WK04-WK09 Contents

WK04: Network Layer Security: IPsec

Why Network Layer Security?

Higher-layer security mechanisms do not necessarily protect an organisation's internal network links from malicious traffic.

If and when malicious traffic is detected at the end-hosts, it is too late,

- bandwidth has already been consumed.

Higher-layer security mechanisms (e.g., TLS) do not conceal IP headers.

- IP addresses of the communicating end-hosts visible to eavesdroppers.

Possible to create Secure VPN (more soon)

What is network-layer confidentiality ?

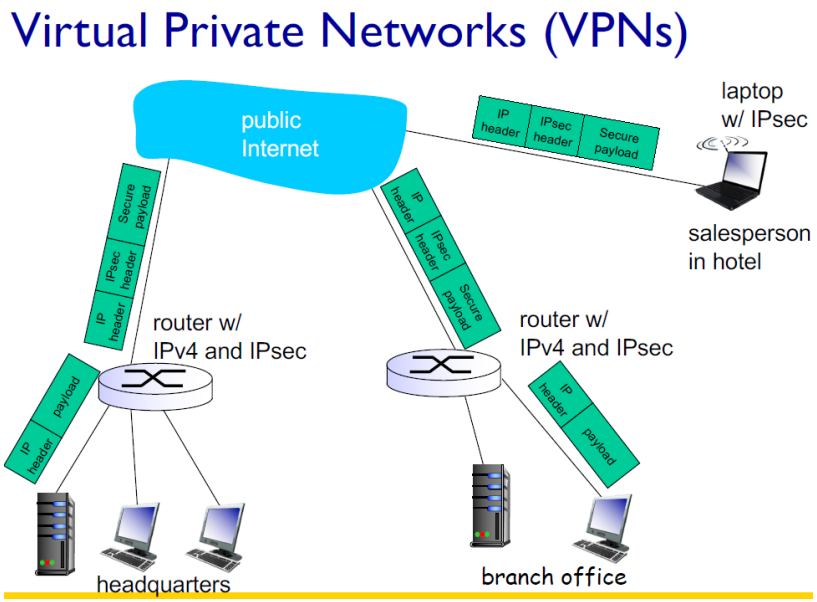
between two network entities:

- sending entity encrypts datagram payload, payload could be:
  - TCP or UDP segment, ICMP message, OSPF message ….
    - Implemented below Transport layer (TCP/UDP)
    - Even when implmented in end-systems, doesn't affect higher layers
- all data sent from one entity to other would be hidden:
  - web pages, e-mail, P2P file transfers, TCP SYN packets …
- "blanket coverage"

Virtual Private Networks (VPNs)

motivation:

- institutions often want private networks for security.
  - costly: separate routers, links, DNS infrastructure.
- VPN: institution's inter-office traffic is sent over public Internet instead
  - encrypted before entering public Internet
  - logically separate from other traffic



IPsec services

Data integrity

Origin authentication

Replay attack prevention

Confidentiality

Two different offerings (AH and ESP) discussed later

Tunnel and Transport Modes

Transport Mode: protects IP Payload received from layers above (Higher layer TCP, UDP, ICMP..)

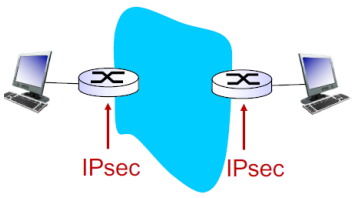
- There are many detailed nuances with AH/ESP support, encapsulation etc.

Tunnel Mode: All of IP including header etc is encapsulated, new IP header is added by Firewall/Routers.

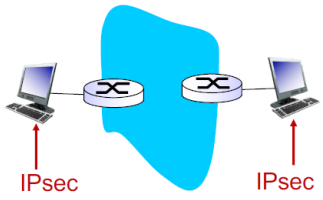
- End hosts behind firewall don't have to worry about IPSec
- We will focus on Tunnel Mode here as it is more widely used

Encapsulation: 封装，nuances:细微差别

IPsec – tunneling mode



- edge routers IPsec-aware
- IP address of Routers/Gateways used, destination address encrypted



- hosts IPsec-aware
- Also possible that one host is IPSec aware and other behind firewall

Advantages of tunnel mode (Edge)

Simple key distribution: fewer keys needed as gateways do encryption/decryption

Traffic analysis difficult as ultimate destination IP header concealed

Less processing burden on end-hosts

Conceal:隐藏

Two IPsec protocols (AH and ESP)

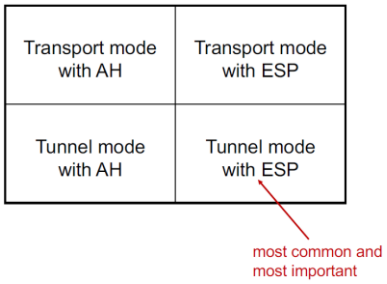
Authentication Header (AH) protocol

- provides source authentication & data integrity but *not* confidentiality

Encapsulation Security Protocol (ESP)

- provides source authentication, data integrity, *and confidentiality*
- more widely used than AH

Four combinations are possible!



Protocol Operations for ESP

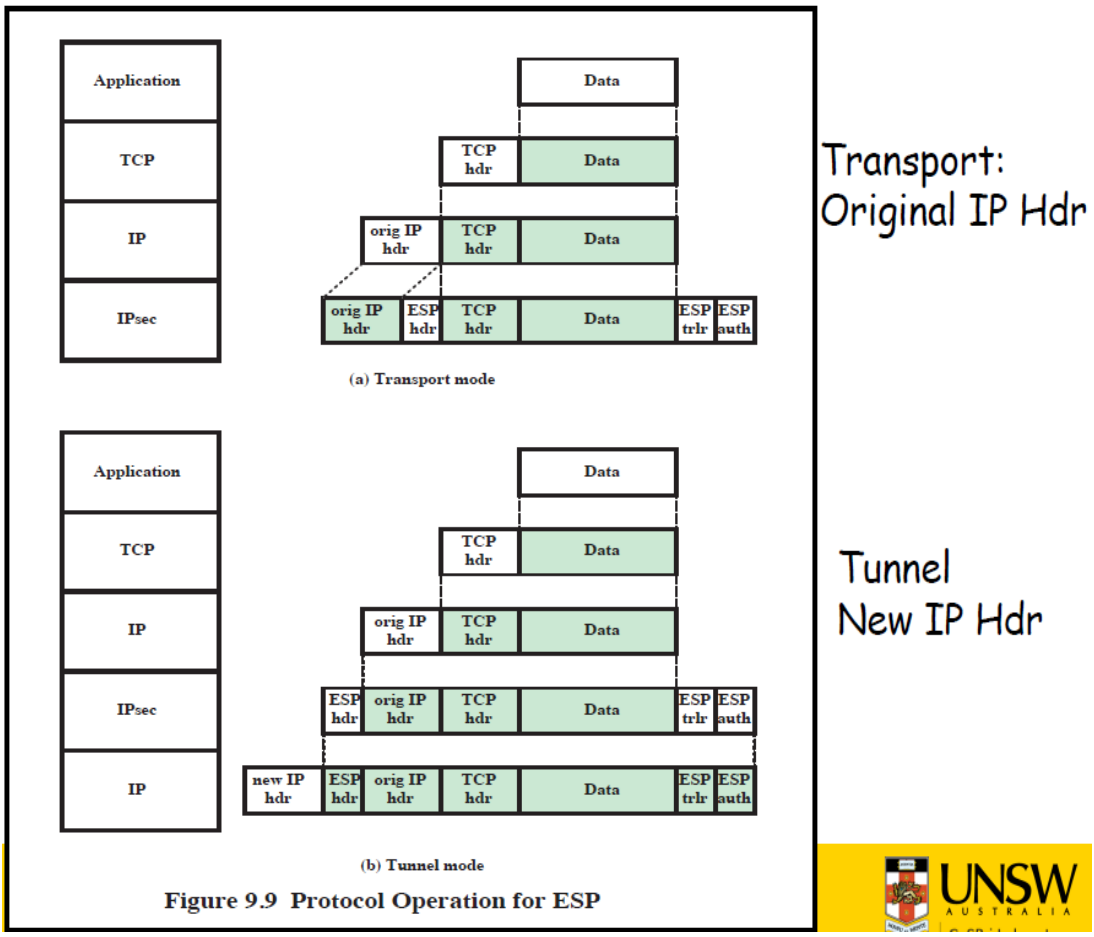


Figure 9.9 Protocol Operation for ESP

Security associations (SAs)

before sending data, "security association (SA)" established from sending to receiving entity

- SAs are simplex: for only one direction

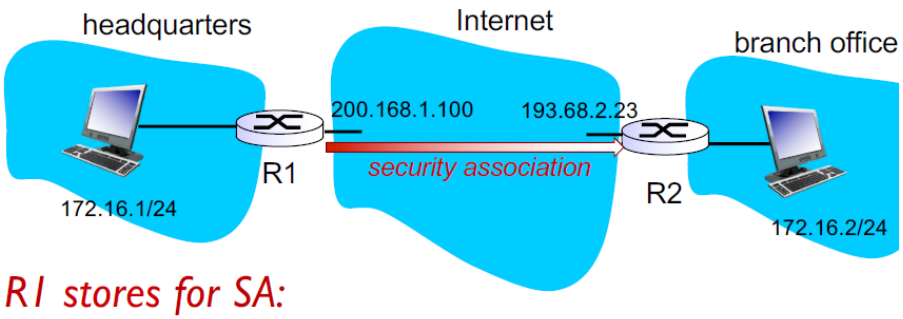
ending, receiving entitles maintain *state information* about SA

- recall: TCP endpoints also maintain state info
- IP is connectionless; ***IPsec is connection-oriented!***

Combination of Security Associations has many advanced features : Read stallings Chapter9 (not examinable)

Entitle:称做，名为

Example SA from R1 to R2



R1 stores for SA:

- 32-bit SA identifier: *Security Parameter Index (SPI)*
- origin SA interface (200.168.1.100)
- destination SA interface (193.68.2.23)
- type of encryption used (e.g., 3DES with CBC)
- encryption key
- type of integrity check used (e.g., HMAC with MD5)
- authentication key

Security Association Database (SAD)

endpoint holds SA state in *security association database (SAD)*, where it can locate them during processing.

when sending IPsec datagram, R1 accesses SAD to determine how to process datagram.

when IPsec datagram arrives to R2, R2 examines SPI in IPsec datagram, indexes SAD with SPI, and processes datagram accordingly.

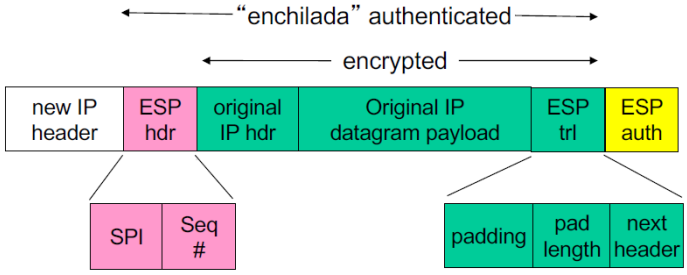
SAD Parameters

Normally defined by the following parameters in a SAD entry:

- Security parameter index
- Sequence number counter
- Sequence counter overflow
- Anti-replay window
- AH information
- ESP information
- Lifetime of this security association
- IPsec protocol mode
- Path MTU

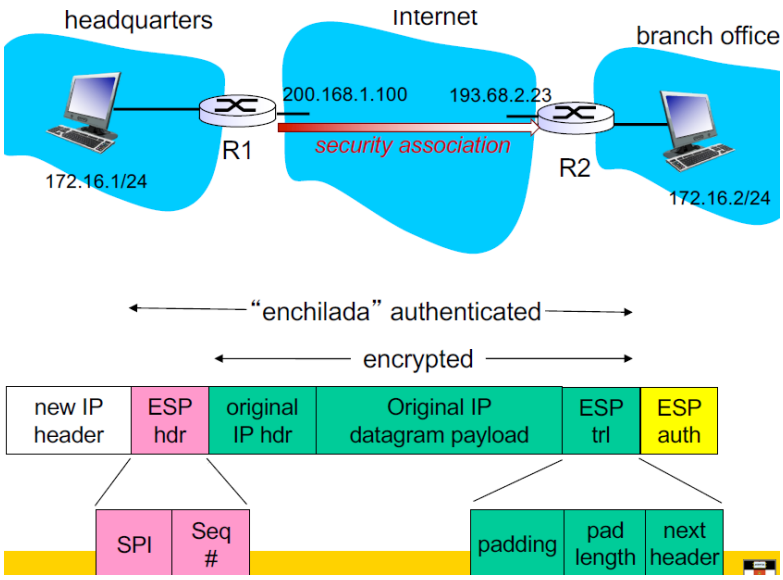
IPsec datagram

focus for now on tunnel mode with ESP



Note; Original IP address/hdr encrypted, destination Router/GW

Example:



R1: convert original datagram to IPsec datagram

appends to back of original datagram (which includes original header fields!) an "ESP trailer" field.

encrypts result using algorithm & key specified by SA.

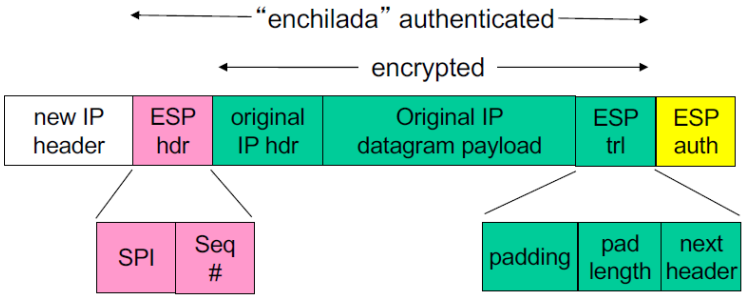
appends to front of this encrypted quantity the "ESP header, creating "enchilada".

creates authentication MAC over the *whole enchilada*, using algorithm and key specified in SA;

appends MAC to back of enchilada, forming *payload*;

creates brand new IP header, with all the classic IPv4 header fields, which it appends before payload.

Inside the enchilada:



ESP trailer: Padding for block ciphers

ESP header:

- SPI, so receiving entity knows what to do
- Sequence number, to thwart replay attacks

MAC in ESP auth field is created with shared secret key (HMAC)

- Note: ESP header is included in MAC calculation

IPsec sequence numbers

for new SA, sender initializes seq. # to 0

- each time datagram is sent on SA:
- sender increments seq # counter
- places value in seq # field

goal:

- prevent attacker from sniffing and replaying a packet
- receipt of duplicate, authenticated IP packets may disrupt service
- method:
  - destination checks for duplicates
  - doesn't keep track of *all* received packets; instead uses a window (remember from 3331)

Security Policy Database (SPD)

policy: For a given datagram, sending entity needs to know if it should use IPsec

needs also to know which SA to use

- may use: source and destination IP address; protocol number

info in SPD indicates "what" to do with arriving datagram

info in SAD indicates "how" to do it

Host SPD Example

Protocol	Local IP	Port	Remote IP	Port	Action	Comment
UDP	1.2.3.101	500	*	500	BYPASS	IKE
ICMP	1.2.3.101	*	*	*	BYPASS	Error messages
*	1.2.3.101	*	1.2.3.0/24	*	PROTECT: ESP intransport-mode	Encrypt intranet traffic
TCP	1.2.3.101	*	1.2.4.10	80	PROTECT: ESP intransport-mode	Encrypt to server
TCP	1.2.3.101	*	1.2.4.10	443	BYPASS	TLS: avoid double encryption
*	1.2.3.101	*	1.2.4.0/24	*	DISCARD	Others in DMZ
*	1.2.3.101	*	*	*	BYPASS	Internet

Src: Stallings. Table9.2

Table 9.2 provides an example of an SPD on a host system (as opposed to a network system such as a firewall or router). This table reflects the following configuration:

A local network configuration consists of two networks. The basic corporate network configuration has the IP network number 1.2.3.0/24. The local configuration also includes a secure LAN, often known as a DMZ, that is identified as 1.2.4.0/24. The DMZ is protected from both the outside world and the rest of the corporate LAN by firewalls. The host in this example has the IP address 1.2.3.10, and it is authorized to connect to the server 1.2.4.10 in the DMZ.

The entries in the SPD should be self-explanatory. For example, UDP port 500 is the designated port for IKE. Any traffic from the local host to a remote host for purposes of an IKE exchange bypasses the IPsec processing.

表9.2提供了主机系统上的SPD示例（与防火墙或路由器等网络系统不同）。此表反映了以下配置：

本地网络配置由两个网络组成。基本企业网络配置的IP网络号为1.2.3.0/24。本地配置还包括一个安全LAN，通常称为DMZ，标识为1.2.4.0/24。DMZ受到防火墙的保护，不受外部世界和企业局域网的其他部分的影响。本例中的主机的IP地址为1.2.3.10，它被授权连接到DMZ中的服务器1.2.4.10。

SPD中的条目应该是不言而喻的。例如，UDP端口500是IKE的指定端口。从本地主机到远程主机的用于IKE交换的任何通信都将绕过IPSec处理。

IPsec Outbound Processing

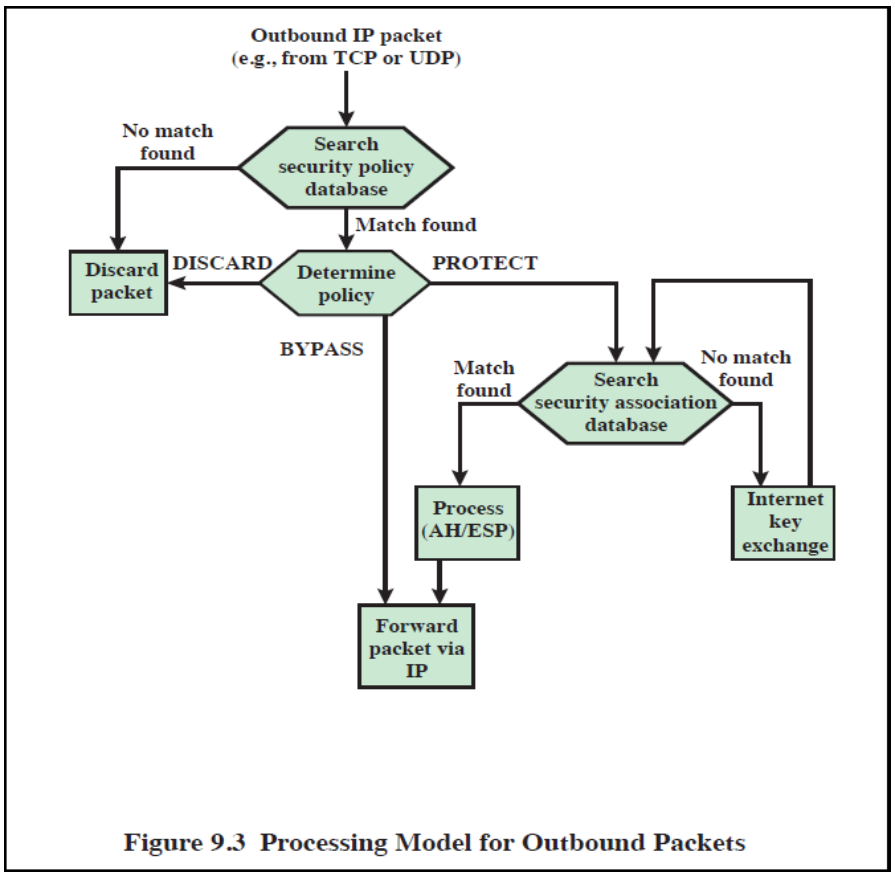


Figure 9.3 Processing Model for Outbound Packets

IPsec Inbound Processing

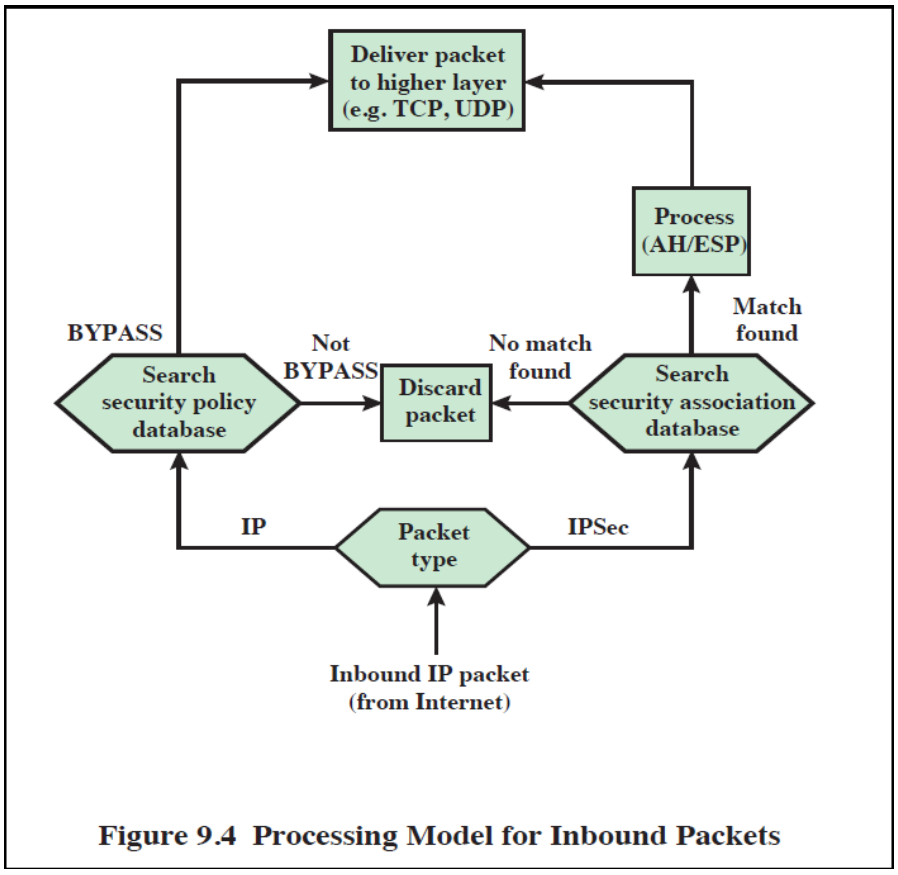
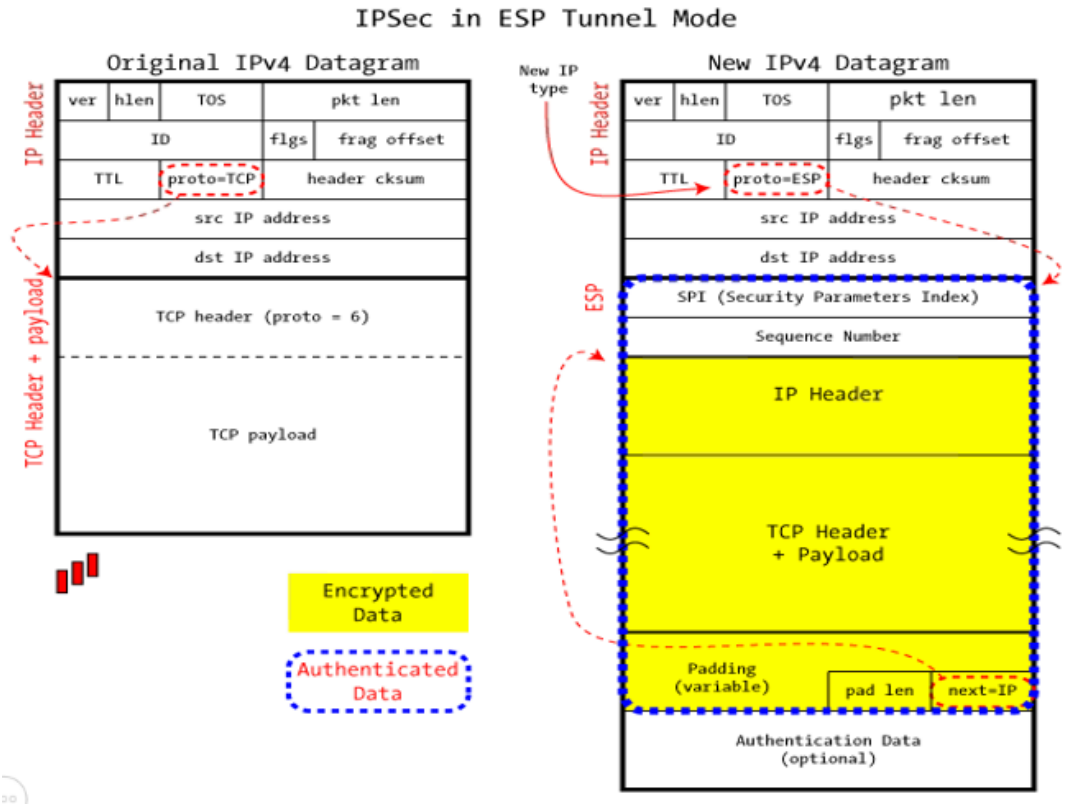


Figure 9.4 Processing Model for Inbound Packets

IPSec in ESP Tunnel Mode

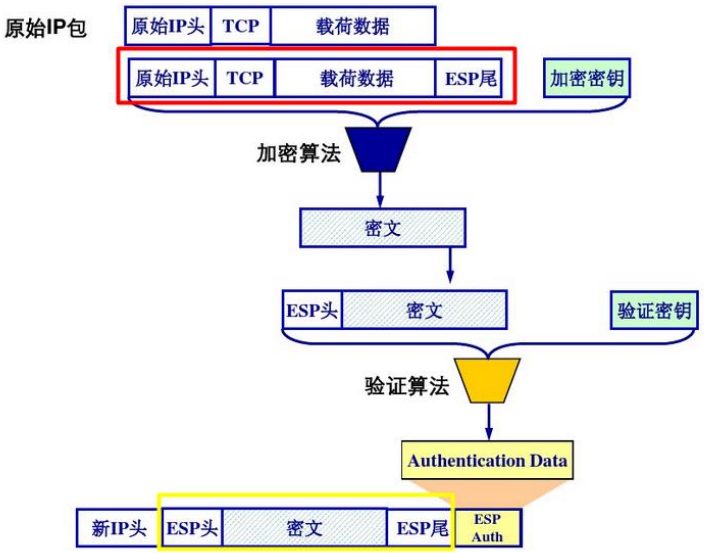
隧道模式下的认证和传输区域



1.原始的IPV4数据包IP header 中，proto是TCP，指向下一个的TCP header

2.新的IP header proto为ESP，然后ESP的trailer的Next Head为整个加密数据的IP头

红色区域是加密区域，整个数据包加ESP尾， 黄色区域是验证区域，这些区域内的数据是不能被修改的。



ESP数据包发送

1.使用分组的相应选择符（目的IP地址、端口、传输协议等）查找安全策略数据库

（SPD）获取策略，如分组需要IPSec处理，且其SA已建立，则与选择符相匹配的SPD项将指向安全关联数据库中的相应SA，否则则使用IKE建立SA。
2.生成或增加序列号
3.加密分组，SA指明加密算法，一般采用对称密码算法
4.计算完整性校验值
ESP数据接收
1.若IP分组分片，先重组
2.使用目的IP地址、IPSec协议、SPI进入SAD索引SA，如果查找失败，则丢弃分组
3.使用分组的选择符进入SPD中查找与之匹配的策略，根据策略检查该分组是否满足IPSec处理要求
4.检查抗重播功能
5.如SA指定需要认证，则检查数据完整性
6.解密

IKE: Internet Key Exchange

previous examples: manual establishment of IPsec SAs in IPsec endpoints:

Example SA

SPI: 12345
Source IP: 200.168.1.100
Dest IP: 193.68.2.23
Protocol: ESP
Encryption algorithm: 3DES-cbc
HMAC algorithm: MD5
Encryption key: 0x7aeaca…
HMAC key:0xc0291f…

manual keying is impractical for VPN with 100s of endpoints

instead use *IPsec IKE (Internet Key Exchange) RFC5996*

IKE（Internet Key Exchange）因特网密钥交换协议是 IPSEC 的信令协议，为 IPSec 提供了自动协商交换密钥、建立安全联盟的服务，能够简化 IPSec 的使用和管理，大大简化 IPSec 的配置和维护工作。IKE 不是在网络上直接传送密钥，而是通过一系列数据的交换，最终计算出双方共享的密钥，并且即使第三者截获了双方用于计算密钥的所有交换数据，也不足以计算出真正的密钥。IKE 具有一套自保护机制，可以在不安全的网络上安全的分发密钥，验证身份，建立 IPSEC 安全联盟。

IKE 的安全机制

IKE 具有一套自保护机制，可以在不安全的网络上安全的分发密钥、认证身份并建立 IPSec 安全联盟。完善的前向安全性（PFS：Perfect Forward Security）是一种安全特性，指一个密钥被破解， 并不影响其他密钥的安全性，因为这些密钥间没有派生关系。

