in \mathbb{Z}$$

ແລະ ຮູບແບບຂອງການຖອດລະຫັດໃນທາງຄະນິດ ສາດ ແມ່ນ

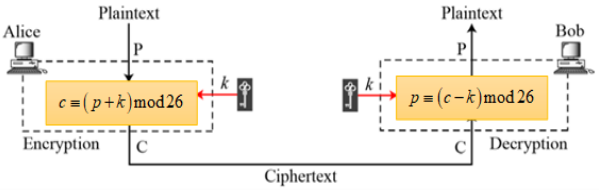
$$p = f^{-1}(c) \equiv (c-k) \bmod 26,$$

$$0 \le p \le 25 \quad \text{ແລະ} \quad k \in \mathbb{Z}$$

ຊຶ່ງວ່າ: p ແທນຕຳແໜ່ງທີ່ເປັນຕົວເລກຂອງແຕ່ລະຕົວອັກສອນ (ມີທັງໝົດ 26 ຕົວ)

k ແທນຈຳນວນຂອງຕຳແໜ່ງທີ່ຕ້ອງການເລື່ອນ ຫຼື Key.

ຮູບທີ 2: ຂັ້ນຕອນການເຂົ້າລະຫັດ ແລະ ການຖອດລະຫັດຂອງ Caesar cipher



2.) Affine cipher

ການເຂົ້າ ແລະ ການຖອດລະຫັດຂອງ Affine cipher ຈະຄ້າຍຄືກັບການເຂົ້າ ແລະ ການຖອດລະຫັດຂອງ Caesar cipher. ການເຂົ້າ ແລະ ຖອດລະຫັດດ້ວຍວິທີນີ້ ໄດ້ເພີ່ມເຕີມຈາກວິທີເກົ່າຄື: ມີຫຼັກການ ການຄູນເຂົ້າມາຕື່ມ ແລະ ເພີ່ມ key ເຂົ້າມາເປັນຕົວນຳໃນການເລື່ອນລະຫັດຕົວອັກສອນ 2 ຕົວ. ຊຶ່ງວ່າ ຮູບແບບການເຂົ້າລະຫັດ (algorithm) ໃນທາງຄະນິດສາດແມ່ນ:

$$c = E(p) \equiv (ap+b) \bmod 26,$$
$$a \in \mathbb{R} \qquad \mathbb{R} \ \ 0 \le p \le 25,$$

ແລະ ຮູບແບບຂອງການຖອດລະຫັດໃນທາງຄະນິດສາດແມ່ນ

$$p = E^{-1}(c) \equiv a^{-1}(c-b) \bmod 26,$$
$$a \in \mathbb{R} \qquad \mathbb{R} \ \ 0 \le p \le 25$$

ຊຶ່ງວ່າ: p ແທນຕຳແໜ່ງທີ່ເປັນຕົວເລກຂອງແຕ່ລະຕົວອັກສອນ (ມີທັງໝົດ 26 ຕົວ)

a, b ແທນຈຳນວນຂອງຕຳແໜ່ງທີ່ຕ້ອງການເລື່ອນ ຫຼື Key.

3.) RSA cipher

RSA ເປັນການເຂົ້າ ແລະ ຖອດລະຫັດຂໍ້ມູນແບບ key ສາທາລະນະ ແລະ key ສ່ວນຕົວ. ຜູ້ທີ່ຄິດຄົ້ນ RSA ຂຶ້ນມາໄດ້ແກ່ Ronald Rives, Adi shamir ແລະ Leonard Adleman ໂດຍຄຳວ່າ RSA ໄດ້ມາຈາກຕົວອັກສອນຕົວທຳອິດຂອງນາມສະກຸນຜູ້ຄິດຄົ້ນທັງ 3 ຄົນ.

ສຳລັບລາຍລະອຽດການສ້າງ key ສາທາລະນະ (Public key) ແລະ key ສ່ວນຕົວ (Private key) ຂອງວິທີ RSA ມີຂັ້ນຕອນດັ່ງນີ້:

ຂັ້ນຕອນທີ 1: ໃຫ້ເລືອກຈຳນວນມູນ 2 ຄ່າ, ໂດຍກຳນົດໃຫ້ເປັນຕົວປ່ຽນ p ແລະ q

ຂັ້ນຕອນທີ 2: ຊອກຄ່າຜົນຄູນລະຫວ່າງ p ແລະ q, ໂດຍກຳນົດໃຫ້ $n = p*q$

ຂັ້ນຕອນທີ 3: ຊອກຄ່າຜົນຄູນລະຫວ່າງ $p-1$ ແລະ $q-1$ ໂດຍກຳນົດໃຫ້ $\phi(n) = (p-1)*(q-1)$

