การเพิ่มความปลอดภัยและประสิทธิภาพในการรับส่งข้อความ SMS
Security and Efficiency in SMS Messaging

ภูกิจ บุรีภักดี[1] และ ปราโมทย์ กั่วเจริญ[2]

[1]นักศึกษาปริญญาโท สาขาวิทยาการคอมพิวเตอร์ คณะสถิติประยุกต์ สถาบันบัณฑิตพัฒนบริหารศาสตร์

[2]ผู้ช่วยศาสตราจารย์ สาขาวิทยาการคอมพิวเตอร์ คณะสถิติประยุกต์ สถาบันบัณฑิตพัฒนบริหารศาสตร์

[1] phookit.b@gmail.com, [2]pramote@as.nida.ac.th

บทคัดย่อ

บทความนี้นำเสนอวิธีการรักษาความปลอดภัยและการเพิ่มประสิทธิภาพในการส่งข้อความ SMS ด้านการรักษาความปลอดภัยสามารถทำได้โดยการเข้ารหัสลับข้อความ SMS โดยใช้เส้นโค้ง (Elliptic Curve Cryptography) ซึ่งเป็นแนวทางในการเข้ารหัสลับแบบกุญแจสาธารณะ โดยมีพื้นฐานมาจากโครงสร้างพีชคณิตของเส้นโค้งเหนือฟิลด์จำกัด เพื่อที่จะแก้ไขข้อจำกัดของขนาดของข้อความ SMS มีการแปลงข้อความเป็นรหัสตัวเลข ซึ่งจะเพิ่มประสิทธิภาพในการส่งข้อความตัวอักษรละตินที่ยาวกว่า 160 ตัวอักษร และ 70 ตัวอักษรสำหรับภาษาไทย

เพื่อตรวจสอบวิธีการนี้ ได้มีการพัฒนาโปรแกรมประยุกต์สำหรับการนี้บนโทรศัพท์มือที่ใช้ระบบปฏิบัติแอนดรอยด์ โดยการนำคำศัพท์ทั้งที่เป็นภาษาอังกฤษและภาษาไทยมาสร้างดัชนีโดยใช้พื้นที่เก็บ 24 บิต ซึ่งฐานข้อมูลเบื้องต้นมีรายการทั้งสิ้น 101,910 รายการ ข้อความที่ถูกแปลงเป็นตัวเลขแล้วจะถูกเข้ารหัสด้วย ECC SEC2 โดยมีความยาวของกุญแจขนาด 160 บิต

คำสำคัญ: ความปลอดภัย, ECC, การเข้ารหัสลับ, เอสเอ็มเอส

Abstract

This paper presents a method for providing security and efficiency in SMS messaging. The security aspect is accomplished by encrypting the SMS message using elliptic curve cryptography (ECC) which is an approach to public-key cryptography based on the algebraic structure of elliptic curves over finite fields. To overcome the limitation of the size of the SMS message, message encoding is performed. This provides efficiency in sending messages longer than 160 characters for Latin alphabets and 70 characters for Thai. By encoding the message, the amount of SMS messages can be reduced.

To verify this method, an application is implemented on an Android phone. Both English and Thai words are indexed using 24-bit space. The initial database contains 101,910 entries. The resulting encoded message is encrypted using ECC SEC2 recommendation with the key of size 160 bits.

Keyword: security, elliptic curve cryptography, encryption, SMS.

1. บทนำ

ปัจจุบันการส่งข้อความ SMS สามารถส่งได้ทุกที่ทุกเวลาไม่ว่าผู้รับจะอยู่ในพื้นที่ที่มีสัญญาณหรือไม่ก็ตาม ถ้าหากปลายทางของผู้รับ SMS ไม่มีสัญญาณ ระบบจะเก็บข้อมูลไว้ จนกว่าปลายทางจะมีสัญญาณ จากนั้นระบบทำการส่งข้อมูลไปในทันที [1]

