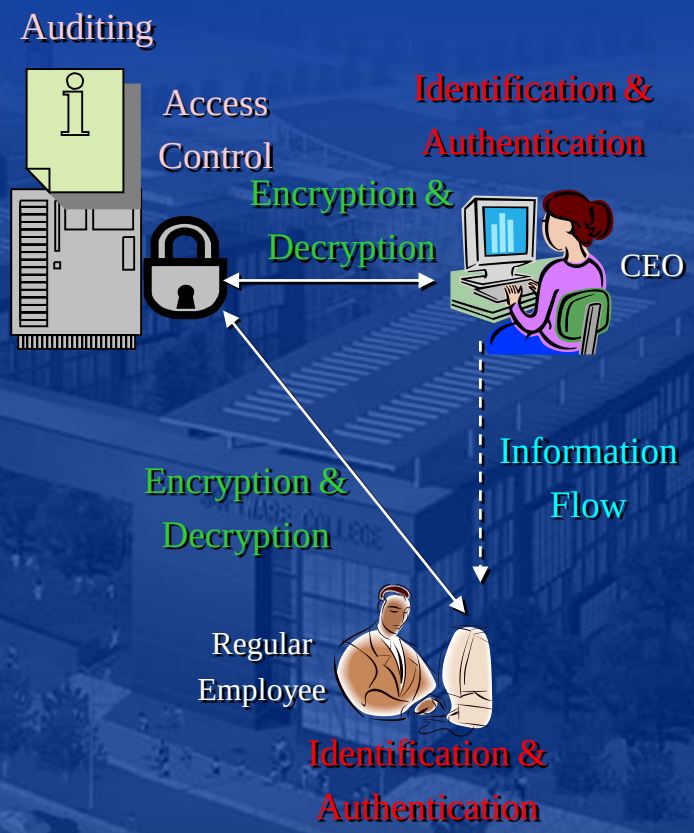
Security and Privacy

Prof. Jingsha He
School of Software Engineering
Beijing University of Technology

Identification and Authentication



Reading Material

- Matt Bishop
  - Chapter 12
  - Chapter 14
  - Chapter 10 (Section 10.2.2)

Identity, Identification and Authentication

- Definitions
  - Identity
    - Representation of an entity inside a computer system
    - It often implies the use of a unique name for an entity
  - Identification
    - Presentation of an identity to a security system
  - Authentication
    - Verification of the identity of an entity
    - A binding of an identity to an entity

Identity

- Purposes
  - For access control
  - For accountability
    - Logging & Auditing
- Identities in a security system
  - A data file (an object in general)
    - File name: for the human being
    - File descriptor: for a process
    - File allocation table entry: for the kernel
  - A user
    - Any name comprised of an arbitrary number of alphanumeric characters
      - May be constrained in some ways

Groups and Roles

- An identity may refer to an entity that is comprised of a group of entities
  - A convenient way of performing access control and other security functions to a set of entities at the same time
  - Models of groups
    - Static: alias to a set of entities
    - Dynamic: construct for grouping a set of entities
- An identity may refer to a role
  - To tie entities together
  - To represent rights or security functions to which entities are assigned or entitled

Identity and Trust

- Requirements
  - Identities should be unique
  - Identities should be bound to the right entities
- Truthfulness of identity
  - A trust issue
  - Certificate
    - To bind an identity to an entity

Identity and Certificate

- Certificate issued by a CA
  - Class 1
    - Authentication of an e-mail address
  - Class 2
    - Verification of real name and address through an online database
  - Class 3
    - Background check by an investigative service

Trust of Identity

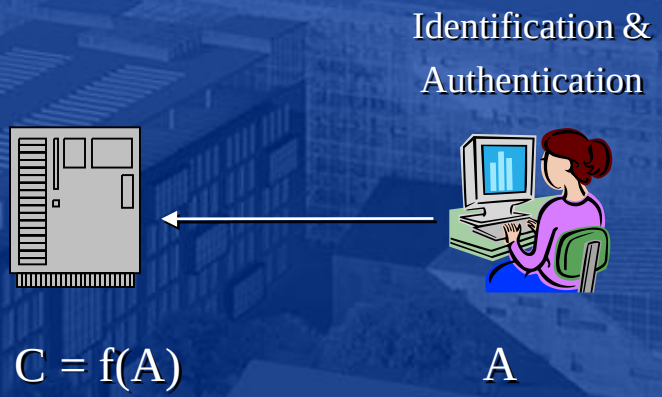
- Trust of a certificate
  - Depending on the trustworthiness of the CA
  - Depending on the level of trust indicated by the CA
    - High: a passport
    - Low: an unsworn statement
  - It's all relative
- The point
  - Identity has the trust issue
  - Certificate also has the trust issue

Authentication

- Purpose
  - To verify that a stated identity really belongs to the right entity
- Methods
  - What the entity knows
    - Password, PIN, DoB, mother's maiden name, etc.
  - What the entity has
    - Badge, ID card, key, etc.
  - What the entity is
    - Fingerprints, personal characteristics, etc.
  - Where the entity is
    - Particular gate, specific terminal, special access device, etc.

