CSCE 465 Computer & Network Security

Instructor: Abner Mendoza

Authentication (III)

Trusted Intermediaries

Roadmap

- Basics
- Kerberos
- Public Key Infrastructure (PKI)

Trusted Intermediaries

- Problem: authentication for large networks
- Solution #1
  - Key Distribution Center (KDC)
    - Representative solution: Kerberos
  - Based on secret key cryptography
- Solution #2
  - Public Key Infrastructure (PKI)
  - Based on public key cryptography

Kerberos

Goals of Kerberos

1. User $\leftrightarrow$ server mutual authentication
2. Users should only need to authenticate once to obtain services from multiple servers
3. Should scale to large numbers of users and servers
   – makes use of a Key Distribution Center so servers don’t need to store information about users

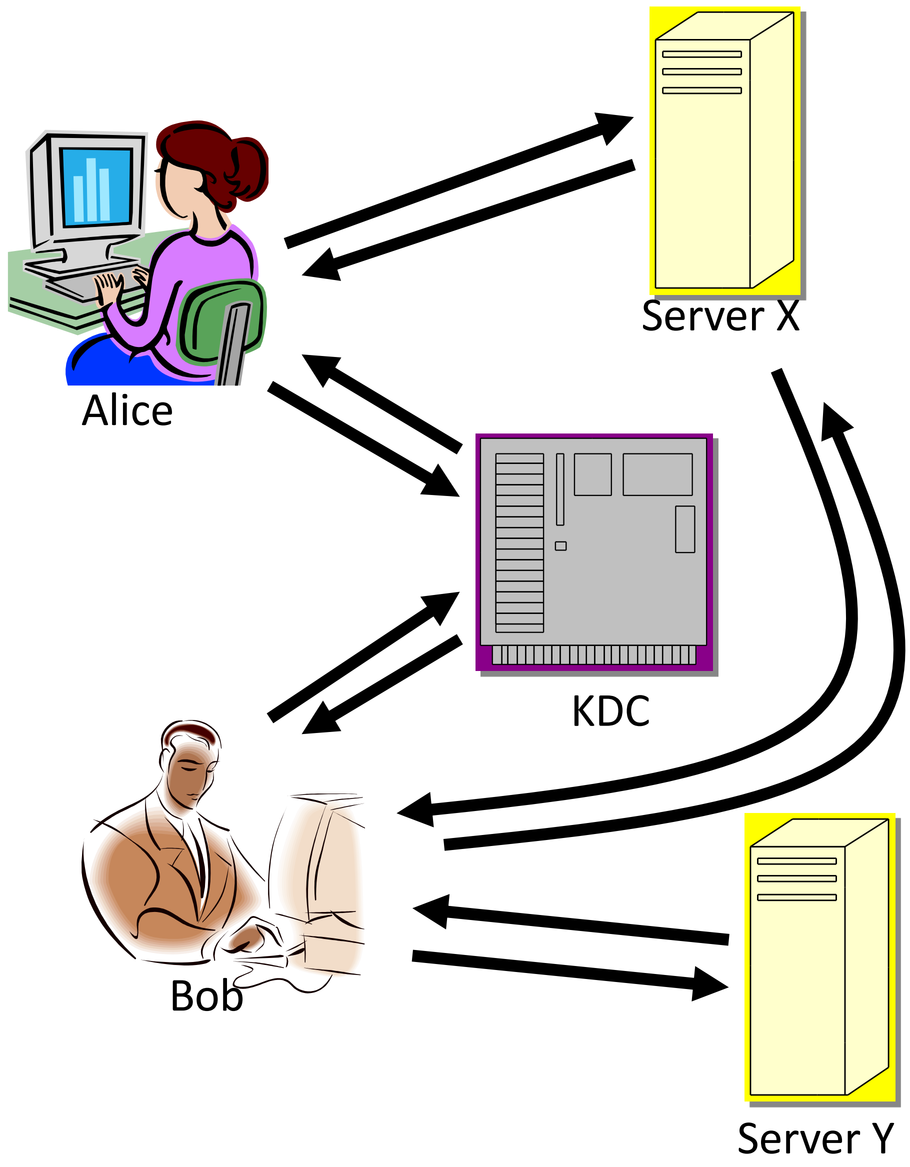
Some Properties

- Kerberos uses only secret key (symmetric) encryption
  - originally, only DES, but now 3DES and AES as well
- A *stateless* protocol
  - KDCs do not need to remember what messages have previously been generated or exchanged
  - the state of the protocol negotiation is contained in the message contents

