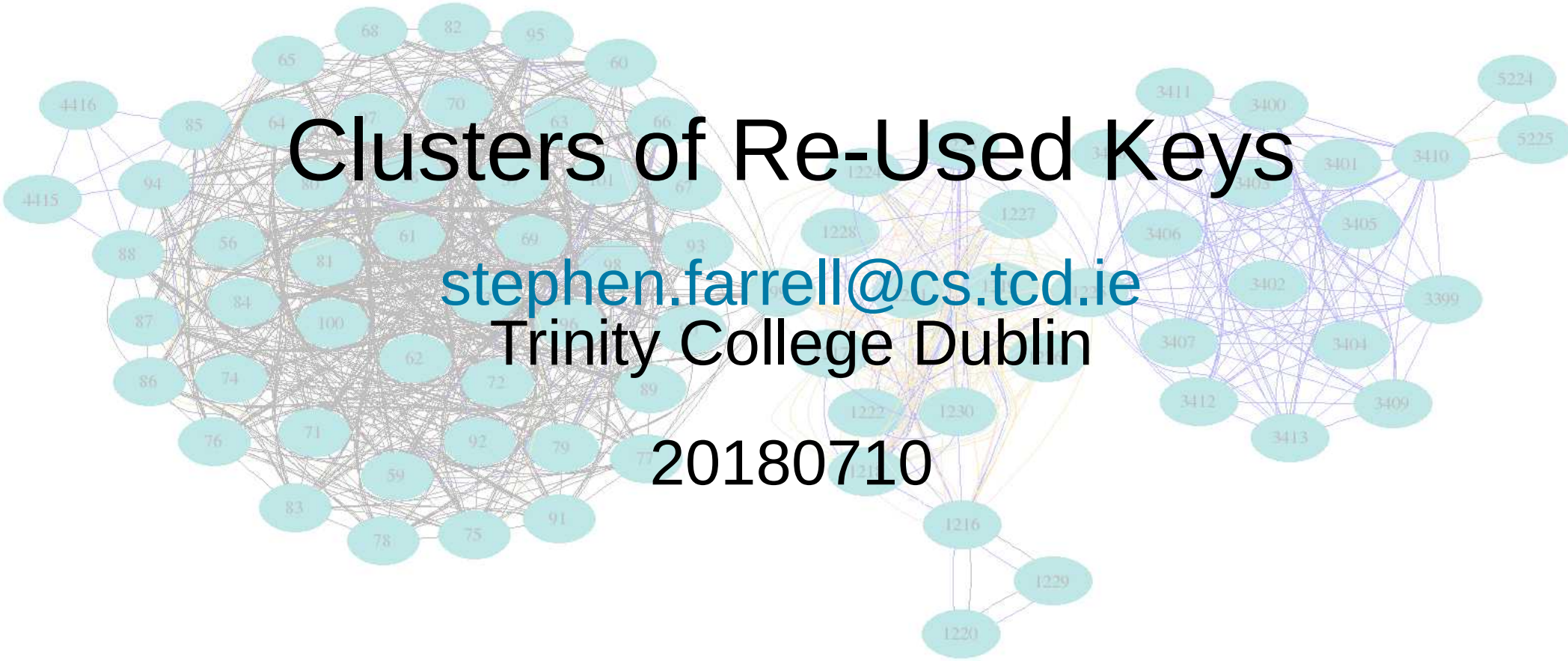


Clusters of Re-Used Keys

stephen.farrell@cs.tcd.ie
Trinity College Dublin

20180710

The TL;DR slide

- Me: Trinity College Dublin, School of Computer Science and Statistics
  - Research topics: security, privacy, delay-tolerant networking
- This talk: (small) country-scale network scans show that there are a **lot** of cryptographic host/server keys for mail, web and SSH services being **re-used** on a **lot** of different IP addresses, which is surprising
  - Around 20180316, if you used TLS/SSH via a standard SSH/TLS port with a randomly chosen mail server in Ireland, (so that host also listens on port 25), then the probability that some other IP address in Ireland shares a host/server key with that mail server was **>=53%**
- Preprint: https://eprint.iacr.org/2018/299
  Code: https://github.com/sftcd/surveys/
  Graphs: https://down.dsg.cs.tcd.ie/runs/
  This: https://down.dsg.cs.tcd.ie/misc/hark.pdf

