UVG UNIVERSIDAD DEL VALLE DE GUATEMALA

Intercambio de claves. Método de Diffie-Hellman.

Alan Reyes-Figueroa

Criptografía y Cifrado de Información (Aula 16) 05.octubre.2021

Logaritmo Discreto

Dado un primo $p > 2$, tomemos $\mathbf{g} \in (\mathbb{Z}/p\mathbb{Z})^*$, un elemento de orden q, esto es

$$\mathbf{g}^q \equiv 1 \pmod{p}, \qquad \text{pero } \mathbf{g}^j \not\equiv 1 \pmod{p}, \text{ para } 1 \leq j < q.$$

Logaritmo Discreto

Dado un primo $p > 2$, tomemos $\mathbf{g} \in (\mathbb{Z}/p\mathbb{Z})^*$, un elemento de orden q, esto es

$$\mathbf{g}^q \equiv 1 \pmod{p}, \qquad \text{pero } \mathbf{g}^j \not\equiv 1 \pmod{p}, \text{ para } 1 \leq j < q.$$

Ya vimos que calcular potencias módulo p es una tarea relativamente simple. Veamos ahora el problema inverso.

Logaritmo Discreto

Dado un primo $p > 2$, tomemos $\mathbf{g} \in (\mathbb{Z}/p\mathbb{Z})^*$, un elemento de orden q, esto es

$$\mathbf{g}^q \equiv 1 \pmod{p}, \qquad \text{pero } \mathbf{g}^j \not\equiv 1 \pmod{p}, \text{ para } 1 \leq j < q.$$

Ya vimos que calcular potencias módulo p es una tarea relativamente simple. Veamos ahora el problema inverso.

Dada la función $\mathbf{x} \longrightarrow \mathbf{g}^{\mathbf{x}} = \mathbf{y} \pmod{p}$,

Logaritmo Discreto

Dado un primo $p > 2$, tomemos $\mathbf{g} \in (\mathbb{Z}/p\mathbb{Z})^*$, un elemento de orden q, esto es

$$\mathbf{g}^q \equiv 1 \pmod{p}, \qquad \text{pero } \mathbf{g}^j \not\equiv 1 \pmod{p}, \text{ para } 1 \leq j < q.$$

Ya vimos que calcular potencias módulo p es una tarea relativamente simple. Veamos ahora el problema inverso.

Dada la función $\mathbf{x} \longrightarrow \mathbf{g}^{\mathbf{x}} = \mathbf{y} \pmod{p}$, consideramos la función inversa

$$\log_{\mathbf{g}}(\mathbf{y}) = \log_{\mathbf{g}}(\mathbf{y}) = \mathbf{x}, \text{ donde } \mathbf{x} \in \{0, 1, \ldots, q-1\}.$$

Logaritmo Discreto

Dado un primo $p > 2$, tomemos $\mathbf{g} \in (\mathbb{Z}/p\mathbb{Z})^*$, un elemento de orden q, esto es

$$\mathbf{g}^q \equiv 1 \pmod{p}, \qquad \text{pero } \mathbf{g}^j \not\equiv 1 \pmod{p}, \text{ para } 1 \leq j < q.$$

Ya vimos que calcular potencias módulo p es una tarea relativamente simple. Veamos ahora el problema inverso.

Dada la función $\mathbf{x} \longrightarrow \mathbf{g}^{\mathbf{x}} = \mathbf{y} \pmod{p}$, consideramos la función inversa

$$\log_{\mathbf{g}}(\mathbf{y}) = \log_{\mathbf{g}}(\mathbf{y}) = \mathbf{x}, \text{ donde } \mathbf{x} \in \{0, 1, \ldots, q-1\}.$$

Esta función se llama el **logaritmo discreto** con base **g** de **y**, módulo p.

Logaritmo Discreto

Dado un primo $p > 2$, tomemos $\mathbf{g} \in (\mathbb{Z}/p\mathbb{Z})^*$, un elemento de orden q, esto es

$$\mathbf{g}^q \equiv 1 \pmod{p}, \qquad \text{pero } \mathbf{g}^j \not\equiv 1 \pmod{p}, \text{ para } 1 \leq j < q.$$

Ya vimos que calcular potencias módulo p es una tarea relativamente simple. Veamos ahora el problema inverso.

Dada la función $\mathbf{x} \longrightarrow \mathbf{g}^{\mathbf{x}} = \mathbf{y} \pmod{p}$, consideramos la función inversa

$$\log_{\mathbf{g}}(\mathbf{y}) = \log_{\mathbf{g}}(\mathbf{y}) = \mathbf{x}, \text{ donde } \mathbf{x} \in \{0, 1, \ldots, q-1\}.$$

Esta función se llama el **logaritmo discreto** con base **g** de **y**, módulo p.

Ejemplo: Para $p = 11$

a	1	2	3	4	5	6	7	8	9	10
$\log_2 a$	0	1	8	2	4	9	7	3	6	5

Definición

*Un **grupo** G es un conjunto no vacío de elementos, junto con una operación binaria $\cdot : G \times G \to G$ que satisface*

1. *(asociatividad)* $(a \cdot b) \cdot c = a \cdot (b \cdot c)$
2. *(elemento neutro) existe* $e \in G$ *tal que* $(a \cdot e) = e \cdot a = a$*, para todo* $a \in G$*.*
3. *(elementos invertos) para todo* $a \in G$*, existe* $b \in G$ *tal que* $a \cdot b = b \cdot a = e$*.*

Definición

*Un **grupo** G es un conjunto no vacío de elementos, junto con una operación binaria $\cdot : G \times G \to G$ que satisface*

1. *(asociatividad)* $(a \cdot b) \cdot c = a \cdot (b \cdot c)$
2. *(elemento neutro) existe* $e \in G$ *tal que* $(a \cdot e) = e \cdot a = a$*, para todo* $a \in G$*.*
3. *(elementos invertos) para todo* $a \in G$*, existe* $b \in G$ *tal que* $a \cdot b = b \cdot a = e$*.*

