

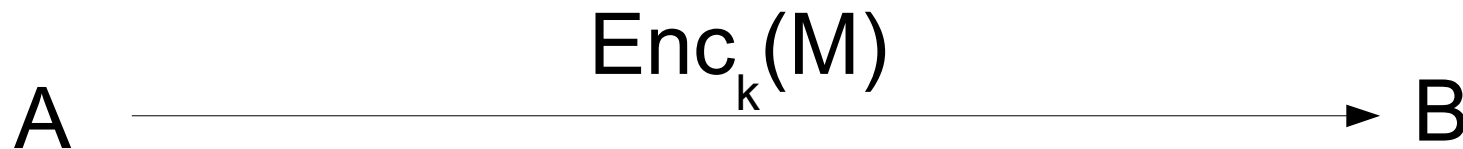


Criptografía y Seguridad

Criptografía:
Cifrado asimétrico y
Firma digital

Distribución de claves

- Un criptosistema CCA-Secure permite enviar información manteniendo:
 - Confidencialidad
 - Integridad



- Pero requiere que ambas partes conozcan una misma clave

Distribución de claves

- Las claves no pueden transmitirse por un canal inseguro.
- ¿Como se comparten las claves?
- Dos puntos → Uso de canal seguro
- Múltiples puntos →
 - Una clave por cada combinación
 - Cada parte debe administrar $n-1$ claves
 - ¿Si hay n puntos cuantas claves se necesitan?
 - Un punto único de confianza
 - También referido como Trusted third party

Distribución de claves

- KDC – Key Distribution Centers
 - Son entidades que centralizan intercambio de claves
 - Comparten una clave con cada entidad que participa (k_a, k_b, k_c, \dots)
 - Si A quiere comunicarse con C, envía un pedido al KDC
 - El KDC crea una nueva clave k_s , clave de sesión, y se la envía a A y C, cifrandola con k_a y k_c respectivamente
- Permiten tener n claves para n entidades
- Pero es un único punto de falla

Distribución de claves

- KDC – Key Distribution Centers



Criptografía asimétrica

- “We stand today on the brink of a revolution in cryptography”, Diffie-Hellman (1976).



Criptografía asimétrica

- Se basa en la asimetría de ciertos problemas
 - Por ejemplo, es fácil cerrar un candado
 - Pero se necesita una llave para abrirlo
- ¿Se puede crear un criptosistema donde existan dos contraseñas?
 - Una para cifrar, la otra para descifrar.
 - Incluso si un atacante posee la clave para cifrar, no puede recuperar el mensaje
 - Se puede publicar dicha clave a proposito !!!
 - También llamada criptografía de clave pública

Criptografía asimétrica

- Intercambio de claves (nuevo!)
 - Permite a dos partes generar una clave compartida en línea
- Cifrado asimétrico
 - Similar al cifrado simétrico
- Firma digital
 - Similar al MAC

Algebra, topología, aritmética, ...



Repaso de algebra

- Grupo algebraico: $G, +$ (conjunto y operación)
 - Clausura
 - Asociatividad
 - Neutro
 - Inverso
- Subgrupo ($G, G', +$)
 - $(G, +)$ es un grupo, $(G', +)$ es un grupo
 - G contiene a G' y $G' \neq \emptyset$
- Grupo abeliano: $G, +$
 - $(G, +)$ es un grupo algebraico
 - Conmutatividad

Repaso de algebra

- Grupos finitos cíclicos
 - $G = (\{g^n \mid n \in \mathbb{Z}\}, +)$
 - (^ es aplicación sucesiva de +)
 - Todos los grupos de tamaño n son isomorfos
 - Grupo canónico $Z_n = (\{1, 2, 3, \dots, n-1\}, +)$
 - $g^n = e$ (neutro)
 - Generador: g / g es primo relativo a n
 - Hay $\Phi(n)$ generadores
 - $\text{Ord}(g) = \#$ elementos del subgrupo ciclico
 - g es elemento primitivo si $\text{ord}(g) = n$

