User Authentication

Paul Francis

CS619, Oct. 14 2004

This week's reading assignment

- "Man-in-the-Middle in Tunneled Authentication Protocols"
- A strange choice…
  - Represents a snapshot of a 15-year process that has taken place mainly in the standards communities
- Hard to understand without significant background knowledge
- Alternative was no reading assignment…
  - I felt this was better than nothing

Some history

- PPP (Point-to-point protocol) first standardized by the IETF in 1989
- Three primary goals:
  - Link protocol framing for multiple network protocols (mainly for dialup modems)
    - IP, ISO CLNP, Xerox NS IDP, DecNet IV, Appletalk, Novell IPX, . . .
  - Negotiate parameters (compression, etc.)
  - ***User authentication***
    - Initially username/password ***in the clear!!!***

1992: Better user authentication

- Challenge Handshake Access Protocol (CHAP)
- Designed to run over PPP
  - Basically a request/reply stop-and-go protocol
- Simple operation:
  - Authenticator and "peer" (client) share a secret ***S***
    - Often something weak like a user password
  - Authenticator sends the peer a random challenge ***C***
  - Peer calculates Response ***R = MD5(S | C)***, sends it to authenticator
  - Authenticator also computes ***R***, and compares with received response

Comments on CHAP

- Challenge ***C*** must be a good nonce (random and not repeated for a long time)
  - Otherwise snooping attacker could replay responses later
- User name transmitted in the clear (violates privacy)
- All other values transmitted in the clear
  - So subject to dictionary attack
- These weaknesses acceptable for the threat
  - Value of password was low---network access only
  - (PPP didn't provide encryption, that was job of IPsec)
  - Cost of snooping and dictionary attack high compared to value

Dictionary attack

- Attacker has a huge list of possible passwords
  - Common words and names, various misspellings, inserted digits, or funny capitalizations
- Snoop challenges and responses
  - Only works if challenge and response are in the clear
- Brute force, solve challenge with all passwords and look for match with response
- Cost of attack independent of the size of the challenge
  - Dependent only on the size of the password dictionary
- Bottom line: either secret must be a large random value, or the challenge/response must be encrypted with a good secret

PPP, CHAP, and roaming

- ISPs offering dial-up also want to offer roaming
  - Travel to any place in the world, dial a local number to access Internet service
- But, ISPs don't have global or even national reach
- Solution is for ISPs to team together to provide global coverage
- Users dial-up locally, reach a local ISP
- Local ISP authenticates user with user's home ISP
- Requires some way to extend PPP authentication across the Internet

RADIUS and iPass

- RADIUS is a AAA protocol
  - Authentication, Authorization and Accounting
- RADIUS is extensible (generic "TLV" format)
- Authentication protocols like CHAP map directly into RADIUS messages
- Companies like iPass provide global RADIUS routing service
  - User names in PPP take the form user@isp.com
  - Local ISP sends RADIUS request to iPass, which passes it on to the appropriate ISP
- There is an implicit trust relationship between iPass and the ISPs it serves
  - RADIUS does not provide transitive trust itself
  - A next-gen AAA called DIAMETER does…

PPP-RADIUS example

Meanwhile, back at the ranch…

- IEEE working on Wireless LAN (WLAN, 802.11)
- Wanted to provide privacy and access control at a level equivalent to that of a wired LAN
- Developed WEP (*Wired Equivalent Privacy*)
- Simple idea:
  - Access Point (AP) and ***all*** clients share a secret key
  - For access, clients encrypt a cleartext challenge sent by AP
  - Packets are encrypted both ways using shared key

Famously, WEP sucked

- Apparently designed by radio/protocol engineers without review from academic or crypto communities
  - No doubt concerned with cost of hardware…
  - Today's paper in fact is an example of the value of open/public specification of security protocols
- Bottom line:
  - With commonly available equipment, an attacker could drive up to a site, listen for a few hours more or less, and crack the secret key
  - Software to do this openly available
  - Details widely published…

Practical WLAN privacy solution

- Put the WLAN outside the firewall
- Require legitimate users to access network via (strong) VPN technology
  - This is actually not such a bad solution…

