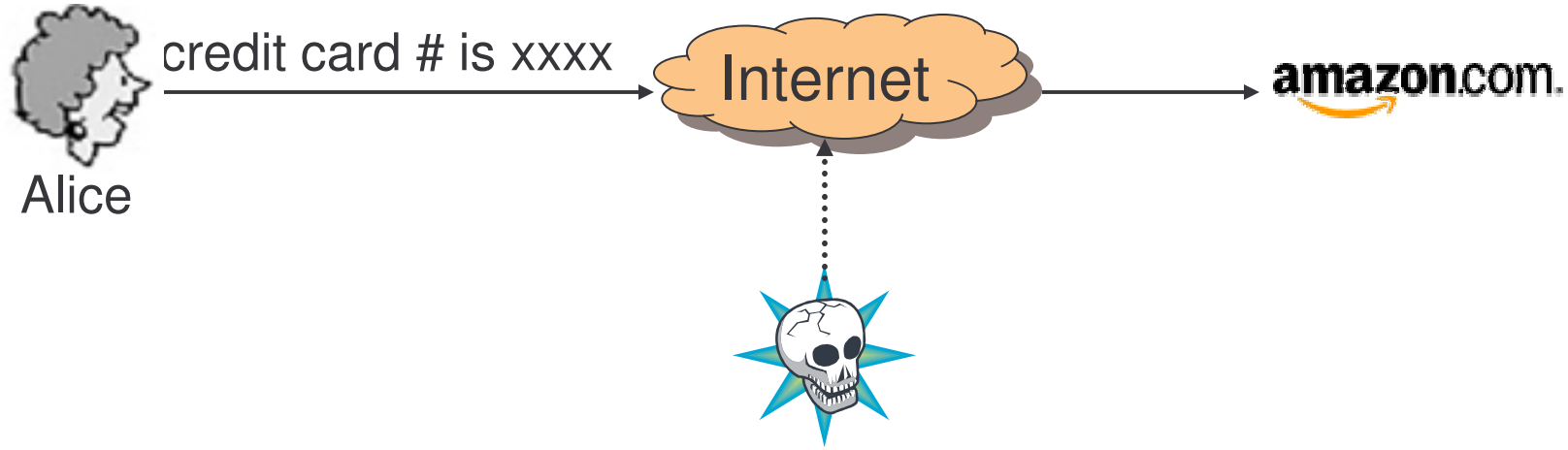
ECE 230x

Network Security Basics

Outline

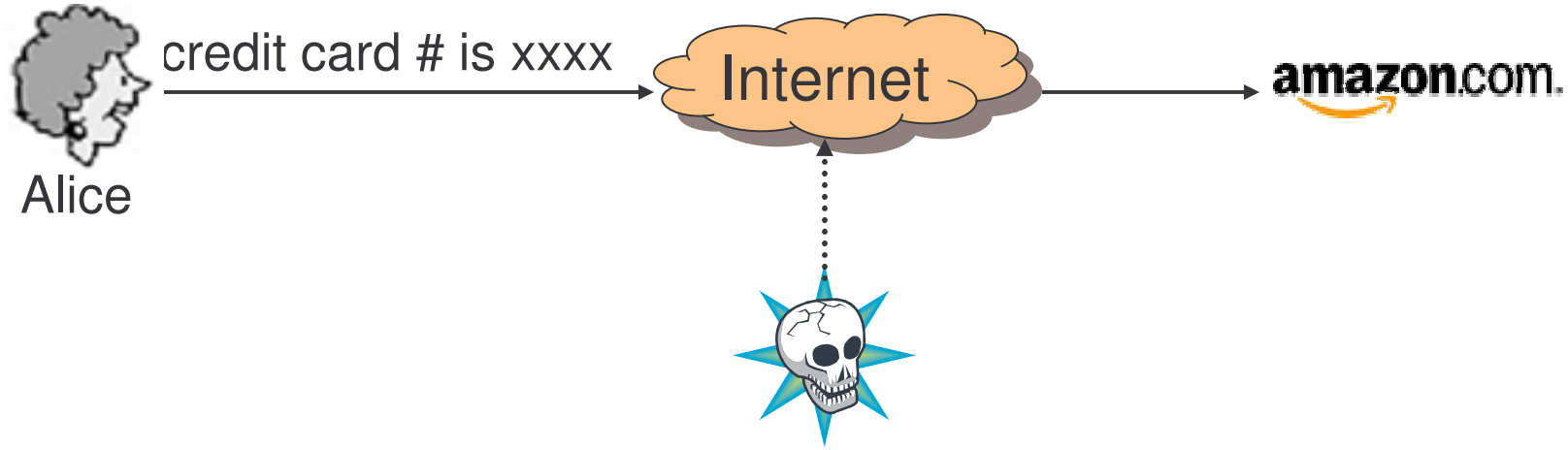
- Common security requirements
- Basic security tools
  - Secret-key cryptography
  - Public-key cryptography
- Example
  - Online shopping with Amazon

