

# Instant Revocation

Jon A. Solworth

Dept. of Computer Science and  
Center for RITES  
University of Illinois at Chicago

16 June 2008

# Part I

## Certificates and Revocation

# Public Key Infrastructure (PKI)

- A primary goal of a PKI is to deliver authentication information (public key bindings)
- In a timely
- And efficient manner
- thus increasing the effectiveness of the PKI
- while enabling it to provide services at low cost.

# Certificates

- A **certificate** binds
  - a public key to
  - some characteristic of a user.
  - examples of characteristic: name, credit card number, club membership
- A certificate is **signed** by a **Certificate Authority (CA)**
- The CA's private key which is used for signing must be well protected, and hence is typically kept offline
- Certificates are therefore typically long-lived (e.g., 1 year)
- Hence, the certificate information may become invalid due to:
  - Lost private key
  - Change of characteristic

# Revocation

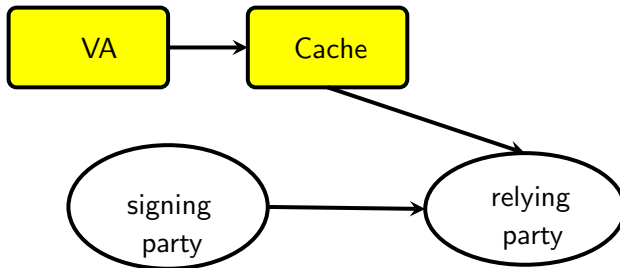
When a certificate becomes invalid, it must be revoked:

- The revocation must be timely, and hence the term “instant revocation” (Sandhu) to mean revocations which take effect in only a second or two.
- They must be efficient, as revocation is *the* major expense of a CA
- Peter Gutmann has called Revocation a Grand Challenge Problem in PKI.
- It is commonly agreed that Revocation in PKI is broken.

# Current revocation mechanisms

CRL	<i>Cert Rev Lists</i> : lists of bad certs, signed by the CA.	Bloated and untimely.
OCSP	<i>On-line Cert Status Protocol</i> : query cert. status, returns a signed response.	CPU and bandwidth expensive
CRS	<i>Cert Rev System</i> : use hash chains to reduce query costs	about 10x cheaper than OCSP
CRT	<i>Cert Rev Tree</i> : use Merkle tree to make updates to caches cheaper.	high query cost

# Anatomy of traditional revocation



- The signing party is the party to be authenticated
- The relying party validates the signature
- The Validity Authority (VA) publishes revocation status
- The cache is queried as to revocation status
- Edges represent Internet connections (\$\$\$)
- Entities incident on arrows pay costs

# Anatomy

- VA provides information to caches, VA holds secrets.
- Caches hold information sufficient to answer validity queries
- Relying parties **query** the cache for certificate revocations
- Relying party generally takes the risk, should check the revocations
- Signing party performs signature, provides if necessary certificate

# Dominant costs

- To reduce costs, must measure dominant costs
- The cost consists of two components:
  - Internet bandwidth cost
  - Crypto CPU costs

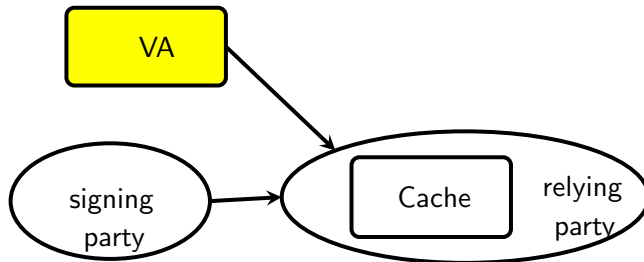
In order of their importance.

- These entities must be sized for *peak*, not average, rates
- LAN networking costs are essentially zero
- Within an ISP (ISP infrastructure plus customer) networking costs are also essentially zero

## Part II

# Certificate Push Revocation

# Certificate Push Revocation (CPR)



- Get rid of queries over the Internet
- Instead, push all information to the relying party
- Relying party can then answer queries without accessing Internet

# CPR Scheme

Very simple scheme, each second:

**updates** Send out certificates revoked and added in last second

**recover cache** Send out part of the total database of valid certificates

Information is digitally signed to ensure it came from VA.

# CPR challenge 1

How to replicate revocation information efficiently at relying party?

- Revocations occur infrequently: given a 10% revocation rate and 100,000,000 certs, revocation rate is .3/second
- Use Internet multicast to “broadcast” revocation information to each relying party
- multicast is unreliable, build in redundancy so that lost packets can be recovered.

# CPR challenge 2

How to recover from cache failure?

