DECSAI
Departamento de Ciencias de la Computación e I.A.
Universidad de Granada

Seguridad

© Fernando Berzal, berzal@acm.org

Seguridad

- Motivación
- Criptografía
  - Criptografía de clave privada
  - Criptografía de clave pública
- Funciones hash
- Firmas digitales y certificados
- Autentificación
- Comunicaciones seguras
- Seguridad en Internet
- Seguridad en aplicaciones web
- Ataques DoS [denial-of-service]
- Seguridad en sistemas distribuidos

Seguridad: Motivación

"COMPUTER security is a contradiction in terms."

The Economist
Apr 8th 2017

Seguridad: Motivación

Motivos por los que alguien podría causar problemas de seguridad:

Adversary	Goal
Student	To have fun snooping on people's e-mail
Cracker	To test out someone's security system; steal data
Sales rep	To claim to represent all of Europe, not just Andorra
Businessman	To discover a competitor's strategic marketing plan
Ex-employee	To get revenge for being fired
Accountant	To embezzle money from a company
Stockbroker	To deny a promise made to a customer by e-mail
Con man	To steal credit card numbers for sale
Spy	To learn an enemy's military or industrial secrets
Terrorist	To steal germ warfare secrets

Seguridad: Aspectos

Aspectos por los que se necesita seguridad:

- Privacidad de la información (p.ej. evitar intrusos).
- Libertad de expresión.
- Derechos de autor (copyright).
- Autentificación (origen y destino fiables).
- Integridad
  (el mensaje ha de recibirse tal como se originó).
- No repudiación
  (una vez enviado un mensaje, el usuario no puede negar su autoría, p.ej. transacciones comerciales).

Seguridad

SEGURIDAD

=

Confidencialidad

+

Integridad

+

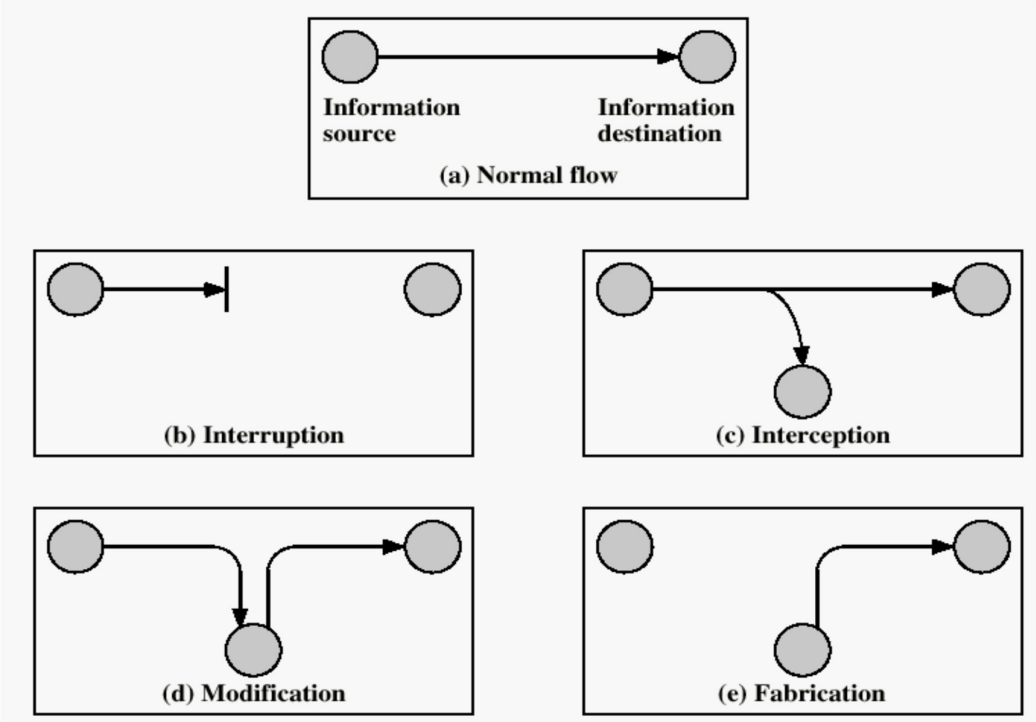
Disponibilidad

+

Autentificación

Seguridad: Ataques

Riesgos (amenazas) ➔ Identificación de vulnerabilidades



Seguridad: Ataques

Ataques pasivos

Difíciles de detectar,
si bien pueden tomarse medidas preventivas.

- **Escuchas** *[eavesdropping]*

  Objetivo: Obtener información.

  Mecanismo: Análisis del tráfico
  (frecuencia y naturaleza de los mensajes).

  vg: Sniffers, scanners, crackers…

Seguridad: Ataques

Ataques activos

"Fáciles" de detectar, aunque difíciles de prevenir:

- **Masquerading = Spoofing**
  (pretender ser quien no se es)

  vg: Direcciones IP (DNS),
  números de secuencias (TCP),
  ataques por repetición [replay],
  MIM [Man in the Middle]…

Seguridad: Ataques

Ataques activos

"Fáciles" de detectar, aunque difíciles de prevenir:

- **Alteración de datos**

  vg: WWW

- **Denegación de servicio** *[denial of service]*

  vg: Ping-of-death, smurf, spam, DDoS (TCP SYN)...

- **Ingeniería social**

Seguridad: "Malware"

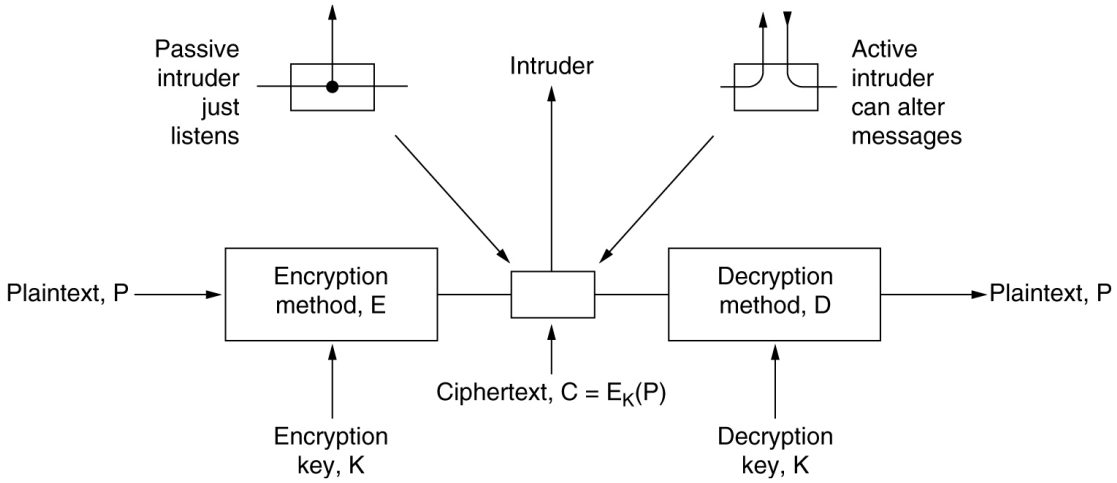
- Virus
- Gusanos [worms]
- Caballos de Troya
- Puertas traseras [trap doors]
- Bombas lógicas

Seguridad

Timeline of Emergence of Security Technologies

	1960s	1970s	1980s	1990s	2000s
emerging concerns	Interactive computing. Time sharing. User authentication. File sharing via hierarchical file systems. Prototypes of "computer utilities."	Packet networks (ARPANET). Local networks (LANs). Communication secrecy and authentication. Object-oriented design. Multilevel security. Mathematical models of security. Provably secure systems.	Adoption of TCP/IP protocols for the Internet. Exponential growth of Internet. Proliferation of PCs and workstations. Client-server model for network services. Viruses, worms, Trojans, and other forms of malware. Buffer overflow attacks.	World Wide Web. Browsers. Commercial transactions. Data repositories and breaches. Portable apps and scripts. Internet fraud. Web-based attacks. Social engineering and phishing attacks. Peer-to-peer (P2P) networks.	Botnets. Denial-of-service attacks. Wireless networks. Cloud platforms. Massive data breaches. Ransomware. Malicious adware. Internet of Things. Surveillance. Cyber warfare.
security technologies	Access controls Passwords Supervisor state	Public-key cryptography Cryptographic protocols Cryptographic hashes Security verification	Malware detection (antivirus) Intrusion detection Firewalls	Virtual private networks (VPNs) Public-key infrastructure (PKI) Secure web connections (SSL/TLS) Biometrics 2-factor authentication Confinement (virtual machines, sandboxes)	Secure coding and development processes Threat intelligence and sharing Adware blocking Denial-of-service mitigation WiFi security

Criptografía



Sistema criptográfico de clave secreta.

Criptografía

Ejemplo: Cifrado por transposición

M	E	G	A	B	U	C	K
7	4	5	1	2	8	3	6
p	l	e	a	s	e	t	r
a	n	s	f	e	r	o	n
e	m	i	l	l	i	o	n
d	o	l	l	a	r	s	t
o	m	y	s	w	i	s	s
b	a	n	k	a	c	c	o
u	n	t	s	i	x	t	w
o	t	w	o	a	b	c	d

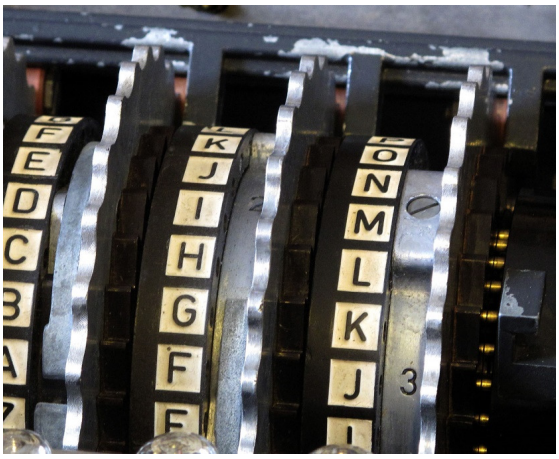
Plaintext

pleasetransferonemilliondollarsto
myswissbankaccountsixtwotwo

Ciphertext

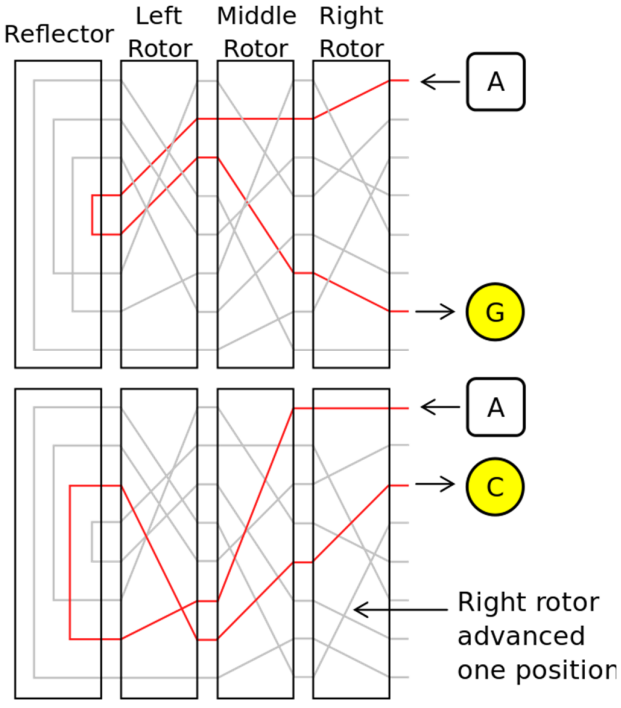
AFLLSKSOSELAWAIATOOSSCTCLNMOMANT
ESILYNTWRNNTSOWDPAEDOBUOERIRICXB

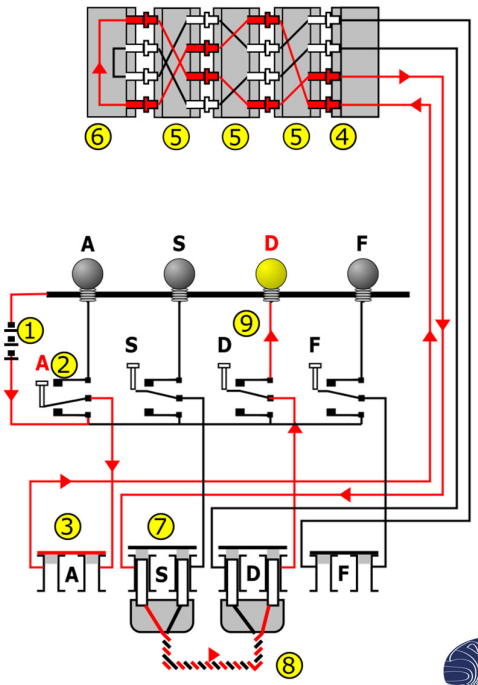
Criptografía



Máquina Enigma https://en.wikipedia.org/wiki/Enigma_machine

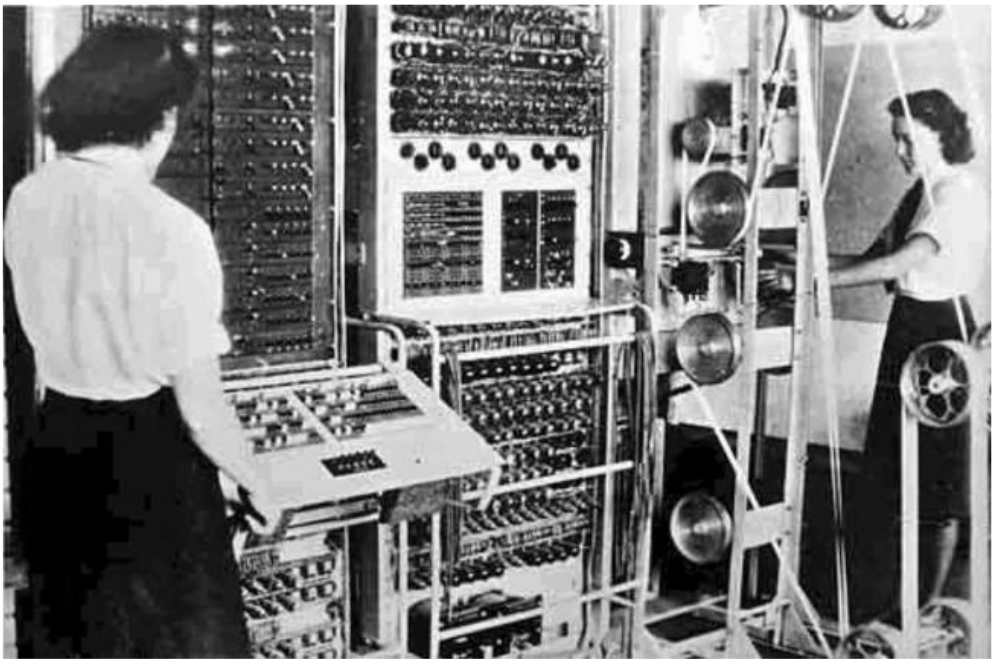
Criptografía





Máquina Enigma https://en.wikipedia.org/wiki/Enigma_machine

Criptografía



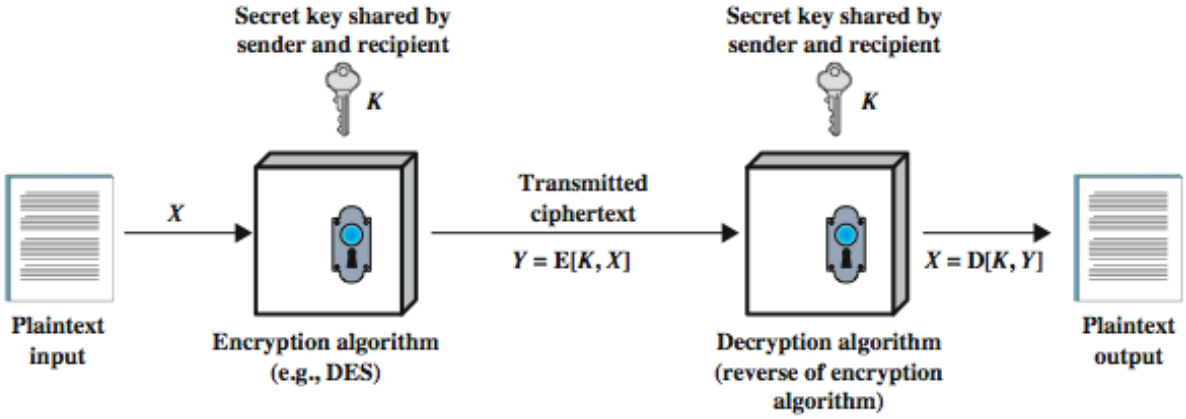
Decodificador de Enigma (Bletchley Park)

Criptografía

Posibles ataques:

- **Criptoanálisis** (basado en el conocimiento de los algoritmos de cifrado y de las características generales de los mensajes).
- **Fuerza bruta** (se analizan todas las posibilidades hasta que se consiga algo).

Criptografía de clave secreta

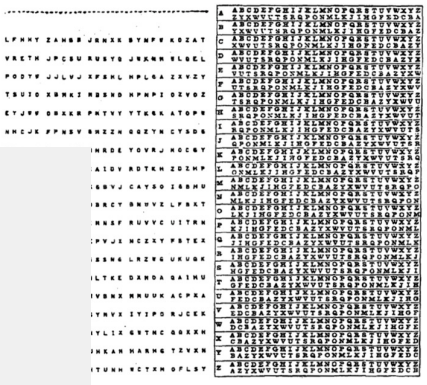


Requisito:

Aunque sea conocido el algoritmo de cifrado, debe ser difícil descifrar el mensaje (aun disponiendo de muchos textos cifrados).

Criptografía de clave secreta

OTP [One-time pad]



Encryption

```
  00101010 Plaintext
⊕ 11000001 Secret Key
  11101011 Cyphertext
```

Decryption

```
  11101011 Cyphertext
⊕ 11000001 Secret Key
  00101010 Plaintext
```


Frank Miller - Inventor of the One-Time Pad

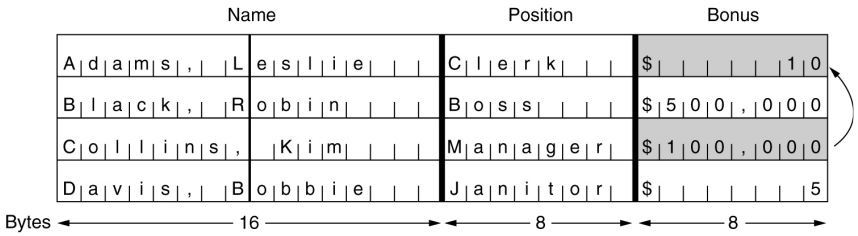
https://en.wikipedia.org/wiki/One-time_pad

Criptografía de clave secreta

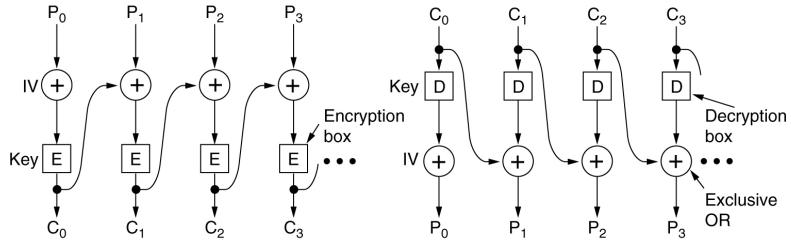
Algoritmos de cifrado en bloque

Bloques de texto de tamaño fijo ⇒ Bloques de texto cifrado.

