

Internet and Network Security Fundamentals

In Conjunction with



28 June – 2 July, 2011

Suva, Fiji

Presenters

- Champika Wijayatunga
Training Manager, APNIC
champika@apnic.net

Day 1

- Intro to Internet Resources
 - Internet Resource Registration
 - Whois Database
- Intro to Security
- Network Security Concepts
 - Terminology
- Threats and Attacks
 - Attacks on Different Layers
- Cryptography
- Public Key Infrastructure
- Network Infrastructure Security

Day 2 - 3

- Security and Information Policy
- Security on Different Layers
- Server Security
- DNS Security
 - Reverse DNS
 - ACLs
 - TSIG
 - DNSSEC
- Understanding TCP/IP

Day 4 - 5

- Network Analysis
- Forensics fundamentals
- Anatomy of a network attack
 - Miscreants, Motivations, & Misconceptions
- Modern Attacks
- Botnets
- DDoS, & Botnet financials
- Penetration Testing



Acknowledgements

- **Merike Kaeo** from Double Shot Security and the author of “Designing Network Security”.
- **APNIC** acknowledges her contribution and support with appreciation and thanks

Internet Resource Registration – Whois Database

What is the APNIC Database?

- **Public network management database**
 - Operated by IRs
 - Public data only
 - For private data: Please see “Privacy of customer assignment” module
- **Tracks network resources**
 - IP addresses, ASNs, Reverse Domains, Routing policies
- **Records administrative information**
 - Contact information (persons/roles)
 - Authorisation

Whois Database Query - Clients

- **Standard whois client**
 - Included with many Unix distributions
 - RIPE extended whois client
 - <http://ftp.apnic.net/apnic/dbase/tools/ripe-dbase-client.tar.gz>
- **Query via the APNIC website**
 - <http://www.apnic.net/apnic-bin/whois2.pl>
- **Query clients - MS-Windows etc**

Object Types

OBJECT

person

role

inetnum

inet6num

aut-num

domain

route

mntner

PURPOSE

contact persons

contact groups/roles

IPv4 addresses

IPv6 addresses

Autonomous System number

reverse domains

prefixes being announced

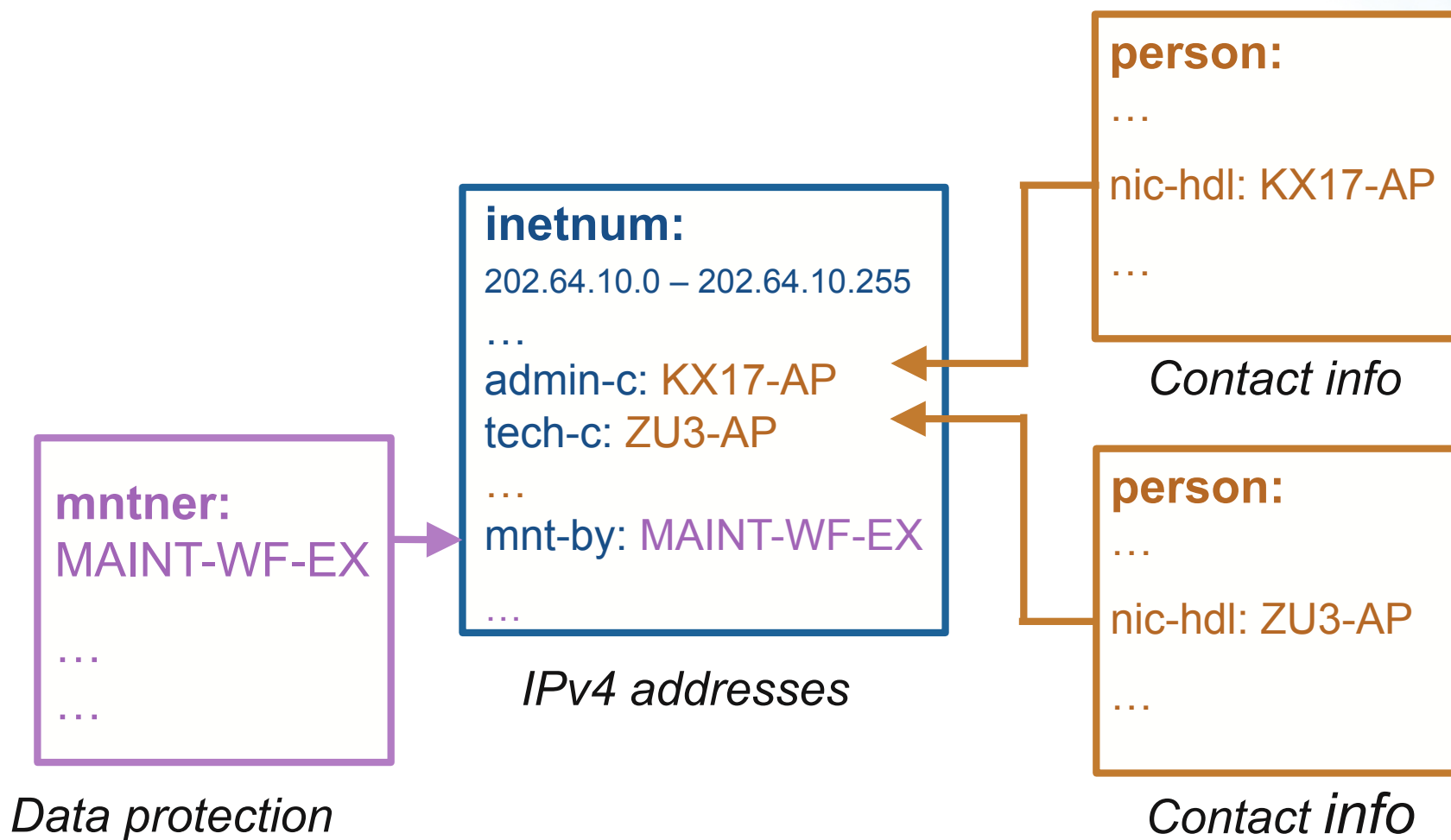
(maintainer) data protection

<http://www.apnic.net/db/>

Database Object

- An object is a set of attributes and values
- Each attribute of an object...
 - Has a value
 - Has a specific syntax
 - Is mandatory or optional
 - Is single- or multi-valued
- Some attributes ...
 - Are primary (unique) keys
 - Are lookup keys for queries
 - Are inverse keys for queries
- Object “templates” illustrate this structure

Inter-related Objects



Database Query – Look-up Keys

| OBJECT TYPE | ATTRIBUTES – LOOK-UP KEYS |
|-------------|---------------------------|
| person | name, nic-hdl, e-mail |
| role | name, nic-hdl, e-mail |
| mntner | maintainer name |
| inetnum | network number, name |
| domain | domain name |
| aut-num | as number |
| as-macro | as-macro name |
| route | route value |
| inet6num | network number, name |

- * Whois supports queries on any of these objects/keys

Object Templates

To obtain template structure*, use :

whois -t <object type>

```
% whois -h whois.apnic.net -t person
```

| | | | |
|----------|-------------|------------|-----------------------|
| person: | [mandatory] | [single] | [primary/look-up key] |
| address: | [mandatory] | [multiple] | [] |
| country: | [mandatory] | [single] | [] |
| phone: | [mandatory] | [multiple] | [] |
| fax-no: | [optional] | [multiple] | [] |
| e-mail: | [mandatory] | [multiple] | [look-up key] |
| nic-hdl: | [mandatory] | [single] | [primary/look-up key] |
| remarks: | [optional] | [multiple] | [] |
| notify: | [optional] | [multiple] | [inverse key] |
| mnt-by: | [mandatory] | [multiple] | [inverse key] |
| changed: | [mandatory] | [multiple] | [] |
| source: | [mandatory] | [single] | [] |

*Recognised by the RIPE whois client/server

Person Object Example

- Person objects contain contact information

Attributes

Values

| | |
|----------|------------------------------|
| person: | Ky Xander |
| address: | ExampleNet Service Provider |
| address: | 2 Pandora St Boxville |
| address: | Wallis and Futuna Islands |
| country: | WF |
| phone: | +680-368-0844 |
| fax-no: | +680-367-1797 |
| e-mail: | kxander@example.com |
| nic-hdl: | KX17-AP |
| mnt-by: | MAINT-WF-EX |
| changed: | kxander@example.com 20100731 |
| source: | APNIC |

What is a nic-hdl?

- Unique identifier for a person
- Represents a person object
 - Referenced in objects for contact details
 - (inetnum, aut-num, domain...)
 - format: <XXXX-AP>
 - Eg: KX17-AP



```
person:   Ky Xander
address:  ExampleNet Service Provider
address:  2 Pandora St Boxville
address:  Wallis and Futuna Islands
country:  WF
phone:    +680-368-0844
fax-no:   +680-367-1797
e-mail:   kxander@example.com

nic-hdl:  KX17-AP
mnt-by:   MAINT-WF-EX
changed:  kxander@example.com 20020731
source:   APNIC
```


Inetnum Object Example

- Contain IP address allocations / assignments

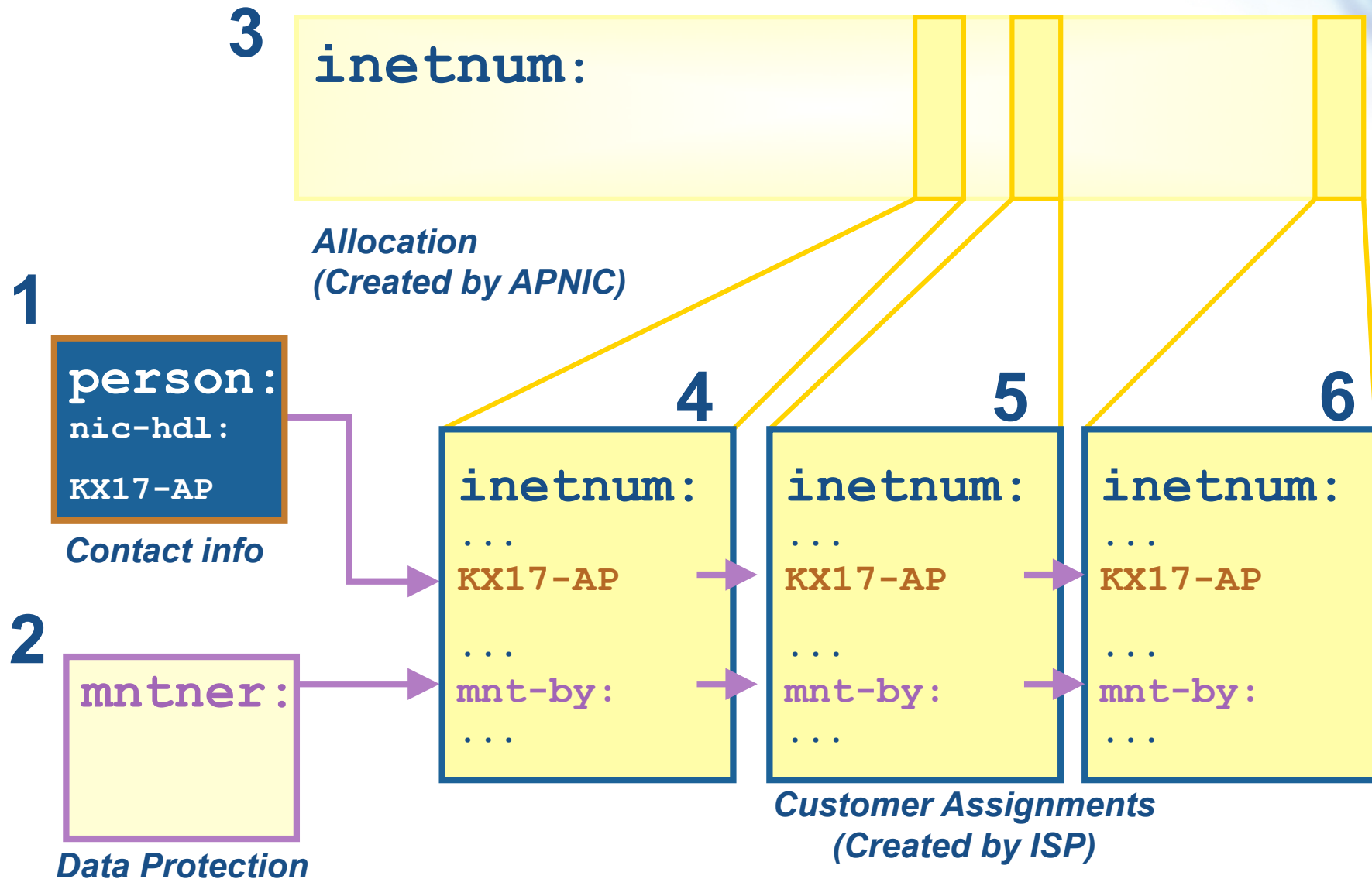
| Attributes | Values |
|------------|---------------------------------------|
| inetnum: | 202.51.64.0 - 202.51.95.255 |
| netname: | CCNEP-NP-AP |
| descr: | Communication & Communicate Nepal Ltd |
| descr: | VSAT Service Provider, Kathmandu |
| country: | NP |
| admin-c: | AS75-AP |
| tech-c: | AS75-AP |
| mnt-by: | APNIC-HM |
| mnt-lower: | MAINT-NP-ARUN |
| changed: | hostmaster@apnic.net 20010205 |
| status: | ALLOCATED PORTABLE |
| source: | APNIC |

ISP Registration Responsibilities

- 1. Create person objects for contacts**
 - To provide contact info in other objects
- 2. Create mntner object**
 - To provide protection of objects
- 3. Create inetnum objects for all customer address assignments as private data**
 - But you may change these to be public data if you wish
 - Allocation object created by APNIC
- 4. Protect all the Objects**



Using the db – Step by Step



Database Protection

- Maintainer Object

mntner: MAINT-WF-EX
descr: Maintainer for ExampleNet Service Provider
country: WF
admin-c: ZU3-AP
tech-c: KX17-AP
upd-to: kxander@example.com
mnt-nfy: kxander@example.com
auth: CRYPT-PW apHJ9zF3o
mnt-by: MAINT-WF-EX
referral-by: MAINT-APNIC-AP
changed: kxander@example.com 20020731
source: APNIC



- protects other objects in the APNIC database

Database Protection

- **Authorisation**
 - “mnt-by” references a mntner object
 - Can be found in all database objects
 - “mnt-by” should be used with every object!
- **Authentication**
 - Updates to an object must pass the authentication rule specified by its maintainer object



Authorisation Mechanism

inetnum: 202.137.181.0 – 202.137.185.255
netname: EXAMPLENET-WF
descr: ExampleNet Service Provider
.....
mnt-by: MAINT-WF-EX

mntner: MAINT-WF-EX
descr: Maintainer for ExampleNet Service Provider
country: WF
admin-c: ZU3-AP
tech-c: KX17-AP
upd-to: kxander@example.com
mnt-nfy: kxander@example.com
auth: CRYPT-PW apHJ9zF3o
mnt-by: MAINT-WF-EX
changed: kxander@example.com 20020731
source: APNIC

Authentication Methods

- **‘auth’ attribute**
 - Crypt-PW
 - Crypt (Unix) password encryption
 - Use web page to create your maintainer
 - PGP – GNUPG
 - Strong authentication
 - Requires PGP keys
 - MD5
 - Available



Mnt-by & Mnt-lower

- **‘mnt-by’ attribute**
 - Can be used to protect any object
 - Changes to protected object must satisfy authentication rules of ‘mntner’ object.
- **‘mnt-lower’ attribute**
 - Also references mntner object
 - Hierarchical authorisation for inetnum & domain objects
 - The creation of child objects must satisfy this mntner
 - Protects against unauthorised updates to an allocated range - highly recommended!