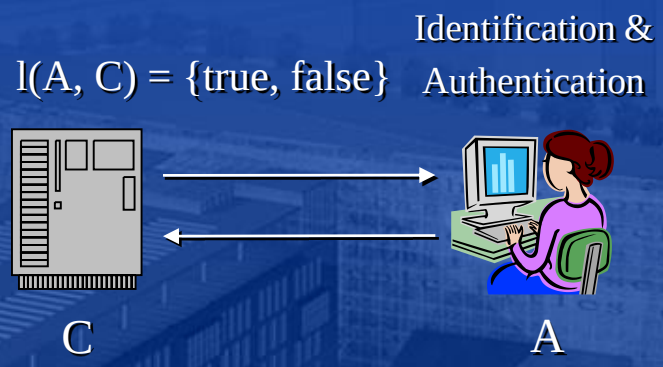
Authentication Components

- For creating and storing authentication information
  - Authentication information: A
    - For an entity to prove its identity
  - Complementary information: C
    - For a system to store authentication information along with the corresponding identity
    - For a system to verify authentication information
  - Complementary functions: F
    - For a system to generate the complementary information from the authentication information
    - For $f \in F$, $f: A \rightarrow C$



Authentication Components

- For performing authentication
  - Authentication functions: L
    - For the system to verify an identity
    - For $l \in L$, $l: A \times C \rightarrow \{true, false\}$
- For managing authentication information
  - For an entity to create or to alter the authentication and the corresponding complementary information



Passwords

- Purpose
  - To use information that an entity knows to verify that a stated identity really belongs to the entity
- Authentication method
  - What an entity knows
- Password protection
  - Passwords are not allowed to be transmitted without proper protection
  - For $f \in F$, $f: A \rightarrow C$ uses a one-way hash function

Password Attacks

- Dictionary attack
  - The guess of a password through repeated trial and error
- Types of attacks
  - Type 1: C and F are available
    - For each guessed password p, compute f(p) for each f∈F until the result matches the stored complementary information c for the same entity
  - Type 2: L is available
    - For each guessed password p, invoke each l∈L and, if “true” is returned, p is the correct password

Counter-Measures to Password Guessing

- Goal
  - To maximize the amount of time consumed before the password is correctly guessed
- Calculation
  - P: probability of correctly guessing a password in a specified period of time
    - In number of time units
  - G: number of password guesses that can be carried out in one time unit
  - T: number of time units for the calculation
  - N: total number of possible passwords
  - Anderson's Formula: $P \geq TG/N$ or $N \geq TG/P$

A Scenario of Password Guessing

- The formula: $P \geq TG/N$
- The scenario
  - R: number of bytes per minutes that can be sent over a communication line
  - E: number of characters for each log-in
  - S: length of a password
  - A: number of characters in the alphabet from which the characters of the password are drawn
  - M: number of months for the password guess
  - Then, $N = A^S$, $G = R/E$, $T = 4.32 \times 10^4 M$

    $P \geq 4.32 \times 10^4 M(R/E)/A^S$ or

    $A^S \geq 4.32 \times 10^4 \times M \times R/(P \times E)$

An Example of Password Guessing

- The objective
  - To determine the minimum length of passwords in a system
- Parameters
  - A = 96 characters
  - G = 10^4 per second
  - P = 0.5
  - T = 365 days = 365 x 24 x 60 x 60 seconds
- Assumptions
  - The length of time required to try out each password is constant
  - All passwords are equally like to be selected
- The result
  - $N \geq TG/P = 6.31 \times 10^{11}$
  - $N = \sum_{i=1}^{S} 96^i \geq 6.31 \times 10^{11} \Rightarrow S \geq 6$

Password Selection

- Theorem
  - When the selection of a password from a set of possible passwords is equally probable, the expected time that is needed for guessing a password is the longest
- In reality?
- Strong passwords
  - At least one digit
  - At least one letter
  - At least one punctuation character
  - At least one control character

