DNSSEC
Security Extensions for DNS

Champika Wijayatunga
<champika@icann.org>

ICANN

+1-202- HealthCare.gov

US-NSTIC effort

VoIP

DNS is a part of all IT ecosystems

Google

OpenID

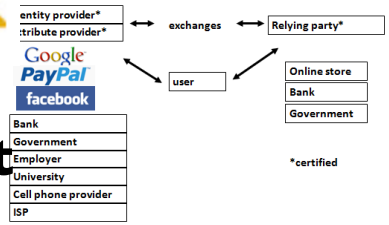
facebook

eBay

AADHAAR

PayPal

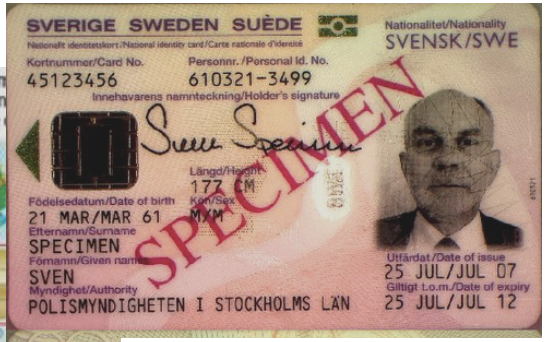
OECS ID effort



amazon Prime

e-Passport symbol

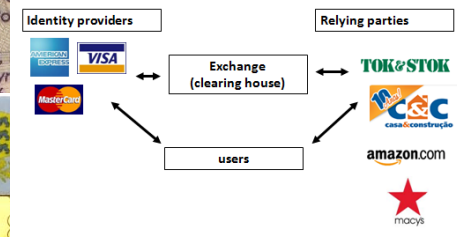
ICP Brasil
O Brasil na era da certificação digital



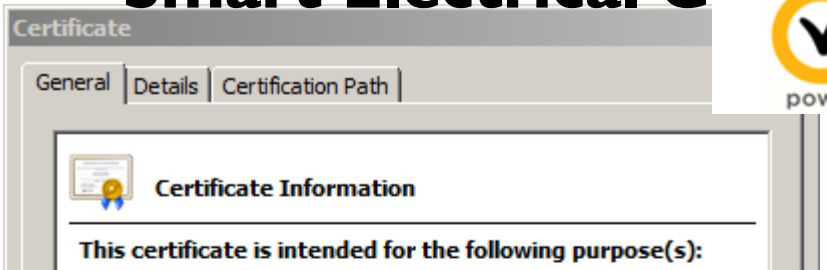
DigiNotar
A VASCO COMPANY

COMODO
Creating Trust Online

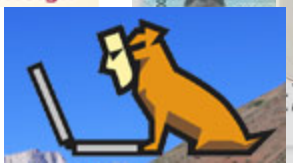
Trust frameworks are not new



Smart Electrical G



Norton SECURED
powered by VeriSign



CANN

mydomainname.com

lamb@xtcn.com

Where DNSSEC fits in

- ..but CPU and bandwidth advances make legacy DNS vulnerable to MITM attacks
- DNS Security Extensions (DNSSEC) introduces digital signatures into DNS to cryptographically protect contents
- With DNSSEC fully deployed a business can be sure a customer gets un-modified data (and visa versa)

ICANN

The Bad: DNSChanger - 'Biggest Cybercriminal Takedown in History' – 4M machines, 100 countries, $14M

Nov 2011 http://krebsonsecurity.com/2011/11/malware-click-fraud-kingpins-arrested-in-estonia/
End-2-end DNSSEC validation would have avoided the problems

ICANN

The Bad: Other DNS hijacks*

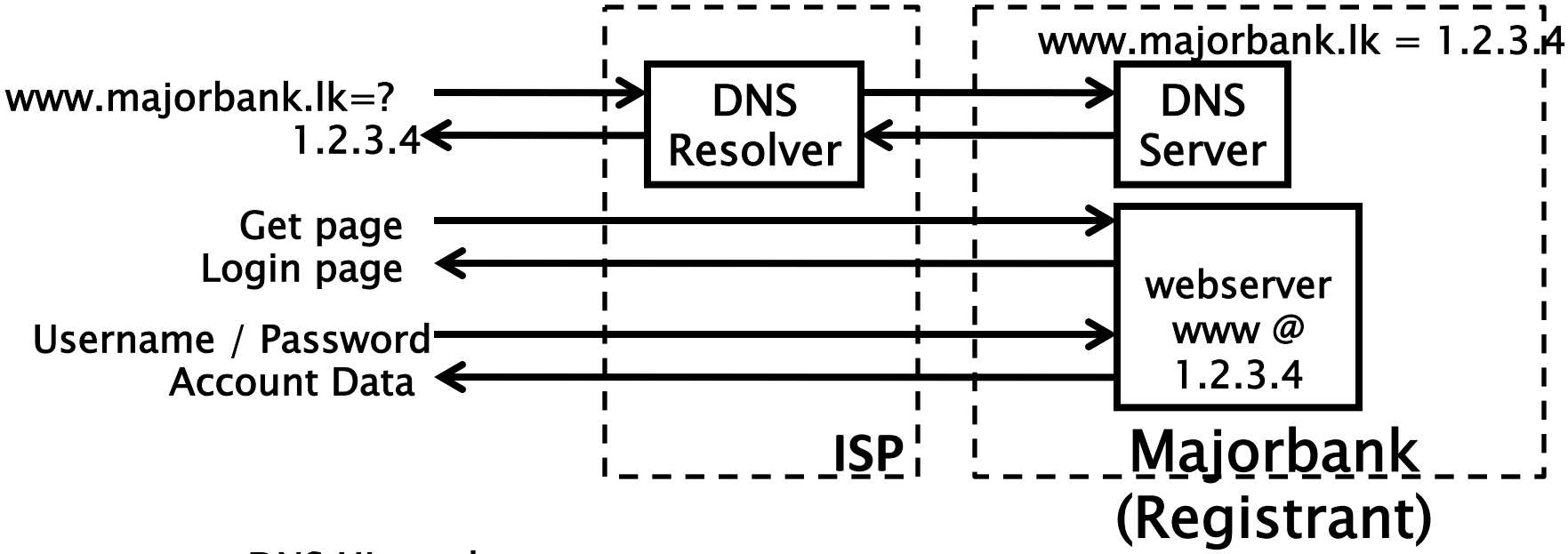
- 25 Dec 2010 - Russian e-Payment Giant ChronoPay Hacked
- 18 Dec 2009 – Twitter – “Iranian cyber army”
- 13 Aug 2010 - Chinese gmail phishing attack
- 25 Dec 2010 Tunisia DNS Hijack
- 2009-2012 google.*
  - April 28 2009 Google Puerto Rico sites redirected in DNS attack
  - May 9 2009 Morocco temporarily seize Google domain name
- 9 Sep 2011 - Diginotar certificate compromise for Iranian users
- SSL / TLS doesn't tell you if you've been sent to the correct site, it only tells you if the DNS matches the name in the certificate. Unfortunately, majority of Web site certificates rely on DNS to validate identity.
- DNS is relied on for unexpected things though insecure.