- Modo ECB [Electronic Code Book]



- Modo CBC [Cipher Block Chaining]:
  Cifrado con encadenamiento

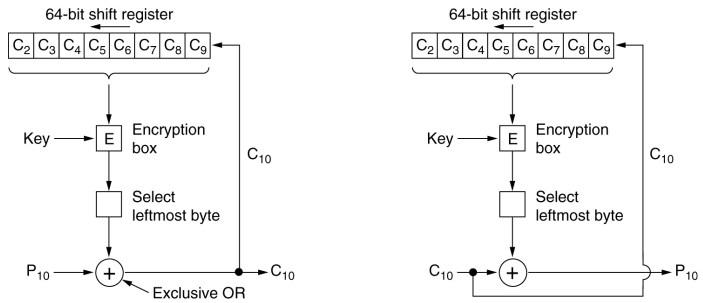


Criptografía de clave secreta

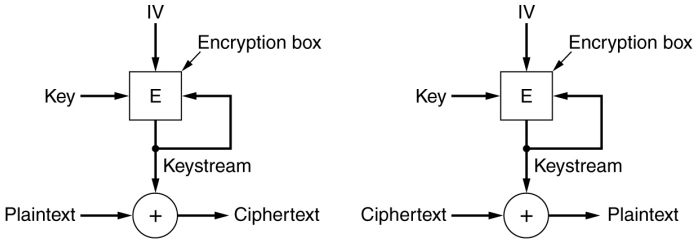
Algoritmos de cifrado en bloque

Bloques de texto de tamaño fijo ⇒ Bloques de texto cifrado.

- Modo CF [Cipher Feedback]: Cifrado con realimentación



- Modo SC [Stream Cipher]



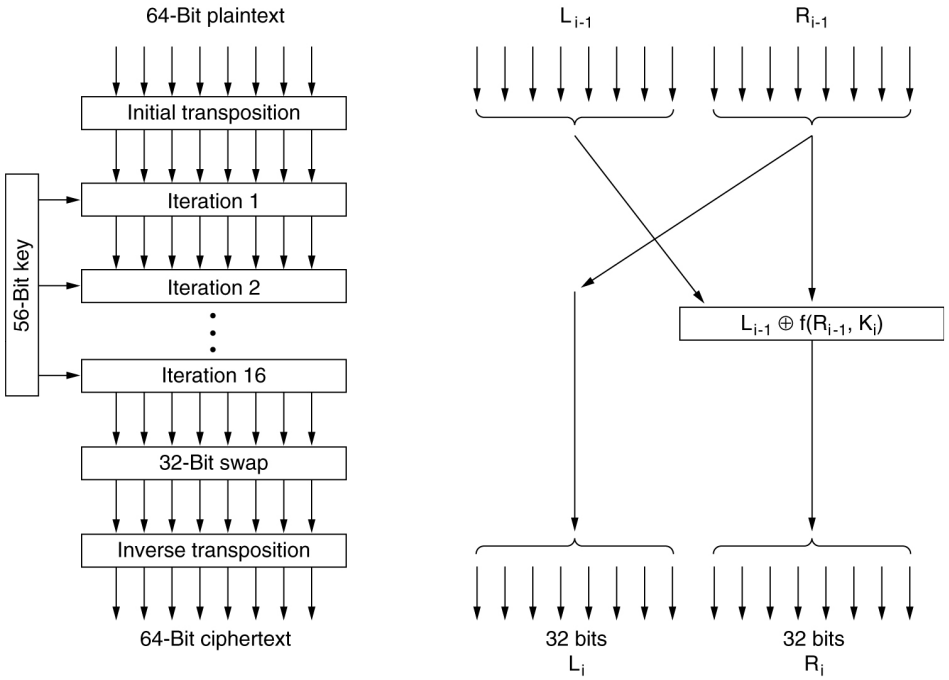
Criptografía de clave secreta

Cipher	Author	Key length	Comments
Blowfish	Bruce Schneier	1–448 bits	Old and slow
DES	IBM	56 bits	Too weak to use now
IDEA	Massey and Xuejia	128 bits	Good, but patented
RC4	Ronald Rivest	1–2048 bits	Caution: some keys are weak
RC5	Ronald Rivest	128–256 bits	Good, but patented
Rijndael	Daemen and Rijmen	128–256 bits	Best choice
Serpent	Anderson, Biham, Knudsen	128–256 bits	Very strong
Triple DES	IBM	168 bits	Second best choice
Twofish	Bruce Schneier	128–256 bits	Very strong; widely used

Criptografía de clave secreta

DES *[Data Encryption Standard], estándar USA*

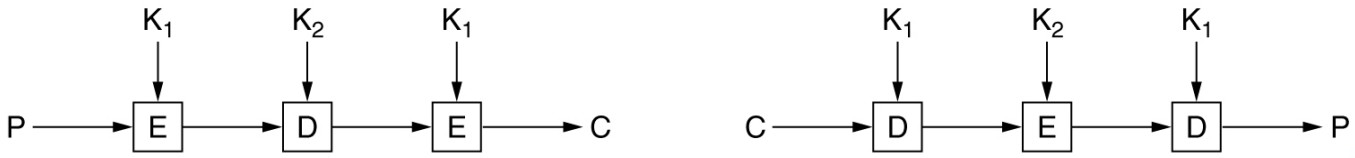
Bloques de 64 bits, clave de 56 bits (inseguro idesde 1998!)



Criptografía de clave secreta

Triple DES *[Data Encryption Standard]*

ANSI X9.17 (1985): 168 bits de clave

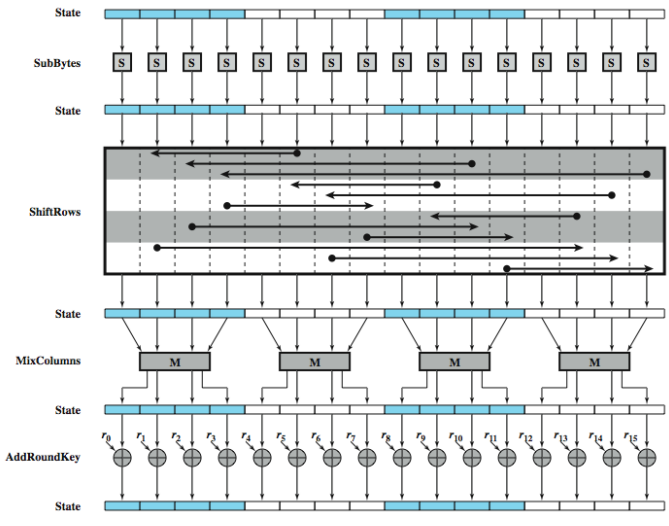


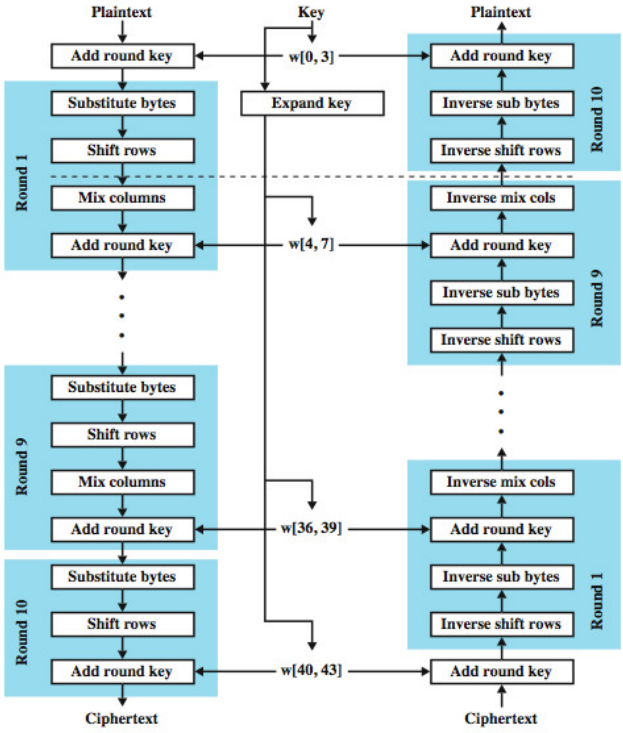
Criptografía de clave secreta

AES *[Advanced Encryption Standard]*

NIST (1997)

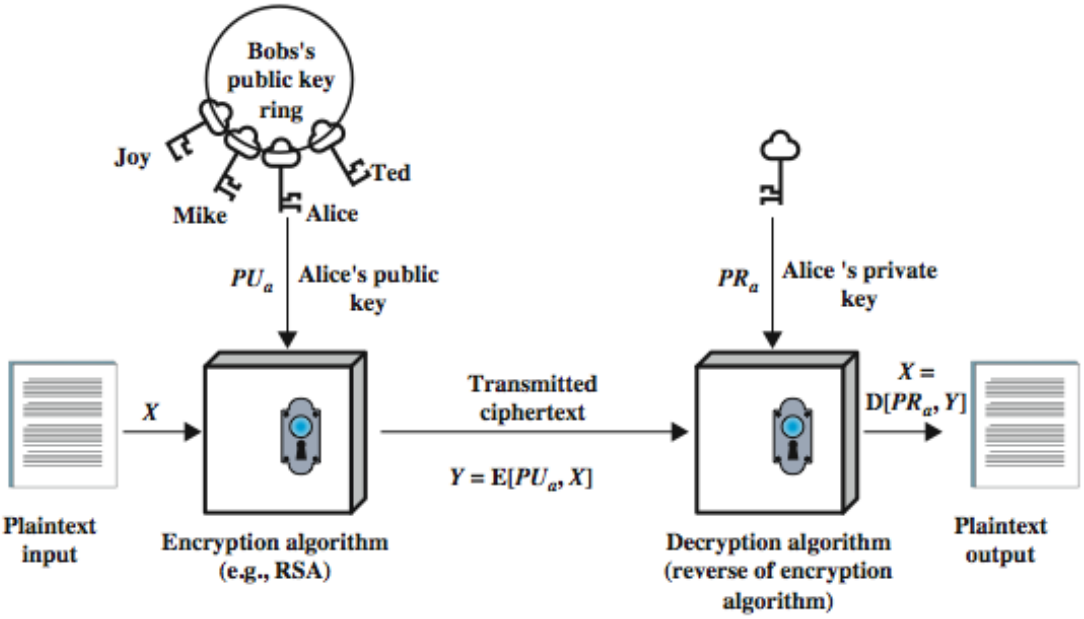
FIPS 197 (2001)





Criptografía de clave pública

Sistema asimétrico con dos claves:



Criptografía de clave pública

Requisitos:

- Debe ser fácil crear un par (clave pública, clave privada).
- Debe existir un algoritmo eficiente para cifrar el texto usando una clave y descifrarlo usando la otra.
- Debe dificultarse al máximo la posibilidad de descubrir la clave privada conociendo la clave pública.
- Debe ser difícil descifrar el texto si sólo disponemos de la clave que se utilizó para cifrarlo y el texto cifrado.

Criptografía de clave pública

Key Generation

Select p, q	p and q both prime
Calculate $n = p \times q$	
Calculate $\phi(n) = (p-1)(q-1)$	
Select integer e	$\gcd(\phi(n), e) = 1;\ 1 < e < \phi(n)$
Calculate d	$d = e^{-1} \bmod \phi(n)$
Public key	$KU = \{e, n\}$
Private key	$KR = \{d, n\}$

Encryption

Plaintext:	$M < n$
Ciphertext:	$C = M^e \pmod n$

Decryption

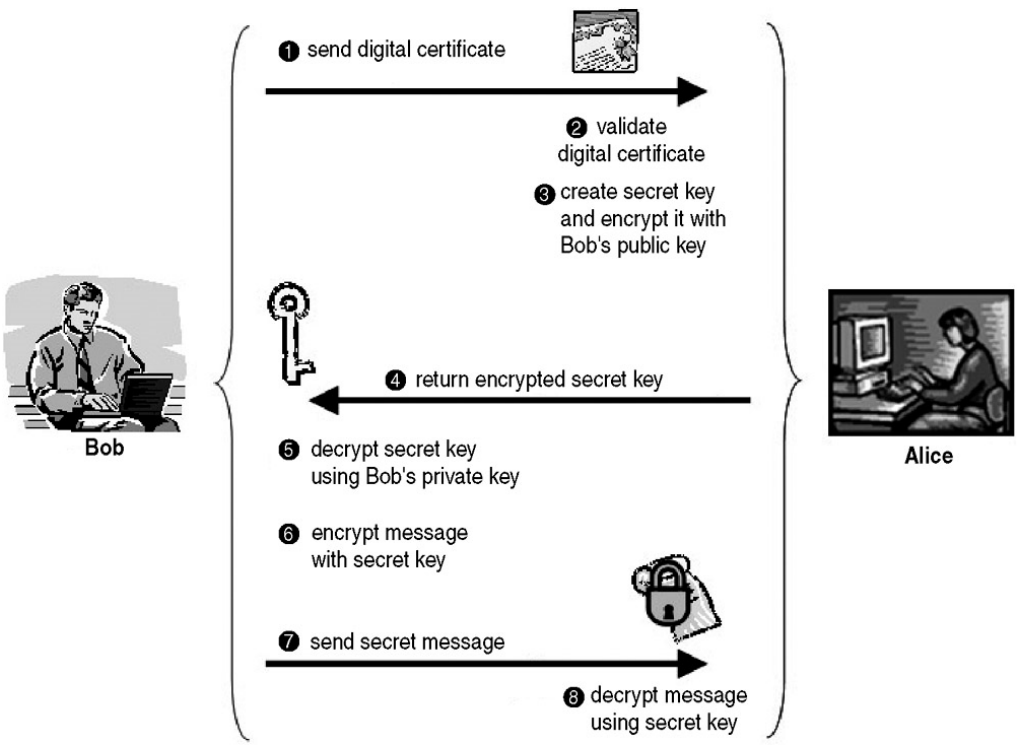
Ciphertext:	C
Plaintext:	$M = C^d \pmod n$

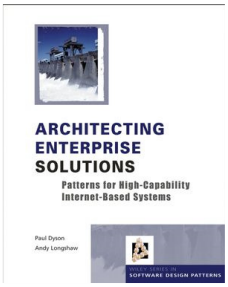
RSA

Rivest, Shamir & Adleman
MIT, 1977

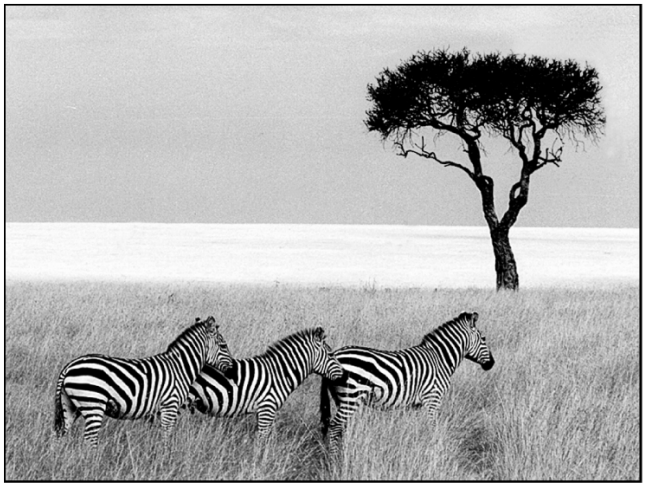
Plaintext (P)			Ciphertext (C)		After decryption	
Symbolic	Numeric	P^3	$P^3 \pmod{33}$	C^7	$C^7 \pmod{33}$	Symbolic
S	19	6859	28	13492928512	19	S
U	21	9261	21	1801088541	21	U
Z	26	17576	20	1280000000	26	Z
A	01	1	1	1	01	A
N	14	2744	5	78125	14	N
N	14	2744	5	78125	14	N
E	05	125	26	8031810176	05	E
			Sender's computation		Receiver's computation	

Criptografía de clave pública

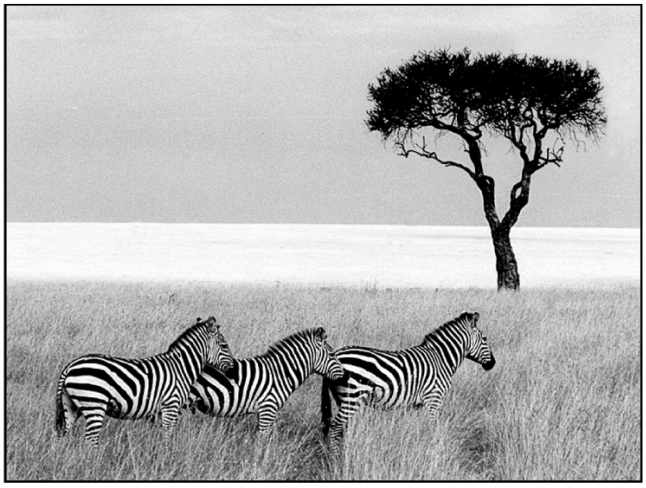




Esteganografía



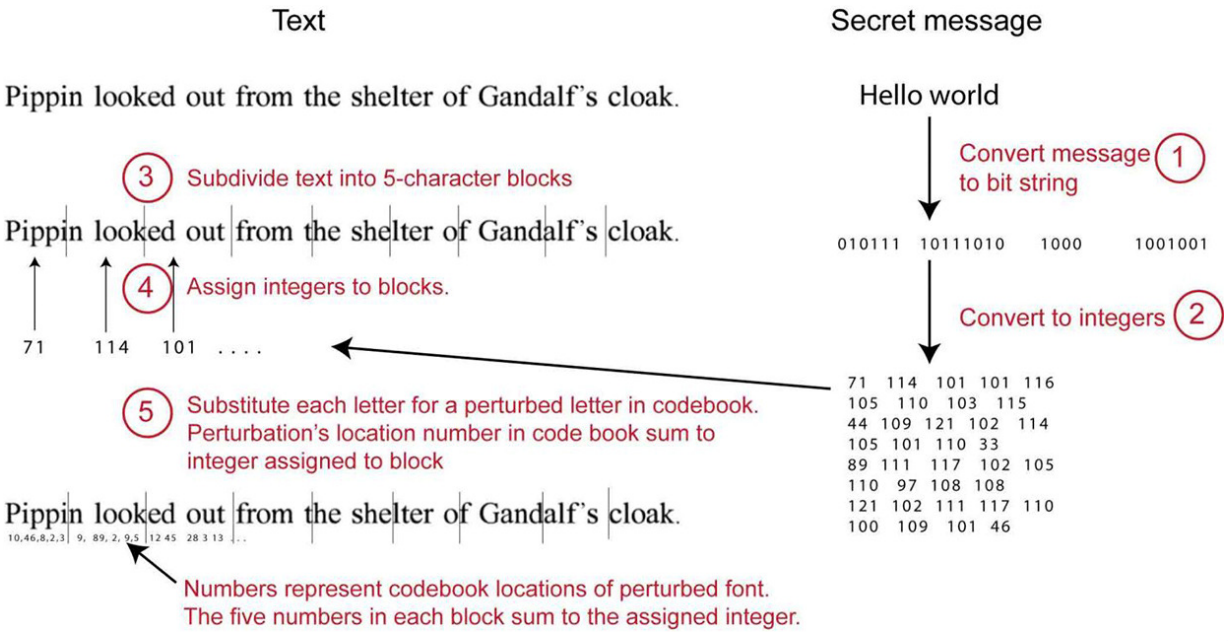
Una foto…



… con cinco obras de Shakespeare.