- validity vector contains a bit per certificate
- Multicast part of the validity bit vector each second
- If the whole bit vector is transmitted in  $T$  seconds, then bandwidth is  $N/T$ , where  $N$  is the number of certificates.
- Assumes the certificate serial numbers are one range, easily generalizable to multiple ranges

## Part III

# Performance

# performance measurements

$Q$  queries/day (from 1-64)

$P_a$  peak/average authorizations in busiest time interval

big VA 100,000,000 valid certificates

In general

- Want to drive  $Q$  up, as that increases usefulness of PKI
- $P_a$  is typically very significant, because of time-of-day imbalances, and also annual cycles (e.g., Christmas shopping)

## OCSP Performance

$Q$	$P_a$	<b>Auth/sec</b> Big VA	<b>bits/sec</b> Big VA	<b>CPU cores</b> Big VA
1	1	1,157.41	2,444,444.44	6.89
4	1	4,629.63	9,777,777.78	27.55
16	1	18,518.52	39,111,111.11	110.19
64	1	74,074.07	156,444,444.44	440.74
1	5	5,787.04	12,222,222.22	34.43
4	5	23,148.15	48,888,888.89	137.73
16	5	92,592.59	195,555,555.56	550.93
64	5	370,370.37	782,222,222.22	2,203.70
1	10	11,574.07	24,444,444.44	68.87
4	10	46,296.30	97,777,777.78	275.46
16	10	185,185.19	391,111,111.11	1,101.85
64	10	740,740.74	1,564,444,444.44	4,407.41

## CRS Performance

$Q$	$P_a$	<b>Auth/sec</b> Big VA	<b>bits/sec</b> Big VA	<b>CPU cores</b> Big VA
1	1	1,157.41	222,222.22	54.6
4	1	4,629.63	888,888.89	54.6
16	1	18,518.52	3,555,555.56	54.6
64	1	74,074.07	14,222,222.22	54.6
1	5	5,787.04	1,111,111.11	54.6
4	5	23,148.15	4,444,444.44	54.6
16	5	92,592.59	17,777,777.78	54.6
64	5	370,370.37	71,111,111.11	54.6
1	10	11,574.07	2,222,222.22	54.6
4	10	46,296.30	8,888,888.89	54.6
16	10	185,185.19	35,555,555.56	54.6
64	10	740,740.74	142,222,222.22	54.6

## CPR performance part 1 (multicast revocation info)

## Steady state costs

$H$	$P_r$	Large VA
1	1	2,378.14
1	10	2,469.44
1	100	3,382.40
10	1	2,469.44
10	10	3,382.40
10	100	12,512.00
20	1	2,570.88
20	10	4,396.80
20	100	22,656.00

$H$  is the redundancy

$P_r$  is the peak revocations

# CPR performance part 2 (multicast recovery info)

- **Recovery cost** relying party only subscribes during recovery
- Loading the initial certificate validity vector

recovery time	bits/second Big VA
60	1,854,127.85
120	928,201.93
300	372,646.37

- Total VA bandwidth costs: 1–2 T1 lines
- Total VA computational power less than one core

# Memory, CPU use at Cache

- Memory at cache is about 12.5 mbytes for validity vector
- Validity vector is by far the dominant memory cost
- Relying party uses about .001 of a core
- Relying party can host cache with trivial CPU, memory overhead, spending bulk of time on other stuff

## Part IV

# Low bandwidth devices

# Low bandwidth devices

- CPR multicast bandwidth (like other Internet protocols) must be paid by **both** the VA and the customer.
- It is very economical for the VA
- But its requirement that the customer always listen for updates may be too expensive or
- There may be organizational reasons why direct multicast might not be allowed (e.g., firewall policy)

# Move the cache to the ISP (or enterprise)

- VA bandwidth requirements unchanged
- ISP minimize Internet bandwidth costs (their biggest cost)
- High performance since the latency is low
- Economic incentives for ISP to set things up, since the alternative is more traditional (and much more expensive) revocation techniques

## 3 Schemes

As usual, **untrusted** is better since

- answers are checkable
- an untrusted cache cannot effect integrity

Schemes:

1. An **untrusted cache** in which query response is signed by the VA.
  - Revocation information kept as a Merkle Hash Tree
  - Query response is logarithmic in size of revocation store

## 3 Schemes (cont'd)

2. In **signer sends**, an untrusted cache at the signer is used to send information to relying party
  - Two types of relying parties, big and small
  - Typically, small signers do not talk to each other
  - Big signers directly get multicast revocations
  - Small signers receive revocation information from big signers
3. A **trusted cache**, unlike untrusted cache, but does not required signed validations since they are trusted.
  - Single-bit response
  - Appropriate for enterprise

## Part V

### Related work

# Related work

- Rivest'98: talked about alternatives to CRL
- Micali'96: invented CRS
- Kocher'98: invented CRT
- Chadwick&Anthony'07: Webdav trusted directory
- Goyal'07: looked query vs. sending to cache attaching

## Part VI

# Conclusion

# Conclusion

- CPR multicasts the slowest changing part of the revocation database
- CPR provides fault tolerance to account for packet loss, cache failure
- CPR is independent of authentication rate, encouraging use of strong authentication
- Instant Revocation can be done for 100,000,000 Certs with
  - 1 core (10s to 1000s of times more efficient than traditional techniques)
  - 1-2 T1 (100s to 1000s of times more efficient than traditional techniques)
- CPR is even more efficient when revocation is not instant
- We are building CPR now