Example Scenario

- Alice wants to make use of services from X, contacts the KDC to authenticate, gets ticket to present to X

- Bob wants to make use of services from X and Y, contacts the KDC, gets tickets to present to X and Y



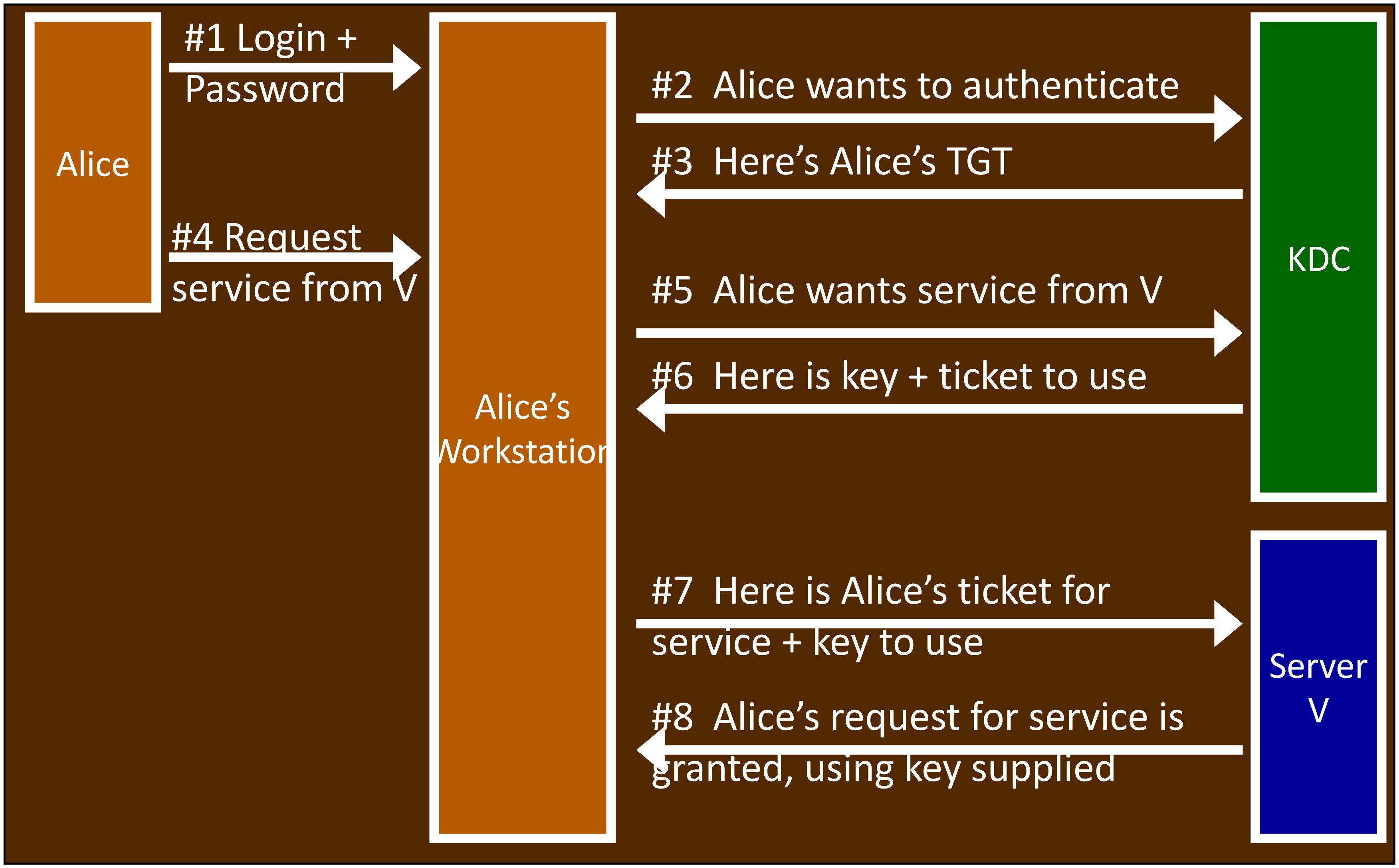
The KDC

- Infrastructure needed (KDC components)
  1. the database of user information (IDs, password hash, shared secret key, etc.)
  2. an authentication server (AS)
  3. a ticket-granting server (TGS)
- The KDC of course is critical and should be carefully guarded