อย่างไรก็ตามข้อจำกัดของ SMS ก็คือความยาวของจำนวนตัวอักษรที่สามารถส่งได้ นั่นคือ กรณีที่ข้อความเป็นอักษรละติน จะส่งได้ 160 ตัวอักษรต่อ SMS หนึ่งข้อความ และกรณีที่ข้อความไม่เป็นตัวอักษรละติน เช่น เป็นภาษาไทยจะส่งได้ 70 ตัวอักษรต่อ SMS หนึ่งข้อความ [2] ดังนั้นถ้าพิมพ์ข้อความเกินจากนี้ก็จะใช้ SMS มากกว่าหนึ่งข้อความในการส่งต่อหมายเลขโทรศัพท์มือถือ อีกประการหนึ่งคือข้อความที่ส่งมานั้นยังไม่มีระบบการรักษาความปลอดภัยในการป้องกันการเปิดอ่านข้อความ

ดังนั้นผู้ศึกษาจึงได้พัฒนาโปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่ เพื่อมาประยุกต์ใช้ในการส่งข้อความโดยเมื่อเป็นฝั่งส่ง ข้อความจะถูกแปลงเป็นรหัสตัวเลข (Encoded Message) ตามดัชนีที่สร้างขึ้นตามพจนานุกรม

ราชบัณฑิตยสถาน [3] และพจนานุกรมภาษาอังกฤษ [4] หลังจากนั้นนำข้อความที่ถูกแปลงเป็นตัวเลขมาเข้ารหัสแบบกุญแจอสมมาตร (Asymmetric Key Cryptography) ด้วยกุญแจสาธารณะ (Public Key) ของฝั่งรับ เมื่อฝั่งรับได้รับจะถอดรหัสแบบกุญแจอสมมาตร ด้วยกุญแจส่วนตัว (Private Key) แล้วจึงจะถอดรหัสตัวเลข (Decode Message) เพื่อให้ได้ข้อความต่อไป วิธีนี้ทำให้สามารถส่งข้อความที่ได้มากขึ้น ลดค่าใช้จ่ายในการส่งข้อความได้ อีกทั้งยังสามารถเพิ่มความปลอดภัยในการเปิดอ่านข้อมูลได้อีกด้วย ถึงแม้ว่าข้อความนี้ไปอยู่ในมือของผู้ประสงค์ร้าย ก็จะไม่สามารถเปิดอ่านได้ เนื่องจากต้องมีกุญแจในการถอดรหัสข้อความ

2. ความรู้พื้นฐานและทฤษฎีที่เกี่ยวข้อง

2.1 SMS (Short Message Service)

SMS คือการบริการส่งข้อความสั้น ๆ ผ่านทางโทรศัพท์มือถือ กรณีที่ส่งเป็นภาษาไทยจะส่งได้ 70 ตัวอักษรต่อหนึ่งข้อความและ 160 ตัวอักษรต่อข้อความภาษาอังกฤษ จุดเด่นของบริการ SMS คือ สามารถส่งไปยังผู้รับโดยไม่ต้องกังวลว่าพื้นที่ของผู้รับจะมีสัญญาณหรือไม่ในขณะนั้น หากทางปลายทางไม่มีสัญญาณระบบ SMS นี้จะเก็บข้อมูลไว้จนกว่าปลายทางมีจะสัญญาณทางระบบจึงจะทำการส่งข้อมูลไปในทันที นอกจากนี้แล้ว SMS ยังสามารถส่งข้อความที่ได้รับมาต่อไปยังหมายเลขอื่น ๆ ได้อย่างไม่จำกัดอีกด้วย [1]

2.2 Character Encoding [5]

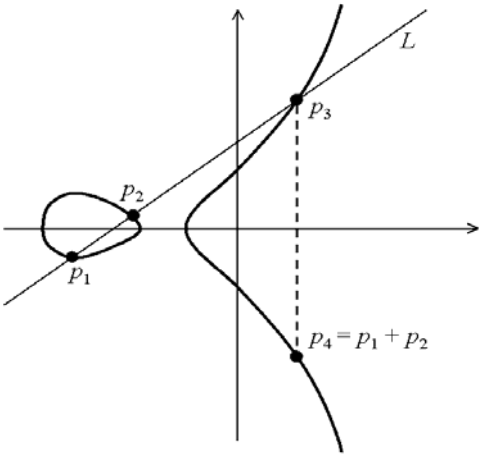
รหัสตัวเลขที่คอมพิวเตอร์ใช้เพื่อแทนตัวอักษร แบ่งเป็น 2 ประเภท ดังต่อไปนี้