ຂັ້ນຕອນທີ 4: ເລືອກຄ່າ e ຂຶ້ນມາ, ໂດຍກຳນົດ $e > 1$ ແລະ $e < \phi(n)$. ນອກຈາກນີ້ ຄ່າຂອງ e ແລະ $\phi(n)$ ຈະຕ້ອງເປັນຈຳນວນມູນຕໍ່ກັນ (e ແລະ $\phi(n)$ ຕ້ອງມີຕົວຫານຮ່ວມເທົ່າກັບ 1) ຫຼື $\gcd(e, \phi(n)) = 1$

ຂັ້ນຕອນທີ 5: ຄຳນວນຫາຄ່າ d ທີ່ຕອບສະໜອງ $ed \bmod \phi(n) = 1$

ເມື່ອຄົບທັງ 5 ຂັ້ນຕອນແລ້ວ, ພວກເຮົາຈະໄດ້ key ສາທາລະນະ (Public key) ແມ່ນ: $\gcd(n,e)$ ແລະ key ສ່ວນຕົວ (Private key) ແມ່ນ: $\gcd(n,d)$. ຊຶ່ງວ່າ ຮູບແບບການເຂົ້າລະຫັດ (algorithm) ໃນທາງຄະນິດສາດແມ່ນ:

$$c \equiv m^d \bmod n$$

ແລະ key ຂອງການຖອດລະຫັດໃນທາງຄະນິດສາດແມ່ນ:

$$m \equiv c^d \bmod n$$

ຊຶ່ງວ່າ m ແທນ key ສາທາລະນະ (Public key) ຫຼື key ສ່ວນຕົວ (Private key)

2. ຜົນໄດ້ຮັບ

ຈາກການສຶກສາ ການເຂົ້າ ແລະ ຖອດລະຫັດຂໍ້ມູນຂ່າວສານໃນຄັ້ງນີ້ພົບວ່າ ຄະນິດສາດ ແລະ ວິທີທາງຄະນິດສາດ ມີຄວາມສຳຄັນຫຼາຍຕໍ່ການນຳມາໃຊ້. ຕໍ່ໄປນີ້ຈະສະແດງຜົນການທົດສອບການຂຽນ script ດ້ວຍພາສາ Python ເຂົ້າກັບ algorithm ຂອງແຕ່ລະວິທີໃນການເຂົ້າ

ແລະ ຖອດລະຫັດຂໍ້ມູນ.

3.1 ຜົນຂອງການເຂົ້າ ແລະ ຖອດລະຫັດ

1.) ການເຂົ້າ ແລະ ຖອດລະຫັດແບບ Caesar cipher

ຂໍ້ຄວາມທີ່ຈະເຂົ້າລະຫັດ message

message = THIS IS MY SECRET MESSAGE PHAILIN

ການຂຽນ script ດ້ວຍພາສາ Python, ຜົນໄດ້ຮັບຈາກການເຂົ້າລະຫັດສະແດງໃຫ້ເຫັນ ໃນຮູບທີ 3 ແລະ ເມື່ອຖອດລະຫັດ ຈະໄດ້ຂໍ້ຄວາມ ດັ່ງຮູບທີ 4.

2.) ການເຂົ້າ ແລະ ຖອດລະຫັດແບບ Affine cipher

ຂໍ້ຄວາມທີ່ຈະເຂົ້າລະຫັດ message

message = hello this is my secret message

ການຂຽນ script ດ້ວຍພາສາ Python, ຜົນໄດ້ຮັບຈາກການເຂົ້າລະຫັດ ສະແດງໃຫ້ເຫັນໃນຮູບທີ 5 ແລະ ເມື່ອຖອດລະຫັດ ຈະໄດ້ຂໍ້ຄວາມ ດັ່ງຮູບທີ 6.

3.) ການເຂົ້າ ແລະ ຖອດລະຫັດແບບ RSA

ຂໍ້ຄວາມທີ່ຈະເຂົ້າລະຫັດ message

message = I was employed part time to develop software for Thipayal Insurance Company and I developed a website application for Phoenix Company. I entered the competition to select the delegate for the Cyber SEA games final in October 2015 and I came third and I have also completed the Training Course on Incident Handling run by NRI Secure Technologies which is a leading provider of information security solutions in Japan

ການຂຽນ script ດ້ວຍພາສາ Python, ຜົນໄດ້ຮັບຈາກການເຂົ້າລະຫັດ ສະແດງໃຫ້ເຫັນໃນຮູບທີ 7 ແລະ ເມື່ອຖອດລະຫັດ ຈະໄດ້ຂໍ້ຄວາມ ດັ່ງຮູບທີ 8.

