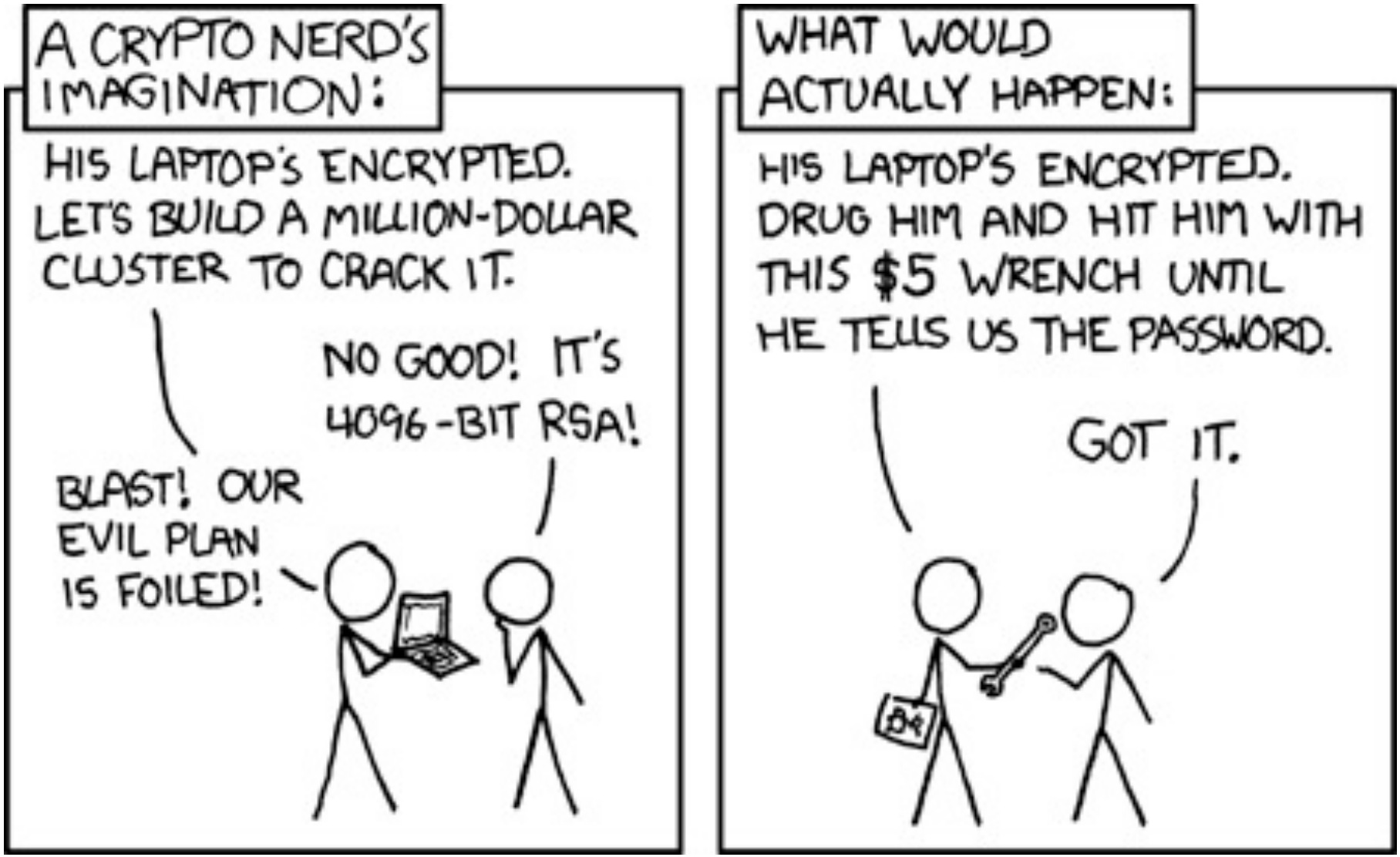

A CRYPTO NERD'S IMAGINATION:
HIS LAPTOP'S ENCRYPTED. LET'S BUILD A MILLION-DOLLAR CLUSTER TO CRACK IT.
NO GOOD! IT'S 4096-BIT RSA!
BLAST! OUR EVIL PLAN IS FOILED!
WHAT WOULD ACTUALLY HAPPEN:
HIS LAPTOP'S ENCRYPTED. DRUG HIM AND HIT HIM WITH THIS $5 WRENCH UNTIL HE TELLS US THE PASSWORD.
GOT IT.
$5

CSE 484 / CSE M 584 (Autumn 2011)

Asymmetric Cryptography

Daniel Halperin
Tadayoshi Kohno

Thanks to Dan Boneh, Dieter Gollmann, John Manferdelli, John Mitchell, Vitaly Shmatikov, Bennet Yee, and many others for sample slides and materials ...

Class updates

- Remember current events and security reviews are due **this Friday**
- **Office hours** today (with Miro) in CSE 210
- (Short) Homework 3; due next Wednesday
  - Individual assignment
  - Available on Catalyst at 3:30 today
- (Short) Lab 3 out tomorrow or Friday
  - Short, fun privacy "scavenger hunt"
  - Groups of 1 to 3

Today

- Wrap up RSA / Public Key *Cryptography*
- Switch to Public Key *Protocols*

Advantages of Public-Key Crypto

- Confidentiality without shared secrets
  - Very useful in open environments
  - No "chicken-and-egg" key establishment problem
    - With symmetric crypto, two parties must share a secret before they can exchange secret messages
    - Caveats to come
- Authentication without shared secrets
  - Use digital signatures to prove the origin of messages
- Reduce protection of information to protection of authenticity of public keys
  - No need to keep public keys secret, but must be sure that Alice's public key is really her true public key

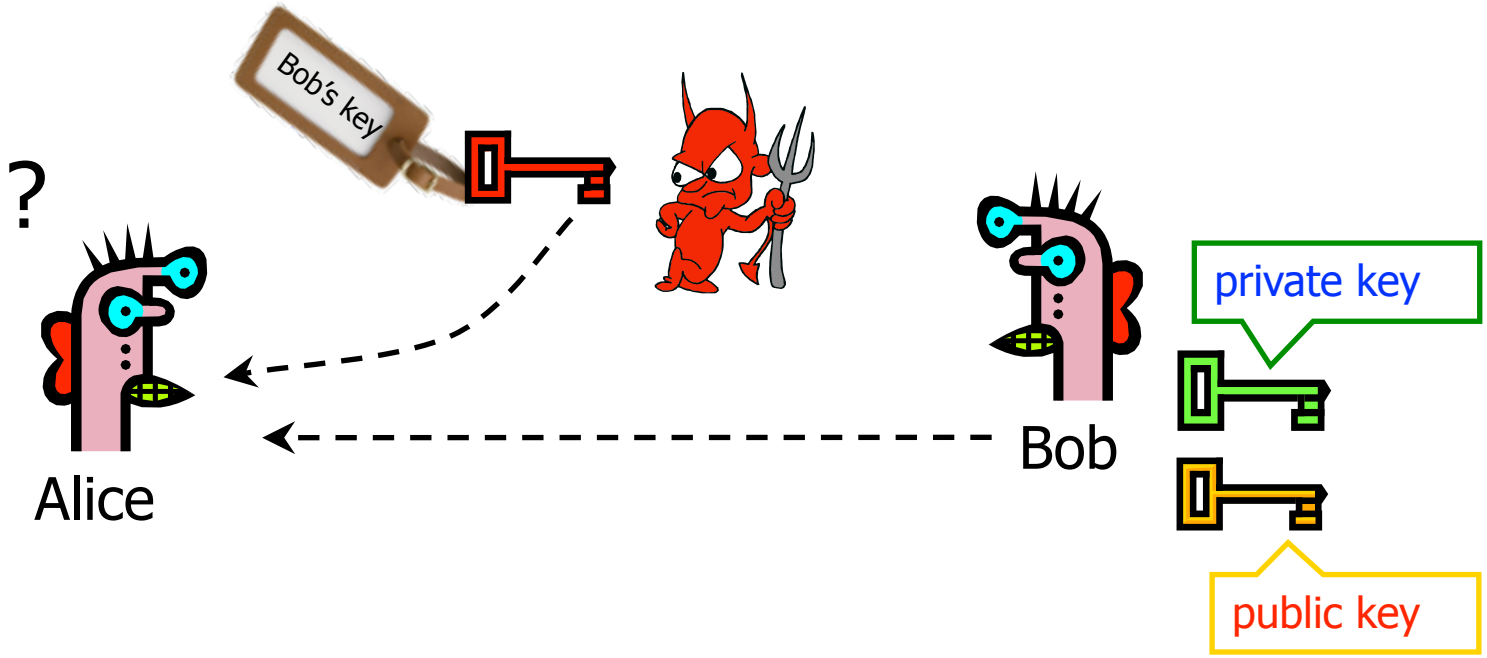
Disadvantages of Public-Key Crypto

- Calculations are 2-3 orders of magnitude slower
  - Modular exponentiation is an expensive computation
  - Typical usage: use public-key cryptography to establish a shared secret, then switch to symmetric crypto
    - E.g., IPsec, SSL, SSH, ...
- Keys are longer
  - 1024+ bits (RSA) rather than 128 bits (AES)
- Relies on unproven number-theoretic assumptions
  - What if factoring is easy?
    - Factoring is believed to be neither P, nor NP-complete
  - (Of course, symmetric crypto also rests on unproven assumptions)

Note: Optimizing Exponentiation

- How to compute M^x mod N? Say x=13
- Sums of power of 2, x = 8+4+1 = $2^3+2^2+2^0$
- Can also write x in binary, e.g., x = 1101
- Can solve by repeated squaring
  - y = 1;
  - y = y^2 * M mod N // y = M
  - y = y^2 * M mod N // y = M^2 *M = M^{2+1} = M^3
  - y = y^2 mod N // y = $(M^{2+1})^2$ = M^{4+2}
  - y = y^2 * M mod N // y = $(M^{4+2})^2$ *M = M^{8+4+1}
- Does anyone see a potential issue?

