Distributed Systems Security

Authentication Principles - 2

Prof. Steve Wilbur
s.wilbur@cs.ucl.ac.uk

Z08

Lecture Objectives

- Examine classic authentication protocols based on PKC and identify weaknesses
- Examine classic ways of dealing with key distribution

Z08

Basic Protocol Map

- We will use the following map to keep our bearings as we explore the various protocols

	Shared Key (SKC)	Public Key (PKC)
Key Distribution	Shared Key Distribution	Public Key Distribution
One-way Authentication	One-way SKC	One-way PKC
Two-way Authentication	Two-way SKC	Two-way PKC

Z08

Notation

A, B	Principals - "good guys" e.g.. Alice, Bob
U, V, W	Domains - "organisations" e.g.. UCL, IBM
E	Eavesdropper - "bad guys", e.g.. Eve
S	Security server/service
IDx	Identity (name) of "x"
KRx	pRivate (secret) key of "x"
KUx	pUblic key of "x"
Ks	Conventional (SKC) "Session" key
{Data}K	Data encrypted with key K

Z08

Two-way PKC Authentication

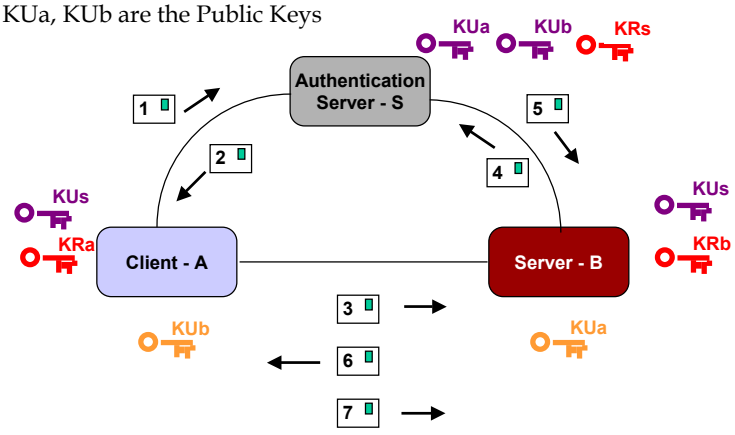
- Assume A, B already have their own private keys, KRa and KRb (do not know each other's key)
- The trusted third party, S knows public keys KUa and KUb of A and B
- This information is public so why does S need to be trusted?

- Protocol needed to mutually authenticate A and B

Z08

Needham & Schroeder Two-Way PKC Authentication

KRa, KRb are Private Keys
KUa, KUb are the Public Keys



Z08

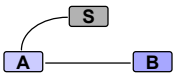
Needham & Schroeder Two-Way PKC Authentication - 2

- Appears more complex: more keys, more messages
- Red keys are private and purple keys are public keys - distributed before protocol begins
- Orange keys are distributed by protocol

- Protocol principle is actually simpler than for SKC
- Not all steps needed after first use

Z08

Needham & Schroeder PKC Protocol



- A, B are parties involved
- S is Authentication Server
- KUa, KUb, KUs are Public keys of A, B, S
- KRa, KRb, KRs are Private Keys of A, B, S
- I is "nonce" used once only
- {x}k means "x encrypted by key k"
- Caching of Public Keys can reduce number of steps

Authentication with Public Keys

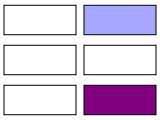
1.	A->S:	IDa, IDb
2.	S->A:	{KUb, IDb}KRs
3.	A->B:	{Ia, IDa}KUb
4.	B->S:	IDb, IDa
5.	S->B:	{KUa, IDa}KRs
6.	B->A:	{Ia, Ib}KUa
7.	A->B:	{Ib}KUb

Data Phase

x.	A->B:	{{Mab}KRa }KUb
y.	B->A:	{{Mba}KRb }KUa

Z08

Needham & Schroeder PKC Protocol - 2

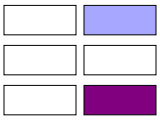


- Note token at step 2 contains identity of B and its public key encrypted by S
- But this is not a secret, so why is it encrypted?

