



# Unit 10 Security in IoT





# การรักษาความปลอดภัยของระบบ IoT



ระบบคอมพิวเตอร์ที่ไม่ถูกแฮ็คจะเป็นระบบในอุดมคติโดยปกติแล้วเทคโนโลยีที่นำมาใช้ในการสร้าง IoT นั้นจะมีระบบรักษาความปลอดภัยอยู่แล้ว เช่น Wi-Fi และการส่งข้อมูลผ่านระบบอินเทอร์เน็ตซึ่งมีความปลอดภัยค่อนข้างสูง ยังสามารถที่จะถูกเจาะได้เนื่องจากการส่งข้อมูลจาก tier1 ไปถึง 4 มีด้วยกันหลายขั้นตอนจึงต้องมีการคิดและออกแบบการรักษาความปลอดภัยของข้อมูลในทุกๆขั้นตอนของการส่ง ทำให้เป็นงานที่/ต้องอาศัยความเชี่ยวชาญและความเข้าใจระบบเป็นอย่างมาก





แนะนำวิธีการที่สามารถนำไปปรับใช้กับการส่งข้อมูลในระบบ IOT

1. การใช้งาน POST Method ในการส่งข้อมูลผ่าน Web socket
2. การเข้ารหัสข้อมูลโดยใช้ AES library
3. การสร้างอัลกอริธึมในการเข้ารหัสด้วยตนเอง
4. การเข้าถึงด้วยการใช้ serial number





# HTTP Request Method

-  คือ วิธีการสื่อสารระหว่าง client และ server เช่น การส่งค่าไป-มา ระหว่าง client และ server ที่จะเป็นแบบ request และ response โดยสำหรับ IoT แล้วการส่งค่าจาก microcontroller ก็สามารถใช้โปรโตคอล HTTPRequest เพื่อส่งข้อมูลไปยังฐานข้อมูลที่เป็นฝั่ง server ได้แต่จริงๆ อุปกรณ์ IoT สามารถเป็นได้ทั้ง server และ client ซึ่งจะใช้คำสั่งที่แตกต่างกัน คือ **HTTPServer** และ **HTTPClient**
-  วิธีการ (method) ส่งข้อความผ่าน HTTP สามารถส่งได้ 2 แบบคือ **GET** และ **POST**





# GET and POST



ทั้งคู่เป็นวิธีการส่งค่าพารามิเตอร์ผ่าน HTTPRequest แต่จะแตกต่างกันที่ GET แสดงค่าบน URL เป็นข้อมูลจริงที่กำลังถูกส่งมาในเวลานั้น จะสามารถถูกสโตนแอมและผู้เจาะระบบจะสามารถอ่านค่าได้ทันทีแต่ POST จะนำข้อมูลที่ส่งผ่านไปในส่วน of body ของ HTTP และจะไม่ถูกแสดงบน URL อีกทั้งความปลอดภัยของข้อมูลจะขึ้นอยู่กับความปลอดภัยของ HTTP protocol ของเว็บไซต์ที่/ทำการรับค่า (http หรือ https)



# ○○○ ส่วนประกอบของการส่งข้อมูลแบบ HTTPRequest บน Arduino IDE



ชื่อ website (Server name) : “http://www.rmutt.ac.th”



Folderย่อย (path) : “/user/student/”



โดยปกติแล้วเวลานำไปใช้จะรวมเข้าด้วยกัน เรียกว่า server IP หรือ server name และบางครั้งก็ใช้เป็นการระบุ IP แทน



“http://www.rmutt.ac.th/user/student/”

“203.158.103.85/user/student/”



ชื่อ file : “data\_receive.php”




การส่งค่าด้วย HTTP Request ค่าจะถูกส่งไปยังไฟล์เพื่อเข้าคำสั่งในการ insert ค่าลงฐานข้อมูลในกรณีที่ใช้ SQL ซึ่งใน IoT จะทำการรวมชื่อไฟล์กับ server name เข้าด้วยกัน



“http://www.rmutt.ac.th/user/student/data\_receive.php” หรือ “203.158.103.85/user/student/data\_receive.php”

# ○○○ ส่วนประกอบของการส่งข้อมูลแบบ HTTPRequest บน Arduino IDE


 ในท้ายที่สุดจะเป็นการระบุค่าที่จะบันทึกลงฐานข้อมูล เช่น ส่งค่า username ลงฐานข้อมูล column ชื่อ username

 `username="john", temp=temp, sensor_data=data`

 หากมีข้อมูลหลายชุดจะถูกคั่นด้วย "&"

 `username="john"&password=xxxxx&age=age`

---

 ข้อมูลจะถูกแยกจากชื่อเว็บไซต์ด้วย "?"