Passwords and Tickets

1. Alice provides a password when she logs into her workstation
2. Alice's workstation…
   - derives Alice's master key from the password
   - asks the KDC for a temporary session key K_A
3. The KDC provides a *ticket-granting ticket* (TGT) for Alice to use; eliminates need for…
   - …repeated authentication
   - …further use of master key

Kerberos v4 Protocol Sketch


Alice
#1 Login + Password
#4 Request service from V
Alice's Workstation
#2 Alice wants to authenticate
#3 Here's Alice's TGT
#5 Alice wants service from V
#6 Here is key + ticket to use
KDC
#7 Here is Alice's ticket for service + key to use
#8 Alice's request for service is granted, using key supplied
Server V

Msg#3: TGT

$K_{KDC}(ID_A \mid Addr_A \mid \mathcal{K}_{A\text{-}KDC} \mid Lifetime_{TGT} \mid TS_{TGT} \mid ID_{KDC})$

- The TGT is what allows the KDC to be stateless
  - means simpler, more robust KDC design
  - allows replicated KDCs
- The TGT contains
  - the session key to be used henceforth
  - the user ID (Alice)
  - the valid lifetime for the TGT

V5: Some Differences with v4

1. v5 uses ASN.1 syntax to represent messages
   - a standardized syntax, not particularly easy to read
   - but, very flexible (optional fields, variable field lengths, extensible value sets, …)
2. v5 extends the set of encryption algorithms
3. v5 supports much longer ticket lifetimes
4. v5 allows "Pre-authentication" to thwart password attacks
5. v5 allows delegation of user access / rights