Esteganografía

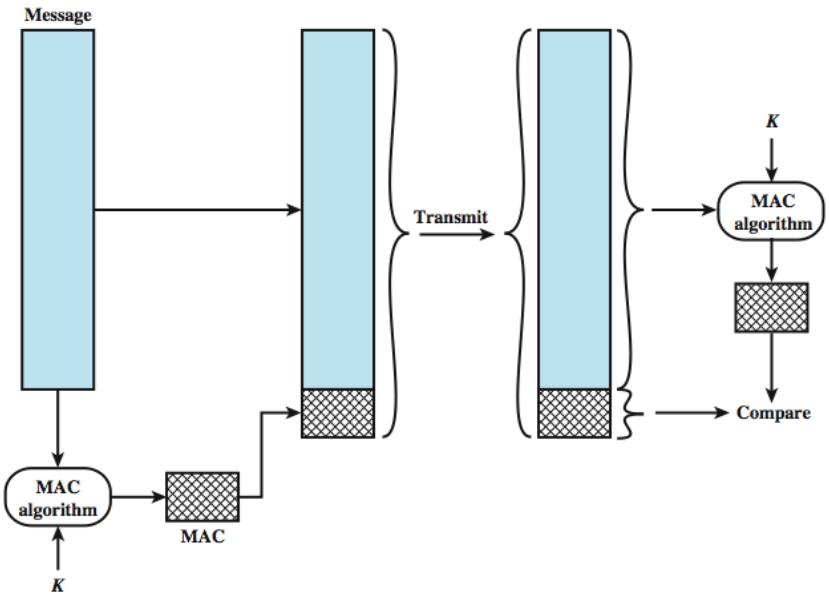
FontCode



"Hiding Information in Plain Text"
IEEE Spectrum, May 2018 & SIGGRAPH'2018
https://spectrum.ieee.org/tech-talk/computing/software/hiding-information-in-plain-text

Funciones hash

Message digests = Message Authentication Codes [MAC]



vg: MD5 [Message Digest 5],
SHA-1 [Secure Hash Algorithm – 1]

Funciones hash

Funciones hash tradicionales

- MD5 [Message Digest 5], retirado en 2008 después de que se expusiera un grave problema de seguridad.
  "2013 attack by Xie Tao, Fanbao Liu, and Dengguo Feng breaks MD5 collision resistance in 2^{18} time. This attack runs in less than a second on a regular computer."
  https://en.wikipedia.org/wiki/MD5

- SHA-1 [Secure Hash Algorithm 1], de 1995, sustituido por su sucesor, SHA-2, en 2017:
  "The cost to spoof an SHA-1 hash function today is estimated to be around $100,000—a number that will continue to drop."
  @ December 2015
  https://en.wikipedia.org/wiki/SHA-1

Funciones hash

Funciones hash actuales

- SHA-2 [Secure Hash Algorithm 2], publicado en 2001, incluye 6 funciones con valores hash de distinto número de bits, p.ej. SHA-256 o SHA-512. https://en.wikipedia.org/wiki/SHA-2
- SHA-3 (estándar NIST, 2015)… https://en.wikipedia.org/wiki/SHA-3

Funciones hash

Otras funciones hash

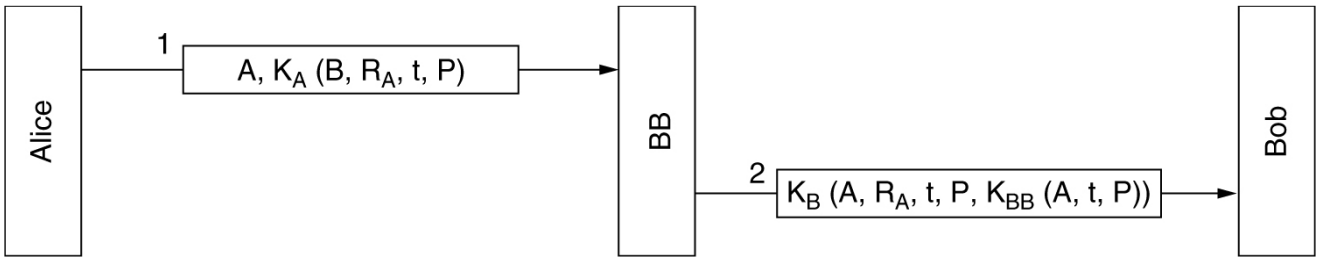
(p.ej. utilizadas para almacenar contraseñas)

- Argon2 (ganador del Password Hashing Competition, 2015, https://password-hashing.net/)
- PBKDF2 [Password-Based Key Derivation Function 2], RFC 2898, https://en.wikipedia.org/wiki/PBKDF2
- scrypt, https://en.wikipedia.org/wiki/Scrypt
- bcrypt, https://en.wikipedia.org/wiki/Bcrypt

Firmas digitales

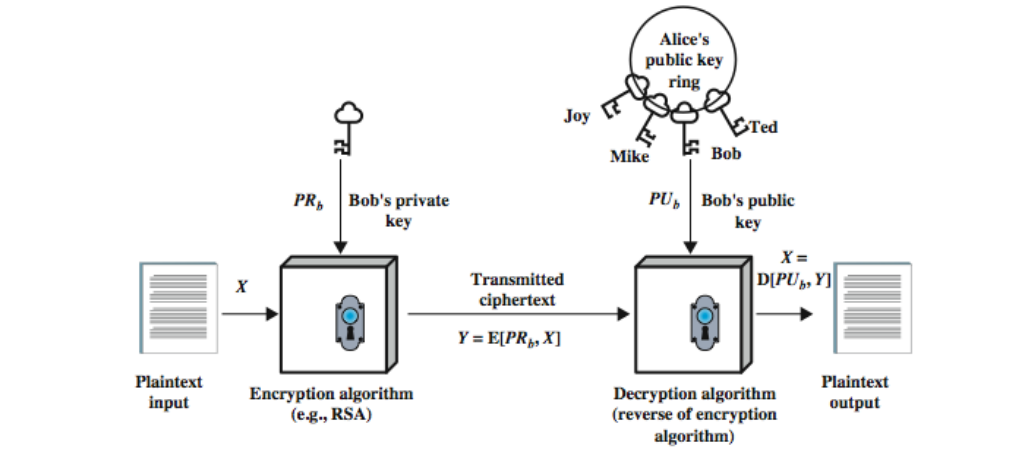
Firmas de clave simétrica

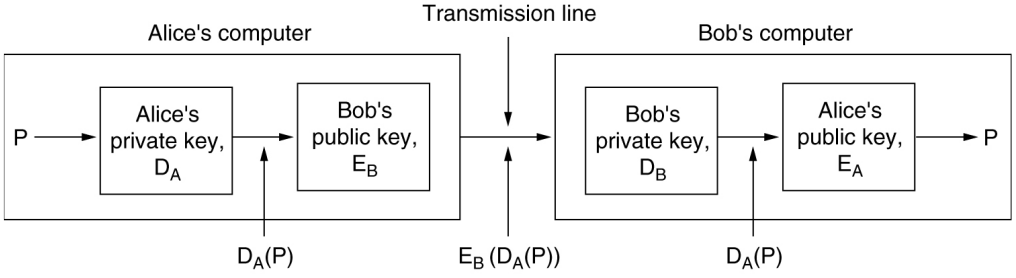
Gran Hermano ("Big Brother")



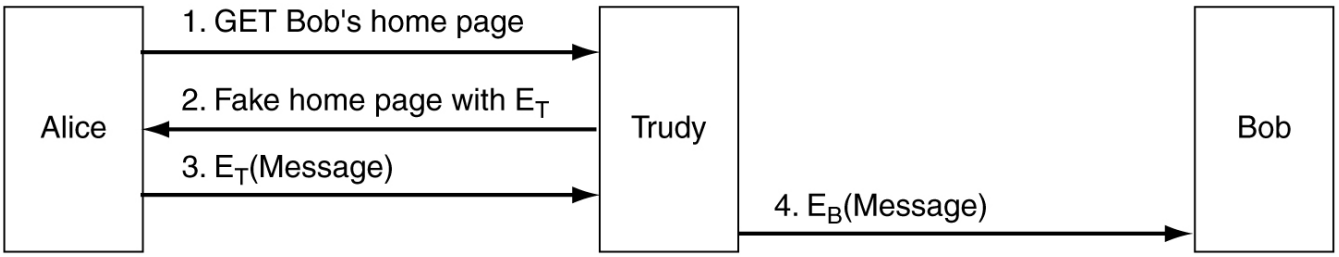
Firmas digitales

Firmas digitales con criptografía de clave pública





Certificados



Intruso en un sistema criptográfico de clave pública

Certificados

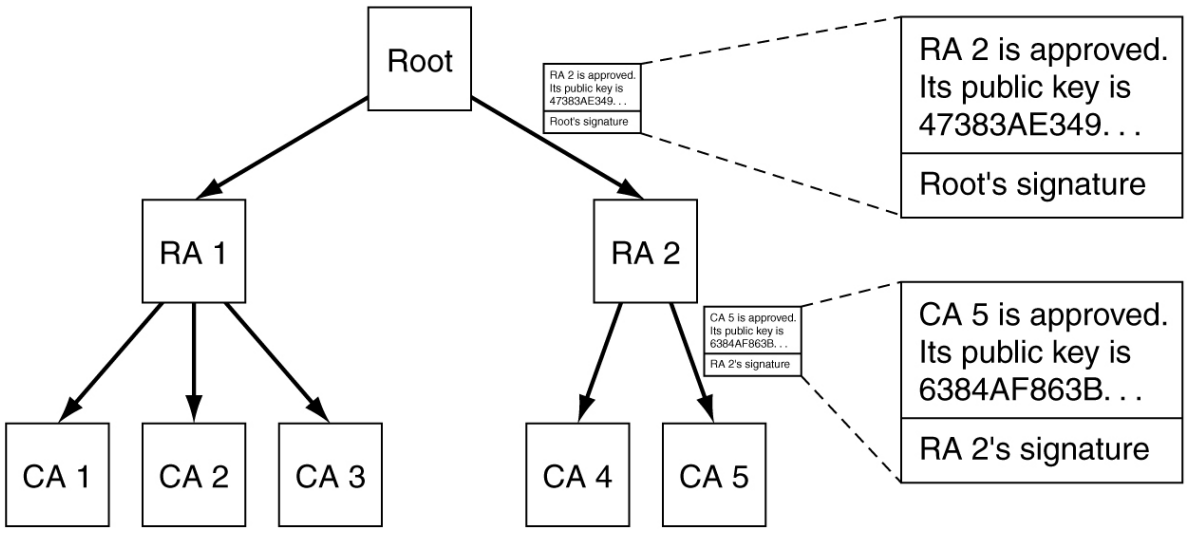
I hereby certify that the public key
19836A8B03030CF83737E3837837FC3s87092827262643FFA82710382828282A
belongs to
Robert John Smith
12345 University Avenue
Berkeley, CA 94702
Birthday: July 4, 1958
Email: bob@superdupernet.com

SHA-1 hash of the above certificate signed with the CA's private key

Certificado digital

Certificados

Gestión de claves públicas:
Infraestructura de clave pública (PKI)



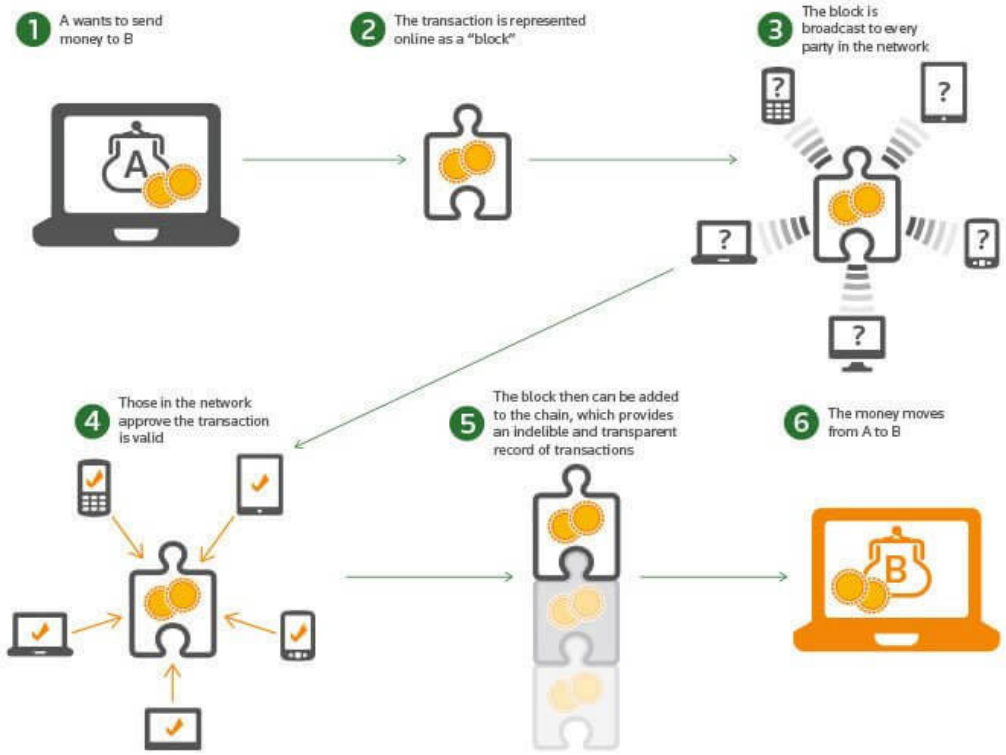
Tecnología "blockchain"

Protocolo Bitcoin
("Satoshi Nakamoto", 2008)

- Criptografía asimétrica de clave pública.
- Direcciones públicas derivadas de la clave pública:
  33 caracteres en Base58
  (derivado de Base64, sin 0, I, O, l, + ni /).
- Transacciones autorizadas con la clave privada:
  A transfiere a B agregando la clave pública de B
  y firmando con su clave privada.

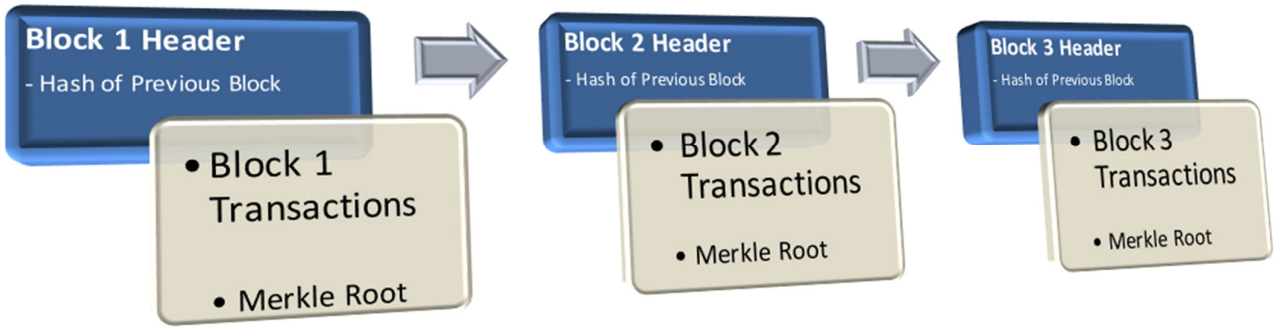
Tecnología "blockchain"

Protocolo Bitcoin



Tecnología "blockchain"

Red P2P: "**distributed ledger**"

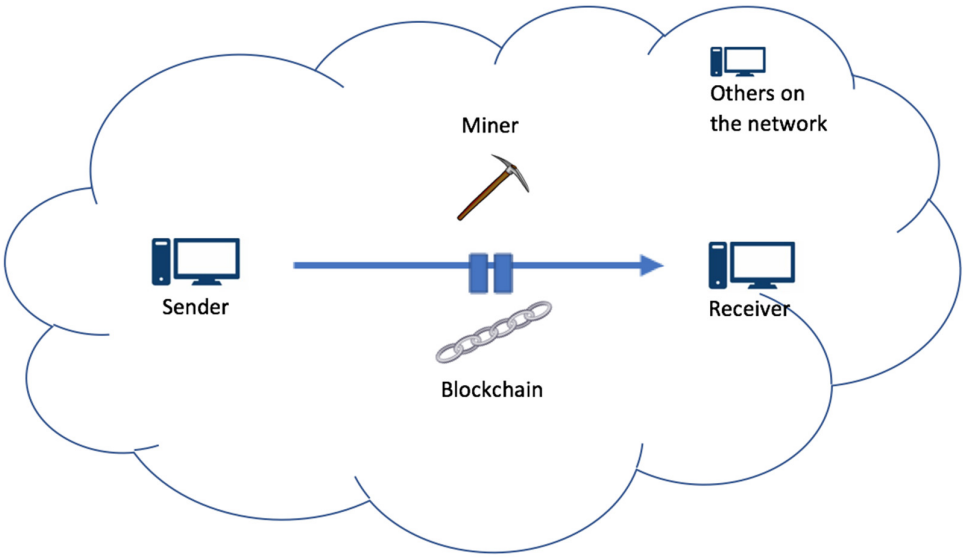


Mantenimiento distribuido de todas las transacciones conocidas en una cadena de bloques:

- Historial de posesión de las monedas desde su creación hasta su propietario actual.

Tecnología "blockchain"

Cadena de bloques



LA CLAVE: HASHING CRIPTOGRÁFICO

Bloques inmutables, cadena inalterable en la práctica

vs. "derecho al olvido" ???

