C|EH™
Certified Ethical Hacker

Module 20

Cryptography

Module Objectives

Module Objectives

- Understanding Cryptography Concepts
- Overview of Encryption Algorithms
- Cryptography Tools
- Understanding Public Key Infrastructure (PKI)
- Understanding Email Encryption
- Understanding Disk Encryption
- Understanding Cryptography Attacks
- Cryptanalysis Tools

Module Flow

1. Cryptography Concepts
2. Encryption Algorithms
3. Cryptography Tools
4. Public Key Infrastructure (PKI)
5. Email Encryption
6. Disk Encryption
7. Cryptanalysis
8. Countermeasures

Cryptography

- Cryptography is the **conversion of data** into a scrambled code that is encrypted and sent across a private or public network
- Cryptography is used to protect confidential data such as **email messages**, chat sessions, **web transactions**, personal data, **corporate data**, e-commerce applications, etc.

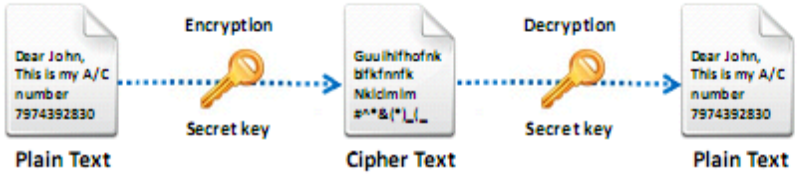
Objectives of Cryptography

- Confidentiality
- Integrity
- Authentication
- Nonrepudiation

Types of Cryptography

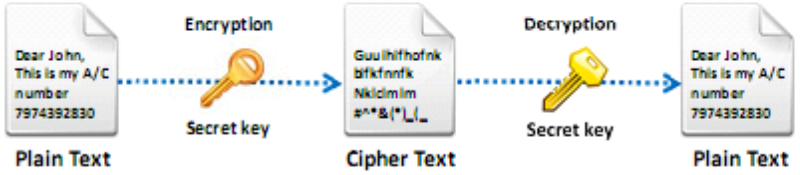
Symmetric Encryption

Symmetric encryption (secret-key, shared-key, and private-key) **uses the same key** for encryption as it does for decryption



Asymmetric Encryption

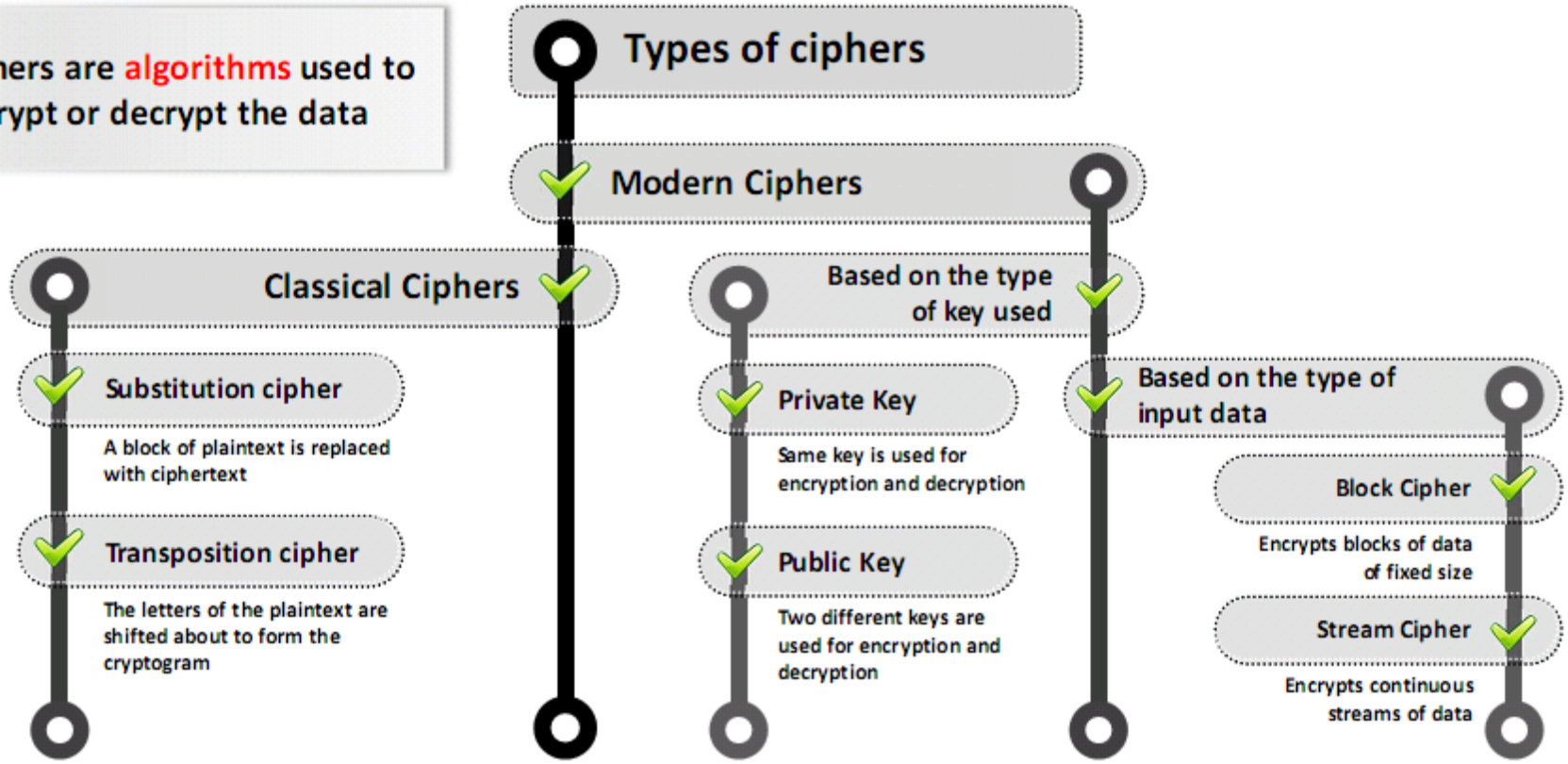
Asymmetric encryption (public-key) **uses different encryption keys** for encryption and decryption. These keys are known as public and private keys



Module Flow

Ciphers

Ciphers are algorithms used to encrypt or decrypt the data



Data Encryption Standard (DES)

DES is designed to **encipher** and **decipher** blocks of data consisting of **64 bits** under control of a 56-bit key

DES is the **archetypal block cipher**—an algorithm that takes a fixed-length string of plaintext bits and transforms it into a ciphertext bitstring of the same length

Due to the **inherent weakness** of DES with today's technologies, some organizations repeat the process thrice(3DES) for added strength, until they can afford to update their equipment to AES capabilities

Advanced Encryption Standard (AES)

- AES is a **symmetric-key** algorithm that secures sensitive but unclassified material by the US government agencies
- AES is an **iterated block cipher**, which works by repeating the same operation **multiple** times
- It has a **128-bit** block size, with key sizes of 128, 192, and 256 bits, respectively, for AES-128, AES-192, and AES-256

AES Pseudocode

```
Cipher (byte in[4*Nb], byte out[4*Nb], word
w[Nb*(Nr+1)])
begin
  byte state[4,Nb]
  state = in
  AddRoundKey(state, w)
   for round = 1 step 1 to Nr-1
     SubBytes(state)
     ShiftRows(state)
     MixColumns(state)
     AddRoundKey(state, w+round*Nb)
   end for
  SubBytes(state)
  ShiftRows(state)
  AddRoundKey(state, w+Nr*Nb)
  out = state
end
```

RC4, RC5, and RC6 Algorithms

RC4

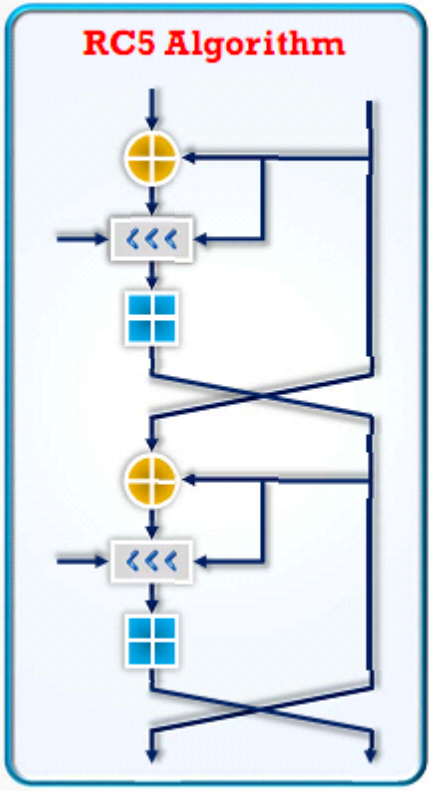
- A variable key size **symmetric key stream cipher** with byte-oriented operations and is based on the use of a random permutation

RC5

- It is a **parameterized algorithm** with a variable block size, a variable key size, and a variable number of rounds. The key size is **128-bits**

RC6

- RC6 is a **symmetric key block cipher** derived from RC5 with two additional features:
  - Uses **integer multiplication**
  - Uses **four 4-bit working registers** (RC5 uses two 2-bit registers)



Twofish

- This algorithm was one of the five finalists to replace DES for the US Government, but it was not chosen

- It uses a block size of 128 bits and key sizes up to 256 bits. It is a Feistel cipher

- It was designed by Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall, and Niels Ferguson

The DSA and Related Signature Schemes

Digital Signature Algorithm

FIPS 186-2 specifies the Digital Signature Algorithm (DSA) that may be used in the generation and verification of digital signatures for sensitive, unclassified applications

Digital Signature

The digital signature is computed using a set of rules (i.e., the DSA) and a set of parameters such that the identity of the signatory and integrity of the data can be verified

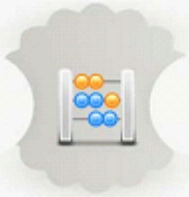
Each entity creates a public key and a corresponding private key

1. Select a prime number q such that $2^{159} < q < 2^{160}$
2. Choose t so that $0 \leq t \leq 8$
3. Select a prime number p such that $2^{511+64t} < p < 2^{512+64t}$ with the additional property that q divides $(p-1)$
4. Select a generator α of the unique cyclic group of order q in Z^*_p
5. To compute α, select an element g in Z^*_p and compute $g^{(p-1)/q} \bmod p$
6. If $\alpha = 1$, perform step five again with a different g
7. Select a random a such that $1 \leq a \leq q-1$
8. Compute $y = \alpha^a \bmod p$

Following are the public keys: p, q, α, y and a is the private key.