Authenticity of Public Keys



Problem: How does Alice know that the public key she received is really Bob's public key?

Distribution of Public Keys

- Public announcement or public directory
  - Risks: forgery and tampering
- Public-key certificate
  - Signed statement specifying the key and identity
    - $sig_{CA}(\text{“Bob”}, PK_B)$
- Common approach: certificate authority (CA)
  - Single agency responsible for certifying public keys
  - After generating a private/public key pair, user proves his identity and knowledge of the private key to obtain CA's certificate for the public key (offline)
  - Every computer is pre-configured with CA's public key

Hierarchical Approach

- Single CA certifying every public key is impractical
- Instead, use a trusted root authority
  - For example, Verisign
  - Everybody must know the public key for verifying root authority's signatures
- Root authority signs certificates for lower-level authorities, lower-level authorities sign certificates for individual networks, and so on
  - Instead of a single certificate, use a certificate chain
    - $sig_{Verisign}$("AnotherCA", $PK_{AnotherCA}$), $sig_{AnotherCA}$("Alice", PK_A)
  - What happens if root authority is ever compromised?

Many Challenges

Spoofing URLs With Unicode

Posted by timothy **on Mon May 27, '02 09:48 PM**
from the **there-is-a-problem-with-this-certificate** dept.

Embedded Geek writes:

> "Scientific American has an interesting article about how a pair of students at the Technion-Israel Institute of Technology registered "microsoft.com" with Verisign, using the Russian Cyrillic letters "c" and "o". Even though it is a completely different domain, the two display identically (the article uses the term "homograph"). The work was done for a paper in the **Communications of the ACM** (the paper itself is not online). The article characterizes attacks using this spoof as "scary, if not entirely probable," assuming that a hacker would have to first take over a page at another site. I disagree: sending out a mail message with the URL waiting to be clicked ("Bill Gates will send you ten dollars!") is just one alternate technique. While security problems with Unicode have been noted here before, this might be a new twist."

http://it.slashdot.org/story/08/12/30/1655234/CCC-Create-a-Rogue-CA-Certificate
http://www.win.tue.nl/hashclash/rogue-ca/

Many Challenges

CCC Create a Rogue CA Certificate

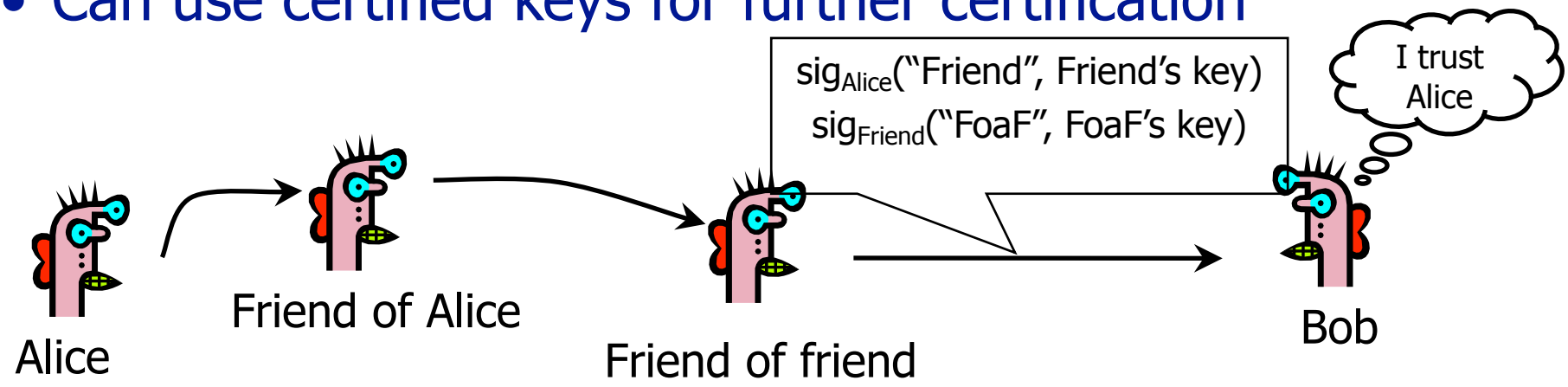
Posted by CmdrTaco on Tue Dec 30, 2008 12:14 PM
from the **they-even-faked-this-dept** dept.

t3rmin4t0r writes

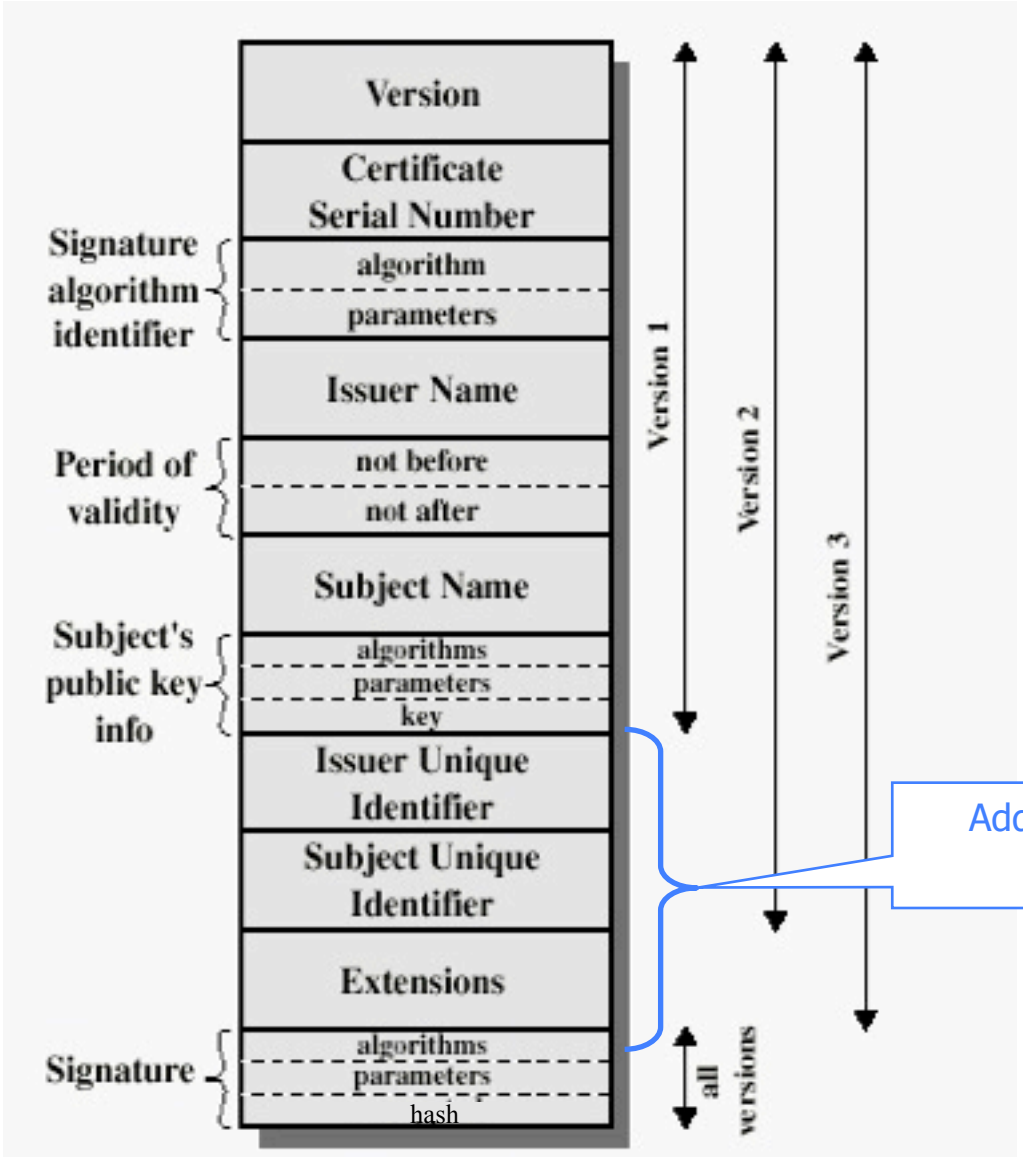
> "Just when you were breathing easy about Kaminsky, DNS and the word hijacking, by repeating the word SSL in your head, the hackers at CCC were busy at work making a hash of SSL certificate security. Here's the scoop on how they set up their own rogue CA, by (from what I can figure) reversing the hash and engineering a collision up in MD5 space. Until now, MD5 collisions have been ignored because nobody would put in that much effort to create a useful dummy file, but a CA certificate for phishing seems juicy enough to be fodder for the botnets now."