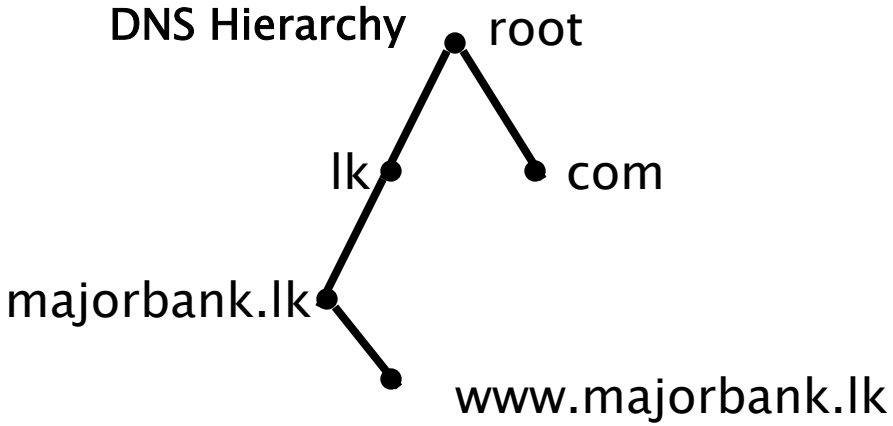
DigiNotar
A VASCO COMPANY

*A Brief History of DNS Hijacking - Google
http://costarica43.icann.org/meetings/sanjose2012/presentation-dns-hijackings-marquis-boire-12mar12-en.pdf

ICANN

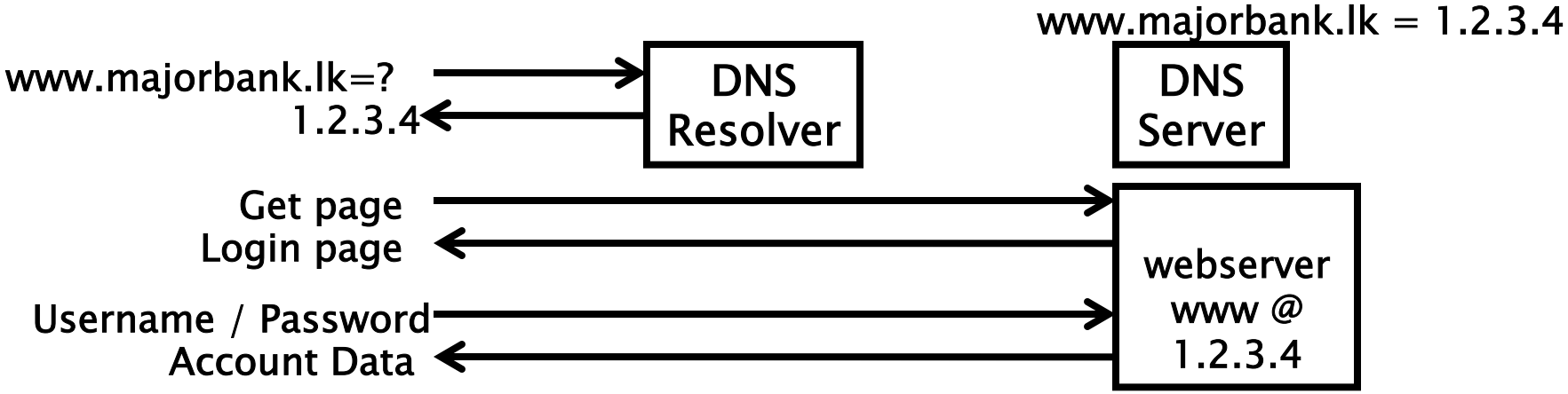
The Internet's Phone Book - Domain Name System (DNS)



