

# Public Key Infrastructure Components

**PKI Outreach Programme (POP)**  
Nationwide Awareness Programme,  
Centre for Development of Advanced Computing (C-DAC)  
Electronics City, Bangalore.

# Digital Signature

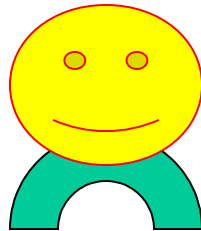
- A *digital signature* of a message is
  - a number dependent on some secret known only to the signer and
  - additionally on the content of the message being signed
- Signatures must be verifiable
- Applications
  - Authentication,
  - Data Integrity
  - Non-repudiation

# Digital Signature

- Key pairs of every individual
  - *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*

# Digital Signature

- Veeru has two keys



**Veeru**

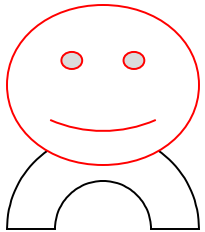


Public Key

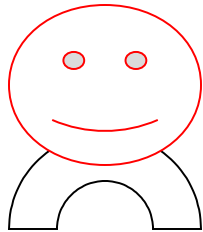


Private Key

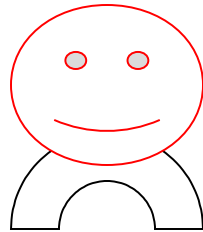
# Digital Signature



**Pat**



**Sam**



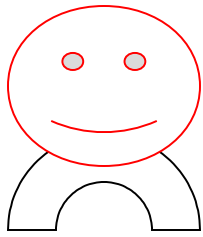
**Don**



Public Key

Anyone can get Veeru's Public Key,  
but Veeru keeps his Private Key to  
himself

# Digital Signature

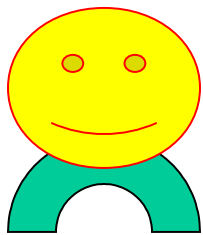


Pat

“Hey  
Veeru,  
shall we go  
for lunch”

Encrypt with  
Veeru's public key

“Xyexyene  
iorourjflree  
fewerdjdf”



Veeru

“Xyexyene  
iorourjflree  
fewerdjdf”

Decrypt with  
Veeru's private key

“Hey  
Veeru,  
shall we go  
for lunch”