3. ວິພາກຜົນ

ຂໍ້ມູນ ຫຼືຂໍ້ຄວາມໃດໜຶ່ງທີ່ຈະເຂົ້າ ແລະຖອດລະຫັດ ຕ້ອງມີການປ່ຽນຂໍ້ມູນໃຫ້ເປັນຕົວເລກກ່ອນ ໂດຍອາໄສວິທີທາງຄະນິດສາດເປັນຂະບວນການຜັນປ່ຽນ (ພຽງສຸຣັຍ, 2010). ຈາກນັ້ນ, ຈະເລືອກໜຶ່ງໃນວິທີໃດໜຶ່ງ (Caesar cipher, Affine cipher ຫຼື RSA cipher) ມາເປັນກຸນແຈໃນການເຂົ້າ ແລະ ຖອດລະຫັດ ໂດຍຈະມີການເລື່ອນຕຳແໜ່ງຕົວເລກແຕ່ລະຕົວ. ຫຼັງຈາກນັ້ນ, ຈະຜັນປ່ຽນຕົວເລກໃຫ້ກັບມາເປັນຕົວອັກສອນ ໂດຍຂະບວນການທາງຄະນິດສາດ ຄືນອີກເທື່ອໜຶ່ງ ເຊິ່ງເອີ້ນວ່າຂະບວນການນີ້ວ່າ ການຖອດລະຫັດ (ພະນະລາສີ, 2019). ເຊິ່ງທັງໝົດນັ້ນແມ່ນເປັນ Algorithms ທີ່ທີມງານຄົ້ນຄວ້າຂອງພວກຂ້າພະເຈົ້າ ໄດ້ຂຽນເປັນ Script ໃນພາສາໂປຼແກຣມຂອງ Python ແລ້ວນຳມາປະມວນຜົນຢູ່ໃນຄອມພິວເຕີ.

ຂໍ້ຄວາມທີ່ຖືກເຂົ້າລະຫັດຈາກ MEET YOU IN THE PARK ໂດຍວິທີຂອງ Caesar cipher ເລີ່ມຕົ້ນ ເຮົາຈະໃຫ້ຂໍ້ຄວາມປ່ຽນມາເປັນຕົວເລກ (ໄຊບຸນເຮືອງ, 2017) ເຊິ່ງຜົນໄດ້ຮັບແມ່ນ: 12 4 4 19 24 14 20 8 13 19 7 4 15 0 17 10. ຫຼັງຈາກນັ້ນ, ແທນແຕ່ລະຕົວເລກ ຫຼື ຄ່າຂອງ p ໃສ່ $c = f(p) \equiv (p+3) \bmod 26$, ແລ້ວຄິດໄລ່ຕາມຫຼັກເກນຂອງ modulo (ໄຊບຸນເຮືອງ ແລະ ວົງບຸນສີ, 2019). ດັ່ງນັ້ນ, ຈະໄດ້ຄ່າຂອງ c ແມ່ນ 15 7 7 22 1 17 23 11 16 22 10 7 18 3 20 13. ສຸດທ້າຍ, ປ່ຽນຕົວເລກໃຫ້ກັບຄືນມາເປັນຕົວອັກສອນຈະໄດ້ຂໍ້ຄວາມທີ່ຖືກເຂົ້າລະຫັດແມ່ນ PHHW BRX LQ WKH SDUN.

ຖອດລະຫັດຈາກຂໍ້ຄວາມ LEWLYPLUJL PZ H NYLHA ALHJOLY ໂດຍການເລືອກ key ຫຼື $k = 7$ ໂດຍວິທີຂອງ Caesar cipher ເລີ່ມຕົ້ນ ເຮົາຈະໃຫ້ຂໍ້ຄວາມປ່ຽນມາເປັນຕົວເລກເຊິ່ງຜົນໄດ້ຮັບແມ່ນ: 11 4 22 11 24 15 11 20 9 11 15 25 713 24 11 7 00 11 7 9 14 11 24 . ຫຼັງຈາກນັ້ນ, ແທນແຕ່ລະຕົວເລກ ຫຼື ຄ່າຂອງ c ໃສ່ $p = f^{-1}(c) \equiv (c-7) \bmod 26$, ແລ້ວຄິດໄລ່ຕາມຫຼັກເກນຂອງ modulo (ໄຊບຸນເຮືອງ ແລະ ວົງບຸນສີ, 2019). ດັ່ງນັ້ນ, ຈະໄດ້ຄ່າຂອງ p ແມ່ນ 4 23 15 4 17 8 4 13 2 48 18 0 6 17 4 0 1919 4 0 2 7 4 17. ສຸດທ້າຍ, ປ່ຽນຕົວເລກໃຫ້ກັບຄືນມາເປັນຕົວອັກ ສອນ ຈະໄດ້ຂໍ້ຄວາມທີ່ຖືກຖອດລະຫັດແມ່ນEXPERIENCE IS A GREAT TEACHER.