Repaso de algebra

- Anillo: $G, +, *$ (conjunto y dos operaciones)
 - $(G, +)$ es un grupo abeliano
 - Clausura de $*$
 - Asociatividad de $*$
 - $*$ distributiva sobre $+$
- Cuerpo o campo: $G, +, *$
 - $(G, +, *)$ es un anillo
 - Conmutatividad de $*$
 - Neutro de $*$
 - Inverso de $*$ en $\{ G - \text{neutro. de } + \}$

Repaso de algebra

- Campo de Galois (campo finito)
 - $(G, +, *)$
 - $|G| = n$
 - Todos los campos de tamaño k son isomorfos
 - Todo campo finito tiene tamaño p^n con p primo y n entero
 - Campo canónico:
 - $\mathbb{Z}_p = \{ k \mid k \in \{0, 1, \dots, p-1\}, +, * \}, p$ primo
 - Tamaño: p

Repaso de aritmética

- Aritmética modular
 - $a = b \bmod n \iff a - b = k * n$
 - Se reduce al grupo canónico: $0 \leq a < n$
- Ejemplo:
 - $2 + 8 \bmod 5 = 0$
 - $4 * 3 \bmod 7 = 5$
- Si n es primo, tenemos un campo finito (\mathbb{Z}_p)
- Si n no es primo, tenemos un anillo
 - Si consideramos solo $k / (k:n) = 1$, tenemos un grupo multiplicativo $(\mathbb{Z}_n^*, *)$

Repaso de aritmética

- Aritmética modular

- Si $(k : n) = 1 \rightarrow \exists k^{-1} / k * k^{-1} = 1 \bmod n$
- $a^{\Phi(n)} = 1 \bmod n$ ($\Phi(n)$ es el tamaño de Z_n^*)
 - Si p es primo, $a^{p-1} = 1 \bmod p$

- Cálculo de $\Phi(n)$

- $\Phi(n*m) = \Phi(n) * \Phi(m)$, si $(n : m) = 1$
- $\Phi(p^a) = p^a - p^{a-1} = p^{a-1} * (p - 1)$, si p es primo

Intercambio de claves

- Es un protocolo $\Pi(n)$, ejecutado por dos partes:
 - $\Pi: (n) \rightarrow \text{Tran}, k_a, k_b$
 - No tiene entrada (salvo el parámetro de seguridad)
 - La salida del protocolo es
 - Un conjunto de mensajes intercambiados
 - Una clave k_a conocida solo por una de las partes
 - Una clave k_b conocida solo por la otra parte
- Condición fundamental:
 - $k_a = k_b$

Seguridad frente a ataques pasivos

- Key-Exchange experiment: $KE_{A,\Pi}$
- Dado un adversario A , y una primitiva de intercambio de claves Π :

1) Se ejecuta Π , sea $k = k_a = k_b$

2) Se genera $b \leftarrow \{0, 1\}$

3) Si $b = 0 \Rightarrow k' \leftarrow \{0, 1\}^n$

Si $b = 1 \Rightarrow k' = k$

4) A obtiene Trans y k' , y emite $b' = \{0, 1\}$

- $KE_{A,\Pi} = 1$ si $b = b'$ (A gana)

Si $\Pr[KE_{A,\Pi} = 1] < 0.5 + \varepsilon \Rightarrow \Pi$ es seguro.

Intercambio Diffie-Hellman

- 1) A define G, q, g , donde G es un grupo, q el tamaño y g un generador.
- 2) A elige $x \leftarrow \mathbb{Z}_q$ y calcula $h_1 = g^x$
- 3) A envía a B: (G, q, g, h_1)
- 4) B elige $y \leftarrow \mathbb{Z}_q$ y calcula $h_2 = g^y$
- 5) B envía a A: (h_2)
- 6) A calcula $k_a = h_2^x$
- 7) B calcula $k_b = h_1^y$

Si A y B se conocen de antemano, pueden tener predefinidos (G, q, g)