Tecnología "blockchain"

Mitos y realidades

	Myth	Reality	
1	Blockchain is Bitcoin	Bitcoin is just one crypto-currency application of blockchain	Blockchain technology can be used and configured for many other applications
2	Blockchain is better than traditional databases	Blockchain's advantages come with significant technical trade-offs that mean traditional databases often still perform better	Blockchain is particularly valuable in low-trust environments where participants can't trade directly or lack an intermediary
3	Blockchain is immutable or tamper-proof	Blockchain data structure is append only, so data can't be removed	Blockchain could be tampered with if >50% of the network-computing power is controlled and all previous transactions are rewritten—which is largely impractical
4	Blockchain is 100% secure	Blockchain uses immutable data structures, such as protected cryptography	Overall blockchain system security depends on the adjacent applications—which have been attacked and breached
5	Blockchain is a "truth machine"	Blockchain can verify all transactions and data entirely contained on and native to blockchain (eg, Bitcoin)	Blockchain cannot assess whether an external input is accurate or "truthful"—this applies to all off-chain assets and data digitally represented on blockchain

McKinsey&Company

Tecnología "blockchain"

Aplicaciones

Criptomonedas ["Cryptocurrencies"]: Monedas digitales

- BTC: **Bitcoin** (2009)
- LTC: Litecoin (2011)
- NXT: Nxt (2013)
- Dash = XCoin (2014)
- NEM (2015)

Transacciones entre bancos

- **Ripple** (2012)

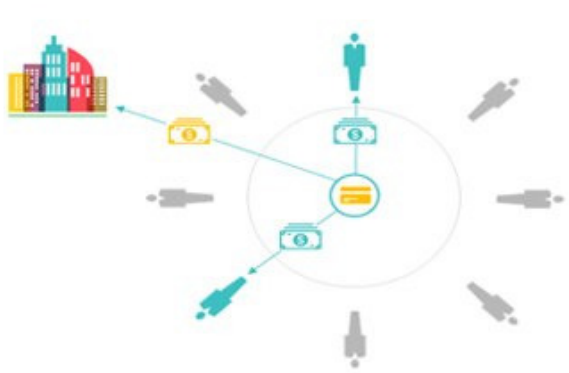
"Smart contracts" (sin intermediarios)

- **Ethereum** (2013)

http://coinmarketcap.com/ **>10k !!!**

Tecnología "blockchain"

Aplicaciones: Pagos sin intermediarios



Current Payment systems require third-party intermediaries that often changes high processing fees…

… but machine-to-machine payment using the Bitcoin protocol could allow for direct payment between individuals, as well as support micropayments

Tecnología "blockchain"

Casos de uso

Record keeping: storage of static information

Transactions: registry of tradeable information

1 Static registry

- Distributed database for storing reference data

Example
- Land title
- Food safety and origin
- Patent

2 Identity

- Distributed database with identity-related information
- Particular case of static registry treated as a separate group of use cases due to extensive set of identity-specific use cases

Example
- Identity fraud
- Civil-registry and identity records
- Voting

3 Smart contracts

- Set of conditions recorded on a blockchain triggering automated, self-executing actions when these predefined conditions are met

Example
- Insurance-claim payout
- Cash-equity trading
- New-music release

4 Dynamic registry

- Dynamic distributed database that updates as assets are exchanged on the digital platform

Example
- Fractional investing
- Drug supply chain

5 Payments infrastructure

- Dynamic distributed database that updates as cash or cryptocurrency payments are made among participants

Example
- Cross-border peer-to-peer payment
- Insurance claim

6 Other

- Use case composed of several of the previous groups
- Standalone use case not fitting any of the previous categories

Example
- Initial coin offering
- Blockchain as a service

McKinsey&Company

Autentificación

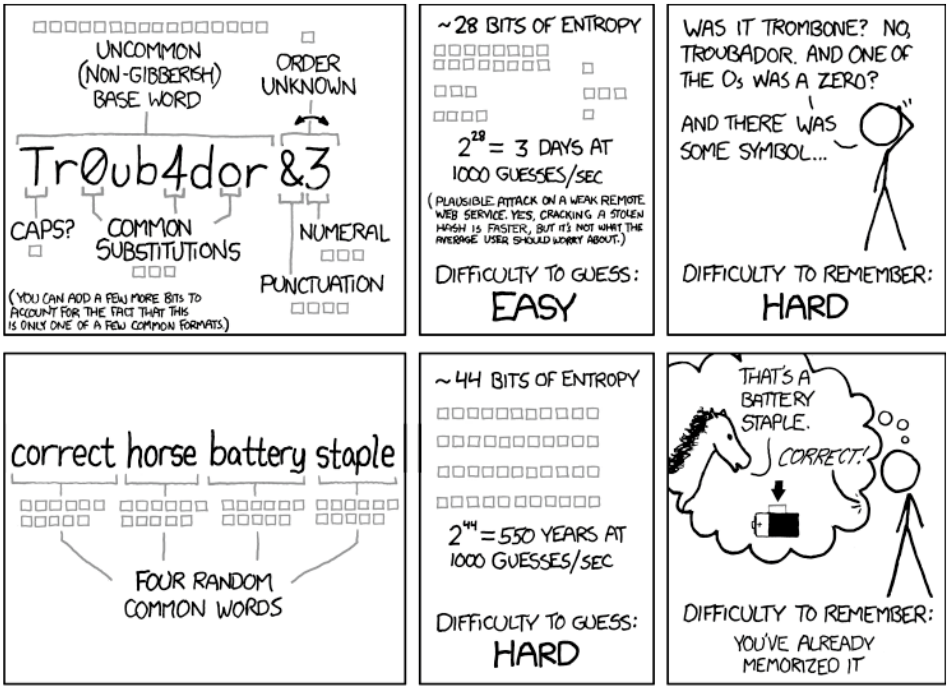
TEST: ¿Qué contraseña es más segura?

p@ssw0rd	pAsswOrd
punk4life	punkforlife
ieatkale88	iloveyou88
jonnyrtxe	jonny1421
sk8erboy	skaterboy
1qaz2wsx3edc	thefirstkiss

Do Users' Perceptions of Password Security Match Reality?, CHI'2016

Autentificación

Contraseñas "seguras" https://xkcd.com/936/



Autentificación

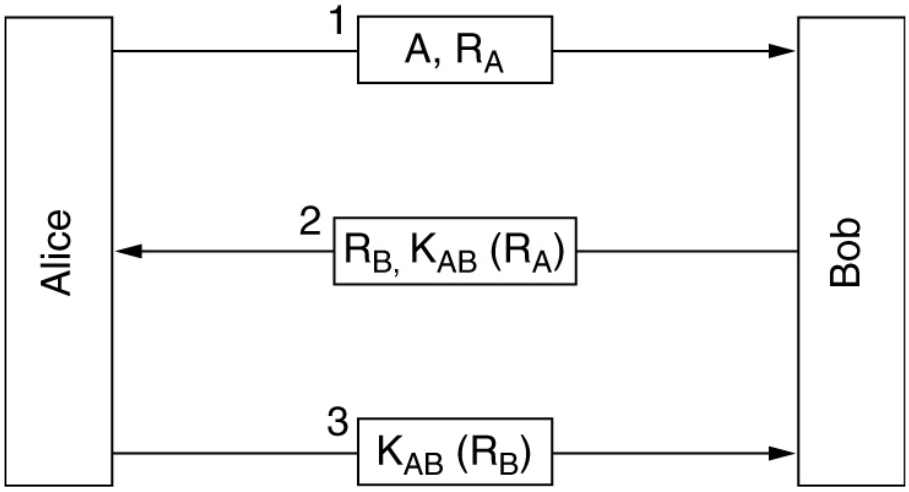
Las peores contraseñas
(usadas por el 10% de los usuarios)

123456	password	admin	abc123
12345678	passw0rd	login	qwerty
12345	letmein	welcome	qazwsx
123456789	football	master	hello
1234567	iloveyou	starwars	whatever
123123	dragon	monkey	freedom
			trustno1

http://jobs.ieee.org/jobs/content/Is-Your-Password-on-This-List--2018-01-31

Autentificación

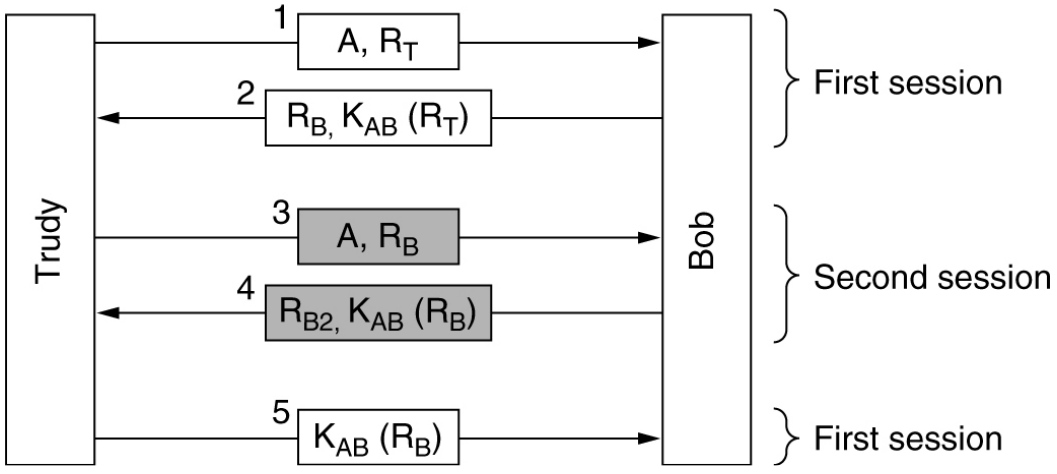
Garantizar que el origen y el destino sean quienes dicen ser…



Autentificación con clave secreta compartida

Autentificación

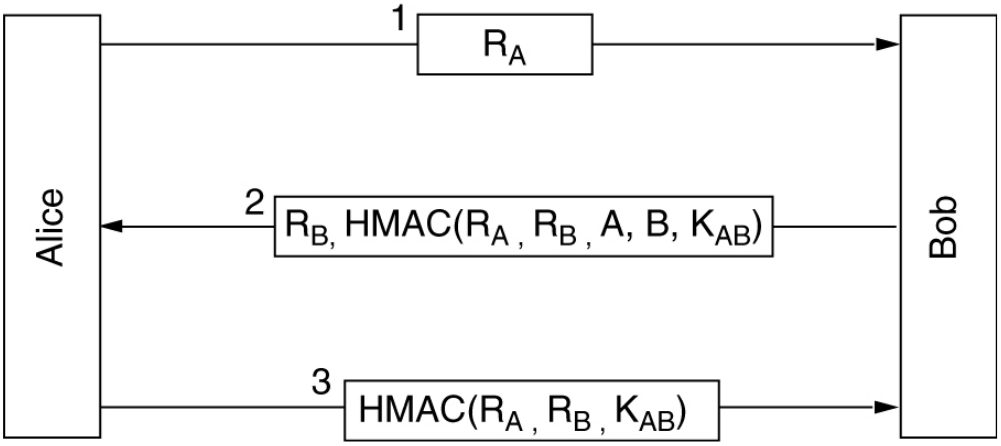
Garantizar que el origen y el destino sean quienes dicen ser…



Problema: Ataque por reflexión

Autentificación

Garantizar que el origen y el destino sean quienes dicen ser…



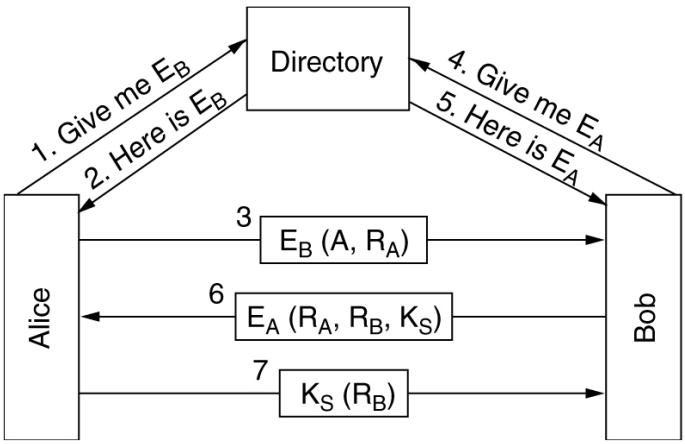
Solución: HMAC

(Keyed-Hashing for Message Authentication)

RFC 2104

Autentificación

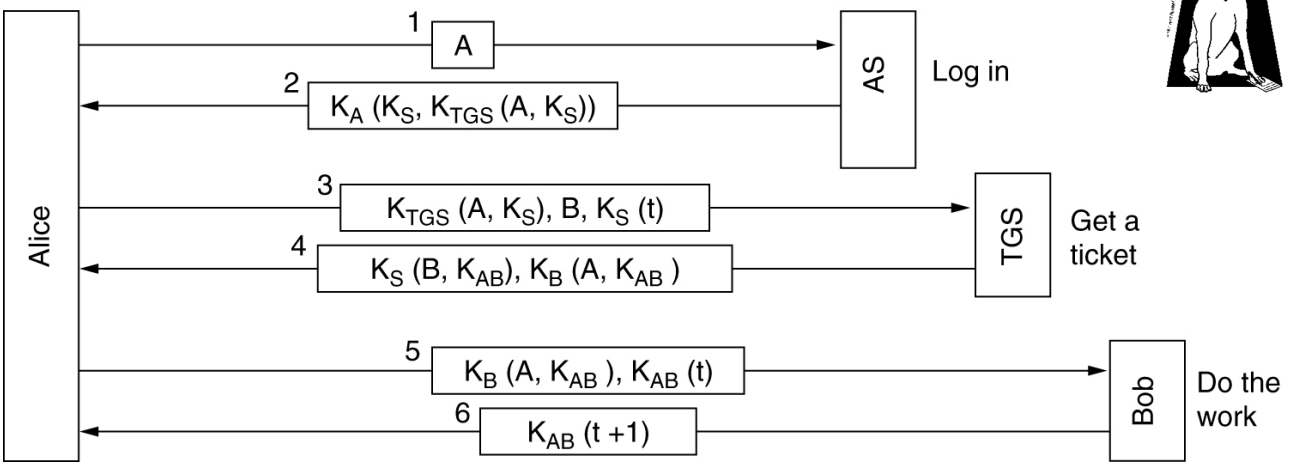
Autentificación basada en criptografía de clave pública



Autentificación mutua

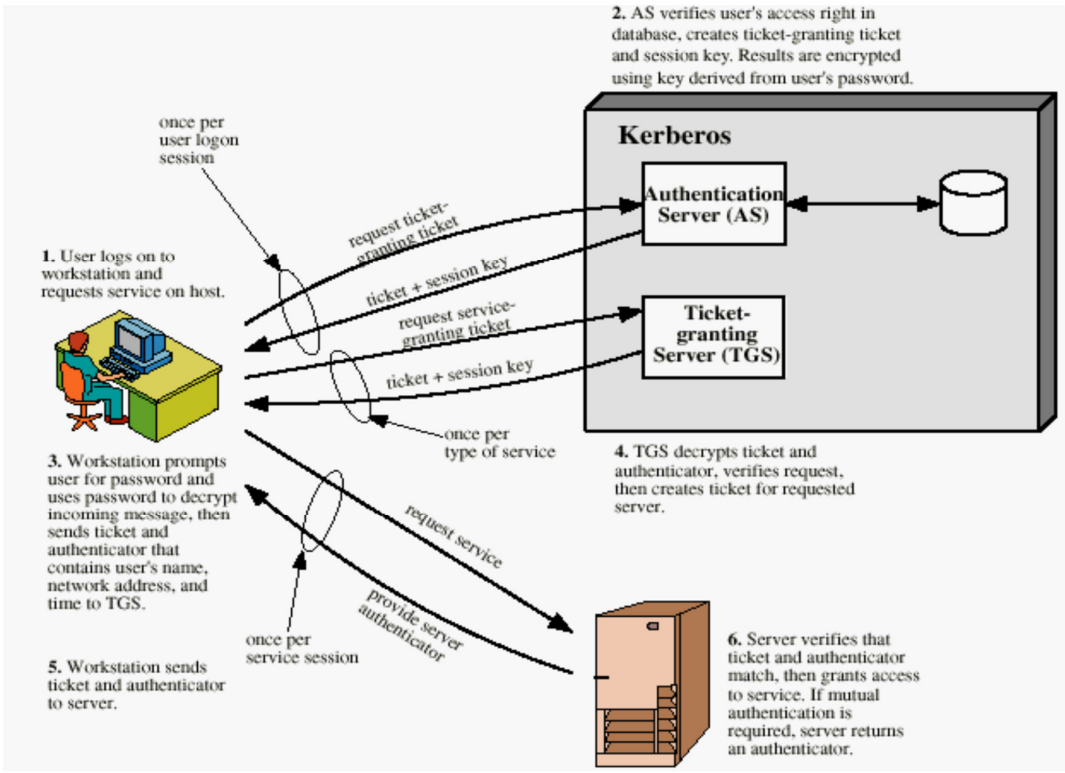
Autentificación

Kerberos



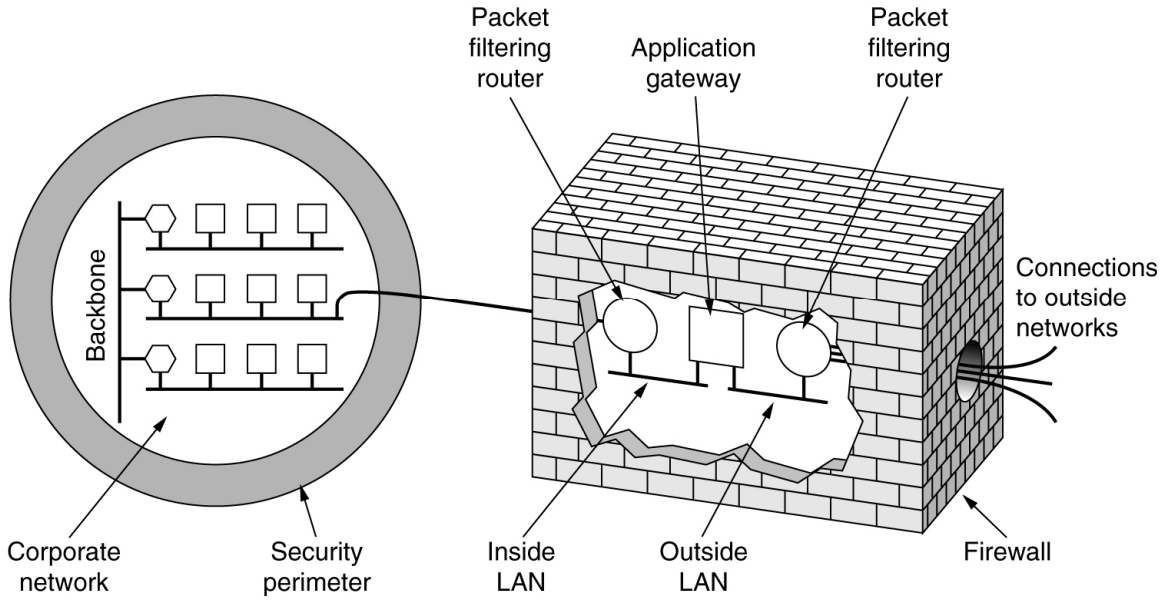
Autentificación

Kerberos



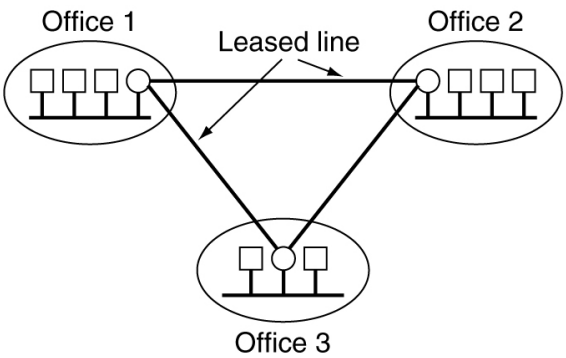
Comunicaciones seguras

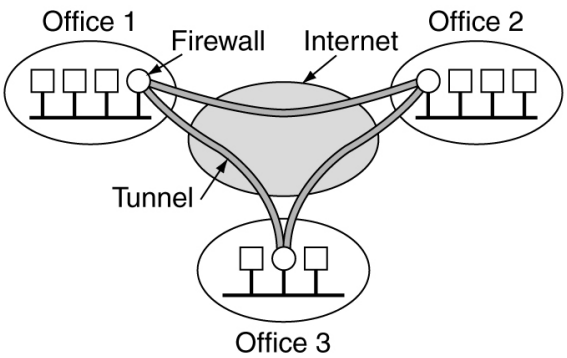
Cortafuegos



Comunicaciones seguras

Redes privadas virtuales

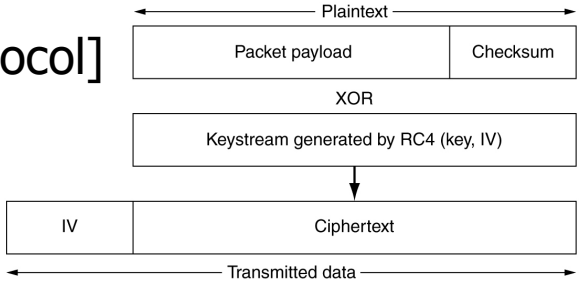




Comunicaciones seguras

Seguridad en redes inalámbricas 802.11

- **WEP** [Wireless Encryption Protocol]
  Inseguro: NO UTILIZAR.