Rivest Shamir Adleman (RSA)

RSA is an **Internet encryption and authentication system** that uses an algorithm developed by Ron Rivest, Adi Shamir, and Leonard Adleman



It is widely used and is one of the **de-facto encryption standard**

It uses **modular arithmetic** and **elementary number theories** to perform computations using two large prime numbers

Rivest Shamir Adleman (RSA) (Cont'd)

The RSA Signature Scheme

Algorithm Key generation for the RSA signature scheme

SUMMARY: each entity creates an RSA public key and a corresponding private key. Each entity A should do the following:

1. Generate two large distinct random primes p and q, each roughly the same size.
2. Compute $n = pq$ and $\phi = (p-1)(q-1)$.
3. Select a random integer e, $1 < e < \phi$, such that $\gcd(e, \phi) = 1$.
4. Use the extended Euclidean algorithm (Algorithm 2.107) to compute the unique integer d, $1 < d < \phi$, such that $ed \equiv 1 \pmod{\phi}$.
5. A's public key is (n, e); A's private key is d.

Algorithm RSA signature generation and verification

SUMMARY: entity A signs a message $m \in \mathcal{M}$. Any entity B can verify A's signature and recover the message m from the signature.

1. *Signature generation.* Entity A should do the following:
   (a) Compute $\tilde{m} = R(m)$, an integer in the range $[0, n-1]$.
   (b) Compute $s = \tilde{m}^d \bmod n$.
   (c) A's signature for m is s.
2. *Verification.* To verify A's signature s and recover the message m, B should:
   (a) Obtain A's authentic public key (n, e).
   (b) Compute $\tilde{m} = s^e \bmod n$.
   (c) Verify that $\tilde{m} \in \mathcal{M}_R$; if not, reject the signature.
   (d) Recover $m = R^{-1}(\tilde{m})$.

Example of RSA Algorithm

```
P  = 61 <= first prime number (destroy this after computing
                               E and D)
Q  = 53 <= second prime number (destroy this after computing
                                E and D)
PQ = 3233 <= modulus (give this to others)
E  = 17   <= public exponent (give this to others)
D  = 2753 <= private exponent (keep this secret!)

Your public key is (E,PQ).
Your private key is D.

The encryption function is: encrypt(T) = (T^E) mod PQ
                                       = (T^17) mod 3233

The decryption function is: decrypt(C) = (C^D) mod PQ
                                       = (C^2753) mod 3233

To encrypt the plaintext value 123, do this:

encrypt(123) = (123^17) mod 3233
             = 337587917446653715596592958817679803 mod 3233
             = 855

To decrypt the cipher text value 855, do this:

decrypt(855) = (855*2753) mod 3233
             = 123
```

Diffie-Hellman

- A cryptographic protocol that allows two parties to establish a **shared key** over an **insecure channel**
- Developed and published by **Whitfield Diffie** and **Martin Hellman** in **1976**
- Actually was independently developed a few years earlier by **Malcolm J. Williamson** of the **British Intelligence Service**, but it was classified

Diffie-Hellman Algorithm

- The system has two parameters called *p* and *g*
  - Parameter *p* is a **prime number**
  - Parameter *g* (usually called a generator) is an integer less than *p*, with the following property: for every number *n* between 1 and *p*-1 inclusive, there is a power *k* of *g* such that $n = g^k \bmod p$
- Many cryptography textbooks use the fictitious characters "**alice**" and "**bob**" to illustrate cryptography, and we will do that here as well:
  - Alice generates a random private value *a* and Bob generates a random private value *b*. Both *a* and *b* are drawn from the **set of integers**
  - They derive their public values using parameters *p* and *g* and their private values. Alice's public value is $g^a \bmod p$ and Bob's public value is $g^b \bmod p$
  - They exchange their public values
  - Alice computes $g^{ab} = (g^b)^a \bmod p$, and Bob computes $g^{ba} = (g^a)^b \bmod p$
  - Since $g^{ab} = g^{ba} = k$, Alice and Bob now have a shared secret key **k**

Message Digest (One-Way Hash) Functions

- Hash functions **calculate a unique fixed-size bit string** representation called a message digest of any arbitrary block of information

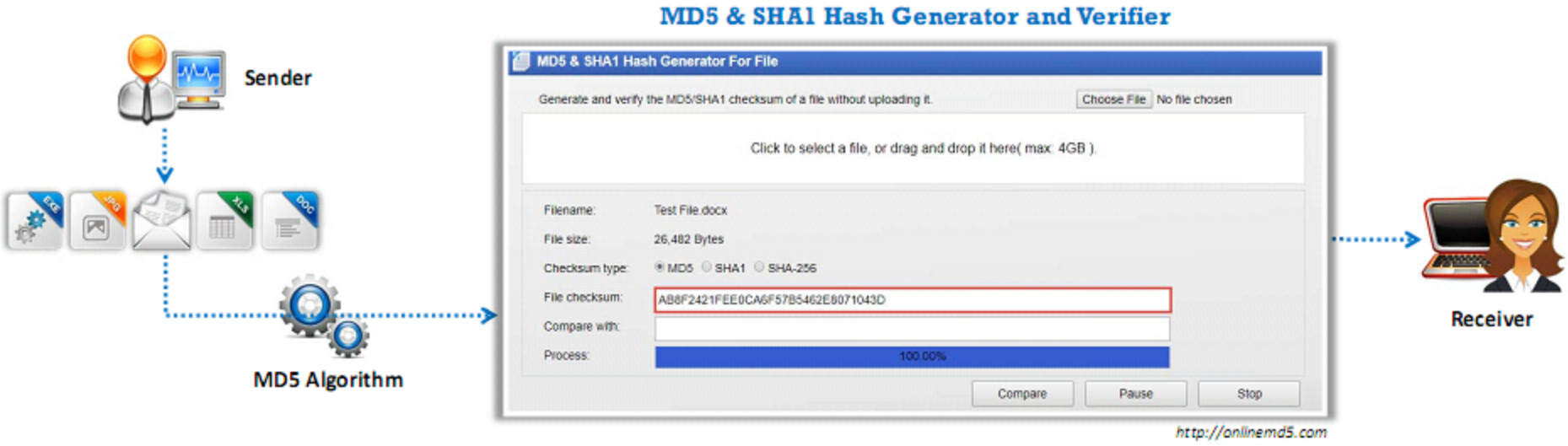
- If any given bit of the function's input is changed, then every output bit has a **50 percent** chance of changing

- It is computationally infeasible to have two files with the **same message digest value**

Note: Message digests are also called one-way hash functions because they cannot be reversed

Message Digest Function: MD5

- MD5 algorithm takes a message of **arbitrary length** as the input and then outputs a **128-bit fingerprint** or message digest of the input
- MD5 is not collision resistant; use of latest algorithms such as **SHA-2** and **SHA-3** is recommended
- It is still deployed for digital signature applications, file integrity checking, and storing passwords



Message Digest Function: Secure Hashing Algorithm (SHA)

It is an algorithm to generate cryptographically secure one-way hash, published by the **National Institute of Standards and Technology** as a **US Federal Information Processing Standard**

SHA1

It produces a **160-bit digest** from a message with a maximum length of **(264 – 1) bits**, and it resembles the MD5 algorithm

SHA2

It is a family of two similar hash functions with different block sizes, namely, **SHA-256** that uses **32-bit words** and **SHA-512** that uses **64-bit words**

SHA3

SHA-3 uses the **sponge construction** in which message blocks are **XORed** into the initial bits of the state, which is then invertibly permuted

RIPEMD-160

- **RACE Integrity Primitives Evaluation Message Digest** (RIPEMD) is a **160-bit hash algorithm** developed by Hans Dobbertin, Antoon Bosselaers, and Bart Preneel

- There exist 128, 256, and 320-bit versions of this algorithm, called **RIPEMD-128**, **RIPEMD-256**, and **RIPEMD-320**, respectively

- The compression function consists of **80 stages made up of 5 blocks** that execute **16 times each**

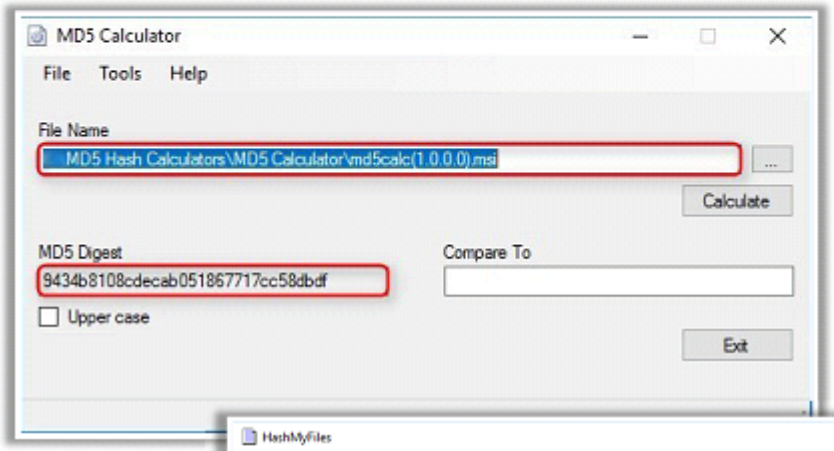
- This process **repeats twice** by combining the results at the bottom using **modulo 32 addition**