Common Security Requirements



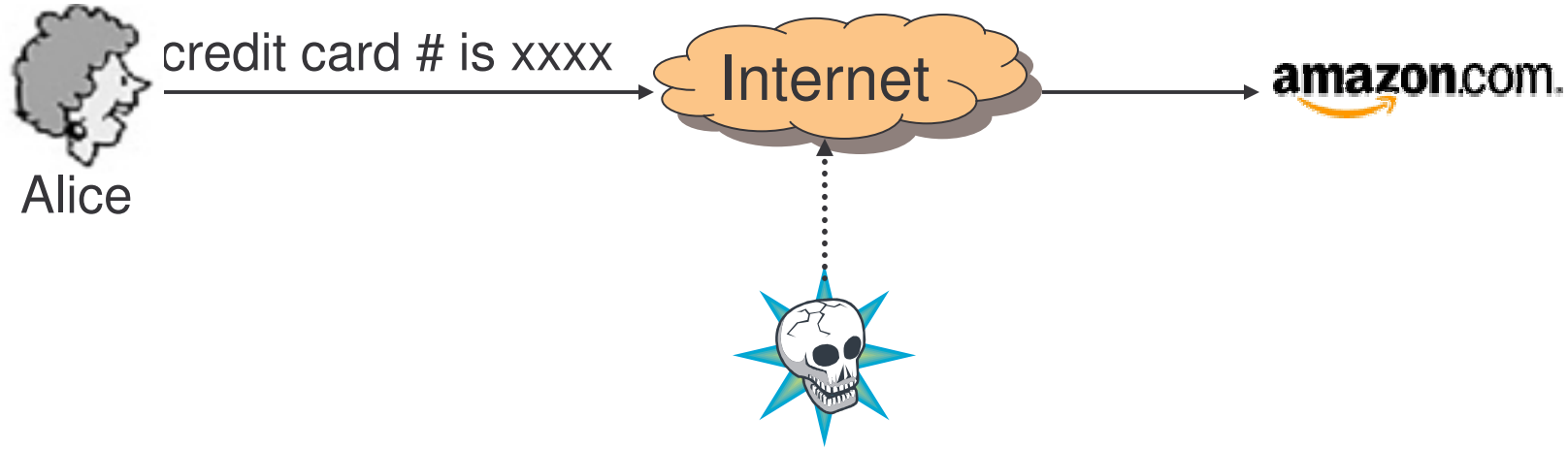
- What could the hacker possibly do?
  - Attempt to read the messages from Alice to Amazon, thus learning Alice's credit card number
  - Impersonate Amazon in the transaction – the *Phishing* attack
  - Modify the delivery address or content of Alice's order
- Do you still feel safe to do online shopping?
  - 85% of Americans have security concerns with online shopping, and 24% have abandoned it

Common Security Requirements



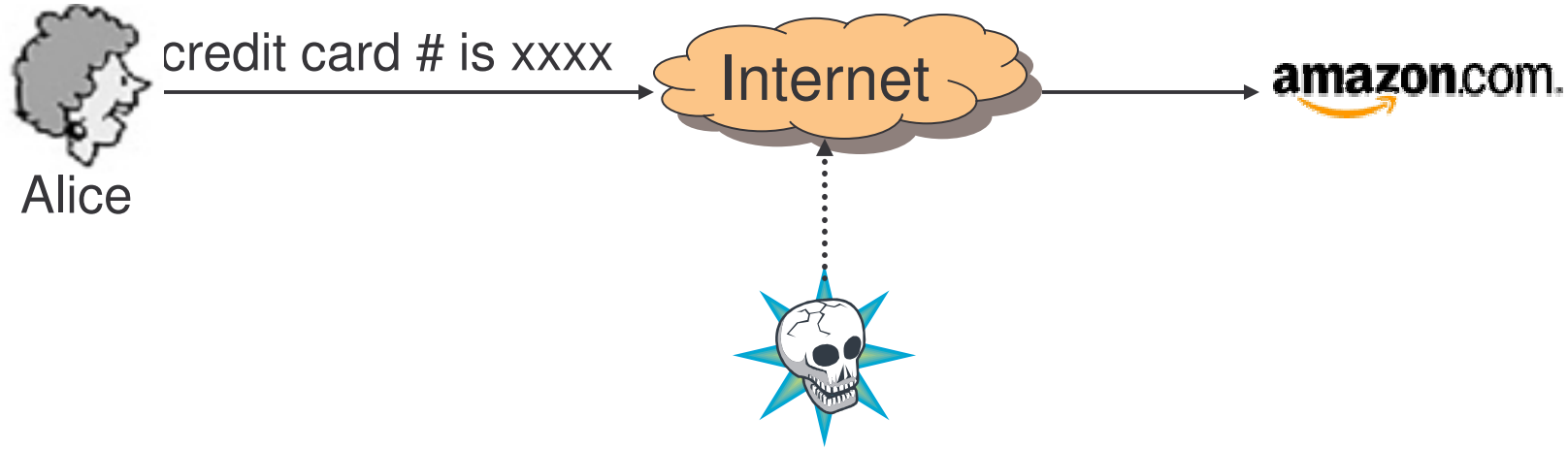
- Data confidentiality or secrecy
  - Message sent between Alice and Amazon should not be readable by or should appear unintelligible to the hacker
  - Preventing unauthorized access to secret data