Authentication / Authorisation

- **APNIC allocation to member**
 - Created and maintained by APNIC


| | |
|----------------|----------------------------------|
| Inetnum: | 203.146.96.0 - 203.146.127.255 |
| netname: | LOXINFO-TH |
| descr: | Loxley Information Company Ltd. |
| Descr: | 304 Suapah Rd, Promprab, Bangkok |
| country: | TH |
| admin-c: | KS32-AP |
| tech-c: | CT2-AP |
| 1 → mnt-by: | APNIC-HM |
| 2 → mnt-lower: | LOXINFO-IS |
| changed: | hostmaster@apnic.net 19990714 |
| source: | APNIC |

1. Only APNIC can change this object
2. Only LOXINFO-TH can create assignments within this allocation



Authentication / Authorisation

- **Member assignment to customer**
 - Created and maintained by APNIC member



```
Inetnum:      203.146.113.64 - 203.146.113.127
netname:      SCC-TH
descr:        Sukhothai Commercial College
Country:      TH
admin-c:      SI10-AP
tech-c:       VP5-AP
mnt-by:       LOXINFO-IS
changed:      voraluck@loxinfo.co.th 19990930
source:       APNIC
```

Only LOXINFO-IS can change this object

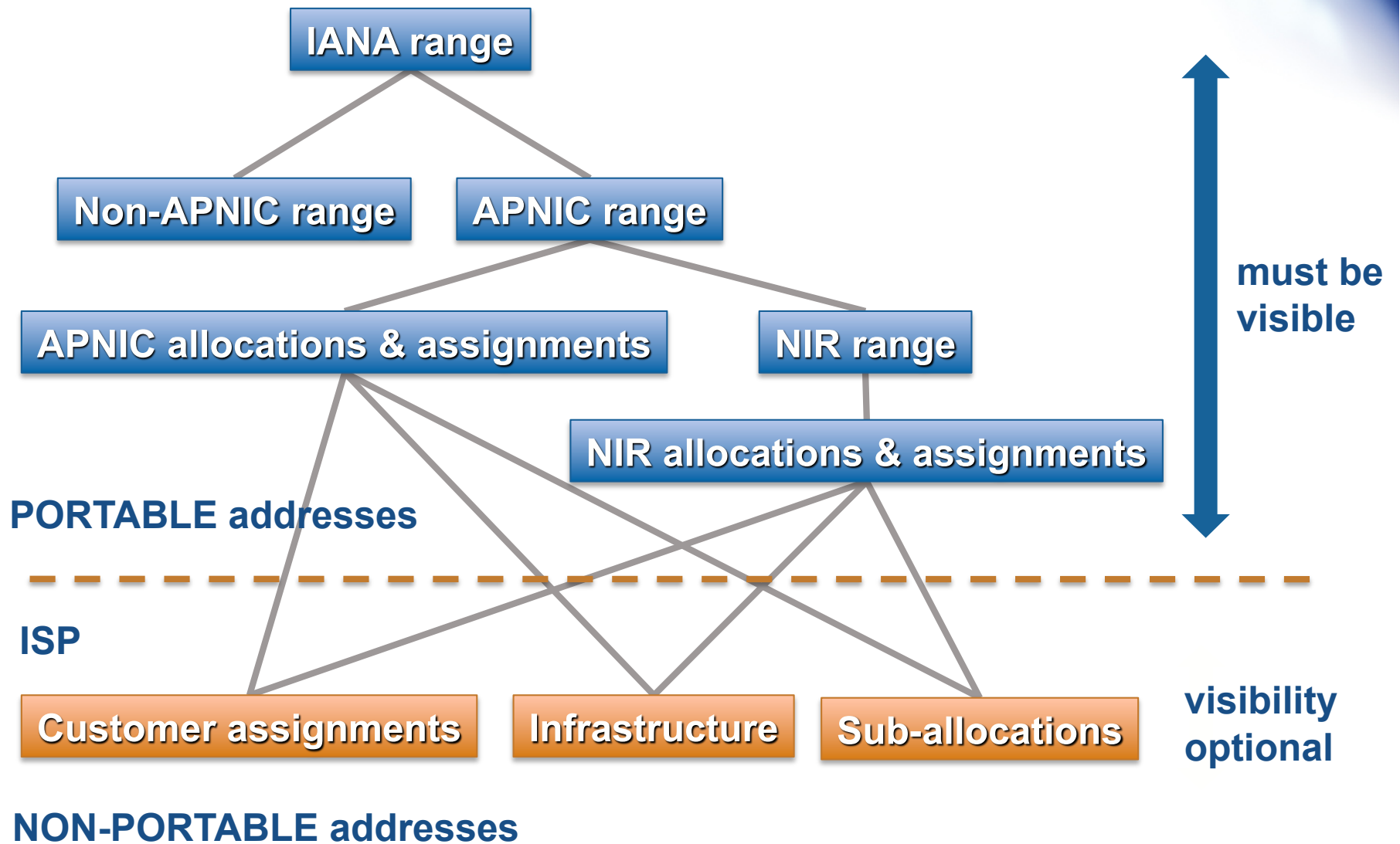
Privacy of Customer Assignments

Customer Privacy

- **Privacy issues**
 - Concerns about publication of customer information
 - Increasing government concern
- **APNIC legal risk**
 - Legal responsibility for accuracy and advice
 - Damages incurred by maintaining inaccurate personal data
- **Customer data is hard to maintain**
 - APNIC has no direct control over accuracy of data
- **Customer assignment registration is still mandatory**



What Needs to be Visible?





APNIC

Asia Pacific Network Information Centre

Introduction to Security



Why Security?

- The Internet was initially designed for connectivity
- Security threats are real...
 - And need protection against
- Fundamental aspects of information must be protected
- We can't keep ourselves isolated from the INTERNET

Why Security?

Most Significant Operational Threats

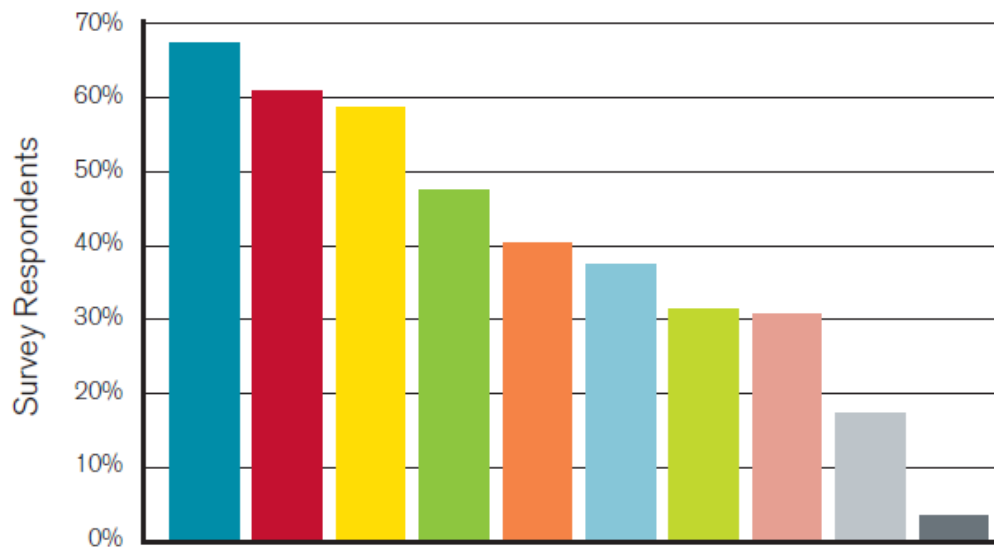
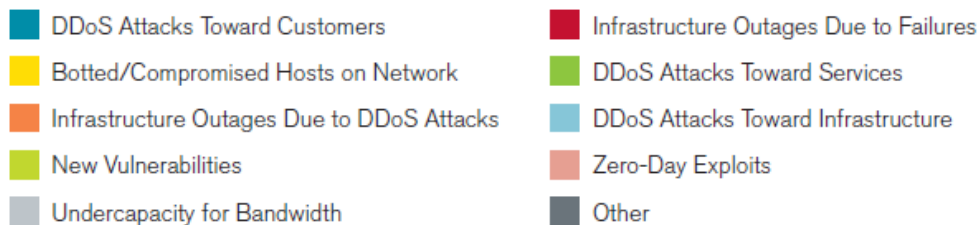
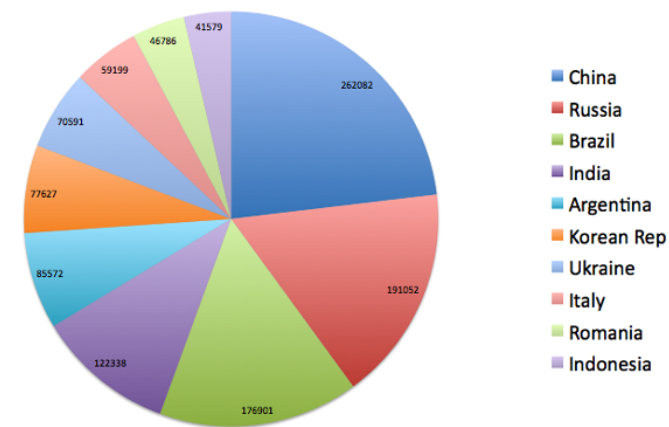


Figure 7
Source: Arbor Networks, Inc.




Source: <http://www.arbornetworks.com/report>

Most infrastructure attacks are unreported

Example Threats

Sony provides PSN update, confirms a 'compromise of personal information' (updated)

By Tim Stevens  posted Apr 26th 2011 4:15PM

BREAKING




PLAYSTATION®Network

It's looking like things are just as bad as we feared and that "external intrusion" got a little deeper than we might have liked. In an update on its *PlayStation.Blog*, Sony just confirmed that the ongoing PSN outage was caused by "malicious actions," which we already knew, but continues by indicating that there has also been "a compromise of personal information." Exactly what that means Sony isn't saying, and it stops short of saying that credit card data for PSN and Qriocity users has been exposed, but the company *does* say "your credit card number (excluding security code) and expiration date may have been obtained." Yes, it *may* have been obtained -- even Sony isn't sure. There's no further ETA for when PSN may be back up online or when you might be able to finally sample *Portal 2*'s delicious online co-op mode, but at least *you* can still watch Netflix.

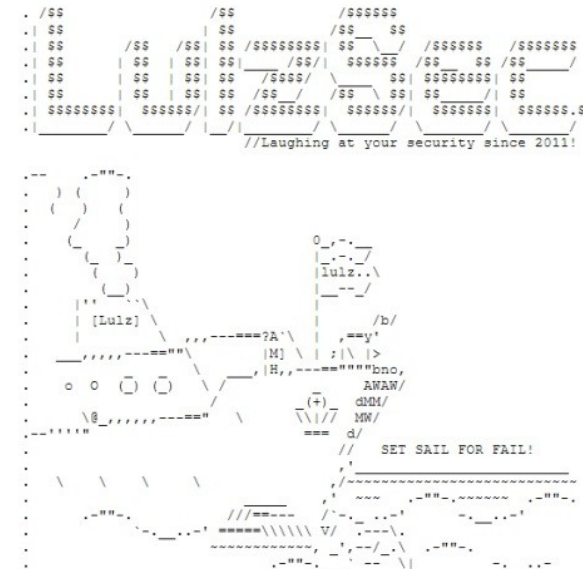
Update: Our friends at *Joystiq* are reporting that Connecticut Senator Blumenthal is [rip roarin' mad](#) about the situation, "demanding answers" from SCEA president Jack Tretton. Right now, we're more curious what Kevin Butler has to say about things.

<http://www.engadget.com/2011/04/26/sony-provides-psn-update-confirms-a-compromise-of-personal-inf/>

Sony Pictures hacked by Lulz Security, 1,000,000 passwords claimed stolen (update)

By Zachary Lutz  posted Jun 2nd 2011 5:47PM

BREAKING



Oh, Sony -- *not again*. We've just received numerous tips that Lulz Security has broken into SonyPictures.com, where it claims to have stolen the personal information of over 1,000,000 users -- all stored (disgracefully) in plain text format. Lulz claims the heist was performed with a simple SQL injection -- just like we saw the [last time around](#). A portion of the group's exploit is posted online in a RAR file, which contains over 50,000 email / password combos of unfortunate users. We've downloaded this file (at our own risk, mind you) and can verify these sensitive bits are now in the wild, though it remains

<http://www.engadget.com/2011/06/02/sony-pictures-hacked-by-lulz-security-1-000-000-passwords-claim/>

Types of Security

- **Computer Security** - generic name for the collection of tools designed to protect data and to thwart hackers
- **Network Security** - measures to protect data during their transmission
- **Internet Security** - measures to protect data during their transmission over a collection of interconnected networks

Goals of Information Security

Confidentiality

**prevents
unauthorized
use or
disclosure of
information**

Integrity

**safeguards the
accuracy and
completeness
of information**

Availability

**authorized
users have
reliable and
timely access
to information**

SECURITY



APNIC

Asia Pacific Network Information Centre

Network Security Concepts





Terminology



Access Control

- Ability to permit or deny the use of an object by a subject.
- It provides 3 essential services:
 - Identification and authentication (who can login)
 - Authorization (what authorized users can do)
 - Accountability (identifies what a user did)



Vulnerability

- A weakness in security procedures, network design, or implementation that can be exploited to violate a corporate security policy
 - Software bugs
 - Configuration mistakes
 - Network design flaw
- Exploit
 - Taking advantage of a vulnerability

Risk

- The possibility that a particular vulnerability will be exploited
 - Risk analysis: the process of identifying:
 - security risks
 - determining their impact
 - and identifying areas require protection

Threat

- Any circumstance or event with the potential to cause harm to a networked system
 - Denial of service
 - Attacks make computer resources (e.g., bandwidth, disk space, or CPU time) unavailable to its intended users
 - Unauthorised access
 - Access without of permission issued by a rightful owner of devices or networks
 - Impersonation
 - Identity theft
 - Worms
 - Viruses

Risk management vs. cost of security

- Risk mitigation
 - The process of selecting appropriate controls to reduce risk to an acceptable level
- The level of acceptable risk
 - Determined by comparing the risk of security hole exposure to the cost of implementing and enforcing the security policy
- Assess the cost of certain losses and do not spend more to protect something than it is actually worth

Attack sources

- Active vs. passive
 - Active = Writing data to the network
 - Common to disguise one's address and conceal the identity of the traffic sender
 - Passive = Reading data on the network
 - Purpose = breach of confidentiality
 - Attackers gain control of a host in the communication path between two victim machines
 - Attackers has compromised the routing infrastructure to arrange the traffic pass through a compromised machine

What are security goals?

- Controlling data / network access
- Preventing intrusions
- Responding to incidences
- Ensuring network availability
- Protecting information in transit

Security services

- Authentication
- Authorisation
- Access control
- Data integrity
- Data confidentiality
- Auditing / logging
- DoS mitigation

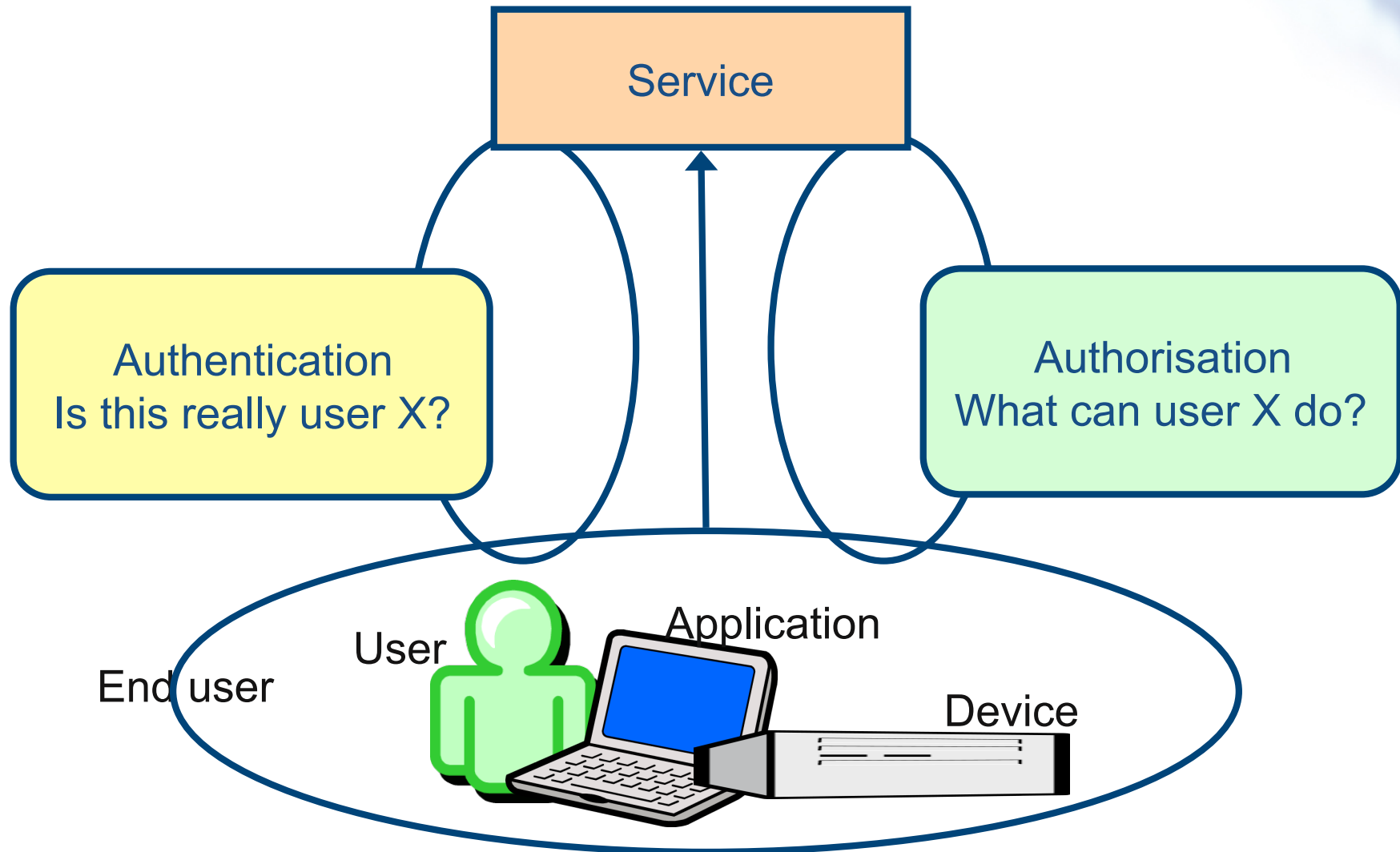
Authentication

- The process of validating the claimed identity of an end user or a device such as a host, server, switch, router, etc.
- Must be careful whether a technology is using:
 - User authentication
 - Device authentication
 - Application authentication

