The OpenPGP Web of Trust

Ralph Holz, Georg Carle

Network Architectures and Services
Technische Universität München

5 Dec 2012

Scope of this lecture

The purpose of this lecture is two-fold.

Introduction to a very different PKI: Web of Trust (WoT)

- We know: X.509, the hierarchical PKI we use for SSL/TLS
- Today: OpenPGP, a user-driven, non-hierarchical PKI

Results from our publication

- A. Ulrich, R. Holz, P. Hauck, G. Carle:
  Investigating the OpenPGP Web of Trust. ESORICS 2011.
- Contains results on the use of the OpenPGP WoT

Part 1:
The concept of a Web of Trust

Introducing the Web of Trust

PGP/GnuPG (GPG)

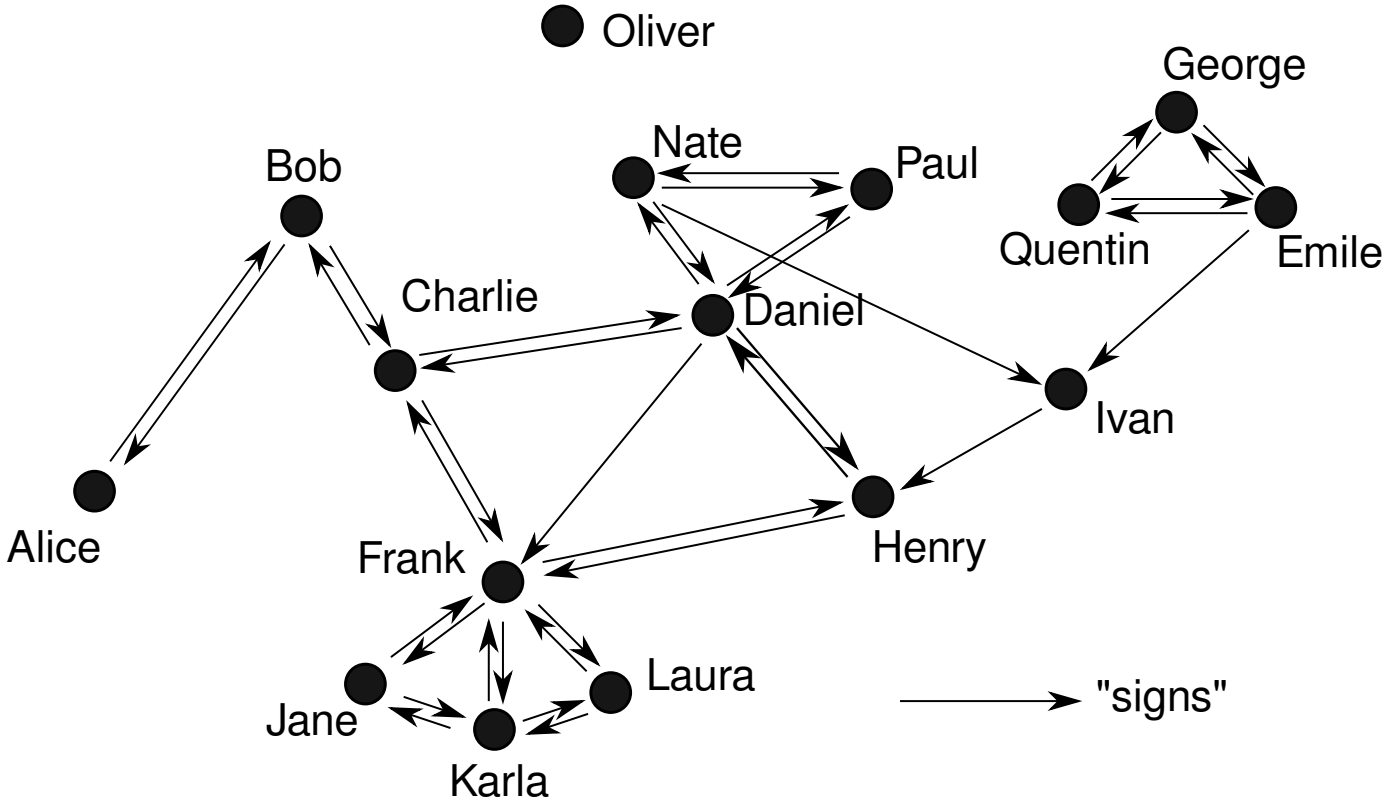
- Widely used implementations of OpenPGP (authentication & encryption)
- Popular with geeks & security community
- Often used for e-mail

Web of Trust (WoT)

- Idea: everyone can certify anyone else
- Decentralized
- Certification Authorities (CAs) not disallowed: just very active users

