COSIC

KU LEUVEN

Entity Authentication and Symmetric Key Establishment

Prof. Bart Preneel
COSIC – KU Leuven - Belgium
Firstname.Lastname(at)esat.kuleuven.be
http://homes.esat.kuleuven.be/~preneel
@bpreneel1
June 2023

1

Goals

- Understand goals of entity authentication
- Understand strength and limitations of entity authentication protocols including passwords
- Understand subtle problems when entity authentication protocols are deployed in practice
- Understand properties of protocols for key establishment and entity authentication

2

2

Definitions

1970s

Confidentiality
Integrity
Availability

	data	entities
confidentiality	encryption	anonymity
authentication	data authentication	identification

Authorisation

Non-repudiation of origin, receipt

Contract signing

Notarisation and Timestamping

E-voting, e-auction,…

Don't use the word authentication without defining it

3

3

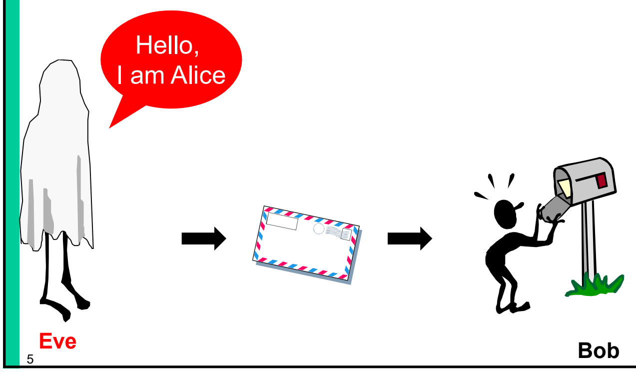
Entity authentication (identification)

- the problem
- passwords
- challenge response with symmetric key and MAC (symmetric tokens)
- challenge response with public key (signatures, ZK)
- biometry

4

4

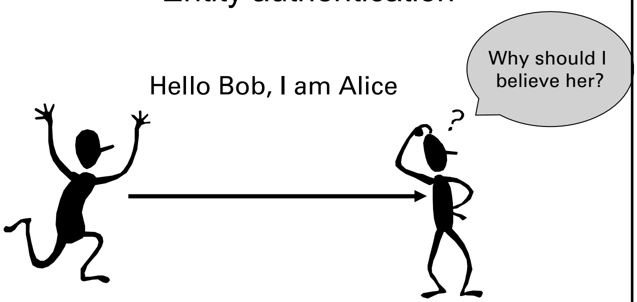
Entity authentication



5

5

Entity authentication



entity authentication: one is corroborated of the identity of another party, and of the fact that this party is **alive (active)** during the protocol

6

6

Entity authentication is based on one or more of the following elements:

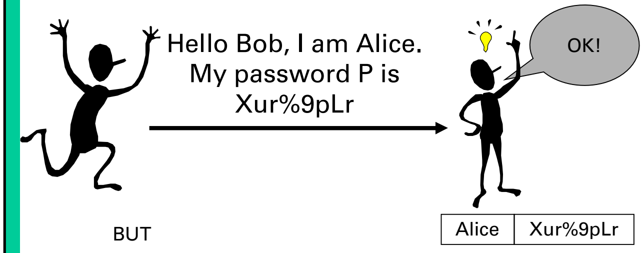
- what someone knows
  - password, PIN
- what someone has
  - magstripe card, smart card
- what someone is (biometrics)
  - fingerprint, retina, hand shape,...
- how someone does something
  - manual signature, typing pattern
- where someone is
  - dialback, location based services (GSM, Galileo)

ert5^r$#89Oy

7

7

"Entity authentication" with passwords



BUT

- Eve can guess the password
- Eve can listen to the channel and learn Alice's password
- Bob needs to know Alice's secret
- Bob needs to store Alice's secret in a secure way

Possibility of replay: liveliness is missing

8

8

Problem: human memory is limited

- Solution: store key K on magstripe, USB key, hard disk
- Stops guessing attacks

But this does not solve the other problems related to passwords

And now you identify the card, not the user….

