COMP90015 Distributed Systems

Security

School of Computing and Informati

2022 Semester II

Policies and Mechanisms

- The challenges of security arise as a result of the need to share or to distribute resourc an be physic
- A *secu* shared r ared resource.
- A security policy is enforced using a *security mechanism*.
- Digital cryptography provides the basis for most comp mechanisms, though computer security and cryptog bjects.

Threats and attacks

- Some threats are obvious – e.g., reading traffic on a shared network to gain information like a password or other personal information.
- Some th ver.
- Some th something with the release.
- Security threats fall into three broad classes:
  - *Leakage* – the acquisition of information by unauthorized r
  - *Tampering* – the unauthorized alteration of information
  - *Vandalism* – interference with the proper operation of a syste perpetrator.

Attacks on distributed systems depend on access to an existing communication channel. A communication channel can be misused in different ways:

- *Eavesdropping* – obtaining copies of messages without authority.
- *Masquerading* – sending or receiving messages using the identity of another princip
- *Messa* passin *man-in* ; e.g. the e
- *Replaying* – storing intercepted messages and sending them at a later date.
- *Denial of service* – flooding a channel or other resour to deny access for others.

Some attacks can be arguable, e.g. to what extent is spam considered a denial of service attack?

Threats from mobile code

- Some distributed systems allow code, called mobile code, to be communicated to a remote host, to be executed by that host. In this case it is necessary to ensure that the host, including all processes and resources available at the host, is secure f hile of course still allo
- Similar t ia the web browse code should b
- The Java VM has undergone revisions to ensure that mob ot pose a security risk.
- Construction of environments for running mobile cod ach is generally more difficult than providing secure channel to validate that the mobile code is not harmful. Trusted Network Computing works along these lines.

Information leakage

- Information leakage can be particularly difficult to prevent. E.g. a flood of messages to a dealer in a particular market can be a meaningful and useful piece of in cure.
- When th hen there is the pot
- E.g. ther t can communicate anonymously with a server. However if the client always makes requests on a Thursday afternoon, then the behavior ma nd the clients' identity may be inferred.
- Basically, the system must appear to be random in order fo ation to be leaked.

Securing electronic transactions

- There are a number of uses of the Internet that require secure transactions:
  - Email – personal information is often transmitted via email, including e.g. credit card details, and in some cases emails are used to authenticate a user, e.g. when a user is signing up to a mailing list.
  - Purchase of goods and services – payments for goods and services commonly happen via a web interface. Digital products are delivered via the Internet.
  - Bank and differ
  - Micr ing of a book, usag , etc, requi e very lo ctions of cent.
- Some example security policies for securing web purch
  - Authenticate the vendor to the buyer, so that the buyer is confide er is operated by the vendor.
  - Ensure that credit card and personal details are transmitted u ely, from the buyer to the vendor and that the details are kept private at all tim
  - Responses from the vendor, including digital goods and services, should be received by the buyer without alteration or disclosure during transmission. In this case, authenticating the buyer is not usually required since the vendor is happy so long as the money is made available.
  - It should be possible for a buyer to complete a secure transaction with a vendor even if there has been no previous contact between buyer and vendor and without the involvement of a third party.

Designing secure systems

- Building a completely secure system is akin to building a completely bug-free system.
- Known t buted system
- Logs of se i security violati her attempts to use supervisor resources have failed, due to incorrect password.
- Costs of implementing a policy mechanism must be bala threat. Costs of attack can be traced, i.e. how much does it co tacker in terms of time and resources
- Security should not needlessly inhibit legitimate uses.

Worst-case assumptions and design guidelines

- Interfaces are exposed – e.g. a socket interface is open to the public, in much the same way as the front door of a house.
- Networks are insecure – messages can be looked at, copied and falsified.
- Limit the lifetime and scope of each secret – keys and passwords can be broken
- Algorit er and more widely d per source code is sc code and this helps to find potential security problems before they are taken advantage of.
- Attackers may have access to large resources – available er needs to be predicted into the life time of the system and syste to be secure against some orders of magnitude beyond this.
- Minimize the trusted base – parts of the system that are responsible for enforcing security are trusted, the greater the number of trusted parts the greater the complexity and so the greater risk of errors and misuse.

