

---

# Entity Authentication

Sweta Kothari

---

BoiseCrypt Fall'09 – 16<sup>th</sup> to 18<sup>th</sup> Dec

---

# INDEX

1. Introduction
  2. Principle & Objectives
  3. Types of Entity authentication
    - ❑ Weak Authentication: Passphrase Advantages and Disadvantages
    - ❑ Towards Strong authentication: Lamport Authentication
    - ❑ Strong Authentication : Symmetric and asymmetric key based, advantages and disadvantages
    - ❑ Zero knowledge based : Schnorr scheme
  4. Various attacks on Identification protocol
  5. Counter measures for these attacks
  6. Comparison among these attacks
-

# Why one more kind of authentication??

- In some cases we might want to check that the person to whom we are communicating is the one meant for rather than just concentrating on the message.
- To check “entity”; we have ENTITY AUTHENTICATION.
- One of its primary usage can be related to the scenario where resources are access privileged like servers, in military operations, etc.

---

# Terminologies involved

- **Verifier** - The person who is in charge of checking that the correct entity is involved in communication is the Verifier. Verifier can also create some tokens by itself during communication which are used.
  - **Claimant** - The person who wants to start communication by proving its identity is Claimant.
  - **Nonce** – Time variant parameter which is served to distinguish one protocol instance from another like random number.
  - **Salt** – Upon arrival of password we may add some bits upon initial entry. This t-bit random string is called “Salt”.
-

---

# More on Entity Authentication

- This provides a basis by which one party (the Verifier) can gain assurance that the identity of another (called as Claimant) is as declared to prevent impersonation and is actually participating.
  - The common technique to guarantee this is by checking the correctness of message which might have been sent earlier and helps us know that claimant has some secret associated by design with the verifier.
-

# Entity Authentication Vs. Message Authentication

Message Authentication	Entity Authentication
<ul style="list-style-type: none"><li>▪ This involves a meaningful message.</li><li>▪ This doesn't provide timeliness guarantees as to when it was created etc.</li></ul>	<ul style="list-style-type: none"><li>▪ This doesn't involve meaningful message; it involves some "claim" for proving its entity.</li><li>▪ Time is important, as in this protocol corroboration of a claimant's identity takes place.</li></ul>

---

# Basis Of Entity Authentication

- **Something Known** : They include standard password, PIN – personal identification numbers, etc.
  - **Something possessed** : This includes some physical accessory like our own student id to access labs, ATM cards etc.
  - **Something Inherent** : They include characteristics like finger prints, handwritten signatures, voice, i.e. some human physical characteristic.
-

---

# Properties of Entity Protocol

- **Reciprocity of identification** : Both claimant and verifier may prove their identities to other providing unilateral or mutual identification.
  - **Computational Efficiency** : The number of operations required to execute a protocol.
  - **Communication Efficiency** : Bandwidth required or number of passes.
  - **Third-part involvement** : E.g. Online trusted third party to distribute common symmetric keys.
  - **Security Guarantees** : E.g. Provable security and zero knowledge security.
  - **Storage of Secrets** : Location and method used.
-

---

# Types of Entity Authentication

1. Weak Authentication – Passphrase
  2. Partially Strong – One time passwords
  3. Strong authentication – using symmetric key  
- using asymmetric key
  4. Zero knowledge based – Schnorr
-

---

# Password (Weak Authentication)

This is amongst the most conventional schemes where in a user has an 'user id' and a 'password'. User id acts like a claim and password as evidence supporting the claim.

The system checks to see if it matches or not. Here demonstration of knowledge of the secret which is password in this case; corroborates that the person is verified.

## **Passphrase :**

- The user types in a sentence rather than a short word for password. The idea is it is easier for a user to remember sentences.
  - The entered sentence is then compared with what is stored in for the corresponding user.
  - The long password can be encrypted or in plain-text when stored on the medium or data base.
-

---

# Advantages and Disadvantages of Weak authentication

## **Advantages :**

- It has better entropy than a short password.
- It is easier to remember than the usual passwords.

## **Disadvantage:**

- This is really weak against attacks as intruder can hear over communication channel and impersonate it later.
  - It is also very easy to replay the same message and use it later.
  - Exhaustive search or password guessing i.e. dictionary attacks can also be used.
  - One has to type extra.
-

---

# One-time passwords

## (Towards Strong Authentication)

- **Lamport's One Way Functions**

Let  $H$  be a one-way function. User  $A$  begins with secret  $w$ .