Possibility of replay: liveliness is missing

9

9

Improvement: Static Data Authentication

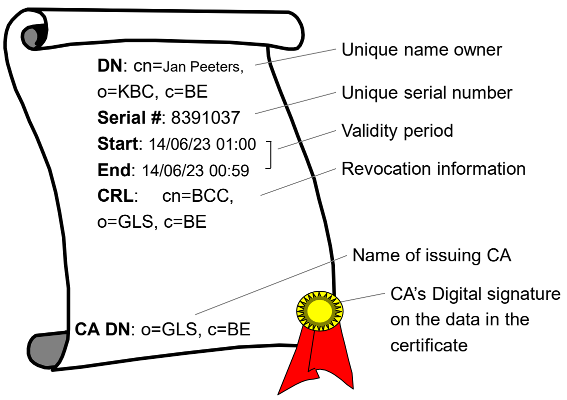
- Replace K by a signature of a third party CA (Certification Authority) on Alice's name: $SigSK_{CA}$ (Alice) = special certificate
- Advantage: can be verified using a public string PK_{CA}
- Advantage: can only be generated by CA
- Disadvantage: signature = 40..128 bytes
- Disadvantage: can still be copied/intercepted

Possibility of replay: liveliness is missing

10

10

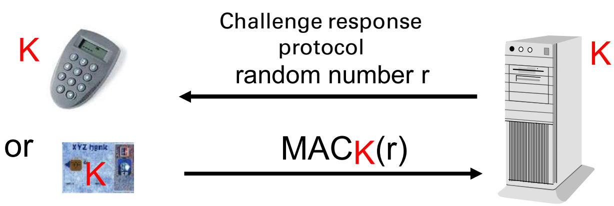
"Certificate" for static data authentication



11

11

Entity authentication with symmetric token



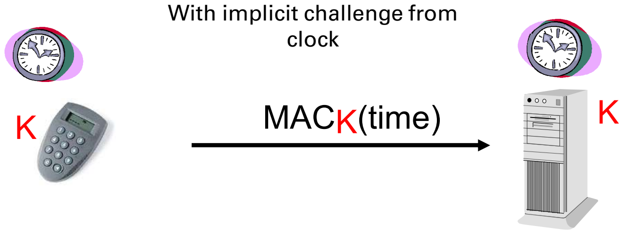
- Eavesdropping no longer effective
- Bob still needs secret key K
- IETF RFC 4226 HOTP (2005) HMAC-based One Time Password

Detects whether Alice is alive!

12

12

Entity authentication with symmetric token

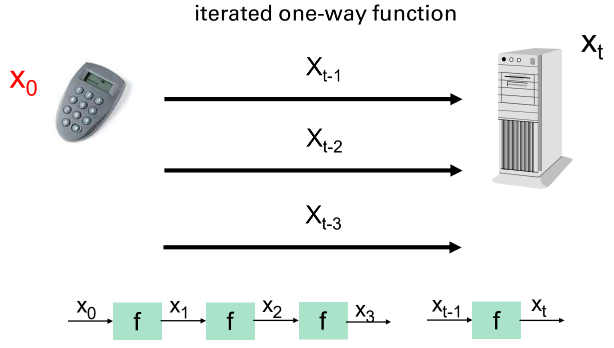


- Eavesdropping no longer effective
- Bob still needs secret key K
- resynchronization mechanism needed
- IETF RFC 5238 TOTP (2011) Time-based One Time Password