Common Security Requirements



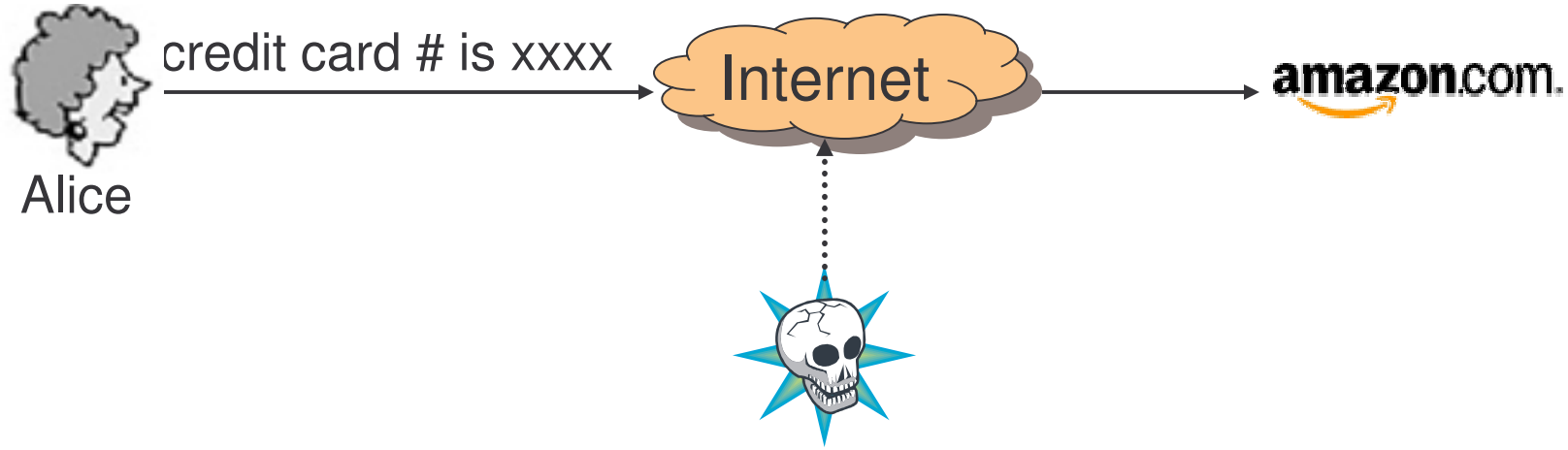
- Data integrity
  - Amazon should be able to detect if data sent by Alice has been modified by the hacker; vice versa
  - Ensuring data has not been altered by unauthorized means such as insertion, deletion, and substitution

Common Security Requirements



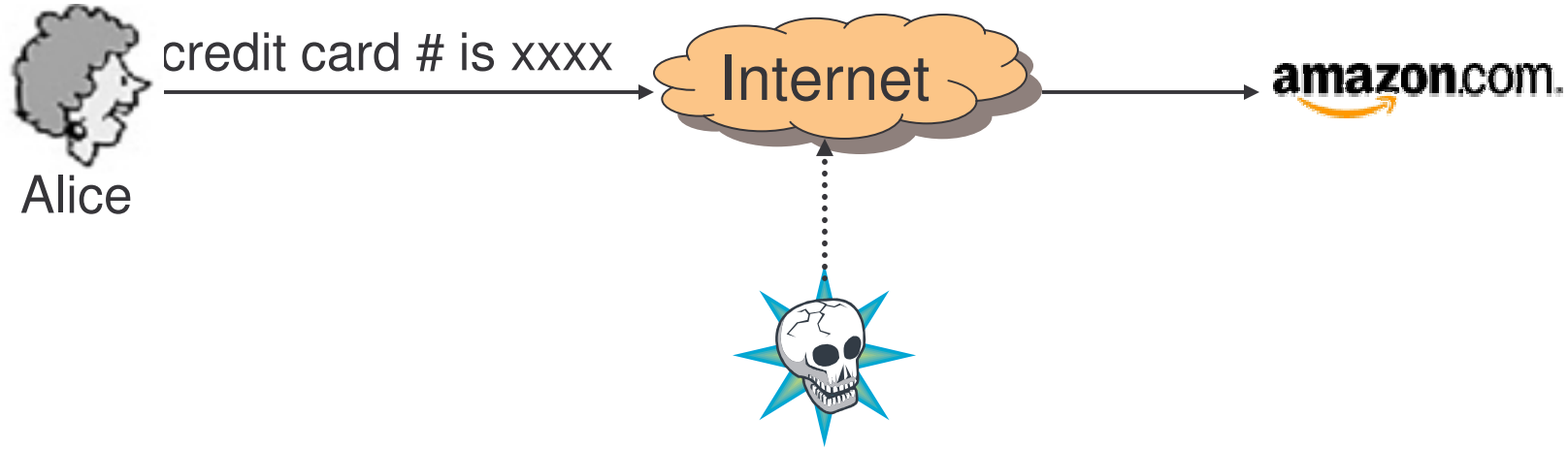
- Authentication
  - Alice and Amazon should be able to verify each other's identity
  - Enabling two communicating parties to identity each other