# Digitally Signing

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

Hash

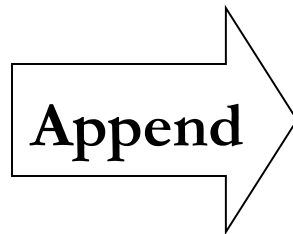
Message  
Digest

# Digitally Signing



# Digitally Signing

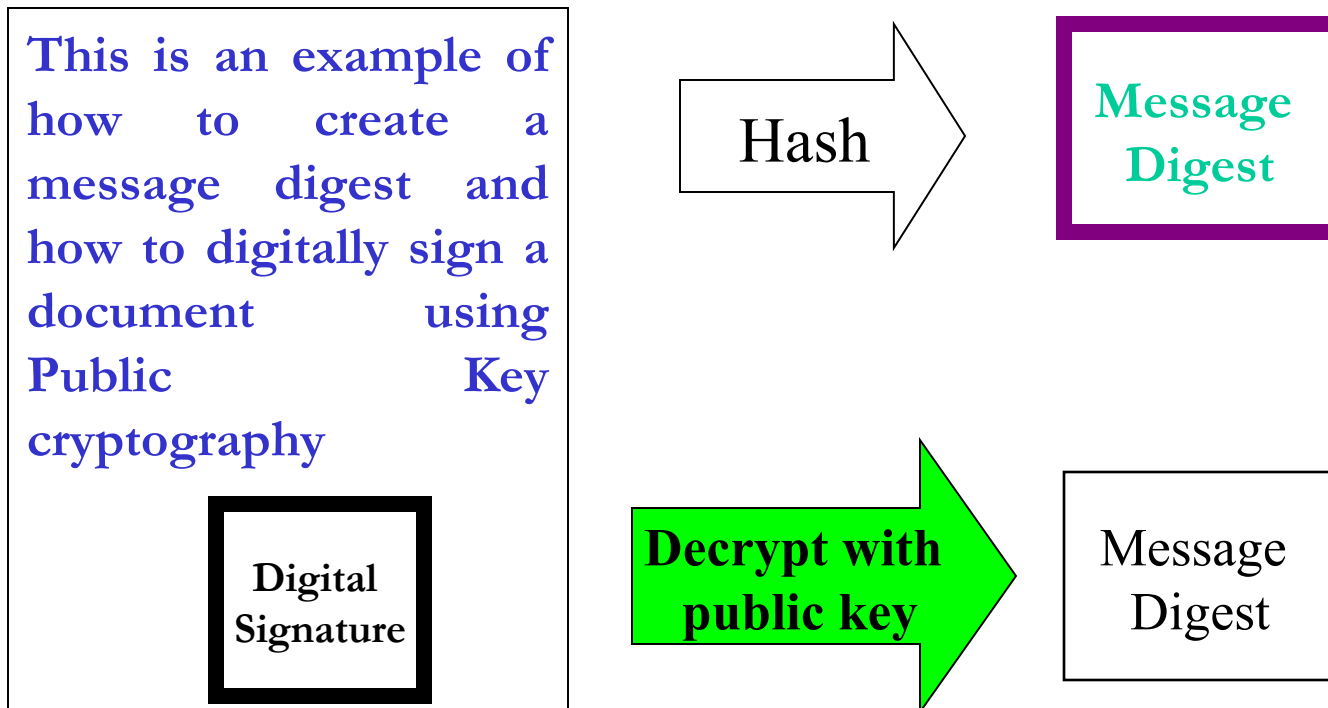
Digital  
Signature



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

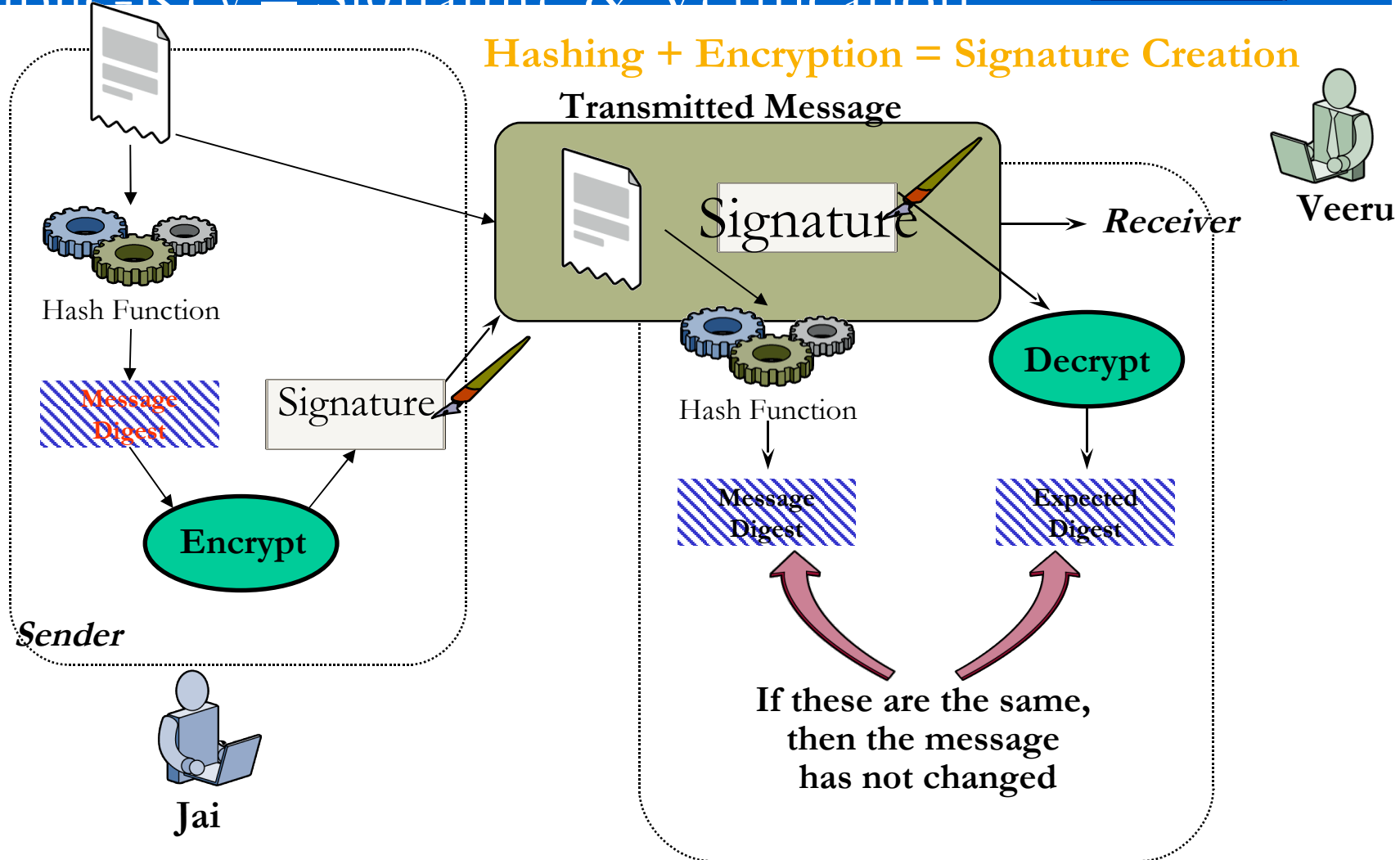
Digital  
Signature

# Digital Signature verification



# Public-Key - Signature & Verification

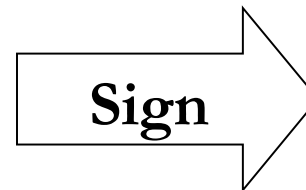
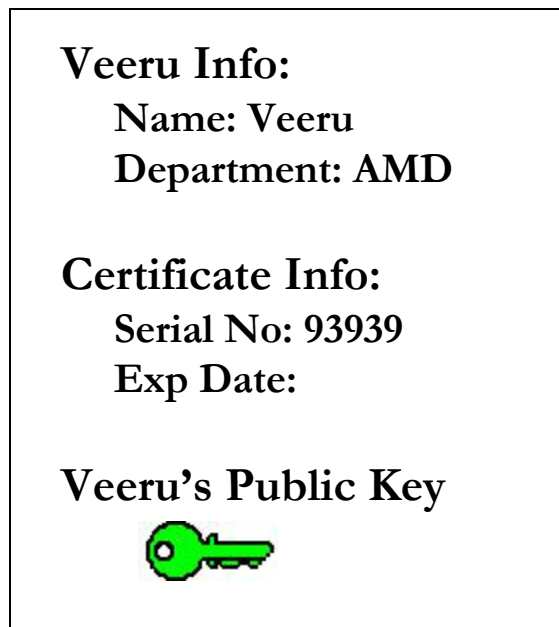
**Hashing + Encryption = Signature Creation**



**Hashing + Decryption = Signature Verification**

# Digital Certificate

- A digital certificate binds the owners public key, name email and other necessary information together



## Structure of Digital Certificate

- **The structure of a X.509 v3 digital certificate is as follows:**
  - **Certificate**
    - **Version**
    - **Serial Number**
    - **Algorithm ID**
    - **Issuer**
    - **Validity**
      - **Not Before**
      - **Not After**
    - **Subject**
    - **Subject Public Key Info**
      - **Public Key Algorithm**
      - **Subject Public Key**
    - **Issuer Unique Identifier (Optional)**
    - **Subject Unique Identifier (Optional)**
    - **Extensions (Optional)**
      - ...

# Sample Certificate

Certificate:

Data:

```

Version: 1 (0x0)
Serial Number: 7829 (0x1e95)
Signature Algorithm: md5WithRSAEncryption
