Information Security Requirements (from Michael T. Goodrich, Roberto Tamassia, Introduction to Computer Security, Pearson, 2011, p. 2-18)

## Information security is defined conceptually as C-I-A (confidentiality – integrity – availability)

Confidentiality

An avoidance of the unauthorized disclosure of information

Confidentiality provides access for those who are allowed to see the data while disallowing others from learning anything about its content or even existence of such data

Tools for protecting sensitive information:

* Encryption using a secret, called an encryption key
* Access control: rules and policies that limit access to confidential information to authorized people
* Authentication: the determination of the identity or role that someone has (token-based, password-based, biometrical)
* Authorization: the determination if a person or system is allowed access to resources based on an access control policy
* Physical security: the establishment of physical barriers to limit access to protected computational resources (e.g., rooms with walls incorporating copper meshes, Faraday cages, so that electromagnetic signals cannot enter/exit the enclosure)

Integrity

Accounts: A=1000$, B=2000$; total=1000+2000=3000$

Transaction: 100$ to be transferred from A to B:

A=900$, B=2100$; total=3000$ commitment

1. A=A-100=900 fail
2. B=B+100=2100
3. B=B+100=2100 fail
4. A=A-100=900 new money

A=900$, B=2000$ total=2900$; loss of 100$ roll-back

A property that information has not be altered in an unauthorized way

Tools for data protection are also used for data integrity. Additional tools:

* Backups: the periodic archiving of data
* Checksums: the computation of a function that maps the contents of a file to a numerical value; this value is kept together with the file. To check the file integrity, its checksum is recalculated and compared versus the stored one
* MSB B7b6b5b4b3b2b1b0 LSB checksum=b0 xor b1 …xor b7
* 10010001 sum=1
* 10010000 sum=0
* Data correcting codes: methods for storing data in a way that small changes can be easily detected and automatically corrected
* DNA spiral
* mirroring

All these tools use redundancy

Not data only, but metadata shall be also protected

Availability

The property that information is accessible in a timely fashion by those authorized to do so

For example, stock quotes are most useful when they are fresh. Another example, stolen credit card number shall be timely discarded

Tools:

* Physical protection: infrastructure meant to keep information available even in the event of physical challenges (buildings to withstand storms, earthquakes, bomb blasts, and outfitted with generators and other electronic equipment to cope with power outages and surges)
* Computational redundancies: computers and storage devices that are used as fallbacks in the case of failures (RAID, redundant array of inexpensive disks; farms of web servers)

An attacker who does not care about the confidentiality or integrity, may choose an attack on availability (DoS attack, denial of service) Distributed DoS

P=Malloc(100);p<>null; 0

If(p!=NULL) normal

Else message;

Buffer overflow

float buf[80]; 4-byte

Buf[0], ..buf[79];

Buf char\*

\*(Buf+400)=100; 5th

Run-time checking out of boundary

.c compliler => .obj linker => .exe

## Assurance, Authenticity, and Anonymity

In addition to CIA concepts, there are also AAA concepts (assurance, authenticity, anonymity)

Assurance: how trust is provided and managed in computer systems. Trust: degree to which we have confidence that people or systems are behaving in the way we expect

Trust involves the interplay inter-play of the following

* Policies specifying behavioral expectations that people or systems have for themselves and others. For example, the designers of an online music system may specify policies how users can access and copy songs
* Permissions describing the behaviors that are allowed by the agents that interact with a person or a system. For instance, an online music store may provide permissions for limited access and copying for people who have purchased certain songs
* Protections describing mechanisms put in place to enforce permissions and policies. Using our running example of an online music store, we could imagine that such a system would build in protections to prevent people from unauthorized access and copying of its sources (trust directed from system to users)

Assurance involves the management of trust in two directions – from users to systems (e.g., valid use of credit card numbers for songs purchasing), and from system to users

Designers of computer systems want the people using the resources of their systems doing so in line with their policies. Also designers may want restricting time of use of their copy, or number of watches, or number of back-up copies of their music.

Thus, trust management deals with the design of effective, enforceable policies, methods for granting permissions to trusted users, and the components that can enforce those policies and permissions for protecting and managing the resources in the system.

Another important part of system assurance involves software engineering. The designers of a system need to know that the software that implements their system is coded so that it conforms to their design. For example, the designer of a system might specify that a Pseudo Random Number Generator (PRNG) is used for encryption with random numbers. If implementer, always uses the same seed value for PRNG then the numbers generated will be again repeated.

Thus, it is necessary having good specification and complying it implementation.

Also, for trust, it is necessary supporting some evidence of the operations conducted to be able using them in the case of going to court (say, in the case being cheated by a web site).