 `www.rmutt.ac.th/user/student/data_receive.php?username="john"`

---

 ควรใช้การส่งแบบ POST เพราะข้อมูลจะถูกเข้ารหัสซึ่งขึ้นอยู่กับ HTTP protocol ด้วยเช่นกัน ทำให้ไม่สามารถถูกอ่านได้หากถูกเจาะระบบหาก HTTP Protocol มีความปลอดภัย เช่น https

 `http.POST(httpRequestData);`



# Data Encryption การเข้ารหัสข้อมูล



คือ การกระทำใดๆ ต่อข้อมูลเพื่อให้รูปลักษณ์ของข้อมูลนั้นเปลี่ยนไปเพื่อป้องกันบุคคลหรือโปรแกรมอื่นไม่พึงประสงค์สามารถอ่านข้อมูลได้เพื่อปกปิดข้อมูลไว้เป็นความลับจากผู้ส่งถึงมือผู้รับ



ข้อมูลปกติ เรียกว่า **Plaintext**



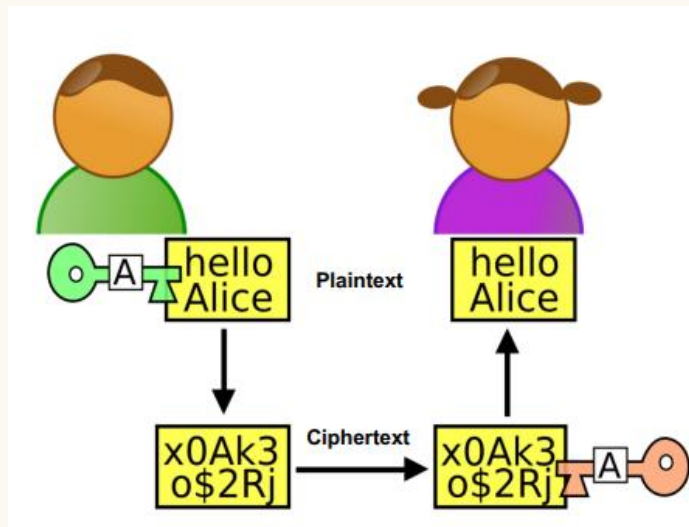
ส่วนข้อมูลที่ถูกเข้ารหัสแล้ว เรียกว่า **Ciphertext** ดังนั้น ผู้ส่งจะแปลงข้อมูลปกติให้เป็น Ciphertext จากนั้น ผู้รับจะทำการแปลง Ciphertext กลับสู่ข้อมูลปกติเพื่ออ่านข้อมูลนั้นๆ



สิ่งที่ใช้ในการเข้ารหัส เรียกว่ากุญแจ (Key) คือ วิธีในการเข้ารหัส เช่น การสลับตำแหน่งตัวอักษร แปลงเป็นเลขฐาน 16



ผู้รับและผู้ส่งจะต้องมี Key ในการเข้าและถอดรหัสจึงจะสามารถอ่านข้อมูลได้โดยที่ผู้อื่นจะต้องไม่ทราบรหัส



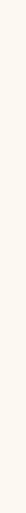


# Data Encryption การเข้ารหัสข้อมูล



มี 2 ประเภท

1. Symmetric Key (Secret Key, private key, กุญแจส่วนตัว, กุญแจสมมาตร)
2. Asymmetric Key (Public Key, กุญแจสาธารณะ, กุญแจไม่สมมาตร)







# 1. Symmetric Key (single key)



จะใช้กุญแจเดียวกันในการเข้าและถอดรหัส หมายความว่าผู้รับและผู้ส่งจะต้องถือกุญแจตัวเดียวกันในการรับและส่งข้อมูลจากรูปจะเห็นว่า Bob และ Alice ใช้กุญแจ A ในการเข้ารหัส



