การเข้ารหัสข้อมูลด้วยRSA

จัดทำโดย

นายสุริยา เสียงใส

B5913862

เสนอ

ผู้ช่วยศาสตราจารย์ ดร.ปรเมศวร์ ห่อแก้ว

โครงงานนี้เป็นส่วนหนึ่งของวิชา COMPUTER ENGINEERING PROJECT I

รหัสวิชา 523495

สำนักวิชา วิศวกรรมศาสตร์ สาขาวิชา วิศวกรรมคอมพิวเตอร์

มหาวิทยาลัยเทคโนโลยีสุรนารี

ภาคการศึกษาที่ 2 ปีการศึกษา 2561

เรื่อง : การเข้ารหัสข้อมูลด้วยRSA

ผู้จัดทำ : B5913862 นายสุริยา เสียงใส

ที่ปรึกษา : ผู้ช่วยศาสตราจารย์ ดร.ปรเมศวร์ ห่อแก้ว

ภาคการศึกษา : 2

ปีการศึกษา : 2561

บทคัดย่อ

ในปัจจุบันระบบสารสนเทศมีความเกี่ยวข้องกับตัวเราอย่างสำคัญ อาทิ การส่งจดหมายอิเล็กทรอนนิกส์ การใช้ระบบแชทโต้ตอบ การจัดเก็บข้อมูล การแลกเปลี่ยนข้อมูลระหว่างองค์กร ข้อมูลบางอย่างมีความสำคัญมากและไม่ต้องการให้บุคคลอื่นทราบ จึงมีแนวความคิดในการเก็บข้อมูลไม่ให้บุคคลอื่นทราบถึงแม้ข้อมูลจะถูกขโมย โดยใช้หลักการซ่อนข้อมูล เปลี่ยนข้อมูลให้เป็นลักษละใหม่ นั้นคือการเข้ารหัสข้อมูล โดยนำเทคนิค KEY PAIR RSA (Rivest–Shamir–Adleman) มาใช้

คำนำ

โครงงานนี้เป็นส่วนหนึ่งของรายวิชา 523295 COMPUTER ENGINEERING PROJECT I โดยมีจุดประส่งค์ศึกษาเกี่ยวกับ การสร้าง KEY ด้วย RSA และการเข้ารหัสข้อมูล

ผู้จัดทำหวังว่าโครงงานนี้จะให้ความรู้และเป็นประโยชน์แก่ผู้อ่าน หากมีข้อเสนอแนะ ประการใดผู้จัดทำขอรับไว้เพื่อนำไปพัฒนาให้ดียิ่งขึ้น

ผู้จัดทำ

สารบัญ

เรื่อง	หน้า
บทคัดย่อ	ก
คำนำ	ข
สารบัญ	ค
บทที่1 : บทนำ	1
1.1 ที่มาและความสำคัญ	1
1.2 วัตถุประสงค์	2
1.3 ขอบเขตการศึกษา	2
1.4 ผลที่คาดว่าจะได้รับ	2
บทที่2 : เอกสารที่เกี่ยวข้อง	2
2.1 ภาษา java	4
2.2 spring boot	6
2.3 angular	7
2.4 KEY PAIR RSA (Rivest–Shamir–Adleman)	9
2.5 เทคนิคการเข้ารหัสข้อมูล	16
2.6 การประยุกต์ใช้งาน RSA	18
บทที่3 : วิธีการจัดทำโครงงาน	22
3.1 วัสดุ อุปกรณ์ เครื่องมือ หรือโปรแกรมที่ใช้ในการพัฒนา	22
3.2 ขั้นตอนการดำเนินโครงงาน	22
บทที่ 4 : ผลการดำเนินงาน	33
4.1 การเข้ารหัสข้อมูล	33
4.2 การถอดรหัสข้อมูล	44

บทที่ 5 : สรุปผลการดำเนินงานและข้อเสนอแนะ 45

5.1 สรุปผลการทำ เข้ารหัสข้อมูลด้วยRSA 45

5.2 วิเคราะห์ผลการดำเนินงาน 45

5.3 อุปสรรคในการทำโครงงาน 46

บรรณานุกรม 47

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญ

ในปัจจุบันระบบสารสนเทศมีความเกี่ยวข้องกับตัวเราอย่างสำคัญ อาทิ การส่งจดหมายอิเล็กทรอนนิกส์ การใช้ระบบแชทโต้ตอบ การจัดเก็บข้อมูล การแลกเปลี่ยนข้อมูลระหว่างองค์กร ข้อมูลบางอย่างมีความสำคัญมากและไม่ต้องการให้บุคคลอื่นทราบ จึงมีแนวความคิดในการเก็บข้อมูลไม่ให้บุคคลอื่นทราบถึงแม้ข้อมูลจะถูกขโมย โดยใช้หลักการซ่อนข้อมูล เปลี่ยนข้อมูลให้เป็นลักษละใหม่ นั้นคือการเข้ารหัสข้อมูล โดยนำเทคนิค KEY PAIR RSA (Rivest–Shamir–Adleman) มาใช้

1.2 วัตถุประสงค์

1. ศึกษาหลักการทำงานสร้าง key pair rsa
2. เพื่อศึกษาการเข้ารหัสข้อมูลด้วยเทคนิค RSAเพื่อ
3. เพื่อศึกษาหาแนวทางกระบวนประยุกต์ใช้ key pair rsa กับ application ด้านอื่นๆ

1.3 ขอบเขตการศึกษา

1.ศึกษาการเกี่ยวกับการสร้าง key ด้วย rsa เท่านั้น

2. ศึกษากระบวนการเข้ารหัสข้อมูลแบบ TEXT

1.4 ผลที่คาดว่าจะได้รับ

1. สามารถสร้าง key ด้วย rsa ได้
2. สามารถเข้ารหัสข้อมูลแบบ TEXT ได้
3. สามารถประยุกต์ ใช้ rsa กับด้านอื่นๆ ได้

บทที่ 2
เอกสารที่เกี่ยวข้อง

ในการจัดทำโครงงานเรื่อง การเข้ารหัสข้อมูลด้วย RSA ผู้จัดทำได้ศึกษาและรวบรวมเนื้อหาที่เกี่ยวข้องกับโครงงานแบ่งออก เป็นดังนี้

2.1 ภาษา java

2.2 spring boot

2.3 angular

2.4 KEY PAIR RSA (Rivest–Shamir–Adleman)

2.5 เทคนิคการเข้ารหัสข้อมูล

2.6 การประยุกต์ใช้งาน RSA

2.1 ภาษา java

Java เป็นภาษาเขียนโปรแกรมเพื่อวัตถุประสงค์ทั่วไป โดยสามารถทำงานได้พร้อมกัน เป็นภาษาที่สร้างมาจากคลาส และสนับสนุนการเขียนโปรแกรมแบบออบเจ็ค ถูกออกแบบมาให้พร้อมสำหรับการใช้งาน โดยมีเมธอดและคลาสช่วยอำนวยความสะดวกมากมาย ภาษา Java นั้นมีความตั้งใจว่าจะทำให้นักพัฒนาออกแบบและพัฒนาโปรแกรมน้อยลง นั่นคือการเขียนเพียงครั้งเดียว แต่นำไปใช้งานได้ทุกที่หรือทุกแพลตฟอร์ม

แอพพลิเคชันของภาษา Java นั้นโดยปกติแล้วจะคอมไพล์เป็น bytecode ที่สามารถรันได้ใน Java virtual machine (JVM) ขึ้นกับสถาปัตยกรรมของคอมพิวเตอร์นั้นๆ และในปี 2016 Java เป็นภาษาที่ได้รับความนิยมและใช้มากที่สุดในโลก โดยเฉพาะการใช้พัฒนาเว็บแอพพลิเคชัน ภาษา Java นั้นพัฒนาโดย James Gosling ที่บริษัท Sun Microsystems (ปัจจุบันถูกซื้อกิจการเป็น Oracle Corporation) และเปิดตัวครั้งแรกเมื่อปี 1995 โดยภาษานั้นได้รับรูปแบบจากภาษา C และ C++ แต่ภาษา Java ถือว่าเป็นภาษาระดับสูงกว่าภาษาทั้งสอง