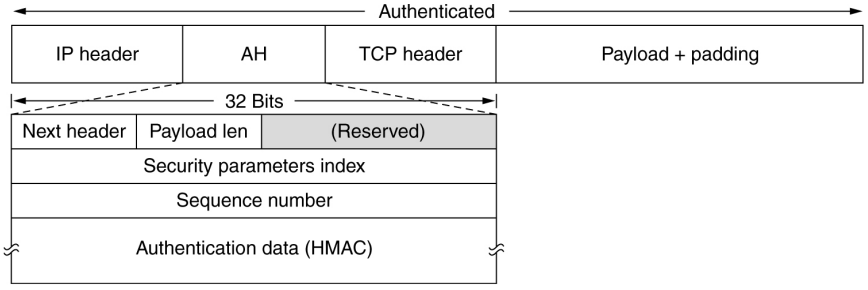


- **WPA** [Wi-Fi Protected Access]
  - Personal Mode = PSK [Pre-Shared Key]:
    Clave compartida (RC4).
    Modo menos seguro, sin servidor de autentificación.
  - TKIP [*Temporal Key Integrity Protocol*], 2002.
    Protocolo de Integridad de Clave Temporal: Contador de secuencia (para prevenir ataques por repetición) y comprobación de integridad [MICHAEL].
  - CCMP [Counter Mode with Cipher Block Chaining Message Authentication Code Protocol] @ WPA2, 2004

Comunicaciones seguras

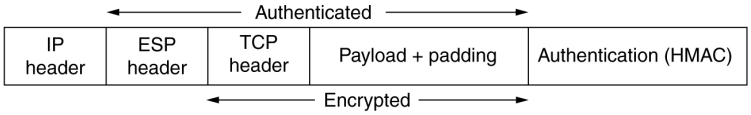
El protocolo IPSec (security extensions for IPv4/v6)

- Cabecera de autentificación

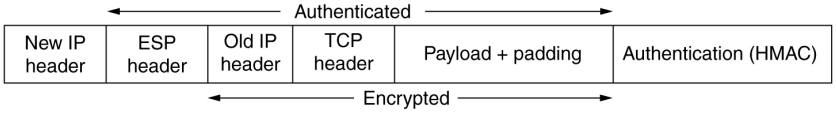


- ESP: Encapsulating Security Payload

Modo de transporte



Modo túnel



- Algoritmo de intercambio de claves

Comunicaciones seguras

SSL [Secure Sockets Layer] & TLS [Transport Layer Security]

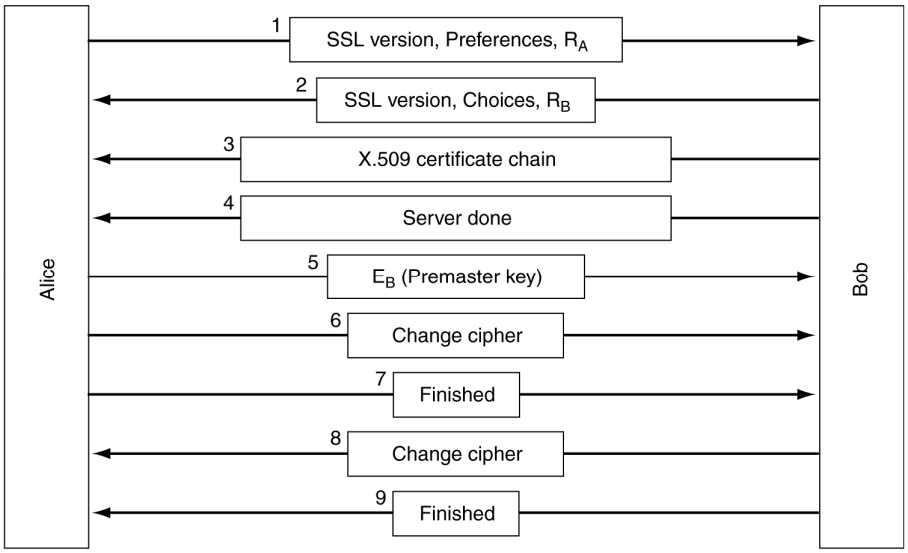
Application (HTTP)
Security (SSL)
Transport (TCP)
Network (IP)
Data link (PPP)
Physical (modem, ADSL, cable TV)

Historia: SSL 3.0 @ Netscape, 1996
TLS 1.0 @ RFC 2246, 1999 ≈ SSL 3.0
TLS 1.1 @ RFC 4346, 2006

Comunicaciones seguras

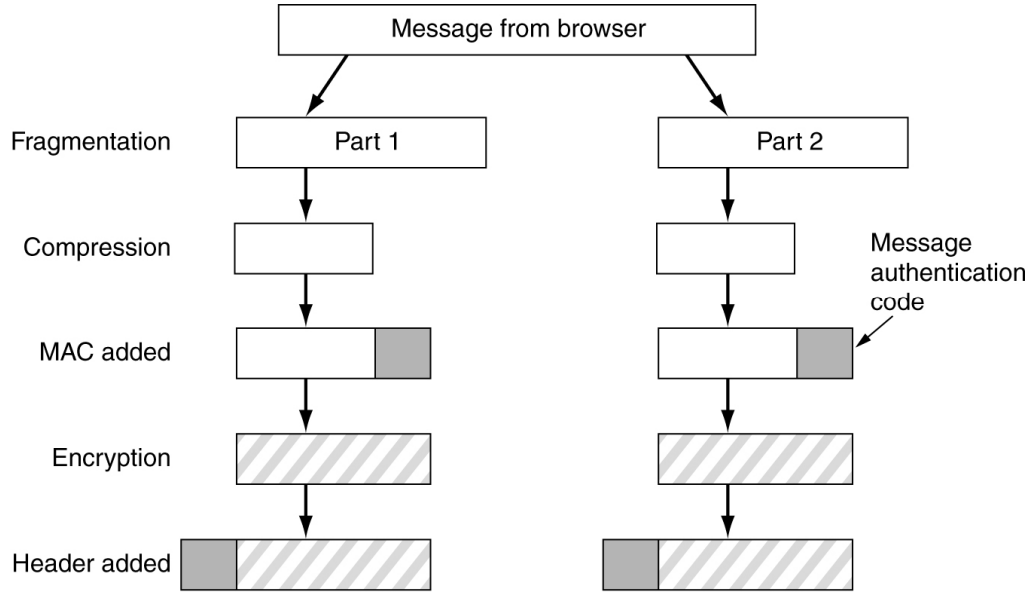
SSL / TLS

Establecimiento de una conexión SSL/TLS

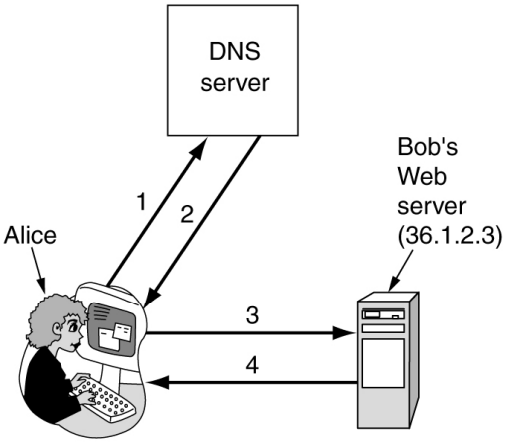


SSL / TLS

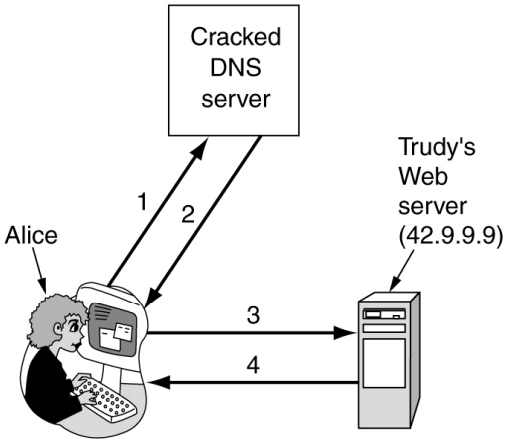
Transmisión de datos con SSL/TLS



Servidores de nombres DNS



1. Give me Bob's IP address
2. 36.1.2.3 (Bob's IP address)
3. GET index.html
4. Bob's home page

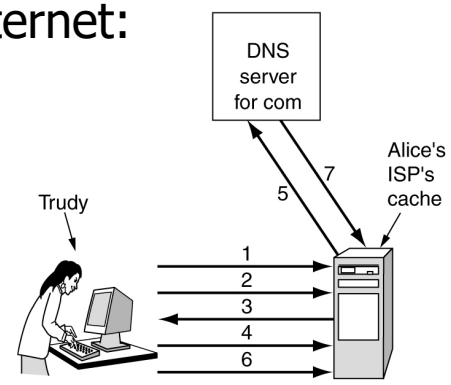


1. Give me Bob's IP address
2. 42.9.9.9 (Trudy's IP address)
3. GET index.html
4. Trudy's fake of Bob's home page

Situación normal vs. Ataque a los servidores DNS

Servidores de nombres DNS

Ataque utilizando la infraestructura del proveedor de acceso a Internet:



1. Look up foobar.trudy-the-intruder.com (to force it into the ISP's cache)
2. Look up www.trudy-the-intruder.com (to get the ISP's next sequence number)
3. Request for www.trudy-the-intruder.com (Carrying the ISP's next sequence number, n)
4. Quick like a bunny, look up bob.com (to force the ISP to query the com server in step 5)
5. Legitimate query for bob.com with seq = n+1
6. Trudy's forged answer: Bob is 42.9.9.9, seq = n+1
7. Real answer (rejected, too late)

Self-certifying URL

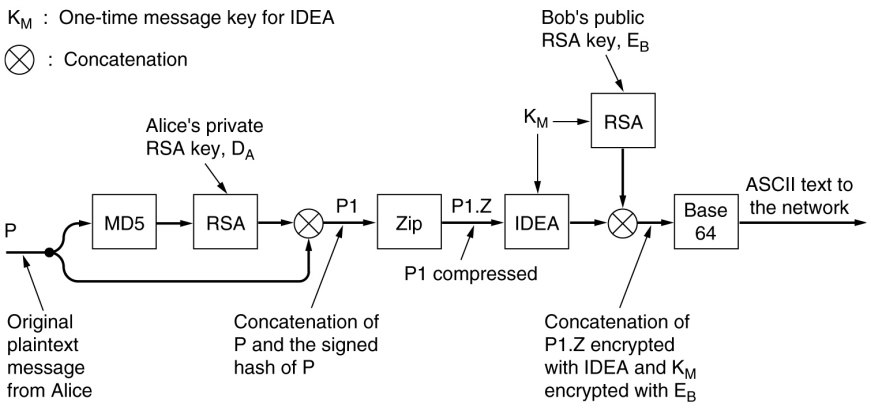
Server | SHA-1 (Server, Server's Public key) | File name

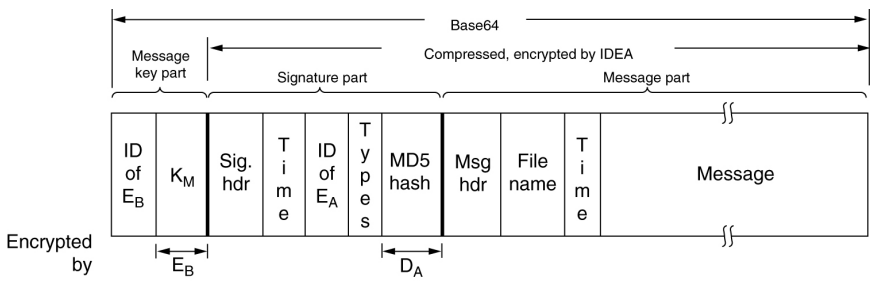
http://www.bob.com:2g5hd8bfjkc7mf6hg8dgany23xds4pe6/photos/bob.jpg

Comunicaciones seguras

Correo electrónico: PGP (Pretty Good Privacy)

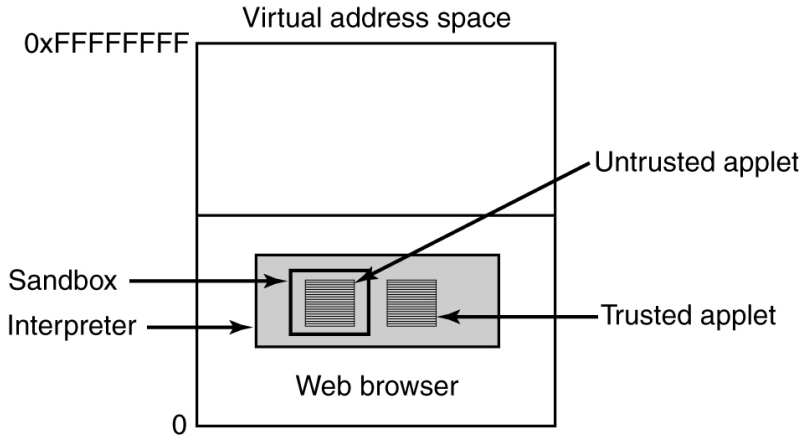
Criptografía, firma digital y compresión





Comunicaciones seguras

Seguridad en los applets Java



Seguridad en aplicaciones web

Top 10: Open Web Application Security Project

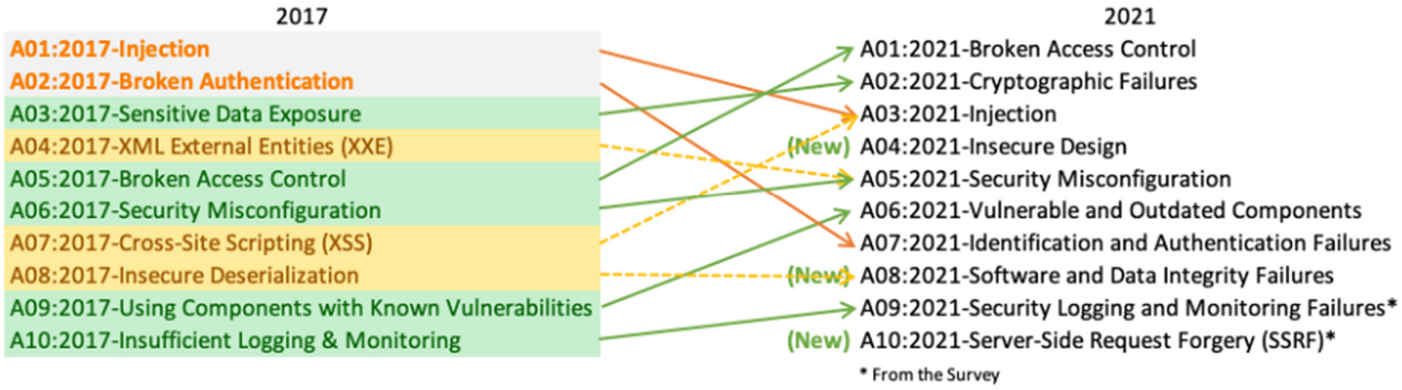
https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

1. Inyección (SQL, ORM, XML, SSI, buffer overflow…)
2. Autentificación y gestión de sesiones de usuario
3. XSS [Cross Site Scripting]
4. Referencias directas a objetos
5. Fallos de configuración (p.ej. PHP)
6. Exposición de datos sensibles
7. Controles de acceso (p.ej. URLs de ficheros PDF)
8. CSRF [Cross Site Request Forgery]
9. Redirecciones no validadas (redirect/forward)
10. Componentes con vulnerabilidades conocidas

Seguridad en aplicaciones web

Top 10: Open Web Application Security Project

https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project



2003-2021

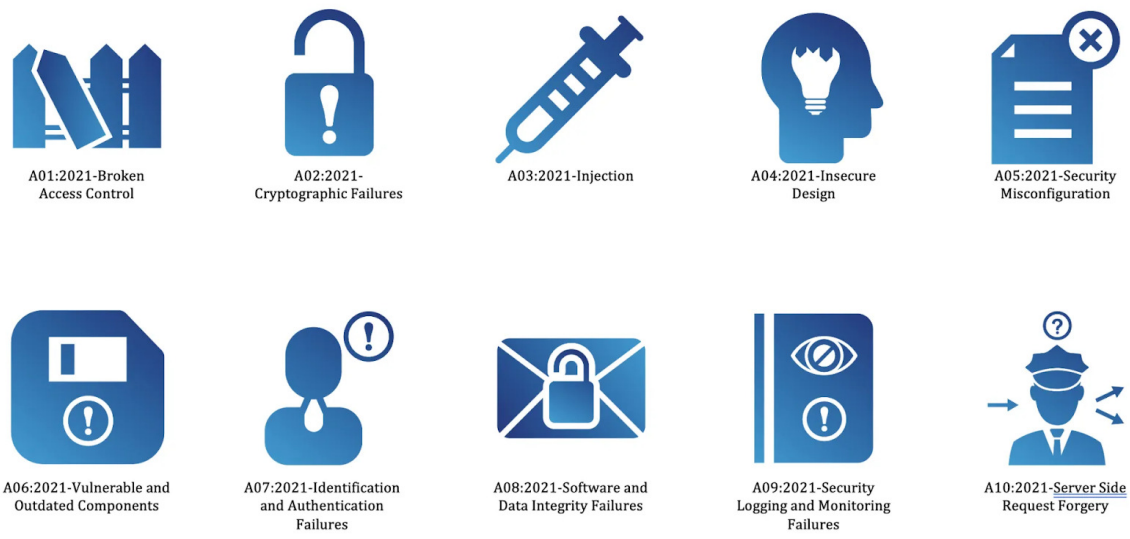
Revisiones del ranking

7ª actualización (2021)

Seguridad en aplicaciones web

Top 10: Open Web Application Security Project

https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project



Seguridad en aplicaciones web

Ataques por inyección

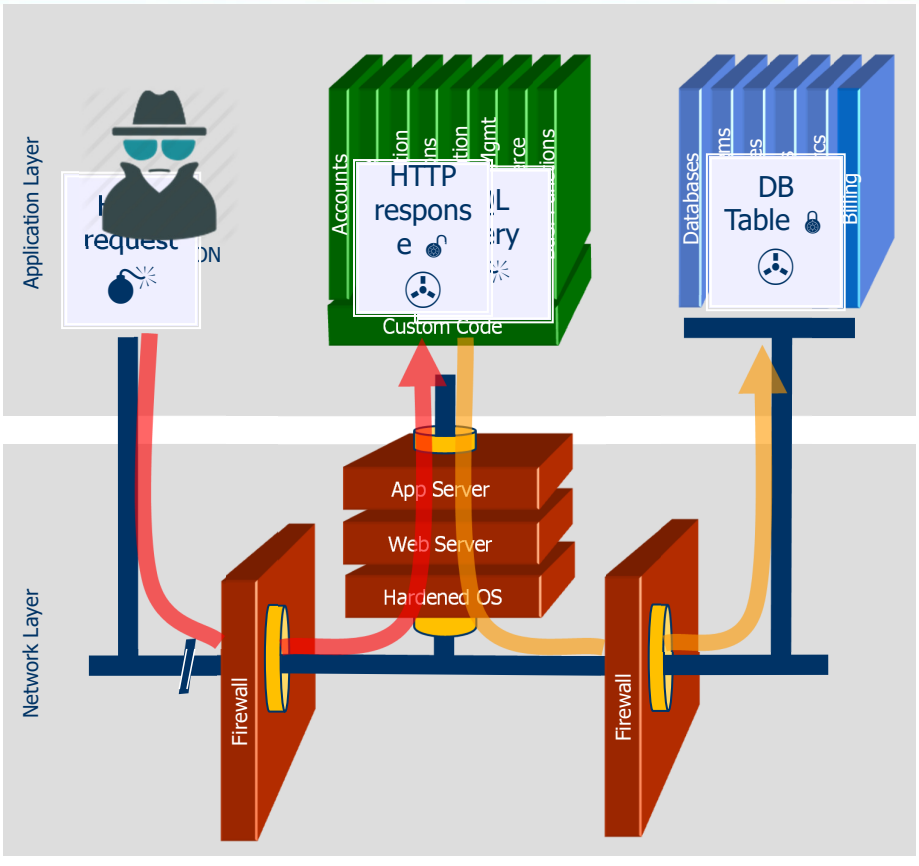
INYECCIÓN: Conseguir que una aplicación ejecute comandos por medio del envío de datos a un intérprete.