Alternative: "Web of Trust"

- Used in PGP (Pretty Good Privacy)
- Instead of a single root certificate authority, each person has a set of keys they "trust"
  - If public-key certificate is signed by one of the "trusted" keys, the public key contained in it will be deemed valid
- Trust can be transitive
  - Can use certified keys for further certification



X.509 Certificate



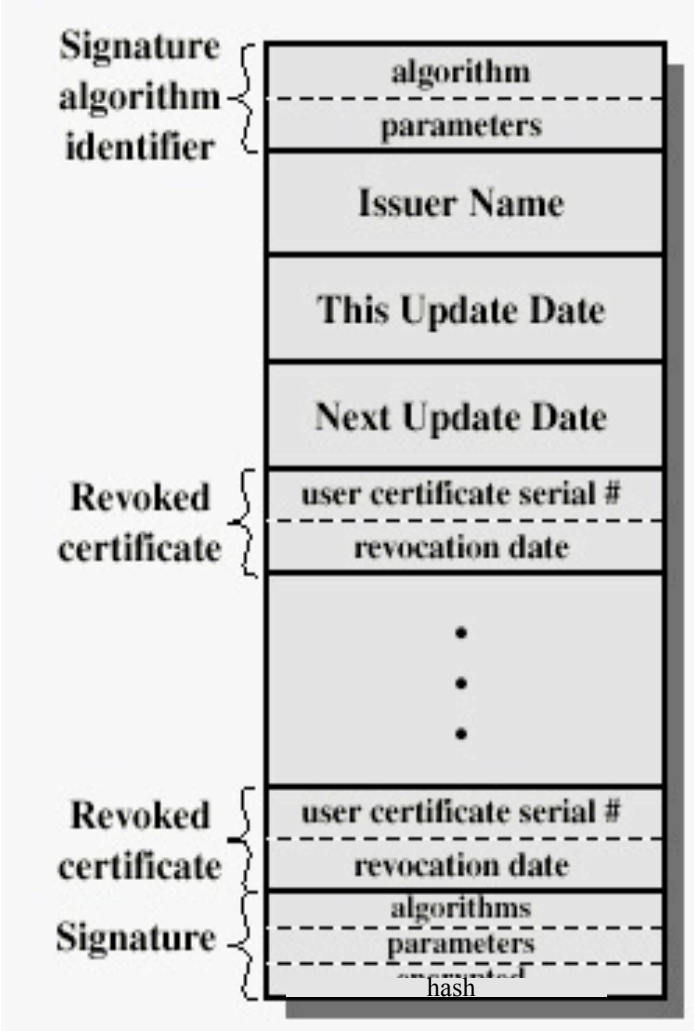
Certificate Revocation

- Revocation is very important
- Many valid reasons to revoke a certificate
  - Private key corresponding to the certified public key has been compromised
  - User stopped paying his certification fee to this CA and CA no longer wishes to certify him
  - CA's private key has been compromised!
- Expiration is a form of revocation, too
  - Many deployed systems don't bother with revocation
  - Re-issuance of certificates is a big revenue source for certificate authorities

Certificate Revocation Mechanisms

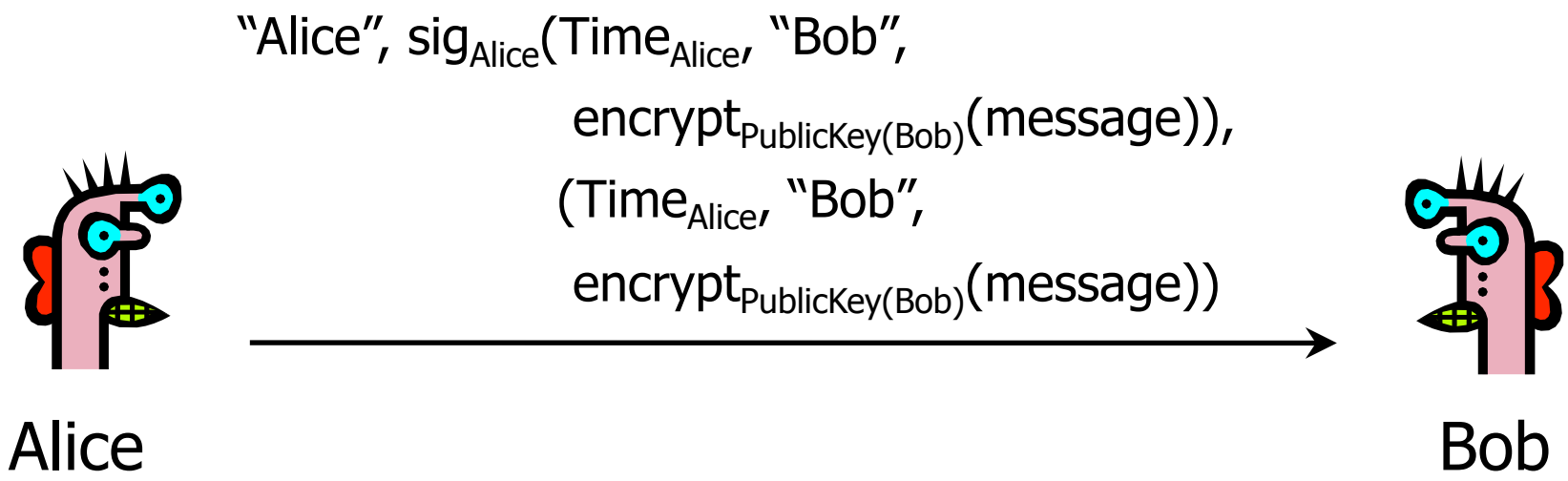
- Online revocation service
  - When a certificate is presented, recipient goes to a special online service to verify whether it is still valid
    - Like a merchant dialing up the credit card processor
- Certificate revocation list (CRL)
  - CA periodically issues a signed list of revoked certificates
    - Credit card companies used to issue thick books of canceled credit card numbers
  - Can issue a "delta CRL" containing only updates

X.509 Certificate Revocation List



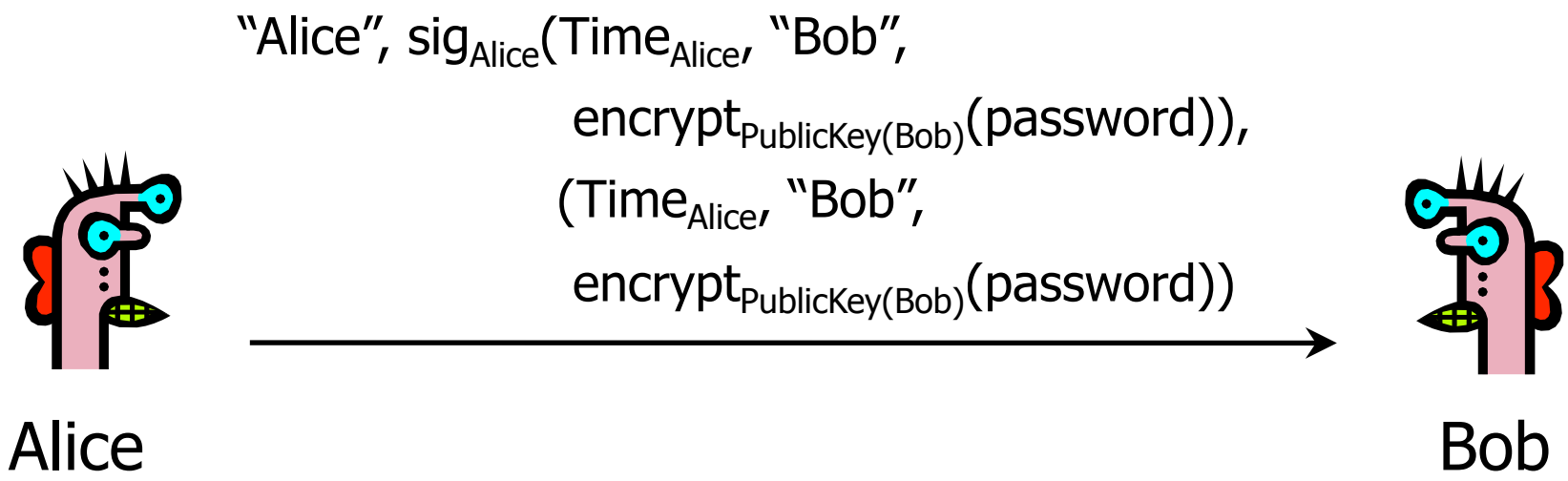
Because certificate serial numbers must be unique within each CA, this is enough to identify the certificate