2.2.1 Fixed-Length Character Encoding เป็นการเข้ารหัสโดยใช้รหัสที่มีจำนวนหลักตายตัว การเข้ารหัสในลักษณะนี้จะเป็นการแทนที่ตัวอักษรใด ๆ ด้วยตัวเลขที่มีจำนวนบิตแบบคงที่ เข้ารหัสด้วยตัวเลข 8 บิต 16 บิต หรือ 32 บิต เช่น ASCII, Extended ASCII, UTF-16 เป็นต้น

2.2.2 Variable-Length Character Encoding เป็นการเข้ารหัสโดยใช้รหัสที่มีจำนวนหลักไม่คงที่ รหัสแต่ละตัวอาจจะสั้นยาวไม่เท่ากัน ตามแต่ผู้ออกแบบกำหนด โดยส่วนใหญ่จะใช้วิธีกำหนดช่วงเอาไว้ว่า รหัสในช่วงใดช่วงหนึ่งจะมีความยาวที่ระดับหนึ่ง ในขณะที่ในอีกช่วงหนึ่งก็จะมีความยาวที่อีกระดับหนึ่ง ยกตัวอย่างเช่น UTF-8 เป็นต้น

2.3 ECC (Elliptic Curve Cryptography)

ECC เป็นอัลกอริทึมที่ใช้ในการเข้ารหัสแบบอสมมาตร (Asymmetric Key Cryptography) ได้รับการนำเสนอโดย Neal Koblitz และ Victor S. Miller ในปี 1985 โดยอัลกอริทึมการเข้ารหัส ECC นี้ได้รับการพัฒนาจากสมการของเส้นโค้งของวงรี $y^2 = x^3 + ax + b$ ดังแสดงในภาพที่ 1

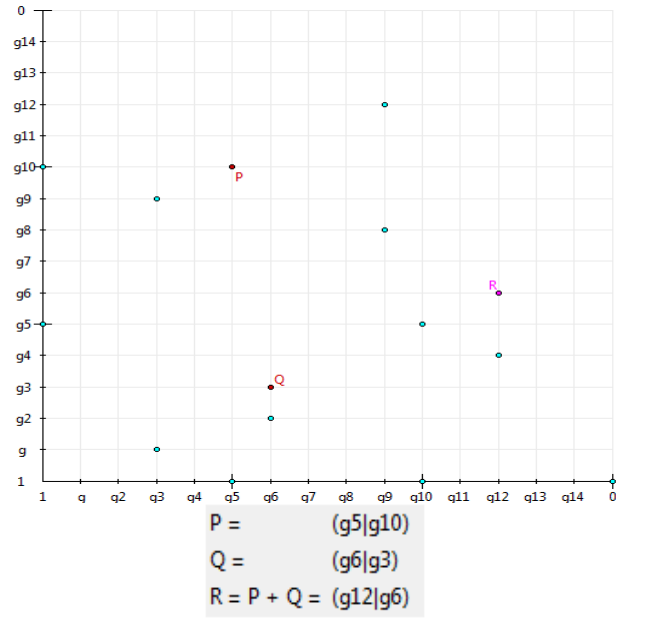


ภาพที่ 1 กราฟแสดงความสัมพันธ์ของสมการ Elliptic Curves

ECC มีข้อดีที่เหนือกว่า RSA (Rivest Shamir Adleman) คือจะใช้ Key ที่สั้นกว่าแต่สามารถให้ความปลอดภัยเท่ากับ RSA [6] ถ้าใช้ Key มีความยาวเท่ากัน ECC จะมีความปลอดภัยสูงกว่านั่นคือ หากต้องการโจมตีแบบ Brute-Force จะใช้เวลามากกว่า RSA เนื่องจาก ECC ใช้ Key ที่มีขนาดเล็กกว่า RSA มาก และมีความสามารถในการคำนวณที่รวดเร็ว ใช้พลังงานต่ำและใช้หน่วยความจำน้อย ดังนั้น ECC จึงเหมาะสำหรับการใช้งานในอุปกรณ์เคลื่อนที่ขนาดเล็ก เช่น โทรศัพท์มือถือ Pocket PC และ PDA เป็นต้น [7]

