Tunghai Science Vol. 8: 139–158
July, 2006

橢圓曲線版的動態秘密分享系統

林 靜 慧* 沈 淵 源*

Abstract

本論文主要在探討橢圓曲線版的動態秘密分享系統。首先, 我們先對密碼學的起源做一簡短的介紹, 接著介紹如何分享秘密以及動態秘密分享系統, 再介紹橢圓曲線密碼系統。最後, 我們推出一個橢圓曲線版的動態秘密分享系統並對其安全性作一簡單的分析。

1 前言

數千年來, 不論是君王或將領, 都需要一套有效率的通訊系統來治理國家、指揮軍隊。他們當然也深知萬一訊息落入不當人士裡, 讓敵國窺知機密, 或讓反對勢力獲取關鍵資訊時, 所會產生的嚴重後果。因此, 在隨後的好幾個世紀裡, 不斷的有一些更高明的加密技術產生。

在19世紀 Kerckhoffs 的演說中, 已經隱約透露出現代密碼學的原理:「沒有秘密算法, 一切盡在密鑰中。」可是, 當時的加密體系仍然缺少數學背景, 因而也缺少測量或評價這些體系抵抗攻擊的能力。要是有人能最終達到密碼學的終極目標, 即找到百分之百無條件的安全體系, 那該有多好。但事實上, 是否存在這樣的密碼系統呢? 答案是肯定的, 的確存在有無法破解的密碼系統, 如單次鑰匙簿密碼 (One-time Pad); 而且也存在有密碼系統如 RSA 者, 只要執行妥當, 在你有生之年也是無法破解的, 即使破解了, 也需耗費大量的時間及物資。

在現今這個網路通訊逐漸發達的社會上, 有很多商業上的機密或私人的通訊, 都會藉由網路來傳遞訊息。因此網路上所流傳的資料的安全性就顯得格外重要, 因此衍生出許多的密碼技術, 如公開金鑰密碼系統、數位簽署、秘密分享、橢圓曲線密碼系統等等, 希望藉由資訊技術達到保護資訊安全的目的, 所以密碼學已成爲資訊科學中重要的一環。

1979年,Shamir 以及 Blakley 首先提出 (t, w) 門檻策略秘密分享 (threshold secret sharing) 方法, 該方法可以簡單且有效地解決一群人分享一個秘密的問題。(t, w) 門檻策略秘密分享允許 w 個參與者

*東海大學應用數學研究所

(particpant) 共享一個秘密, 一旦有多於 (或等於) t 個參與者的同意之下, 秘密才可重建。(t,w) 門檻策略秘密分享的特色如下：

1. 分發者 (U_D) 將秘密分割成 w 個子秘密 (share), 並利用安全通道 (secure channel) 分送給 w 個參與者。

2. 只要有 t 個參與者拿出子秘密, 即可重建該秘密。

3. 少於 t 個子秘密將無法得到任何有關該秘密的資訊。

然而在傳統的 (t,w) 門檻秘密分享方法中存在有三個缺點。首先, 每個參與者所得到的子秘密只能用來重建一個秘密, 一旦秘密重建後, 子秘密即失去效用。其次是, 無法解決分發者在子秘密產生階段以及參與者在秘密重建階段時可能引發的欺騙問題。再者, 子秘密傳送過程必須通過安全通道, 此時容易招致攻擊者的竊取。

本研究主要的論述, 是利用動態秘密分享方法來解決包括分發者等所有參與者可能有的欺騙行爲。這裡所提到的動態, 是指可以利用相同的子秘密便可同時分享不同的秘密, 這可減少許多計算量。接著我們利用橢圓曲線密碼系統, 將其改良, 使得子秘密在傳送過程可以不經過安全通道, 便擁有更高的安全性。最後我們探討並分析此秘密分享系統之優缺點。

2 秘密分享

在這章節我們將介紹最受歡迎的幾個秘密分享系統, 以及其推導之過程。

2.1 分散秘密

假設你擁有一個秘密, 可用一個整數M來表示。你想把這個整數拆開成兩半給三毛與四郎, 使得他們當中的任何一人都無法從已知的那一部分得到完整的秘密M。那該如何解決此問題呢? 其實答案就在問題中。首先, 選取一隨機整數 r 交給三毛, 而後再將整數 $M-r$ 交給四郎。如果要重建此秘密 M , 三毛與四郎兩人僅需聚在一起, 然後亮出各自擁有的數並相加即可。

但有一技術層面問題必先克服的, 那就是我們不可能選取隨機整數使得所有的整數具有同等的可能性 (無限多個具有相同概率的數加在一起不可能會等於 1)。所以我們選取一個整數 n , n 會大過所有可能出現的訊息 M , 並將 M 與 r 看做模 n 下的數。如此一來, 只要在模 n 數系下的每個整數之概率均爲 $\frac{1}{n}$, 那就毫無問題的可在 模 n 數系下隨機選取一整數 r 。

同理, 如果今天我們想將一個秘密 M 分散給 w 個人, 那我們就在模 n 下選取 $w-1$ 個隨機整數 $r_1, r_2, \cdots, r_{w-1} \pmod n$ 並一一交給其中的 $w-1$ 個人, 剩下的那個人則交給整數 $M - \sum_{k=1}^{w-1} r_k \pmod n$。

2.2 門檻法 (Threshold Schemes)

在上面我們所介紹的是將一秘密 M 分散給 w 個人。但那方法必須所有 w 個人都參與, 才能解出 M , 比較不實用。現在我們介紹僅需部分的人參與即可重建整個訊息的方法。

定義: 令 $t \leq w$ 爲兩正整數。一個(t,w)-門檻法乃是將訊息 M 分享給 w 位參與者的一種方法: 在此方法中, 只需其中任何 t 位參與者就可重建訊息 M , 若少於 t 位便無法重建 M 。

2.2.1 Shamir門檻法

在一九七九年由沙密爾 (Shamir) 所提出, 所以稱之爲沙密爾門檻法或拉格蘭茲內插法 (Lagrange Interpolation Scheme)。其演算法如下:

1. 選取一質數 p , 大於所有的訊息也大於所有參與者的人數。此處所有的計算都在模 p 的數系中進行。若用合成數代替, 那下面所得到的矩陣有可能沒有乘法反元素。

2. 將訊息 M 表示成模 p 數系中的一個數, 而我們要將訊息 M 分享給 w 位參與者, 但只需其中的 t 位便可解出此一訊息。

3. 隨機選取 $t-1$ 個整數 s_i $(i = 1,2,\cdots,t-1)$ 作爲多項式 $s(x)$ 中 x^i 項的係數, 然後將 M 放在此多項式的常數項位置。所以我們得到一多項式

$$s(x) = M + s_1 x + s_2 x^2 + s_3 x^3 + \cdots + s_{t-1} x^{t-1} \pmod{p}$$

   其常數項就是原訊息 M , 亦即 $s(0) \equiv M \pmod{p}$。

4. 對這 w 位參與者, 我們先選取相異的整數 $x_1, x_2, \cdots, x_w \pmod{p}$, 然後再交給每個人一秘密數對 (x_i, y_i), 其中$y_i \equiv s(x_i) \pmod{p}$。例如:$1,2,3,\cdots,w$乃是這些 x 值既合理而又自然的一個選擇。所以我們就將數對$(1,s(1)),(2,s(2)),\cdots,(w,s(w))$交給這 w 個人, 一人一組。質數 p 是公開的, 但多項式 $s(x)$ 則保密。