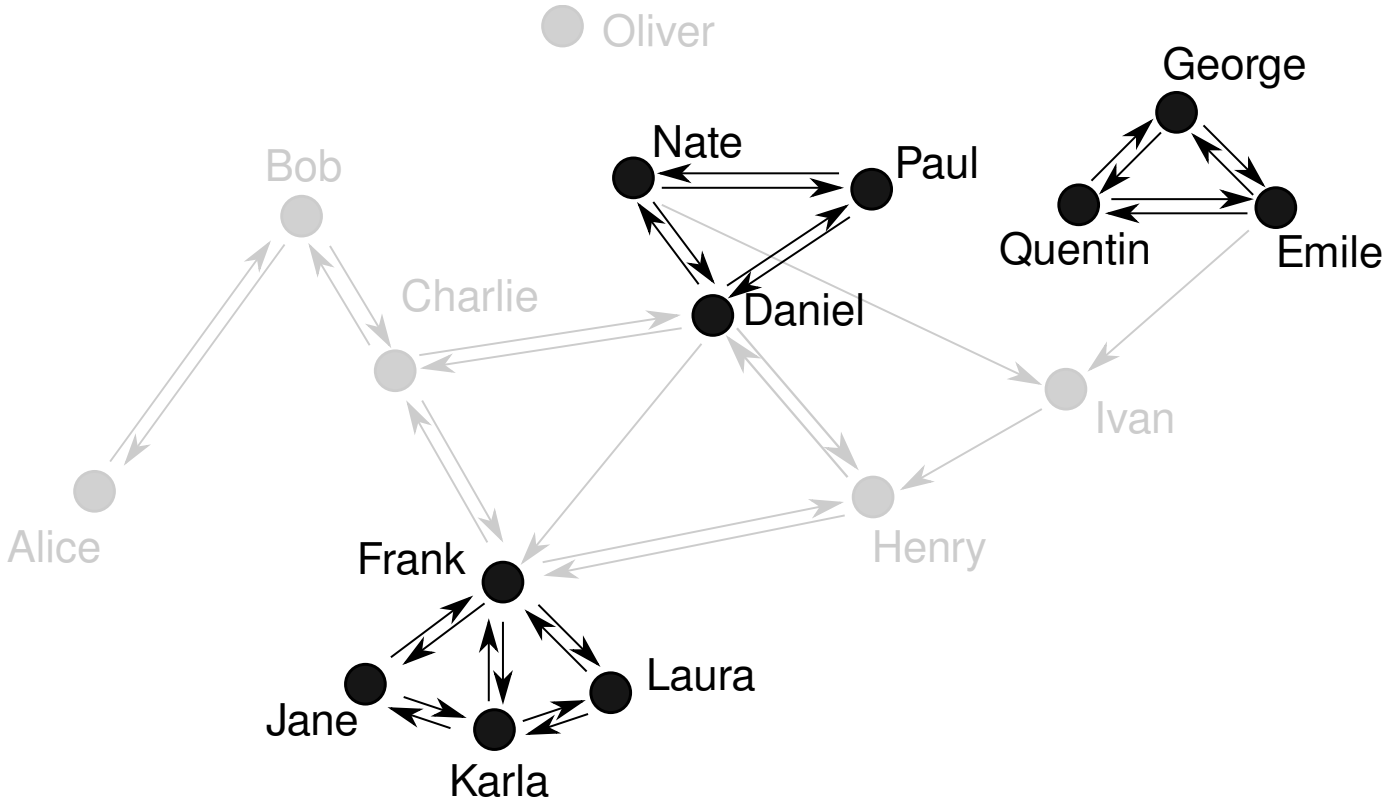
Web of Trust (WoT): Directed Graph

Signing results in a directed graph



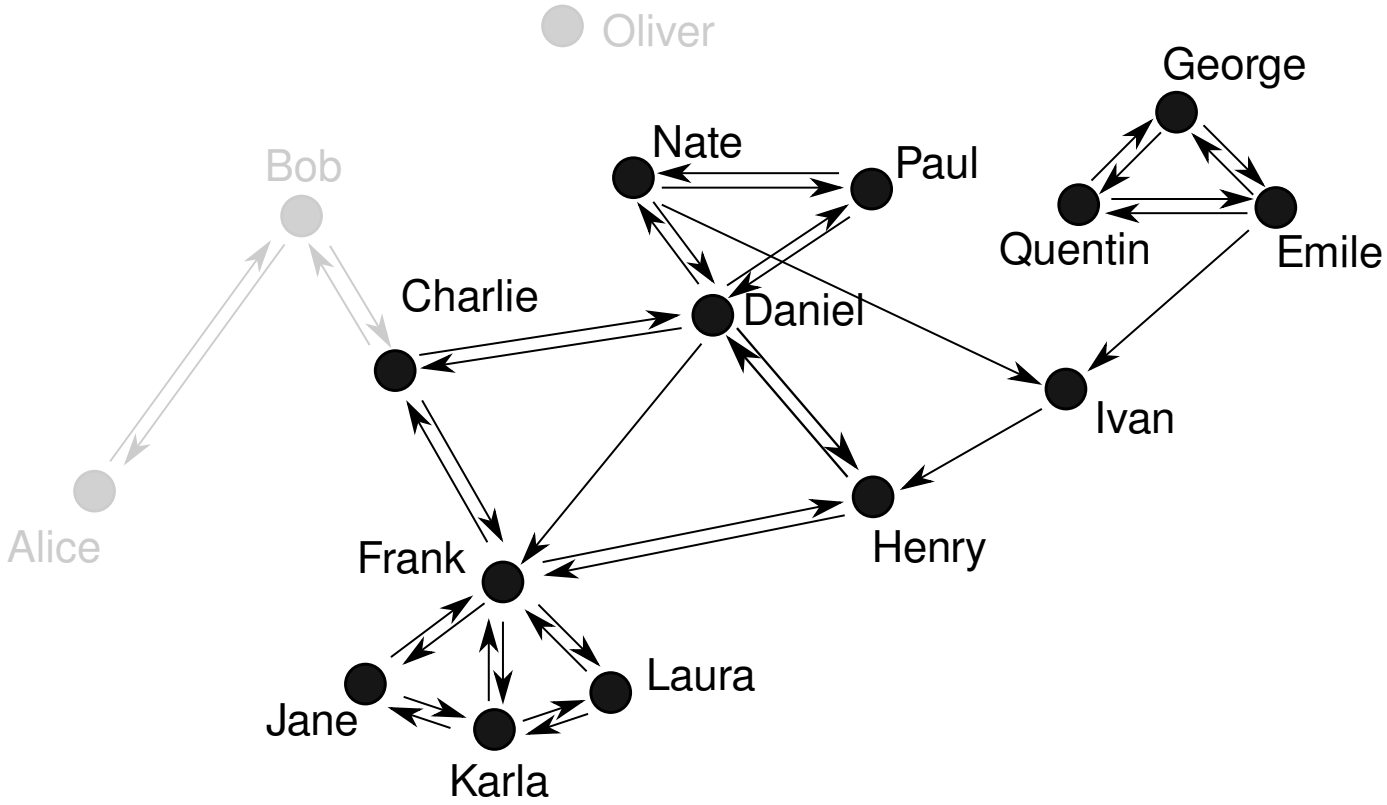
Web of Trust (WoT): Communities

If signing follows social relationships: communities form



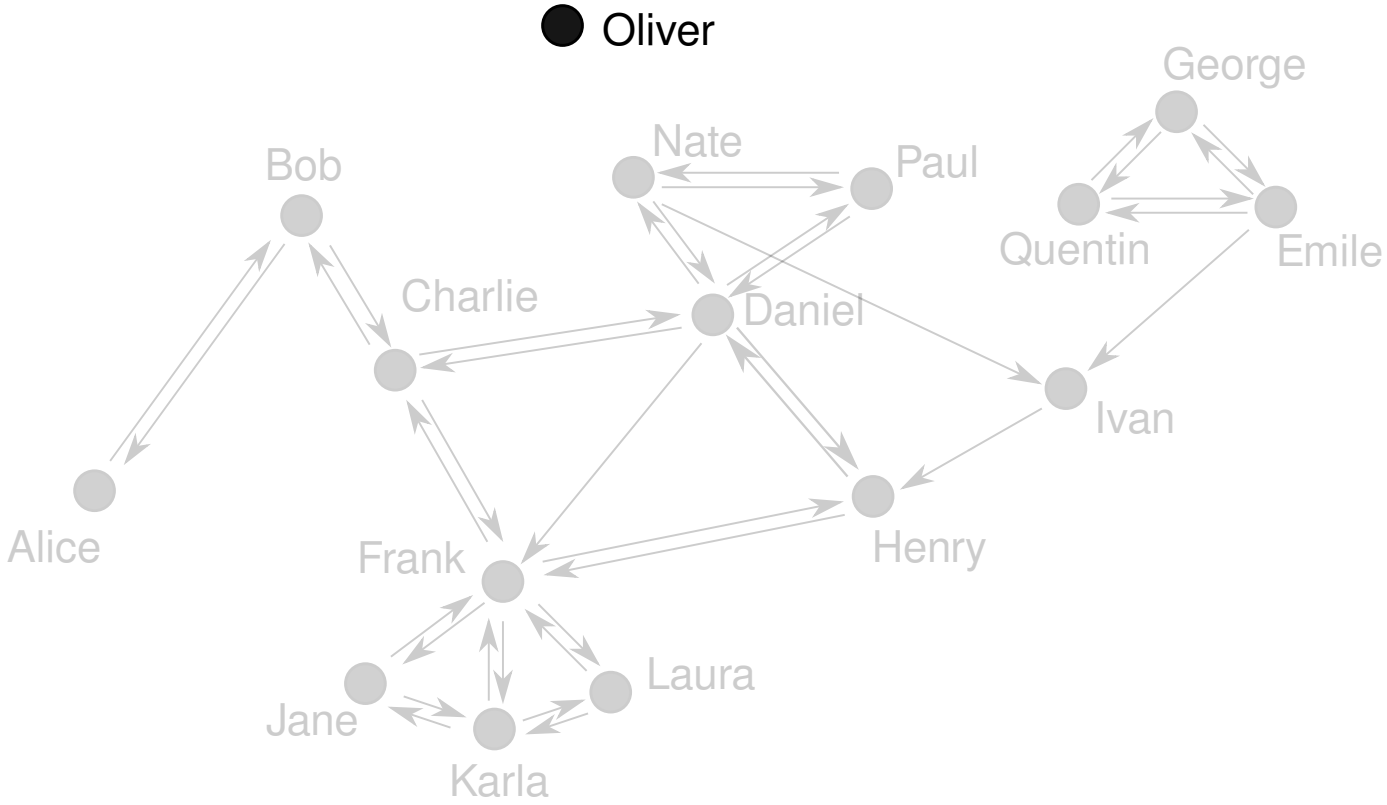
Web of Trust (WoT): Linked Communities

Communities may be linked via some social links again



Web of Trust (WoT): Isolated Islands

Disconnected ‘islands’ may also exist



OpenPGP Web of Trust

Certification

- Public/private key pair for every user: e.g. `pub 2048R/69B003EF`
- User ID: `[Ralph Holz, <holz@net.in.tum.de>]`
- Issue a certificate = `sign(User ID, public key)`

Web of Trust (WoT)

- Network of *key servers* to upload keys
- Uses the Synchronizing Keyservers (SKS) protocol
- Mechanism of creation: personal contact, key signing parties (but not fully known)
- Complete history of the network (SKS knows no ‘delete’ operation!)

Trust in OpenPGP (1)

Owner Trust

- Alice: “I trust Bob [*very much*/*somewhat*/*not*] to properly identify a person before signing.”
- Private assessment – stored *locally*

Used to determine ‘valid keys’

- GnuPG comes with adjustable default settings
- Path length ≤ 5
- Either ‘full’ trust in all owners on path
- Or ≥ 3 distinct paths with ‘marginal’ trust in owners

Trust in OpenPGP (2)

Public Key Trust

- A second form of trust in OpenPGP
- Corresponds to this statement made by Alice:
  'I have checked [*carefully*/*somewhat*/*not very much*] that this is Bob's key.'
- Stored *as part of signature*
- Most often used with a default setting

OpenPGP Trust Metrics

OpenPGP favours a user-centric model

- Focus is on owner trust:
  - Either ‘full’ trust in all owners on certification path
  - Or at least 3 distinct paths with ‘somewhat’ trusted owners
  - But never a path length > 5!
- Does not scale to ‘Find Paths to All Possible Keys’
- Best used in a local ‘neighbourhood’
- This is also the intended use

Part 2:
Investigation of the current OpenPGP WoT

Deriving Requirements

A good WoT should...