Si adicionalmente G cumple con

4. *(conmutatividad)* $a \cdot b = b \cdot a$*, para todo* $a, b \in G$

*entonces G se llama un **grupo conmutativo**.*

Logaritmo Discreto

Definición

*Un **grupo** G es un conjunto no vacío de elementos, junto con una operación binaria $\cdot : G \times G \to G$ que satisface*

1. *(asociatividad)* $(a \cdot b) \cdot c = a \cdot (b \cdot c)$
2. *(elemento neutro) existe* $e \in G$ *tal que* $(a \cdot e) = e \cdot a = a$*, para todo* $a \in G$*.*
3. *(elementos invertos) para todo* $a \in G$*, existe* $b \in G$ *tal que* $a \cdot b = b \cdot a = e$*.*

Si adicionalmente G cumple con

4. *(conmutatividad)* $a \cdot b = b \cdot a$*, para todo* $a, b \in G$

*entonces G se llama un **grupo conmutativo**.*

Ejemplos: $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$, $\mathbb{C}$,

Logaritmo Discreto

Definición

*Un **grupo** G es un conjunto no vacío de elementos, junto con una operación binaria $\cdot : G \times G \to G$ que satisface*

1. *(asociatividad)* $(a \cdot b) \cdot c = a \cdot (b \cdot c)$
2. *(elemento neutro) existe* $e \in G$ *tal que* $(a \cdot e) = e \cdot a = a$*, para todo* $a \in G$*.*
3. *(elementos invertos) para todo* $a \in G$*, existe* $b \in G$ *tal que* $a \cdot b = b \cdot a = e$*.*

Si adicionalmente G cumple con

4. *(conmutatividad)* $a \cdot b = b \cdot a$*, para todo* $a, b \in G$

*entonces G se llama un **grupo conmutativo**.*

Ejemplos: $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$, $\mathbb{C}$, $\mathbb{Z}/p\mathbb{Z}$

Definición

*Un **grupo** G es un conjunto no vacío de elementos, junto con una operación binaria $\cdot : G \times G \to G$ que satisface*

1. *(asociatividad)* $(a \cdot b) \cdot c = a \cdot (b \cdot c)$
2. *(elemento neutro) existe* $e \in G$ *tal que* $(a \cdot e) = e \cdot a = a$*, para todo* $a \in G$*.*
3. *(elementos invertos) para todo* $a \in G$*, existe* $b \in G$ *tal que* $a \cdot b = b \cdot a = e$*.*

Si adicionalmente G cumple con

4. *(conmutatividad)* $a \cdot b = b \cdot a$*, para todo* $a, b \in G$

*entonces G se llama un **grupo conmutativo**.*

Ejemplos: $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$, $\mathbb{C}$, $\mathbb{Z}/p\mathbb{Z}$
$\mathbb{R}^n$, $\mathbb{R}^{m \times n}$, $F(a, b)$, $C(a, b)$,

Logaritmo Discreto

Definición

*Un **grupo** G es un conjunto no vacío de elementos, junto con una operación binaria $\cdot : G \times G \to G$ que satisface*

1. *(asociatividad)* $(a \cdot b) \cdot c = a \cdot (b \cdot c)$
2. *(elemento neutro) existe* $e \in G$ *tal que* $(a \cdot e) = e \cdot a = a$*, para todo* $a \in G$*.*
3. *(elementos invertos) para todo* $a \in G$*, existe* $b \in G$ *tal que* $a \cdot b = b \cdot a = e$*.*

Si adicionalmente G cumple con

4. *(conmutatividad)* $a \cdot b = b \cdot a$*, para todo* $a, b \in G$

*entonces G se llama un **grupo conmutativo**.*

Ejemplos: $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$, $\mathbb{C}$, $\mathbb{Z}/p\mathbb{Z}$
$\mathbb{R}^n$, $\mathbb{R}^{m \times n}$, $F(a, b)$, $C(a, b)$,
S_n es el grupo de permutaciones de n elementos. se **g** de **y**, módulo p.

Logaritmo Discreto

Definición

*Un **grupo** G es un conjunto no vacío de elementos, junto con una operación binaria $\cdot : G \times G \to G$ que satisface*

1. *(asociatividad)* $(a \cdot b) \cdot c = a \cdot (b \cdot c)$
2. *(elemento neutro) existe* $e \in G$ *tal que* $(a \cdot e) = e \cdot a = a$*, para todo* $a \in G$*.*
3. *(elementos invertos) para todo* $a \in G$*, existe* $b \in G$ *tal que* $a \cdot b = b \cdot a = e$*.*

Si adicionalmente G cumple con

4. *(conmutatividad)* $a \cdot b = b \cdot a$*, para todo* $a, b \in G$

*entonces G se llama un **grupo conmutativo**.*

Ejemplos: $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$, $\mathbb{C}$, $\mathbb{Z}/p\mathbb{Z}$
$\mathbb{R}^n$, $\mathbb{R}^{m \times n}$, $F(a, b)$, $C(a, b)$,
S_n es el grupo de permutaciones de n elementos. se **g** de **y**, módulo p.

Logaritmo Discreto

Dado un grupo **cíclico** y un generador $\mathbf{g}$ para el grupo G:

$$G = \{1, \mathbf{g}, \mathbf{g}^2, \mathbf{g}^3, \mathbf{g}^4, \ldots, \mathbf{g}^{q-2}, \mathbf{g}^{q-1}\}.$$

(recordemos que q es el orden de $\mathbf{g}$).

Logaritmo Discreto

Dado un grupo **cíclico** y un generador $\mathbf{g}$ para el grupo G:

$$G = \{1, \mathbf{g}, \mathbf{g}^2, \mathbf{g}^3, \mathbf{g}^4, \ldots, \mathbf{g}^{q-2}, \mathbf{g}^{q-1}\}.$$

(recordemos que q es el orden de $\mathbf{g}$).

Definición

Decimos que el problema del logaritmo discreto $\log_{\mathbf{g}} \mathbf{y} = \log_{\mathbf{g}}(\mathbf{g}^{\mathbf{x}}) = \mathbf{x}$ *es* ***difícil*** *en* G *si para todo algoritmos eficiente* $\mathcal{A}$*, se cumple*

$$\mathbb{P}_{\mathbf{g} \leftarrow G,\ \mathbf{x} \leftarrow \mathbb{Z}/q\mathbb{Z}}[\mathcal{A}(G, q, \mathbf{g}, \mathbf{y}) = \mathbf{x}] < \varepsilon$$

es negligible.