5. 假設現在有 t 個人聚在一起分享彼此間的數對。 爲了簡化符號, 我們可假設這些數對爲$(x_1,y_1),(x_2,y_2),\cdots,(x_t,y_t)$。由此, 我們要尋回原訊息 M 。

6. 假設有一個 $t-1$ 次的多項式 $s(x)$, 已知其中的 t 個值爲

$$y_k \equiv s(x_k) \pmod{p}$$

   我們要從這些資訊來重建這多項式, 所以對任意的 k , $1 \leq k \leq t$,我們有

$$y_k(x_k) = M + s_1 x_k^1 + s_2 x_k^2 + s_3 x_k^3 + \cdots + s_{t-1} x_k^{t-1} \pmod{p}$$

   此處 s_i $(1 \leq i \leq t-1)$ 及 M 均爲未知數。

7. 將上面的 t 個同餘式寫成矩陣形式

$$\begin{pmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{t-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{t-1} \\ 1 & x_3 & x_3^2 & \cdots & x_3^{t-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_t & x_t^2 & \cdots & x_t^{t-1} \end{pmatrix} \begin{pmatrix} M \\ s_1 \\ s_2 \\ \vdots \\ s_{t-1} \end{pmatrix} \equiv \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_t \end{pmatrix} \pmod{p}$$

此係數矩陣, 暫且稱之爲 V , 是一個 Vandermonde 矩陣。我們知道, 如果這個矩陣的行列式值在模 p 下不等於零, 則此矩陣有唯一的解。此行列式值可被證明就是

$$\text{detV}=\prod_{1\le j<k\le t}(x_k-x_j)$$

此值只有當兩個 x_i 一樣時才是0 (mod p)(此處的 p 爲質數)。所以只要 x_i 相異, 則此系統有唯一解。

8. 現在我們換一個角度來重建多項式 $s(x)$, 由此引導我們得到這個多項式的一個公式從而可推得訊息 M 的一個公式。我們的目標是重建一多項式 $s(x)$, 而其中的 t 個值爲 $(x_k,y_k), 1\le k\le t$, 很自然的 我們會想到 t 次多項式

$$u(x)=(x-x_1)(x-x_2)(x-x_3)\cdots(x-x_t)$$

對每一個 $k=1,2,3\cdots,t$, 將 $u(x)$ 除以 $x-x_k$ 得到一個 $t-1$ 次多項式

$$u_k(x)=\prod_{\substack{j=1\\ j\ne k}}^{t}(x-x_j)$$

此多項式滿足 $u_k(x_j)=0, \forall j\ne k$; 若這些 x_i 兩兩相異, 則 $u_k(x_k)\ne 0$ 。再將每一個 $u_k(x)$ 單位化, 亦即除以 $u_k(x_k)$, 稱之爲 $l_k(x)$ 。因此得到

$$l_k(x)\equiv\frac{u_k(x)}{u_k(x_k)}\equiv\prod_{\substack{j=1\\ j\ne k}}^{t}\frac{x-x_j}{x_k-x_j} \pmod{p}$$

顯而易見, 我們有

$$l_k(x_j)\equiv\begin{cases} 1, & k=j \\ 0, & k\ne j \end{cases} \pmod{p}。$$

因此我們得到拉格蘭茲內插多項式

$$L(x) = \sum_{k=1}^{t} y_k l_k(x) = \sum_{k=1}^{t} y_k \prod_{\substack{j=1 \\ j\neq k}}^{t} \frac{x - x_j}{x_k - x_j}。$$

這個多項式滿足所有的條件$L(x_j) = y_j, 1 \leq j \leq t$, 因爲

$$\begin{aligned} L(x_j) &\equiv \sum_{k=1}^{t} y_k l_k(x_j) \equiv y_j l_j(x_j) + \sum_{\substack{j=1 \\ j\neq k}}^{t} y_k l_k(x_j) \\ &\equiv y_j \cdot 1 + \sum_{\substack{j=1 \\ j\neq k}}^{t} y_k \cdot 0 \equiv y_j \pmod{p} \end{aligned}$$

因此, 透過 Vandermonde 矩陣的證明, 我們知道 $s(x)$ 爲經過這些點的唯一的 $t-1$ 次多項式, 所以得到$L(x) = s(x)$。

9. 如果要重建秘密訊息 M , 只需計算 $L(0)$ 之值。因此得到秘密訊息的公式爲

$$M \equiv \sum_{k=1}^{t} y_k \prod_{\substack{j=1 \\ j\neq k}}^{t} \frac{-x_j}{x_k - x_j} \pmod{p}。$$

例子: 假設我們有一個$(5,8)$–門檻法, 其質數 $p = 987541$ 。而其中的五份爲

$$(9853,853),(4421,4387),(6543,1234),(93293,78428),(12398,7563)$$

試求出訊息。

解:

使用拉格蘭茲內插法, 算出經過這五點之多項式在 Mathematica 中輸入指令如下: PolynomialMod[InterpolatingPolynomial[
{{9853,853},{4421,4387},{6543,1234},{93293,78428},{12398,7563}} ,x],987541]
其輸出爲
$M = 678987, s_1 = 14728, s_2 = 1651, s_3 = 574413, s_4 = 4567414$; 所以解得此一多項式爲

$$s(x) = 678987 + 14728x + 1651x^2 + 574413x^3 + 456741x^4,$$

可得訊息 $M = 678987$。

2.2.2 分析

Shamir 門檻法秘密分享有三個主要的問題:

1. 不能有效地防止分發者的欺騙, 即是分發者在分發子秘密時, 可能會給某個分享者分發假的子秘密, 使在恢復秘密的成員無法恢復出正確的秘密, 多數的文獻忽略了這一點。

2. 不能抵抗分享者的欺騙, 即是有些分享者在恢復秘密時提供假的子秘密, 而使其他的成員無法恢復正確的秘密。

3. 在恢復秘密的階段, 參與秘密恢復的成員會將自己的子秘密交付給一位合成者, 由合成者來代表所有參與者恢復共享秘密 , 大部份的文獻中, 合成者本身就是參與秘密恢復的共享者, 這就難以保證合成者在恢復秘密後不會提供假的秘密給其他參與秘密恢復的分享者。

3 動態秘密分享

在本章中, 我們提出基於離散對數問題的動態秘密分享策略; 此策略不僅檢測出包括分發者在內的所有參與者的欺騙行爲, 又可以無限次分發且還原不同的秘密, 並具有結構簡單、安全性高等優點。與已有可驗證的秘密分享方案相比較, 此驗證計算複雜度較小, 數據傳輸量小, 因此效率較高。其演算法如下:

