DNS Security

Anoop Kumar Pandey
Principal Technical Officer
Centre for Development of Advanced Computing (C-DAC)
Electronics City, Bangalore 560 100

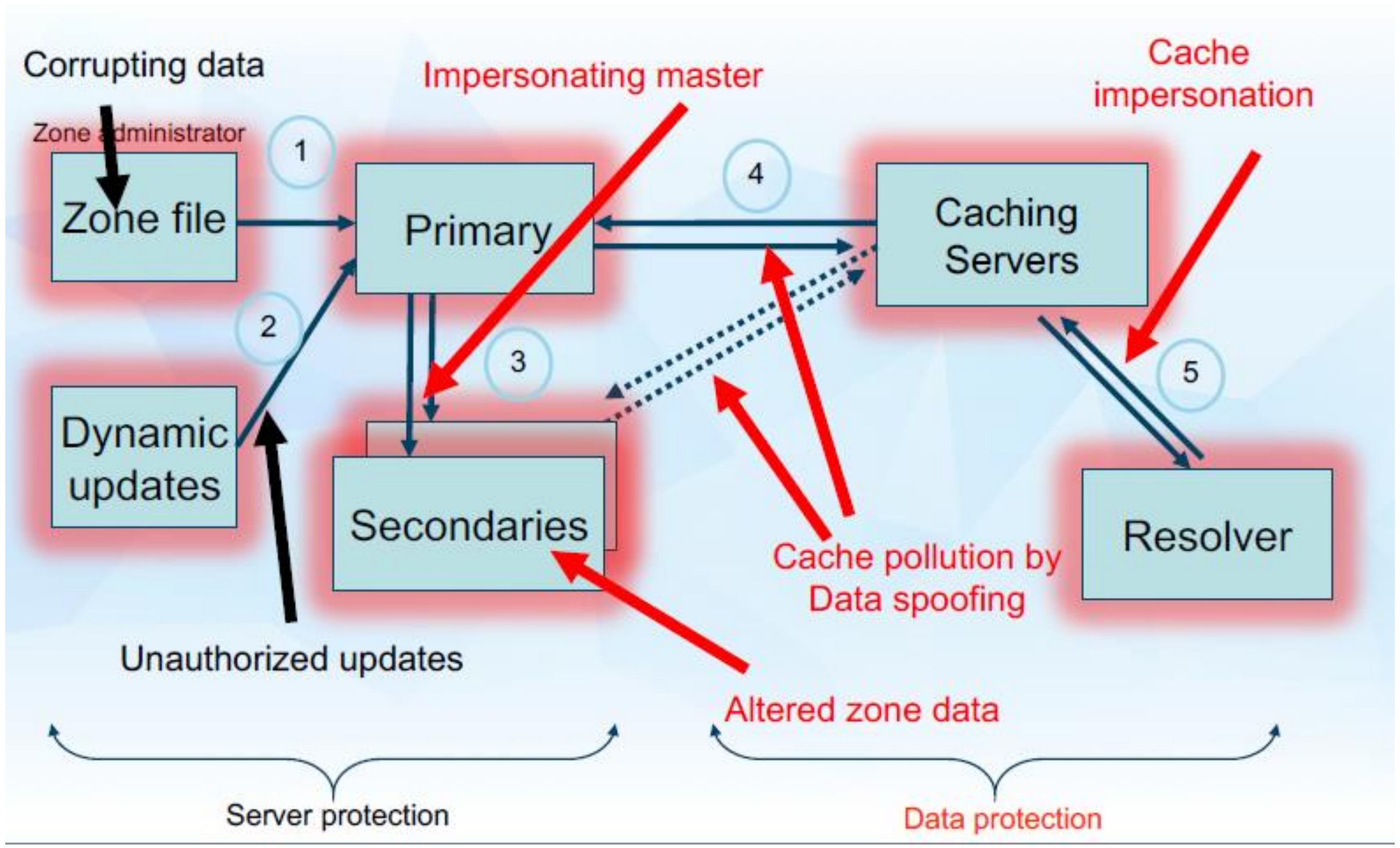
Centre of Excellence in DNS Security
22nd January 2020

/iiref coednssecurity.in @IIRnEF

Agenda

- DNSSEC
  - Need
  - Introduction
  - New Record Types
- Working of DNSSEC
- Implementation of DNSSEC

Need for DNSSEC



Centre of Excellence in DNS SECURITY

DNSSEC

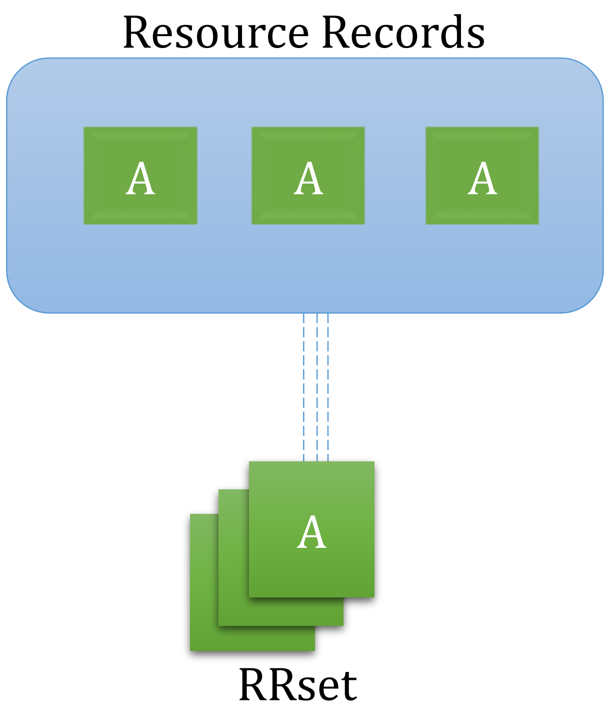
- Adds a layer of Trust through authentication
  - Adds cryptographic signature to existing DNS Records
- Verify Signature
  - Data coming from authoritative server
  - Ensures
    - No modification en-route
    - No fake record injection
- Hierarchical Trust Model

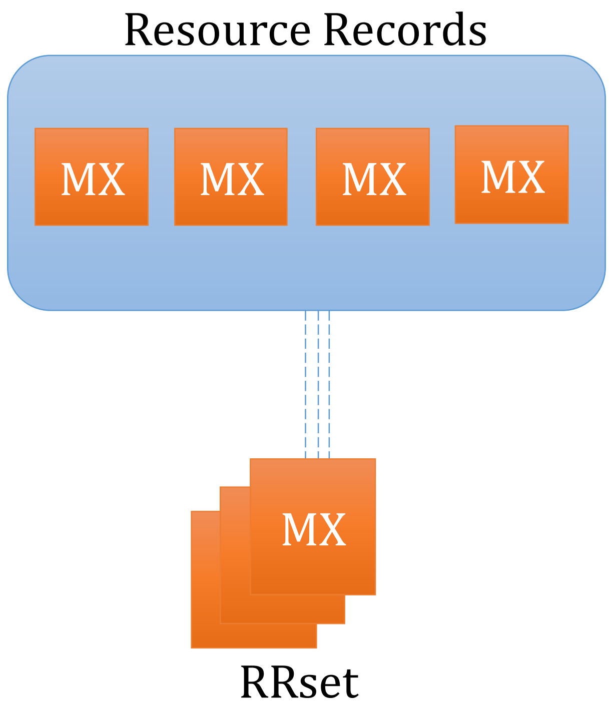
New Record Types

- **RRSIG** - Contains a cryptographic signature
- **DNSKEY** - Contains a public signing key
- **DS** - Contains the hash of a DNSKEY record
- **NSEC** and **NSEC3** - For explicit denial-of-existence of a DNS record
- **CDNSKEY** and **CDS** - For a child zone requesting updates to DS record(s) in the parent zone.