Logaritmo Discreto

Dado un grupo **cíclico** y un generador $\mathbf{g}$ para el grupo G:

$$G = \{1, \mathbf{g}, \mathbf{g}^2, \mathbf{g}^3, \mathbf{g}^4, \ldots, \mathbf{g}^{q-2}, \mathbf{g}^{q-1}\}.$$

(recordemos que q es el orden de $\mathbf{g}$).

Definición

Decimos que el problema del logaritmo discreto $\log_{\mathbf{g}} \mathbf{y} = \log_{\mathbf{g}}(\mathbf{g}^{\mathbf{x}}) = \mathbf{x}$ *es* ***difícil*** *en* G *si para todo algoritmos eficiente* $\mathcal{A}$*, se cumple*

$$\mathbb{P}_{\mathbf{g} \leftarrow G,\ \mathbf{x} \leftarrow \mathbb{Z}/q\mathbb{Z}}[\mathcal{A}(G, q, \mathbf{g}, \mathbf{y}) = \mathbf{x}] < \varepsilon$$

es negligible.

Ejemplos:

- $U(p)$ grupos de unidades módulo p, para p muy grande.
- Grupos de curvas elípticas módulo p.

Intercambio de Claves

En esta aula veremos protocolos para el ntercambio de claves **k** de modo que dos entes puedan iniciar un esquema de intercambio de información segura.

Intercambio de Claves

En esta aula veremos protocolos para el ntercambio de claves **k** de modo que dos entes puedan iniciar un esquema de intercambio de información segura.

Intercambio de Claves

En esta aula veremos protocolos para el ntercambio de claves **k** de modo que dos entes puedan iniciar un esquema de intercambio de información segura.

Observaciones:

- En este caso nos centramos únicamente en el problema de intercambiar una clave y evita que sea leída por terceros. No se analiza el problema de autenticación, esto es, verificar que no ha ocurido modificación de información.

Intercambio de Claves

En esta aula veremos protocolos para el ntercambio de claves **k** de modo que dos entes puedan iniciar un esquema de intercambio de información segura.

Observaciones:

- En este caso nos centramos únicamente en el problema de intercambiar una clave y evita que sea leída por terceros. No se analiza el problema de autenticación, esto es, verificar que no ha ocurido modificación de información.
- Existen protocolos de intercambio de claves basados en funciones hash (*e.g.* SHA-256), que se conocen tomo *Merkle puzzles*. Sin embargo, éstos se consideran demasiando inseguros desde el punto de vista práctico.

Método de Diffie-Hellman

Desarrollado por Whitfield Diffie y Martin Hellman en 1975, publicado en “New Directions in Cryptography”, IEEE Transactions on Information Theory. 22 (6): 644–654.

Método de Diffie-Hellman

Desarrollado por Whitfield Diffie y Martin Hellman en 1975, publicado en “New Directions in Cryptography”, IEEE Transactions on Information Theory. 22 (6): 644–654.

- Fijamos un primo p grande (600 o más dígitos, alrededor de 2000 bits).

Método de Diffie-Hellman

Desarrollado por Whitfield Diffie y Martin Hellman en 1975, publicado en "New Directions in Cryptography", IEEE Transactions on Information Theory. 22 (6): 644–654.

- Fijamos un primo p grande (600 o más dígitos, alrededor de 2000 bits).
- Fijamos un entero $g \in \{1, 2, \ldots, p-1\}$.

Método de Diffie-Hellman

Desarrollado por Whitfield Diffie y Martin Hellman en 1975, publicado en "New Directions in Cryptography", IEEE Transactions on Information Theory. 22 (6): 644–654.

- Fijamos un primo p grande (600 o más dígitos, alrededor de 2000 bits).
- Fijamos un entero $g \in \{1, 2, \ldots, p-1\}$. Así, (p, g) se vuelven parámetros del método de Diffie-Hellman. Una vez elegidos, no cambian.

Método de Diffie-Hellman

Desarrollado por Whitfield Diffie y Martin Hellman en 1975, publicado en "New Directions in Cryptography", IEEE Transactions on Information Theory. 22 (6): 644–654.

- Fijamos un primo p grande (600 o más dígitos, alrededor de 2000 bits).
- Fijamos un entero $g \in \{1, 2, \ldots, p-1\}$. Así, (p, g) se vuelven parámetros del método de Diffie-Hellman. Una vez elegidos, no cambian.
- Alice elige un entero aleatorio $\mathbf{a} \in \{1, 2, \ldots, p-1\}$.

Método de Diffie-Hellman

Desarrollado por Whitfield Diffie y Martin Hellman en 1975, publicado en "New Directions in Cryptography", IEEE Transactions on Information Theory. 22 (6): 644–654.

- Fijamos un primo p grande (600 o más dígitos, alrededor de 2000 bits).
- Fijamos un entero $g \in \{1, 2, \ldots, p-1\}$. Así, (p, g) se vuelven parámetros del método de Diffie-Hellman. Una vez elegidos, no cambian.

- Alice elige un entero aleatorio $\mathbf{a} \in \{1, 2, \ldots, p-1\}$.
- Bob elige un entero aleatorio $\mathbf{b} \in \{1, 2, \ldots, p-1\}$.

Método de Diffie-Hellman

Desarrollado por Whitfield Diffie y Martin Hellman en 1975, publicado en "New Directions in Cryptography", IEEE Transactions on Information Theory. 22 (6): 644–654.

- Fijamos un primo p grande (600 o más dígitos, alrededor de 2000 bits).
- Fijamos un entero $g \in \{1, 2, \ldots, p-1\}$. Así, (p, g) se vuelven parámetros del método de Diffie-Hellman. Una vez elegidos, no cambian.