ต้นกำเนิดการอ้างอิงในการพัฒนาของ Java compiler virtual machines และคลาสไลบรารีในตอนต้นนั้นถูกเผยแพร่โดย Sun ภายใต้ลิขสิทธิที่เหมาะสม ใน May 2007 ในการปฏิบัติตามกับข้อระบุของ Java Community Process Sun ได้จดทะเบียนใหม่ในเทคโนโลยีของ Java เป็นจำนวนมากภายใต้ GNU General Public License และการพัฒนาเทคโนโลยีอื่นๆ เช่น GNU Compiler for Java (bytecode compiler) GNU Classpath (ไลบรารี่มาตฐาน) และ IcedTea-Web เวอร์ชันล่าสุดของ Java คือเวอร์ชัน 8 ที่สนับสนุนโดย Oracle ถึงแม้ในเวอร์ชันก่อนหน้าก็ได้สนับสนุนโดย Oracle และบริษัทอื่น

หนึงของวัตถุประสงค์ในการออกแบบภาษา Java คือให้มันสามารถพกพาได้ ซึ่งหมายความโปรแกรที่เขียนในภาษา Java จะต้องสามารถรัยได้กับฮาร์ดแวร์และระบบปฏิบัติการที่ได้รับการสนับสนุนจาก Java Runtime ซึ่งนี่บรรลุผลได้โดยการคอมไพล์โค้ดของ Java ให้อยู่ในรูปแบบการแสดงผลระดับกลางที่เรียกว่า Java bytecode แทนการแปลงไปยังภาษาเครื่องโดยตรง คำสั่งของ Java bytecode นั้นคล้ายคลึงกับภาษาเครื่อง แต่มันจะทำงานโดย virtual machine (VM) ที่เขียนเฉพาะสำหรับฮาร์ดแวร์แต่ละประเภท ซึ่งโดยปกติผู้ใช้ทั่วไปจะใช้ Java Runtime Environment (JRE) ที่ติดตั้งในเครื่องของพวกเขาสำหรับรัน Standalone Java applications หรือในเว็บบราวน์เซอร์สำหรับ Java applets นอกจากนี้ ภาษา Java ยังมีไลบรารี่มาตฐานในการพัฒนาแอพพลิเคชันในเรื่อง Graphics threading และ networking

2.2 spring boot

Spring เป็น lightweight สำหรับ Java Enterprise ซึ่งเกิดมาเพื่อให้ programmer สามารถเขียนโปรแกรมง่ายกว่าสมัยก่อนที่ heavyweight มากหรือที่รู้จักกันดี Enterprise JavaBeans (EJBs) และมาพร้อมกันความสามารถ dependency injection, aspect-oriented programming และ Plain Old Java Objects (POJOs) ซึ่งเป็นคลาสง่ายๆธรรมดา และที่ดีอีกอย่างคือเขียน Test ได้ง่าย

Spring Boot เป็น project นึงของ Spring Framework ที่ทำให้พัฒนา Application ได้รวดเร็วโดยที่มีการทำ Auto Configuration ทำให้ไม่ต้องเสียเวลาไป Config ทุกอย่างเองเหมือนแต่ก่อน เราสามารถสร้าง standalone application ที่ export เป็น jar หรือจะทำเป็น war แล้วนำไป deploy ที่ application server เหมือนเดิมก็ได้เช่นกัน

Spring boot เป็นเครื่องมือที่ทำให้ Developer สามารถใช้งาน Spring Framework ได้ง่ายและรวดเร็ว ลดขั้นตอนการ configuration ด้วยวิธีการทำ Auto Configuration แต่ยังไงก็ตาม Spring Boot ได้เพิ่ม Annotation ใหม่ๆเข้ามา ดังนั้นผู้ใช้ก็ต้องไปศึกษาเรียนรู้วิธีการใช้งานแต่ละตัว การทำงานของ Spring Boot ยังทำงานอยู่บน Spring Framework เพียงแค่มันทำ Interface มันให้เราใช้งานง่ายๆ ดังนั้น Developer จำเป็นต้องเข้าใจการทำงานของ Spring Framework

2.3 angular

Angular เป็น Front-end JavaScript Framework ที่ทำงานบนฝั่ง Client ที่เรานำไปสร้าง Reactive Single Page Applications (SPA) ซึ่งก็คือทุกๆหน้าจะถูกโหลดมารวมอยู่ในหน้าเดียว การคลิ๊กเปลี่ยนหน้าหรือการคลิ๊กปุ่มต่างๆ จะทำให้เรารู้สึกเหมือนเป็น Desktop Application ที่ไม่มีการโหลดเปลี่ยนหน้า Angular เป็น Model-View-Controller (MCV) และยังเป็น Model-View-ViewModel (MVVM) อีกด้วย มีการเชื่อมการทำงานระหว่าง JavaScript เข้ากับ DOM Element ของ HTML ใช้การทำงาน client-side template สามารถสร้าง template ไปใส่ไว้ในที่ที่เรากำหนดไว้ได้ และเป็น 2-way data binding เพื่อ sync Model กับ View

เดิม Angular 1 เป็น javascript และตั้งแต่ Angular 2 ขึ้นไปเป็น typescript ปัจจุบัน Angular 6 ซึ่งมีอะไรใหม่ๆเพิ่มเข้ามา

Angular เป็นหนึ่งใน Front-end framework ที่ได้รับความนิยมสูงที่สุดในปัจจุบัน พัฒนาโดย Google เพื่อนำมาใช้ในการสร้างโปรเจคแบบ SPA (Single Page Application) แปลตรงตัวเลยก็คือ application ที่มีเพียง page เดียว โดยที่ client จะติดต่อกับ server ด้วยการเรียก AJAX ไปที่ Restful API ของ server

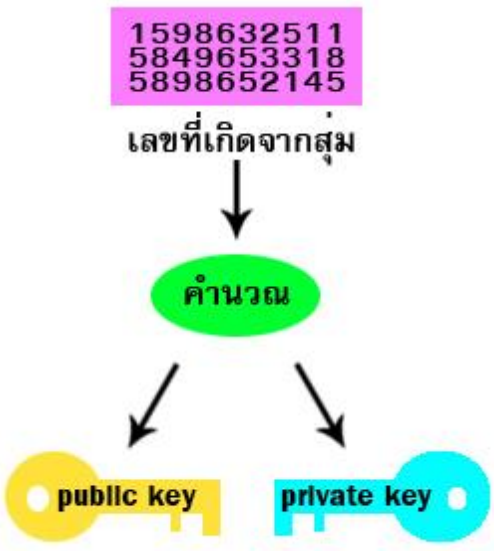
Feature หลักของ Angular

- Data-binding : การ sync ข้อมูลแบบ auto ระหว่าง controller และ view
- Scope : ส่วนที่ทำหน้าที่เชื่อมโยงระหว่าง controller กับ view
- Controller : เป็นฟังก์ชั่นที่กำหนดค่าเริ่มต้น รวมถึงควบคุมการทำงานต่างๆของ scope
- Services : Angular มี built-in services ต่างๆ ที่จะช่วยสามารถทำงานต่างๆ ได้สะดวกขึ้น เช่น $http สำหรับทำ HTTP Request, $q สำหรับจัดการ Promise
- Directives : เป็น custom HTML tag ที่เราสามารถกำหนด และควบคุมการทำงานของ tag ได้เอง
- Templates : ส่วนที่จะ render ข้อมูลที่ได้รับจาก controller
- Routing : การเปลี่ยนแปลง View ที่แสดงผล โดยไม่เกิดการ refresh ใหม่ทั้ง
- DI (Dependency Injection) : ช่วยให้เราสามารถเข้าถึงส่วนย่อยต่างๆ ของ Application ทั้งที่เป็น built-in และ custom ได้อย่างง่ายดาย

2.4 KEY PAIR RSA (Rivest–Shamir–Adleman)

การเข้ารหัสแบบ RSA เป็นอัลกอริธึมการเข้าหรัสแบบกุญแจอสมมาตร ในการเข้ารหัสโดยใช้ความรู้เรื่องเลขคณิตมอดุลาร์เข้ามาช่วยในการคำนวณ (modular arithmetic)