La seguridad de DH

- Primero, dado g^x y g^y , no debería ser posible obtener x o y .
 - Esto se conoce como el problema del logaritmo discreto, y no tiene solución eficiente
- Condición necesaria pero no suficiente.
- Se necesita la conjetura de decisión DH:
 - Dados g , g^x y g^y , un adversario no puede distinguir g^{xy} de un valor aleatorio
 - Nota: ¡La formulación se realizó muchos años después de la publicación del algoritmo!
 - Hoy se sabe que es un problema NP-Hard

Diffie Hellman en la práctica

- La versión original requiere un canal de transmisión autenticado
 - O sea, un atacante que pueda modificar mensajes rompe la seguridad de DH
- Se complementa con firmas digitales

Criptosistema asimétrico

- Es una terna de algoritmos
 - $\text{Gen}: () \rightarrow \text{pk}, \text{sk}$ (public key, secret key)
 - Enc (cifrado): $\text{Enc}_{\text{pk}}(m)$
 - Dec (descifrado): $\text{Dec}_{\text{sk}}(c)$
- Propiedades
 - Para todo m y k válidos: $\text{Dec}_{\text{sk}}(\text{Enc}_{\text{pk}}(m))=m$

Prueba de indistinguibilidad

- Eavesdropping Indistinguishability test: $\text{Eav}_{A,\Pi}$

- Dado un adversario A , y un Criptosistema Π :

- Se genera una clave: $(pk, sk) \leftarrow K$
 - 1) A recibe pk y emite m_0 y m_1
 - 2) Se genera $b \leftarrow \{0, 1\}$
 - 3) Se calcula $c \leftarrow \text{Enc}_{pk}(m_b)$ y se le envía a A
 - 4) A emite $b' = \{0, 1\}$

- $\text{Eav}_{A,\Pi} = 1$ si $b = b'$ (A gana)

Si $\Pr[\text{Eav}_{A,\Pi} = 1] < 0.5 + \varepsilon \Rightarrow \Pi$ es indisting.

Consecuencias

- Un atacante que conoce pk
 - Puede cifrar cualquier mensaje
 - O sea, tiene acceso a la función de cifrado
- Para un criptosistema asimétrico:
 - Si Π es indistinguible para un adversario pasivo
 $\Rightarrow \Pi$ es CPA-Secure
- Recordar que:
 - CPA-Secure REQUIERE cifrado no determinístico

“Textbook” RSA

- Generación de claves
 - Elegir $p, q \leftarrow$ Números primos. $n = p \cdot q$.
 - $e \leftarrow (0, \Phi(n)) \mid \text{mcd}(e, \Phi(n)) = 1$
 - Calcular $d \mid e \cdot d \equiv 1 \pmod{\Phi(n)}$
 - $P_k = (n, e), \text{ sk} = (n, d)$
- $\text{Enc}_{p_k}(p) \equiv p^e \pmod{n}$
- $\text{Dec}_{s_k}(c) = c^d \pmod{n}$

Problemas

- Tal cual está formulado, el cifrado es determinístico ¿Que problemas trae?
- Si e y m son pequeños: $m^e < n$, y entonces se puede calcular el logaritmo
 - Durante mucho tiempo se utilizo $e=3$ para ahorrar tiempo
- Módulos repetidos
 - Si dos pares de claves comparten el mismo n , es posible recuperar n (y luego la clave privada)

Ejemplo RSA

- Parámetros muy pequeños a modo ilustrativo:
 - $p=2.357$, $q=2.551$, $n=p*q=6.012.707$
 - $\Phi(n) = (p-1)*(q-1) = 6.007.800$
 - $e=3.674.911$ (elegida al azar)
 - Usando euclides extendido: $d=422.191$
- Cifrado de $m=5.234.673$
 - $e(m) = 5.234.673^{3.674.911} \bmod 6.012.707 = 3.650.502$
- Descifrado de $m'=3.650.502$
 - $d(m') = 3.650.502^{422.191} \bmod 6.012.707 = 5.234.673$