Common Security Requirements



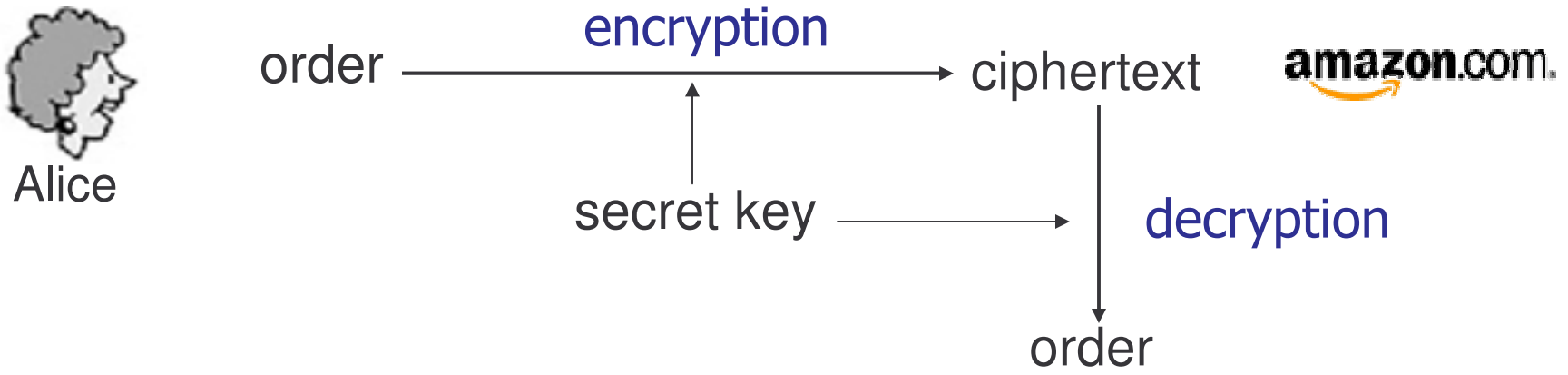
- Non-repudiation
  - Both Alice and Amazon cannot deny the transaction
  - Preventing an entity from denying previous commitments or actions

Secret-Key Cryptography



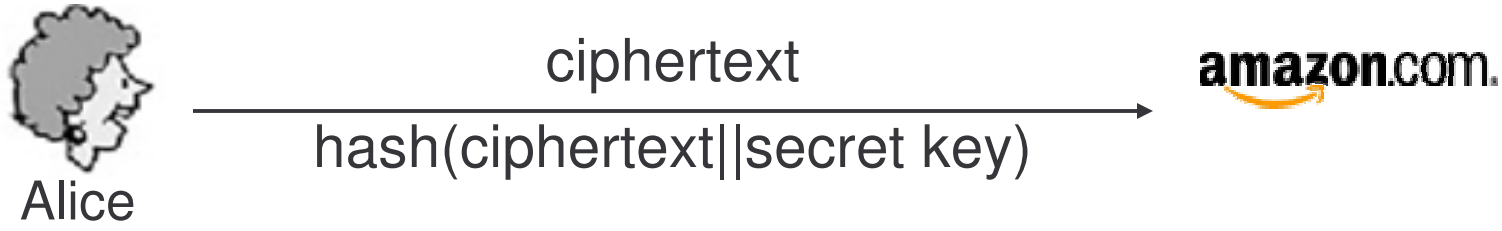
- Alice and Amazon share a common secret, which is called a secret key and only known to them two

Secret-Key Cryptography



- How is data confidentiality achieved?
  - Alice encrypts the order using the secret key and generates the ciphertext seeming unrelated to the original order
  - Amazon decrypts the ciphertext using the secret key
  - No one else can decrypt the ciphertext without the secret key

Secret-Key Cryptography



- Hash function
  - Taking an input message and produces a fixed-length output, called a *hash value*
  - It is infeasible to find two distinct input messages which hash to a common hash value
- How are data integrity and authentication achieved?
  - Alice sends to Amazon the ciphertext along with a hash value computed over the ciphertext and the secret key
  - Before doing decryption, Amazon recalculates the hash value and checks its equality to the received one
  - If the hash check is successful, Amazon determines the message was indeed sent by Alice and was not modified

