บทที่ 4

ผลการทดสอบการทำงานของอัลกอริทึมไอเดส

การวิเคราะห์ประสิทธิภาพของอัลกอริทึมไอเดสเมื่อเปรียบเทียบกับอัลกอริทึมเดส ที่กล่าวไว้ในบทที่ 3 เป็นการวิเคราะห์เชิงทฤษฎี โดยใช้วิธีการทางคณิตศาสตร์ เพื่อหาปริมาณงานที่ต้องทำเพื่อใช้ทำลายอัลกอริทึม ซึ่งเป็นวิธีการวัดความซับซ้อนของอัลกอริทึม จะเห็นว่าอัลกอริทึมไอเดสมีความซับซ้อนมากกว่าอัลกอริทึมเดส

ในบทที่ 4 นี้จะเป็นการทดสอบประสิทธิภาพการทำงานของอัลกอริทึมไอเดสเปรียบเทียบกับอัลกอริทึมเดส หากจะกล่าวโดยเจาะจง ก็คือ ให้พิจารณาประสิทธิภาพการทำงานในส่วนของการแบ่งกลุ่มและการเปิดค่าจากตาราง S-boxes เนื่องจากความแตกต่างระหว่างอัลกอริทึมไอเดสและอัลกอริทึมเดสจะอยู่ที่ส่วนนี้เท่านั้น

ได้มีนักวิชาการ คือ Gordon และ Retkin ได้เสนอว่า S-boxes ที่ดี ควรมีคุณสมบัติดังนี้ คือ (Gordon and Retkin, 1983)

1. เมื่อข้อมูลเข้าเปลี่ยนไป 1 บิต ข้อมูลออก ควรจะมีการเปลี่ยนแปลงที่มากพอ
2. ความสัมพันธ์ของแต่ละบิตของข้อมูลออกควรจะขึ้นต่อแต่ละบิตของข้อมูลเข้า
3. การเข้ารหัสลับข้อมูลเข้าที่ไม่ทราบค่า ควรจะให้ผลลัพธ์เป็นข้อมูลออก ที่ไม่สามารถคาดเดาได้ว่าข้อมูลเข้า คืออะไร
4. ความสัมพันธ์ระหว่างข้อมูลเข้าและข้อมูลออกไม่เป็นความสัมพันธ์แบบเชิงเส้น (Linearity)

และมีความเห็นว่า S-boxes ของอัลกอริทึมเดส มีคุณสมบัติเหล่านี้อย่างครบถ้วน

ดังนั้น ในบทนี้จะศึกษาลงไปในรายละเอียดการทำงานของอัลกอริทึมไอเดส และอัลกอริทึมเดส จะทำการตรวจสอบว่า S-boxes ของอัลกอริทึมไอเดส มีคุณสมบัติดังกล่าวหรือไม่ พร้อมทั้งจะทำการเปรียบเทียบเวลา และหน่วยความจำที่อัลกอริทึมทั้งสองแบบใช้

เนื่องจากอัลกอริทึมไอเดสมีวิธีการแบ่งกลุ่มข้อมูลเพื่อนำมาใช้เปิดตาราง S-boxes ได้จำนวนมากมายหลายวิธี ดังนั้น ในการวิเคราะห์จะเป็นการยกตัวอย่างวิธีการแบ่งกลุ่มเพียง 9 วิธี เพื่อใช้เป็นกรณีศึกษานำมาเปรียบเทียบกับการทำงานของอัลกอริทึมเดส และทดสอบว่ามีคุณสมบัติของ S-boxes ที่ดีหรือไม่ วิธีการแบ่งกลุ่มที่จะยกเป็นกรณีศึกษา มีดังนี้ คือ

กรณีที่ 1 แบ่งข้อมูลเข้า ออกเป็นกลุ่มละ 5 5 5 5 5 5 5 5 8
ได้ผลลัพธ์ออกมาเป็นกลุ่มละ 3 3 3 3 3 3 3 3 8

กรณีที่ 2 แบ่งข้อมูลเข้า ออกเป็นกลุ่มละ 6 6 6 6 12 12
ได้ผลลัพธ์ออกมาเป็นกลุ่มละ 4 4 4 4 8 8

กรณีที่ 3 แบ่งข้อมูลเข้า ออกเป็นกลุ่มละ 7 7 7 7 7 7 6
ได้ผลลัพธ์ออกมาเป็นกลุ่มละ 5 5 5 5 4 4 4

กรณีที่ 4 แบ่งข้อมูลเข้า ออกเป็นกลุ่มละ 8 8 8 8 8 8
ได้ผลลัพธ์ออกมาเป็นกลุ่มละ 6 6 6 6 4 4

กรณีที่ 5 แบ่งข้อมูลเข้า ออกเป็นกลุ่มละ 5 6 10 8 9 10
ได้ผลลัพธ์ออกมาเป็นกลุ่มละ 3 4 7 5 6 7

กรณีที่ 6 แบ่งข้อมูลเข้า ออกเป็นกลุ่มละ 7 10 12 8 11
ได้ผลลัพธ์ออกมาเป็นกลุ่มละ 5 6 8 6 7

กรณีที่ 7 แบ่งข้อมูลเข้า ออกเป็นกลุ่มละ 12 8 6 10 12
ได้ผลลัพธ์ออกมาเป็นกลุ่มละ 8 5 4 7 8

กรณีที่ 8 แบ่งข้อมูลเข้า ออกเป็นกลุ่มละ 12 12 12 12
ได้ผลลัพธ์ออกมาเป็นกลุ่มละ 8 8 8 8

กรณีที่ 9 แบ่งข้อมูลเข้า ออกเป็นกลุ่มละ 7 13 8 12 8
ได้ผลลัพธ์ออกมาเป็นกลุ่มละ 4 9 6 8 5

ในการทดสอบเพื่อดูประสิทธิภาพการทำงานของอัลกอริทึมทั้งสองแบบ จะทำการทดสอบบนเครื่องไมโครคอมพิวเตอร์ รุ่น AT CPU 80286 ความเร็ว 16 MHz และเขียนโปรแกรมด้วยภาษาซี ใช้คอมไพเลอร์ของเทอร์โบ ซี เวอร์ชั่น 2.0 โดยจะมีวิธีการในการทดสอบ ดังนี้

4.1 ความสัมพันธ์ของข้อมูลขาเข้าและข้อมูลขาออก

เนื่องจากผู้ออกแบบอัลกอริทึมเดส มีจุดประสงค์ที่ต้องการให้อัลกอริทึมนี้สามารถแปลงข้อมูลเนื้อแท้ไปเป็นรูปแบบอื่นในลักษณะที่มีความซับซ้อน คือจะไม่สามารถหาความสัมพันธ์อย่างเป็นระบบระหว่างข้อมูลเนื้อแท้กับข้อมูลเข้ารหัสได้ หรือหาความสัมพันธ์ระหว่างคีย์สำหรับเข้ารหัสลับกับข้อมูลเข้ารหัสได้ และการเปลี่ยนแปลงข้อมูลเนื้อแท้ไป 1 บิต หรือเปลี่ยนคีย์สำหรับเข้ารหัสลับไป 1 บิต จะต้องมีผลกระทบทำให้ได้ข้อมูลเข้ารหัสที่มีความแตกต่างกันมากพอ หรือควรจะมีการเปลี่ยนแปลงข้อมูลไปประมาณครึ่งหนึ่งของข้อมูลเข้ารหัสทั้งหมด

การที่ข้อมูลเนื้อแท้ที่ต่างกันเพียงบิตเดียว แต่มีผลทำให้ได้ผลลัพธ์ที่แตกต่างกันมาก แสดงว่า โครงสร้างของอัลกอริทึมมีประสิทธิภาพ หาความสัมพันธ์อย่างเป็นระบบระหว่างข้อมูลเข้ารหัสและข้อมูลเนื้อแท้ไม่ได้ การจะทำลายอัลกอริทึมก็จะทำได้ยาก

ในการวัดความเปลี่ยนแปลงระหว่างข้อมูลเข้ารหัส 2 กลุ่มที่เกิดจากข้อมูลเนื้อแท้ที่ต่างกันเพียงบิตเดียว จะวัดโดยการใช้ ระยะแฮมมิ่ง (Hamming Distance) ซึ่งเป็นวิธีการเปรียบเทียบบิตต่อบิตในตำแหน่งเดียวกันของข้อมูลเข้ารหัสทั้งสองกลุ่ม ถ้าบิตต่างกันจะวัดเป็น 1 ถ้าบิตเหมือนกันจะวัดเป็น 0 ดังนั้น ค่าเฉลี่ยของการวัดความเปลี่ยนแปลงของข้อมูลเข้ารหัส 2 กลุ่ม ซึ่งข้อมูลแต่ละกลุ่มมีขนาด 64 บิต ควรจะมีการเปลี่ยนแปลงไปประมาณเท่ากับครึ่งหนึ่งของข้อมูลเข้ารหัส คือ 32 บิต

เพื่อดูว่าอัลกอริทึมไอเดส ซึ่งมีการปรับเปลี่ยนมาจากอัลกอริทึมเดสจะมีคุณสมบัตินี้หรือไม่ โดยจะทำการทดสอบด้วยการเข้ารหัสข้อมูลเนื้อแท้ที่ต่างกัน 1 บิต ด้วยอัลกอริทึมไอเดส ตามลักษณะการแบ่งกลุ่มที่ยกมาเป็นกรณีศึกษา และในการทดสอบจะแยกเป็น 2 กรณี ดังนี้ คือ ดูการเปลี่ยนแปลงของข้อมูลเข้ารหัสเมื่อข้อมูลเนื้อแท้เปลี่ยนไป 1 บิต และ ดูการเปลี่ยนแปลงของข้อมูลเข้ารหัส เมื่อคีย์สำหรับการเข้ารหัสลับเปลี่ยนไป 1 บิต รายละเอียดดังจะได้กล่าวในตอนต่อไป

4.1.1 การเปลี่ยนแปลงของข้อมูลเข้ารหัสเมื่อข้อมูลเนื้อแท้เปลี่ยนไป 1 บิต

ในการทดสอบได้นำข้อมูล 15 กลุ่ม แต่ละกลุ่มมีขนาด 64 บิต และแต่ละกลุ่มมีความแตกต่างกันเพียงบิตเดียว นำมาเข้ารหัสลับโดยใช้อัลกอริทึมเดส และอัลกอริทึมไอเดส ซึ่งจะมีการใช้คีย์สำหรับการเข้ารหัสลับเดียวกันในการเข้ารหัส ในการทดสอบจะมีการวัดระยะแฮมมิ่ง เพื่อดูการเปลี่ยนแปลงของข้อมูลเข้ารหัสที่เกิดจากข้อมูลเนื้อแท้ที่ต่างกัน 1 บิต ดูตัวอย่างผลการทดสอบในตารางที่ 4.1 เมื่อใช้อัลกอริทึมไอเดส ที่มีการแบ่งกลุ่มเป็นกลุ่มละ 5, 6, 10, 8, 9 และ 10 ได้ผลลัพธ์เป็น 3, 4, 7, 5, 6 และ 7 ตามลำดับ (ผลของการทดสอบวิธีการแบ่งกลุ่มแบบอื่น ๆ ดูในภาคผนวก)

จากตารางที่ 4.1 วิธีการหาค่าระยะแฮมมิ่ง จะทำโดยเข้ารหัสข้อมูลเนื้อแท้ชุดแรก คือ 4142434445464748 ซึ่งเป็นข้อมูลอยู่ในรูปฐานสิบหก และทำการเข้ารหัสข้อมูลเนื้อแท้ชุดอื่น ๆ ที่มีความแตกต่างจากข้อมูลชุดแรกเพียง 1 บิต แล้วนำข้อมูลเข้ารหัสของข้อมูลเนื้อแท้ชุดแรก นำมาเปรียบเทียบกับข้อมูลเข้ารหัสของข้อมูลเนื้อแท้ชุดอื่น ๆ โดยวัดค่าออกมาเป็นระยะแฮมมิ่ง ค่าระยะแฮมมิ่งที่วัดได้ ค่าต่ำสุด คือ 28 และค่าสูงที่สุด คือ 36 คิดเป็น ค่าเฉลี่ยของระยะแฮมมิ่งวัดได้ทั้งหมดเท่ากับ 32.21 ซึ่งใกล้เคียงกับค่าฉลี่ยที่ควรจะเป็น คือ 32

ผลการทดสอบทั้งหมดสามารถสรุปออกมาเป็นตารางที่ 4.2 ซึ่งเป็นการแสดงค่าเฉลี่ยของระยะแฮมมิ่งของข้อมูลเข้ารหัส โดยใช้วิธีการเข้ารหัสโดยผ่านอัลกอริทึมเดส และอัลกอริทึมไอเดส ซึ่งมีวิธีการแบ่งกลุ่มตามกรณีศึกษาที่ได้ยกมา จะเห็นได้ว่า การเข้ารหัสลับโดยผ่านอัลกอริทึมเดส วัดระยะแฮมมิ่งได้ 30.93 ส่วนการเข้ารหัสลับโดยใช้อัลกอริทึมไอเดส ค่าระยะแฮมมิ่งที่วัดได้จะอยู่ในช่วง 29.79 - 32.21 ซึ่งมีค่าใกล้เคียงกับ 32 ทั้งสิ้น

ดังนั้น จากตัวเลขในตารางที่ 4.2 จะเห็นได้ว่า การเข้ารหัสโดยผ่านอัลกอริทึมไอเดส แม้ข้อมูลเนื้อแท้จะมีความแตกต่างกันเพียง 1 บิต แต่ก็ให้ผลลัพธ์เป็นข้อมูลเข้ารหัสที่มีความแตกต่างกันมาก แสดงว่าการทำงานของอัลกอริทึมไอเดสมีการทำงานที่ซับซ้อน ไม่ต่างจากอัลกอริทึมเดส ซึ่งเป็นอัลกอริทึมที่มีโครงสร้างการทำงานที่มีประสิทธิภาพ