Authenticity

It is the ability to determine the statements, policies, and permissions issued by persons or systems are genuine. If such things can be faked, there is no way to enforce the implied contracts that people and systems engage in when buying or selling items online.

Nonrepudiation is the property that authentic statements issued by some person or system cannot be denied. Nonrepudiation is achieved by the use of digital signature.

Anonymity

There is an unfortunate side effect from using personal identities in electronic transactions (medical history, purchase history, legal records, e-mail communications, employment records). Therefore, anonymity is needed, which is the property that certain records or transactions not to be attributable to any individual.

If organizations need publishing data about their members or clients, we should expect that they do so in a privacy preserving fashion, using some of the following tools.

* Aggregation (sum, average, min, max, count)
* Mixing: the intertwining of transactions, information, or communications in a way that cannot be traced to any individual
* Proxies: trusted agents that are willing to engage in actions for an individual in a way that cannot be traced back to that person. Internet searching proxies are web sites that themselves provide an Internet browser interface, so that individuals can visit web sites that they might be blocked from, for instance, because of the country they are located in.
* Pseudonyms: fictional identities that can fill in for real identities in communications and transactions, but are otherwise known to a trusted party only. For example, many social networks allow users interacting user pseudonyms without revealing their actual identity.

## Threats and Attacks

**Threat** is a potential for violation of security, which exists when there is a circumstance, capability, action, or event that could breach security and cause harm. That is, a threat is a possible danger that might exploit a vulnerability.

**Attack** is an assault on system security that derives from an intelligent threat: that is, an intelligent act that is a deliberate attempt to evade security services and violate the security policy of a system

Most common attack types

* Eavesdropping (on confidentiality)
* Alteration (on integrity)
* Denial of Service (on availability)
* Masquerading: the fabrication of information that is purported to be from someone who is not actually an author. For example, phishing, which creates a web site that looks like a real bank or other e-commerce site. It is intended for gathering passwords and spoofing (sending network data packets with false return addresses). It is attack on authenticity, confidentiality (stealing passwords), and/or anonymity (use of passwords to get actual individual’s data).
* Repudiation: the denial of a commitment or data receipt. This involves an attempt to back out of a contract or a protocol that requires the different parties to provide receipts acknowledging that data has been received. This is an attack on assurance.
* Correlation and traceback: the integration of multiple data sources and information flows to determine the source of a particular data stream or piece of information. This is an attack on anonymity.

Sum(salaries)=1000

A,B,C: A+B+C=1000; A? B? C?

D; A+B+C+D=1200; (A+B+C+D)-(A+B+C)=1200-1000=200

## Cryptographic principles

A cryptographic system consists of seven components

1. The set of possible plain-texts
2. The set of possible cipher-texts
3. The set of encryption keys
4. The set of decryption keys
5. The correspondence between encryption keys and decryption keys

For RSA: Rivest Shamir Adleman: 1977: e\*d=1 mod fi(N) reciprocals, multiplicative inverses

Fi(N)=(p-1)\*(q-1) Euler totient function

N=p\*q; p, q are prime numbers

Primes=> 2,3,5,7,11,13,..

13=1\*13; 54=1\*2\*3^3

Composite numbers: 6=1\*2\*3; 8=1\*2\*2\*2=1\*2^3

Factored

P=3; q=5; N=15; ; modulus

E=3; public key

D=?3 private key, secret; d=e^(-1) mod fi(N)=3^(-1) mod 8 =3

Encryption by E Allows data hiding: only holder of secret d can decrypt ciphertext

Encryption by d, then any person can decrypt the ciphertext correctly=> digital signature

Brute force attack

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| x | 0 | 1 | 2 | 3=d | 4 | 5 | 6 | 7 |
| 3(=E)\*x mod 8 | 0 | 3 | 6 | 9 mod 8 =1 |  |  |  |  |

0<=R=A mod B<B

R=A – q\*B

Q=floor(A/B)

Floor(x): maximal integer not exceeding x

A=15; B=4; q=floor(15/4)=flloor(3.75)=3

Floor(-3.75)=-4

Ceiling(x)

15 mod 4 = 15-floor(15/4)\*4=15-3\*4=15-12=3

0<=A<B=> A mod B =A

A mod B =A-floor(A/B)\*B=A-0\*B=A

B=4; A=-3

-3 mod 4 = -3-floor(-3/4)\*4=-3-(-1)\*4=-3+4=4-3=1

Floor(-0.75)=??-1

If A<0, |A|<B => A mod B = B-|A|= B+A

-5 mod 21 =?? 16= 21-5

-7 mod 27=??20

7 mod 27=7

%

Alice Bob knows pubic key of Alice

Bob: 1) generates symmetric key, SK, for DES (data encryption standard) or AES (advanced encryption standard)

C=RSAEncr(SK, Alice public key)

SK =RSADec(C, Alice private key)