3.1 秘密分發階段

1. U_D 是系統秘密的分發者, U_D 選取一個大質數 p 及模 p 的一個原根 g , p 大於所有的訊息也大於所有參與者的人數。U_i 是第 i 個參與者, 其身份識別碼爲 δ_i $(i = 1,2,\cdots,w)$。

2. 將訊息 k 表示成模 p 數系中的一個數, 而我們要將訊息 k 分享給 w 位參與者, 但只需其中的 t 位便可解出此一訊息。U_D 先計算 $K = g^k \pmod p$。

3. 每一個 U_i $(i = 1,2,\cdots,w)$ 隨機選一整數 x_i 爲私鑰, 算出 $X_i = g^{x_i}$ 並傳送給 U_D 。

4. U_D 隨機選取 t 個整數 a_j $(j = 0,1,2,\cdots,t-1)$ 作爲多項式 $f(x)$ 中 x^j 項的係數 。所以得到一多項式

$$f(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + \cdots + a_{t-1}x^{t-1} \pmod{p-1},$$

U_D 計算 $s = k + a_0 \pmod{p-1}$, 並算出 $A_j = g^{a_j} \pmod p$,
$j = 0,1,2,\cdots,t-1$; 以及每一個 $f(\delta_i) \pmod{p-1}$, $i = 1,2,\cdots,w$ 。

5. U_D 隨機選取整數 r , 並計算 $R = g^r \pmod p$。
對 $i = 1,\cdots,w$ 算出 $b_i = f(\delta_i) + X_i^r \pmod p$, 以及 $c_i = s + X_i^r \pmod p$。

6. U_D 將 p , g , R , K ; b_i , c_i , δ_i ($i = 1,\cdots,w$) ; A_j ($j = 0,1,\cdots,t-1$) 公開。

3.2 訊息驗證階段

1. 每一個 U_i $(i = 1,2,\cdots,w)$ 使用其私鑰 x_i 算出 $R^{x_i} \pmod p$稱之爲 D_i , 然後算出 $B_i = b_i - D_i \pmod p$ 及 $C_i = c_i - D_i \pmod p$。

2. 每一個 U_i $(i = 1, 2, \cdots, w)$ 可驗證下列等式是否成立

$$g^{C_i} = K \cdot A_0 \pmod p, \quad g^{B_i} = \prod_{j=1}^{t} {A_{j-1}}^{{\delta_i}^{j-1}} \pmod p$$

若等式成立, 說明了 U_D 沒有欺騙行爲。
若等式不成立, 則公開 U_D 的欺騙行爲。

3.2.1 證明驗證成立

1. 驗證 $g^{C_i} = K \cdot A_0 \pmod p$
證明:
因爲 $D_i = R^{x_i} = (g^r)^{x_i} = (g^{x_i})^r = {X_i}^r \pmod p$
所以 $C_i = c_i - D_i = s + {X_i}^r - {X_i}^r = s \pmod p$
又因爲 $s = k + a_0 \pmod{p-1}$
$\Rightarrow C_i = (k + a_0) + n \cdot (p-1)$
且 $K = g^k \pmod{p-1}, A_0 = g^{a_0} \pmod{p-1}$
$\therefore g^{C_i} = g^{(k + a_0) + n \cdot (p-1)} = g^k \cdot g^{a_0} \cdot g^{n \cdot (p-1)} = K \cdot A_0 \cdot (g^{p-1})^n$
$= K \cdot A_0 \pmod p$ 因此得到左式=右式, 故得證。

2. 驗證 $g^{B_i} = \prod_{j=1}^{t} {A_{j-1}}^{{\delta_i}^{j-1}} \pmod p$
證明:
因爲 $D_i = {X_i}^r \pmod p$
所以 $B_i = b_i - D_i = f(\delta_i) + {X_i}^r - {X_i}^r = f(\delta_i) \pmod p$
已知 $A_i = g^{a_i} \pmod p$
將 δ_i 代入 $f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots + a_{t-1} x^{t-1} \pmod{p-1}$
得到
$f(\delta_i) = (a_0 + a_1 \delta_i + a_2 {\delta_i}^2 + a_3 {\delta_i}^3 + \cdots + a_{t-1} {\delta_i}^{t-1}) + n \cdot (p-1)$
$\therefore$ 左式= $g^{f(\delta_i)} = g^{(a_0 + a_1 \delta_i + a_2 {\delta_i}^2 + a_3 {\delta_i}^3 + \cdots + a_{t-1} {\delta_i}^{t-1}) + n \cdot (p-1)}$
$= g^{(a_0 + a_1 \delta_i + a_2 {\delta_i}^2 + a_3 {\delta_i}^3 + \cdots + a_{t-1} {\delta_i}^{t-1})} \cdot (g^{p-1})^n$
$= g^{(a_0 + a_1 \delta_i + a_2 {\delta_i}^2 + a_3 {\delta_i}^3 + \cdots + a_{t-1} {\delta_i}^{t-1})} \pmod p$
$= g^{a_0} \cdot g^{a_1 \delta_i} \cdot g^{a_2 {\delta_i}^2} \cdot \cdots \cdot g^{a_{t-1} {\delta_i}^{t-1}}$
$= (g^{a_0}) \cdot (g^{a_1})^{\delta_i} \cdot (g^{a_2})^{{\delta_i}^2} \cdot \cdots \cdot (g^{a_{t-1}})^{{\delta_i}^{t-1}}$
$= A_0 \cdot {A_1}^{\delta_i} \cdot {A_2}^{{\delta_i}^2} \cdot \cdots \cdot {A_{t-1}}^{{\delta_i}^{t-1}}$
$= \prod_{j=1}^{t} {A_{j-1}}^{{\delta_i}^{j-1}}$ = 右式, 故得證。

3.3 秘密還原階段

1. 假設有 t 位說是 U_1 , U_2 ,$\cdots$, U_t 共同還原秘密。每個參與的成員 U_i 將各自的 B_i 傳送給合成者 U_A 。

2. 合成者 U_A 首先驗證 $g^{B_i} = \prod_{j=1}^{t} A_{j-1}{}^{\delta_i{}^{j-1}} \pmod p$, $i = 1, \cdots, t$ 是否成立。若成立, 說明分享者 U_i 沒有欺騙行爲; 若不成立, 則要求其重新發送自己手中的資訊。

3. 當 U_A 獲得一組數據 (δ_i , B_i), $i = 1, \cdots, t$, 由 Lagrange 內插多項式求出經過此 t 點的 $t-1$ 次 多項式 $L(T)$, 並將 $L(0)$ 告知參與秘密還原的分享者。

4. 分享者驗證 $A_0 = g^{L(0)} \pmod p$ 是否成立, 如果成立, 則說明 U_A 沒有欺騙行爲, 最後, 分享者 U_i 計算 $C_i - L(0)$; 就是所要求的秘密 k , 因爲 $C_i = s$,$L(0) = a_0$

4 橢圓曲線密碼系統

在1980年代的中期, 米勒 (Miller) 與寇伯立茲 (Koblitz) 將橢圓曲線引進密碼術當中, 從而設計了一套新的密碼系統。橢圓曲線密碼系統相較於傳統密碼系統的優點之一, 在於後者使用了相當大的鑰匙來保持安全性, 而前者似乎不需要如此龐大的鑰匙便能提供某種程度的安全。