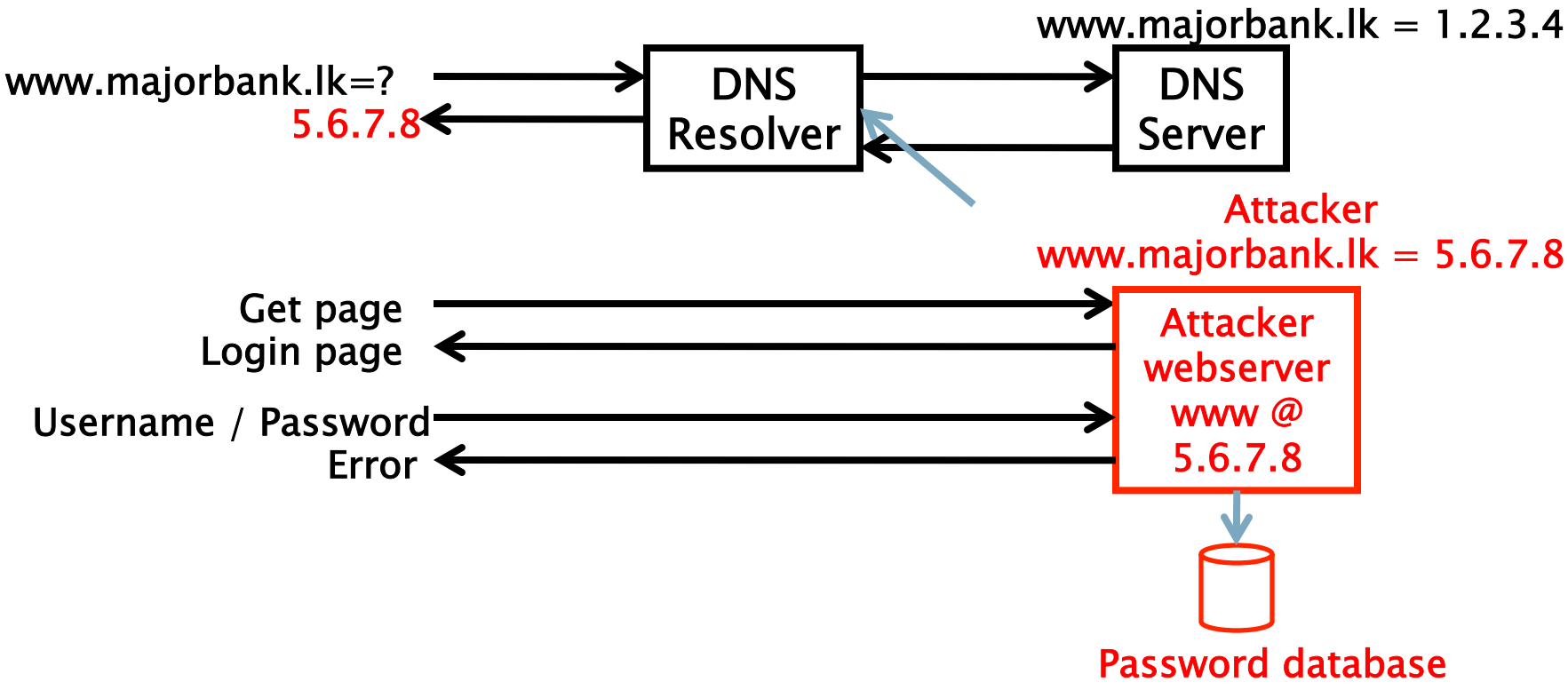


ICANN

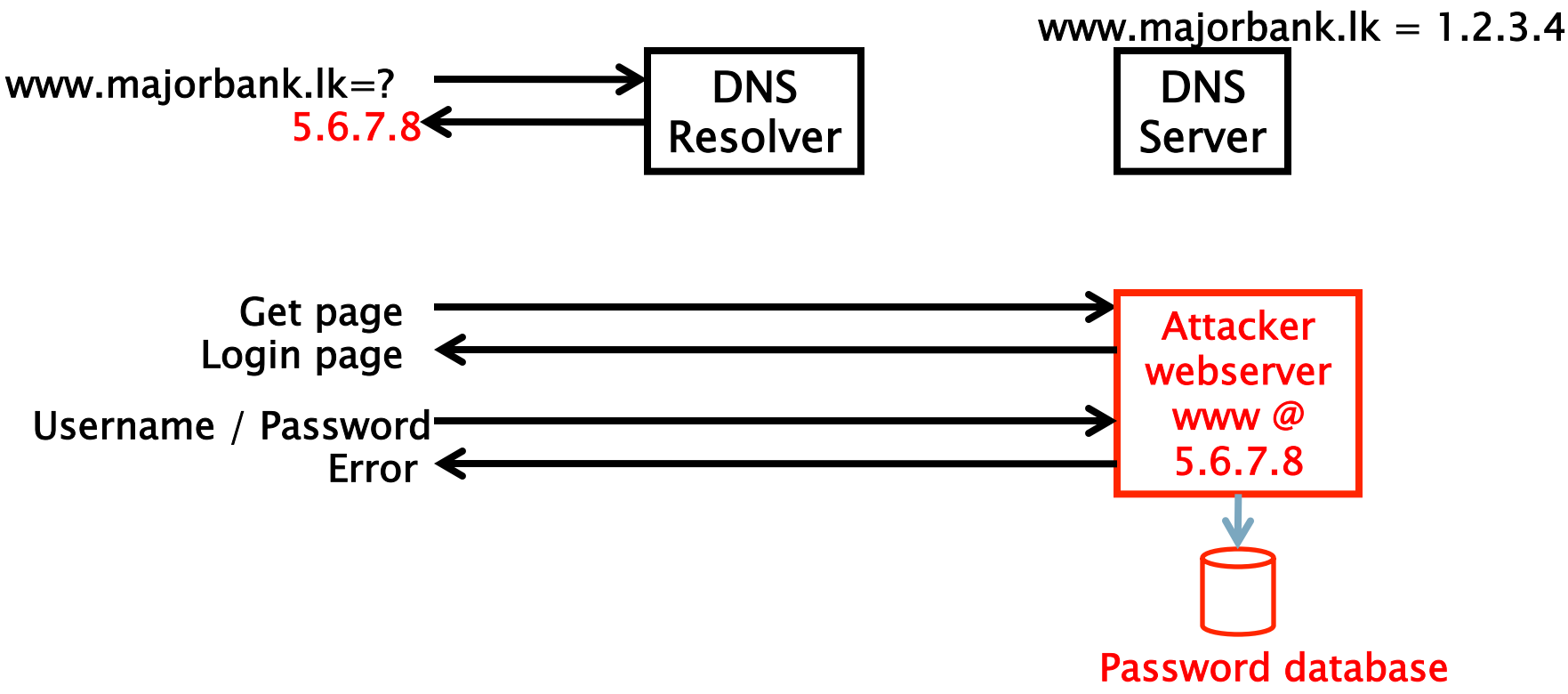
Caching Responses for Efficiency



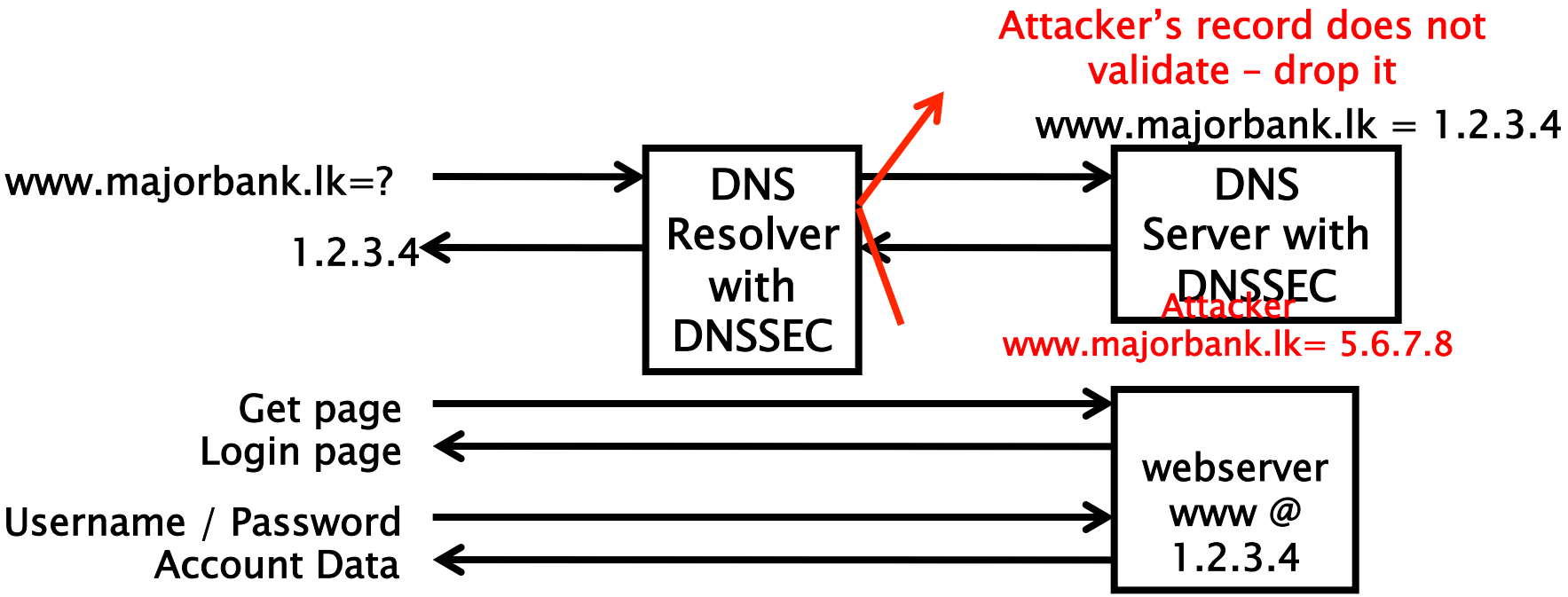
The Problem:
DNS Cache Poisoning Attack



Argghh! Now all ISP customers get sent to attacker.

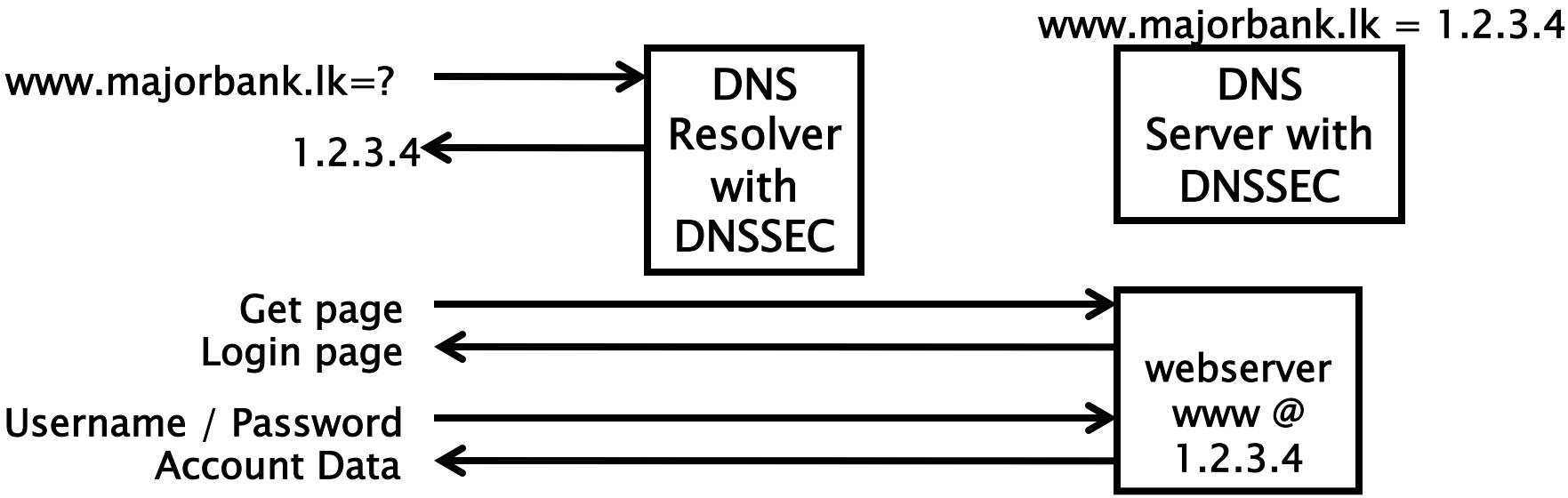


Securing The Phone Book - DNS Security Extensions (DNSSEC)



ICANN

Resolver only caches validated records



The Business Case for DNSSEC

- Cyber security is becoming a greater concern to enterprises, government, and end users. DNSSEC is a key tool and differentiator.
- DNSSEC is the biggest security upgrade to Internet infrastructure in over 20 years. It is a platform for new security applications (for those that see the opportunity).
- DNSSEC infrastructure deployment has been brisk but requires expertise. Getting ahead of the curve is a competitive advantage.