Authorisation

- The act of granting access rights to a user, groups of users, system, or program
 - Typically this is done in conjunction with authentication

Authentication and Authorisation



Non-repudiation

- A property of a cryptographic system that prevents a sender from denying later that he or she sent a message or performed a certain action
- Assure that both parties are involved in the transaction
 - This prevents one party from denying involvement at a later date

Integrity

- Assurance that the data has not been altered except by the people who are explicitly intended to modify it

Confidentiality

- Assurance that data is not read or accessed by unauthorised persons

Availability

- A state in computing systems and networks in which the system is operable and can run services it is supposed to offer

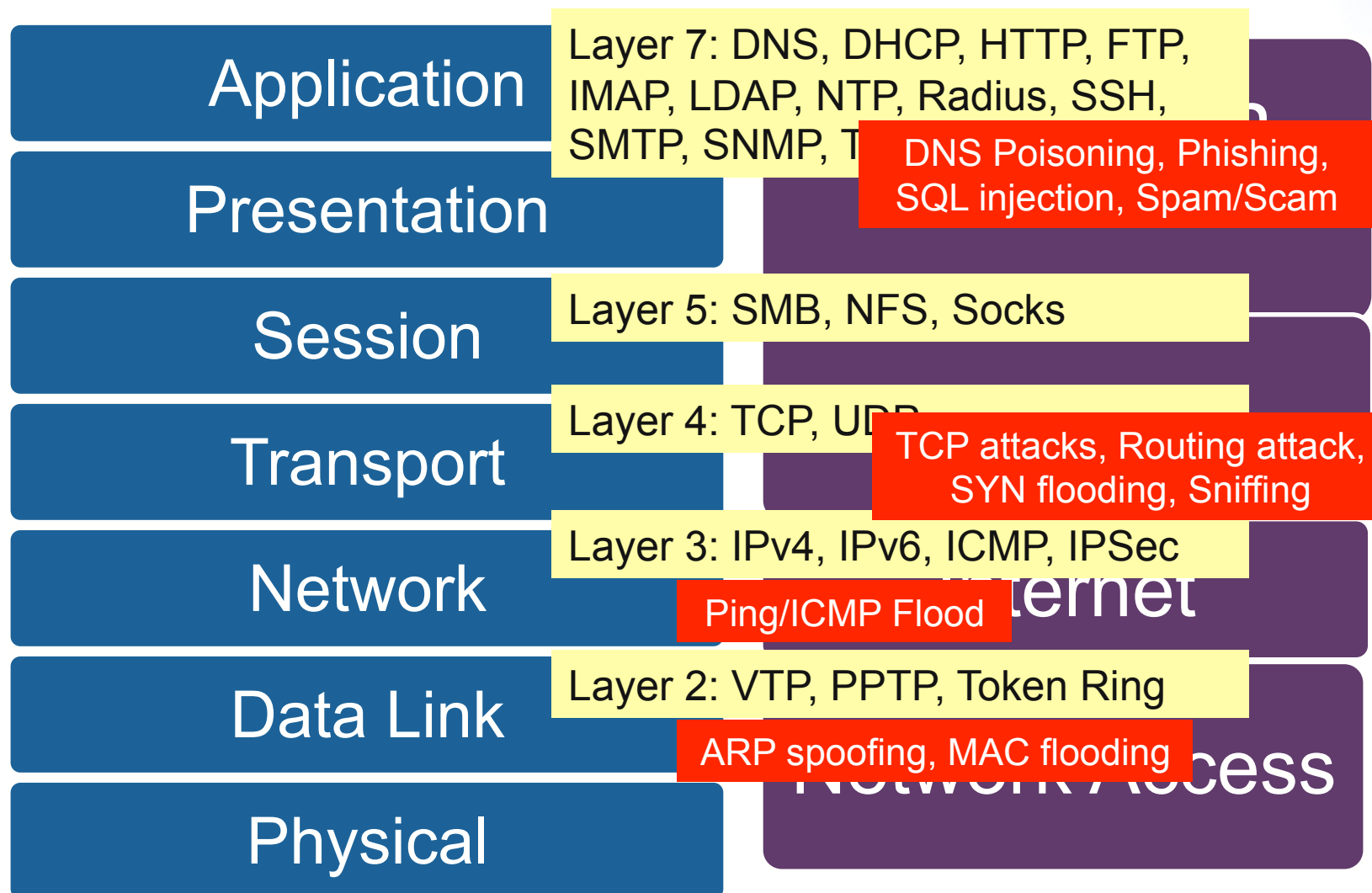
Audit

- A chronological record of system activities that is sufficient to enable the reconstruction and examination of a given sequence of events



Threats and Attacks

Attacks on Diff Layers



Layer 2 Attacks

- VLAN hopping
- MAC attacks
- DHCP attacks
- ARP Attacks

VLAN Hopping

- Attack on a network with multiple VLANs
- Two primary methods:
 - Switch spoofing – attacker initiates a trunking switch
 - Double tagging – packet is tagged twice.
- Solution: configure the switch's edge ports to accept only untagged packets

```
switchport nonnegotiable
```

```
switchport mode access
```

MAC Flooding

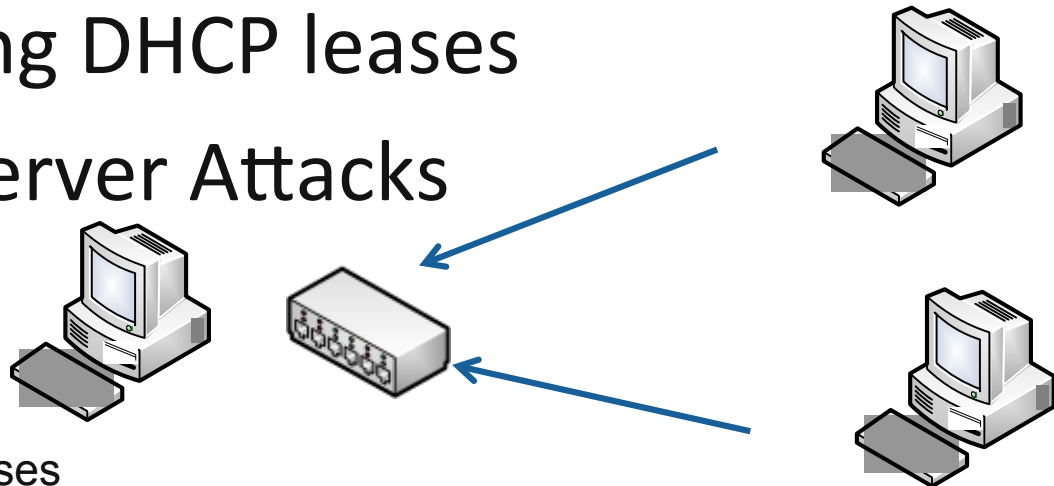
- Attack that exploits the CAM Table
- CAM = Content Addressable memory = stores info on the mapping of individual MAC addresses to physical ports on the switch.
- Exploits the limitation of all switches – fixed CAM table size



DHCP Attack Types

- DHCP Starvation Attack - Broadcasting vast number of DHCP requests with spoofed MAC address simultaneously.
- DoS attack using DHCP leases
- Rogue DHCP Server Attacks

Server runs out of IP addresses to allocate to valid users



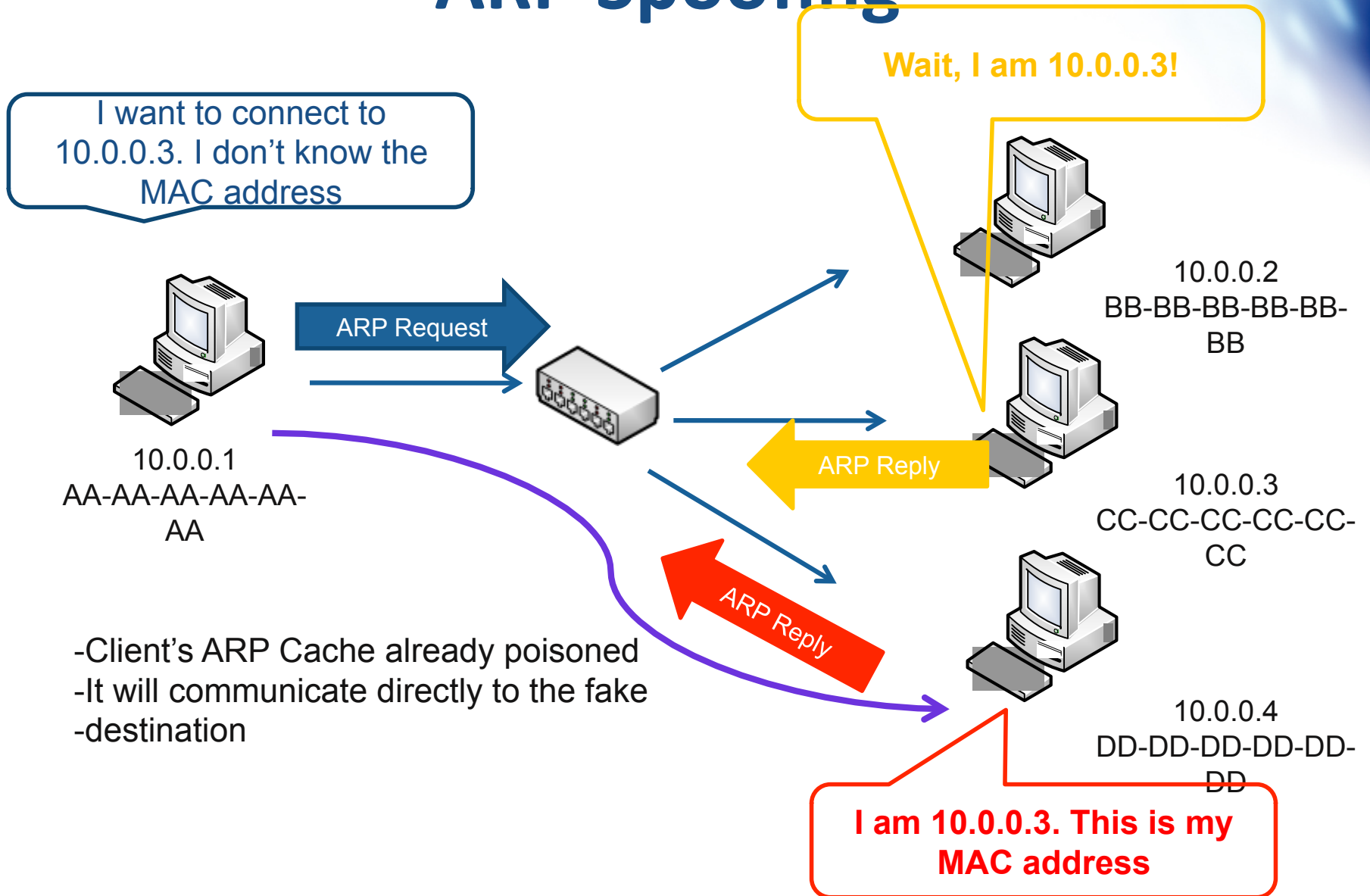
Attacker sends many different DHCP requests with many spoofed addresses.

DHCP Attack Types

- Solution: enable DHCP snooping

```
ip dhcp snooping (enable dhcp snooping globally)
ip dhcp snooping vlan <vlan-id> (for specific
vlans)
ip dhcp snooping trust
ip dhcp snooping limit rate <rate>
```

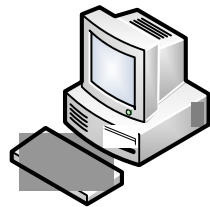
ARP Spoofing



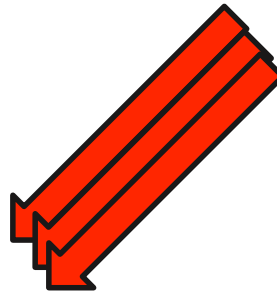
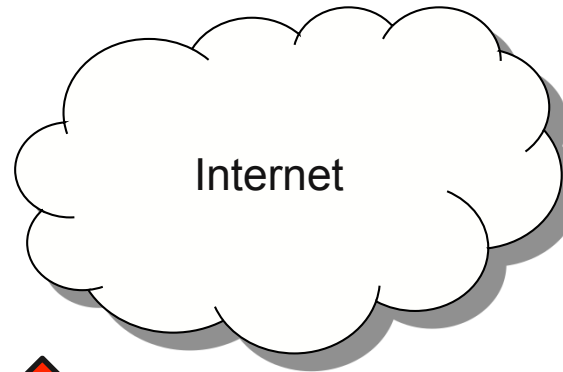
Layer 3 Attacks

- ICMP Ping Flood
- ICMP Smurf
- Ping of death

Ping Flood

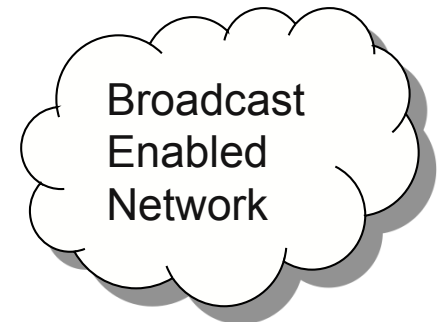
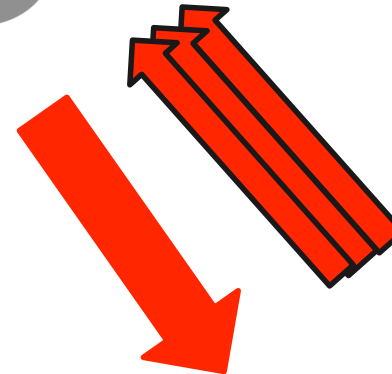


Attacking System



Victim System

Other forms of ICMP attack:
-Ping of death
-ICMP ping flood



TCP Attacks

- **SYN Flood** – occurs when an attacker sends SYN requests in succession to a target.
- Causes a host to retain enough state for bogus half-connections such that there are no resources left to establish new legitimate connections.

TCP Attacks



- Server needs to keep waiting for ACK y+1
- Server recognizes client based on IP address/port and y +1



Routing Attacks

- Attempt to poison the routing information
- Distance Vector Routing
 - Announce 0 distance to all other nodes
 - Blackhole traffic
 - Eavesdrop
- Link State Routing
 - Can drop links randomly
 - Can claim direct link to any other routers
 - A bit harder to attack than DV
- BGP attacks
 - ASes can announce arbitrary prefix
 - ASes can alter path

Application Layer Attacks

- Applications don't authenticate properly
- Authentication information in clear
 - FTP, Telnet, POP
- DNS insecurity
 - DNS poisoning
 - DNS zone transfer

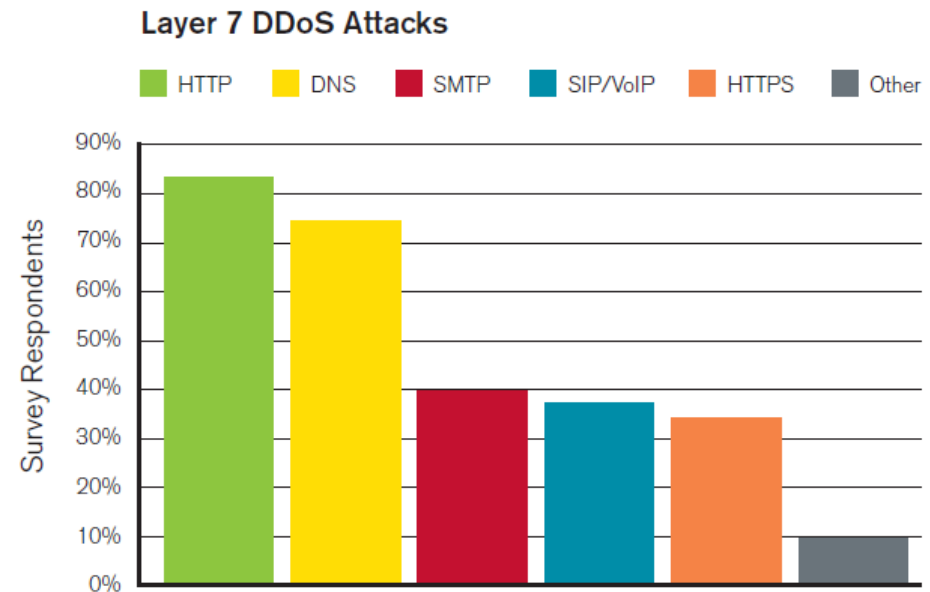


Figure 8

Source: Arbor Networks, Inc.