การเข้ารหัสแบบกุญแจอสมมาตร (public-key cryptography) เป็นการเข้ารหัสที่นิยมใช้กันอย่างแพร่หลายในการทำธุรกรรมอิเล็กทรอนิคส์ เช่นการยืนยันตัวตนด้วยระบบลายเซ็นอิเล็กทรอนิคส์ (digital signature) และการค้าผ่านอินเตอร์เน็ต (e-commerce) โดยการเข้ารหัสจะต้องมี public key และ private key ซึ่งสร้างจากตัวเลขที่สุ่มขึ้นมา และนำมาผ่านขั้นตอนของ RSA



โดย public key จะเป็นตัวที่สามารถเผยแพร่และใช้ร่วมกันได้ แต่ private key จะมีอยู่เฉพาะที่ผู้รับสารเท่านั้น หรือก็คือ การเข้ารหัสของแต่ละคนสามารถใช้ public key เดียวกันได้ แต่ในการถอดรหัสออกมาจะขึ้นอยู่กับ private key ของผู้รับสารที่จะถอดรหัส

ประวัติ

การเข้ารหัสแบบ RSA เป็นอัลกอลิธึมที่ถูกอธิบายเมื่อพ.ศ. 2520 โดย รอน ริเวสต์ (Ron Rivest) , อาดี ชามีร์ (Adi Shamir) และเล็น เอเดิลแมน (Len Adleman) ที่ MIT โดยที่ RSA นั้นเป็นตัวย่อมาจากนามสกุลของทั้ง 3 คน (Rivest-Shamir-Adleman) การเข้ารหัสแบบ RSA ได้จดสิทธิบัตรโดยสถาบัน MIT ในสหรัฐอเมริกาเมื่อปี พ.ศ. 2526 และได้สิ้นสุดลงเมื่อปี พ.ศ. 2543 เพราะเป็นผลงานที่เคยถูกตีพิมพ์เผยแพร่แล้วก่อนที่จะจดสิทธิบัตร

การสร้าง public key

- ขั้นที่ 1

  สุ่มตัวเลข 2 ตัว ที่เป็นจำนวนเฉพาะ กำหนดให้เป็น p และ q

- ขั้นที่ 2

  หาค่า n โดยที่ n=pq

- ขั้นที่ 3

  หาค่า $\varphi(n) = (p-1)(q-1)$

  (ค่า totient)

- ขั้นที่ 4

  เลือกจำนวนเต็ม e หนึ่งค่าโดยที่

  $1 < e < \varphi(n)$; e เป็นจำนวนเฉพาะสัมพัทธ์กับ $\varphi(n)$ (ไม่มีตัวประกอบร่วมยกเว้น 1)

- ขั้นที่ 5

หาค่า d โดยที่ $de \equiv 1 (\mathrm{mod}\, \phi(n))$

การเข้ารหัส

เมื่อได้ค่า d, e, n มาแล้ว ก็จะสามารถนำค่าเหล่านี้มาเข้ารหัสได้ดังต่อไปนี้

- ขั้นที่ 1

  เมื่อให้ M เป็นข้อความที่ยังไม่ได้เข้ารหัส (plain text) ให้นำ M มาเปลี่ยนเป็นตัวเลข m

  (อาจเปลี่ยนโดยวิธีการแทน a เป็น 1 b เป็น 2 ไปเรื่อยๆ) โดยที่ m < n

- ขั้นที่ 2

  นำค่า m มาคำนวณในสมการ

  $$c = m^e \bmod n$$

  โดยที่ c เป็นค่าหลังจากที่เข้ารหัสแล้ว

  ตัวอย่างการเข้ารหัส

  การสร้าง public key

- ขั้นที่ 1 สุ่มจำนวนเฉพาะ

  ให้ p และ q เป็น 11 และ 23 ตามลำดับ

- ขั้นที่ 2 หาค่า n

  n = pq

  n = 11(23) = 253

- ขั้นที่ 3 หาค่า totient $\varphi(n)$

$$\varphi(n) = (p-1)(q-1)$$

- ขั้นที่ 4 เลือกค่า e

ให้ e = 3

- ขั้นที่ 5 หาค่า d

จะได้ d = 147

ตัวอย่างการเข้ารหัส

ให้ m = 97 (สมมุติว่าเปลี่ยนมาจากข้อความ M แล้ว)

นำไปคำนวณในสมการ $c = m^e \bmod n$

จะได้ c = 102

เพราะฉะนั้นจะได้ว่าข้อความที่เข้ารหัสแล้วจะมีค่าเป็นตัวเลขคือ 102

การถอดรหัส

การถอดรหัสจะมีขั้นตอนคล้ายกับการเข้ารหัส แต่จะมีการนำค่า d ที่หาไว้มาใช้ เมื่อได้ค่า d, e, n มาแล้ว ให้นำค่าเหล่านี้มาถอดรหัสได้ดังต่อไปนี้

- ขั้นที่ 1

นำค่า c ที่เป็นตัวเลขที่เข้ารหัสแล้วมาคำนวณในสมการ $m = c^d \bmod n$

โดยที่ m เป็นตัวเลขที่ยังไม่ได้เข้ารหัส

- ขั้นที่ 2

เมื่อได้ตัวเลข m มาแล้วให้นำไปเปลี่ยนเป็นตัวอักษร M เพื่อให้ได้ข้อความที่ยังไม่ได้เข้ารหัส (plain text)

ตัวอย่างการถอดรหัส

จากตัวอย่างการเข้ารหัสที่ได้กล่าวไว้แล้ว เราจะนำค่าตัวเลขที่เข้ารหัสไว้แล้วและค่า d, e, n ที่คำนวณไว้แล้ว มาถอดรหัสได้ด้วยวิธีการดังต่อไปนี้

นำค่า c=102 มาคำนวณในสมการ $m = c^d \bmod n$

จะได้ m=97

เมื่อถอดรหัสแล้วจะได้ค่า m =97 ซึ่งตรงกับค่าตัวเลขที่ยังไม่ได้เข้ารหัส

2.5 เทคนิคการเข้ารหัสข้อมูล

การเข้ารหัส คือ การทำให้ข้อมูลเป็นความลับ เพื่อให่ข้อมูลมีคุณสมบัติดังนี้ *Confidentiality, Integrity, Authentication/Non-repudiation*

ประเภท

1. **Symmetric Encryption** หรือ **Secret Key** เข้าและถอดรหัสด้วย key เดียวกัน
    1. เอา plain text และ secret key มาเข้ารหัสด้วยอัลกอริธึม จะได้ ciphertext ออกมา
    2. ผู้เข้ารหัสส่ง/แจกจ่าย secret key ให้ผู้รับ อย่างเป็นความลับ
    3. ผู้รับได้ cipher text มา ก็ใช้อัลกอริธึมพร้อม secret key ที่มีอยู่ถอดรหัส จะได้ plain text ออกมา
2. **Asymmetric Encryption** หรือ **Public Key** เข้าและถอดรหัสด้วย key ต่างกัน
    1. **ใช้สำหรับการเข้ารหัส**
        1. เรามีกุญแจ 2 ดอก เอา Private key ไว้กับตัว เอา Public Key ไปวางในที่สาธาณะไว้แจก
        2. คนที่ต้องการจะส่งข้อความเข้ารหัสมาให้เรา ให้เค้ามาดาวน์โหลดกุญแจสาธารณะนั้น ไปทำการเข้ารหัสข้อความที่ต้องการส่งด้วยกุญแจสาธารณะ แล้วจึงส่งข้อความที่เข้ารหัสไปให้กับเราผู้เป็นเจ้าของกุญแจสาธารณะ

3. วิธีนี้จะไม่มีผู้อื่นสามารถเปิดดูข้อความที่เข้ารหัสนั้นได้ยกเว้นเราที่เป็นผู้ที่ถือกุญแจส่วนตัว (ที่เป็นคู่ของกุญแจสาธารณะนั้น) จึงจะสามารถเปิดข้อความนี้ดูได้

