

1). ให้นิสิตออกแบบ Documents ในการเก็บข้อมูล User และ Profile ที่ใช้สำหรับการ authentication และ authorization (ส่งเป็น data decryption)

Authentication คือการยืนยันตัวตน ก็คือการบอกว่าคนที่กำลังจะเข้ามาในระบบนั้นคือใคร และเป็นคนๆ นั้นจริงหรือเปล่า ตัวอย่างเช่น

1. ขั้นตอนการตรวจสอบว่าผู้ถือบัตรนิสิตมีใบหน้าตรงกับรูปบนบัตร เรียกว่า Authentication
2. โรงพยาบาลที่เคาท์เตอร์จ่ายยา จะเรียกชื่อจริง และให้คนไข้เอ่ยนามสกุล เพื่อตรวจสอบว่าเป็นคนที่ เรียกจริง เป็นการทำ authentication

Authorization เป็นการมอบสิทธิให้กับผู้ใช้ คือก่อนที่ผู้ใช้สักคนจะทำอะไรได้จะต้องได้รับสิทธิในการกระทำนั้น ๆ ก่อน ดังที่เราน่าจะเคยเห็นประตูที่เขียนว่า Authorized Person only หรือเฉพาะผู้ที่ได้รับอนุญาตเท่านั้นตัวอย่างการใช้งานเช่น

1. ถ้า login เข้าเว็บบอร์ดด้วย account ระดับ admin สามารถเข้าไปแก้ไขข้อมูลในเว็บไซต์ได้ ขั้นตอนการกำหนดสิทธิให้ account ที่ login เข้ามาด้วยระดับ admin สามารถแก้ไขข้อมูลได้ เรียกว่า authorization

User: เป็นตารางรหัสของผู้ใช้งานใช้สำหรับการเข้าสู่ระบบเพื่อใช้บริการต่างของเว็บหรือแอปพลิเคชัน

Attribute	Data type	Key	Data format	Description	Example
User ID	INT(11)	PK	xxxxx	id	001
Username	String(60)		nnnnnn	username	สมชาย
Password	String(60)		nnnnnn	password	อ๊อ

Profile: เป็นตารางแสดงรายละเอียดข้อมูลของผู้ใช้งาน

Attribute	Data type	Key	Data format	Description	Example
User ID	INT(11)	PK	xxxxx	id	001
Username	String(60)		nnnnn	username	สมชาย
Password	String(60)		nnnnn	password	อีอี
Name	String(60)		nnnnn	name	สมชาย
Last name	String(60)		nnnnn	last name	5 นาที่
Email	String(60)		n@mail	email	admin@mail.com
Birthday	date /time		00/00/00 00:00	date of birth	26/02/2561 17:30
Timestamp	date /time		00/00/00 00:00	date stamp	26/02/2561 17:30
Status	String(60)		nnnnn	status login	ADMIN

Authentication: เป็นตารางแสดงการล็อกอินเข้าสู่ระบบโดยระบบจะให้ Token สำหรับไว้เช็คเว็บเซอรัวิส
เพื่อใช้งานและกำหนดสิทธิ์ในการใช้งาน Authorization

Attribute	Data type	Key	Data format	Description	Example
User ID	INT(11)	PK	xxxxx	id	001
TOKEN	String(60)		nnnnn	Token	62896ecfac3f4c3ea20aaf58c0d6fbc0
Authorization	String(60)		nnnnn	Authorization	admin

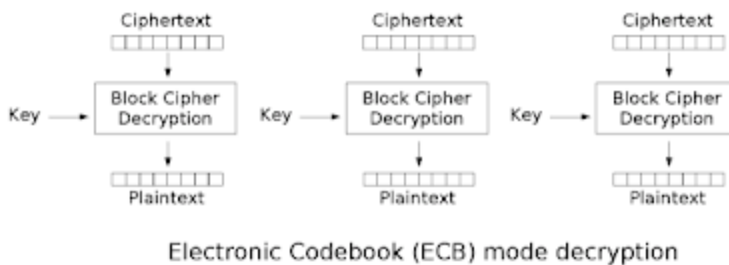
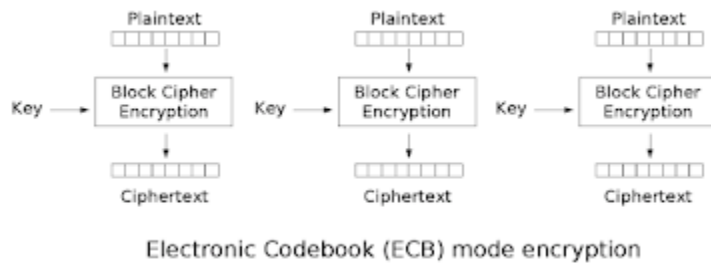
Authorization: เป็นการบอกว่า Token นั้นสามารถเข้าถึงระบบอะไรได้บ้าง