ICANN

DNSSEC - Where we are

- Deployed on 596/779 TLDs (15 Dec 2014 - 76%)
- Root signed** and audited
- Required in new gTLDs. Basic support by ICANN registrars
- Growing ISP support*.
- 3rd party signing solutions***
- Growing S/W H/W support: NLNetLabs, ISC, Microsoft, PowerDNS, Secure64…? openssl, postfix, XMPP, mozilla: early DANE support
- IETF standard on DNSSEC SSL certificates (RFC6698)
- Growing support from major players…(Apple iPhone/ iPad, Google 8.8.8.8,…)

*** COMCAST /w 20M and others; most ISPs in SE ,CZ. AND ~12% of resolvers validate using DNSSEC**

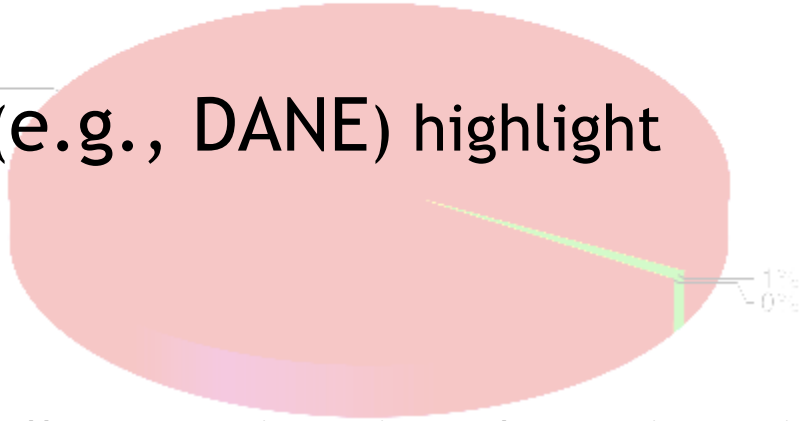
****Int'l bottom-up trust model /w 21 TCRs from: TT, BF, RU, CN, US, SE, NL, UG, BR, Benin, PT, NP, Mauritius, CZ, CA, JP, UK, NZ…**

***** Partial list of registrars: https://www.icann.org/en/news/in-focus/dnssec/deployment**

ICANN

But…

- But deployed on ~1-2% (3.5M) of 2nd level domains. Many have plans. Few have taken the step (e.g., yandex.com, paypal.com*, comcast.com).
- DNSChanger and other attacks highlight today's need. (e.g end-2-end DNSSEC validation would have avoided the problems)
- Innovative security solutions (e.g., DANE) highlight tomorrow's value.

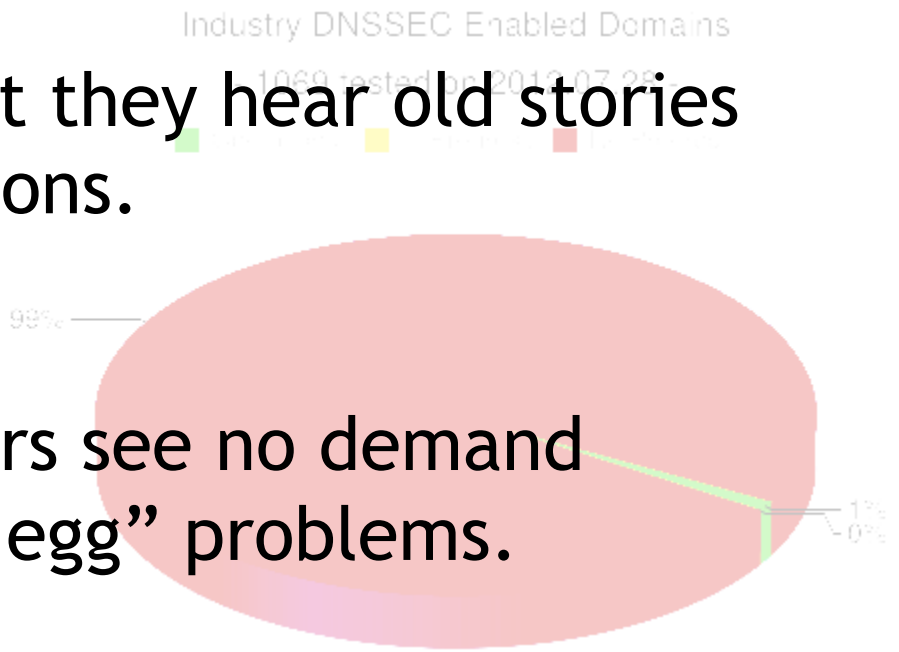


* http://fedv6-deployment.antd.nist.gov/cgi-bin/generate-com http://www.thesecuritypractice.com/the_security_practice/2011/12/all-paypal-domains-are-now-using-dnssec.html
http://www.nacion.com/2012-03-15/Tecnologia/Sitios-web-de-bancos-ticos-podran-ser-mas-seguros.aspx

ICANN

DNSSEC: So what's the problem?

- Not enough IT departments know about it or are too busy putting out other security fires.
- When they do look into it they hear old stories of lack of turnkey solutions.
- Registrars*/DNS providers see no demand leading to "chicken-and-egg" problems.



*but required by new ICANN registrar agreement

ICANN

What you can do

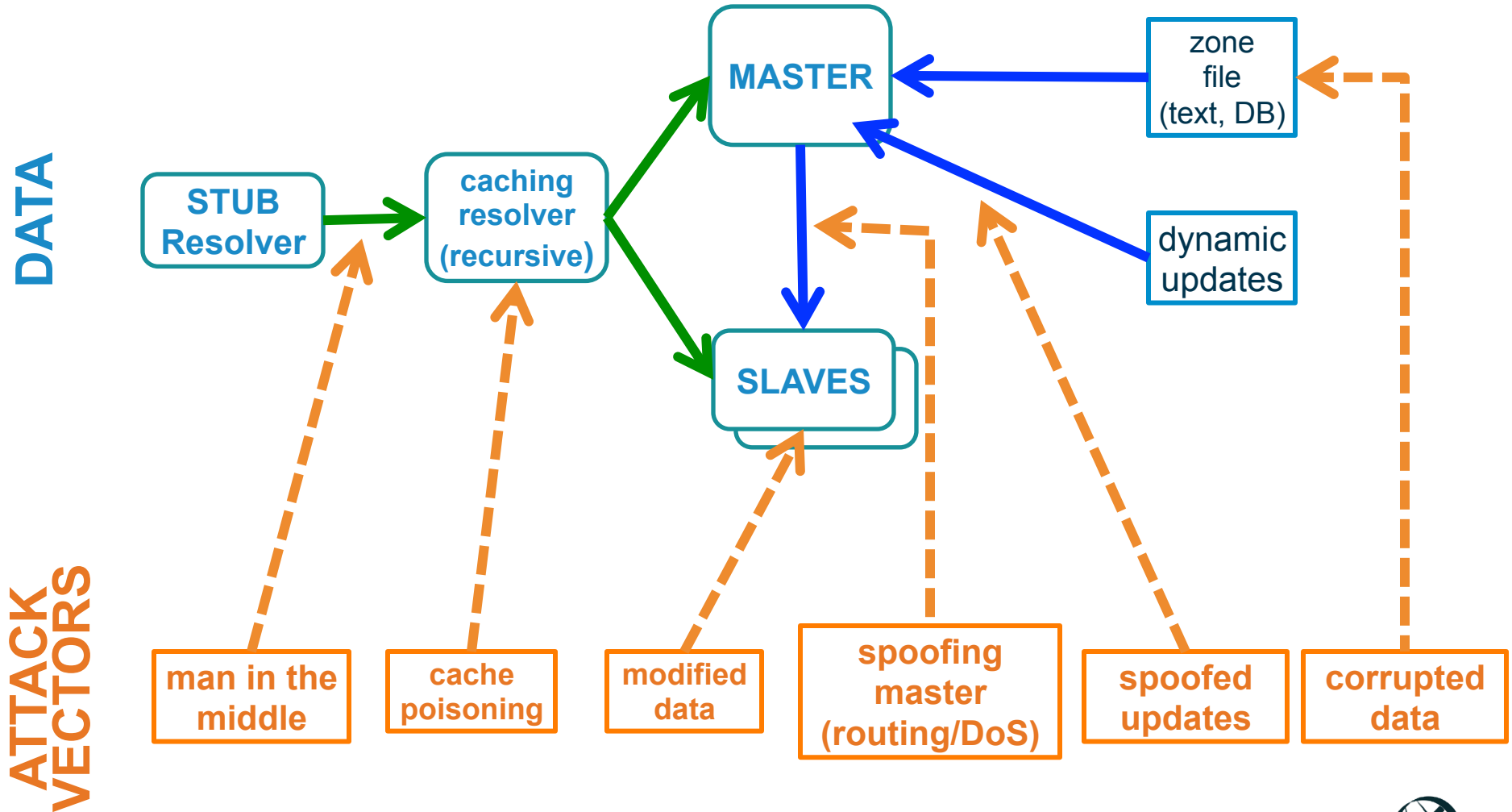
- ***For Companies:***
  - Sign your corporate domain names
  - Just turn on validation on corporate DNS resolvers
- ***For Users:***
  - Ask ISP to turn on validation on their DNS resolvers
- ***For All:***
  - Take advantage of organizations offering DNSSEC education and training

