YALE UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

CPSC 467b: Cryptography and Computer Security | *Handout #20*
Zheng Ma
April 18, 2005

Solutions to Problem Set 5

Each problem counts 10 points

Problem 15: Intergral Broadcasting

Suppose user A is broadcasting packets to n recipients $B_1, ..., B_n$. Privacy is not important but integrity is. In other words, each of $B_1, ..., B_n$ should be assured that the packets he is receiving were sent by A. User A decides to use a MAC.

(a) Suppose user A and $B_1, ..., B_n$ all share a secret key k. User A MAC's every packet she sends using k. Each user B_i can then verify the MAC. Using at most two sentences explain why this scheme is insecure, namely, show that user B_1 is not assured that packets he is receiving are from A.

(b) Suppose user A has a set $S = \{k_1, ..., k_m\}$ of m secret keys. Each user B_i has some subset $S_i \subseteq S$ of the keys. When A transmits a packet she appends m MAC's to it by MACing the packet with each of her m keys. When user B_i receives a packet he accepts it as valid only if all MAC's corresponding to keys in S_i are valid. What property should the sets $S_1, ..., S_n$ satisfy so that the attack from part (a) does not apply? We are assuming all users $B_1, ..., B_n$ are succiently far apart so that they cannot collude.

(c) Show that when $n = 6$ (i.e. six recipients) the broadcaster A need only append 4 MAC's to every packet to satisfy the condition of part (b). Describe the sets $S_1, ..., S_6 \subseteq \{k_1, ..., k_4\}$ you would use.

Solution:

(a) Since A and all $B_i \in B$ share the secret key k, any B_i can send to the parties $B \setminus \{B_i\}$ a message M appended with the MAC under k of M. To the recipients this looks exactly like a message sent by A, and hence, the recipient is not sure the message came from A.

(b) B_i can successfully fool $B_j (i \neq j)$ iff B_i has every key that B_j has. This is easy to see since B_j verifies a message by verifying the MACs corresponding to the keys he has. Thus, for the scheme to work, each pair $B_i, B_j (i \neq j)$ must have at least one key not shared between them. Then, no B_i can fool a B_j because B_i will lack one of the keys that B_j uses to verify the message.

Note that we use the assumption of non-collusion to ensure that if some proper subset of parties $\hat{B} \in B$ have between them every key, they cannot work together to fool the parties in $B \setminus \hat{B}$.

(c) Let the keys be k_1, k_2, k_3, k_4. We note that $6 = \mathrm{C}_4^2$. Hence each S_i need only contain two keys. The subsets are:

$$S_1 : \{k_1, k_2\}$$
$$S_2 : \{k_1, k_3\}$$

$$S_3 : \{k_1, k_4\}$$
$$S_4 : \{k_2, k_3\}$$
$$S_5 : \{k_2, k_4\}$$
$$S_6 : \{k_3, k_4\}$$

Note that for any two S_i, S_j $(i \neq j)$, S_i and S_j differ in at least one element.

Problem 16: Combining Signatures and Encryption

Let (S_A, V_A) be Alice's digital signature scheme, and let (E_B, D_B) be Bob's public key encryption scheme. Alice wants to send a private signed message m to Bob. She thinks of several possible ways to proceed:

i. Encrypted signed message: Alice sends $E_B(\langle m, S_A(m)\rangle)$ to Bob.

ii. Signed encrypted message: Alice sends $\langle E_B(m), S_A(E_B(m))\rangle$ to Bob.

iii. Hybrid scheme: Alice sends $\langle E_B(m), S_A(m)\rangle$ to Bob.

(The notation $\langle x, y\rangle$ denotes the ordered pair (x, y), suitably encoded as a string.)

(a) For each scheme, describe how Bob decodes the message and verifies the signature.

(b) Alice comes to you for a recommendation of which scheme to use. Your job is to write a brief report giving your best professional advice to her. You should consider in your report any aspects that you feel would be important in practice, e.g., overall security and reliability of each scheme, possibility of known or unanticipated attacks, efficiency of implementation, and so forth.

Solution:

(a) The procedures are straightforward. Note that for scheme (ii), the signature is of the encrypted message; hence, the encrypted message must be given to the signature verification predicate, not the plaintext message.

(b) The signed encrypted scheme (ii) is vulnerable to plagerism, since anyone can remove Alice's signature and put her/his own on instead, even though they can't read the contents of the message.

When choosing between (i) and (iii), speed and secrecy issues have to be balanced. For efficiency of decryption and signature verification, (iii) is faster since the two can be done in parallel. However, (iii) reveals more information about m, and if, in particular, the same public key encryption method is used for both encryption and signatures (a bad idea) so that $S_A(m) = D_A(m)$, then m can be fully recovered because E_A is public. Since now, every computer is very fast, efficiency is not such a key issue, so (i) is the wisest choice.