Methods against Password Guessing

- Exponential back-off
  - Wait for t^{n-1} seconds before the next log-in when the n^{th} authentication attempt fails
    - t is a system parameter
- Disconnection
  - Disconnect after a specified number of failed attempts
- Disabling
  - Disable after a specified number of failed attempts
- Jailing (honey pot)
  - Fool the attacker, then record all the activities that the attacker conducts

Challenge and Response

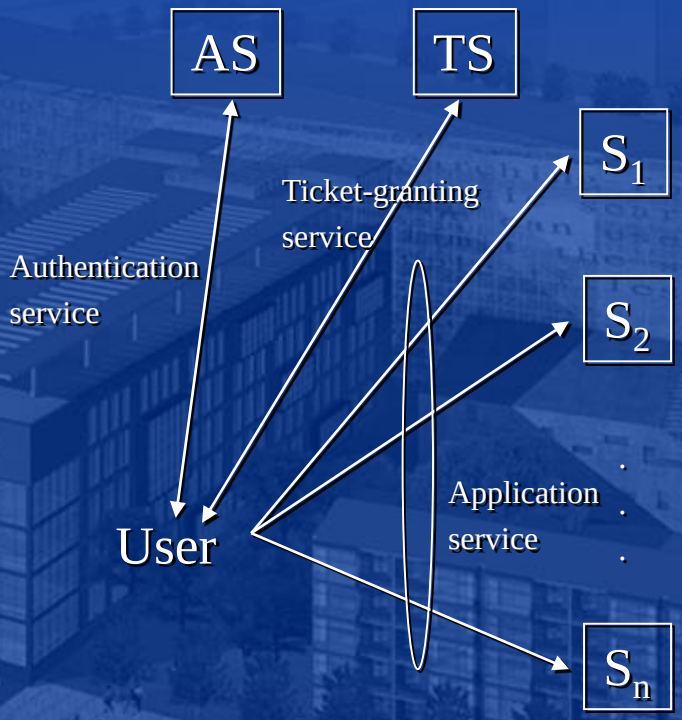
- Purpose
  - To fight against replay attacks
- Methods
  - Pass algorithm
    - Response calculation function is kept secret
  - One-time password
    - Password is invalidated as soon as it is used
  - Challenge and response
    - A challenge is generated and sent to the authenticating entity
    - A response is calculated based on the challenge and sent back to the authentication system for verification

Biometrics

- Purpose
  - The use of automated measurement of biological or behavioral features to characterize and, hence, identify an entity
- Methods
  - Fingerprints
  - Voices
  - Eyes
  - Faces
  - Keystrokes (pressure, interval, duration, position, etc.)
  - Some combinations of the above

Kerberos Authentication

- Foundation
  - Needham-Schroeder protocol plus Denning and Sacco modification
- Kerberos application scenario
  - A system consists of a central authentication server AS, a ticket-granting server TS and one or more application servers S_1, …, S_n
  - AS authenticates a user to the Kerberos system
  - TS issues tickets to the user to authenticate to the application servers
  - S_1, S_2, …, S_n can be accessed by the user by presenting tickets issued by TS



