Online Credit Card Purchases are Risky for the Merchant

- Many security procedures that credit card companies rely on are not applicable in online environment (CNP).
- Merchant pays when goods are not delivered, order is disputed or in cases of credit card fraud.
- Percentage of Internet transactions charged back to online merchants much higher than for traditional retailers (3-10% compared to ½-1%).
- To protect selves, merchants can:
  - Refuse to process overseas purchases
  - Insist that credit card and shipping address match
  - Require users to input 3-digit security code printed on back of card

Security

- Confidentiality:
  - prevent unauthorized disclosure of information (attacker cannot spy on you)
- Integrity:
  - prevent unauthorized modification of information (attacker cannot modify your data)
- Authentication:
  - prove your identity to the system (attacker cannot pretend to be an authorized user)
- Availability:
  - guarantee access to the information (attacker cannot prevent you from accessing your information)

Non Repudiation:
- prevent false denial of performed action (user cannot pretend he/she did not do something)

Authorization:
- define what each user is allowed to do

Auditing:
- securely record evidence of performed actions

Fault Tolerance:
- provide some degree of service despite failures or attacks
Points of Vulnerability

Threats
- Unauthorized access
- Masquerading
- Eavesdropping
- Modification, insertion, replay of data
- Disclosure of secret data
- Denial of Service
- Lack of accountability

Countermeasures
- Access control
- Identification and Authentication
- Transport message protection
- Controllable delegation mechanisms
- Non-repudiation
- Security audit logging

Friends and Enemies
- well-known in network security world
- Bob, Alice (lovers!) want to communicate “securely”
- Trudy, the “intruder” may intercept, delete, add messages

Who might Bob, Alice be?
- … well, real-life Bobs and Alices!
- Web browser/server for electronic transactions (e.g., on-line purchases)
- on-line banking client/server
- DNS servers
- routers exchanging routing table updates
- other examples?

Packet Sniffing
- broadcast media
- promiscuous NIC reads all packets passing by
- can read all unencrypted data (e.g. passwords)
- e.g.: C sniffs B’s packets
IP Spoofing

- Can generate “raw” IP packets directly from application, putting any value into IP source address field
- Receiver can’t tell if source is spoofed
- E.g.: C pretends to be B

Denial of Service (DoS)

- Flood of maliciously generated packets “swamp” receiver
- Distributed DOS (DDOS): multiple coordinated sources swamp receiver
- E.g., C and remote host SYN-attack A

Cryptography

- Symmetric key crypto: sender, receiver keys identical
- Public-key crypto: encrypt key public, decrypt key secret

Cryptography

- Substitution cipher: substituting one thing for another
  - Monoalphabetic cipher: substitute one letter for another
- E.g.: plaintext: bob. i love you. alice
ciphertext: nkn. s gktc wky. mgsbc

Perfect Cipher

- Definition:
  - Let C = E[M]
  - Pr[C -> M] = Pr[C -> M | C]
- Example: one time pad
  - Generate random bits b_1 ... b_n
  - E[M_1, ... , M_n] = (M_1 ⊕ b_1 ... M_n ⊕ b_n)
- Cons: size!
- Pseudo Random Generator (seed, offset)
  - G(R) = b_1 ... b_n
  - Indistinguishable from random (efficiently)

Symmetric Key Cryptography

- Bob and Alice share known (symmetric) key: K_{A,B}
  - E.g., key is knowing substitution pattern in monoalphabetic substitution cipher
- Q: how do Bob and Alice agree on key value?
DES

**Data Encryption Standard**
- US encryption standard [NIST 1993]
- 56-bit symmetric key, 64 bit plaintext input
- How secure is DES?
  - DES Challenge: 56-bit-key-encrypted phrase (“Strong cryptography makes the world a safer place”) decrypted (brute force) in 4 months
  - no known “backdoor” decryption approach
- making DES more secure
  - use three keys sequentially (3-DES) on each datum
  - use cipher-block chaining