ข้อมูลเข้า แบ่งเป็นกลุ่มละ 5 6 10 8 9 10
ข้อมูลออก แบ่งเป็นกลุ่มละ 3 4 7 5 6 7
คีย์สำหรับเข้ารหัสลับ : 0123456789abcdef

ข้อมูลเนื้อแท้	ข้อมูลเข้ารหัส	ระยะแฮมมิ่ง
4142434445464748	07cdae1ba8d16ee2	
5142434445464748	ab05e62a35acf285	32
4152434445464748	25e1d2ddbdd77080	26
4142534445464748	df57796428c13242	29
4142435445464748	434b30e2856ea7a9	35
414243445546474e	64a491baf872a924	32
41424344454e4748	387ca245aad2136f	30
4142434445465748	6d4da6f8d1871886	28
4142434445464758	4c3ac75491cabe62	32
6142434445464748	7051014af393bd33	35
4162434445464748	7fdd5b6ae8088742	28
4142634445464748	982209db69a7d443	36
4142436445464748	f524b0a2df4d20b5	38
4142434465464748	5a06a3f88f5f753a	34
4142434445464768	685734c845d2104a	36
ค่าเฉลี่ยระยะแฮมมิ่ง		32.21

ตารางที่ 4.1 แสดงการเปลี่ยนแปลงของข้อมูลเข้ารหัสเมื่อข้อมูลเนื้อแท้มีความแตกต่างกัน 1 บิต

หมายเหตุ : ข้อมูลเนื้อแท้และข้อมูลเข้ารหัส และคีย์สำหรับเข้ารหัส ในตารางแสดงเป็นตัวเลขอยู่ในรูปฐานสิบหก

กรณี	วิธีการแบ่งกลุ่ม	ค่าเฉลี่ยระยะแฮมมิ่ง
	การเข้ารหัสลับแบบอัลกอริทึมเดส 6 6 6 6 6 6 6 6 4 4 4 4 4 4 4 4	30.93
	การเข้ารหัสลับแบบอัลกอริทึมไอเดส	
1	5 5 5 5 5 5 5 5 8 3 3 3 3 3 3 3 3 8	31.86
2	6 6 6 6 12 12 4 4 4 4 8 8	31.86
3	7 7 7 7 7 7 6 5 5 5 5 4 4 4	31.14
4	8 8 8 8 8 8 6 6 6 6 4 4	31.93
5	5 6 10 8 9 10 3 4 7 5 6 7	32.21
6	7 10 12 8 11 5 6 8 6 7	31.86
7	12 8 6 10 12 8 5 4 7 8	29.79
8	12 12 12 12 8 8 8 8	31.86
9	7 13 8 12 8 4 9 6 8 5	30.64

ตารางที่ 4.2 แสดงระยะแฮมมิ่งของข้อมูลเข้ารหัสเมื่อข้อมูลเนื้อแท้เปลี่ยนไป 1 บิต โดยใช้อัลกอริทึมเดส และอัลกอริทึมไอเดส

4.1.2 การเปลี่ยนแปลงของข้อมูลเข้ารหัส เมื่อคีย์สำหรับการเข้ารหัสลับเปลี่ยนไป 1 บิต

เพื่อดูความเปลี่ยนแปลงของข้อมูลเข้ารหัส เมื่อคีย์สำหรับการเข้ารหัสลับเปลี่ยนไป 1 บิต จะมีการทดสอบในลักษณะเดียวกับการทดสอบในหัวข้อ 4.1.1 แต่ต่างกันที่ข้อมูลที่นำมาเข้ารหัสที่มีความแตกต่างกัน 1 บิตจะเป็นคีย์สำหรับเข้ารหัสลับแทนที่จะเป็นข้อมูลเนื้อแท้ จำนวนคีย์ที่จะนำมาทดสอบมีทั้งหมด 15 คีย์ แต่ละคีย์มีความแตกต่างกันเพียง 1 บิต และจะมีข้อมูลเนื้อแท้ 1 ชุด นำข้อมูลทั้งหมดมาเข้ารหัสผ่านอัลกอริทึมเดสและอัลกอริทึมไอเดส โดยจะมีการวัดระยะแฮมมิ่ง เพื่อดูความเปลี่ยนแปลงของข้อมูลเข้ารหัสที่เกิดจากการเข้ารหัสลับที่ใช้คีย์ที่แตกต่างกันเพียง 1 บิต และใช้ข้อมูลเนื้อแท้ชุดเดียวกัน ดูตัวอย่างการทดสอบจากตารางที่ 4.3 เมื่อใช้อัลกอริทึมไอเดส ที่มีการแบ่งกลุ่มเป็นกลุ่มละ 5, 6, 10, 8, 9 และ 10 ได้ผลลัพธ์ 3, 4, 7, 5, 6 และ 7 ตามลำดับ (ส่วนผลการทดสอบวิธีการแบ่งกลุ่มแบบอื่น ๆ ดูในภาคผนวก)

จากตารางที่ 4.3 วิธีการหาค่าระยะแฮมมิ่ง จะทำโดยเข้ารหัสข้อมูลเนื้อแท้ชุดเดียว คือ 4142434445464748 ซึ่งข้อมูลอยู่ในรูปฐานสิบหก โดยใช้คีย์สำหรับเข้ารหัสลับชุดแรก คือ 0123456789abcdef และทำการเข้ารหัสข้อมูลเนื้อแท้ชุดเดิม แต่ใช้คีย์สำหรับเข้ารหัสลับอื่น ๆ ที่มีความแตกต่างจากคีย์แรกเพียง 1 บิต แล้วนำข้อมูลเข้ารหัสที่เกิดจากคีย์ชุดแรกนำมาเปรียบเทียบกับข้อมูลเข้ารหัสที่เกิดจากการเข้ารหัสด้วยค่าคีย์อื่น ๆ โดยวัดค่าออกมาเป็นระยะแฮมมิ่ง ค่าระยะแฮมมิ่งที่วัดได้ ค่าต่ำสุด คือ 29 และได้ค่าสูงที่สุด คือ 35 และคิดเป็นค่าเฉลี่ยของระยะแฮมมิ่งวัดได้ทั้งหมดเท่ากับ 32.21 ซึ่งใกล้เคียงกับค่าฉลี่ยที่ควรจะเป็น คือ 32

ผลการทดสอบทั้งหมดสามารถสรุปออกมาเป็นตารางที่ 4.4 ซึ่งเป็นการแสดงค่าเฉลี่ยของระยะแฮมมิ่งของข้อมูลเข้ารหัส โดยใช้วิธีการเข้ารหัสโดยผ่านอัลกอริทึมเดส และอัลกอริทึมไอเดส ซึ่งมีวิธีการแบ่งกลุ่มตามตัวอย่างที่ได้ยกมาเป็นกรณีศึกษา จะเห็นได้ว่า การเข้ารหัสลับโดยผ่านอัลกอริทึมเดส วัดระยะแฮมมิ่งได้ 33.21 ส่วนการเข้ารหัสลับโดยใช้อัลกอริทึมไอเดส ค่าระยะแฮมมิ่งที่วัดได้จะอยู่ในช่วง 29.64 - 34.14 ซึ่งมีค่าใกล้เคียงกับ 32

ดังนั้น จากตัวเลขในตารางที่ 4.4 จะเห็นได้ว่า การเข้ารหัสโดยผ่านอัลกอริทึมไอเดส แม้ว่าจะใช้ข้อมูลเนื้อแท้ชุดเดียวกัน และมีการใช้คีย์สำหรับเข้ารหัสที่มีความแตกต่างกันเพียง 1 บิต แต่ก็ให้ผลลัพธ์เป็นข้อมูลเข้ารหัสที่มีความแตกต่างกันมาก แสดงว่าการทำงานของอัลกอริทึมไอเดสมีการทำงานที่ซับซ้อน ไม่แตกต่างจากอัลกอริทึมเดส

สรุปการทดลองในหัวข้อ 4.1.1 และ 4.1.2 จากการวัดระยะแฮมมิ่งในตาราง ที่ 4.2 และตารางที่ 4.4 แสดงว่า S-boxes ของอัลกอริทึมไอเดสมีคุณสมบัติตามข้อ 1 ที่ได้กล่าวมาข้างต้น นั่นหมายถึงว่า การทำงานภายในของอัลกอริทึมไอเดส มีการทำงานที่มีประสิทธิภาพไม่แตกต่างอัลกอริทึมเดส ซึ่งเป็นอัลกอริทึมที่ได้รับการยอมรับทั่วไป เพราะการเปลี่ยนแปลงข้อมูลเพียงบิตเดียว ได้ส่งผลกระทบทำให้ข้อมูลเข้ารหัสมีการเปลี่ยนแปลงไปจากเดิมมาก ทำให้ไม่สามารถที่จะหาความสัมพันธ์ระหว่างข้อมูลเหล่านี้อย่างเป็นระบบได้ ความพยายามที่จะทำลายอัลกอริทึมก็จะเป็นไปได้ยาก

ข้อมูลเข้า แบ่งเป็นกลุ่มละ 5 6 10 8 9 10
ข้อมูลออก แบ่งเป็นกลุ่มละ 3 4 7 5 6 7
ข้อมูลเนื้อแท้ : 4142434445464748

คีย์สำหรับเข้ารหัสลับ	ข้อมูลเข้ารหัส	ระยะแฮมมิ่ง
0123456789abcdef	07cdae1ba8d16ee2	
1123456789abcdef	9d8ea0366c0c82e8	30
0162456789abcdef	294d38a1ccab9507	34
0123656789abcdef	dfe73e5416c6fd19	35
0123454789abcdef	83b832dc0d2cda46	34
0123456709abcdef	251abf315a4155d7	29
01234567898bcdef	26d592ca3d8ffc9e	29
0123456789ab8def	6cee3531f6d7f18e	33
0123456789abcd6f	8fcc75476268623b	29
0323456789abcdef	24efd3070b178a06	30
0127456789abcdef	563674b7d70b87c0	38
01234d6789abcdef	bc00a88db293c90c	33
0123456589abcdef	44874870b85b4459	29
0123456781abcdef	3e9f24bfdb63354c	32
0123456789afcdef	0bdf0635427b1419	32
ค่าเฉลี่ยระยะแฮมมิ่ง		31.93

ตารางที่ 4.3 แสดงการเปลี่ยนแปลงของข้อมูลเข้ารหัสเมื่อคีย์สำหรับเข้ารหัสลับมีความแตกต่างกัน 1 บิต

หมายเหตุ : ข้อมูลเนื้อแท้และข้อมูลเข้ารหัส และคีย์สำหรับเข้ารหัส ในตารางแสดงเป็นตัวเลขอยู่ในรูปฐานสิบหก

กรณี	วิธีการแบ่งกลุ่ม	ค่าเฉลี่ยระยะแฮมมิ่ง
	การเข้ารหัสลับแบบอัลกอริทึมเดส 6 6 6 6 6 6 6 6 4 4 4 4 4 4 4 4	33.21
	การเข้ารหัสลับแบบอัลกอริทึมไอเดส	
1	5 5 5 5 5 5 5 5 8 3 3 3 3 3 3 3 3 8	33.07
2	6 6 6 6 12 12 4 4 4 4 8 8	33.43
3	7 7 7 7 7 7 6 5 5 5 5 4 4 4	32.29
4	8 8 8 8 8 8 6 6 6 6 4 4	32.00
5	5 6 10 8 9 10 3 4 7 5 6 7	31.93
6	7 10 12 8 11 5 6 8 6 7	31.29
7	12 8 6 10 12 8 5 4 7 8	29.64
8	12 12 12 12 8 8 8 8	34.14
9	7 13 8 12 8 4 9 6 8 5	30.14

ตารางที่ 4.4 แสดงระยะแฮมมิ่งของข้อมูลเข้ารหัสเมื่อคีย์สำหรับเข้ารหัสลับเปลี่ยนไป 1 บิต โดยใช้อัลกอริทึมเดส และอัลกอริทึมไอเดส

4.2 การวิเคราะห์ความสัมพันธ์ของข้อมูลเข้ารหัสที่ขึ้นต่อข้อมูลเนื้อแท้

ในการออกแบบอัลกอริทึมที่ดี แต่ละบิตของข้อมูลออกจะต้องมีความสัมพันธ์กับบิตของข้อมูลเข้า ซึ่งความสัมพันธ์นี้จะเพิ่มขึ้นเรื่อย ๆ ตามจำนวนรอบของการทำงาน ซึ่งอัลกอริทึมเดสก็มีคุณสมบัติข้อนี้ กล่าวคือ แต่ละบิตของข้อมูลเข้ารหัสจะเป็นฟังก์ชั่นที่ซับซ้อนกับทุก ๆ บิตของข้อมูลเนื้อแท้ และกับทุก ๆ บิตของคีย์สำหรับเข้ารหัส ซึ่งจะมีการหาวิธีวิเคราะห์เพื่อวัดความสัมพันธ์ว่าเมื่อใดข้อมูลเข้ารหัสจึงจะเป็นฟังก์ชั่นกับทุก ๆ บิตของข้อมูลเนื้อแท้ ซึ่งจะวัดเป็นจำนวนรอบของการทำงานที่ซ้ำ ๆ กัน ดูรายละเอียดของการวิเคราะห์จากหัวข้อ Analysis of Intersymbol Dependencies for Data Encryption Standard ใน (Meyer and Matyas, 1982)