4.1 橢圓曲線

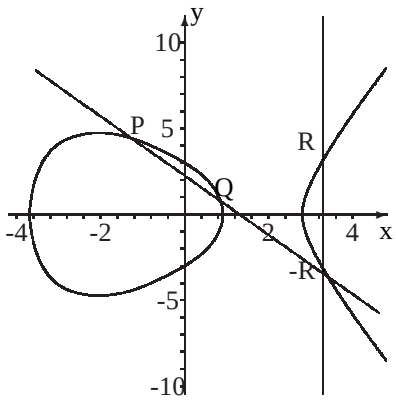
下列方程式的圖形我們稱之爲橢圓曲線

$$E : y^2 \equiv x^3 + ax + b \ \ (\bmod p)$$

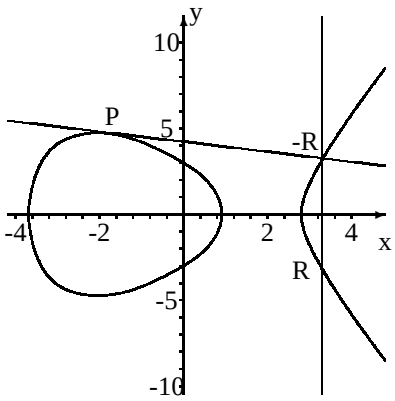
在此處 a , b 爲任何適用的集合, 如有理數、實數、複數、模 p 之下的整數或有限數體。令 $E_p(a,b) = E \cup \{\infty\}$, 其中 $4a^3 + 27b^2 \neq 0$ 。而在此處的 ∞ 稱之爲『無限遠點 (Point at infinity) 或零點 (Zero point)』。此點最簡單的一個處理方式就是將它看作在 y 軸的最上方。這可以放在投影幾何的背景下嚴密的處理, 但上面直觀的概念對我們來講已足夠了。可參考 Silverman 與 Tate 二人所合寫的書: 橢圓曲線上的有理點 (RationalPoint On EllipticCurves)[11]; 或 L.C. Washington 的 Elliptic Curves Number Theory and Cryptography [8]。若將 y 軸最下方的點看成最上方的點, 則 ∞ 也是位於 y 軸的最下方。在實數的領域下, 圖形只有兩種可能的形式, 就是右邊那個三次多項式有三個相異的實根或是一個實根而定。重根的情況又另當別論, 通常我們假設 三次多項式 $E : y^2 = x^3 + ax + b$ 沒有重根。

4.2 橢圓曲線上的加法律

現在回到我們原先的橢圓曲線 $E_p(a,b)$ 上, 來看看橢圓曲線上的點是如何相加的。



圖一 橢圓曲線上 $P+Q=R$



圖二 橢圓曲線上 $P+P=R$

1. 橢圓曲線 E 上的相異兩點 P 及 Q 相加, 可經由圖一中觀察 $P+Q=R$, 如下: 經過 P 及 Q 二點劃一直線 L 。直線 L 與橢圓曲線 E 交於 $-R$, 然後取 E 上與 $-R$ 對稱於x-軸的點 R (即 y 座標變號)。

2. 橢圓曲線 E 上的一點 P , 計算 $P+P=R$, 可經由圖二中觀察 $P+Q=R$, 如下: 經過 P 點畫橢圓曲線 E 的切線 L 。直線 L 與橢圓曲線 E 交於 $-R$, 然後取 E 上與 $-R$ 對稱於x-軸的點 R 。

3. 加法單位元素: 我們發現 $P+O=P$, 因爲經過 P 與 O 點的直線是垂直x-軸的, 所以此直線將會與 E 自然相交於 $-P$, 而再取對稱點即爲 P , 使得 $P+O=P$ 。同時亦可證明此加法具有結合性與交換性:

$$(P+Q)+R=P+(Q+R)$$

$$P+Q=Q+P$$

因此, 橢圓曲線 E 上的點在此加法運算之下則形成一個交換群, 無限遠的點 ∞ 就是橢圓曲線的加法單位元素。

4. 橢圓曲線加法運算的公式:
令 $P=(x_1,y_1), Q=(x_2,y_2)$爲橢圓曲線上的兩點, 並且 $P\neq Q$, 則 $P+Q=R=(x_3,y_3)$ 。此處

$$\begin{aligned} x_3 &\equiv m^2-x_1-x_2 \pmod p, \\ y_3 &\equiv m(x_1-x_3)-y_1 \pmod p. \end{aligned}$$

其中的m爲

$$m = \begin{cases} \dfrac{y_2-y_1}{x_2-x_1}, & \text{if P} \neq \text{Q} \\ \dfrac{3x_1^2+a}{2y_1}, & \text{if P = Q} \end{cases}$$

4.3 如何用橢圓曲線上的點來表示明文**?**

將明文信息編碼而成爲橢圓曲線上之點的問題, 並不像傳統的情況那樣簡單。這裡有一個 Koblitz 的方法。其想法如下: 令 $E : y^2 \equiv x^3 + ax + b \pmod{p}$ 爲一橢圓曲線。已經數字化的信息 m 將看成是這橢圓曲線上的某一點的 x 座標。然而 $m^3 + am + b$ 在模 p 之下是平方數的概率至少 1/2。因此我們在 m 後面接上一個位元成爲另一個數, 稱之爲 x , 藉著調整此位元直到我們得到一個 x 使得 $x^3 + ax + b$ 在模 p 之下爲平方數。

更明確地說, 令 K 爲一個大整數使得將信息編碼成爲橢圓曲線上的點時, 其失敗率爲 $1/2^K$ 是可接受的。假設 m 滿足 $(m+1)K < p$, 將此信息 m 表示成一個形如 $x = mK + j$ 的數, 此數 $0 \le j \le K$。對 $j = 0, 1, \cdots, K-1$, 計算 $x^3 + ax + b$ 並計算在模 p之下的平方根。若有一平方根 y, 則取 $P_m = (x, y)$, 否則將 j 增加 1 形成新的 x, 然後重複上述的步驟。如此這般地, 直等到找著了一個平方根或是 $j = K$。如果 j 總是等於 K, 那麼對這個信息而言, 我們的任務就無法達成。因爲 $x^3 + ax + b$ 大約有一半的時間是一個平方數, 所以我們大約有 $1/2^K$ 失敗的機會。

由點 $P_m = (x, y)$, 如何回復到原信息呢? 僅需計算 $\frac{x}{K}$, 取其整數部分即可。所以 $m = [\frac{x}{K}]$, 此處 $[\frac{x}{K}]$ 爲高斯符號, 就是小於或等於 $\frac{x}{K}$ 的最大整數。