A little on VPNs

- Two main kinds: IPsec and PPTP
- IPsec developed by IETF
- PPTP is an extension of PPP developed by Microsoft
  - Point-to-Point Tunneling Protocol
- PPTP also had/has security problems
  - MS-CHAPv1 had serious issues
  - MS-CHAPv2 fixed most of them, but could still be broken
    - See suggested reading (Schneier and Mudge)
    - I don't know what has happened since then

State of affairs circa late '90s

- WEP clearly broken
- WEP not at all designed for "hotspot" type service
  - Requires per-user, per session authentication and privacy
- IEEE802 requested help from IETF on this, but . . .
- CHAP designed under the assumptions of moderate physical protection and minimal value
- Also, CHAP does not authenticate the network, only the user
  - Trivial to setup a rogue hotspot, so user and network must mutually authenticate each other

State of affairs circa late '90s

- PPTP perhaps has many of the right attributes
  - Mutual authentication, privacy, per-session keys
  - But gives business advantage to Microsoft
  - Somewhat tied to PPP…
- IPsec assumes IP connectivity
  - But for WLAN hotspots, we want to authenticate before we assign an IP address (scarce, and therefore DDoS weakness)

New Auth protocols needed…

- Decoupled from PPP
- Provides mutual authentication
- Not subject to Man-in-the-middle (mitm) attack
  - Or any other attack!
- Allows for a wide range of authentication techniques
  - Legacy, proprietary, new ones . . .
  - Password, certificate, pre-shared secret . . .
- Leverage huge existing AAA infrastructure
- Provides per-session keys to encryption machinery
  - Client and Access Point

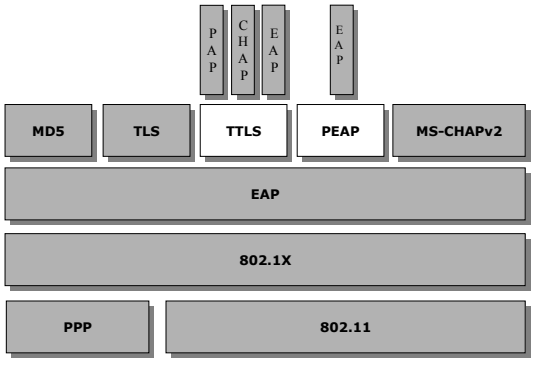
New auth key ideas

- Decouple authentication message *transport* from PPP
  - Define ***EAP*** (Extensible Authentication Protocol)
    - Extensible in part because can "pass through" boxes that understand EAP but not the auth method being used
    - Box knows where to send EAP message, and can tell that auth was granted
  - Runs over any transport
    - (certainly links like Ethernet or PPP, but also transports like TCP or UDP!)
  - Define how to run various auth methods over EAP (CHAP etc.)
    - Thus plugs into extensive RADIUS backend
- Utilize ***tunneled authentication*** to obtain mutual authentication from well-understood legacy user-auth protocols

Tunneled Authentication

- Simple idea:
- First run TLS (Transport Layer Security) between the client and the network
  - This authenticates the network
  - And generates a secret for the subsequent cipher stream
- Then, run CHAP within private TLS tunnel
  - This authenticates the user
- Strong parallel with secure web sites
  - TLS authenticates web site and creates secure transport
  - Then website asks for user name and password

EAP Architecture *



* Stolen from student talk from the Hebrew Univ of Jerusalem

EAP certainly is extensible!

Shared Key Methods
- 2.1 EAP-MD5
- 2.2 EAP-Cisco Wireless
- 2.3 EAP-SIM
- 2.4 SRP-SHA1
- 2.5 EAP-AKA
- 2.6 MS-EAP-Authentication
- 2.7 EAP MSCHAP-V2
- 2.8 EAP-HTTP Digest
- 2.9 EAP-SPEKE
- 2.10 EAP-FAST
- 2.11 EAP-Archie
- 2.12 EAP-GSS
- 2.13 EAP-IKEv2
- 2.14 EAP-LDAP
- 2.15 EAP-MD5 Tunneled Authentication Protocol
- 2.16 EAP-PSK
- 2.17 EAP-SKE
- 2.18 EAP-SSC

