Lecture 6.1:
Protocols - Authentication and Key Exchange I

CS 436/636/736
Spring 2012

Nitesh Saxena
Course Admin

- HW2 due now
- Solution to be provided soon
- Will start grading right away
  - Hope to return before the exam
Course Admin

• **Mid-Term Exam**
  – On March 08
  – In class, from 11am-12:15pm

• Covers lectures up to Feb 23

• In-class review on Mar 06 (Tuesday)

• Strictly closed-book (no cheat-sheets are allowed)

• A sample exam will be provided as we near the exam date
Outline of Today’s lecture

• Today we try to put everything together
  – Encryption (public-key/private-key)
  – MACs
  – Signing
  – Key-Distribution
• Secure protocols (for secure communication)
  – Authentication
    • We studied it somewhat while talking about key distribution
  – (Authenticated-) Key Exchange
• Designing secure protocols is hard – we’ll only be able to learn the basics today
• We’ll use the board extensively today – be prepared to take notes
Protocol

- A protocol is a set of rules using which *two or more* entities exchange messages
- It consists of *messages* and *rounds*
Messages and Rounds

- A message is a unit of information send from one entity to other
- A round is a basic unit of protocol time
  1. Wake up because of
     1. Alarm (or clock)
     2. Initial start or
     3. Receipt of message(s) from other(s)
  2. Compute something
  3. Send message(s) to other(s)
  4. Repeat 2-3 if needed
  5. Wait for message(s) or clock
Types of Adversaries

• Passive
  – Eavesdrop, delay, drop, replay messages

• Active
  – Eavesdrop, delay, drop, replay and modify messages
Model

- N parties
- Any party can initiate the protocol with any other party
- Each party can be running a number of sessions with any other party at any point
Adversary and Security Model

- Different session should use different keys
- Compromise of one session should not lead to the compromise of any other session
- Adversary is an active adversary and a part of the system
- “Simply forwarding” adversary is NOT considered an adversary against the protocol
  - Why?
  - Message authentication (m’)
  - Key exchange
Properties of a Secure Protocol

• Correctness
  – If entities taking part in the protocol behave honestly, (and also if there are no transmission errors) the protocol achieves its desired goal
    • In other words, if everything works as expected, does the protocol satisfy its desired goal?

• Security
  – No adversary can defeat the goal of the protocol (with a significantly high probability)
    • Adversary could be passive or active, depending upon the application (we consider the latter)
    • We won’t consider denial-of-service attacks
Origin authentication: forwarding a signed message (one can verify that the message was generated by the source but not from the party it is coming from)
Basis for Authentication

• Something you know
  – e.g., a PIN, a password

• Something you have
  – e.g., an access key or a card; a certificate; a smart card; an RFID tag

• Something you are
  – e.g., biometric (such as fingerprint)
Weak Authentication

- PINs, Passwords provide weak authentication
  - Security is based upon how hard the pin/password is to guess
  - Usually, the passwords are short and weak
  - Vulnerable to dictionary attacks
- Widely used in practice
  - Unix, web emails,......
- Protocol (A authenticates to B using a password P, that A shares with B)

1. $A \rightarrow B$: Hi, this is A!
2. $B \rightarrow A$: $r$ (random challenge)
3. $A \rightarrow B$: $H(p,r)$

Talk about UNIX password authentication (look at HAC)
Strong Authentication

- An entity authenticates to the other by proving the knowledge of a secret associated with that entity, without revealing anything meaningful about the secret itself.
- Can be achieved through:
  - Private/Public Key Encryption
  - MAC
  - Digital signatures
- Strong because the security reduces to the security of the underlying cryptographic primitive, which is assumed to be hard to break.
- Our focus in the rest of the lecture.
- We’ll study both private-key and public-key based authentication.
Symm. Encryption-based authentication

- Uses encryption to authenticate Alice to Bob (assuming Alice-Bob have established a shared secret $K$, say through Kerberos)

A auth B

1. $A \rightarrow B$: Hi Bob, this is Alice!
2. $B \rightarrow A$: $r$ (a challenge)
3. $A \rightarrow B$: $\text{Enc}_K(r, B)$ (response)

Why not simply send $\text{Enc}_K(r)$ in msg 3? Reflection Attack
Security of the previous protocol

- An attacker needs to come up with a valid response
  - Impossible if encryption is secure
- r must not be re-used by Bob
  - Why?
Freshness

• Assurance that message has not been used previously and originated within an acceptably recent timeframe

• Two methods:
  – Nonce (Number used once)
  – Timestamps
Nonces

• One-time random number
• We depended on B to make sure r is a good nonce
• Choose nonces “randomly” from a large space (such as $2^{128}$) to avoid re-use and for unpredictability – good RNG
Timestamps