HMAC

1. HMAC is a type of **message authentication code** (MAC) that makes use of **cryptographic key** with a combination of a cryptographic hash function
2. It is widely used to verify the **integrity of the data** and **authentication** of a message
3. This algorithm includes an embedded hash function such as **SHA-1** or **MD5**
4. The strength of HMAC depends on the **embedded hash function**, key size, and the size of the hash output
5. As HMAC executes the underlying hash function twice, it protects from various **length extension attacks**

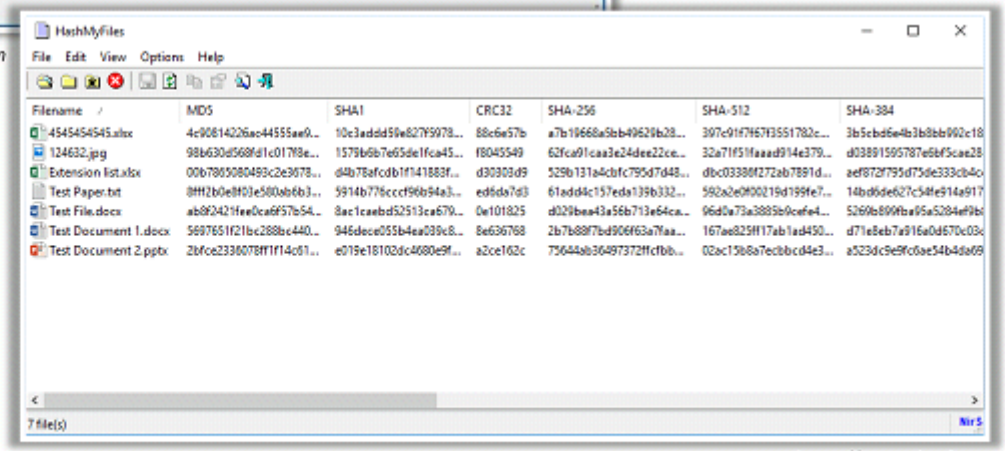
Module Flow

MD5 Hash Calculators



http://www.bullzip.com

MD5 Calculator



HashMy Files

https://www.nirsoft.net

HashCalc
https://www.slavasoft.com

Hash Calculator
https://www.mcafee.com

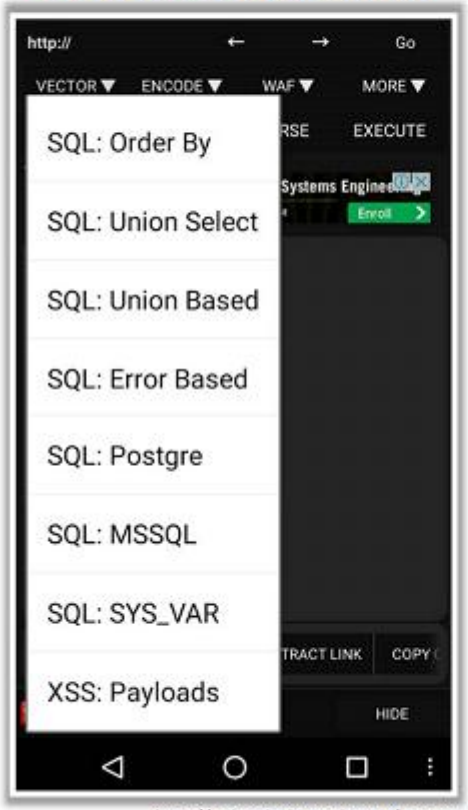
HashTool
https://www.digitalvolcano.co.uk

OnlineMD5.com
http://onlinemd5.com

MD5 Hash generator
https://hash.online-convert.com

Hash Calculators for Mobile

MD5 Hash Calculator

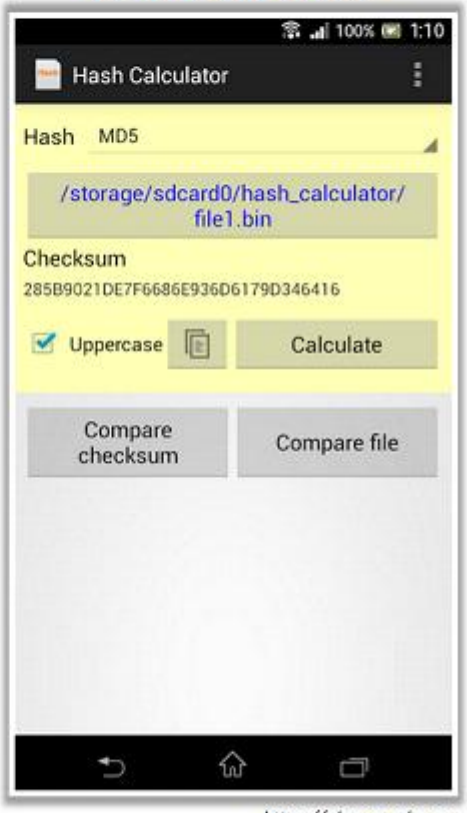


http://md5calculator.chromefans.org

Hash Droid

https://play.google.com

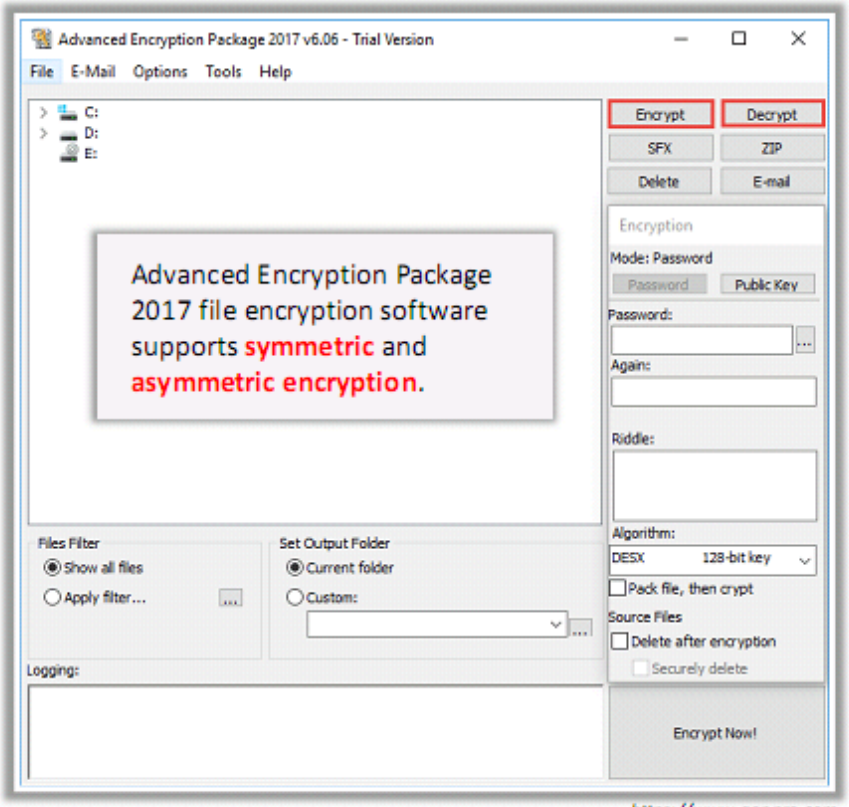
Hash Calculator



https://play.google.com

Cryptography Tools: Advanced Encryption Package 2017 and BCTextEncoder

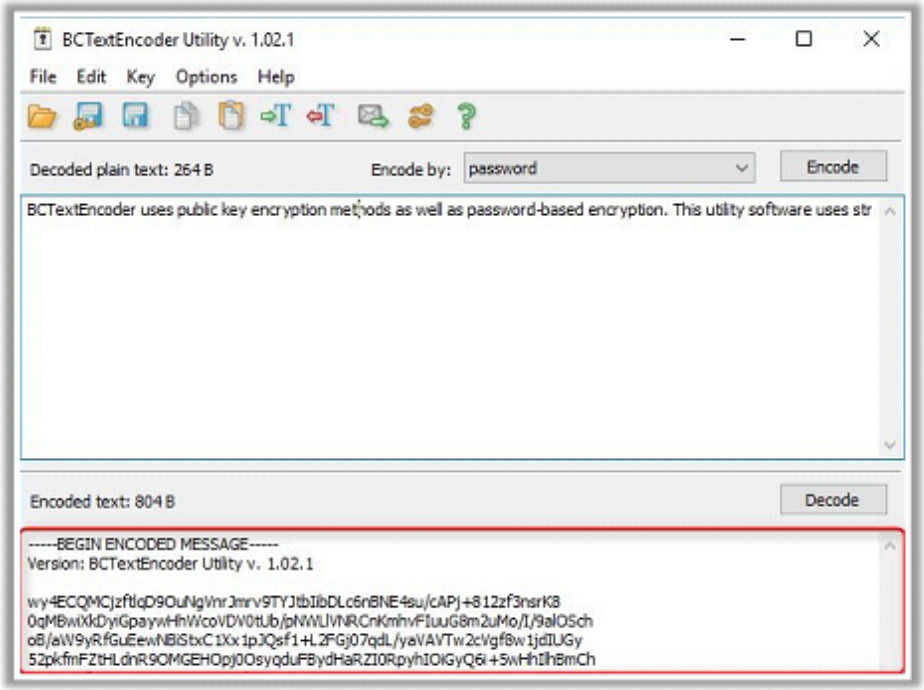
Advanced Encryption Package 2017



https://www.aeppro.com

BCTextEncoder

- It encrypts confidential text in your message
- It uses strong symmetric and public key algorithms for data encryption