13

13

Lamport's one-time passwords

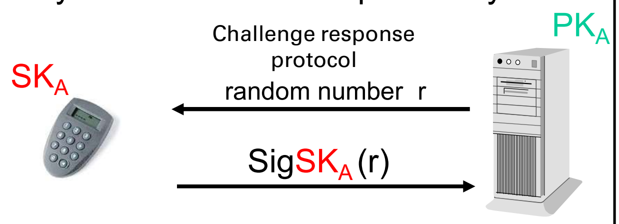


Disadvantage: only works with one Bob

14

14

Entity authentication with public key token

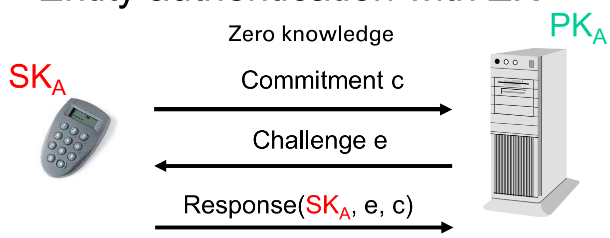


- Eavesdropping no longer effective
- Bob no longer needs a secret – only PK_A

15

15

Entity authentication with ZK



- Mathematical proof that Bob only learns that he is talking to Alice (1 bit of information)
- Bob cannot use this information to convince a third party that he is/was talking to Alice

16

16

ZK definitions

- **complete:** if Alice knows the secret, she can carry outthe protocol successfully
- **sound:** Eve (who wants to impersonate Alice) can only convinceBob with a very small probability that she is Alice;
- **zero knowledge:** even a dishonest Bob does not learn anything except for 1 bit (he is talking to Alice); he could have produced himself all the other information he obtains during the protocol.

17

17

ZK: Fiat-Shamir (1986)

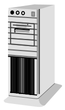
- central RSA modulus n
- per user:
  - identity I_A
  - secret key s_A $(0 < s_A < n)$
  - public key $y_A = s_A^2 \mod n$
- facts from number theory:
  - if one knows the factorization of n, it is easy to compute the square roots modulo n (if they exist);
  - if one can compute square roots modulo n, it is easy to factor n

18

18

ZK: Fiat-Shamir

All operations mod n



s_A

$r \in_R [1,n-1]$
$x = r^2$

I_A, y_A, x →

$x \in [1,n-1]$?

← Challenge e

$e \in_R \{0,1\}$

$z = r \cdot s_A^e$

Response z →

$z^2 = x \cdot y_A^e$?

1. Complete: trivial

2. Sound: Eve's probability of success = ½

Eve picks a random value for z and computes $x = z^2/y_A^e$
- Eve gambles that Bob will choose e=0: $x_0 = z^2$
- Eve gambles that Bob will choose e=1: $x_1 = z^2/y_A$

If Eve can correctly answer both cases for the same value of x (sent in step 1) then $z_0 = r$ and $z_1 = r \cdot s_A$ and thus she can compute $s_A = z_1/z_0$

19

19

ZK: Fiat-Shamir

3. Zero knowledge: Bob learns nothing about Alice's secret
- e=0: B learns r and r^2
- e=1: B learns $r \cdot s_A$ and $(r \cdot s_A)^2$
  $= r^2 \cdot y_A$ so Bob can compute this from r^2 and y_A

- $r \cdot s_A$ is a Vernam encryption of s_A: statistically independent of s_A
- hence B only sees a random value and its square mod n, which he could have produced himself (yet he is convinced that he has spoken to Alice!)

In practice: more iterations (20...40) for better security ($1/2^{20}$ ...$1/2^{40}$)

20

20

Overview Identification Protocols

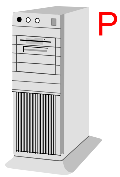
	Guess	Eavesdrop channel (liveliness)	Impersonation by Bob	Secret info for Bob	Mathema-tical proof	Security
Password	-	-	-	-	-	1
Magstripe (SK)	+	-	-	-	-	2
Magstripe (PK)	+	-	-	+	-	3
Dynamic password	+	+	-	-	-	4
Smart card (SK)	+	+	-	-	-	4
Smart Card (PK)	+	+	+	+	-	5
ZK	+	+	+	+	+	6

21

21

Entity authentication with password

Challenge response protocol



P

P

← random number r

$MAC_P(r)$ →

- Eavesdropping no longer effective
- Bob still needs secret key P
- Exhaustive search for P is easy based on a single transcript: not very secure