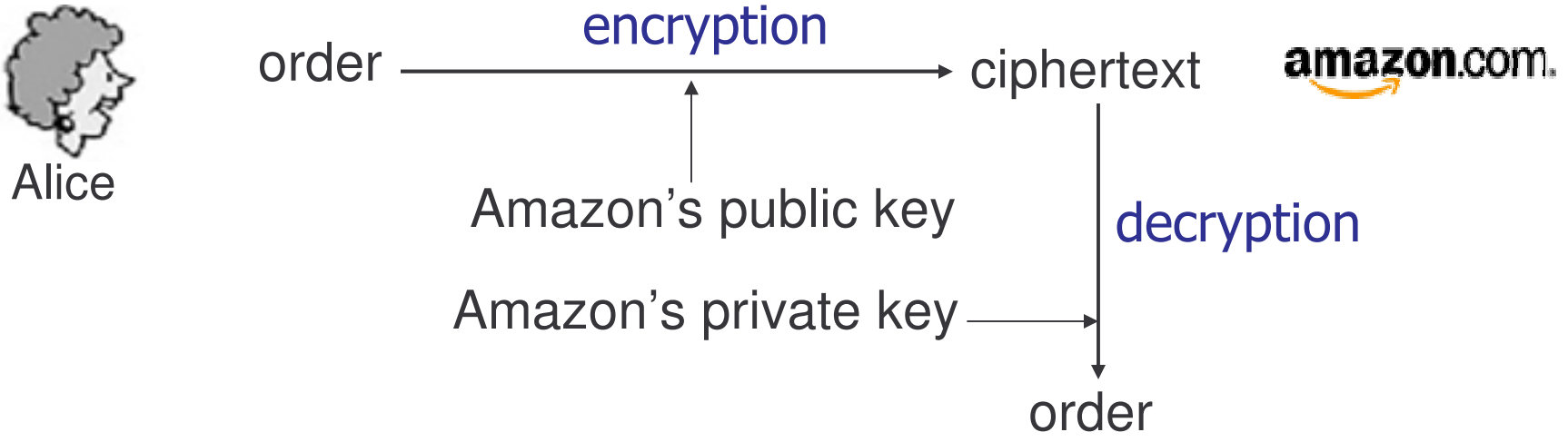
Secret-Key Cryptography

- No support for non-repudiation
  - The secret key is known to two parties, e.g., both Alice and Amazon, and cannot be used to generate a unique digital signature
- Difficult secret-key establishment
  - How could Amazon establish a unique secret key with each of its millions of customers?

Public-Key Cryptography

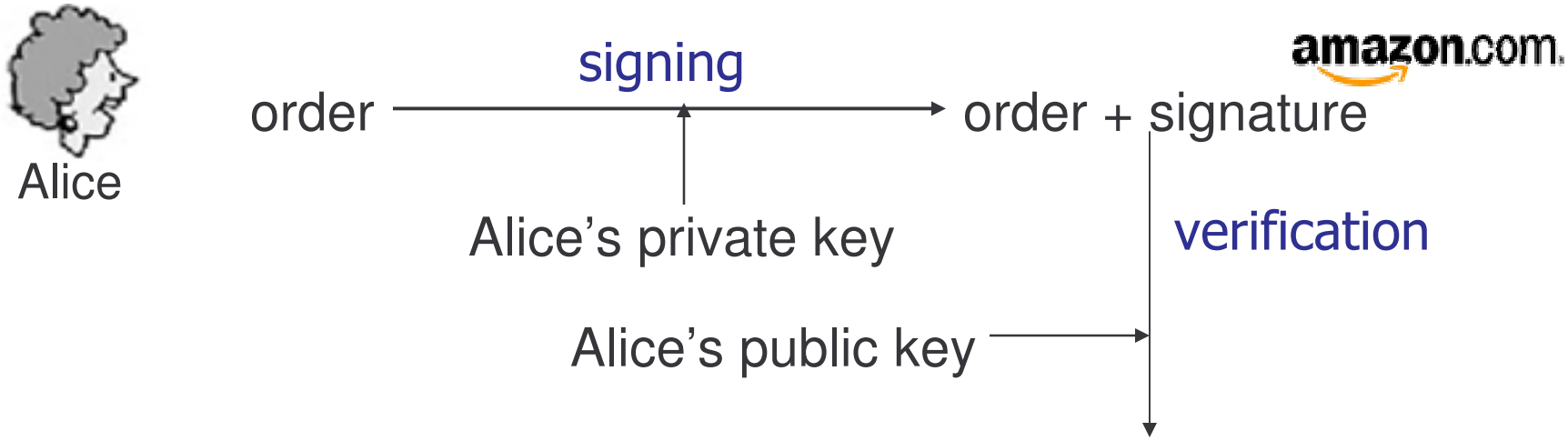
- Alice has two unique keys
  - A private key (K^{-1}), kept confidential to herself
  - A public key (K), preferably known to the entire world
  - There is a one-to-one correspondence between K & K^{-1}
  - It is computationally infeasible to determine K^{-1} given K
- Amazon also has unique pair of public and private keys