PKCS 1 v1.5

- RSA Labs Public Key Cryptography Standard
- Define una versión con padding aleatorio de RSA:
 - Sea k la longitud de n en bytes
 - Solo se permiten cifrar mensajes de hasta $n-11$ bytes. Sea D la longitud de m en bytes
 - $m' = 00000000 \parallel 00000010 \parallel r \parallel 00000000 \parallel m$
 - Donde $r = k - D - 3$ bytes aleatorios $\neq 0$
 - La condición es para evitar ambigüedades
- Se cree que es CPA – Secure
 - Pero se encontraron ataques que muestran que no es CCA-Secure

Tamaño de claves

$$n = p * q \text{ (rsa-2048)}$$

251959084756578934940271832400483985714292821262
040320277771378360436620207075955562640185258807
844069182906412495150821892985591491761845028084
891200728449926873928072877767359714183472702618
963750149718246911650776133798590957000973304597
488084284017974291006424586918171951187461215151
726546322822168699875491824224336372590851418654
620435767984233871847744479207399342365848238242
811981638150106748104516603773060562016196762561
338441436038339044149526344321901146575444541784
240209246165157233507787077498171257724679629263
863563732899121548314381678998850404453640235273
81951378636564391212010397122822120720357

El gamal (basado en DH)

- Generación de claves
 - Seleccionar G , q , g (campo G de tamaño q)
 - $x \leftarrow \mathbb{Z}_q$, $h = g^x$
 - $pk = (G, q, g, h)$, $sk = (G, q, g, x)$
- $Enc_{pk}(m)$: $y \leftarrow \mathbb{Z}_q$, salida: $c=(c_1, c_2)=(g^y, h^y * m)$
- $Dec_{sk}(c) = c_2 / c_1^x$
- Resultados conocidos:
 - Si la prueba de decisión DH es difícil en G , El gamal es CPA-Secure

Diferencias con RSA

- El Gamal es probabilístico
- Permite reutilizar los parámetros G , q , g
- No está limitado a campos numéricos
 - Permite usar anillos de polinomios
 - Tienen 2^n elementos → Fácil mapear mensajes
 - Permite usar curvas elípticas
 - El problema de decisión DH es más complejo

Ejemplo El Gamal

- (con parámetros muy pequeños)
 - $G = Z_q^*$, $q = 2.357$, $g = 2$
 - $x = 1.751 \rightarrow g^x \bmod q = 2^{1.751} \bmod 2.357 = 1185$
- Cifrado de $m=2.035$
 - Seleccionar $y = 1.520$ (aleatorio)
 - $e(m) = (2^{1.520} \bmod 2.357, 2.035 * 1.185^{1.520} \bmod 2.357)$
 - $e(m) = (1.430, 697)$
- Descifrado de $m'=(1.430, 697)$
 - $d(m') = 1.430^{-1.751} * 697 \bmod 2.357 = 2.035$

Cifrado asimétrico en números

- El nivel de seguridad es relativo a los tamaños de los conjuntos involucrados
 - En RSA: $n=p*q$
 - En El Gamal: $n=q$
- Cuando se utilizan campos numéricos, $n \geq 1024$ bits
 - En la actualidad se recomiendan 1536 o 2048 bits
- Para campos de otro tipo, los números pueden variar
 - Por ejemplo, sobre curvas elípticas, $n \geq 320$ bits

Firmas digitales

- Similares a los MACs
 - Su objetivo es la integridad
- Pero tienen ciertas ventajas:
 - Publicamente verificables
 - Es transferible: Puede enviarse simultáneamente a varios destinatarios o reenviarse y sigue siendo verificable
 - Proveen no repudio: Quien firma no puede negar haberlo hecho
 - Propiedad muy importante cuando está reglamentada jurídicamente