EJEMPLOS: SQL, shell SO, XPath, Hibernate…

La inyección de código SQL es muy común, pese a ser muy sencilla de prevenir :-(

https://www.owasp.org/index.php/Top_10_2013-A1-Injection

Seguridad en aplicaciones web



Account: ' OR 1=1 --

SKU:

Submit

1. Application presents a form to the attacker
2. Attacker sends an attack in the form data
3. Application forwards attack to the database in a SQL query
4. Database runs query containing attack and sends encrypted results back to application
5. Application decrypts data as normal and sends results to the user

Seguridad en aplicaciones web

Ataques por inyección

EJEMPLO @ http://elvex.ugr.es

```
[pid 4652:tid 900] [client 184.107.172.42:38400]
 AH00128: File does not exist: …/php-ofc-library/ofc_upload_image.php
...
```

Ejecución de código PHP ("Open Flash Chart")

http://www.securityfocus.com/bid/37314/exploit

http://www.example.com/libs/open-flash-chart/php-ofc-library/ofc_upload_image.php?name=shell.php&HTTP_RAW_POST_DATA=<?system($_GET['cmd']);?>

Seguridad en aplicaciones web

Ataques por inyección

EJEMPLO @ http://elvex.ugr.es

```
92.63.97.93 - - [29/Oct/2015:02:19:47 +0100] "POST /cgi-bin/php/%63%67
%69%6E/%70%68%70?%2D%64+%61%6C%75%6F%6E+%2D%64+%6D%6F%64+%2D%64+%73%75
%68%6F%6E%3D%6F%6E+%2D%64+%75%6E%63%74%73%3D%22%22+%2D%64+%64%6E%65+
%2D%64+%61%75%74%6F%5F%70%72%%74+%2D%64+%63%67%69%2E%66%6F%72%63%65
%5F%72%65%64%69%72%65%63%74%3D%30+%2D%64+%74%5F%3D%30+%2D%64+%75%74
+%2D%6E HTTP/1.1" 404 218 ...
```

Ejecución de código PHP

URL decodificada

/cgi-bin/php/cgin/php?-d+aluon+-d+mod+-d+suhon=on+-d+uncts=""+-d+dne+-d+auto_pr%t+-d+cgi.force_redirect=0+-d+t_=0+-d+ut+-n

Seguridad en aplicaciones web

Ataques por inyección

RECOMENDACIONES

- Evitar el intérprete por completo, o bien…
- Utilizar una interfaz que permita el uso de variables para distinguir entre código y datos (p.ej. "prepared statements", procedimientos almacenados…)
- Codificar siempre las entradas provenientes del usuario antes de pasárselas al intérprete (p.ej. "escaping")
- Validar todas las entradas del usuario.
- Minimizar los privilegios de la aplicación en la BD para reducir el impacto del fallo de seguridad.

Seguridad en aplicaciones web

Ataques por inyección

EJEMPLO EN JAVA: JDBC

VERSIÓN NO SEGURA

```
String query = "SELECT account_balance FROM user_data"
  +" WHERE user_name = "+ request.getParameter("customerName");
Statement statement = connection.createStatement( … );
ResultSet results = statement.executeQuery( query );
```

VERSIÓN SEGURA: PREPARED STATEMENTS

```
String custname = request.getParameter("customerName");
String query = "SELECT account_balance FROM user_data"
  +" WHERE user_name = ? ";
PreparedStatement pstmt = connection.prepareStatement( query );
pstmt.setString( 1, custname);
ResultSet results = pstmt.executeQuery( );
```

Ataques por inyección

EJEMPLO EN JAVA: JDBC

VERSIÓN SEGURA: PROCEDIMIENTO ALMACENADO

```
String custname = request.getParameter("customerName");
CallableStatement cs = connection.prepareCall(
  "{call sp_getAccountBalance(?)}");
cs.setString(1, custname);
ResultSet results = cs.executeQuery();
```

VERSIÓN SEGURA: VALIDACIÓN DE ENTRADAS (WHITE LISTS)

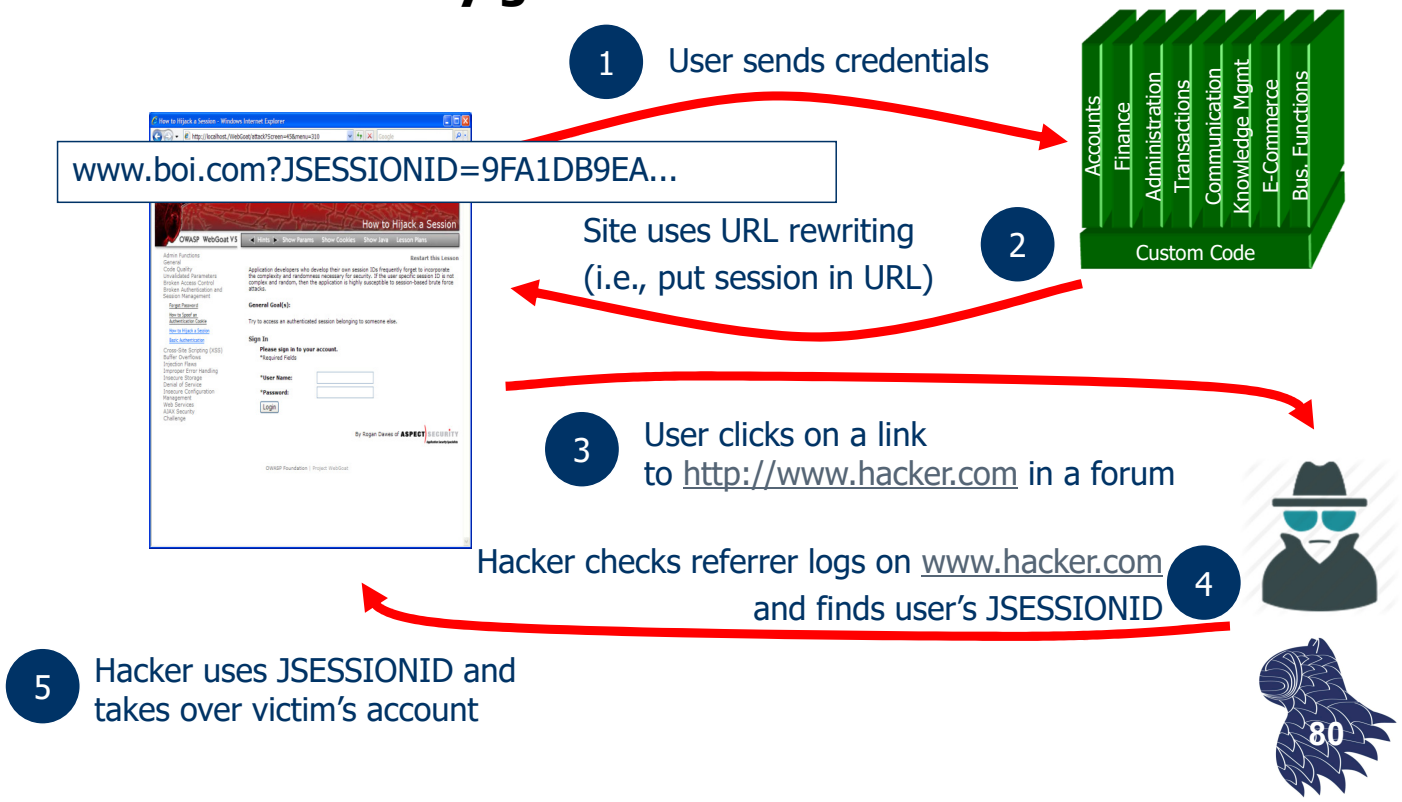
```
switch(PARAM) {
  case "Value1": tableName = “Table1"; break;
  …
  default:       throw new InputValidationException
                    ("unexpected value for table name");
}
```


Seguridad en aplicaciones web

Autentificación y gestión de sesiones de usuario

- HTTP es un protocolo sin estado: las credenciales se suministran con cada solicitud y se debería utilizar SSL para todo lo que requiera autentificación.
- SESSION_ID sustituye a las credenciales pero es igual de útil para un atacante (dato expuesto en el navegador del cliente, en los logs…)
- Puertas laterales: cambiar/recordar contraseña, pregunta secreta, correo electrónico…

Autentificación y gestión de sesiones de usuario



Seguridad en aplicaciones web

Autentificación y gestión de sesiones de usuario

RECOMENDACIONES

- Autentificación simple, centralizada y estandarizada.
- Usar el ID de sesión proporcionado por el contenedor (servidor de aplicaciones).
- Usar SSL para proteger las credenciales y los identificadores de sesión en todo momento.
- Verificar el certificado SSL de la página.
- Verificar que "salir" destruye realmente la sesión.
- Establecer "mínimos" para la complejidad de las contraseñas y mecanismos seguros de recuperación.
- Diseñar el mecanismo de almacenamiento de claves asumiendo un eventual compromiso de seguridad...

Autentificación y gestión de sesiones de usuario

ALMACENAMIENTO DE CLAVES

- Las claves en sí: sin limitar el juego de caracteres permitido y fijando la longitud máxima (p.ej. 160) en prevención de ataques DoS [Denial of Service].

- Almacenamiento "salteado" utilizando funciones hash:

```
[protected form] = [salt]
+ protect([protection func], [salt] + [credential]);
```

"sal" [salt]: Valor aleatorio de longitud fija, p.ej. 64 bits, específico para cada credencial almacenada.

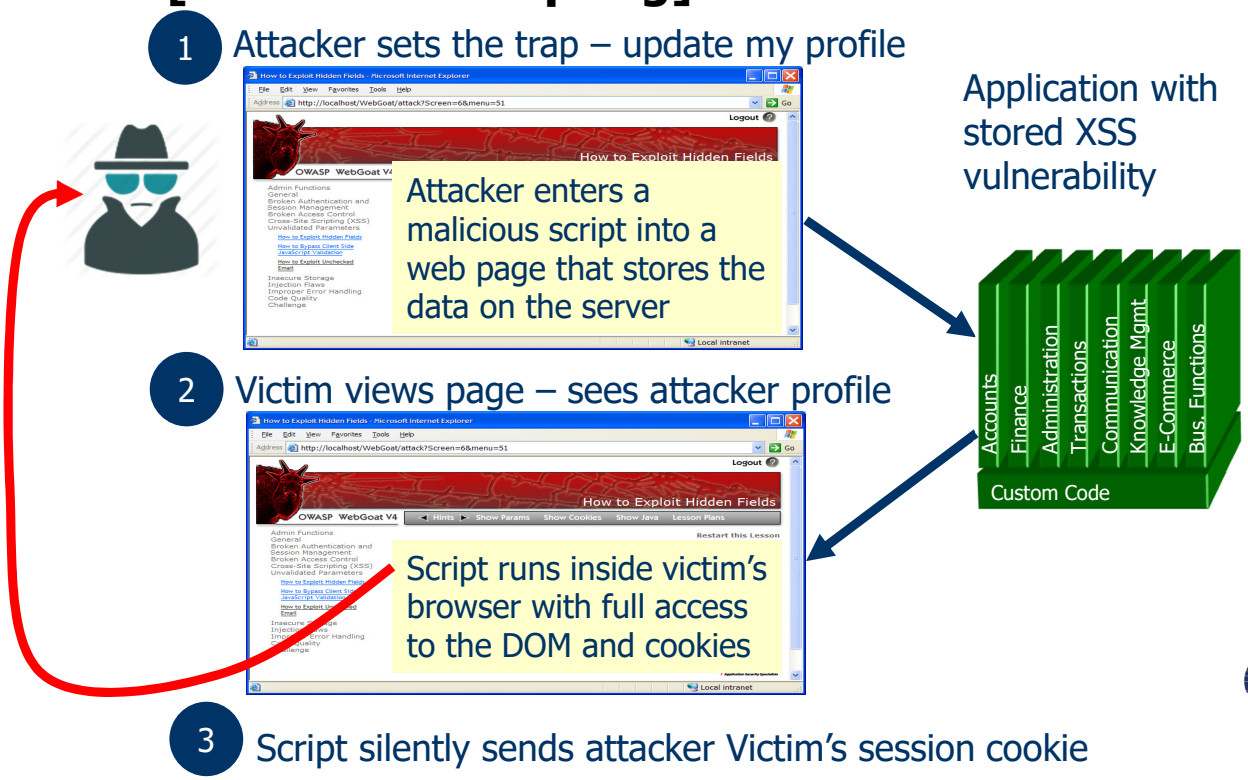
Seguridad en aplicaciones web

XSS [Cross-Site Scripting]

- El atacante consigue enviar datos al navegador de la víctima, en el que consigue ejecutar código en JavaScript para apoderarse de su sesión de usuario, robar datos, manipular la página web visualizada, redirigirlo a una página falsa o incluso instalar un proxy para observar su comportamiento.

- Casi todas las aplicaciones web son vulnerables a este tipo de ataques, p.ej. `javascript:alert(document.cookie)`

XSS [Cross-Site Scripting]



Seguridad en aplicaciones web

XSS [Cross-Site Scripting]

RECOMENDACIONES

Eliminación de la vulnerabilidad:

- No incluir nunca datos suministrados por el usuario en la página generada por la aplicación web.

Defensas frente a la vulnerabilidad:

- Codificar todas las entradas suministradas por el usuario (p.ej. "escaping", "sanitization"…).
- Validación usando listas blancas ["white lists"].
- CSP [Content Security Policy], W3C specification.

Seguridad en aplicaciones web

Referencias directas a objetos

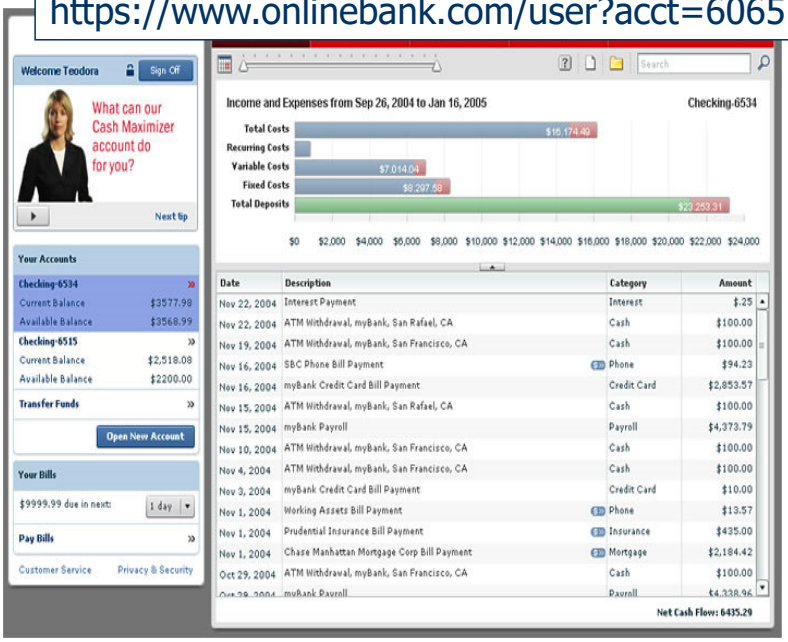
- Error habitual:
  Control de acceso en la capa de presentación
  Referencias a objetos en campos ocultos de formularios, cuyo control de acceso no se controla correctamente en el servidor.

- El atacante puede modificar esa referencia para acceder a datos a los que no debería tener acceso.

Seguridad en aplicaciones web

Referencias directas a objetos

https://www.onlinebank.com/user?acct=6065



- Attacker notices his acct parameter is 6065
  ?acct=6065
- He modifies it to a nearby number
  ?acct=6066
- Attacker views the victim's account information

Seguridad en aplicaciones web

Fallos de configuración

En cualquier parte del sistema que da soporte a la aplicación web: sistema operativo, servidor HTTP, contenedor / servidor de aplicaciones, intérpretes (p.ej. PHP), DBMS, intranet…

Impacto:

- Instalación de puertas traseras [backdoors].
- Accesos no autorizados
- …

Seguridad en aplicaciones web

Fallos de configuración

EJEMPLO @ http://elvex.ugr.es

```
[pid 4616:tid 848] [client 75.119.221.245:34465]
 AH00128: File does not exist: …/www/elvex/wp-admin/
[pid 4100:tid 912] [client 85.25.136.37:18395]
 AH00128: File does not exist: …/www/elvex/test/wp-admin/
[pid 4100:tid 912] [client 180.210.204.141:59963]
 AH00128: File does not exist: …/www/elvex/wordpress/wp-admin/
[pid 4100:tid 868] [client 108.171.217.244:35884]
 AH00128: File does not exist: …/www/elvex/blog/wp-admin/
[pid 4100:tid 904] [client 216.104.160.77:60732]
 AH00128: File does not exist: …/www/elvex/wp/wp-admin/
[pid 4100:tid 872] [client 193.143.77.22:38799]
 AH00128: File does not exist: …/www/elvex/old/wp-admin/
...
```

Intentos de acceso a una aplicación web (WordPress)

Seguridad en aplicaciones web

Exposición de datos sensibles

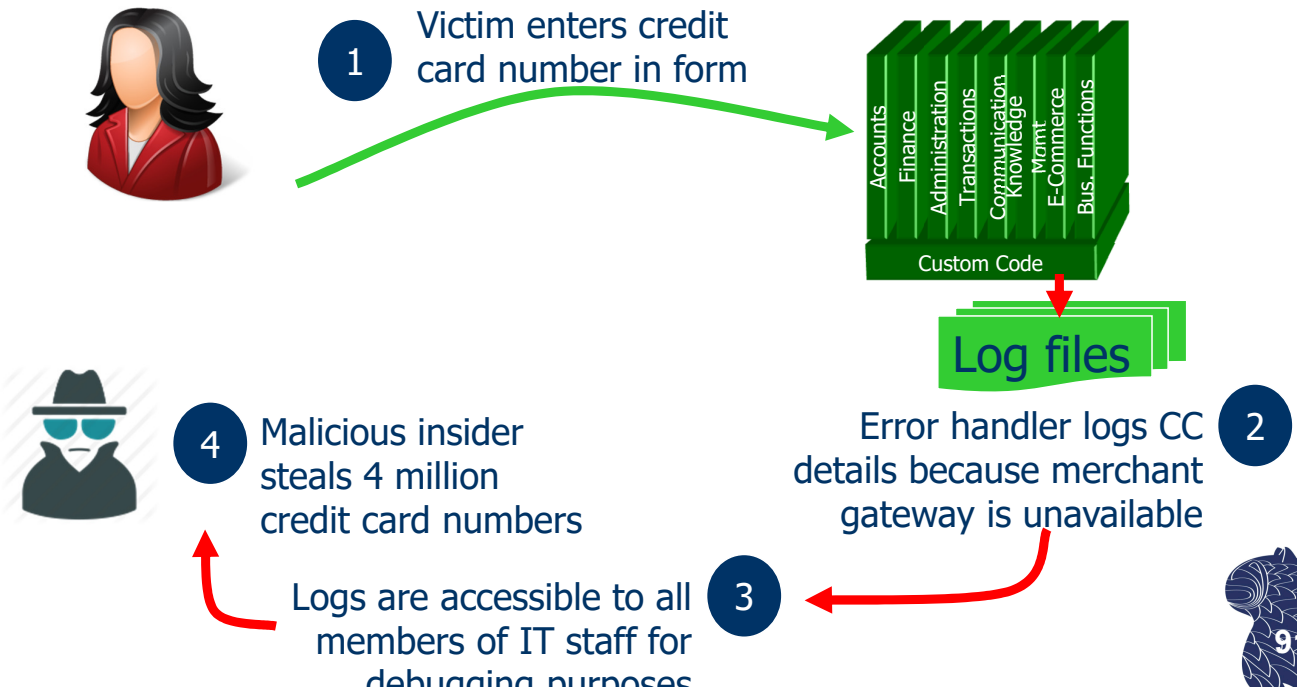
- Falta de seguridad en el almacenamiento (bases de datos, ficheros, logs, copias de seguridad…) o la transmisión de datos sensibles (uso de HTTPS).