$A$  sends  $w_0 = H^t(w)$ .  $B$  initializes its counter for  $A$  to  $i_A = 1$ .

The  $i^{\text{th}}$  identification proceeds from

$A \rightarrow B : A, i, w_i (= H^{t-i}(w))$

$B$  checks that  $i = i_A$  and that the received password  $w_i$  satisfies  $H(w_i) = w_{i-1}$ .

Once verified and successful it sets  $i_A = i_A + 1$  and saves  $w_i$ .

- **Advantages:** A partial solution to this is one-time passwords. It addresses the major concern of one-time password schemes i.e. eavesdropping and replay of password. They are safe against passive adversaries who eavesdrop and later try and impersonate.
  - **Disadvantages:** It's still vulnerable to an active adversary who intercepts and traps. Challenge response technique addresses this threat.
-

---

# Challenge-response identification

## ( Strong Authentication)

- The central idea of challenge response is that claimant proves its identity to verifier by demonstrating knowledge of a secret known to be associated with entity without revealing the secret itself to the verifier during the protocol.
  - The challenge is usually time variant and is random number.
  - As every time the challenge is different, even if the adversary is monitoring the network it won't help as challenge changes every time.
-

# Challenge-Response :

## Symmetric Key

- Challenge-response by symmetric-key techniques - unilateral authentication, using random number.
- Here, A is the claimant and B is the verifier.
- Communication takes place as

$A \leftarrow B : r_B$  -----(1)

$A \rightarrow B : E_K(r_B, B^*)$  -----(2)

B sends A a random number. To prove its claim, A then encrypts the random number send by B using the symmetric encryption key 'k'. It also sends the optional field of the verifier as 'B'. This prevents reflection attack as the key used is bi-directional key 'k'.

B then decrypts the message sent by A to see the random number is the same as it had sent. It also sees if the identifier matches. If either of them is not true it stops any further communication.

# Challenge-Response :

## Public key decryption

- Here PK keys are used.
- This is how protocol works :

$$A \leftarrow B : h(r) , B , P_A(r,B) \text{ ---- (1)}$$

$$A \rightarrow B : r \text{ ---- (2)}$$

B chooses a random number and computes  $h(r)$ . 'h' is a one way hash function,  $P_A$  is the public key of A e.g. RSA, B is its own identifier.

### **Verification Steps :**

1. A decrypts  $r$  and B and computes hash of  $r$  to see if that matches  $h(r)$  sent or not. Stops if that doesn't match.
2. Matches the encrypted identifier after decrypting it and see if that matches the one sent in plain text. Stops if that doesn't match.
3. If both 1 and 2 are true it then sends the random number in plain text to B. Now B checks to see if it is the same random number or not. If not then it stops.

---

# Advantages and Disadvantages

- **Advantages::**

- It is a time variant challenge where the response depends on entity's secret and challenge.
- Even if the communication line is monitored, the response from one execution of identification protocol will not provide an adversary with useful information.

- **Disadvantages::**

- This protocol still would reveal some partial information about the claimant's secret, an adversarial verifier might still be able to strategically select challenges to obtain responses providing information.
-

---

# Zero-knowledge Identification Protocols

- To address the impersonation issues, zero knowledge protocols are used. It allows a claimant to demonstrate knowledge of a secret while revealing no information of use to verifier.
-

---

# Schnorr Identification Protocol

- Schnorr identification protocol is zero knowledge based protocol.
  - Its security is based on the intractability of the discrete logarithm problem.
  - It is a three pass protocol where a claimant proves its identity to verifier using it.
-

# Schnorr Id: System Parameters

- A prime  $p$  is selected such that  $(p-1)$  is divisible by another prime  $q$ .
- $\beta$  is chosen such that  $1 \leq \beta \leq p - 1$ , having multiplicative order  $q$ .
- Each party obtains authentic copy of  $(p, q, \beta)$  and verification function, allowing verification of  $T$ 's signatures  $S_T(m)$  on message  $m$ .
- A parameter  $t$  is chosen such that  $2^t < q$

# Schnorr Id : Selection of per-user parameters

- Each claimant is given unique identity like  $A$  is given  $I_A$ .
- $A$  chooses a private key 'a',  $0 \leq a \leq q-1$  and computes  $v = \beta^{-a} \pmod p$ .
- $A$  identifies itself to  $T$ , transfers  $v$  to  $T$  and obtains certificate as  $\text{cert } A = (I_A, v, S_T(I_A, v))$  from  $T$  binding  $I_A$  with  $v$ .