- have certification paths between many (all) keys
  - else it is not useful
- have short certification paths
  - less entities to trust
  - chances of accurately assessing key authenticity
- have redundant paths between keys
  - beneficial for GnuPG trust metric
- be robust
  - removal of a key must have little impact on reachability
- capture social relations between users well
  - trust assessment is easier in communities

Deriving Requirements

A good WoT should...

- have certification paths between many (all) keys
  - else it is not useful
- have short certification paths
  - less entities to trust
  - chances of accurately assessing key authenticity
- have redundant paths between keys
  - beneficial for GnuPG trust metric
- be robust
  - removal of a key must have little impact on reachability
- capture social relations between users well
  - trust assessment is easier in communities

Deriving Requirements

A good WoT should...

- have certification paths between many (all) keys
  - else it is not useful
- have short certification paths
  - less entities to trust
  - chances of accurately assessing key authenticity
- have redundant paths between keys
  - beneficial for GnuPG trust metric
- be robust
  - removal of a key must have little impact on reachability
- capture social relations between users well
  - trust assessment is easier in communities

Deriving Requirements

A good WoT should...

- have certification paths between many (all) keys
  - else it is not useful
- have short certification paths
  - less entities to trust
  - chances of accurately assessing key authenticity
- have redundant paths between keys
  - beneficial for GnuPG trust metric
- be robust
  - removal of a key must have little impact on reachability
- capture social relations between users well
  - trust assessment is easier in communities

Deriving Requirements

A good WoT should...

- have certification paths between many (all) keys
  - else it is not useful
- have short certification paths
  - less entities to trust
  - chances of accurately assessing key authenticity
- have redundant paths between keys
  - beneficial for GnuPG trust metric
- be robust
  - removal of a key must have little impact on reachability
- capture social relations between users well
  - trust assessment is easier in communities

Deriving Requirements

A good WoT should...

- have certification paths between many (all) keys
  - else it is not useful
- have short certification paths
  - less entities to trust
  - chances of accurately assessing key authenticity
- have redundant paths between keys
  - beneficial for GnuPG trust metric
- be robust
  - removal of a key must have little impact on reachability
- capture social relations between users well
  - trust assessment is easier in communities

Our Questions (Problem Statement)

Analyze the Web of Trust's graph w.r.t.

- Macro structure
  - How can users profit from the WoT?
- Usefulness to users
  - How effectively can the WoT used?
- Robustness
  - How does the WoT react to changes?
- Further Aspects
  - Social structures? Crypto algorithms?

Our Questions (Problem Statement)

Analyze the Web of Trust's graph w.r.t.

- Macro structure
  - How can users profit from the WoT?
- Usefulness to users
  - How effectively can the WoT used?
- Robustness
  - How does the WoT react to changes?
- Further Aspects
  - Social structures? Crypto algorithms?

Our Questions (Problem Statement)

Analyze the Web of Trust's graph w.r.t.

- Macro structure
  - How can users profit from the WoT?
- Usefulness to users
  - How effectively can the WoT used?
- Robustness
  - How does the WoT react to changes?
- Further Aspects
  - Social structures? Crypto algorithms?

Our Questions (Problem Statement)

Analyze the Web of Trust's graph w.r.t.

- Macro structure
  - How can users profit from the WoT?
- Usefulness to users
  - How effectively can the WoT used?
- Robustness
  - How does the WoT react to changes?
- Further Aspects
  - Social structures? Crypto algorithms?

Our Questions (Problem Statement)

Analyze the Web of Trust's graph w.r.t.

- Macro structure
  - How can users profit from the WoT?
- Usefulness to users
  - How effectively can the WoT used?
- Robustness
  - How does the WoT react to changes?
- Further Aspects
  - Social structures? Crypto algorithms?

Let's start:
Obtaining our dataset

Used Dataset

Obtained full snapshot of SKS database of late 2009

- Stored relevant key properties in SQL DB
- Snapshot contains complete history of network
- Time stamps of key creation, signatures, expiry, revocations, ...

Unknowns

- Unknown number of non-public (not published) signatures
- Unknown number of really active users

Resulting Key Set

Many keys available on the servers

All keys	2.7 millions
Expired, revoked, broken keys	570,000

But not many used for signatures

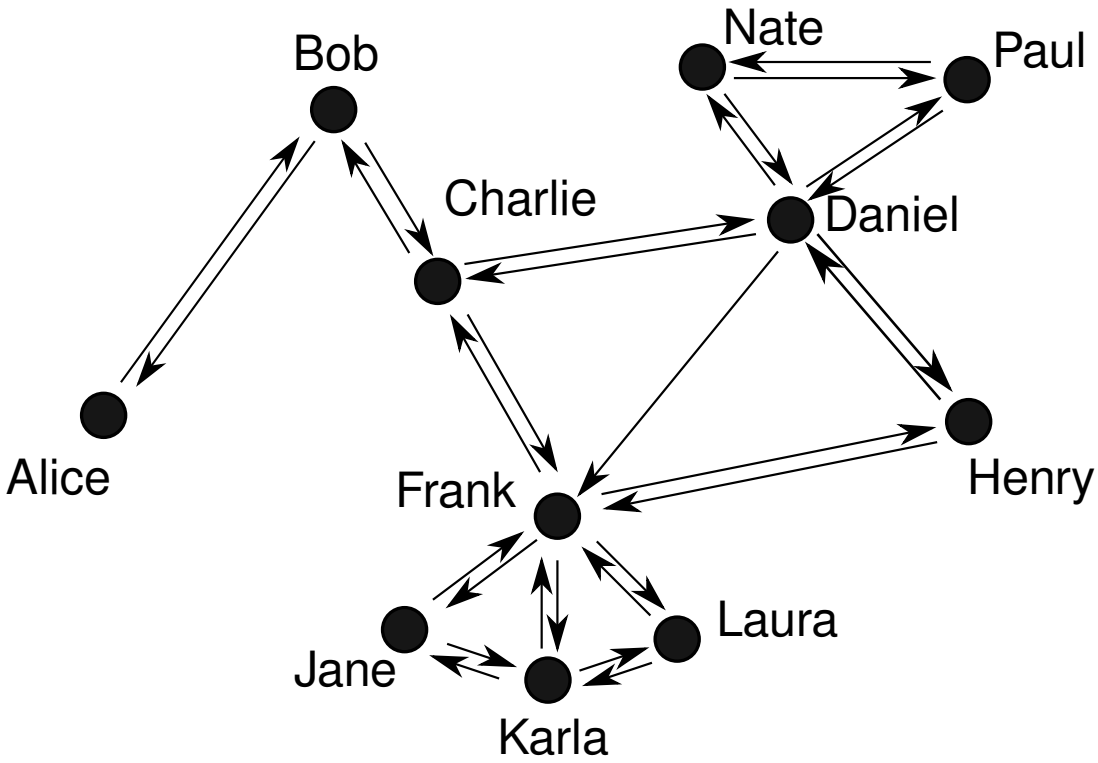
Keys with incoming or outgoing signatures	325,000
Resulting signatures	817,000

Majority of available keys are not verifiable: no signature chains.

Macro Structure

Macro Structure

Strongly Connected Components (SCCs)



Within an SCC, there is ≥ 1 signature chain between any key pair.