Contents

- Background
- Methodology
- Results
  - Skip to slide #25 for annotated graphs
  - Skip to slide #35 for numbers
- Causes/Mitigations
- Future Work and Conclusions

Background

- I've been wondering whether it's possible to help improve Internet security and privacy based on local actions and measurements, compared to Internet-scale action and measurement
- We had a workshop in Dublin in Sep 2017
  - https://responsible.ie/
- One outcome there was that local (Irish) Internet measurement should have low-hanging fruit
  - So I set a student project...

Student Project

Develop a useful survey tool of Irish mail server deployments
censys.io collate Internet-scale surveys and make the results of those public via their web site and an API. A quick search there says there are 12582 email servers in Ireland, of whom 62% do some form of STARTTLS for mail transport security. The goal here is to use the API to develop a tool that can be run periodically, that analyses that data and produces a list of email addresses to which one might send useful advice as to how they could improve their mail transport security. For example, if the data indicate some deployments are running a server that could easily be fixed, (say by just updating a certificate), then crafting a mail with deployment-specific instructions as to how to do that could be good. Actually contacting the postmasters involved is not a part of this project but may be done later based on results found.
https://studentprojects.scss.tcd.ie/#sfarrel6

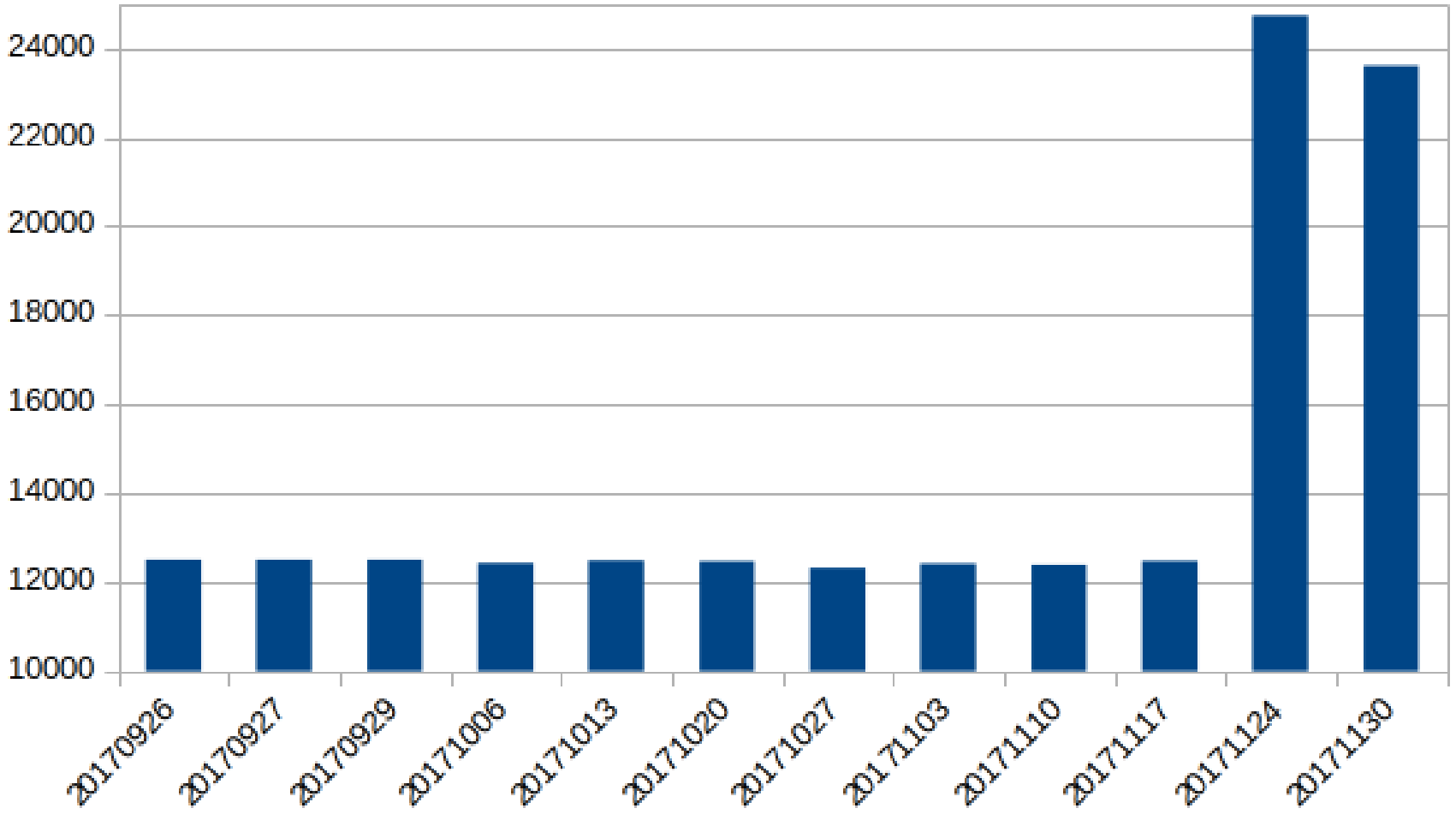