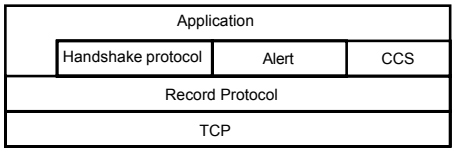
Non-Shared Key Methods
- 4.1 EAP-OTP
- 4.2 EAP-GTC
- 4.3 EAP RSA Public Key Authentication
- 4.4 EAP-DSS
- 4.5 EAP-KEA
- 4.6 EAP-TLS
- 4.7 Defender Token (AXENT)
- 4.8 RSA Security SecurID EAP and SecurID EAP
- 4.9 Arcot systems EAP
- 4.10 EAP-TTLS
- 4.11 Remote Access Service
- 4.12 EAP-3Com Wireless
- 4.13 PEAP
- 4.14 EAP-MAKE
- 4.15 CRYPTOcard
- 4.15 CRYPTOcard
- 4.16 DynamID
- 4.17 Rob EAP
- 4.18 MS-Authentication TLV
- 4.19 SentriNET
- 4.20 EAP-Actiontec Wireless
- 4.21 Cogent systems biometrics authentication EAP
- 4.22 AirFortress EAP
- 4.23 Securesuite EAP
- 4.24 DeviceConnect EAP
- 4.25 EAP-MOBAC
- 4.26 ZoneLabs EAP (ZLXEAP)
- 4.27 EAP Bluetooth Application
- 4.28 EAP-GPRS
- 4.29 EAP support in smart cards
- 4.30 EAP-TLS SASL

From draft-bersani-eap-synthesis-sharedkeymethods-00.txt

TLS (Transport Layer Security)

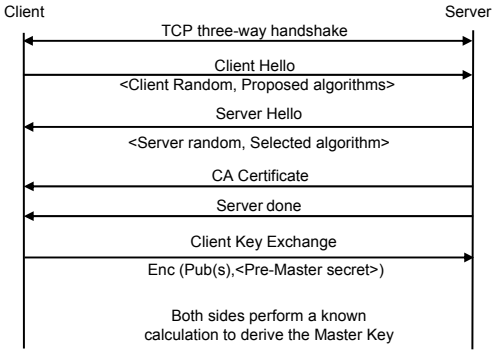
- Original version (SSL) designed by Netscape for secure HTTP
  - IETF version called TLS
- Runs over TCP
- Based on public key encryption using certs
  - Web browsers shipped with certs of major CAs (Certificate Authorities)
  - Therefore, can locally validate certs of web sites
- Client authenticates server
  - Though mutual authentication is possible

Quick summary of TLS*

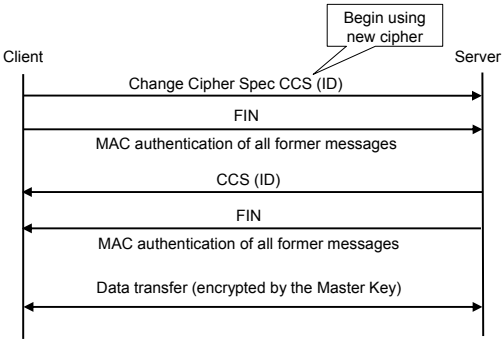


* Stolen from student talk from the Hebrew Univ of Jerusalem

Quick summary of TLS (2)



Quick summary of TLS (3)

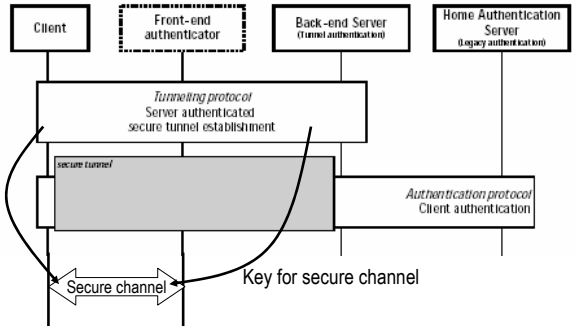


Now we get to the paper…

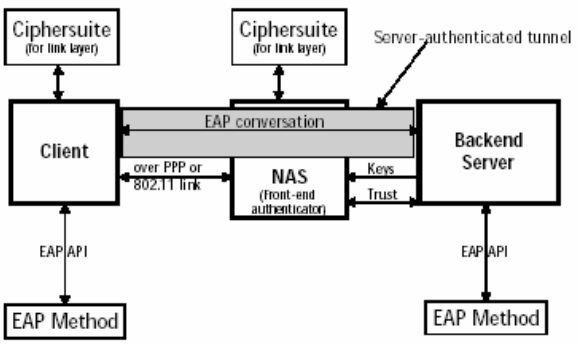
- "Man-in-the-Middle in Tunneled Authentication Protocols"
- Looked generally at tunneled auth protocols, and specifically at:
  - PEAP (Cisco/MS/RSA protocol)
  - EAP-TTLS (IETF, driven by Funk)
  - PIC (Uses EAP to create IPsec material?)
  - PANA over TLA (POTLS)
  - SLA (added by more recent version)
- But all these protocols have the same basic approach: use auth tunneling to maximize use of existing protocols and infrastructure