https://www.jetico.com

Cryptography Tools

AxCrypt
https://www.axcrypt.net

Folder Lock
www.newsoftwares.net

CryptoExpert 8
http://www.cryptoexpert.com

CertainSafe
https://certainsafe.com

VeraCrypt
https://veracrypt.codeplex.com

Cryptainer LE Free Encryption Software
http://www.cypherix.com

CryptoForge
https://www.cryptoforge.com

winAES
https://winaes.com

EncryptOnClick
https://www.2brightsparks.com

GNU Privacy Guard
https://www.gnupg.org

Steganos Safe 19
https://www.steganos.com

Secure IT
http://www.cypherix.com

AES Crypt
https://www.aescrypt.com

Steganos LockNote
https://sourceforge.net

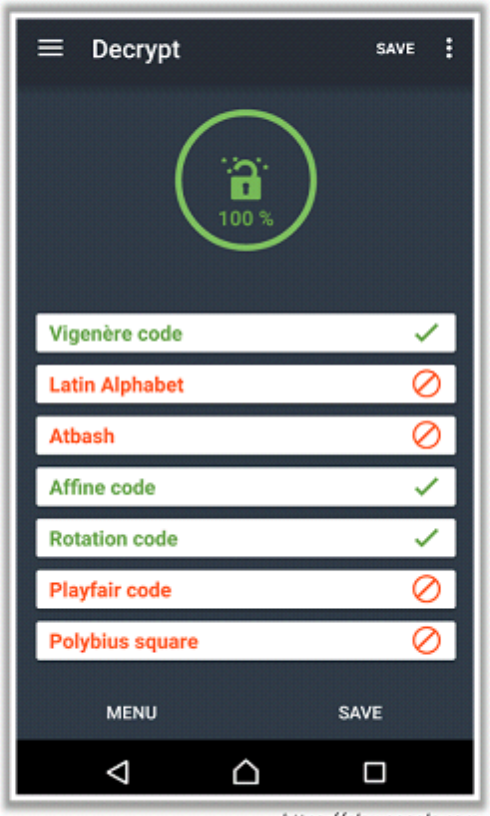
Autokrypt
http://www.hiteksoftware.com

Cryptography Tools for Mobile

Secret Space Encryptor

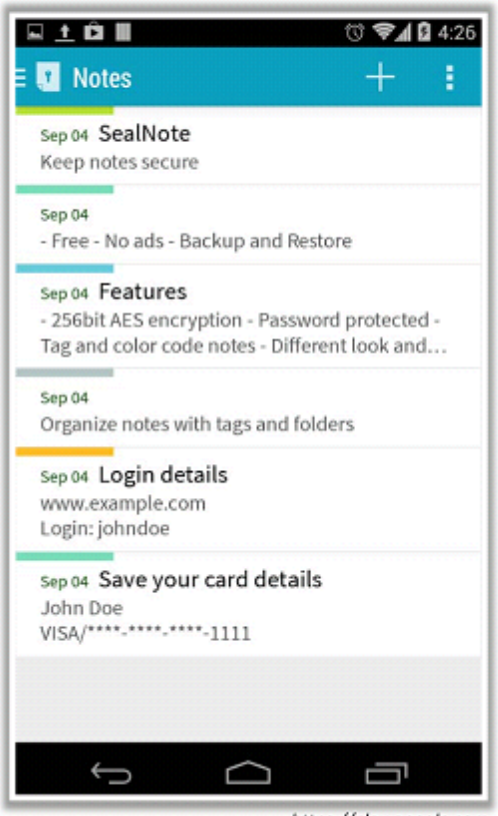
https://paranoiawarks.mobi

Decrypto



https://play.google.com

SealNote Secure Encrypted Note



https://play.google.com

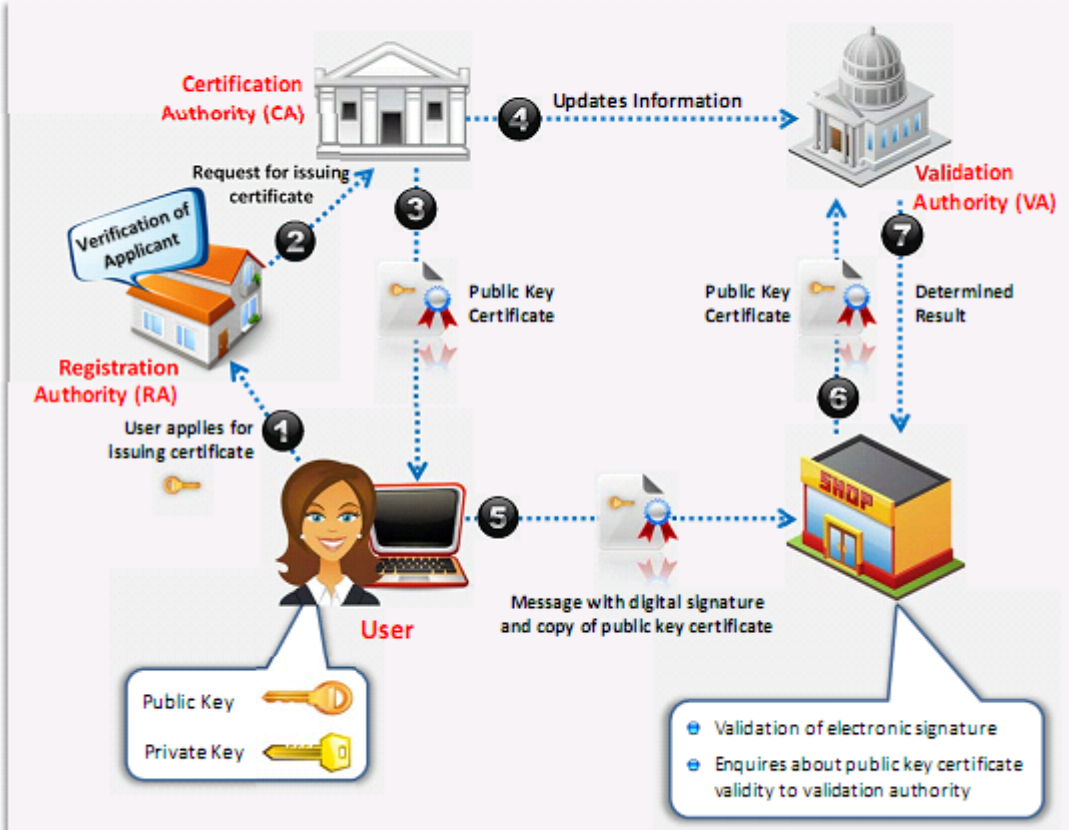
Module Flow

Public Key Infrastructure (PKI)

PKI is a **set of hardware**, **software**, **people**, **policies**, and **procedures** required to create, manage, distribute, use, store, and revoke **digital certificates**

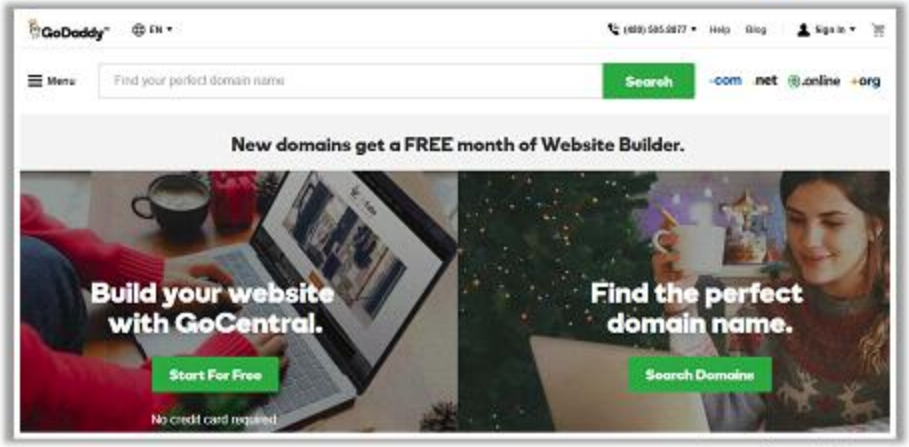
Components of PKI

- **Certificate Management System**: Generates, distributes, stores, and verifies certificates
- **Digital Certificates**: Establishes credentials of a person when doing online transactions
- **Validation Authority (VA)**: Stores certificates (with their public keys)
- **Certificate Authority (CA)**: Issues and verifies digital certificates
- **End User**: Requests, manages, and uses certificates
- **Registration Authority (RA)**: Acts as the verifier for the certificate authority



Certification Authorities

https://www.comodo.com



https://in.godaddy.com

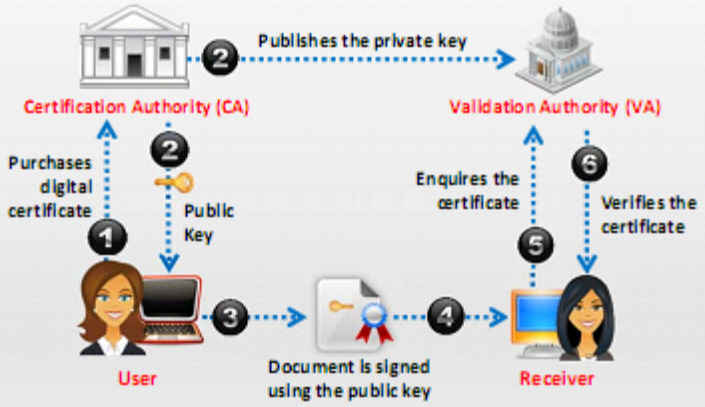
https://www.identrust.com

https://www.websecurity.symantec.com

Signed Certificate (CA) Vs. Self Signed Certificate

Signed Certificate

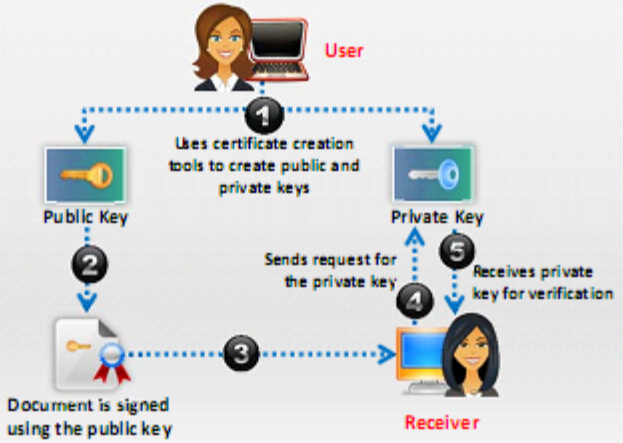
- User approaches a trustworthy **Certification Authority** (CA) and purchases digital certificate
- User gets the **public key** from the CA; he signs the document using it
- The signed document is delivered to the receiver
- The receiver can verify the certificate by enquiring in **Validation Authority** (VA)
- VA verifies the certificate to the receiver, but it does not **share private key**