RRset

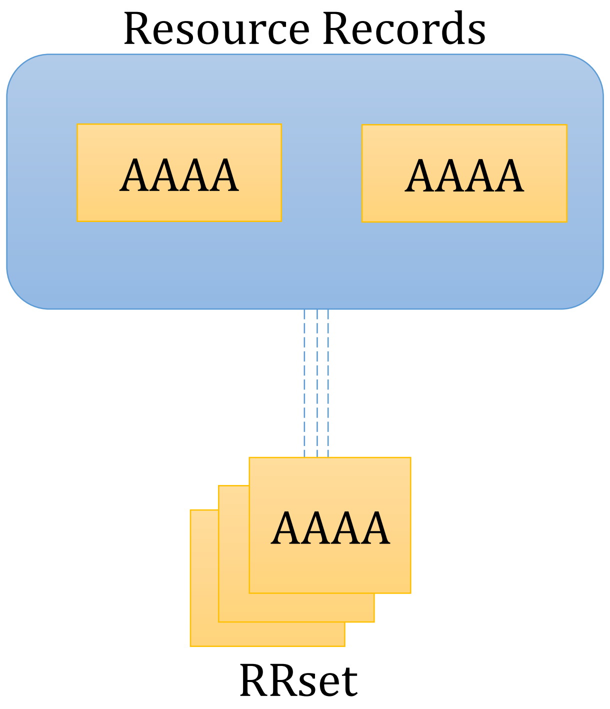
- All the records with the same type (A, AAAA etc) on the same label (i.e. www.microsoft.com)





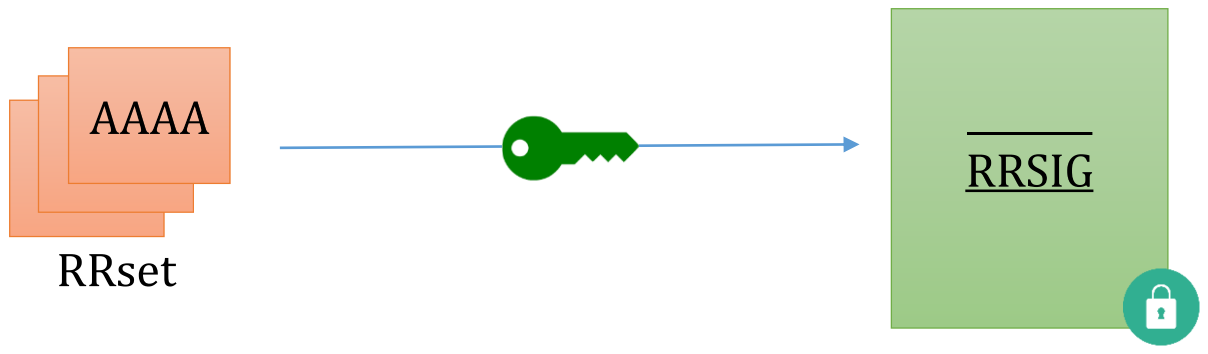
RRset

- Full RRset is signed
  - Not one record
- Request and validate entire RRset for one label
  - Validating one won't suffice

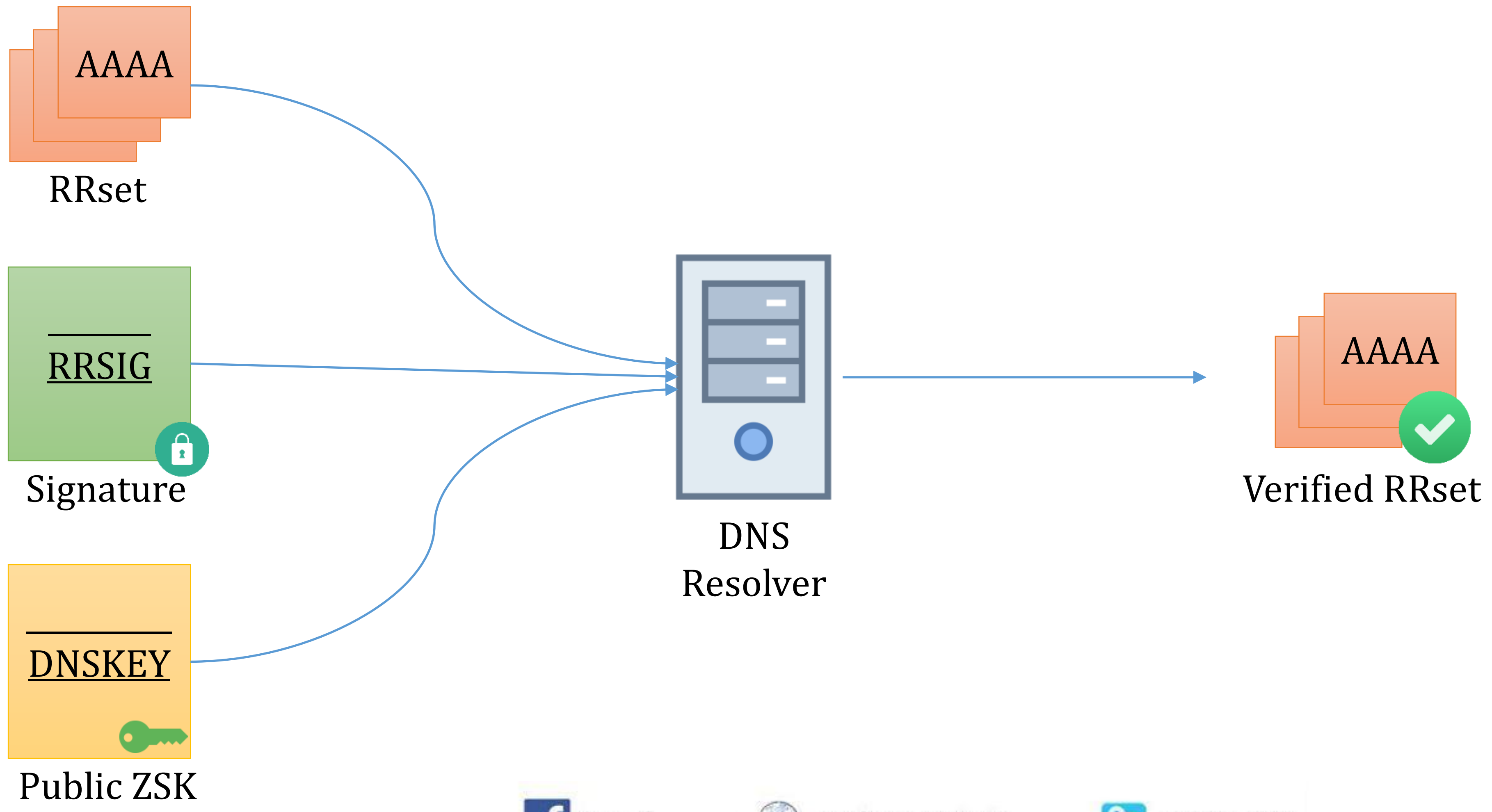


Zone Signing Key (ZSK)

- Key Pair
  - Private key to sign RRset
  - Public key to verify the signature



Zone Signing Key (ZSK)