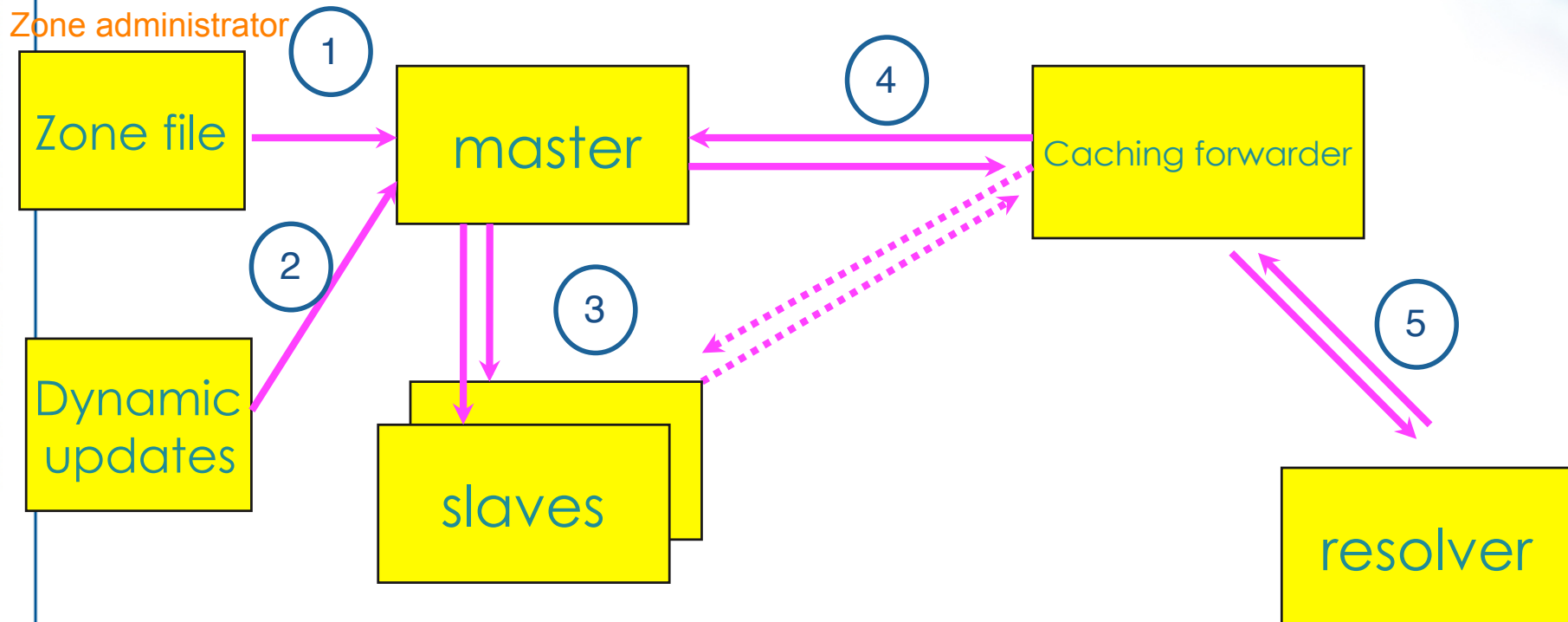
Server Side Scripting

- **Server-side scripting** – program is executed on the server and not on the user's browser or plugin.
- ASP.NET, PHP, mod_perl, CGI, Ruby, Python
- Benefits:
 - Cross-platform
 - No plugin required on user side
- Disadvantages:
 - Dynamic scripts create new security concern, exploiting code flaws

SQL Injection

- **SQL Injection** – a subset of unverified user input vulnerability that injects malicious code (or SQL query) into strings. This code is executed when passed on to the SQL server.

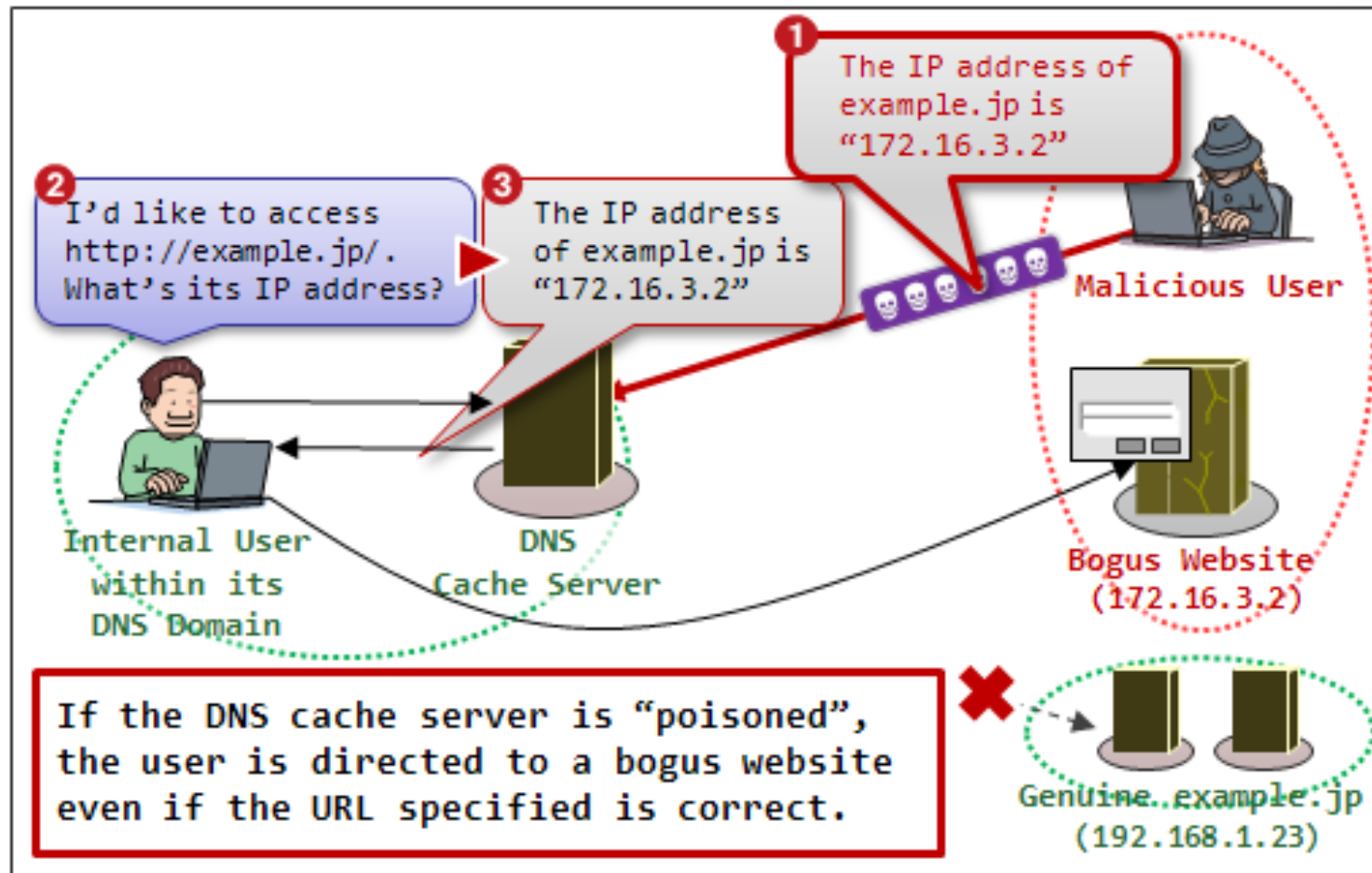
DNS Points of Attack



DNS Cache Poisoning

- Caching incorrect resource record that did not originate from authoritative DNS sources.
- Result: connection (web, email, network) is redirected to another target (controlled by the attacker)

DNS Cache Poisoning



http://www.ipa.go.jp/security/english/vuln/200809_DNS_en.html

Common Types of Attack

- Man-in-the-middle attack – intercepts messages that are intended for a valid device
- Ping sweeps and port scans
- Hijacking and Spoofing -sets up a fake device and trick others to send messages to it
- Sniffing – capture packet as they travel through the network
- DoS and DDoS

Wireless Attacks

- WEP – first security mechanism for 802.11 wireless networks
- Weaknesses in this protocol were discovered by Fluhrer, Mantin and Shamir, whose attacks became known as “FMS attacks”
- Tools were developed to automate WEP cracking
- Chopping attack were released to crack WEP more effectively and faster

Man in the Middle Attacks (Wireless)

- Creates a fake access point and have clients authenticate to it instead of a legitimate one.
- Capture traffic to see usernames, passwords, etc that are sent in clear text.

network
security.

who needs it
when there's tape?

How do we protect
our system?



<http://m-a-t-t-h-e-w.deviantart.com/art/network-security-26703428?q=boost%3Apopular%20network%20security&qo=24>



APNIC

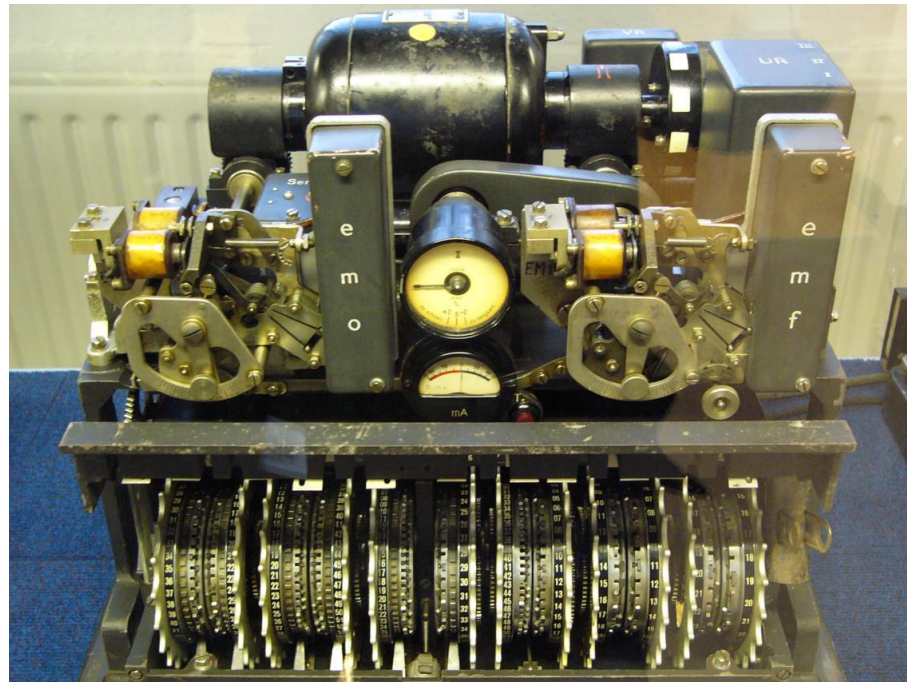
Asia Pacific Network Information Centre

Cryptography



Cryptography

- Has evolved into a complex science in the field of information security



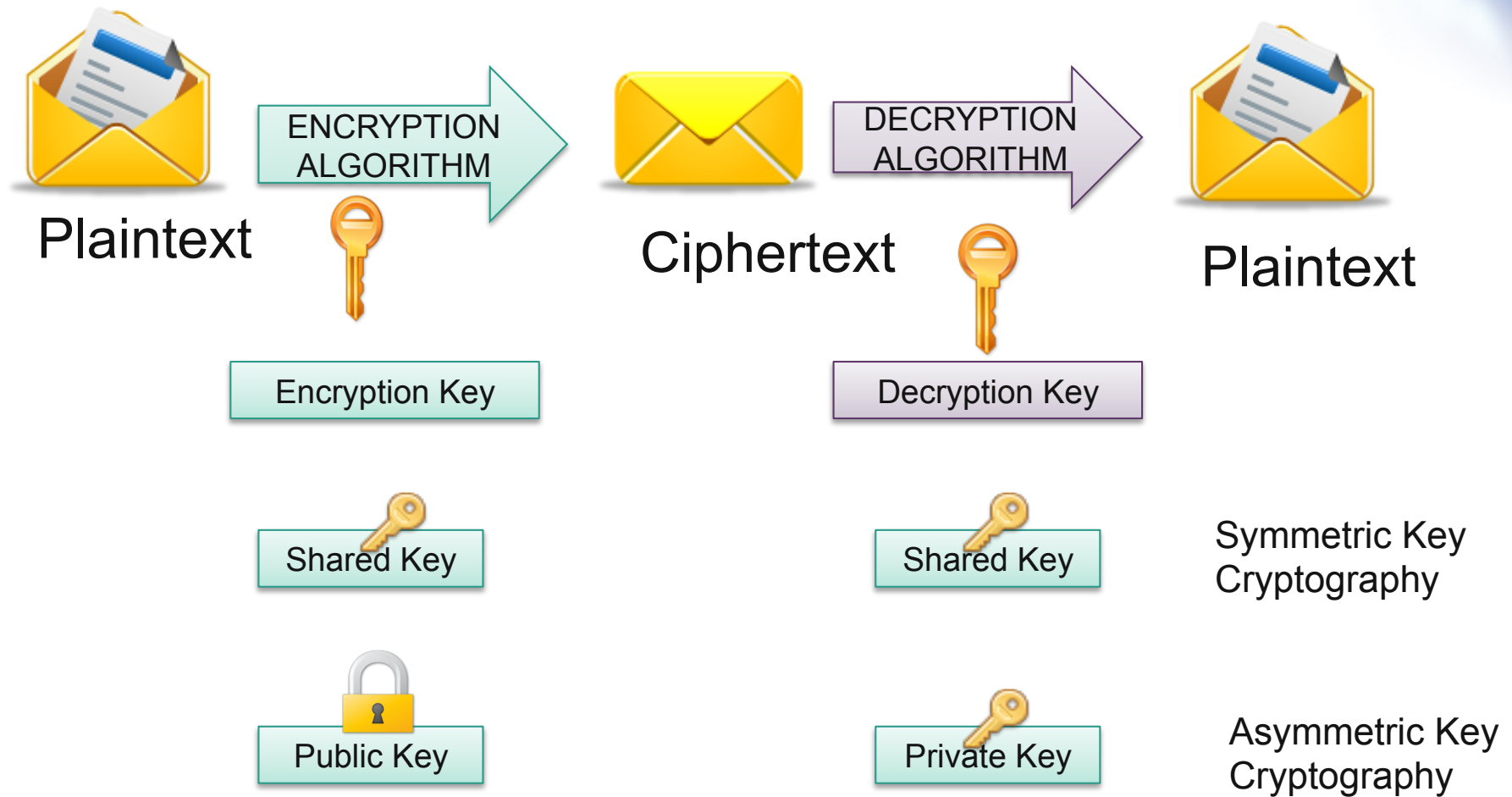
What is cryptography?

- Part of field of study known as **cryptology**
- Cryptology includes:
 - **Cryptography**
 - study of methods for secret writing
 - transforming messages into unintelligible form
 - recovering messages using some secret knowledge (key)
 - **Cryptanalysis:**
 - analysis of cryptographic systems, inputs and outputs
 - to derive confidential information

Cryptography

- **Encryption** – process of transforming plaintext to ciphertext using a cryptographic key
- **Symmetric key cryptography** – uses a single key to both encrypt and decrypt information. Also known as private key.
 - Includes DES, 3DES, AES, IDEA, RC5, Blowfish
- **Asymmetric key cryptography** – separate keys for encryption and decryption (public and private key pairs)
 - Includes RSA, Diffie-Hellman, El Gamal

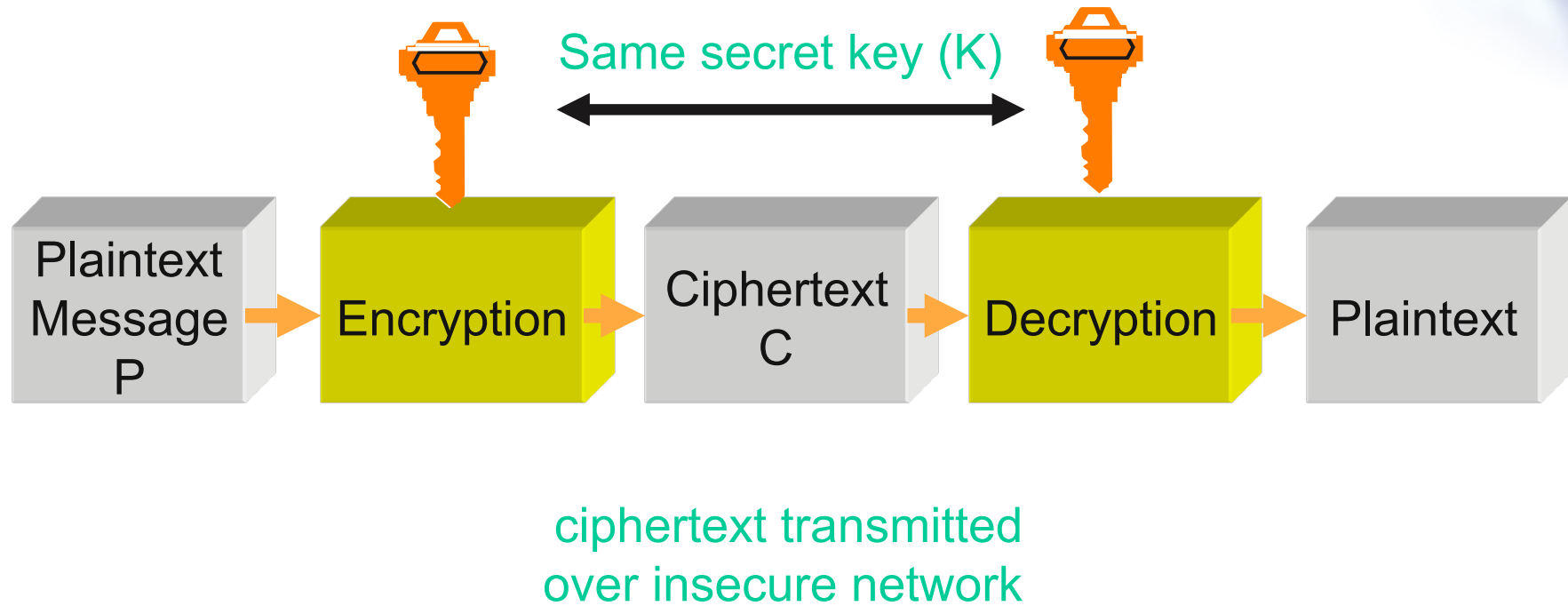
Cryptography



Terminology of cryptography

- Cipher
 - cryptographic technique (algorithm) applying a secret transformation to messages
- Plaintext / cleartext
 - original message or data
- Encryption
 - transforming plaintext, using a secret key, so meaning is concealed
- Ciphertext
 - Unintelligible encrypted plaintext
- Decryption
 - transforming ciphertext back into original plaintext
- Cryptographic key
 - secret knowledge used by cipher to encrypt or decrypt message