I started gathering project data…

- Using censys.io python client API
- Weekly cron job downloading details of mail servers (port 25 listeners) geo-located to Ireland
- Two odd things, and one less surprising thing seen…
  - Unsurprising: mega-scalers cause me problems:-)
  - Oddity#1: changes in number of listeners...
  - **Oddity#2: fewer keys than expected!**
    - Oddity#2 caused this work

Aside: Mega-scalers…

- I tend to not have any “live” accounts with mega-scalers – not 100% and not due to zealotry but I do always push back on things wanting me to create new accounts
- Censys.io went commercial in Dec 2017 and so wanted their users (whether paid or free-researchers) to use a mega-scaler’s storage
  - Entirely reasonable, but not for me:-)
- Result: I decided to do my own scans, which in the end was beneficial (for me)

Oddity#1: port 25 listener numbers...



Didn't investigate that jump (yet), could be due to GPDR preparation maybe?
Numbers haven't gone back down: 20180316: 24,774 port 25 listeners.

Oddity#2: Fewer keys seen...

- Counting keys in run IE-20171130…

```
$ HostPortKeyCount.py -f all-key-fingerprints.json
...lots of output…
hosts: 10657
hostsports: 25935
fps: 12889
```

- From the ~23.6k port 25 listeners, ~10.4k do some kind of crypto, with a total of **~25.9k host/port** combinations doing crypto, but we only see **~12.9k different keys**
  - Not a huge discrepancy, but worth a look...
- Ports checked: 22, 25, 110, 143, 443, (587,) 993
  - 587 wasn't part of 2017/Censys.io runs, but was added for later/local runs

