CS 378 - Network Security and Privacy
Spring 2005

FINAL

May 16, 2005

DO NOT OPEN UNTIL INSTRUCTED

YOUR NAME: ______________________________

Collaboration policy

No collaboration is permitted on this exam. Any cheating (*e.g.*, submitting another person's work as your own, or permitting your work to be copied) will automatically result in a failing grade. The Computer Sciences department code of conduct can be found at `http://www.cs.utexas.edu/users/ear/CodeOfConduct.html`

Final exam (125 points)

Problem 1: “Beauty is truth, truth beauty” (21 points)

Circle only one of the choices **(3 points each)**.

1. **TRUE** **FALSE** “Salting” makes the known-plaintext attack on password files more difficult.
2. **TRUE** **FALSE** To take advantage of the authentication and confidentiality services provided by IPSec, applications such as Web browsers and FTP must be modified accordingly.
3. **TRUE** **FALSE** Circuit-level firewall gateways are stateless and, therefore, must allow all incoming packets to high-numbered ports.
4. **TRUE** **FALSE** Random scanning is used by Internet worms because it is the optimal way to find IP addresses of vulnerable hosts.
5. **TRUE** **FALSE** Stepping stone detection can help identify zombie computers used in distributed denial of service attacks.
6. **TRUE** **FALSE** Bytecode verification and runtime checking in the Java Virtual Machine prevent all buffer overflow attacks on Java code.

Problem 2: “A good name is better than riches” (9 points)

Denning and Sacco proposed the following protocol, which enables Alice and Bob to establish a shared symmetric key K with the help of a trusted public-key certificate directory S. Alice contacts the directory and obtains the certificates for her own public key $pk(A)$ and Bob’s public key $pk(B)$. Alice then generates a fresh random key K, signs it together with the current timestamp, and sends the result to Bob, encrypted with Bob’s public key and accompanied by the two certificates. Bob decrypts the message and verifies Alice’s signature using the public key contained in Alice’s certificate. If verification succeeds and the timestamp is recent, Bob concludes that he now shares key K with Alice.

The protocol is summarized below:

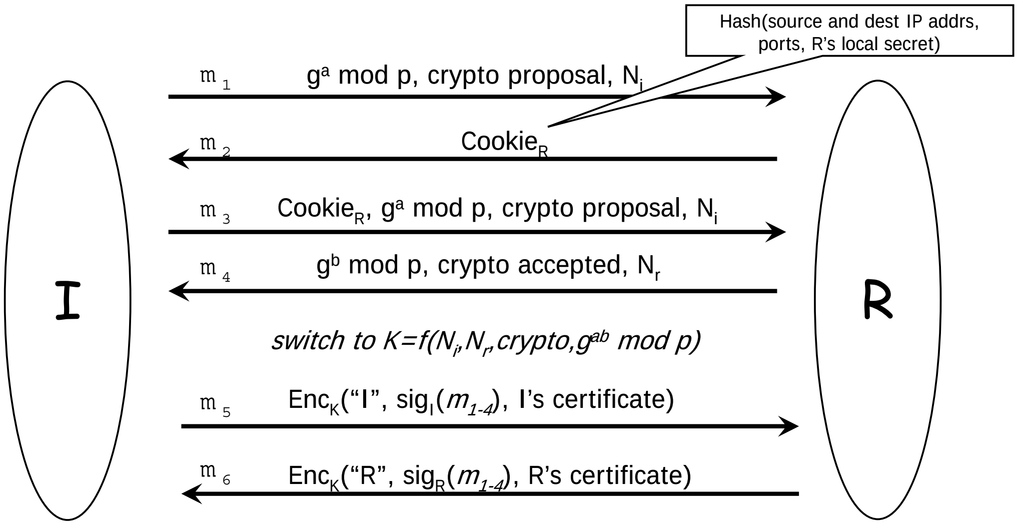
$$\begin{array}{llll}
\textit{Alice} & \rightarrow & \textit{Directory} & A, B \\
\textit{Directory} & \rightarrow & \textit{Alice} & cert_A, cert_B \\
\textit{Alice} & \rightarrow & \textit{Bob} & cert_A, cert_B, encrypt_{pk(B)}(sig_{pk(A)}(K, time_A))
\end{array}$$

You can assume that public-key certificates are signed by a trusted certificate authority, and that Alice's and Bob's clocks are synchronized.