X.509 Version 1



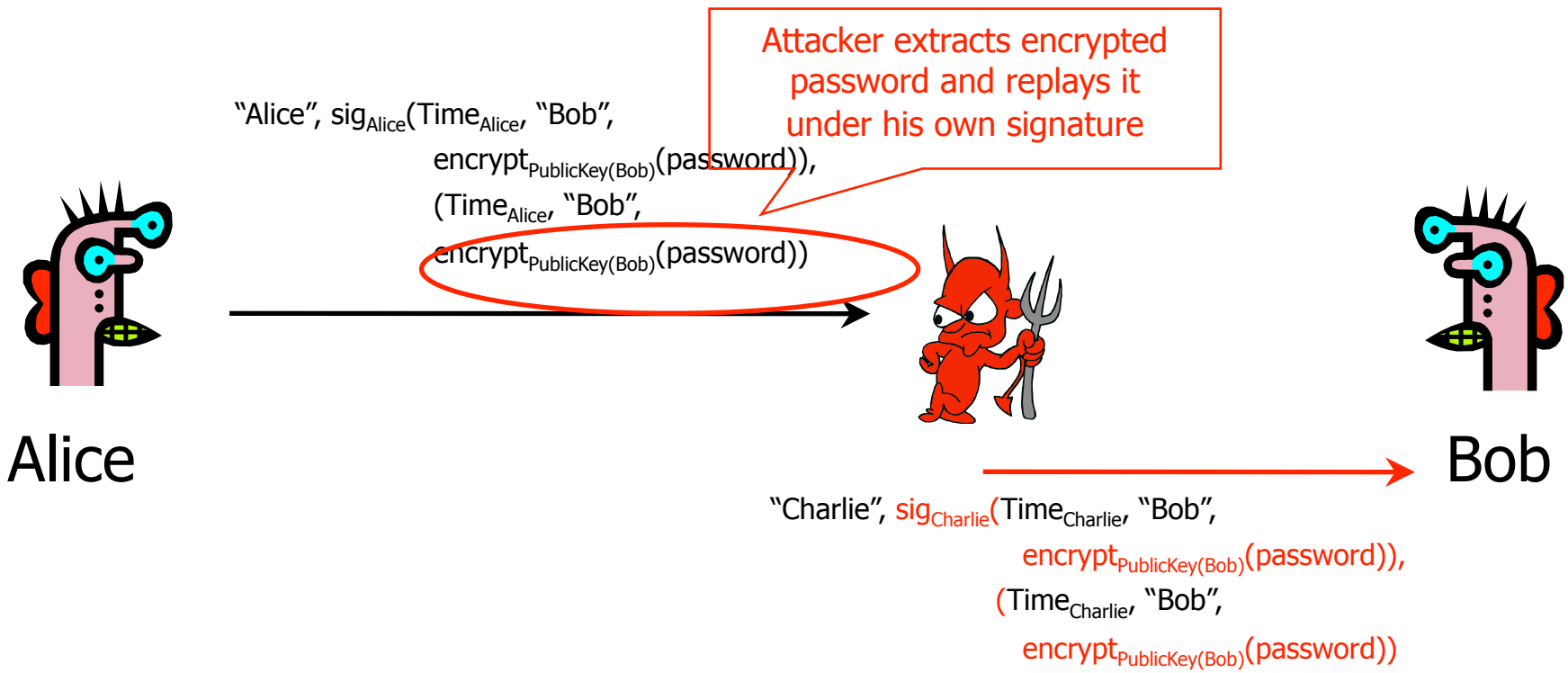
- Encrypt, then sign
  - Goal: achieve both confidentiality and authentication
  - E.g., encrypted, signed password for access control (for next slide: assume one password for whole system)
- Does this work?

X.509 Version 1 (message is passwd)



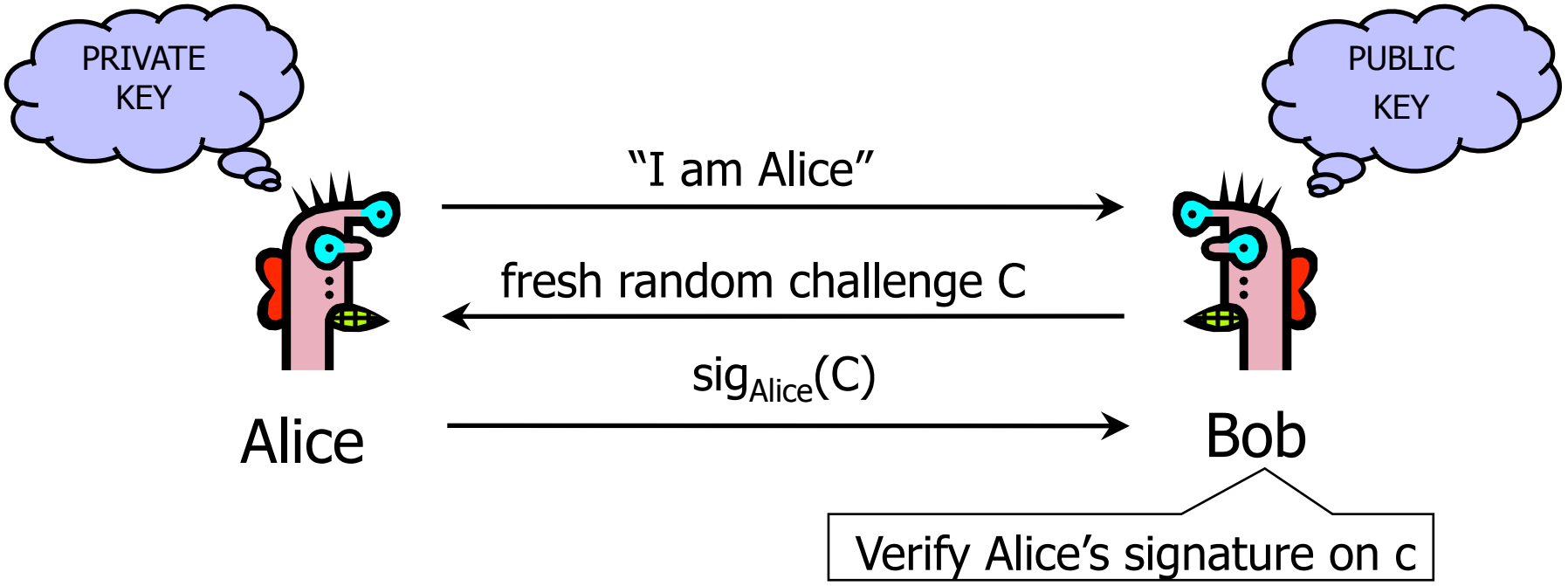
- Encrypt, then sign
  - Goal: achieve both confidentiality and authentication
  - E.g., encrypted, signed password for access control (for next slide: assume one password for whole system)
- Does this work?

Attack on X.509 Version 1



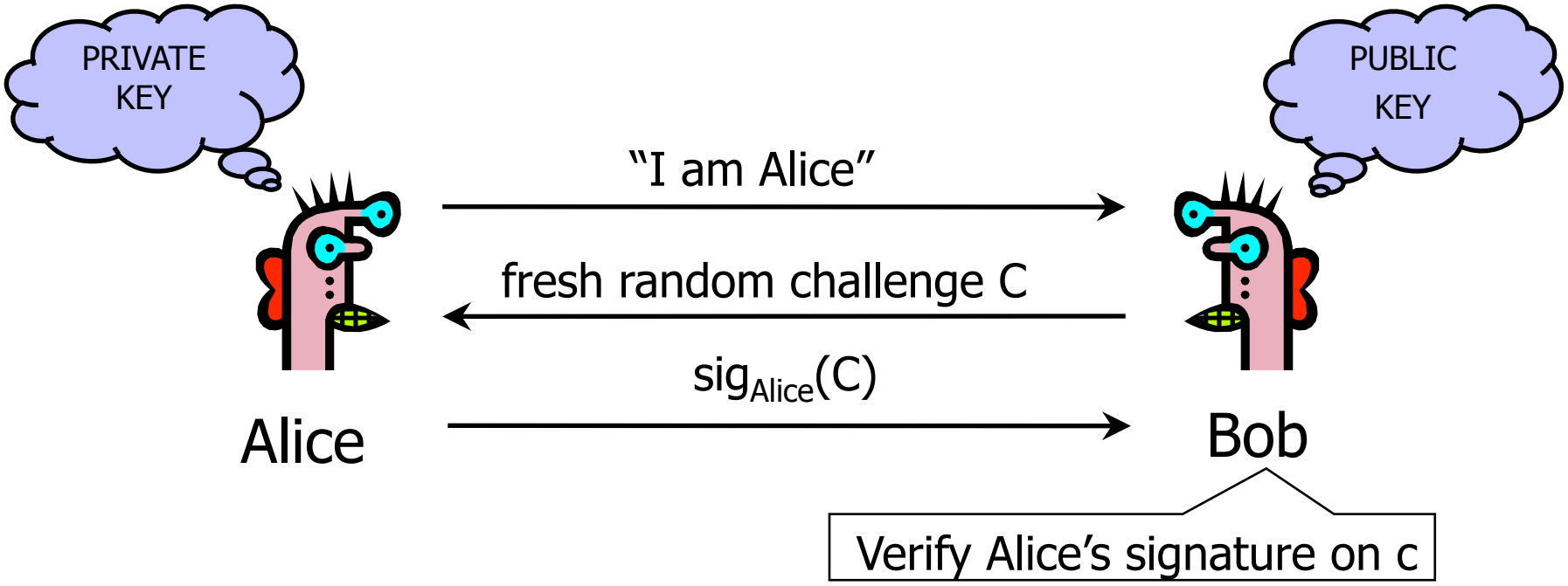
- Receiving encrypted password under signature does not mean that the sender actually knows the password!

