ECE 443/518 – Computer Cyber Security
Lecture 13 Public Key Infrastructure

Professor Jia Wang
Department of Electrical and Computer Engineering
Illinois Institute of Technology

October 2, 2024

Outline

Public Key Infrastructure (PKI)

Secure Network Communication

Midterm Exam

- Lecture 1 $\sim$ Lecture 13, see Homework 1 and 2 for sample.
    - Points may be deducted if key steps are missing.
- Students registered for main campus section: Wed. 10/9, 11:25 AM – 12:40 PM, in class.
    - A physical calculator is allowed. Laptop or any other electronic device or calculator apps running on them are not allowed.
    - Closed book/notes. A letter-size page of cheat sheet is allowed.
- Online students may take the exam as above, or follow the instruction in the email "ECE 443/513 Exams for Online Section Students" from Charles Scott (scott@iit.edu).
    - Contact Scott and me if you cannot find the email.
    - No make-up exam will be offered if you fail to do so.
- ADA Accommodations: contact Center for Disability Resource (disabilities@iit.edu)
- Emergency/extraordinary reasons for make-up midterm exams are accepted only with documented proof like docter's notes.

Reading Assignment

- This lecture: UC 13
- Next lecture (10/14): Secure Collaborations

Outline

Public Key Infrastructure (PKI)

Secure Network Communication

Key Establishment using Public-Key Cryptography

- Consider RSA: use keys for both encryption and signature.
- For Alice to send k_{ses} to Bob,
  - $x = (k_{ses}, sig_{k_{pr,A}}(k_{ses}))$, then $y = e_{k_{pub,B}}(x)$.
  - Bob decrypts y first to get x and then verifies it.
- No PFS: k_{ses} is exposed if $k_{pr,B}$ is leaked.
- Double RSA is not efficient.

Efficient PFS Key Establishment

- Combine authentication with key exchange.
  - Both can be done via public-key cryptography.
- Authentication via digital signatures.
  - Alice: $k_{pub,A}$ and $k_{pr,A}$. Bob: $k_{pub,B}$ and $k_{pr,B}$.
  - A.k.a. authentication keys as these keys are never used for encryption.
- Apply key exchange to establish session key, e.g. DHKE.
  - Alice sends $(\alpha^a \bmod p, sig_{k_{pr,A}}(\alpha^a \bmod p))$ to Bob.
  - Bob sends $(\alpha^b \bmod p, sig_{k_{pr,B}}(\alpha^b \bmod p))$ to Alice.
  - After Alice and Bob both verify the signatures, they both compute $k_{ses} = \alpha^{ab} \bmod p$.
- No replay attack as long as a and b are randomly chosen.
- What about Man-in-the-Middle attacks?
  - Alice and Bob need to authenticate each other's public key.
  - How to create an authentic channel if Alice and Bob won't be able to meet each other?

Certificate Authority (CA)

- Another trusted third-party.
  - Make use of public-key cryptography: $k_{pub,CA}$ and $k_{pr,CA}$.
  - For digital signatures only.
- Everyone knows $k_{pub,CA}$ from an authentic channel.
  - To verify digital signatures from CA.
- How Alice proves to Bob $k_{pub,A}$ is from Alice?
  - Using an authentic channel, Alice sends $k_{pub,A}$ to CA and ask CA to sign $(k_{pub,A}, ID_A)$.
  - CA returns Alice her certificate: $Cert_A = ((k_{pub,A}, ID_A), sig_{k_{pr,CA}}(k_{pub,A}, ID_A))$.
  - Alice presents Bob $Cert_A$ that Bob can verify with $k_{pub,CA}$.
- If CA trusts Alice, CA may allow Alice to sign additional certificates using $k_{pub,A}$.
  - $Cert_A$ will need to include a field indicating so, and whoever certified by Alice should also present $Cert_A$.
  - Chain of Certificate Authorities (CAs)

Discussions

- There is still need for authentic channels.
  - Inevitable if we need to associate public keys to entities.
  - But we don't need $O(n^2)$ authentic channels between each pair of parties – we just need $O(n)$ of them between each party and CA.
  - However, this remains a very complicated matter in real world.
- CA doesn't need to be online.
  - No performance concern.
  - Much less chance of being compromised.
- While CA remains a single point of failure, it is less disastrous if compromised in comparison to KDC.
  - Only allow Man-in-the-Middle attacks.
  - If Alice has already authenticated Bob's public key and stored it, Man-in-the-Middle attacks could be even more difficult.