Self-signed Certificate

- User creates public and private keys using a tool such as **Adobe Reader**, **Java's keytool**, **Apple's Keychain**, etc.
- User uses public key to **sign the document**
- The **self-signed document** is delivered to the receiver
- The receiver request the user for his **private key**
- User **shares the private key** with the receiver

Note: The certificate verification rarely occurs due to necessity of disclosing the private key

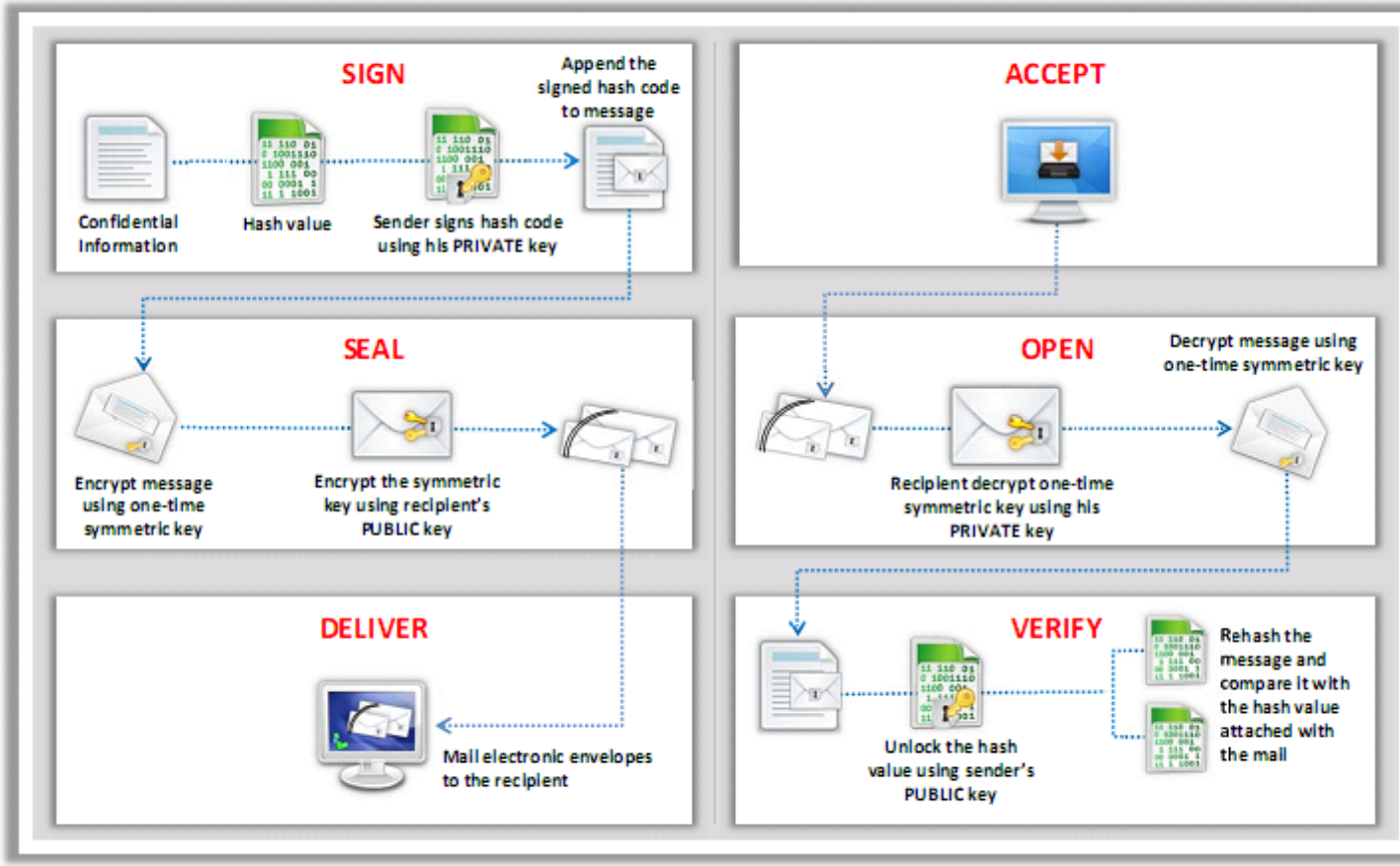


Module Flow

1. Cryptography Concepts
2. Encryption Algorithms
3. Cryptography Tools
4. Public Key Infrastructure (PKI)
5. Email Encryption
6. Disk Encryption
7. Cryptanalysis
8. Countermeasures

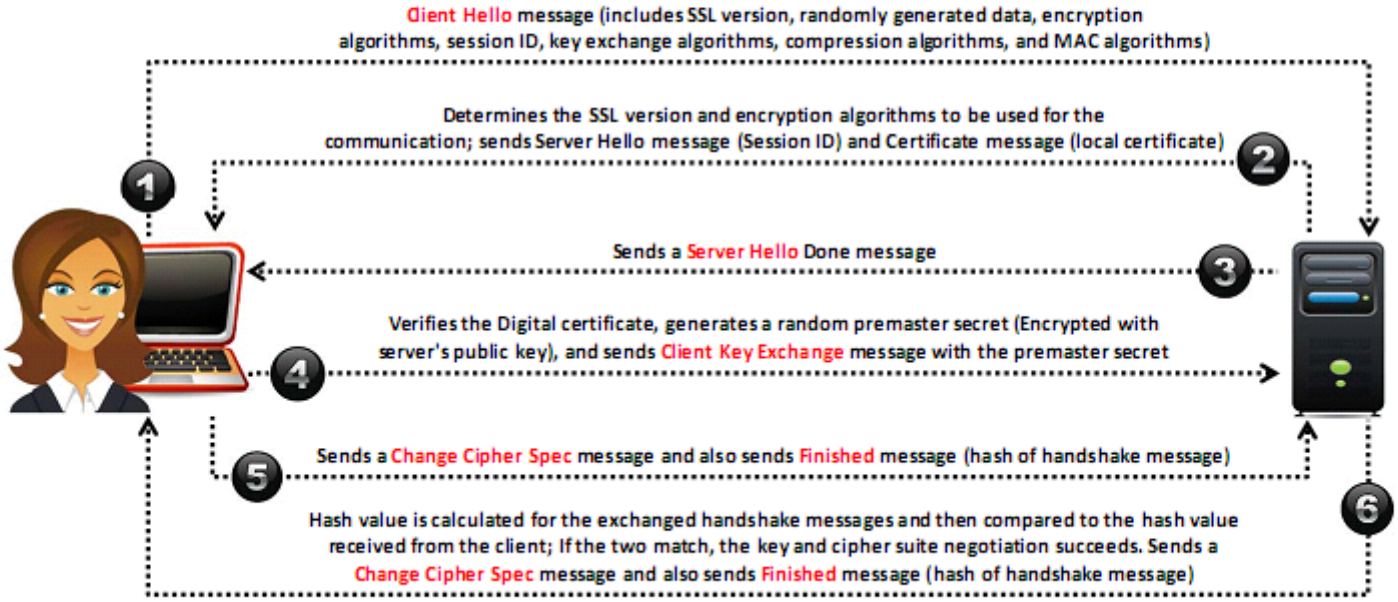
Digital Signature

- Digital signature uses asymmetric cryptography to simulate the security properties of a **signature in digital, rather than written form**
- A digital signature may be further protected, by **encrypting the signed email** for confidentiality



Secure Sockets Layer (SSL)

- SSL is an application layer protocol developed by Netscape for **managing the security** of a message transmission on the Internet
- It uses **RSA asymmetric (public key) encryption** to encrypt data transferred over SSL connections



Transport Layer Security (TLS)

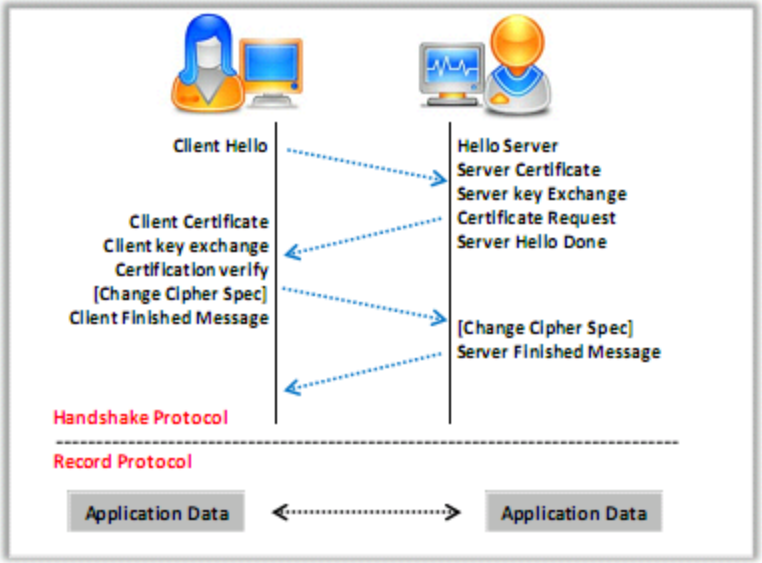
- TLS is a protocol **to establish a secure connection** between a client and a server and ensure privacy and integrity of information during transmission
- It uses the **RSA algorithm** with 1024 and 2048 bit strengths

TLS Handshake Protocol

It allows the client and server to authenticate each other, select encryption algorithm, and exchange symmetric key prior to data exchange