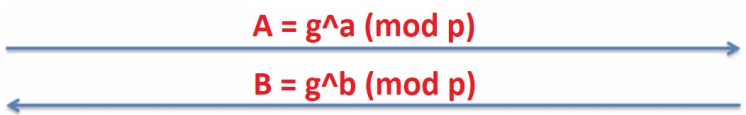
- Alice elige un entero aleatorio $\mathbf{a} \in \{1, 2, \ldots, p-1\}$.
- Bob elige un entero aleatorio $\mathbf{b} \in \{1, 2, \ldots, p-1\}$.
- Alice envía $A = g^{\mathbf{a}} \pmod{p}$, y Bob envía $B = g^{\mathbf{b}} \pmod{p}$.

Método de Diffie-Hellman

Desarrollado por Whitfield Diffie y Martin Hellman en 1975, publicado en "New Directions in Cryptography", IEEE Transactions on Information Theory. 22 (6): 644–654.

- Fijamos un primo p grande (600 o más dígitos, alrededor de 2000 bits).
- Fijamos un entero $g \in \{1, 2, \ldots, p-1\}$. Así, (p, g) se vuelven parámetros del método de Diffie-Hellman. Una vez elegidos, no cambian.

- Alice elige un entero aleatorio $\mathbf{a} \in \{1, 2, \ldots, p-1\}$.
- Bob elige un entero aleatorio $\mathbf{b} \in \{1, 2, \ldots, p-1\}$.
- Alice envía $A = g^{\mathbf{a}} \pmod{p}$, y Bob envía $B = g^{\mathbf{b}} \pmod{p}$.

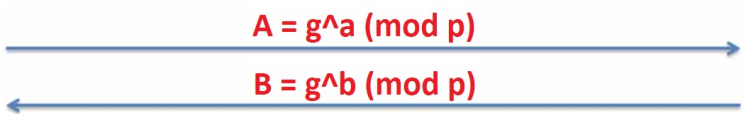


Método de Diffie-Hellman

Desarrollado por Whitfield Diffie y Martin Hellman en 1975, publicado en "New Directions in Cryptography", IEEE Transactions on Information Theory. 22 (6): 644–654.

- Fijamos un primo p grande (600 o más dígitos, alrededor de 2000 bits).
- Fijamos un entero $g \in \{1, 2, \ldots, p-1\}$. Así, (p, g) se vuelven parámetros del método de Diffie-Hellman. Una vez elegidos, no cambian.

- Alice elige un entero aleatorio $\mathbf{a} \in \{1, 2, \ldots, p-1\}$.
- Bob elige un entero aleatorio $\mathbf{b} \in \{1, 2, \ldots, p-1\}$.
- Alice envía $A = g^{\mathbf{a}} \pmod{p}$, y Bob envía $B = g^{\mathbf{b}} \pmod{p}$.



Ahora, ambos comparten la clave secreta $\mathbf{k} = g^{\mathbf{ab}} \pmod{p}$.

Método de Diffie-Hellman

Basta con que Alice calcule (Alice conoce cuánto vale **a** y B):

$$B^{\mathbf{a}}$$

Método de Diffie-Hellman

Basta con que Alice calcule (Alice conoce cuánto vale **a** y B):

$$B^{\mathbf{a}} \equiv (g^{\mathbf{b}})^{\mathbf{a}}$$

Método de Diffie-Hellman

Basta con que Alice calcule (Alice conoce cuánto vale **a** y B):

$$B^{\mathbf{a}} \equiv (g^{\mathbf{b}})^{\mathbf{a}} \equiv g^{\mathbf{ab}} \pmod{p}.$$

Método de Diffie-Hellman

Basta con que Alice calcule (Alice conoce cuánto vale **a** y B):

$$B^{\mathbf{a}} \equiv (g^{\mathbf{b}})^{\mathbf{a}} \equiv g^{\mathbf{ab}} \pmod{p}.$$

Similarmente, basta que Bob calcule (Bob conoce cuánto vale **b** y A):

$$A^{\mathbf{b}}$$

Método de Diffie-Hellman

Basta con que Alice calcule (Alice conoce cuánto vale **a** y B):

$$B^{\mathbf{a}} \equiv (g^{\mathbf{b}})^{\mathbf{a}} \equiv g^{\mathbf{ab}} \pmod{p}.$$

Similarmente, basta que Bob calcule (Bob conoce cuánto vale **b** y A):

$$A^{\mathbf{b}} \equiv (g^{\mathbf{a}})^{\mathbf{b}}$$

Método de Diffie-Hellman

Basta con que Alice calcule (Alice conoce cuánto vale **a** y B):

$$B^{\mathbf{a}} \equiv (g^{\mathbf{b}})^{\mathbf{a}} \equiv g^{\mathbf{ab}} \pmod{p}.$$

Similarmente, basta que Bob calcule (Bob conoce cuánto vale **b** y A):

$$A^{\mathbf{b}} \equiv (g^{\mathbf{a}})^{\mathbf{b}} \equiv g^{\mathbf{ab}} \pmod{p}.$$

Método de Diffie-Hellman

Basta con que Alice calcule (Alice conoce cuánto vale **a** y B):

$$B^{\mathbf{a}} \equiv (g^{\mathbf{b}})^{\mathbf{a}} \equiv g^{\mathbf{ab}} \pmod{p}.$$

Similarmente, basta que Bob calcule (Bob conoce cuánto vale **b** y A):

$$A^{\mathbf{b}} \equiv (g^{\mathbf{a}})^{\mathbf{b}} \equiv g^{\mathbf{ab}} \pmod{p}.$$

De estos elementos, sólo **a** y **b** son secretos:

- p = primo (público), , conocido por Alice, Bob, y cualquier otro.
- g = generador base (público), conocido por Alice, Bob, y cualquier otro.
- **a** = llave privada de Alice.
- **b** = llave privada de Bob.
- A = llave pública de Alice, conocida por Alice, Bob, y cualquier otro.
- B = llave pública de Bob's, conocida por Alice, Bob, y cualquier otro.

Seguridad

¿Por qué este método es seguro?

Seguridad

¿Por qué este método es seguro?

Cualquier persona conoce p, g, $A = g^{\mathbf{a}} \pmod{p}$ y $B = g^{\mathbf{b}} \pmod{p}$.

Seguridad

¿Por qué este método es seguro?