4. ສະຫຼຸບຜົນ

ການເຂົ້າ ແລະ ຖອດລະຫັດດ້ວຍວິທີການທາງຄະນິດສາດທັງ 3 ຮູບແບບ: Caesar cipher, Affine cipher ແລະ RSA cipher ເຫັນວ່າມີຄວາມແຕກຕ່າງກັນ ແລະ ລະດັບຄວາມປອດໄພກໍ່ແຕກຕ່າງກັນ, ຊຶ່ງການເຂົ້າ ແລະ ຖອດລະຫັດແບບ Caesar cipher ນັ້ນເປັນ

ການເຂົ້າລະຫັດແບບເລື່ອນ (ເຄື່ອນຍ້າຍຕົວໜັງສືໄປໃນຕຳແໜ່ງໃໝ່), ການເຂົ້າ ແລະ ຖອດລະຫັດໃນຮູບແບບນີ້ key ຈະເປັນຕົວກຳນົດໃນການເລື່ອນຕົວອັກສອນຂອງແຕ່ລະຕຳແໜ່ງໃນການເຂົ້າ ແລະ ຖອດລະຫັດຂໍ້ມູນ.

ສຳລັບການເຂົ້າ ແລະ ຖອດລະຫັດແບບ Affine cipher ນັ້ນຈະຄ້າຍຄືກັບການເຂົ້າ ແລະ ຖອດລະຫັດແບບ Caesar cipher ແຕ່ຈະມີຂໍ້ແຕກຕ່າງຢູ່ບ່ອນວ່າ ນອກຈາກການເພີ່ມ Key ເປັນ 2 ຕົວໄປເປັນຕົວກຳນົດໃນການເລື່ອນຕົວອັກສອນແລ້ວ ຍັງນຳໃຊ້ວິທີການຄູນ ແລະ ນຳໃຊ້ເຕັກນິກຄະນິດສາດເຂົ້າມາແກ້ໄຂບັນຫາຕື່ມ.

ດັ່ງນັ້ນ, ການເຂົ້າ ແລະ ຖອດລະຫັດແບບ RSA cipher ໄດ້ແນວຄວາມຄິດມາຈາກການເຂົ້າ ແລະ ຖອດລະຫັດແບບ Caesar Cipher ຊຶ່ງຈະເພີ່ມຄວາມສາມາດໃນການເຂົ້າລະຫັດໄດ້ດີຂຶ້ນຕື່ມ ແລະ ຮອງຮັບໃນເລື່ອງຄວາມເປັນສ່ວນຕົວ ແລະ ຢືນຢັນຕົວບຸກຄົນໄດ້ວ່າ ຂໍ້ມູນການເຂົ້າລະຫັດນັ້ນ ໄປເຖິງຜູ້ຮັບຜູ້ໃດແທ້.

5. ຄວາມຮູ້ບຸນຄຸນ

ພວກຂ້າພະເຈົ້າຂໍສະແດງຄວາມຂອບໃຈ ແລະ ຮູ້ບຸນຄຸນຢ່າງສູງ ມາຍັງການຈັດຕັ້ງທຸກຂັ້ນ ພາຍໃນພາກວິຊາ ກໍ່ຄື ຄະນະວິທະຍາສາດທຳມະຊາດ ມະຫາວິທະຍາໄລແຫ່ງຊາດ ທີ່ໃຫ້ໂອກາດ ແລະ ອຳນວຍຄວາມສະດວກທຸກຢ່າງໃຫ້ພວກຂ້າພະເຈົ້າ ບໍ່ວ່າຈະເປັນເອກະສານ, ຂໍ້ມູນຕ່າງໆໃນການຄົ້ນຄວ້າຄັ້ງນີ້. ພິເສດ ຂໍສະແດງຄວາມຂອບໃຈມາຍັງ ທ່ານ ອຈ. ທອງສຸກ ໄຊບຸນເຮືອງ (ພະແນກຄົ້ນຄວ້າ ແລະ ບໍລິການວິຊາການ) ແລະ ອຈ. ສຸລິວັນ ຫຼ້າໜໍ່ສະຫວັນ (ພາກວິຊາ ຊີວະວິທະຍາ) ຄະນະວິທະຍາສາດທຳມະຊາດ ມະຫາວິທະຍາໄລແຫ່ງຊາດ ທີ່ໃຫ້ແນວທາງໃນການຄົ້ນຄວ້າ ແລະ ຍາມໃດກໍ່ຊ່ວຍປະກອບຄຳຄິດຄຳເຫັນ ຊ່ວຍເຫຼືອ ພວກຂ້າພະເຈົ້າ ຕະຫຼອດມາ ຈົນເຮັດໃຫ້ບົດຂອງພວກຂ້າພະເຈົ້າ ສຳເລັດລົງໄດ້.