• Inclusion of date/time-stamp in the message allows recipient to check it for freshness
  – Need to be protected with cryptographic means

• $A \rightarrow B: \text{Enc}_k(T,B)$
  – Results in fewer messages and rounds

• But, requires synchronized clocks
  – hard to achieve in practice!
Encryption-based Mutual Authentication (1)

- Run two copies of the uni-lateral authentication protocol $\rightarrow$ 4 rounds
- We can piggyback common flows

1. $A \rightarrow B$: $A, rA$
2. $B \rightarrow A$: $\text{Enc}_K(rB, rA, A)$
3. $A \rightarrow B$: $\text{Enc}_K(rA, rB)$
Encryption-based Mutual Authentication (2)

1. $A \rightarrow B$: $A, \text{Enc}_K(T, B)$
2. $B \rightarrow A$: $\text{Enc}_K(T+1, A)$
Session Key Exchange with KDC – Needham-Schroeder Protocol

• A -> KDC \ ID_A | ID_B | N_1
  \(\text{Hello, I am Alice, I want to talk to Bob, I need a session Key and here is a random}
  \text{nonce identifying this request)}\)

• KDC -> A \ E_{K_A}(K | ID_B | N_2 | E_{K_B}(K | ID_A))
  \(\text{Encrypted} (\text{Here is a key, for you to talk to Bob as per your request N}_2 \text{ and also an}
  \text{envelope to Bob containing the same key)}\)

• A -> B \ E_{K_B}(K | ID_A)
  \(\text{(I would like to talk using key in envelope sent by KDC)}\)

• B -> A \ E_K(N_2)
  \(\text{(OK Alice, But can you prove to me that you are indeed Alice and know the key?)}\)

• A -> B \ E_K(f(N_2)) \(\text{(Sure I can!)}\)
Session Key Exchange with KDC – Needham-Schroeder Protocol (corrected version with mutual authentication)

- **A -> KDC**: $\text{ID}_A \ || \ \text{ID}_B \ || \ N_1$
  (Hello, I am Alice, I want to talk to Bob, I need a session Key and here is a random nonce identifying this request)

- **KDC -> A**: $E_{K_A}(K || \text{ID}_B || N_1 || E_{K_B}(\text{TS}_1, K || \text{ID}_A))$
  (Encrypted (Here is a key, for you to talk to Bob as per your request $N_1$ and also an envelope to Bob containing the same key))

- **A -> B**: $E_K(\text{TS}_2, B)$, $E_{K_B}(\text{TS}_1, K || \text{ID}_A)$
  (I would like to talk using key in envelope sent by KDC; here is an authenticator)

- **B -> A**: $E_K(\text{TS}_2+1, A)$
  (OK Alice, here is a proof that I am really Bob)
Version 4 summary

(a) Authentication Service Exchange: to obtain ticket-granting ticket

1. $C \rightarrow AS$: $ID_c || ID_{gs} || TS_1$
2. $AS \rightarrow C$: $E_{K_c}[K_{C,gs} || ID_{gs} || TS_2 || Lifetime_2 || Ticket_{gs}]$

$Ticket_{gs} = E_{K_{C,gs}}[K_{C,gs} || ID_c || AD_c || ID_{gs} || TS_2 || Lifetime_2 ]$

(b) Ticket-Granting Service Exchange: to obtain service-granting ticket

3. $C \rightarrow TGS$: $ID_c || Ticket_{gs} || Authenticator_c$
4. $TGS \rightarrow C$: $E_{K_{C,gs}}[K_{C,V} || ID_c || TS_1 || Ticket_c ]$

$Ticket_{gs} = E_{K_{C,gs}}[K_{C,gs} || ID_c || AD_c || ID_{gs} || TS_2 || Lifetime_2 ]$
$Ticket_c = E_{K_{C,V}}[K_{C,V} || ID_c || AD_c || ID_s || TS_1 || Lifetime_2 ]$
$Authenticator_c = E_{K_{C,gs}}[ID_c || AD_c || TS_1 ]$

(c) Client/Server Authentication Exchange: to obtain service

5. $C \rightarrow K$: $Ticket_c || Authenticator_c$
6. $K \rightarrow C$: $E_{K_{C,V}}[TS_5 + 1]$ (for mutual authentication)

$Ticket_c = E_{K_{C,V}}[K_{C,V} || ID_c || AD_c || ID_s || TS_1 || Lifetime_2 ]$
$Authenticator_c = E_{K_{C,gs}}[ID_c || AD_c || TS_5 ]$
Further Reading

• HAC – chapter 10
• Stallings – Chapter 15