Outline

Public Key Infrastructure (PKI)

Secure Network Communication

TCP/IP Networking

- Most widely used networking protocols today.
- Layered structure: upper layers implement services using services provided by lower layers.
- IP Address: provide means to identify hosts
  - IPv4: 32 bits, usually quad-dotted like 216.47.143.249.
  - IPv6: 128 bits, very slowly adopted.
  - Special addresses: e.g. 127.0.0.1 (localhost).
  - Packet routing: store and forward communication
- TCP: transport layer protocol
  - Port: 16 bits for different applications on the same host
  - Communication as a reliable and ordered byte stream
- Domain Name System (DNS): application layer protocol
  - DNS query: map easy-to-memorize domain names, e.g. www.iit.edu, into numerical IP addresses.
  - Name servers: servers at well-known IP addresses that can answer DNS queries.

TCP/IP Security

- TCP/IP was designed to survive a nuclear war.
  - Not much against our passive and active adversaries.
- Security risks: here are a few
  - Fake Internet: a network that runs the same set of protocols but all important hosts are controlled by adversaries.
  - Eavesdropping: passive adversaries may see all packets passing through a router.
  - IP address spoofing: active adversaries may insert new packets with fake source addresses.
  - DNS spoofing: active adversaries may intercept and replace DNS query responses in order to redirect communication to a host controlled by adversaries.
- Network as a blackbox.
  - Well, we know that secure communications can be established over insecure channels.
  - TCP/IP networking can be made secure by introducing new services without affecting existing users.

HyperText Transfer Protocol (HTTP)

- An application protocol to transfer hypertext.
  - HTML files, etc.
  - Domain name is resolved by DNS.
  - On top of TCP, usually use port 80.
  - Request-response: clients (browser) request resources from servers.
- Foundation of data communication over World Wide Web.
  - Widely deployed and supported infrastructure: firewalls, proxies, content delivery networks, load balancers, etc.
- Not secure
  - Everything is in plaintext and there is no authentication.
  - One can insert something to a webpage during transmission.

Transport Layer Security (TLS)

- Successor of Secure Sockets Layer (SSL)
  - SSL has been deprecated because of security concerns.
  - However, the name 'SSL' remains in use, e.g. when mentioning TLS as TLS/SSL, or using Java API.
  - You should use TLS 1.1 or above, and avoid SSL 1.0,2.0,3.0, as well as TLS 1.0 .
- Provide confidentiality and integrity over TCP connections.
  - Client connects to server via TCP, then negotiates via a handshaking procedure to determine cipher parameters and to perform authentication and key establishment.
  - Finally the byte streams are protected by authenticated encryption and sent over the TCP transport.

TLS Authentication

- Via public key infrastructure (PKI).
- Server authentication
  - Server provides its certificate.
  - Client verifies the server certificate using the corresponding CA's public key.
- Client authentication
  - Server provides a list of CAs that it would trust.
  - Client provides one of its certificates that is signed by one of server's CAs.
  - Server verifies the client certificate using the corresponding CA's public key.
- Usually server authentication only.
- In either case, where did client or server get their CAs' public keys?
- What if we need to revoke server's or client's certificate if they lost their private keys?

More on PKI

- CA certificates (public key) distribution.
  - Usually as part of your OS installation.
  - Can be updated manually.
  - That's why you should only install OS from legitimate sources and why you should not give other people/software root access of your computer.
- Certificate revocation list (CRL)
  - Each certificate has an expiration date. An expired certificate won't be accepted.
    - Could attackers change that expiration date?
  - CAs will provide a list of all revoked certificates that are not expired, which should be refered when verifying certificates.
  - Clients and servers need to get this list on a timely basis.

HyperText Transfer Protocol Secure (HTTPS)

- A.k.a. HTTP over SSL or HTTP over TLS.
  - HTTP communication entirely on top of TLS (over TCP), usually use port 443.
  - Provide confidentiality and integrity.
  - Usually server authentication only, but client authentication could also be added.
- Domain name authentication
  - HTTPS server certificates need to include matching domain names and/or ip addresses for the connection to be considered secure by browsers.
  - Provide protection against IP address spoofing and DNS spoofing.
  - CA certificates can also be included with new browser installations – don't install browser from unknown sources!

HTTPS Issues

- HTTP or HTTPS?
  - It used to be costly to setup HTTPS websites as one need to buy certificates from known CAs.
  - Free certificates are widely available now due to awareness of security concerns and you should move your HTTP websites to use HTTPS.
  - Check website of Let's Encrypt.
- HTTPS only authenticate domain names
  - If someone attacks the web server to modify the web pages, HTTPS provides no protection.
  - This becomes even more tricky if content delivery networks (CDN) are used.

Summary

- TCP/IP network is not secure.
- But we can establish secure communication over it with proper system setup and choice of protocols.
    - Without breaking existing network infrastructure and applications.