Attribute	Data type	Key	Data format	Description	Example
User ID	INT(11)	PK	xxxxx	id	001
Name	String(60)		nnnnn	Name	All system
Authorization				Authorization	

2). การ encryption และ decryption สำหรับเข้ารหัส password

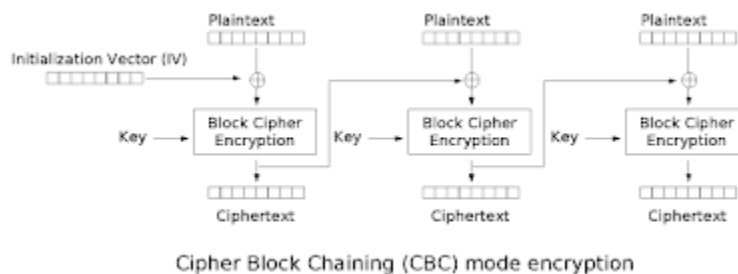
วิธีการเข้ารหัส และถอดรหัส aes-128-cbc

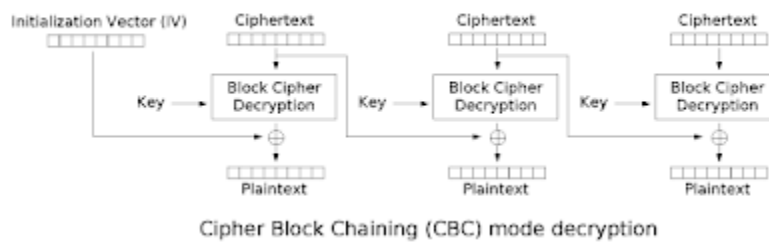
encryption algorithm โดยทั่วไป จะทำการเข้ารหัสเป็น block เช่น AES จะมี block size เป็น 128 bits (16 bytes) ซึ่งถ้าเรานำ key มาเข้ารหัสข้อมูลทีละ block ก็จะเรียก mode นี้ว่า ECB (Electronic Codebook) ตามรูปข้างล่าง (รูปจาก http://en.wikipedia.org/wiki/Block_cipher_modes_of_operation)



รูปข้างบนสำหรับ encryption และรูปข้างล่างสำหรับ decryption ปัญหาของ mode นี้ไม่ขอกว่าในนี้ เพราะว่ามันจะมีอยู่แล้วในหนังสือ cryptography ทั่วไป

CBC (Cipher Block Chaining) mode ซึ่งมีการเข้ารหัสตามภาพข้างล่าง (รูปจาก http://en.wikipedia.org/wiki/Block_cipher_modes_of_operation)





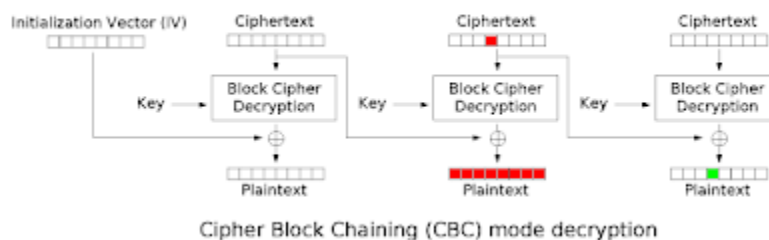
IV (Initialization Vector) คือค่าเริ่มต้น โดยปกติจะได้รับการ random ขนาดเท่ากับ block operation ของ algorithm นั้นๆ และเพื่อให้่ายต่อการอธิบายต่อ ผมจะขอเรียกผลลัพธ์จากการ decryption ที่ยังไม่ทำ XOR ว่า "Intermediate value"