เนื่องจากความซับซ้อนของอัลกอริทึมเดส อยู่ที่การทำงานที่ซับซ้อนของฟังก์ชั่น f ซึ่งถูกกระทำซ้ำ ๆ กัน 16 รอบ โดยที่ในแต่ละรอบจะมีสับคีย์และข้อมูลเข้าสำหรับรอบที่ i คือ X(i-1) โดยที่ i = 1, 2, 3, ..., 16 ค่า X(i-1) จะถูกแบ่งเป็น 2 ส่วน ๆ ละ 32 บิต เป็นส่วนซ้าย L(i-1) และส่วนขวา R(i-1) นั่นคือ

$$X(i-1) = L(i-1), R(i-1)$$

และข้อมูลออกในรอบที่ i คือ X(i) และ

$$X(i) = L(i), R(i)$$

จากการทำงานในลักษณะที่มีการแบ่งข้อมูลออกเป็น 2 ส่วน เป็นส่วนซ้ายและส่วนขวา ทำให้เราสามารถวิเคราะห์หาความสัมพันธ์ของข้อมูลออก ในรอบที่ i เมื่อ i = 1, 2, 3, .., 16 กับข้อมูลเข้า ในรอบที่ 0 โดยการพิจารณาแยกข้อมูลออกเป็นส่วน ๆ ดังนี้ คือ

วิเคราะห์ความสัมพันธ์ระหว่างข้อมูลออกส่วนซ้าย L(i) ในรอบที่ i
กับข้อมูลเข้าส่วนซ้าย L(0) ในรอบที่ 0

วิเคราะห์ความสัมพันธ์ระหว่างข้อมูลออกส่วนซ้าย L(i) ในรอบที่ i
กับข้อมูลเข้าส่วนขวา R(0) ในรอบที่ 0

วิเคราะห์ความสัมพันธ์ระหว่างข้อมูลออกส่วนขวา R(i) ในรอบที่ i
กับข้อมูลเข้าส่วนซ้าย L(0) ในรอบที่ 0

วิเคราะห์ความสัมพันธ์ระหว่างข้อมูลออกส่วนขวา R(i) ในรอบที่ i
กับข้อมูลเข้าส่วนขวา R(0) ในรอบที่ 0

และจากการวิเคราะห์อัลกอริทึมเดส พบว่าการทำงานซ้ำ ๆ กันเพียง 5 รอบก็เพียงพอที่จะทำให้ข้อมูลเข้ารหัสมีความสัมพันธ์กับทุก ๆ บิตของข้อมูลเนื้อแท้ ดังแสดงไว้ในตารางที่ 4.5

รอบที่	ความสัมพันธ์ระหว่าง ข้อมูลออก / ข้อมูลเข้า (ร้อยละ)				
(i)	L(i) & L(0)	L(i) & R(0)	R(i) & L(0)	R(i) & R(0)	X(i) & X(0)
1	0.00	3.13	3.13	18.75	6.25
2	3.13	18.75	18.75	87.60	32.06
3	18.75	87.60	87.60	100.00	73.49
4	87.60	100.00	100.00	100.00	96.90
5	100.00	100.00	100.00	100.00	100.00

ตารางที่ 4.5 แสดงความสัมพันธ์ระหว่างข้อมูลออกกับข้อมูลเข้า ตามจำนวนรอบของการทำงานโดยใช้อัลกอริทึมเดส

จากตารางที่ 4.5 ในรอบที่ 1 แสดงความสัมพันธ์ระหว่างบิตของข้อมูล L(1) ที่ขึ้นกับ L(0) จะเป็น 0 คือ ไม่มีความสัมพันธ์กันเลยระหว่างส่วนซ้ายของข้อมูลออก กับส่วนซ้ายของข้อมูลเข้า ส่วนความสัมพันธ์ระหว่างบิตของข้อมูล L(1) ที่ขึ้นกับ R(0) จะเป็นร้อยละ 13.3 นั่นคือ ข้อมูลออกส่วนซ้ายมีความสัมพันธ์ขึ้นต่อข้อมูลเข้าส่วนขวา เป็นร้อยละ 13.3 และความสัมพันธ์ระหว่างบิตของข้อมูล R(1) ที่ขึ้นกับ L(0) จะเป็นร้อยละ 13.3 เช่นเดียวกัน

และความสัมพันธ์ระหว่างบิตของข้อมูลออก R(1) ที่ขึ้นกับ R(0) จะเป็นร้อยละ 18.75 ซึ่งเมื่อคิดเฉลี่ยแล้ว ความสัมพันธ์ของข้อมูลออกที่ขึ้นต่อข้อมูลเข้า ในรอบที่ 1 จะเท่ากับร้อยละ 6.25 และความสัมพันธ์นี้จะเพิ่มขึ้นเรื่อย ๆ ในรอบที่ 2, 3 และ 4 เป็นร้อยละ 32.06, 73.49 และ 96.90 ตามลำดับ จนกระทั่งในรอบที่ 5 ความสัมพันธ์ของข้อมูลออกที่ขึ้นต่อข้อมูลเข้าจะเท่ากับร้อยละ 100 ซึ่งจากผลการวิเคราะห์ที่ได้ทำให้สรุปได้ว่า การทำงานในอัลกอริทึมเดส เพียง 5 รอบ ก็สามารถทำให้แต่ละบิตของข้อมูลออกขึ้นกับทุก ๆ บิตของข้อมูลเข้าได้

ดังนั้นเพื่อวิเคราะห์การทำงานของอัลกอริทึมไอเดส โดยการวิเคราะห์ความสัมพันธ์ของข้อมูลเข้ารหัสกับข้อมูลเนื้อแท้ว่าเป็นอย่างไร โดยจะทำการทดสอบในลักษณะเดียวกับของอัลกอริทึมเดส ซึ่งการทดสอบจะใช้กับวิธีการแบ่งกลุ่มที่ยกมาเป็นกรณีศึกษา 9 ตัวอย่างที่กล่าวมาข้างต้น และผลการทดลองได้สรุปไว้ในตารางที่ 4.6 ถึงตารางที่ 4.14 ดังนี้ คือ

กรณีที่ 1 : ข้อมูลเข้า แบ่งเป็นกลุ่มละ 5 5 5 5 5 5 5 5 8
ข้อมูลออก แบ่งเป็นกลุ่มละ 3 3 3 3 3 3 3 3 8

รอบที่	ความสัมพันธ์ระหว่าง ข้อมูลออก / ข้อมูลเข้า (ร้อยละ)				
(i)	L(i) & L(0)	L(i) & R(0)	R(i) & L(0)	R(i) & R(0)	X(i) & X(0)
1	0.00	3.13	3.13	14.36	5.15
2	3.13	14.36	14.36	57.13	22.24
3	14.36	57.13	57.13	100.00	57.15
4	57.13	100.00	100.00	100.00	89.28
5	100.00	100.00	100.00	100.00	100.00

ตารางที่ 4.6 แสดงความสัมพันธ์ระหว่างข้อมูลออกกับข้อมูลเข้า ตามจำนวนรอบของการทำงานโดยใช้อัลกอริทึมไอเดส แบ่งกลุ่มแบบกรณีที่ 1

กรณีที่ 2 : ข้อมูลเข้า แบ่งเป็นกลุ่มละ 6 6 6 6 12 12
ข้อมูลออก แบ่งเป็นกลุ่มละ 4 4 4 4 8 8

รอบที่	ความสัมพันธ์ระหว่าง ข้อมูลออก / ข้อมูลเข้า (ร้อยละ)				
(i)	L(i) & L(0)	L(i) & R(0)	R(i) & L(0)	R(i) & R(0)	X(i) & X(0)
1	0.00	3.13	3.13	25.00	7.82
2	3.13	25.00	25.00	95.31	37.11
3	25.00	95.31	95.31	100.00	78.91
4	95.31	100.00	100.00	100.00	98.83
5	100.00	100.00	100.00	100.00	100.00

ตารางที่ 4.7 แสดงความสัมพันธ์ระหว่างข้อมูลออกกับข้อมูลเข้า ตามจำนวนรอบของการทำงาน โดยใช้อัลกอริทึมไอเดส แบ่งกลุ่มแบบกรณีที่ 2

กรณีที่ 3 : ข้อมูลเข้า แบ่งเป็นกลุ่มละ 7 7 7 7 7 7 6
ข้อมูลออก แบ่งเป็นกลุ่มละ 5 5 5 5 4 4 4

รอบที่	ความสัมพันธ์ระหว่าง ข้อมูลออก / ข้อมูลเข้า (ร้อยละ)				
(i)	L(i) & L(0)	L(i) & R(0)	R(i) & L(0)	R(i) & R(0)	X(i) & X(0)
1	0.00	3.13	3.13	16.89	5.79
2	3.13	16.89	16.89	77.34	28.57
3	16.89	77.34	77.34	100.00	67.90
4	77.34	100.00	100.00	100.00	94.34
5	100.00	100.00	100.00	100.00	100.00

ตารางที่ 4.8 แสดงความสัมพันธ์ระหว่างข้อมูลออกกับข้อมูลเข้า ตามจำนวนรอบของการทำงาน โดยใช้อัลกอริทึมไอเดส แบ่งกลุ่มแบบกรณีที่ 3

กรณีที่ 4 : ข้อมูลเข้า แบ่งเป็นกลุ่มละ 8 8 8 8 8 8
ข้อมูลออก แบ่งเป็นกลุ่มละ 6 6 6 6 4 4

รอบที่	ความสัมพันธ์ระหว่าง ข้อมูลออก / ข้อมูลเข้า (ร้อยละ)				
(i)	L(i) & L(O)	L(i) & R(O)	R(i) & L(O)	R(i) & R(O)	X(i) & X(O)
1	0.00	3.13	3.13	18.75	6.25
2	3.13	18.75	18.75	88.18	32.20
3	18.75	88.18	88.18	100.00	73.78
4	88.18	100.00	100.00	100.00	97.05
5	100.00	100.00	100.00	100.00	100.00

ตารางที่ 4.9 แสดงความสัมพันธ์ระหว่างข้อมูลออกกับข้อมูลเข้า ตามจำนวนรอบของการทำงาน โดยใช้อัลกอริทึมไอเดส แบ่งกลุ่มแบบกรณีที่ 4

กรณีที่ 5 : ข้อมูลเข้า แบ่งเป็นกลุ่มละ 5 6 10 8 9 10
ข้อมูลออก แบ่งเป็นกลุ่มละ 3 4 7 5 6 7

รอบที่	ความสัมพันธ์ระหว่าง ข้อมูลออก / ข้อมูลเข้า (ร้อยละ)				
(i)	L(i) & L(O)	L(i) & R(O)	R(i) & L(O)	R(i) & R(O)	X(i) & X(O)
1	0.00	3.13	3.13	20.12	6.59
2	3.13	20.12	20.12	77.05	30.10
3	20.12	77.05	77.05	100.00	68.55
4	77.05	100.00	100.00	100.00	94.26
5	100.00	100.00	100.00	100.00	100.00

ตารางที่ 4.10 แสดงความสัมพันธ์ระหว่างข้อมูลออกกับข้อมูลเข้า ตามจำนวนรอบของการทำงาน โดยใช้อัลกอริทึมไอเดส แบ่งกลุ่มแบบกรณีที่ 5

กรณีที่ 6 : ข้อมูลเข้า แบ่งเป็นกลุ่มละ 7 10 12 8 11
ข้อมูลออก แบ่งเป็นกลุ่มละ 5 6 8 6 7

รอบที่	ความสัมพันธ์ระหว่าง ข้อมูลออก / ข้อมูลเข้า (ร้อยละ)				
(i)	L(i) & L(0)	L(i) & R(0)	R(i) & L(0)	R(i) & R(0)	X(i) & X(0)
1	0.00	3.13	3.13	24.32	7.64
2	3.13	24.32	24.32	92.68	36.11
3	24.32	92.68	92.68	100.00	77.42
4	92.68	100.00	100.00	100.00	98.17
5	100.00	100.00	100.00	100.00	100.00

ตารางที่ 4.11 แสดงความสัมพันธ์ระหว่างข้อมูลออกกับข้อมูลเข้า ตามจำนวนรอบของการทำงาน โดยใช้อัลกอริทึมไอเดส แบ่งกลุ่มแบบกรณีที่ 6

กรณีที่ 7 : ข้อมูลเข้า แบ่งเป็นกลุ่มละ 12 8 6 10 12
ข้อมูลออก แบ่งเป็นกลุ่มละ 8 5 4 7 8

รอบที่	ความสัมพันธ์ระหว่าง ข้อมูลออก / ข้อมูลเข้า (ร้อยละ)				
(i)	L(i) & L(0)	L(i) & R(0)	R(i) & L(0)	R(i) & R(0)	X(i) & X(0)
1	0.00	3.13	3.13	24.80	7.77
2	3.13	24.80	24.80	96.00	37.18
3	24.80	96.00	96.00	100.00	79.20
4	96.00	100.00	100.00	100.00	99.00
5	100.00	100.00	100.00	100.00	100.00

ตารางที่ 4.12 แสดงความสัมพันธ์ระหว่างข้อมูลออกกับข้อมูลเข้า ตามจำนวนรอบของการทำงาน โดยใช้อัลกอริทึมไอเดส แบ่งกลุ่มแบบกรณีที่ 7

กรณีที่ 8 : ข้อมูลเข้า แบ่งเป็นกลุ่มละ 12 12 12 12
ข้อมูลออก แบ่งเป็นกลุ่มละ 8 8 8 8

รอบที่	ความสัมพันธ์ระหว่าง ข้อมูลออก / ข้อมูลเข้า (ร้อยละ)				
(i)	L(i) & L(0)	L(i) & R(0)	R(i) & L(0)	R(i) & R(0)	X(i) & X(0)
1	0.00	3.13	3.13	31.32	9.38
2	3.13	31.25	31.25	100.00	41.41
3	31.25	100.00	100.00	100.00	82.81
4	100.00	100.00	100.00	100.00	100.00