Symmetric cipher



Symmetric Key Algorithm

- **Stream ciphers** – encrypts bits of the message at a time
- **Block ciphers** – takes a block of bits and encrypts them as a single unit

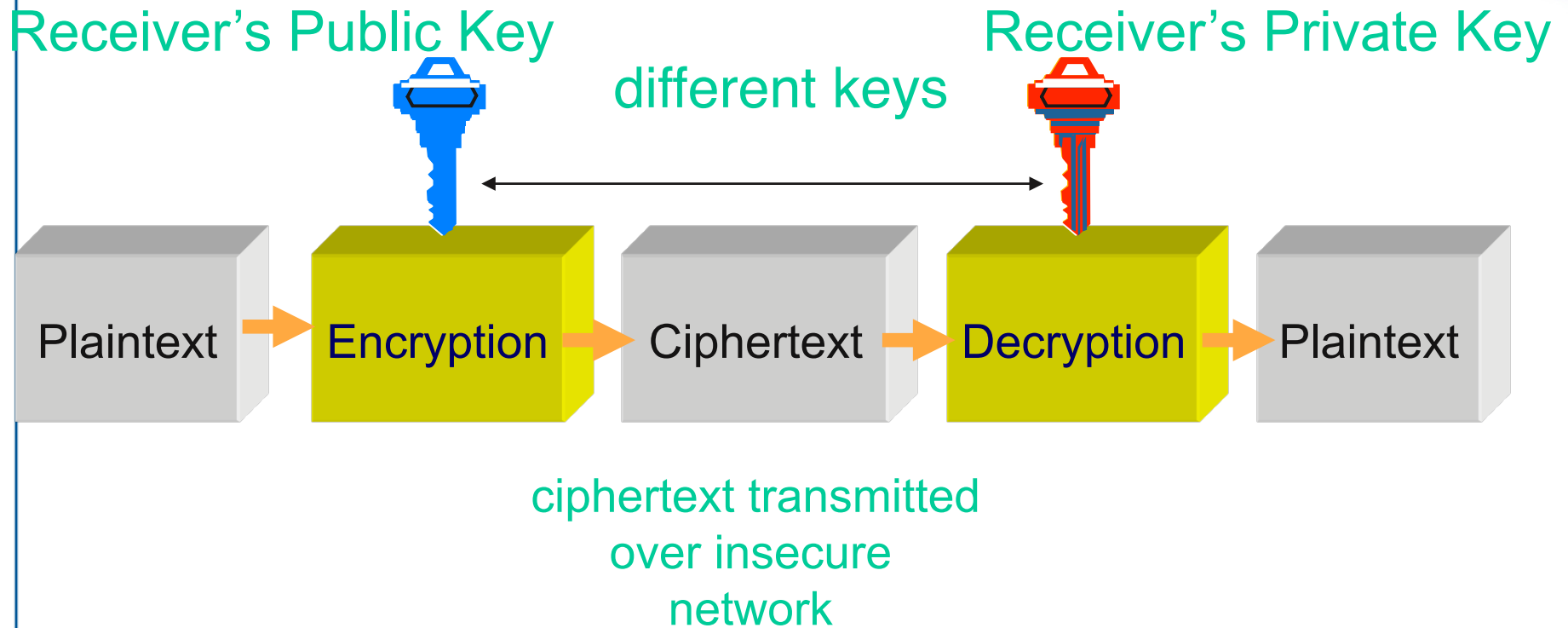
Symmetric ciphers

- Two categories:
 - Stream ciphers:
 - data is encrypted one bit at a time
 - Uses a keystream generator to produce pseudorandom key
 - Fast
 - No current standard
 - Eg RC4
 - Block ciphers:
 - Data is encrypted in blocks
 - EG DES has block size of 64 bits
 - AES (Advanced Encryption Standard)

Asymmetric ciphers

- Two different keys (key pair):
 - A message encrypted with one key is decrypted using the other key
 - two keys are related
 - but it is *computationally infeasible* to derive one key from the other
- Each participant requires a pair of keys
 - encryption key K_{pub} (made public)
 - decryption key K_{priv} (kept private)
- Also known as public key cryptography
- Security depends on
 - algorithm strength
 - key size
 - protection measures of private key K_{priv}

Asymmetric ciphers



Asymmetric ciphers

- Everyone knows the public key
 - no need for secure means of public key distribution
- For **confidentiality**, anyone can encrypt a message for Alice using her **public** key K_{pub}
 - Encryption: $C = E(P, K_{pub})$
 - Only Alice knows her private key
 - so only Alice can decrypt encrypted message
 - Decryption: $P = D(C, K_{priv})$
C=ciphertext, E=encrypt,
P=plaintext, K=key, D=decrypt

Asymmetric ciphers

- Role of public and private keys can be reversed for authentication and non-repudiation:
 - Alice encrypts a message using her private key, K_{priv}
 - Encryption: $C = E(P, K_{\text{priv}})$
 - Everyone knows Alice's corresponding public key, K_{pub}
 - Decryption: $P = D(C, K_{\text{pub}})$
 - Successful decryption means message must have been encrypted using Alice's private key

Example asymmetric cipher

- *RSA algorithm (1977)*
 - Currently most widely used public key cryptosystem
 - Named after designers:
 - Rivest, Shamir, and Adleman
 - Based on difficulty of factoring large integers
 - Encryption and decryption involve exponentiation mod n
 - performed one data block at a time

Asymmetric ciphers

- Advantages:
 - Simple key exchange/distribution
 - public keys are not secret
 - so they don't need to be distributed over a secure channel
 - Any user need only have a single key pair
 - Rather than sharing a different key with every other user
 - Fewer keys needed – more scalable



Asymmetric ciphers

- Disadvantages:
 - Complexity of operations greater than in symmetric ciphers
 - Longer keys required for equivalent security (*previous slide*)
 - Speed
 - Encryption/decryption is computationally intensive
 - so much slower than symmetric ciphers
 - Association between an entity and his public key must be verified
 - Trusted Certification Authority (CA) required
 - Digital certificates

Message digests

- **Message digests** – produces a condensed representation of a message (hashing)
 - MD5
 - SHA-1
 - HMAC

Secret Key Algorithms

- **DES** – block cipher using shared key encryption, 56-bit
- **3DES** (Triple DES) – a block cipher that applies DES three times to each data block
- **RC4** – variable-length key, “stream cipher” (generate stream from key, XOR with data)
- **AES** – replacement for DES; current standard

Cryptography: AES

- Advanced Encryption Standard (AES) Cipher
- Has a fixed block size of 128 bits
- Has a key size of 128, 192, or 256 bits
- Based on Rijndael cipher which was developed by Joan Daemen and Vincent Rijmen

Hashing

- Also called a *message digest* or *checksum*
- A form of signature that represents the data.
- Uses:
 - Verifying file integrity - if the hash changes, it means the data is either compromised or altered in transit.
 - Digitally signing documents – sign the hash with a private key
 - Hashing passwords – store as hash rather than cleartext

Hashing

- **MD5** Message Digest Algorithm
 - Outputs a 128-bit fingerprint of an arbitrary-length input
- **SHA-1** (Secure Hash Algorithm)
 - Outputs a 160-bit message digest similar to MD5
 - Widely-used on security applications (TLS, SSL, PGP, SSH, S/MIME, IPsec)

Diffie-Hellman

- **Diffie-Hellman Protocol** – requires that both the sender and recipient of a message have key pairs.
- Combining one's private key and the other's public key, both parties can compute the same shared secret number.

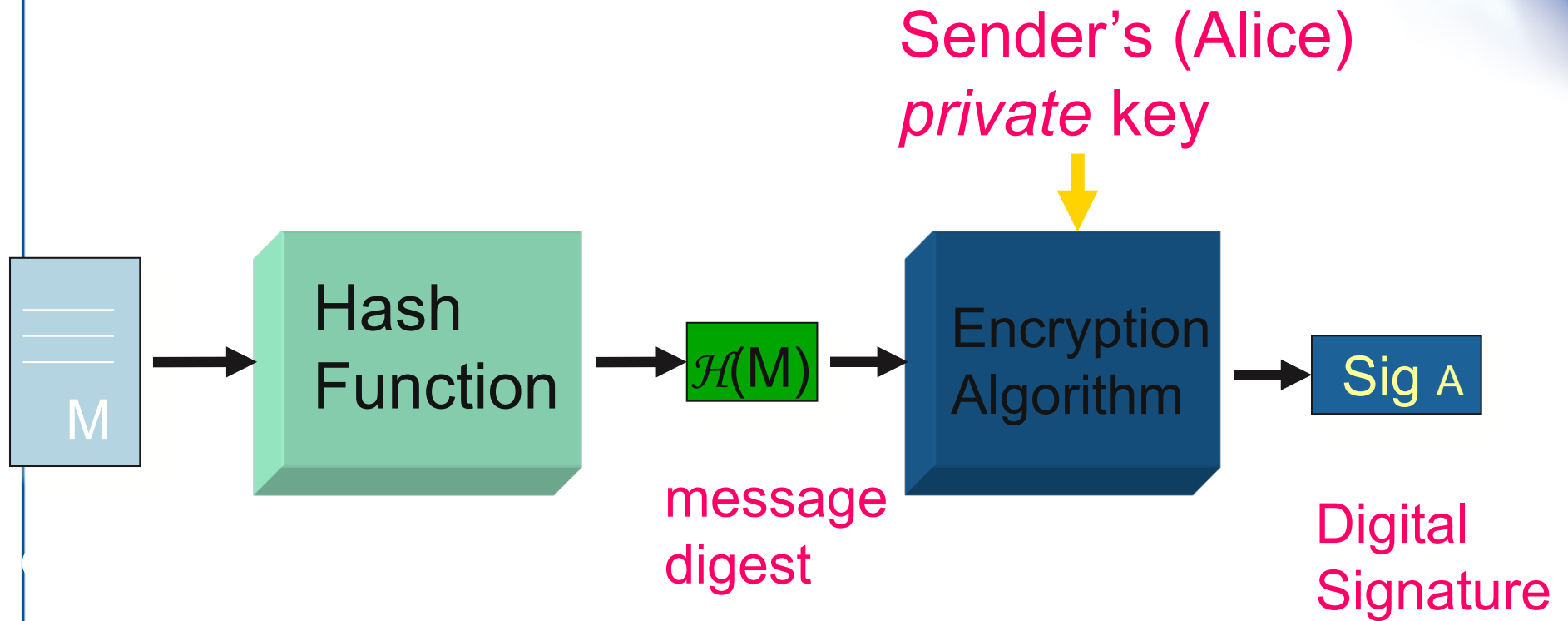
Trusted Network

- Standard defensive-oriented technologies
 - Firewall
 - Intrusion Detection
- Build TRUST on top of the TCP/IP infrastructure
 - Strong authentication
 - Public Key Infrastructure (PKI)

Digital Signature

- Used to provide:
 - Authentication
 - Integrity
 - Non-repudiation
- Uses public-key encryption
- Normal to sign a hash (condensed version) of document rather than signing whole document
 - For efficiency reasons
 - Particularly if messages are long