Authentication with Public Keys



1. Only Alice can create a valid signature
2. Signature is on a fresh, unpredictable challenge

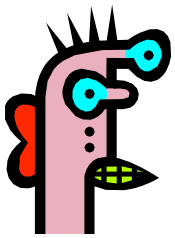
Authentication with Public Keys



1. Only Alice can create a valid signature
2. Signature is on a fresh, unpredictable challenge

Potential problem: Alice will sign anything

Mafia-in-the-Middle Attack [from Anderson's book]

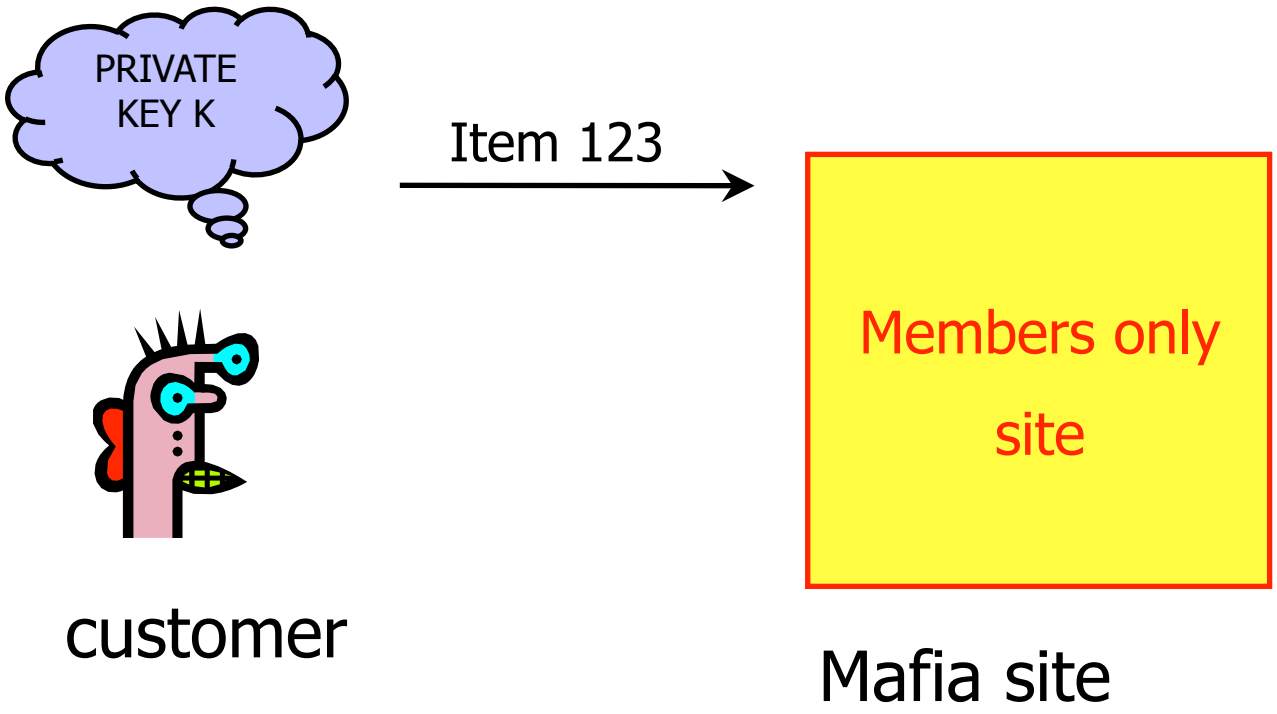


customer

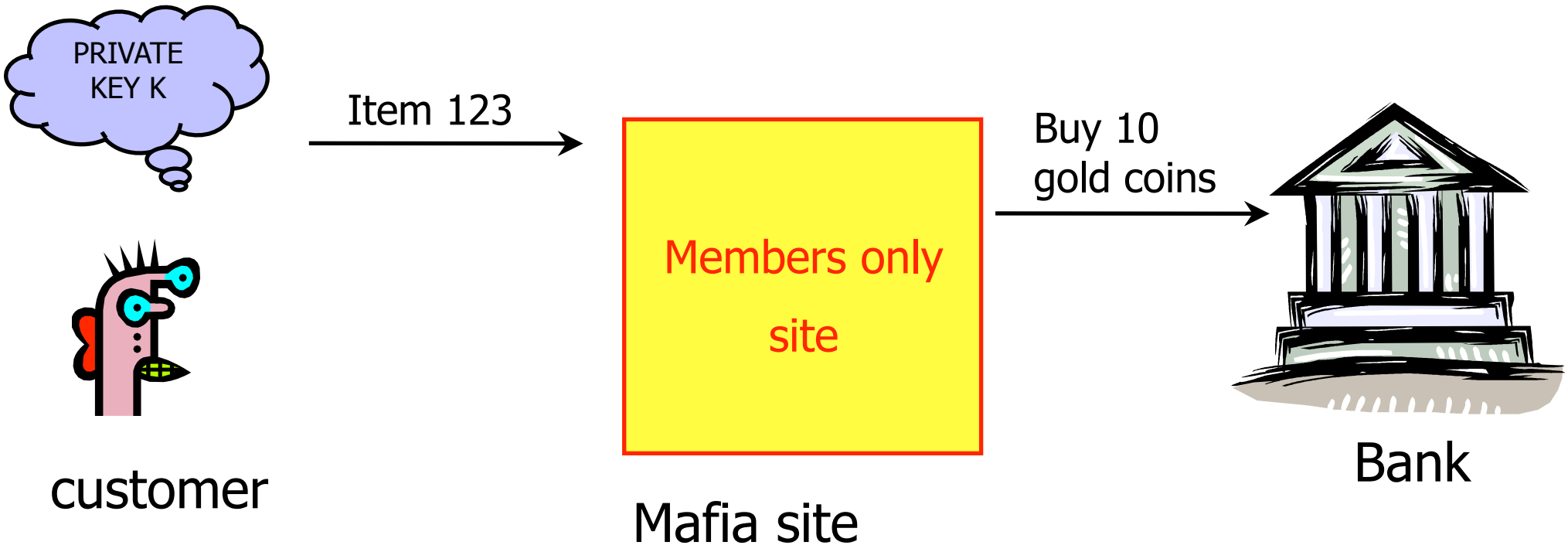
Members only site

Mafia site

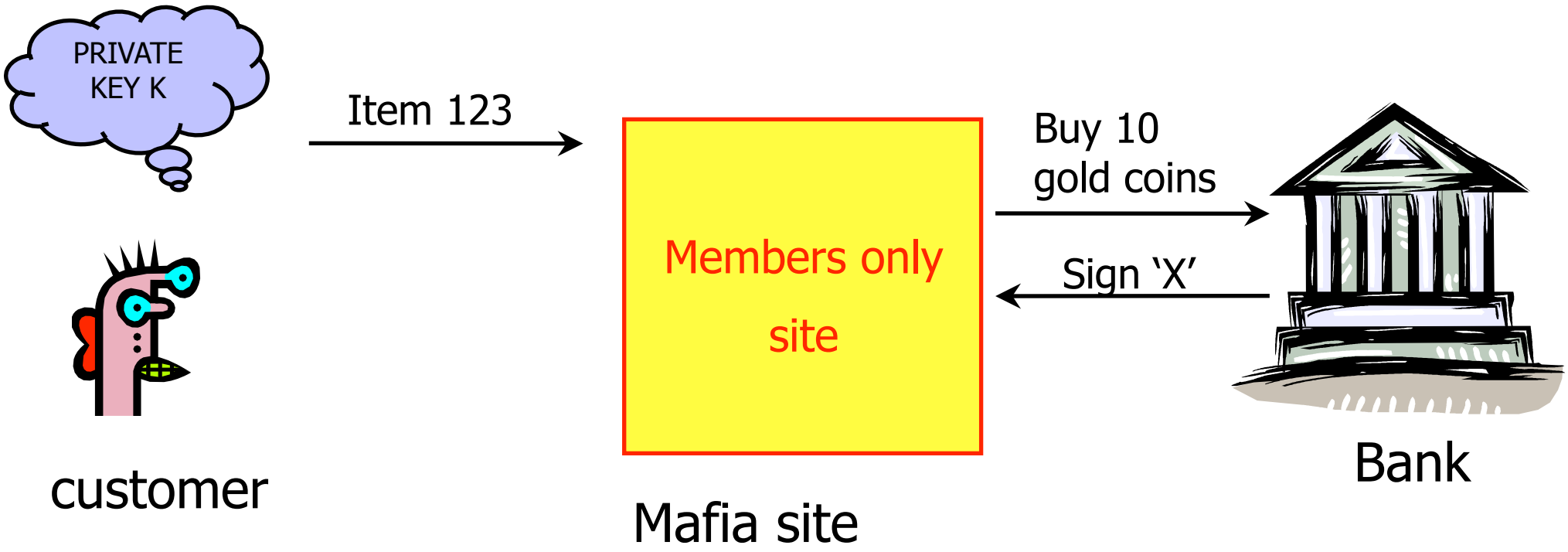
Mafia-in-the-Middle Attack [from Anderson's book]



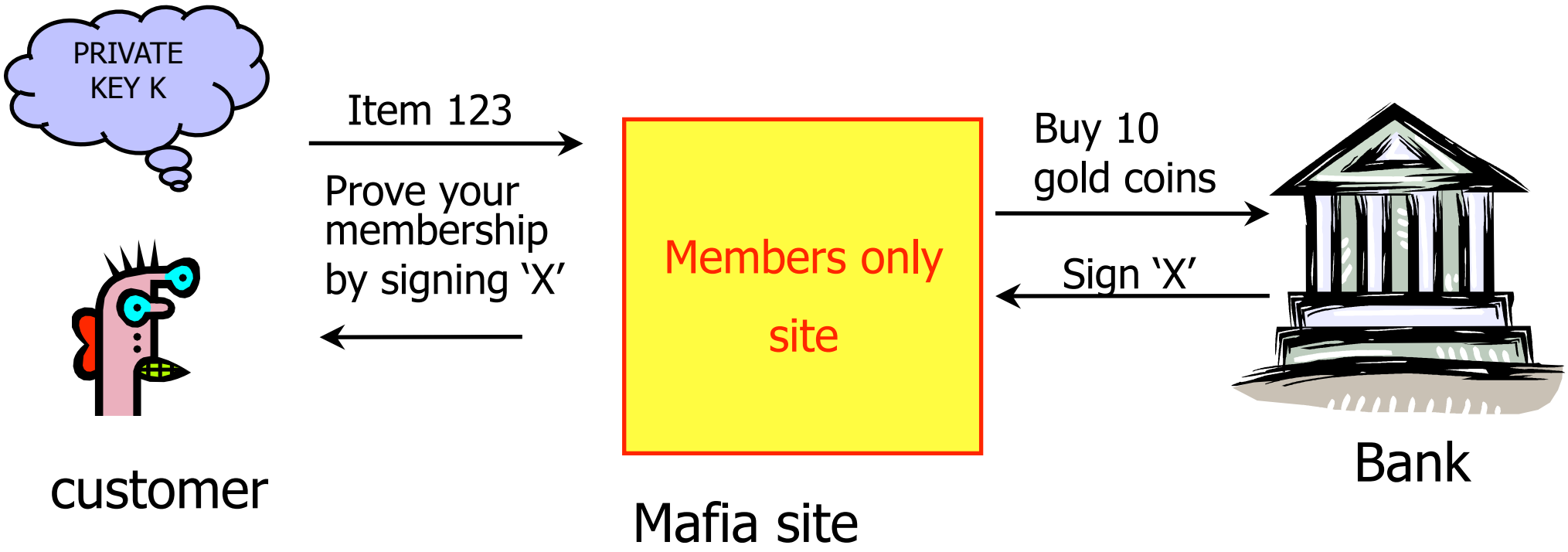
Mafia-in-the-Middle Attack [from Anderson's book]