Cryptography

Familiar names for the protagonists in security protocols:

- *Alice* –
- *Bob* – S
- *Carol*
- *Dave*
- *Eve* – Eavesdropper.
- *Mallory* – Malicious attacker.
- *Sara* – A server.

Encryption keys

There are two main classes of encryption algorithms: *shared secret key* algorithms and *public/private key* algorithms.
Some com

- k_A – Ali
- k_B – Bo
- k_{AB} – S
- k_A^{priv} – Alice's private key (known only to Alice).
- k_A^{pub} – Alice's public key (published by Alice for everyon

- $\{M\}_k$ – Message M encrypted with key k.
- $[M]_k$ – Message M signed with key k.

Basic properties

- Given an encryption algorithm, E, a decryption algorithm, D, a key, k, and a message, M, $\{M\}_k = E(M, k)$ and $M = D(\{M\}_k, k)$.
- If $k = k_A$ is Alice's secret key then $\{M\}_k$ can only be decrypted by Alice using k.
- If $k =$ $\}_k$ can only be decry
- If $k =$ $\{M\}_k$ can be decrypted by anyone who has k_A .
- If $k = k_A^{pub}$ is Alice's public key, from a public/privat $M\}_k$ can only be decrypted by Alice using k_A^{priv}
- Private/secret keys should be securely maintained sin compromised if an attacker obtains a copy of them.
- Public/private key encryption algorithms typically require *100 to 1000 times more processing power* than secret-key algorithms.

Secrecy and integrity

A fundamental policy is one of ensuring secrecy of a message. If Alice and Bob have agreed to a shared key and encryption/decryption algorithm then for a sequence of messages $M_1, M_2, \ldots$:

1. Alice us o Bob.
2. Bob use

If the messa ontains some agreed upon value such as a checksum, then Bob can be confident that the message is from Alice and that it has not been tamp ith.

Some problems:

- How can Alice communicate a shared key k
- How does Bob know that a received message is a not a copy of some previous meessage $\{M_i\}_{k_{AB}}$ captured by an attacker Mallory and resent to Bob?

Authentication using a trusted third party

Consider the case when Alice wants to access a resource held by Bob. Sara is an authentication server that is securely managed. Sara issues passwords to all users including Alice and Bob. Sara knows k_A and k_B because they are derived from the passwords.

1. Alice sends an (unencrypted) message to Sara stating her identity and reques
2. Sara sen t encryp g with Bob: $\{$
3. Alice decrypts the response using k_A. Alice can before sending it, because it is encrypted with
4. Alice sends a request R to Bob: $\{Ticket\}_{k_B}$
5. Bob receives the encrypted ticket and decrypts it using hi e the ticket is actually $Ticket = \{k_{AB}, Alice\}$. Alice and Bob can now communicate using the shared key or *session key*, k_{AB}.

The previous algorithm is a simplified version of the authentication protocol originally developed by Roger Needham and Michael Schroeder.

Kerberos overview

Challenge Response

- The use of an authentication server is practical in situations where all users are part of a single organization. It is not practical when access is required between parties t
- Simple s the clear" Such pa
- The *ch*

passwords in the clear. The identity of a client is establishe g the client an encrypted message that only the client should be a pt, this is called a challenge message. If the client cannot decrypt t e message then the client cannot properly respond.