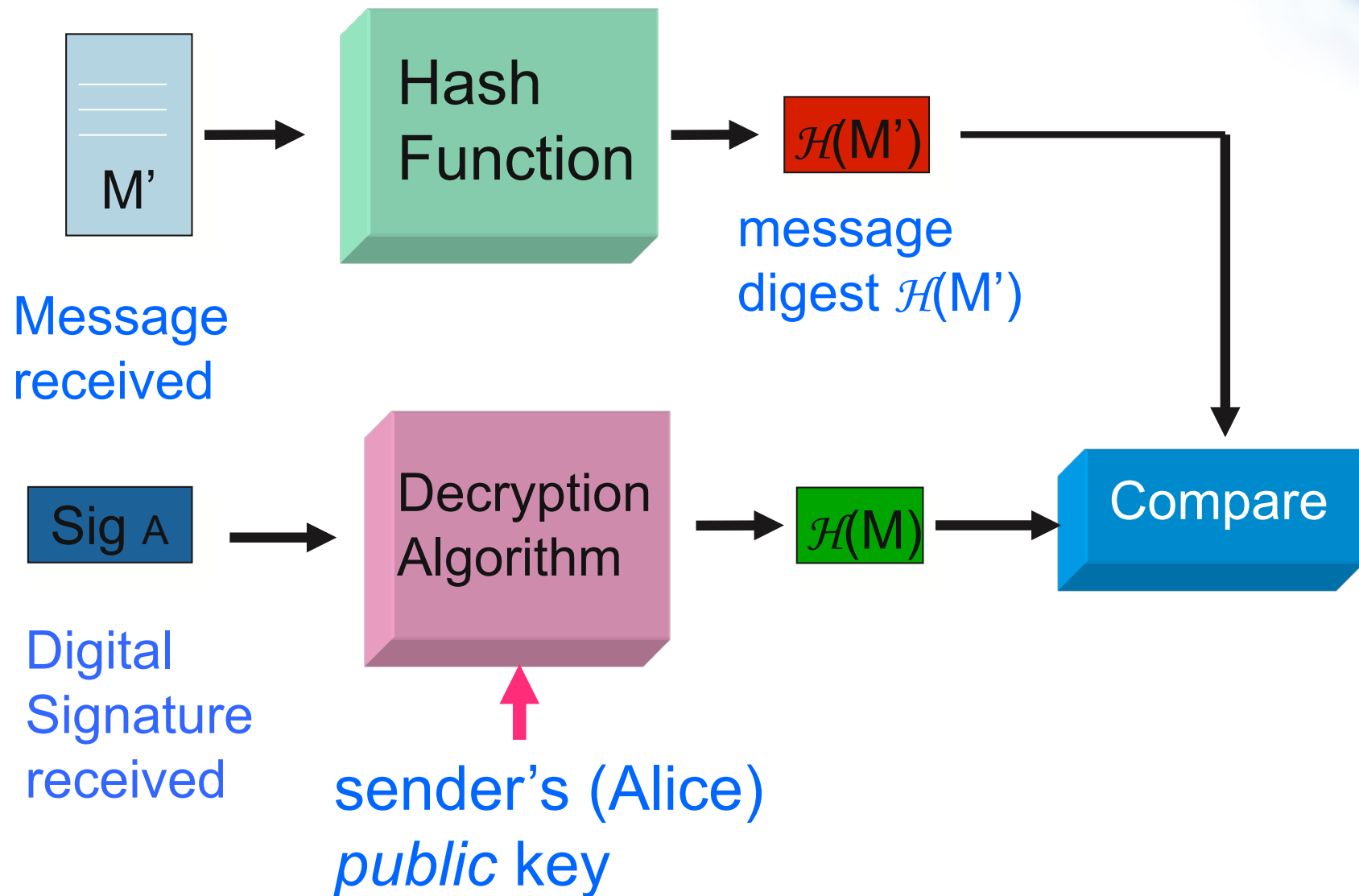
Creating an RSA Digital Signature



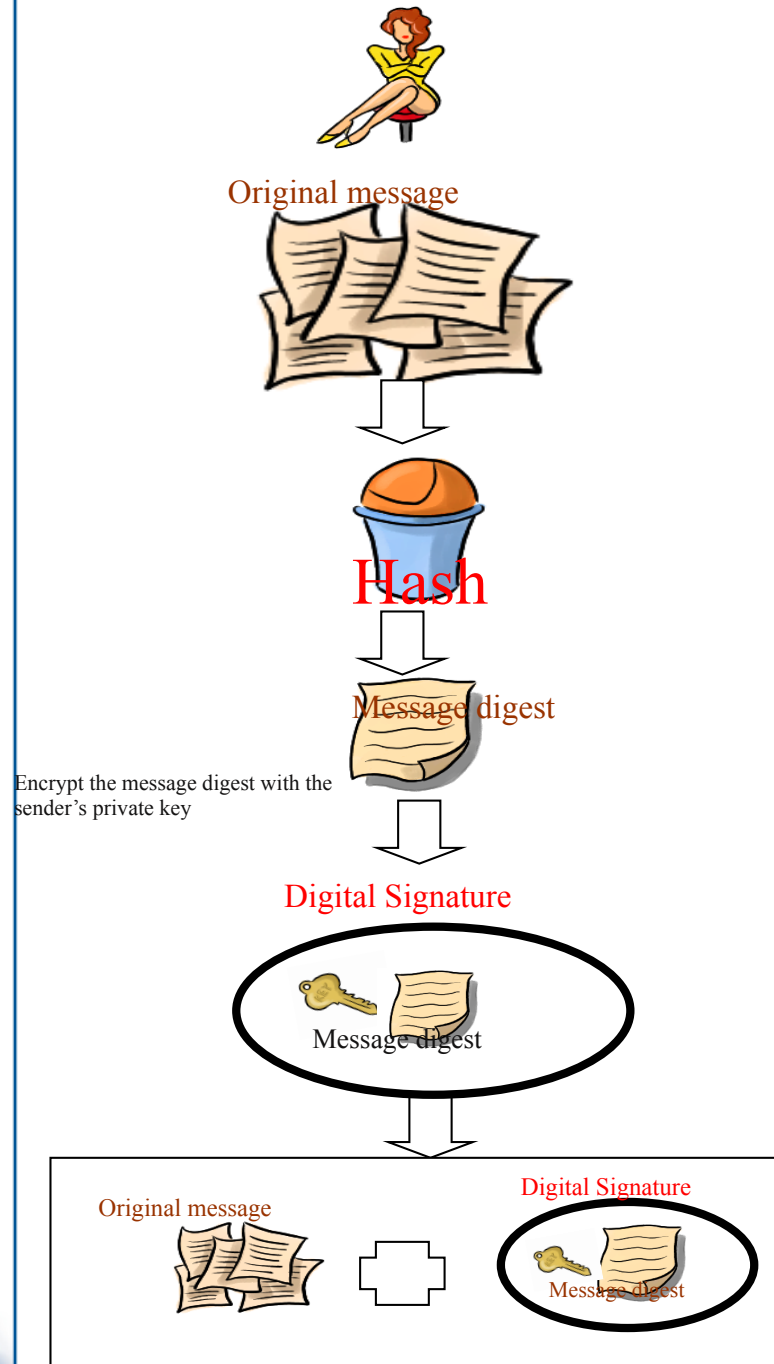
Authenticating message sender

- *Verifying an RSA Digital Signature:*
 - Bob (message receiver):
 - generates $\mathcal{H}(M')$ from M' he received
 - determines $\mathcal{H}(M) = D_{\text{RSA}}(\text{Sig}_A(M), K_{A_{\text{pub}}})$
 - compares $\mathcal{H}(M')$ and $\mathcal{H}(M)$
 - If $\mathcal{H}(M')$ and $\mathcal{H}(M)$
 - then integrity and authenticity of message are guaranteed
 - also sender cannot deny sending the message (non-repudiation)

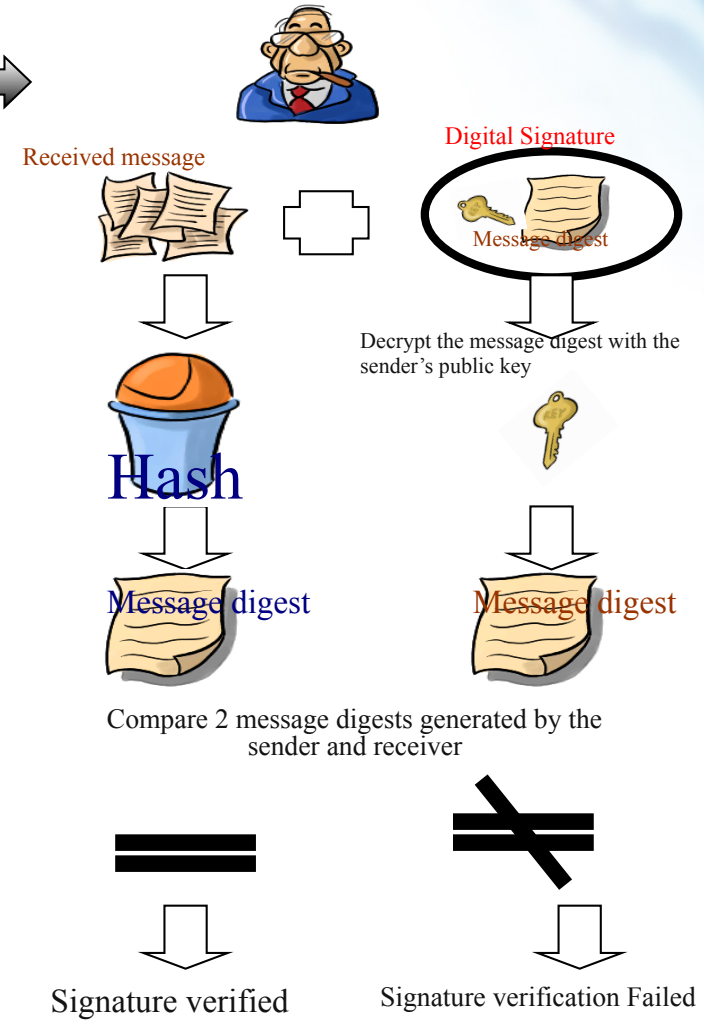
Verifying an RSA Digital Signature



Digital Signing Process



Digital Verification Process





APNIC

Asia Pacific Network Information Centre

Public Key Infrastructure

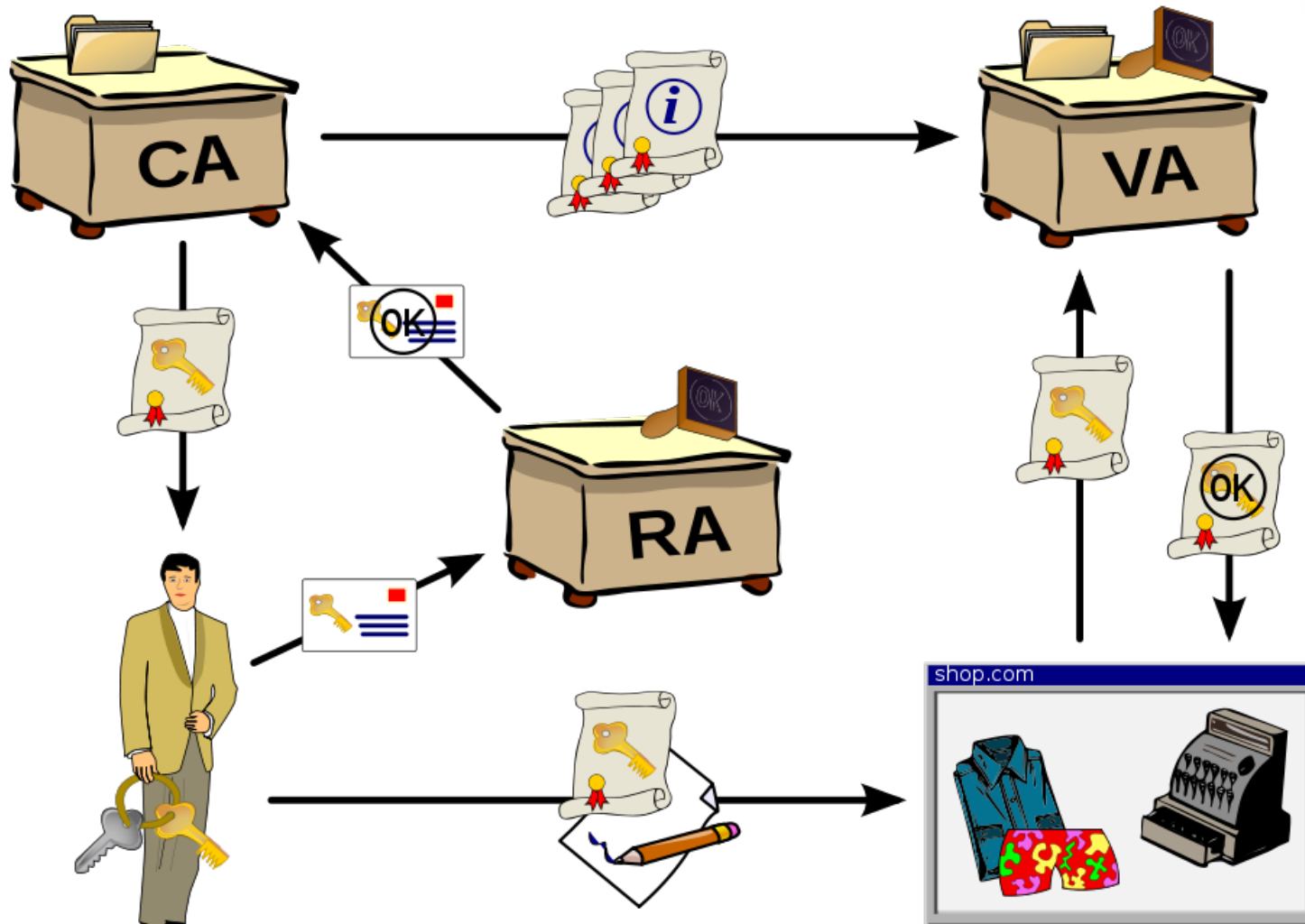
Public Key Infrastructure

- Framework that builds the network of trust
- Combines public key cryptography, digital signatures, to ensure confidentiality, integrity, authentication, nonrepudiation, and access control
- Protects applications that require high level of security

PKI - Components

- Certificate Authority (CA) – a trusted third party
 - Trusted by both the owner of the certificate and the party relying upon the certificate.
- Registration Authority (RA) – binds keys to users
 - Users who wish to have their own certificate registers with the RA
- Validation Authority (VA) –
 - Validates the user is who he says he is

Public Key Infrastructure - Process



Source: <http://commons.wikimedia.org>

Digital Certificate

- **Digital certificate** – basic element of PKI; secure credential that identifies the owner
- Also called public key certificate





Digital certificates

- Digital certificates deal with the problem of
 - binding a public key to an entity
 - A major legal issue related to eCommerce
- A digital certificate contains:
 - user's public key
 - user's ID
 - other information e.g. validity period
- Certificate examples:
 - X509 (standard)
 - PGP (Pretty Good Privacy)
- Certificate Authority (CA) creates and digitally signs certificates

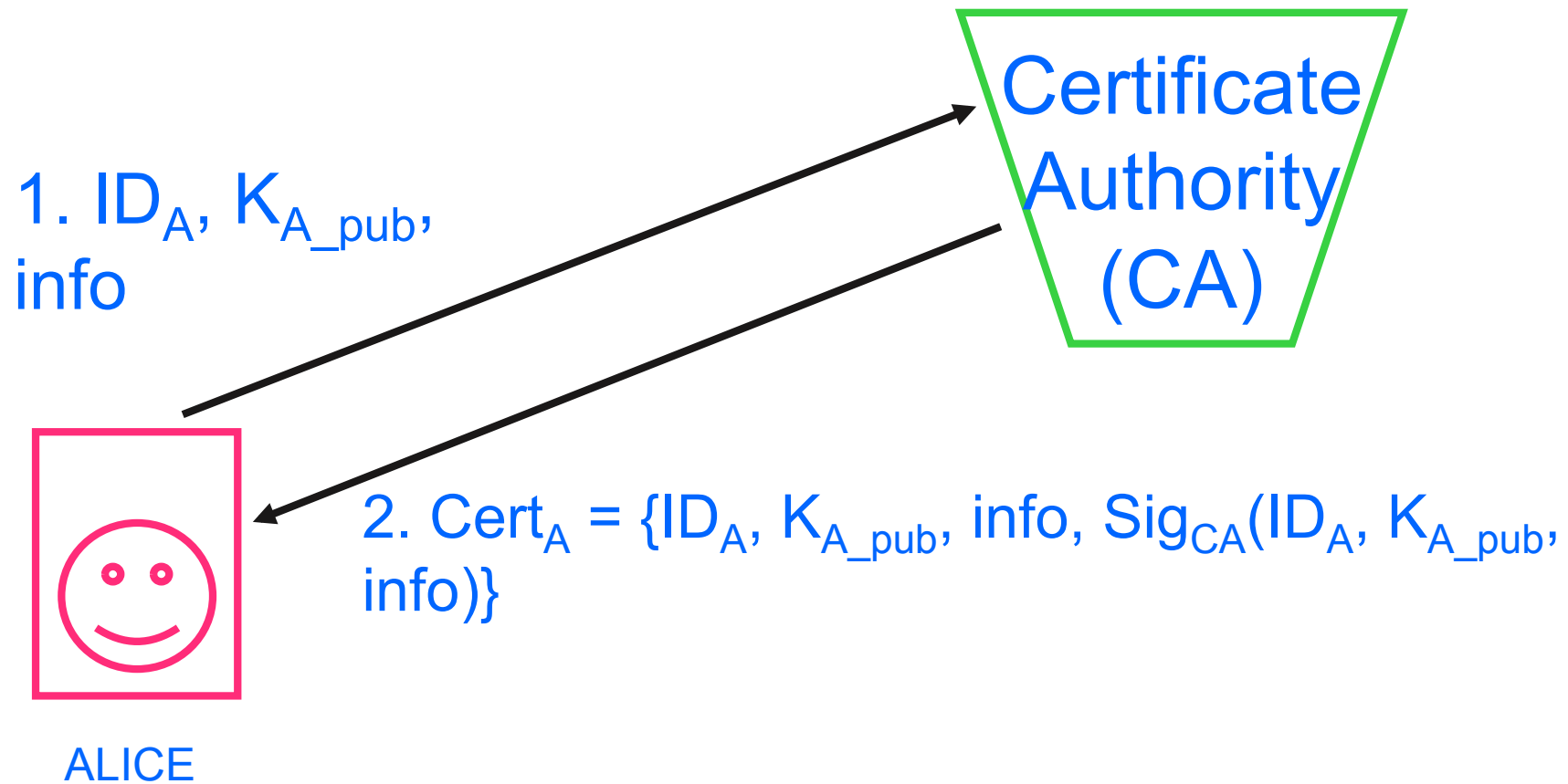


Digital certificates

- To obtain a digital certificate Alice must:
 - make a certificate signing request to the CA
 - Alice sends to CA:
 - her identifier ID_A
 - her public key K_{A_PUB}
 - additional information
 - Alice must supply proof that she is indeed Alice
- CA returns Alice's digital certificate, cryptographically binding her identity to public key:
 - $CertA = \{ID_A, K_{A_pub}, info, Sig_{CA}(ID_A, K_{A_pub}, info)\}$

Digital certificates

How does Alice obtain a digital certificate?