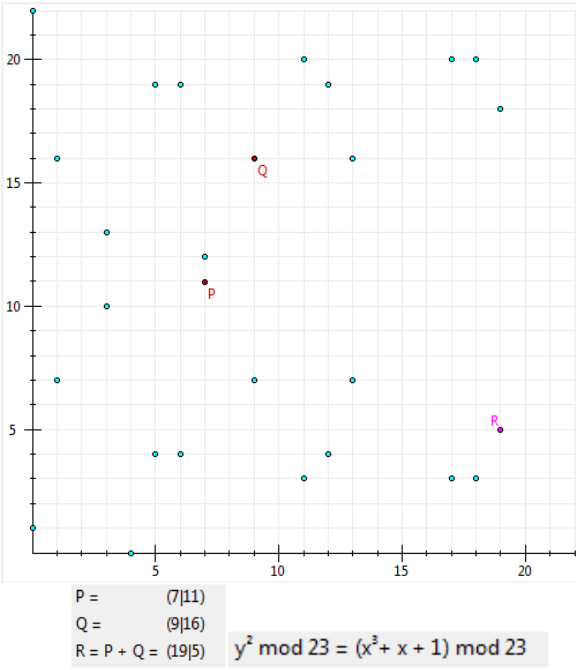
2.3.1 Elliptic Curve Group Over Finite Fields

- **Over GF(2^m) (Polynomials)** เป็นกราฟที่ได้จากสมการ $y^2 + xy = x^3 + ax + b$ โดยที่ $b \neq 0$ กำหนดให้ $P = (x_1, y_1), Q = (x_2, y_2)$ คือจุดบนกราฟของสมการ ดังแสดงในตัวอย่างในภาพที่ 2



ภาพที่ 2 กราฟ Elliptic Curve Over GF(2^m) [13]

● **Over GF(p) (Prime Number)** เป็นกราฟที่ได้จากสมการ $y^2 \bmod p = (x^3 + ax + b) \bmod p$ โดยที่ $4a^3 + 27b^2 \bmod p \neq 0$ ภาพที่ 3 แสดงตัวอย่างของกราฟ Elliptic Curve Over GF(23)



ภาพที่ 3 กราฟ Elliptic Curve Over GF(23) [13]

2.3.2 กฎการบวกระหว่างจุดบนกราฟ GF(p) [8]

กำหนดให้ $P = (x_1, y_1)$ และ $Q = (x_2, y_2)$ คือจุดบนกราฟของสมการ $P + Q = R = (x_3, y_3)$

โดยที่ $x_3 = \lambda^2 - x_1 - x_2$, $y_3 = \lambda(x_1 - x_3) - y_1$

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} & if \quad P \neq Q \\ \dfrac{2x_1^2 + a}{2y_1} & if \quad P = Q \end{cases}$$

2.3.3 กฎการลบระหว่างจุดบนกราฟ GF(p) [9]

กำหนดให้ $P = (x_1, y_1)$ และ $Q = (x_2, y_2)$ คือจุดบนกราฟของสมการ $P - Q = P + (-Q)$

โดยที่ $-Q = (x_2, y_2 \bmod p)$

2.3.4 กฎการคูณค่าคงที่กับจุดบนกราฟ GF(p) [8]

กำหนดให้ $P = (x_1, y_1)$ และ $Q = (x_2, y_2)$ คือจุดบนกราฟของสมการ ถ้า P = Q จะได้

$$P + P = 2P = R = (x_3, y_3)$$

เมื่อ k คือจำนวนเต็มบวกใดๆจะได้

$$Q = kP = \underbrace{P + P + \cdots + P}_{k}$$

เช่น ถ้า $k = 9$, $Q = kP = 9P = 2(2(2P)) + P$

2.3.5 การเข้ารหัสและถอดรหัส (ECC Encryption and Decryption) [10], [11]

การเข้ารหัส (Encryption) ผู้ส่ง A นำข้อความ P_m มาคำนวณหาค่าข้อความที่เข้ารหัสลับ C_m แล้วส่งไปหาผู้รับ B ซึ่ง $C_m = \{kG, P_m + kP_B\}$ โดยที่

G คือจุดที่ได้จากการ Generate บน Elliptic Curve

k คือตัวเลขสุ่มจำนวนเต็มบวกที่เลือกโดย A

P_B คือ Public Key ของ B ซึ่ง $P_B = n_B \times G$

n_B คือ Private Key ของ B

การถอดรหัส (Decryption) B นำ Private Key มาคูณค่าจุดแรก และนำผลลัพธ์ไปลบออกจากค่าจุดที่สอง ดังต่อไปนี้

$$P_m + kP_B - n_B(kG) = P_m + k(n_B)G - n_B(kG) = P_m$$

ตัวอย่างการเข้ารหัสและถอดรหัส GF(p)