ICANN

DNSSEC Implementation

ICANN

DNS Data flow



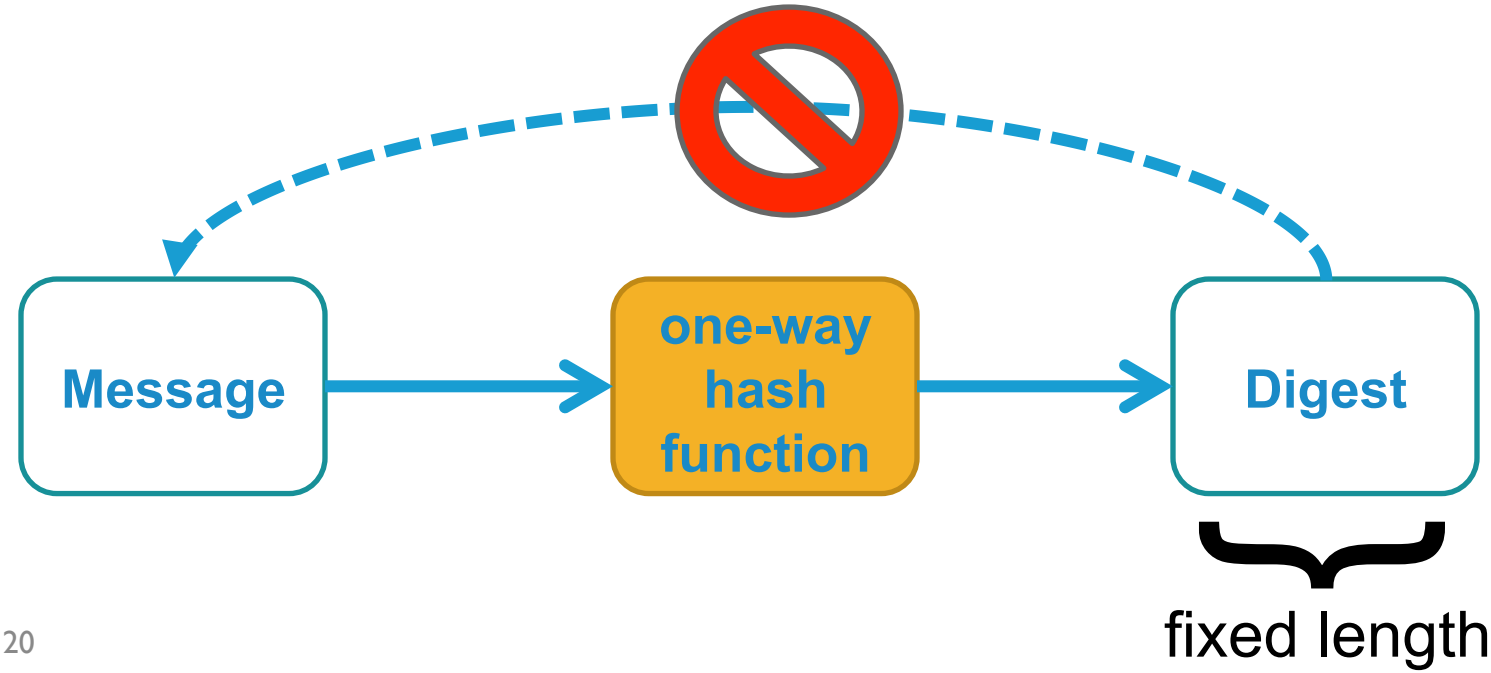
ICANN

Brief reminder on cryptography

- Nowadays most of our Security Services are based in one (or a combination) of the following areas:
  - One-way hash functions
  - Symmetric key crypto
  - Public-key crypto (or asymmetric)

One-way hash functions

- Takes a **message** of any length as input
- Delivers a short and fixed length output (usually called **hash** or **digest**)
- It can be used as a *digital signature*.

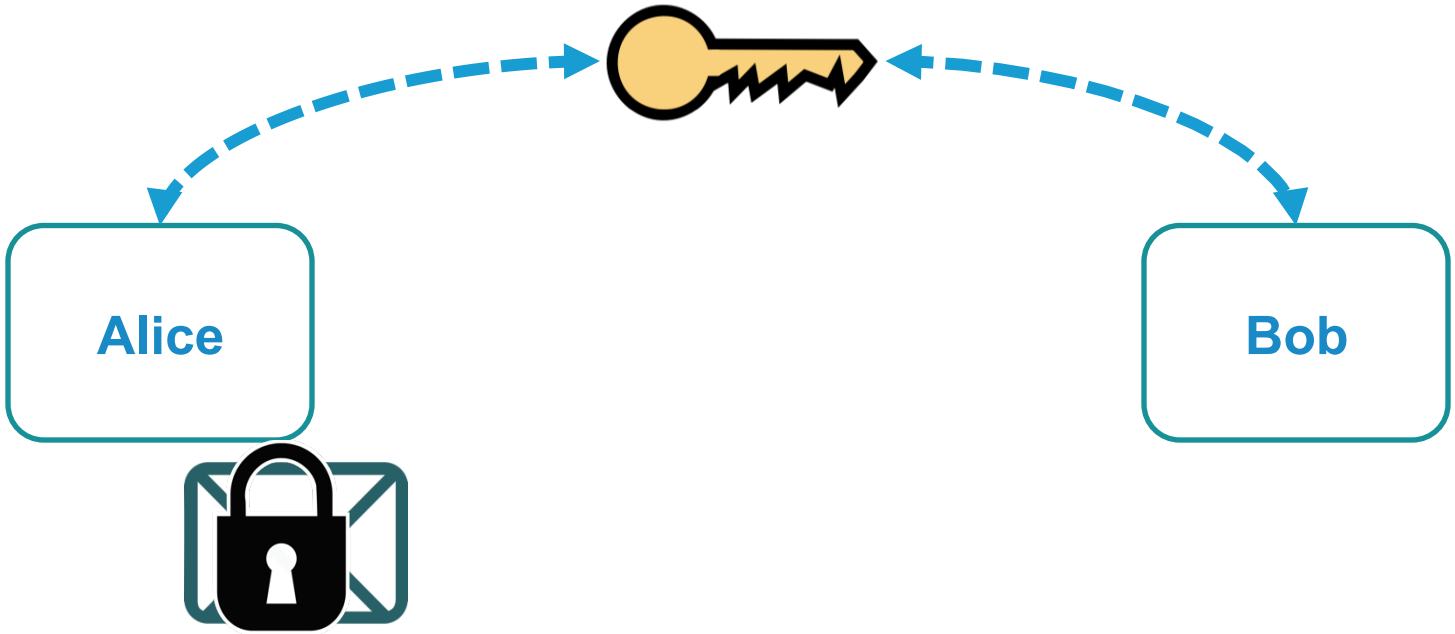


One-way hash functions

- Most common
  - MD5
  - SHA-1
  - SHA-2
  - GOST
  - HAVAL
  - DES
  - checksum
  - CRC{16,32,64}

Symmetric key crypto

- Both sender and receiver share a key.
- Key is used to encrypt a message, which can be decrypted by the same key



