RevCast: Fast, Private Certificate Revocation over FM radio

Aaron Schulman
Stanford University

Dave Levin
University of Maryland

Neil Spring
University of Maryland
Authentication in the PKI
I want an encrypted connection.
I want an encrypted connection.
I want an encrypted connection.

Certificate #12

Bank of America

Signed by CA:

VERISIGN

The CA (✓) attests that

is controlled by

Bank of America
Is bound to Bank of America?

Certificate #12

Signed by CA: VERISIGN

The CA (✓) attests that

is controlled by Bank of America

Authentication in the PKI
Authentication in the PKI

Is bound to Bank of America?

Certificate #12

Signed by CA: VERISIGN

The CA (Verisign) attests that is controlled by Bank of America
Authentication in the PKI

Is 🛡 bound to **Bank of America**?

The CA (✔) attests that

is controlled by

**Bank of America**

**Certificate #12**

Signed by CA: **VERISIGN**

Trusted Root CAs

Bank of America
Revocation in the PKI

Trusted Root CAs

Certificate #12

Signed by CA: VERISIGN

Bank of America
Revocation in the PKI

Trusted Root CAs

Certificate #12
Signed by CA:

Bank of America
Revocation in the PKI

Trusted Root CAs

Certificate #12

Bank of America
Signed by CA: VERISIGN

Bank of America

Chrome browser loading HTTPS://www.bankofamerica.com
Revocation in the PKI
Revocation in the PKI

Trusted Root CAs

Certificate #12

Signed by CA: VERISIGN

Bank of America

Bank of America
Revocation in the PKI

Trusted Root CAs

Certificate #12
Bank of America
Signed by CA: VERISIGN

Bank of America
VERISIGN
Revocation in the PKI
Revocation in the PKI

Trusted Root CAs

Certificate #12
Signed by CA: VERISIGN

Bank of America

Certificate #12
Signed by CA: VERISIGN

Revocation

Bank of America

VERISIGN™

thawte™

Go Daddy.COM

https://www.bankofamerica.com
Revocation in the PKI

Trusted Root CAs

Certificate #12
Signed by CA: 
Bank of America

Revocation Certificate #12
Signed by CA: 
Bank of America

The CA (ystore) breaks the binding of with Bank of America
Revocation in the PKI

Trusted Root CAs

Certificate #12
Bank of America
Signed by CA: VERISIGN

The CA (pery) breaks the binding of with Bank of America

Revocation Certificate #12
Signed by CA: VERISIGN
Revocation in the PKI

Trusted Root CAs

Certificate #12
Signed by CA: VERISIGN

Certificate #12
Signed by CA: VERISIGN

The CA (山路) breaks the binding of with

✔

❌

Bank of America

Bank of America

The CA (山路) breaks the binding of with Bank of America
Revocation in the PKI

Trusted Root CAs

Certificate #12
Signed by CA: VERISIGN

Certificate #12
Signed by CA: VERISIGN

The CA (VERISIGN) breaks the binding of with Bank of America

The browser displays an error for the Revocation Certificate #12.
Revocation in the PKI

One revocation every 1.1 seconds for all CAs on the Internet

Trusted Root CAs

The CA (_VERISIGN_) breaks the binding of with

Certificate #12
Signed by CA:

The CA (_VERISIGN_) breaks the binding of with Bank of America

Certificate #12
Signed by CA:

One revocation every 1.1 seconds for all CAs on the Internet
Every device needs revocations

Revocation Certificate #12
Signed by CA: VERISIGN
Every device needs revocations
Properties of revocation systems
Properties of revocation systems

**Timeliness**

Clients’ revocation state should be up-to-date, ideally within 10s of seconds.
Properties of revocation systems

**Timeliness**
Clients' revocation state should be up-to-date, ideally within 10s of seconds.

**Low-cost dissemination**
The distribution mechanism must scale with CAs, certificates, and clients.
Properties of revocation systems

Timeliness
Clients’ revocation state should be up-to-date, ideally within 10s of seconds

Low-cost dissemination
The distribution mechanism must scale with CAs, certificates, and clients

Privacy
Users’ browsing habits should not have to be revealed
Properties of revocation systems

It is generally regarded that no system can possibly achieve all three.