6. ເອກະສານອ້າງອີງ

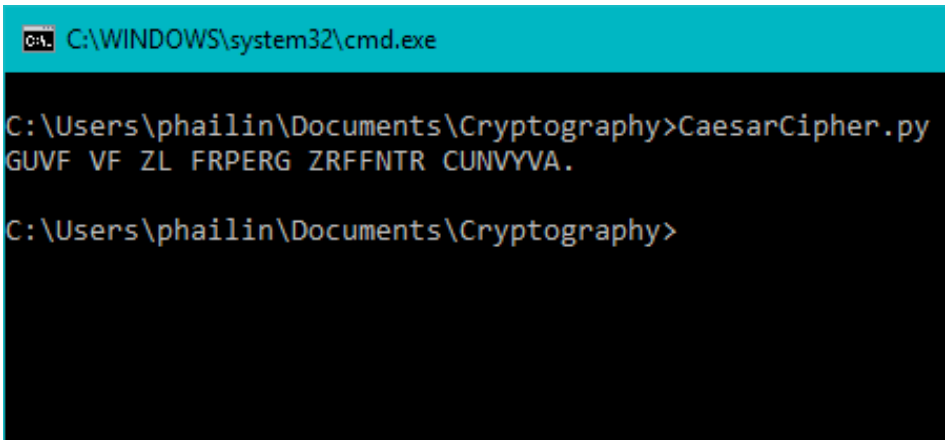
ສີສະເວງສັບ. (2015). *ລະບົບລະຫັດລັບເພີ່ມຄວາມປອດໄພໃຫ້ກັບຂໍ້ມູນ (Cryptosystem for Data Security).*

ຊົງວິໄລ, ສອນໄຊ. (2014). *ພຶດຊະຄະນິດຂັ້ນສູງ.* ພາກວິຊາຄະນິດສາດ, ຄະນະວິທະຍາສາດທຳມະຊາດ, ມະຫາວິທະຍາໄລແຫ່ງຊາດ. ນະຄອນຫຼວງວຽງຈັນ. ພິມຄັ້ງທີ 1.

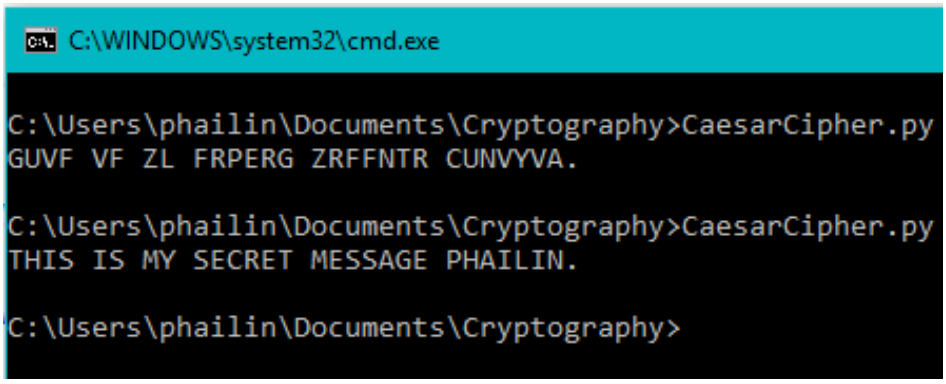
ໄຊບຸນເຮືອງ, ທອງສຸກ ແລະ ວົງບຸນສີ, ປານທອງ. (2019). *ຄະນິດສາດສຳລັບຄອມພິວເຕີ 1*. ພາກວິຊາຄະນິດສາດ, ຄະນະວິທະຍາສາດທຳມະຊາດ, ມະຫາວິທະຍາໄລແຫ່ງຊາດ. ນະຄອນຫຼວງວຽງຈັນ. ພິມຄັ້ງທີ 1.

ໄຊບຸນເຮືອງ, ທອງສຸກ. (2017). *ທິດສະດີກຸ່ມ.* ພາກວິຊາຄະນິດສາດ, ຄະນະວິທະຍາສາດທຳມະຊາດ, ມະຫາວິທະຍາໄລແຫ່ງຊາດ. ນະຄອນຫຼວງວຽງຈັນ. ພິມຄັ້ງທີ 1.