例題**:** 令 $p = 179$ 且假設我們的橢圓曲線爲 $y^2 = x^3 + 2x + 7$。若可以接受 $1/2^{10}$ 的失敗率, 則取 $K = 10$。因爲我們要求 $(m+1)K < 179$, 故 $0 \le m \le 16$。假設我們的信息爲 $m = 5$。考慮形如 $mK + j = 50 + j$ 的 m 值, 可能的選擇爲 $50, 51, \cdots, 59$。對 $x = 51$, 我們得到

$$x^3 + 2x + 7 \equiv 121 \pmod{179}\ ,\ 11^2 \equiv 121 \pmod{179}$$

因此我們用點 $P_m = (51, 11)$ 來表示信息 $m = 5$。因此信息 m 可還原如下:$m = [\frac{51}{11}] = 5$。

4.4 橢圓曲線加密系統

在現有的數種橢圓曲線的加/解密方法中, 我們只看其中最簡易的一種方法。首先系統將要送的明文 m 編碼成橢圓曲線上的點 P_m。點 P_m 會被加密成密文, 並且稍後會被解碼。在這裡要注意一點, 我們不能單純的將訊息編碼成某個點的x或y座標, 因爲並不是所有的這類座標都會落在$E \pmod{p}$ 上。在一個鑰匙交換系統中, 其加/解密系統需要兩個參數,α 和橢圓曲線$E \pmod{p}$。

首先, 使用者 A 選擇一私密鑰匙 a_A , 然後在產生一公開鑰匙 $\beta_A = a_A\alpha$。同樣的使用者 B 也選擇一私密鑰匙 a_B , 然後在產生一公開鑰匙 $\beta_B = a_B\alpha$。

爲了將訊息 P_m 加密後傳送給 B , A 選擇一隨機整數 k , 並且產生一個由兩點所組成密文 C_m。

$$C_m \quad = \quad \{\, k\alpha\ ,\ P_m + k\beta_B\}$$

在這 A 用的是 B 的公開鑰匙 β_B。爲了解開密文, B 用自己的私密鑰匙乘上第一點, 再用第二點減去得到的結果, 可得

$$P_m + k\beta_B - a_B(k\alpha) \quad = \quad P_m + k(a_B\alpha) - a_B(k\alpha) = P_m$$

A 藉由加上 $k\beta_B$ 來隱藏訊息 P_m。除了 A 之外沒人知道 k 值, 所以即使 β_B 是公開的鑰匙, 也沒人能移除 或隱藏用的 $k\beta_B$。

4.5 例子

考慮在模179之下的橢圓曲線

$$E : y^2 \equiv x^3 + 2x + 7 \pmod{179}$$

以及橢圓曲線上的一點 $\alpha = (111,11)$, 在將此點及橢圓曲線 $E_{179}(2,7)$ 公開。
使用者 A 選取一私密鑰匙 $a_A = 12$ ，然後 A 產生一公開鑰匙 $\beta_A \equiv 12 \cdot (111,11) \equiv (111,168) \pmod{179}$ 。使用者 B 同樣選取一私密鑰匙 $a_B = 9$ ，並計算其公開鑰匙 $\beta_B \equiv 9 \cdot (111,11) \equiv (20,23) \pmod{179}$ 。

若 A 要將訊息 $m = 5$ 加密後傳給 B , 其進行步驟如下:

1. 欲傳送訊息 $m = 5$ 加密後傳送給B, 我們先將 m 轉換橢圓曲線上的點 $P_m = (51,11)$。
2. 選取隨機整數 $k = 11$ 。
3. 計算

$$\begin{aligned} y_1 &\equiv k\alpha \pmod{p} \\ &\equiv 11 \cdot (111,11) \pmod{179} \\ &\equiv (152,26) \end{aligned}$$

計算

$$\begin{aligned} y_2 &\equiv P_m + k\beta \pmod{p} \\ &\equiv (51,11) + 11 \cdot (20,23) \pmod{179} \\ &\equiv (51,11) + (164,19) \\ &\equiv (156,18) \end{aligned}$$

因此 A 所產生的密文 $C_m = \{(152,26),(156,18)\}$ 。
B 可由 A 所產生的密文 C_m 來加以解密:
計算

$$\begin{aligned} y_2 - a_B y_1 &\equiv P_m + k\beta_B - a_B(k\alpha) \pmod{p} \\ &\equiv (51,11) + 11 \cdot (20,23) - 9 \cdot [11 \cdot (111,11)] \pmod{179} \\ &\equiv (51,11) + (164,19) - 9 \cdot (152,26) \\ &\equiv (51,11) + (164,19) - (164,19) \\ &\equiv (51,11) \\ &\equiv P_m \end{aligned}$$

因此, A 藉由加上 $k\beta_B$ 來隱藏訊息 P_m 。除了 A 之外沒人知道 k 的值, 所以即使 β_B 是公開的鑰匙, 也沒人能移除隱藏用的 $k\beta_B$

5 橢圓曲線版之動態秘密分享系統

接下來我們介紹橢圓曲線版之動態秘密分享系統。

5.1 秘密分發階段

1. U_D 是系統秘密的分發者, U_D 選取一個大質數 p 及模 p 的一個原根 g , p 大於所有的訊息也大於所有參與者的人數。U_i 是第 i 個參與者, 其身份識別碼爲 δ_i $(i = 1,\cdots,w)$。

2. 將訊息 k 表示成模 p 數系中的一個數, 而我們要將訊息 k 分享給 w 位參與者, 但只需其中的 t 位便可解出此一訊息。U_D 先計算 $K = g^k \pmod p$。

3. U_D 隨機選取 t 個整數 a_j $(j = 0,1,2,\cdots,t-1)$ 作爲多項式 $f(x)$ 中 x^j 項的係數 。所以得到一多項式

$$f(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + \cdots + a_{t-1}x^{t-1} \pmod{p-1},$$

U_D 計算 $s = k + a_0 \pmod{p-1}$, 並算出每一個 $A_j = g^{a_j} \pmod p$, $j = 0,1,2,\cdots,t-1$; 以及每一個 $f(\delta_i) \pmod{p-1}$, $i = 1,2,\cdots,w$。

4. U_D 選取一橢圓曲線 $E_p(a,b)$, 並公開之。每一個 U_i $(i = 1,2,\cdots,w)$ 在 $E_p(a,b)$ 上選取一點 α_i , 再隨機選取一整數 x_i 爲私鑰, 然後計算 $\beta_i = x_i\alpha_i$ 並將 α_i , β_i 公開, 但 x_i 保持私密。

5. U_D 將其 $f(\delta_i)$ 轉換成 $E_p(a,b)$ 上的一點 P_i , 並將 s 轉換成 $E_p(a,b)$ 上的一點 Q 。選取一隨機整數 r , 並算出 $z_{1i} = r\alpha_i$, $z_{2i} = P_i + r\beta_i$ 與 $z_{3i} = Q + r\beta_i$, 將 (z_{1i}, z_{2i}, z_{3i}) 傳給 U_i , $i = 1,\cdots,w$。

6. U_D 將 p , g , K , z_{1i} , z_{2i} , z_{3i} , δ_i ($i = 1,\cdots,w$);
   A_j ($j = 0,1,\ \cdots,\ t-1$) 公開。

5.2 訊息驗證階段

1. 每一個 U_i $(i = 1,2,\cdots,w)$ 使用私鑰 x_i 計算 $b_i = z_{2i} - x_i\, z_{1i}$ 及 $c_i = z_{3i} - x_i\, z_{1i}$, 再將 b_i 及 c_i 回復其原值分別爲 B_i 及 C_i , (此 b_i 即 P_i , c_i 即 Q , 因而 B_i 就是 $f(\delta_i)$, C_i 就是 s)。