- This is embryo form of a Public Key Certificate
  - Provides information vouched for by S
  - Securely distributed - encrypted with S's private key
  - KUb and B's identity bound by encryption process
  - Could send in clear with accompanying digital signature instead

Z08

Needham & Schroeder PKC Protocol - 3

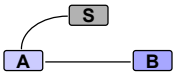


- Steps 3, 6 and 7 establish "live-ness" of A and B by showing they can use their private key to encrypt and value chosen by the other party
- One case of a challenge-response protocol

- Encryption of messages using:
  - sender's private key for authentication (anyone can decrypt it) and,
  - recipient's public key for secrecy (only recipient can decrypt it)
- But, use of PKC for message exchange is slow...

Z08

Denning Hybrid Protocol



- Timestamp in certificates provides validity period
- Timestamp in message 3 for Ks distribution protects against replay of session key
- Session key chosen by A
- BUT, usual problems with clock synchronisation

Authentication with Public Key Cryptography

1. A->S: IDa, IDb
2. S->A: CERTa, CERTb
3. A->B: CERTa, CERTb, {{Ks, T}KRa }KUb

CERTa = {IDa, KUa, Ta}KRs

CERTb = {KUb, IDb, Tb}KRs

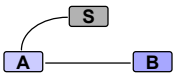
Data Phase with SKC

x. A->B: {Mab} Ks

y. B->A: {Mba} Ks

Z08

Woo and Lam Hybrid Protocol



- Certificates (red) do not have validity - but could have
- Token (purple) contains session key and info for mutual authentication
- This binding assures A that Ks is fresh
- Session key chosen by S

Authentication with Public Key Cryptography

1. A->S: IDa, IDb
2. S->A: {KUb, IDb}KRs
3. A->B: {Na, IDa}KUb
4. B->S: IDb, IDa, {Na}KUs
5. S->B: {IDa, KUa}KRs, {{Na, Ks, IDb}KRs }KUb
6. B->A: {{Na, Ks, IDb}KRs, Nb}KUa
7. A->B: {Nb} Ks

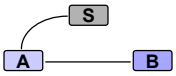
Data Phase with SKC

x. A->B: {Mab} Ks

y. B->A: {Mba} Ks

Z08

Woo and Lam Hybrid Revised Protocol



- Weakness in that Na may only be unique among A's nonces
- So IDa, Na uniquely identifies session request
- Note: many published protocols have been revised after careful analysis!!

Authentication with Public Key Cryptography

1. A->S: IDa, IDb
2. S->A: {KUb, IDb}KRs
3. A->B: {Na, IDa}KUb
4. B->S: IDb, IDa, {Na}KUs
5. S->B: {IDa, KUa}KRs, {{Na, Ks, IDa, IDb}KRs }KUb
6. B->A: {{Na, Ks, IDa, IDb}KRs, Nb}KUa
7. A->B: {Nb} Ks

Data Phase with SKC

x. A->B: {Mab} Ks

y. B->A: {Mba} Ks

Z08

Public Key Distribution

- In principle we can distribute public keys by:
  - Public announcement
  - Publicly available directory
  - Public-key authority
  - Public-key certificates
- Public Announcement
  - Distribution of public key alone means that recipient is responsible for labeling it with owner's identity
  - Precarious - label may come "unglued"
  - Rogue may substitute their own PK allowing them to intercept/fake messages and denies service to legitimate party

Z08

Public Key Distribution - 2

- Publicly Available Directory
  - Rely on OS file protection to bind owner's identity and Public Key
  - Allow owners update access
  - All other users read only access
  - Most file system security can be cracked
- Public Key Authority
  - Similar to Authentication Server (S) in N&S protocols
  - For volume use S would be bottleneck
  - So need approach which has many more distribution points, yet still retains proof of authenticity