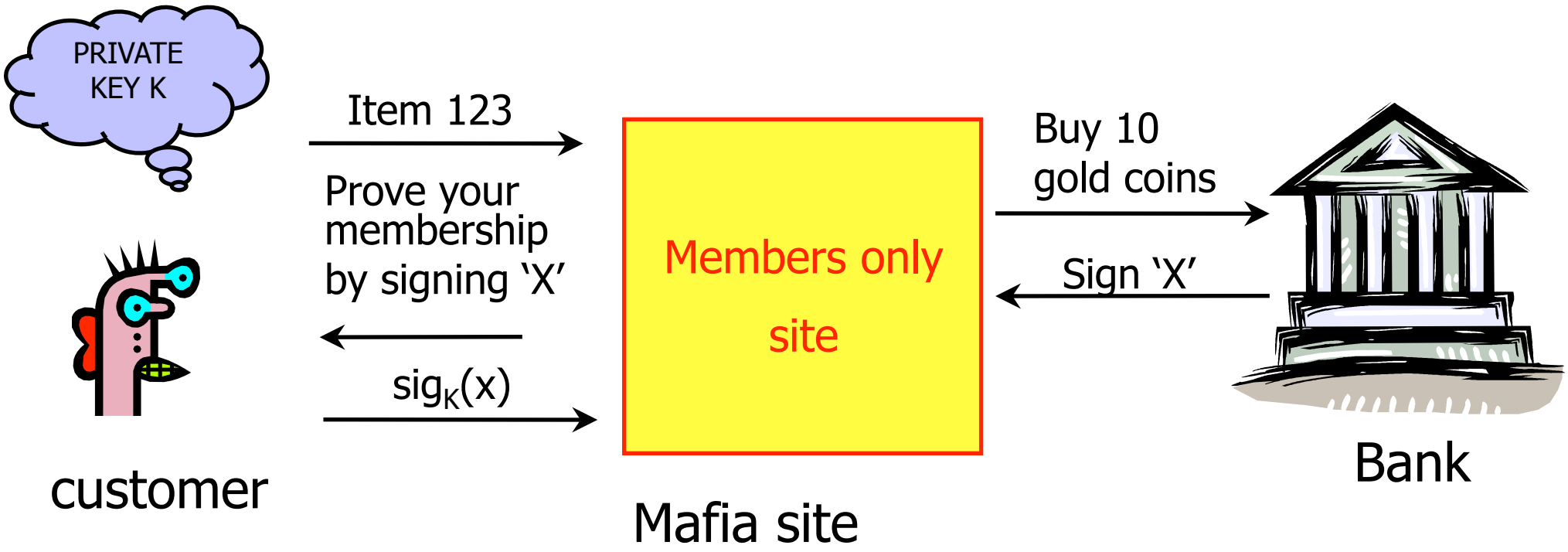
Mafia-in-the-Middle Attack [from Anderson's book]



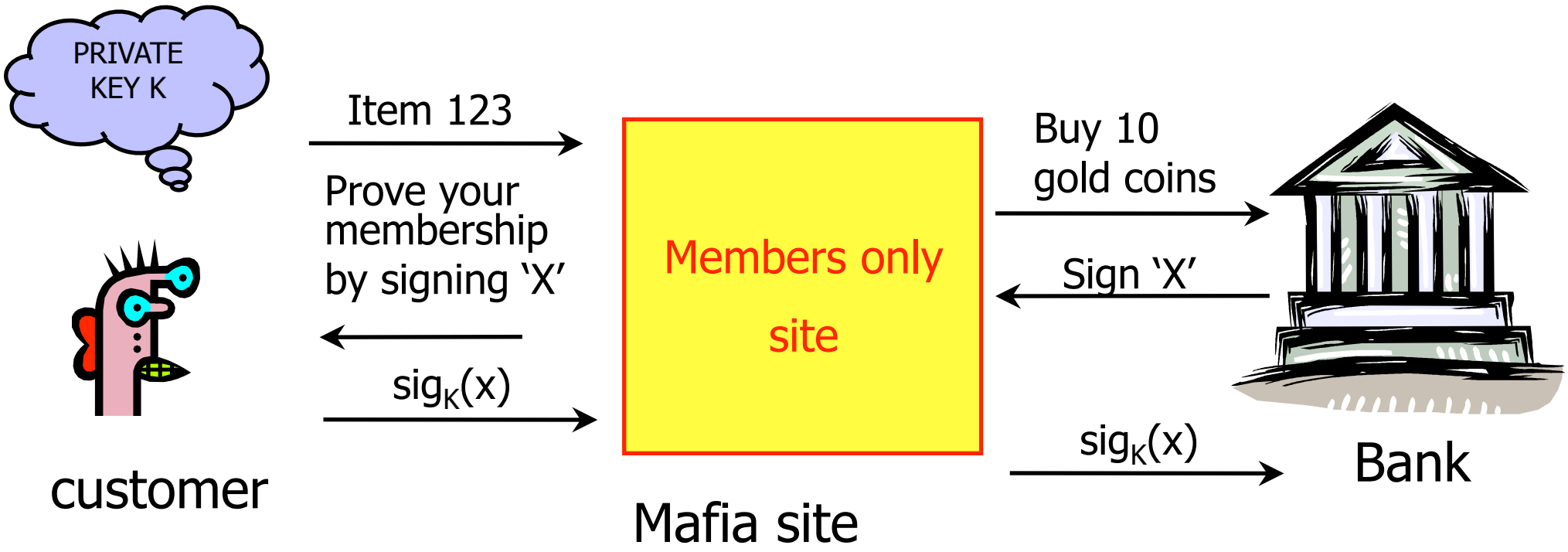
Mafia-in-the-Middle Attack [from Anderson's book]



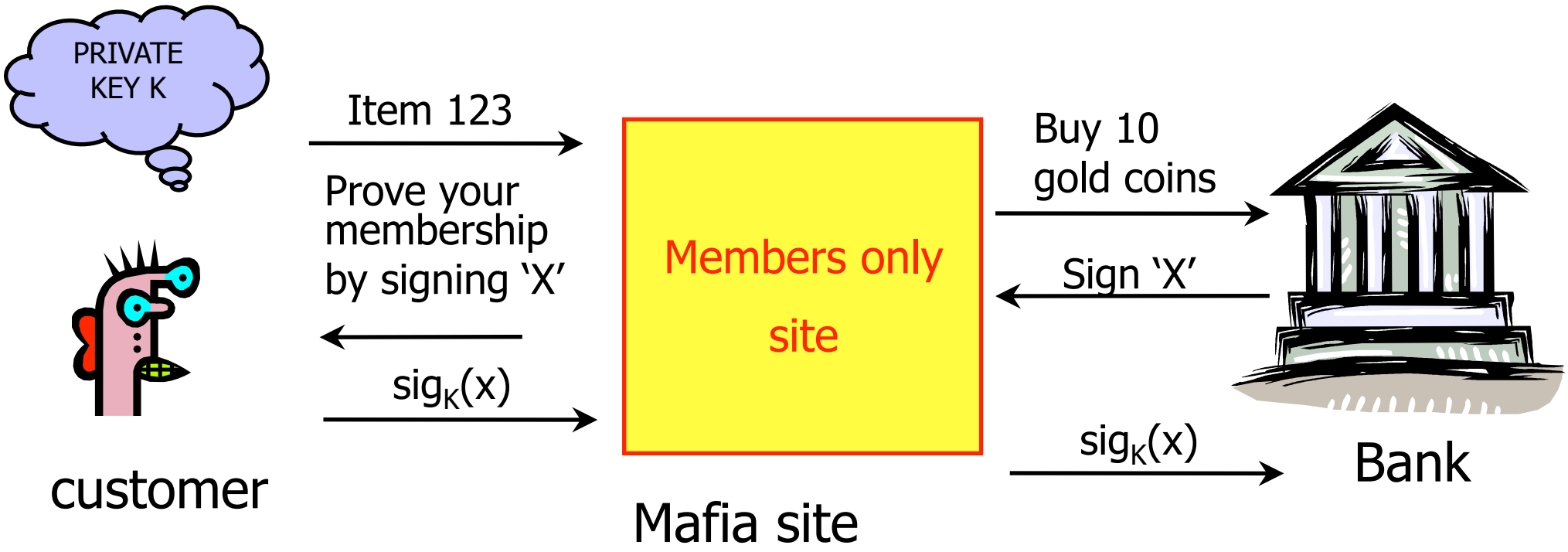
Mafia-in-the-Middle Attack [from Anderson's book]



Mafia-in-the-Middle Attack [from Anderson's book]



Mafia-in-the-Middle Attack [from Anderson's book]



One key recommendation: Don't use same public key / secret key pair for multiple applications. (Or make sure messages have different formats across applications.)

