

Álgebra de las votaciones para procesos electorales seguros

Eduardo Ruiz Duarte

Facultad de Ciencias UNAM

Febrero 6, 2010

- ¿Qué es esto?
- Logaritmo discreto
- Esquema DHMEW
- Algoritmo Shamir para compartir secretos
- Prueba Zero-Knowledge
- Un ejemplo de votación
- Conclusiones

Mucho se ha debatido sobre elecciones electrónicas, aquí mencionaré algunas ideas. La problemática socio-política no es parte de esta presentación ya que para mí es beta también aunque al final podremos discutirlo un poco, aquí trataré de explicar por qué sería seguro algebraicamente hablando, pongan mucha atención.

¿Qué es esto?

En un esquema de votación electrónica existen dos distintos tipos de participantes:

Votantes (Votan)

Autoridad de votación (Junta los votos)

Las siguientes propiedades son necesarias:

1. Verificabilidad universal

Todos podrán checar que no haya trampa

2. Privacidad

Asegura la confidencialidad de los votos

3. Robusto

Debe de funcionar aunque haya tramposos y solo votantes autorizados pueden emitir un voto

El logaritmo discreto es la base de muchos protocolos criptográficos intuitivamente consiste en el problema de calcular b de $a^b = c \pmod{J}$ en un grupo finito dado a, c, J

Problema de logaritmo discreto:

Sea $\langle G, * \rangle$ cíclico, $|G| = n$, $b \in G$ y $\langle b \rangle = G \Rightarrow \forall g \in G$
 $g = b^k$, k entero y definimos el morfismo de grupos:

$$\log_b : G \rightarrow \mathbb{Z}_n$$

$$g \mapsto [k]$$

de tal manera que $b^k = g \pmod n$

Problema de logaritmo discreto

No existe hasta ahora un algoritmo para calcular esta k en tiempo polinomial dados g y b aunque existen cosas mas rápidas que "intentar todo" como *pollard* – ρ o la criba de campo de funciones, criba numerica, etc.., también es muy rápido si usas tu computadora cuántica usando el algoritmo de Shor para el problema de logaritmo discreto y la transformada de fourier cuántica :p

Problema de logaritmo discreto

Una aplicación del problema de logaritmo discreto es intercambiar llaves a través de medios no seguros, Martin Hellman, Whitfield Diffie, Ralph Merkle pensaron en un algoritmo usando el problema de logaritmo discreto y no olvidemos a los que lo descubrieron antes pero por razones de secreto militar no podían publicarlo, Malcolm J. Williamson y James H. Ellis de Inglaterra en la GCHQ. Este esquema permitirá comunicar datos cifrados entre personas que no tienen nada en común, por ejemplo un password
Por ejemplo si los votantes quieren transmitir datos cifrados a la autoridad de votación

Protocolo Diffie-Hellman-Merkle-Ellis-Williamson

Alberto (A) quiere cifrar un mensaje a Berenice (B) pero necesitan un password en común, el problema es que no pueden comunicarse el password ya que podrían estar intervenidos, y Eulalio (E) podría estar capturando su tráfico entonces:

- A y B se ponen de acuerdo en un $\langle \mathbb{Z}_p^*, * \rangle$ y en un $g \in G$ tal que g es generador, (Nótese que E ya tiene esta información)
- A toma un $a \in \mathbb{Z}_p$, calcula $A_1 = g^a \pmod p$ y manda A_1 a B
- B toma un $b \in \mathbb{Z}_p$, calcula $B_1 = g^b \pmod p$ y manda B_1 a A
- A calcula $S_a = B_1^a \pmod p$
- B calcula $S_b = A_1^b \pmod p$
- $S_a = S_b$ ya que $S_a = (g^b)^a = S_b = (g^a)^b = g^{ab} = S$

Alberto y Berenice ya tienen un secreto S y no importa que Eulalio conozca A_1, B_1, p y g

Esquema Shamir para compartir secretos

Recordemos antes un requerimiento:

¿Cómo podemos generar una función continua que pase por ciertos puntos dados en el plano?

Esquema Shamir para compartir secretos

Interpolación de Lagrange.

Dados $n + 1$ puntos (x_i, y_i) con $x_i \neq x_j$ queremos un $p(x)$ tal que $p(x_i) = y_i \forall i$

$$p(x) = \sum_{j=0}^n y_j \mathcal{L}_j(x)$$

Donde $\mathcal{L}_j(x)$ es el j -ésimo polinomio de Lagrange definido por los puntos dados

$$\mathcal{L}_j(x) = \prod_{i=0, i \neq j}^n \frac{x - x_i}{x_j - x_i}$$

Este polinomio existe no es parte de la presentación demostrarlo pero para los curiosos, la respuesta está detrás de la matriz de Vandermonde.

Esquema Shamir para compartir secretos

Regresando al esquema de compartir secretos, éste puede ser muy útil para integrar en un protocolo de elecciones, usen su imaginación.