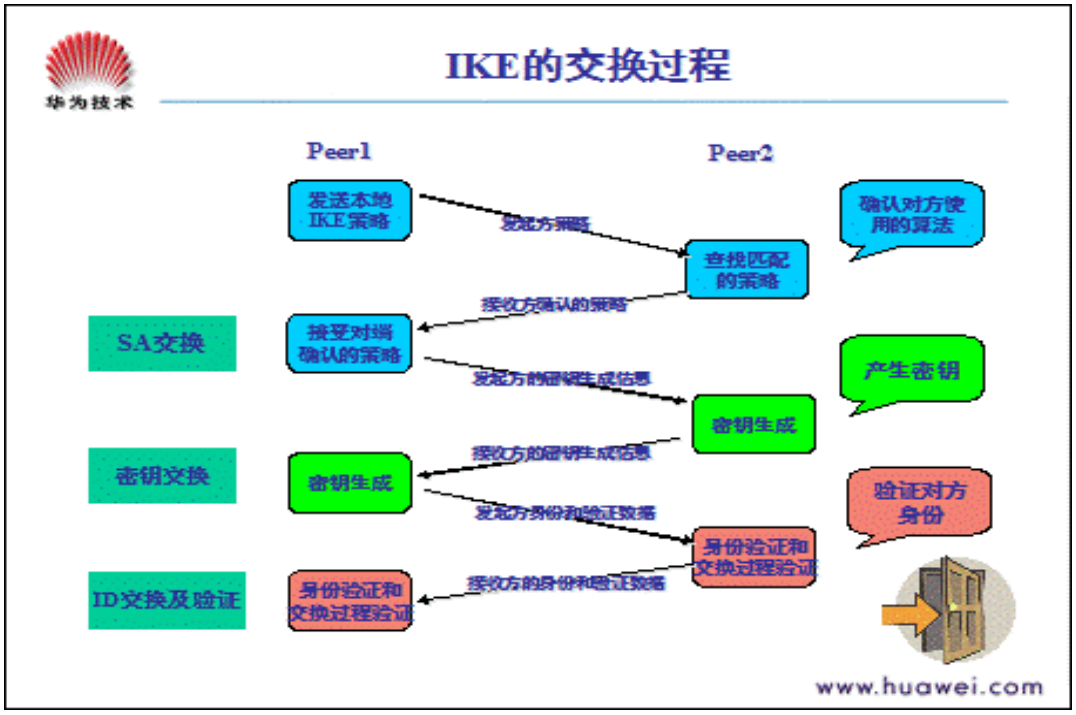
数据验证有两个方面的概念：
1）保证数据完整性（发送的数据未被第三方修改过）
2）身份保护

身份验证确认通信双方的身份。验证字用来作为一个输入产生密钥，验证字不同是不可能在双方产生相同的密钥的。验证字是验证双方身份的关键。身份数据在密钥产生之后加密传送，实现了对身份数据的保护。

DH 交换和密钥分发：Diffie-Hellman 算法是一种公共密钥算法。通信双方在不传送密钥的情况下通过交换一些数据，计算出共享的密钥。

PFS 特性是由 DH 算法保障的。

IKE 的交换过程



IKE 协商分为两个阶段，分别称为阶段一和阶段二。

阶段一：在网络上建立 IKE SA，为其它协议的协商（阶段二）提供保护和快速协商。通过协商创建一个通信信道，并对该信道进行认证，为双方进一步的 IKE 通信提供机密性、消息完整性以及消息源认证服务，是主模式；

阶段二：快速模式，在 IKE SA 的保护下完成 IPSec 的协商。

IKE 协商过程中包含三对消息：

第一对叫 SA 交换，是协商确认有关安全策略的过程；

第二对消息叫密钥交换，交换 Diffie-Hellman 公共值和辅助数据（如：随机数），加密物在这个阶段产生；

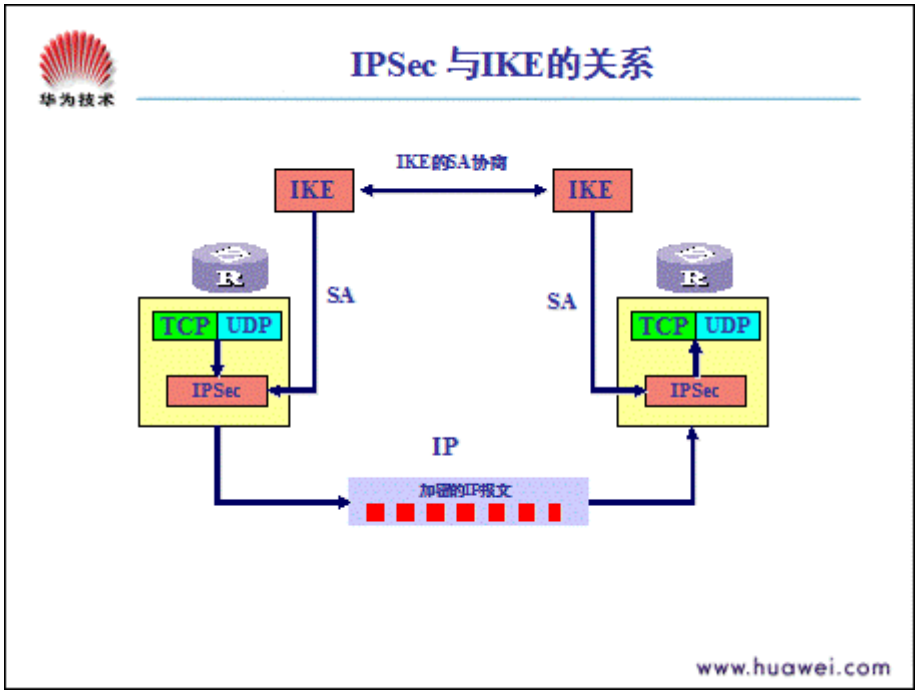
最后一对消息是 ID 信息和验证数据交换，进行身份验证和对整个 SA 交换进行验证。

IKE 在 IPSec 中的作用

- 降低手工配置的复杂度;
- 安全联盟定时更新;
- 密钥定时更新;
- 允许 IPSec 提供反重放服务;
- 允许在端与端之间动态认证;

因为有了信令协议，很多参数（如：密钥）都可以自动建立。
IKE 协议中的 DH 交换过程，每次的计算和产生结果都是毫无关系的。为保证每个安全联盟所使用的密钥互不相关，必须每次安全联盟的建立都运行 DH 交换过程。
IPSEC 使用 IP 报文头中的序列号实现防重放。此序列号是一个 32 比特的值，此数溢出后，为实现防重放，安全联盟需要重新建立，这个过程与要 IKE 协议的配合。
对安全通信的各方身份的的验证和管理，将影响到 IPSEC 的部署。IPSEC 的大规模使用，必须有 CA-Certification Authority（认证中心）或其他集中管理身份数据的机构的参与。

IPSec与IKE的关系



IKE 是 UDP 之上的一个应用层协议，是 IPSEC 的信令协议。
IKE 为 IPSEC 协商建立安全联盟，并把建立的参数及生成的密钥交给 IPSEC。
IPSEC 使用 IKE 建立的安全联盟对 IP 报文加密或验证处理。
IPSEC 处理做为 IP 层的一部分，在 IP 层对报文进行处理。AH 协议和 ESP 协议有自己的协议号，分别是 51 和 50。

IPsec summary

IKE message exchange for algorithms, secret keys, SPI numbers

either AH or ESP protocol (or both)

- AH provides integrity, source authentication
- ESP protocol (with AH) additionally provides encryption

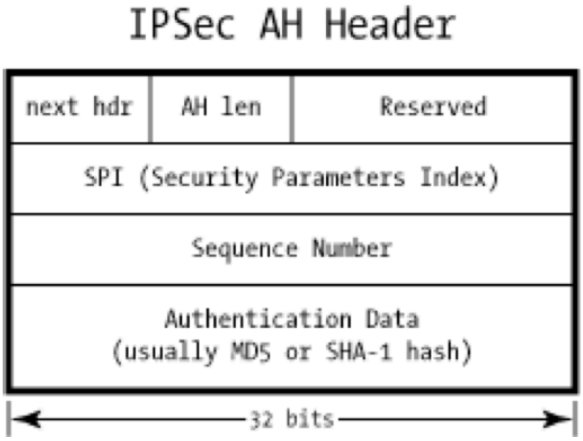
IPsec peers can be two end systems, two routers/firewalls, or a router/firewall and an end system

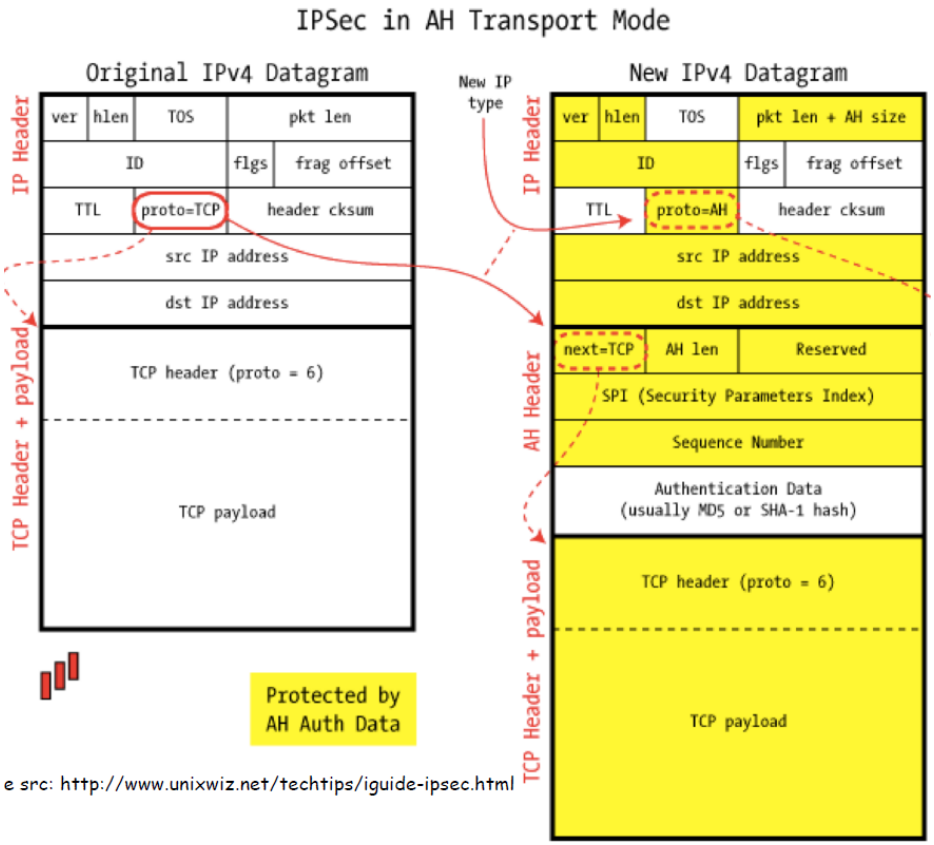
Authentication Header (AH)

AH provides ***authentication*** but not ***privacy***

a special hashing algorithm and a specific key known only to the source and the destination used to generate authentication header

Parts of the datagram used for the calculation, and the placement of the header, depends on the mode (tunnel or transport) and the version of IP (IPv4 or IPv6)





e src: http://www.unixwiz.net/techtips/iguide-ipsec.html

Tunnel Mode and Transport Mode Functionality

	Transport Mode SA	Tunnel Mode SA
AH	Authenticates IP payload and selected portions of IP header and IPv6 extension headers.	Authenticates entire inner IP packet (inner header plus IP payload) plus selected portions of outer IP header and outer IPv6 extension headers.
ESP	Encrypts IP payload and any IPv6 extension headers following the ESP header.	Encrypts entire inner IP packet.
ESP with Authentication	Encrypts IP payload and any IPv6 extension headers following the ESP header. Authenticates IP payload but not IP header.	Encrypts entire inner IP packet. Authenticates inner IP packet.

IKE Features

Each entity has a certificate (incl. Public Key)

Similarity with SSL handshake

- Exchange certificates
- Negotiate authentication and encryption algorithms
- Securely exchange Key Material for creating session keys in the IPSec SAs

IKE phases

IKE has two phases

- *phase 1:* establish bi-directional IKE SA *different* from IPSec SA
  - Authenticated and encrypted tunnel between two end points for IKE messages
  - Est. a Master Key for use in IPSec SA in phase 2
  - Another exchange for identity by signing their messages (Identities now protected by IKE SA, can't be sniffed)
    - » Also negotiated encryption/auth algorithms
- *phase 2: two sides* negotiate IPSec of SA in each direction
  - No PKI in second phase, hence large number of SA negotiation possible for scalability.

IKE有两个阶段

第一阶段：建立不同于ipsec sa的双向ike sa

－IKE消息的两个端点之间经过身份验证和加密的通道

-EST。在第2阶段的ipsec sa中使用的主密钥

－通过签署他们的信息来交换身份信息（现在由ike sa保护的身份信息，不能被嗅探）

还协商了加密/认证算法

第二阶段：双方在各个方向协商SA的ipsec

•第二阶段没有PKI，因此可以进行大量SA协商以实现可扩展性。

WK05: WLAN 802.1X Authentication

Overview

Security at Layer2

Authentication and Authorization in WLAN

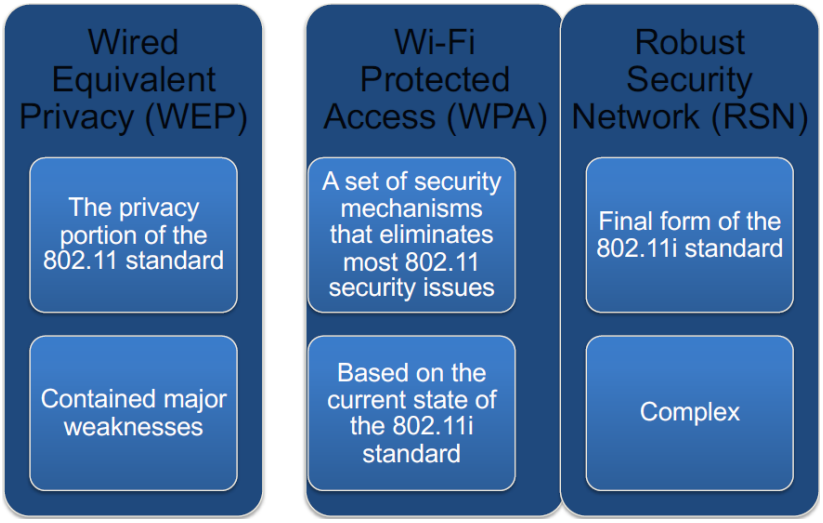
802.1X Extensible Authentication Protocol (EAP)

– Authentication and Authorisation for both wired and wireless network

Robust Security Network (RSN)/802.11i for Key Management

Authentication:证明 Authorization: 授权

WLAN Security Summary



Challenges for Enterprise

Pre Shared Key (PSK) not scalable

– Max 64 hex characters, configure manually in each device

E.g. 100 Employees, all share same Key.

One leaves the company

– Configure 99 devices with new key

We have learnt the vulnerability with WEP/WPA – and labs.

WPA2 provides CCMP/AES.

We learnt about SSL and IPSec

– Provide lot of flexibility/Option in configuring security at network and transport layers

Advanced Authentication Methods based on "Extensible Authentication Protocol (EAP)" – topic of this lecture

AAA

Authentication: verification of user identity and credentials
- May be multifactor: biometric etc.

Authorization: granting access to resources and services
- Needs authentication first.

Accounting: tracking network use by users
- Important to keep log
- Required by many industry regulators
- Helpful for billing/charging

Credential：证书

Accounting：跟踪用户使用的网络
- 记录很重要
- 许多行业监管机构要求
- 有助于计费/收费

Authentication in WLAN

Username and passwords

Digital Certificates

Dynamic/One Time passwords

Smartcards or credential on USBs

Machine authentication (based on embedded identity)

Pre-shared Keys (We saw WEP, WPA using this earlier)

Wi-Fi Protected Setup (WPS) – push button/Pin

WLAN Example of MF: A registered computer and a legitimate user which has entry in a DB e.g. A Microsoft Active Directory

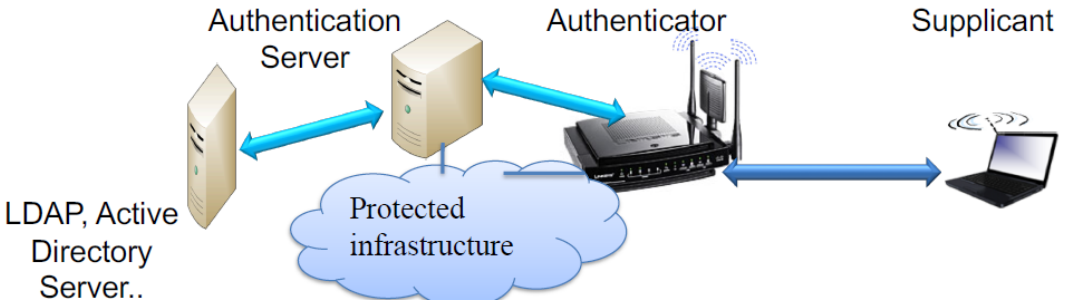
Various applications and higher layer protocols have their own authorization schemes.

WLAN can provide authorization via 801.X framework at Layer-2 (can be used with Robust Security Network (RSN))
- Port based access control (more later), for both wired and wireless network
- Lot of standard documents for various bits/pieces – not focus of this subject

Accounting is an important part but not within scope of WSN
- Useful for forensics though

IEEE 802.1X Port based Authentication

- Port Based: User must authenticate to switch they are physically connected to.
- Involves 3-party communications (nomenclature from 802.1X standard)
  - Supplicant
    - User
  - Authenticator
    - Ethernet switch, wireless access point
  - Authentication server
    - RADIUS (Remote access dial-in user service) database, Kerberos, LDAP or AD (Can be co-located with Authenticator)



nomenclature:命名 supplicant:请求者 dial:拨号

Supplicant

Device to be authenticated for resource use

Uses EAP protocol to connect to Auth. Server

Until identity verified – can't use higher layer protocols (3 – 7)

Can be software/app running 802.1X client

OS based supplicants:

- Microsoft Wireless Zero Conf – WZC
  - Known problems with supplicant software
- Apple's airport client

Chipset vendors may provide supplicant software

- Intel, Atheros, Broadcom

Chipset：芯片集 vendors：提供商

Authenticator (Access Point)

For EAP, authenticator acts as a relay between Supplicant and Auth. Server

Two Virtual Ports:

- Uncontrolled : allows EAP authentication traffic
- Controlled: Only authenticated traffic

With WLAN Bridging solution:

- Root bridge (a nominated bridge) is authenticator and other connected ones are supplicant

Configured with address of Authentication Server

- Possible co-location of Auth. Server with Authenticator
- Shared Secret with Auth. Server

Nominated:任命，指定

Authentication Server: RADIUS

RADIUS provides centralized authentication, authorization and accounting management for user/host to access a network service/resource

- Details in RFC 2865

Supports AAA (Authentication, Authorization and Accounting) – a.k.a " Triple A"

- RFC 3579 (AAA protocols such as RADIUS/EAP)
- RADIUS is used to shuttle RADIUS-encapsulated EAP Packets between authenticator and an authentication server

Most network equipment supports RADIUS

- Wireless AP, VPN appliance, SSL, etc.

Keeps an audit log of user's activity – accountability

Radius Server

- Standalone – local DB
- Use External DB – e.g Active Directory
- UDP Port 1812 for Auth, 1813 for Acct.

Any other server can also be directly used in place of RADIUS

Shuttle:穿梭； encapsulated:封装 audit:审计 standalone:独立的电脑

Radius Server and Authenticator configured with a shared secret.

Authenticator sends a RADIUS Access Request message to the RADIUS server, requesting authorization to grant access via the RADIUS protocol

- This request includes access credentials (e.g., username and password hash)
- Authentication server checks the credentials using the RADIUS server, Kerberos server, LDAP or Active Directory server
- returns one of three responses
  - Access Accept, Access Reject, Access Challenge for extra credentials

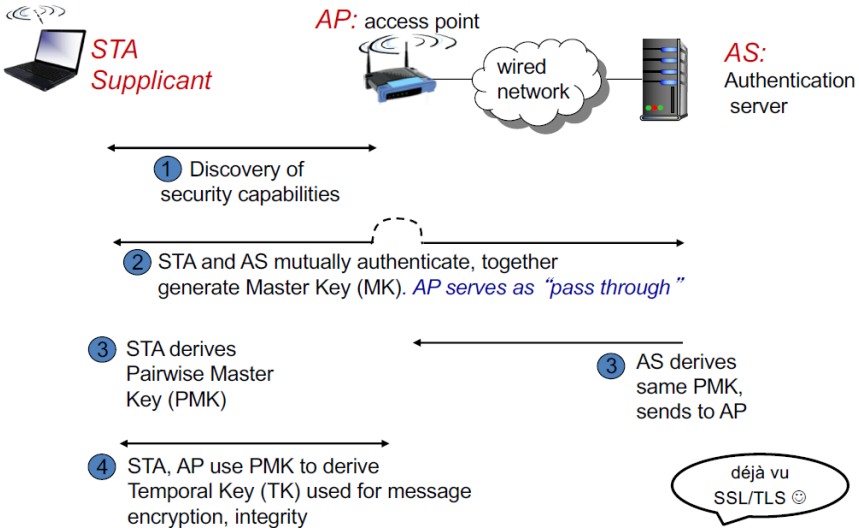
RADIUS server examples

Elektron (US$750) is an entry-level and user-friendly server

ClearBox (US$599) is designed for small networks, but it also scales to larger networks

FreeRADIUS (open source) is a solid and economical choice for Unix/Linux admins offering the most customization and flexibility

Four phases of operation (Short Story)

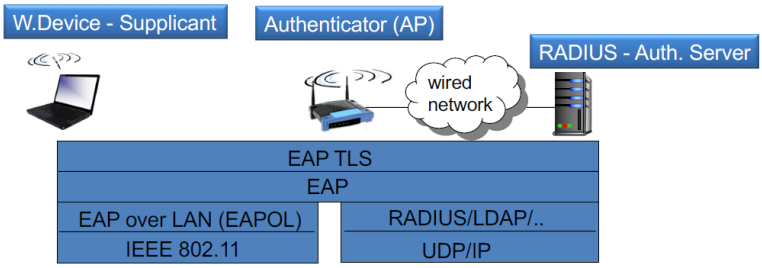


EAP: extensible authentication protocol

EAP: end-end client (mobile) to authentication server protocol

EAP sent over separate "links"

- Wireless Device-to-AP (EAP over LAN)
- AP to authentication server (RADIUS over UDP)



802.1X protocol - Long Story Continue

When a new client (supplicant) is connected to an authenticator, the port on the switch/wireless AP (authenticator) is enabled and set to the "unauthorized" state

- In this state, only 802.1X traffic is allowed
- Other traffic, such as DHCP and HTTP, is blocked at the data link layer
- Steps
  - Authenticator sends out the EAP-Request identity to the supplicant
  - Supplicant responds with the EAP-response packet that the authenticator forwards to the authenticating server
  - If the authenticating server accepts the request, the authenticator sets the port to the "authorized" mode and normal traffic is allowed
  - When the supplicant logs off, it sends an EAP-logoff message to the authenticator; the authenticator then sets the port to the "unauthorized" state, once again blocking all non-EAP traffic

当新客户机（请求方）连接到验证器时，交换机/无线AP（验证器）上的端口启用并设置为“未授权”状态。

- 在此状态下，只允许802.1X流量
- 其他流量（如DHCP和HTTP）在数据链路层被阻塞

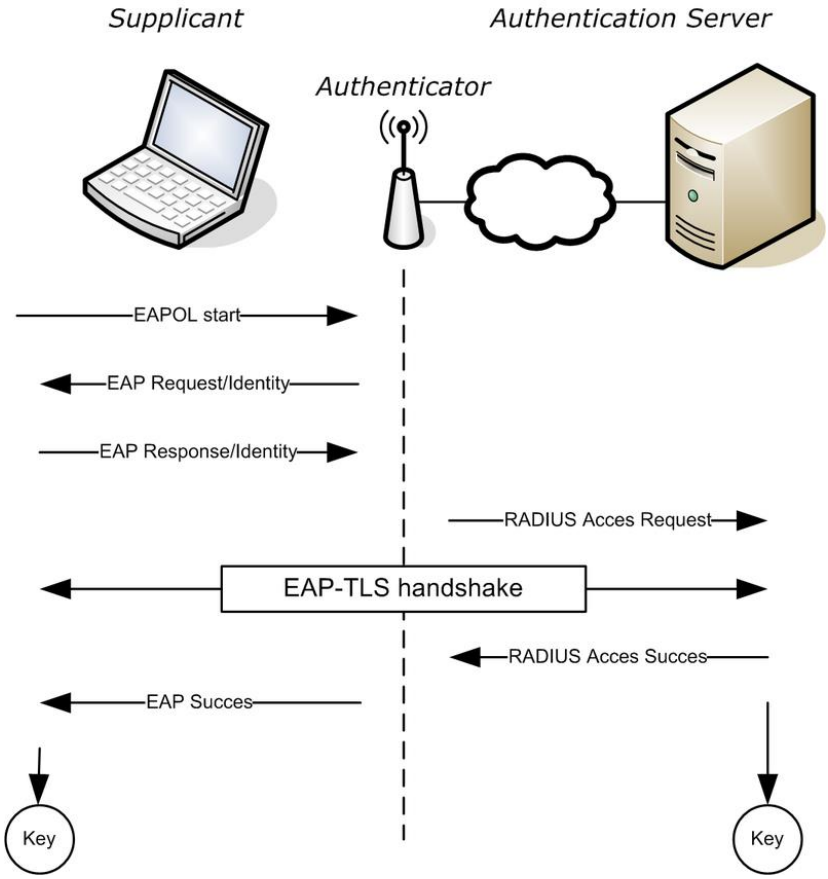
-步骤

o authenticator向请求方发送EAP请求标识

o请求方用认证方转发给认证服务器的EAP响应包进行响应。

o如果认证服务器接受该请求，则认证器将端口设置为“授权”模式，并允许正常通信。

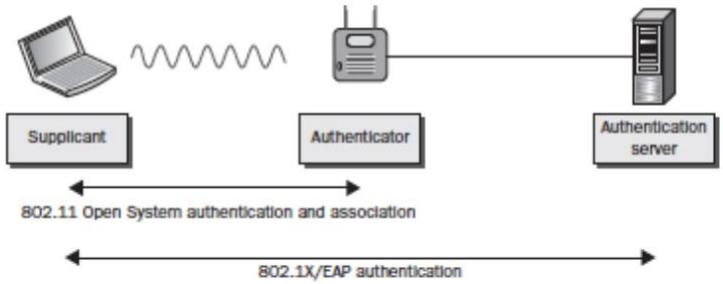
o当请求方注销时，它向验证器发送EAP注销消息；然后验证器将端口设置为“未授权”状态，再次阻止所有非EAP流量。



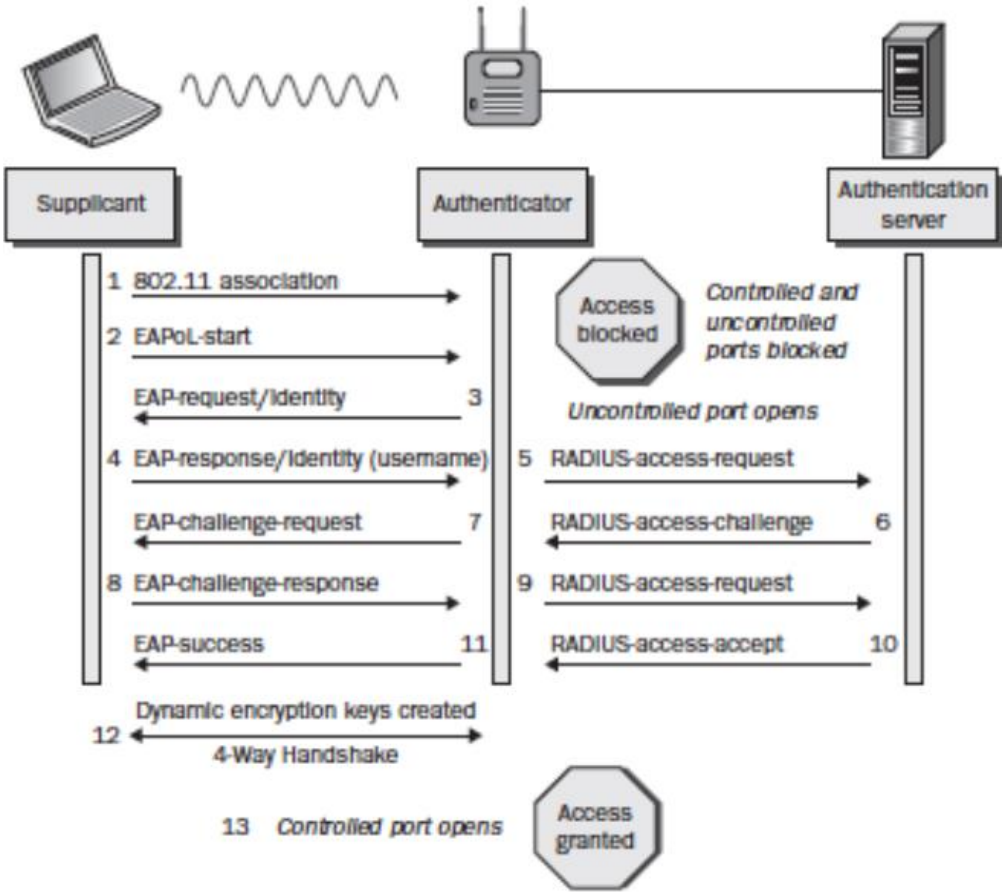
Association and EAP

First step is usual 802.11 association to establish L2 connection

If 802.1X framework used, network unusable unless the shown authorization process is complete.