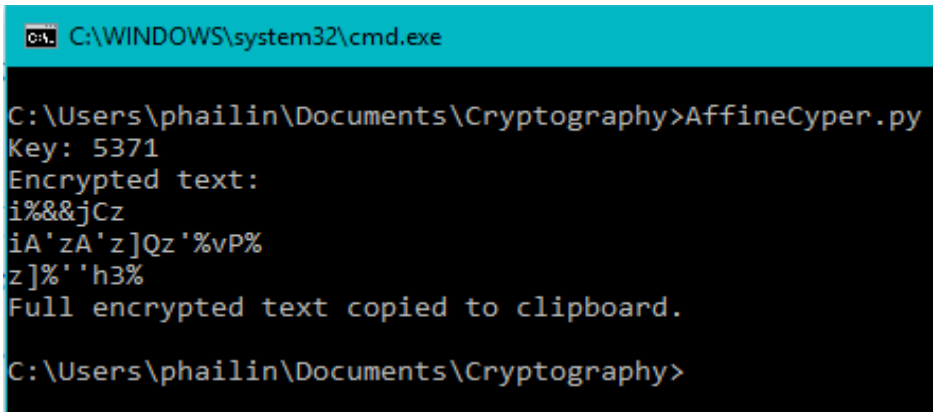
ພະນະລາສີ, ອຸດອນ. (2019). *ຄະນິດສາດເຕັມໜ່ວຍ 1*. ພາກວິຊາຄະນິດສາດ, ຄະນະວິທະຍາສາດທຳມະຊາດ, ມະຫາວິທະຍາໄລແຫ່ງຊາດ. ນະຄອນຫຼວງວຽງຈັນ. ພິມຄັ້ງທີ 1.



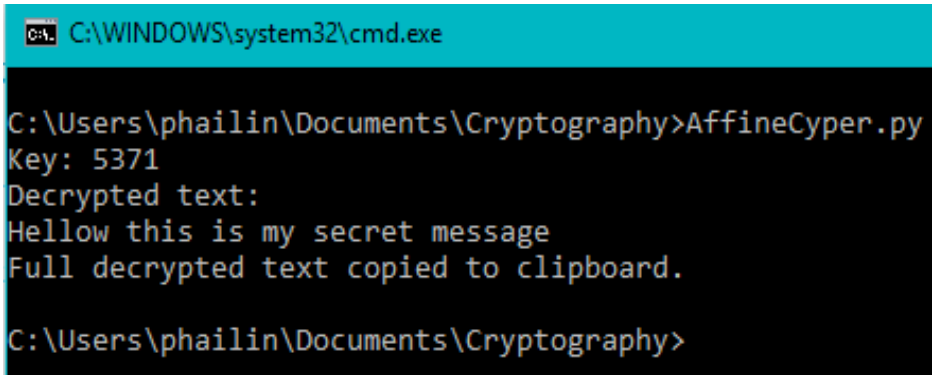
ຮູບທີ 3: ການເຂົ້າລະຫັດດ້ວຍວິທີຂອງ Caesar cipher



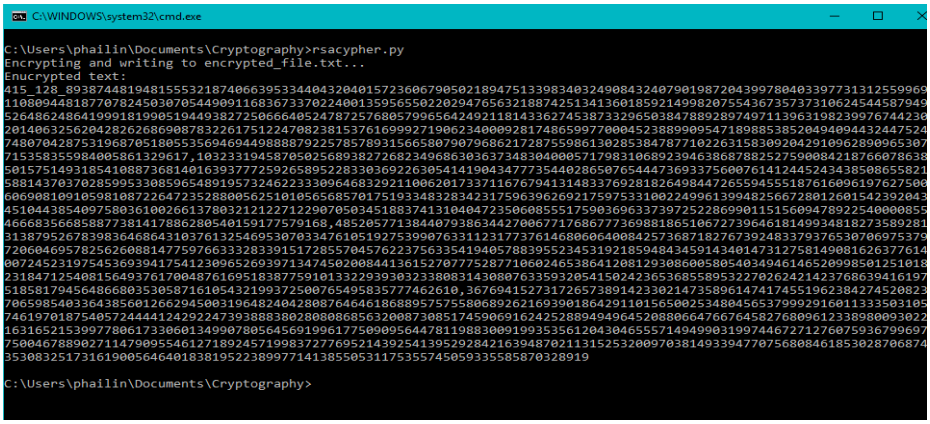
ຮູບທີ 4: ການຖອດລະຫັດດ້ວຍວິທີຂອງ Caesar cipher