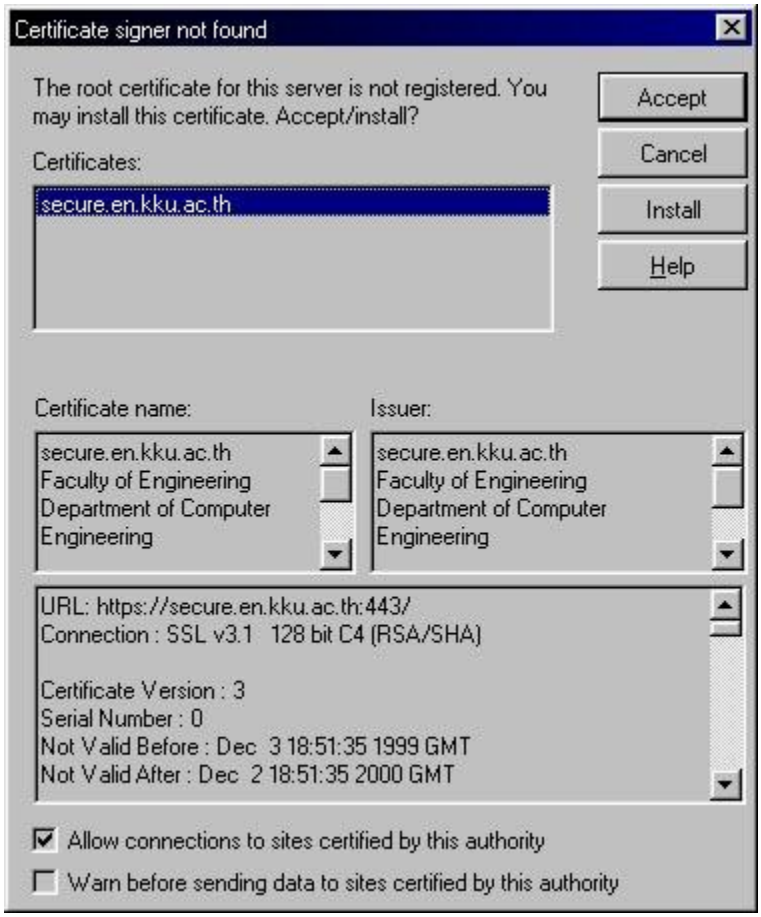
2. **ใช้สำหรับการลงลายมือชื่ออิเล็กทรอนิกส์ (Digital Signature)**

    1. ผู้ส่งนำข้อความ (Plaintext) ไปแปลงผ่าน Hash Function กลายเป็น Digest
    2. ผู้ส่งเอา Digest นั้นมาเข้ารหัสด้วย Private Key ของตัวเอง กลายเป็น Digital Signature
    3. ส่งทั้ง Plaintext และ Digital Signature ไปให้ผู้รับผ่านเน็ต
    4. ผู้รับ เอา Plaintext ไปแปลงผ่าน Hash Function กลายเป็น Digest
    5. ผู้รับ เอา Digital Signature มาถอดรหัสด้วย Public Key ที่ตนเองมี ได้ Digest
    6. ผู้รับ เอา Digest ทั้งคู่มาเทียบกันว่าตรงกันรึเปล่า

3. **Message Digest Algorithm** เป็นการเข้ารหัสทางเดียว ถอดย้อนกลับไม่ได้ จะออกมาเป็นข้อมูลที่แฮชไปแล้ว 1 ชุด มีขนาดเท่าตามกำหนด (ปกติ128-256 บิต) กันไม่ว่าข้อความตั้งต้นจะยาวแค่ไหน ปกติไว้ทดสอบ Integrity ว่าข้อมูลถูกเปลี่ยนระหว่างทางรึเปล่า

2.6 การประยุกต์ใช้งาน RSA

Secure Shell (SSH) .. SSH เป็น protocol ที่แก้ปัญหาเรื่องความไม่ปลอดภัยในการ remote login เข้าใช้งาน service ต่างๆ อย่าง telnet, X11, และ r* command ต่างๆ, etc. เนื่องจาก service เหล่านี้เวลา login จะส่ง plaintext password ผ่านเครือข่าย ซึ่งทำให้ใครๆ ก็ดักจับได้ พอ password ถูกขโมยความเสียหายอื่นๆ ก็ตามมา .. SSH จึงเสนอทางแก้โดยให้มี encryption ระหว่าง client และ server ตั้งแต่เริ่มต้น connect ดังนั้นข้อมูลที่ส่งผ่านระหว่าง SSH client และ SSH server ก็จะปลอดภัย วิธีการของ SSH อนุญาตให้ตั้ง security algorithm/protocol ได้หลายตัวเช่น IDEA, Triple-DES, IDEA, ArcFour, Blowfish, และ Twofish .. algorithms พวกนี้เป็น secret key cryptography หมดเลย ดังนั้น Key จะแลกกันโดยใช้ Public Key Cryptography ช่วย .. ภายหลัง SSH สามารถทำเป็น tunnel สำหรับให้โปรแกรมอื่นใช้งานได้ด้วย และเมื่อไม่นานนี้เองก็มีความพยายามตั้ง SSH ให้เป็นมาตรฐานในอินเทอร์เน็ต ตอนนี้ SSH เข้าเป็น working group ของ Internet Engineering Task Force (IETF) แล้วและกำลังร่างมาตรฐาน (Internet Draft) เพื่อประกาศเป็นมาตรฐาน Request For Comment (RFC) กันต่อไป


Certificate signer not found
The root certificate for this server is not registered. You may install this certificate. Accept/install?
Accept
Cancel
Install
Help
Certificates:
secure.en.kku.ac.th
Certificate name:
Issuer:
secure.en.kku.ac.th
Faculty of Engineering
Department of Computer Engineering
secure.en.kku.ac.th
Faculty of Engineering
Department of Computer Engineering
URL: https://secure.en.kku.ac.th:443/
Connection : SSL v3.1 128 bit C4 (RSA/SHA)
Certificate Version : 3
Serial Number : 0
Not Valid Before : Dec 3 18:51:35 1999 GMT
Not Valid After : Dec 2 18:51:35 2000 GMT
Allow connections to sites certified by this authority
Warn before sending data to sites certified by this authority

SSL (Secure Socket Layer) .. SSL เริ่มมาจากบริษัท Netscape ที่ทำ browser นั่นแหละครับ ลักษณะของ SSL เรียกว่าเป็น security protocol ก็คือเป็นตัวให้บริการความปลอดภัยในการสื่อสารข้อมูล SSL เป็น protocol ที่ทำงานแทรกอยู่ระหว่าง application กับ transport layer (TCP) ปัจจุบันมี service มากมายที่ทำงานกับ SSL เช่น http, ftp, telnet, pop3, smtp หรือแม้แต่ VPN การทำงานของ SSL จะเริ่มจาก server ส่ง certificate เพื่อยืนยันตัวตนกับผู้ใช้ ขั้นตอนนี้เรียกว่า authentication certificate ที่ใช้กันเป็นมาตรฐาน X.509 จะรับรอง (ด้วย digital signature) โดยผู้ที่เชื่อถือได้เช่น US Post Service หรือถ้าเป็นบริษัทที่นิยมใช้ก็จะเป็นของ VeriSign ซึ่งต้องซื้อและมีราคาแพงตาม strength ของความปลอดภัย บางเจ้าขาย 500 certificates ในราคาแสนกว่าเหรียญ แต่ certificate เหล่านี้จะตรวจสอบตัวตนจริงๆ ได้ชัวร์สุดๆ สำหรับคนที่ไม่อยากจ่าย (อย่างผม..) ก็จะใช้วิธี self signing ในการสร้าง ceritifcate คือ เซ็นเองใช้เอง (อย่าง POP3S, SMTPS ของ gear/intania จะเป็น self signed certificate, HTTPS ของ secure.en.kku.ac.th ก็เช่นเดียวกัน)