Generic EAP Exchange



Notes on General Exchange

Most steps are self explanatory

Step4: Only identity is sent to the AP in clear text

- This allows for uncontrolled port to open

Step8: Supplicant doesn't send password, just MD5 (or other hash of password)

Step12: A complex process to generate dynamic encryption key

- uses 4 way handshake (see additional optional reading foils)

Step13: Controlled port is unblocked for the user.

- Proceeds to obtain an IP address using DHCP

Note: Step4 and 8 create security risk,

- hash algorithms can be cracked
- Dictionary attack possible
- Would it help to have an encrypted tunnel for these steps 4 -9?

Most schemes use tunneled authentication to pass identity credentials

Tunneling of EAP

EAP Methods defined in commonly used modern EAP standards include

- EAP-TLS (EAP-Transport Layer Security)
  - RFC 5216
- EAP-SIM (EAP for GSM Subscriber Identity)
  - RFC 4186
- EAP-AKA (EAP for UMTS Authentication and Key Agreement)
  - RFC 4187
- PEAP (Protected Extensible Authentication Protocol)
  - RFC 3748, (Microsoft Windows MS-CHAPv2
- EAP-FAST (Flexible Authentication via Secure Tunneling)
  - RFC 4851
- EAP-TTLS (EAP-Tunneled Transport Layer Security)
  - RFC 5281

No need to remember these acronyms or RFCs (some foils at then end are for optional reading

WK05 Guest Lecture: HTTPS seven years after DigiNotar

Transport Layer Security (TLS)

TLS: between transport layer and application layer

- Backbone of secure Internet communication
  - Secures HTTP, SMTP, IMAP, POP3, XMPP, LDAP, FTP, …
  - Used to create Virtual Private Networks (VPNs)
- Involves many different technologies:
  - Authenticated Encryption (AEAD)
  - **X.509**
  - **HTTP Pinning** and **Certificate Transparency**
  - DNS(SEC) & Co.: **CAA**, DANE

TLS problems developed over the years

Lesser-known flaws

- **Triple Handshake (renegotiation flaw)**
- SLOTH (weak hash functions)
- Logjam (precomputation of DLog due to export-grade crypto)
- DROWN (Bleichenbacher + cross-protocol attack (SSL2))

Plus vulnerabilities of implementations:

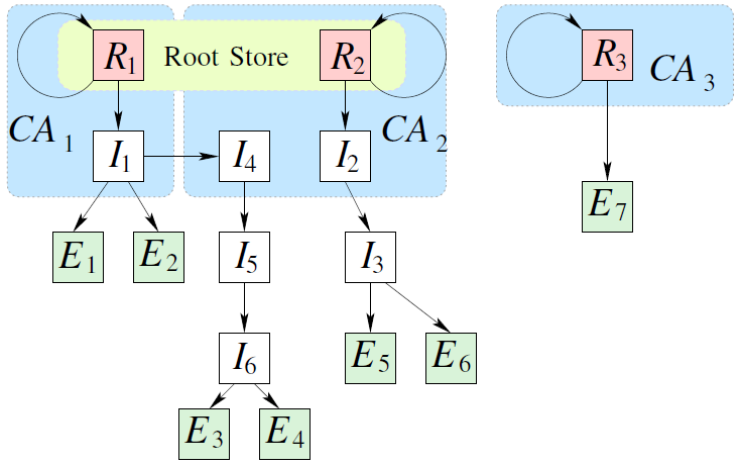
- Heartbleed (memory corruption)
- SMACK (bugs in popular implementations)
- FREAK (downgrade of cipher suite)
- Insecure nonce use in AES-GCM

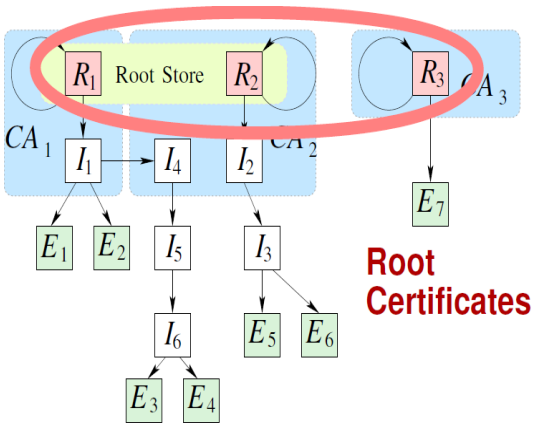
The X.509 Public Key Infrastructure (PKI)

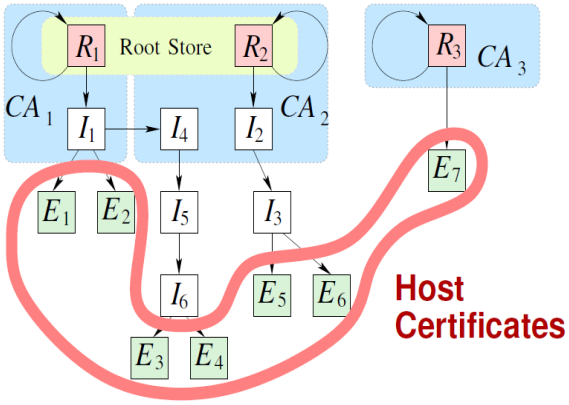
Much of our Internet security is built on X.509

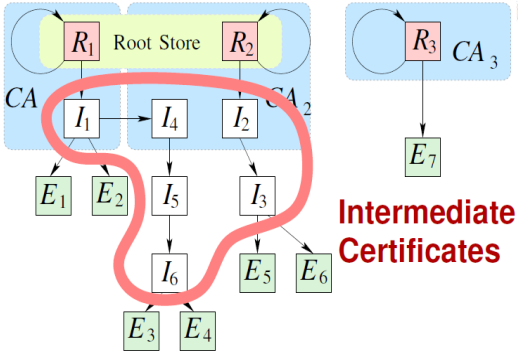
- Certificates bind an entity name to a public key
- Certification Authorities (CAs) act as certificate issuers
- Browsers/OSes preconfigured with CAs' 'root' certificates

Basic idea of X.509 PKI









PKI weaknesses in 2008

- Early December 2008:
  - 'Error' in Comodo CA: no identity check
  - Reported by Eddy Nigg of StartSSL (a CA)
  - A regional sub-seller just took the credit card number and gave you a certificate
  - No real reaction by Mozilla
- Late December 2008: whitehat hacks StartSSL CA
  - Technical report: simple flaw in Web front-end
  - Certificate for `mozilla.com` issued
  - Caught by 2nd line of defence: human checks for high-value domains

DigiNotar compromise

July 2011: DigiNotar CA hacked

Attacker claims to be the same one as in March

531 fake certificates, high-value domains

E.g., Google, Facebook, Mozilla, CIA, Mossad, Skype

Some hints pointed at Man-in-the-middle attack in Iran

The Netherlands' PKI was operated by DigiNotar...

For the first time, a Root CA is removed from a browser for being compromised

The (in)security of X.509

Since 2011, several studies have shown that:

- X.509 certificates are often wrongly issued
- X.509 PKI is poorly maintained
- Enormous dynamics, and huge attack surface

This is true for both Web and email

Examples:

- Holz et al., IMC 2011
- Durumeric et al., IMC 2013
- Amann et al., ACSAC 2013
- Durumeric et al., IMC 2015 (email)
- Holz et al., NDSS 2016 (email)

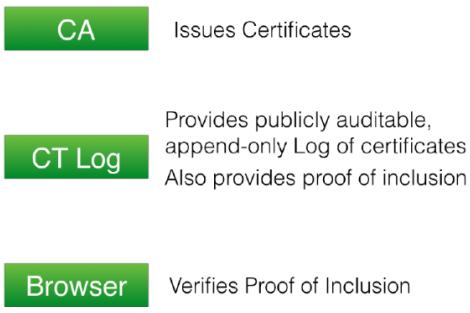
TLS/HTTPS security extensions

Introduced as a reaction to the attack(s)

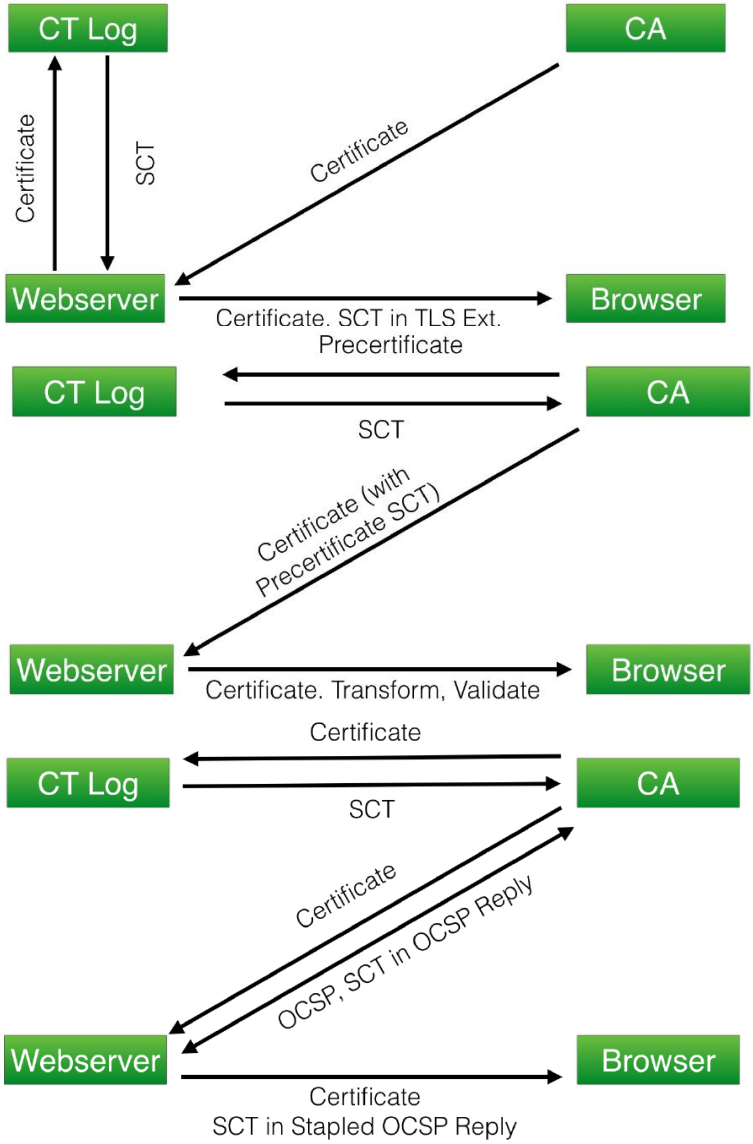
- Certificate Transparency: detect CA misbehaviour **fast**
- HTTP Strict Transport Security (HSTS)
- HTTP Public Key Pinning (HPKP)
- SCSV (TLS fallback signalling cipher suite)
- Certificate Authority Authorization (CAA)
- DNS-based authentication of named entities (DANE-TLSA)

Transparency：透明

Certificate Transparency



Auditable:可审计的；inclusion:包含,内容



任何人都可以将证书提交给日志，但大多数证书将由证书颁发机构和服务器运营商提交。当某人向日志提交有效的证书时，日志会使用签名证书时间戳（SCT）作出响应，这仅仅是在某个时间段内将证书添加到日志的承诺。时间段被称为最大合并延迟（MMD）。

MMD有助于确保日志服务器在合理的时间范围内将证书添加到日志中，并且不会阻止证书的颁发或使用，同时允许日志运行分布式服务器群以提供恢复能力和可用性。SCT在整个证书的整个生命周期都附带证书。特别是，TLS服务器必须在TLS握手期间向SCT交付证书。证书透明度支持三种方法来交付带有证书的SCT，这里只介绍两种：

TLS扩展

服务器运营商可以通过使用特殊的TLS扩展来提供SCT（见图2）。在这种情况下，CA向服务器运营商颁发证书，服务器运营商将证书提交给日志。日志将SCT发送给服务器运营商，并且服务器运营商使用具有类型的TLS扩展signed_certificate_timestamp 在TLS握手期间将SCT交付给客户端。（下图左边）

此方法不会更改CA颁发SSL证书的方式。但是，它确实需要更改服务器以适应TLS扩展。

OCSP装订

服务器运营商也可以通过使用联机证书状态协议（OCSP）订书机交付SCT（参见图2）。在这种情况下，CA同时向证书服务器和服务器运营商颁发证书。然后，服务器运营商向CA进行OCSP查询，并且CA用SCT进行

响应，在TLS握手期间服务器可以将其包括在OCSP扩展中。

该方法允许CA负责SCT，但不会延迟证书的颁发，因为CA可以异步获取SCT。但是，它确实需要修改服务器才能执行OCSP装订。（下图右边那种）

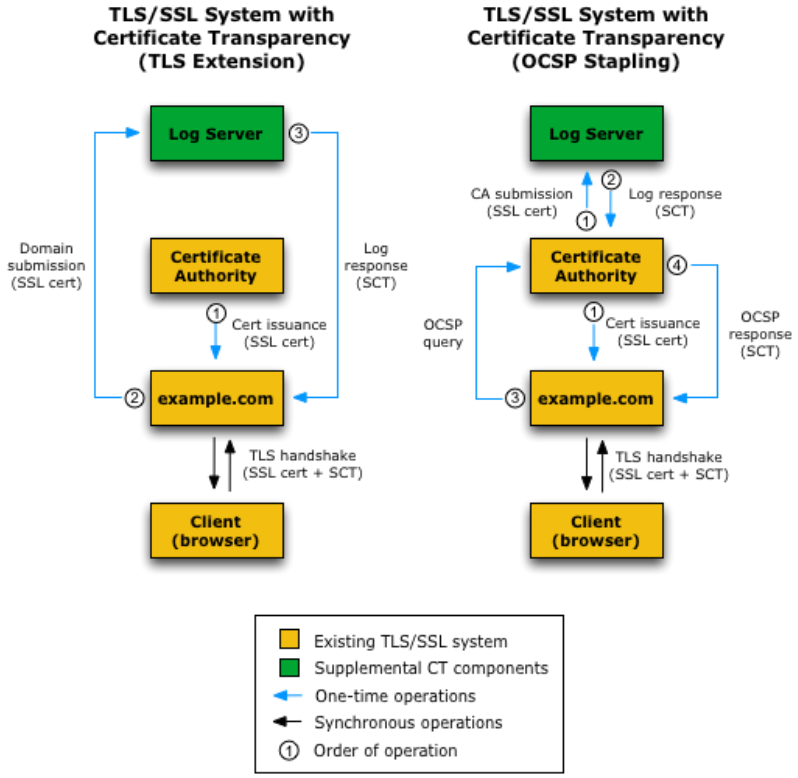


Figure 2

Methodology

Active & passive scans

- Shared pipeline where possible

Active measurements from 2 continents

- Largest Domain-based TLS scan so far
- More than 192 Million domains

Passive measurements on 3 continents

- More than 2.4 Billion observed TLS connections

Curious certificates

425 cases monitored at UCB

Exact clones of existing certificates

- CA signature does not check out

No SCT data - just 'random string'

For high-value domains

- Cloudfront
- Twitter

IPs belong to various hosters

Connecting actively to the IPs: no certs sent in the TLS handshake!

Cover traffic?

HTTP Strict Transport Security (HSTS)

- Dynamic 'pinning': tell clients that this site supports HTTPS
- Example:
  `Strict-Transport-Security: max-age=31536000; includeSubDomains`
- Instructs browser:
  - To expect HTTPS for next 12 months, including subdomains
  - To redirect on port 443
  - To disallow user override of certificate warnings
- Simple, powerful
- Very little danger for server operators to misconfigure (and lose customers)

Pinning:探针；override:推翻

HTTP Public Key Pinning (HPKP)

- Servers communicate life-time and hash value of their X.509 public key in the HTTP header
  - `Public-Key-Pins: pin-sha256="cUPcT..."; max-age=5184000; report-uri="https://www.example.net/hpkp-report"`
- Addresses short-comings of simple pinning:
  - Life-cycle management for key upgrade/compromise: 'backup pins' communicated in addition to the primary ones
- Misconfiguration leads to lock-out of site visitors!

（HPKP）是一个互联网安全通过交付机制HTTP 标头，它允许HTTPS网站抵制假冒使用不当签发或欺诈攻击数字证书。[1]它通过向客户端（例如Web浏览器）提供一组公钥来实现这一点，该公钥应该是未来连接到同一域名的唯一受信任者。例如，攻击者可能会破坏证书颁发机构，然后错误地为Web源颁发证书。为了克服这种风险，HTTPS Web服务器提供一段有效的给定时间的"固定"公钥哈希值；在后续连接中，在该有效时间内，客户端希望服务器在其证书链中使用一个或多个这些公钥。如果没有，则显示错误消息，该消息不能被用户（容易地）绕过。

HPKP was never effective

What about SCSV?

- Best result ever: 96% support
- Automatically introduced with library updates
- Auto-updates on browser side

SCSV是为了解决传统服务器的互操作性问题，因为许多TLS客户端实现不依赖于TLS协议版本协商机

制，而是在初始握手尝试失败时有意使用降级协议重新连接。这样的客户端可能会退回到它们宣布版本低至TLS 1.0（或甚至其前身的安全套接层（SSL）3.0））的连接，作为最高支持版本。
虽然这种后备重试可能是与实际遗留服务器连接的有用的最后手段，但主动攻击者可能会利用降级策略来削弱连接的加密安全性。此外，由于网络故障引起的握手错误也可能被误解为与旧服务器的交互，并导致协议降级。
所有不必要的协议降级是不希望的（例如，如果客户端和服务器实际上都支持TLS 1.2，则从TLS 1.2到TLS 1.1）;当结果是通过降级到SSL 3.0而导致TLS扩展功能的丢失时，它们可能特别有害。

Certificate Authority Authorization (CAA)

- Idea: define in a special CAA record, which CAA is allowed to issue certificates for your domain
  - CA must check this record before issuing
  - Per vote of the CA/Browser Forum, enforced since Sep 2017

Domain	Type	Flags	Tag	Value
tum.de	CAA	0	issue	"letsencrypt.org"
tum.de	CAA	0	issue	"pki.dfn.de"
tum.de	CAA	0	issuewild	";"
tum.de	CAA	0	iodef	"mailto:a@b"

Table 1: Exemplary CAA section of DNS zone file

idea：在一个特殊的CAA记录中定义允许哪个CAA为您的域颁发证书
- CA必须在发布前检查此记录
-自2017年9月起，CA/BROWSER论坛每次投票

CAA是一种安全措施，允许域所有者在其域名服务器（DNS）中指定哪些CA有权为该域颁发证书。如果CA收到具有CAA记录的域的证书订单，并且该CA未被列为授权颁发者，则禁止他们将证书颁发给该域或任何子域。

CAA的好处

CAA 的一个好处是补充证书透明度（CT）。CT 提供了一些机制，可以帮助域名所有者在发布后识别其域名的错误发布或频繁颁发的证书，而 CAA 可以帮助防止未经授权的发布。他们共同构建了一套比其中任何一个更好的安全性。CAA 还可以帮助已标准化或希望标准化或限制其使用的 CA 的组织。在 CAA 之前，组织没有一种简单的方法来实施此类策略，但现在所有 CA 都必须检查 CAA 记录，这些策略实际上可以由 CA 强制执行。

实施 CAA

虽然 CA 必须检查 CAA 记录，但对于域所有者，使用 CAA 是可选的。您可以自行决定是否要实施，如果您决定这样做，您可以根据需要指定多个 CA。以下是 CA 的处理规则：没有 CAA 记录：CA 可以发布；CAA 记录包括 CA：CA 可以发布；CAA 记录，但不包括 CA：CA 无法发出。CAA 支持以下属性:Issue：允许 CA 颁发证书（包括通配符证书，除非受 Issuewild 限制）；Issuewild：允许 CA 颁发通配符证书，但不允许非通配符证书。

End-to-end auditing

idea：使用历史、每日的 TLS 扫描和 CAA 记录来揭示过去发行的异常情况。
–限制:
–即使扫描之间，CAA 记录也会快速改变。

–拆分地平线视图：CA 可能收到了不同的 DNS 答复
–证书颁发时间是基于字段和 SCT 之前无效的估计值。

Summary: deployment

Deployment correlates with:
- Ease of configuration
- Low risk to availability

CT is very powerful—but would it be deployed if it wasn't by Google?

HSTS works, but HPKP is rightly deprecated

Auto-updates are powerful

Auditing by independent parties is useful—see CAA
- In fact, recommend to store CAA audit trail in CT

部署与以下内容相关：–易于配置 –低可用性风险
CT 非常强大，但如果不是谷歌，它会被部署吗？
–HSTS 有效，但 HPKP 被正确否决。 –自动更新功能强大 -独立方的审计很有用，见 CAA。–事实上，建议在 CT 中存储 CAA 审计跟踪

WK06: Operational Security: Firewalls and IDS

Overview

Firewall
- Stateless
- Stateful
- Application level gateways

IDS
- Host Based
- Network based

Snort

Firewalls

Internet connectivity is essential but is vulnerable to threats

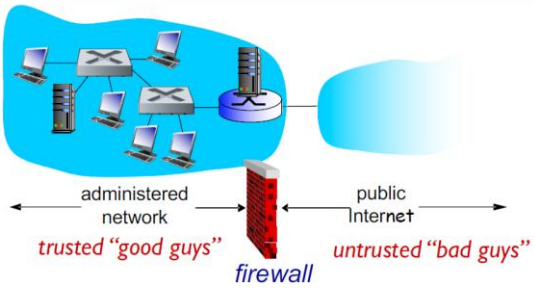
Use firewall as a "Perimeter Defense" in part of a comprehensive security policy

A firewall is a control point for monitoring and implementing access policies
- Interconnects networks with different trusts

Perimeter defense:周边防御; comprehensive:综合的

firewall

isolates organization's internal net from larger Internet, allowing some packets to pass, blocking others



Prevent denial of service attacks

– SYN flooding: attacker establishes many bogus TCP connections, no resources left for "real" connections

Prevent illegal modification/access of internal data

– e.g., attacker replaces CIA's homepage with something else

Allow only authorized access to inside network

– set of authenticated users/hosts

Manage access for authorized users

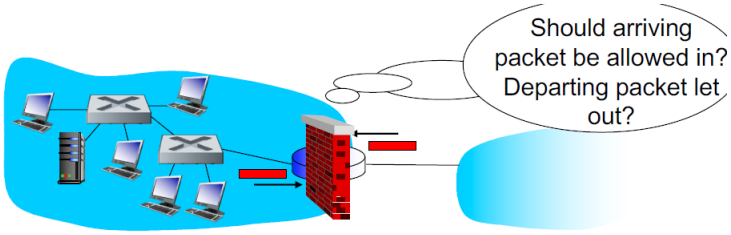
– which user is allowed to access what type of services outside of the intranet

Type:

Three types of firewalls:

– Stateless packet filters
– Stateful packet filters
– Application gateways

Stateless Packet Filters



Internal network connected to the Internet via *router firewall*

Firewall *filters packet-by-packet*, decision to forward/drop packet based on:

– source IP address, destination IP address
– TCP/UDP source and destination port numbers
– ICMP message type
– TCP flag bits (SYN, ACK, FIN)

Example 1: block incoming and outgoing datagrams with IP protocol field = 17, and with either source or dest port = 23

– result: all incoming, outgoing UDP flows and telnet connections are blocked

Example 2: block inbound TCP segments with SYN = 1 & ACK = 0.

– result: prevents external clients from making TCP connections with internal clients

Policy	*Firewall Setting*
No outside Web access.	Drop all outgoing packets to any IP address, port 80 (or HTTP ports)
No incoming TCP connections, except those for institution's public Web server only.	Drop all incoming TCP SYN packets to any IP except 130.207.244.203, port 80
Prevent your network from being used for a smurf DoS attack.	Drop all ICMP packets going to a "broadcast" address (e.g. 130.207.255.255/16).
Prevent your network from being tracerouted	Drop all outgoing ICMP TTL expired traffic

Access Control Lists (ACL)

ACL: table of rules, applied top to bottom to incoming packets: (action, condition) pairs [222.22/16 is the home network]

action	source address	dest address	protocol	source port	dest port	flag bit
allow	222.22/16	outside of 222.22/16	TCP	> 1023	80	*
allow	outside of 222.22/16	Allows Web Surfing to internal users	TCP	80	> 1023	ACK
allow	222.22/16	outside of 222.22/16	UDP	> 1023	53	---
allow	outside of 222.22/16	222.2 Allows DNS to operate	UDP	53	> 1023	----
deny	*	*	*	*	*	*

Two default policies (Discard/Forward):

Discard/Deny – prohibit unless explicitly permitted

– more conservative, controlled, services only added on case to case basis

Forward/Allow – permit unless explicitly prohibited

– Easier to manage but less secure

Explicitly:明确地; conservative:保守地

Stateful Packet Filtering

stateless packet filter: heavy handed tool

– admits packets that "make no sense," e.g., source port = 80, ACK bit set, even though no TCP connection has been initiated

action	source address	dest address	protocol	source port	dest port	flag bit
allow	outside of 222.22/16	222.22/16	TCP	80	> 1023	ACK

stateful packet filter: track status of every connection

– track TCP connection setup (SYN), teardown (FIN): determine whether incoming, outgoing packets "makes sense"

– timeout inactive connections at firewall: no longer admit packets

- ACL augmented to check connection state table before admitting packet for the rule

action	source address	dest address	proto	source port	dest port	flag bit	check conxion
allow	222.22/16	outside of 222.22/16	TCP	> 1023	80	*	
allow	outside of 222.22/16	222.22/16	TCP	80	> 1023	ACK	x
allow	222.22/16	outside of 222.22/16	UDP	> 1023	53	---	
allow	outside of 222.22/16	222.22/16	UDP	53	> 1023	---	x
deny	*	*	*	*	*	*	

- ACL rule

action	source address	dest address	protocol	source port	dest port	flag bit	Conxion
allow	outside of 222.22/16	222.22/16	TCP	80	> 1023	ACK	X

- Connection table

source address	dest address	protocol	source port	dest port	Timer
222.22.2.2	199.1.205.1	TCP	12559	80	Valid
222.22.22.77	203.77.5.55	TCP	47855	80	Valid

Packet arrives: Source IP 199.5.5.20, Source port 80, ACK = 1,
Destination IP=222.22.2.2, Destination port = 36500

No existing connection found in Connection table: Reject the packet

Application Gateways

Firewalls only read packet headers

What if you want to allow user based access instead of host based (using IP addresses)?

- Requires user authentication
- This is beyond the capability of stateless/stateful filters

Application layer is involved

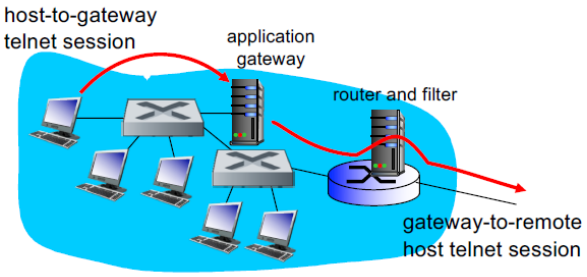
- overhead more than inspecting packets at the network and transport layers

Overhead:开销

An application specific filter

- Example: allow select internal users to telnet outside

AG prompts for username/passwords



1. require all telnet users to telnet through gateway.
2. for authorized users, gateway sets up telnet connection to dest host. Gateway relays data between 2 connections
3. router filter blocks all telnet connections not originating from gateway.

Host based Firewalls

A software module to secure an individual host

Available in many OS and often used in servers

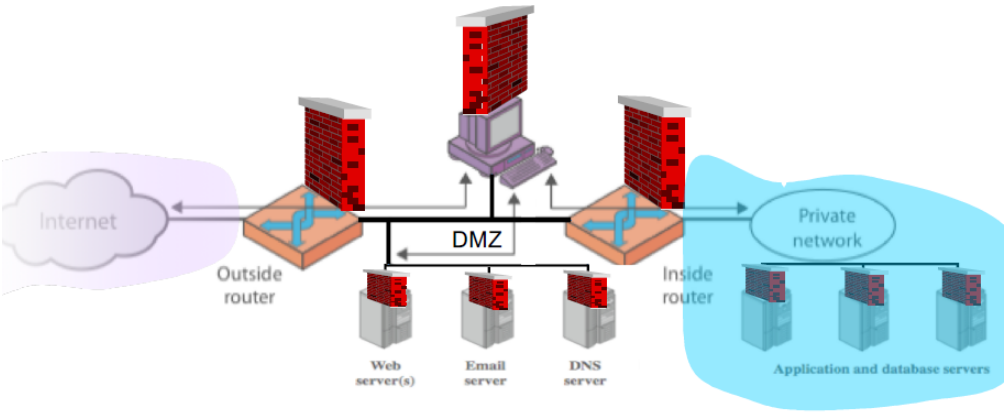
Can tailor filtering rules to match the host environment

Independent of the network topology

Provides additional layer of protection

Tailor:定制

Firewall Configurations (DMZ)



DMZ: De Militarized Zone that hosts organisation's external facing services

Outside router only advertises servers in the DMZ

Limitations of firewalls/gateways

IP spoofing: router can't know if data "really" comes from claimed source

if multiple app's. need special treatment, each has own app. gateway

client software must know how to contact gateway.