Z08

Public Key Certificates

- Bind owner's identity to their public key under specified issuing authority
- Can be freely distributed
- Tampering will be detected, ∴ route taken for delivery is very unlikely to weaken trust in the certificate *per se*
- Can be distributed by anyone, e.g. certificates that might be needed by the recipient can be sent with message
  - Recipient can check their validity, or
  - Recipient can get "fresh" ones from its favourite Certificate Distribution Centre

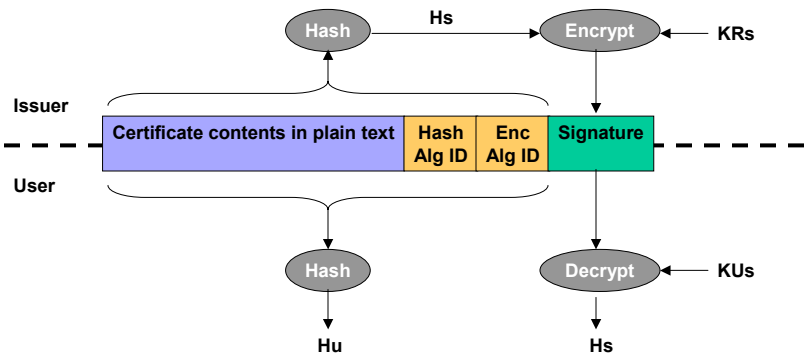
Z08

Public Key Certificates

- In practice: need additional information such as:
  - Validity period
  - Encryption algorithm identifier
  - Mode of use of algorithm, e.g.. key length
  - etc.
- Certificates have to be revoked if:
  - Private key compromised
  - Key details change, e.g.. owner leaves company
- These and other practical issues covered later

Z08

Certificate Life-Cycle



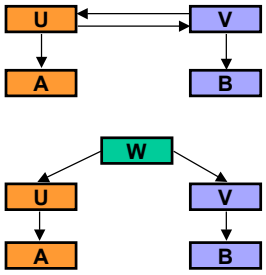
If Hu = Hs then the Certificate has not been tampered with

Similar technique can be used for ensuring integrity of any plaintext message

Z08

PKC Authentication Multiple Domains

- Consider A and B in different domains U and V
- ∴ A's PK distributed via certificate issued by U and B's by V
- How does A get B's PK and vice versa?
- Trust relationships:
  - U and V trust each other - Cross-Certification
  - U and V trust a third party W - Super-Domain



Z08

Certification Paths

- Let us first take simplest case where A and B are in same domain U
- Cu(A) = Certificate for A issued by U
  = {IDa, KUa}KRu
- Cu(B) = Certificate for B issued by U
  = {IDb, KUb}KRu
- For communication between A and B:

A has:		B has:	
	Cu(A)		Cu(B)
	Cu(B)		Cu(A)
	Pu		Pu

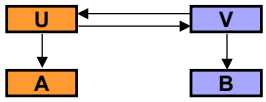
- Using Pu, A and B can both validate other party's certificate ie. verify that U issued the certificate and that U believes that KUx is the public key of IDx
- ∴ A and B can believe that KUx is the public key of IDx

Z08

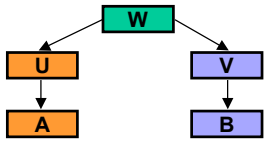
Certification Paths - 2

- If A in domain U then Cu(A) issued by U
- If B in domain V then Cv(B) issued by V

- Cross Certification:
  - U issues Cu(V) = {IDv, KUv}KRu
  - V issues Cv(U) = {IDu, KUu}KRv



- Super Domain:
  - W issues Cw(U) = {IDu, KUu}KRw
  - W issues Cw(V) = {IDv, KUv}KRw