Secure Sessions

- Secure sessions are among the most important applications in network security
  - Enable us to talk securely on an insecure network
- Goal: secure bi-directional communication channel between two parties
  - The channel must provide confidentiality
    - Third party cannot read messages on the channel
  - The channel must provide authentication
    - Each party must be sure who the other party is
  - Other desirable properties: integrity, protection against denial of service, anonymity against eavesdroppers

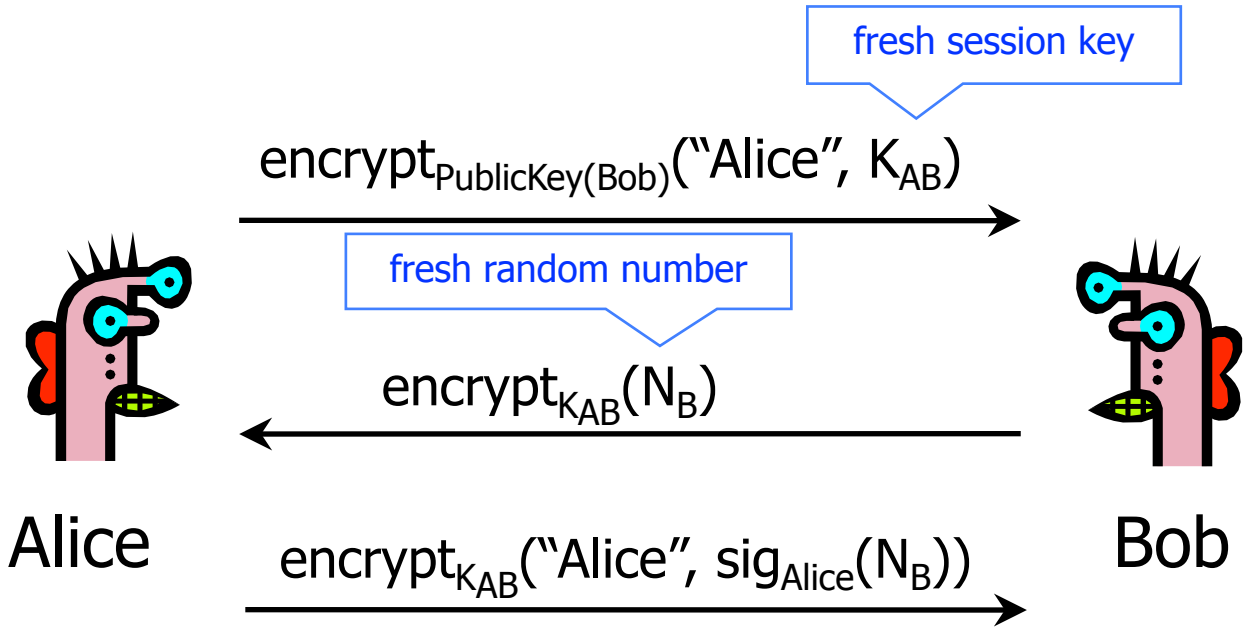
Key Establishment Protocols

- Common implementation of secure sessions:
  - Establish a secret key known only to two parties
  - Then use block ciphers for confidentiality, HMAC for authentication, and so on
- Challenge: how to establish a secret key
  - Using only public information?
  - Even if the two parties share a long-term secret, a fresh key should be created for each session
    - Long-term secrets are valuable; want to use them as sparingly as possible to limit exposure and the damage if the key is compromised
    - (Background: For N parties, there are N choose 2 = N*(N-1)/2 pairs of parties.)

Key Establishment Techniques

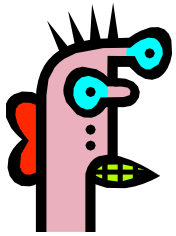
- Use a trusted key distribution center (KDC)
  - Every party shares a pairwise secret key with KDC
  - KDC creates a new random session key and then distributes it, encrypted under the pairwise keys
    - Example: Kerberos
- Use public-key cryptography
  - Diffie-Hellman authenticated with signatures
    - Example: IKE (Internet Key Exchange)
  - One party creates a random key, sends it encrypted under the other party's public key
    - Example: TLS (Transport Layer Security)

Early Version of SSL (Simplified)

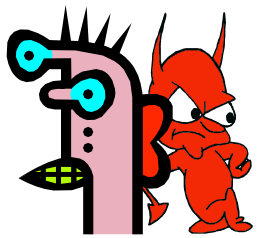


- Bob's reasoning: I must be talking to Alice because…
  - Whoever signed N_B knows Alice's private key… Only Alice knows her private key… Alice must have signed N_B… N_B is fresh and random and I sent it encrypted under K_{AB}… Alice could have learned N_B only if she knows K_{AB}… She must be the person who sent me K_{AB} in the first message…

Breaking Early SSL



Alice

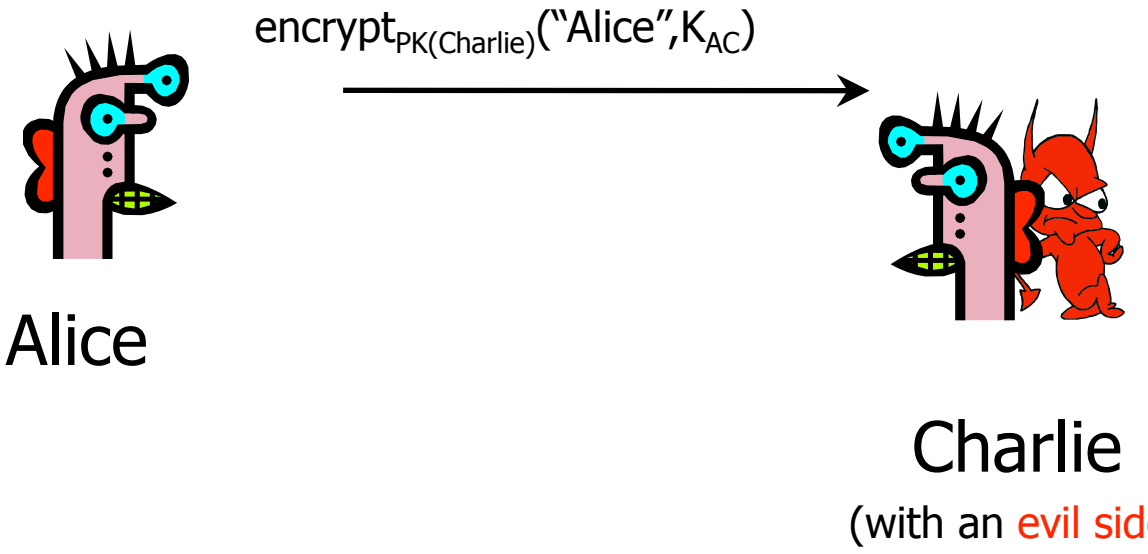


Charlie

(with an evil side)

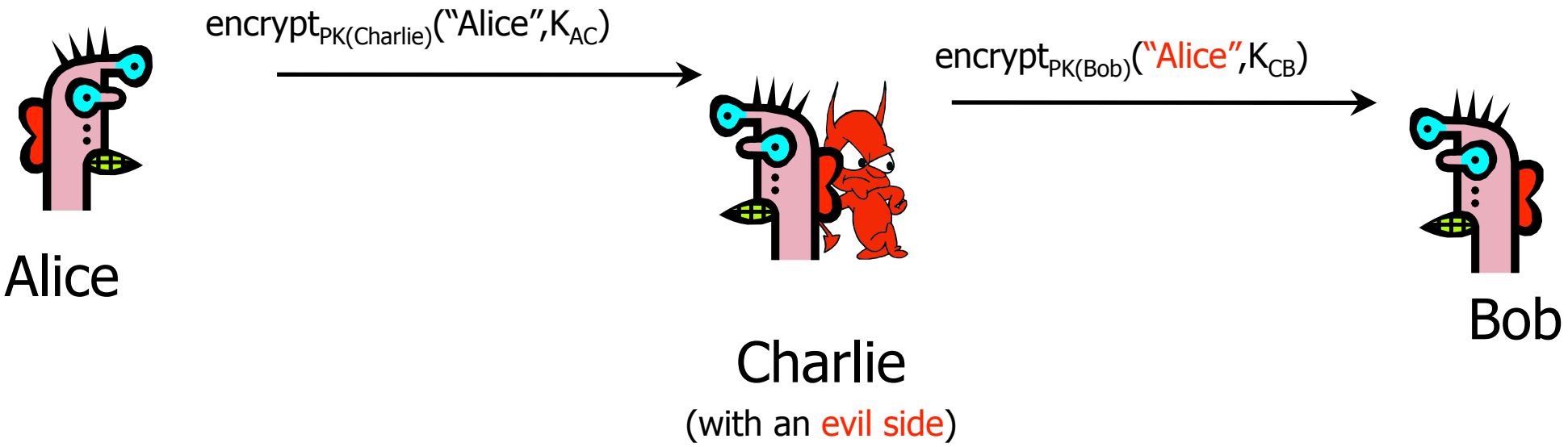
- Charlie uses his legitimate conversation with Alice to impersonate Alice to Bob
  - Information signed by Alice is not sufficiently explicit