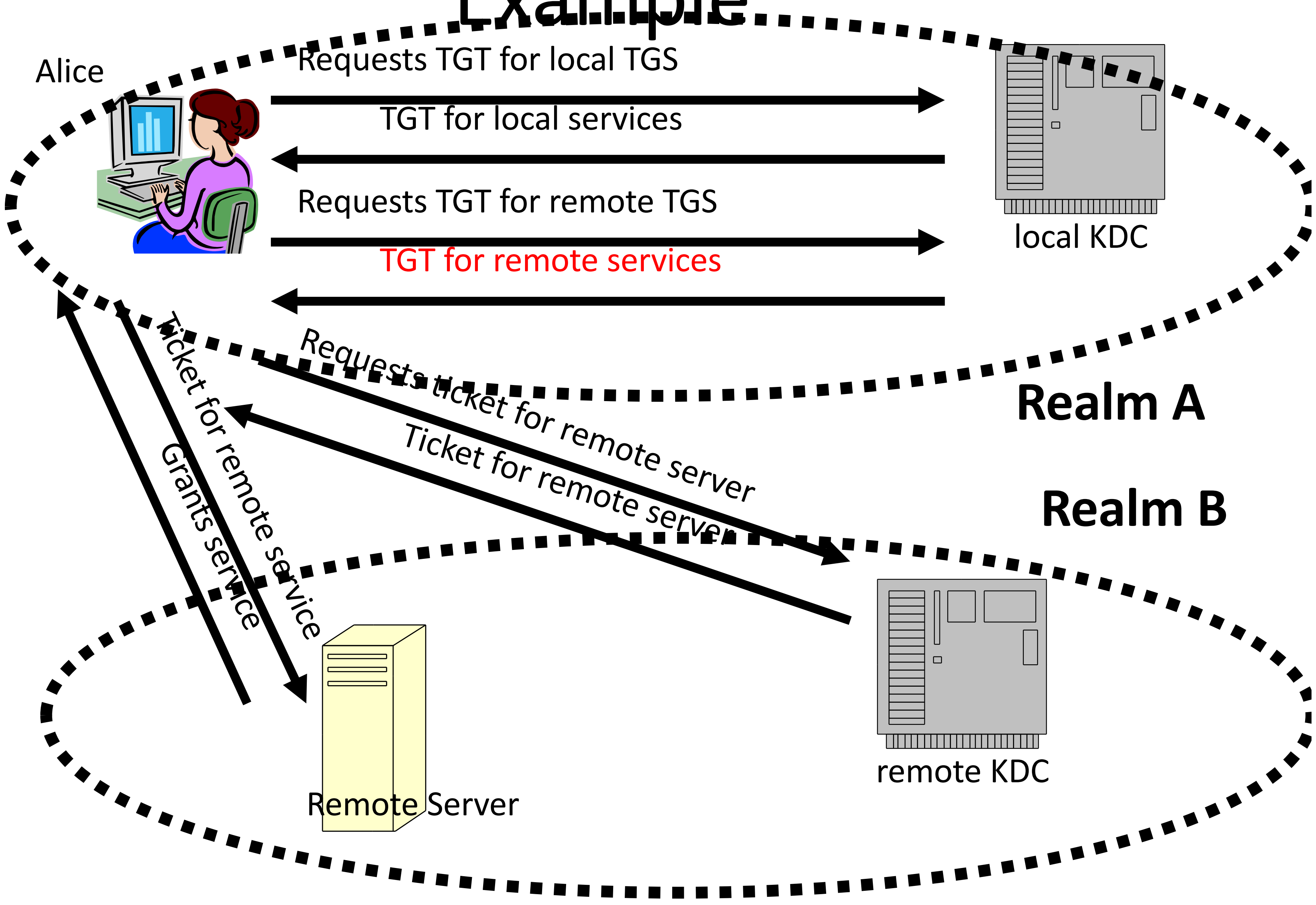
Realms

- A *realm* is a group of resources sharing a single authority for authorization
  - frequently the same as a DNS domain, and referred to by the domain name (e.g., “cse.tamu.edu”)
- A realm consists of…
  1. KDC (TGS, AS, and database)
  2. users
  3. servers

Inter-Realm Authentication

- What if a user wants access to services located in a different realm?
- Simple solution: require Alice to be registered in each realm, has to undergo separate authentication in each
- More complex solution: the KDCs cooperate to perform inter-realm authentication
  - these KDCs must have previously-negotiated shared secret keys
  - receiving KDC can decide for itself whether to accept credentials issued by another KDC

Example


Alice
Requests TGT for local TGS
TGT for local services
Requests TGT for remote TGS
TGT for remote services
local KDC
Realm A
Realm B
Requests ticket for remote server
Ticket for remote server
Ticket for remote service
Grants service
remote KDC
Remote Server

Kerberos Summary

1. Kerberos is the most widely used authentication service
2. Modeled on the Needham-Schroeder protocol, but adds the TGT
3. v5 extends and fixes problems of v4; v4 no longer in active use
4. Inter-realm authentication scales to very large systems (e.g., the Internet)

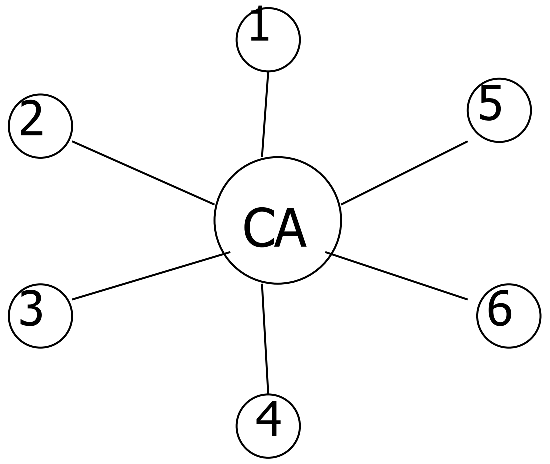
Public Key Infrastructure

What Is PKI

- Informally, the infrastructure supporting the use of public key cryptography.
- A PKI consists of
  - Certificate Authority (CA)
  - Certificates
  - A repository for retrieving certificates
  - A method of revoking certificates
  - A method of evaluating a chain of certificates from known public keys to the target name