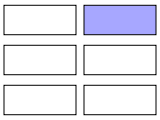
Z08

Certification Paths - 3

- Both use same approach for finding target PK so let us just consider Cross Certification case
- A might send to B
  - Message, Cu(A), Cv(U)
  - These are certificates and are public - so Cv is not just available to entities within domain V
- B has Pv
- ∴ B can extract Public Key of U (strictly believe that KUu is the public key of IDu)
- B can then can extract Public Key of A using KUu from Cu(A) (strictly believe that KUa is the public key of IDa)
- Chain can be longer, e.g.
  Message, Cx1(A), Cx2(x1), Cx3(x2), ... Cxn(Xn-1)
  where B is in domain Xn and possesses Pxn

Z08

Certification Paths - 4



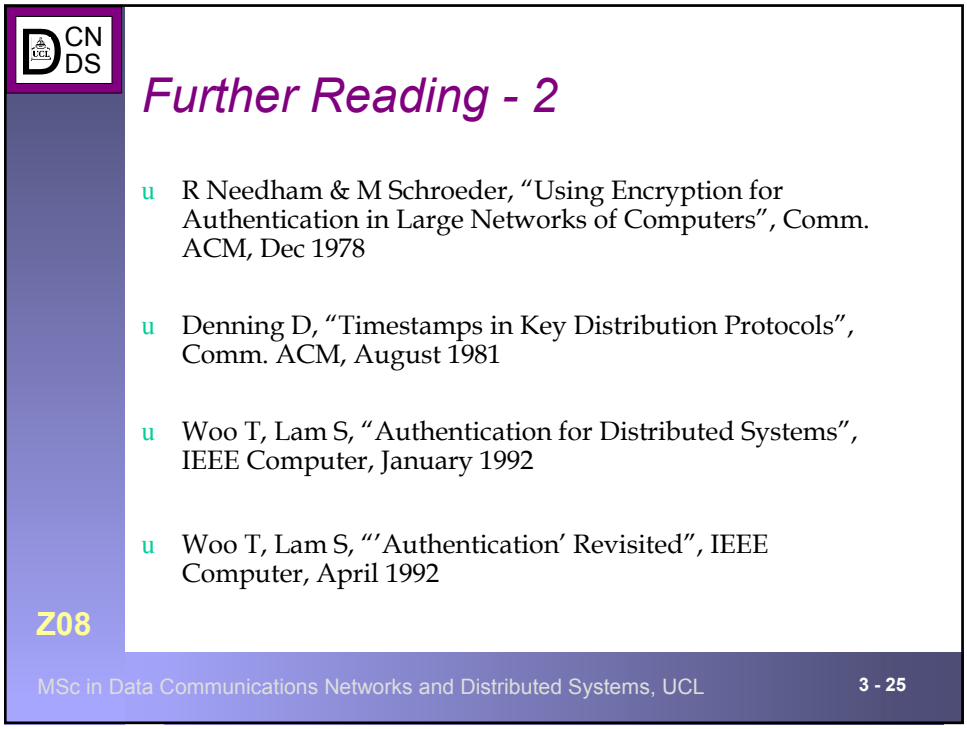
- Who can sign a certificate?
  Jurisdiction
- What paths will be acceptable to recipient?
  Trust
  Forward and reverse chains may not be identical
- Should names be personal or related to an organisation?
  Possible double signatures

Z08

Further Reading

- W Stallings, "Cryptography and Network Security: Principles and Practice", 2ed, Prentice Hall, 1999, 0-13-869017-0
  - **Public Key Authentication:** pp 308-309
  - **Key Management:** pp 182-193
- C Pfleeger, "Security in Computing", 2ed, Prentice Hall, 1997, 0-13-185794-0
  - **Public Key Authentication:** `pp 132-134

Z08


CN
DS
Further Reading - 2
u R Needham & M Schroeder, "Using Encryption for Authentication in Large Networks of Computers", Comm. ACM, Dec 1978
u Denning D, "Timestamps in Key Distribution Protocols", Comm. ACM, August 1981
u Woo T, Lam S, "Authentication for Distributed Systems", IEEE Computer, January 1992
u Woo T, Lam S, "'Authentication' Revisited", IEEE Computer, April 1992
Z08
MSc in Data Communications Networks and Distributed Systems, UCL
3 - 25