– e.g., must set IP address of proxy in Web browser

- filters often use all or nothing policy for UDP
- tradeoff: degree of communication with outside world, level of security
- many highly protected sites still suffer from attacks

Intrusion detection systems (IDS)

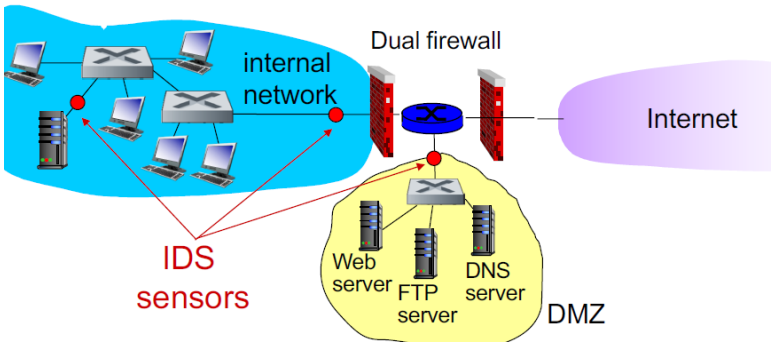
Packet filtering:

- operates on TCP/IP headers only
- no correlation check among sessions

IDS: intrusion detection system

- deep packet inspection: look at packet contents (e.g., check character strings in packet against database of known virus, attack strings)
- examine correlation among multiple packets
  - port scanning
  - network mapping
  - DoS attack

multiple IDSs: different types of checking at different locations



Intrusion [RFC 2828 Internet Security Glossary]

- A security event or a combination of security events in which an intruder gains or attempts to gain, access to a system (or system resources) without having authorization to do so
- Intruder may be from outside the network or a legitimate user of the network
- Intruder attacks range from gentle (just looking around) to the serious (reading privileged data, perform un-authorized modifications, disrupt services etc.)

Intrusion detection

- A security service that monitors and analyzes system events for the purpose of finding, and providing real-time or near real-time warning of attempts to access system resources in an un-authorized manner

–一种安全事件或安全事件的组合，其中入侵者未经授权而获得或试图获得、访问系统（或系统资源）。

–入侵者可能来自网络外部或网络的合法用户。

–入侵者攻击的范围从温和（只是环顾四周）到严重（读取特权数据、执行未经授权的修改、中断服务等）。

–一种安全服务，用于监控和分析系统事件，以查找和提供以非授权方式访问系统资源的尝试的实时或近

实时警告。

Denial of service

– Attempts to crash a service or machine, overload network links, CPU, or fill up the disk, e.g. by sending lots of packets

Port Scanning

– Intruder sends packets to a list of ports trying to find open vulnerable ports. Next step could be to deliver malicious code at a vulnerable port

Securing remote shell privileges

– Intruder opens a shell on the victim machine, allowing arbitrary code execution

Network mapping

Worms, viruses, trojans

OS vulnerabilities attacks

拒绝服务

–试图使服务或机器崩溃、过载网络链接、CPU 或填充磁盘，例如发送大量数据包。

•端口扫描

–入侵者将数据包发送到试图查找打开的易受攻击端口的端口列表。下一步可能是在易受攻击的端口上交付恶意代码

•保护远程 shell 权限

–入侵者打开受害者机器上的外壳，允许任意代码执行

•网络映射

•蠕虫、病毒、特洛伊木马

•操作系统漏洞攻击

Intrusion Techniques

Target identification and information gathering

– OSINT (Open Source Intelligence)

– Nmap

Gaining Access

– Vulnerability identification

– Acquire passwords (guess or brute force)

– Install reverse shell

Privilege Escalation

– Exercise access rights of owner

目标识别和信息收集 –OSInt（开源智能） NMAP

•获取访问权限 –漏洞识别 –获取密码（猜测或暴力）–安装后壳体

•权限提升

–行使所有者的访问权

Motivated by thrill of access and status
- hacking community a strong meritocracy
- status is determined by level of competence

Benign intruders might be tolerable
- do consume resources and may slow performance
- can't know in advance whether benign or malign

Awareness led to establishment of Computer Emergency Response Teams (CERTs)
- collect / disseminate vulnerability info / responses
- hackers also have access to CERT reports

受到进入和地位的刺激 –黑客社区是一个强大的精英阶层 –状态由能力水平决定

•良性入侵者可能是可以容忍的。 –消耗资源并可能降低性能 –不能提前知道是良性还是恶性

•意识导致建立计算机紧急情况响应小组（CERT） –收集/传播漏洞信息/响应 –黑客还可以访问证书报告

Elements of Intrusion Detection

Primary assumptions:
- System activities are observable
- Normal and intrusive activities have distinct evidence

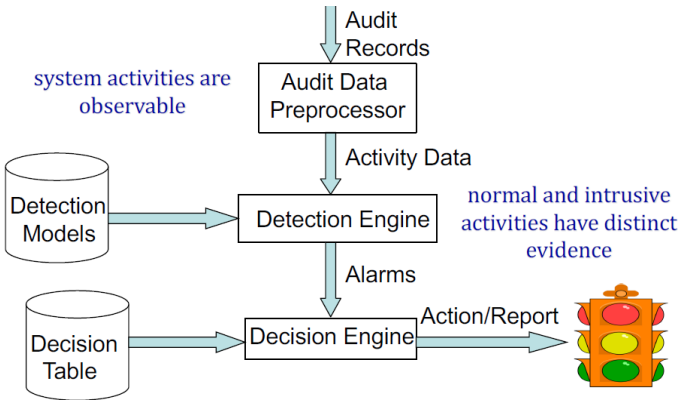
Components of intrusion detection systems:
- From an algorithmic perspective:
  - Features - capture intrusion evidences
  - Models - piece evidences together
- From a system architecture perspective:
  - Various components: audit data processor, knowledge base, decision engine, alarm generation and responses

主要假设：-系统活动可观察 -正常和侵入性活动有明显证据

入侵检测系统组件：–从算法的角度来看： •功能-捕获入侵证据 •模型-将证据拼凑在一起

–从系统架构的角度来看： •各种组件：审计数据处理器、知识库、决策引擎、警报生成和响应

Components of Intrusion Detection



Elements of Intrusion Detection

Audit
- Recording of all security relevant events of a supervised system
- Collects the input for intrusion detection module

Audit data delivers information on:
- Who accessed?
- When, where and how?
- Who's and which resource?

Audit data requires integrity protection
- Attacker can wipe out traces of malicious behavior

审计 -记录监控系统的所有安全相关事件 –收集入侵检测模块的输入

•审计数据提供以下信息： —谁访问？ –何时、何地、如何？ —谁和哪种资源？

•审计数据需要完整性保护 –攻击者可以清除恶意行为的痕迹

Wipe out:擦除

Intrusion Detection Approaches

Features: evidences extracted from audit data

Analysis approach: piecing the evidences together
- Misuse detection (a.k.a. signature-based)
- Anomaly detection (a.k.a. statistical-based)

特点：从审计数据中提取证据

•分析方法：将证据拼凑在一起 –误用检测（A.K.A.基于签名）–异常检测（A.K.A.基于统计）

Anomaly:异常的； misuse:滥用

Signature based IDS

Uses predefined proper (or bad) set of rules and patterns
- Event audit analysis reveals signatures for known past attacks

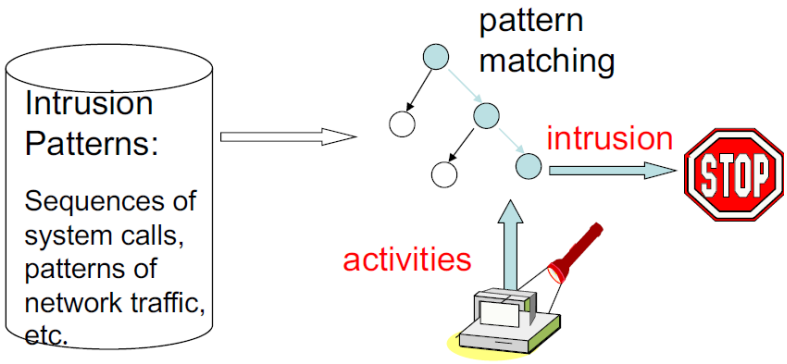
ID system is programmed to interpret a certain series of packets, or a certain piece of data contained in those packets as an attack

Mostly based on Pattern Matching systems
- An IDS that watches web servers might be programmed to look for the string "phf" as an indicator of a CGI program attack (for example, the ``phf'' in ``GET /cgi-bin/phf?'')

使用预定义的正确（或错误）规则和模式集 –事件审计分析显示已知过去攻击的特征

•ID 系统被编程为将某一系列数据包或这些数据包中包含的某一数据段解释为攻击

•主要基于模式匹配系统 –监视 Web 服务器的 ID 可能被编程为查找字符串“phf”作为 CGI 程序攻击的指示器（例如，“get/cgibin/phf”中的“phf”）。



Example: ***if*** (traffic contains “x90+de[^\r\n]{30}”) ***then***
“attack detected”
Problems?

Can't detect new attacks

Drawback of Signature based IDS

Drawbacks:

- Requires prior knowledge of potential attacks and only work if the attack signature is in the database
- Signature database requires continuous updating
- Higher rate of “false negative” with outdated database

Anomaly based IDS

Consider normal/expected behavior of legitimate users over a period of time; apply statistical tests to detect intruder

Intruder unlikely to mimic the behavior pattern of the legitimate user

– Profile based (time/duration/IP for login)
– Threshold based (various events such as %age of ICMP traffic)

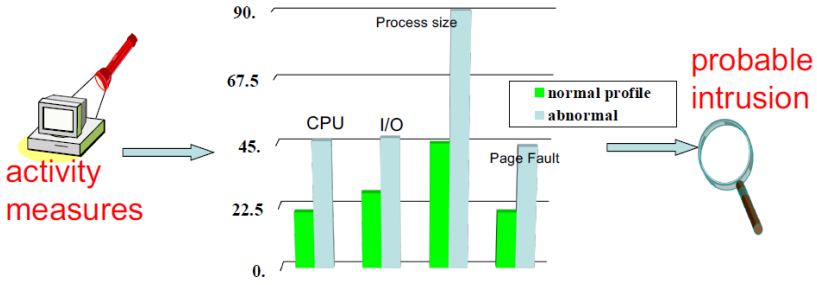
An attack scenario needs not to be defined a priori

考虑合法用户在一段时间内的正常/预期行为；应用统计测试检测入侵者

•入侵者不太可能模仿合法用户的行为模式 –基于配置文件（登录时间/持续时间/IP） –基于阈值（各种事件，如 ICMP 流量的百分比）

•无需预先定义攻击场景

Anomaly Detection



Define a profile describing normal behavior, then detect deviations

Deviation:偏差

Relatively high “false positive” rates

- Anomalies could be just new normal activities
- Anomalies caused by other elements faults e.g., router misconfigurations

Privacy of users

- Collecting specific user patterns
- Work related and personal habits

"false negative", if a normal behavior pattern matches an attack pattern

“假阳性率比较高。”–只是正常的活动异常可能是新的 –例如带给其他元素的异常信号，misconfigurations 路由器

•用户隐私的 –collecting 特定用户模式 –个人和工作相关的广告

•“假阴性”，如果一个攻击模式和正常模式的不良行为 matches

IDS Deployment

Network based

- Monitor network traffic

Host based

- Monitor single host activity and computer processes

Hybrid

- Permits combined analysis of system events and network traffic

Host based IDS

Specialized software to monitor system activity for detecting suspicious behavior

- Log all relevant system events (e.g., file/device accesses)
- Monitor shell commands and system calls executed by user applications and system programs
- Pay a price in performance if every system call is filtered

Problems:

- User dependent: install/update IDS on all user machines!
- If attacker takes over machine, can tamper with IDS binaries and modify audit logs
- Only local view of the attack

用于监视系统活动以检测可疑行为的专用软件 –记录所有相关系统事件（如文件/设备访问） –监控用户应用程序和系统程序执行的 shell 命令和系统调用 –如果过滤了每个系统调用，就要为性能付出代价

问题： –用户相关：在所有用户机器上安装/更新 ID！ –如果攻击者接管计算机，则可以篡改 IDS 二进制文件并修改审计日志。 –只有攻击的局部视图

Tamper:篡改; binaries:文件

Network IDS

NIDS monitors traffic at selected points on a network

- In near real-time to detect intrusion patterns

Deploying sensors at strategic locations

- Packet sniffing via tcpdump at routers

Inspecting network traffic

- Watch for violations of protocols and unusual connection patterns
- Look into the packet payload for malicious code

Limitations

- Cannot execute the payload or do any code analysis
- Record and process huge amount of traffic
- Easily defeated by encryption

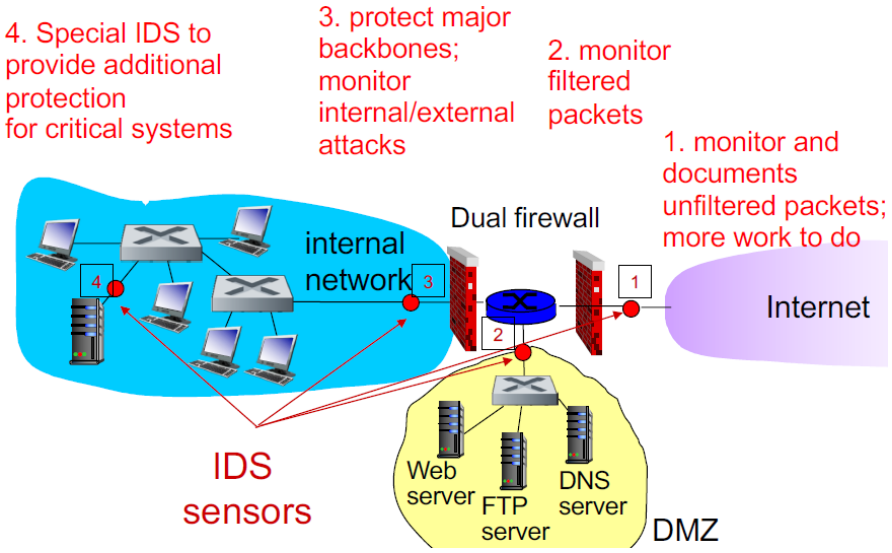
NIDS 监视网络上选定点的流量 –近实时检测入侵模式

•在战略位置部署传感器 –通过路由器上的 tcpdump 进行包嗅探

•检查网络流量 –注意违反协议和异常连接模式 –查看数据包有效负载中是否存在恶意代码

局限性 –无法执行有效负载或执行任何代码分析 –记录和处理大量流量 –容易被加密打败

NIDS Deployment



Wireless IDS

Wireless inherent characteristics provide the relative ease of accessing (and injecting) network communications

– Each frame is broadcasted

On the other hand, there are complexities involved in monitoring wireless communications

– 2 common frequency bands (5GHz and 2.4GHz)

– Several channels within each band

A wireless sensor can monitor a single channel at a time

– A sensor will miss malicious activity occurring on other channels when it is monitoring one particular channel

– Attacker can launch attack simultaneously on two different channels

无线固有特性提供了相对容易的访问（和注入）网络通信 –每个帧都被广播

•另一方面，在监控无线通信方面存在复杂性 –2 个公共频段（5GHz 和 2.4GHz） –每个波段内有几个频道

•无线传感器可以一次监测一个信道 –当传感器监视某个特定通道时，它会错过其他通道上发生的恶意活动。 –攻击者可以在两个不同的通道上同时发起攻击

Inherent:内在的； simultaneously:同时地

Wireless sensors normally perform channel scanning

- They can monitor each channel a few times per second
  - Attacker can attack in short bursts on un-scanned channels
- Each sensor sees only a fraction of the activity on each channel
  - Forensics data is incomplete

Wireless IDS can use specialized hardware with multiple radios and antennas

The actual range of the wireless sensor depends on the surrounding facilities, location of people within the facility and other changing characteristics

无线传感器通常执行信道扫描 –他们可以每秒对每个频道进行几次监控 ： 攻击者可以在未扫描通道上进行短时间突发攻击 –每个传感器只能看到每个通道上活动的一小部分：取证数据不完整

•无线 ID 可以使用具有多个无线电和天线的专用硬件

•无线传感器的实际范围取决于周围设施、设施内人员的位置和其他变化特征。

Forensics data:取证数据

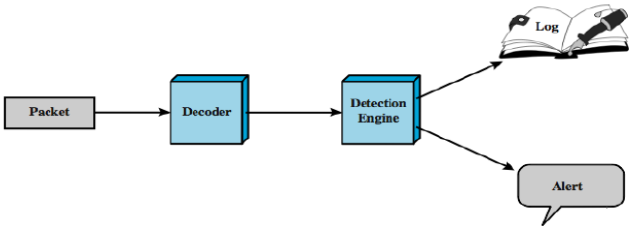
Snort IDS

An open source light weight IDS

- Real time packet capture and rules analysis
- Can work in inline or passive modes

Components:

- Decoder
- Detector
- Logger
- Alerter



Decoder:解码器

SNORT Rules

- use a simple, flexible rule definition language
- each rule consists of a fixed header and zero or more options

  action protocol SIP Sport <dir> DIP Dport [options]

Action	Description
alert	Generate an alert using the selected alert method, and then log the packet.
log	Log the packet.
pass	Ignore the packet.
drop	Drop the packet and log.
reject	Drop the packet, log it, and then send a TCP reset if the protocol is TCP or an ICMP port unreachable message if the protocol is UDP.
sdrop	Silently drop the packet but do not log.

- Last three actions only in inline mode
- Protocols supported : TCP, UDP, IP & ICMP
- Direction -> & < >

SNORT Rule Options (subset)

meta-data
msg Defines the message to be sent when a packet generates an event. **reference** Defines a link to an external attack identification system, which provides additional information. **classtype** Indicates what type of attack the packet attempted. **sid** Signature ID for the rule
payload
content Enables Snort to perform a case-sensitive search for specific content (text and/or binary) in the packet payload. **depth** Specifies how far into a packet Snort should search for the specified pattern. Depth modifies the previous content keyword in the rule. **offset** Specifies where to start searching for a pattern within a packet. Offset modifies the previous content keyword in the rule. **nocase** Ignore case. Nocase modifies the previous content keyword in the rule.
non-payload
ttl Check the IP time-to-live value. This option was intended for use in the detection of traceroute attempts. **id** Check the IP ID field for a specific value. Some tools (exploits, scanners and other odd programs) set this field specifically for various purposes, for example, the value 31337 is very popular with some hackers. **dsize** Test the packet payload size. This may be used to check for abnormally sized packets. In many cases, it is useful for detecting buffer overflows. **flags** Test the TCP flags for specified settings. **seq** Look for a specific TCP header sequence number.
post-detection
logto Log packets matching the rule to the specified filename. **session** Extract user data from TCP Sessions. There are many cases where seeing what users are typing in telnet, rlogin, ftp, or even web sessions is very useful.

SNORT Rules

log udp any any -> 192.168.1.0/24 1:1024

Log UDP traffic with any source IP any source port and destination is any IP within 192.168.1.0/24 and destination port any port less than and equal to 1024

alert tcp any any -> any any (flags: SF; msg: "Possible SYN FIN scan";)

Alert if both SYN and FIN flags set at the same time

alert tcp any any -> any any (msg:"Possible exploit"; content:"|90|"; offset:40; depth:75; dsize: >6000;)

Alert for NOP instructions between bytes 40 and 75 of the data portion of a packet and payload size is > 6000 bytes.

WK06: Insider Threats, Access Control, and Network Security

Overview

Insider Threats
- Definition
- Types
- High-risk insiders
- Influential cases
- Challenges in protecting against insider threats

Detection of Insider Threats
- Key intuition
- Attack phases
- Detection Approaches

Preventing Against Insider Threats
- Access Control

Recent research in this area

Tuition: 直觉

Security model

The term "secure" is widely abused
- When somebody says "XYZ is secure", ask them what they mean by the word "secure".

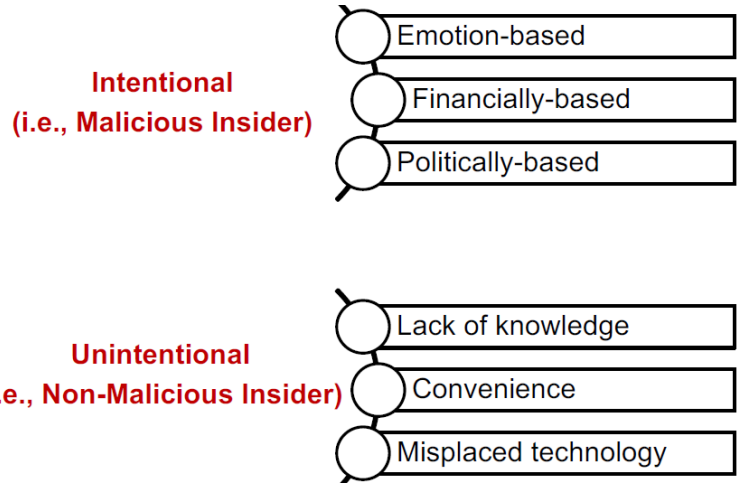
Security Model
- **What do we want to protect?**
  - What are the conditions for the adversary to win?
- **What resources does the adversary have?**
  - Money, computing power, …
- **What system access does the adversary have?**
  - Insider/outsider, sees public key, wiretapping,…
- **What is the system's desired lifetime?**
  - 10 seconds, 1 year, infinity,…

Wiretapping:窃听; adversary:敌人

Insider threat: definition

CERT: Someone who **exploits** his access to the organization's network, system, and data to take actions that negatively affect the Confidentiality, Integrity, and Availability (**CIA**) of the organization's information and Information and Communications Technology (**ICT**) infrastructure.

Types of Insiders

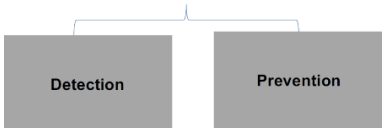


Intentional:意图的，故意的

What makes insider threats challenging?

1. People/users are the weakest link.
   - *"Security is only as good as it's weakest link, and **people** are the weakest link in the chain." [Bruce Schneier, Secrets and Lies, 2000]*
2. **Perimeter defences** are irrelevant.
   - *One who **has legitimate access** to information/systems (i.e. is trusted).*
3. Growing number of **mobile devices** with access to data and users with excessive privileges.
   - BYOD policy adopted by many organizations.
   - How easier would it be for next Snowden to copy files when accessing using own device?
4. Still at the **early stages** of developing usable and efficient insider threat **solutions**.
   - Still more concerned about outsiders! And, not investing enough.

Protection against insider threats



How to detect: the key intuition.

Irrespective of intent, malicious or unusual behaviour will deviate from normal behavioural patterns.

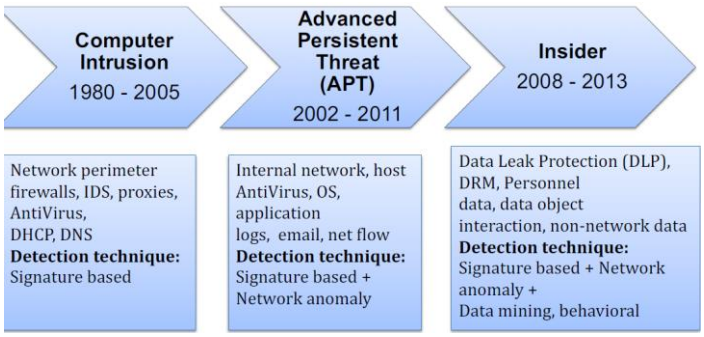
Irrespective:无关的；deviate:脱离 不管意图如何，恶意或异常行为都会偏离正常的行为模式。

Malicious insider attack phases

Kill chain	Threat
Reconnaissance	Port scan Network vulnerability scan Web application vulnerability scan database vulnerability scan
Weaponisation	Social engineering
Delivery	Email spam (URL or attachments) Malicious or phishing websites Removable media
Exploit	Privilege escalation
Install	RAT or backdoor
C2	DDoS Email spam Click fraud and bitcoin mining
Actions on objectives	Data exfiltration Violation against data integrity or availability Sabotage of ICT systems

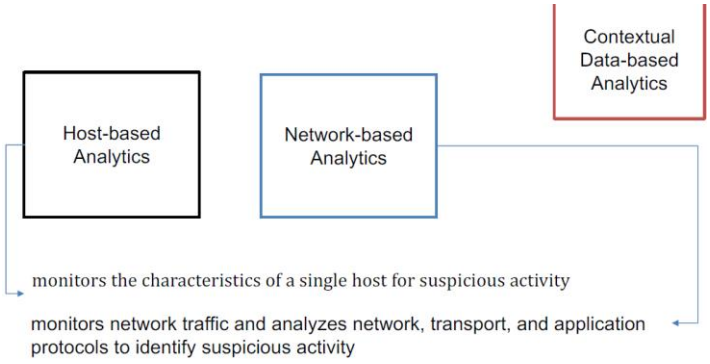
Reconnaissance: 侦察 weaponisation：武器化 Spam:垃圾短信; escalation:增加; fraud:欺骗;exfiltration:漏出;violation:违反;sabotage:妨害

Security Attack Detection Evolution

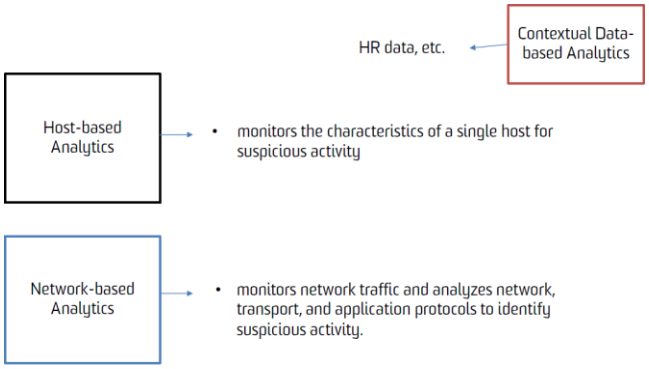


网络外围防火墙、ID、代理、防病毒、DHCP、DNS 检测技术：基于签名
内部网络、主机防病毒、操作系统、应用程序日志、电子邮件、网络流检测技术：基于签名+网络异常
数据泄漏保护（DLP）、DRM、人员数据、数据对象交互、非网络数据检测技术：基于签名+网络异常+数据挖掘、行为

Detection approaches



监视单个主机的特征以了解可疑活动
监控网络流量并分析网络、传输和应用程序协议以识别可疑活动



IDS requirements

run continually
be fault tolerant
resist subversion
impose a minimal overhead on system
configured according to system security policies
adapt to changes in systems and users
scale to monitor large numbers of systems
provide graceful degradation of service
allow dynamic reconfiguration

•连续运行 •容错 •抵制颠覆 •对系统施加最小的开销 •根据系统安全策略配置 •适应系统和用户的变化
•大规模监控大量系统 •提供优雅的服务降级 •允许动态重新配置

Host-based approaches to intrusion detection

anomaly detection

threshold detection
- involves counting the number of occurrences of a specific event type over an interval of time

profile based
- profile of the activity of each user is developed and used to detect changes in the behavior of individual accounts

signature detection

- involves an attempt to define a set of rules or attack patterns that can be used to decide that a given behavior is that of an intruder

异常检测：阈值检测 –包括计算特定事件类型在一段时间内发生的次数。

基于配置文件 –开发每个用户的活动概要，并用于检测单个帐户行为的变化。

签名检测：涉及到定义一组规则或攻击模式的尝试，这些规则或攻击模式可用于确定给定行为是入侵者的行为。

Data source for Host-based Analytics

Measure	Model	Type of Intrusion Detected
Login and Session Activity		
Login frequency by day and time	Mean and standard deviation	Intruders may be likely to log in during off-hours.
Frequency of login at different locations	Mean and standard deviation	Intruders may log in from a location that a particular user rarely or never uses.
Time since last login	Operational	Break-in on a "dead" account.
Elapsed time per session	Mean and standard deviation	Significant deviations might indicate masquerader.
Quantity of output to location	Mean and standard deviation	Excessive amounts of data transmitted to remote locations could signify leakage of sensitive data.
Session resource utilization	Mean and standard deviation	Unusual processor or I/O levels could signal an intruder.
Password failures at login	Operational	Attempted break-in by password guessing.
Failures to login from specified terminals	Operational	Attempted break-in.
Command or Program Execution Activity		
Execution frequency	Mean and standard deviation	May detect intruders, who are likely to use different commands, or a successful penetration by a legitimate user, who has gained access to privileged commands.
Program resource utilization	Mean and standard deviation	An abnormal value might suggest injection of a virus or Trojan horse, which performs side-effects that increase I/O or processor utilization.
Execution denials	Operational model	May detect penetration attempt by individual user who seeks higher privileges.
File Access Activity		
Read, write, create, delete frequency	Mean and standard deviation	Abnormalities for read and write access for individual users may signify masquerading or browsing.
Records read, written	Mean and standard deviation	Abnormality could signify an attempt to obtain sensitive data by inference and aggregation.
Failure count for read, write, create, delete	Operational	May detect users who persistently attempt to access unauthorized files.