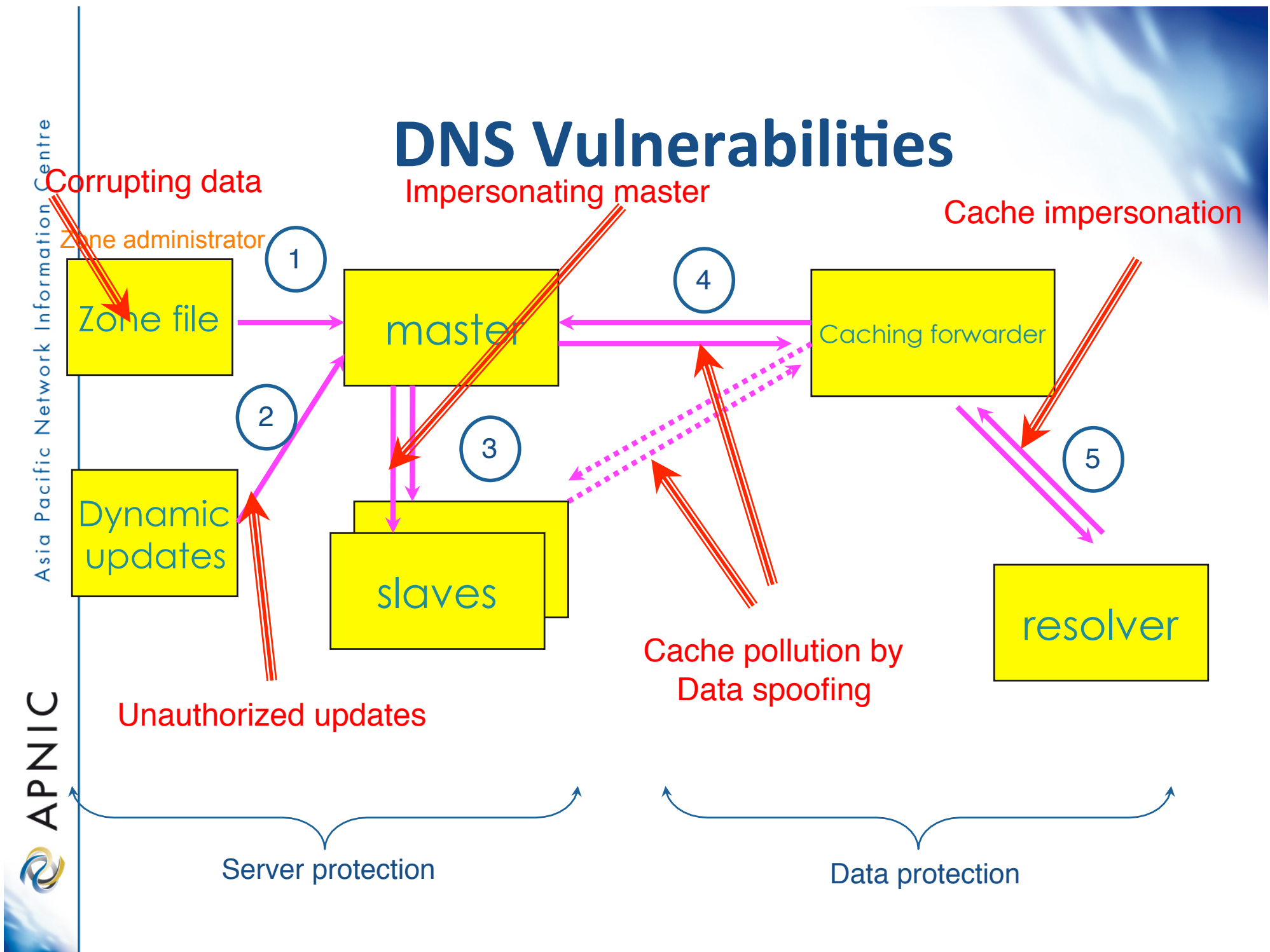
Non-repudiation

- provided using digital signatures:
 - If signature uses something known only to the signer
 - then only signer can have formed the signature
 - so signer cannot deny it
- If Alice denies sending message:
 - Her private key can be tested on original plaintext to prove she must have sent it
- Assumes no compromises of system, keys, etc

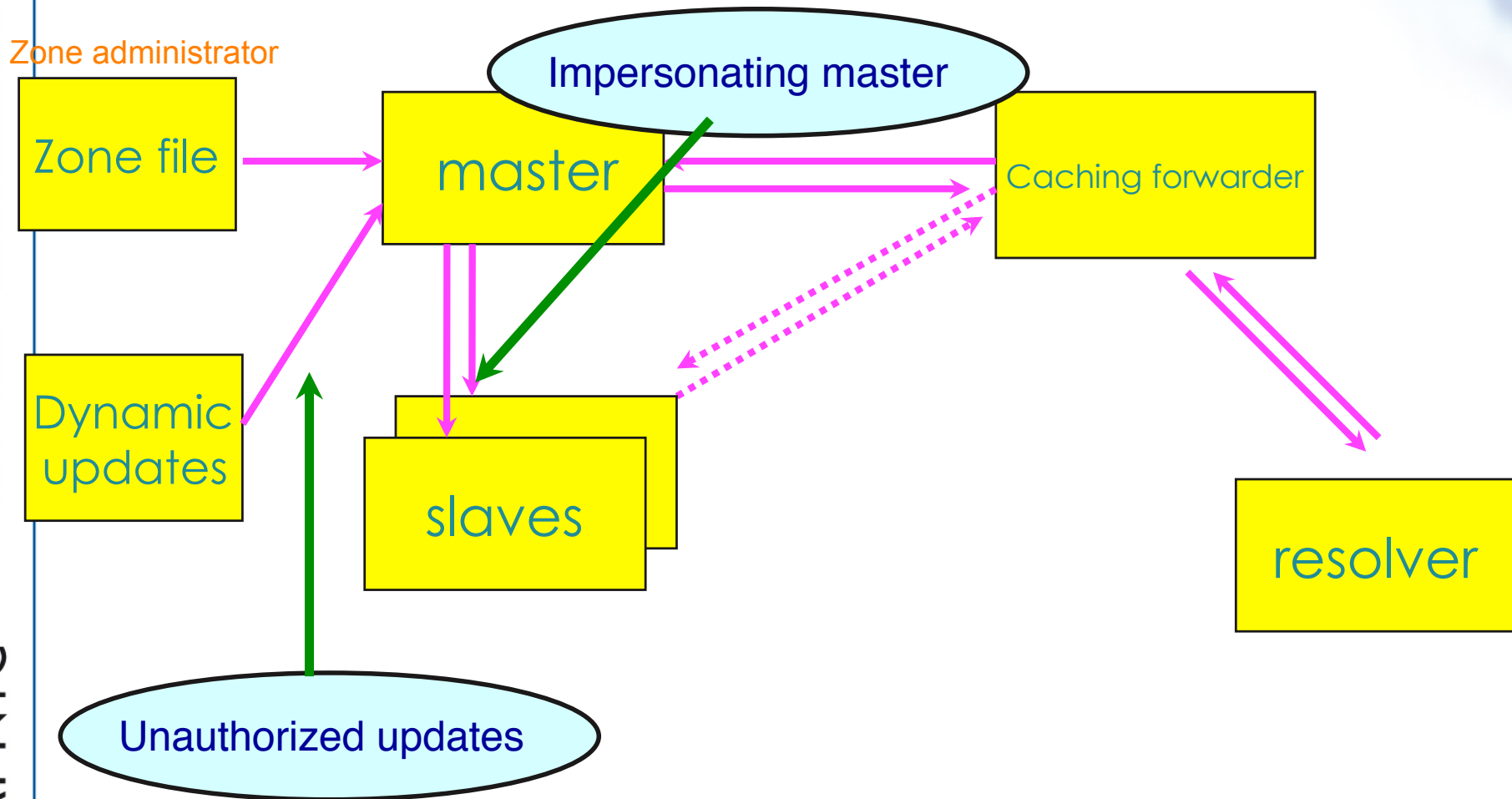
DNS Security



DNS Vulnerabilities

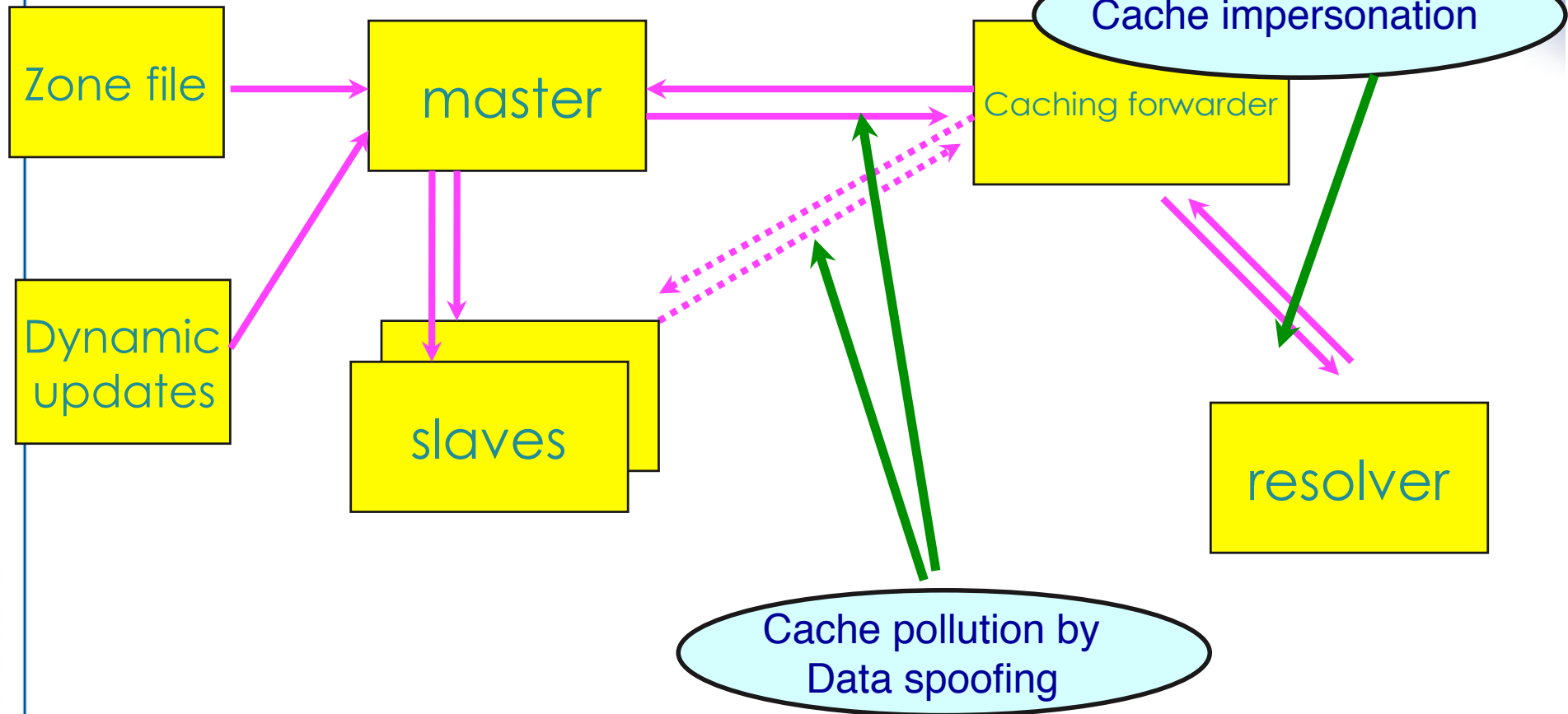


TSIG Protected Vulnerabilities



Vulnerabilities protected by DNSKEY / RRSIG / NSEC

Zone administrator





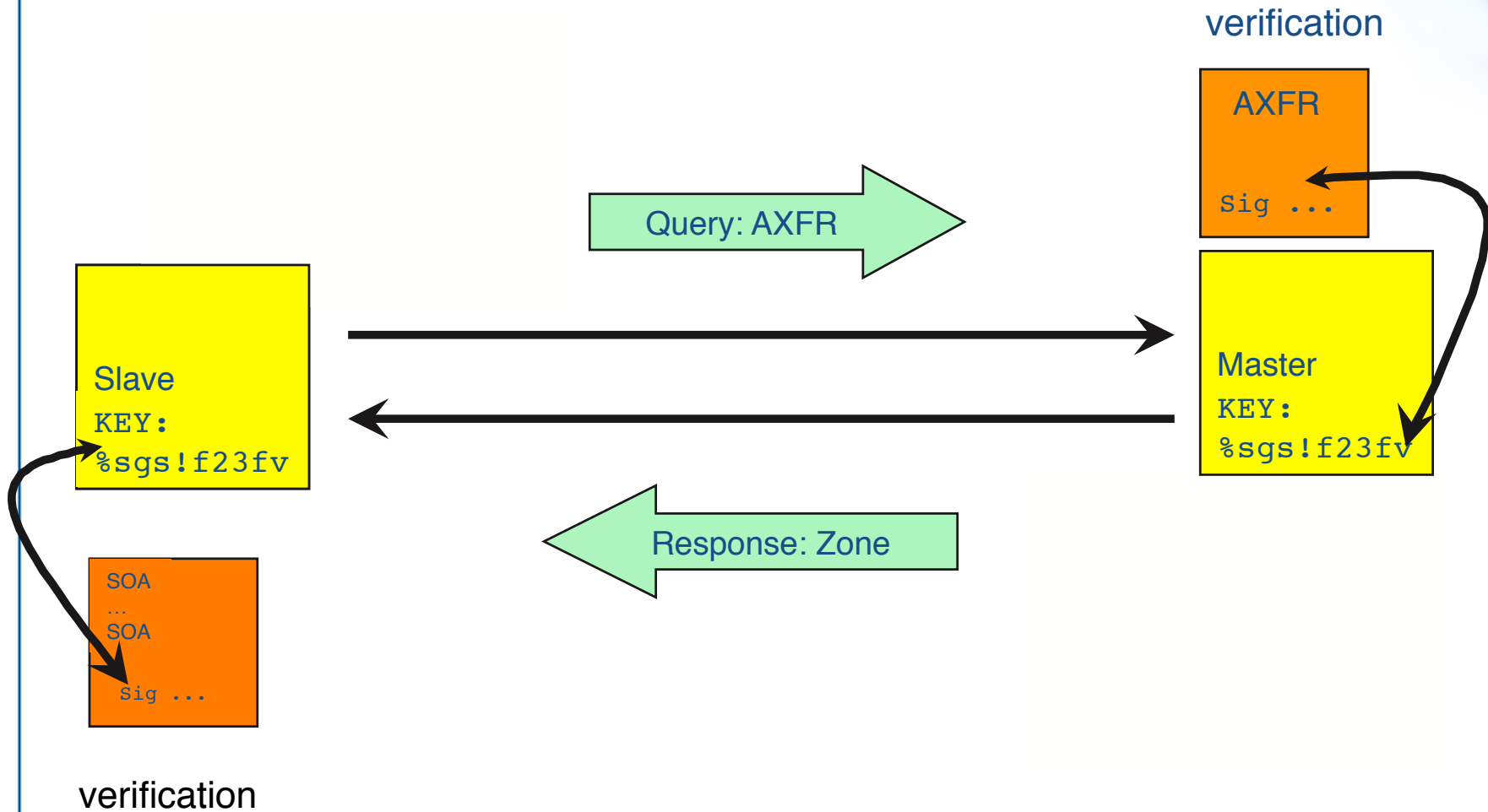
What is TSIG - Transaction Signature?

- A mechanism for protecting a message from a primary to secondary and vice versa
- A keyed-hash is applied (like a digital signature) so recipient can verify message
 - DNS question or answer
 - & the timestamp
- Based on a shared secret - both sender and receiver are configured with it

What is TSIG - Transaction Signature?

- TSIG (RFC 2845)
 - authorizing dynamic updates & zone transfers
 - authentication of caching forwarders
- Used in server configuration, not in zone file

TSIG example



TSIG steps

1. Generate secret
2. Communicate secret
3. Configure servers
4. Test

TSIG - Names and Secrets

- TSIG name
 - A name is given to the key, the name is what is transmitted in the message (so receiver knows what key the sender used)
- TSIG secret value
 - A value determined during key generation
 - Usually seen in Base64 encoding

TSIG – Generating a Secret

- dnssec-keygen
 - Simple tool to generate keys
 - Used here to generate TSIG keys
- `dnssec-keygen -a <algorithm> -b <bits> -n host <name of the key>`

TSIG – Generating a Secret

- Example

```
> dnssec-keygen -a HMAC-MD5 -b 128 -n  
HOST ns1-ns2.pcx.net
```

This will generate the key

```
> Kns1-ns2.pcx.net.+157+15921
```

```
>ls
```

```
➤ Kns1-ns2.pcx.net.+157+15921.key
```

```
➤ Kns1-ns2.pcx.net.+157+15921.private
```

TSIG – Generating a Secret

- TSIG should never be put in zone files!!!
 - might be confusing because it looks like RR:

```
ns1-ns2.pcx.net. IN KEY 128 3 157 nEfRX9...bbPn7lyQtE=
```

TSIG – Configuring Servers

- Configuring the key
 - in named.conf file, same syntax as for rndc
 - `key { algorithm ...; secret ...; }`
- Making use of the key
 - in named.conf file
 - `server x { key ...; }`
 - where 'x' is an IP number of the other server

Configuration Example – named.conf

Primary server 10.33.40.46

Secondary server 10.33.50.35

```
key ns1-ns2.pcx. net {  
    algorithm hmac-md5;  
    secret "APlaceToBe";  
};  
server 10.33.50.35 {  
    keys {ns1-ns2.pcx.net};  
};  
zone "my.zone.test." {  
    type master;  
    file "db.myzone";  
    allow-transfer {  
        key ns1-ns2..pcx.net ;}; };  
};
```

```
key ns1-ns2.pcx.net {  
    algorithm hmac-md5;  
    secret "APlaceToBe";  
};  
server 10.33.40.46 {  
    keys {ns1-ns2.pcx.net};  
};  
zone "my.zone.test." {  
    type slave;  
    file "myzone.backup";  
    masters {10.33.40.46};  
};
```

You can save this in a file and refer to it in the named.conf using 'include' statement:

```
include "/var/named/master/tsig-key-ns1-ns2";
```