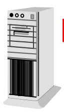
22

22

Entity authentication with password: EKE

[Bellovin,Merritt '92]

All operations mod p



P

P

$x \in_R [1,p-1]$

$A \,||\, E_P(\alpha^x)$ →

$y \in_R [1,p-1]$

r_B 128-bit string

← $A \,||\, E_P(\alpha^y || r_B)$

r_A 128-bit string

$k = (\alpha^y)^x$

$E_k(r_A || r_B)$ →

$k = (\alpha^x)^y$

← $E_k(r_A)$

- Adds entity authentication to Diffie Hellman
- Attacker cannot perform off-line exhaustive search for the password P
- Attacker can still try on-line attacks; need to restrict number of online attempts per account
- Bob needs to store the password P in clear

Literature: PAKE: Password Authenticated Key Establishment

23

23

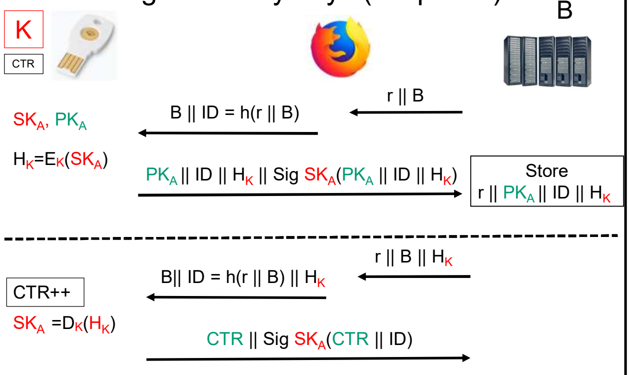
Google's security keys

- Standardized by FIDO Alliance
- Threat model
  - web attackers (host malicious web content)
  - related site attackers
  - network level attackers
  - malware (but not in browser)
- Hardware: public key + button to press
- Generate key pair for each website and authenticate using device key pair

24

24

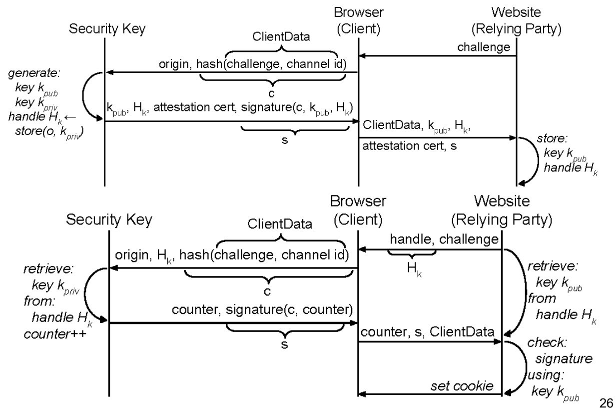
Google security keys (simplified)



25

25

Google's security keys

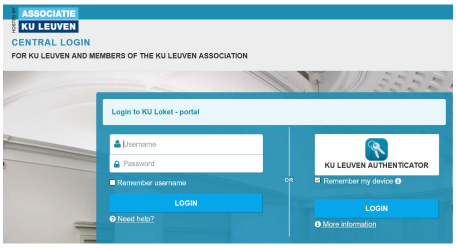


26

26

KU Leuven Authenticator - nextAuth

Public key protocol with local password verification on device



27

27

Entity authentication in practice

- Phishing – mutual authentication
- Losing devices – local authentication to device – need to check proper linking of two protocols (e.g. EMV)
- Sharing devices – biometry
- Interrupt after initial authentication – authenticated key establishment
- Mafia fraud – distance bounding

28

28

Mutual entity authentication

- Phishing is impersonating of the verifier (e.g. the bank)
- Most applications need entity authentication in two directions
- User needs to make judgment: difficult!
- Mutual entity authentication is not equivalent to 2 parallel unilateral protocols for entity authentication