หากผู้ ใช้ยอมรับ certificate นั้นโปรแกรมก็จะเริ่มตกลงกันว่าจะใช้ protocol อะไรในการเข้าและถอดรหัส ขึ้นกับว่าโปรแกรมและตัว SSL server รองรับได้ขนาดไหน อย่าง HTTPS ของ IE4 จะใช้ RC4 stream cipher เป็น secret key cryptography ขนาด 40-bit (ซึ่งแกะได้ด้วยเครื่องซูเปอร์คอมพิวเตอร์ความเร็วสูงๆ ได้ในเวลาวินาทีเดียว) หรือถ้าเป็น IE5 ก็จะเป็น 1024-bit RSA Public Key Encryption กับ MD5/RSA Digital Signature ส่วน Opera 3.6 รองรับ SSLv3.1 จะใช้ 1024-bit RSA Public Key Encryption กับ SHA/RSA Digital Signature

PGP (Pretty-Good Privacy) เป็น public-domain program ใช้ IDEA (International Data Encryption Algorithm) เป็น algorithm สำหรับ encryption ใช้ RSA สำหรับจัดการ key และใช้ MD5 (Message Digest v.5) สำหรับสร้าง hash วิธีการสร้าง key ของ PGP จะใช้ latency ในการพิมพ์ keyboard มาเป็นตัวหาเลขสุ่ม แล้วจึงเอาเลขสุ่มนี้ไปหา key อีกที ส่งที่น่าสนใจของ PGP คือการกระจาย public key ครับ ปกติแล้วการกระจาย public key จะเป็นหน้าที่ของ key certification authorities แต่สำหรับ PGP จะให้ผู้ใช้คนไหนก็ได้ sign (digital) กำกับลงใน public key ของผู้อื่น การตรวจสอบความน่าเชื่อถือของ key ก็จะดูจาก digital signature ว่าใครเป็นคน sign ตัวอย่าง เช่นฉาก A, B, C

บทที่ 3
วิธีการจัดทำโครงงาน

3.1 วัสดุ อุปกรณ์ เครื่องมือ หรือโปรแกรมที่ใช้ในการพัฒนา

3.1.1 เครื่องคอมพิวเตอร์ หรือ โน๊ตบุ๊ค

3.1.2 โปรแกรมที่ใช้ในการพัฒนา

1) Netbean

2.) Visual Studio Code

3.) IntelliJ IDEA

3.2 ขั้นตอนการดำเนินโครงงาน

3.2.1 วางแผนขั้นตอนการดำเนินงาน

1.) คิดหัวข้อโครงงาน

2.) ศึกษาและค้นคว้าข้อมูลจาก google และ youtube

3.) ศึกษาและติดตั้งโปรแรมที่ต้องใช้

4.) พัฒนาระบบ

5.) นำเสนอโครงงานครั้งสุดท้ายต่ออาจารย์ที่ปรึกษาโครงงาน

3.2.3 การพัฒนาระบบ

1). สร้าง Key Pair ด้วย RSA

```
                private  void generateKeys() throws
Exception {

        KeyPairGenerator kpg =
KeyPairGenerator.getInstance("RSA");
        kpg.initialize(2048);
        KeyPair kp = kpg.genKeyPair();
        System.out.println("pu   " + kp.getPublic());
        System.out.println("pri  " + kp.getPrivate());
        PublicKey publicKey = kp.getPublic();
        PrivateKey privateKey = kp.getPrivate();
        System.out.println("keys created");
        KeyFactory fact = KeyFactory.getInstance("RSA");
        RSAPublicKeySpec pub = fact.getKeySpec(publicKey,
RSAPublicKeySpec.class);
        RSAPrivateKeySpec priv =
fact.getKeySpec(privateKey, RSAPrivateKeySpec.class);
        saveToFile("public.key", pub.getModulus(),
pub.getPublicExponent());
        saveToFile("private.key", priv.getModulus(),
priv.getPrivateExponent());



        System.out.println("keys saved");
    }


    private  void saveToFile(String fileName, BigInteger
mod,
                                    BigInteger exp) throws
IOException {
        ObjectOutputStream fileOut = new
ObjectOutputStream(
```

```
                new BufferedOutputStream(new
FileOutputStream(fileName)));
        try {

            fileOut.writeObject(mod);
            fileOut.writeObject(exp);



        } catch (Exception e) {
            System.out.println(e.getMessage());
            throw new IOException("Unexpected error");
        } finally {
            fileOut.close();

            System.out.println("Closed writing file.");
        }
    }
```

2). เก็บ Private Key ไว้สำหรับถอดรหัส

```
private static final String DEFAULT_FILE_NAMEkey =
"private.key";



    @GetMapping("/downloadFile/key")
    public ResponseEntity<ByteArrayResource>
downloadFilekey(
            @RequestParam(defaultValue =
DEFAULT_FILE_NAMEkey) String fileName) throws
IOException {

        MediaType mediaType
=MediaType.parseMediaType("application/octet-
stream");
        System.out.println("fileName: " +
fileName);
        System.out.println("mediaType: " +
mediaType);



        Path path = Paths.get(DIRECTORY + "\\" +
DEFAULT_FILE_NAMEkey);
        byte[] data = Files.readAllBytes(path);
        ByteArrayResource resource = new
ByteArrayResource(data);

        initRoot();

        return ResponseEntity.ok()
```

```
                // Content-Disposition

.header(HttpHeaders.CONTENT_DISPOSITION,
"attachment;filename=" +
path.getFileName().toString())
                // Content-Type
                .contentType(mediaType) //
                // Content-Lengh
                .contentLength(data.length) //
                .body(resource);
    }
```

3). เข้ารหัสข้อมูลด้วย Public Key และเก็บข้อมูลที่ถูกเข้ารหัส

```
private  void rsaEncrypt( MultipartFile file_loc, String
file_des)
            throws Exception {


            byte[] data = new byte[32];
            int i;
            System.out.println("start encyption");
            Key pubKey = readKeyFromFile("public.key");
            Cipher cipher = Cipher.getInstance("RSA");
            cipher.init(Cipher.ENCRYPT_MODE, pubKey);
            FileInputStream fileIn;
            try {
            FileInputStream fileInc = (FileInputStream)
file_loc.getInputStream();
                fileIn=fileInc;
            }
            catch (Exception e) {
                System.out.println(e.getMessage());
                throw e;
            }

            FileOutputStream fileOut = new
FileOutputStream(file_des);
            CipherOutputStream cipherOut = new
CipherOutputStream(fileOut, cipher);
            // Read in the data from the file and encrypt
it
            while ((i = fileIn.read(data)) != -1) {
                cipherOut.write(data, 0, i);
            }
            // Close the encrypted file
            System.out.println("Encode    " +
cipherOut.toString());
            cipherOut.close();
```

```
            fileIn.close();
            System.out.println("encrypted file created");
            System.out.println("output    " +
fileOut.getChannel());
        Path moveprivate = Files.move
        (Paths.get(".\\private.key"),
        Paths.get(".\\file\\private.key"));

        if(moveprivate != null)
        {
            System.out.println("File private key renamed
and moved successfully");
        }
        else
        {
            System.out.println("Failed to move the file
private key");
        }


        Path movepublic = Files.move
        (Paths.get(".\\public.key"),
        Paths.get(".\\file\\public.key"));

        if(movepublic != null)
        {
            System.out.println("File public key renamed
and moved successfully");
        }
        else
        {
            System.out.println("Failed to move the file
public key ");
        }
    }

}
```

4.) ถอดรหัสข้อมูลที่ถูกเข้ารหัส ด้วย Private Key

```
          private void rsaDecrypt( String file_des)
            throws Exception {

        int i;
        System.out.println("start decyption");
        Key priKey = readKeyFromFile();
        System.out.println("pass read key");
        Cipher cipher = Cipher.getInstance("RSA");
        cipher.init(Cipher.DECRYPT_MODE, priKey);
        FileInputStream fileIn = new
FileInputStream(fileNameEncyp);
        CipherInputStream cipherIn = new
CipherInputStream(fileIn, cipher);
        FileOutputStream fileOut = new
FileOutputStream(file_des);
        // Write data to new file
        while ((i = cipherIn.read()) != -1) {
            fileOut.write(i);
        }
        // Close the file
        fileIn.close();
        cipherIn.close();
        fileOut.close();
        System.out.println("decrypted file created");
    }
```