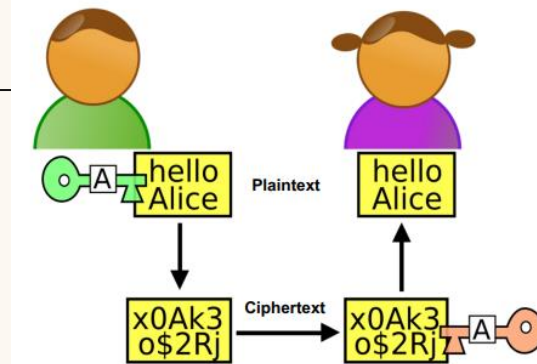
ข้อดี : มีความรวดเร็วกว่าและใช้ทรัพยากรน้อย การเข้ารหัสมีความปลอดภัยสูงมาก ซึ่งหน่วยงานรัฐบาล USA ก็ใช้การเข้ารหัสแบบนี้ในการรักษาความปลอดภัยของข้อมูล



ข้อเสีย

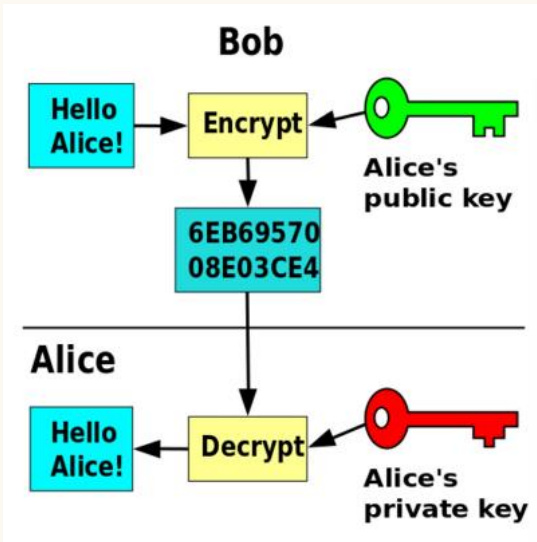
1. ทั้งผู้รับและผู้ส่งจะต้องเจอกันก่อนเพื่อตกลงรับทราบถึง Key ที่จะใช้ในการเข้าและถอดรหัส บางครั้งอาจจะเป็นการยากที่จะทำให้ทั้งสองมาเจอกันได้ หรือในบางกรณีอาจจะเป็นไปไม่ได้เลยที่ทั้งสองจะมาเจอกัน ในกรณีนี้จะต้องใช้การเข้ารหัสอีกรูปแบบคือ Asymmetric Key

2. จะต้องแชร์กุญแจให้กับผู้อื่น ซึ่งผู้ที่ถือกุญแจนั้นจะสามารถถอดรหัสของทุกๆ ข้อมูลที่ถูกเข้ารหัสด้วยกุญแจตัวนั้น





## 2. Asymmetric Key



จากรูปจะเห็นว่า Bob เป็นผู้ส่งข้อมูลและใช้กุญแจสาธารณะ (Public Key) ที่ Alice สร้างขึ้นเพื่อทำการเข้ารหัสจากนั้นส่งข้อมูลหา Alice โดย Alice มี key อีกตัวหนึ่งใช้สำหรับถอดรหัส ซึ่ง key ตัวนี้ Alice จะมีแต่เพียงผู้เดียวไม่สามารถบอกใครได้ เรียกว่า กุญแจส่วนตัว (Private Key)



ข้อดี : ไม่ต้องมีการแลกเปลี่ยนกุญแจกันก่อน เช่น การนัดเจอกันเพราะการเข้ารหัสและถอดรหัสจะใช้ key คนละตัว



ข้อเสีย : ใช้เวลานานและใช้ทรัพยากรมากกว่าในการถอดรหัสเพราะการถอดรหัสจะใช้ key คนละตัวกับการเข้ารหัส ทำให้การส่งข้อมูลแบบนี้มีความซับซ้อนมากกว่า



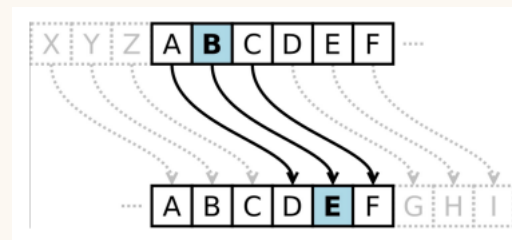
# Caesar Cipher (รหัสซีซาร์)



รูปแบบการเข้ารหัสแบบซีซาร์ใช้หลักการแทนที่ตัวอักษร จากตัวอย่างแสดงให้เห็นถึงการแทนที่ตัวอักษรด้วยตัวอักษรที่อยู่ถัดไป 3 ตัว