How can this protocol be used to impersonate Alice to a third party, Charlie?

Problem 3: "We think of the key, each in his prison"

Below is the (slightly simplified) version of the Internet Key Exchange (IKE) protocol:


Hash(source and dest IP addrs, ports, R's local secret)
m 1
g^a mod p, crypto proposal, N_i
m 2
Cookie_R
m 3
Cookie_R, g^a mod p, crypto proposal, N_i
m 4
g^b mod p, crypto accepted, N_r
I
R
switch to K=f(N_i,N_r,crypto,g^ab mod p)
m 5
Enc_K("I", sig_I(m_1-4), I's certificate)
m 6
Enc_K("R", sig_R(m_1-4), R's certificate)

Problem 3a (5 points)

Is it possible to stage a denial of service attack on the IKE responder (R in the picture) by replaying old connection requests? If yes, describe how the attack results in the denial of service to honest clients. If no, describe the defense.

Problem 3b (5 points)

What is the purpose of encrypting message m_6?

Problem 4: "All hope abandon, ye who enter here"

One idea used in virus checkers is to loop through all the entry points to a file, and look for possible viruses in the next sequence of bytes.

Problem 4a (5 points)

Why would a virus writer design virus code to place infections just after program entry points?

Problem 4b (5 points)

How would a virus writer circumvent virus checkers that only look at file entry points?

Problem 5: "Listen to all sides and filter them from your self"

Problem 5a (5 points)

What information is used by a packet-filtering firewall when deciding whether to allow a packet to enter the protected network?

Problem 5b (10 points)

List at least **three** weaknesses of any (properly configured) packet-filtering firewall.

Problem 6: “But see, amid the mimic route a crawling shape intrude!” (15 points)

Suppose that every packet observed by a network-based intrusion detection system (NIDS) belongs to one of the following mutually exclusive categories: legitimate (88% of all traffic), known worm (4%), distributed denial of service (4%) or port scan (4%).

The NIDS correctly classifies all known-worm packets. A legitimate packet is classified as legitimate with probability 91%, and misclassified as belonging to any of the three attack categories with equal probability. A DDoS packet is classified as DDoS with probability 50%, as a known worm with probability 40%, and as a legitimate packet with probability 10%. A port-scan packet is classified correctly with probability 85%, and misclassified as a legitimate packet with probability 15%.

If the NIDS announces that a particular packet belongs to a known worm, what is the probability that this packet is **not** a legitimate packet? Show your calculations.

Problem 7: “Such protection as vultures give to lambs”

What defenses are built into the SSL/TLS handshake protocol against the following threats? For each threat, describe specifically **which part of SSL** provides protection. Do not make any assumptions about the specific encryption or signature scheme used by SSL (SSL is supposed to be compatible with multiple schemes).

Problem 7a (5 points)

Protection against replay attacks:

Problem 7b (5 points)

Protection against man-in-the-middle attacks:

Problem 7c (5 points)

Protection against known-plaintext attacks:

Problem 8: “More is thy due than more than all can pay”

Remember Peppercoin? In this problem, we look at another micropayment scheme, called Payword, that aims to provide an easy way for merchants to **aggregate** very small payments. For simplicity, suppose that there is only one Merchant in the system. Anonymity is not a concern in this scheme.

Payword works as follows. Every morning, the User picks a random number r and generates a “chain” of **paywords** $w_0, \ldots, w_n$ so that $w_n = r$, $w_i = H(w_{i+1})$ for i between 0 and $n-1$, where H is a cryptographically strong hash function. Assume that n is sufficiently large. The User presents w_0 to the Bank, who signs it together with the User’s identity, producing $S = sig_{bank}(U, w_0)$.

Let k be the current index of the payword chain, maintained internally by the User (in the beginning, $k = 0$). To pay 1 cent to the Merchant, the User sends him the triple $\langle S, w_{k+1}, k+1 \rangle$, and increments the internal counter k. To reduce computational overhead, the User’s message to the Merchant is not signed.