**Timeliness**
Clients’ revocation state should be up-to-date, ideally within 10s of seconds.

**Low-cost dissemination**
The distribution mechanism must scale with CAs, certificates, and clients.

**Privacy**
Users’ browsing habits should not have to be revealed.
Properties of revocation systems

It is generally regarded that no system can possibly achieve all three.

- **Timeliness**: Clients' revocation state should be up-to-date, ideally within 10s of seconds.
- **Low-cost dissemination**: The distribution mechanism must scale with CAs, certificates, and clients.
- **Privacy**: Users' browsing habits should not have to be revealed.

RevCast

✔ ✔ ✔
Existing revocation systems

Certificate Revocation Lists (CRL)

Online Certificate Status Protocol (OCSP)

Short lived certs

OCSP Stapling
<table>
<thead>
<tr>
<th>Existing revocation systems</th>
<th>Timeliness</th>
<th>Low-cost dissemination</th>
<th>Privacy</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRLs</td>
<td></td>
<td></td>
<td>✔️</td>
</tr>
<tr>
<td>OCSP</td>
<td>✔️</td>
<td>☓</td>
<td>☓</td>
</tr>
<tr>
<td>Short lived</td>
<td>✔️</td>
<td>☓</td>
<td>✔️</td>
</tr>
<tr>
<td>Stapling</td>
<td>✔️</td>
<td>☓</td>
<td>✔️</td>
</tr>
</tbody>
</table>

- **CRLs**: Client receives a CRL to check certificate revocation. Taps on Timeliness and Privacy.
- **OCSP**: Client requests revocation via OCSP. Taps on Privacy. No refreshes on Timeliness.
- **Short lived**: Client checks certificate with Org. Taps on Privacy and Timeliness.
- **Stapling**: Client checks certificate with Org. Taps on Privacy and Timeliness.
Existing revocation systems

<table>
<thead>
<tr>
<th></th>
<th>Timeliness</th>
<th>Low-cost dissemination</th>
<th>Privacy</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CRLs</strong></td>
<td>❌</td>
<td>❌</td>
<td>✔️</td>
</tr>
<tr>
<td><strong>OCSP</strong></td>
<td>✔️</td>
<td>❌</td>
<td>❌</td>
</tr>
<tr>
<td><strong>Short lived</strong></td>
<td>✔️</td>
<td>❌</td>
<td>✔️</td>
</tr>
<tr>
<td><strong>Stapling</strong></td>
<td>✔️</td>
<td>❌</td>
<td>✔️</td>
</tr>
</tbody>
</table>

All of these protocols rely on unicast transmission of revocations.
Unicast is not well suited for distributing revocations

Doesn’t \textbf{scale} to distributing to every device on the Internet

Failures are \textbf{benign} indication of connectivity issues (soft-fail)

\textit{Multicast revocation is also flawed (Sybils, MITM, DoS)}
RevCast

We propose broadcasting revocations over FM RDS

Tower: http://cityspottercards.com/
We propose broadcasting revocations over FM RDS.

Tower: http://cityspottercards.com/
FM RDS coverage is ideal for disseminating revocations

- Transmitters are where people are
- Up to 10 million people per tower
Properties of revocation systems

Low-cost dissemination

One transmission covers up to 10 million & Under-monotized

Privacy

Radio broadcasts are inherently receiver anonymous
Properties of revocation systems

**Low-cost dissemination**

One transmission covers up to 10 million & Under-monotizized

**Privacy**

Radio broadcasts are inherently receiver anonymous
Solved. Let’s go party like it’s 1989!
One tiny problem. RDS has an effective bitrate of 421.8 bps.
Rest of the talk

RevCast protocol - fitting revocations in 421.8 bps

Evaluate RevCast with 2 months of revocations
Revoking over FM RDS

CAs

Radio station

Receivers

VERISIGN

thawte

GoDaddy.com
Revoking over FM RDS

CAs

VERISIGN

thawte

Go Daddy

Radio station

Receivers

1

2

3
Revoking over FM RDS

CAs

VERISIGN

thawte™
its a trust thing™

GoDaddy.com

Radio station

R1

R2

R3

R1 R2 R3

Receivers

Chrome

Android

Smartphone
Losses can go undetected
Losses can go undetected
Losses can go undetected
Losses can go undetected