ตารางที่ 4.13 แสดงความสัมพันธ์ระหว่างข้อมูลออกกับข้อมูลเข้า ตามจำนวนรอบของการทำงาน โดยใช้อัลกอริทึมไอเดส แบ่งกลุ่มแบบกรณีที่ 8

กรณีที่ 9 : ข้อมูลเข้า แบ่งเป็นกลุ่มละ 7 13 8 12 8
ข้อมูลออก แบ่งเป็นกลุ่มละ 4 9 6 8 5

รอบที่	ความสัมพันธ์ระหว่าง ข้อมูลออก / ข้อมูลเข้า (ร้อยละ)				
(i)	L(i) & L(0)	L(i) & R(0)	R(i) & L(0)	R(i) & R(0)	X(i) & X(0)
1	0.00	3.13	3.13	22.95	7.30
2	3.13	22.95	22.95	96.88	36.48
3	22.95	96.88	96.88	100.00	79.17
4	96.88	100.00	100.00	100.00	99.22
5	100.00	100.00	100.00	100.00	100.00

ตารางที่ 4.14 แสดงความสัมพันธ์ระหว่างข้อมูลออกกับข้อมูลเข้า ตามจำนวนรอบของการทำงาน โดยใช้อัลกอริทึมไอเดส แบ่งกลุ่มแบบกรณีที่ 9

จากการหาความสัมพันธ์ของข้อมูลโดยการทดสอบกับวิธีการแบ่งกลุ่มทั้ง 9 กรณี พบว่าการทำงานซ้ำ ๆ กันเพียง 5 รอบ ก็ทำให้ความสัมพันธ์ของข้อมูลเข้ารหัสขึ้นกับข้อมูลเนื้อแท้เต็มร้อยละ 100 ซึ่งหมายความว่า ข้อมูลเข้ารหัสมีความสัมพันธ์กับข้อมูลเนื้อแท้ทุก ๆ บิตแล้ว ยกเว้นการเข้ารหัสแบบกรณีที่ 8 ที่พบว่าทำงานเพียง 4 รอบ ก็ทำให้ความสัมพันธ์ของข้อมูลเข้ารหัสขึ้นกับข้อมูลเนื้อแท้ทุก ๆ บิตแล้ว และเมื่อเปรียบเทียบความสัมพันธ์ในแต่ละรอบของการเข้ารหัสแบบต่าง ๆ ดูตารางที่ 4.15 จะพบว่า ในรอบที่ 1 การเข้ารหัสที่ใช้อัลกอริทึมไอเดสแบบที่ 1 และ 3 มีความสัมพันธ์น้อยกว่าการเข้ารหัสที่ใช้อัลกอริทึมเดส ส่วนการเข้ารหัสโดยใช้อัลกอริทึมไอเดสแบบอื่น ๆ จะมีความสัมพันธ์มากกว่า ส่วนในรอบที่ 2, 3 และ 4 การเข้ารหัสที่ใช้อัลกอริทึมไอเดสแบบที่ 1, 3 และ 5 จะมีความสัมพันธ์น้อยกว่าการเข้ารหัสที่ใช้อัลกอริทึมเดส ส่วนแบบอื่น ๆ อีก 6 แบบ จะมีความสัมพันธ์มากกว่าที่ใช้เดส โดยเฉพาะในแบบที่ 8 ในรอบที่ 4 ความสัมพันธ์ของข้อมูลก็เต็มร้อยละ 100 แล้ว

รอบ	อัลกอริ-ทึมเดส	อัลกอริทึมไอเดส								
		1	2	3	4	5	6	7	8	9
1	6.25	5.15	7.8	5.79	6.25	6.59	7.64	7.77	9.38	7.30
2	32.06	22.24	37.11	28.57	32.20	30.10	36.11	37.18	41.41	36.48
3	73.49	57.15	78.91	67.90	73.78	68.55	77.42	79.20	82.81	79.17
4	96.90	89.28	98.83	94.34	97.05	94.26	98.17	99.00	100.00	99.22
5	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00

ตารางที่ 4.15 แสดงความสัมพันธ์ระหว่างข้อมูลออกกับข้อมูลเข้า ตามจำนวนรอบของการทำงาน โดยเปรียบเทียบระหว่างการใช้อัลกอริทึมเดสกับอัลกอริทึมไอเดส

สรุปผลจากการทดสอบเพื่อหาความสัมพันธ์ของข้อมูลเข้ารหัสที่ขึ้นต่อข้อมูลเนื้อแท้ โดยการวิเคราะห์หาจำนวนรอบในการทำงานที่ทำให้ข้อมูลเข้ารหัสขึ้นต่อข้อมูลเนื้อแท้ ผลการทดสอบได้แสดงให้เห็นว่าจำนวนรอบในการทำงานของอัลกอริทึมเดสกับอัลกอริทึมไอเดส ที่ทำให้ข้อมูลเข้ารหัสมีความสัมพันธ์กับข้อมูลเนื้อแท้ทุก ๆ บิต จะใช้จำนวนรอบที่เท่ากันหรือใกล้เคียงกัน นั่นคือ

สามารถกล่าวได้ว่า S-boxes ของอัลกอริทึมไอเดสมีคุณสมบัติที่ดี ที่ S-boxes ควรจะมีตามข้อ 2 และได้แสดงว่า อัลกอริทึมไอเดส ซึ่งมีการปรับเปลี่ยนมาจากอัลกอริทึมเดสเป็นอัลกอริทึมที่มีประสิทธิภาพในการทำงานไม่แตกต่างจากอัลกอริทึมเดส

จากการทดสอบและเปรียบเทียบประสิทธิภาพการทำงานของอัลกอริทึมไอเดสกับอัลกอริทึมเดส ได้แสดงว่าการแบ่งกลุ่มเพื่อเปิดค่าจากตารางใน S-boxes ของอัลกอริทึมไอเดสมีคุณสมบัติของ S-boxes ที่ดีตามข้อ 1 และ 2 นั่นคือ การเปลี่ยนแปลงของข้อมูลออก เมื่อข้อมูลเข้าเปลี่ยนไป 1 บิต มีการเปลี่ยนแปลงมากพอ และความสัมพันธ์ของข้อมูลออกทุก ๆ บิต จะขึ้นกับแต่ละบิตของข้อมูลเข้าทุก ๆ บิต เมื่อทำงานในฟังก์ชั่น f เพียง 5 รอบ เท่านั้น และจากการทดสอบนี้ทำให้เราสามารถสรุปได้ว่า S-boxes นี้มีคุณสมบัติตามข้อที่ 3 ด้วย คือ การเข้ารหัสข้อมูลเข้าที่ไม่ทราบค่า ควรจะให้ข้อมูลออก ที่ไม่สามารถคาดเดาได้ว่าข้อมูลเข้า คืออะไร เพราะในการทดสอบข้อมูลเข้าที่แตกต่างกัน 1 บิต จะให้ข้อมูลออกที่มีค่าแตกต่างกันมาก ทำให้ไม่สามารถคาดเดาได้ว่าข้อมูลเข้าคืออะไร

ส่วนคุณสมบัติที่ควรมีของ S-boxes อีกข้อหนึ่ง คือ ความสัมพันธ์ของข้อมูลเข้าและข้อมูลออกต้องไม่เป็นความสัมพันธ์แบบเชิงเส้น Gordon และ Retkin ได้ศึกษารายละเอียดและเสนอเป็นบทความชื่อ "Are Big S-boxes Best ?" (Gordon and Retkin, 1983) ซึ่งได้นำมาสรุปไว้ในหัวข้อต่อไป

4.3 ความสัมพันธ์ของข้อมูลเข้าและข้อมูลออกต้องไม่เป็นความสัมพันธ์แบบเชิงเส้น

Gordon และ Retkin มีความเห็นว่า ในการเข้ารหัสที่ใช้เทคนิคการแทนที่ข้อมูลด้วยค่าในตาราง มีความเป็นไปได้ที่จะก่อให้เกิดความสัมพันธ์ที่เป็นแบบเชิงเส้นระหว่างข้อมูลเข้าและข้อมูลออก ซึ่งความสัมพันธ์แบบเชิงเส้นเป็นความสัมพันธ์ที่ไม่พึงประสงค์จะให้เกิดขึ้นในการเข้ารหัสลับ ดังนั้น ทำให้เกิดคำถามว่า ควรมีการแบ่งกลุ่มข้อมูลอย่างไร ถ้าเลือกค่าจากตารางที่มีค่าเป็นเลขสุ่ม จึงจะไม่ทำให้เกิดความสัมพันธ์แบบเชิงเส้นเกิดขึ้น

จากบทความที่เขาเสนอ ได้แสดงให้เห็นว่า ความน่าจะเป็นของความสัมพันธ์แบบเชิงเส้นจะลดลงมาก เมื่อขนาดของกลุ่มที่ใช้เปิดค่าจากตารางใหญ่ขึ้น ในอัลกอริทึมเดส ข้อมูลเข้ามีขนาด 6 บิต ได้ข้อมูลออกขนาด 4 บิต คือ (6, 4) S-boxes ซึ่งมีทั้งหมด 8 กลุ่ม ในการศึกษาเขาจะพิจารณาแต่ละกลุ่มในลักษณะ(4,4) S-boxes เป็นการแทนที่ข้อมูลเข้า 4 บิต ด้วยข้อมูลออก 4 บิต ส่วนอีก 2 บิต จะเป็นตัวกำหนดว่าจะเลือกตาราง S-box ใด ใน 4 S-boxes ซึ่งการพิจารณาแบบนี้ วิธีการทำงานยังเหมือนเดิม แต่มีมุมมองที่แตกต่างไปจาก

เดิมเท่านั้น และการหาความน่าจะเป็นที่ความสัมพันธ์ระหว่างข้อมูลออก 1 บิต หรือหลาย ๆ บิต กับข้อมูลเข้า มีความสัมพันธ์เป็นแบบเชิงเส้น หรือหมายถึงความสัมพันธ์ระหว่างตัวเลขสุ่มในตาราง S-boxes กับข้อมูลเข้า มีความสัมพันธ์เป็นแบบเชิงเส้น จะเท่ากับ 9.3×10^{-3} และความน่าจะเป็นนี้จะลดลงอย่างรวดเร็ว เมื่อขนาดของกลุ่มข้อมูลใหญ่ขึ้น และประมาณการได้ดังนี้ คือ

ข้อมูลขนาด 5 บิต มีความน่าจะเป็น เท่ากับ 5.1×10^{-7}
ข้อมูลขนาด 6 บิต มีความน่าจะเป็น เท่ากับ 4.1×10^{-16}
ข้อมูลขนาด 7 บิต มีความน่าจะเป็น เท่ากับ 7.5×10^{-35}
ข้อมูลขนาด 8 บิต มีความน่าจะเป็น เท่ากับ 7.1×10^{-73}

ดังนั้น การเปรียบเทียบประสิทธิภาพการทำงานของอัลกอริทึมไอเดสว่าดีกว่าอัลกอริทึมเดส หรือไม่ เมื่อพิจารณาจากคุณสมบัติข้อ 4 นี้ จะขึ้นอยู่กับการแบ่งกลุ่มเพื่อใช้เปิดค่าจากตาราง S-boxes ถ้าแบ่งกลุ่มข้อมูลใหญ่กว่า 4 บิต ความน่าจะเป็นที่ความสัมพันธ์ของข้อมูลเข้ากับข้อมูลออกจะเป็นเชิงเส้น ก็จะน้อยลงมาก แต่ในทางตรงกันข้าม ความน่าจะเป็นจะเพิ่มมากขึ้น และโอกาสที่อัลกอริทึมจะถูกทำลายจะมากขึ้น

นอกจากการทดสอบที่ได้กล่าวมาแล้ว มีอีกสิ่งหนึ่งที่ควรทดสอบ คือ การเปรียบเทียบเวลาที่ใช้ในการประมวลผลของอัลกอริทึมทั้งสองแบบ และการวัดขนาดของหน่วยความจำที่อัลกอริทึมทั้งสองแบบใช้ ดังนี้ คือ

4.4 การวัดเวลาที่ใช้ในการประมวลผล

วิธีหนึ่งที่ใช้ในการเปรียบเทียบประสิทธิภาพการทำงานของอัลกอริทึม ก็คือ การวัดเวลาที่ใช้ในการประมวลผล ดังนั้น เพื่อที่จะเปรียบเทียบประสิทธิภาพระหว่างอัลกอริทึมไอเดสและอัลกอริทึมเดสในแง่ของเวลาที่ใช้ในการประมวลผลข้อมูลจึงได้ทำการทดสอบโดยทำการเข้ารหัสข้อมูลเนื้อแท้ที่มีขนาดต่าง ๆ กัน ตั้งแต่ขนาด 11 ไบต์ ไปจนถึงข้อมูลขนาด 85 กิโลไบต์ ให้ผ่านอัลกอริทึมเดส และอัลกอริทึมไอเดส ที่มีวิธีการแบ่งข้อมูลตามที่ได้ยกเป็นกรณีศึกษา ว่าต้องใช้เวลาในการประมวลผลเท่าใด ผลการทดสอบได้แสดงไว้ในตารางที่ 4.16 และ 4.17 ซึ่งเป็นตารางแสดงเวลาการเข้ารหัสลับและถอดรหัสลับของข้อมูล ที่ใช้วิธีการเข้ารหัสแบบต่าง ๆ และแสดงอัตราส่วนของเวลาที่อัลกอริทึมไอเดสใช้เมื่อเปรียบเทียบกับอัลกอริทึมเดส จากตารางที่ 4.16 และตารางที่ 4.17 ได้แสดงให้เห็นว่าการเข้ารหัสลับข้อมูลที่มีขนาดเดียวกัน