ຮູບທີ 5: ການເຂົ້າລະຫັດດ້ວຍວິທີຂອງ Affine cipher



ຮູບທີ 6: ການຖອດລະຫັດດ້ວຍວິທີຂອງ Affine cipher



ຮູບທີ:7 ການເຂົ້າລະຫັດດ້ວຍວິທີຂອງ RSA cipher

ຮູບທີ 8: ການຖອດລະຫັດດ້ວຍວິທີຂອງ RSA cipher

Script ຂອງການເຂົ້າ ແລະ ຖອດລະຫັດ

```
#!/usr/bin/python
#   Caesar Cipher
#import os
#import pyperclip
# the string to be encrypted / decrypted
#message = 'This is my secret message phailin. -=+'
message = 'GUVF VF ZL FRPERG ZRFFNTR CUNVYVA. -=+'
#message = 'CUNVYVA CUBFVCBAT'
#the encryption / decryption key
key = 13
 # tells the program to encryption or decrypt
mode = 'decrypt' # set to 'encrypt' or 'decrypt'
 # every possible symbol that can be encrypted
Letters = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
 # STORE THE ENCRYPTED / DECRYPTED FORM OF THE MESSAGE
translated = ''
 # capitalize = message.upper()
message = message.upper()
 #run the encryption / decryption code on each symbol in the message string
for symbol in message:
        if symbol in Letters:
                num = Letters.find(symbol)
                if mode == 'encrypt':
                        num = num + key
                elif mode == 'decrypt':
                        num = num - key
                if num >= len(Letters):
                        num = num - len(Letters)
                elif num < 0 :
                        num = num + len(Letters)
                translated = translated + Letters[num]
        else:
                translated = translated + symbol

                print (translated)
                #pyperclip.copy(translated)

# Affine Cipher
import sys,cryptomath, random
SYMBOLS = """ !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]
^_`abcdefghijklmnopqrstuvwxyz{|}~""" # note the space at the front
def main():
        #myMessage = "hellow this is my secret message"
        myMessage="""r%&&jCz
iA'zA'z]Qz'%vP%
z]%"h3%"""
        myKey = 5371
        myMode = 'decrypt' # set to 'encrypt' or 'decrypt'
        if myMode == 'encrypt':
                translated = encryptMessage(myKey, myMessage)
        elif myMode == 'decrypt':
                translated = decryptMessage(myKey, myMessage)
        print('Key: %s' % (myKey))
        print('%sed text:' % (myMode.title()))
        print(translated)
        print('Full %sed text copied to clipboard.' % (myMode))
def getKeyParts(key):
        keyA = key // len(SYMBOLS)
        keyB = key % len(SYMBOLS)
        return (keyA, keyB)
```

```
def checkKeys(keyA, keyB, mode):
	if keyA == 1 and mode == 'encrypt':
		sys.exit('The affine cipher becomes incredibly weak when key A isset to 1. Choose a different key.')
	if keyB == 0 and mode == 'encrypt':
		sys.exit("The affine cipher becomes incredibly weak when key B isset to 0. Choose a different key.")
	if keyA < 0 or keyB < 0 or keyB > len(SYMBOLS)-1:
		sys.exit("Key A must be greater than 0 and Key B must be between 0 and %s." % (len(SYMBOLS) - 1))
	if cryptomath.gcd(keyA,len(SYMBOLS)) != 1:
		sys.exit('key A (%s) and the symbol set size (%s) are not relatively prime. Choouse a different key.'% (keyA,len(SYMBOLS)))
def  encryptMessage(key,message):
	keyA,keyB = getKeyParts(key)
	checkKeys(keyA,keyB,'encrypt')
	ciphertext = ''
	for symbol in message:
		if symbol in SYMBOLS:
			symIndex = SYMBOLS.find(symbol)
			ciphertext += SYMBOLS[(symIndex*keyA+keyB)%len(SYMBOLS)]
		else:
			ciphertext += symbol
	return ciphertext
def decryptMessage(key,message):
	keyA,keyB = getKeyParts(key)
	checkKeys(keyA,keyB,'decrypt')
	plaintext = ''
	modInverseOfKeyA = cryptomath.findModInverse(keyA,len(SYMBOLS))
	for symbol in message:
		if symbol in SYMBOLS:
			symIndex = SYMBOLS.find(symbol)
			plaintext += SYMBOLS[(symIndex - keyB)*modInverseOfKeyA %len(SYMBOLS)]
		else:
			plaintext += symbol
	return plaintext
def getRandomKey():
	while True:
		keyA = random.randint(2,len(SYMBOLS))
		keyB = random.randint(2,len(SYMBOLS))
	if cryptomath.gcd(keyA,keyB,len(SYMBOLS)) == 1:
		return keyA * len(SYMBOLS) + keyB
if __name__ == '__main__':
	main()

#RSA Cypher
import sys
#Important: The block size must be less than or equal to the key size!
#NoteL The block zie is in bytes, the key size is in bitsD There
#are 8  bits in 1 byte.
DEFAULT_BLOCK_SIZE = 128 #128 byte
BYTE_SIZE = 256 # One btye has 256 different values.
def main():
	filename = 'encrypted_file.txt'
	#mode = 'encrypt' #set to 'encrypt ' or 'decrypt'
	mode = 'decrypt'
	if mode == 'encrypt':
		# message ="""Journalists belong in the gutter because that is
		# where the ruling classes throw their guilty secrets." -Gerald Priestland "The