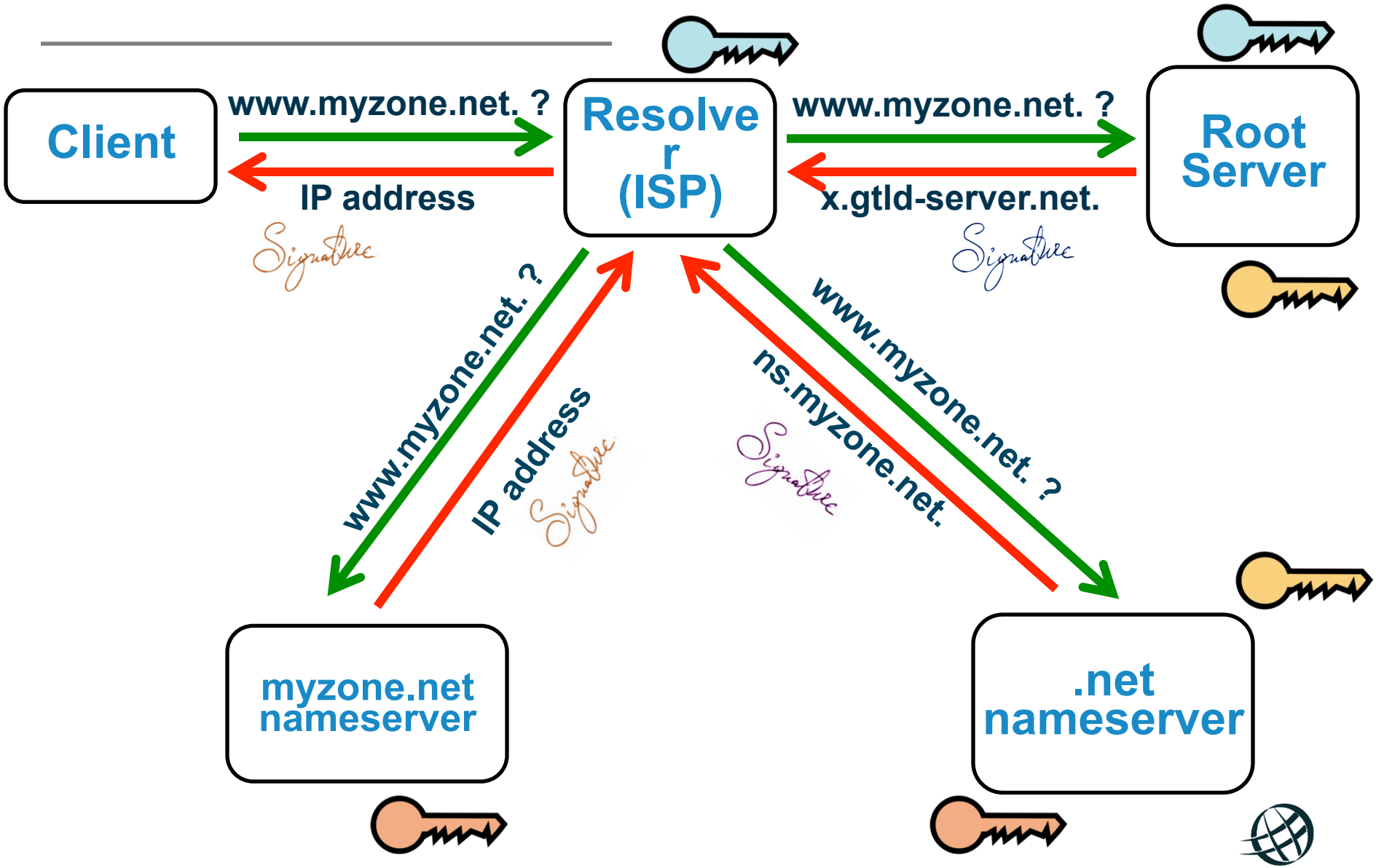
Asymmetric or Public Key crypto

- Sender has a pair of keys. One is private and the other one is public.
- Each key is able to encrypt a message, while the other key can do the opposite.

Public Key crypto functions

- RSA (Riverst Shamir Adleman)
- DSA (Digital Signature Algorithm)
- ElGamal
- ECDSA (Elliptic Curve Digital Signature Algorithm)

How DNSSEC Works



ICANN

How DNSSEC Works

- Data authenticity and integrity by signing the Resource Records Sets with a private key
- Public DNSKEYs published, used to verify the RRSIGs
- Children sign their zones with their private key
  - Authenticity of that key established by parent signing hash (DS) of the child zone's key
- Repeat for parent…
- Not that difficult on paper
  - Operationally, it is a bit more complicated

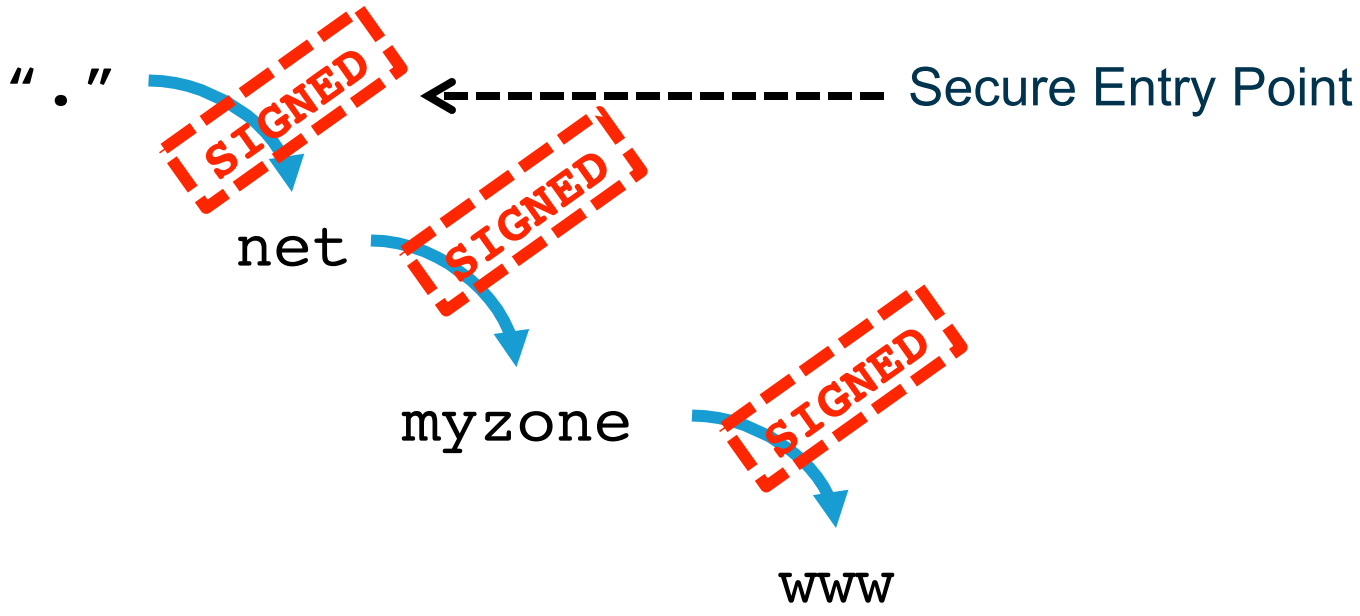
New concepts

New Concepts

- Secure Entry Point and Chain of Trust
  - Delegating Signing Authority
- New RRs
  - DNSKEY, RRSIG, NSEC/NSEC3 and DS
- Signature expiration
- Key Rollovers

Chain of trust and Secure Entry Point

- Using the existing delegation based model of distribution/
- Don’t sign the entire zone, sign a RRset
- Parent **DOES NOT** sign the child zone. The parent signs a pointer (hash) to the key used to sign the data of the child zone (DS record)
- Example with **`www.myzone.net.`**