29

29

Limitations of devices

- Device authenticates user
  - but if the user looses the device…
  - solution: authenticate user to device using password, PIN or biometrics
  - but need to connect both phases properly! (EMV example)
- Device can be passed on to others (delegation, fraud)
  - solution: biometrics

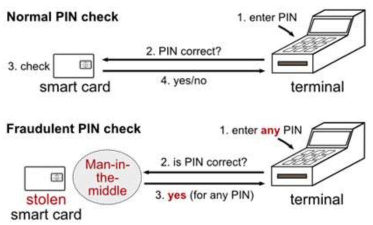
30

30

Warning about EMV

http://www.cl.cam.ac.uk/research/security/banking/nopin/oakland10chipbroken.pdf

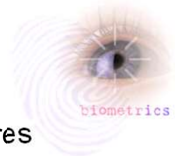
EMV PIN verification "wedge" vulnerability S.J. Murdoch, S. Drimer, R. Anderson, M. Bond, IEEE Security & Privacy 2010



31

31

Biometry

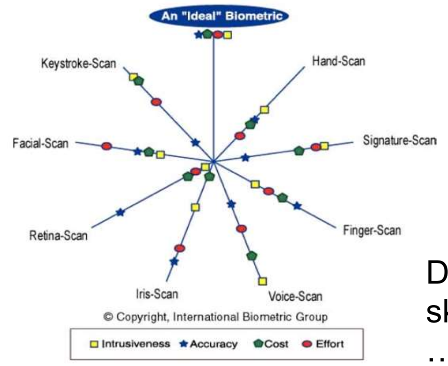


- Based on our unique features
- Identification or verification
  - Is this Alice?
  - Check against watchlist
  - Has this person ever registered in the system?

32

32

Some unique features



Source: Zephyr Analysis

DNA
skin
…

33

33

Biometry: pros and cons

- Real person
- User friendly
- Cannot be forwarded
- Little effort for user
- More suitable for supervised entity authentication (e.g. border controls)
- Secure implementation: derive key in a secure way from the biometric

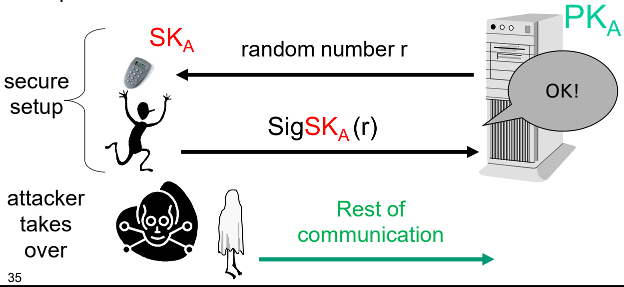
- Privacy (medical)
- Intrusive?
- Liveliness?
- Cannot be replaced
- Risk for physical attacks
- Hygiene
- Does not work everyone, e.g., people with disabilities
- Reliability
- No cryptographic key

34

34

Keeping authenticity alive

- Establish who someone is
- Establish that this person is active/liveliness
- But what if the connection is broken after the initial phase?



35

35

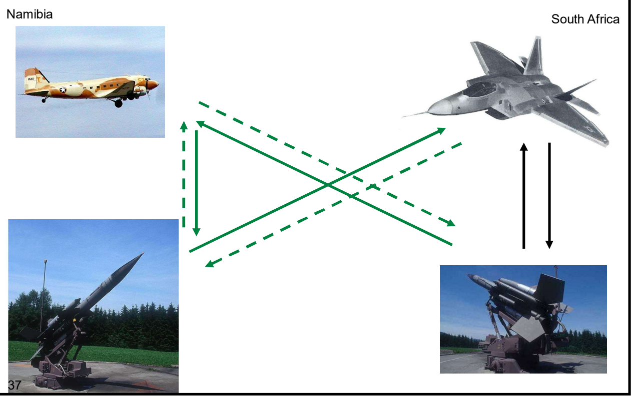
Solution

- Authenticated key agreement
- Run a mutual entity authentication protocol
- Establish a key
- Encrypt and authenticate all information exchanged using this key

36

36

The mafia fraud
– or the grandmaster chess problem



37

Location-based authentication

- Distance bounding: try to prove that you are physically close to the verifier
- Other uses of “location”
  - Dial-back: can be defeated using fake dial tone
  - IP addresses and MAC addresses can be spoofed
  - Mobile/wireless communications: operator knows access point, but how to convince others?
  - Trusted GPS: Galileo?