2. 每一個 U_i $(i = 1,2,\cdots,w)$ 可驗證下列等式是否成立

$$g^{C_i} = K \cdot A_0 \pmod p \text{ , } g^{B_i} = \prod_{j=1}^{t} {A_{j-1}}^{{\delta_i}^{j-1}} \pmod p$$

若等式成立, 說明了 U_D 沒有欺騙行爲; 若等式不成立, 則公開 U_D 的欺騙行爲。

5.3 秘密還原階段

假設有 t 位說是 U_1 , U_2 ,$\cdots$, U_t 共同還原秘密。每個參與的成員 U_i 將 b_i 加密後傳送給合成者, 再由合成者將其秘密還原。

1. 每一個 U_i $(i=1,2,\cdots,t)$ 用合成者 U_A 的 α_A , β_A , 以及其手上的 x_i , b_i 算出 $e_{1i} = x_i\alpha_A$ 以及 $e_{2i} = b_i + x_i\beta_A$, 再將 (e_{1i}, e_{2i}) 傳給 U_A 。

2. 合成者 U_A 得到一組 (e_{1i}, e_{2i}), $i = 1$, $\cdots$, t 之後, 使用自己的私鑰 x_A 解出 $e_{2i} - x_A\, e_{1i}$, 再回復其原值 T_i $(i = 1,\cdots, t)$ 。

3. U_A 首先驗證

$$g^{T_i} = \prod_{j=1}^{t} A_{j-1}{}^{\delta_i{}^{j-1}} \pmod p,\ i = 1\ ,\ \cdots\ ,\ t$$

(因爲 T_i 就是前面的 B_i , $i=1,2,\cdots,t$) 是否成立。若成立, 說明分享者 U_i 沒有欺騙行爲; 若不成立, 則要求其重新發送自己手中的資訊。

4. 當 U_A 算好一組數據 (δ_i , T_i), $i = 1$, $\cdots$, t, 經由 Lagrange 內插多項式求出 $L(T)$, 並算出其常數項 $L(0) \pmod{p-1}$, 將 $L(0)$ 表示爲$E_p(a,b)$上的一點 R , U_A 再計算 $m_{1i} = x_A\alpha_i$以及 $m_{2i} = R + x_A\beta_i$ $(i = 1\ ,\ \cdots\ ,\ t)$, 並將 (m_{1i}, m_{2i}) 傳送給每一個參與秘密還原的分享者 U_i $(i=1,2,\cdots,t)$。

5. 分享者 U_i 利用私鑰 x_i 算出 $n_i = m_{2i} - x_i\, m_{1i}$ $(i = 1\ ,\ \cdots\ ,\ t)$, 再將 n_i 回復其原值 N_i (此 n_i 就是 R , 因而 N_i 就是 $L(0)$), 分享者驗證 $A_0 = g^{N_i} \pmod p$ 是否成立, 如果成立, 則說明 U_A 沒有欺騙行爲, 如果不成立, 則要求 U_A 重新發送計算之結果。最後, 分享者計算 $k = C_i - N_i \pmod{p-1}$ 。

5.4 例子 **-** 橢圓曲線版之動態 **(3,4)** 秘密分享

假設有四個成員參與秘密分享, 但是只要三位成員即可還原其秘密。

分發秘密階段:

1. U_D 選取質數 $p = 7919$, 以及 p 的一個原根 $g = 7$, 假設 $k = 229$是要分享的秘密。

2. 選取 $a = (a_0 , a_1 , a_2) = (401 , 7 , 11)$, 所以 U_D 可以得到
$f(x) = 401 + 7x + 11x^2 \pmod{7918}$。

3. U_D 利用 $f(x)$ 得到 (δ_i , $f(\delta_i)$) : (1 , 419),(2 , 459),(3 , 521),(4 , 605), 並計算 $s = k + a_0 = 630 \pmod{7918}$。

4. U_D 選取一橢圓曲線$E_{7919}(-1,0) = E$

$$E\ :\ y^2 = x^3 - x \pmod{7919},$$

$U = (U_1, U_2, U_3, U_4)$,在 E 上選取一點 α_i , 其密鑰爲 $x = (6\ ,5\ ,4\ ,3\ )$, 利用私鑰 x_i 計算 $\beta_i = x_i\ \alpha_i$ 則 α_i : $(91,3808),(73,437),(64,2416),(46,900),$

β_i ： $(5391,2634),(7813,3147),(1013,4732),(7686,7642)$, 並將 α_i , β_i 公開, 但 x_i 保持私密。

5. U_D 將其 $f(\delta_i)$ 寫成 $E_{7919}(-1,0)$ 上的一點 P_i , 以及將 s 表示爲 $E_{7919}(-1,0)$ W的一點 Q , 則 P_i ： $(3774,1296),(4132,2632),(4690,1417),(5446,447)$, Q ： $(5671,1883)$, 選取一隨機整數 $r = 8$, 並算出 $z_{1i} = r\alpha_i$, $z_{2i} = P_i + r\beta_i$, $z_{3i} = Q_i + r\beta_i$, 則
z_{1i} ： $(1900,7027),(3983,5774),(3374,3837),(5171,4270)$,
z_{2i} ： $(6969,6940),(5802,4911),(7780,7066),(5310,1858)$,
z_{3i} ： $(5067,3241),(1855,3291),(2658,2623),(6372,6178)$,
將(z_{1i},z_{2i},z_{3i})傳給 U_i 。

6. U_D 計算$K = 7^{229} = 6171 \pmod{7919}$, 計算 $A_i = g^{a_i} \pmod p$, $i = 0,1,2$ 得到 $A = (6807,7886,7876) \pmod{7919}$, 並將其皆公開。

訊息驗證階段:

1. U 計算 $c_i = z_{3i} - x_i z_{1i}$ ：$(5671,1883)$, 以及 $b_i = z_{2i} - x_i z_{1i}$ ： $(3774,1296),(4132,2632),(4690,1417),(5446,447)$, 並將 b_i 及 c_i 回復其原值分別爲 (B_i) = (419 , 459 , 521 , 605) 及 (C_i) = $(630,630,630,630)$ 。

2. U_i 可驗證下列等式是否成立
$g^{C_i} = K \cdot A_0 \pmod{7919}$, $g^{B_i} = \prod_{j=1}^{t} A_{j-1}{}^{\delta_i{}^{j-1}} \pmod{7919}$

3. 由上面我們可以知道 U_i 所得到的 B_i 以及 C_i 是正確的。

秘密還原階段:

1. 假設 U_1 , U_2 , U_3 共同還原秘密。參與的成員 U_1 , U_2 分別將 b_1, b_2 加密後傳送給合成者 U_3 。

2. 已知$(\alpha_3$, $\beta_3)$, U_i 將其手上的 b_i 作運算得 $e_{1i} = x_i\ \alpha_3$ 以及 $e_{2i} = b_i + x_i\ \beta_3$, 則 e_{1i} ： $(2824,2736),(7287,5299)$, e_{2i} ： $(6302,\ 773),(7682,7586)$, 並將其 (e_{1i},e_{2i}) 傳給 U_3 。

3. U_3 得到二組 (e_{1i},e_{2i}), $i = 1$,2, U_3 利用自己的私鑰 x_3 解出 $e_{2i} - x_j\ e_{1i}$ 得到 $(3774,1296)$及$(4132,2632)$, 並回復其原值分別爲 $T_1 = 419$ 及 $T_2 = 449$ 。

4. 合成者 U_3 首先驗證 $g^{T_i} = \prod_{j=1}^{t} A_{j-1}{}^{\delta_i{}^{j-1}} \pmod p$, $i = 1$,2 是否成立。若成立, 說明分享者 U_i 沒有欺騙行爲; 若不成立, 則要求其重新發送自己手中的資訊。

5. 當 U_3 獲得二組數據 (δ_i ,T_i), $i = 1$,2 , 由 Lagrange 內插多項式求出 $L(T)$, 並得其常數項爲 $L(0) = 401 \pmod{7918}$, 將此數表示爲 E 上的一點 $R = (3610,34)$, 並利用 U_i 的 $(\alpha_i$,$\beta_i), i = 1$, 2 計算 $m_{1i} = x_3\ \alpha_i$以及 $m_{2i} = R + x_3\beta_i$, $i = 1$, 2, 則 (m_{11} , m_{12}) = ($(3879,3381),(5491,6825)$), (m_{21} , m_{22}) = ($(1618,6597),(3141,1411)$), 分別傳給 U_1 及 U_2 。

6. U_1 及 U_1 利用私鑰 x_1,x_2 分別解出 $n_1 = m_{12} - x_1\ m_{11} = (3610,34)$, 以及 $n_2 = m_{22} - x_2\ m_{21} = (3610,34)$, 將其回復原值爲 $N_i = 401$ 。

7. 分享者驗證 $A_0 = g^{N_i} = 6807 \pmod{7919}$ 是否成立, 如果成立, 則說明 U_3 沒有欺騙行爲, 如果不成立, 則要求 U_3 重新發送計算之結果, 最後, 分享者計算 $k = 630 - 401 = 229 \pmod{7919}$ 。

5.5 安全性分析

在研究中, 我們利用橢圓曲線去改良動態秘密分享系統, 強化其加密性。此方法改正了當分享者將自己手上的訊息傳送給分享者時, 不需要經過安全秘密通道 ,假如 $t - 1$ 個成員想合謀還原秘密 k , 他們無法知道其他人的訊息也就無法求得秘密 k 。秘密 k 可以被無限次更新, 只要分發者重覆秘密分發過程3-6即可。而且在此方法的子秘密生成的算法及驗證算法具有數據公開以及數據量小的優點。在許多情況下, 秘密分享後並不馬上要求還原秘密 ⟨即是秘密的延遲恢復性⟩, 這就要求分享者在還原秘密之前先驗證自己手中的子秘密是否有效, 而大多文獻中, 這是無法做到的。因此, 難以避免分發者在分發假的秘密給某些合法的分享者, 而使包括他們的一組合法成員不能還原出正確的秘密。本方法的驗證過程有效地避免了這種情況的發生。另外, 文中的合成者 U_A , 有效避免以前方法中合成者本身也是參與秘密還原的分享者, 而他在恢復秘密後不提供或提供假的秘密給其他的參與秘密恢復的分享者的問題。

References

[1] A Shamir (1979) “How to share a secret[J],” *TComm. ACM*, **22(11)**, 612-613.

[2] H. M. Sun, S. P. Shieh (1994), “Construction of dynamic threshold schemes[J],” *lectronics Letters*, **30(24)** , 2023-2025.

[3] E. Dawson (1995), “Multisecret - sharing scheme based on one - way function[J] ,” *Electronics Letters*.

[4] J. Pieprzyk, T. Hardjono, J. Seberry (1998), “Fundamentals of Computer Security,” 361-366.

[5] N. Koblitz (1987), “Elliptic curve cryptosystems,” *Mathematics of Computation*, **48(177)**, 203-209.

[6] V. S. Miller (1986), “Use of elliptic curves in cryptography. In H. C. Williams, editor,” *Advances in Cryptology (CRYPTO’85)*, Lecture Notes in Computer Science No. **218**, pp. 417-426, Springer, Berlin Heidelberg New York.

[7] 沈淵源 (2006), “密碼學之旅與 MATHEMATICA 同行,” 全華科技圖書股份有限公司.

[8] Lawrence C. Washington (2003), “Elliptic Curves Number Theory and Cryptography,” 59-173.

[9] 翁榮茂 (2005), “橢圓曲線版的 ElGamal 門檻密碼系統,” 私立東海大學, 碩士論文.

[10] James K. Strayer (1994), “Elementary Number Theory”.

[11] Silverman, Joseph H./Tate, John (1992), “Rational Points on Elliptic Curves,” *Undergraduate Texts in Math*, Springer - Verlag, New York.

6 橢圓曲線版的動態**(3,4)** 秘密分享之 **Mathematica** 演算法

```
In[1]:= <<NumberTheory‘NumberTheoryFunctions‘
In[2]:= ecadd[p1_,p2_,a_,b_,n_]:=Module[z,m,x3,y3,p3,
        z=0;z1=1;If[p1==”infinity”,”infinity”,p3=p2;z=1,” ”];
        If[z==1,” ”,If[p2==”infinity”,”infinity”,p3=p1;z=1,” ”]];
        If[z==1,” ”, If[p1[[1]]==p2[[1]]&&p1[[2]]==p2[[2]]==0,
        p3=”infinity”,”infinity”;z=1,” ”]];
        If[z==1,” ”,If[p1[[1]]==p2[[1]]&&p1[[2]]≠2[[2]],
        p3=”infinity”,”infinity”;z=1,” ”]];
        If[z==1,” ”, If[p1==p2&&GCD[p1[[2]],n]≠1&&
        GCD[p1[[2]],n]≠n,z=1;
        z1=GCD[p1[[2]],n],” ”]];
        If[z==1,” ”,If[p1==p2,
        m=Mod[(3*p1[[1]]^2+a)*PowerMod[2*p1[[2]],-1,n],n];
        z=1;x3=m^2-p1[[1]]-p2[[1]];
        y3=m*(p1[[1]]-x3)-p1[[2]];p3=Mod[x3,y3,n],” ”]];
        If[z==1,” ”,If[GCD[p2[[1]]-p1[[1]],n]≠1,z=1;
        z1=GCD[p2[[1]]-p1[[1]],n],” ”]];
        If[z==1,” ”,m=Mod[(p2[[2]]-p1[[2]])*
        PowerMod[p2[[1]]-p1[[1]],-1,n],n];
        x3=m^2-p1[[1]]-p2[[1]];y3=m*(p1[[1]]-x3)-p1[[2]];
```

```
        p3=Mod[x3,y3,n]];If[z1==1,p3,"factor=",z1]];
        ecmus[p1_,m_,a_,b_,n_]:=Module[z,z=p1;
        For[i=1,i¡m&&z[[Length[z]]][[1]]≠"factor=",i++,
        z=Append[z,ecadd[p1,z[[Length[z]]],a,b,n]]];z]
        ecmlt[p1_,m_,a_,b_,n_]:=Last[ecmus[p1,m,a,b,n]]
```