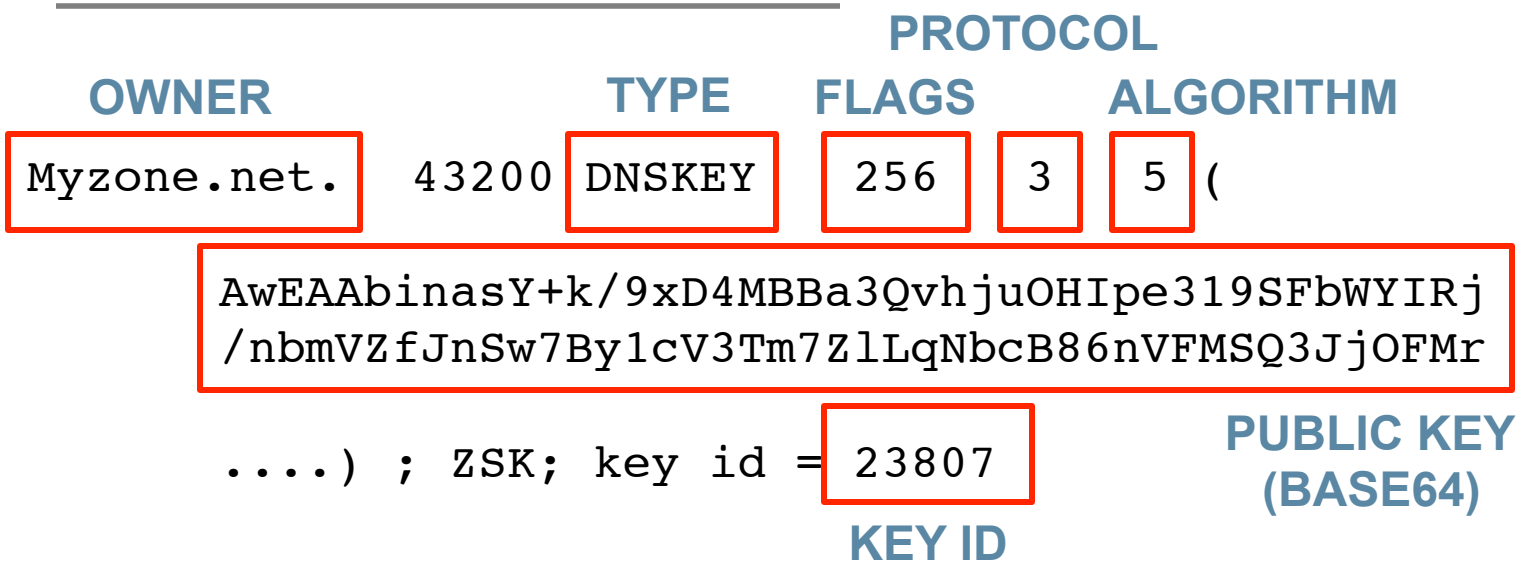


New Resource Records

New RRs

- Adds five new DNS Resource Records:
  1. **DNSKEY**: Public key used in zone signing operations.
  2. **RRSIG**: RRset signature
  3. **NSEC** &
  4. **NSEC3**: Returned as verifiable evidence that the name and/or RR type does not exist
  5. **DS**: Delegation Signer. Contains the hash of the public key used to sign the key which itself will be used to sign the zone data. Follow DS RR's until a ”trusted” zone is reached (ideally the root).

New RR: DNSKEY



- FLAGS determines the usage of the key
- PROTOCOL is always 3 (DNSSEC)
- ALGORITHM can be (3: DSA/SHA-1, 5: RSA/SHA1, 8: RSA/SHA-256, 12: ECC-GOST)
  - http://www.iana.org/assignments/dns-sec-alg-numbers/dns-sec-alg-numbers.xml

DNSKEY: Two Keys, not one…

- There are in practice at least **two** DNSKEY pairs for every zone
- Originally, **one** key-pair (public, private) defined for the zone
  - **private**: key used to sign the zone data (RRsets)
  - **public**: key published (DNSKEY) in the zone
- DNSSEC works fine with a single key pair
- Problem with using a single key:
  - Every time the key is updated, the DS record must be updated on the parent zone as well
  - Introduction of **K**ey **S**igning **K**ey (flags=257)

KSK and ZSK

- Key Signing Key (KSK)
  - Pointed to by parent zone in the form of DS (Delegation Signer). Also called Secure Entry Point.
  - Used to sign the Zone Signing Key
  - Flags: 257
- Zone Signing Key (ZSK)
  - Signed by the KSK
  - Used to sign the zone data RRsets
  - Flags: 256
- This decoupling allows for independent updating of the ZSK without having to update the KSK, and involve the parents (i.e. less administrative interaction)

New RR: RRSIG (Resource Record Signature)

```
Myzone.net.    600    A      192.168.10.10
Myzone.net.    600    A      192.168.23.45
```

TYPE COVERED #LABELS

OWNER TYPE ALG TTL

```
myzone.net.    600    RRSIG    A    7    2    600    (
```

SIG. EXPIRATION SIG. INCEPTION KEY ID SIGNER NAME

```
20150115154303    20141017154303    23807    myzone.net.
```

SIGNATURE

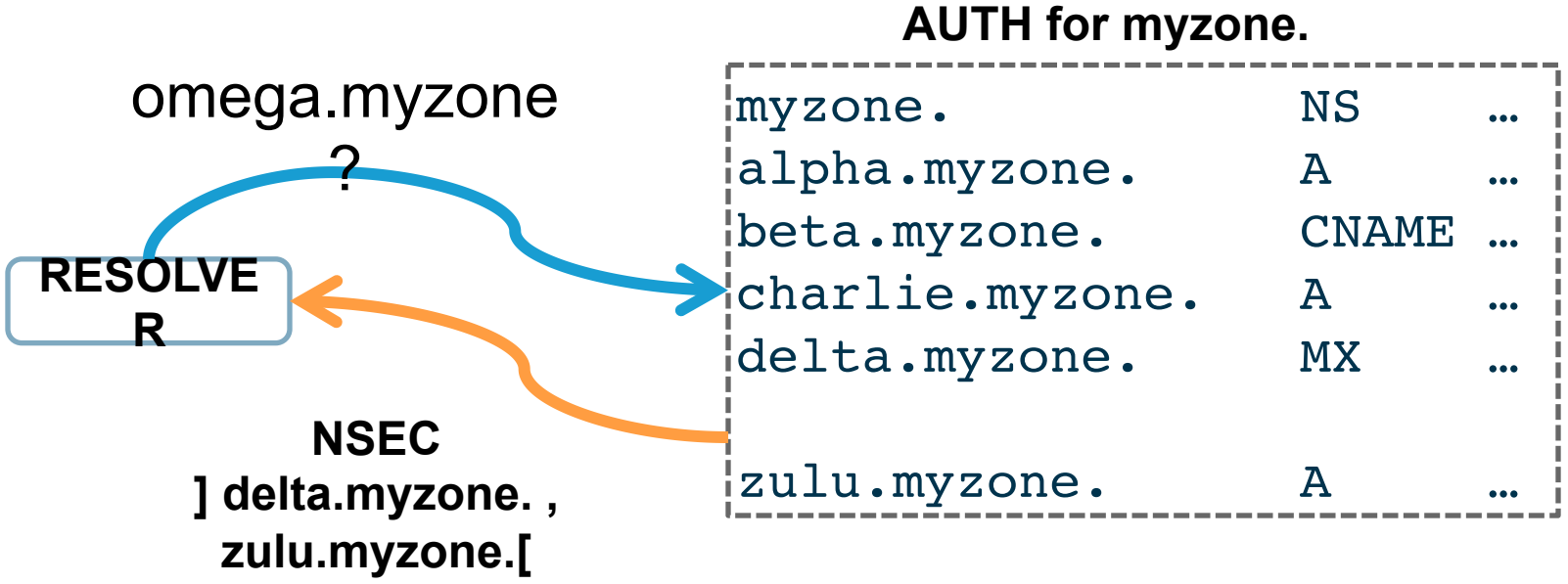
```
CoYkYPqE8Jv6UaVJgRrh7u16m/cEFGtFM8TArbJdaiPu
W77wZhrvonoBEyqYbhQ1yDaS74u9whECEe08gfoe1FGg
. . .
)
```

RRSIG

- Typical default values
  - Signature expiration is 30 from now
  - Proper timekeeping (NTP) is required
- What happens when signatures run out?
  - SERVFAIL
  - Domain effectively disappears from the Internet for validating resolvers
- Note that *keys* do **not** expire
- Not all RRSets need to be resigned at the same time