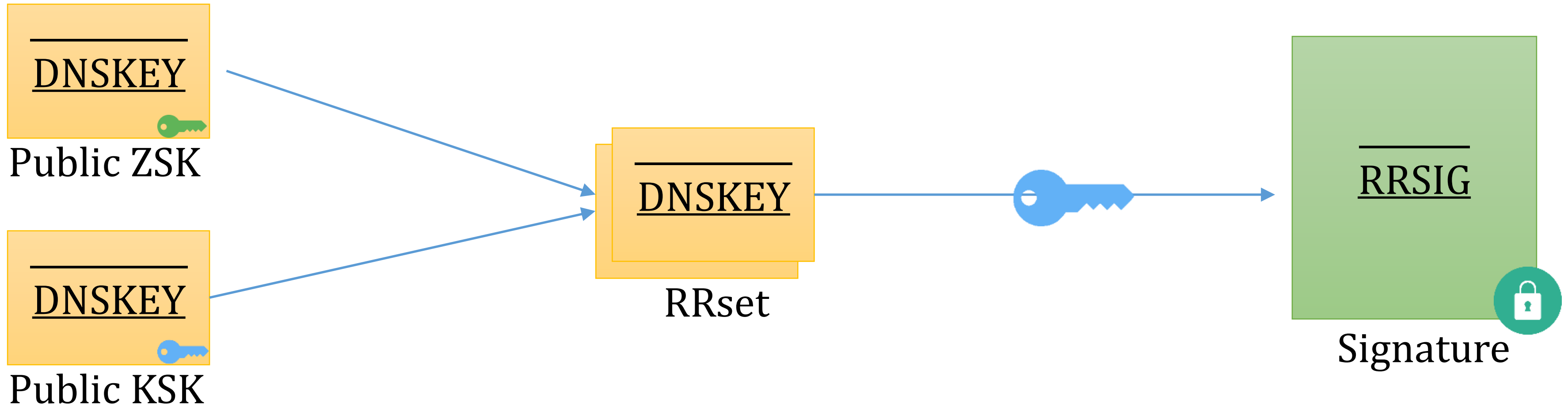
Centre of Excellence in DNS SECURITY

Zone Signing Key (ZSK)

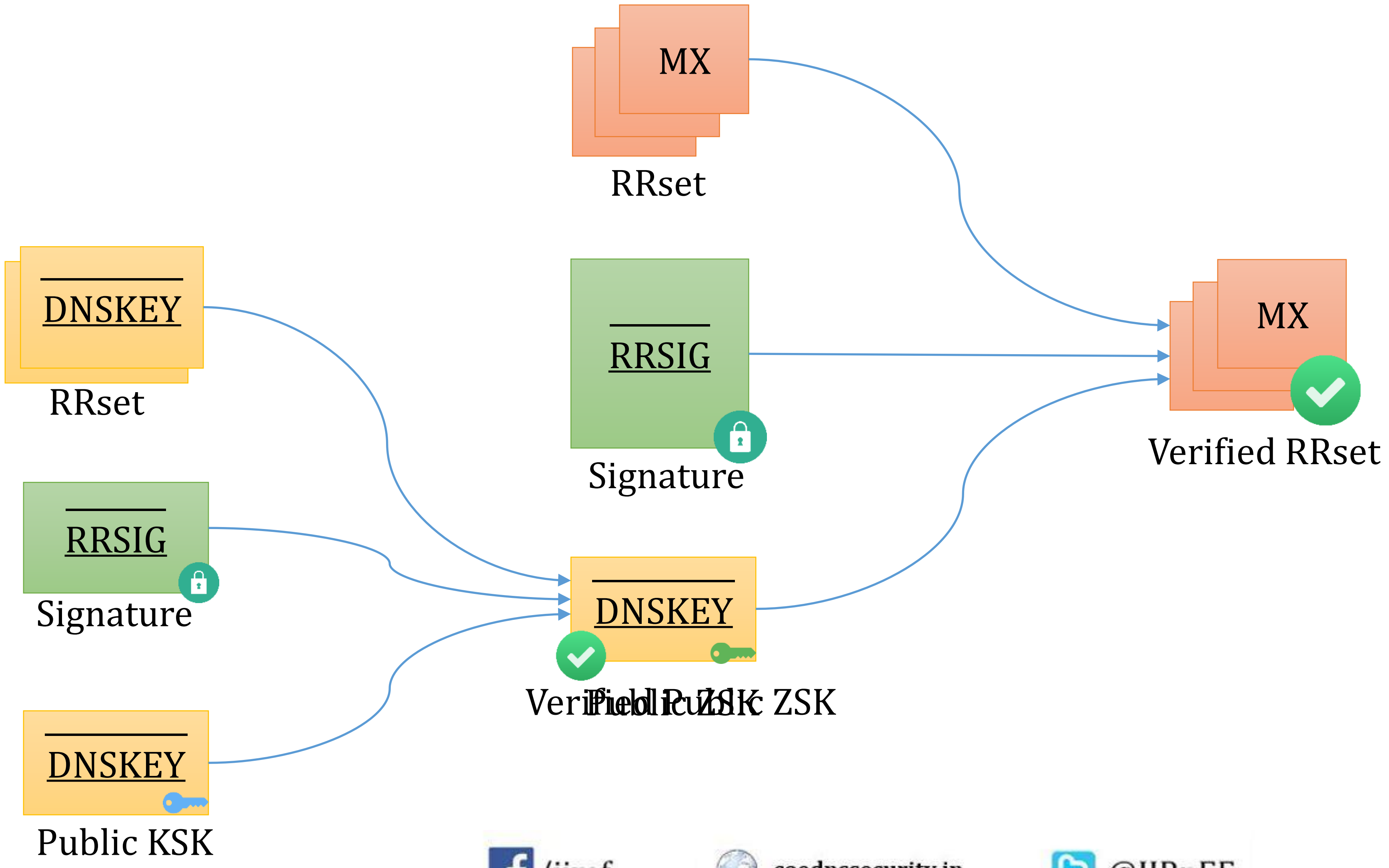
- ZSK can validate records
- But what if ZSK is compromised?

Key Signing Key (KSK)

- Signs public ZSK (stored in DNSKEY)
    - Creates RRSIG for DNSKEY
- Public KSK is stored in another DNSKEY record
- Both Public ZSK and KSK are signed using private KSK
- Resolvers use public KSK to validate public ZSK



Record Verification

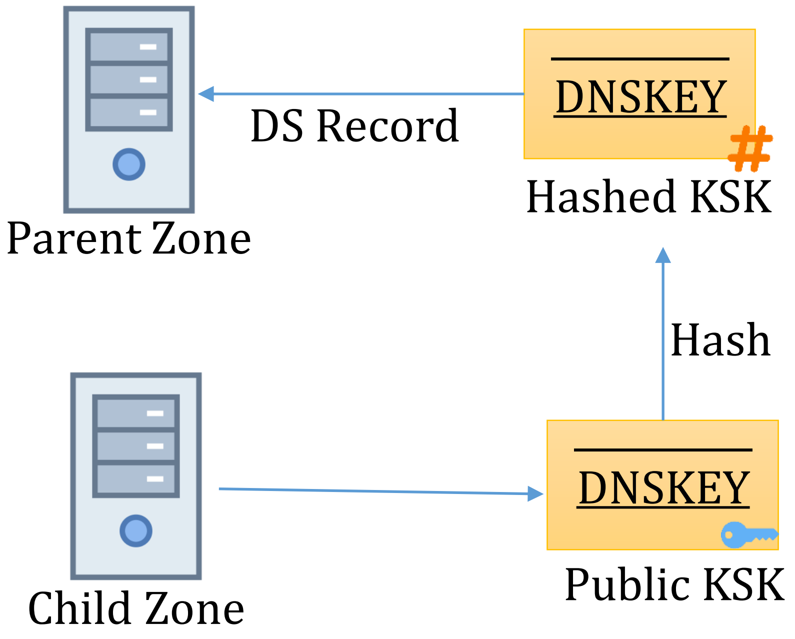


DNS Hierarchy

- Trust established in self zone
- But KSK is self-signed (Trust ends there itself)
- Trust should be propagated to parent zone for full trust