Impacto:

- Modificación y robo de datos confidenciales.
- Pérdida de confianza de los clientes.
- Costes derivados de la brecha de seguridad.
- Problemas legales.

Seguridad en aplicaciones web

Exposición de datos sensibles: Almacenamiento



Seguridad en aplicaciones web

Exposición de datos sensibles: Almacenamiento

Twitter, 3 de mayo de 2018

Hi @fberzal,

When you set a password for your Twitter account, we use technology that masks it so no one at the company can see it. We recently identified a bug that stored passwords unmasked in an internal log. We have fixed the bug, and our investigation shows no indication of breach or misuse by anyone.

Out of an abundance of caution, we ask that you consider changing your password on all services where you've used this password. You can change your Twitter password anytime by going to the password settings page.

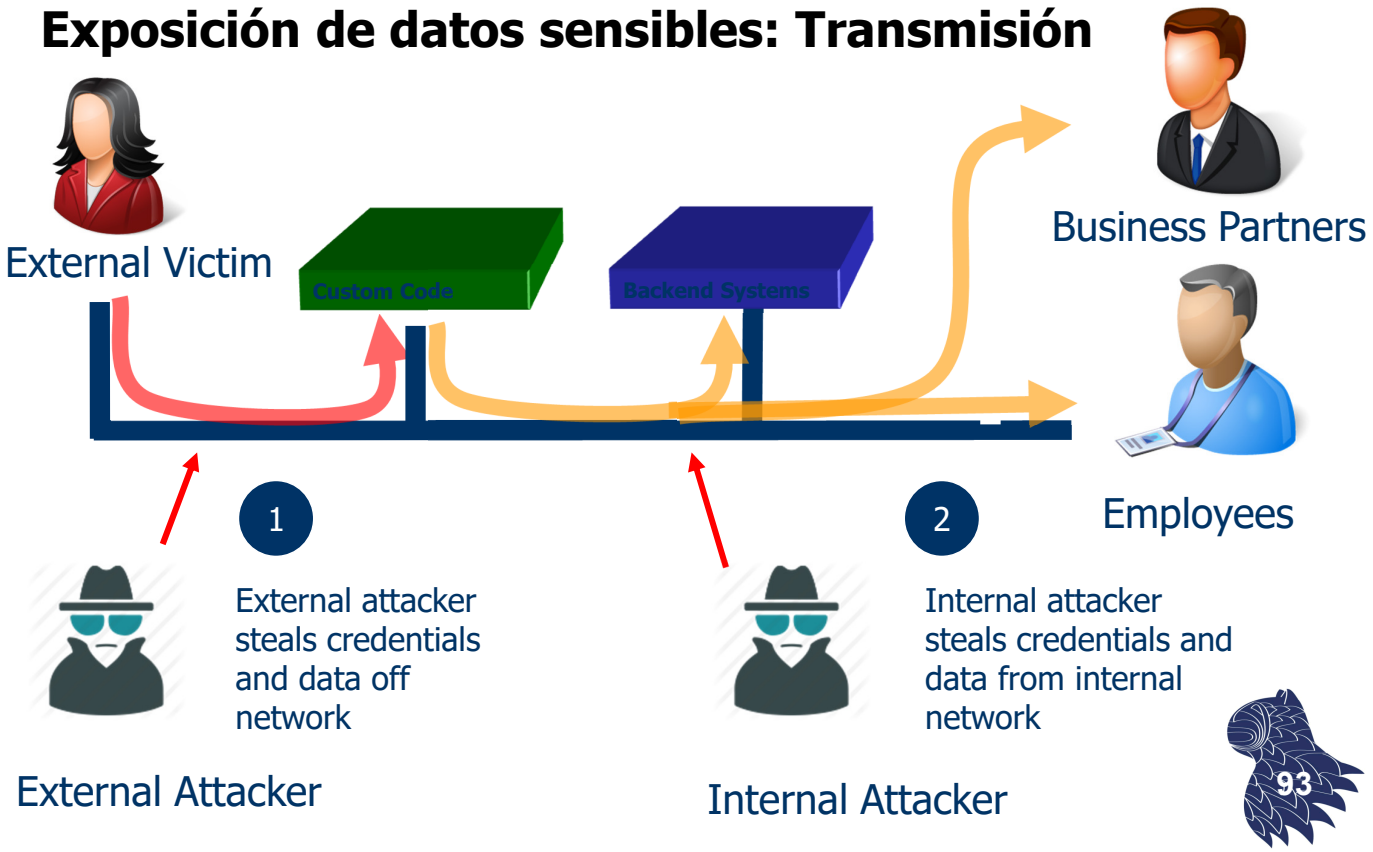
About The Bug

We mask passwords through a process called hashing using a function known as bcrypt, which replaces the actual password with a random set of numbers and letters that are stored in Twitter's system. This allows our systems to validate your account credentials without revealing your password. This is an industry standard.

Due to a bug, passwords were written to an internal log before completing the hashing process. We found this error ourselves, removed the passwords, and are implementing plans to prevent this bug from happening again.

Seguridad en aplicaciones web

Exposición de datos sensibles: Transmisión



Controles de acceso

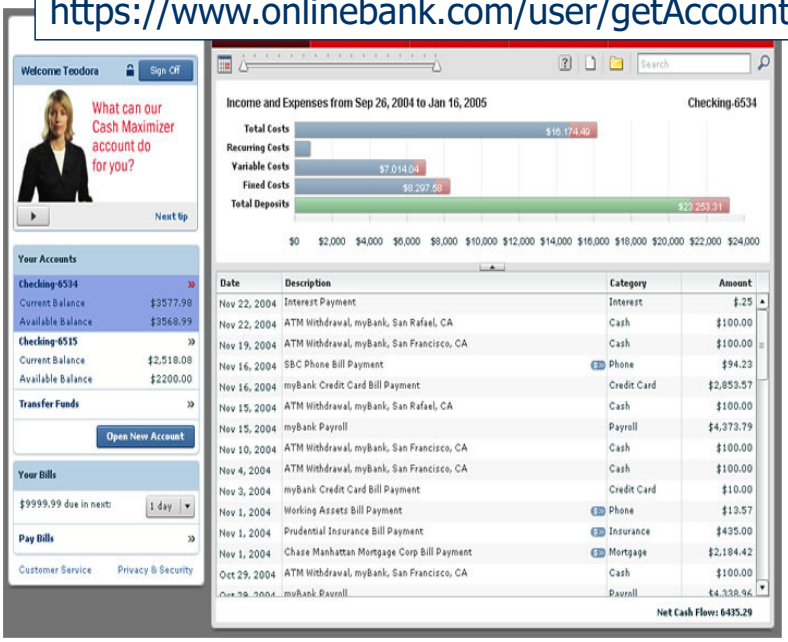
- Protección del acceso a URLs o a funciones de la aplicación a las que se hace referencias por medio de URLs+parámetros.

- Error habitual:
  Control de acceso en la capa de presentación
  Conociendo la URL usada por la aplicación, el atacante puede tener acceso a funciones a las que no debería.

Seguridad en aplicaciones web

Controles de acceso

https://www.onlinebank.com/user/getAccounts



- Attacker notices the URL indicates his role
  /user/getAccounts
- He modifies it to another directory (role)
  /admin/getAccounts, or
  /manager/getAccounts
- Attacker views more accounts than just their own

CSRF [Cross Site Request Forgery]

- Ataque en el que el navegador de la víctima envía comandos a una aplicación web vulnerable:
  La víctima es usuaria legítima de la aplicación, pero es el atacante el que realiza las acciones en su nombre.

- Causa: Los navegadores web envían automáticamente las credenciales del usuario con cada solicitud (cookie, ID sesión, dirección IP), aunque ésta se genere con un script, un formulario o una imagen de otro sitio web!!!

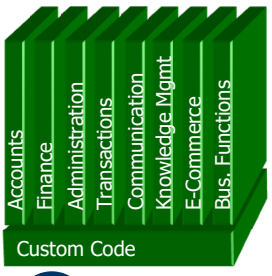
- Casi todas las aplicaciones web son vulnerables :-(

Seguridad en aplicaciones web

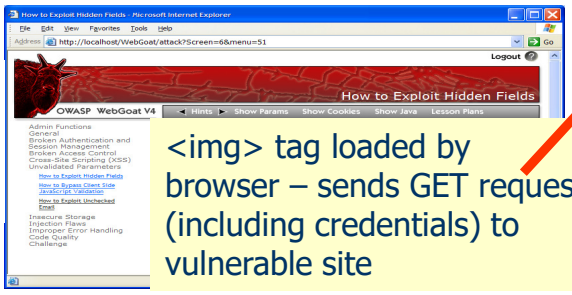
CSRF [Cross Site Request Forgery]

1 Attacker sets the trap on some website on the internet (or simply via an e-mail)

Application with CSRF vulnerability



2 While logged into vulnerable site, victim views attacker site



3 Vulnerable site sees legitimate request from victim and performs the action requested

SSRF [Server Side Request Forgery]

Vulnerabilidad que se produce cuando una aplicación accede a un recurso externo sin validar la URL suministrada por un usuario (incluso cuando se usan cortafuegos o VPNs).

EJEMPLOS

- Escaneo de puertos internos
- Acceso a ficheros y recursos locales: `file:///etc/passwd`
- Acceso a metadatos (cloud computing): IP "mágica" `http://169.254.169.254/`
- Abuso de servicios internos: RCE [Remote Code Execution] o DoS [Denial of Service]

Seguridad en aplicaciones web

Redirecciones no validadas [redirect / forward]

- Las redirecciones son frecuentes en aplicaciones web (y suelen enviar información del usuario), por lo que si no se validan correctamente, un atacante puede enviar a la víctima al sitio que quiera.

- Los "forwards" son también frecuentes y, en ocasiones, los parámetros determinan el destino, por lo que pueden utilizarse fraudulentamente para saltarse comprobaciones de seguridad (autentificación y control de acceso).

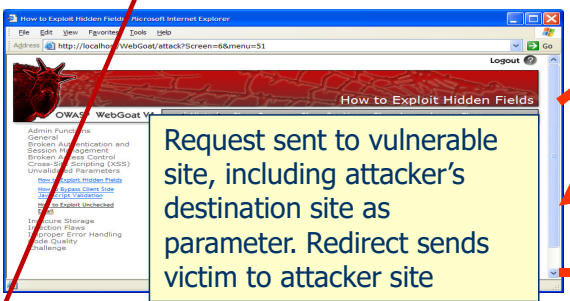
Seguridad en aplicaciones web

Redirecciones no validadas: redirect

1 Attacker sends attack to victim via email or webpage

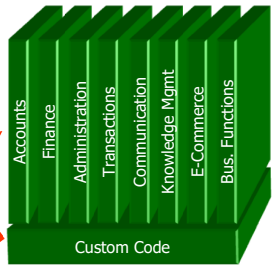
From: Internal Revenue Service
Subject: Your Unclaimed Tax Refund
Our records show you have an unclaimed federal tax refund. Please click here to initiate your claim.

2 Victim clicks link containing unvalidated parameter



Request sent to vulnerable site, including attacker's destination site as parameter. Redirect sends victim to attacker site

3 Application redirects victim to attacker's site



Evil Site

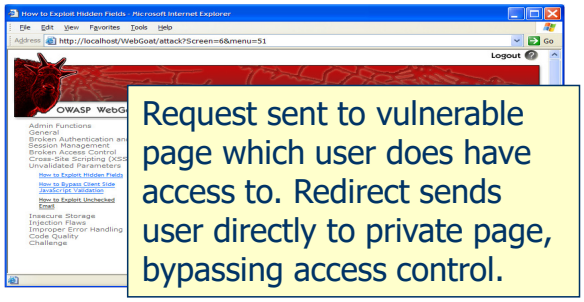
http://www.irs.gov/taxrefund/claim.jsp?year=2006& … &dest=www.evilsite.com

4 Evil site installs malware on victim, or phishes for private information

Seguridad en aplicaciones web

Redirecciones no validadas: forward

1 Attacker sends attack to vulnerable page they have access to



Request sent to vulnerable page which user does have access to. Redirect sends user directly to private page, bypassing access control.

```
public void sensitiveMethod(
HttpServletRequest request,
HttpServletResponse response) {

        // Do sensitive stuff here.
        ...
```

2 Application authorizes request, which continues to vulnerable page

Filter

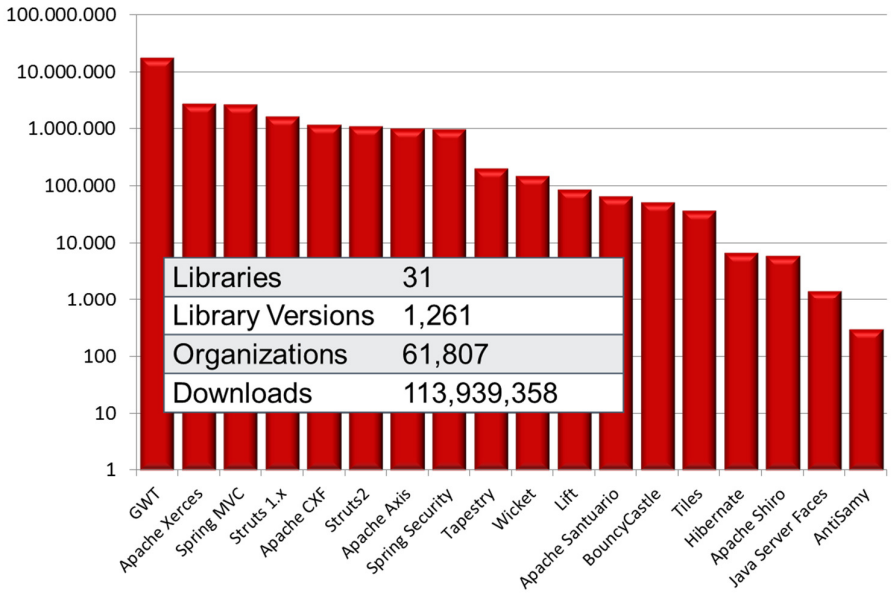
3 Forwarding page fails to validate parameter, sending attacker to unauthorized page, bypassing access control

```
public void doPost( HttpServletRequest request, HttpServletResponse response) {

    String target = request.getParameter( "dest" );
    ...
    request.getRequestDispatcher(target).forward(request, response);
```

Componentes con vulnerabilidades conocidas

Todo el mundo usa bibliotecas vulnerables…



https://www.aspectsecurity.com/news/press/the-unfortunate-reality-of-insecure-libraries

Seguridad en aplicaciones web

Componentes con vulnerabilidades conocidas

Automatización usando el plugin 'Versions' de Maven

Dependencies

Status	Group Id	Artifact Id	Current Version	Scope	Classifier	Type	Next Version	Next Incremental	Next Minor	Next Major
⚠	com.fasterxml.jackson.core	jackson-annotations	2.0.4	compile		jar		**2.0.5**	**2.1.0**	
⚠	com.fasterxml.jackson.core	jackson-core	2.0.4	compile		jar		**2.0.5**	**2.1.0**	
⚠	com.fasterxml.jackson.core	jackson-databind	2.0.4	compile		jar		**2.0.5**	**2.1.0**	
⚠	com.google.guava	guava	11.0	compile		jar		**11.0.1**	**12.0-rc1**	**12.0**
⚠	com.ibm.icu	icu4j	49.1	compile		jar				**50.1**
⚠	com.theoryinpractise	halbuilder	1.0.4	compile		jar		**1.0.5**		
⚠	commons-codec	commons-codec	1.3	compile		jar			**1.4**	
✓	commons-logging	commons-logging	1.1.1	compile		jar				
⚠	joda-time	joda-time	2.0	compile		jar			**2.1**	
⚠	net.sf.ehcache	ehcache-core	2.5.1	compile		jar		**2.5.2**	**2.6.0**	
⚠	org.apache.httpcomponents	httpclient	4.1.2	compile		jar		**4.1.3**	**4.2**	
⚠	org.apache.httpcomponents	httpclient-cache	4.1.2	compile		jar		**4.1.3**	**4.2**	
⚠	org.apache.httpcomponents	httpcore	4.1.2	compile		jar		**4.1.3**	**4.2**	
⚠	org.jdom	jdom	1.1	compile		jar		**1.1.2**		**2.0.0**
✓	org.slf4j	slf4j-api	1.7.2	provided		jar				

Most out of Date!