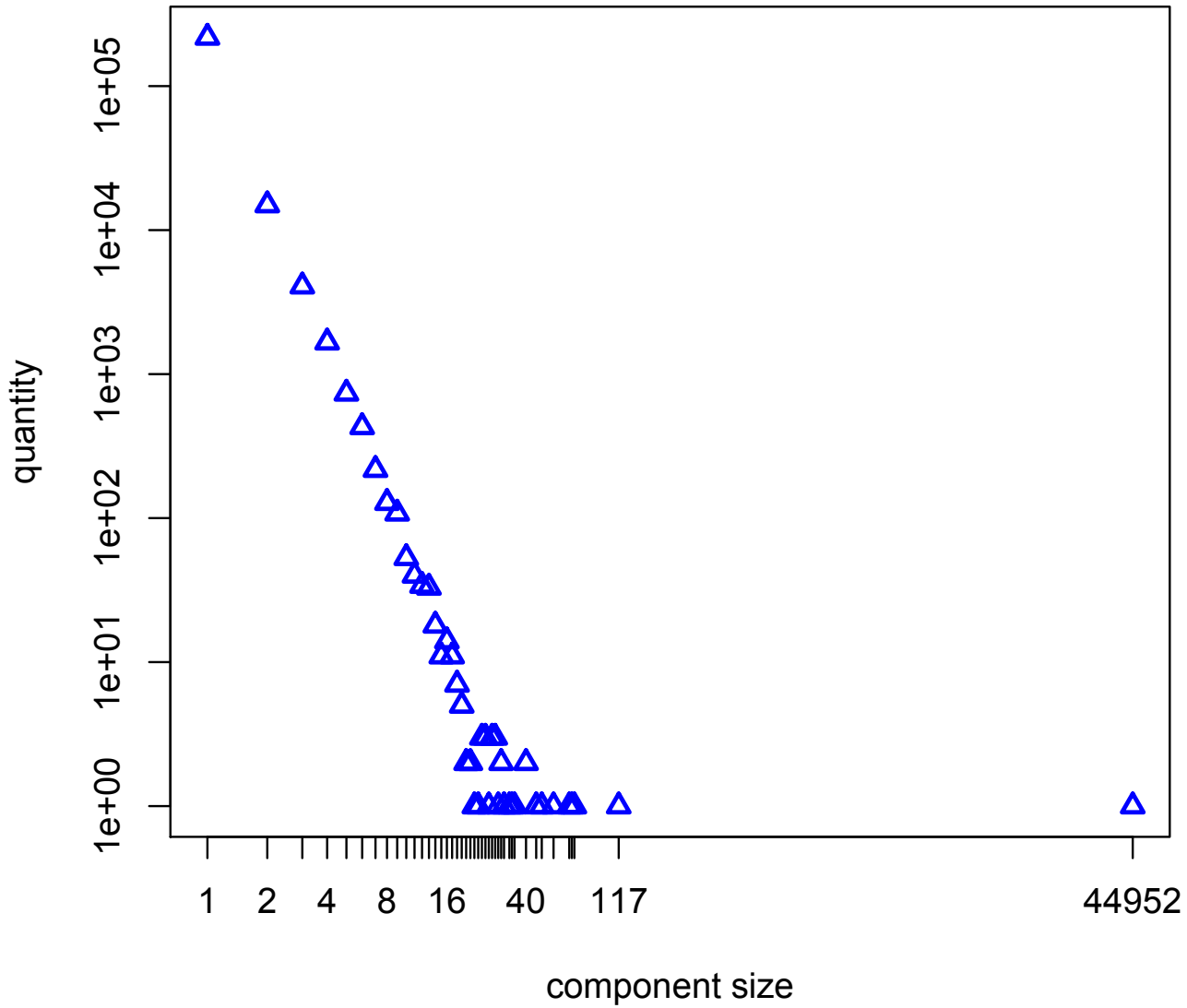
Macro Structure

SCCs are important:
mutual authentication only within the same SCC

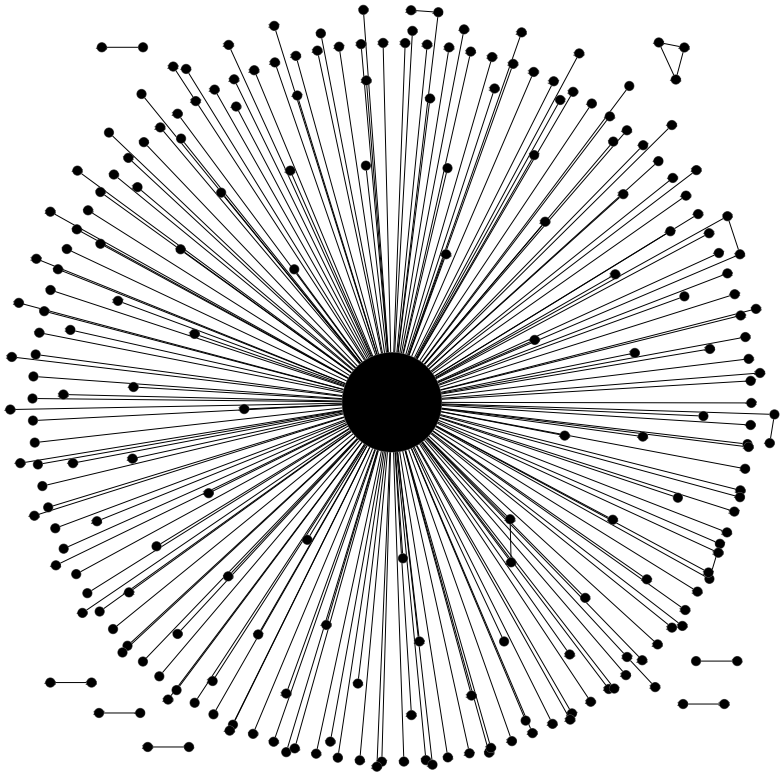
SCCs in the Web of Trust

- Largest SCC (LSCC) of just **45,000** keys (!)
- But there are **240,283** SCCs...
- ... $>$ 100,000 are single nodes (trivial sub-graphs)
- ... $\approx$ 10,000 node pairs

Macro Structure: SCC Sizes



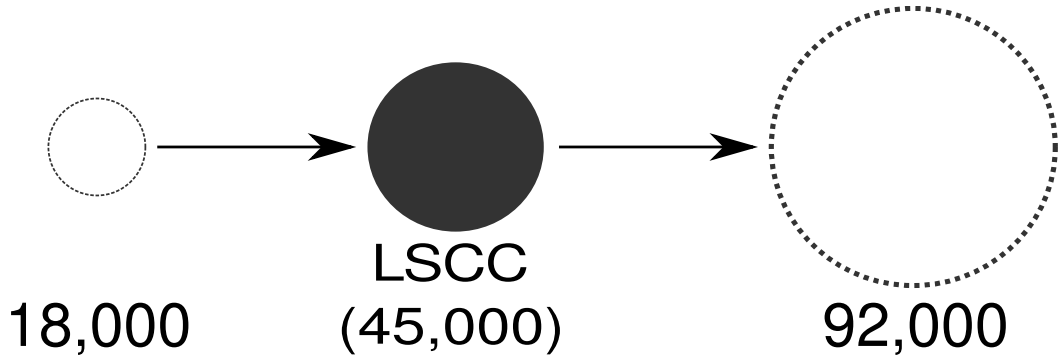
Macro Structure: SCCs and LSCC



SCCs of size > 8 – LSCC in the middle

Macro Structure: Pecularities

Links in/out of LSCC (uni-directional!)



Certification Authorities

- Prominent: Heise, CACert and DFN-Verein (4,200 keys signed in LSCC)
- Heise signed 21,000 keys outside LSCC, too

Impact on Usability

2.7m keys – just 45,000 really profit from the WoT

Significant user activity only in LSCC

- Ratio edges/nodes in LSCC is 9.85,
  and in whole WoT 2.51
- Most users in smaller SCCs cannot verify keys in the WoT
- Recommendation to new users:
  - Get a signature from someone in the LSCC
  - Get a signature from a CA

Focusing on LSCC

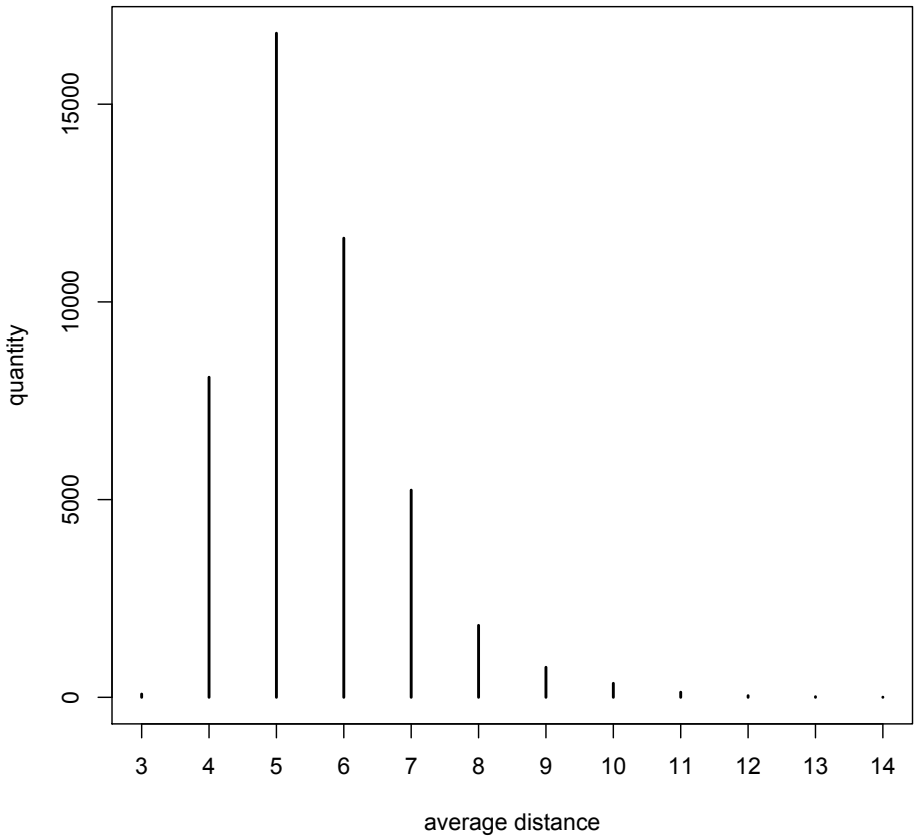
The remainder of this talk will focus on the LSCC