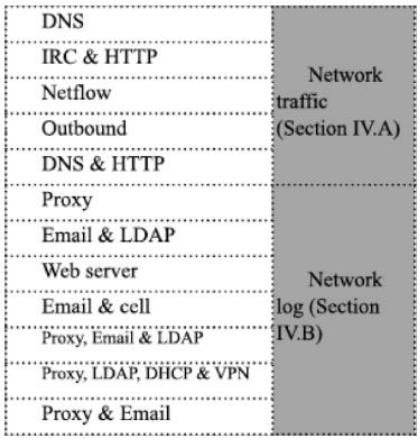
System-call

Command, Keyboard, and Mouse

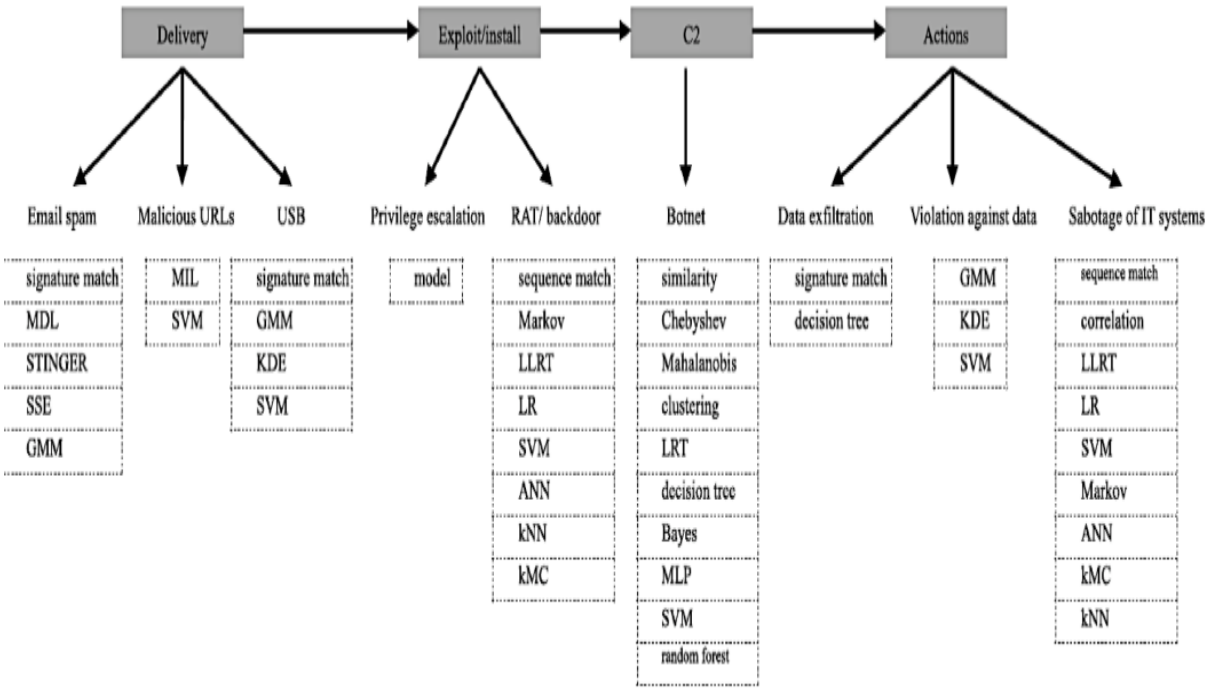
Host log

Data source for Network-based Analytics

Network traffic
Network logs

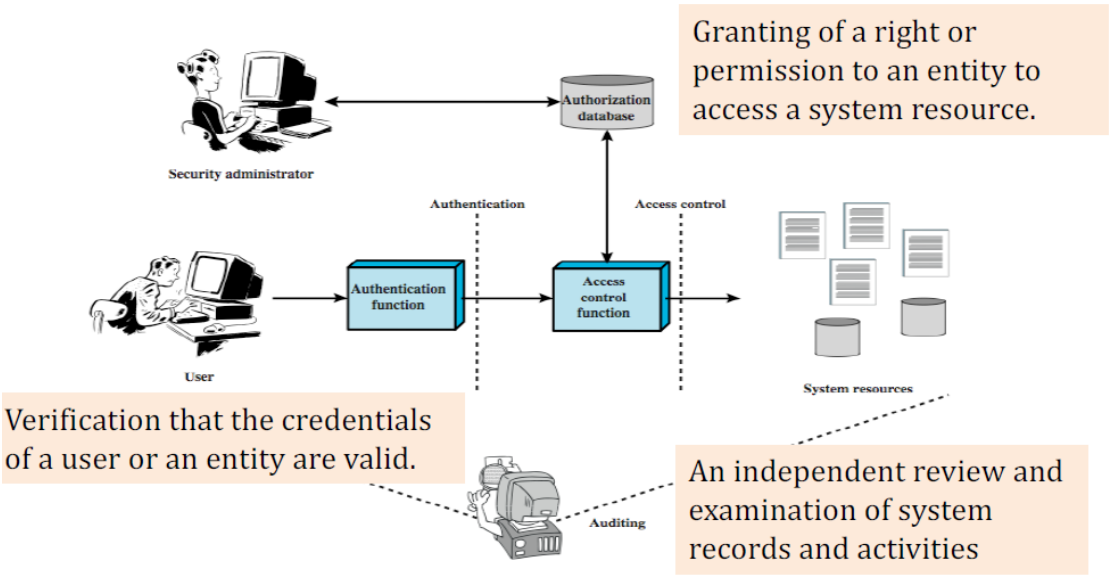


Overview of algorithms used for insider threats



Preventing insider threats

Access Control: the oldest information security mechanism (pre-dates WWW).
Who can access what, under what conditions, and for what purposes.



Access control process

Subject (initiator, data requestor)	**Action** (request)	**Access Right** (permission)	**Object** (target)
• Submits the access request (e.g. the user)	• Requested Operation (e.g., read, delete, etc.)	• A specific ability for the subject to do the action to the object.	• What is being requested (e.g., a file).

Initiator:发起人

Access control policies

Discretionary Access Control (DAC), 1970

- Owner controls access
- Grounded in pre-computer policies of researchers

Mandatory Access Control (MAC), 1970

- Synonymous to Lattice-Based Access Control (LBAC)
- Access based on security labels
- Labels propagate to copies
- Grounded in pre-computer military and national security policies

Role-Based Access Control (RBAC), 1995

- Access based on roles
- Can be configured to do DAC or MAC
- Grounded in pre-computer enterprise policies
- Most commonly used model today

		OBJECTS			
		File 1	File 2	File 3	File 4
SUBJECTS	User A	Own Read Write		Own Read Write	
	User B	Read	Own Read Write	Write	Read
	User C	Read Write	Read		Own Read Write

(a) Access matrix

自由访问控制（DAC），1970 年

o 所有者控制访问

o 以研究人员的计算机前政策为基础

强制访问控制（MAC），1970 年

o 与基于网格的访问控制（lbac）同义

o 基于安全标签的访问

o 标签传播到副本

o 以计算机前军事和国家安全政策为基础

•基于角色的访问控制（RBAC），1995 年

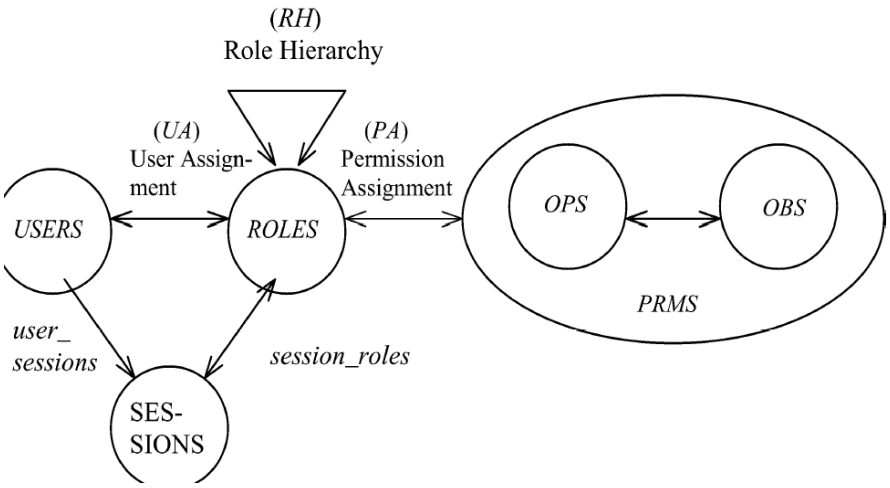
o 基于角色的访问

O 可以被配置为执行 DAC 或 MAC

o 以计算机前企业政策为基础

o 目前最常用的型号

RBAC96 model



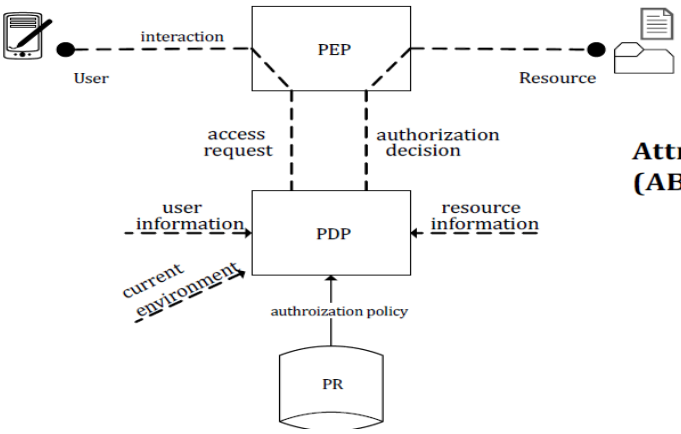
Question

What's missing when looking at the insider attacks discussed at the beginning of this lecture? Why these were not effective in preventing those incidents?

在看本课开始讨论的内幕攻击时，遗漏了什么？为什么这些措施不能有效地预防这些事件？

Access!!!即是访问控制！

Context-based access control



Attribute-based access control (ABAC):

Based on **attributes** of the **user**, the **resource** to be accessed, and current **environment**.

Note: By 2015, majority of AC solutions were predicted to be context-aware in organisation *.

Attribute-based Access Control (ABAC)

Simple Rule Example

Any person must have the same clearance level or higher as the classification of the document he or she is requesting to view, must also be working on that project, be a current employee, and have finished the appropriate training.

```
A user with role=="junior" OR role=="senior" AND
user.clearance>=record.classification can actionId=="view" on
objectType=="record" if classification=="secret" AND user.project ==
"record.project" AND status=="current" and record.training is in user.training
```

Another Rule Example

Junior personnel cannot access top secret information between 20:00 and 6:00.

```
policy checkTimeAccess {
  apply firstApplicable
    rule checkNightAccess {
      target clause role == "junior" and classification = "top secret"
      condition timeInRange(timeOneAndOnly(currentTime),
        "20:00:00":time, "06:00:00":time)
    deny
    }
}
```

XACML policy example – Cont'd

```
<Policy PolicyId="ExamplePolicy"
      RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:permit-overrides">
  <Target>
   <Subjects> <AnySubject/></Subjects>
   <Resources><Resource>
            <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
            <AttributeValue
            DataType="http://www.w3.org/2001/XMLSchema#anyURI">http://server.example.com/code
                                    /docs/developer-guide.html</AttributeValue>
             <ResourceAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#anyURI"
                      AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"/>
            </ResourceMatch>
    </Resource></Resources>
   <Actions><AnyAction/></Actions>
  </Target>
  <Rule RuleId="ReadRule" Effect="Permit">
            ...
    </Rule>
 </Policy>
```

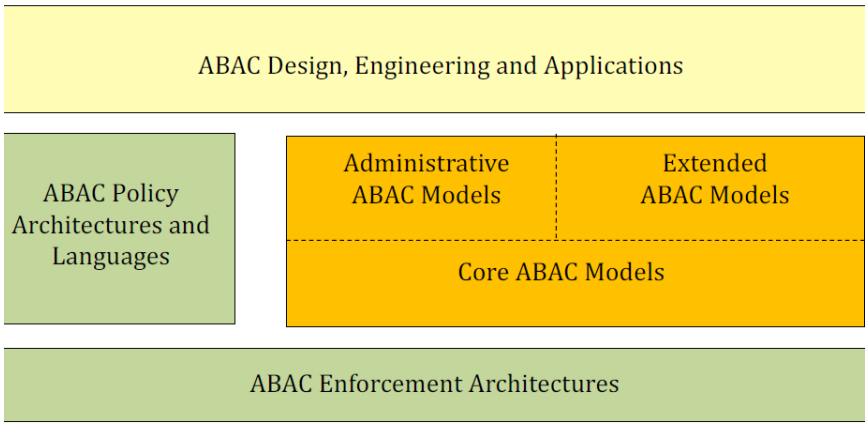
XACML policy example

```
<Rule RuleId="ReadRule" Effect="Permit">
      <Target>
            <Subjects><AnySubject/></Subjects>
            <Resources><AnyResource/></Resources>
            <Actions>
            <Action>
                        <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                        <AttributeValue
    DataType="http://www.w3.org/2001/XMLSchema#string">read</AttributeValue>
                        <ActionAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#string"
                                    AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"/>
                        </ActionMatch>
            </Action>
            </Actions>
    </Target>
    <Condition FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
            <SubjectAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#string"
                                    AttributeId="group"/>
            </Apply>
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">developers</AttributeValue>
   </Condition>
</Rule>
```

ABAC: an active research area

ABAC is orders of magnitude more complex than anything that has been an Access Control.

– Policy specification, enforcement, and implementation.



Magnitude:量级 ABAC 比任何访问控制都要复杂得多。 –政策规范、执行和实施。

Function-based Access Control (FBAC)

Simple intuition: Model intention by **Action**.

In the case of Pentagon's access to State Department: one should be able to run data mining techniques and/or **search** documents' **without being authorized to preform any other operation** such as Print, Copy, E-mail, etc.

Contribution: Providing a systematic solution to an open problem in AC. In other words,

- **AC at level of data,**
- **Control over operation execution,** and
- **Avoid Y/N approach when making access decisions.**

简单直觉：用行动来模拟意图。

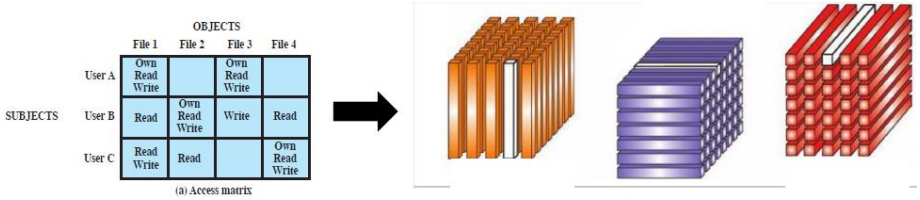
在五角大楼进入国务院的情况下：应该能够运行数据挖掘技术和/或搜索文档，而无需授权执行任何其他操作，如打印、复印、电子邮件等。

贡献：为交流中的一个开放问题提供一个系统的解决方案。

Move from a 2-dimensional Access Control Matrix to a 3-dimensional **Access Control Tensor**.

(Subject, Object, Operation) → (Subject, Object, Function)

User A can execute **Copy, Paste and Email** functions on **data blocks A and B**.



Idea in a nutshell:

Slice data files and store them as separate entities. Separate entities are called “Atom”.

We assign Function permissions for each Atom.

The .**ADoc** file is created on each access request by combining atoms.

对数据文件进行切片并将其存储为单独的实体。单独的实体称为“原子”。

我们为每个原子分配函数权限。

.adoc 文件通过组合原子在每个访问请求上创建。

.ADoc and Atoms

Each Atom has a:

Classification level

Each child Atoms has also a classification level: Classified, confidential, public, etc. The Atom classification level cannot be different from the container (.ADoc file).

$$D = \bigcup Atom(i, j, k, ...)$$

$F(i)$={allowed functions for Atom(i)}
$F(D)$={NOT allowed functions for Document(D)}

$C(i)$={Classification level of Atom(i)}
$C(D)$={Classification level of Document(D)}

$Atom(i)$ is an Atom for Document(D) IFF:
C1: $F(i) \bigcap F(D) = \{\phi\} \wedge C(i) \subset C(D)$

Document(D) is .ADoc file complaint when:
$\forall Atom(i) \in D$, C1 holds.

Relaxed-FBAC

1. Atom’s Functions: When Atoms are created the set of Functions (F) allowed on them is specified.
2. .ADoc policy: Depending on **specified policy** a set of non-allowed functions is created for Atoms.

ATOM’s Function permission list: {Copy, Paste, Email, Send to Bluetooth, Send to Wi-Fi direct, Send to messaging Apps}

.ADoc policy: If user A (member of low-level serviceman) and is after 6 PM and outside of organization safe-list zones in NY and is using APP A, or B, or C then disable Copy, Paste and Email for all atoms in this document.

ATOM’s Function permission list ∩ .Adoc policy

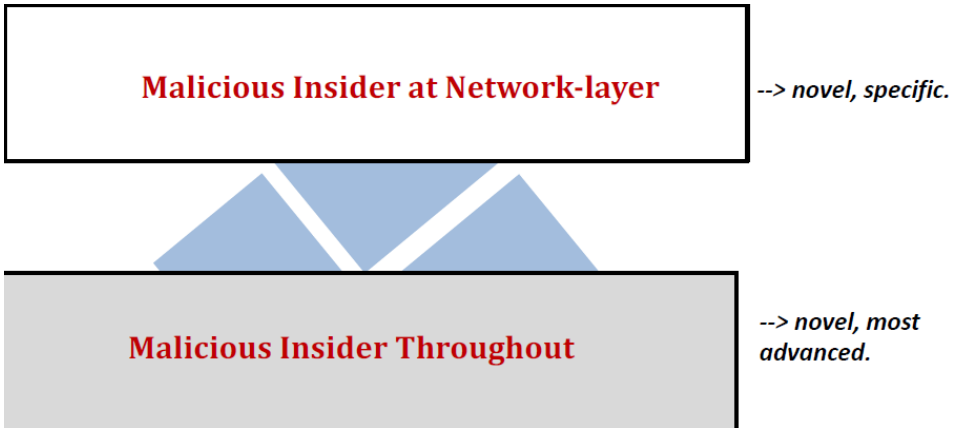
1.原子的函数：当原子被创建时，就指定了允许它们使用的函数集（f）。

2。.adoc 策略：根据指定的策略，为原子创建一组不允许的函数。

Atom 的函数权限列表：

复制、粘贴、发送电子邮件、发送至蓝牙、发送至
Wi-Fi Direct，发送至消息应用程序
.adoc 策略：如果用户 A（低级别服务人员的成员）在下午 6 点之后在纽约的组织安全列表区域之外并且正在使用应用程序 A、B 或 C，则禁用此文档中所有原子的复制、粘贴和电子邮件。

Advanced malicious insiders

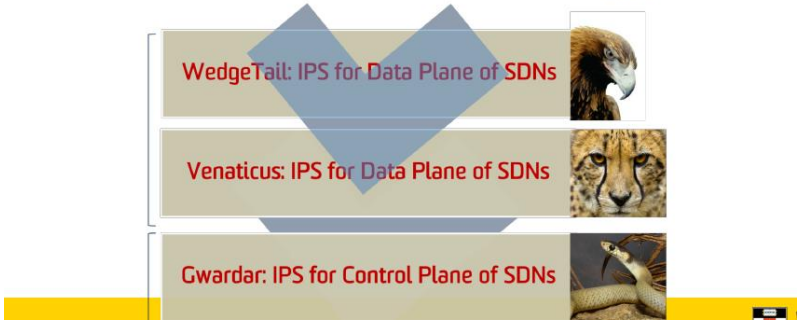


Novel:新奇，异常

Malicious insider at Network-layer

Has access to network infrastructure
Maliciously tampers with the network configurations
Changes the packet forwarding setup throughout or for a subset of the network.

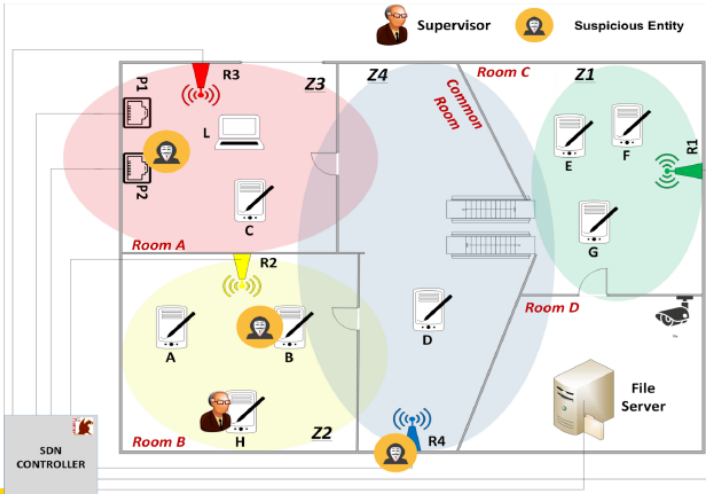
We have proposed and developed early solutions to address this for Software-Defined Networks (SDNs) - the next generation of network architectures. If interested, see relevant papers.



可以访问网络基础设施
•恶意篡改网络配置
•更改整个网络或网络子集的数据包转发设置。
•我们已经为软件定义网络（SDN）——下一代网络架构提出并开发了早期解决方案。如有兴趣，请参阅相关文件。

Malicious Insider Throughout

"BYOD", "Anytime, Anywhere"
Access both at application-layer and network-layer
Assumed to exploit his access to compromise confidentiality, integrity, and availability (CIA) of the organization's data.



What's missing?

1. **Fail to adapt to the 'negative changes' in user's behaviour -- even if it suggests attacking the system.**

   E.g. *hacking tools*

2. **Assuming that users follow the security rules.**

   E.g. *An employee to access confidential information in a secure room over a secure connection and in the presence of a supervisor rather than having an access control system enforcing this policy.*

3. **Trusting the user's device integrated sensors for retrieving the context attributes.**

   E.g. *GPS hack*

4. **A binary approach to access decisions. Context is not deterministic, so can not be access control.**

   E.g. *A user accessing sensitive information in an `insecure context' must not share (e.g. Email) but can read (e.g., View)*

5 **Relying on single access enforcement point.**

> ***E.g.*** *if an attacker compromises a mobile device, it can still be prevented from targeting the organization's services.*

1. 无法适应用户行为中的"负面变化"——即使它建议攻击系统。
2. 假设用户遵循安全规则。
3. 信任用户的设备集成传感器来检索上下文属性。
4. 访问决策的二进制方法。上下文不具有确定性，因此不能是访问控制。
5. 依赖于单一访问强制点。

WK07: Bluetooth Security

Introduction

Open wireless protocol for exchanging data over short distances from fixed and mobile devices, creating personal area network.

A reliable wireless protocol for voice and data transmission

Bluetooth Evolution

Bluetooth Special Interest Group (SIG)

Founded in Spring 1998

By Ericsson, Intel, IBM, Nokia, Toshiba

Now more than 2,000 organizations have joined the SIG

Features

Bluetooth-enabled devices can automatically locate each other

Topology is established on a temporary and random basis

Up to eight Bluetooth devices may be networked together in a master-slave relationship to form a piconet

Piconet:微微网络

This Bluetooth network is called a Piconet. National Institute of Standards and Technology (NIST) defines this network as Wireless Personal Area Network (WPAN).

One is master, which controls and sets up the network (piconet)

Two or more piconet interconnected to form a scatter net

Only one master for each piconet

A device can't be masters for two piconet

The slave of one piconet can be the master of another piconet

一个是 master，它控制和设置网络（piconet）

•两个或多个 piconet 相互连接形成一个散点网络

•每个 piconet 只有一个 master

•一个设备不能是两个 piconet 的主设备

•一个 piconet 的奴隶可以是另一个 piconet 的主人

Security Issues

Authenticity: Are you the device you claim you are?

– Impersonation

Confidentiality: Is the exchanged data only available to the intended devices?

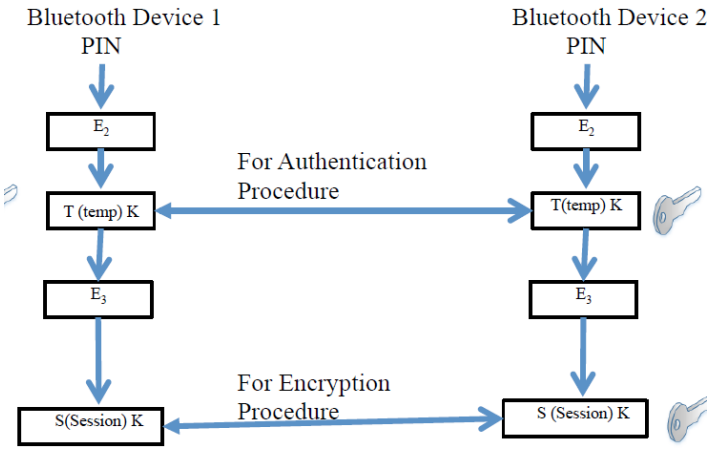
– Packet sniffing

Authorisation: Are only the intended devices accessing the specified data and control?

– Prerequisite: authenticity and confidentiality

Impersonation:扮演，模仿； authenticity:可靠性;authorization:授权，批准

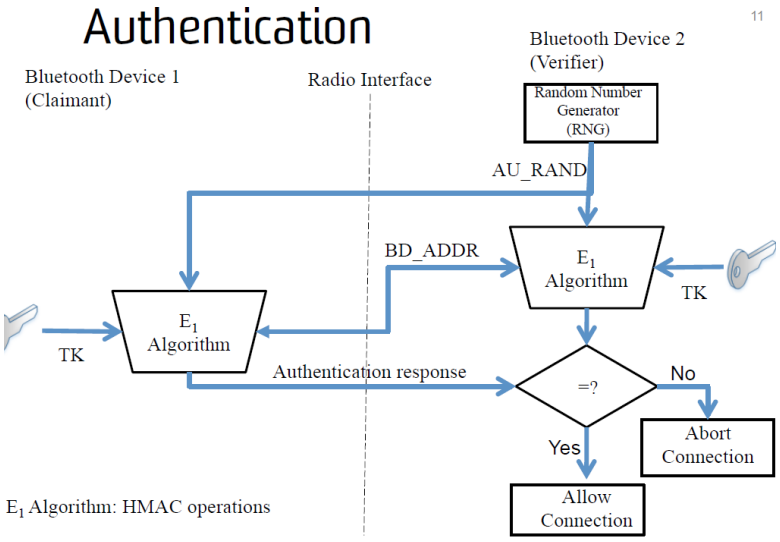
Temporary Key Generation



E_1: HMAC algorithm
E_2: it would be discussed in case study
E_3: it is described in slide 13.