Certification Authorities (CA)

- A CA is a trusted node that maintains the public keys for all nodes (Each node maintains its own private key)



If a new node is inserted in the network, only that new node and the CA need to be configured with the public key for that node

Certificates

- A CA is involved in authenticating users' public keys by generating certificates
- A certificate is a signed message vouching that a particular name goes with a particular public key
- Example:
  1. [Alice's public key is 876234]$_{\text{carol}}$
  2. [Carol's public key is 676554]$_{\text{Ted}}$ & [Alice's public key is 876234]$_{\text{carol}}$

- Knowing the CA's public key, users can verify the certificate and authenticate Alice's public key

Certificates

- Certificates can hold expiration date and time
- Alice keeps the same certificate as long as she has the same public key and the certificate does not expire
- Alice can append the certificate to her messages so that others know for sure her public key

CA Advantages

1. The CA does not need to be online. [Why?]

2. If a CA crashes, then nodes that already have their certificates can still operate.

3. Certificates are not security sensitive (in terms of confidentiality).
   - Can a compromised CA decrypt a conversation between two parties?
   - Can a compromised CA fool Alice into accepting an incorrect public key for Bob, and then impersonate Bob to Alice?

CA Problems

- What if Alice is given a certificate with an expiration time and then is revoked (fired) from the system?
  - Alice can still use her certificate till the expiration time expires.
  - What kind of harm can this do?
  - Alice can still exchange messages with Bob using her un-expired certificate.

- Solution:
  - Maintain a Certificate Revocation List (CRL) at the CA. A Certificate is valid if (1) it has a valid CA signature, (2) has not expired, and (3) is not listed in the CA's CRL list.

PKI Models

1. Monopoly model
2. Monopoly + Registration Authorities (RA)
3. Delegated CAs
4. Oligarchy model
5. Anarchy model (Web of Trust): fully distributed
6. Name constraints
7. Top-down with name constraints
8. Bottom-up with name constraints (hierarchical name space, similar to DNS)

Certificate Revocation

- Certificates for public keys (Campus IDs) might need to be revoked from the system
  - Someone is fired
  - Someone graduated
  - Someone’s certificate (card) is stolen

Certificate Revocation

- Certificates typically have an associated expiration time
  - Typically in the order of months (too long to wait if it needs to be revoked)

- Solutions:
  - Maintain a Certificate Revocation List (CRL)
  - A CRL is issued periodically by the CA and contains all the revoked certificates
  - Each transaction is checked against the CRL

CRLs

1. Why are CRLs issued periodically even if no certificates are revoked?

2. How frequent should CRLs be issued?

3. If a CRL is maintained, why associate an expiration time with certificates?

Delta CRL

- Certificates (1) may be huge, and (2) need to be issued periodically
- A Delta CRL includes lists changes from the last complete CRL
- Delta CRLs may be issued periodically (frequently) and full CRLs are issued less frequently