38

38

Key establishment

- The problem
- How to establish secret keys using secret keys?
- How to establish secret keys using public keys?
  - Diffie-Hellman and STS
- How to distribute public keys? (PKI)

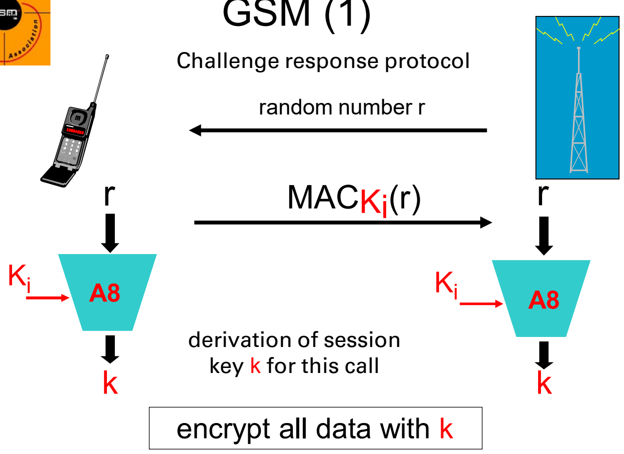
39

Key establishment: the problem

- Cryptology makes it easier to secure information, by replacing the security of information by the security of keys
- The main problem is how to establish these keys
  - 95% of the difficulty
  - integrate with application
  - if possible transparent to end users

40

GSM (1)



41

GSM (2)

- SIM card with long term secret key K_i (128 bits)
- secret algorithms
  - A3: MAC algorithm
  - A8: key derivation algorithm
  - A5.1/A5.2: encryption algorithm
- anonymity: IMSI (International Mobile Subscriber Identity) replaced by TIMSI (temporary IMSI)
  - the next TIMSI is sent (encrypted) during the call set-up

42

Point-to point symmetric key distribution

Before: Alice and Bob share long term secret K_{AB}

generate session key k → $E K_{AB}(k \| time \| Bob)$ → *decrypt extract k*

← $E k(time \| Alice \| hello)$ ←

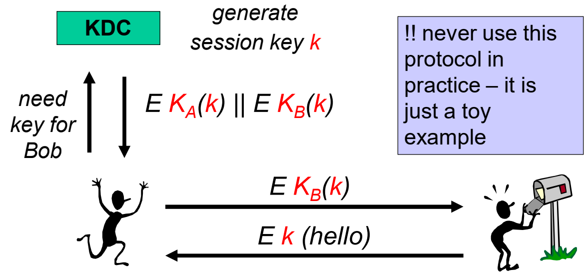
- After: Alice and Bob share a short term key *k*
  - which they can use to protect a specific interaction
  - which can be thrown away at the end of the session
- Alice and Bob have also authenticated each other

43

Symmetric key distribution with 3rd party

Before (KDC=Key Distribution Center)

- Alice shares a long term secret with KDC: K_A
- Bob shares long term secret with KDC: K_B



44

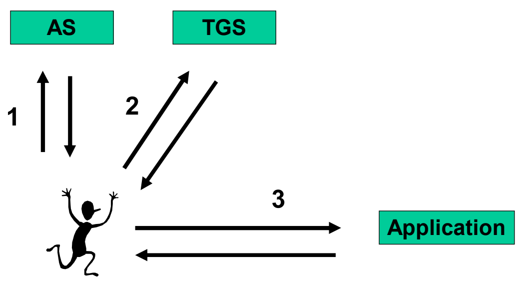
Symmetric key distribution with 3rd party(2)