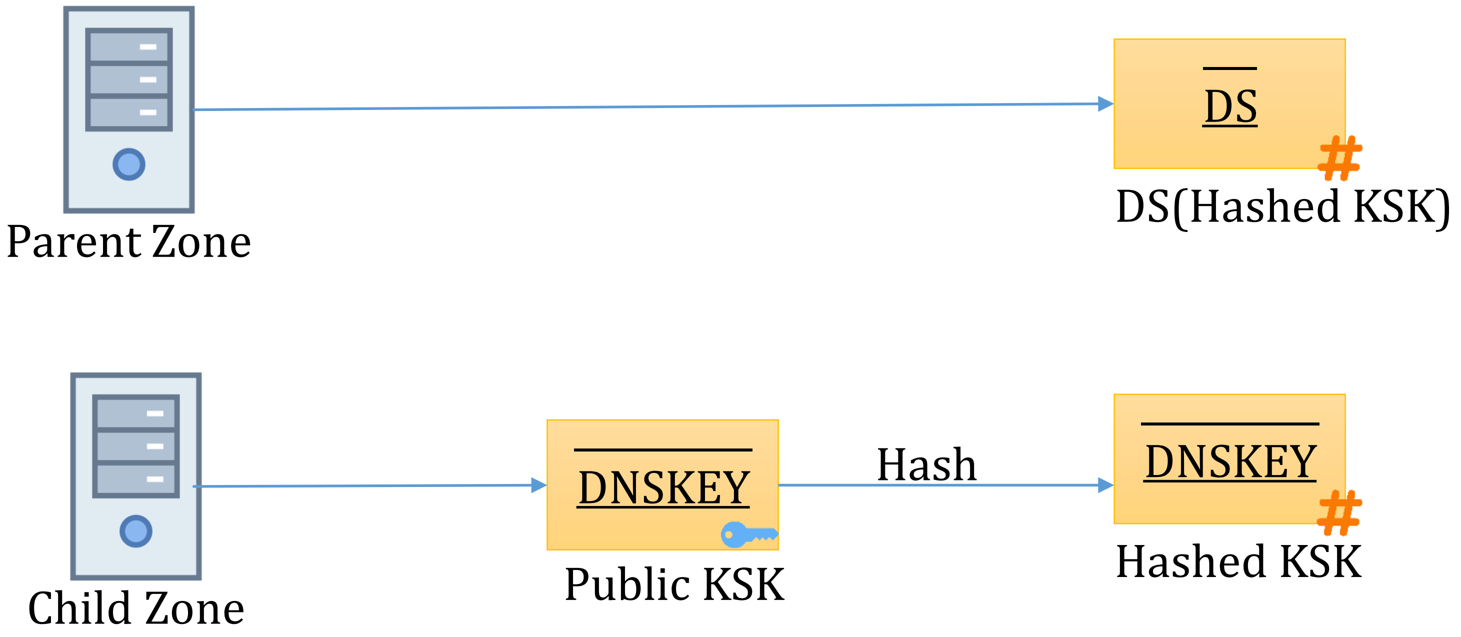
Delegation Signer Records

- Hash of DNSKEY record containing Public KSK
- Published as a DS record in Parent Zone
- DS Record establishes that child zone is DNSSEC enabled.