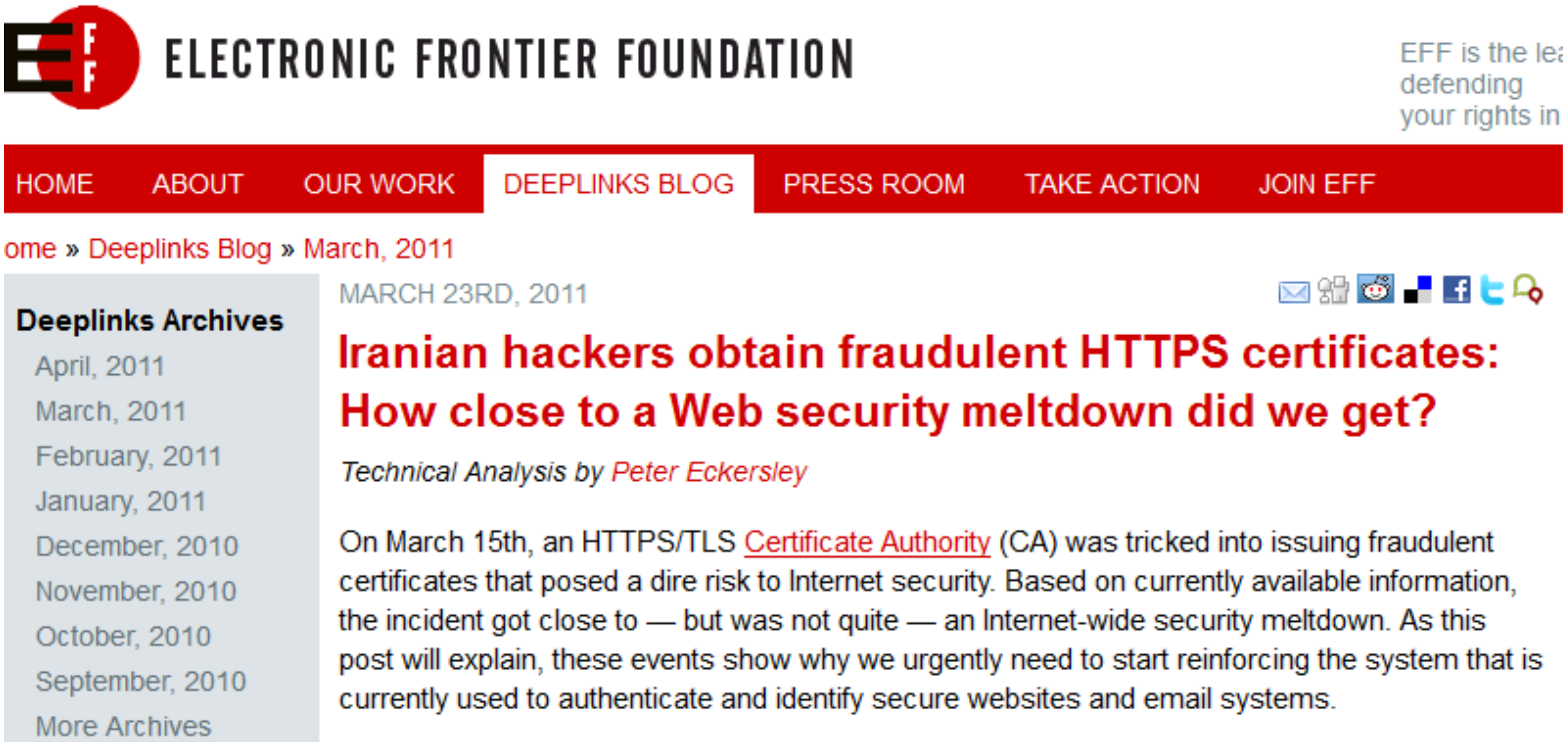
On-line Revocation Servers (OLRS)

- An OLRS is a system that can be queried over the network for the revocation status of individual certificates
- An OLRS maintains the full CRL list
- What if someone impersonates an OLRS?

  .....
  - Solution?

  .....

Famous Attack on CA



ELECTRONIC FRONTIER FOUNDATION

EFF is the le
defending
your rights in

HOME ABOUT OUR WORK DEEPLINKS BLOG PRESS ROOM TAKE ACTION JOIN EFF

ome » Deeplinks Blog » March, 2011

Deeplinks Archives

- April, 2011
- March, 2011
- February, 2011
- January, 2011
- December, 2010
- November, 2010
- October, 2010
- September, 2010
- More Archives

MARCH 23RD, 2011

Iranian hackers obtain fraudulent HTTPS certificates: How close to a Web security meltdown did we get?

Technical Analysis by Peter Eckersley

On March 15th, an HTTPS/TLS Certificate Authority (CA) was tricked into issuing fraudulent certificates that posed a dire risk to Internet security. Based on currently available information, the incident got close to — but was not quite — an Internet-wide security meltdown. As this post will explain, these events show why we urgently need to start reinforcing the system that is currently used to authenticate and identify secure websites and email systems.

Recent Attack on CA

- 9 fraudulently issued certificates from Comodo include following domains:
  - mail.google.com, www.google.com, login.yahoo.com, login.skype.com, login.live.com, addons.mozilla.org, global trustee
- The master CA private keys in its Hardware Security Modules (HSMs) were not compromised
- The future?