5.) สร้างหน้า User Interface สำหรับ เข้ารหัสข้อมูล

```
<br>

<br>
<mat-card class="position">
<div *ngIf="currentFileUpload" class="progress">
  <div class="progress-bar progress-bar-info progress-bar-
striped" role="progressbar" attr.aria-
valuenow="{{progress.percentage}}"
    aria-valuemin="0" aria-valuemax="100"
[ngStyle]="{width:progress.percentage+'%'}">
    {{progress.percentage}}%</div>
</div>
<div class="center">
<button class="center" mat-raised-button color="warn">
<label class="btn btn-default">
  <input class="center" type="file"
(change)="selectFile($event)">
</label>
</button>
</div>
<br>
เลือกไฟล์ TEXT ที่ต้องการเข้ารหัส
<br><br>
<button  class = "center" mat-raised-button
color="primary" [disabled]="!selectedFiles"
(click)="upload()">Upload</button>
<br>
<br>
<button  class = "center" mat-raised-button
color="primary"  (click)="downloadEncrypFile()">Download
Encryption File</button>
  
<button  class = "center" mat-raised-button color="warn"
(click)="downloadkeyFile()">Download Key File</button>

</mat-card>
```

6.) สร้างหน้า User Interface สำหรับ ถอดรหัสข้อมูล

```
<br>
<br>
<mat-card class="position">
<div *ngIf="currentFileUpload" class="progress">
  <div class="progress-bar progress-bar-info progress-bar-
striped" role="progressbar" attr.aria-
valuenow="{{progress.percentage}}"
    aria-valuemin="0" aria-valuemax="100"
[ngStyle]="{width:progress.percentage+'%'}">
    {{progress.percentage}}%</div>
</div>
<div class="center">
<button class="center" mat-raised-button color="warn">
<label class="btn btn-default">
  <input class="center" type="file"
(change)="selectFile($event)">
</label>
</button>
</div>
<br>
เลือกไฟล์ที่ต้องการถอดรหัส
<br><br>
<button  class = "center" mat-raised-button
color="primary" [disabled]="!selectedFiles"
(click)="uploaden()">Upload Encryption File</button>
<br>
<br>
<div *ngIf="currentFileUploadkey" class="progress">
  <div class="progress-bar progress-bar-info progress-bar-
striped" role="progressbar" attr.aria-
valuenow="{{progresskey.percentage}}"
    aria-valuemin="0" aria-valuemax="100"
[ngStyle]="{width:progress.percentage+'%'}">
    {{progress.percentage}}%</div>
</div>
<div class="center">
<button class="center" mat-raised-button color="warn">
```

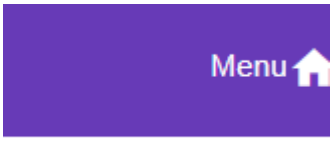
```
<label class="btn btn-default">
  <input class="center" type="file"
(change)="selectFilekey($event)">
</label>
</button>
</div>
<br>
ใส่กุญแจเพื่อถอดรหัส
<br><br>
<button  class = "center" mat-raised-button
color="primary" [disabled]="!selectedFileskey"
(click)="uploadkey()">Upload Key File</button>
<br>
<br>
<button  class = "center" mat-raised-button
color="primary"  (click)="downloadDecrypFile()">Download
Decryption File</button>
</mat-card>
```