$E_p(a,b) = E_{23}(1,1)$ จะได้ a = 1, b = 1, p = 23 เขียนเป็นสมการได้ดังนี้ $y^2 \bmod 23 = (x^3 + x + 1) \bmod 23$

จุดบนกราฟทั้งหมด GF(23) แสดงในภาพที่ 4

(0, 1)	(6, 4)	(12, 19)
(0, 22)	(6, 19)	(13, 7)
(1, 7)	(7, 11)	(13, 16)
(1, 16)	(7, 12)	(17, 3)
(3, 10)	(9, 7)	(17, 20)
(3, 13)	(9, 16)	(18, 3)
(4, 0)	(11, 3)	(18, 20)
(5, 4)	(11, 20)	(19, 5)
(5, 19)	(12, 4)	(19, 18)

ภาพที่ 4 จุด GF23(23) ทั้งหมด

กำหนดให้

เลือกใช้จุด G = (1,7)

P_m = (9,7) ซึ่งแทนด้วยอักษร “M”

ผู้ส่ง A

Private Key = $n_A = 3$

Public Key = $n_A \times G$ = 3*(1,7) = (18,20)

ผู้รับ B

Private Key = $n_B = 5$

Public Key = $n_B \times G$ = 5*(1,7) = (0,1)

เมื่อ A ต้องการส่งข้อความให้ B

1. A สุ่มตัวเลขได้ k = random = 9
2. คำนวณ $C_m = \{kG, P_m + kP_B\}$

C_m = {9*(1,7), (9,7) + 9*(0,1)}

= {(9,16), (9,7) + (19,18)}

= {(9,16), (13,7)}

A ส่ง C_m = {(9,16), (13,7)}

เมื่อ B รับข้อความจากสมการ

$$P_m + kP_B - n_B(kG) = P_m + k(n_B)G - n_B(kG) = P_m$$

P_m = (13,7) - 5*(9,16)

= (13,7) - (19,18)

= (13,7) + (19,-18) (จากกฎการลบ)

= (13,7) + (19,5) (5 = -18 mod 23)

= (9,7)

B ได้รับจุด (9,7) แทนด้วยอักษร “M”

3. การออกแบบการส่ง SMS ที่มีความปลอดภัยและมีประสิทธิภาพ

การออกแบบวิธีการส่ง SMS ที่มีความปลอดภัยและมีประสิทธิภาพ ประกอบด้วย 2 ส่วนด้วยกัน ดังต่อไปนี้

3.1 การออกแบบฐานข้อมูล

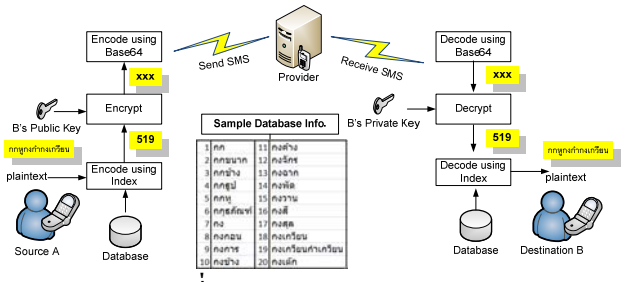
ขั้นตอนแรกเป็นการนำคำศัพท์ที่ได้ทั้งภาษาอังกฤษและภาษาไทยมาสร้างดัชนี (Index) โดยปกติตัวเลขจะเก็บในรูปของตัวเลข 32 บิต หรือ 4 ไบต์ แต่ในที่นี้จะเก็บค่าดัชนีด้วยตัวเลข 24 บิต หรือ 3 ไบต์ ซึ่งข้อมูลในฐานข้อมูลจะใช้ตัวเลขไม่เกิน 24 บิต และดัชนีตัวสุดท้ายคือหมายเลข 101910 ดังแสดงในภาพที่ 5