TSIG Testing : dig

- You can use dig to check TSIG configuration

- `dig @<server> <zone> AXFR -k <TSIG keyfile>`

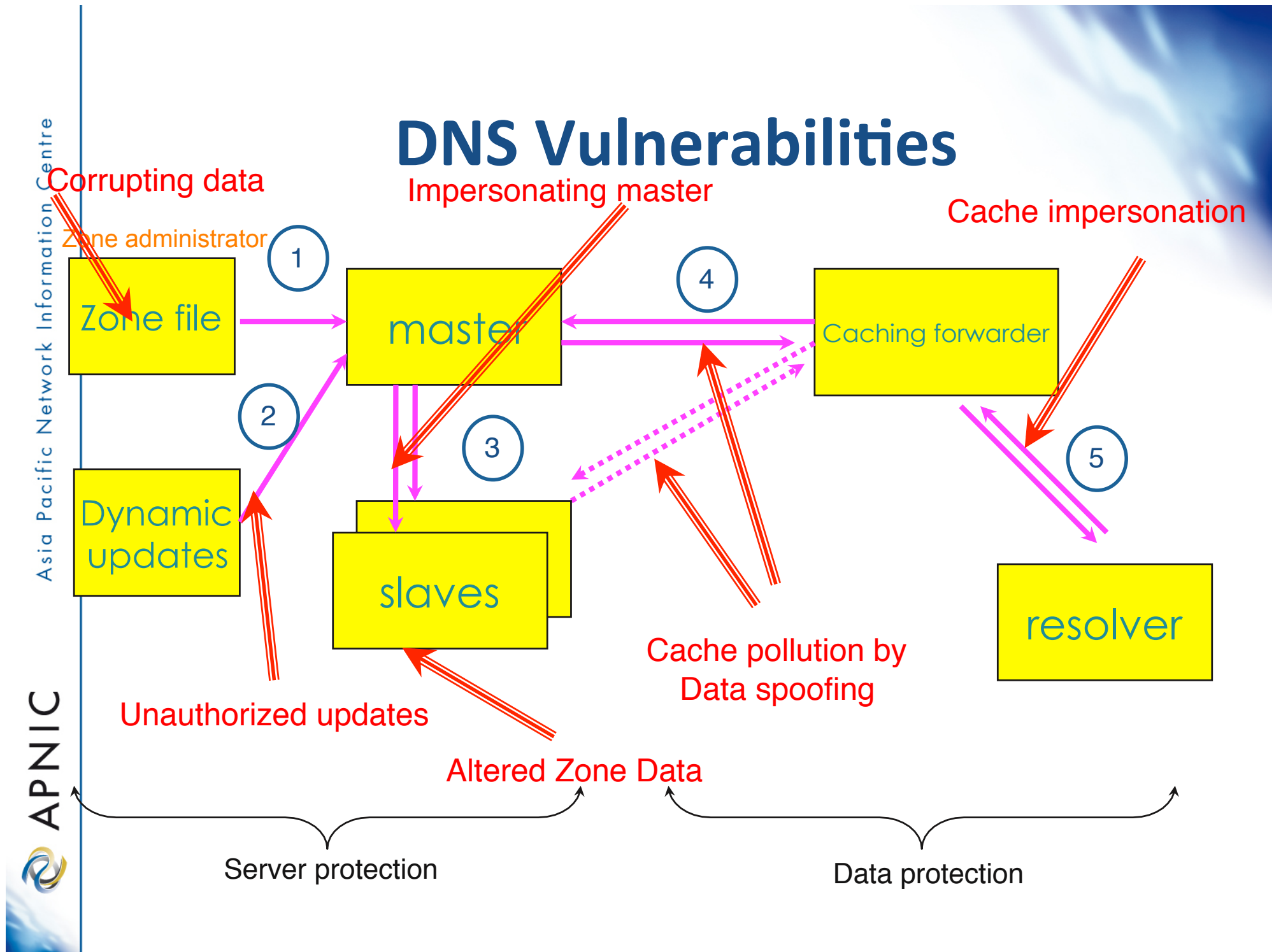
```
$ dig @127.0.0.1 example.net AXFR \  
-k Kns1-ns2.pcx.net.+157+15921.key
```

- Wrong key will give “Transfer failed” and on the server the security-category will log this.

TSIG Testing - TIME!

- TSIG is time sensitive - to stop replays
 - Message protection expires in 5 minutes
 - Make sure time is synchronized
 - For testing, set the time
 - In operations, (secure) NTP is needed

DNS Vulnerabilities



DNSSEC mechanisms

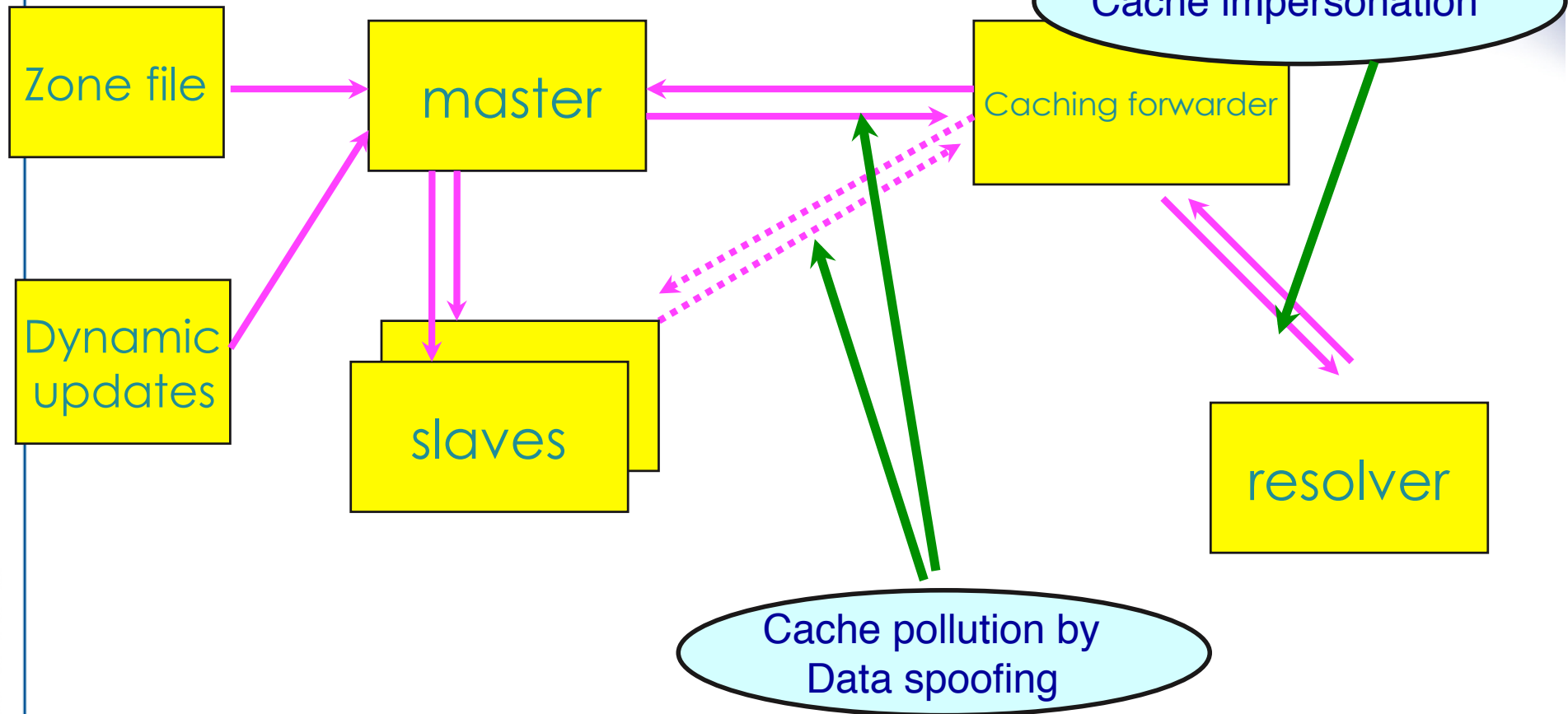
- TSIG: provides mechanisms to authenticate communication between servers
- DNSKEY/RRSIG/NSEC: provides mechanisms to establish authenticity and integrity of data
- DS: provides a mechanism to delegate trust to public keys of third parties
- A secure DNS will be used as an infrastructure with public keys
 - However it is **NOT** a PKI

DNSSEC mechanisms

- Key pair
 - A private(secret) key and a corresponding public key
- In DNSSEC,
 - If you know the public key, you can verify a signature created with the private key
 - Only uses signatures
- Public Key Crypto
 - If you know the public key, you can encrypt data that can only be decrypted with the private key

Vulnerabilities protected by DNSKEY / RRSIG / NSEC

Zone administrator



Authenticity and Integrity

- Authenticity
 - Is the data published by the entity we think is authoritative
- Integrity
 - Is the data received the same as what was published?
- Islands of security
 - We cannot expect that every name server in the world would configure to support DNSSEC and every zone is secured
 - Security aware name servers and Security not aware name servers



Publishing keys

- A zone is signed using its private key
- Receiving name server must have access to zone's public key in order to perform the security verification
- How to obtain public key
 - Publish the public key using DNSKEY RR in the zone file
 - Obtain the key using out of band process
 - Trusted anchor (defined using *trusted-keys* statement in config file)

Response from name servers

- Secure
 - A trusted anchor is present for the zone and has been used to validate the received data successfully
 - Authenticated Data (AD) bit is set
- Insecure
 - A trusted anchor is present and information allows the name server to prove that at a delegation point there is no secure link to the zone
 - Parent is secure but child is not secure

Response from name servers

- Bogus
 - A trusted anchor exists, but the data failed to authenticate at the receiving name server using the trusted anchor
 - An attempt to spoof any response from the domain
- Indeterminate
 - There is no trusted anchor for the domain

DNSSEC RRs

- Data authenticity and integrity by signing the Resource Records Sets with private key
- Public DNSKEYs used to verify the RRSIGs
- Children sign their zones with their private key
 - Authenticity of that key established by signature/checksum by the parent (DS)
- Ideal case: one public DNSKEY distributed

New Resource Records

- 3 Public key crypto related RRs
 - RRSIG
 - Signature over RRset made using private key
 - DNSKEY
 - Public key, needed for verifying a RRSIG
 - DS
 - Delegation Signer; 'Pointer' for building chains of authentication
- One RR for internal consistency
 - NSEC
 - Indicates which name is the next one in the zone and which typecodes are available for the current name
 - authenticated non-existence of data

RR's and RRsets

- Resource Record:

- Name TTL class type rdata
www.example.net. 7200 IN A 192.168.1.1

- RRset: RRs with same name, class **and** type:

| | | | | |
|------------------|------|----|---|-------------|
| www.example.net. | 7200 | IN | A | 192.168.1.1 |
| | | | A | 10.0.0.3 |
| | | | A | 172.10.1.1 |

- RRsets are signed, not the individual RRs

DNSKEY RDATA

Example:

```
example.net. 3600 IN DNSKEY 256 3 5  
(  
    AQOvhvXXU61Pr8sCwELcqqq1g4JJ  
    CALG4C9EtraBKVd+vGIF/unwigfLOA  
    O3nHp/cgGrG6gJYe8OWKYNgq3kDChN)
```

RRSIG RDATA

```
example.net. 3600 IN RRSIG A 5 2 3600 (
    20081104144523 20081004144523 3112 example.net.
    VJ+8ijXvbrTLeoAiEk/qMrdudRnYZM1VlqhNvhYuAcYKe2X/
    jqYfMfjfSURmhPo+0/GOZjW66DJubZPmNSYXw== )
```

Delegation Signer (DS)

- Delegation Signer (DS) RR indicates that:
 - delegated zone is digitally signed
 - indicated key is used for the delegated zone
- Parent is authoritative for the DS of the child's zone
 - Not for the NS record delegating the child's zone!
 - DS **should not** be in the child's zone

DS RDATA

```
$ORIGIN .net.
example.net.      3600 IN      NS       ns.example.net
ns.example.net.   3600 IN      DS       3112  5 1 (
                                     239af98b923c023371b52
                                     1g23b92da12f42162b1a9
                                     )
```


NSEC RDATA

- Points to the next domain name in the zone
 - also lists what are all the existing RRs for “name”
 - NSEC record for last name “wraps around” to first name in zone
- Used for authenticated denial-of-existence of data
 - authenticated non-existence of TYPEs and labels

NSEC Record example

```
$ORIGIN example.net.
```

```
@ SOA      ...
```

```
    NS      NS.example.net.
```

```
    DNSKEY   ...
```

```
    NSEC     mailbox.example.net. SOA NS NSEC DNSKEY
```

```
    RRSIG
```

```
mailbox    A      192.168.10.2
```

```
           NSEC   www.example.net.  A NSEC RRSIG
```

```
WWW        A      192.168.10.3
```

```
           TXT    Public webserver
```

```
           NSEC   example.net. A NSEC RRSIG TXT
```

Setting up a secure zone

Enable dnssec

- In the named.conf,

```
Options {  
    directory “....”  
    dnssec-enable yes;  
};
```

Creation of keys

- Zones are digitally signed using the private key
- Can use RSA-SHA-1, DSA-SHA-1 and RSA-MD5 digital signatures
- The public key corresponding to the private key used to sign the zone is published using a DNSKEY RR

Keys

- Two types of keys
 - Zone Signing Key (ZSK)
 - Sign the RRsets within the zone
 - Public key of ZSK is defined by a DNSKEY RR
 - Key Signing Key (KSK)
 - Signed the keys which includes ZSK and KSK and may also be used outside the zone
 - Trusted anchor in a security aware server
 - Part of the chain of trust by a parent name server
- Using a single key or both keys is an operational choice (RFC allows both methods)

Create key pairs

- To create Zone Signing Key (ZSK)
 - `dnssec-keygen -a rsasha1 -b 1024 -n zone champika.net`
- After key generations (ZSK) you will see 2 files have been created
 - Files with *.key* and *.private* extensions
 - *.key* file contains your public key where as *.private* file contains your private key

Publishing your public key

- Using \$INCLUDE you can call the public key (DNSKEY RR) inside the zone file
 - \$INCLUDE /path/Kchampika.net.+005+33633.key ; ZSK
- You can also manually enter the DNSKEY RR in the zone file

Signing the zone

- > `dnssec-signzone -o champika.net db.champika.net
Kchampika.net.+005+33633`
- Once you sign the zone a file with a .signed extension will be created
 - db.champika.net.signed

Signed Zone

- Observe the signed zone file
- Resource Records
 - DNSKEY
 - RRSIG
 - NSEC

Only authoritative records are signed

- NS records for the zone itself are signed
- NS records for delegations are not signed
 - DS RRs are signed!
- Glue is not signed
- Difference in the file size
 - *Db.champika.net* Vs *db.champika.net.signed*

Updates to the config file

- Modify the zone statement
- Replace the previous zone file with the signed zone file

Publishing the signed zone

- Edit named.conf:

```
zone "champika.net" {  
    type master;  
    file "champika.net.signed";  
    allow-transfer { 192.168.1.2 ;  
                    key ns1-ns2.champika.net.; };  
    notify yes;  
};
```

- Use named-checkconf
- Reload zone
- Test

Testing the server

- Ask a dnssec enabled question from the server and see whether the answer contains dnssec-enabled data
 - Basically the answers are signed
- `dig @localhost www.champika.net +dnssec +multiline`

Testing with dig: an example

```

Terminal — bash — 144x46
bash-3.2# dig @localhost www.champika.net +dnssec +multiline

; <<>> DiG 9.6.0-APPLE-P2 <<>> @localhost www.champika.net +dnssec +multiline
; (3 servers found)
;; global options: +cmd
;; Got answer:
;; -->HEADER<-- opcode: QUERY, status: NOERROR, id: 37425
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITIONAL: 3

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;www.champika.net.      IN A

;; ANSWER SECTION:
www.champika.net.      86400 IN A 192.168.1.2
www.champika.net.      86400 IN RRSIG A 5 3 86400 20091123163643 (
                        20091024163643 22827 champika.net.
                        EypIIVyQyYBLK0X2u/LT1+40xjBomXzLrcdwSErgioMb
                        pGyDwDLzP+FTbE3QCfBMLNDt2AGoYcty1cfY4li9sHkw
                        fue6hTQTsm0LhisBkVKQBy6ZD5oGiJQgaIkBgMltVhPh
                        jGJ8Z1UhbWkCgGK13doAa+5X8mx6MXNCudiNWeg= )

;; AUTHORITY SECTION:
champika.net.          86400 IN NS ns.champika.net.
champika.net.          86400 IN RRSIG NS 5 2 86400 20091123163643 (
                        20091024163643 22827 champika.net.
                        CZsPewlhPwPyTl8wPh09QhD6pWt0If2mLVshviGKq4no
                        ISNVoijmX0LyIns+o3DZz/2+TtwoQCRFLbfi99YMS3fx
                        BHGYqFDeGItYVx3oBpmTuAtMu2+od5WFS+LC1sJsEP/N
                        QvUDgtWrrj8+Z0wVVj8aLe+I51h29ek7Mzk7+P4E= )

;; ADDITIONAL SECTION:
ns.champika.net.       86400 IN A 192.168.1.1
ns.champika.net.       86400 IN RRSIG A 5 3 86400 20091123163643 (
                        20091024163643 22827 champika.net.
                        eTP05c4GscnoC9V5sR6vgDo02WgCr1T5arU7YZhWctXI
                        vkmU1ni+wguaW6xezfb/Eu4J69bMnpQoX2zWUDtLUCM
                        +FVLsFx4Bbt+BjPEJKV03g9vv6IdkkR/pxyE1kJWJWmI
                        tR49P2dywlzqqTyvnj3F1yuFRTLHhJvfCvc+n8w= )

;; Query time: 3 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Sun Oct 25 03:40:38 2009
;; MSG SIZE rcvd: 610

```

Thank you! 😊

champika@apnic.net