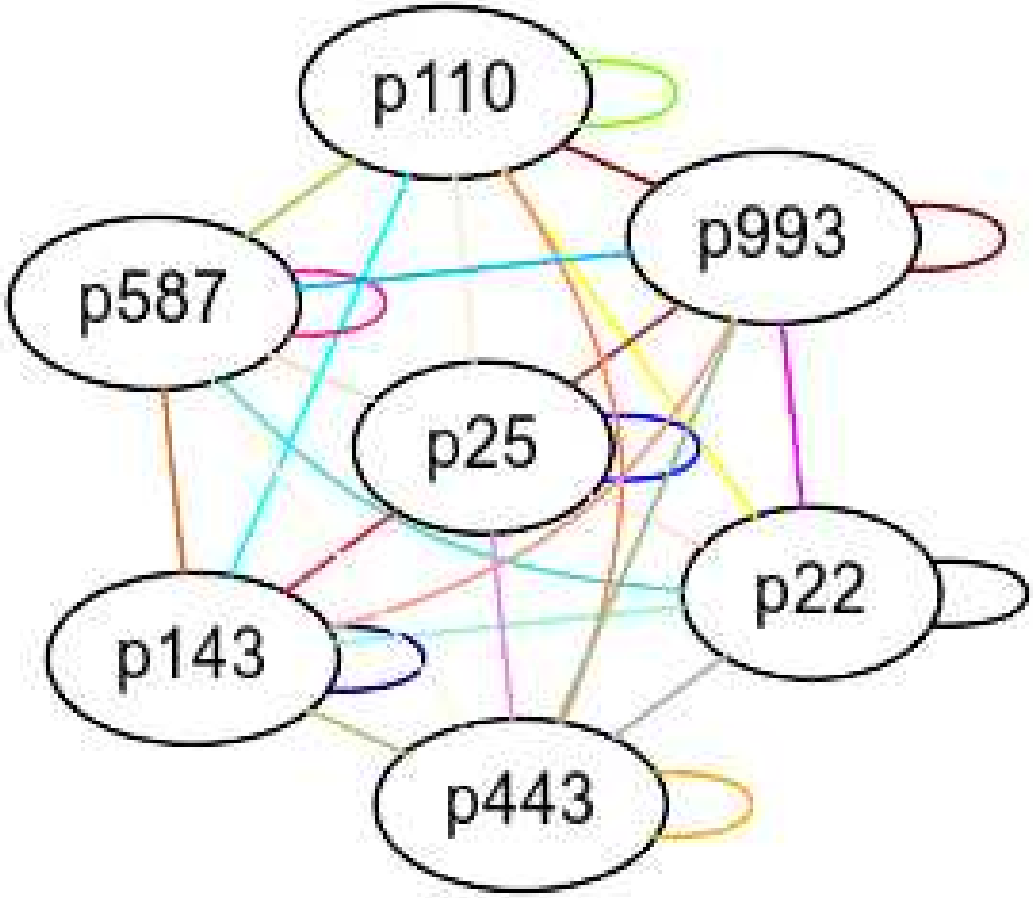
Clusters

- If there are fewer keys than services using keys, then clearly some keys are being re-used, e.g. same key for port 25 and 587 on one host is reasonable, less so on two hosts
- Examining the data, it became clear there was some structure visible in the key re-uses…
- Definition: If the same key is seen on two IP addresses that are part of a scan population, then those IP addresses are part of the same cluster
  - Regardless of the port/service with which the key is used
  - Single-address key re-uses do not contribute to clusters
- We see re-uses on **every** possible combination of port/services
  - **Except** we've yet to see any key shared between SSH and TLS

Causes...

- There are many possible causes for key re-use on different IP addresses (more later) but minimally, one could be mirroring the entire content/disk of a web server for redundancy
  - From an Internet vantage-point that looks the same as any other form of key re-use
  - Note that not all re-uses are as benign as this one…
- We can visualise that with a graph like this:
  - Nodes represent IP addresses
  - Node numbers are anonymised IP addresses
  - Node background colour represents the AS of the IP address
  - Edges represent key re-uses
  - Edge-colour represents the pair of ports,
    - e.g. black is for 22-22 (SSH)…

Legend for edge-colours



Issues with key re-use

- Not all re-uses are “bad” but there are risks…
  - Multi-homed hosts aren’t bad:-)
- Leaking keys allow masquerade
  - With RSA key transport, it’s worse
- Risk may increase faster than linearly with cluster-size
- Lots of related work - see pre-print and Heninger/ Holz etc
- Clusters and scale of scans here may be new-ish

“Safe” re-use

- Same listener process on same IP (e.g. ports 25/587)
- Multi-homed hosts with >1 IPv4 address
- Mirroring, as in earlier slide
- Some HSM use-cases, maybe for a top-of-rack HSM
- Multiple machines or VMs where the one and only admin has really thought it all through:-)
  - Some wildcard cert use-cases can fit this scenario
- Notes:
  - “Real” keys may not be visible via IPv4 address scanning, but only if e.g. a DNS name is supplied in a TLS SNI extension or HTTP Host header
  - In general, we can’t distinguish any of the above from less safe re-uses, from an Internet vantage point

Dangers of re-use (1)