ตารางที่ 4.16
แสดงเวลาการเข้ารหัสลับของอัลกอริทึมเคสและอัลกอริทึมไฮเคส
และแสดงอัตราส่วนของเวลาที่อัลกอริทึมไฮเคสใช้เมื่อเทียบกับอัลกอริทึมเคส

หน่วย : วินาที

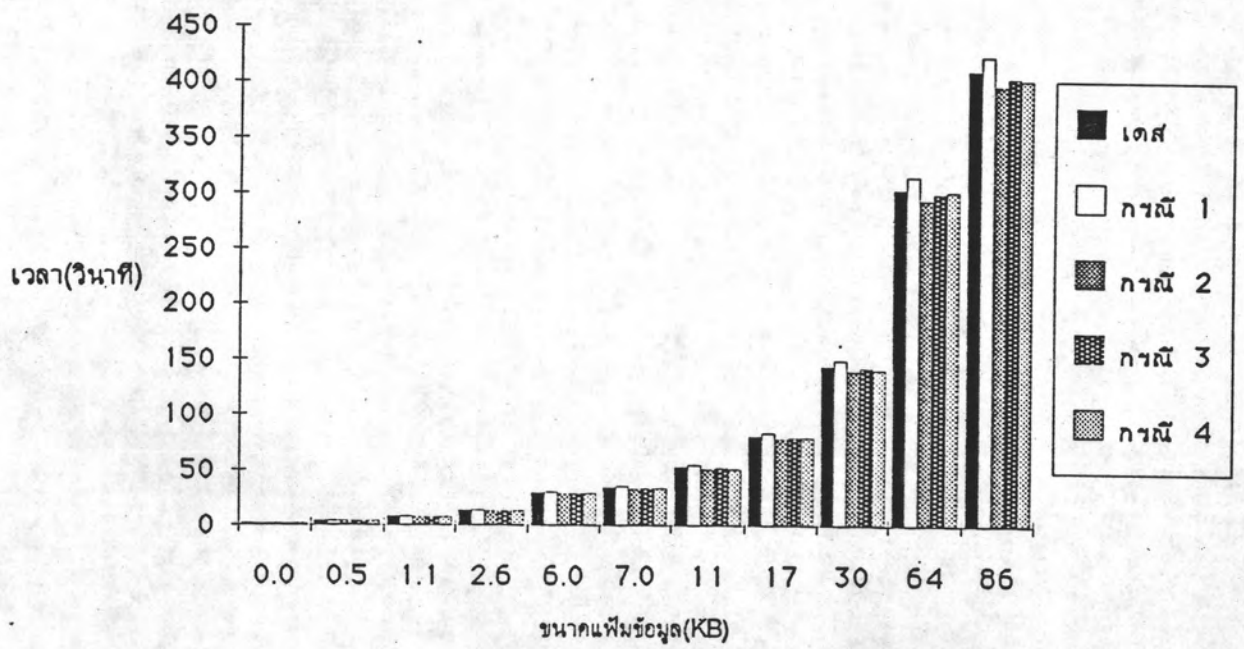
ขนาดของแฟ้มข้อมูล (ไบต์)		11		547		1131		2568		5961		6998		10846		16848		30207		64036		85998	
เคส		1.70	1.00	4.34	1.00	7.96	1.00	14.01	1.00	30.49	1.00	35.53	1.00	54.11	1.00	82.77	1.00	146.92	1.00	309.01	1.00	417.77	1.00
ไฮเคส	กรณี 1	1.54	0.91	4.29	0.99	8.07	1.01	14.39	1.03	31.42	1.03	36.69	1.03	56.35	1.04	86.13	1.04	153.20	1.04	318.62	1.03	428.09	1.02
	กรณี 2	1.70	1.00	4.22	0.97	7.69	0.97	13.62	0.97	29.33	0.96	34.38	0.97	52.34	0.97	79.98	0.97	142.20	0.97	300.55	0.97	404.74	0.97
	กรณี 3	1.48	0.87	4.01	0.92	7.63	0.96	13.73	0.98	29.72	0.97	34.77	0.98	53.16	0.98	81.84	0.99	145.93	0.99	305.28	0.99	408.75	0.98
	กรณี 4	1.65	0.97	4.12	0.95	7.64	0.96	13.40	0.96	29.27	0.96	34.32	0.97	51.74	0.96	79.04	0.95	140.01	0.95	302.86	0.98	401.94	0.96
	กรณี 5	1.60	0.94	4.06	0.94	7.74	0.97	13.39	0.96	30.04	0.99	34.83	0.98	53.33	0.99	81.78	0.99	146.54	1.00	301.16	0.97	407.49	0.98
	กรณี 6	1.76	1.04	4.34	1.00	7.91	0.99	13.84	0.99	29.82	0.98	34.50	0.97	52.78	0.98	81.18	0.98	142.86	0.97	301.04	0.97	405.24	0.97
	กรณี 7	1.70	1.00	4.23	0.97	7.74	0.97	13.68	0.98	29.60	0.97	34.55	0.97	52.62	0.97	80.52	0.97	143.19	0.97	303.41	0.98	406.06	0.97
	กรณี 8	1.82	1.07	4.29	0.99	7.75	0.97	13.67	0.98	29.17	0.96	34.05	0.96	52.07	0.96	79.53	0.96	141.32	0.96	296.71	0.96	396.29	0.95
	กรณี 9	1.70	1.00	4.12	0.95	7.69	0.97	13.52	0.97	29.33	0.96	34.11	0.96	52.34	0.97	80.36	0.97	142.70	0.97	296.49	0.96	395.08	0.95

ตารางที่ 4.17
แสดงเวลาการถอดรหัสลับของอัลกอริทึมเคสและอัลกอริทึมไฮเคส
และแสดงอัตราส่วนของเวลาที่อัลกอริทึมไฮเคสใช้เมื่อเทียบกับอัลกอริทึมเคส

หน่วย : วินาที

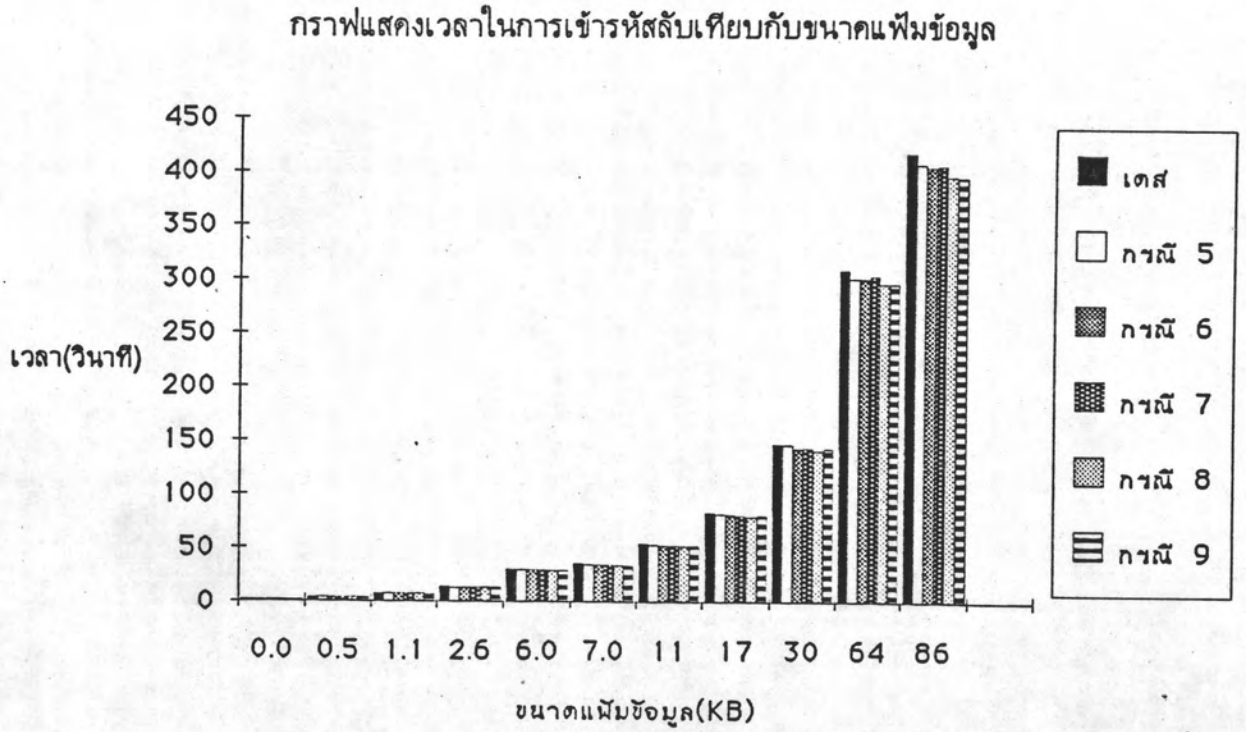
ขนาดของแฟ้มข้อมูล (ไบต์)		11		547		1131		2568		5961		6998		10846		16848		30207		64036		85998	
เคส		1.76	1.00	4.28	1.00	7.85	1.00	13.79	1.00	29.55	1.00	34.33	1.00	52.45	1.00	80.68	1.00	143.57	1.00	302.86	1.00	409.36	1.00
ไฮเคส	กรณี 1	1.53	0.87	4.18	0.98	7.91	1.01	14.00	1.02	30.38	1.03	35.54	1.04	54.43	1.04	83.27	1.03	148.52	1.03	314.83	1.04	422.92	1.03
	กรณี 2	1.70	0.97	4.23	0.99	7.64	0.97	13.43	0.97	28.72	0.97	33.28	0.97	51.08	0.97	78.38	0.97	139.12	0.97	293.85	0.97	396.07	0.97
	กรณี 3	1.48	0.84	4.01	0.94	7.58	0.97	13.40	0.97	29.05	0.98	33.67	0.98	51.85	0.99	79.42	0.98	141.98	0.99	299.72	0.99	402.94	0.98
	กรณี 4	1.65	0.94	4.06	0.95	7.47	0.95	13.19	0.96	28.73	0.97	33.61	0.98	51.13	0.97	79.20	0.98	140.00	0.98	301.32	0.99	401.34	0.98
	กรณี 5	1.60	0.91	4.06	0.95	7.64	0.97	13.51	0.98	29.22	0.99	33.89	0.99	51.96	0.99	80.46	1.00	143.52	1.00	300.61	0.99	402.99	0.98
	กรณี 6	1.76	1.00	4.29	1.00	7.80	0.99	13.51	0.98	29.00	0.98	33.50	0.98	51.47	0.98	78.76	0.98	140.45	0.98	291.21	0.96	397.44	0.97
	กรณี 7	1.70	0.97	4.29	1.00	7.74	0.99	13.46	0.98	28.94	0.98	33.78	0.98	51.30	0.98	78.54	0.97	140.28	0.98	296.21	0.98	397.44	0.97
	กรณี 8	1.81	1.03	4.28	1.00	7.74	0.99	13.35	0.97	28.51	0.96	33.01	0.96	50.37	0.96	77.23	0.96	137.42	0.96	290.83	0.96	389.81	0.95
	กรณี 9	1.65	0.94	4.18	0.98	7.58	0.97	13.24	0.96	28.56	0.97	33.12	0.96	50.75	0.97	77.83	0.96	138.80	0.97	292.54	0.97	392.06	0.96

กราฟแสดงเวลาในการเข้ารหัสลับเทียบกับขนาดแฟ้มข้อมูล



รูปที่ 4.1

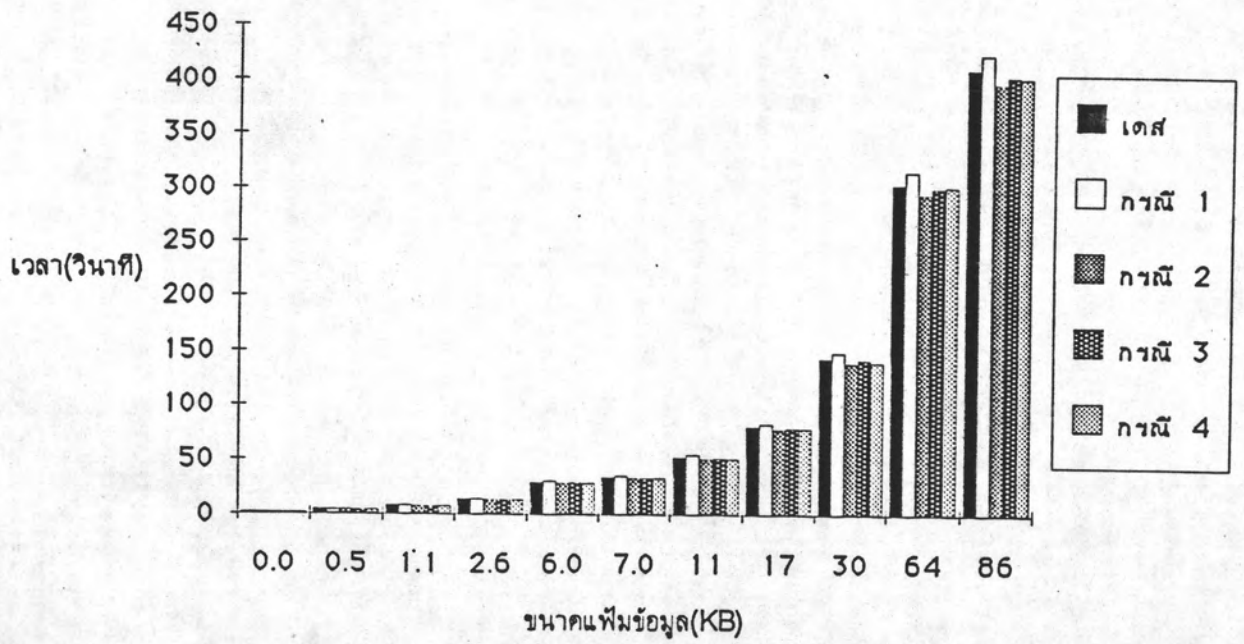
การเปรียบเทียบเวลาที่ใช้ในการเข้ารหัสลับข้อมูลที่มีขนาดต่าง ๆ กัน
โดยใช้อัลกอริทึมเดส และอัลกอริทึมไอเดส แบบกรณีที่ 1-4