PIN is the default password set in the Bluetooth devices. It is a source of security vulnerability in Bluetooth. They have algorithms to generate keys at different stages. E2 is the algorithm to generate the Temporal Key (TK) that is used for authentication process. Next the devices use another algorithm E3 to generate another Session key that is subsequently used for encryption while transferring data among the devices. This later key (SK) provides confidentiality and the earlier key (TK) was for authenticity.
最初，设备具有初始化步骤中的 PIN。PIN 是蓝牙设备中设置的默认密码。它是蓝牙中安全漏洞的来源。他们有在不同阶段生成密钥的算法。e2 是生成用于身份验证过程的时间密钥（tk）的算法。我们将在后面的幻灯片中讨论这个算法。接下来，设备使用另一个算法 e3 生成另一个会话密钥，随后在设备之间传输数据时用于加密。这个后一个密钥（sk）提供机密性，而前一个密钥（tk）是为了真实性。

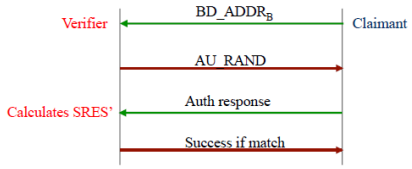
Authentication



For authentication, Bluetooth device 1 is the claimant which has to be authenticated by the verifier Bluetooth device 2.
Step 1: The claimant sends its BD_ADDR across to the verifier. The verifier generates a random number (AU_RAND, 128 bit) and sends it to the claimant.
Step 2: The claimant uses the E1 algorithm (Hash-based Message Authentication Code HMAC) to compute an authentication response using its unique 48-bit Bluetooth device address (BD_ADDR), the TK (generated in previous slide), and AU_RAND as inputs. The verifier performs the same computation. Only the 32 most significant bits of the E1 output are used for authentication purposes.
Step 3. The claimant returns the most significant 32 bits of the E1 output as the authentication response to the verifier.
Step 4. The verifier compares the authentication response from the claimant with the value that it has computed.
Step 5. If the two 32-bit values are equal, the authentication is considered successful. If the two 32-bit values are not equal, the authentication fails. Device only gets verified if it possesses the correct TK and can produce the correct authentication response using the same E1 algorithm. This is one way authentication (claimant gets verified). Devices can switch roles to get two way authentication.
对于认证，蓝牙设备1是索赔人，必须由验证蓝牙设备2进行认证。 **步骤1：**索赔人将其bd_地址发送给验证者。验证器生成一个随机数（au_rand，128位），并将其发送给索赔人。 **步骤2：**索赔人使用e1算法

（基于哈希的消息验证代码hmac）计算验证响应，使用其唯一的48位蓝牙设备地址（bd_addr）、tk（在上一张幻灯片中生成）和au_rand作为输入。验证器执行相同的计算。只有e1输出的32个最高有效位用于身份验证。 **步骤3。**索赔人返回e1输出中最重要的32位作为对验证器的身份验证响应。 **步骤4。**验证器将索赔人的身份验证响应与它计算的值进行比较。 **步骤5。**如果两个32位值相等，则认为身份验证成功。如果两个32位值不相等，则验证失败。只有当设备具有正确的tk并且能够使用相同的e1算法生成正确的认证响应时，设备才会得到验证。这是单向验证（索赔人得到验证）。设备可以切换角色以获得双向身份验证

Authentication Summary



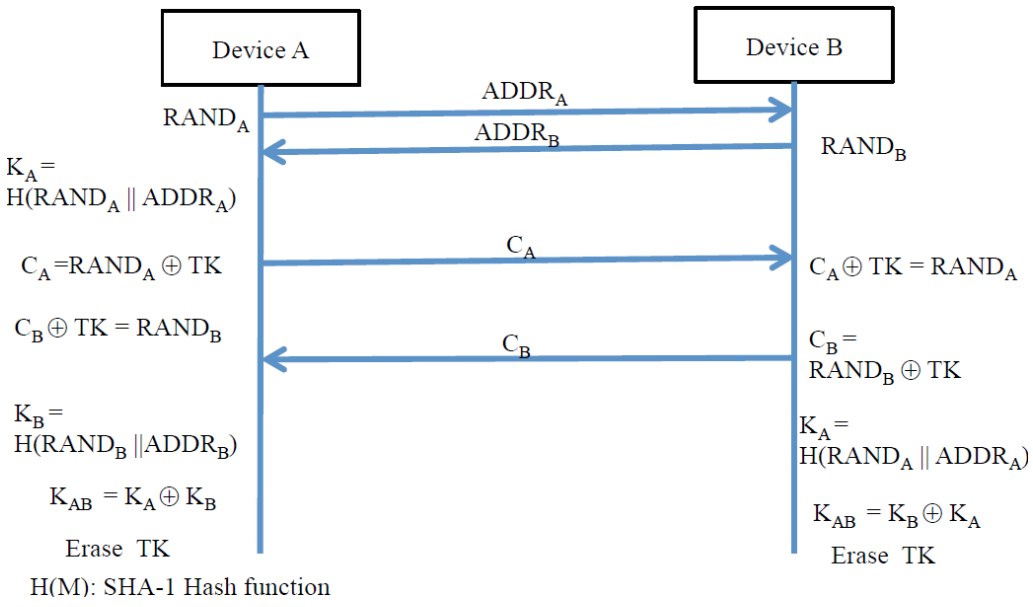
Authentication Process

Parameter	Length	Secrecy parameter
Device Address	48 Bits	Public
Random Challenge	128 Bits	Public
Authentication(Auth) Response	32 Bits	Public
Temporary Key	128 Bits	Secret

If you look at the summary of this authentication process, all the information (parameters) are public except the Temporary Key. That means these are also known to the third parties (attackers) that can use this information.

如果查看此身份验证过程的摘要，除了临时密钥之外，所有信息（参数）都是公共的。这意味着可以使用此信息的第三方（攻击者）也知道这些信息。如果攻击者想冒充索赔人，它就拥有这些公开信息。唯一的秘密部分是攻击者不知道的临时密钥。如果攻击者知道临时密钥，则可以很容易地将其冒充为索赔人。

Session Key Generation



Step 1: Exchange BT addresses (ADDRA and ADDRB)
Step 2: Generate a Random number and use one-way hash function on the generated random number and own BT address. Get KA at A and KB at B. Hash function is one-way means even if the attacker knows KA, it cannot retrieve RANDA
Step 3: TK already known to both A and B. XOR the generated Random Number with TK to get CA (and CB at B).
Step 4: Exchange CA and CB
Step 5: Device A XOR CB with TK to get RANDB and device B XOR CA with TK to get RANDA
Step 6: Device A calculates the Hash using RANDB and ADDRB to get the KB (similarly Device B gets KA)
Step 7: XOR KA and KB to get the final session key KAB . TK can now be discarded. This is how we achieve Authenticity and Confidentiality using TK and SK. Assumption is that TK has already been established to perform this algorithm. It is reusing TK that is a significant vulnerability as the source of the Temporary Key is not secure making this Session key generation algorithm vulnerable.
如果两个设备想要建立通信和交换数据，首先他们必须为此会话派生一个sk **步骤1：**交换BT地址（addra和addrb） **第二步：**生成一个随机数，对生成的随机数和自己的BT地址使用单向散列函数。hash函数是单向的，即使攻击者知道ka，它也无法检索randa **步骤3:**tk已经为a和b所知。xor用tk生成的随机数得到ca（和b）。 **步骤4：**交换CA和CB **步骤5：**用tk设置一个xor-cb以获取randb，用tk设置一个b-xor-ca以获取randa **步骤6：**设备A使用randb和addrb计算散列值以得到kb（同样，设备B得到kA） **步骤7:**xor-ka和kb获取最后一个会话密钥kab。现在可以丢弃tk。这就是我们如何使用tk和sk实现真实性和机密性的方法。假设已经建立了tk来执行该算法。它正在重用Tk，这是一个严重的漏洞，因为临时密钥的来源不安全，这使得会话密钥生成算法容易受到攻击。

Is Channel Hopping Secure?

Channel Hopping (Bluetooth Smart only) –
Both communication parties would hop to a different wireless channel per packet in a fixed channel hopping increment.
Adversary could not achieve data by monitoring one wireless channel only.
We will study its vulnerability soon.

信道跳频（仅蓝牙智能）–两个通信方将以固定信道跳频增量跳到每个数据包的不同无线信道。
•对手仅通过监控一个无线通道无法获得数据。
•我们将很快研究其脆弱性。
Another mechanism to secure wireless networks is called Channel Hopping.
In Channel Hopping, we have multiple channels available within a given frequency for wireless devices to use. Devices select the same channel frequency to communicate.
In Bluetooth Smart protocol, the Bluetooth Special Interest Group (SIG) has specified channel hopping to increase the security. The reason to do this is that by doing channel hopping to different channels for each data packet, the attacker is not able to fully capture the whole conversation by monitoring a single channel. The SIG expected that in order for the attacker to intercept all data packets/conversation, the attacker needs to monitor all channels which increases the cost for the attacker.

This channel hopping basically works with fixed channel hopping increments and we also have vulnerability in this channel hopping mechanism that we will study in the later slides.
另一种保护无线网络的机制称为信道跳频。在信道跳频中，我们在给定频率内有多个信道可供无线设备使用。设备选择相同的信道频率进行通信。 在蓝牙智能协议中，蓝牙特殊兴趣组（bluetouthspecialinterest group，SIG）指定了跳频通道以提高安全性。这样做的原因是，通过对每个数据包跳到不同的通道，攻击者无法通过监视单个通道完全捕获整个会话。 SIG预计，为了让攻击者拦截所有数据包/会话，攻击者需要监控所有通道，这会增加攻击者的成本。这个通道跳跃基本上与固定的通道跳跃增量一起工作

Case study: Bluetooth Low Energy (BTLE)

Introduced in Bluetooth 4.0 (2010)

New modulation and link layer for low power devices

- Incompatible with classic Bluetooth devices
- PHY and link layer different (no channel hopping in classic Bluetooth)
- High-level protocols reused (L2CAP, ATT)

Modulation:调制;incompatible:矛盾的

BTLE applications

High end smart phones

Sports/fitness devices

Door locks

Upcoming medical devices (e.g., blood glucose monitor)

BTLE Protocol review

GATT (Generic Attribute Profile) – how to discover and provide services based on ATT
ATT (Attribute protocol) – how to discover/read/write attributes on a peer device
L2CAP (Logical Link Control and Adaptation Protocol) – packet segmentation and reassemble
Link Layer
Physical Layer

We will focus on Link layer and Physical layer security only.
Other layers are similar to their counterparts in wired networks.

Reassemble:重新装配;counterpart:配对物，副本

Physical Layer

Physical layer: channels for hopping (40 available channels in 2.4Ghz)

- Advertising: 3 channels
- Data: 37 channels

Each channel is of 2 Mhz width. There are two phases of communication;

First is advertising. Advertising phase is required for device discovery (broadcast and connection establishment) so that devices can contact each other. There are 3 channels available for advertising phase. Next phase is the data transfer (after connection establishment) that uses the rest of the 37 data channels.

在蓝牙低能量通信中，2.4GHz范围内共有40个可用信道供通信方使用。在这40个频道中，有3个用于广告，其余37个用于数据传输。每个信道的宽度为2兆赫。沟通有两个阶段：第一阶段是广告。设备发现（广播和连接建立）需要广告阶段，以便设备可以相互联系。广告阶段有3个频道。下一阶段是数据传输（连接建立后），它使用37个数据通道的其余部分。

Channel Hopping

- Hop along 37 data channels
- One data packet per channel
- Next channel = current channel + hop increment (mod 37)
- Time between hops: hop interval, it is the duration when both communication parties stays in one channel. It is equal to one Round Trip Time + channel switch latency.

3→10→17→24→31→1→8→15→…(hop increment = 7)

沿着37个数据通道跳跃

•每个通道一个数据包

•下一个通道=当前通道+跃点增量（mod 37）

•跳之间的时间：跳间隔，是指两个通信方保持在一个信道中的持续时间。等于一次往返时间+信道切换延迟。

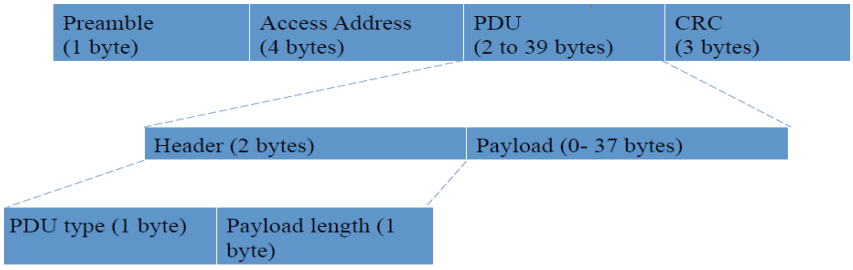
Hop interval is the time between each hop or the time both devices spend in one channel. = RTT + channel switching latency. Not always the same otherwise it would be very easy to guess. Communicating parties negotiate Hop increment and Hop interval in connection establishment phase.

一个主要的观察是，您可以沿着37个通道跳跃，并以循环方式每个通道传输一个数据包。

如果超过37，则环绕并从最低数字开始选择一个通道（如果通道增量为1，则为1）。

下一个通道是当前通道+跳增量模块37跳增量是选择下一个通道时跳过的通道数。您可以使用mod 37来选择一个小于等于37的通道值。跃点间隔是每个跃点之间的时间或两个设备在一个通道中花费的时间。=rtt+信道切换延迟。不总是一样的，否则很容易猜出来。在连接建立阶段，通信方协商跃点增量和跃点间隔。跳跃增量为7的示例。达到31后，加7=38%37为1。等等。37是素数。这很重要。

Link Layer



- Only PDU encrypted which creates security vulnerability
  - packet sniff to break the confidentiality in PDU.

Preamble: 8-bit sequence (01010101), it is for receiving radio to synchronize the transmission.

Access Address: it is a 32-bit address for recipient to pick up the intended Bluetooth packets.

PDU = Protocol Data Unit

CRC = Cyclic Redundancy Check which provides the integrity check whether the packet has experienced any bit corruption.

Note that only the PDU gets encrypted using the Session key (SK) that we discussed earlier, rest three (Preamble, Access address and CRC) are sent in plaintext. Packet sniffing can be used to break the confidentiality.

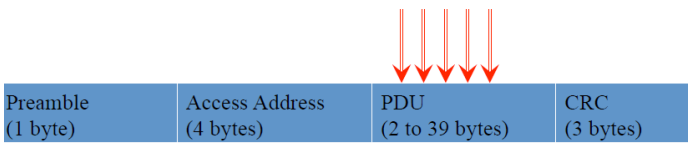
前导码：8位序列（01010101），用于接收无线电同步传输。 访问地址：它是一个32位的地址，收件人可以接收预期的蓝牙数据包。 PDU=协议数据单元 CRC=循环冗余检查 在每个蓝牙包中，您有4个部分。1字节前导码，4字节访问地址，（2-39字节）PDU和3字节CRC。 CRC提供完整性检查数据包是否出现任何位损坏。 在实际的PDU中，有一个2字节的头段和高达37字节的有效负载。请注意，只有PDU使用前面讨论过的会话密钥（SK）进行加密，其余三个（前导码、访问地址和CRC）以明文形式发送。包嗅探可以用来破坏机密性。

Encryption and MACs

Encrypts and MACs PDU section

AES-CCM algorithm

AES-CCM is secure but the key exchange protocol is weak!



It relies on the Temporary Key which is very easy to crack.

Packet Sniff Process

Configuration

- Set modulation parameters to match BTLE (e.g., set to the same frequency, 2.4GHz)
- Tune to proper channel – it needs to know the channel hopping pattern.
  - Hop Increment
  - Hop interval
  - Both can be sniffed from connection packet or recovery in promiscuous mode

Promiscuous:混乱的 modulation, tune:调整

Two ways to get the hop increment and interval.

1. Sniff the connection packets in the advertisement phase. This is the easiest way. You can sniff the negotiating devices when they are setting up the scheduling for channel hopping to agree on these parameters so that they know how to hop. Have to listen on at most 3 advertisement channels.

Here we assume that the attacker has missed the connection advertising phase and data phase is in progress.

2. In data transfer stage, we can use pattern detection in promiscuous mode and use some mathematics to recover the hop increment and hop duration

首先在攻击者处配置包嗅探过程，我们将调制参数设置为匹配BTLE（2.4 GHz）。其次，它需要调到适当的频道。基本上需要监控1-2个频道，而不是所有频道。攻击者需要知道什么？需要通道跳跃模式：跳跃增量和跳跃持续时间，参与方将在特定通道上停留多长时间，以及哪个通道将用作下一个通道。有两种方法可以得到这种模式。1。在广告阶段嗅探连接数据包。这是最简单的方法。您可以在协商设备设置频道跳转计划时嗅探它们，以就这些参数达成一致，以便它们知道如何跳转。最多只能收听3个广告频道。这里我们假设攻击者错过了连接广告阶段，数据阶段正在进行中。2。在数据传输阶段，我们可以在混杂模式下使用模式检测，并使用一些数学方法来恢复跳跃增量和跳跃持续时间。

What Information do we need?

Preamble (1 byte)	Access Address (4 bytes)	PDU (2 to 39 bytes)	CRC (3 bytes)

- Access Address (AA)
  - Advertising: Fixed 0x8E89BED6
  - Connection: Actual device address
- Channel Information:
  - Hop interval
  - Hop increment

  Where to get this info: Connection packet!

  - easy if you get the starting packets

In the advertising phase all devices uses a fixed 0x8E89BED6 as Access address while in data phase it

has the actual device address when they are exchanging data. So it is very easy to identify packets exchanged in advertising phase (on one of three advertisement channels)
Channel information is contained in the connection packets exchanged during the advertising phase. Devices negotiate to come in agreement of how much is hop increment and what is the hop interval. Wait for the connection packet to arrive and retrieve the PDU. The connection packet contains the channel hop increment, hop interval in the payload. Easy if you can capture the starting packets
我们需要什么信息？1。需要访问地址才能知道此通信是否符合我们的兴趣。2。另一个是PDU，它包含在广告阶段具有通道信息（跳跃间隔和增量）的有效负载。 需要通道信息，以便攻击者知道通信方将如何使用通道跳跃机制发送数据。在广告阶段，所有设备使用固定的0x8e89bed6作为访问地址，而在数据阶段，当它们交换数据时，它具有实际的设备地址。因此，很容易识别在广告阶段（在三个广告渠道中的一个）交换的数据包，在广告阶段交换的连接数据包中包含渠道信息。设备协商得出跃点增量是多少，跃点间隔是多少。等待连接数据包到达并检索PDU。连接数据包包含有效负载中的通道跃点增量、跃点间隔。如果你能捕获开始的数据包，就很容易了

Promiscuous mode

What if I missed the connection packets?

- Capture a number of data packets.
- Perform the pattern search (promiscuous mode) to recover access addresses, hop interval and hop increment values (easily done!)

Crack the session key to decrypt PDU

广告阶段非常短，因此大部分时间攻击者都会错过此阶段（并错过连接数据包！）.在这种情况下，当通信设备交换数据包时，攻击者可以捕获许多数据包来执行模式搜索以恢复通道跳跃信息。

Recovery of Access Address

Preamble (1 byte)	Access Address (4 bytes)	PDU (2 to 39 bytes)	CRC (3 bytes)

What we know: Preamble (01010101)
What we have: Sea of bits
What we want: Access Address
10001110111101010101 → likely preamble!
10011100000100011001.. -> part of AA
100011001…100011101 →32 bit complete of AA! After that, PDU!

A preamble is “01010101” but “01010101” is not always a preamble.

CRC is here to help (for attacker)!

Attacker could use CRC (after PDU) to verify the access address and PDU.

– If CRC passes, the access address is correct. Otherwise, the “01010101” is false positive for preamble.

The PDU length is variable, 2-39 bytes? How to know where the CRC starts?

In Bluetooth, two connected devices need to exchange packets in each hop interval even though they do not have anything to transmit.

Those are empty packets. The PDUs of these empty packets are fixed to 2 bytes (header only) and the format and content is always the same. Attacker can use those empty packets to infer the access addresses.

前导码模式是固定的，但每种模式都不是前导码。这个模式可以出现在包的任何地方。攻击者如何确保他找到的位模式确实是一个前导码？如果模式不是前导码，则可能是误报。攻击者可以使用CRC验证模式是否确实是一个前导码。CRC是为了确保数据包的前一部分在传输中没有损坏。根据模式匹配前导码的位置重新计算CRC，如果与捕获的CRC匹配，则模式是实际的前导码，而不是假阳性。PDU长度是可变的，2-39字节？如何知道CRC从哪里开始？在蓝牙技术中，两个连接的设备需要在每个跃点间隔内交换数据包，即使它们没有任何要传输的内容。那些是空包。这些空数据包的PDU固定为2个字节（仅限头），并且格式和内容始终相同。攻击者可以使用这些空数据包推断访问地址。

Recovery of Hop Interval

Observation: 37 is a prime

Sit on one data channel and wait for two consecutive packets. Measure the time difference.

$$\Delta t/37 = \text{hop interval}$$

攻击者能够检索访问地址后，就能够识别设备。如果它想监视任何特定设备的数据，那么就可以启动攻击的第二阶段。首先，它要发现跳间隔，即两个设备在一个通道上的停留时间。这里的一个观察是，BT Smart中可用的信道数为37，37是一个质数。攻击者将要做的是坐在任何特定的数据通道上并获取一个数据包。等待它在同一个通道上得到下一个数据包，然后测量这两个数据包之间的时间差。这个时差除以37就是跳跃间隔。因为37是一个质数，为了回到同一个频道，设备必须跳37次。因此，我们测量在同一个通道上接收到的两个数据包之间的时间差，并将其除以37，得到跳跃间隔。因为37是一个素数，所以每一个通道在37次跳后循环回到同一个通道之前都会使用一次。例如，跳增量1在37之后会返回，2在37个跳间隔之后也会返回，依此类推。

Recovery of Hop Increment

Start on data channel 0, jump to data channel 1 when a packet arrives.

We know hop interval, we can calculate how many channels have been hopped between channel 0 and 1.

– Δt/hop interval = channel hops

对于跳跃增量，这有点棘手。跃点增量是在一个跃点中跳过/跳跃了多少个通道？
在前面的例子中，我们有7个通道跳跃，导致跳跃增量为7。为此，我们监视两个通道以获取跃点增量。从通道0开始，一旦在通道0上接收到数据包，就跳到下一个通道1。测量在通道1上接收另一个数据包时的时差。我们知道跳跃间隔（上一张幻灯片），时间差除以跳跃间隔将给出跳跃的频道数。在这之后，我们有一些数学计算来找到跳跃增量。

Calculate Hop Increment

$$\text{HopIncrement} * \text{channel hops} \equiv 1 \pmod{37}$$

$$\text{HopIncrement} \equiv \text{channel hops}^{-1} \pmod{37}$$

Apply Fermat's little theorem : $a^{p-1} \equiv 1 \pmod{p}$,

$$\text{HopIncrement} \equiv \text{channel hops}^{37-2} \pmod{37}$$

跳跃增量是攻击者想要知道的。跃点增量(hop increment)乘以channel hops（就是一共跳了多少个channel）恒等于 1（两个数据包之间的一个通道差（通道0和通道1）（mod 37）

Sniff summary

Connections packets

Promiscuous mode: recovery of

– Access Address

– Hop Interval

– Hop Increment

Hop Interval by monitoring one channel and Hop increment by monitoring two channels. And for cracking, we have only monitored maximum of two data channels, one at a time instead of 37.
通过监视一个通道的间隔和通过监视两个通道的跃点增量在已知跳间隔和跳增量的情况下，我们已经破解了通道跳变机制，因为攻击者可以遵循跳变模式，不必同时监视所有37个数据通道。对于破解，我们最多只能监控两个数据通道，一次一个，而不是37个。

Custom Key exchange protocol

Three pairing methods
- Just Works™
- 6-digit PIN
- 00B
- "None of these key pairing methods provide protection against a passive eavesdropper" – Bluetooth Core Spec

还有一件事没有被破解，那就是PDU。此部分使用前面幻灯片中讨论的会话密钥（SK）机制进行加密。对于蓝牙中的密钥交换，目前有三种配对方法：JustWorks、6位PIN和00B。然而，蓝牙核心规范指出，这些配对方法都不能防止被动窃听器。

Cracking Temporary Key

Temporary Key (TK) = AES (PIN, AES(PIN, rand XOR p1) XOR p2) (E_2 in slide 10)

Green – transmitted in plaintext
Red – wanted to know
PIN: integer between 0 and 999,999
JustWork™ is always 0!

回想一下，临时密钥tk用于身份验证。下面是生成tk的公式。绿色的信息以明文形式传输，并为通信方和攻击者所知。红色的只有蓝牙通讯设备知道。因此，临时密钥仅为通信设备所知。PIN用于生成TK，而TK用于生成SK。对于6位数的pin，pin值在0到999999（100万）之间，在JustWorks中最坏的情况是始终为0。

Cracking the PIN

Total Time to crack:
< 1 second

Subsequent Key crack

PIN → STK
STK → LTK
LTK → Session key!

- Every key is known.
- Attacker can learn about PDU
- Attacker can inject the packets in the networks with the session key!

通过pin，您可以知道临时密钥，通过临时密钥，您可以生成会话密钥。你已经知道跳槽了。事实上，当你需要一些数学计算来破解通道跳跃增量时，破解pin比通道跳跃更容易。

BTLP术语（SMART）

STK=短期密钥

LTK=长期密钥

Conclusions

Bluetooth has some security mechanisms

Bluetooth is not secure. There exist loopholes and they are easy to exploit.

Security in Bluetooth is yet to be improved.

Loophole:漏洞

WK07 Guest Lecture: IoT Security

Top cloud security risks

Data Breach / Loss

Insufficient Identity, Credential and Access Management (including Account Hijacking and Malicious Insiders)

Cyber Threats / Abuse and Nefarious Use of Cloud Services / DoS

Insecure Interfaces and APIs

Insufficient Due Diligence

System and / or Shared Technology Vulnerabilities

数据泄露/丢失

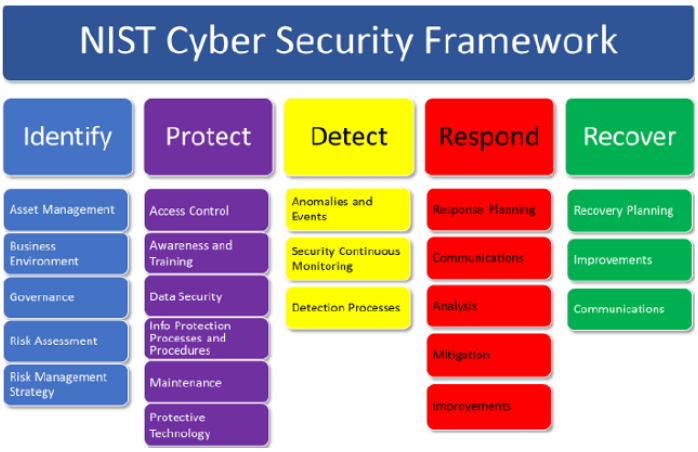
•身份、凭证和访问管理不足（包括帐户劫持和恶意内部人员）

•网络威胁/滥用和恶意使用云服务/DOS

•不安全的接口和API

•尽职调查不足

•系统和/或共享技术漏洞



Key IoT security challenges

Low powered	Limited computing capabilities mean difficulty implementing security controls (e.g. encryption).
Standards and regulation	Lack of government regulation and standards mean most are not designed with security in mind.
Lifecycle management	Keeping devices up to date is not something that is currently well managed, leading to security vulnerabilities potentially remaining unpatched indefinitely.
Transport protocols	The sheer number of emerging connection protocols makes them difficult to manage and secure.
Physical access	Devices are increasingly unlikely to be located in physically secure sites, significantly increasing the opportunities for attackers to compromise their integrity.
Number of devices	IoT deployments are largely uncontrolled environments (in the context of security) where the number of devices grows exponentially, making it extremely challenging for cyber security teams to govern and manage.
Availability and continuity	Devices are often not designed without alternate options to maintain availability of both functionality and connectivity in the event of failure.