ใน mode นี้ ก่อนที่จะนำข้อมูลไปเข้ารหัส จะไปทำ XOR กับผลลัพธ์จากการเข้ารหัสของ block ก่อนหน้า ส่วน block แรกจะทำการ XOR กับ IV ส่วนการทำ decryption ก็จะทำในทางกลับกัน ตามรูปข้างบน

เมื่อเข้าใจ CBC mode กันแล้ว คราวนี้ก็มาถึง attack หนึ่งเพื่อทำ bit flipping ซึ่ง attack ที่ขั้นตอนการทำ XOR โดย XOR มีคุณสมบัติดังนี้

1. $A \text{ xor } 0 = A$
2. $A \text{ xor } 1 = \text{not } A$
3. ถ้า $A \text{ xor } B = C$ แล้ว $B \text{ xor } C = A$, $C \text{ xor } A = B$

จากรูปการ decryption ของ CBC mode ถ้าเราสนใจใน block ที่ 3 จะเห็นว่า intermediate value ของ block ที่ 3 จะทำการ XOR กับ encrypted block ที่ 2 ได้ผลลัพธ์ออกมา ดังนั้นถ้าเราแก้ค่าใน encrypted block ที่ 2 ไป 1 bit ผลของการ decryption ใน block ที่ 3 ก็จะไปเปลี่ยนไป 1 bit แต่ผลของการ decryption ใน block ที่ 2 จะเปลี่ยนแปลงทั้งหมด ตามรูปข้างล่าง



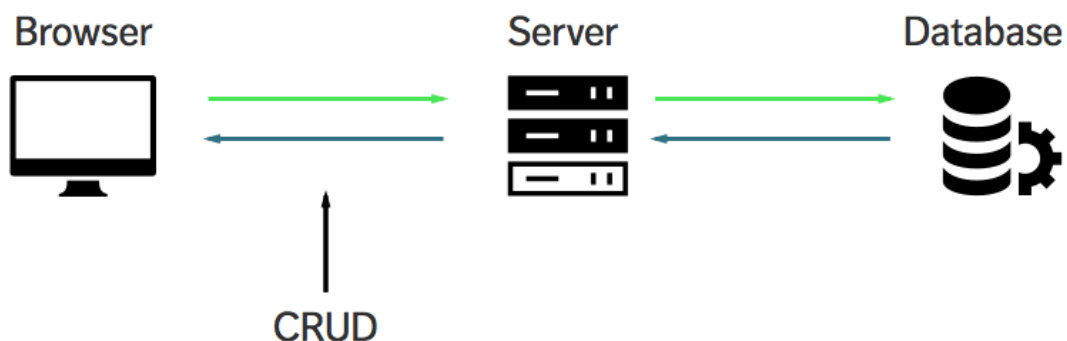
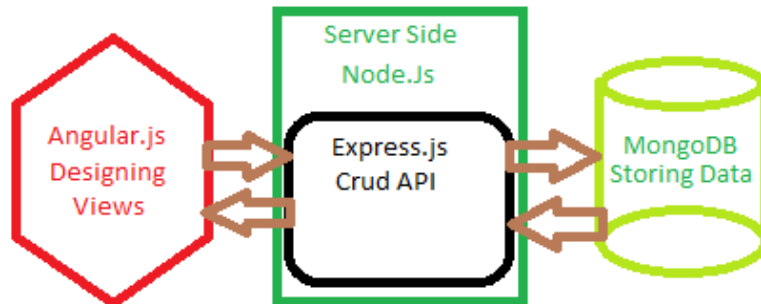
attack นี้จะทำได้ เราต้องรู้ว่า plaintext บางส่วน(หรือทั้งหมด) แล้วเราต้องการเปลี่ยนค่าบ้างค่า และที่สำคัญข้อมูลใน block ที่ถูกเปลี่ยนแปลงทั้งหมดนั้น เป็นเพียงข้อมูล ที่ไม่ผลต่อการทำงานของโปรแกรมที่อ่านข้อมูล

ถ้าเรารู้ข้อมูล plaintext และตำแหน่งทั้งหมด เราสามารถที่จะเปลี่ยนข้อมูลได้ทั้ง block เนื่องจากเราสามารถหา intermediate

value ของ block ที่เราจะเปลี่ยนได้จากการ XOR กันของ plaintext กับ encrypted block ก่อนหน้า และนำ plaintext ที่เราต้องการจะให้เป็นไป XOR กับ intermediate value ก็จะได้ encrypted block ก่อนหน้าที่เราต้องเปลี่ยน