Index	Word	24 bit	32 bit
0	0	00000000.00000000.00000000	00000000.00000000.00000000.00000000
1	1	00000000.00000000.00000001	00000000.00000000.00000000.00000001
2	2	00000000.00000000.00000010	00000000.00000000.00000000.00000010
...	...	...	...
62117	zygotic	00000000.11110010.10100101	00000000.00000000.11110010.10100101
62118	zymurgy	00000000.11110010.10100110	00000000.00000000.11110010.10100110
62119	ก	00000000.11110010.10100111	00000000.00000000.11110010.10100111
62120	ก็	00000000.11110010.10101000	00000000.00000000.11110010.10101000
...	...	...	...
101909	ไฮสปีด	00000001.10001110.00010101	00000000.00000001.10001110.00010101
101910	ไฮเอนด์	00000001.10001110.00010110	00000000.00000001.10001110.00010110

ภาพที่ 5 Index ของการวิเคราะห์ระบบฐานข้อมูล[3][4]

3.2 การออกแบบการทำงานของระบบ

ภาพที่ 6 แสดงการทำงานภาพรวมของระบบ

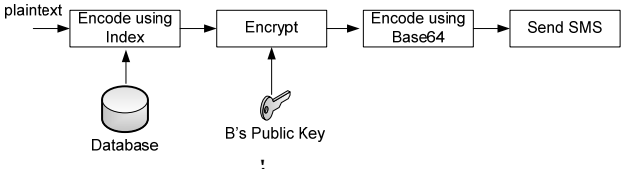


ภาพที่ 6 การทำงานภาพรวมของระบบ

การรับและส่งข้อความมีขั้นตอนดังนี้

ฝั่งส่ง การส่งข้อความแสดงดังภาพที่ 7

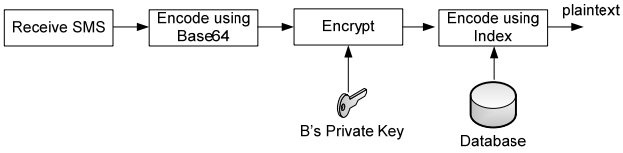
1. ผู้ส่งพิมพ์ข้อความหรือเลือกข้อความจากฐานข้อมูล
2. นำข้อความที่ได้แปลงเป็นรหัสตัวเลข (Encode Message) จากฐานข้อมูลที่ได้ถูกออกแบบไว้
3. นำข้อความที่ถูกแปลงเป็นตัวเลขมาเข้ารหัสด้วย Public Key ของฝั่งรับโดยใช้ ECC
4. นำผลลัพธ์ที่ได้จากข้อ 3 มาแปลงเป็น Base64 Encoding ส่ง SMS ให้ฝั่งรับ



ภาพที่ 7 การส่งข้อความ

ฝั่งรับ การอ่านข้อความแสดงดังภาพที่ 8

1. นำข้อความที่ได้มาถอดรหัสโดยใช้ Base64 Decoding
2. นำผลลัพธ์ที่ได้จากข้อ 1 มาถอดรหัสด้วย private key ของฝั่งรับโดยใช้ ECC
3. นำข้อความที่ได้จากการถอดรหัสจะได้เป็นรหัสตัวเลข นำมาถอดรหัสตัวเลข (Decode Message)
4. ผู้รับได้ข้อความที่ต้องการ



ภาพที่ 8 การอ่านข้อความ

4. การพัฒนาระบบ

ในการพัฒนาระบบ ได้เลือกใช้ ECC SEC2 Standard [12] ซึ่งเป็นมาตรฐานการเข้ารหัสที่มีประสิทธิภาพที่กำหนดโดยกลุ่ม The Standards for Efficient Cryptography Group (SECG) ก่อตั้งขึ้นในปี 1998 version ปัจจุบัน SEC3, draft version 0.5 ส่วน key ที่ใช้คือ secp160r1 จะมีความยาว 160 บิต ซึ่งมีประสิทธิภาพเท่ากับ RSA 1024 บิต แต่ความยาวของ key สั้นกว่า

คุณสมบัติของ secp160r1 [12]

a = 1461501637330902918203684832716283019653785059324

b = 163235791306168110546604919403271579530548345413

p = 1461501637330902918203684832716283019653785059327

G(x,y) = (425826231723888350446541592701409065913635568770, 203520114162904107873991457957346892027982641970)

ตัวอย่าง P, Q ที่ Generate ได้ [13]

P(x,y) = (1003252683990840279727753451475179984407358396999, 571206832083043554837947831606568067794209538782)

Q(x,y) = (883524120876639534704801724854422205545288316044, 888900324514296331591103235077175145553071214967)