Authenticated communication with public keys

1. Alice accesses a *key distribution service*, Sara, to obtain a *public-key certificate* giving Bob's public key. The public-key certificate, *Cert*, is a message signed by Sara using k_S^{priv}. The key k_S^{pub} is widely known by Alice and others and is used to check the signature. Among other things the the certificate contains Bob, k_B^{pub}
2. Alice creates a message. She sends the message and her public key pair ($\{keyname, \{k_{AB}\}_{k_B^{pub}}\}$.
3. Bob selects the appropriate private key k_B^{priv} obtain k_{AB}. Alice and Bob can now securely communic

If the message from Alice to Bob was tampered with then the decrypted k_{AB} will not match and messages back from Bob will not make sense. Having said this, Alice can also encrypt some additional identification in the original message, e.g. a checksum or Alice's email address, etc.

Digital signature

- A *digital signature* serves the same role as a signature, binding an identity to a message.
- In the case of public/private keys, the identity is the public/private key pair itself.
- A digital s
  Digest
  digest fu
  to from t
  this.
- If Alice wants to sign a message, M, then Alice
- A receiver Bob decrypts the digest using k
  of M locally. If the message or the encrypted digest wer
  the results will not match.
- This is effectively a signature based on the identity k_A^{priv} since no other private key would produce that encrypted digest and no other message is likely to produce that digest. Alice cannot deny that she signed the message.

Digital Signature with pub/priv keys

Digital Signature with shared key

Certificate chains

- For Alice to authenticate a certificate from Sara concerning Bob's public key, Alice must first have Sara's public key. This poses a recursive problem.
- In the simplest case, Sara creates a *self-signed* certificate, which is attesting to her own p e.g. by distrib ertificate is *trusted* e integri
- Howev ob's public key and that Sara has signed a certificate attesting to Caro . This is an example of a *certificate chain*. If Alice trust can authenticate Bob's identity. Otherwise Alice mus e Carol's identity using Sara's certificate.
- *Revoking* a certificate is usually by using predefined expiry dates. Otherwise anyone who may make use of the certificate must be told that the certificate is to be revoked.

http://publib.boulder.ibm.com

A standard for digital certificates is X.509. From Wikipedia, the structure of an X.509 version 3 certificate is:

* Certificate
    * Version
    * Serial Number
    * Algorithm ID
    * Iss
    * Val
    * Subject
    * Subject Public Key Info
        * Public Key Algorithm
        * Subject Public Key
    * Issuer Unique Identifier (Optional)
    * Subject Unique Identifier (Optional)
    * Extensions (Optional)
* Certificate Signature Algorithm
* Certificate Signature