ดังนั้น ตัวอักษร "A" จะถูกแทนที่ด้วย "D" และ "B" จะถูกแทนที่ด้วย "E" ไปเรื่อยๆ



ธรรมดา (plain text): ABCDEFGHIJKLMNOPQRSTUVWXYZ  
รหัส (cipher text): XYZABCDEFGHIJKLMNOPQRSTUVWXYZ

ข้อความรหัส: QEB NRFZH YOLTK CLU GRJMP LSBO QEB IXWV ALD  
ข้อความธรรมดา: THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG



การแปลงสามารถแสดงด้วยการปรับแนวสอง  
พยัญชนะ พยัญชนะรหัสเป็นพยัญชนะธรรมดาที่  
หมุนซ้ายหรือขวาบางตำแหน่ง ตัวอย่างเช่น  
ต่อไปนี้ F เป็นรหัสซีซาร์ที่ใช้การหมุนซ้ายสาม  
ตำแหน่ง เทียบเท่าหมุนขวา 23 ตำแหน่ง



เมื่อเข้ารหัสบุคคลมองหาแต่ละอักษรของสารใน  
บรรทัด "ปกติ" แล้วเขียนอักษรที่ตรงกันใน  
บรรทัด "รหัส" การถอดรหัสให้ทำกลับกัน โดย  
เลื่อนขวา 3



## Encryption in use



การนำหลักการดังกล่าวมาใช้ในชีวิตประจำวัน เช่น การเข้าเว็บไซต์ที่มีการรักษาความปลอดภัย โดยใช้โปรโตคอล HTTPS, การส่งอีเมล, การใช้ Bluetooth

## Data encryption for Arduino IDE



ในการพัฒนาระบบ IoT บน Arduino IDE สามารถนำความรู้ด้านการเข้ารหัสมาใช้ได้ 2 รูปแบบคือ

1. ดาวน์โหลด library ที่มีผู้พัฒนาไว้โดยปกติแล้วจะสามารถใช้งานได้โดยไม่มีค่าใช้จ่าย ปัจจุบันสามารถใช้ AESLib ที่เป็นเทคนิคการเข้ารหัสแบบ AES
2. พัฒนาอัลกอริธึมขึ้นมาใช้งานเองโดยสามารถสร้างเป็น library ของตัวเองหรือจะเป็นแค่การเขียนโปรแกรมลงไปบน Arduino IDE เลย





# AESLib



คือ library สำหรับการเข้ารหัสข้อมูลสำหรับ microcontroller ที่โปรแกรมบน Arduino IDE โดยจะใช้หลักการในการเข้ารหัสแบบ AES

---

## AES Advanced Encryption Standard



แปลว่า มาตรฐานการเข้ารหัสระดับสูง



โดดเด่นในด้านการเข้ารหัสที่รวดเร็วเหมาะสำหรับระบบที่มีการส่งข้อมูลอยู่ตลอดเวลาและนิยมใช้ในการส่งข้อมูลระหว่างอุปกรณ์ฮาร์ดแวร์ต่างๆ รวมไปถึงทางด้านการทหารอีกด้วย