有限的计算能力意味着难以实现安全控制（例如加密）。
缺乏政府监管和标准意味着大多数设计时都没有考虑到安全问题。
使设备保持最新不是当前管理良好的事情，这会导致安全漏洞可能无限期地保持未修补状态。
大量出现的连接协议使得它们难以管理和安全。
设备越来越不可能位于物理安全的站点，这大大增加了攻击者破坏其完整性的机会。
物联网部署在很大程度上是不受控制的环境（在安全环境中），设备数量呈指数级增长，这使得网络安全团队管理和管理非常具有挑战性。
设备的设计通常没有备用选项，以在发生故障时保持功能和连接性的可用性。

Common IoT security weaknesses

Interface (web/cloud/mobile/physical)

Security configurability

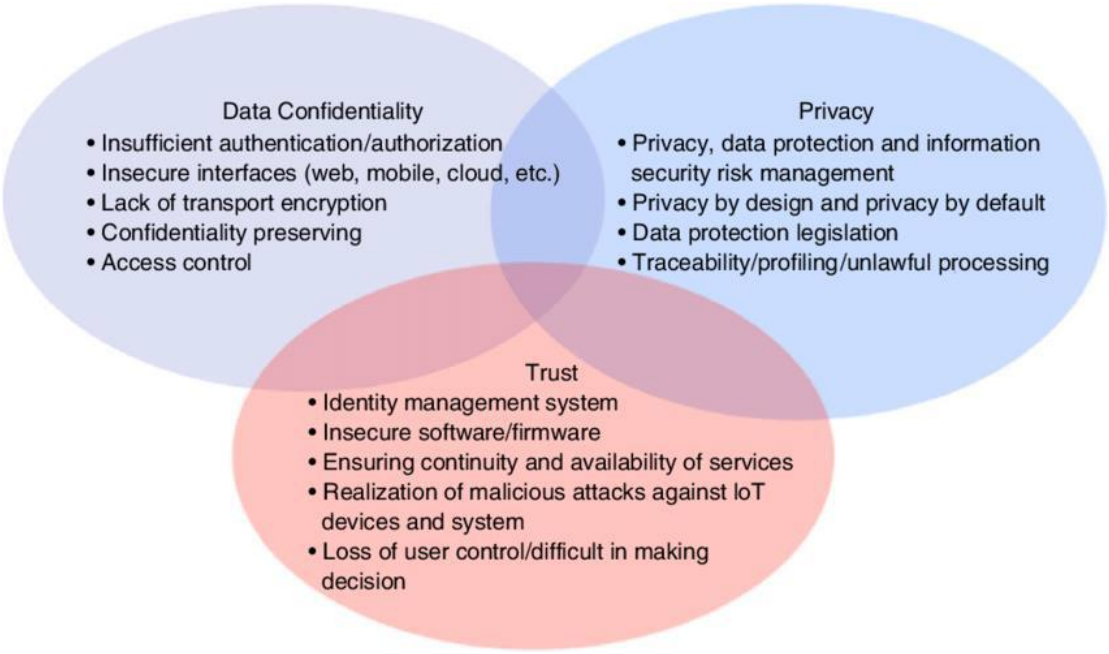
Authentication/authorisation

Network connectivity and services

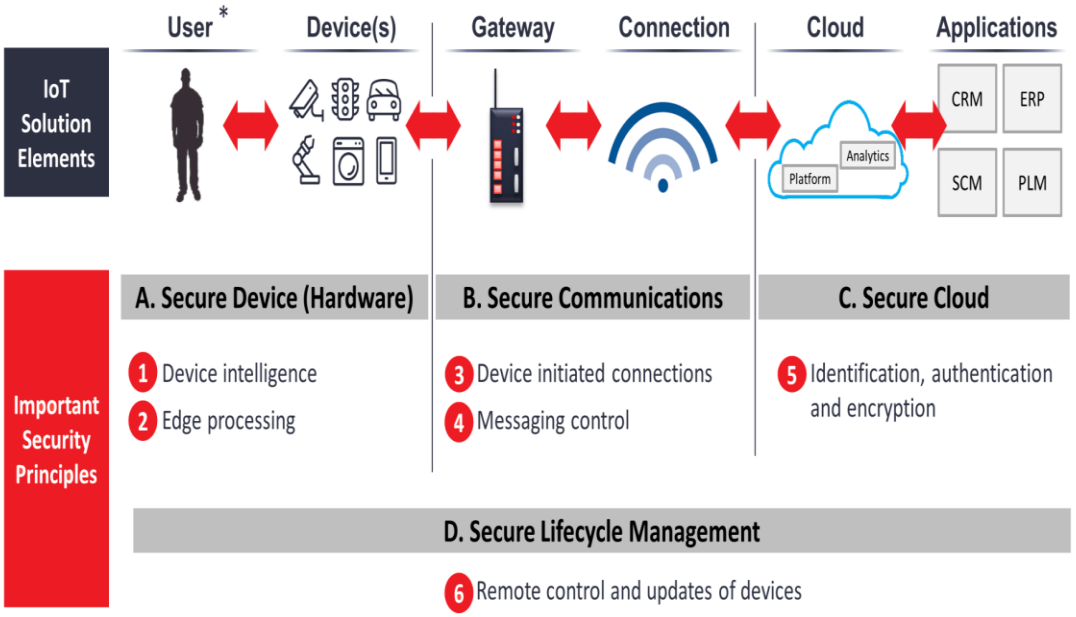
Confidentiality & integrity verification

Software & hardware (incl. firmware, memory, sensors)

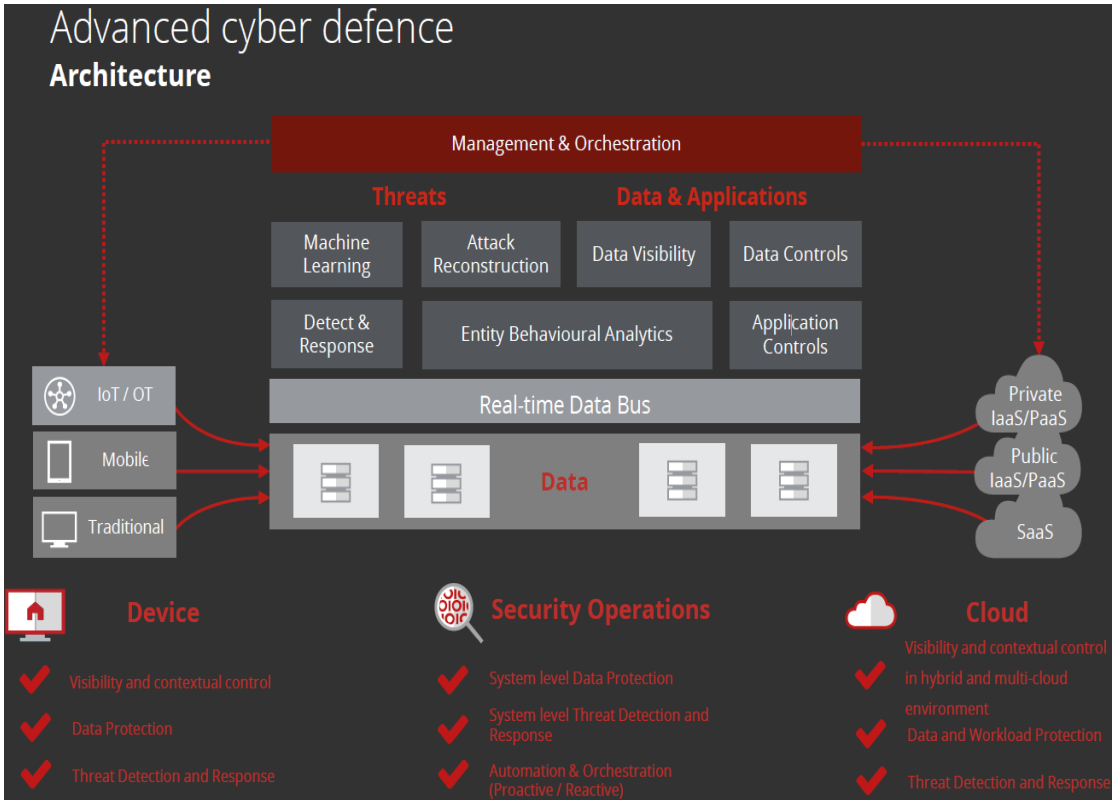
IoT data security concerns



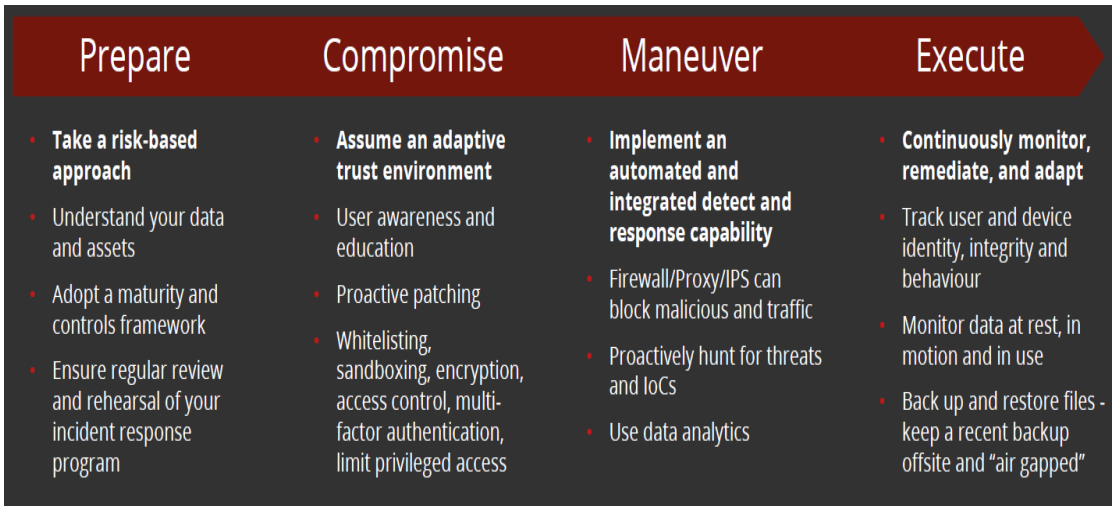
Six principles of IoT Security Architecture



Source: IoT Analytics ** User: can represent a person, device, system, or application*



Advanced cyber defence



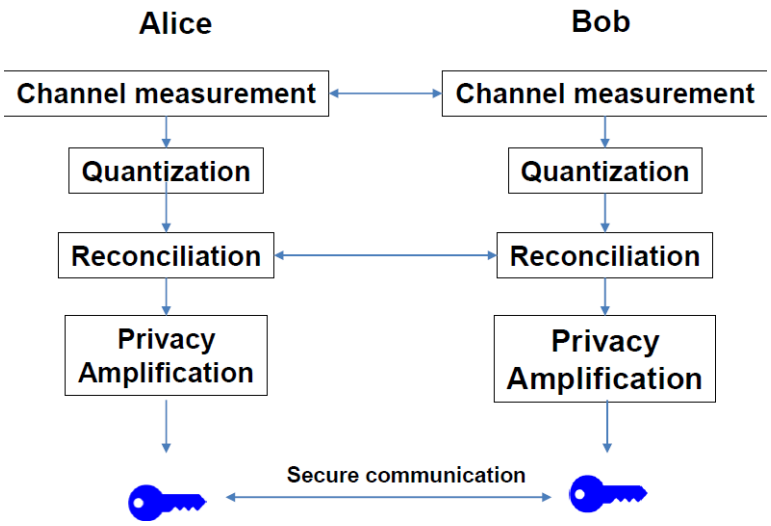
How do we solve the problem and give ourselves more time for fun?

•Do not underestimate the threat; assume they are already in the system

•Cooperation and collaboration is critical; enough with protectionism

•Stop playing the hapless victim; take responsibility and be proactive

LoRa



Privacy Amplification(放大)

Reconciliation step reveal information to attackers
Alice and Bob exchange a number of packets for this step

Universal hash function-SHA

After key generation, Alice and Bob can use symmetric encryption method to secure their communication such as AES.

WK-08: Security in Wireless Broadcast

Overview

Quick overview of Elliptic Curve Deffie-Hellman (ECDH) and Datagram TLS as lightweight solution for many wireless (IoT) solutions.

Security Challenges in wireless broadcast

Advanced techniques using hash-chains, Merkle Tree

Application case study of Code dissemination in a multi-hop wireless network

Dissemination:散播

Datagram TLS (DTLS)

SSL Designed to run on top of TCP
Datagram TLS developed later to run over connectionless UDP
– RFC 4347 for details
Already supported by several implementations
Very similar to TLS
– Needs extra control messages as UDP doesn't provide these like TCP
– Sequence number in record header to protect from Replay attack
If very lossy network, may have issues with lot of restranmissons for reliability

设计在TCP上运行的SSL •稍后开发的数据报TLS用于运行无连接的UDP -详细信息请参见RFC 4347。
•已得到多个实施的支持
•非常类似于TLS –需要额外的控制消息，因为udp不提供类似tcp的控制消息。 –记录头中的序列号以防止重播攻击
•如果网络非常有损，可能会出现许多可靠性方面的限制问题。

Elliptic Curve (ECC) Scheme

Key Agreement, aka, Elliptic Curve Deffie-Hellman (ECDH)
- Allows for establishment of shared secret similar to DH
- The shared key is then used for symmetric encryption or for further session/temporal key derivation

Digital Signature: Elliptic Curve Digital Signature Algorithm (ECDSA), allows use of public/private key for signing a message and verification of signature, more efficient than RSA based DSA.

密钥协议，aka，椭圆曲线deffie hellman（ecdh）o允许建立类似于dh的共享秘密 o然后将共享密钥用于对称加密或进一步的会话/时间密钥派生。•数字签名：椭圆曲线数字签名算法（ECDSA），允许使用公钥/私钥签署消息和验证签名，比基于RSA的DSA更高效。

Elliptic Curve Cryptography

Elliptic curve cryptography (ECC) is an approach to public-key cryptography based on the algebraic structure of elliptic curves over finite field.

ECC presents various benefits over RSA such as:
– fast computation
– small key size
– compact signatures.

For example, to provide equivalent security to 1024-bit RSA, an ECC scheme only needs 160 bits.

Compact:紧凑的 椭圆曲线密码学（ECC）是基于有限域椭圆曲线代数结构的公钥密码学方法。

•ECC提供了与RSA相比的各种好处，例如： –快速计算 –小钥匙尺寸 –紧凑签名。

•例如，为了提供与1024位RSA同等的安全性，ECC方案只需要160位。

Elliptic Curve Diffie-Hellman Key Exchange

1. Alice and Bob publicly agree on an elliptic curve E over a finite field Z_p.
2. Next Alice and Bob choose a public *base point* B on the elliptic curve E.
3. Alice chooses a random integer $1<\alpha<|E|$, computes $P = \alpha B$, and sends P to Bob. *Alice keeps her choice of α secret.*
4. Bob chooses a random integer $1<\beta<|E|$, computes $Q = \beta B$, and sends Q to Alice. *Bob keeps his choice of β secret.*

5. Alice computes $K_A = \alpha Q = \alpha(\beta B)$.
5. Bob computes $K_B = \beta P = \beta(\alpha B)$.
7. The shared secret key is $K = K_A = K_B$.

- Even if Eve knows the base point B, or P or Q, she will not be able to figure out α or β, so K remains secret!

1. Alice and Bob choose E to be the curve $y^2 = x^3+x+6$.
2. Alice and Bob choose the public base point to be B=(2,4).
3. Alice chooses $\alpha = 4$, computes $P = \alpha B = 4(2,4) = (6,2)$, and sends P to Bob. *Alice keeps α secret.*
4. Bob chooses $\beta = 5$, computes $Q = \beta B = 5(2,4) = (1,6)$, and sends Q to Alice. *Bob keeps β secret.*

5. Alice computes $K_A=\alpha Q = 4(1,6) = (4,2)$.
6. Bob computes $K_B = \beta P = 5(6,2) = (4,2)$.
7. The shared secret key is K = (4,2).

How to use keys?

- Rule of thumb:
- Public Key Cryptography: slow
- Symmetric Cryptography: fast

- Hence, do not encrypt large messages with Public Key Cryptography
- Encrypt a random, fresh symmetric key with Public Key Cryptography
- Use this key and symmetric encryption to encrypt a large message

- For signature, only sign the hash value of messages

- Send the encrypted key, signature, and symmetrically encrypted message to your communication partner

Challenges for Broadcast Security

Broadcast applications need security
- Packet injection or eavesdropping is easy

Security solutions for point-to-point communication not suitable secure for broadcast

Broadcast challenges
- Scale to large audiences
- Dynamic membership
- Low overhead (computation & communication)
- Packet loss
  - How to achieve reliability in broadcasts?

广播应用程序需要安全性 –包注入或窃听很容易

•点对点通信的安全解决方案不适合广播安全

•广播挑战 –扩展到大量受众 –动态会员 –低开销（计算和通信）-包丢失

Scale & Dynamics

Small groups contain up to ~100 members

Medium-size groups contain 100-1000 members

Large groups contain 1000-10^9 members – e.g. IoT

How does scale affect security?

Dynamic membership: members may join and leave at any time

How do dynamics affect security?

Communication Pattern

Group can be single-source broadcast
- One-to-many
- SSM: Single-source multicast, source-specific multicast

Multiple-source broadcast
- Some-to-many

All members broadcast
- Many-to-many

Reliable Broadcast Transmission

How to reliably and scalably disseminate data to large numbers of receivers?

Challenges
- Ack implosion problem if receivers return Ack to sender for received packets
- Nack implosion problem is severe as well
- For large numbers of receivers, there usually is a fraction of them that do not obtain message
- Local repair mechanisms (create tree topology and ask upstream parent for packet) faced numerous scalability difficulties
  - Accumulate ack (delayed) and send to parent node on the tree.

Disseminate:宣传，传播; implosion:内爆 •如何向大量接收器可靠、可伸缩地传播数据?

–如果接收端将接收到的数据包的ACK返回给发送方，则ACK内爆问题

–Nack内爆问题也很严重

–对于大量的接收器，通常有一小部分接收不到信息。

–本地修复机制（创建树拓扑并向上游父级请求数据包）面临许多可扩展性困难。o累积ACK（延迟）并发送到树上的父节点。

End-to-End approach

- Trusted server authenticates each node and distributes the key.
- Use well-known E2E protocols such as (D)TLS, IPSEC, SSL.

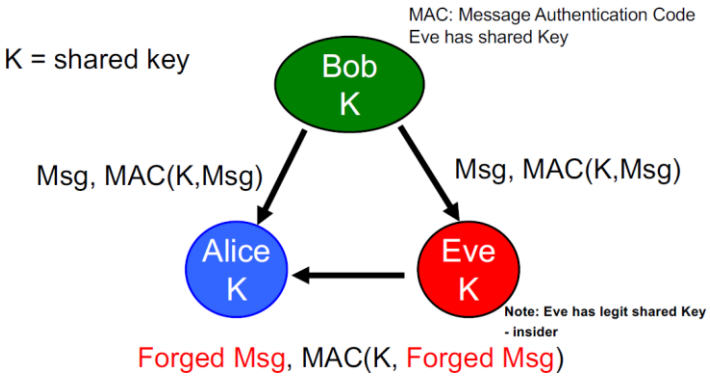
Pros

- Higher security level
- Dynamic grouping

Cons

- Extremely higher ciphertext overhead due to one by one transfer
- Higher computation requirements

Shared Key: Easy to Forge



Forge:伪造

Asymmetric Key: Digital Signature

Sign each packet and verify using Asymmetric key

However, Signatures are expensive esp. for low end processors, e.g., RSA 2048:

- High generation cost (~1 millisecond), High verification cost (~0.1 millisecond), High communication cost (256 bytes/packet)

If we use one signature over multiple packets, intolerant to packet loss

Intolerant:无法忍受 对每个数据包签名并使用非对称密钥进行验证 但是，签名成本很高，特别是对于低端处理器，例如RSA 2048： •发电成本高（~1毫秒），验证成本高（~0.1毫秒），通信成本高（256字节/包） 并且如果我们在多个数据包上使用一个签名，则不允许数据包丢失。

Trivial broadcast key distribution

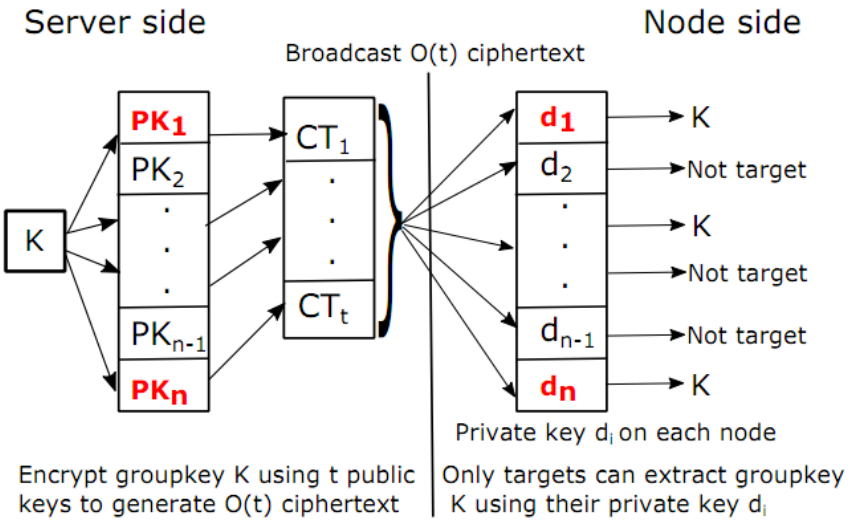


Figure 1: Trivial Broadcast Encryption scheme in an n nodes network targeting t nodes. K: shared group key. PK: Public Key. CT: Ciphertext. d: Decryption using each node's private key.

Trivial:不重要的

Very simple and effective system. Easy to implement.

The only issue is linear ciphertext O(t) for key distribution.

Good for applications with smaller number of target nodes.

Possible solution for non-sensor applications such as Desktop, smart-phone

非常简单有效的系统。易于实施。

•唯一的问题是密钥分配的线性密文o（t）。

•适用于目标节点数量较少的应用程序。

•适用于桌面、智能手机等非传感器应用的可能解决方案

Shamir's Secret Sharing Schemes

Split secret into multiple parts aka shares

Threshold: a minimum amount of shares needed to unlock the secret/vault.

- Example: To access an important resource, a key can be generated by use of shares from various participants. Even if a some shares get compromised, they can't Secret.

Maths beyond scope: e.g. parabola uniquely described with 3 points,

- multiple points (shares) can be distributed, threshold = 3 to define a parabola uniquely and find the secret S (e.g coefficient a0 of the polynomial)

Parabola：抛物线，coefficient:系数，polynomial:多项式

将秘密分成多个部分，即股份 •**阈值**：解锁秘密/保险库所需的最小份额。

–示例：要访问重要资源，可以使用来自不同参与者的共享来生成密钥。即使一些股票被泄露了，他们也不能保密。

•超出范围的数学：例如，用3个点唯一描述的抛物线，

–可以分布多个点（共享），阈值=3以唯一定义抛物线并找到秘密s（例如多项式的系数a0）

Hash Chain -basics

Client generate 1000 hashes for password

– Suggested by Lamport for password protection

Server stores H^{1000}(password)

Client willing to authenticate sends H^{999}(password)

Server computes H^{1000}(password) = H(H^{999}(password))

- Match found, store H^{999}(password)for next time

Eavesdropper can't use H^{999}(password) since server expects H^{998}(password)

客户端为密码生成1000个哈希 –由lamport建议用于密码保护

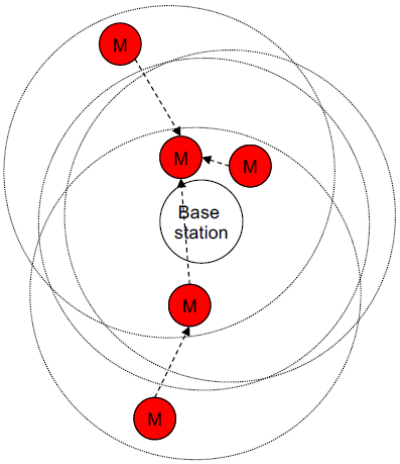
•服务器存储h1000（密码）

•客户端愿意验证发送的H999（密码）

•服务器计算h1000（密码）=h（h999（密码）） –找到匹配项，下次存储H999（密码）

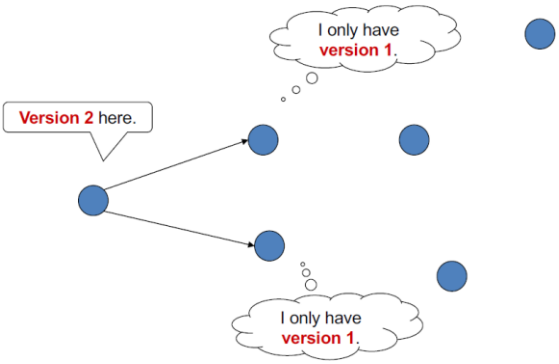
•窃听器不能使用H999（密码），因为服务器需要H998（密码）

Secure Code Image Programming in WSN



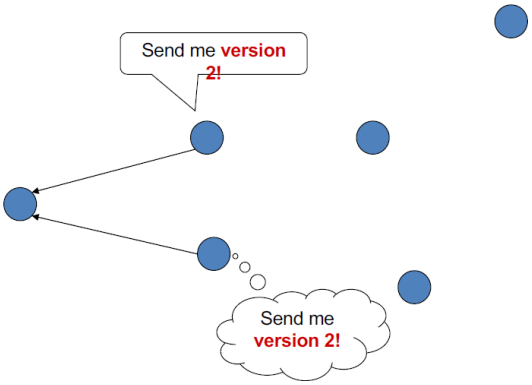
Epidemic propagation

1. Nodes periodically advertise

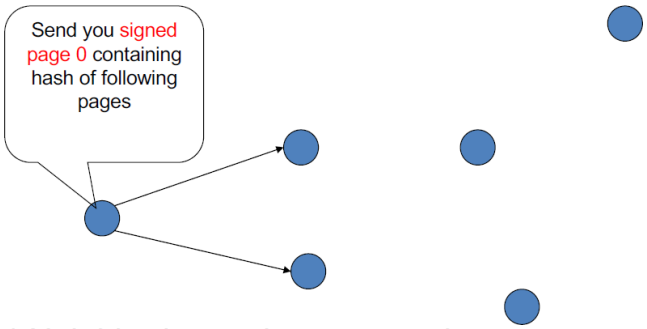


Periodically:周期的 epidemic:传染的

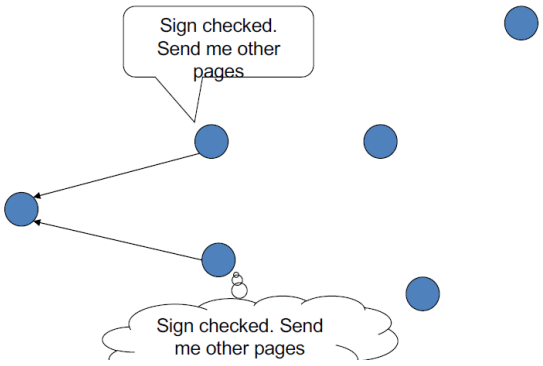
2.Neighboring nodes request new version



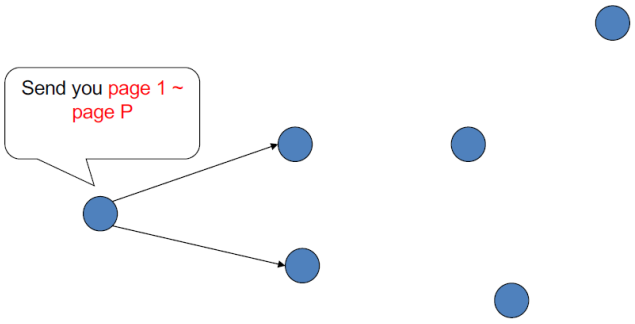
3. Server sends signed hash of each page



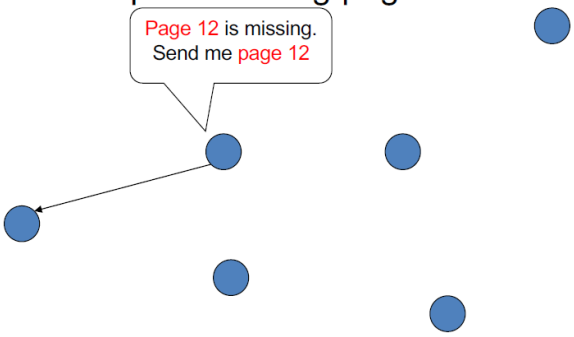
4.Neighboring nodes request data



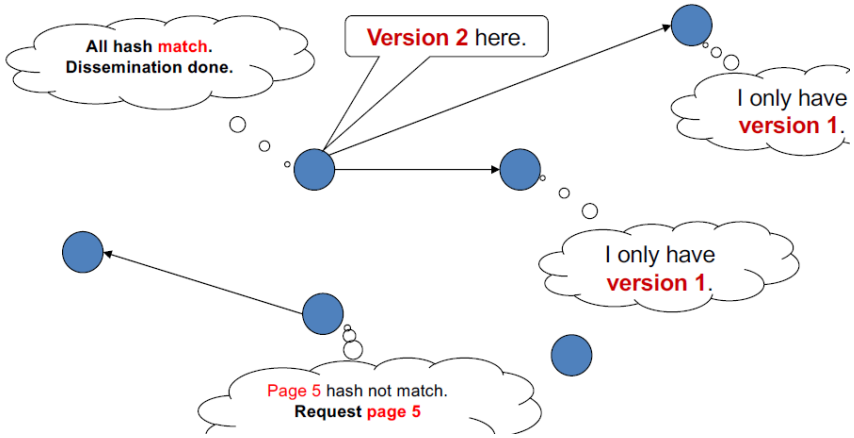
5. Transmit each page



6.Nodes request missing page



7.Check hash of each page



Threat Model

Adversary has power laptop/computer

External Attacks: adversary doesn't control any node

- Eavesdrop, inject forged messages, replay intercepted messages, impersonate valid node, *Wormhole Attack* (fake non-existing links), *Sybil Attack* (one node presents several identities to defeat fault tolerance)

Internal Attacks: take control of nodes

- manipulate nodes until detected, intercept sensitive information even if encrypted, (selectively) drop packets, and launch Sybil attacks

Adversary can distribute illegal code, drain battery, disconnect network etc.

对手有电源笔记本电脑/电脑

•外部攻击：对手无法控制任何节点 –窃听、注入伪造的消息、重放拦截的消息、模拟有效节点、虫洞攻击（假不存在的链接）、Sybil攻击（一个节点提供多个标识以消除容错性）

•内部攻击：控制节点 –操作节点直到检测到，拦截敏感信息，即使加密，（有选择地）丢弃数据包，并发起Sybil攻击

•对手可以分发非法代码、耗尽电池、断开网络等。

Setting: Metrics

Security metrics
- Can external adversary forge a message?
- Can single or several receivers forge a message that at least one other receiver accepts?

