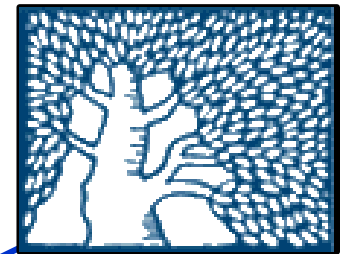


Primary-Secondary-Resolvers Membership Proof Systems and their Applications to DNSSEC

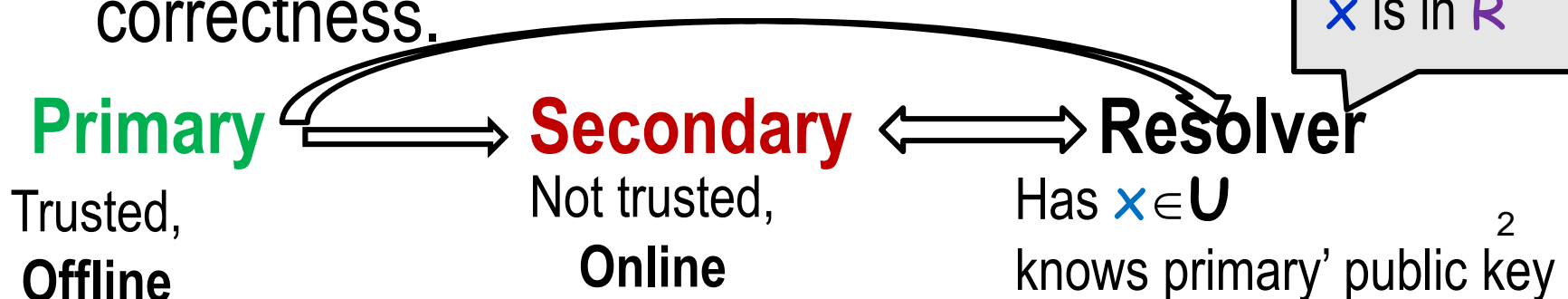


Weizmann Institute

**Sharon Goldberg, Moni Naor, Dimitris Papadopoulos,
Leonid Reyzin, Sachin Vasant, Asaf Ziv**

The (non) membership problem

- Database \mathcal{R} of n elements from universe \mathcal{U}
 - With object $x \in \mathcal{R}$ associated information y
- Want to allow **lookups** in \mathcal{R} such that
 - If $x \in \mathcal{R}$ then answer is 'yes' and associated y retrieved
 - If $x \notin \mathcal{R}$ then answer is 'no'
- Don't want to leak more information than this!
- **Entity** providing answer: **not** trusted wrt to correctness.



Motivation: Secure DNS Lookups

- DNS: Domain Name Server

Example.com: 172.16.254.1

- Allows the translation of names to IP Addresses
- Plain DNS does **not** guarantee **authenticity** to users

- DNSSEC: Security extension of DNS

- Retrieved records are authenticated (signed)
- What about non-existing records? **Denial of existence**
- Current methods leak information about the set
- Allow 'zone enumeration'

Listing all names in a domain

- Want to improve DNSSEC

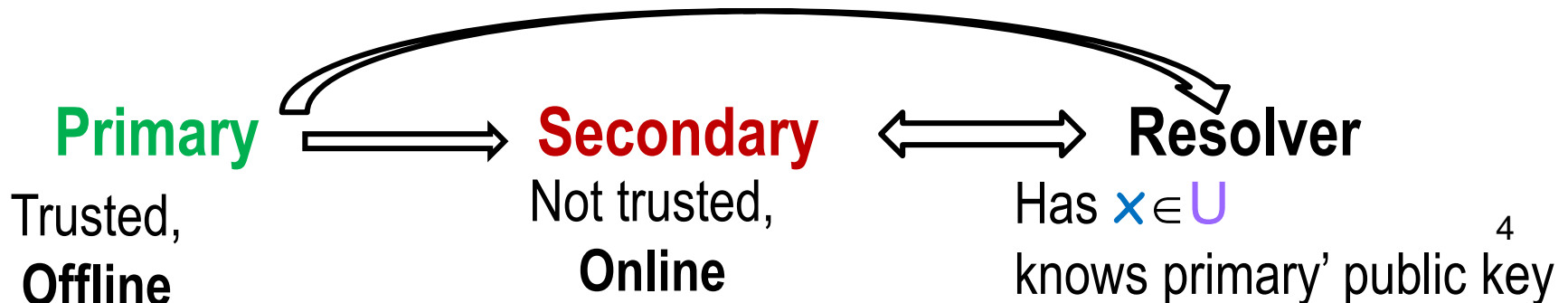
How NSEC Works (Roughly)

- The **primary** signs all existing records
 - plus link to the next record in sorted order
 - Gives all signatures to secondary
 - Public key: signing key

After a while: learn all of **R**

- Given query **x**
 - If $x \in R$ then **secondary** gives signature on record
 - If $x \notin R$ then **proof of non existence** is:

signed pair (x_1, x_2) such that $x_1 < x < x_2$



Is Zone Enumeration a Real Problem?