รูปที่ 4.2

การเปรียบเทียบเวลาที่ใช้ในการเข้ารหัสลับข้อมูลที่มีขนาดต่าง ๆ กัน
โดยใช้อัลกอริทึมเดส และอัลกอริทึมไอเดส แบบกรณีที่ 5-9

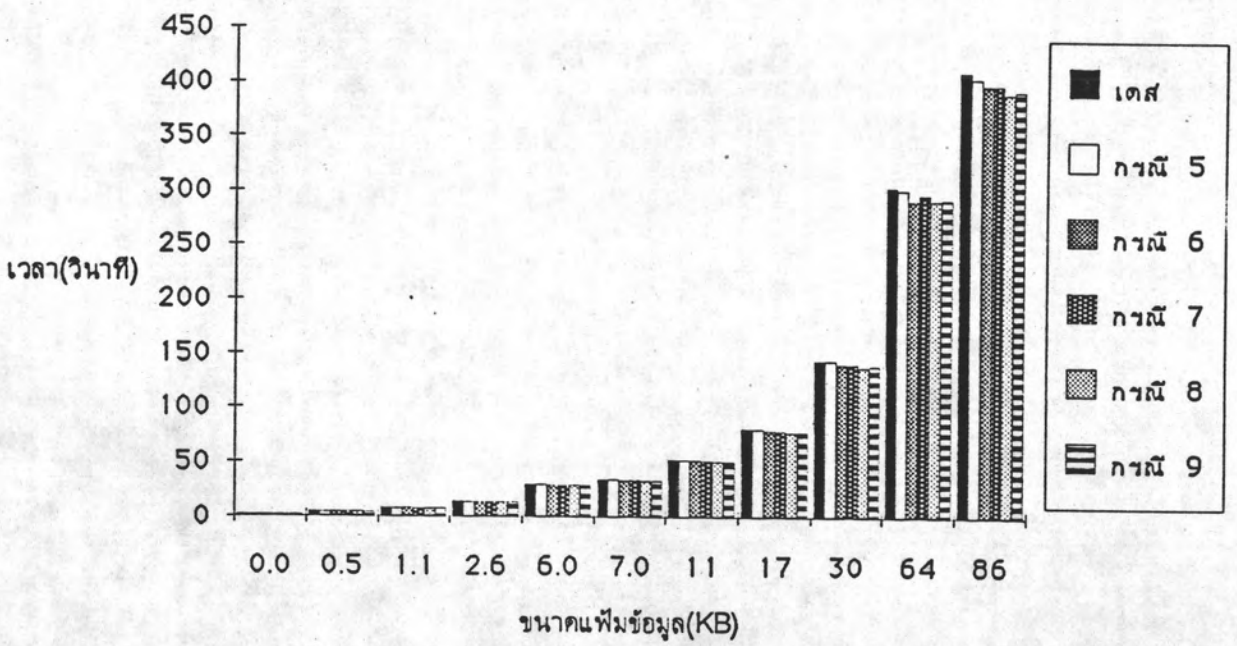
กราฟแสดงเวลาในการถอดรหัสเทียบกับขนาดแฟ้มข้อมูล



รูปที่ 4.3

การเปรียบเทียบเวลาที่ใช้ในการถอดรหัสลับข้อมูลที่มีขนาดต่าง ๆ กัน
โดยใช้อัลกอริทึมเดส และอัลกอริทึมไอเดส แบบกรณีที่ 1-4

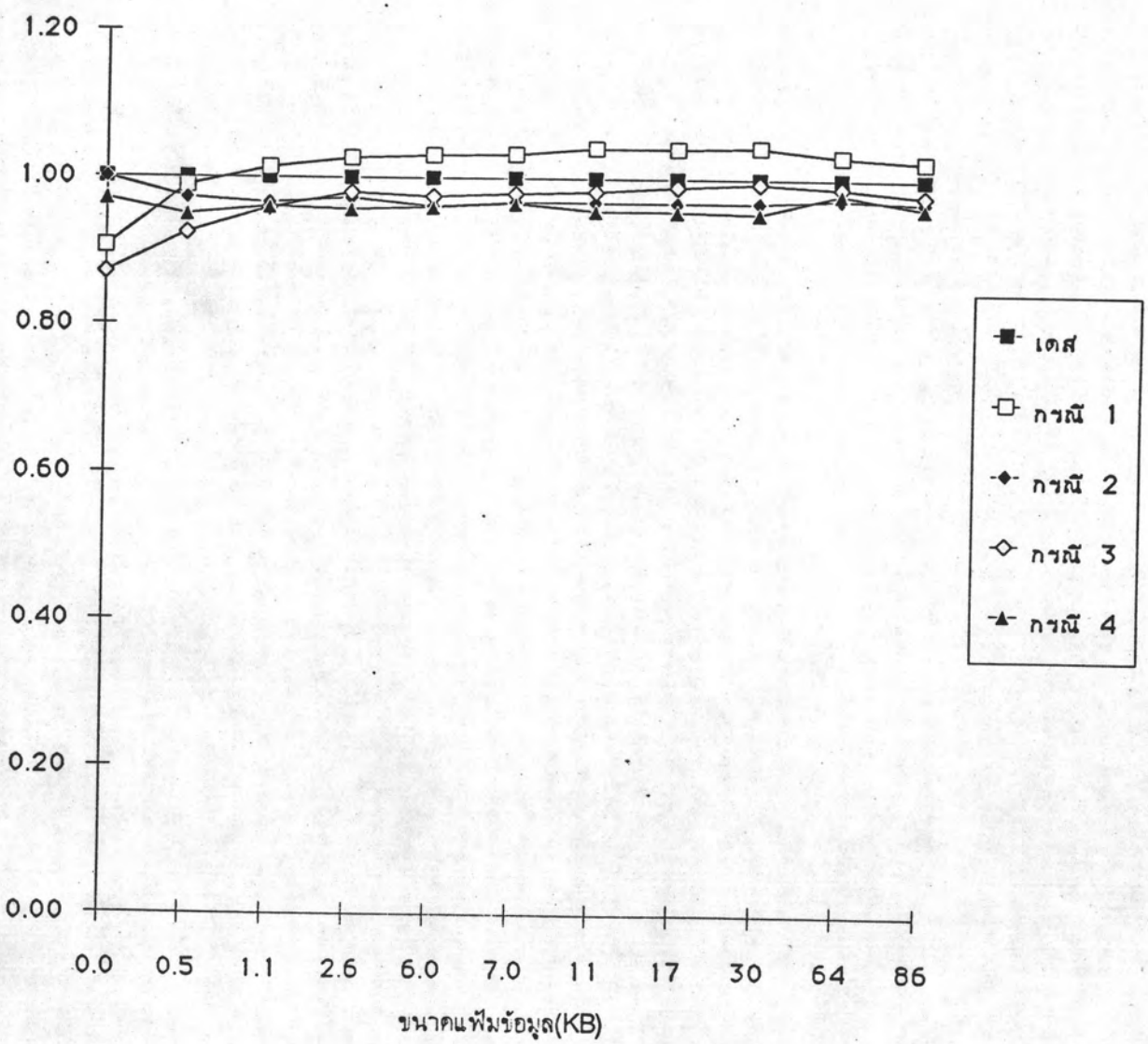
กราฟแสดงเวลาในการถอดรหัสเทียบกับขนาดแฟ้มข้อมูล



รูปที่ 4.4

การเปรียบเทียบเวลาที่ใช้ในการถอดรหัสลับข้อมูลที่มีขนาดต่าง ๆ กัน
โดยใช้อัลกอริทึมเดส และอัลกอริทึมไอเดส แบบกรณีที่ 5-9

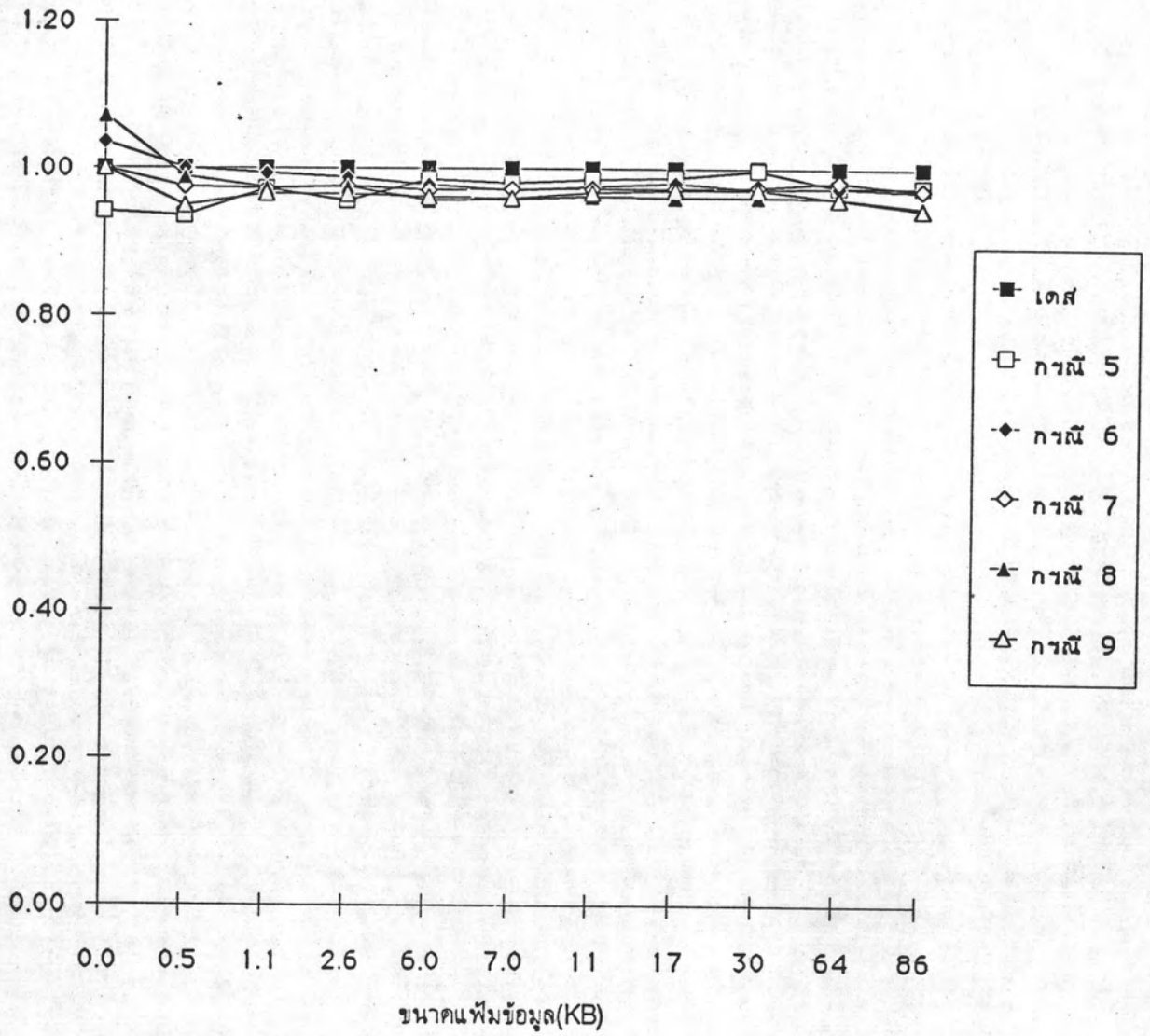
กราฟเปรียบเทียบอัตราส่วนของเวลาในการเข้ารหัสลับของไอเดสเมื่อเทียบกับเดส



รูปที่ 4.5

การเปรียบเทียบอัตราส่วนของเวลาที่ใช้ในการเข้ารหัสลับข้อมูลที่มีขนาดต่าง ๆ กัน โดยใช้อัลกอริทึมไอเดส แบบกรณีที่ 1-4 เทียบกับอัลกอริทึมเดส