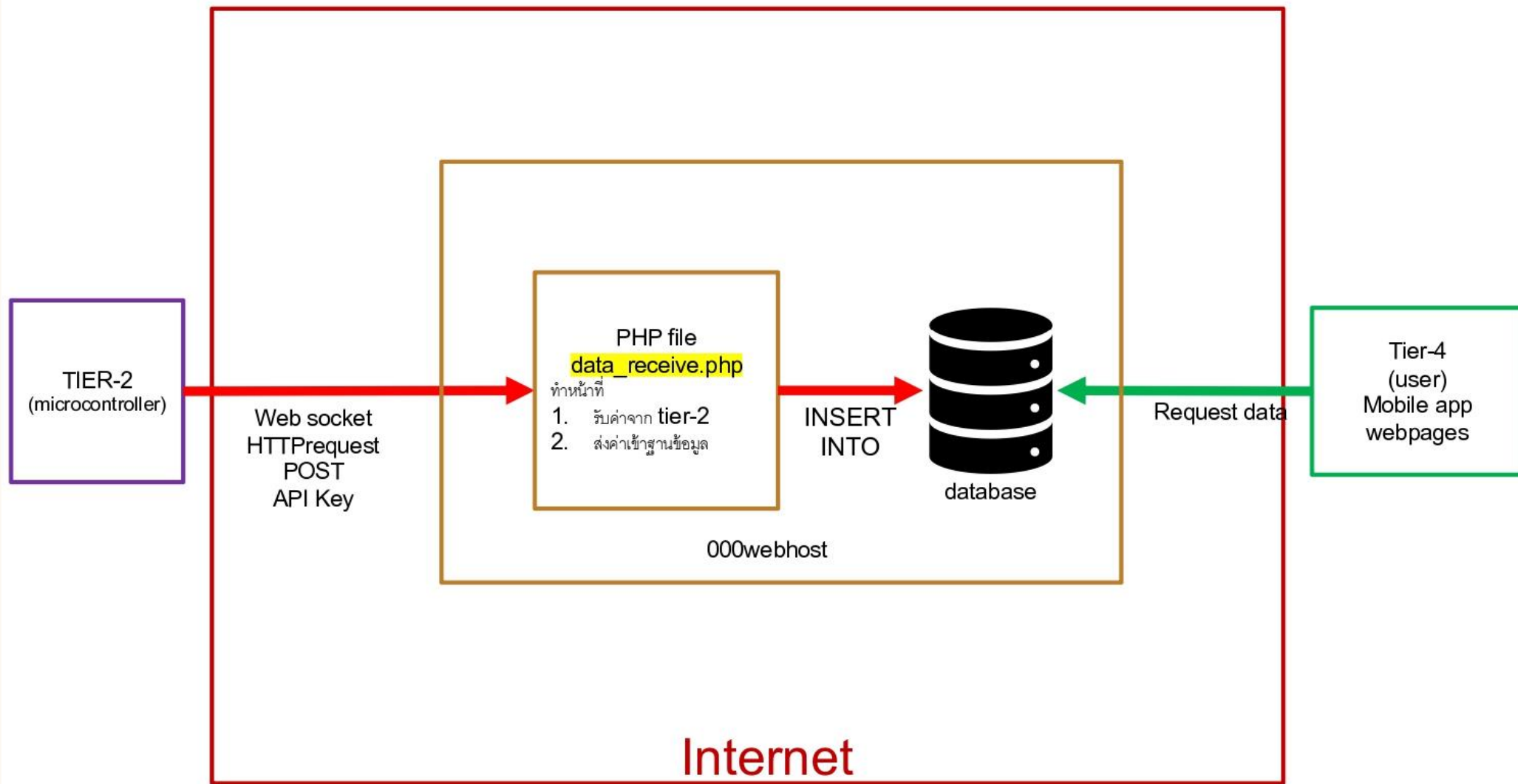


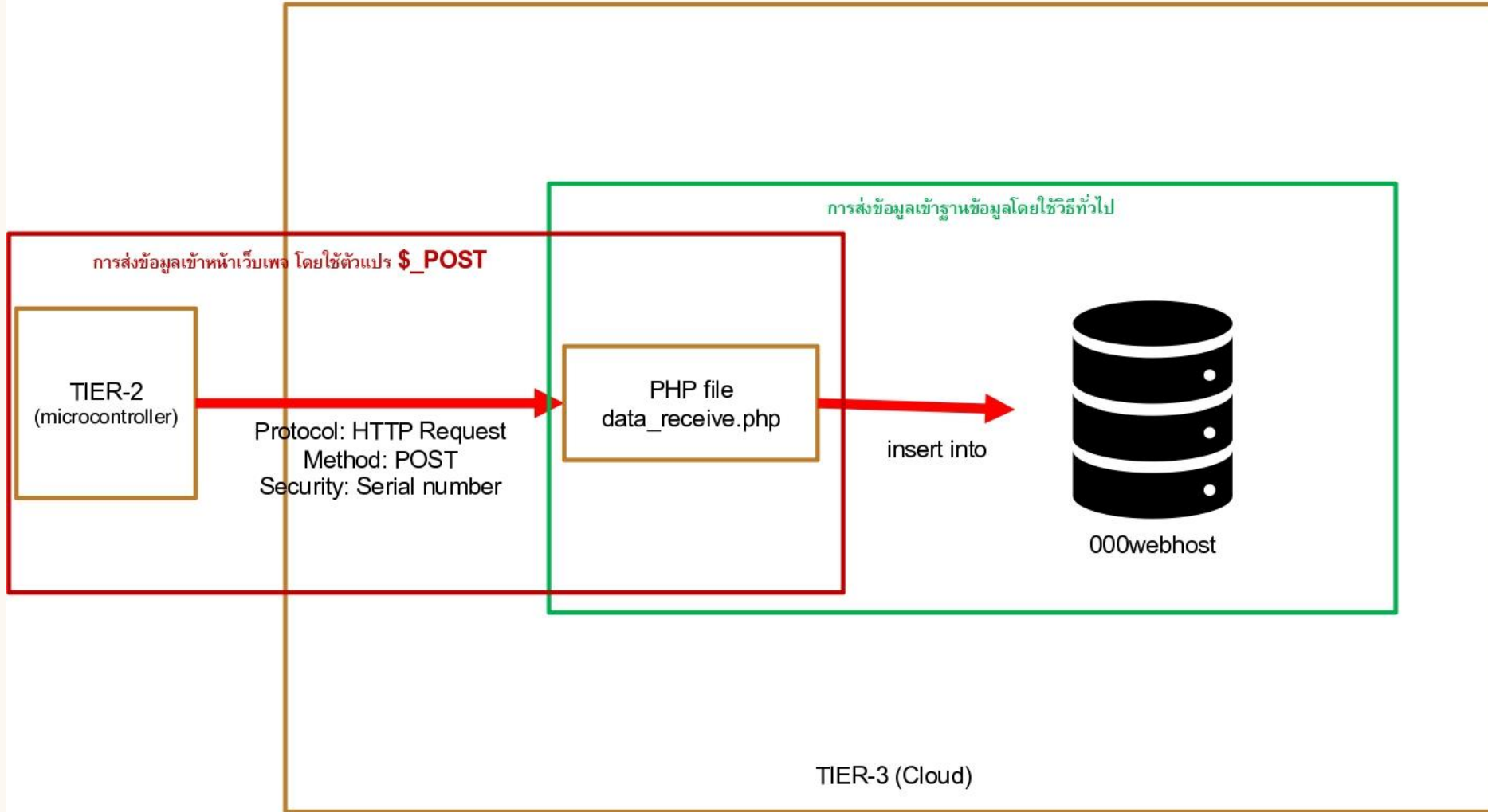
AES เป็นการเข้ารหัสแบบสมมาตร (symmetric) โดยใช้กุญแจตัวเดียวกันในการเข้าและถอดรหัส AES ยังมีการเข้ารหัสข้อมูลหลายรอบ ดังนั้นการที่จะเข้าถึง plaintext แทบเป็นไปไม่ได้



AES ได้รับการรับรองจาก NSA (สำนักงานความมั่นคงแห่งชาติแห่งสหรัฐอเมริกา) ให้ปกป้องข้อมูลลับสุดยอดของประเทศ









## Serial Number



เป็นข้อความที่อาจประกอบไปด้วยตัวเลขหรือตัวอักษรเพียงอย่างเดียวหรือเป็นการผสมกัน โดยการใช้สามารถนำมาใช้ในการยืนยันตนเพื่อเชื่อมต่อกับระบบ เช่น microcontroller ต้องการส่งข้อมูลไปยัง cloud ซึ่ง microcontroller จะต้องแจ้ง serial number แต่ cloud หาก cloud ตรวจสอบแล้วว่า serial number ที่ได้รับแจ้งตรงกันก็สามารถให้ทำการเชื่อมต่อและทำการแลกเปลี่ยนข้อมูลกันได้แต่หากข้อมูลไม่ตรงกันก็จะไม่ให้ทำการเชื่อมต่อโดยตามหลักการแล้วการเชื่อมต่อที่ไม่ได้รับอนุญาตจะไม่ทราบ serial number ที่ใช้ในระบบ ดังนั้นก็จะไม่สามารถเข้าถึงข้อมูลในระบบได้










# Authentication



-  คือการรับรองความถูกต้องหรือการตรวจสอบความถูกต้องการสื่อสารกันระหว่าง tier2 ถึง 4 จำเป็นที่จะต้องมีการระบุตัวตนว่าใช้การเชื่อมต่อ (connection) ที่ได้รับอนุญาตกระบวนการนี้จะเรียกว่า connection authentication
-  connection authentication คือ การตรวจสอบความถูกต้องในการเชื่อมต่อ
-  วิธีการสร้างกลไกการ authenticate คือ การใช้ serial number ที่สร้างขึ้นเองเพื่อส่งข้อมูลจาก tier-2 ไปยังระบบฐานข้อมูลบน tier-3 ที่เป็น cloud service