1. The encryption algorithm to use

RSA-512 =>155 decimal digits; RSA-2048 =>617 decimal digits

Max bit 64, 128;

C=M^e mod N

N=15; e=3; M is from Z(N)={0,.,N-1}; Z any integer; ZN

M=3;

C=3^3 mod 15=27 mod 15 =??12

1. The decryption algorithm to use

M’=C^d mod N = 12^3 mod 15

C1=C^1 mod N = 12 mod 15 = 12

C2=c1^2 mod N = 144 mod 15 = (9\*15+9) mod 15 =0+9=9

((A mod n)\*(b mod n)) mod n= (A\*B) mod n

M’=C1\*C2 mod 15 = 12\*9 mod 15 =108 mod 15 = (7\*15+3) mod 15 = 0+3 =3=M

For example, Caesar cipher

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Numerical code | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
| Plaintext | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z |
| Ciphertext | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c |

C=(p+3)mod26; C=(p+k)mod26; k is from 0,..,25; key space is 1..25 its cardinality 25

P=(c-3)mod26; P=(c-k)mod26;

Symmetric cryptosystems or shared-key cryptosystems

C=EK(P); P=DK(C)

Asymmetric cryptosystems or public-key cryptosystems

C=EK1(P); P=DK1(C)

K1 and K2 constitute a pair of keys, one of them is public, and the other one is private (secret)

For asymmetric cryptography, the use of a secret key for encryption serves as a digital signature of the key owner. Say, Alice has SA (secret of Alice), PA (public of Alice), and Bob has SB (secret of Bob), PB (public of Bob). Then the signature of Bob for message M is C=ESB(M), that is checked by Alice by decryption of C by PB: M=DPB(С).

Public-key cryptography drawback is its high computational complexity, e.g., in RSA, it is necessary exponentiation of numbers with hundreds of decimal digits (300-500 digits). It is mainly used for key exchange. 512 bits used=> decimal digits 155; 2048 bits => 617 decimal digits

Cryptosystems may be attacked by brute force attack, assuming enumeration of all possible keys. With AES key of 128 bit size, number of possible keys is 2128, that is infeasible to enumerate. Cryptosystems with large key sizes are considered as computationally (conditionally) secure.

SSL protocol

Client

Bank (private, public keys)

Bank=>Client: PublickeyB

Client: generates session key, K

Client->Bank: C=RSAEnc(K, PublickeyB)

Bank: K’=RSADec(C,PrivayeKeyB)’ K’=K assume

## Hash functions

Digital signatures are usually made on an entire message but on its hash value. Hash function, h, is similar to encryption transformation, also may have secret key, but contrary to an encryption transformation which is one-to-one (invertible), it is many-to-one (one-way, not invertible). Hash function returns result of a fixed size (e.g., 64 bits) for any input message.

F:XxY->Z; Z=F(X,Y)

Hash functions are many-to-one mappings

Record=(stId,stdname,stdsurn); stid is from D1, stdname from D2, stdsurn from D3

Y=x^2; x=3=>y=9=>x1=3, x2=-3

Hash: message of any size-> fixed size; MD5 output is 128 bits

Record is from D1xD2xD3

(M, H(M))=>(M’,h); H(M’)==h? true => original, otherwise => corrupted

Message authentication code (MAC) of a message, M, encrypted with a key, K,

MAC(M,K)=h(K||M); K=’secret’, M=’message”=> K||M=” secretmessage”

(M,MAC)=>(M’,mac); (MAC(M’,K)=h(K||M’))==mac?yes => trust; otherwise corrupted

MAC is transferred together with the ciphertext

C=EK(M)

After decryption,

M’=DK(C),

a receiver check authenticity of the message received by re-computation of MAC,

MAC(K,M’)=h(K||M’)

And comparing it versus MAC(K,M). If they match, the receiver decides the message is authentic.

Digital signatures:

(M, RSAEnc(H(M), PrivateKeyA))=>(M’,ds)

Verification: H(M’)==RSADec(ds, PublicKeyA)?yes=> true; Alice; otherwise, reject

## Digital certificates

For public-key cryptography, it is necessary having trusted public key of the correspondent. Digital certificates (DC) issued by trusted third parties (certification authority, CA) are used for that purpose. DC usually contains the following data:

- Name of the CA (e.g., Thawte)

- date of the certificate issue (e.g., 29.09.2016)

- expiration date (e.g., 28.09.2018)

- address of the web site (e.g., mail.google.com)

- name of the organization operating the web site (e.g., Google, Inc.)

- public key used of the web server (e.g., an RSA 1024-bit key)

- name of the cryptographic hash function used (e.g., SHA-256)

- digital signature

One way of defending against a phishing attack is to check that the digital certificate contains the name of the organization associated with the website.