Public-Key Cryptography



- How is data confidentiality achieved?
  - Alice encrypts the order using Amazon's public key
  - Amazon decrypts the ciphertext using its private key
  - No one else can decrypt the ciphertext without knowing Amazon's private key

Public-Key Cryptography



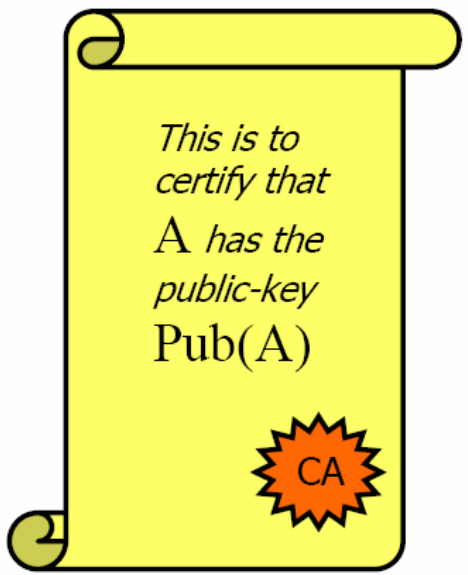
- Digital signatures provide guarantees for authentication, data integrity, and non-repudiation
  - Amazon trusts the order was sent by Alice and was not maliciously modified as nobody else can fake her digital signature
  - Alice cannot deny the order because only she knows her private key and can generate the signature on the order

Public-Key Cryptography

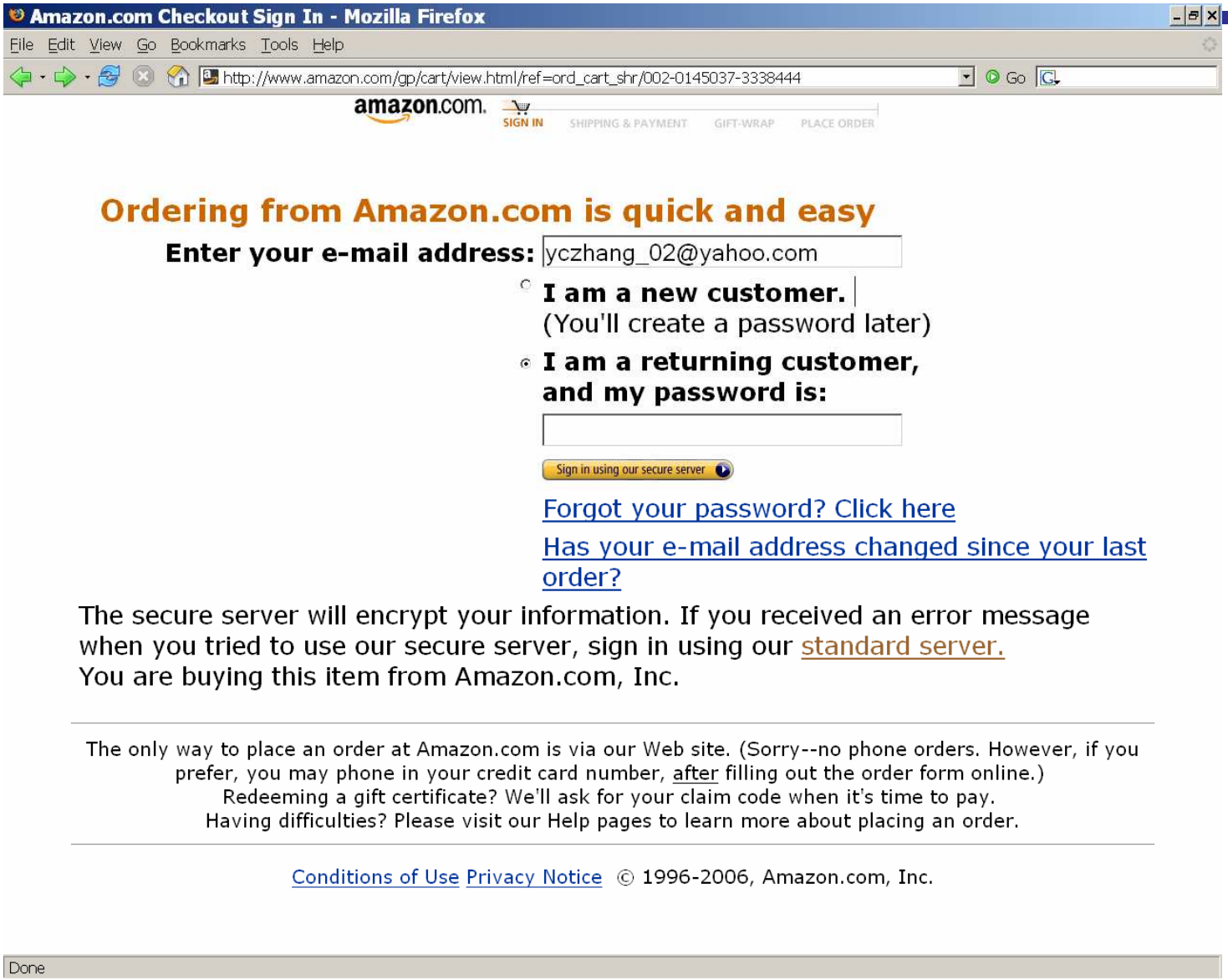
- How could Amazon know Alice's public key?
  - Alice sends her public key directly to Amazon
  - Or Alice puts her public key in some central public-key repository for Amazon to retrieve
- What is the security issue here?
  - The hacker modifies Alice's public key to his and thus is able to impersonate Alice to Amazon
  - The hacker modifies Amazon's public key to his and thus is able to impersonate Amazon to Alice
  - Called public-key impersonation attacks