- Leakage – most private keys are stored in files, enabling...
- Masquerade – anyone with a re-used private key can masquerade as other IPs in the same cluster, enabling...
  - Misdirected mail – if I have the right key, I can defeat even MTA-STS via usual MX/BGP tricks
  - Credential exposure – SSH password logins, IMAP/SMTP passwords could be captured and may reach beyond the cluster
  - Web origins – h2/ORIGIN frame/secondary cert proposals could enable spoofing any web server in the cluster from any other sharing the same key

Dangers of re-use (2)

- RSA key transport – if using old TLS ciphersuites, then offline decryption of sessions based on recorded ciphertext is easy
- Cross-protocol attacks may be more likely to succeed
- (Not sure if significant, but) Formal analyses of protocol security (e.g. for TLS1.3) didn't take large-scale key re-use into account
- Data recovery – if I can recover old key values from the disk on my new VM, then key re-use makes that vuln. more dangerous
- Laziness – re-using keys like this gives an impression of carelessness to the Internet

Dangers of re-use (3)

- Risk may increase as cluster-size increases…
- Let's assume:
  - Per-host cost of a leak is '***c***'
  - Probability of a leak of one key is '***p***'
  - Cluster size is '***n***':
- Cost of leaks without re-use for 'n' hosts: ~***c*p*n***
- Cost of leaks for cluster of 'n' hosts: ~***c*p*n²***
- Won't always apply, but seems correct in general that risk increases faster than linearly with key re-use

Related Work

- By no means 1st to survey keys, e.g.
  - Heninger, et al., “Mining your ps and qs: Detection of widespread weak keys in network devices.” 2012
  - Holz, et al., “TLS in the wild: An internet-wide analysis of tls-based protocols for electronic communication,” 2015.
  - More in preprint
- This work differs a bit by:
  - Looking at structure (clusters)
  - Small-country scale scans
  - Slightly different background/research questions

Methods

- 2017: Using censys.io; 2018: Ran zmap/zgrab locally
  - Port 25 listeners for IE, EE, FI, PT, LU, NZ, NA, UY, SI and SG
  - Then scan those for ports: 22, 25, 110, 143, 443, 587, 993
  - Usual zgrab SSH/TLS metadata stored (loadsa json;-)
- Analysis code:
  - Find clusters and make pretty pictures
  - Compare runs over time and cross-border
- Ethics: slowly scanning port 25 listeners isn't intrusive
  - Leave usual breadcrumbs in case someone objects

Methods (1)

- Different methods in 2017 and 2018
- 2017: Using censys.io, selected port 25 listeners in two similarly sized countries: IE (Ireland) and EE (Estonia), which gives zgrab output
- 2018: Locally use zmap to find port 25 listeners based on MaxMind prefixes for IE, EE, FI, PT, LU, NZ, NA, UY, SI and SG
- For each country: use zgrab to gather banner, SSH and TLS information from hosts for these ports:
  - 22, 25, 110, 143, 443, 587, 993
  - Port 587 wasn't included in censys.io 2017 scans

Methods (2)

- Use SHA256 fingerprint of SSH/TLS server keys seen in zgrab outputs to find clusters of re-used keys
  - Pretty obvious but we do see clusters “merging” during analysis runs, so >1 iteration over the data is needed
- Store names from reverse-DNS, certificates and banners, and matching A record, if present
- Store usual main TLS/SSH crypto parameters
  - Key type/size, cipher-suite, browser-trusted?, self-signed?, expired?
- Generate per-cluster JSON files, graphviz dot files and images (of graph and word-cloud of names)

Methods (3)

- Validation: my code or zmap or zgrab could have bugs
  - s/could/invariably does/ for my code at least:-)
- Developed tool to validate a cluster using different code base from scanning/analysis code:
  - Based on ‘ssh-keyscan’ and ‘openssl s_client’ instead of zgrab
  - Likely still some shared lower level library code but seems ok
- That, and manual inspection, shows clusters are real

Methods (4)