We investigate

- Usefulness (distances, paths, clustering)
- Robustness
- Communities

Usefulness: Distances and Node Degrees

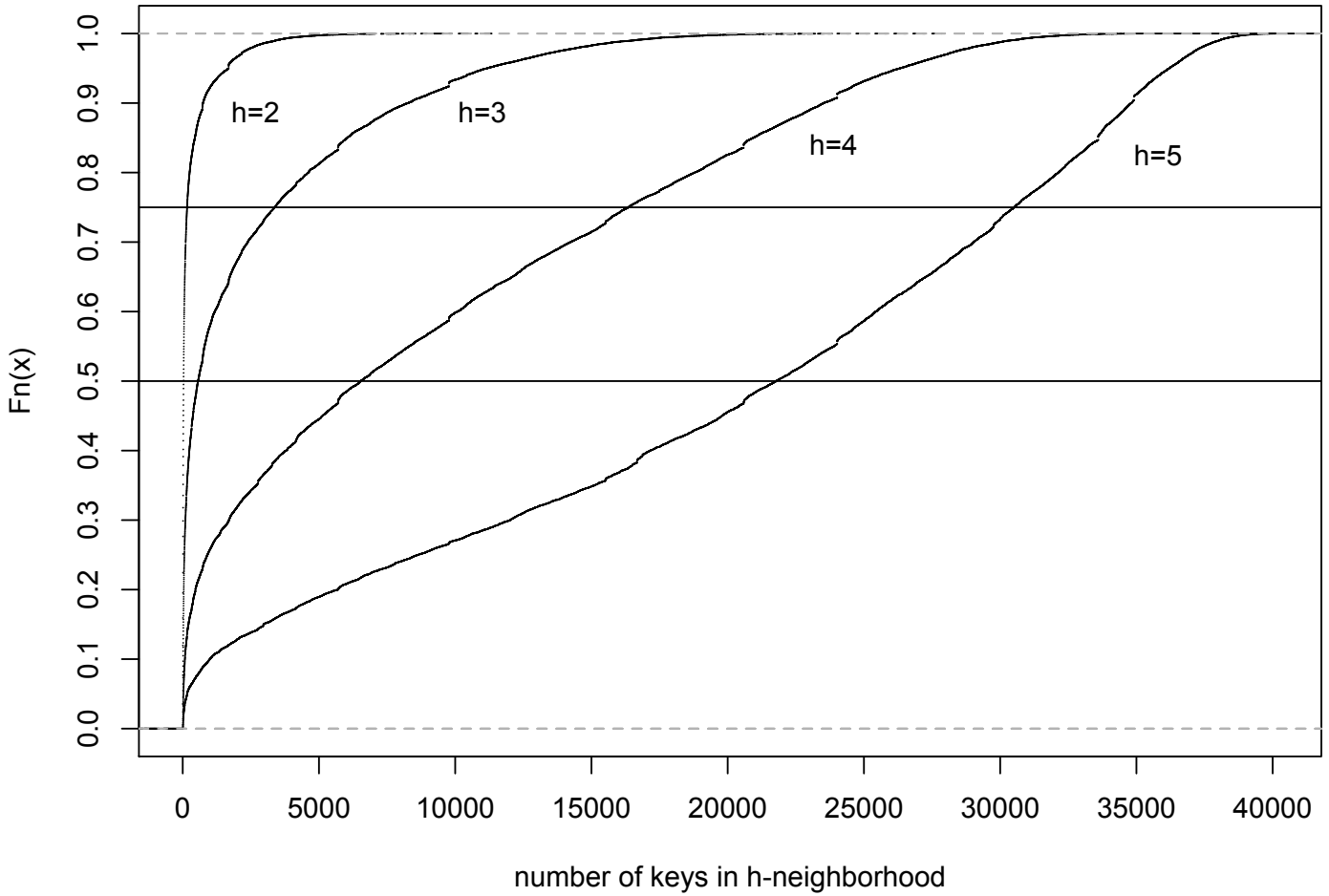
Distances in the WoT



- Distance between two nodes is length of shortest path
- Recall: GnuPG's path limit is 5

Nodes reachable via 1,..., 5 hops

CDF for 1-, 2-, ..., 5-neighborhoods



Nodes reachable via 1,..., 5 hops

The LSCC is well meshed

- 2-neighborhood (2 hops)
  - Mostly very small neighborhood
  - Very few keys can reach a few hundred keys
- 5-neighborhood (5 hops)
  - 50% chance that a key can reach $\leq$ 22,000 keys
  - Some keys can reach up to almost 38,000 keys

Significance

- Good finding: path lengths not a problem
- But recall: availability of paths is important, too

Node Degrees

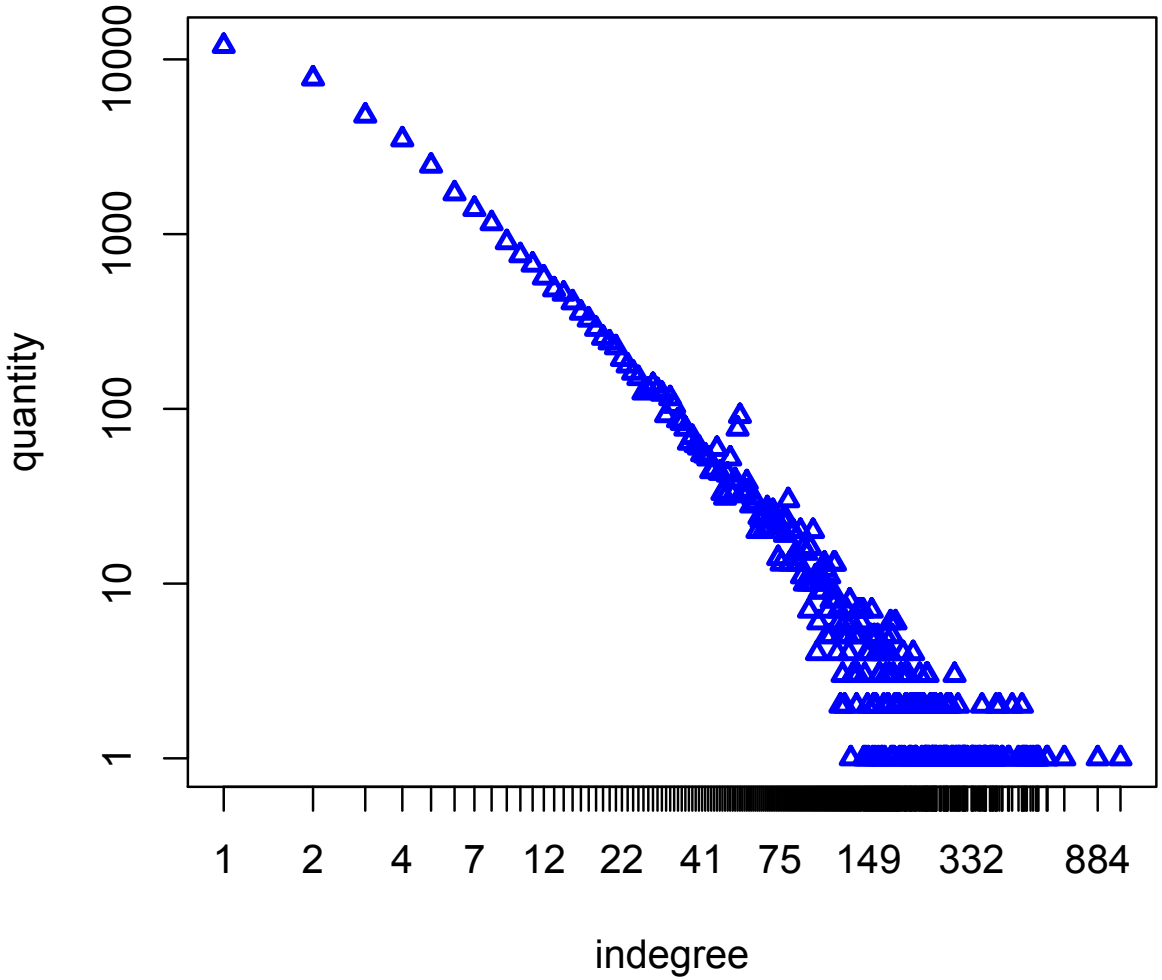
GnuPG views redundant paths as beneficial

- High indegree: key more likely to be verifiable
- High outdegree: higher likeliness of redundant paths

Mutual signatures are also beneficial

- Improves overall verifiability of keys
- Strengthens indegree and outdegree

Node Indegrees



Note: Outdegrees have practically the same distribution

Majority of nodes: low in/outdegree

This is a *bad* finding

- Almost half of keys have indegree 1 or 2
- About $1/3$ of nodes have outdegree 1 or 2
- Most nodes cannot use redundant paths
- Mutual signatures: only in 50% of cases...