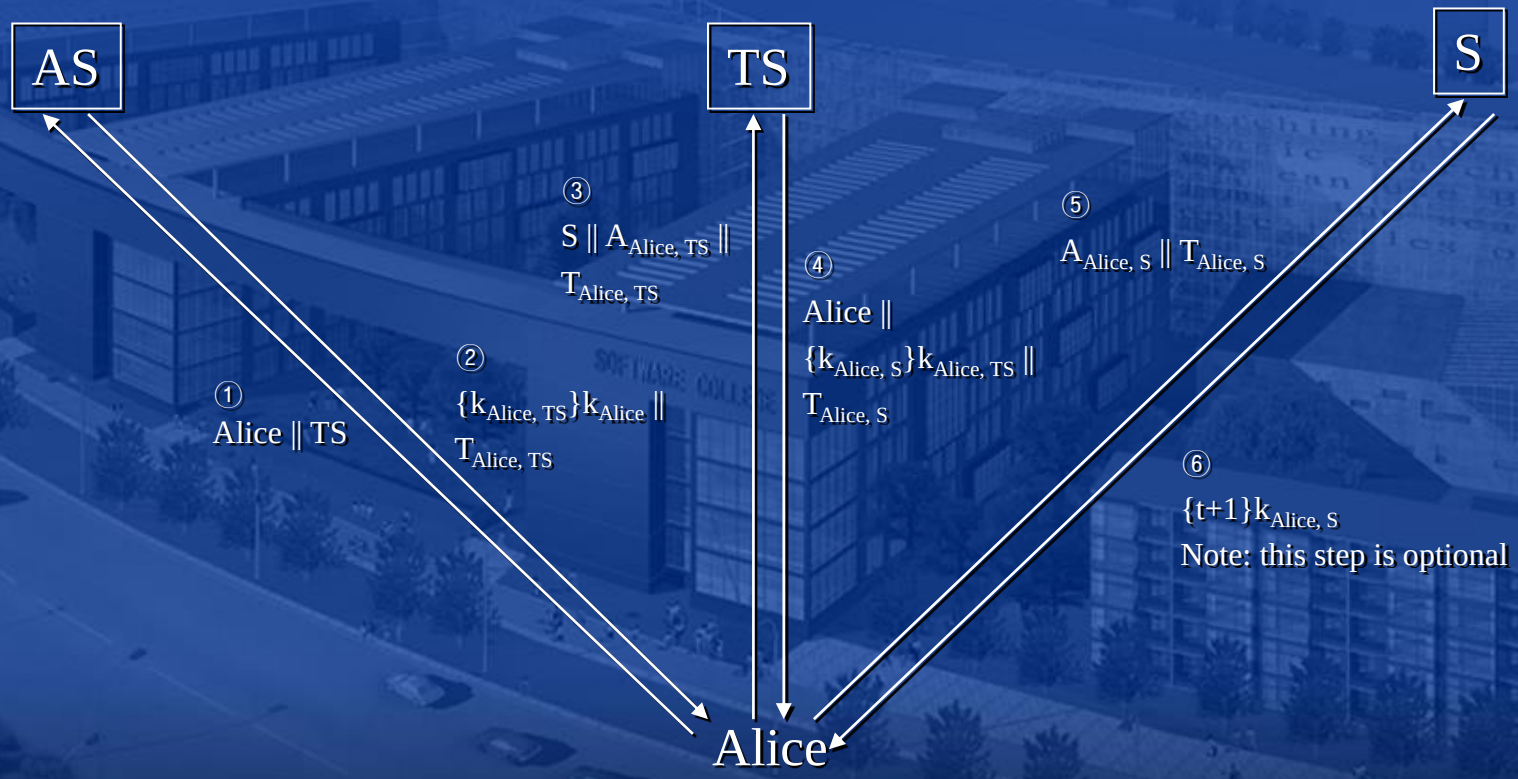
Components of the Kerberos Protocol

- Secret key based cryptography
- The Authentication Server AS shares a secret key with each and every user and with the Ticket-Granting Server TS
  - Question: how to achieve the above?
- The Ticket-Granting Server TS shares a secret key with each and every of the applications servers S_1, …, S_n

Components of the Kerberos Protocol

- Ticket
  - $T_{Alice, Server}$ = {Alice || Alice's address || valid time || $k_{Alice, Server}$}k_{Server}
    - $k_{Alice, Server}$ is the session key generated by the server that created the ticket to be shared between "Alice" and "Server" so as to access "Server"
    - k_{Server} is the secret key that "Server" shares with the server that created the ticket
  - To be presented by Alice to Server for access
- Authenticator
  - $A_{Alice, Server}$ = {Alice || t || k_t}$k_{Alice, Server}$
    - $k_{Alice, Server}$ is the session key that is shared between "Alice" and "Server" so as to access "Server"
    - t is the timestamp when the authenticator is created
    - k_t is an alternative session key
  - To prove to Server that Alice has the session key

The Kerberos Protocol



Significance of Kerberos

- Single sing-on
  - User only needs to log in once with the Authentication Server (AS)
    - Result: a ticket-issuing ticket is issued to the user to access the Ticket-Granting Server (TS)
  - TS issues tickets to the user to access the application servers
    - Result: logging-in to the application servers is transparent to the user
- Widely used in financial systems and large-scale e-commerce applications

Summary

- Identity
- Identification
- Authentication
  - Passwords and password attacks
  - Challenge and response
  - Biometrics
  - The Kerberos protocol

Thought of the Lecture

- Do you know how authentication works in your network?
- What scheme is used?
- How susceptible is it to attacks?
- If somebody points a gun at a user, the gunner could get any authentication desired.
- Do you know how reliable your data storage is?

Q & A