บทที่ 4

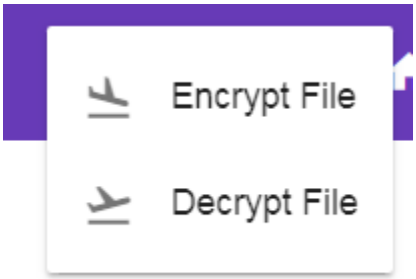
ผลการดำเนินงาน

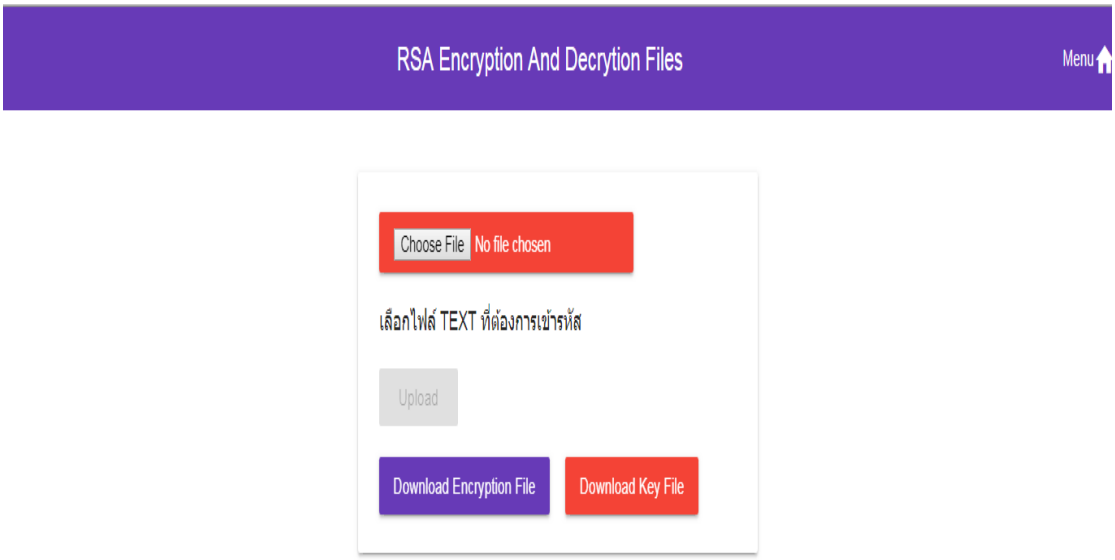
4.1 การเข้ารหัสข้อมูล

คลิก Menu

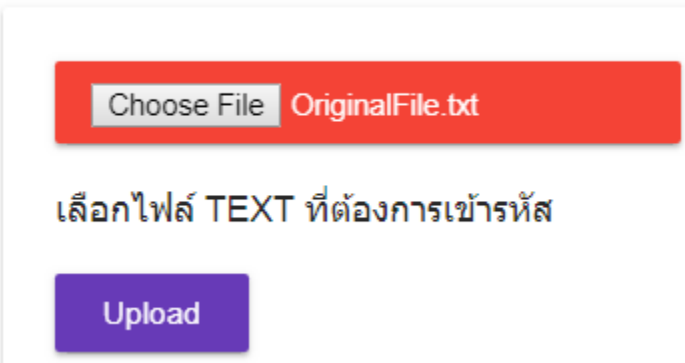


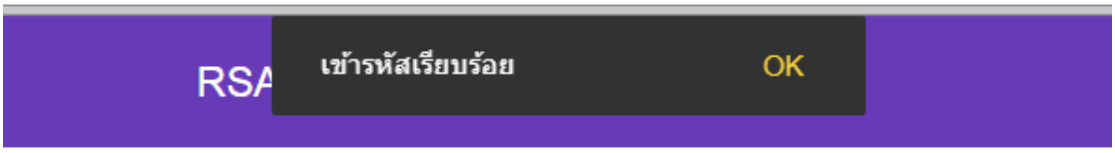
เลือก Encrypt File

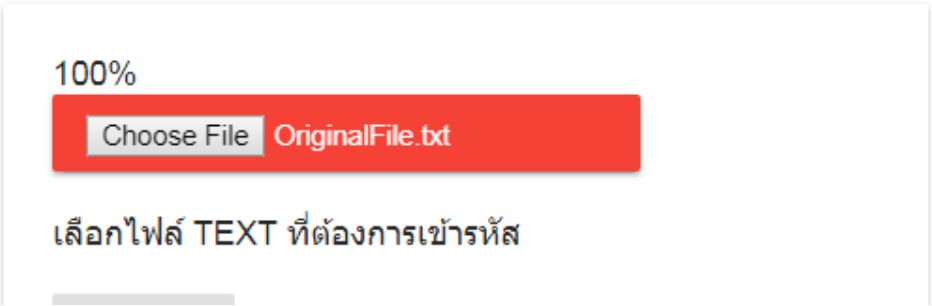




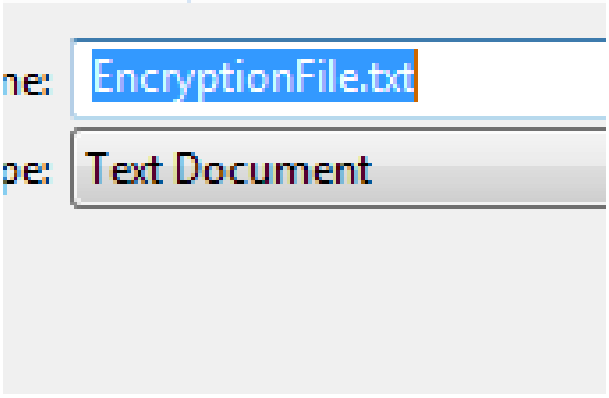
เลือกไฟล์ คลิก Upload



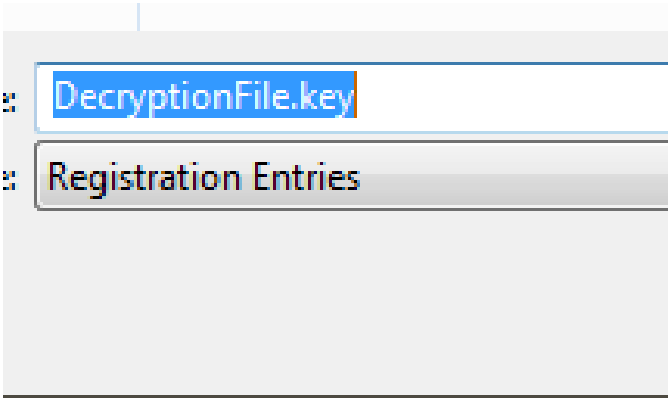




คลิก Download Encryption File แล้ว save



คลิก Download Key File แล้ว save



ข้อมูลก่อนเข้ารหัส

suriya siangsai B5913862

ข้อมูลที่ถูกเข้ารหัส

㚑疇□춈塚ꀀ 샛틿릴ꀁ 腊珞客ῷ㖞砝ɛ纋㫼鸲
렐뤏峠纃ʆ鲅□紆죵斵碣ल粗蓎ᥰ鷗җ纀϶护
앗□繠□矴㊍庌⸗뇡犌□繿□褈□絨□図□軓嫢ɛ́
迨ḝ씍 좗 ∣ 탠 弓 褚□뵯 煉 릺□沥廒ؠ鋕□羝塢24Ꮩ
ȼʒʕ峙憢옛ˊϒ□黽ʊ艺ロ꼍 妢）□□□鳮□霌8
坟ڗ□ڎ□□抽□□饅ǀ瀸艒ꝺ 끕 阼鳇□췝

Key สำหรับถอดรหัส

¬í sr java.math.BigIntegerŒüŸ©;û————————————— I

bitCountI bitLengthI firstNonzeroByteNumI

lowestSetBitI signum[magnitudet [Bxr java.lang.Number†¬•

”à‹ xpÿÿÿÿÿÿÿÿÿÿþÿÿÿþ ur [B¬óøTà xp □ØO^Ñ°

J’»»«†¢hZïšµÇOñnÓ%:ûÁœœo°òá«Lh)·éŽ9"eœÛæo¤üH9>ñåÞ@øzâ+úÛ

————————————íÜéì{rX?^z®,=□□z‡Ãb{ËàÎýà°Æ½|

[åÑjì,{Séµs————————————¥—d˜

ÿZGïÜº¹{Ê´PÆÂæN%·ä·Ušk‡ê•ÙEuÿ:äsäZQÚ}Z#t™–¡

Ûö□-¤€è9Gy*nºG¿šgŠ¼m

„m¸-33

¥’CwHIø+‘ƒ¥Z’□Ž‘ÀqÒè ¦ˆ„›“(‰ç(Ò— 4û8“*2GTÊÙõ!ó_/xsq ~

ÿÿÿÿÿÿÿÿÿÿÿÿþÿÿÿþ uq ~

Ï˝

Fα¨ÑD

ï‘M<ä½´‡{µrc¤ØjLÂm·é-wŒòÛ¶KÒ»5:#6½PŸÌ»>“kŸ×…— ¶¾"F‘…

ÊjV,CTVšÙ²-w„é:îâ

BPìN

Û»jôÌJúö~ÄAå¦ !ìUyÁ˜éÌ–½EÜØ,æ:¸@!ÈHØtÁ

z&k½

Q?àèÈÚ|ph÷ÏÉâÞµ.ç"|Ÿ

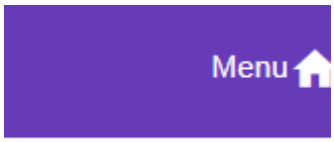
vïÙ®µXåiÅ\ÈÃfTO¯à4d'E;eõù^íª¯”h„t/æÌ‘[Õ

————————————ƒp□žn²í□o¸ □ç@]èÜ“&□ÞñÝo}è›ì`©×(39¨’;?žÜ‡ƒ'