TLS Record Protocol

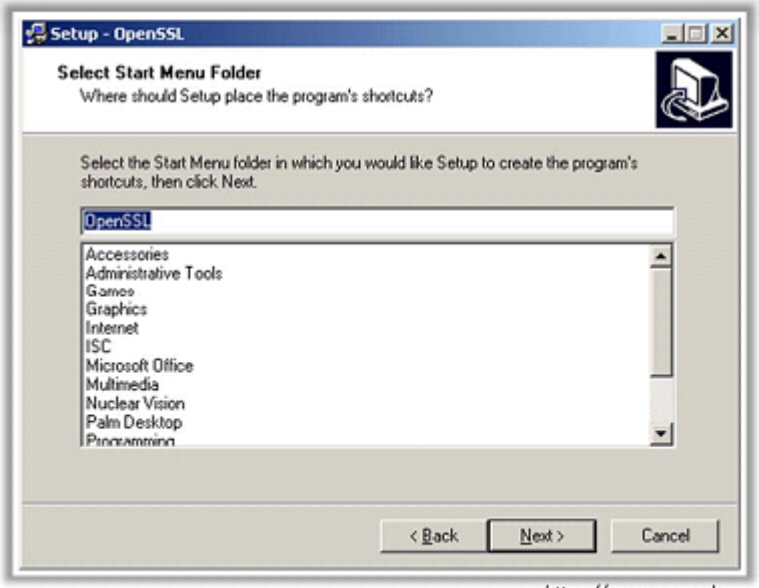
It provides secured connections with an encryption method such as DES



Cryptography Toolkits: OpenSSL and Keyczar

OpenSSL

- OpenSSL is an open source cryptography toolkit implementing the **Secure Sockets Layer (SSL v2/v3)** and **Transport Layer Security (TLS v1)** network protocols and related cryptography standards required by them



https://www.openssl.org

Keyczar

- Keyczar is an open source cryptographic toolkit designed to make it easier and safer for developers to use **cryptography in their applications**
- It **supports authentication** and **encryption** with both symmetric and asymmetric keys

Features:

- Key rotation and versioning
- Safe default algorithms, modes, and key lengths
- Automated generation of initialization vectors and ciphertext signatures
- Java, Python, and C++ implementations
- International support in Java

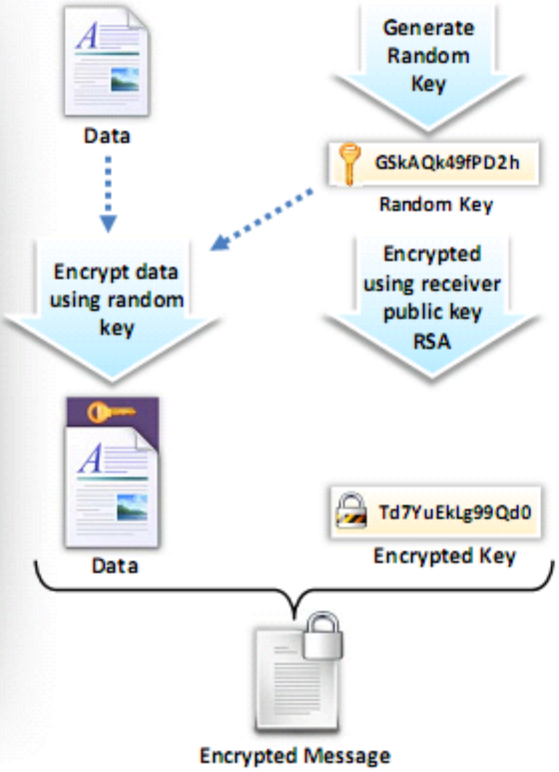
https://github.com

Pretty Good Privacy (PGP)

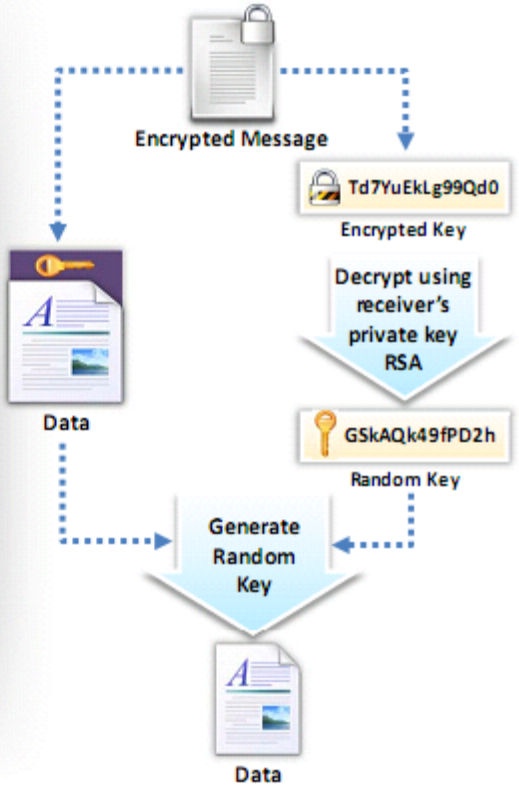
Pretty Good Privacy

- PGP is a protocol used to **encrypt** and **decrypt** data that provides **authentication** and **cryptographic privacy**
- It is often used for data **compression**, **digital signing**, encryption and decryption of **messages**, **emails**, **files**, **directories**, and to enhance privacy of email communications
- It combines the best features of both **conventional** and **public key cryptography** and is therefore known as **hybrid cryptosystem**

PGP Encryption



PGP Decryption



Module Flow

Disk Encryption

Confidentiality

Disk encryption protects **confidentiality of the data** stored on disk by converting it into an unreadable code using disk encryption software or hardware

- Privacy
- Passphrase
- Hidden Volumes

Encryption

It works in a similar way as **text message encryption** and protects data even when the OS not active

- Volume Encryption

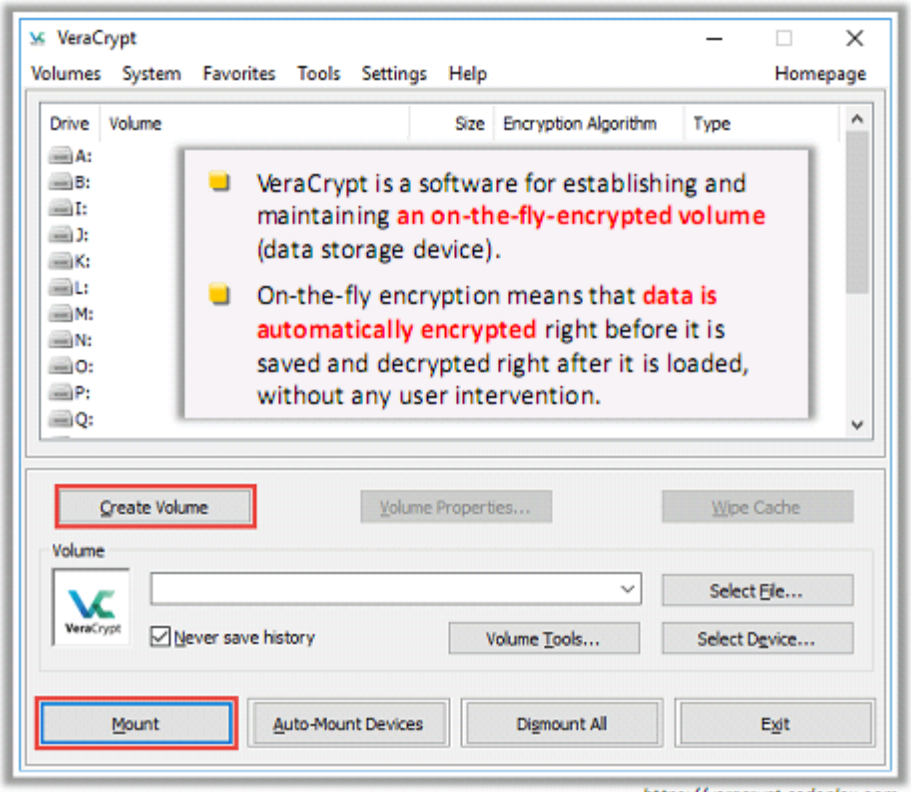
Protection

With the use of an encryption program for your disk, you can **safeguard any information** to burn onto the disk, and keep it from falling into the wrong hands

- Blue Ray
- DVD
- Backup

Disk Encryption Tools: VeraCrypt and Symantec Drive Encryption

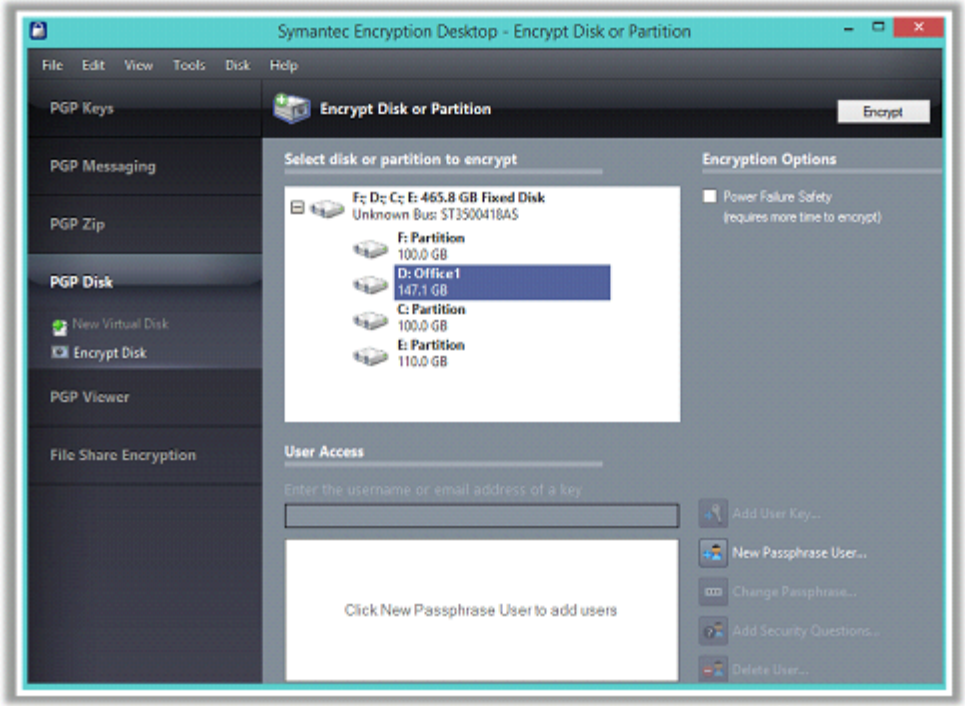
VeraCrypt



https://veracrypt.codeplex.com

Symantec Drive Encryption

- Symantec Drive Encryption provides full disk encryption for all data (user files, swap files, system files, etc.) on desktops, laptops, and removable media