P+Q = R =(405944969580074619918673191645050138460318078016, 110364466032477370623978385686291437487092645214 9)

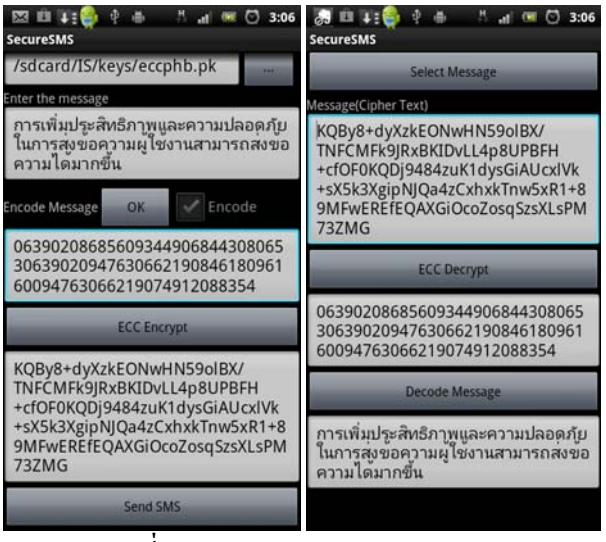
ได้มีการพัฒนาโปรแกรมประยุกต์เพื่อการใช้งานบนโทรศัพท์มือที่ใช้ระบบปฏิบัติการแอนดรอยด์ ใช้ Library ECC [14] ภาพที่ 9 แสดง Method ในภาษาจาวาที่ใช้ในการแปลงตัวเลขเป็นไบต์และไบต์เป็นตัวเลข ซึ่งใช้การเลื่อนบิต

```
public static final byte[] intTo3ByteArray(int value) {
    return new byte[] {
            (byte)(value >>> 16),
            (byte)(value >>> 8),
            (byte)value};
}

public static final int byte3ArrayToInt(byte [] b) {
    return (b[0] << 16)
    + ((b[1] & 0xFF) << 8)
    + ((b[2] & 0xFF));
}
```

ภาพที่ 9 Method ที่ใช้ในการแปลงตัวเลขเป็นไบต์และไบต์เป็นตัวเลข

สำหรับการส่งและอ่านข้อความ SMS ของโปรแกรมประยุกต์ได้พัฒนาทำขึ้นมา ได้มีการทดสอบการทำงาน ภาพหน้าจอในภาพที่ 10 แสดงขั้นตอนการทำงาน



ภาพที่ 10 การส่งและอ่านข้อความ Secure SMS

5. ผลการดำเนินงาน

คำศัพท์ทั้งภาษาอังกฤษและภาษาไทยได้ถูกนำมาสร้างดัชนี โดยแต่ละดัชนีจะใช้พื้นที่เก็บขนาด 24 บิต ซึ่งโดยเฉลี่ยแล้ว จะแทนค่าตัวอักษรได้ 7.9 ตัว จะส่งผลให้จำนวนตัวอักษรที่จะส่งลดลง ตัวอย่างเช่น ย่อหน้าแรกของบทคัดย่อมีตัวอักษรทั้งไทยและละตินจำนวน 470 ตัว หรือ 470 ไบต์ ซึ่งจะต้องส่งด้วย SMS จำนวน 7 ข้อความ แต่ถ้าแปลงเป็นตัวเลขจะใช้พื้นที่เก็บ 348 ไบต์ การแปลงตัวเลขขนาด 3 ไบต์ให้อยู่ในรูปของ Base64 จะเปลี่ยนเป็นตัวอักษรละตินจำนวน 464 ตัว แต่จะสามารถส่งได้ 160 ตัวอักษรต่อหนึ่งข้อความ ซึ่งจะใช้ SMS เพียง 3 ข้อความ จะเห็นได้ว่าการแปลงข้อความโดยใช้ดัชนี สามารถเพิ่มประสิทธิภาพการส่ง SMS

การเข้ารหัสลับข้อความ SMS โดยใช้มาตรฐาน ECC SEC2 โดยใช้ Key ที่มีความยาว 160 บิต จะช่วยเสริมสร้างความปลอดภัยในการส่ง SMS ซึ่งผู้ส่งจะต้องติดตั้ง Public