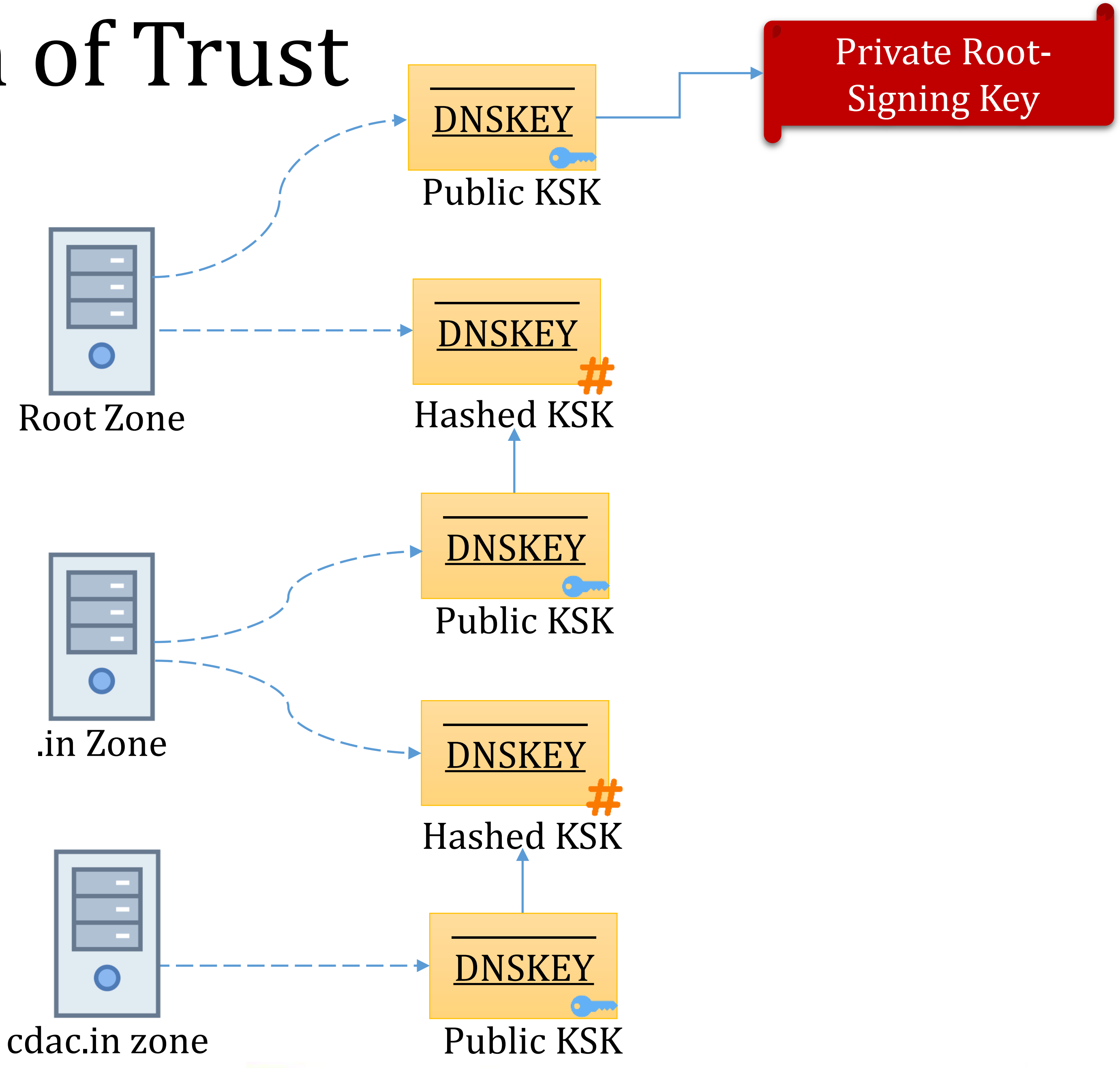


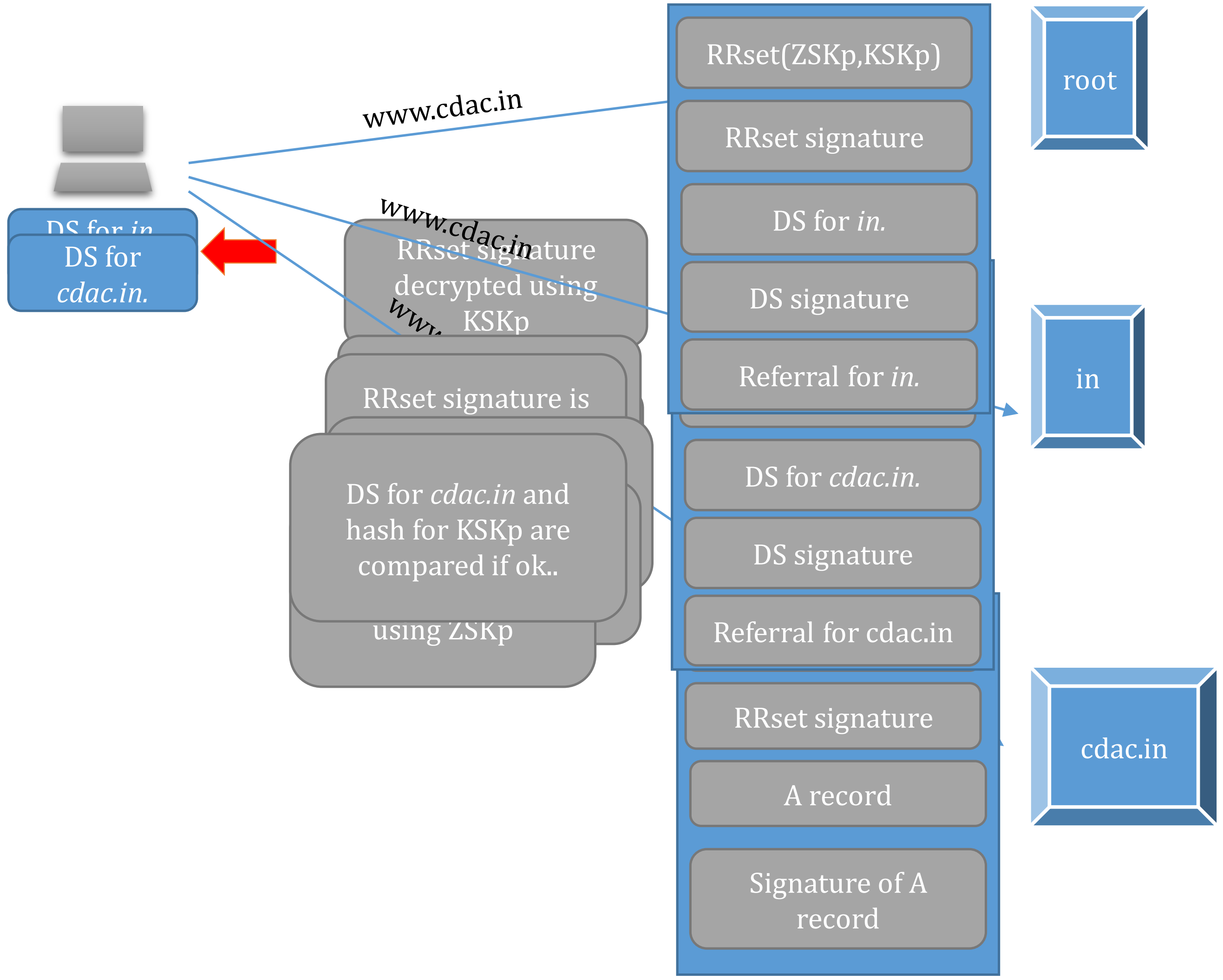
Delegation Signer Records

- Verification of Child Zone KSK



Chain of Trust




RRset(ZSKp,KSKp)
RRset signature
DS for *in.*
DS signature
Referral for *in.*
DS for *cdac.in.*
DS signature
Referral for cdac.in
RRset signature
A record
Signature of A record
root
in
cdac.in
www.cdac.in
www.cdac.in
DS for *in.*
DS for *cdac.in.*
RRset signature decrypted using KSKp
RRset signature is
DS for *cdac.in* and hash for KSKp are compared if ok..
using ZSKp

NSEC/NSEC3 Record

- For non-existent Domain
  - Empty answer
    - What to sign??
- Explicit authenticated denial of existence of a record
- NSEC (Next Secure) Record
  - Returns the "next secure" record for a non-existent domain (alphabetical order)
    - 'A' record for *auth, drive, mail, www*
    - Request for 'A' record for *smtp* → NSEC Record containing *www* will be returned
  - Record can be verified
  - Prone to zone walking
    - Security threat if some sub-domains need to be kept private
- NSEC3 Record
  - Uses cryptographically hashed record names to avoid the enumeration of the record names in a zone.

Summary

- DNSSEC Guarantees:
  - Authenticity of DNS answer origin
  - Integrity of reply
  - Authenticity of denial of existence
- Accomplishes this by signing DNS replies at each step in the hierarchy
- Uses public-key cryptography to sign responses
- Typically use trust anchors, entries in the OS to bootstrap the process

Summary

- DNSSEC does not
  - Provide confidentiality for DNS data
  - Protect against Denial of Data

Implementation

Create keys folder in /var/named and generate keys in that folder

- dnssec-keygen -a ECDSAP256SHA256 -3 anoop.in
- dnssec-keygen -f KSK -a ECDSAP256SHA256 -3 anoop.in
- chmod 777 /var/named/keys
- chgrp named /var/named/keys
- nano /etc/named.conf

```
zone "anoop.in" IN {
        type master;
        file "anoop";
        key-directory "/var/named/keys/";
        auto-dnssec maintain;
        inline-signing yes;
};
```

Implementation

- systemctl restart named
- systemctl status named
- rndc loadkeys anoop.in
- openssl rand -hex 4
  2514ca31
- rndc signing -NSEC3PARAM 1 0 10 2514ca31 anoop.in
- dig @localhost dnskey anoop.in | dnssec-dsfromkey -f - anoop.in

Centre of Excellence in DNS SECURITY

Thank You

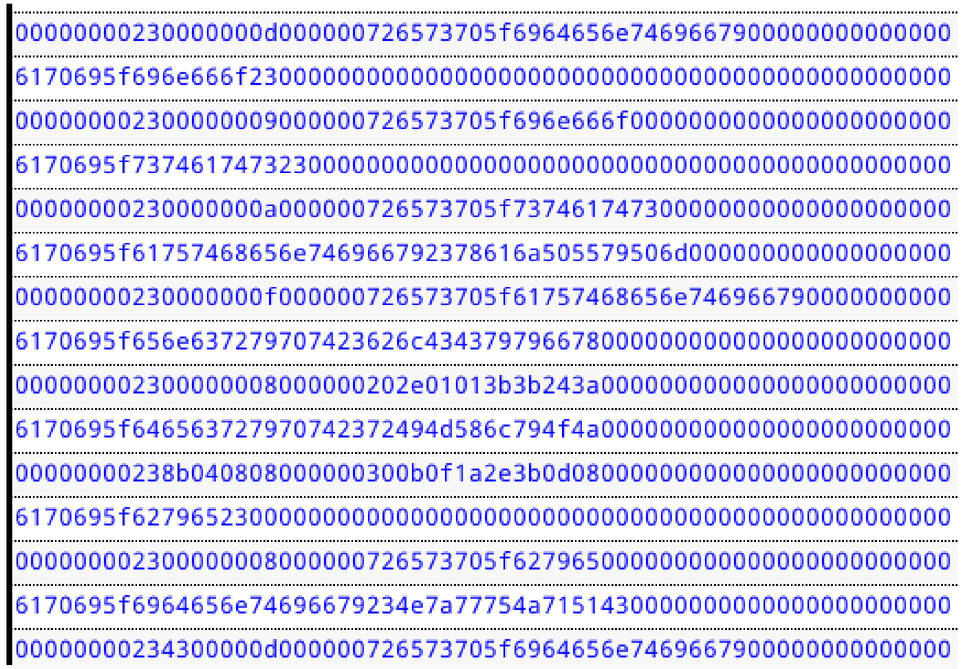
Centre of Excellence in DNS SECURITY

Digital Signatures

What is a Digital Signature ?