Breaking Early SSL



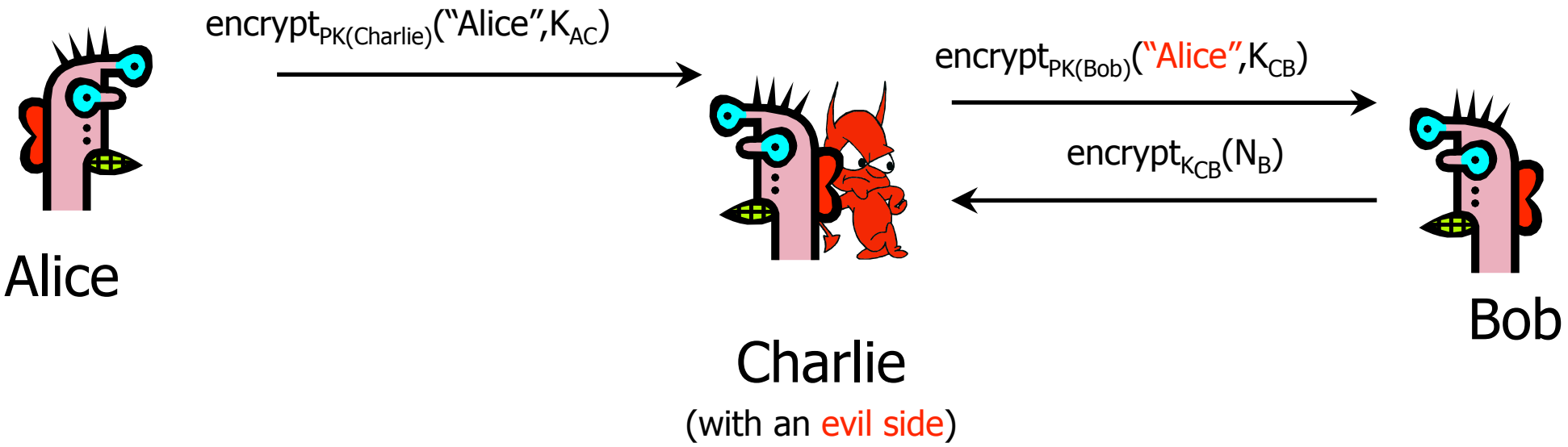
- Charlie uses his legitimate conversation with Alice to impersonate Alice to Bob
  - Information signed by Alice is not sufficiently explicit

Breaking Early SSL



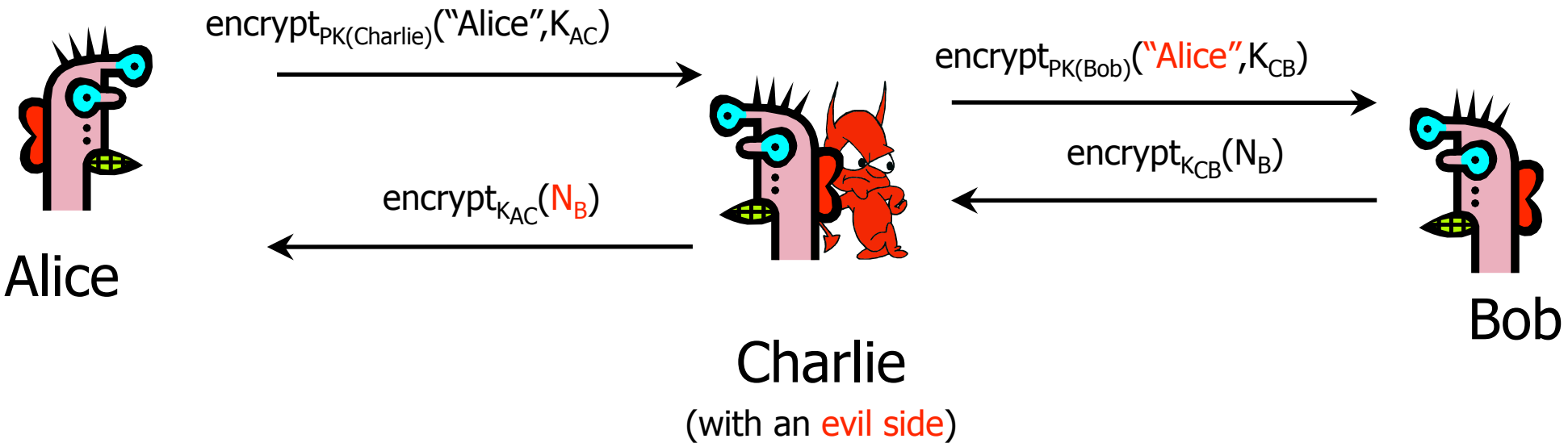
- Charlie uses his legitimate conversation with Alice to impersonate Alice to Bob
  - Information signed by Alice is not sufficiently explicit

Breaking Early SSL



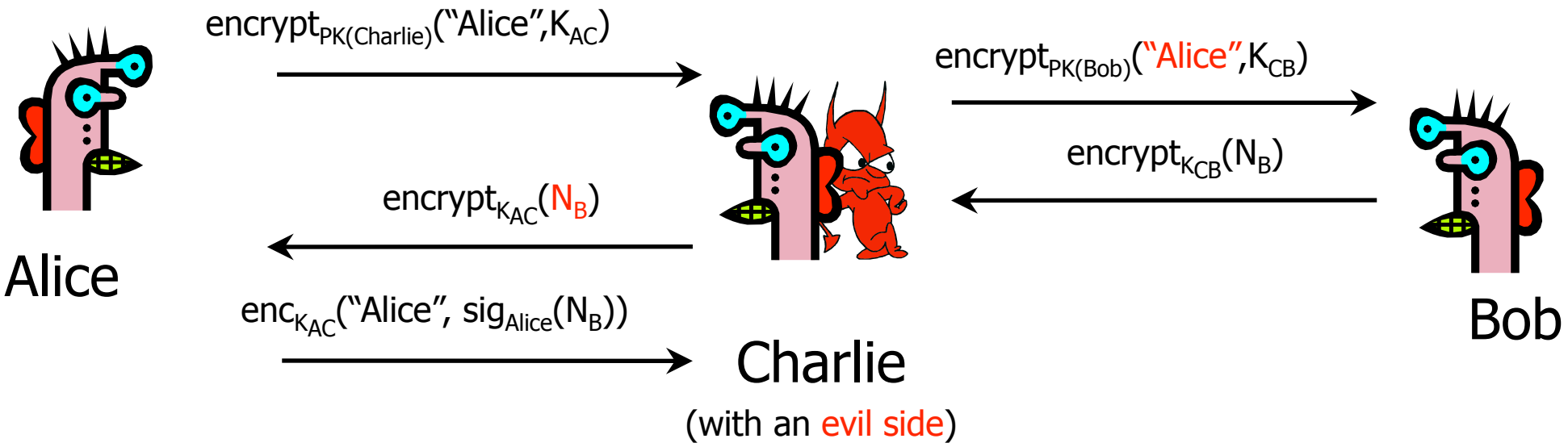
- Charlie uses his legitimate conversation with Alice to impersonate Alice to Bob
  - Information signed by Alice is not sufficiently explicit

Breaking Early SSL



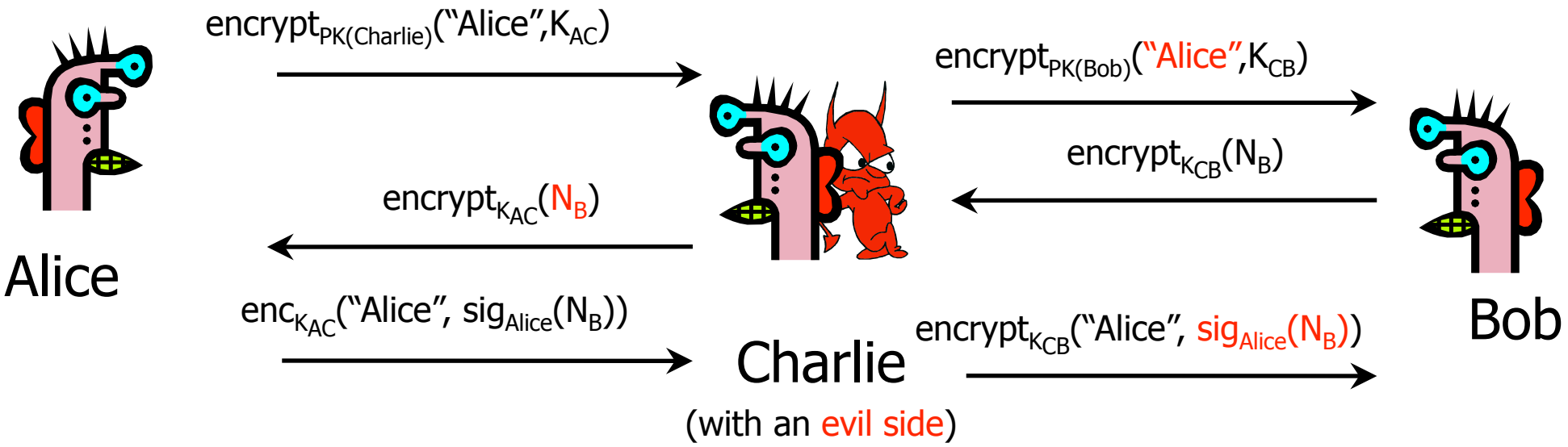
- Charlie uses his legitimate conversation with Alice to impersonate Alice to Bob
  - Information signed by Alice is not sufficiently explicit

Breaking Early SSL



- Charlie uses his legitimate conversation with Alice to impersonate Alice to Bob
  - Information signed by Alice is not sufficiently explicit

Breaking Early SSL



- Charlie uses his legitimate conversation with Alice to impersonate Alice to Bob
  - Information signed by Alice is not sufficiently explicit