This means: redundant paths are too rare

- Verify another key: needs direct signatures
- Be verifiable: only via very few other keys

Robustness:
Resilience Against Change

Robustness

What happens when keys expire, are revoked, ...

- Paths over these keys become invalid
- Simulated this by randomly removing nodes

Targeted attacks...

- Difficult: either compromise the key...
- ... or delete it on all SKS servers
- Simulated this: remove nodes with high degree first

Is the LSCC a Scale-free Graph?

Scale-free graphs...

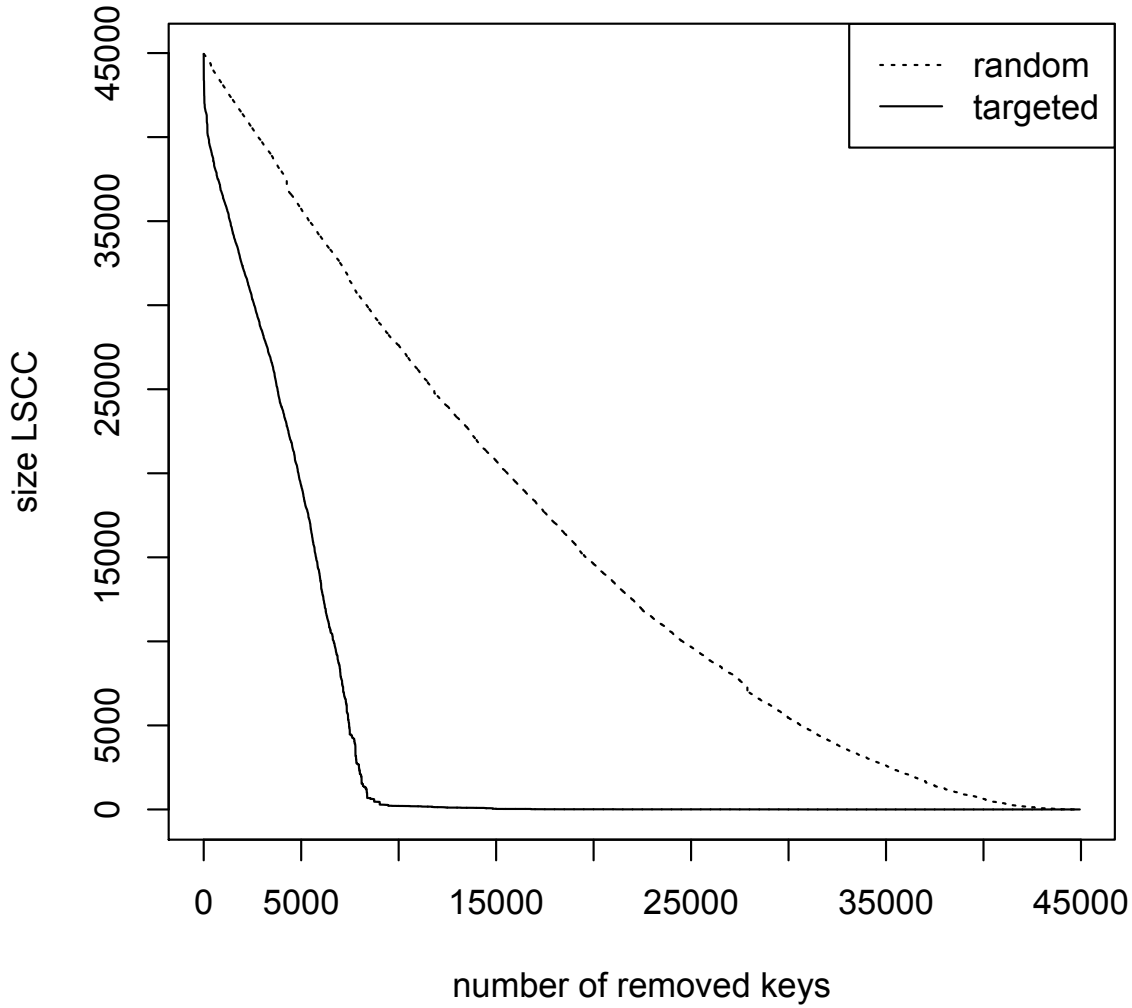
- ... strong hub structure, node degrees follow Power Law, i.e. distribution of node degree follows power law: $P(k) \propto k^{-\gamma}$
- ... robust against random removal, sensitive to targeted removal of nodes

The LSCC is *not* scale-free

- (Clauset, 2009) recommend Maximum-Likelihood + Kolmogorov-Smirnov test
- The values we obtained rule out Power Law

But similar: many inter-connected hubs

Remove keys, recompute LSCC size



Removing keys

Random removal (expiry, revocation, ...)

- Very robust
- Need to remove $1/3$ of keys to cut LSCC by half

Targeted removal (attack)

- Quite robust – decay not too bad
- Remove all nodes of degree:
  - $>$ 160 ($\approx 0.5\%$ of nodes) $\rightarrow$ LSCC shrinks to 88%
  - $>$ 18 ($\approx 11\%$ of nodes) $\rightarrow$ LSCC shrinks to 50%

Removing keys

Assume CA keys are compromised/revoked

- The LSCC does not care: new size at 94.4%
- Average distances stay the same
- Many paths around the CAs:
  they are not critical components

Key removal is not an efficient attack

- There are *many* hubs, and they are inter-connected
- Not a typical scale-free network