- After: Alice and Bob share a short term key ***k***
- Need to trust third party!
- Single point of failure in system

45

Kerberos/Single Sign On (SSO)

Alice uses her password only once per day



46

Kerberos/Single Sign On (2)

- Step 1: Alice gets a “day key” K_A from AS (Authentication Server)
  - based on a Alice’s password (long term secret)
  - K_A is stored on Alice’s machine and deleted in the evening
- Step 2: Alice uses K_A to get application keys k_i from TGS (Ticket Granting Server)
- Step 3: Alice can talk securely to applications (printer, file server) using application keys k_i

47

A public-key distribution protocol: Diffie-Hellman

Before: Alice and Bob have never met and share no secrets; they know a public system parameter α

generate x, compute α^x → α^x → *generate y, compute* α^y

← α^y ←

compute $k=(\alpha^y)^x$ — *compute* $k=(\alpha^x)^y$

After: Alice and Bob share a short term key *k*

- Eve cannot compute ***k***: in several mathematical structures it is hard to derive x from α^x (this is known as the discrete logarithm problem)

48

Diffie-Hellman

generate x / *compute* α^x → α^x → *generate* y / *compute* α^y

← α^y

compute $k=(\alpha^y)^x$ — *compute* $k=(\alpha^x)^y$

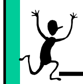
- how does Alice know that she shares this secret key ***k*** with Bob?
- answer: Alice has no idea at all about who the other person is! The same holds for Bob
- no authentication or key confirmation

49

49

Person-in-the middle attack

- Eve shares a key k_1 with Alice and a key k_2 with Bob
- Requires *active* attack
- Example: SSH rekeying on server



α^{x_1} → ; ← α^{y_1}

α^{x_2} → ; ← α^{y_2}

$k_1 = (\alpha^{y_1})^{x_1} = (\alpha^{x_1})^{y_1}$ $k_2 = (\alpha^{y_2})^{x_2} = (\alpha^{x_2})^{y_2}$

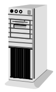
50

Entity authentication with password: EKE

[Bellovin,Merritt '92]

All operations mod p

P



P

$x \in_R [1,p-1]$ — A || $E_P(\alpha^x)$ → — $y \in_R [1,p-1]$, r_B 128-bit string

← A || $E_P(\alpha^y || r_B)$

r_A 128-bit string, $k = (\alpha^y)^x$ — $E_k(r_A || r_B)$ → — $k = (\alpha^x)^y$

← $E_k(r_A)$

- Adds entity authentication to Diffie Hellman
- Attacker cannot perform off-line exhaustive search for the password P
- Attacker can still try on-line attacks; need to restrict number of uses of the account
- Literature: PAKE: Password Authenticated Key Establishment

51

51

Station to Station protocol (STS)

- The problem can be fixed by adding digital signatures
- This protocol plays a very important role on the Internet (under different names)

SK_A, PK_B — SK_B, PK_A

choose x — α^x →

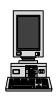
← α^y — *choose* y

$k=(\alpha^y)^x$ — *SigA*(α^x || α^y) → — $k=(\alpha^x)^y$

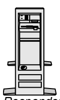
√*SigB* — ← *SigB*(α^y || α^x) — √*SigA*

52

IKE - Main Mode with Digital Signatures



Initiator



Responder

proposed attributes →

← selected attributes

g^x, N_i →

← g^y, N_r

K derived from master = prf(N_i || N_r, g^{xy})

SIG_i = Signature on H(master, g^x || g^y || ... || ID_i)

E(K, ID_i, [Cert(i)], SIG_i) →

SIG_r = Signature on H(master, g^y || g^x || ... || ID_r)

← E(K, ID_r, [Cert(r)], SIG_r)

H is equal to prf or the hash function tied to the signature algorithm (all inputs are concatenated)

53

Key transport using RSA

generate k, $E_{PK_B}(k)$ — $E_{PK_B}(k)$ → — *decrypt using* SK_B *to obtain* k

- How does Bob know that ***k*** is a fresh key?
- How does Bob know that this key ***k*** is coming from Alice?
- How does Alice know that Bob has received the key ***k*** and that Bob is present (entity authentication)?