- A *Digital signature* of a message is a **number (fingerprint) dependent** on
  - a secret known only to the signer **and**
  - the content of the message being signed

- Digital Signatures can be
  - Verified for Authenticity
  - Verified for Integrity
  - Verified for Non-Repudiation



```
00000000230000000d000000726573705f6964656e7469667900000000000000
6170695f696e666f230000000000000000000000000000000000000000000000
000000002300000009000000726573705f696e666f0000000000000000000000
6170695f737461747323000000000000000000000000000000000000000000000
00000000230000000a000000726573705f737461747300000000000000000000
6170695f61757468656e74696679237861 6a505579506d0000000000000000
00000000230000000f000000726573705f61757468656e746966790000000000
6170695f656e637279707423626c434379796678000000000000000000000000
000000002300000008000000202e01013b3b243a000000000000000000000000
6170695f64656372797074237 2494d586c794f4a000000000000000000000000
00000000238b040808000000300b0f1a2e3b0d08000000000000000000000000
6170695f627965230000000000000000000000000000000000000000000000000
000000002300000008000000726573705f627965000000000000000000000000
6170695f6964656e74696679234e7a77754a7151430000000000000000000000
00000000234300000d000000726573705f6964656e7469667900000000000000
```

Creating Digital Signature

- Every individual is given a pair of **keys**
  - ***Public key*** : known to everyone
  - ***Private key*** : known only to the owner

- To ***digitally sign*** an electronic document the signer uses his/her ***Private key***
- To ***verify*** a digital signature the verifier uses the signer’s ***Public key***

What is a key pair?

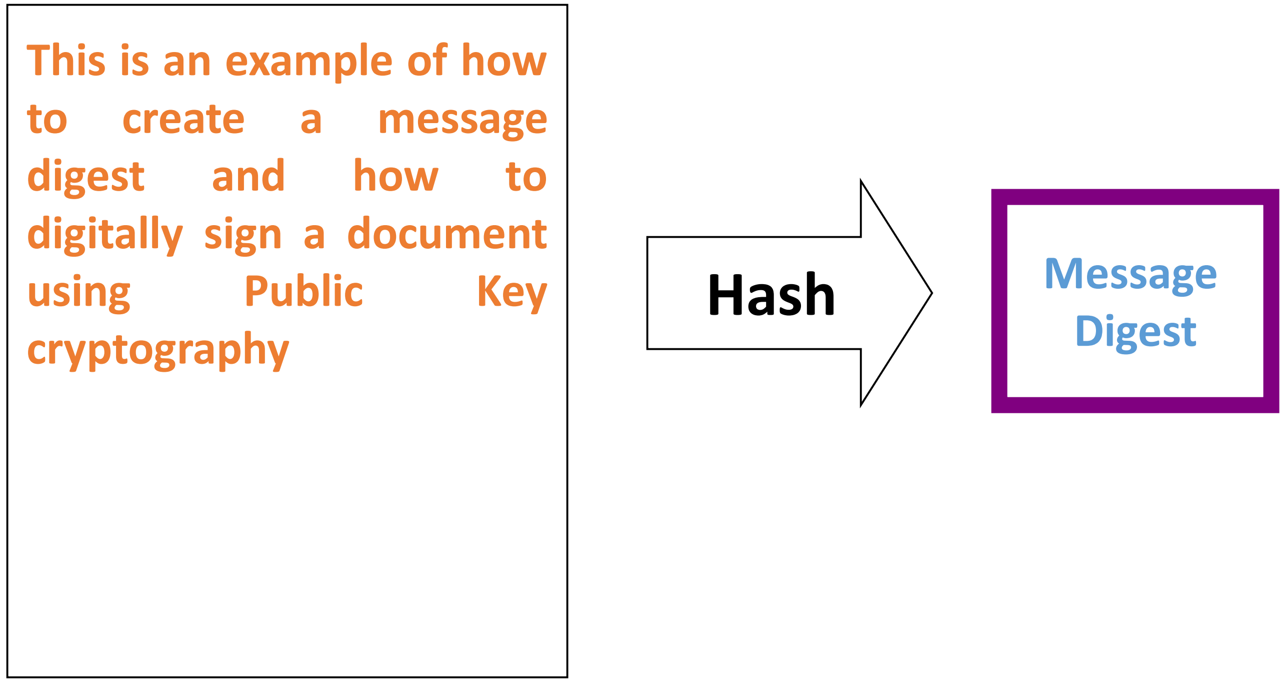
Private Key

```
3082 010a 0282 0101 00b1 d311 e079 5543 0708 4ccb 0542 00e2 0d83
463d e493 bab6 06d3 0d59 bd3e c1ce 4367 018a 21a8 efbc ccd0 a2cc
b055 9653 8466 0500 da44 4980 d854 0aa5 2586 94ed 6356 ff70 6ca3
a119 d278 be68 2a44 5e2f cfcc 185e 47bc 3ab1 463d 1ef0 b92c 345f
8c7c 4c08 299d 4055 eb3c 7d83 deb5 f0f7 8a83 0ea1 4cb4 3aa5 b35f
5a22 97ec 199b c105 68fd e6b7 a991 942c e478 4824 1a25 193a eb95
9c39 0a8a cf42 b2f0 1cd5 5ffb 6bed 6856 7b39 2c72 38b0 ee93 a9d3
7b77 3ceb 7103 a938 4a16 6c89 2aca da33 1379 c255 8ced 9cbb f2cb
5b10 f82e 6135 c629 4c2a d02a 63d1 6559 b4f8 cdf9 f400 84b6 5742
859d 32a8 f92a 54fb ff78 41bc bd71 28f4 bb90 bcff 9634 04e3 459e
a146 2840 8102 0301 0001
```

Public Key

```
3082 01e4 f267 0142 0f61 dd12 e089 5547 0f08 4ccb 0542 00e2 0d83 463d
e493 bab6 0673 0d59 bf3e c1ce 4367 012a 11a8 efbc ccd0 a2cc b055 9653
8466 0500 da44 4980 d8b4 0aa5 2586 94ed 6356 ff70 6ca3 a119 d278 be68
2a44 5e2f cfcc 185e 47bc 3ab1 463d 1df0 b92c 345f 8c7c 4c08 299d 4055
eb3c 7d83 deb5 f0f7 8a83 0ea1 4cb4 3aa5 b35f 5a22 97ec 199b c105 68fd
e6b7 a991 942c e478 4824 1a25 193a eb95 9c39 0a8a cf42 b250 1cd5 5ffb
6bed 6856 7b39 2c72 38b0 ee93 a9d3 7b77 3ceb 7103 a938 4a16 6c89 2aca
da33 1379 c255 8ced 9cbb f2cb 5b10 f82e 6135 c629 4c2a d02a 63d1 6559
b4f8 cdf9 f400 84b6 5742 859d 32a8 f92a 54fb ff78 41bc bd71 28f4 bb90
bcff 9634 04de 45de af46 2240 8410 02f1 0001
```

Digital Signing – Step 1


This is an example of how to create a message digest and how to digitally sign a document using Public Key cryptography
Hash
Message Digest

Hash Function

- A hash function is a cryptographic mechanism that operates as **one-way** function
  - Creates a digital representation or "fingerprint“ (Message Digest)
  - **Fixed size output**
  - Change to a message produces different digest

Examples : MD5 , Secure Hashing Algorithm (SHA)

Hash - Example

Message

Hi Jai,

I will be in the park at

3 pm

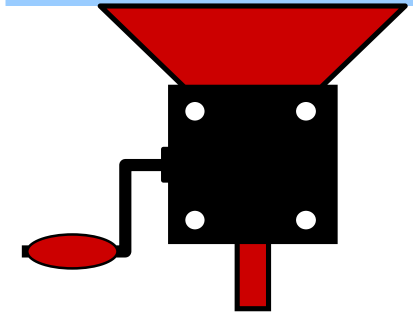
Veeru

Hi Jai,

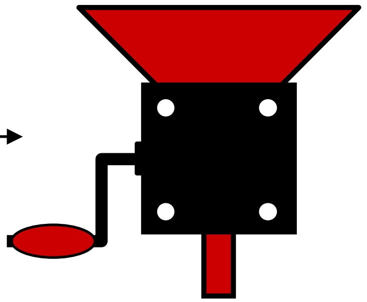
I will be in the park at

3 pm.