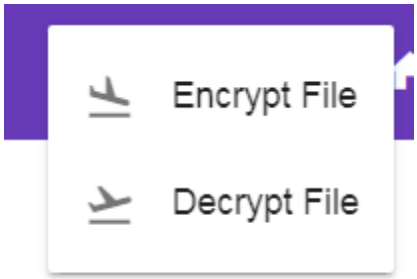
‰ox

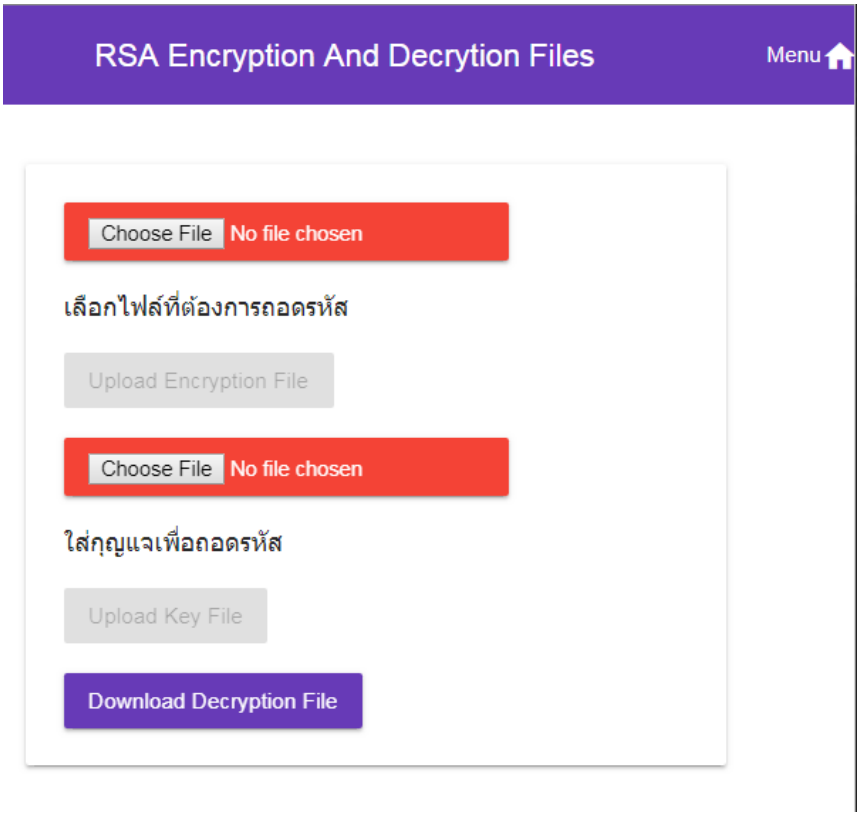
4.2 การถอดรหัสข้อมูล

คลิก Menu

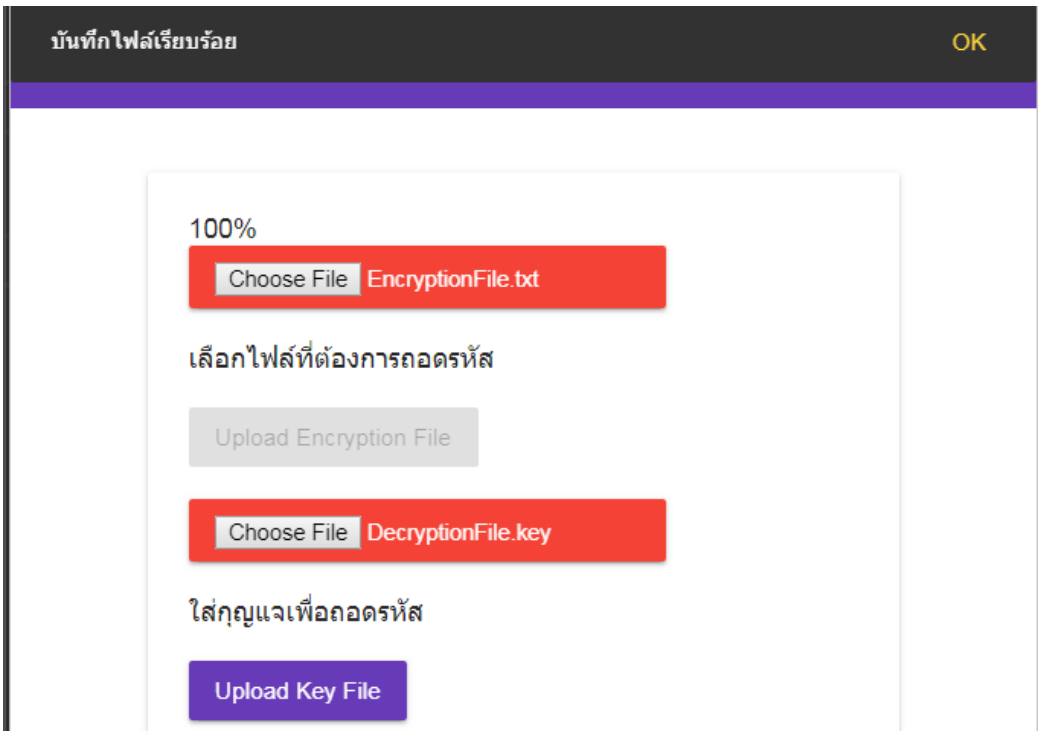


เลือก Decrypt File

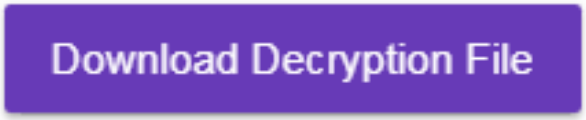


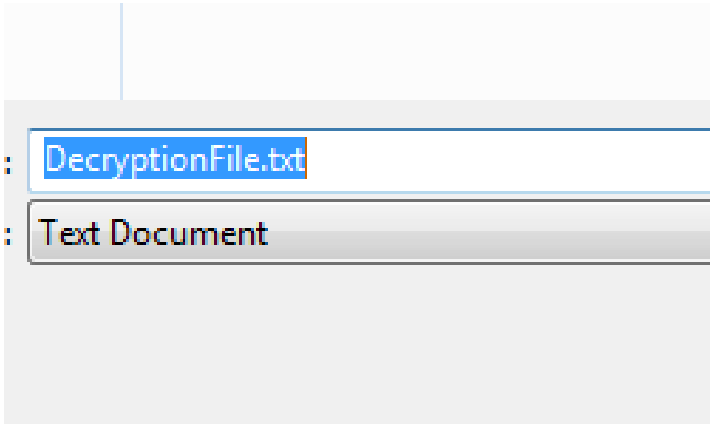


Upload ข้อมูลที่ถูกเข้ารหัสและKeyสำหรับถอดรหัส



คลิก Download Decrytion File จะได้ ข้อมูลที่ถูกถอดรหัส





บทที่ 5

สรุปผลการดำเนินงานและข้อเสนอแนะ

5.1 สรุปผลการทำ เข้ารหัสข้อมูลด้วยRSA

ผู้จัดทำได้พัฒนา การเข้าข้อมูลด้วยRSA โดยมีลายละเอียดดังนี้

- ศึกษาหาข้อมูลเกี่ยวกับการเข้ารหัสข้อมูล
- สร้าง Key Pair ด้วย RSA
- เก็บ Private Key ไว้สำหรับถอดรหัส
- เข้ารหัสข้อมูลด้วย Public Key และเก็บข้อมูลที่ถูกเข้ารหัส
- ถอดรหัสข้อมูลที่ถูกเข้ารหัส ด้วย Private Key
- สร้างหน้า User Interface สำหรับ เข้ารหัสข้อมูล
- สร้างหน้า User Interface สำหรับ ถอดรหัสข้อมูล
- นำเสนอโครงงานครั้งสุดท้ายต่ออาจารย์ที่ปรึกษาโครงงาน

5.2 วิเคราะห์ผลการดำเนินงาน

จากผลการดำเนินงาน ข้อมูลที่ถูกนำเข้าserver หลังจากเข้ารหัสเรียบร้อยจะถูกลบออกทั้งหมด ไม่มีการจัดเก็บข้อมูลใดๆทั้งสิ้น รวมทั้ง Private Key และ Public Key ส่วนการถอดรหัสข้อมูลก็ทำนองเดียวกัน ข้อมูลที่ถูกเข้าแล้วจะไม่สามารถเข้ารหัสซ้ำได้อีก ส่วนการถอดรหัสหากใส่ key ไม่ถูกต้องจะไม่สามารถถอดรหัสข้อมูลได้ ข้อมูลที่ได้จะว่างเปล่า

5.3 อุปสรรคในการทำโครงงาน

ผู้จัดทำในช่วงแรก ยังไม่มีความรู้เกี่ยวกับเทคนิคการเข้ารหัสมูลด้วยวิธีอื่นๆ และไม่รู้หลักการสร้าง key ด้วย RSA ทำให้มีความสับสนต่อแนวทางพัฒนาระบบ

ข้อเสนอแนะและแนวทางในการพัฒนาต่อ

1. ควรพัฒนาการสร้าง key ด้วย RSA ด้วยขั้นตอนและวิธีการต่างกัน แล้วนำมาเปรียบเทียบประสิทธิภาพในการสร้าง key
2. ควรเข้ารหัสข้อมูลด้วยวิธีการต่างๆ และนำผลมาเปรียบกับวิธีการเข้ารหัสด้วย RSA แล้วนำมาวิเคราะห์ผล
3 . ควรนำ key ที่สร้างขึ้นไปประยุกต์ใช้กับ application อื่นๆ

บรรณานุกรม

ภาษา Java. [ออนไลน์ .]

สืบค้นจาก : http://marcuscode.com/lang/java

สืบค้นเมื่อ : 20 กุมภาพันธ์ 2562

Angular คืออะไร ทำความรู้จัก และวิธีใช้งาน. [ออนไลน์ .]

สืบค้นจาก : http://www.helloho.me/getting-started-with-angular/

สืบค้นเมื่อ : 20 กุมภาพันธ์ 2562

เริ่มต้นทำความรู้จักกับ Spring Boot. [ออนไลน์ .]

สืบค้นจาก : http://assanai.com/getting-started-spring-boot/

สืบค้นเมื่อ : 20 กุมภาพันธ์ 2562

การเข้ารหัสแบบ RSA. [ออนไลน์ .]

สืบค้นจาก : https://writesara.wordpress.com/2008/04/10A-rsa/

สืบค้นเมื่อ : 20 กุมภาพันธ์ 2562

การเข้ารหัส. [ออนไลน์ .]

สืบค้นจาก : http://vzrnote.blogspot.com/2015/11/blog-post.html

สืบค้นเมื่อ : 20 กุมภาพันธ์ 2562

PUBLIC KEY CRYPTOGRAPHY .. เทคโนโลยีกุญแจสาธารณะ. [ออนไลน์ .]

สืบค้นจาก : https://kitty.in.th/index.php/articles/public-key-cryptography/

สืบค้นเมื่อ : 20 กุมภาพันธ์ 2562