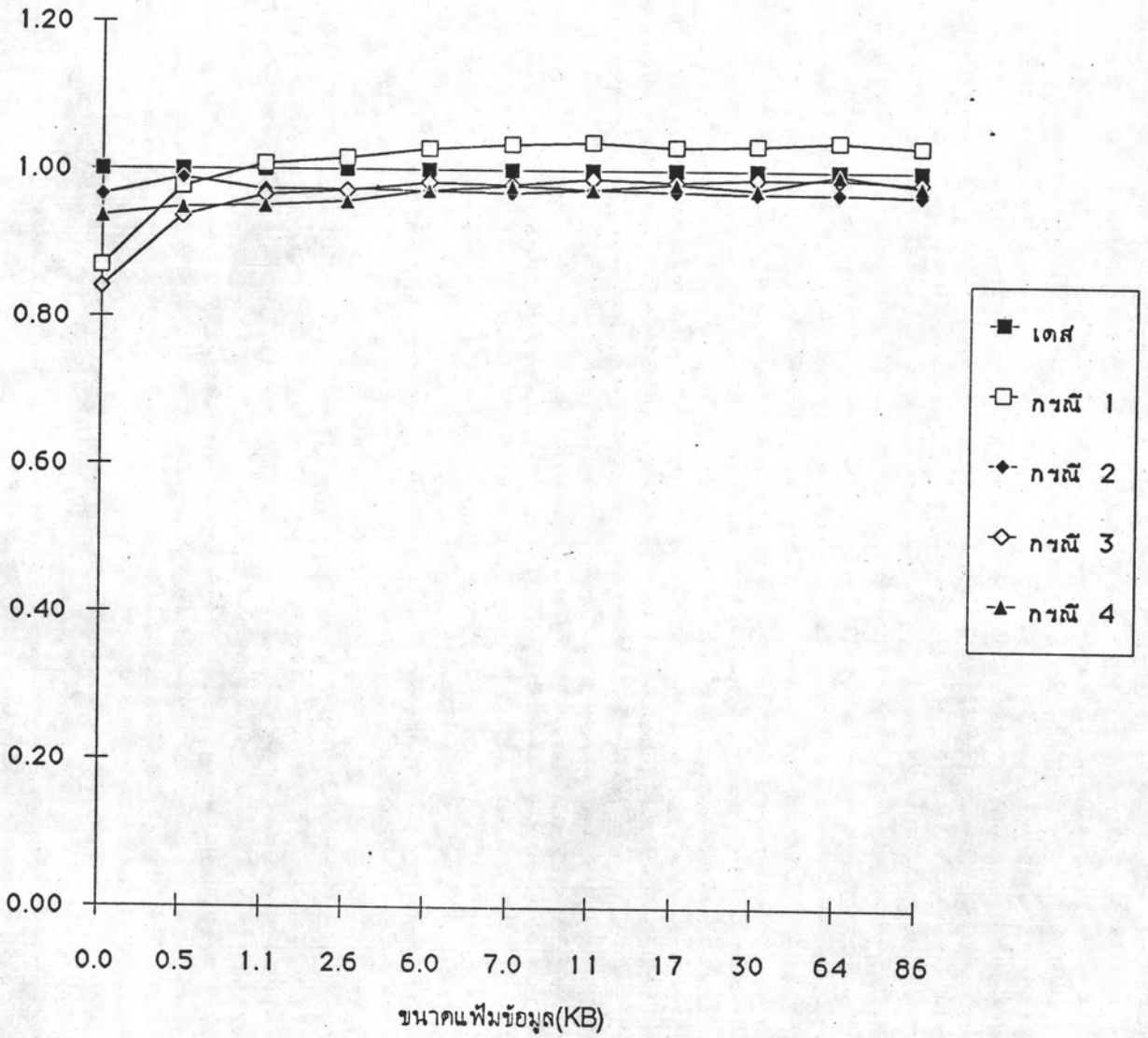
กราฟเปรียบเทียบอัตราส่วนของเวลาในการเข้ารหัสลับของไอเดสเมื่อเทียบกับเดส



รูปที่ 4.6

การเปรียบเทียบอัตราส่วนของเวลาที่ใช้ในการเข้ารหัสลับข้อมูลที่มีขนาดต่าง ๆ กัน
โดยใช้อัลกอริทึมไอเดส แบบกรณีที่ 5-9 เทียบกับอัลกอริทึมเดส

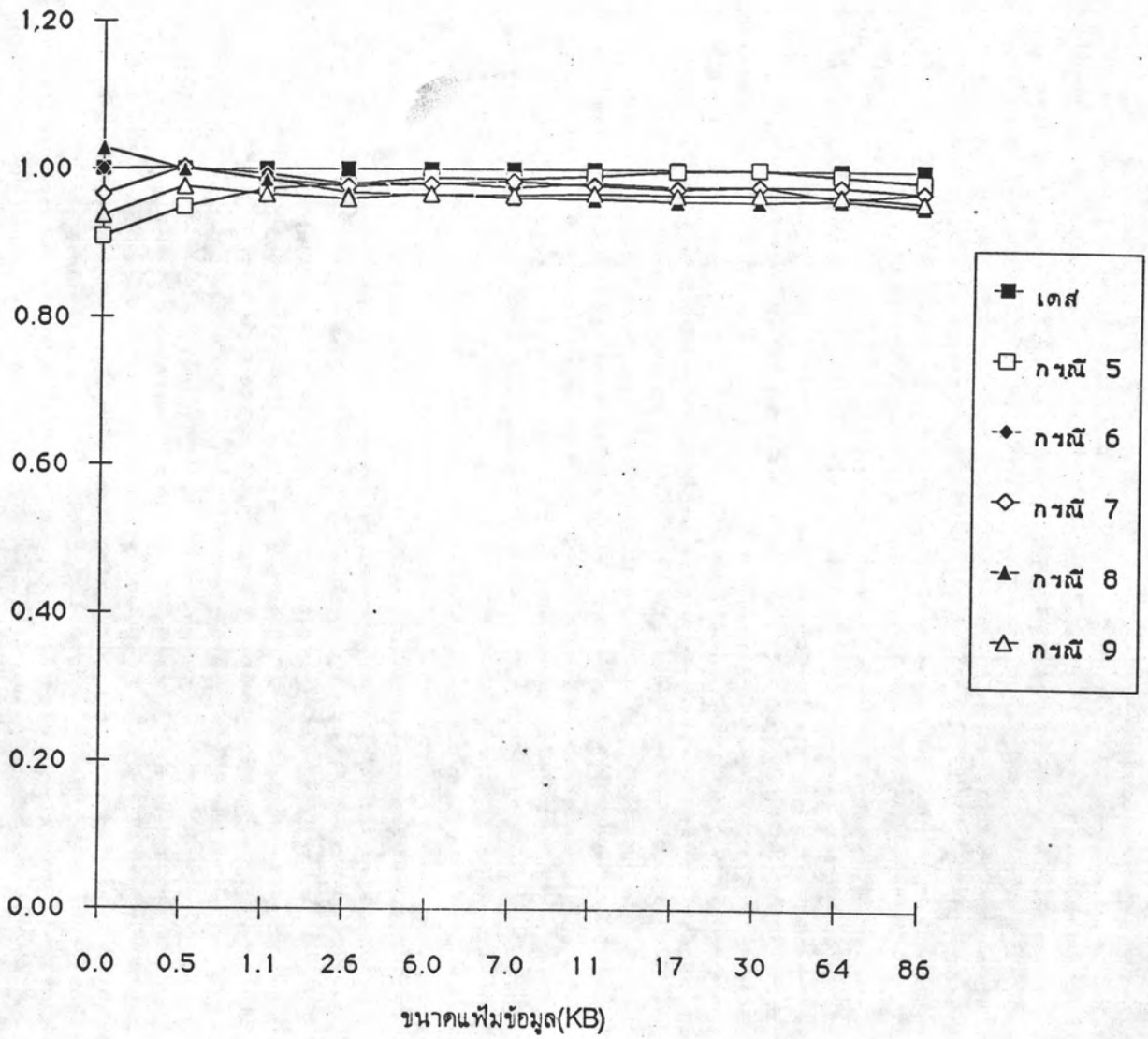
กราฟเปรียบเทียบอัตราส่วนของเวลาในการถอดรหัสลับของไอเดสเมื่อเทียบกับเดส



รูปที่ 4.7

การเปรียบเทียบอัตราส่วนของเวลาที่ใช้ในการถอดรหัสลับข้อมูลที่มีขนาดต่าง ๆ กัน
โดยใช้อัลกอริทึมไอเดส แบบกรณีที่ 1-4 เทียบกับอัลกอริทึมเดส

กราฟเปรียบเทียบอัตราส่วนของเวลาในการถอดรหัสลับของไอเดสเมื่อเทียบกับเดส



รูปที่ 4.8

การเปรียบเทียบอัตราส่วนของเวลาที่ใช้ในการถอดรหัสลับข้อมูลที่มีขนาดต่าง ๆ กัน
โดยใช้อัลกอริทึมไอเดส แบบกรณีที่ 5-9 เทียบกับอัลกอริทึมเดส

จะใช้เวลาในการประมวลผลใกล้เคียงกันไม่แตกต่างกันมาก ไม่ว่าจะเป็นการเข้ารหัสลับผ่านอัลกอริทึมแบบใด แต่จะเห็นได้ว่าการเข้ารหัสลับและการถอดรหัสลับโดยใช้อัลกอริทึมไอเดสในทุก ๆ กรณีศึกษายกเว้นในกรณีที่ 1 จะมีแนวโน้มในการใช้เวลาในการประมวลน้อยกว่าการเข้ารหัสหรือถอดรหัสโดยใช้อัลกอริทึมเดส และเพื่อที่จะเปรียบเทียบเวลาที่ใช้จากการเข้ารหัสทั้งสองแบบให้เห็นชัดเจนยิ่งขึ้นจะแสดงเป็นรูปกราฟ ดังรูปที่ 4.1 ถึงรูปที่ 4.4 และจะทำการเปรียบเทียบเวลาที่ใช้ในการเข้ารหัสลับและถอดรหัสลับ โดยให้เวลาที่ใช้ในการประมวลผลโดยอัลกอริทึมเดสเป็น 1 และเวลาที่ใช้ในการประมวลผลโดยอัลกอริทึมไอเดส แบบต่าง ๆ เป็นอัตราส่วนกับอัลกอริทึมเดส ผลที่ได้แสดงไว้ในรูปที่ 4.5 ถึงรูปที่ 4.8 จะเห็นได้ว่าเวลาที่ใช้ในการประมวลโดยอัลกอริทึมไอเดสแบบต่าง ๆ ยกเว้นแบบที่ 1 ใช้เวลาน้อยกว่าการประมวลผลด้วยอัลกอริทึมเดสทั้งสิ้น

การที่ผลการทดสอบออกมาในลักษณะนี้ เราอาจจะวิเคราะห์ได้ว่าในการเข้ารหัสอัลกอริทึมไอเดส กรณีที่ 1 ซึ่งมีวิธีการแบ่งข้อมูลเพื่อนำไปเปิดหาค่าจากตาราง S-boxes จะได้จำนวนกลุ่มทั้งหมดเท่ากับ 9 กลุ่ม ในขณะที่กรณีอื่น ๆ มีวิธีการแบ่งข้อมูลออก ตั้งแต่ 4 ถึง 7 กลุ่ม ส่วนการแบ่งกลุ่มของอัลกอริทึมเดสจะเท่ากับ 8 กลุ่ม จะเห็นได้ว่าจำนวนวิธีในการแบ่งกรณีที่ 1 จะมีจำนวนกลุ่มมากที่สุด การที่มีจำนวนกลุ่มมากเวลาที่ใช้ในการประมวลผลจะมากขึ้น เนื่องจากต้องใช้เวลาในการเปิดค่าจากตาราง S-boxes มากขึ้น

4.5 การวัดขนาดของหน่วยความจำที่อัลกอริทึมไอเดสใช้

ในการวัดประสิทธิภาพการทำงานของอัลกอริทึม นอกจากที่ได้กล่าวมาแล้ว อีกสิ่งหนึ่งที่สามารถนำมาประกอบการพิจารณาได้ คือ ขนาดของหน่วยความจำที่อัลกอริทึมนั้นใช้ จากการวัดขนาดของหน่วยความจำของอัลกอริทึมเดส และอัลกอริทึมไอเดสที่มีวิธีการแบ่งกลุ่มตามตัวอย่างที่ยกมาเป็นกรณีศึกษา แสดงไว้ในตารางที่ 4.18 นอกจากนี้ยังได้แสดงขนาดของตาราง S-boxes ของอัลกอริทึมแต่ละแบบไว้ด้วย

	ขนาดหน่วยความจำ (ไบต์)	ขนาดตาราง S-boxes (ไบต์)
อัลกอริทึมเดส	41072	2546
อัลลกอริทึมไอเดส		
กรณีที่ 1	48807	16640
2	49457	31863
3	41981	3874
4	42220	6767
5	44306	12735
6	48780	28216
7	50768	36343
8	57144	60856
9	54175	50229

ตารางที่ 4.18 แสดงขนาดหน่วยความจำที่อัลกอริทึมเดส และอัลกอริทึมไอเดส กรณีต่าง ๆ ใช้

จากตาราง 4.18 จะเห็นว่าอัลกอริทึมเดสใช้หน่วยความจำน้อย และที่ใช้หน่วยความจำมากที่สุดคือ อัลกอริทึมไอเดส กรณีที่ 8 ซึ่งมีการแบ่งข้อมูลเข้าเป็นกลุ่มละ 12 บิต และได้ผลลัพธ์กลุ่มละ 8 บิต และจากตารางจะเห็นว่าขนาดของหน่วยความจำที่อัลกอริทึมใช้จะสัมพันธ์สอดคล้องกับขนาดของตาราง S-boxes นั่นคือ ถ้าตารางมีขนาดใหญ่ อัลกอริทึมจะใช้หน่วยความจำมาก แต่ถ้าตารางมีขนาดเล็ก อัลกอริมทึมก็จะใช้หน่วยความจำน้อยกว่า แต่จะมีกรณีที่ 6 ที่ขนาดของตารางมีขนาดใหญ่กว่ากว่ากรณีที่ 1 แต่ใช้หน่วยความจำน้อยกว่าเล็กน้อย

แต่ขณะเดียวกันขนาดของตาราง S-boxes จะขึ้นอยู่กับขนาดของการแบ่งกลุ่ม นั่นคือ ถ้ากลุ่มข้อมูลมีขนาดใหญ่ จะทำให้ตารางมีขนาดใหญ่ ถ้ากลุ่มมีขนาดเล็ก ตารางก็จะมีขนาดเล็ก เราสามารถคำนวณขนาดของตาราง S-box ได้ดังนี้ คือ

ถ้า ข้อมูลเข้า มีขนาด X บิต
ข้อมูลออก มีขนาด Y บิต
ดังนั้น จำนวนคอลัมน์ในตารางเท่ากับ 2^{Y}
จำนวนแถว ในตารางเท่ากับ 2^{X-Y}
จะได้ค่าทั้งหมดในตารางเท่ากับ $2^{Y} \times 2^{X-Y} = 2^{X}$ จำนวน