At the end of the day, the Merchant sends to the Bank the latest triple (S, w_l, l) received from the User. The Bank transfers l cents from the User’s account to the Merchant’s account.

Problem 8a (5 points)

How does the User make a payment of m cents (rather than 1 cent) with a **single** triple $\langle S, \ldots, \ldots \rangle$?

Problem 8b (5 points)

How does the Bank verify that the l value sent by the Merchant had indeed been received from the User? Recall that the User’s messages to the Merchant are **not** signed.

Problem 8c (5 points)

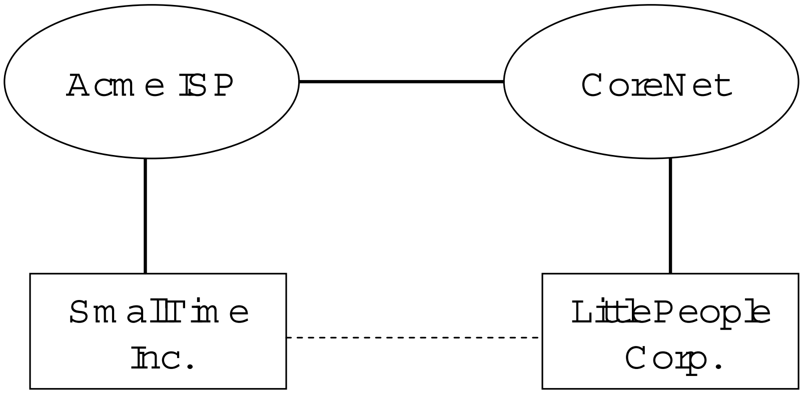
Which property of the hash function H prevents the Merchant from getting more money from the Bank than the User authorized? Explain.

Problem 9: "All, soon or late, are doom'd that path to tread"

Business relationships often determine routing policies. On the Internet, there are usually two types of relationships: *customer-provider* relationships and *peer-peer* relationships. In a customer-provider relationship, one autonomous system (AS), the *customer*, purchases service from another AS, the *provider*. In a peer-peer relationship, two ASs form a mutual agreement to carry each other's traffic and share routes, often providing shortcuts to longer routes that would have to go through the provider.

These relationships affect BGP routing policies as follows. An AS advertises all of its routes to its customers and advertises routes through its customers to all its neighbors. On the other hand, an AS does **not** advertise routes through a peer or a provider to any neighbor other than a customer. In addition, routes through customers are preferred first, then routes through peers, and finally routes through providers.

In the graph below, assume that Acme and CoreNet are peers, that SmallTime is a customer of Acme, and that LittlePeople is a customer of CoreNet.



Problem 9a (5 points)

Assume that the link between SmallTime and LittlePeople is not present, that LittlePeople is advertising itself to neighbors, and so on. What are the AS paths from **each** AS to LittlePeople?

Problem 9b (5 points)

Now suppose that LittlePeople is nervous about its link to CoreNet and decides to purchase a link from SmallTime, *i.e.*, it becomes a customer of SmallTime. However, it tells SmallTime (through a special agreement) that this link is a **backup** link and is to be given lower preference than any other path SmallTime may have to LittlePeople. Thus there is no change to the AS paths from the previous question, because this is taken into account when SmallTime chooses and advertises the "best" route to LittlePeople.

When a link fails, the corresponding route is withdrawn and should be replaced by another one. Suppose the link between CoreNet and LittlePeople fails, and CoreNet withdraws its route to LittlePeople. SmallTime has a backup link that it will advertise as long as it is not aware of any other route to LittlePeople. Given this change, what are the new AS paths from **each** AS to LittlePeople (once the routes stabilize)?

Problem 9c (5 points)

Suppose that the link between CoreNet and LittlePeople is restored. Keeping in mind the preference rules for business relationships, what are the AS paths from **each** AS to LittlePeople?

Bonus problem (0 points)

Match names to problem titles:

Dante Alighieri ______
T.S. Eliot ______
John Keats ______
Edgar Allan Poe ______
Alexander Pope ______
Miguel de Cervantes Saavedra ______
William Shakespeare ______
Richard Brinsley Sheridan ______
Walt Whitman ______