秘密分發階段

```
In[5]:= p=Prime[1000] g=PrimitiveRoot[p]
Out[5]= 7919
Out[6]= 7
In[7]:= k=229; d={1,2,3,4}; x={6,5,4,3}; a={401,7,11};
In[11]:= f[x_]:=Mod[∑_{i=0}^{3} a[[i+1]]*x^i,p-1];
In[12]:= Table[{d[[i]],f[d[[i]]]},i,4]
Out[12]= {{1,419},{2,459},{3,521},{4,605}}
In[13]:= s=Mod[k+a[[1]],p-1] K=PowerMod[g,k,p]
         A=PowerMod[g,a,p]
Out[13]= 630
Out[14]= 6171
Out[15]= {6807,7886,7876}
In[16]:= alpha={{91,3808},{73,437},{64,2416},{46,900}};
         bb=Table[ecmlt[alpha[[j]],x[[j]],-1,0,p],j,4]
Out[17]= {{5391,2634},{7813,3147},{1013,4732},{7686,7642}}
In[18]:= P={{3774,1296},{4132,2632},{4690,1417},{5446,447}};
         Q={5671,1883};
In[20]:= z=Table[ecmlt[alpha[[j]],8,-1,0,p],{j,4}]
         zz=Table[ecadd[P[[j]],ecmlt[bb[[j]],8,-1,0,p],-1,0,p],{j,4}]
         zzz=Table[ecadd[Q,ecmlt[bb[[j]],8,-1,0,p],-1,0,p],{j,4}]

Out[20]= {{1900,7027},{3983,5774},{3374,3837},{5171,4270}}
Out[21]= {{6969,6940},{5802,4911},{7780,7066},{5310,1858}}
Out[22]= {{5067,3241},{1855,3291},{2658,2623},{6372,6178}}
```

訊息驗證階段

```
In[23]:= tmp=Table[ecmlt[z[[j]],x[[j]],-1,0,p],{j,4}];
         z=Table[tmp[[j]][[1]],-tmp[[j]][[2]],{j,4}];
         pp=Table[ecadd[zz[[j]],z[[j]],-1,0,p],{j,4}]
         qq=Table[ecadd[zzz[[j]],z[[j]],-1,0,p],{j,4}]
```

Out[25]= {{3774,1296},{4132,2632},{4690,1417},{5446,447}}

Out[26]= {{5671,1883},{5671,1883},{5671,1883},{5671,1883}}

In[27]:= u=Table[{Floor[pp[[i]][[1]]/9]},{i,4}]

v=Table[{Floor[qq[[i]][[1]]/9]},{i,4}]

Out[27]= {{419},{459},{521},{605}}

Out[28]= {{630},{630},{630},{630}}

In[29]:= PowerMod[g, v, p] == Table[{Mod[K*A[[1]], p]},

{i, 4}]

PowerMod[g, u, p] == Table[Mod[$\prod_{j=1}^{3}$ Power[A[[j]],

d[[i]]$^{(j-1)}$], p], {i, 4}]

Out[29]= True

Out[30]= True

秘密還原階段

In[31]:= e=Table[ecmlt[alpha[[3]],x[[j]],-1,0,p],{j,2}]

ee=Table[ecadd[pp[[j]],ecmlt[bb[[3]],x[[j]],

-1,0,p],-1,0,p],{j,2}]

Out[31]= {{2824,2736},{7287,5299}}

Out[32]= 6302,773,7682,7586

In[33]:= tmp=Table[ecmlt[e[[j]],x[[3]],-1,0,p],{j,2}];

e=Table[{tmp[[j]][[1]],-tmp[[j]][[2]]},{j,2}]

Y=Table[ecadd[ee[[j]],e[[j]],-1,0,p],{j,2}]

Out[34]= {{23,-65},{4072,-5660}}

Out[35]= {{3774,1296},{4132,2632}}

In[36]:= T=Table[{Floor[Y[[i]][[1]]/9]},{i,2}]

Out[36]= {{419},{459}}

In[37]:= PowerMod[g, T, p] == Table[Mod[$\prod_{j=1}^{3}$ Power[A[[j]],

d[[i]]$^{(j-1)}$], p], {i, 2}]

Out[37]= True

In[38]:= PolynomialMod[InterpolatingPolynomial[Table[

{d[[i]],f[d[[i]]]},{i,3}],xx],p-1]

Out[38]= 401 + 7 xx + 11 xx$\hat{}$2

In[39]:= %/.xx→ 0

Out[39]= 401

In[40]:= R=3610,34;

In[41]:= m=Table[ecmlt[alpha[[j]],x[[3]],-1,0,p],j,2]

```
        mm=Table[ecadd[R,ecmlt[bb[[j]],x[[3]],-1,0,p],
        -1,0,p],{j,2}]
Out[41]= {{3879,3381},{1618,6597}}
Out[42]= {{5491,6825},{3141,1411}}
In[43]:= tmp=Table[ecmlt[m[[j]],x[[j]],-1,0,p],{j,2}];
        m=Table[{tmp[[j]][[1]],-tmp[[j]][[2]]},{j,2}]
        n=Table[ecadd[mm[[j]],m[[j]],-1,0,p],{j,2}]
Out[44]= {{381,-7329},{2967,-7424}}
Out[45]= {{3610,34},{3610,34}}
In[46]:= q=Table[Floor[n[[i]][[1]]/9],{i,2}]
Out[46]= {401,401}
In[47]:= A[[1]]==PowerMod[g,q[[1]],p]
        A[[1]]==PowerMod[g,q[[2]],p]
Out[47]= True
Out[48]= True
In[49]:= Mod[s-q[[1]],p-1]
Out[49]= 229
In[50]:= %==k
Out[50]= True
```

利用 Koblitz 方法找出 E 上的點

```
In[51]:= k=9;mm=630;jj=1;xp=mm*k+jj
        Solve[{y^2== xp^3-xp,Modulus==p},
        Mode → Modular]
Out[51]= 5671
Out[52]= {{Modulus → 7919,y → 1883},
        {Modulus → 7919,y → 6036}}
```

Ellipitic curve version of the Dynamic Secret Sharing Scheme

Ching-Huai Lin* Yuan-Yuan Shen

Abstract

In this thesis, we propose an ellipitic curve version of Dynamic Secret Sharing Scheme. We give a very brief introduction to the history of cryptography first, and then discuss how to share secret. Next, we introduce the Dynamic Secret Sharing Scheme and the elliptic curve cryptosystem in chapter 3 and chapter 4 respectively. Finally, we present an ellipitic curve version of the Dynamic Secret Sharing Scheme.

* Department of mathematics, Tunghai University