Veeru



← Hash Algorithm →



Message Digest

B5EA1EC376E61DB2680D0312FC26D3773F384E43

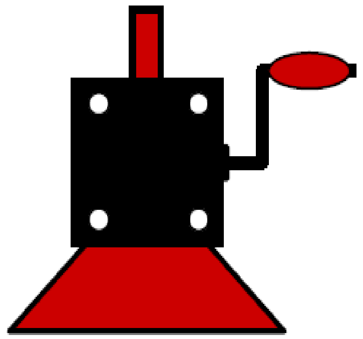
86D19C25294FB0D3E4CF8A02682343906459800
9

$\underbrace{\hspace{10cm}}$

Digests are Different

Hash – One-way

B5EA1EC376E61DB2680D0312FC26D3773F384E43



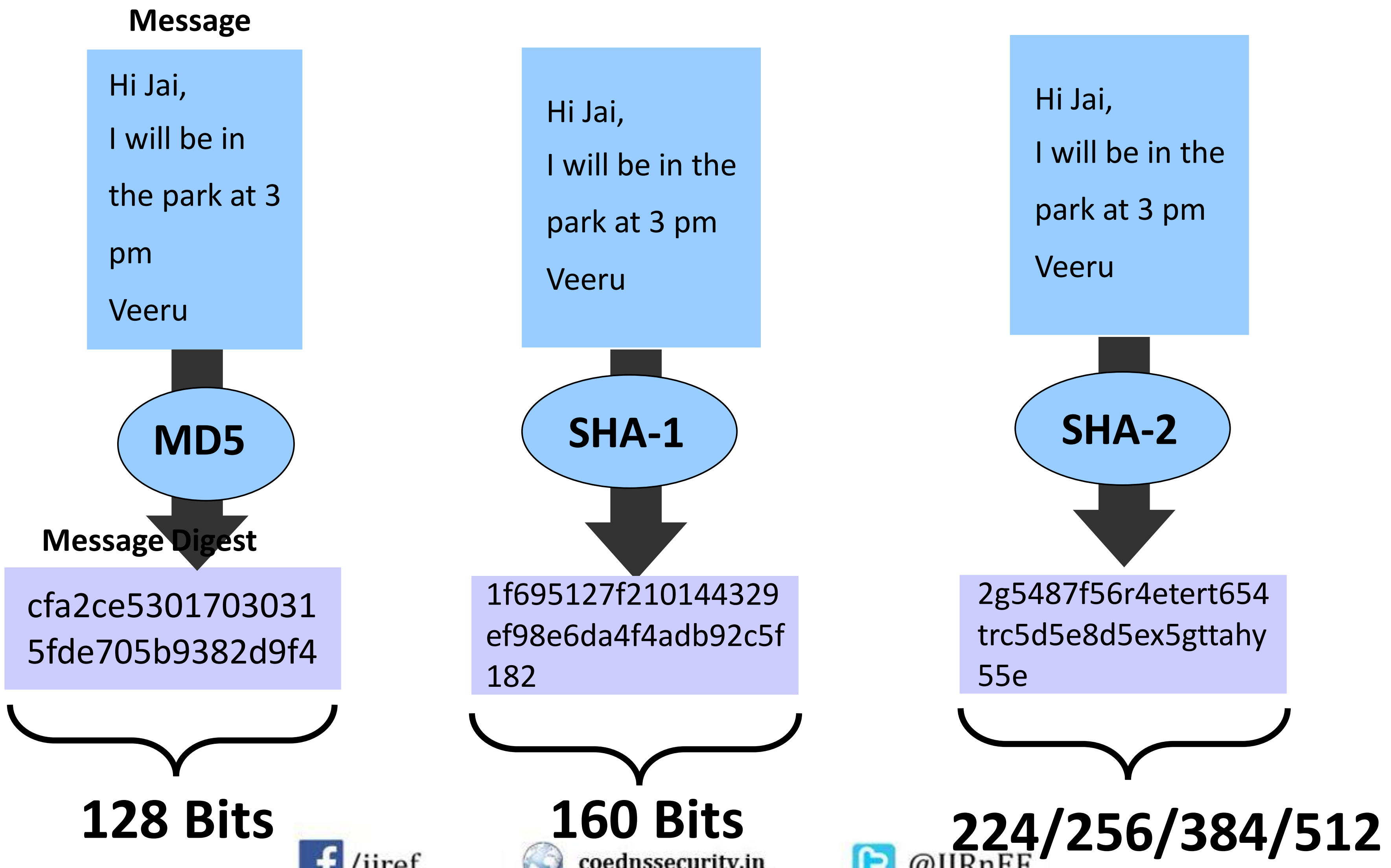
Hi Jai,

I will be in the park at

3 pm

Veeru

MD5 and SHA



Message

Hi Jai,
I will be in
the park at 3
pm
Veeru

MD5

Message Digest

cfa2ce5301703031
5fde705b9382d9f4

128 Bits

Hi Jai,
I will be in the
park at 3 pm
Veeru

SHA-1

1f695127f210144329
ef98e6da4f4adb92c5f
182

160 Bits

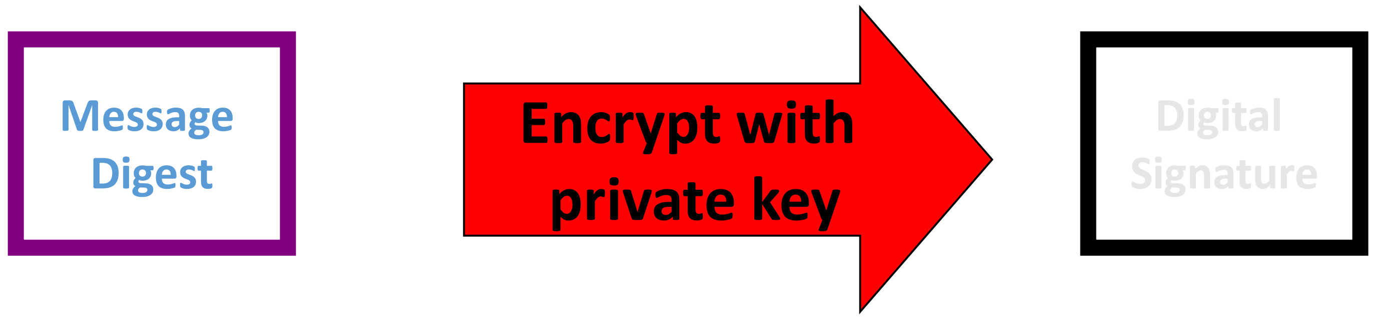
Hi Jai,
I will be in the
park at 3 pm
Veeru