A very good finding for a WoT

Further Aspects

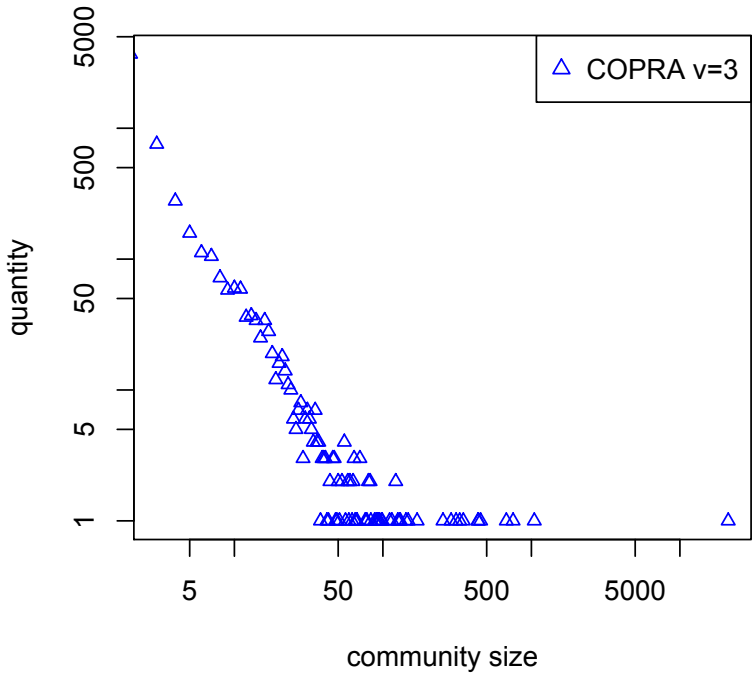
Communities

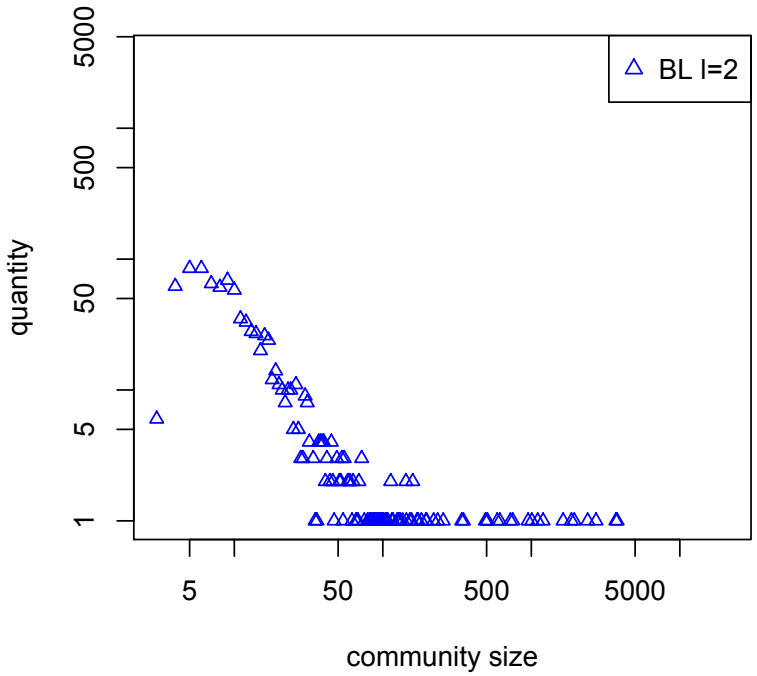
Analysis of community structure

- The LSCC shows a clear Small World Effect
- Used two algorithms for community detection
- Findings:
  - Very strong community structure
  - Communities often dominated by a top-level domain
  - Second-level domains less clearly identifiable

Distribution of Communities by Size

We tried two methods: COPRA and Blondel et al. (BL)





Communities Dissection

COPRA and BL: 94% vs. 99% of nodes in communities of size > 3.

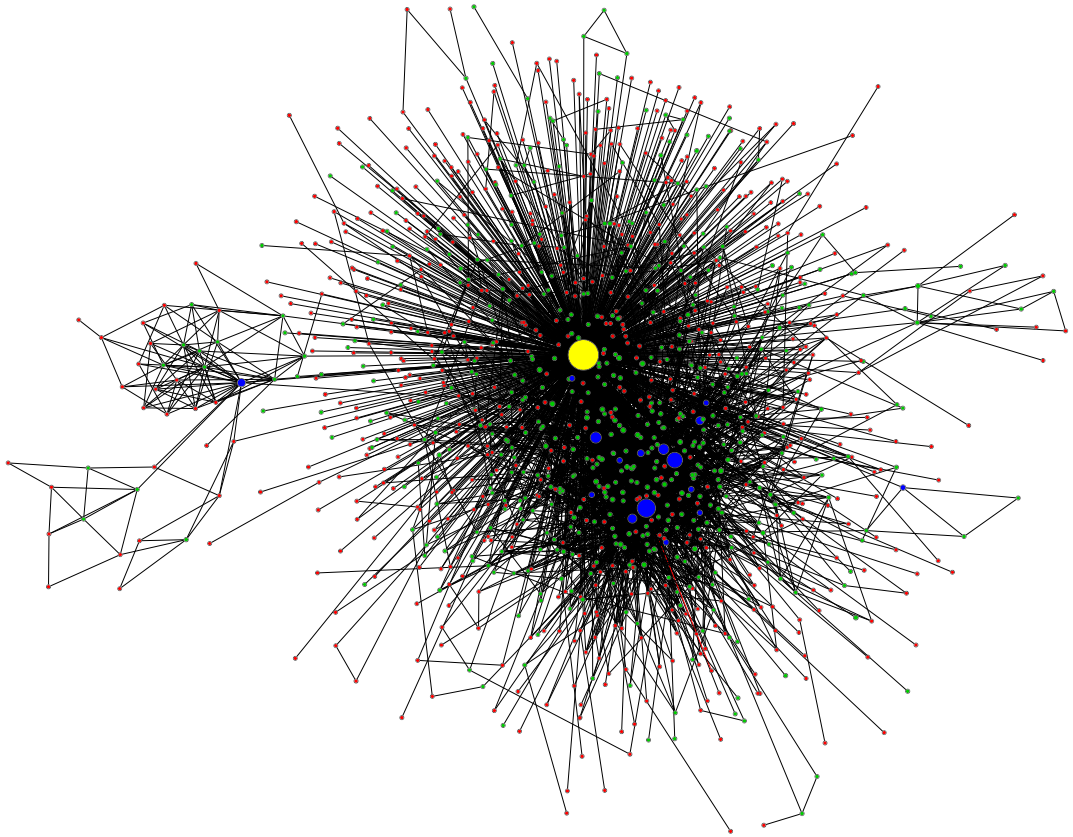


Figure : COPRA dissection for communities > 5.

Mapping to DNS and Events

Problem: little information about social membership in User IDs

- Option 1: group by Top-Level Domains and Second-Level Domains
- Option 2: group by signatures within 30 days

Mapping to DNS and Events

Quick results (details in paper):

- Question: how often are 80% of User IDs in a community in the same TLD?
- Very often: 47%-58%, depending on detection algorithm
- Picture changes entirely for SLDs: only 13%

Quick results (details in paper):

- Top Level Domains communities is dominated by a TLD (COPRA: 58%, BL: 47%)
- Case 1: 80% of nodes of a community are in same TLD/SLD (‘dominated’)
- Case 2: 40% of nodes of a community are in same TLD/SLD (‘assignable’)

Mapping to DNS and Events

Results

- Large percentage of communities is dominated by a TLD (COPRA: 58%, BL: 47%)
- Of the *remaining* communities, many are assignable (COPRA: 38%, BL: 47%)
- Without generic TLDs: similar. E.g., COPRA: 38% dominated; 23% assignable

Picture changes entirely for SLDs:

- E.g., COPRA: 13% dominated, 30% assignable

Signatures within 30d

- Inconclusive. COPRA: 40% of communities; BL: 14%.

Tentative Conclusions w.r.t. Communities

Difficult to reach compelling conclusions

- Algorithms agree that pronounced community structure exists
- Mapping to TLDs works OK, but not for SLDs

Consider: there is a huge number of TLDs and SLDs

- Signing process is supported by social links (that's good)
- Current algorithms yet too imprecise for better analysis
- Might be worthwhile to follow up on this

Crypto strength

Algorithms in LSCC

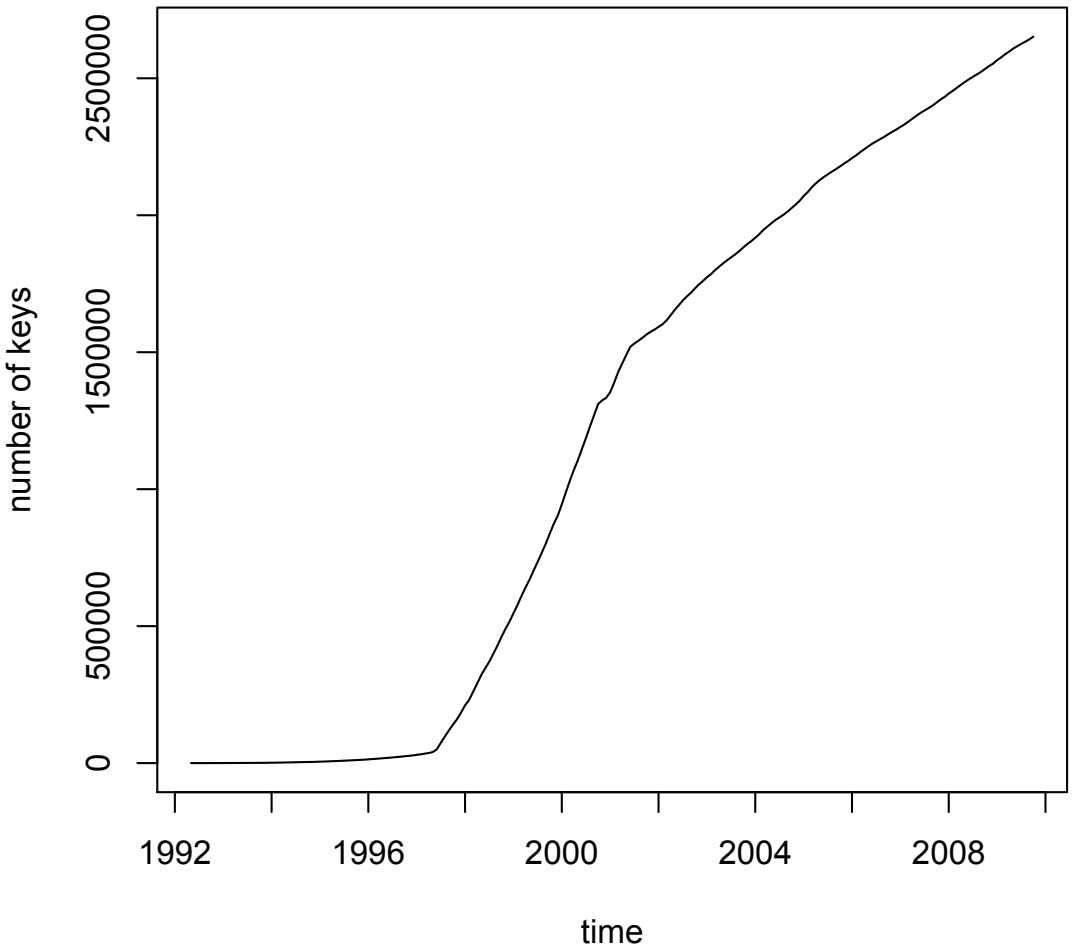
Hash Algorithm	Occ.
SHA1	89.36%
MD5	9.34%
SHA256	1.12%