		# Founding Fathers gave the free press the protection it must have to bare the
		# secrets of government and inform the people." -Hugo Black'"
		message = "I was employed part time to develop software for Thipayal Insurance Company and I developed a website
application for Phoenix Company. I entered the competition to select the delegate for the Cyber SEA games final in October 2015 and I came third
and I have also completed the Training Course on Incident Handling run by NRI Secure Technologies which is a leading provider of information
security solutions in Japan"
		pubKeyFilename = 'al_sweigart_pubkey.txt'
		print('Encrypting and writing to %s...' % (filename))
		encryptedText = encryptAndWriteToFile(filename, pubKeyFilename, message)
		print('Enucrypted text:')
		print(encryptedText)
	elif mode == 'decrypt':
		privKeyFilename = 'al_sweigart_privkey.txt'
		print('Reading from %s and decrypting...' % (filename))
		decryptedText = readFromFileAndDecrypt(filename, privKeyFilename)
		print('Decrypted text:')
		print(decryptedText)
def getBlocksFromText(message, blockSize=DEFAULT_BLOCK_SIZE):
	messageBytes = message.encode('ascii')
	blockInts = []
	for blockStart in range(0, len(messageBytes), blockSize):
		blockInt = 0
		for i in range(blockStart, min(blockStart + blockSize,len(messageBytes))):
			blockInt += ord(messageBytes[i]) * (BYTE_SIZE ** (i % blockSize))
		blockInts.append(blockInt)
```

```
        return blockInts
def getTextFromBlocks(blockInts, messageLength,blockSize=DEFAULT_BLOCK_SIZE):
        message = []
        for blockInt in blockInts:
                blockMessage = []
                for i in range(blockSize - 1, -1, -1):
                        if len(message) + i < messageLength:
                                asciiNumber = blockInt // (BYTE_SIZE ** i)
                                blockInt = blockInt % (BYTE_SIZE ** i)
                                blockMessage.insert(0, chr(asciiNumber))
                message.extend(blockMessage)
        return ''.join(message)
def encryptMessage(message, key, blockSize=DEFAULT_BLOCK_SIZE):
        encryptedBlocks = []
        n, e = key
        for block in getBlocksFromText(message, blockSize):
                encryptedBlocks.append(pow(block, e, n))
        return encryptedBlocks
def decryptMessage(encryptedBlocks, messageLength, key,blockSize=DEFAULT_BLOCK_SIZE):
        decryptedBlocks = []
        n, d = key
        for block in encryptedBlocks:
                decryptedBlocks.append(pow(block, d, n))
        return getTextFromBlocks(decryptedBlocks, messageLength, blockSize)
def readKeyFile(keyFilename):
        fo = open(keyFilename)
        content = fo.read()
        fo.close()
        keySize, n, EorD = content.split(',')
        return (int(keySize), int(n), int(EorD))
def encryptAndWriteToFile(messageFilename, keyFilename, message,blockSize=DEFAULT_BLOCK_SIZE):
        keySize, n, e = readKeyFile(keyFilename)
        if keySize < blockSize * 8:
                sys.exit('ERROR: Block size is %s bits and key size is %s bits.The RSA cipher requires the block size to be equal to or less
than the key size. Either increase the block size or use different keys.' % (blockSize * 8,keySize))
        encryptedBlocks = encryptMessage(message, (n, e), blockSize)
        for i in range(len(encryptedBlocks)):
                encryptedBlocks[i] = str(encryptedBlocks[i])
        encryptedContent = ','.join(encryptedBlocks)
        encryptedContent = '%s_%s_%s' % (len(message), blockSize,encryptedContent)
        fo = open(messageFilename, 'w')
        fo.write(encryptedContent)
        fo.close()
        return encryptedContent
def readFromFileAndDecrypt(messageFilename, keyFilename):
        keySize, n, d = readKeyFile(keyFilename)
        fo = open(messageFilename)
        content = fo.read()
        messageLength, blockSize, encryptedMessage = content.split('_')
        messageLength = int(messageLength)
        blockSize = int(blockSize)
        if keySize < blockSize * 8:
                sys.exit('ERROR: Block size is %s bits and key size is %s bits.The RSA cipher requires the block size to be equal to or less
than the key size. Did you specify the correct key file and encrypted file?' % (blockSize *8, keySize))
        encryptedBlocks = []
        for block in encryptedMessage.split(','):
                encryptedBlocks.append(int(block))

        return decryptMessage(encryptedBlocks, messageLength, (n, d),blockSize)

if __name__ == '__main__':
                main()
```