Efficiency metrics
- Communication overhead
- Computation overhead
- Storage overhead
- Delay for authentication / signature
- Resilience to packet loss

安全指标 –外部对手能伪造信息吗？ –单个或多个接收器能否伪造至少一个其他接收器接受的消息？

效率指标 –通信开销 –计算开销 –存储开销 –认证/签名延迟 –数据包丢失的恢复能力

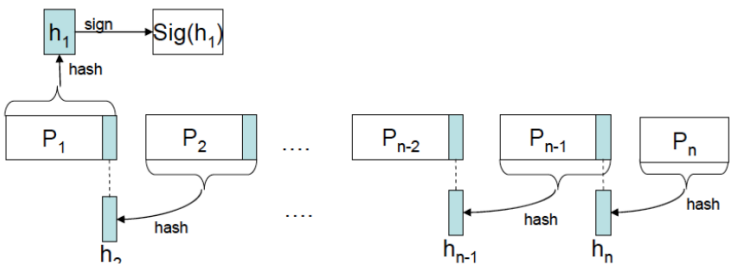
Code Dissemination – Hash Chain

Image fragmented into fixed size segments – called pages

Calculate hash value (h_n) of last page P_n and append this to previous page P_{n-1} , until first page is reached

Hash value of first page P1 signed with private key of base station (BS)

Receivers verify this using public key of BS and recursively authenticate each page



图像分割成固定大小的片段——称为页面

•计算最后一页PN的哈希值（hn），并将其附加到上一页PN-1，直到到达第一页。

•用基站私钥（bs）签名的第一页p1的哈希值

•接收者使用BS的公钥验证这一点，并递归地验证每个页面

Merkle Hash Tree

Hash Chain elements can only be revealed sequentially

– May not be acceptable for many applications e.g. P2P

Merkle-trees : allowing for the pre-authentication of a set of values with a single digital signature (on the root u_0 of the tree) and for the revelation of those values in *any* order

when revealing a value v_i, reveal all the values assigned to the sibling vertices on the path from v_i' to the root

– One way hash property ensures: this disclosure not sufficient to calculate any other unrevealed v_j

Hash Chain 只能按顺序显示哈希链元素 –可能不适用于许多应用，如P2P。

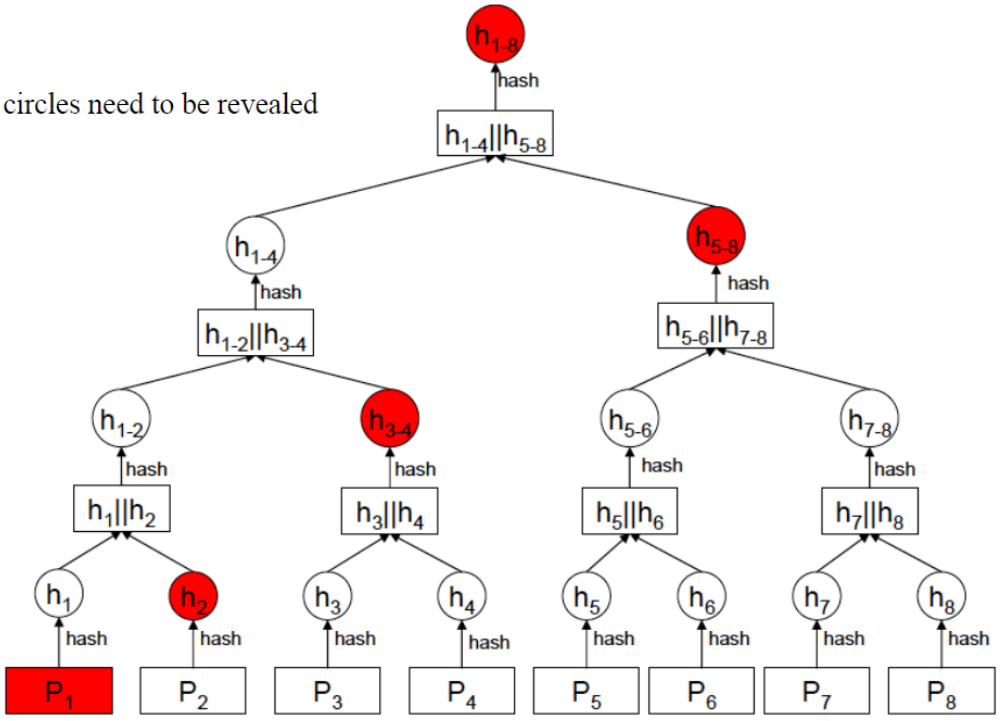
•Merkle Trees：允许使用单个数字签名（在树的根U0上）对一组值进行预验证，并**允许以任何顺序**显示这些值。

•显示值vi时，显示从vi到根路径上分配给同级顶点的所有值。

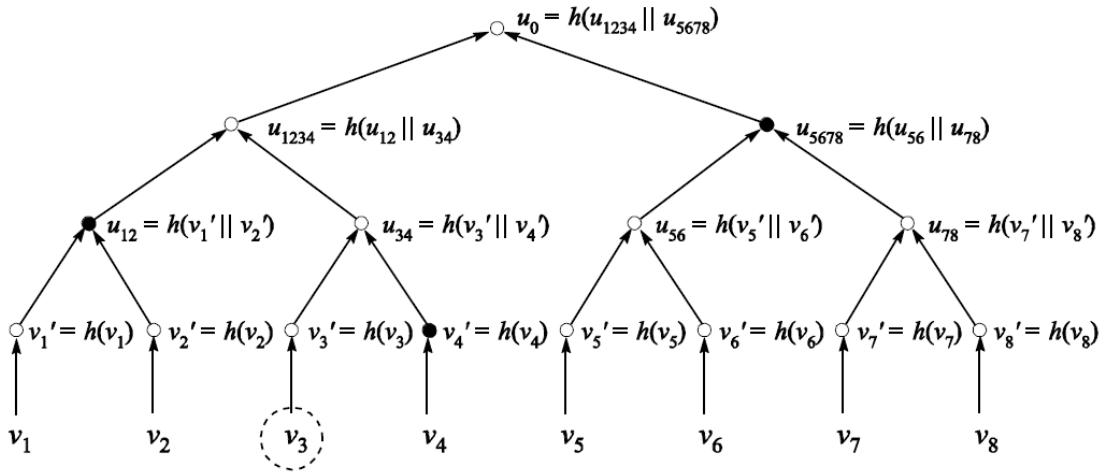
–单向哈希属性确保：此公开不足以计算任何其他未公开的VJ

Code Dissemination-MT Example

*Red circles need to be revealed



Merkle Tree Another Example



- Authenticate v_3, v_3 is revealed together with v_4', u_{12}, u_{5678})
- verifiers hash the revealed values appropriately and check if the result is u_0
  - $u_0 = H(H(u_{12} || H(H(v_3) || v_4')) || u_{5678})$

Code Dissemination – MT

We could further divide each page into packets

Take hash of each packet

Construct a Merkle tree of these hash values and get root for each page

Recursive hash value of the roots of each page is done using Hash-Chain

Digital signature for the first root value is sent

Along with root value, also need sibling vertices as discussed earlier

我们可以进一步把每一页分成包

•获取每个包的哈希值

•构造这些哈希值的Merkle树，并为每个页面获取根目录

•每个页面根的递归哈希值使用哈希链完成

•发送第一个根值的数字签名

•除了根值，还需要前面讨论过的同级顶点

Signature Based Attack

Signature expensive operation on small embedded devices

Attacker keeps sending forged data

Node depletes energy in doing signature verification

These approaches need to know number of data-packets in advance

– may not be available in many applications

Deplete:耗尽

在小型嵌入式设备上的签名昂贵操作

•攻击者不断发送伪造数据

•节点在进行签名验证时消耗能量

•这些方法需要提前知道数据包的数量。–在许多应用中可能不可用

Small RSA Signature

Idea
- Use short-lived small RSA keys (e.g., 384 bit)
- Periodically send out new public key signed with strong signature
  - 48 byte signature per packet
  - Signature generation ~0.1ms, verification ~10us

Advantages
- Relatively low computation overhead
- No buffering, no verification delay
- Scalable

Disadvantages
- Relatively high communication overhead (> 50 bytes/packet)
- Need time synchronization
- Not perfectly robust to packet loss

–使用寿命短的小型RSA密钥（如384位）

–定期发送带有强签名的新公钥：每包48字节签名；签名生成~0.1ms，验证~10us

优势 –计算开销相对较低 –无缓冲，无验证延迟 -可扩展的

•缺点 –通信开销相对较高（>50字节/包）–需要时间同步 –对数据包丢失不完全可靠

Learning Outcomes

Appreciate how Broadcast/multicast fundamentally changes protocol design space for authentication.

Understand tradeoff between reliability and security
- Key exchange etc must be reliable?

Understand hash chain and Merkle tree algorithms and their application for security.

Understand various threats in adhoc, wireless sensor networks and IoT

Appreciate that different points of the design space have different “best solution”

了解广播/多播如何从根本上改变认证的协议设计空间。

•了解可靠性和安全性之间的权衡 –密钥交换等必须可靠?

•了解哈希链和Merkle树算法及其安全应用。

•了解临时、无线传感器网络和物联网中的各种威胁

•了解设计空间的不同点有不同的最佳解决方案

WK-09 Guest Lecture: Privacy and Its Impacts & CRYPTO CURRENCIES

Security vs. Privacy

Bypass the protection
Without authorization
Illegal
Gain information directly
Will cause loss directly

- Don't have to hack a system
- All the data is there (usually)
- Gain information via sophisticated methods (e.g., AI)
- Illegal or legal?

k-Anonymity: Example

- *k*-anonymity [Sweeney 2002]
  - requires that each (Age, ZIP) combination can be matched to at least *k* patients
- How?
- Make Age and ZIP less specific in the medical records

Name	Age	ZIP
Andy	20	10000
Bob	30	20000
Cathy	40	30000
Diane	50	40000

adversary's knowledge

Age	ZIP	Disease
20	10000	flu
30	20000	dyspepsia
40	30000	pneumonia
50	40000	gastritis

medical records

2-anonymous table

"generalization"

Name	Age	ZIP
Andy	20	10000
Bob	30	20000
Cathy	40	30000
Diane	50	40000

adversary's knowledge

Age	ZIP	Disease
[20,30]	[10000,20000]	flu
[20,30]	[10000,20000]	dyspepsia
[40,50]	[30000,40000]	pneumonia
[40,50]	[30000,40000]	gastritis

medical records

Intuition:

- Hiding in a group of *k* is not sufficient
- The group should have a diverse set of sensitive value

Name	Age	ZIP
Andy	20	10000
Bob	30	20000
Cathy	40	30000
Diane	50	40000

adversary's knowledge

QI | sensitive

Age	ZIP	Disease
[20,30]	[10000,20000]	dyspepsia
[20,30]	[10000,20000]	dyspepsia
[40,50]	[30000,40000]	pneumonia
[40,50]	[30000,40000]	gastritis

2-anonymous table

l-Diversity [Machanavajjhala et al. 2006]

- Approach: (similar to *k*-anonymity)
  - Divide tuples into groups, and make the QI of each group identical
- Requirement: (different from *k*-anonymity)
  - Each group has at least *l* "well-represented" sensitive values
- Several definitions of "well-represented" exist
  - Simplest one: in each group, no sensitive value is associated with more than 1/*l* of the tuples

Age	ZIP	Disease
[20,30]	[10000,20000]	flu
[20,30]	[10000,20000]	dyspepsia
[40,50]	[30000,40000]	pneumonia
[40,50]	[30000,40000]	gastritis

2-diverse table

UNSW

l-Diversity: Vulnerability

- Intuition:
  - It is not sufficient to impose constraints of the diversity of sensitive values in each group
  - Need to take into account the adversary's background knowledge (e.g., males are unlikely to have breast cancer)

Name	Age	ZIP
Andy	20	10000
Bob	30	20000
Cathy	40	30000
Diane	50	40000

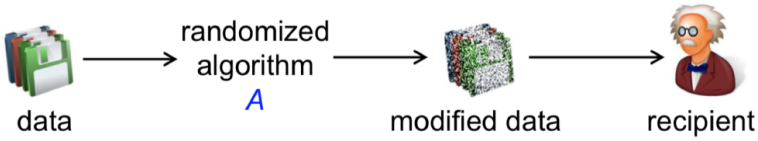
adversary's knowledge

Age	ZIP	Disease
[20,30]	[10000,20000]	breast cancer
[20,30]	[10000,20000]	dyspepsia
[40,50]	[30000,40000]	pneumonia
[40,50]	[30000,40000]	gastritis

2-diverse table

Differential Privacy: Intuition

- In general, we should only publish information that does not highly depend on any particular individual
- This motivates the definition of differential privacy

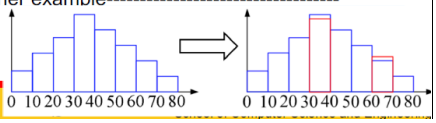


The Laplace Mechanism

- In general, if we want to release a set of values (e.g., counts) from a dataset,
  - We add i.i.d. Laplace noise to each value to achieve differential privacy
- This general approach is called *the Laplace mechanism*
- Figuring out the correct amount of noise to use could be a research issue
  - λ noise leads to (1/ λ)-differential privacy
  - 1/ λ = ϵ

----------Another example----------

- 2 λ noise leads to (1/ λ)-differential privacy



UNSW

1. Collision Free

Assume, given a space of *N* possible hash values, already picked a single value.

N−1 remaining values that are unique from the first
- The probability of randomly generating two integers that are unique from each other I = *(N-1)/N*

N−2 remaining values (out of a possible *N*) that are unique from the first two
- Probability of randomly generating three integers that are all unique = *(N-1)/N x (N-2)/N*
  - Multiply the probabilities because each random number generation is an independent event

In general,
- The probability of randomly generating *k* integers that are all unique is:

$$(N-1)/N \times (N-2)/N \times \cdots \times (N-\{k-\})N \times (N-\{k-1\})/N$$

Can be approximated by:

$$e^{-k(k-1)/2N}$$

1. Collision Free cont.

- If you have 2^{130} randomly chosen inputs
  - 99.8% chance that two of the inputs will collide
- This is the case regardless of what ***H( )*** is
- Takes a long time to compute

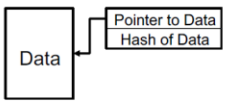
- **No** *H( )* has been proven to be collision free

- Despite – safe to assume
  - If *H(x) = H(y)* then *x = y*
  - Therefore you can use *hash* to recognise large files *(fingerprint)*

Hash Pointers

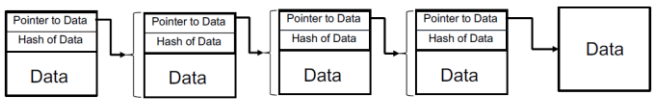
Hash pointer
- Pointer to where some information is stored
- Cryptographic has of the info

Hash stored in the hash pointer is the
- Hash of the ***whole data*** of the previous block
- And the ***hash pointer*** to the block before that one



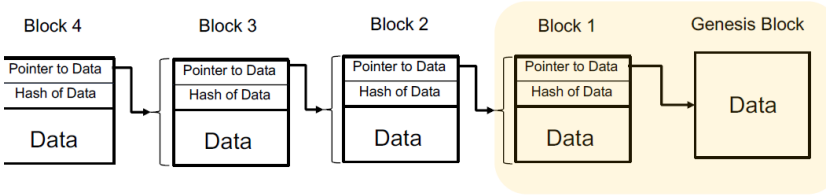
Hash Pointers cont.

- Can be used to create any data structure as long as there are no cycles
- Simple linked list

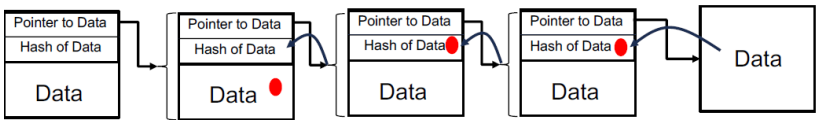


- If we have a hash pointer
  - Ask to get the information back
  - Verify that it has not changed

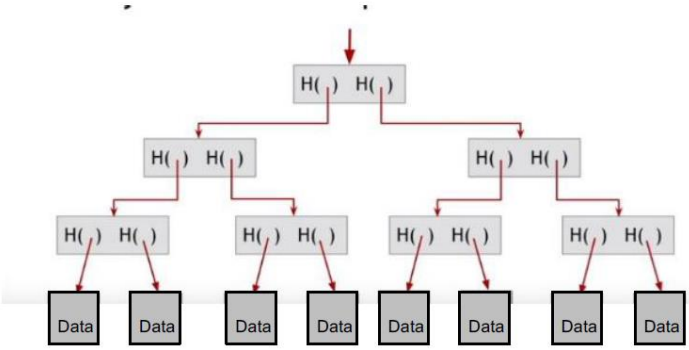
Blockchain



- Tamper-evident block



Binary Tree – Merkle Tree



Advantages

- Can hold many items but need remember the root hash
- Can verify membership in O(log n) time/space
- Sorted Merkle trees
- Can verify non membership in O(log n)
  - Show items before, after missing one

- More generally
  - Hash pointers can be used in any pointer-based data structure that has no cycles

Digital Signatures again.

- Only you can sign, but anyone can verify
  - Uses public private key pair
- Signature is tied to particular document
  - Cannot cut and paste
  - Public Key == an identity

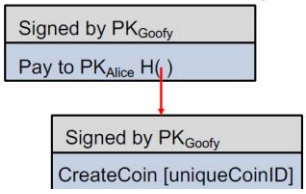
- Decentralised ID
  - Anybody can make a new identity at any time
  - No central point of coordination
  - *Address* in crypto currencies

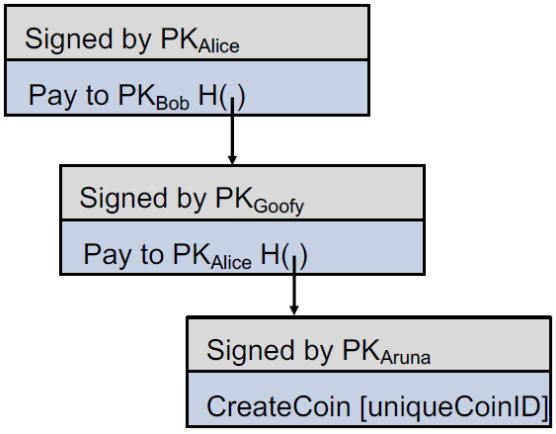
A Simple Cryptocurrency

Goofy coins

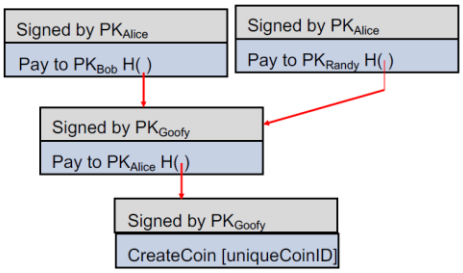
Goofy can create new coins

Whoever owns a coin can spend it



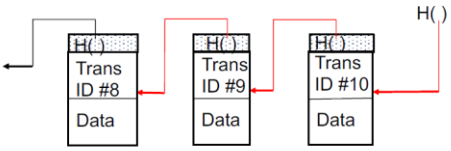


Simple Cryptocurrency – Double Spending



Overcoming Double Spend

- Creator published a history of all transactions
  - Blockchain, signed by the creator (goofy)



- Can optimise by putting multiple transaction into the same block
- What does the history do?
  - Detect double spending

Overcoming Double Spend cont.

- *CreateCoins* transaction creates new coins

transID #10	type:CreateCoins	
Coins created		
num	value	recipient
0	3.3	0x....
0	1.4	0x....
0	7.1	0x....

Coin10(0)
Coin10(1)
Coin10(2)

Paycoins

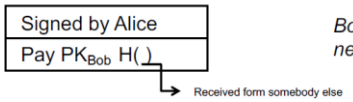
- *PayCoins* transaction consumes (and destroys) some coins and creates new coins of the same total vale

transID #10	type:PayCoins	
Consumed coinIDs: 68(1), 42(0), 72(3)		
Coins created		
num	value	recipient
0	3.3	0x....
0	1.4	0x....
0	7.1	0x....
signatures		

- Valid if:
  - *Consumed coins are valid*
  - *Not already consumed*
  - *Total value = Total value in, and*
  - *Signed by owners of all consumed coins*

Consensus

- Definition
  - Protocol terminates all legitimate nodes decide on the same value
  - The value has to have been proposed by a some legitimate node
- P2P Bitcoin
  - Alice broadcasts the transaction to all nodes on P2P network

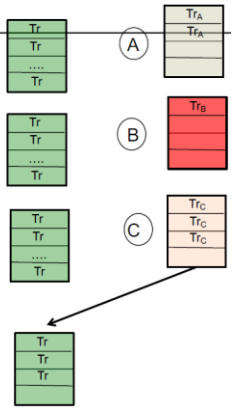


Bob's machine does not need to be on the network

- The which transactions were broadcast and order in these transactions took place

Consensus – Bitcoin

- All nodes have
  - A sequence of blocks of transactions that they have received consensus on
  - A set of outstanding transactions they head about
- Select any valid block
- Really hard technical problem



Why

- Nodes may crash or be malicious
- Network is imperfect
  - Not all pairs of nodes are connected
  - Faults in the network
  - Latency
- Many impossibility results
  - Byzantine generals problem
  - Fischer-Lynch-Paterson – consensus impossible with a single faulty node
- Some well known protocols
  - Paxos: Never produces inconsistent results, but can get stuck (rarely)

Some Observations

- Models say more about the model than the problem
- Models were developed to study systems like distributed data bases
- Bitcoin is a practical solution
- Bitcoin
  - Introduces the notion of incentives
  - Embraces randomness
    - No specific end-point
    - Consensus happens over long time scales (~1 hour): as time goes on the probability increases

Consensus without Identities

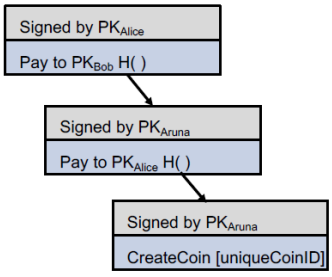
- Identity is hard in P2P systems
  - No central entities
- Pseudonmity is a goal
- Implicit Consensus
  - Assume that it is possible to pick a random node
  - In each round pick a node at random
  - The selected ndoe proposes the next block in the chain
  - Others nodes accepts the and extends the blockchain, if all transactions are valid(unspent, valid signature) or
  - Reject this block extends the blockchain from an earlier block

Consensus Algorithm

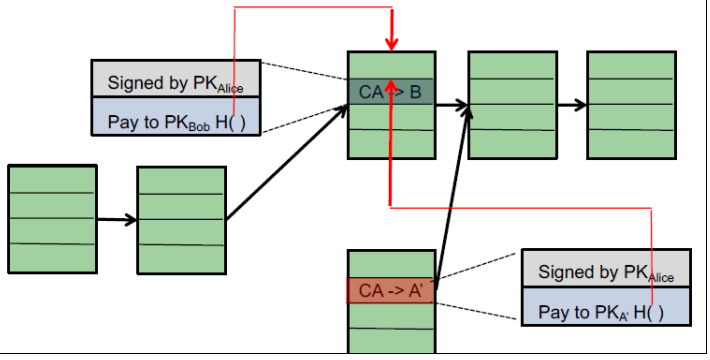
1. New transactions are broadcast to all nodes
2. Each node collects new transactions into a new block
3. In each round a random node gets to broadcast its block
4. Other nodes accepts the block
5. Nodes express their acceptance of the block by including its hash in the next block they create
6. Extend the longest valid branch

Validation

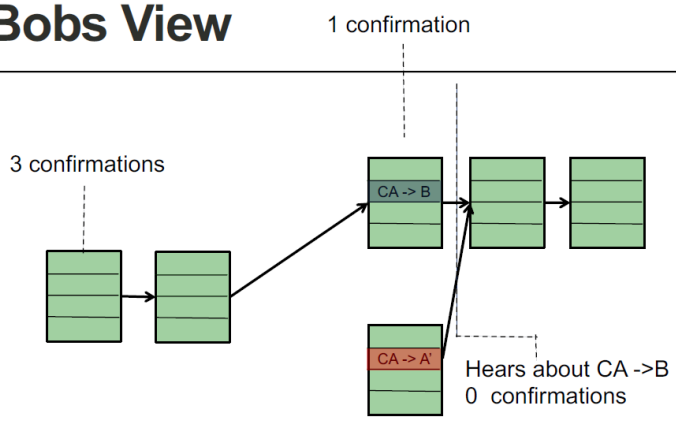
- Stealing somebody else's bit coins?
  - Cannot because cannot forge signature
- Denial of service by not including any of the transactions from a give user
  - Only an a delay
- Double spending



Double Spend (1)



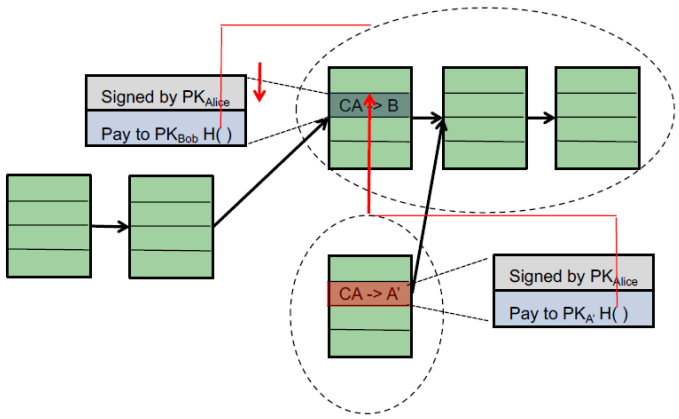
Bobs View



- Double spend probability decreases exponentially with the number of confirmations
- Common heuristic: 6

Incentives not to act maliciously

Reward the node that created these blocks



Punish the node that created this block

Incentive 1: Block Reward

- Creator of a block gets to:
  - Include special con-creation transaction in the block
  - Choose the recipient address of this transaction
- Block creator gets to to “collect” the reward only if the block ends up on the long-term concensus branch
- Value is fixed: currently 25BTC, halves every 4 years

Summary

- Identities
  - No real-world ID any user can create an ID
- Transactions
  - Messages that are broadcast to the P2P network giving instructions as to what to do with coins
  - Coins are chain of transactions
- Peer to Peer Network
  - Transfers the transactions to all the nodes in the network – best effort. Security comes form the blockchain and consensus protocol
- Blockchain and Consensus
  - Transaction to be in a blockchain needs a number of confirmations (6 is the heuristic). Could have a orphan of blocks.
- Hash puzzles and mining
  - Randomly finding nodes

Sample Question

Sample Question1 - Solution

Answer:
Since IV = 11, the key stream is 111110100000 .....

Given, m = 10100000
Hence, ICV = 1010 XOR 0000 = 1010
The three fields will be:

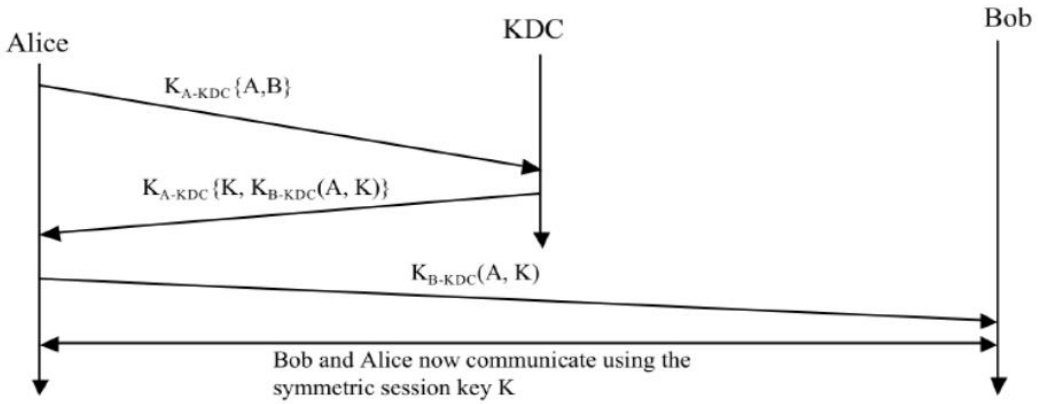
IV: 11
Encrypted message: 10100000 XOR 11111010 = 01011010
Encrypted ICV: 1010 XOR 0000 = 1010 (remember the key bits from 9-12 above are all 0000.

Sample 2

Suppose Alice wants to communicate with Bob using Symmetric Key Cryptograpy using a session key Ks. In this question we use a Key Distribution Centre in place of Public Key Cryptography. KDC shares a unique secret symmetric key with each registered user. For Alice and Bob, denote these keys by $K_{A\text{-}KDC}$ and $K_{B\text{-}KDC.}$ Design a scheme that uses the KDC to distribute the session key: A message from Alice to the KDC; A message from KDC to Alice; and finally a message from Alice to Bob. The first message is $K_{A\text{-}KDC}$ (A,B). Using the notation $K_{A\text{-}KDC}$, $K_{B\text{-}KDC}$, S, A, and B answer the following questions:

A) What is the second message?

B) What is the third message?



First message; Request a session key with Bob from KDC
Second Message: KDC to Alice, sends Key as well as the key encrypted with Bob's shared key with KDC for Bob to verify that it is from a legitimate source.
Third Message: Alice sends this encrypted key to Bob (saying that use this key for session With A (Alice)

Correction: Curly braces should Be parenthesis in first message

Sample 3

Suppose Bob initiates a TCP connection to Trudy who is pretending to be Alice. During the handshake, Trudy sends Bob Alice's certificate. In what step of the SSL handshake algorithm will Bob discover that he is not communicating with Alice?

Answer:

After the client will generate a pre-master secret (PMS), it will encrypt it with Alice's public key, and then send the encrypted PMS to Trudy. Trudy will not be able to decrypt the PMS, since she does not have Alice's private key. Thus Trudy will not be able to determine the shared authentication key. She may instead guess one by choosing a random key. During the last step of the handshake, she sends to Bob a MAC of all the handshake messages, using the guessed authentication key. When Bob receives the MAC, the MAC test will fail, and Bob will end the TCP connection.