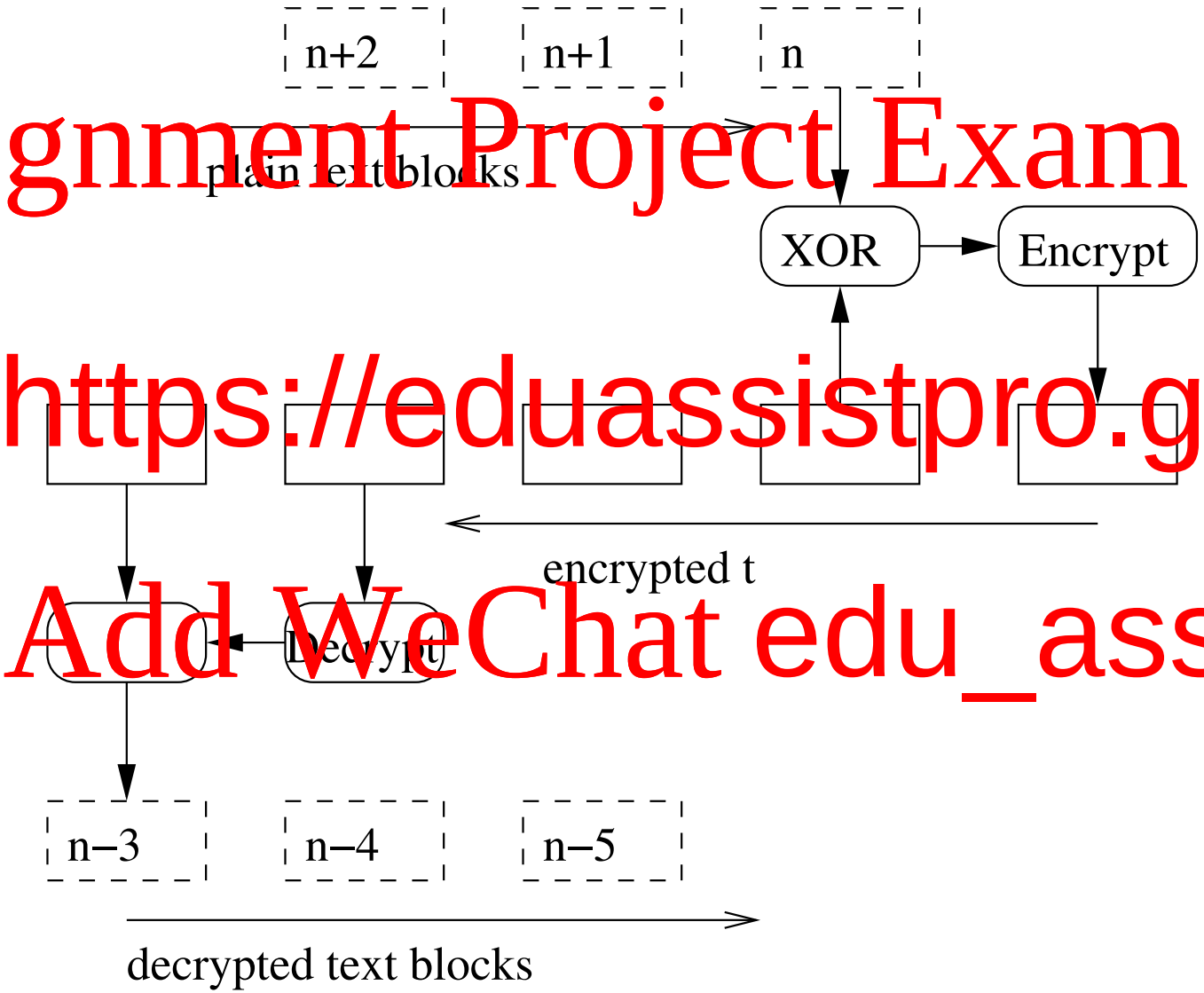
```
Certificate:
  Data:
   Version: 3 (0x2)
   Serial Number: 1 (0x1)
   Signature Algorithm: md5WithRSAEncryption
   Issuer: C=ZA, ST=Western Cape, L=Cape Town, O=Thawte Consulting cc,
           OU=Certification Services Division,
           CN=Thawte Server CA/Email=server-certs@thawte.com
   Validity
       Not Before: Aug  1 00:00:00 1996 GMT
       Not After : Dec 31 23:59:59 2020 GMT
   Subje

   Subje
       P
       R
           Modulus (1024 bit):
               00:d3:a4:50:6e:c8:ff:56:6b:e6:cf:5d:b6:ea:
               68:75:47:a2:aa:c2:da:84:25:fc:a8:f4:47:51:
               85:b5:20:74:94:86:1e:0f:75:c9:e9:08:61:f5:
               6d:30:6e:15:19:02:e9:52:c0:62:db:4d:99:9e:
               6a:dc:44:38:cd:fe:be:e3:64:09:70:c5:fe:b1:
               29:b6:2f:49:c8:3b:d4:27:04:25:10:97:2f:e7:
               6d:c0:28:42:99:d7:4c:43:de:c3:f5:21:6d:54:9f:
               5d:c3:58:e1:c0:e4:d9:5b:b0:b8:dc:b4:7b:df:36:
               3a:c2:b5:66:22:12:d6:87:0d
           Exponent: 65537 (0x10001)
   X509v3 extensions:
       X509v3 Basic Constraints: critical
           CA:TRUE
Signature Algorithm: md5WithRSAEncryption
```

Cryptographic algorithms

- Secret key cryptography is often referred to as *symmetric* cryptography. Public/private key cryptography is often referred to as *asymmetric*.
- *Block ciphers* operate on fixed-size blocks of data; 64 bits is a popular size for the bloc s padded to the stan k is transm
- *Cipher b* encrypted to identical encrypted blocks. However, if the same message is send to do different recipients then it will still look the same and thi information leakage weakness. To guard against this a b *initialization vector* is used to start each message in a di
- *Stream ciphers* are used when the data cannot be easil this case, an agreed upon *key stream* (such as from a random number generator with known seed) is encrypted and the output is XOR'ed with the data stream.