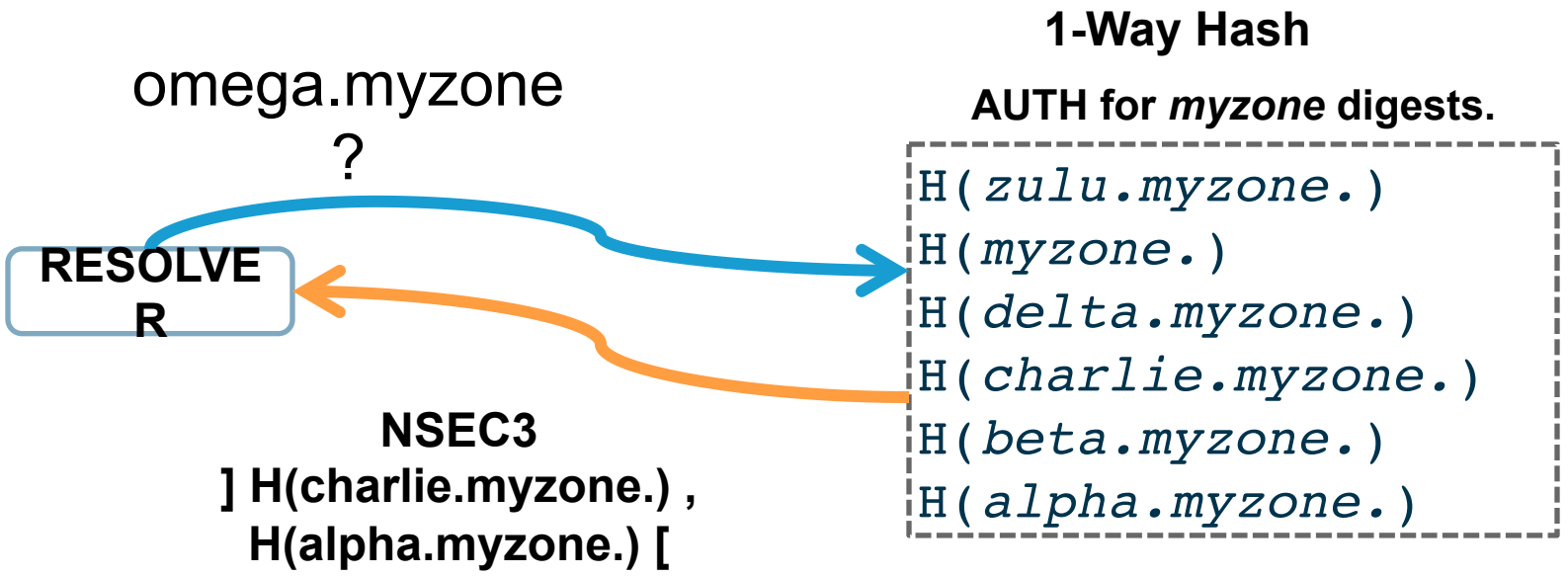
New RR: NSEC

- NXDomains also must be verified
- NSEC provides a pointer to the **N**ext **SEC**ure record in the chain of records.



New RR: NSEC3

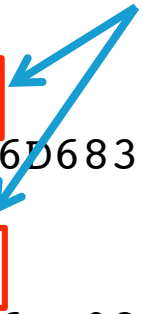
- To avoid concerns about “zone enumeration”
- To avoid large zone-files: opt-out concept



New RR: DS (Delegation Signer)

- Hash of the KSK of the child zone
- Stored in the parent zone, together with the NS RRs indicating a delegation of the child zone.
- The DS record for the child zone is signed together with the rest of the parent zone data
- NS records are NOT signed (they are a hint/ pointer)

Digest type 1 = SHA-1, 2 = SHA-256



```
myzone.       DS     61138   5     1
F6CD025B3F5D030408950535 4A0115584B56D683

myzone.       DS     61138   5     2
CCBC0B557510E4256E88C01B0B1336AC4ED6FE08C8268CC1AA
5FBF00 5DCE3210
```

DNSSEC - Setting up a Secure Zone

- Enable DNSSEC in the configuration file (named.conf)

```
dnssec-enable yes;

dnssec-validation yes;
```

- Create key pairs (KSK and ZSK)

```
dnssec-keygen -a rsasha1 -b 1024 -n zone myzone.net

dnssec-keygen -a rsasha1 -b 1400 -f KSK -n zone myzone.net
```

- Publish your public key

```
$INCLUDE /path/Kmyzone.net.+005+33633.key ; ZSK

$INCLUDE /path/Kmyzone.net.+005+00478.key ; KSK
```

- Signing the zone
- Update the config file
  - Modify the zone statement, replace with the signed zone file
- Test with dig

ICANN

Signing the Zone

- Sign the zone using the secret keys:

```
dnssec-signzone –o myzone.net -t -k
Kmyzone.net.+005+00478 db.myzone.net
Kmyzone.net.+005+33633
```

- Once you sign the zone a file with a .signed extension will be created
  - `db.myzone.net.signed`

ICANN

Testing with dig: an example

`dig @localhost www.myzone.net +dnssec +multiline`

Terminal — bash — 144×46

```
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.
                                Eyp1IVyQyYBLKOX2u/LT1+40xjBomXzLrcdwSErgioMb
                                pGyDWDLzP+FTbE3QCfBMLNDt2AGoYcty1cfY4li9sHkw
                                fue6hTQTSm0LhisBkVKQBy6ZD5oGiJQgaIkBGmLtVkPh
                                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+LClsJsEP/N
                                QvUDgtWrj8+ZOwVVj8aLe+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.
                                eTPO5c4GscnoC9V5sR6vgDo02WgCr1T5arU7YZhWctXI
                                vkmU1ni+wguwqW6xezfB/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
```

ICANN


Questions?

Signatures expiration and Key Rollovers

Signature expiration

- Signatures are per default 30 days (BIND)
- Need for regular resigning:
  - To maintain a constant window of validity for the signatures of the existing RRset
  - To sign new and updated RRsets
  - Use of jitter to avoid having to resign all expiring RRsets at the same time
- The keys themselves do NOT expire…
- But they may need to be rolled over...

Key Rollovers

- Try to minimise impact
  - Short validity of signatures
  - Regular key rollover
- Remember: DNSKEYs do not have timestamps
  - the RRSIG over the DNSKEY has the timestamp
- Key rollover involves second party or parties:
  - State to be maintained during rollover
  - Operationally expensive

Key Rollovers

- Two methods for doing key rollover
  - Pre-Publish
  - Double Signature
- KSK and ZSK rollover use different methods.
  - Remember that KSK needs to interact with parent zone to update DS record.

Key Rollovers: Pre-Publish method

- ZSK Rollover using the pre-publish method
  1. The zone contains only those key sets with which the zone has currently been signed
  2. New DNSKEY. A new key is created and published in the zone, but the zone is not signed with the new key until the pre-roll phase is complete.
  3. New RRSIGs. The zone is signed with the new DNS key. The old DNS key is not removed from the zone and remains published until the RRSIGs that were generated by the old key expire.
  4. DNSKEY Removal. When the RRSIGs that were generated by the old DNS key expire, the old DNS key is removed from the zone.

Key Rollovers: Double Signature

- KSK Rollover using the Double Signature method

1. The zone contains only those key sets with which the zone has currently been signed.
2. New DNSKEY. The new key is published in the zone and the zone is signed with the new key.

   - The zone contains the RRSIGs that are generated by the old and the new keys.

   - The minimum duration for which the zone must contain both sets of RRSIGs is the time required for all the RRSIGs to expire.

3. DNSKEY Removal. When the RRSIGs that were generated by the old DNS key expire, the old DNS key is removed from the zone.

Tools to help the process

Tools to use in DNSSEC

- Authoritative Servers that support DNSSEC
  - NSD (by NLNetLabs)
  - Knot (by CZ NIC Labs)
  - BIND (by ISC)
  - Vantio (by Nominum)
  - Y:A:D::I::F:A (by EURid)
  - MS DNS Server (by Microsoft)
  - TinyDNSSEC (based on tinydns by D.J. Bernstein)

Tools to use in DNSSEC

- Resolvers that support DNSSEC
  - Unbound (by NLNetLabs)
  - BIND (by ISC)
  - MS Windows Server (by Microsoft)
- Tools to automate DNSSEC
  - OpenDNSSEC (by NLnetLabs, .SE, Nominet…et al)
  - DNSSEC-Tools (by Sparta)
  - BIND (by ISC)


Questions?