Issuer: C=ZA, ST=Western Cape, L=Cape Town, O=Thawte Consulting cc,
        OU=Certification Services Division,
        CN=Thawte Server CA/emailAddress=server-certs@thawte.com
Validity
  Not Before: Jul  9 16:04:02 1998 GMT
  Not After : Jul  9 16:04:02 1999 GMT
Subject: C=US, ST=Maryland, L=Pasadena, O=Brent Baccala,
        OU=FreeSoft, CN=www.freesoft.org/emailAddress=baccala@freesoft.org
Subject Public Key Info:
  Public Key Algorithm: rsaEncryption
  RSA Public Key: (1024 bit)
    Modulus (1024 bit):
      00:b4:31:98:0a:c4:bc:62:c1:88:aa:dc:b0:c8:bb:
      33:35:19:d5:0c:64:b9:3d:41:b2:96:fc:f3:31:e1:
      66:36:d0:8e:56:12:44:ba:75:eb:e8:1c:9c:5b:66:
      70:33:52:14:c9:ec:4f:91:51:70:39:de:53:85:17:
      16:94:6e:ee:f4:d5:6f:d5:ca:b3:47:5e:1b:0c:7b:
      c5:cc:2b:6b:c1:90:c3:16:31:0d:bf:7a:c7:47:77:
      8f:a0:21:c7:4c:d0:16:65:00:c1:0f:d7:b8:80:e3:
      d2:75:6b:c1:ea:9e:5c:5c:ea:7d:c1:a1:10:bc:b8:
      e8:35:1c:9e:27:52:7e:41:8f
    Exponent: 65537 (0x10001)
Signature Algorithm: md5WithRSAEncryption
93:5f:8f:5f:c5:af:bf:0a:ab:a5:6d:fb:24:5f:b6:59:5d:9d:
92:2e:4a:1b:8b:ac:7d:99:17:5d:cd:19:f6:ad:ef:63:2f:92:
ab:2f:4b:cf:0a:13:90:ee:2c:0e:43:03:be:f6:ea:8e:9c:67:
d0:a2:40:03:f7:ef:6a:15:09:79:a9:46:ed:b7:16:1b:41:72:
0d:19:aa:ad:dd:9a:df:ab:97:50:65:f5:5e:85:a6:ef:19:d1:
5a:de:9d:ea:63:cd:cb:cc:6d:5d:01:85:b5:6d:c8:f3:d9:f7:
8f:0e:fc:ba:1f:34:e9:96:6e:6c:cf:f2:ef:9b:bf:de:b5:22:
68:9f
  