Cipher Block chaining



Streaming Cipher in IEEE 802.11 WEP

WEP (Wired Equivalent Privacy) security is seriously flawed and everyone should now be using at least WPA2 (WiFi Protected Access version 2).

Some symmetric algorithms include:

- *TEA*: Tiny Encryption Algorithm, and subsequently the Extended (XTEA) version that guards against some minor weaknesses. The algorithm uses 128 bit keys to encrypt 64 bit blocks. The algorithm consists of only a few lines of code (hence the name), is secure and reasonably fast.
- *DES*: Data Encryption Standard uses a 56 bit key to encrypt a 64 bit block. This alg he triple DES ver
- *IDEA*: I o encrypt 64 bit blo
- *RC4*: A stream cipher that uses keys of any length up to 256 by t 10 times as fast as DES and was widely used in WiFi networks un kness was expose
- *AES*: Advanced Encryption Standard has variable blo ngth with specifications for keys with a length of 128, 192 or 256 bi ypt blocks with length of 128, 192 or 256 bits. Block and key lengths can be extended by multiples of 32 bits.

- The most widely known asymmetric algorithm is *RSA* or the Rivest, Shamir and Adelman algorithm.
- RSA is based on the use of the product of two very large prime numbers (greater than 10e100). Its strength comes from the fact that the determination of the prime factors of such large numbers is very computationally expensive.
- There are no known flaws in RSA.
- One pot
  ker, having
  the publ
  ible messages
  and find
  k can be
  defend
  s many bits
  as the key
  s.
  - One analysis done in 2014 (`https://www.di-mg`
    orts
    that "to crack some ciphertext encrypted with a 64-bit key by the
    ethod of
    trying every combination of keys possible means you have 2
    ts or
    1.8×10^{19}. If you have a computer that can carry out one encrypt
    ur
    millisecond, it will take about 292 million years to find the correct
    computer by a million times and it will still take about 3 centuries to solve. In practice, a set of supercomputers operating in parallel can crack a 64-bit key in a relatively short time. If an attacker has access to a large selection of messages all encrypted with the same key, there are other techniques that can be used to reduce the time to derive the key."

Secure socket layer

The *Secure Socket Layer* and its successor the *Transport Layer Security* (TLS) protocol are intended to provide a flexible means for clients and servers to communicate using a secure channel.

In typical use the server is authenticated while the client remains unauthe s to commun pering and message f

There are t

- Peer negotiation for algorithm support.
- Public key encryption-based key exchange and certifi cation.
- Symmetric cipher-based traffic encryption.

Generally the protocol uses a Record Protocol layer th records; each record can be optionally compressed, encrypted and packed with a message authentication code (a signature that uses a shared secret key). Each record has a type that specifies an upper level protocol including Handshake, Change Cipher Spec and Alert Protocol.

TLS protocol stack

TLS handshake

The first phase allows the client and server to establish which cipher, compression method and other connection parameters are to be used. The second phase exchanges certificates. A *master secret* or common secret is ne ta for the purpose of Security

- Numbe ignatures.
- The message that ends the handshake sends a hash of all the d data seen by both parties.
- Hashing is done by combining (XORing) the results of bo HA, in case one is found to be vulnerable.

TLS record protocol

Optional compression is included because the computations can share work with the encryption and thereby save work overall; i.e. it is faster than compressing separately and then securely transmitting the compressed data.