Problem 17: Strong Collision-Free Hash Functions

Let h be a given strong collision-free hash function that maps bitstrings of length $2n$ to bitstrings of length n. We wish to construct a new one-way hash function that maps bitstrings of length $4n$ to bitstrings of length n. Write $x = x_1 \cdot x_2 \cdot x_3 \cdot x_4$, where each x_i has length n. Consider the following candidates:

i. $h_1(x) = h((x_1 \oplus x_2) \cdot (x_3 \oplus x_4))$.

ii. $h_2(x) = h(h(x_1 \cdot x_2) \cdot h(x_3 \cdot x_4))$.

iii. $h_3(x) = h(x_1 \cdot x_2) \oplus h(x_3 \cdot x_4)$.

iv. $h_4(x) = h(h(h(x_1 \cdot x_2) \cdot x_3) \cdot x_4)$.

(Here, "$\oplus$" denotes bitwise exclusive-or and "$\cdot$" denotes concatenation.)

For each function h_i, say whether or not you think it is a strong collision-free hash function. If you think it is, show that the ability to find collisions for h_i would allow one to find collisions for h (contradicting the assumption that h is a strong collision-free hash function). If you think it is not, exhibit a pair of (distinct) colliding words for h_i.

Solution:

(i) h_1 is not strong collision-free because for any x_1, x_2, x_3 and x_4, $h_1(x_1 \cdot x_2 \cdot x_3 \cdot x_4) = h_1(x_2 \cdot x_1 \cdot x_4 \cdot x_3)$. This constitutes a collision if $x_1 \neq x_2$ or $x_3 \neq x_4$. For example, when $n = 1$, (0100, 1000) is a colliding pair for h_1.

(ii) h_2 is a strong collision-free hash function. Suppose not. Then one could find a colliding pair (x, x') for h_2, where $x = x_1 \cdot x_2 \cdot x_3 \cdot x_4$ and $x' = x'_1 \cdot x'_2 \cdot x'_3 \cdot x'_4$. Hence, $x \neq x'$ and $h_2(x) = h_2(x')$.

Let $y = h(x_1 \cdot x_2) \cdot h(x_3 \cdot x_4)$ and $y' = h(x'_1 \cdot x'_2) \cdot h(x'_3 \cdot x'_4)$. If $y \neq y'$, then (y, y') is a colliding pair for h. If $y = y'$, then $h(x_1 \cdot x_2) = h(x'_1 \cdot x'_2)$ and $h(x_3 \cdot x_4) = h(x'_3 \cdot x'_4)$. Hence, either $(x_1 \cdot x_2, x'_1 \cdot x'_2)$ or $(x_3 \cdot x_4, x'_3 \cdot x'_4)$ is a colliding pair for h since $x \neq x'$.

(iii) h_3 is not a strong collision-free hash function since for any $x = x_1 \cdot x_2 \cdot x_3 \cdot x_4$, if $x' = x_3 \cdot x_4 \cdot x_1 \cdot x_2$ and $x' \neq x$, then (x, x') is a colliding pair for h_3. Such x and x' always exist. For example, when $n = 1$, (0011, 1100) is a colliding pair for h_3.

(iv) h_4 is strong collision free. Suppose not. Then one could find a colliding pair (x, x') for h_4, where $x = x_1 \cdot x_2 \cdot x_3 \cdot x_4$ and $x' = x'_1 \cdot x'_2 \cdot x'_3 \cdot x'_4$. Hence, $x \neq x'$ and $h_4(x) = h_4(x')$. Let

$$\begin{array}{ll} y_1 = h(x_1 \cdot x_2) & y'_1 = h(x'_1 \cdot x'_2) \\ y_2 = h(y_1 \cdot x_3) & y'_2 = h(y'_1 \cdot x_3) \\ y_3 = h(y_2 \cdot x_4) & y'_3 = h(y'_2 \cdot x_4) \end{array}$$

We proceed to derive a contradiction. Clearly, $y_3 = h_4(x) = h_4(x') = y'_3$. Because h is strong collision-free, we must have $y_2 = y'_2$ and $x_4 = x'_4$. Similarly, we must have $y_1 = y'_1$ and $x_3 = x'_3$. Applying this same reasoning once again, we conclude that $x_1 = x'_1$ and $x_2 = x'_2$. Putting this all together, it follows that $x = x'$, contradicting the assumption that (x, x') is a colliding pair for h_4. Hence h_4 is strong collision free.