https://www.symantec.com

Disk Encryption Tools

Gillsoft Full Disk Encryption
http://www.gillsoft.in

Full Disk Encryption Software
https://www.winmagic.com

PocketCrypt
http://www.securstar.com

Endpoint Full Disk Encryption
https://www.checkpoint.com

SafeGuard Encryption
https://www.sophos.com

DriveCrypt Plus Pack
http://www.securstar.com

Dell Data Protection | Encryption
http://www.dell.com

Alertsec
https://www.alertsec.com

Rohos Disk Encryption
http://www.rohos.com

AxCrypt
https://www.axcrypt.net

DriveCrypt
http://www.securstar.com

east-tec SafeBit
https://www.east-tec.com

Folder Lock
http://www.newsoftwares.net

ShareCrypt
http://www.securstar.com

Cryptainer LE
http://www.cypherix.com

Module Flow

1. Cryptography Concepts
2. Encryption Algorithms
3. Cryptography Tools
4. Public Key Infrastructure (PKI)
5. Email Encryption
6. Disk Encryption
7. Cryptanalysis
8. Countermeasures

Cryptanalysis Methods

Linear Cryptanalysis

- It is commonly used on **block ciphers**
- It is a known plaintext attack and uses a **linear approximation** to describe the behavior of the block cipher
- Given enough pairs of **plaintext** and **corresponding ciphertext**, bits of information about the key can be obtained
- For example, with the **56 bit DES key brute force** could take up to **256 attempts**

Differential Cryptanalysis

- Differential cryptanalysis is a form of cryptanalysis applicable to **symmetric key algorithms**
- It is the **examination of differences** in an input and how that affects the resultant difference in the output
- It originally worked only **with chosen plaintext**
- It can also work with **known plaintext** and **ciphertext only**

Integral Cryptanalysis

- This attack is useful against block ciphers based on **substitution-permutation networks** an extension of differential cryptanalysis
- Integral analysis, for block size b, holds **b-k bits constant** and runs the other k through all **2k possibilities**
- For **k=1**, this is just differential cryptanalysis, but with **k>1** it is a new technique

Code Breaking Methodologies

- One can measure the **strength of an encryption algorithm** using various code-breaking techniques

Brute-Force	Cryptography keys are discovered by **trying every possible combination**
Frequency Analysis	• It is the study of the frequency of letters or groups of letters in a **ciphertext** • It works on the fact that, in any given stretch of written language, certain letters and **combinations of letters** occur with varying frequencies
Trickery and Deceit	It involves the use of **social engineering techniques** to extract cryptography keys
One-Time Pad	A one-time pad contains many **non-repeating groups of letters** or number keys, which are chosen randomly

Cryptography Attacks

- Cryptography attacks are based on the assumption that the cryptanalyst has access to the **encrypted information**

Ciphertext-only Attack	Attacker has access to the cipher text; goal of this attack to **recover encryption key** from the ciphertext
Adaptive Chosen-plaintext Attack	Attacker makes a **series of interactive queries**, choosing subsequent plaintexts based on the information from the previous encryptions
Chosen-plaintext Attack	Attacker **defines his own plaintext**, feeds it into the cipher, and analyzes the resulting ciphertext
Related-key Attack	Attacker can obtain ciphertexts encrypted under **two different keys**, and this attack is useful if the attacker can obtain the plaintext and matching cipher text
Dictionary Attack	Attacker constructs a **dictionary of plaintext** along with its corresponding ciphertext that he/she has learnt for a certain period of time

Cryptography Attacks (Cont'd)

Known-plaintext Attack	Attacker has **knowledge of some part of the plain text**; using this information, the key used to generate ciphertext is deduced so as to decipher other messages
Chosen-ciphertext Attack	Attacker obtains the plaintexts corresponding to an **arbitrary set** of ciphertexts of his own choosing
Rubber Hose Attack	Extraction of cryptographic secrets (e.g. the password to an encrypted file) from a person by **coercion or torture**
Chosen-key Attack	Attacker usually breaks an **n bit** key cipher into **$2^{n/2}$** number of operations
Timing Attack	It is based on repeatedly measuring the **exact execution times** of modular exponentiation operations
Man-in-the-middle Attack	Attacker performs this attack on the **public key cryptosystems** where key exchange is required before communication takes place

Brute-Force Attack

Attack Scheme	Defeating a cryptographic scheme by **trying a large number of possible keys** until the correct encryption key is discovered
Brute-Force Attack	Brute-force attack is a **high resource and time intensive process**, however, more certain to achieve results
Success Factors	Success of brute-force attack depends on **length of the key**, **time constraint**, and **system security mechanisms**

Power/Cost	40 bits (5 char)	56 bit (7 char)	64 bit (8 char)	128 bit (16 char)
$ 2K (1 PC. Can be achieved by an individual)	1.4 min	73 days	50 years	10^20 years
$ 100K (this can be achieved by a company)	2 sec	35 hours	1 year	10^19 years
$ 1M (Achieved by a huge organization or a state)	0.2 sec	3.5 hours	37 days	10^18 years

Estimate Time for Successful Brute-force Attack

Birthday Attack

- A birthday attack is a name used to refer to a class of brute-force attacks against cryptographic hashes that makes the brute forcing easier
  - **Birthday paradox**: The probability that two or more people in a group of 23 share the same birthday is greater than half

Birthday Paradox

- How many people do you need to have a high likelihood that **2 share the same birth day** (i.e., same day and month, but not the same year)
- There are **365 days** in a year, so you might think at least half of that or 182 people, but it is actually only **23!**

- The basic idea is this: How **many people** would you need to have in a room to have a **strong likelihood** that two would have the **same birthday** (month and day, but not the same year)
- Obviously, if you put **367 people** in a room, at least 2 of them must have the same birthday, since there are only 365 days in a year, plus one more in a leap year
- The paradox is not asking how many people you need to **guarantee a match**, just how many you need to have a strong probability
- Even with 23 people in the room, you have a **50 percent chance** that 2 will have the same birthday

Birthday Paradox: Probability

01 The probability that the first person does not share a birthday with any previous person is **100 percent**, because there are no previous people in the set. This can be written as **365/365**

02 The second person has only one preceding person, and the **odds that the second person** has a **birthday** different from the first are **364/365**

03 The third person might share a birthday with two preceding people, so the odds of having a birthday from either of the two preceding people are **363/365**

04 Because each of these are independent, we can compute the probability as follows:

365/365 * 364/365 * 363/365 * 362/365 ... * 342/365

(342 is the probability of the 23rd person shares a birthday with a preceding person)

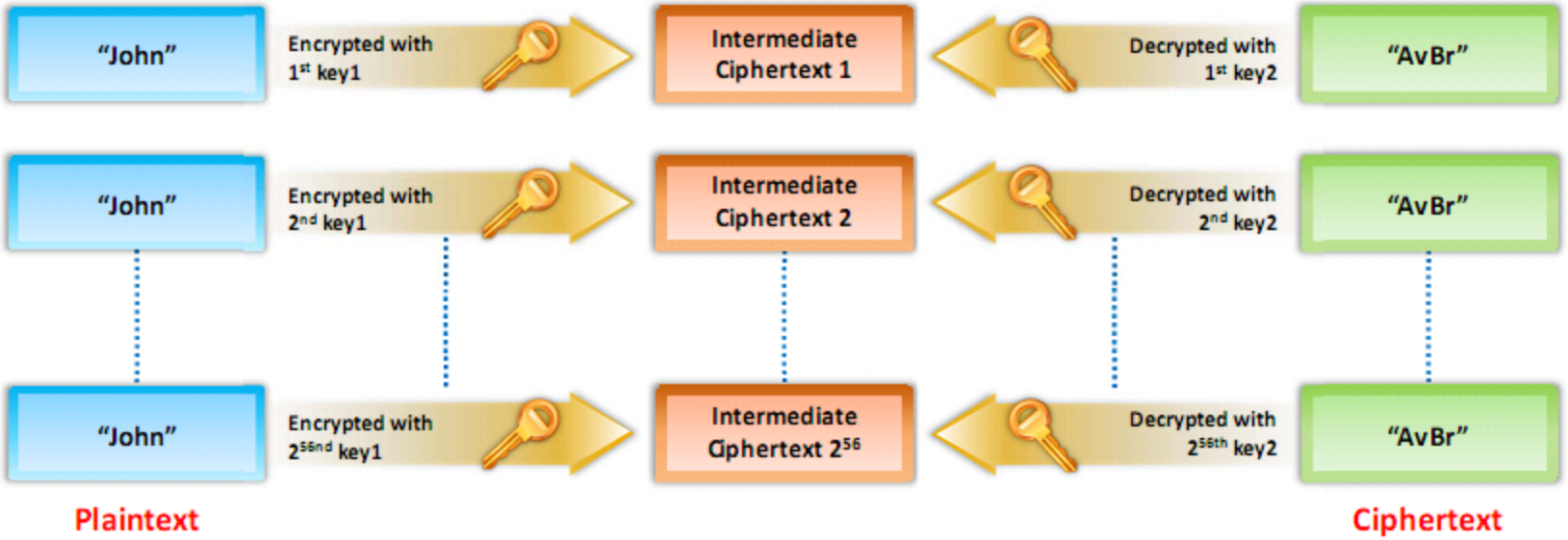
05 When we convert these to decimal values, it yields (truncating at the third decimal point):