Block Cipher Chaining
- How do we encode a large message
  - blocks
  - identical blocks should be encoded differently
- Encoding:
  - \( C_i = E[M_i \oplus C_{i-1}] \)
- Decoding:
  - \( M_i = D[C_i] \oplus C_{i-1} \)
- Problems:
  - loss of data, modification

Advanced Encryption Standard (AES)
- new (Nov. 2001) symmetric-key NIST standard, replacing DES
- processes data in 128 bit blocks
- 128, 192, or 256 bit keys
- brute force decryption (try each key) takes 149 trillion years for AES

Key Exchange
- Diffie & Hellman, 1976
- Exchange secret key over insecure medium
- Alice chooses secret value \( K_A \) and Bob chooses secret value \( K_B \)
- Alice chooses \( p \) and \( g \), both prime numbers and \( g \) is primitive module \( p \) (can be public)
- Alice sends to Bob \( (g^k \mod p, p, g^k \mod p) \)
- Bob sends to Alice \( (g, p, g^{K_B} \mod p) \)
- Alice computes \( k = (g^{K_B} \mod p)^k \mod p \) and Bob computes \( k' = (g^k \mod p)^{K_B} \mod p \)
- \( k = k' \), therefore \( k \) is the new secret key

Public Key Cryptography
- **symmetric key crypto**
  - requires sender, receiver know shared secret key
  - Q: how to agree on key in first place (particularly if never “met”)?
- **public key cryptography**
  - radically different approach [Diffie-Hellman76, RSA78]
  - sender, receiver do not share secret key
  - encryption key public (known to all)
  - decryption key private (known only to receiver)
Public Key Cryptography

RSA
- Rivest, Shamir, Adleman, 1977
- Most commonly used encryption/authentication algorithm (included in Web browsers)
- Algorithm:
  - use two large prime numbers to derive a public key and a private key
  - prime numbers can be discarded
  - keep private key secret, publish public key

RSA: Choosing Keys
1. Choose two large prime numbers \( p, q \) (e.g., 1024 bits each)
2. Compute \( n = pq \), \( z = (p-1)(q-1) \)
3. Choose \( e \) (with \( e \neq n \)) that has no common factors with \( z \). (\( e, z \) are "relatively prime")
4. Choose \( d \) such that \( ed-1 \) is exactly divisible by \( z \). (in other words: \( ed \mod z = 1 \)).
5. Public key is \( (n,e) \). Private key is \( (n,d) \).

RSA: Encryption/Decryption
0. Given \( (n,e) \) and \( (n,d) \) as computed above
1. To encrypt bit pattern, \( m \), compute
   \[ c = m^e \mod n \] (i.e., remainder when \( m^e \) is divided by \( n \))
2. To decrypt received bit pattern, \( c \), compute
   \[ m = c^d \mod n \] (i.e., remainder when \( c^d \) is divided by \( n \)).

Magic happens!

RSA Example
Bob chooses \( p=5, q=7 \). Then \( n=35, z=24 \).
\( e=5 \) (so \( e, z \) relatively prime),
\( d=29 \) (so \( ed-1 \) exactly divisible by \( z \)).

<table>
<thead>
<tr>
<th>Letter</th>
<th>( m )</th>
<th>( m^e \mod n )</th>
<th>( c )</th>
</tr>
</thead>
<tbody>
<tr>
<td>l</td>
<td>12</td>
<td>1524832</td>
<td>17</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Encrypt:</th>
<th>( c = m^e \mod n )</th>
<th>( m = c^d \mod n )</th>
<th>Letter</th>
</tr>
</thead>
<tbody>
<tr>
<td>m</td>
<td>12</td>
<td>41116482720672081905111518232372065</td>
<td>l</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Decrypt:</th>
<th>( m = c^d \mod n )</th>
<th>Letter</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>12</td>
<td>l</td>
</tr>
</tbody>
</table>