Key ของผู้รับบนเครื่องโทรศัพท์ก่อนที่สามารถเข้ารหัสลับข้อมูลได้

6. สรุปผล

ด้านประสิทธิภาพในการส่งข้อความ ผู้ใช้งานสามารถส่งข้อความได้มากขึ้น ส่งผลให้ผู้ใช้งานประหยัดค่าใช้จ่ายในการส่งข้อความ ประสิทธิภาพในการส่งข้อความนั้นขึ้นอยู่กับขนาดของคำศัพท์ที่ถูกทำเป็น Index ยิ่งคำศัพท์มีขนาดของความยาวมากก็จะสามารถส่งข้อความได้มากขึ้น อีกปัจจัยหนึ่งที่มีผลคือ จะต้องสูญเสีย Overhead ในกระบวนการเข้ารหัสทั้ง ECC Encryption และ Base64Encoding

ด้านความปลอดภัย เนื่องจาก ECC เป็นอัลกอริทึมที่ใช้ในการเข้ารหัสแบบอสมมาตร (Asymmetric Key Cryptography) จะใช้ Key สองอันในการเข้าและถอดรหัส โดยหากเข้ารหัสด้วย Key อันหนึ่ง จะต้องทำการถอดรหัสด้วย Key อีกอันหนึ่งที่เหลือ อีกทั้งขนาด Key ที่ใช้มีขนาดเล็กกว่าแต่มีความปลอดภัยสูงเทียบเท่ากับ RSA ที่มีขนาด Key เท่ากัน

3. เอกสารอ้างอิง

[1] บริษัท ทรีโอ อินเกทธเทล จำกัด, "ประวัติความเป็นมา SMS", 2010, http://www.trio4u.com/index.php?lay=show&ac=article&Id=538693173

[2] P. Gupta, “Short Message Service: What, How and Where?” http://www.wirelessdevnet.com/channels/sms/features/sms.html

[3] ศูนย์สารสนเทศ ราชบัณฑิตยสถาน, “พจนานุกรม ฉบับราชบัณฑิตยสถาน พ.ศ. ๒๕๔๒”, http://rirs3.royin.go.th/dictionary.asp

[4] phoenix-sg, "en_us.dic", Dec 01, 2010, http://hg.phoenixviewer.com/phoenix-sg/diff/98fa31757412/indra/newview/app_settings/dictionaries/en_us.dic

[5] “เรื่องวุ่น ๆ กับตัวหนังสือ - ตอนที่ 3 - Character Encoding” 26 May 2009, http://www.thaigamedevx.com/features/63

[6] W. Stallings, Cryptography and Network Security, 5th ed. Upper Saddle River, NJ: Prentice Hall, 2006, pp. 332-347.

[7] Beekie39, “วิทยาการรหัสลับ (Cryptography)”, August 2010, http://beekie39.blogspot.com/2010/08/cryptography.html

[8] Arnaud Tisserand CNRS, IRISA laboratory, CAIRN research team, "Introduction to Elliptic Curve Cryptography (ECC) Hardware Implementation", Nov. 2009, http://www.irisa.fr/prive/Arnaud.Tisserand/docs/slides-semcairn09-ecc-4p.pdf

[9] Anoop MS, "Elliptic Curve Cryptography", http://www.tataelxsi.com/whitepapers/ECC_Tut_v1_0.pdf?pdf_id=public_key_TEL.pdf

[10] Padma Bh, D.Chandravathi, P.Prapoorna Roja, “Encoding And Decoding of a Message in the Implementation of Elliptic Curve Cryptography using Koblitz’s Method” 2010, http://www.enggjournals.com/ijcse/doc/IJCSE10-02-05-08.pdf

[11] Megha Kolhekar, Anita Jadhav, "Implementation of Elliptic Curve Cryptography on text and image", July 2, 2011, http://www.ijecbs.com/July2011/14.pdf

[12] Certicom Research, "SEC 2: Recommended Elliptic Curve Domain Parameters" September 20, 2000, http://www.secg.org/download/aid-386/sec2_final.pdf

[13] Cryotool , August 2010, http://www.cryptool.org/en/

[14] Troels Bjerre Sørensen, Thomas Kragh, Mikkel Kamstrup Erlandsen, “The Java Elliptic Curve Cryptography project (JECC) borzoi 1.02 – an open source Elliptic Curve Cryptography Library”, http://sourceforge.net/projects/jecc/