Details Developer Needs

Vulnerabilidades conocidas

EJEMPLO @ http://elvex.ugr.es

```
[Mon Sep 07 23:50:51.927154 2015] [core:info]
[pid 488:tid 2384] [client 77.58.42.200:52493]
 AH00128: File does not exist: …/xmlrpc.php
...
```

XML-RPC

- XML-RPC Pingback Vulnerability (para lanzar ataques DDoS)
- Brute Force Amplification Attack: system.multicall()

p.ej. WordPress, 302 vulnerabilidades!!! @ mayo'2017

https://www.cvedetails.com/vendor/2337/Wordpress.html

Seguridad en aplicaciones web

… y accesos "curiosos"

EJEMPLO @ http://elvex.ugr.es

```
[Tue Sep 15 20:54:19.264862 2015] [core:info]
[pid 6612:tid 2424] [client 113.53.215.62:37887]
 AH00128: File does not exist: …/Ringing.at.your.dorbell!,
 referer: http://google.com/search?q=2+guys+1+horse
```

```
[Sat Sep 19 23:26:36.354299 2015] [core:info]
[pid 2304:tid 2464] [client 24.199.207.74:8809]
 AH00561: Request header exceeds LimitRequestFieldSize: Cookie
```

```
31.10.155.27 - - [20/Sep/2015:20:02:38 +0200] "-" 408 –
```

HTTP 408 / REQUEST TIMEOUT

Seguridad en aplicaciones web

Top 3 @ OWASP'2021

#1 Control de acceso: Exposición de información sensible (CWE-200), CSRF (CWE-352)…

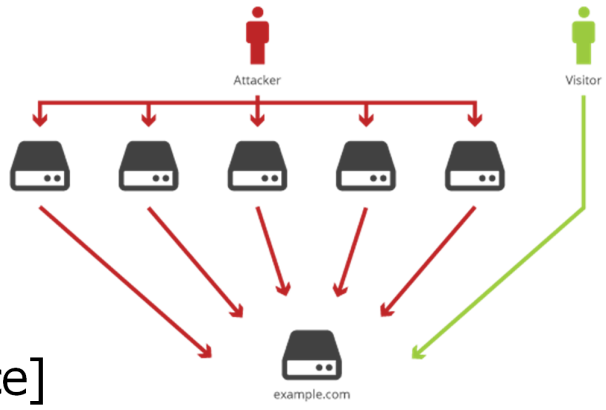
#2 Fallos criptográficos: Almacenamiento o transmisión de datos no encriptados, uso de HTTP en lugar de HTTPS, uso de algoritmos criptográficos con vulnerabilidades conocidas…

#3 Inyección: Consultas y llamadas no parametrizadas

https://cwe.mitre.org/

Ataques DoS [Denial of Service]

- Intentos de conseguir que una máquina o recurso no esté disponible para sus legítimos usuarios.
- Usualmente, se coordina el ataque usando distintas máquinas (a menudo, miles de direcccciones IP) para dificultar las medidas de defensa: **DDoS** [Distributed Denial-of-Service]

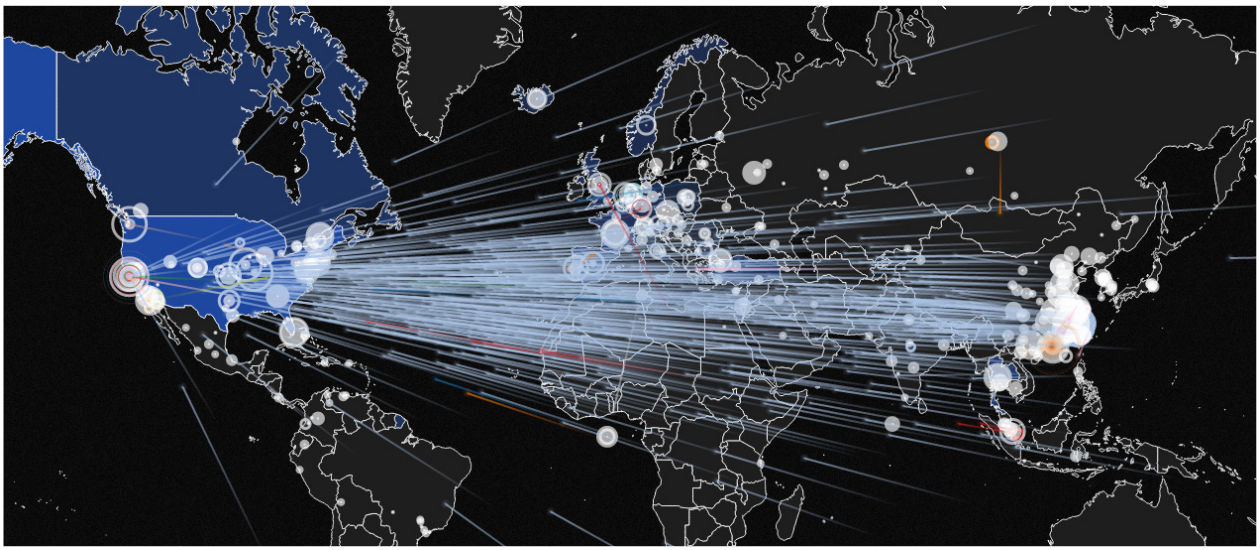


https://en.wikipedia.org/wiki/Denial-of-service_attack

Ataques DoS [Denial of Service]

Digital Attack Map

http://www.digitalattackmap.com/



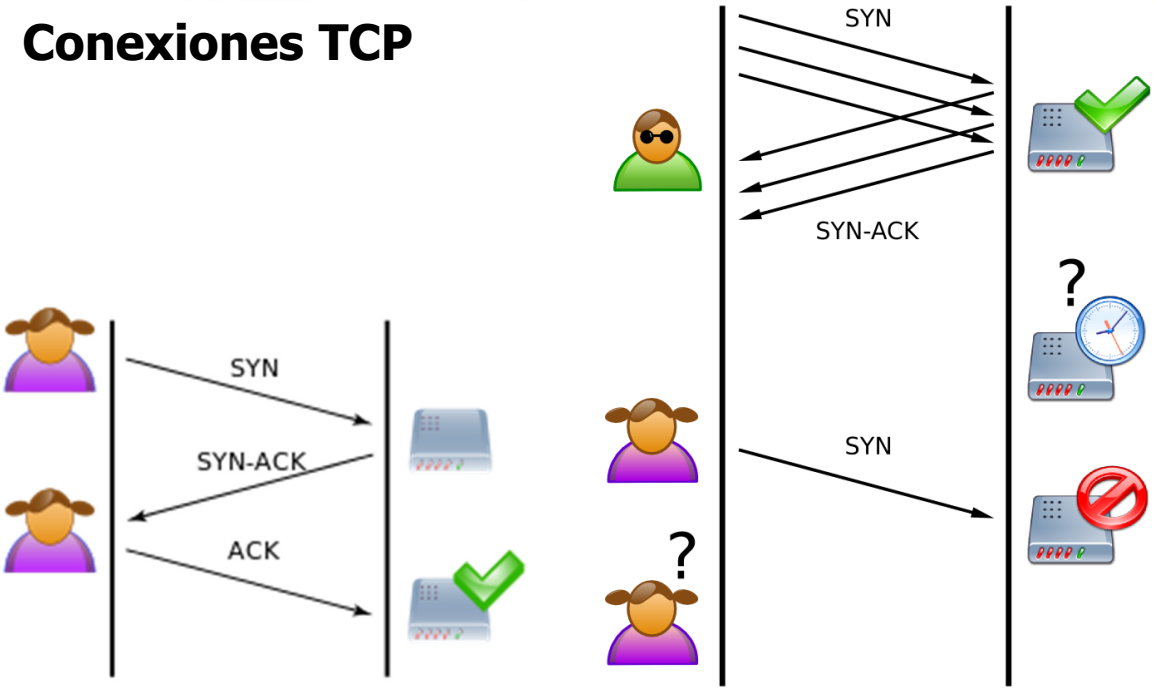
Ataques DoS [Denial of Service]

Tipos de ataques de denegación de servicio

- **Conexiones TCP:** Se ocupan todas las conexiones TCP disponibles para impedir que los usuarios legítimos de la aplicación puedan conectarse a ella.
- **Volumétricos:** Consumen todo el ancho de banda disponible para congestionar la red (hasta 400 Gbps!!)
- **Por fragmentación:** Inundan la víctima con fragmentos de paquetes TCP/UDP.
- **Aplicación:** Específicos para una aplicación web…

Ataques DoS [Denial of Service]

Conexiones TCP



Conexión normal SYN flood

https://en.wikipedia.org/wiki/SYN_flood

Ataques DoS [Denial of Service]

Ejemplos

- **SYN flood:** Envío de paquetes TCP/SYN, falsificando la dirección IP del remitente, cada uno de los cuales se trata como una solicitud de conexión… dejando conexiones medio abiertas a falta de la respuesta de confirmación.
- **Slowloris:** Ataque a servidores web en los que se envían solicitudes parciales HTTP que nunca se completan para mantener conexiones abiertas el mayor tiempo posible.
- **Teardrop:** Envío de fragmentos IP desordenados

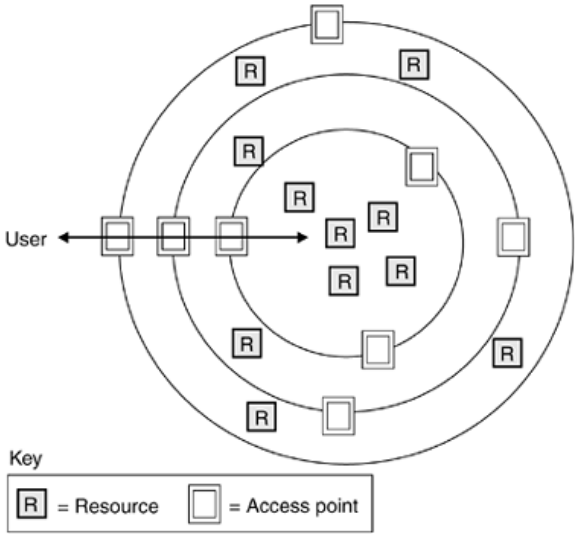
Ataques DoS [Denial of Service]

Ejemplos

- **Smurf** ["pitufo"]: Broadcast de paquetes ICMP en los que se falsea la IP fuente para que la respuesta inunde con tráfico a la víctima.
- **Ping of death:** Envío de "pings" mal formados
- **Reflexión DNS:** Se envían [pequeñas] solicitudes DNS a un servidor de nombres para que la respuesta [mayor] se la envíe a la víctima.
- **Chargen:** Antiguo servicio de prueba que genera una secuencia aleatoria de caracteres (y puede usarse para amplificar un ataque DoS por reflexión)

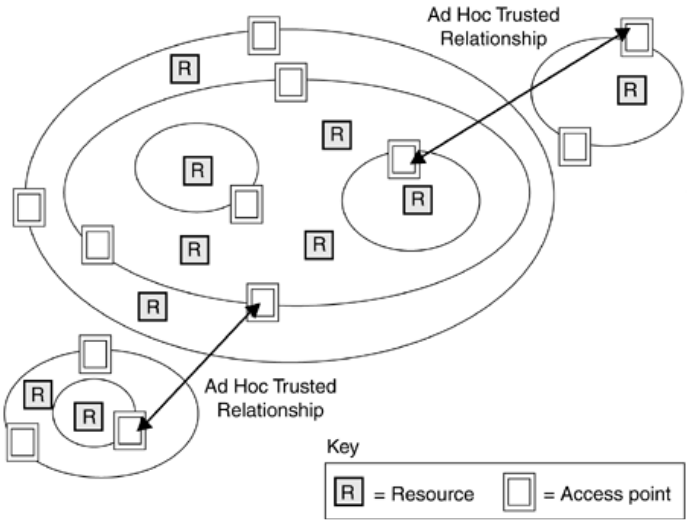
Seguridad en sistemas distribuidos

Onion model



Seguridad en sistemas distribuidos

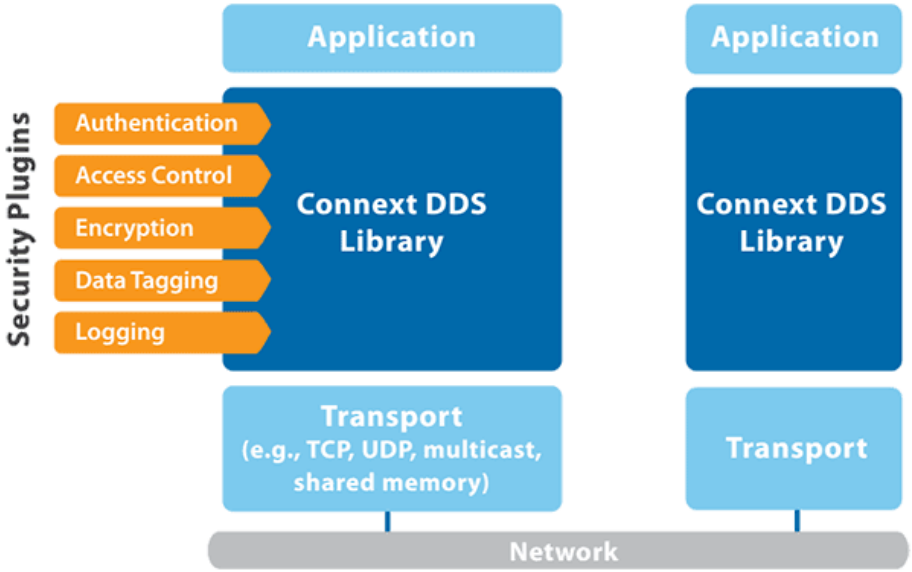
Boiling lava model



Seguridad en sistemas distribuidos

Middleware

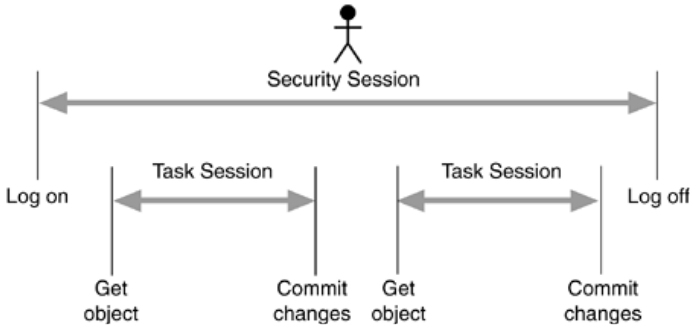
rti Your systems. Working as one.



http://www.omg.org/spec/DDS-SECURITY/

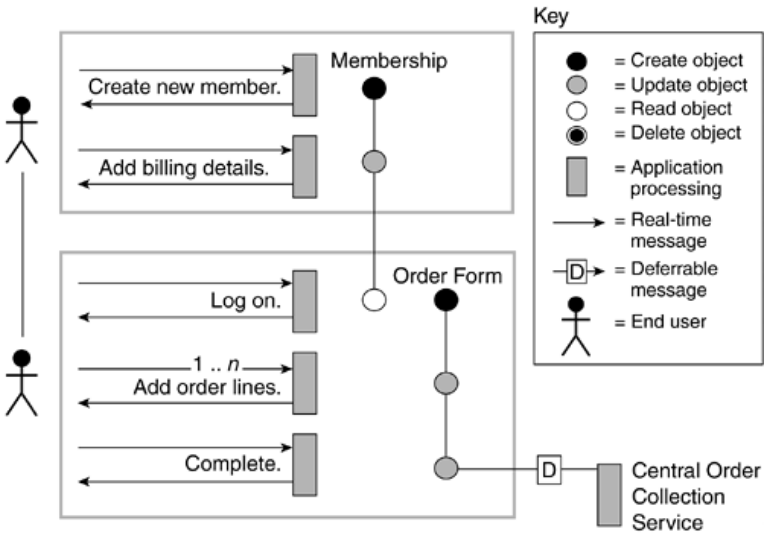
Seguridad en sistemas distribuidos

Transacciones largas



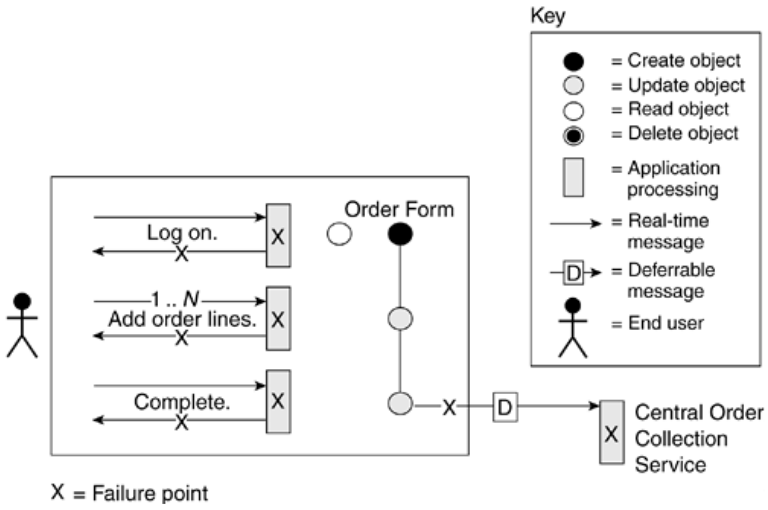
Seguridad en sistemas distribuidos

Diagrama de tareas/mensajes



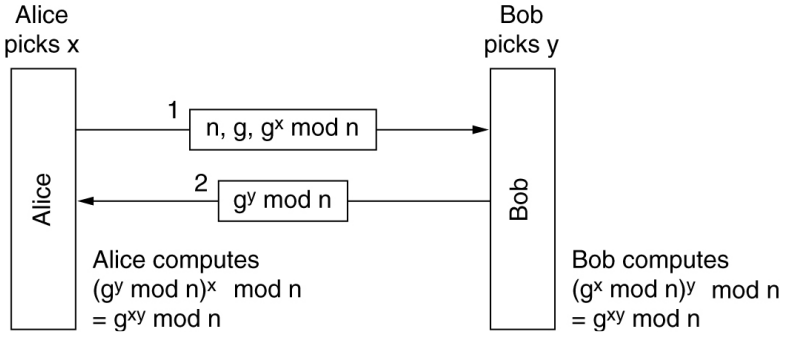
Seguridad en sistemas distribuidos

Identificación de puntos de fallo

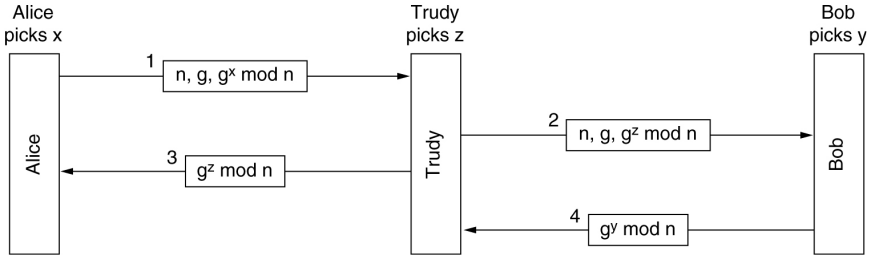


Apéndice
Intercambio de claves

Algoritmo de Diffie-Hellman

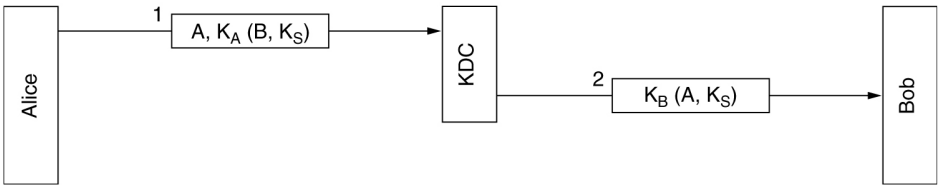


Ataque de un intermediario ("bucket brigade"):

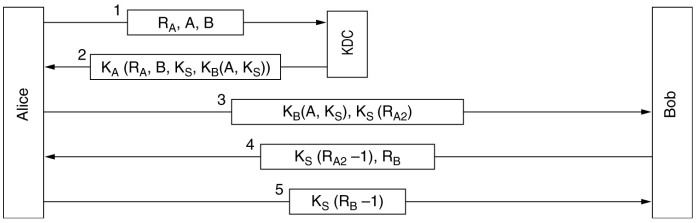


Apéndice
Intercambio de claves

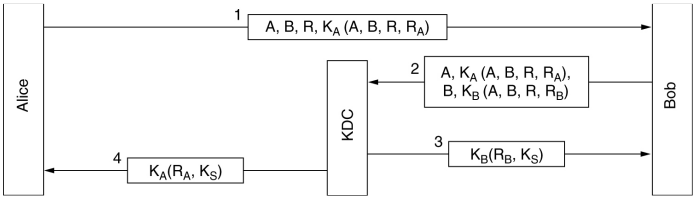
Uso de un centro de distribución de claves (KDC):



- Protocolo de Needham-Schroeder:



- Protocolo de Otway-Rees:



Bibliografía

- Sean Smith & John Marchesini: **The Craft of System Security**. Addison-Wesley Professional, 2007, ISBN 0-321-43483-8.
- John E. Canavan: **Fundamentals of Network Security**. Artech House, 2001. ISBN 1-58053-176-8.
- Amparo Fúster Savater, Dolores de la Guía Martínez, Luis Hernández Encinas, Fausto Montoya Vitini & Jaime Muñoz Masqué: **Técnicas criptográficas de protección de datos**. RA-MA, 1997. ISBN 84-7897-288-9.
- Jesús E. Díaz Verdejo; Juan Manuel López Soler & Pedro García Teodoro: **Transmisión de datos y redes de computadores**. Prentice-Hall, 2003. ISBN 84-205-3919-8.
- William Stallings: **Comunicaciones y redes de computadores**. Prentice-Hall, 2004 [7ª edición]. ISBN 84-205-4110-9.
- Andrew S. Tanenbaum: **Redes de computadoras**. Prentice-Hall, 2003 [4ª edición]. ISBN 970-260-162-2.