Tunneled authentication architecture

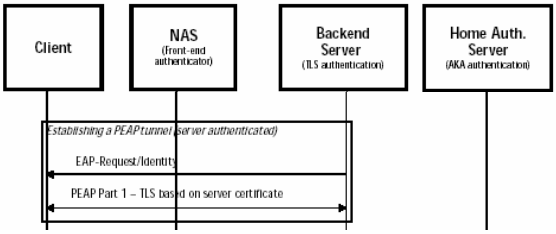
Basic idea: the component protocols are secure (i.e. against MITM) so the whole thing is secure



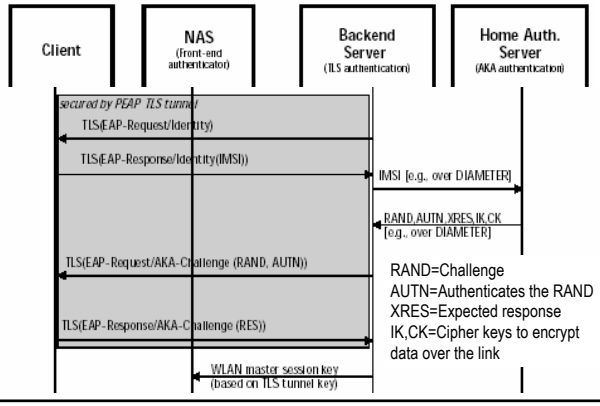
PEAP Example



PEAP with EAP AKA: TLS Part



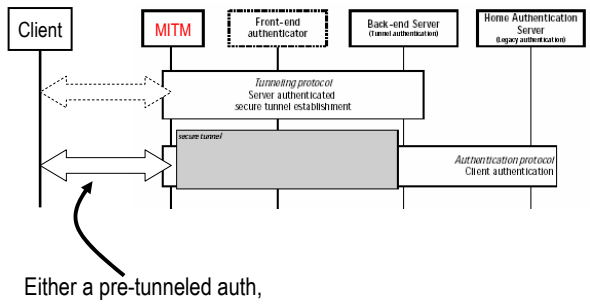
PEAP with EAP AKA: EAP Part



MITM Attack

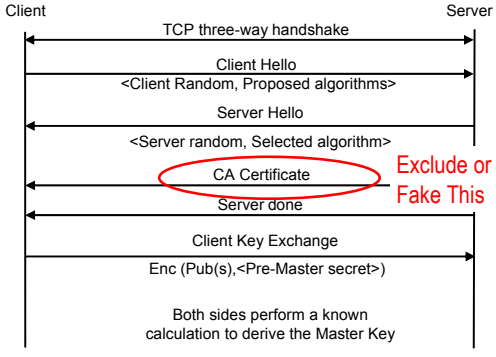
- Can occur under two conditions:
  - The client doesn't run the outer auth (TLS)
    - Because "negotiated down" to legacy protocol
  - The client run's the outer auth incorrectly
    - Because doesn't properly verify cert
- Are these conditions bogus???
  - No: net sec always involved a parameters/methods negotiation
    - Else can't evolve the protocols in the field
  - A classic MITM attack is to negotiate down, then break the weaker protocol
  - Also, sometimes cert acceptance is pushed up to the human user!!!!