Much debate in the networking world: After all this is public information?

- There is a difference between **willing to answer questions** and **revealing everything you know**
- Enumerating hostnames creates a foothold for more complex attacks
- Legal reasons to protect host names (e.g. EU Data Protection laws)
- IETF rewrote the DNSSEC standard to 'deal' with this issue in 2008

How NSEC3 Works (Roughly)

- Instead of storing x itself: store $h(x)$
 - h is some one-way/random oracle function
- The problem is now similar to the case where one is given oracle access to the membership function
 - At best: this is an **obfuscated membership program** and allows the adversary “unlimited” queries
- Bernstein’s NSEC3 walker

May also add salt

What Do We Have to Say

- **Model** the problem
 - **Primary-Secondary-Resolvers Membership Proof Systems**
Completeness, Soundness & Privacy (Zero-Knowledge)
- **Explain** why current attempts have all failed
 - Show that the **secondary** must be performing **online public-key authentication**
 - Can convert to **signatures** in some circumstances
- **Suggest** various constructions to PSRs
 - Based on RSA plus random oracles
 - Based on VRFs and VUFs
 - Based on HIBEs

NSEC5

How Our NSEC5 Works (Roughly)

- Instead of storing x itself: store

$$F(x) = h_2(\text{RSA}^{-1}(h_1(x)))$$

where h_1 and h_2 are random oracles

- Unlike $h(x)$ in NSEC3: not everybody can compute it.
- Equip the **secondary** with the RSA secret key
- To prove that $F(x) = z$:
 - **secondary** sends $S(x) = \text{RSA}^{-1}(h_1(x))$
- **Resolver** needs to know public RSA key
 - One additional RSA computation

How NSEC5 Works (Roughly)

Primary preparation

- Choose Signing key **plus** RSA key (N, e) and hash functions

$h_1: U \rightarrow [N]$ and $h_2: [N] \rightarrow \{0,1\}^*$ Random oracles

Denote $S(x) = \text{RSA}^{-1}(h_1(x))$ and $F(x) = h_2(S(x))$

- For every $x_i \in R$ compute $y_i = F(x_i)$ Plays the role of $h(x)$ in NSEC3
- Sign them in **pairs** by **lexicographical** order: $\text{Sign}(y_i, y_{i+1})$
- For every $x_i \in R$ also sign their values: $\text{Sign}(x_i, v_i)$

Secondary's Public key $PK_S = (N, e)$

Secondary's secret key $SK_S = d$ and

- Set R and $\text{Sign}(x_i, v_i)$
- For all pairs $\text{Sign}(y_i, y_{i+1})$

} I_S

NSEC5 RSA Construction

Denote $S(x) = \text{RSA}^{-1}(h_1(x))$ and $F(x) = h_2(S(x))$

- For every $x_i \in \mathcal{R}$ compute $y_i = F(x_i)$
- Sign them in **pairs** by **lexicographical** order: $\text{Sign}(y_i, y_{i+1})$
- For every $x_i \in \mathcal{R}$ also sign their values: $\text{Sign}(x_i, v_i)$

Secondary

- Given a query $x \in \mathcal{R}$, the **secondary** returns $\text{Sign}(x_i, v_i)$
- Given query $x \notin \mathcal{R}$, the **secondary** returns:

$\text{Sign}(y_i, y_{i+1})$ and $S(x)$ such that $y_i < F(x) < y_{i+1}$

A **Resolver** verifies query x by checking that:

- $y_i < h_2(S(x)) = F(x) < y_{i+1}$
- $\text{RSA}(S(x)) = h_1(x)$

NSEC5 RSA Performance

Performance comparable to NSEC3

Primary: Signature on pairs $\text{Sign}(y_i, y_{i+1})$

Signature on values: $\text{Sign}(x_i, v_i)$

For every $x_i \in \mathcal{R}$ compute $y_i = F(x_i)$

From lower bound:
must work
as hard as signing!

Secondary

- For query $x \notin \mathcal{R}$: secondary computes $y = F(x)$ and returns:
 $\text{Sign}(y_i, y_{i+1})$ and $S(x)$

A Resolver verifies query x by checking that:

- $y_i < h_2(S(x)) = F(x) < y_{i+1}$
- $\text{RSA}(S(x)) = h_1(x)$

Based on

- **NSEC5: Provably Preventing DNSSEC Zone Enumeration** Sharon Goldberg, Moni Naor, Dimitris Papadopoulos, Leonid Reyzin, Sachin Vasant, Asaf Ziv
Cryptology ePrint Archive: Report 2014/582
- **PSR Membership Proof Systems**, Moni Naor and Asaf Ziv

ITCS 2015 at Weizmann Institute

- The **6th Innovations in Theoretical Computer Science (ITCS)** conference, will be held at the Weizmann Institute of Science, Israel
January 11-13, 2015
- **Deadline: Aug 8th 2014**
- Program Chair: Tim Roughgarden



http://www.wisdom.weizmann.ac.il/~naor/itcs2015_main.html

- Sponsored by ACM SIGACT