```

# Key Management

- Key management deals with all the provisions made in the cryptosystem design to deal with the keys
  - Generation
  - Exchange or Distribution
  - Storage
  - Safeguarding
  - Use
  - Vetting
  - replacement

# Certifying Authority (CA)

- Certifying authority is an entity which issues

Digital Certificate

- It is a Trusted third party
- CA's are the important characteristics of Public

Key Infrastructure (PKI)

---

- Verify the credentials of the person requesting for the certificate (RA's responsibility)
- Issue certificates
- Revoke certificate
- ~~Generate and upload CRL~~

# Trust Models for CA

- The received certificate shows it belongs to Veeru.
- Does it really belong to Veeru?
- Eve can generate a certificate with name “Veeru”
  - Intercept emails
  - Man-in-the-middle attack

# Trust Models for CA

- Direct Trust
  - Call Veeru and verify
  - What if you don't know Veeru?

# Trust Models for CA

- Subordinate Hierarchy
- Cross-certified Mesh
- Hybrid
- Bridge CA
- Trust List

# Subordinate Hierarchy

- Hierarchical Trust
  - Hierarchic Structure with well known Root CA
  - The Root CA certifies its immediate descendents, which in turn certifies its descendents
  - Each participant must have knowledge of root CA's public key.
  - Compromise of private key is disastrous for the security of its hierarchy

# Cross certified Mesh

- Any CA may certify any other CA
- The model serves well to represent dynamically changing organization structure
- But generality can make it more difficult to determine whether particular CA is or not appropriate certifier for another CA

# Hybrid Model

- Properties of Hybrid Model
  - Multiple root CA's exist
  - All non-root CA's are certified within a root CA's hierarchy
  - Root CA's establish cross certified mesh among themselves, so each hierarchy can reach every other hierarchy via a single cross-certificate at the root level
  - Selective cross-certification between non-root CA's is permissible

# Bridge CA

- Hybrid model does not scale to large numbers
- Cross certification grows to  $n(n-1)$  for 'n' number of hierarchies to be fully connected
- Bridge CA model embodies a central 'cross-certification authority'
  - Provides cross-certificates
- The Bridge CA appears to be prime candidate to extend PKI's across large number of organizations

## Trust List

- In this model, client systems are provided with the public key of the set of trusted roots.
- To be validated, the certificate must chain to one of these trusted roots, sometimes directly and without intervening CA's.
- End points are required to be configured with large set of public keys if communication with large number of CA's is to be supported

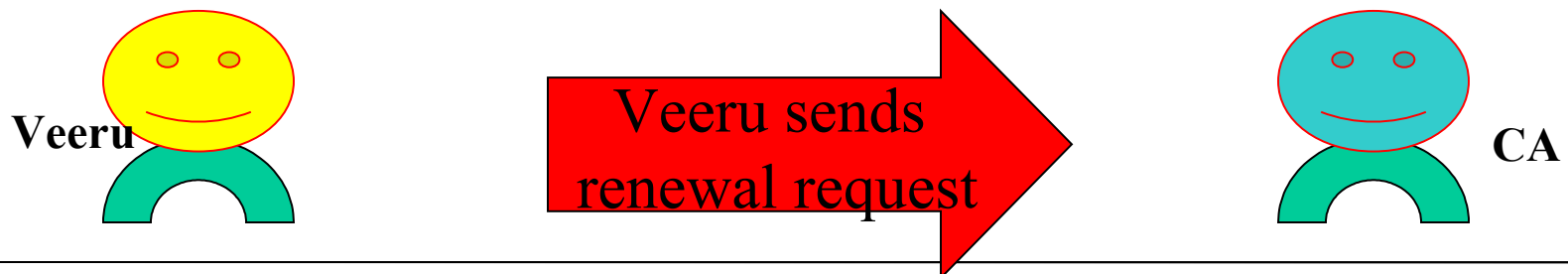
## Certificate Validation Methods

- This is key process of authenticating users and systems and securing network communication through use of digital certificates
- Validating a certificate requires the certificate validation logic in the PKI enabled application to perform series of checks on the different parts of the application
- The validation process performs following checks
  - Digital signature
  - Trust (public key verification)