- Port 25 listeners are mail servers, so we're less likely to see sensitive data
- Slowly scanning for public services isn't intrusive, so we go slow
- We published PTR, TXT RR etc. findable from scanning host source IP
- We don't publish identifying information as that could help with lateral movement
  - That could change in future, but only on a case-by-case basis, e.g. for some product with hard-coded keys if vendor unresponsive
  - Happy to take advice on that aspect as I'm a first-timer for that
- Happy to send full information to asset-holders where we have a relevant contact
  - We anonymise identifying information for IP address that are linked to them, but are not theirs, except for the AS number of the linking IP
  - Have tooling to extract data from scans for a specific IPv4 prefix or ASN, that does the above anonymisation step for "outside" addresses

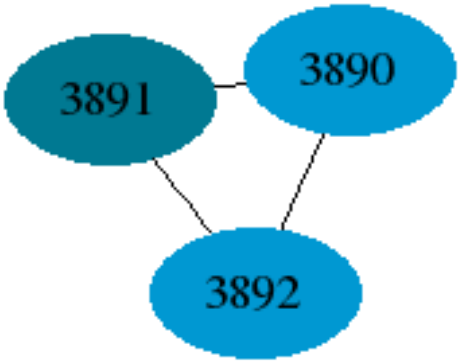
Methods (5)

- Compare clusters in same population (country code) over time
- Compare clusters in different populations to see what cross-border clusters we find
- Developed code to be usable from very modest scanning machine (for zmap/zgrab stages at least)
  - 0.75 GB RAM, 7.5GB disk, “25%” of an AMD Opteron 62xx class CPU, 10Mbps bandwidth
  - Analysis stages benefit from more RAM/disk, usually a laptop is fine
  - Doesn’t need to be fast – days per run are just fine
- Only IPv4 scanning, no IPv6 for now
- Note that our trigger is “listens on port 25” not “does TLS on port 25”
- All the code is at:

  https://github.com/sftcd/surveys

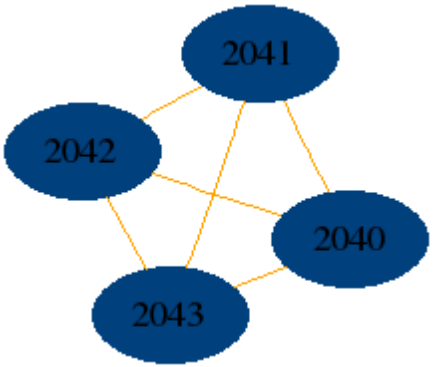
Results...

IE-20180318 Cluster 76
3 hosts sharing ecdsa-sha2-nistp256 SSH host keys in 2 different ASes
SMTP banners not obviously related



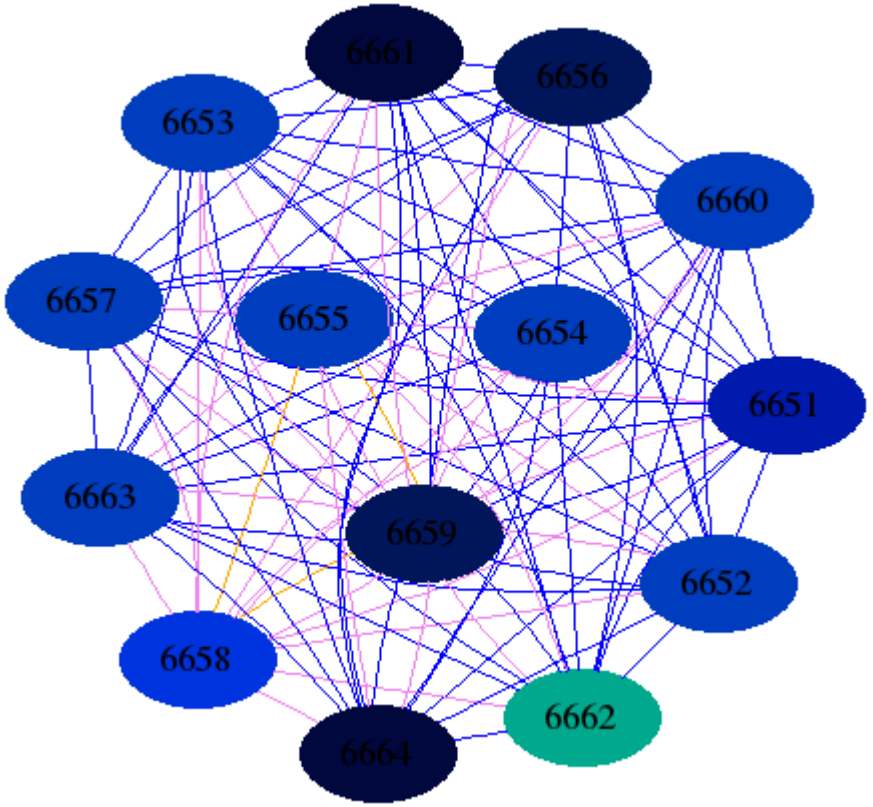
Results...