CAs

VERISIGN

thawte™

GoDaddy.com

Radio station

Receivers

GoDaddy didn’t revoke
Making losses detectible with "nothing now"
Making losses detectible with “nothing now”
Making losses detectible with "nothing now"

GoDaddy says they didn’t revoke
Making losses detectible with “nothing now”
Making losses detectible with “nothing now”

CAs
VERISIGN

Radio station

Receivers

Danger!!! I am not up-to-date with GoDaddy

Go Daddy.com
thawte

Chrome
Android
Sleeping receivers can lose synchronization.
Sleeping receivers can lose synchronization.
Sleeping receivers stay up-to-date with “Nothing since”
Sleeping receivers stay up-to-date with "Nothing since"

CAs
- VERISIGN
- thawte™
- GoDaddy.com

Radio station

Receivers
- Chrome
- Android
- Mobile

I didn’t miss anything from GoDaddy
RevCast messages

Revocation

Nothing now

Nothing since

All other CAs
Must sign every 10s

Revoking CAs
Shortening “nothing now” and “nothing since”
Shortening “nothing now” and “nothing since”
Shortening “nothing now” and “nothing since”

Problem: FM RDS doesn’t scale to \textit{hundreds of} signatures
Shortening “nothing now” and “nothing since”

Problem: FM RDS doesn’t scale to *hundreds of* signatures
Shortening “nothing now” and “nothing since”

Problem: FM RDS doesn’t scale to hundreds of signatures

Multi-signatures: combine multiple CA signatures into one

[Boldyreva 2003]
Shortening “nothing now” and “nothing since”

Problem: FM RDS doesn’t scale to hundreds of signatures

Multi-signatures: combine multiple CA signatures into one

2.89 seconds for both “nothing new” and “nothing since”
RevCast summary

CAs
- VERISIGN

Radio station

Receivers
- Chrome
- Android
- Smartphone
Evaluation

1. How quickly can RevCast send updates?

2. How would RevCast handle a worst case scenario?

3. Is RevCast practical?
Evaluation

978 CRLs extracted from Rapid7’s scan of the entire IPv4 space

[Graph showing the number of revocations per day for different days of the week and years 2013 and 2014, with a notable increase around the Heartbleed event.]
Evaluation

978 CRLs extracted from Rapid7’s scan of the entire IPv4 space

Security takes the weekends off
Evaluation

978 CRLs extracted from Rapid7’s scan of the entire IPv4 space

Security takes the weekends off

# of Revocations Per Day

Month: Year: 2013

Heartbleed

Weekday
Saturday
Sunday

114,021
402,747
How quickly can RevCast update?

CDF

Fraction of interval required

Interval (s)

10
20
60
120
How quickly can RevCast update?

96% of 10sec intervals

99.999% of 2min intervals

CDF

Fraction of interval required

Interval (s)

10
20
60
120

0.01 0.1 1 10 100

0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1

Fraction of interval required

96% of 10sec intervals
99.999% of 2min intervals
Worst-case scenario

CDF

Fraction of interval required

Interval (10s)
Pre-heartbleed
Post-heartbleed
Worst-case scenario

70% of time, up-to-date within 10 seconds
Worst-case scenario

70% of time, up-to-date within 10 seconds

The most extreme takes 15.5 minutes
Why does RevCast work?
Why does RevCast work?

In a small window, there are usually few revocations.
Why does RevCast work?

In a small window, there are usually few revocations.

Different CAs rarely revoke within the same window.

CDF

Interval (s)

20

120

CAs Revoking Per Interval
Why does RevCast work?

In a small window, there are usually few revocations

Different CAs rarely revoke within the same window

• Most CAs co-sign “nothing now” messages
• When they do have something to revoke, it’s a small list
FM RDS is ideal for disseminating revocations

**Receivers:**
- Tiny and cheap (2.5 x 2.5 mm)
- Already built into many devices
  *receivers not antennas*

**Robustness:**
- 10 error correcting bits for every 16 bits
- VHF & FM (same used for emergency weather radio)
Conclusions

It is possible to design a revocation system that provides timelines, privacy, and is low cost.

Broadcasting revocations is a novel application of multi-signatures.

Practical in today’s Internet, and necessary in tomorrow’s.