Key Algorithm	Occ.
DSA-1024	81.32%
RSA-1024	8.68%
RSA-2048	5.36%

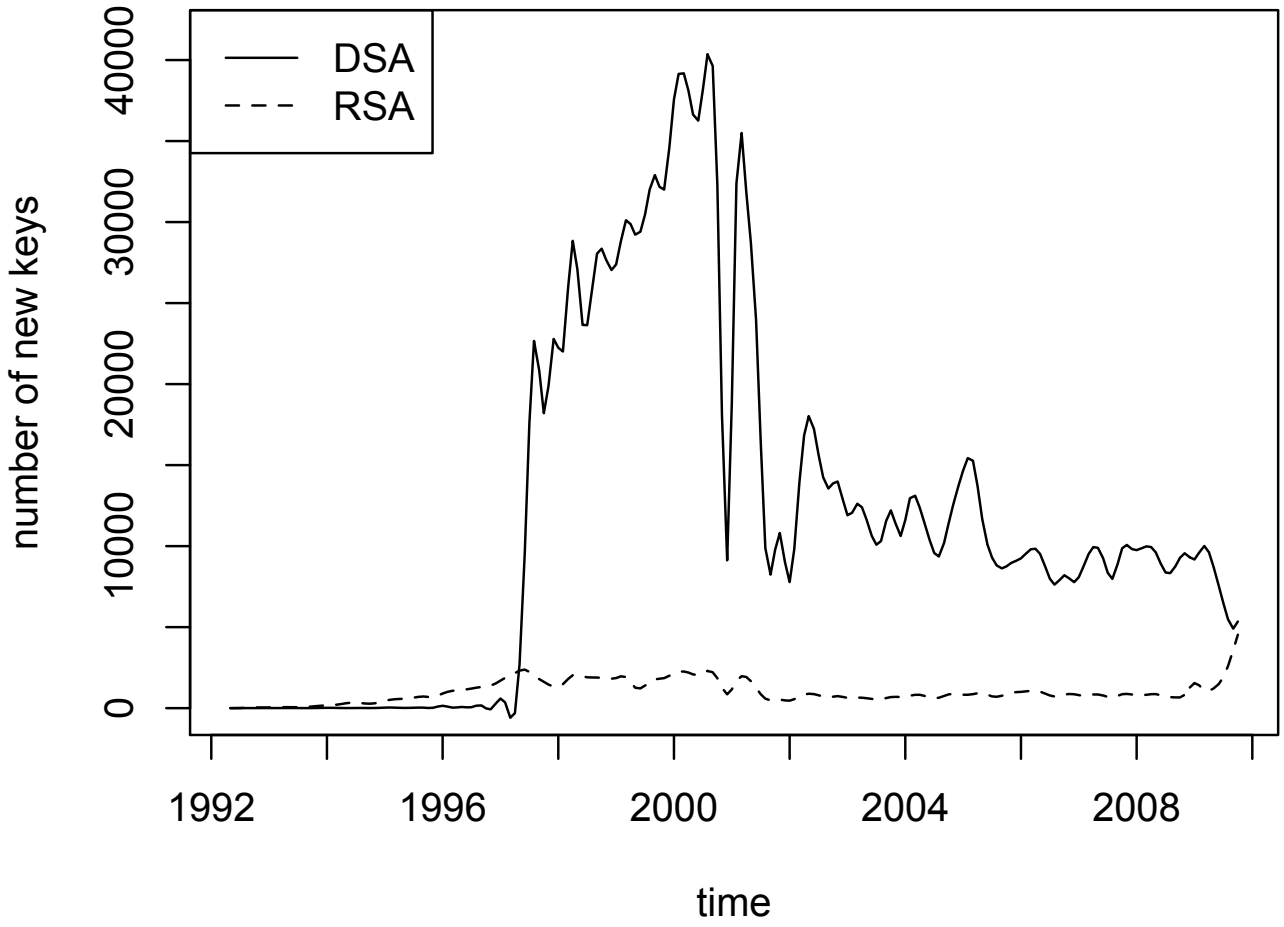
Not too much to criticize here

- Some RSA keys of $\leq$ 1,024 bit are well-connected
- Length of < 768 bit occurs ≈ 500 times (problematic)
- 1,024 bit not a problem today, but maybe tomorrow
- Thankfully, few MD5-based signatures

Number of keys in WoT and LSCC



RSA and DSA keys



Related Work

Capkun et al., 2001

- LSCC at 12,000 keys only; claims Small-World Effect and Power Law distribution

Arenas et al., 2004

- Investigated network as undirected graph
- Degree distribution, clustering: Power Law
- Community Dissection: also claim Power Law

wotsap, Penning

- Continous snapshots and some statistics of LSCC
- Distances, degree distribution, robustness
- Less in-depth; wotsap extraction algorithm is faulty

Conclusions

Conclusions

We have found light and dark

- Macro structure
  - ☹ Only users in LSCC really profit from WoT
  - ☺ CAs are useful, but not critical
- Usefulness
  - ☺ Good reachability via ≤ 5 hops
  - ☹ Redundant paths too rare!
- Robustness
  - ☺ Very robust against expiration, revocation, ...
  - ☺ Key removal is not an efficient attack

WoT works well in ‘close neighborhoods’ of active nodes – but not otherwise.

Conclusions

We have found light and dark

- Macro structure
  - ☹ Only users in LSCC really profit from WoT
  - ☺ CAs are useful, but not critical
- Usefulness
  - ☺ Good reachability via ≤ 5 hops
  - ☹ Redundant paths too rare!
- Robustness
  - ☺ Very robust against expiration, revocation, ...
  - ☺ Key removal is not an efficient attack

WoT works well in ‘close neighborhoods’ of active nodes – but not otherwise.

Conclusions

We have found light and dark

- Macro structure
  - ☹ Only users in LSCC really profit from WoT
  - ☺ CAs are useful, but not critical
- Usefulness
  - ☺ Good reachability via ≤ 5 hops
  - ☹ Redundant paths too rare!
- Robustness
  - ☺ Very robust against expiration, revocation, ...
  - ☺ Key removal is not an efficient attack

WoT works well in ‘close neighborhoods’ of active nodes – but not otherwise.

Conclusions

We have found light and dark

- Macro structure
  - ☹ Only users in LSCC really profit from WoT
  - ☺ CAs are useful, but not critical
- Usefulness
  - ☺ Good reachability via ≤ 5 hops
  - ☹ Redundant paths too rare!
- Robustness
  - ☺ Very robust against expiration, revocation, ...
  - ☺ Key removal is not an efficient attack

WoT works well in ‘close neighborhoods’ of active nodes – but not otherwise.

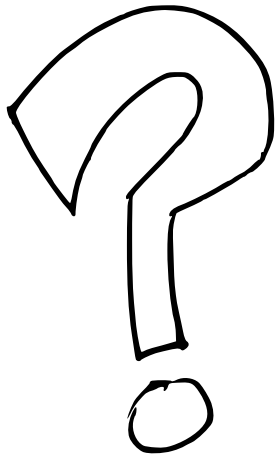
Conclusions

We have found light and dark

- Macro structure
  - ☹ Only users in LSCC really profit from WoT
  - ☺ CAs are useful, but not critical
- Usefulness
  - ☺ Good reachability via ≤ 5 hops
  - ☹ Redundant paths too rare!
- Robustness
  - ☺ Very robust against expiration, revocation, ...
  - ☺ Key removal is not an efficient attack

WoT works well in ‘close neighborhoods’ of active nodes – but not otherwise.

That was PKI. Thank you!



- Download datasets from `pki.net.in.tum.de`
- We encourage work to repeat our investigations of X.509 and OpenPGP
- May be suitable for IDP or BSc thesis?