1 * 0.997 * 0.994 * 0.991 * 0.989 * 0.986 * ... 0.936 = 0.49, or **49 percent**

06 This **49 percent is the probability** that **23 people** will not have any **birthdays** in common; thus there is a **51 percent** (better than even odds) chance that **2** of the **23** will have a **birthday** in common

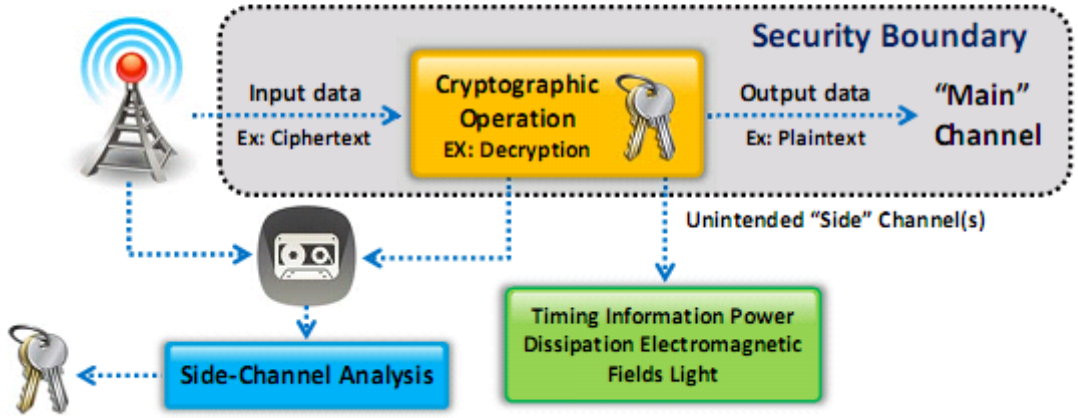
Meet-in-the-Middle Attack on Digital Signature Schemes

- The attack works by **encrypting from one end** and **decrypting from the other end**, thus meeting in the middle
- It can be used for **forging signatures** even on digital signatures that use multiple-encryption schemes



Side Channel Attack

- Side channel attack is a **physical attack** performed on a cryptographic device/cryptosystem to gain sensitive information
- Cryptography is generally part of the hardware or software that runs on physical devices such as semi-conductors (includes resistor, transistor, and so on)
- These physical devices are affected by various **environmental factors** that include power consumption, electro-magnetic field, light emission, timing and delay, and sound
- In side channel attack, an attacker **monitors these channels (environmental factors)** and tries to acquire the information useful for cryptanalysis



- Assume that an encrypted data is to be decrypted and displayed as a plain text, inside a **trusted zone**
- At the time of decryption in a cryptosystem, **physical environmental factors** such as timing, power dissipation, etc., acting on the components of a computer are recorded by an attacker
- The attacker analyzes this information in an attempt **to gain useful information** for cryptanalysis

Hash Collision Attack

- Hash collision attack is performed by finding **two different input messages** that result into same hash output

- This allows an attacker to perform **cryptanalysis** by exploiting the **digital signature** to decode the encoded data

- SHA-1 algorithm converts input message into **constant length of unstructured strings** of numbers and alphabets, which act as a finger print for the sent file

- Attacker is able to forge victim's **digital signature** of message a1 on the incorrect message a2

- Once the attacker is able to detect any collisions in the hash, then he/she tries to identify more collisions by **concatenating data** to the matching messages

DUHK Attack

1. DUHK (Don't Use Hard-Coded Keys) is a **cryptographic vulnerability** that allows an attacker to **obtain encryption keys** used to secure VPNs and web sessions
2. This attack mainly affects any hardware/software using ANSI X9.31 **Random Number Generator** (RNG)
3. The **pseudorandom number generators** (PRNGs) generate random sequences of bits based on the initial secret value called a seed and the current state
4. Both the factors are the key issues of DUHK attack as any attacker could combine ANSI X9.31 with the hard-coded seed key **to decrypt the encrypted data** sent or received by that device
5. Using this attack, attackers identify encryption keys and **steal confidential information** such as critical business data, user credentials, credit card details, etc.

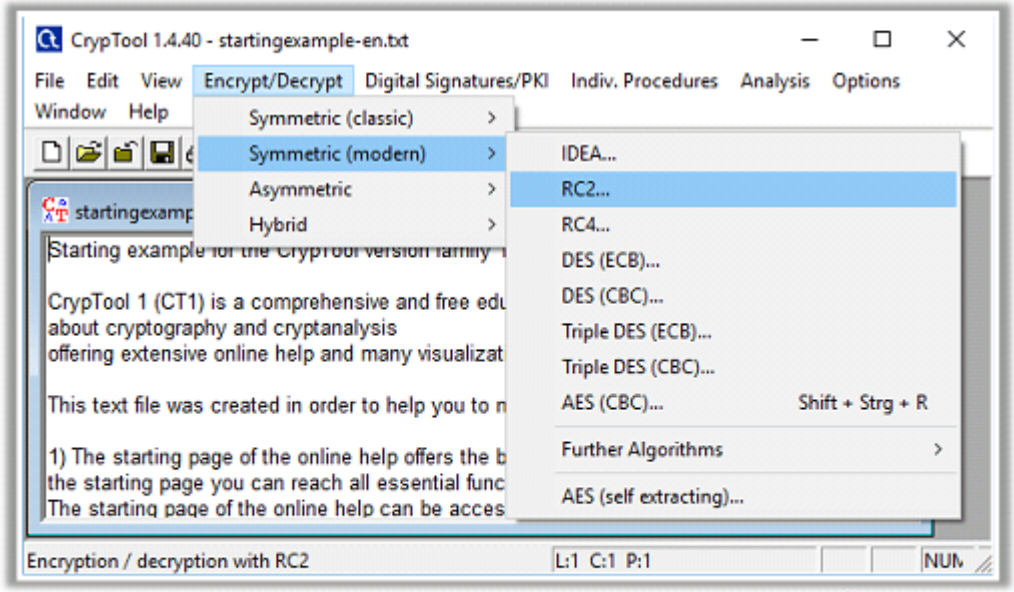
Rainbow Table Attack

- A rainbow table attack is a type of cryptography attack where an **attacker uses a rainbow table for reversing cryptographic hash functions**
- A rainbow table is a **precomputed table** which **contains word lists** like dictionary files and brute force lists and their hash values
- It uses the **cryptanalytic time-memory trade-off technique** to crack the cryptography, which requires less time than some other techniques
- An attacker computes the hash for a list of possible passwords and compares it to the precomputed hash table (rainbow table). If attackers find a match, **they can crack the password**

Cryptanalysis Tools

CrypTool

- CrypTool is a free e-learning program in the area of cryptography and cryptanalysis
- It consists of e-learning software (CT1, CT2, JCT, and CTO)



https://www.cryptool.org

- **CryptoBench** — *http://www.addario.org*
- **Cryptol** — *https://cryptol.net*
- **Ganzúa** — *http://ganzua.sourceforge.net*
- **EverCrack** — *http://evercrack.sourceforge.net*
- **AlphaPeeler** — *http://alphapeeler.sourceforge.net*

Online MD5 Decryption Tools

MD5 Decoder
https://www.dcode.fr

MD5 Decrypt
http://www.md5decrypt.org

MD5 Decrypter
http://www.md5online.org

MD5Decrypter
https://www.md5decrypter.com

OnlineHashCrack.com
https://www.onlinehashcrack.com

HashKiller.co.uk
https://hashkiller.co.uk

Md5.My-Addr.com
http://md5.my-addr.com

cmd5.org
http://www.cmd5.org

CrackStation
https://crackstation.net

md5this
http://www.md5this.com

MD5/Sha1 hash cracker
https://crackstation.net

Md5() Encrypt & Decrypt
http://md5decrypt.net

MD5Decryption
http://md5decryption.com

md5hashing
https://md5hashing.net

Online Reverse Hash Lookup
http://reverse-hash-lookup.online-domain-tools.com

Module Flow

How to Defend Against Cryptographic Attacks

1. Access of cryptographic keys should be given to the application or to the user directly
2. Intrusion detection system should be deployed to monitor exchanging and access of keys
3. Passphrases and passwords must be used to encrypt the key, if stored in disk
4. Keys should not be present inside the source code or binaries
5. For certificate signing, transfer of private keys should not be allowed
6. For symmetric algorithms, key size of 168 bits or 256 bits should be preferred for a secure system, especially in case of large transactions
7. Message authentication must be implemented for encryption of symmetric-key protocols
8. For asymmetric algorithms, key size of 1536 bits and 2048 bits should be considered for secure and highly protected application
9. In case of hash algorithm, key size of 168 or 256 bit should be considered
10. Only recommended tools or products should be preferred rather than creating self-engineered crypto algorithms or functions
11. Put a limit on number of operations per key
12. The output of the hash function should have larger bit length, which makes it hard to decrypt

Module Summary

- Cryptography is the conversion of data into a scrambled code that is decrypted and sent across a private or public network
- Symmetric encryption uses the same key for encryption as it does for decryption and asymmetric encryption uses different encryption keys for encryption and decryption
- Ciphers are algorithms used to encrypt or decrypt the data
- PKI is a set of hardware, software, people, policies, and procedures required to create, manage, distribute, use, store, and revoke digital certificates
- Digital signature uses asymmetric cryptography to simulate the security properties of a signature in digital, rather than in a written form
- Disk encryption protects confidentiality of the data stored on the disk by converting it into an unreadable code using disk encryption software or hardware
- Cryptography attacks are based on the assumption that the cryptanalyst has access to the encrypted information