Esquema de Compartilhamento de Segredos de Shamir

28 de fevereiro de 2011

(Este texto é adptado notas de aula de Paradigmas de Programação)

Temos um número secreto que queremos esconder, mas gostaríamos que ele fosse revelado quando algumas pessoas de confiança decidissem fazê-lo. Distribuímos então "chaves" a estas n pessoas, e quando uma parte delas (um terço, metade, ou a quantidade que decidirmos) combinar as chaves, o número será revelado. O problema é que queremos que *qualquer* grupo de tamanho suficiente possa revelar o segredo.

Por exemplo, suponha que há vinte pessoas emuma empresa responsáveis por guardar um segredo, mas que queiramos definir o quórum de oito delas para que o segredo seja revelado. Um grupo de sete não deve conseguir encontrar o segredo, mas um grupo de oito sim. Desta forma, se oito pessoas combinarem seus "pedaços de chave", podem obter o segredo.

O esquema de compartilhamento de segredos que implementaremos foi desenvolvido por Adi Shamir em 1979, por isso o chamaremos de *SSSS (Shamir's Secret Sharing Scheme)*. A idéia que faz o sistema funcionar é esta: com dois pontos conseguimos representar uma única reta; com três pontos, uma parábola; com quatro, um polinômio de grau 3 e, de maneira geral, podemos representar unicamente um polinômio de grau k usando $k+1$ pontos. Além disso, com um ponto a menos não há como adivinhar ou aproximar o polinômio de maneira eficiente: há infinitas retas passando pelo ponto $(2,3)$, e o mesmo acontece com polinômios de grau maior. Por exemplo, com os pontos$\{(2,3),(3,j),(4,6)\}$ temos, para $j=4$,

$$\frac{x^2}{2} - \frac{3x}{2} + 4;$$

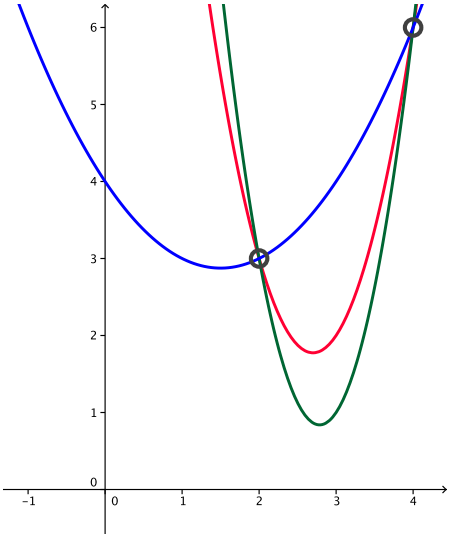
para $j=1$,

$$\frac{7x^2}{2} - \frac{-39x}{2} + 28;$$

para $j=2$,

$$\frac{5x^2}{2} - \frac{-27x}{2} + 20.$$

Estas parábolas passando por $(2,3)$ e $(4,6)$ são ilustradas na figura a seguir:



Se quisermos então compartilhar um segredo entre vinte pessoas, determinando que quaisquer 5 delas podem juntas revelar o segredo, simplesmente criamos um polinômio $a(x)$ de grau 4 cujo termo constante a_0 é o segredo.

$$a_4x^4 + a_3x^3 + a_2x^2 + a^1x + a_0 = 0$$

Damos um ponto polinômio para cada uma das vinte pessoas, e com cinco pontos conseguiremos determinar o polinômio. Tendo o polinômio completo, podemos encontrar a_0.

Por razões que fogem ao objetivo deste texto, o SSSS usa aritmética modular – na verdade, aritmética módulo p, onde p é um número primo grande (em termos práticos isto significa que depois de fazer alguma conta e obter o resultado x, devemos tomar o resto da divisão de x por p). Usaremos $p = 983226812132450720708095377479$.

Para compartilhar um segredo entre w partes com limiar (quórum) igual a t:

1. Escolhemos aleatoreamente $t-1$ números menores que p, que chamaremos de $a_1, \cdots, a_{t-1}$. Temos agora um polinômio:

$$a(x) = a_0 + \sum_{j=1}^{t-1} a_j x^j \pmod{p}$$

   onde a_0 é o segredo.

2. Para cada parte $1 \leq i \leq w$, calculamos $a(i)$ e entregamos a essa parte o par $(i, a(i))$.

Para obter o segredo a partir de k chaves, usaremos o polinômio interpolador de Lagrange: dados os t pares $(x_i, a(x_i))$, o valor do polinômio a no ponto x é dado por

$$\begin{aligned} l(x) &= \sum_{j=1}^{n} l_j(x) \pmod p \\ l_j(x) &= y_j \prod_{k=1;k\neq j}^{t} \frac{(x - x_k)}{(x_j - x_k)} \pmod p. \end{aligned}$$

O segredo é o termo constante a_0 do polinômio, por isso basta obtermos o valor do polinômio no ponto zero, $l(0)$. Isso simplifica a computação do segredo:

$$\begin{aligned} l(0) &= \sum_{j=1}^{n} y_j l_j(x) \pmod p \\ l_j(x) &= \prod_{k=1;k\neq j}^{t} \frac{x_k}{(x_k - x_j)} \pmod p. \end{aligned}$$

O procedimento `lagrange-aux` calcula $l_j(x)$:

A recuperação do segredo é feita pelo procedimento `ssss-restore-integer`, que calcula $l(0)$.