ตัวอย่างเช่น

ข้อมูลเข้า 5 บิต ได้ข้อมูลออก 3 บิต ขนาดของตาราง $2^{5} = 32$
ข้อมูลเข้า 6 บิต ได้ข้อมูลออก 4 บิต ขนาดของตาราง $2^{6} = 64$
ข้อมูลเข้า 7 บิต ได้ข้อมูลออก 4 บิต ขนาดของตาราง $2^{7} = 128$
ข้อมูลเข้า 12 บิต ได้ข้อมูลออก 8 บิต ขนาดของตาราง $2^{12} = 4096$

ขนาดของตารางที่ได้จากการคำนวณนี้ จะเป็นขนาดของตารางสำหรับข้อมูลเพียงกลุ่มเดียว และตัวเลขที่ได้จะเป็นจำนวนตัวเลขทั้งหมดในตาราง ไม่ได้หมายถึงขนาดของหน่วยความจำมีหน่วยเป็นไบต์

จะเห็นได้ว่า เมื่อขนาดของกลุ่มข้อมูลเพิ่มขึ้นเพียง 1 บิต ขนาดของตาราง S-boxes จะเพิ่มขึ้นอีก 1 เท่าจากตารางเดิม ดังนั้น ถ้าขนาดของกลุ่มข้อมูลใหญ่ขึ้น จะทำให้ตารางมีขนาดใหญ่มาก ทำให้เปลืองเนื้อที่ของหน่วยความจำมาก และเป็นผลทำให้ขนาดของหน่วยความจำของอัลกอริทึมใหญ่ขึ้น ดังนั้น เราสามารถสรุปได้ว่า ขนาดของหน่วยความจำนี้จะขึ้นอยู่กับขนาดของกลุ่มที่แบ่งนั่นเอง

จากการทดสอบและเปรียบเทียบประสิทธิภาพการทำงานของอัลกอริทึมไอเดส และอัลกอริทึมเดส ทั้งหมดที่ได้กล่าวมาแล้วนั้น เราอาจกล่าวได้ว่าอัลกอริทึมไอเดสมีประสิทธิภาพในการทำงานไม่ด้อยไปกว่าอัลกอริทึมเดส โดยดูจากวิธีการทดสอบแบบต่าง ๆที่ออกมาจะให้ผลที่ใกล้เคียงหรือดีกว่าอัลกอริทึมเดส และยังแสดงให้เห็นว่ามีคุณสมบัติที่ดีครบถ้วน และจากบทที่ 3 ที่สรุปได้ว่าอัลกอริทึมไอเดสมีความซับซ้อนกว่า ทำให้เราสามารถสรุปได้ว่าอัลกอริทึมไอเดส เป็นอัลกอริทึมที่มีประสิทธิภาพ สามารถนำมาใช้เข้ารหัสข้อมูล เพื่อป้องกันข้อมูลให้ปลอดภัยได้

4.6 การทดสอบการเปลี่ยนวิธีการคำนวณค่าสับคีย์

แนวความคิดในการปรับเปลี่ยนอัลกอริทึมเดสอีกวิธีหนึ่ง ซึ่งได้เสนอไว้ในบทที่ 3 คือ การเปลี่ยนวิธีการคำนวณค่าสับคีย์ ดังนั้น เพื่อที่จะดูว่าเมื่อมีการเข้ารหัสลับข้อมูลที่มีการใช้วิธีการคำนวณค่าสับคีย์ที่แตกต่างกันจะให้ผลลัพธ์ที่แตกต่างกันอย่างไร จึงได้มีการทดสอบ ดังนี้คือ

สร้างอัลกอริทึมเดส 2 อัลกอริทึม ให้อัลกอริทึมหนึ่งมีการคำนวณค่าสับคีย์โดยใช้วิธีการเลื่อนตำแหน่งบิตแบบหมุนเวียนไปทางซ้าย ซึ่งเป็นวิธีที่อัลกอริทึมเดสแบบมาตรฐานใช้กันอยู่ และอัลกอริทึมอีกอัลกอริทึมหนึ่งใช้วิธีการเลื่อนตำแหน่งบิตแบบหมุนเวียนไปทางขวา นำอัลกอริทึมทั้งสองมาเข้ารหัสลับข้อมูลเนื้อแท้ชุดเดียวกัน คือ 4142434445464748 ซึ่งแสดงไว้ในรูปของเลขฐานสิบหก และใช้คีย์สำหรับเข้ารหัสลับชุดเดียวกัน คือ 0123456789abcdef ซึ่งเป็นเลขฐานสิบหกเช่นเดียวกัน ได้ผลลัพธ์ คือ

เมื่อเลื่อนตำแหน่งบิตแบบหมุนเวียนไปทางซ้าย ได้ผลลัพธ์ fa46606cfc0449cc

เมื่อเลื่อนตำแหน่งบิตแบบหมุนเวียนไปทางขวา ได้ผลลัพธ์ ce55e6a0809b3236

และวัดระยะแฮมมิ่งของข้อมูล 2 ชุดนี้จะได้เท่ากับ 37 แสดงว่าข้อมูลเข้ารหัสบิตต่อบิตมีความแตกต่างกัน ถ้าหากมาดูผลของการคำนวณค่าสับคีย์ ในส่วนของการเลื่อนตำแหน่งบิตแบบหมุนเวียนจะได้ผลลัพธ์ดังตารางที่ 4.19 โดยที่ค่าเริ่มต้นก่อนการเลื่อนตำแหน่งบิตของทั้งสองแบบ คือ ข้อมูลในรอบที่ 0 ของตาราง ค่าเริ่มต้นของอัลกอริทึมทั้งสองแบบนี้จะเหมือนกัน คือ เป็นค่าคีย์สำหรับเข้ารหัสลับที่ผ่านการจัดลำดับตาม PC-1 แล้ว ดังรูป 2.4 และการเลื่อนตำแหน่งบิตแบบหมุนเวียนจะเลื่อนบิตตามตารางที่ 4.20 ซึ่งเป็นตารางการเลื่อนบิตที่อัลกอริทึมเดสใช้ในการเลื่อนตำแหน่งบิต การเลื่อนตำแหน่งบิตในแต่ละรอบในทิศทางตรงกันข้าม จะให้ได้ผลลัพธ์ที่แตกต่างกันในแต่ละรอบ แต่ก็มีโอกาสที่ผลลัพธ์ที่ได้จะไปซ้ำกับรอบอื่น ๆ เช่น ในรอบที่ 15 ผลลัพธ์ของการหมุนเวียนไปทางขวาจะเหมือนกับรอบที่ 1 ของการหมุนเวียนไปทางซ้าย หรือในรอบที่ 16 ผลลัพธ์ของการหมุนเวียนทั้งสองวิธีจะเหมือนกัน และจะเท่ากับข้อมูลเริ่มต้นในรอบที่ 0 ส่วนข้อมูลอื่น ๆ จะมีความแตกต่างกันแต่เป็นในลักษณะที่บางส่วนของสับคีย์จะเหมือนกัน หรือเป็นลักษณะที่สับคีย์เหมือนกัน แต่อยู่ในตำแหน่งที่แตกต่างกัน

ดังนั้น วิธีการเข้ารหัสโดยการเปลี่ยนวิธีการคำนวณสับคีย์ให้เป็นการเลื่อนตำแหน่งบิตไปทางขวา ก็เป็นวิธีหนึ่งที่สามารถมาปรับเปลี่ยนอัลกอริทึมเดสได้ แต่ควรมาใช้ในลักษณะที่ประกอบกับอัลกอริทึมไอเดส

รอบที่	การเลื่อนตำแหน่งบิตแบบ			
	หมุนเวียนไปทางซ้าย		หมุนเวียนไปทางขวา	
	C	D	C	D
0	f033550	5533f00	f033550	5533f00
1	e066aa1	aa67e00	7819aa8	2a99f80
2	c0cd543	54cfc01	3c0cd54	154cfc0
3	033550f	533f005	0f03355	05533f0
4	0cd543c	4cfc015	43c0cd5	154cfc0
5	33550f0	33f0055	50f0335	005533f
6	cd543c0	cfc0154	543c0cd	c0154cf
7	3550f03	3f00553	550f033	f005533
8	d543c0c	fc0154c	d543c0c	fc0154c
9	aa87819	f802a99	6aa1e06	7e00aa6
10	aa1e066	e00aa67	9aa8781	9f802a9
11	a87819a	802a99f	66aa1e0	67e00aa
12	a1e066a	00aa67e	19aa878	99f802a
13	87819aa	02a99f8	066aa1e	a67e00a
14	1e066aa	0aa67e0	819aa87	a99f802
15	7819aa8	2a99f80	e066aa1	aa67e00
16	f033550	5533f00	f033550	5533f00

ตารางที่ 4.19 แสดงผลลัพธ์ของข้อมูลที่ถูกเลื่อนตำแหน่งบิตโดยการหมุนเวียนไปทางซ้ายและขวา

รอบที่	จำนวนบิตที่เลื่อน
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
9	1
10	2
11	2
12	2
13	2
14	2
15	2
16	1

ตารางที่ 4.20 ตำแหน่งสำหรับการเลื่อนบิต

4.7 การหลีกเลี่ยงการใช้วีคคีย์ (Weak Key) และเซมิ-วีคคีย์ (Semi-Weak Key)

ในการเข้ารหัสลับข้อมูลเพื่อป้องกันให้ข้อมูลมีความปลอดภัย นอกจากจะต้องเลือกใช้อัลกอริทึมที่มีการทำงานที่ซับซ้อนแล้ว อีกสิ่งหนึ่งที่มีความสำคัญ ก็คือ การเลือกคีย์เพื่อใช้สำหรับการเข้ารหัสลับ จะต้องเป็นคีย์ที่ดีไม่ไปลดประสิทธิภาพและความซับซ้อนของอัลกอริทึมนั้น

ความซับซ้อนในการทำงานของอัลกอริทึมเดสจะลดลง ถ้าหากว่าสับคีย์ที่ใช้สำหรับการเข้ารหัสข้อมูลในแต่ละรอบมีค่าเหมือนกันทุก ๆ ค่า นั่นคือ $K_1 = K_2 = \ldots = K_{16}$ เช่น มีค่าสับคีย์เป็น 0 หรือ 1 ทั้งหมด ดังนั้นจึงควรหลีกเลี่ยงคีย์สำหรับเข้ารหัสลับที่เมื่อคำนวณหาค่าสับคีย์แล้วทำให้ได้ค่าสับคีย์ที่เหมือนกัน เราจะเรียกลักษณะคีย์แบบนี้ว่าวีคคีย์ (Weak-key) ซึ่งจะเป็นคีย์ที่ทำให้การเข้ารหัสและการถอดรหัสมีผลเหมือนกัน นั่นคือ ถ้าข้อมูลเนื้อแท้เข้ารหัสลับด้วยคีย์นี้ แล้วมีการเข้ารหัสลับซ้ำด้วยคีย์นี้อีกครั้งหนึ่งก็จะสามารถถอดรหัสได้ข้อมูลเนื้อแท้ออกมา จะมีคีย์ต่าง ๆ ต่อไปนี้ที่ถือว่าเป็นวีคคีย์ เนื่องจากคีย์เหล่านี้เมื่อนำมาคำนวณหาค่าสับคีย์แล้วจะได้คีย์ที่เป็น 0 หรือ 1 ทั้งหมด

01	01	01	01	01	01	01	01
FE	FE	FE	FE	FE	FE	FE	FE
IF	IF	IF	IF	0E	0E	0E	0E
E0	E0	E0	E0	F1	F1	F1	F1

นอกจากวีคคีย์ ยังมีคีย์สำหรับเข้ารหัสที่เกิดเป็นคู่ เรียกว่า เซมิ-วีคคีย์ (Semi-Weak Key) โดยที่แต่ละคู่ของคีย์เหล่านี้จะให้ค่าสับคีย์ที่เหมือนกัน แต่ตรงกันข้าม คือ สับคีย์ K_1 ของคีย์หนึ่งจะเหมือนกับสับคีย์ K_{16} ของอีกคีย์หนึ่ง ส่วน K_2 จะเหมือนกับ K_{15} เป็นเช่นนี้เรื่อยไป ดังนั้นการเข้ารหัสข้อมูลด้วยคีย์หนึ่ง แล้วเข้ารหัสซ้ำด้วยอีกคีย์หนึ่งที่คู่กัน จะทำให้สามารถถอดรหัสได้ข้อมูลเนื้อแท้ออกมาได้ ค่าเซมิ-วีคคีย์ มีทั้งหมด 6 คู่ คือ

01	FE	01	FE	01	FE	01	FE
FE	01	FE	01	FE	01	FE	01
1F	E0	1F	E0	0E	F1	0E	F1
E0	1F	E0	1F	F1	0E	F1	0E

01	E0	01	E0	01	F1	01	F1
E0	01	E0	01	F1	01	F1	01
1F	FE	1F	FE	0E	FE	0E	FE
FE	1F	FE	1F	FE	0E	FE	0E
01	1F	01	1F	01	0E	01	0E
1F	01	1F	01	0E	01	0E	01
E0	FE	E0	FE	F1	FE	F1	FE
FE	E0	FE	E0	FE	F1	FE	F1

ดังนั้นในการเข้ารหัสลับด้วยอัลกอริทึมเดส ควรจะหลีกเลี่ยงคีย์เหล่านี้ ในส่วนของอัลกอริทึมไอเดสซึ่งมีขั้นตอนการทำงานในการคำนวณสับคีย์เหมือนกับอัลกอริทึมเดส ต่างกันที่วิธีการแบ่งกลุ่มข้อมูลที่ใช้เปิดค่าจากตาราง S-boxes เท่านั้น ก็จะได้รับผลกระทบจากการใช้คีย์เหล่านี้เช่นเดียวกัน จึงควรจะหลีกเลี่ยงการใช้คีย์เหล่านี้ด้วย