# Schnorr Id : Protocol Messages

- Protocol involves 3 messages :-

1.  $A \rightarrow B : \text{cert}_A, x = \beta^r \text{ mod } p$

B checks to see that  $S(I_A), S(v)$  is equal to the value of  $I_A$  and  $v$  sent in certificate when signed and in return sends 'e'.

2.  $A \leftarrow B : e$  (where  $1 \leq e \leq 2^t$ )

A checks that the value of  $e$  send is in the appropriate range.

3.  $A \rightarrow B : y = ae + r \text{ mod } q$

B now computes  $z = \beta^y v^e \text{ mod } p$  and accepts A if  $z=x$ .

# Protocol Actions

## Protocol actions - Claimant 'A' identifies to verifier 'B' as :

- a) 'A' chooses random #  $r$ ,  $0 \leq r \leq q-1$  computes  $x = \beta^r \text{ mod } p$  and sends (1) to B.
- b) B authenticates A's public key  $v$  by verifying T's signature on  $\text{cert}_A$ , then sends to 'A' a random 'e'.
- c) A checks that  $1 \leq e \leq 2^t$  and sends B :  $y = ae + r \text{ mod } q$
- d) B computes  $z = \beta^y v^e \text{ mod } p$  and accepts A's identity provided  $z=x$

# Advantages and Disadvantages of Schnorr

**Advantages :** Since it is zero - knowledge based protocol, the knowledge that the claimant proves is insufficient for verifier to plan chosen – text attack

## **Disadvantages:**

- 1) It is assumed that computing discrete logs modulo a prime  $p$  is intractable.
- 2) Even though lot of computations at the claimant side is pre-computed, still there is lot of computation at verifier side.

---

# Attacks on Entity Protocol

1. **Impersonation:** A deception whereby one entity purports to be another.
  2. **Replay attack:** A deception involving use of information from a single previous protocol execution, on the same or a different verifier.
  3. **Interleaving Attack:** An impersonation or other deception involving selective combination of information from one or more previous or simultaneously ongoing protocol executions.
  4. **Reflection Attack:** An attack involving sending information from an ongoing protocol execution back to the originator.
  5. **Forced Delay:** A forced delay occurs when an adversary intercepts a message and relays it at some later point in time.
  6. **Chosen-text attack:** An attack on a challenge-response protocol wherein an adversary strategically chooses challenges in an attempt to extract information about the claimant's long-term key.
-

# Counter-measures

<b>Attack Type</b>	<b>Measures</b>
Replay	Use of nonce, challenge response techniques, embedding identity.
Interleaving	Linking messages
Reflection	Uni-directional keys, embed identifier of target party in challenge responses
Chosen-text	Use of zero-knowledge techniques; embed in a self-chosen random number
Forced-delay	combined use of random numbers with short response time-outs; timestamps

# Comparison among the Entity Schemes that we have seen so far

<i>Properties</i>	<b>Passphrase (Weak Authentication)</b>	<b>Challenge response identification (Symmetric keys)</b>	<b>Zero knowledge based – Schnorr</b>
<i>Reciprocity of identification</i>	Unilateral	Unilateral or mutual	Usually Mutual
<i>Computational efficiency</i>	Not much	Only random number generation, key generation and validation is needed	Claimant data could be pre computed but there is still lot of computation at verifier's end.
<i>Communication efficiency</i>	Not much	Mostly is 1,2 or 3 pass with not much data.	Needed
<i>Involvement of 3rd party</i>	No need	Might be needed if we want some party to generate keys for us.	Not needed.
<i>Security guarantee</i>	Not good	Moderate	Good
<i>Storage of secret</i>	Local disk	Local disk, Software or hardware tokens	Local disk or software

---

# Relation between Entity & Signature Schemes

- Entity schemes are closely related to digital signatures schemes but they are simpler.
  - Digital signatures have variable message length whereas semantics of entity schemes is fixed.
  - Also entity authentication is real time and whether a user is accepted or rejected is known immediately whereas digital signatures provide non-repudiation feature allowing disputes to be resolved after the fact.
  - Also entity authentication doesn't have life time signatures.
-

---

# References

- Handbook of applied cryptology : A.J. Menezes, P.C. Van Oorschot and S.A. Vanstone,  
[http://www.cacr.math.uwaterloo.ca/hac/about/  
chap10.pdf](http://www.cacr.math.uwaterloo.ca/hac/about/chap10.pdf)
  - <http://java.sun.com/j2se/1.4.2/docs/api/>
-

---

# Q & A

---