Consiste en que un secreto se divide en partes a varias entidades de tal manera que el secreto solo puede construirse cuando ciertas entidades juntan la información

Esquema Shamir para compartir secretos

Dividiremos a D (una clave, un número) en n partes D_1, D_2, \dots, D_n

1. El saber k o más piezas D_i hace que D sea fácilmente computable
2. El saber $k-1$ o menos piezas D_i hace que D sea imposible de computar

Si $k=n$ entonces todos son requeridos para construir el secreto.

Esquema Shamir para compartir secretos

Veamos un ejemplo del algoritmo.

-Supongamos que el secreto es $s = 1234$

-Dividiremos en $n=6$ partes pero queremos que con tan solo $k=3$ partes sea suficiente construir el secreto

-Calculamos $k-1=2$ números aleatorios $a_1 = 166, a_2 = 94$

-Construimos un polinomio

$$f(x) = s + a_1x + a_2x^2 = 1234 + 166x + 94x^2$$

-Sacamos $n=6$ puntos de ese polinomio para cada entidad, por facilidad será $f(i) = y_i \quad i = 1, 2 \dots 6$

-Obtenemos $(1,1494);(2,1942);(3,2578);(4,3402);(5,4414);(6,5614)$ y cada quien le damos un punto.

Esquema Shamir para compartir secretos

Reconstrucción de secreto:

Supongamos que 3 entidades quieren construir el secreto, ellos tienen $(2,1942);(4,3402);(5,4415)$

-Computamos los polinomios de Lagrange $\mathcal{L}_j(x)$

$$\mathcal{L}_0(x) = \frac{x^2}{6} - \frac{x}{2} + 1$$

$$\mathcal{L}_1(x) = \frac{-x^2}{2} - \frac{3x}{2} - 5$$

$$\mathcal{L}_2(x) = \frac{x^2}{3} - 2x + \frac{4}{3}$$

Ahora el polinomio de lagrange es:

$$f(x) = \sum_{j=0}^2 y_j \mathcal{L}_j(x) = 1942 \left(\frac{x^2}{6} - \frac{x}{2} + 1 \right) + 3401 \left(\frac{-x^2}{2} - \frac{3x}{2} - 5 \right) +$$

$$4414 \left(\frac{x^2}{3} - 2x + \frac{4}{3} \right) = 1234 + 166x + 94x^2 \quad \text{y aqui tenemos}$$

$$s=1234$$

¿Cómo podemos demostrarle a alguien que sabemos un secreto sin revelárselo de tal manera que la otra persona quede totalmente convencida? Esto nos puede ayudar a adquirir confianza en la entidad que cuenta los votos, veremos un algoritmo utilizando logaritmo discreto

Una prueba zero-knowledge tiene 3 propiedades:

1. **Compleitud:** El que verifica si es verdadera cierta proposición será convencido de ésta
2. **Validez:** Si la proposición es falsa el que intenta probar tiene una mínima probabilidad de enganar al otro
3. **Zero-Knowledge:** Si la proposición es cierta el que prueba no debió haber revelado mas que la veracidad y no la info secreta

Prueba Zero-Knowledge

Prueba de igualdad de logaritmo:

En esta prueba A le proba a B que posee dos mensajes que tienen el mismo logaritmo discreto con respecto a dos bases diferentes.

Ambos conocen $(g_1, y_1 = g_1^x, g_2, y_2 = g_2^x)$ con x random

1. A escoge $r \in 0, \dots, q - 1$ aleatorio y hace $p := (p_1, p_2) = (g_1^r, g_2^r)$, y le manda p a B
2. B escoge $c \in 0, \dots, q - 1$ y le manda c a A
3. A computa $q := r - cx$ y se le manda q a B
4. B acepta si y solo si $a_1 = g_1^q y_1^c$ y $a_2 = g_2^q y_2^c$

Esto funciona ya que aunque B no sabe r

$$a_1 = g_1^q y_1^c = g_1^{r-cx} y_1^c = g_1^r g_1^{-cx} g_1^{xc} = g_1^r$$

Análogamente con a_2 esto hace que B acepte que existe tal r (password) sin que le sea revelado.

Ya con estas herramientas:

Cada votante V_i selecciona su voto $v_i \in -1, 1$ y codifica v_i como g^{v_i} y lo cifra con Elgamal.

$$c_i := (c_{i,1}, c_{i,2}) := (g^{\alpha_i}, h^{\alpha_i} g^{v_i})$$

Este mensaje lo firma y prueba con zero knowledge que realmente siguió este procedimiento

Conteo de m votos

1. Todos los V_i pueden computar:

$$c = (c_1, c_2) = \left(\prod_{i=1}^m c_{i,1}, \prod_{i=1}^m c_{i,2} \right)$$

Noten que $c = (c_1, c_2)$ es g^d donde d es la diferencia de los votos 1 o -1 (si o no)

2. Para obtener d solo hay que comparar la sucesión

$g^{-m}, g^{-m+1}, \dots, g^{-m+i} \dots$ y que cada quien compare con g^{-d} cuando $g^{-m+i} = g^{-d}$ habrán encontrado $i=d$.

¡Gracias! Eduardo Ruiz Duarte
beck@math.co.ro
<http://math.co.ro>
<http://b3ck.blogspot.com>
PGP Key ID: **FEE7 F2A0**
Twitter: **@toorandom**