SHA-2

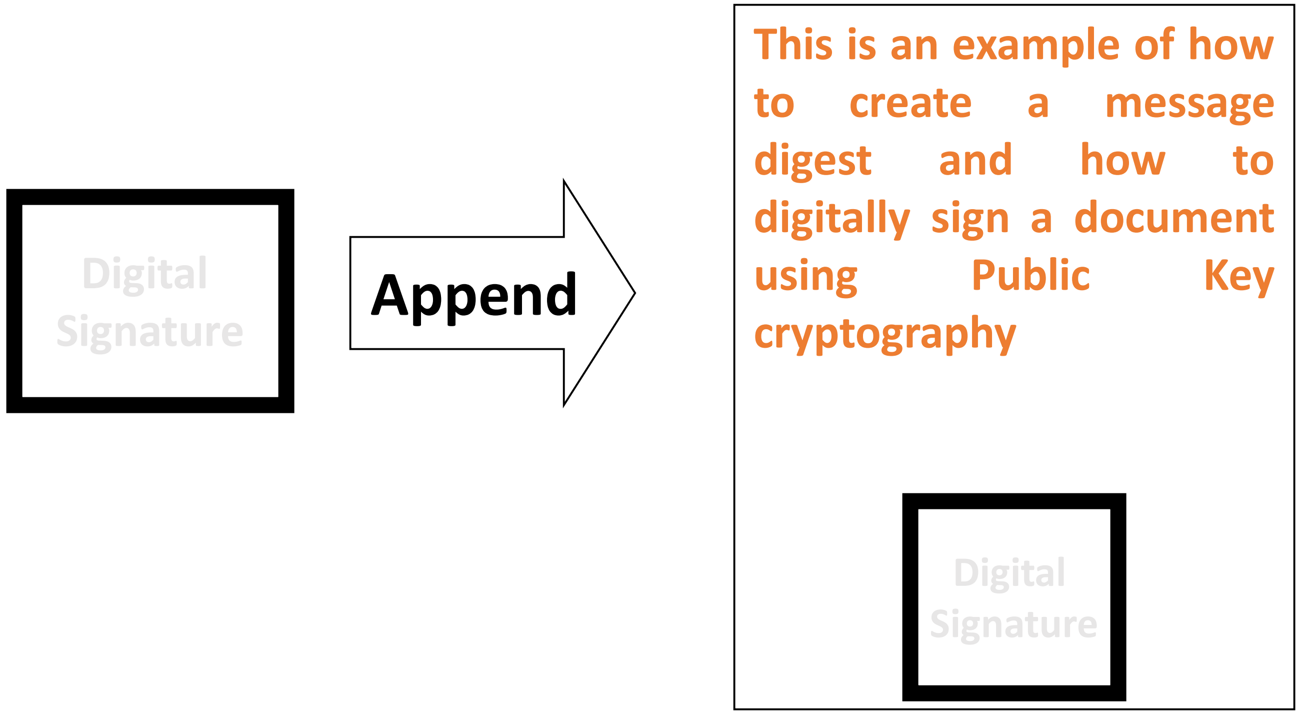
2g5487f56r4etert654
trc5d5e8d5ex5gttahy
55e

224/256/384/512

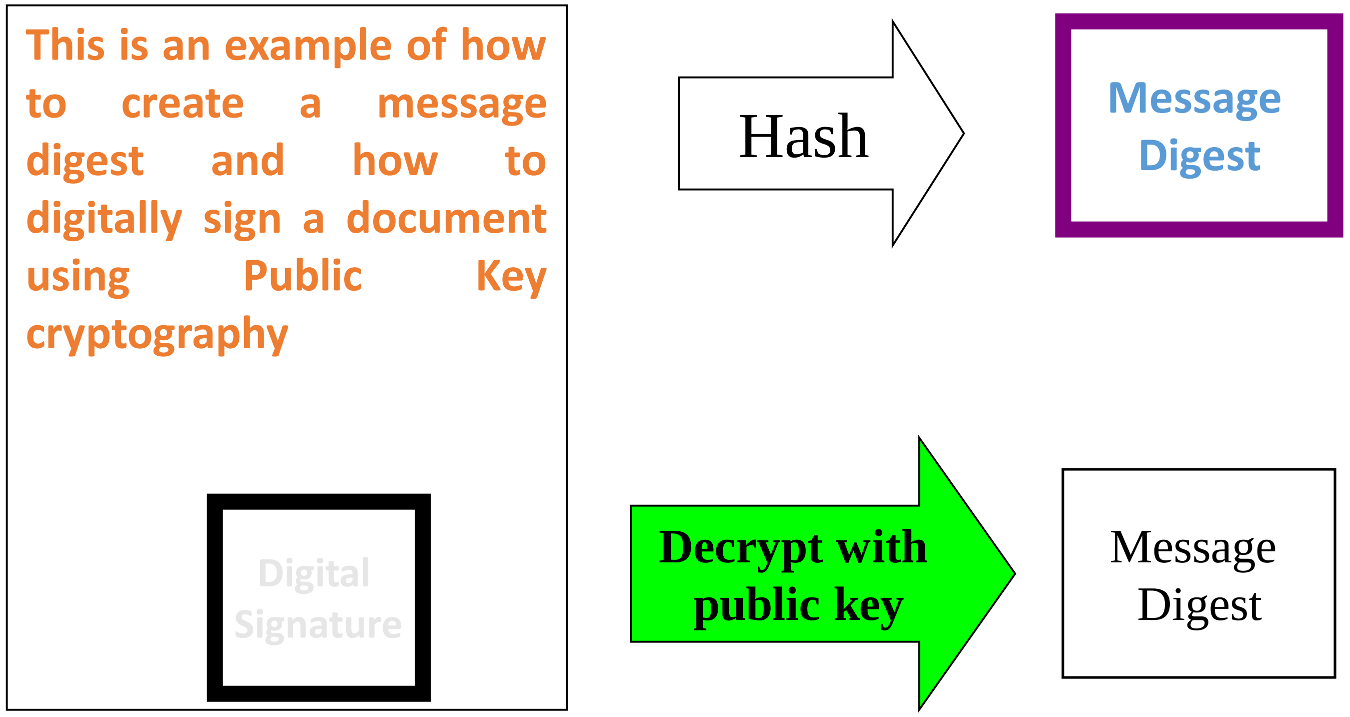
Digital Signing – Step 2



Digital Signing – Step 3


Digital Signature
Append
This is an example of how to create a message digest and how to digitally sign a document using Public Key cryptography
Digital Signature

Digital Signature Verification


This is an example of how to create a message digest and how to digitally sign a document using Public Key cryptography
Digital Signature
Hash
Message Digest
Decrypt with public key
Message Digest

DNSSEC Manual

1. **Key Generation: (/var/named)**

```
#dnssec-keygen -3 -a ECDSAP256SHA256 anoopmis.in.
#dnssec-keygen -f KSK -3 -a ECDSAP256SHA256 anoopmis.in.
```

2. **Permission Change:**

```
#chgrp -R named Kanoopmis.in*
#chmod -R 640 Kanoopmis.in*
```

3. **Manual Signing:**

3.1 Include keys location in zone file (key and zone file located at /var/named)

```
# vi /var/named/anoopmis
        $INCLUDE “Kanoopmis.in.+008+40578.key”;
        $INCLUDE “Kanoopmis.in.+008+40577.key”;
```

3.2 Sign the zone

```
#openssl rand -hex 4    (Generates a 32 bit salt represented in Hexadecimal notation)
#dnssec-signzone -A -3 47755c4e -N INCREMENT -o anoopmis.in -t anoopmis
```

3.3 Point to the new signed zone file

```
#vi /etc/named.conf
        zone "anoopmis.in" IN {
                type master;
                file "anoopmis.signed";
        };
```

3.4 Restart named service

```
#systemctl restart named
```

4. **Check for Delegation Signer:**

```
#dig @localhost dnskey anoopmis.in | dnssec-dsfromkey -f - anoopmis.in
```

5. **Check for DNSSEC:**

```
#dig www.anoopmis.in @localhost +dnssec
```

6. **Automatic Signing:**

6.1 Create a folder “keys” in “/var/named” and move the keys to “/var/named/keys”

```
#mkdir keys
#mv Kanoopmis* keys/
```

6.2 Change Permission of the folder “keys”

```
#chgrp -R named keys
#chmod -R 770 keys
```

6.3 Add path of keys in Configuration File

```
#vi /etc/named.conf
        zone "anoopmis.in" IN {
                type master;
                file "anoopmis";
                key-directory "/var/named/keys";
                auto-dnssec maintain;
                inline-signing yes;
        };
#systemctl restart named
```

6.4 DNS Zone Signing:

```
#rndc loadkeys anoopmis.in
#openssl rand -hex 4
#rndc signing -NSEC3PARAM 1 0 10 8721f7cd anoopmis.in
```