Public-Key Cryptography

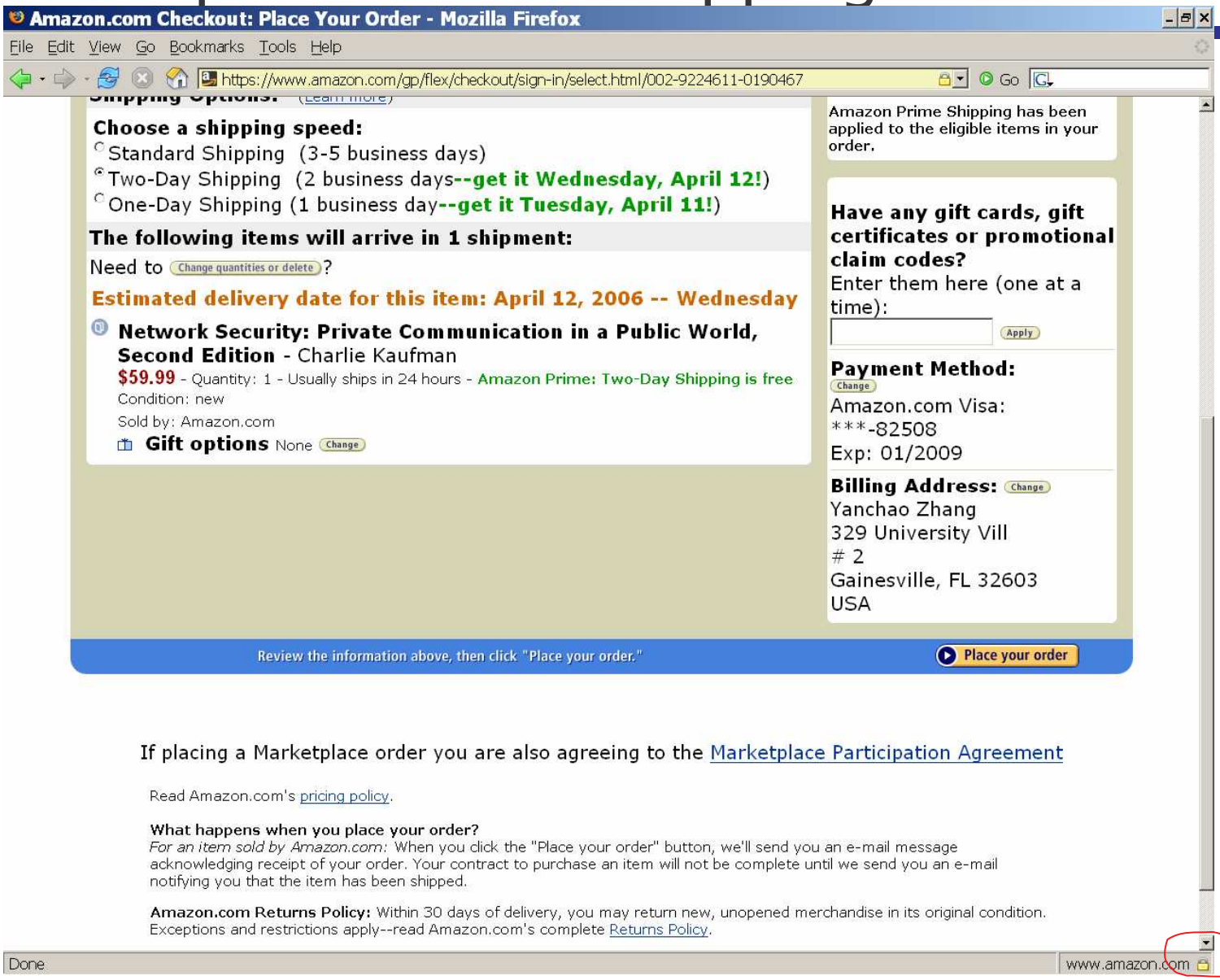
- Solution: public-key certificates
  - Signed messages specifying a name (Alice) and the corresponding public key
  - Generated by a trusted third party known as a Certification Authority (CA) whose public key is publicly known and trustable
  - For a list of common CAs, see http://dmoz.org/Computers/Security/Public_Key_Infrastructure/PKIX/Tools_and_Services/Third_Party_Certificate_Authorities/
  - $cert_{Alice}$:=<Alice, K_{Alice}, expiration-time, CA's signature>
  - Amazon trusts K_{Alice} to be Alice's public key after verifying the CA's signature