Cualquier persona conoce p, g, $A = g^{\mathbf{a}} \pmod{p}$ y $B = g^{\mathbf{b}} \pmod{p}$.

¿Qué tan fácil es calcular $\mathbf{k} = g^{\mathbf{ab}}$ a partir de estos datos?

Seguridad

¿Por qué este método es seguro?

Cualquier persona conoce p, g, $A = g^{\mathbf{a}} \pmod{p}$ y $B = g^{\mathbf{b}} \pmod{p}$.

¿Qué tan fácil es calcular $\mathbf{k} = g^{\mathbf{ab}}$ a partir de estos datos?

Más generalmente, definimos la función $DH_g(g^{\mathbf{a}}, g^{\mathbf{b}}) = g^{\mathbf{ab}} \pmod{p}$.

Seguridad

¿Por qué este método es seguro?

Cualquier persona conoce p, g, $A = g^{\mathbf{a}} \pmod{p}$ y $B = g^{\mathbf{b}} \pmod{p}$.

¿Qué tan fácil es calcular $\mathbf{k} = g^{\mathbf{ab}}$ a partir de estos datos?

Más generalmente, definimos la función $DH_g(g^{\mathbf{a}}, g^{\mathbf{b}}) = g^{\mathbf{ab}} \pmod{p}$. Qué tan compleja de la función DH?

Seguridad

¿Por qué este método es seguro?

Cualquier persona conoce p, g, $A = g^{\mathbf{a}} \pmod{p}$ y $B = g^{\mathbf{b}} \pmod{p}$.

¿Qué tan fácil es calcular $\mathbf{k} = g^{\mathbf{ab}}$ a partir de estos datos?

Más generalmente, definimos la función $DH_g(g^{\mathbf{a}}, g^{\mathbf{b}}) = g^{\mathbf{ab}} \pmod{p}$. Qué tan compleja de la función DH?

Supongamos que el primo p es de longitud n bits.

Seguridad

¿Por qué este método es seguro?

Cualquier persona conoce p, g, $A = g^{\mathbf{a}} \pmod{p}$ y $B = g^{\mathbf{b}} \pmod{p}$.

¿Qué tan fácil es calcular $\mathbf{k} = g^{\mathbf{ab}}$ a partir de estos datos?

Más generalmente, definimos la función $DH_g(g^{\mathbf{a}}, g^{\mathbf{b}}) = g^{\mathbf{ab}} \pmod{p}$. Qué tan compleja de la función DH?

Supongamos que el primo p es de longitud n bits. Actualmente los mejores algoritmos (GNFS, *General number field sieve*) para calcular g^{ab} corren en tiempo $\exp(\widetilde{O}(\sqrt[3]{n}))$.

Seguridad

¿Por qué este método es seguro?

Cualquier persona conoce p, g, $A = g^{\mathbf{a}} \pmod{p}$ y $B = g^{\mathbf{b}} \pmod{p}$.

¿Qué tan fácil es calcular $\mathbf{k} = g^{\mathbf{ab}}$ a partir de estos datos?

Más generalmente, definimos la función $DH_g(g^{\mathbf{a}}, g^{\mathbf{b}}) = g^{\mathbf{ab}} \pmod{p}$. Qué tan compleja de la función DH?

Supongamos que el primo p es de longitud n bits. Actualmente los mejores algoritmos (GNFS, *General number field sieve*) para calcular g^{ab} corren en tiempo $\exp(\widetilde{O}(\sqrt[3]{n}))$.

Tamaño de la clave	Tamaño del módulo
80 bits	1024 bits
128 bits	3072 bits
256 bits (AES)	15360 bits

Seguridad

¿Es posible aplicar el método de Diffie-Hellman en otros ambiente que no sea aritmética modular?

Seguridad

¿Es posible aplicar el método de Diffie-Hellman en otros ambiente que no sea aritmética modular?
Sí, y el tipo de objeto matemático es mucho más complicado: una curva elíptica.

Seguridad

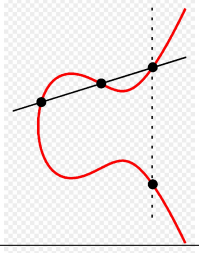
¿Es posible aplicar el método de Diffie-Hellman en otros ambiente que no sea aritmética modular?
Sí, y el tipo de objeto matemático es mucho más complicado: una curva elíptica.

Tamaño de la clave	Tamaño del módulo	Tamaño de curva elíptica
80 bits	1024 bits	160 bits
128 bits	3072 bits	256 bits
256 bits (AES)	15360 bits	512 bits

Seguridad

¿Es posible aplicar el método de Diffie-Hellman en otros ambiente que no sea aritmética modular?
Sí, y el tipo de objeto matemático es mucho más complicado: una curva elíptica.

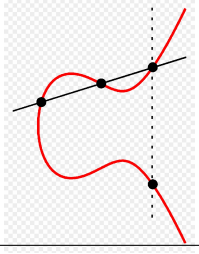
Tamaño de la clave	Tamaño del módulo	Tamaño de curva elíptica
80 bits	1024 bits	160 bits
128 bits	3072 bits	256 bits
256 bits (AES)	15360 bits	512 bits

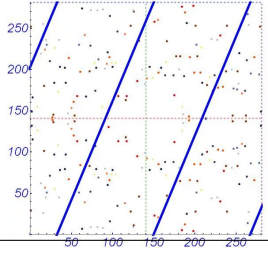


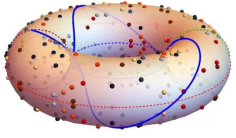
Seguridad

¿Es posible aplicar el método de Diffie-Hellman en otros ambiente que no sea aritmética modular?
Sí, y el tipo de objeto matemático es mucho más complicado: una curva elíptica.