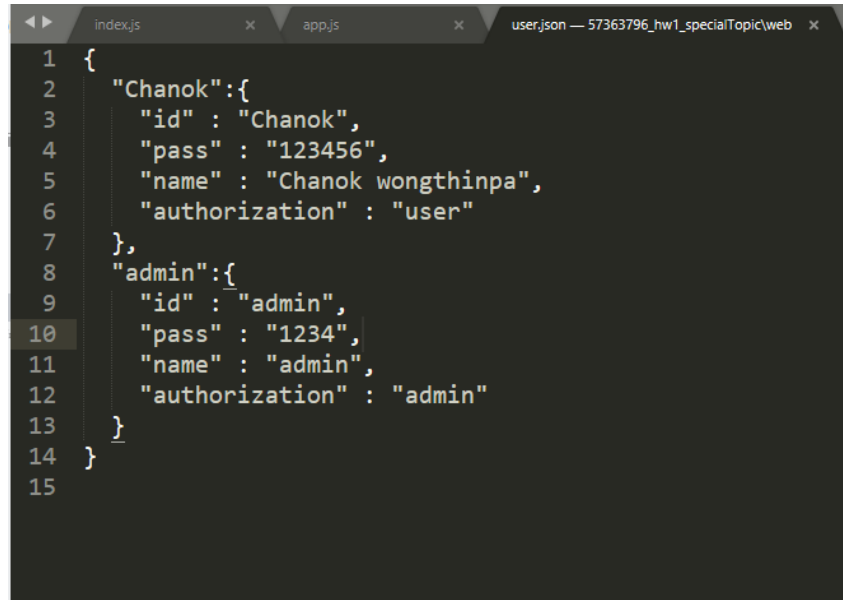
ถึงแม้ว่า attack นี้จะมีมานานแล้ว แต่ก็ถือว่าเป็นตัวอย่างที่ดีว่า การทำ encryption ช่วยได้เฉพาะเรื่องความลับของข้อมูล (Confidential) แต่ไม่ได้ป้องกันเรื่อง ความถูกต้องของข้อมูล (Integrity)

3).ออกแบบและพัฒนา web service สำหรับการ authentication และ authorization



User.json

สำหรับเก็บ userid, password ,name และ authorization สำหรับกำหนดสิทธิ์

A screenshot of a code editor with three tabs: 'index.js', 'app.js', and 'user.json'. The 'user.json' tab is active and displays the following JSON code:

```
1 {  
2   "Chanok":{  
3     "id" : "Chanok",  
4     "pass" : "123456",  
5     "name" : "Chanok wongthinpa",  
6     "authorization" : "user"  
7   },  
8   "admin":{  
9     "id" : "admin",  
10    "pass" : "1234",  
11    "name" : "admin",  
12    "authorization" : "admin"  
13  }  
14 }  
15
```