MITM

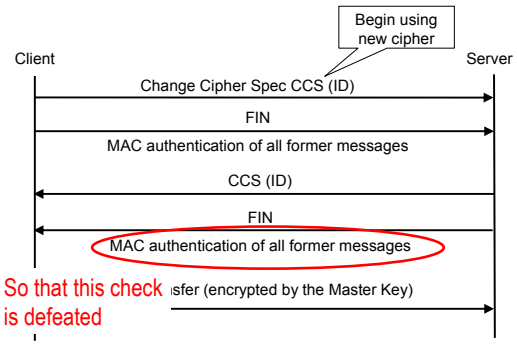


Either a pre-tunneled auth,
or tunneled auth with a bogus TLS cert

TLS slides again



TLS slides again

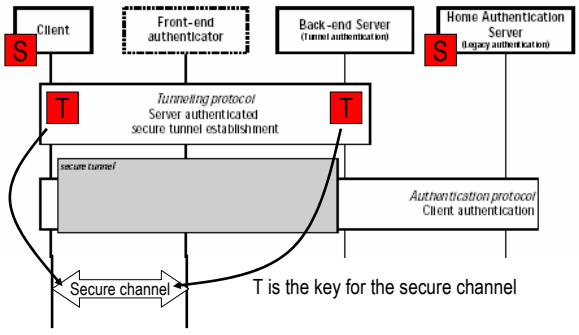


Avoiding the attack

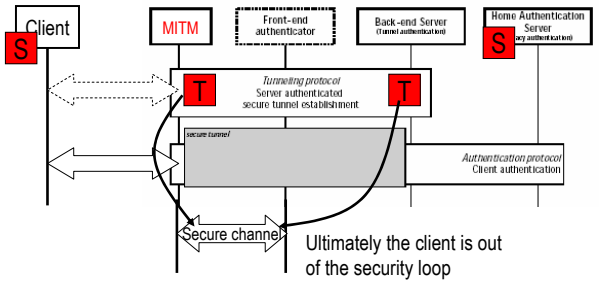
- Basic problem is that the "inner" (EAP) and "outer" (TLS) protocols are not bound in any way
  - Can't verify that the entity that ran one protocol also ran the other…
- Solution then, is to cryptographically bind the two protocols
  - Implicit or explicit

Lack of inner/outer binding

Pre-shared key S is not used to produce the shared channel, only TLS-derived master key T

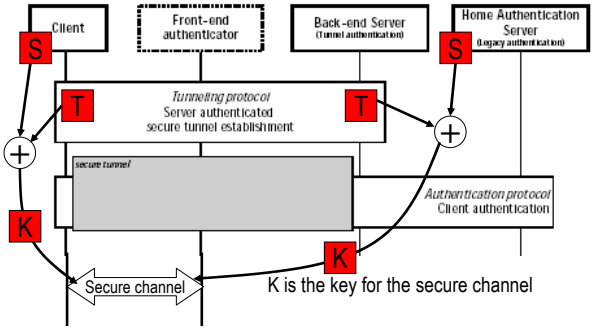


Lack of inner/outer binding



Bind inner and outer authentication

Note that the outer protocol no longer needed for authenticating the network per se if S is strong!



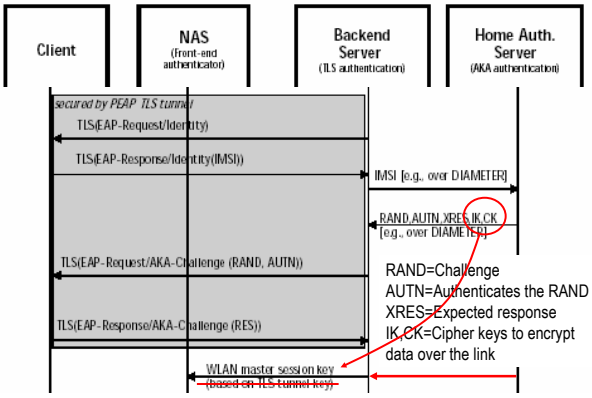
Implicit or Explicit binding

- Implicit:
  - Client and Network locally produce K from S and T
- Explicit:
  - Client and Network produce V from S and T
  - Both V's are sent to a trusted network box that can compare them
  - K can be produced any way (i.e. from T, as before)

EAP AKA uses neither of these

- Rather, AKA produces its own session keying material K from S
  - So this is used instead of T

PEAP with EAP AKA: EAP Part



Conclusions

- Combining otherwise secure protocols can introduce vulnerabilities
  - Understanding this important as need to re-use legacy protocols increases
- Always define security protocols in public
  - Today's paper an example of this
  - Still a problem if wider community doesn't care about a given protocol (i.e. researcher not given kudos for pointing out a problem)