Tamaño de la clave	Tamaño del módulo	Tamaño de curva elíptica
80 bits	1024 bits	160 bits
128 bits	3072 bits	256 bits
256 bits (AES)	15360 bits	512 bits







Curvas Elípticas

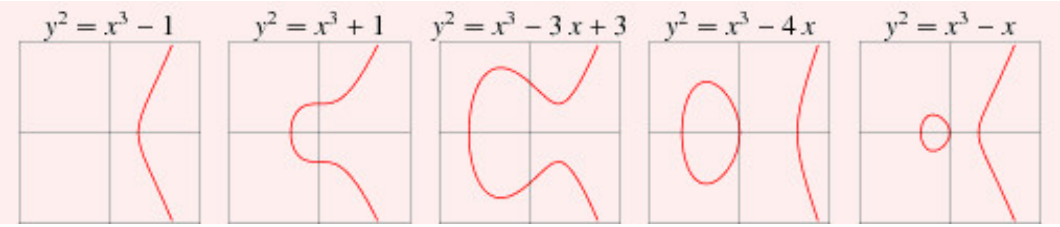
Una curva elíptica sobre $\mathbb{R}$ es una curva algebraica plana, de la forma

$$y^2 = x^3 + ax + b.$$

Curvas Elípticas

Una curva elíptica sobre $\mathbb{R}$ es una curva algebraica plana, de la forma

$$y^2 = x^3 + ax + b.$$

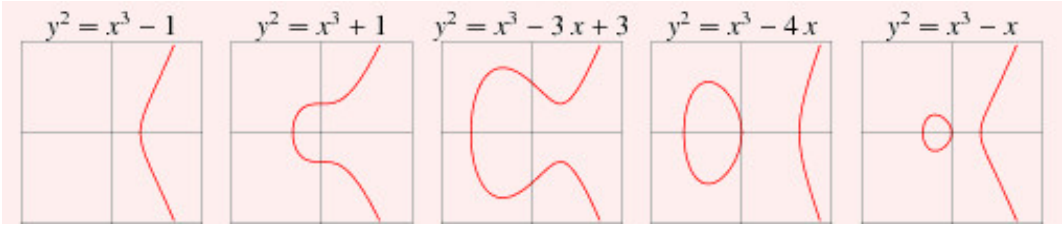


Ejemplos de curvas elípticas sobre $\mathbb{R}$.

Curvas Elípticas

Una curva elíptica sobre $\mathbb{R}$ es una curva algebraica plana, de la forma

$$y^2 = x^3 + ax + b.$$



Ejemplos de curvas elípticas sobre $\mathbb{R}$.

Los puntos de una curva elíptica E tienen la propiedad de formar un grupo. Es posible definir una "suma" de puntos sobre E, de forma que cumpla las propiedades: conmutativa, asociativa, neutro, inversos aditivos.

Curvas Elípticas

La suma de puntos sobre una curva elíptica se define de la siguiente forma: Dados dos puntos P y Q en la curva E, tomamos la recta PQ que pasa por ambos puntos, y encontramos su intersección con E.

Curvas Elípticas

La suma de puntos sobre una curva elíptica se define de la siguiente forma: Dados dos puntos P y Q en la curva E, tomamos la recta PQ que pasa por ambos puntos, y encontramos su intersección con E. Llamamos $R = -(P + Q)$ a dicho punto.

Ahora, definimos $P + Q$ como el punto que es la reflexión de R con respecto al eje x, el cual también está sobre la curva E.

Curvas Elípticas

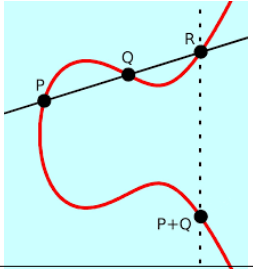
La suma de puntos sobre una curva elíptica se define de la siguiente forma: Dados dos puntos P y Q en la curva E, tomamos la recta PQ que pasa por ambos puntos, y encontramos su intersección con E. Llamamos $R = -(P + Q)$ a dicho punto.

Ahora, definimos $P + Q$ como el punto que es la reflexión de R con respecto al eje x, el cual también está sobre la curva E. Dicho punto también puede obtenerse como la intersección de la recta vertical que pasa por R y por el punto al infinito ($+\infty$ que también está sobre la curva E).

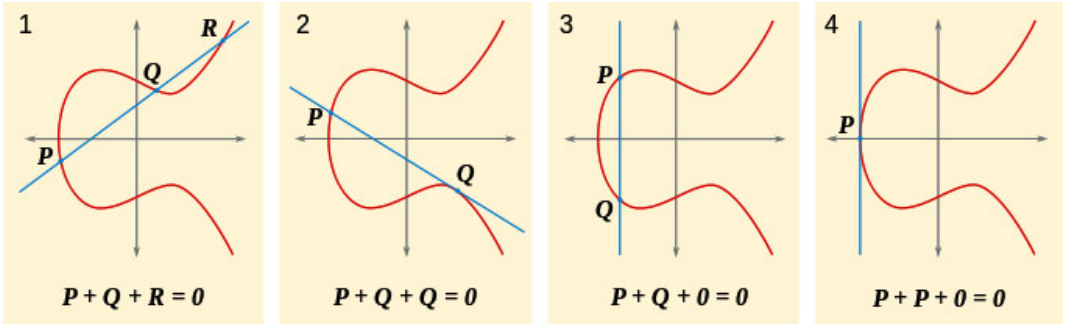
Curvas Elípticas

La suma de puntos sobre una curva elíptica se define de la siguiente forma: Dados dos puntos P y Q en la curva E, tomamos la recta PQ que pasa por ambos puntos, y encontramos su intersección con E. Llamamos $R = -(P + Q)$ a dicho punto.

Ahora, definimos $P + Q$ como el punto que es la reflexión de R con respecto al eje x, el cual también está sobre la curva E. Dicho punto también puede obtenerse como la intersección de la recta vertical que pasa por R y por el punto al infinito ($+\infty$ que también está sobre la curva E).



Curvas Elípticas



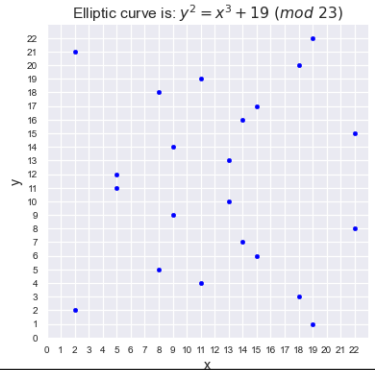
Varios ejemplos de la ley de grupo para una curva elíptica.