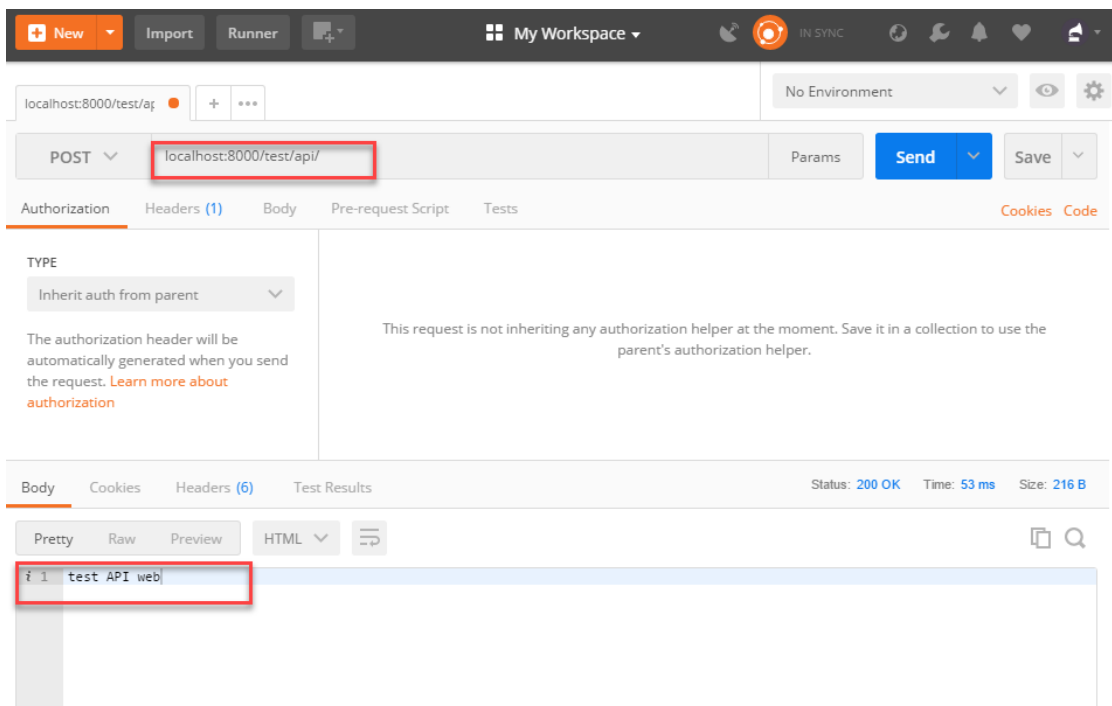
รูปที่ 1

API Method: Post

URL: localhost:8000/test/api

Params: Null

Descriptions: ทดสอบว่าสามารถติดต่อ service ได้



รูปที่ 1

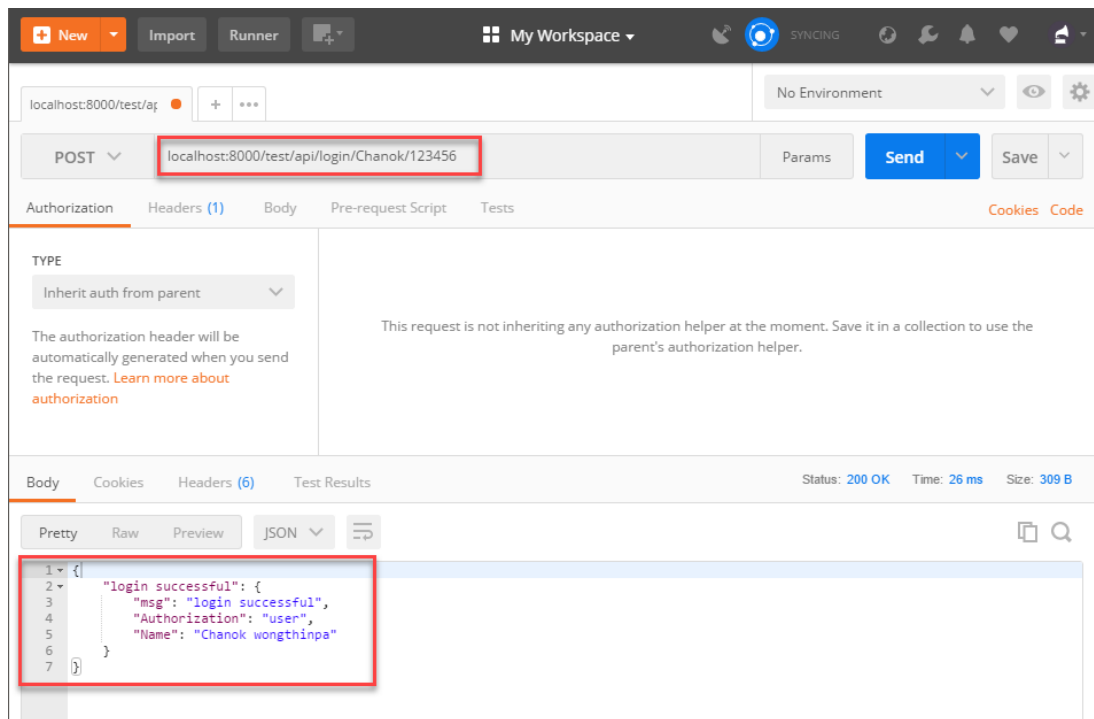
รูปที่ 2

Login Method: Post

URL: localhost:8000/test/api/login/[ID]/[Pass]

Params: Id, Pass

Descriptions: Loginโดยส่ง id และ pass มาทาง พารามิเตอร์



รูปที่ 2