Firma digital

- Es una terna de algoritmos
 - Gen: $(n) \rightarrow k = (sk, pk)$
 - Sign (firma): $s \leftarrow \text{Sign}_{sk}(m)$
 - Vrfy (verificación): $b = \text{Vrfy}_{pk}(m, s)$
- Propiedades
 - Para todo m y k válidos: $\text{Vrfy}_{pk}(m, \text{Sign}_{sk}(m)) = 1$

Seguridad de una firma digital

- $\text{Sig-forge}_{A,\Pi}$
- Dado un nivel de seguridad n , un adversario A , y una firma digital $\Pi(n)$:

- 1) Se genera una clave $k=(sk, pk) \leftarrow K$
- 2) A obtiene $f(x)=\text{Sign}_{sk}(x)$ y pk
- 3) A realiza las evaluaciones que quiera de $f(x)$
(Llamese Q al conjunto de las evaluaciones)
- 4) A emite (m, s)

- $\text{Sig-forge}_{A,\Pi} = 1$ si $\text{Vrfy}_{pk}(m, s) = 1$ y m no pertenece a Q

RSA-Signature

- Identico a RSA-Encryption, pero invirtiendo los papeles de las claves:
- Generación de claves
 - Elegir $p, q \leftarrow$ Números primos. $n = p * q$.
 - $d \leftarrow (0, \Phi(n)) \mid \text{mcd}(e, \Phi(n)) = 1$
 - Calcular $d / e * d \equiv 1 \text{ mod } \Phi(n)$
 - $pk = (n, e), sk = (n, d)$
- $\text{Sign}_{sk}(m) \equiv m^d \text{ mod } n$
- $\text{Vrfy}_{pk}(m, s) = ¿m = s^e \text{ mod } n?$

Este esquema,
aunque común en
la literatura, es inseguro

Problemas de RSA-Signature

- Considerar el adversario A:
 - $s \leftarrow S$ (selecciona una firma al azar)
 - Calcula $m = s^e \bmod n$
 - Emite (m, s)
- ¡Consigue pasar exitosamente la prueba de falsificación!
- Otro ataque:
 - $s_1 = f(m_1), s_2 = f(m_2)$
 - Emite $(m_1 * m_2, s_1 * s_2)$
 - Ejercicio: Verificar que la salida pasa la verificación

Hashed RSA

- Busca solucionar los problemas anteriores:
- Introduce una función de hash libre de colisiones
 - $\text{Sign}_{sk}(m) \equiv H(m)^d \bmod n$
 - $\text{Vrfy}_{pk}(m, s) = \text{¿} H(m) = s^e \bmod n \text{?}$
- Pero no posee una prueba de seguridad a menos que se asuma un modelo ideal de H

Digital Signature Standard

- Generación de claves
 - Seleccionar $H(x)$: SHA1 o SHA2
 - Tamaño de claves: (L, N) : $(1024, 160)$, $(2048, 224)$, $(2048, 256)$ o $(3072, 256)$.
 - $q \leftarrow$ primo de tamaño N (bits)
 - $p \leftarrow$ primo de tamaño P / $(p - 1) = 0 \bmod q$
 - $g \leftarrow$ generador de orden $q \bmod p$
 - $g^{(p-1)/q} \neq 1$
 - Seleccionar p, q, g ($G = \mathbb{Z}_p^*$)
 - $x \leftarrow \mathbb{Z}_q, y = g^x \bmod p$
 - $pk = (p, q, g, y), sk = (p, q, g, x)$

Digital Signature Standard

- $\text{Sign}_{sk}(m)$: $k \leftarrow Z_q$, $r = (g^k \bmod p) \bmod q$
 - $s = [H(m) + x*r] * k^{-1} \bmod q$
 - $\text{Sign}_{sk}(m) = (r, s)$
- $\text{Vrfy}_{pk}(m, (r, s))$: $v1 = [H(m) * s^{-1}] \bmod q$
 - $v2 = r*s^{-1} \bmod q$
 - $\checkmark r = g^{u1} * y^{u2} \bmod p \bmod q?$

Lectura Recomendada

Capítulos 9-12

Introduction to Modern Cryptography
Katz & Lindell