- Typical Life cycle scenario of Digital Certificates
  - Use until renewal
  - Use until re-keying
  - Use until revocation

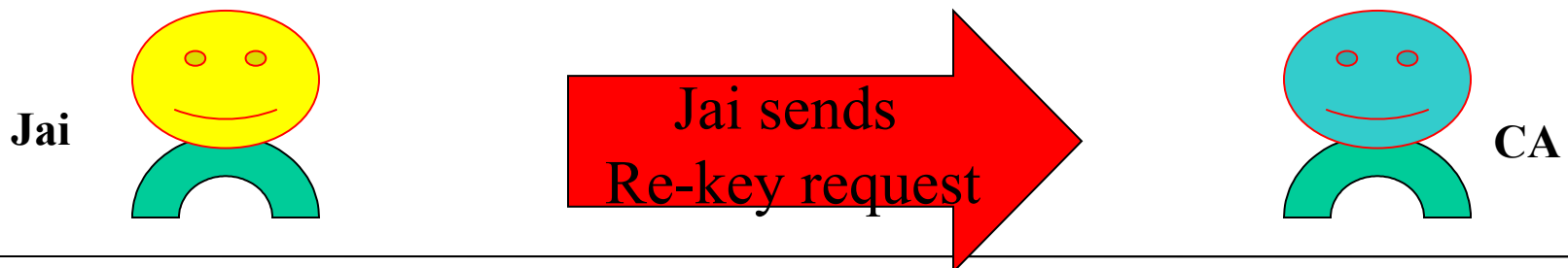
## Use until renewal

- A digital certificate has a defined validity period
- If Veeru wants renew his expiring certificate he sends a renewal request to CA, digitally signed with his old certificate and his private key
- CA issues a new certificate with new validity period
- If there is an overlap in the validity periods, CA can place the old certificate in his CRL



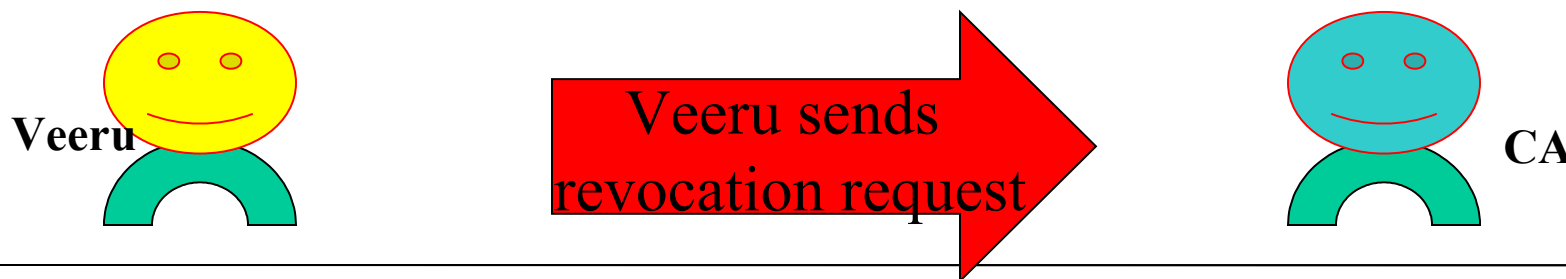
# Use until re-keying

- Suppose Jai decides to change her public and private key pairs (Old keys need not necessarily be compromised)
- He generated new key pair
- He creates a re-key request including her new public key, digitally signs with her old private key and sends request to CA
- CA creates new certificate with the new public key and adds the old certificate to CRL



## Use until revocation

- Veerus private key has been compromised
- Before some uses his key pretending Veeru, he wants to revoke his certificate
- He generates new key pair and sends public key to CA and obtains a new certificate
- CA adds the old certificate on the CRL



# References

- Cryptography and Network security – principles and practice :  
William Stallings
- Applied Cryptography, Second Edition: Bruce Schneier
- [www.certicom.com/index.php/ecc-tutorial](http://www.certicom.com/index.php/ecc-tutorial)
- [http://campustechnology.com/articles/39190\\_2](http://campustechnology.com/articles/39190_2)
- <http://csrc.nist.gov/>
- Handbook of Applied Cryptography, by Menezes
- <http://en.wikipedia.org>
- Cryptographic Techniques for N/w Security

# Thank You