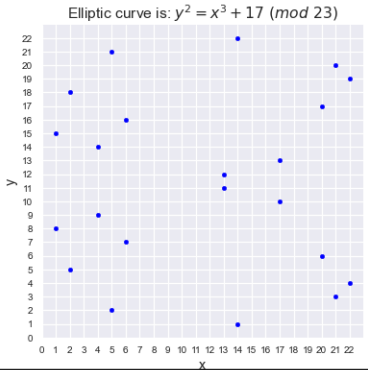
Curvas Elípticas

Cuando trabajamos sobre un cuerpo finito de números como $\mathbb{Z}/p\mathbb{Z}$, podemos también definir una curva elíptica como una relación de la forma $y^2 = x^3 + ax + b$.

Curvas Elípticas

Cuando trabajamos sobre un cuerpo finito de números como $\mathbb{Z}/p\mathbb{Z}$, podemos también definir una curva elíptica como una relación de la forma $y^2 = x^3 + ax + b$.
En este caso, la "curva elíptica" se reduce a una colección finita de puntos que vive en el plano $(\mathbb{Z}/p\mathbb{Z}) \times (\mathbb{Z}/p\mathbb{Z})$, como se ilustra en la siguiente figura.





Inseguro contra MiTM

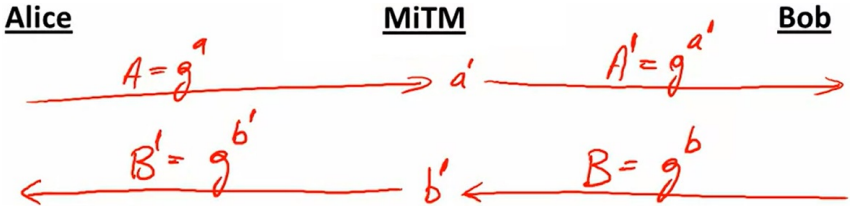
Descrito como antes, el método de Diffie-Hellman es inseguro contra ataques activos.

Inseguro contra MiTM

Descrito como antes, el método de Diffie-Hellman es inseguro contra ataques activos. En particular, contra un ataque llamado *man-in-the-middle*.

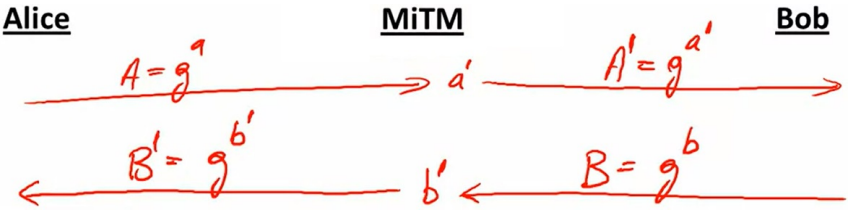
Inseguro contra MiTM

Descrito como antes, el método de Diffie-Hellman es inseguro contra ataques activos. En particular, contra un ataque llamado *man-in-the-middle*.



Inseguro contra MiTM

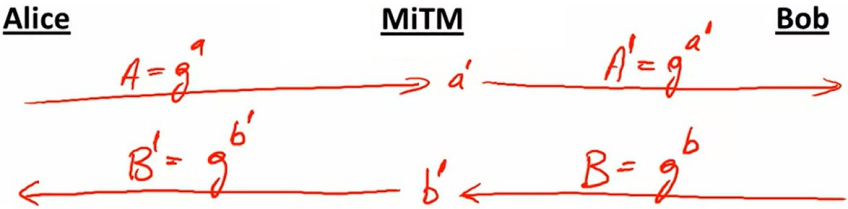
Descrito como antes, el método de Diffie-Hellman es inseguro contra ataques activos. En particular, contra un ataque llamado *man-in-the-middle*.



Ahora, Alice va a calcular $(B')^{\mathbf{a}} = (g^{\mathbf{b}'})^{\mathbf{a}} \equiv g^{\mathbf{ab}'} \pmod{p}$.

Inseguro contra MiTM

Descrito como antes, el método de Diffie-Hellman es inseguro contra ataques activos. En particular, contra un ataque llamado *man-in-the-middle*.

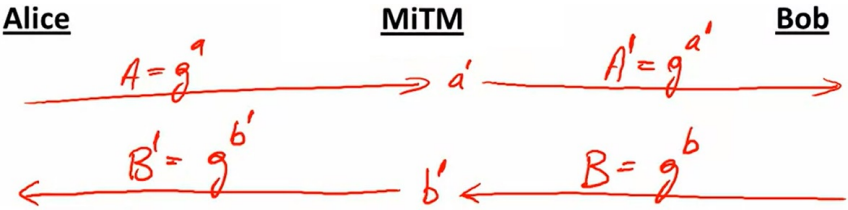


Ahora, Alice va a calcular $(B')^{\mathbf{a}} = (g^{\mathbf{b}'})^{\mathbf{a}} \equiv g^{\mathbf{ab}'} \pmod{p}$.

Bob va a calcular $(A')^{\mathbf{b}} = (g^{\mathbf{a}'})^{\equiv} g^{\mathbf{a}'\mathbf{b}} \pmod{p}$.

Inseguro contra MiTM

Descrito como antes, el método de Diffie-Hellman es inseguro contra ataques activos. En particular, contra un ataque llamado *man-in-the-middle*.



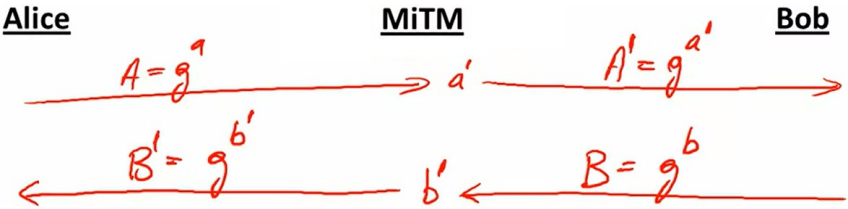
Ahora, Alice va a calcular $(B')^{\mathbf{a}} = (g^{\mathbf{b}'})^{\mathbf{a}} \equiv g^{\mathbf{ab}'} \pmod{p}$.