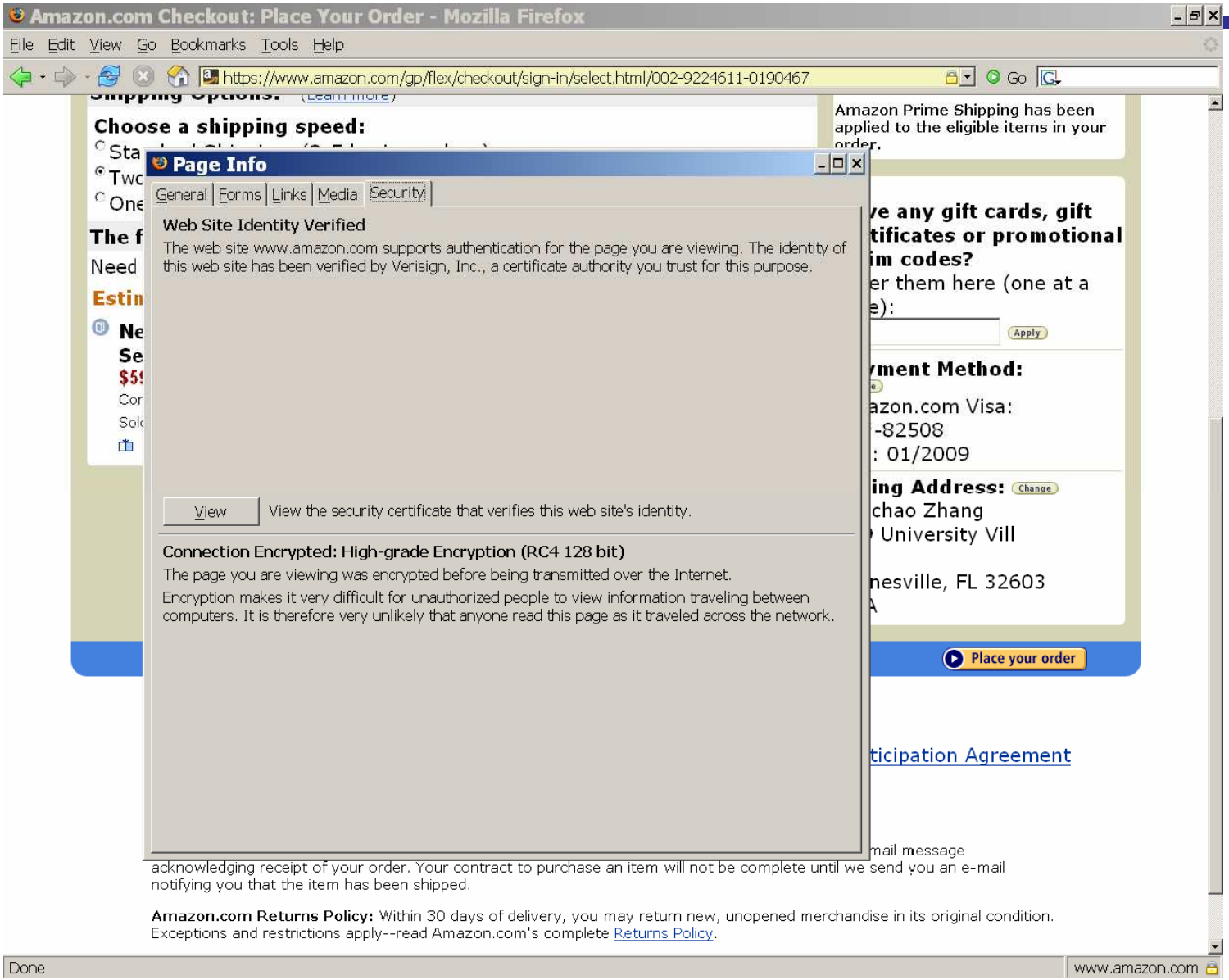
Example: Online Shopping with Amazon



Example: Online Shopping with Amazon



Example: Online Shopping with Amazon



Example: Online Shopping with Amazon

- Major steps (SSL: Secure Sockets Layer Protocol)
  - Firefox initiates a session request to Amazon
  - Amazon returns its public-key certificate issued by VeriSign
  - Firefox uses the pre-stored VeriSign's public key to verify its signature in Amazon's public-key certificate
  - Firefox generates a secret key (a 48-byte number)
  - Firefox encrypts the secret key, and your web account and password with Amazon's public key, and sends the ciphertext to Amazon
  - Amazon uses its private key to decrypt the ciphertext and verifies your password
  - Subsequent communications between Firefox and Amazon will be encrypted and authenticated using secret-key techniques based on the shared secret key