54

Key transport using RSA (2)

generate k

$E_{PKB}(k)$

$E_{PKB}(k \| t_A)$ →

decrypt using SK_B *to obtain* k

- Freshness is solved with a timestamp t_A

55

Key transport using RSA (3)

generate k

$Sig_{SKA}(E_{PKB}(k \| t_A))$ →

decrypt using SK_B *and verify using* PK_A

- Alice authenticates by signing the message
- There are still attacks (signature stripping…)

56

Key transport using RSA (4): X.509

generate k

$Sig_{SKA}(B \| t_A \| E_{PKB}(A \| k))$

$\| t_A \| E_{PKB}(A \| k)$ →

decrypt using SK_B *and verify using* PK_A

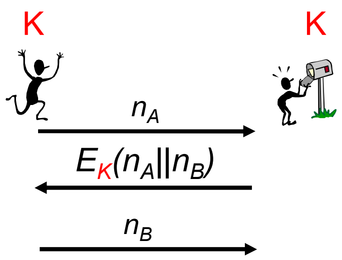
Mutual: B can return a similar message including part of the first message

Problem (compared to D-H/STS): lack of **forward secrecy**

If the long term key SK_B of Bob leaks, all past session keys can be recovered!

57

A simple protocol

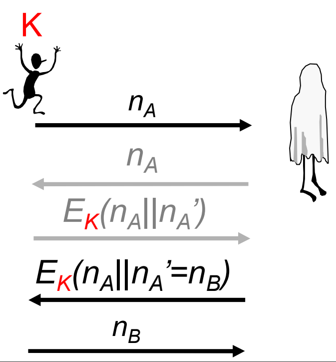


58

58

Reflection attack

Eve does not know K and wants to impersonate Bob



59

59

Conclusions

- Properties of protocols are subtle
- Many standardized protocols exist
  - ISO/IEC, IETF
- Difficulty: which properties are needed for a specific application
- Rule #1 of protocol design: **Don't**
  - not even by simplifying existing protocols

60

60

Recommended reading: entity authentication

- NIST Special Publication 800-63 Version 1.0.2 (2006): Electronic Authentication Guideline: identifies four levels of assurance
  http://csrc.nist.gov/publications/nistpubs/800-63/SP800-63V1_0_2.pdf
- D. Balfanz, R. Chow, O. Eisen, M. Jakobsson, S. Kirsch, S. Matsumoto, J. Molina, P.C. van Oorschot: The Future of Authentication. IEEE Security & Privacy 10(1): 22-27 (2012)
- J. Bonneau, C. Herley, P.C. van Oorschot, F. Stajano: The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes. IEEE Symposium on Security and Privacy 2012: 553-567
- J. Lang, A. Czeskis, D. Balfanz, M. Schilder, S. Srinivas, Security Keys: Practical Cryptographic Second Factors for the Modern Web. Financial Cryptography 2016: 422-440
- R. Peeters, J. Hermans, P. Maene, K. Grenman, K. Halunen, J. Häikiö, n-Auth: Mobile Authentication Done Right. ACSAC 2017: 1-15

See http://csrc.nist.gov/publications/PubsSPs.html

for about 120 Special Publications (800 Series) from NIST on computer security and cryptography

61

61

Recommended reading: key establishment

- A.J. Menezes, P.C. van Oorschot, S.A. Vanstone, Handbook of Applied Cryptography, CRC Press, 1997. Chapter 12.
- C. Boyd, A. Mathuria, Protocols for Authentication and Key Establishment. Information Security and Cryptography, Springer 2010, ISBN 978-3-642-07716-6.
- H. Krawczyk, SIGMA: The 'SIGn-and-MAc' Approach to Authenticated Diffie-Hellman and Its Use in the IKE-Protocols. CRYPTO 2003: 400-425.

62

62