Bob va a calcular $(A')^{\mathbf{b}} = (g^{\mathbf{a}'})^{\equiv} g^{\mathbf{a}'\mathbf{b}} \pmod{p}$.

Estas no son la misma clave.

Inseguro contra MiTM

Descrito como antes, el método de Diffie-Hellman es inseguro contra ataques activos. En particular, contra un ataque llamado *man-in-the-middle*.



Ahora, Alice va a calcular $(B')^{\mathbf{a}} = (g^{\mathbf{b}'})^{\mathbf{a}} \equiv g^{\mathbf{ab}'} \pmod{p}$.

Bob va a calcular $(A')^{\mathbf{b}} = (g^{\mathbf{a}'})^{\equiv} g^{\mathbf{a}'\mathbf{b}} \pmod{p}$.

Estas no son la misma clave.

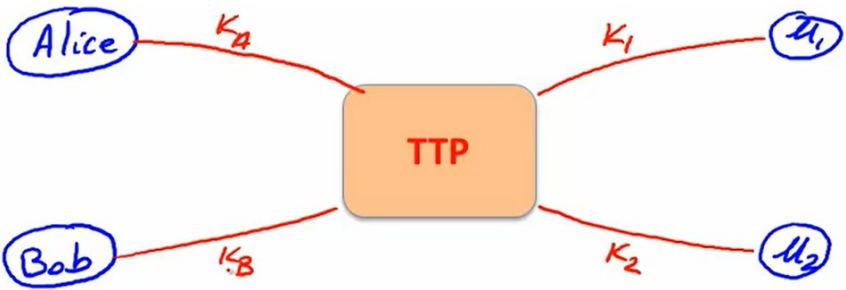
MiTM conoce $A, B, \mathbf{a}', \mathbf{b}'$, y sin problema puede calcular $g^{\mathbf{ab}'}, g^{\mathbf{a}'\mathbf{b}} \pmod{p}$.

Protocolos de Intecambio

Online Trusted Third Party (TTP):

Protocolos de Intecambio

Online Trusted Third Party (TTP):

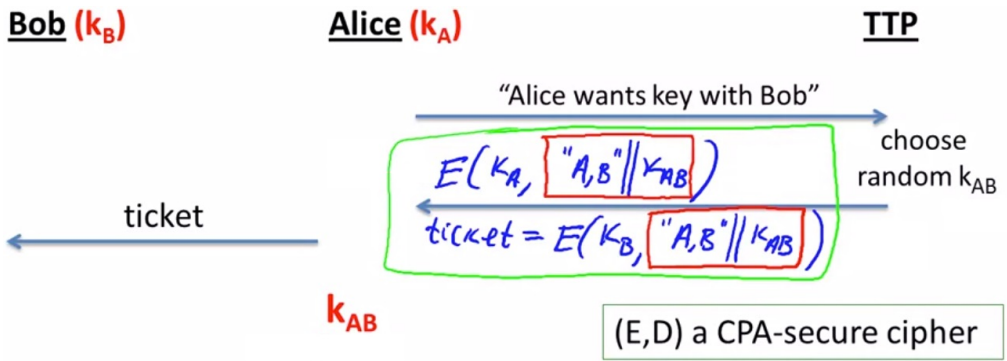


Protocolos de Intecambio

Online Trusted Third Party (TTP):

Protocolos de Intecambio

Online Trusted Third Party (TTP):



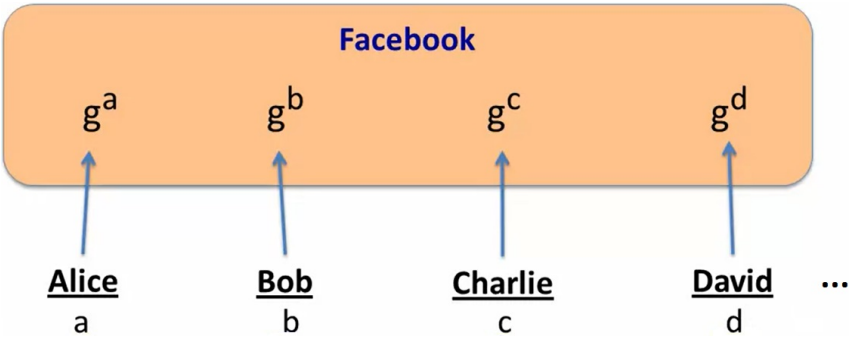
Ejemplo de un protocolo de intercambio de claves mediante una TTP.

Intercambio de Claves

Key exchange:

Intercambio de Claves

Key exchange:



Intercambio de Claves

En la práctica, se utilizan ya ciertos protocolos establecidos como seguroa para el intercambio Diffie-Helmann. Estos consisten el elegir alguno primo p con una cierta longitud de bits.

Intercambio de Claves

En la práctica, se utilizan ya ciertos protocolos establecidos como seguroa para el intercambio Diffie-Helmann. Estos consisten el elegir alguno primo p con una cierta longitud de bits.
El primo p, el generador $\mathbf{g}$, y los parámetros adicionales se pueden generar de forma aleatoria.

Intercambio de Claves

En la práctica, se utilizan ya ciertos protocolos establecidos como seguroa para el intercambio Diffie-Helmann. Estos consisten el elegir alguno primo p con una cierta longitud de bits.
El primo p, el generador **g**, y los parámetros adicionales se pueden generar de forma aleatoria.

Grupos de Diffie-Hellman:

- Grupo 1: potenciación modular con un módulo de 768-bit.
- Grupo 2: potenciación modular con un módulo de 1024-bit.
- Grupo 5: potenciación modular con un módulo de 1536-bit.
- Grupo 14: potenciación modular con un módulo de 2048-bit.
- Grupo 19: grupo de una curva elíptica aleatoria de 256-bit.
- Grupo 20: grupo de una curva elíptica aleatoria de 384-bit.
- Grupo 21: grupo de una curva elíptica aleatoria de 521-bit.
- Group 24: potenciación modular con un módulode 2048-bit y un subgrupo de orden primo de 256-bit.