IE-20180318 Cluster 8
4 hosts sharing HTTPS server keys
The certificate is browser-trusted, issued in 2017 (according to crt.sh)
The hosts each have a different SSH host key
Most clusters are small like this and fairly boring



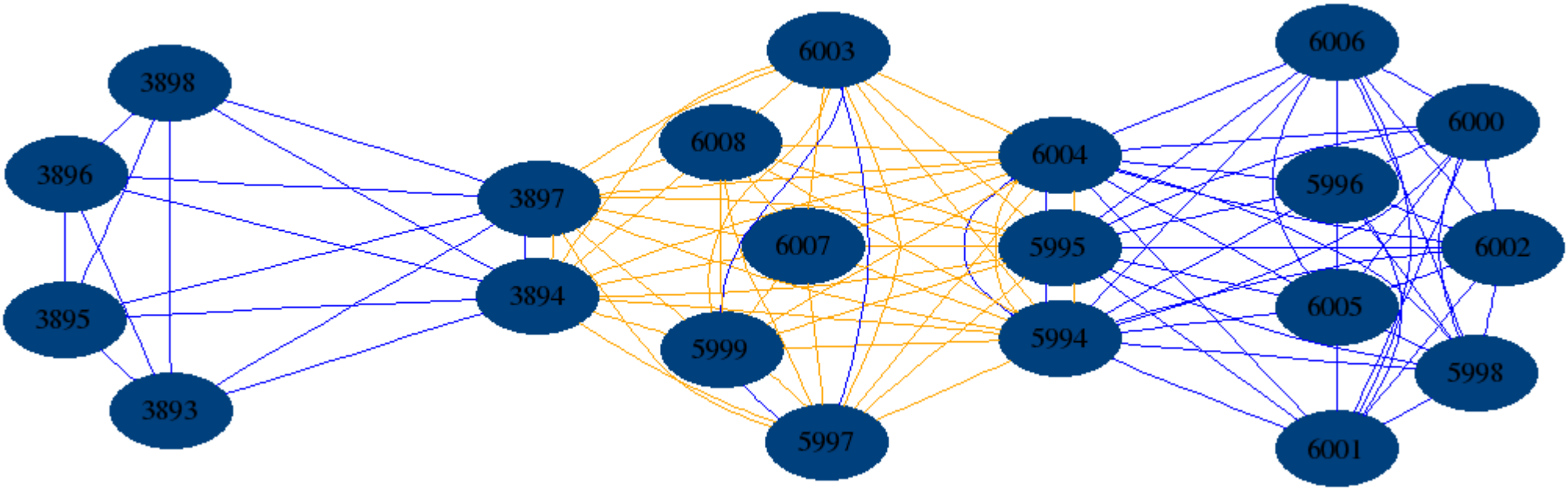
Results...

IE-20180318 Cluster 462
14 hosts sharing a 1024-bit RSA key over 6 ASes
Cipher-suite uses RSA key transport
This is due to a vendor product with a “demonstration” TLS key pair
Same key visible in all country scans (see cross-border later)
One host I checked here uses that “live” for it’s MX and is a destination for sensitive emails
I let the relevant CERT-like entity know about that
On 20180607 the vendor concerned stated that they would contact affected customers and fix the issue in the product. (Yay!)



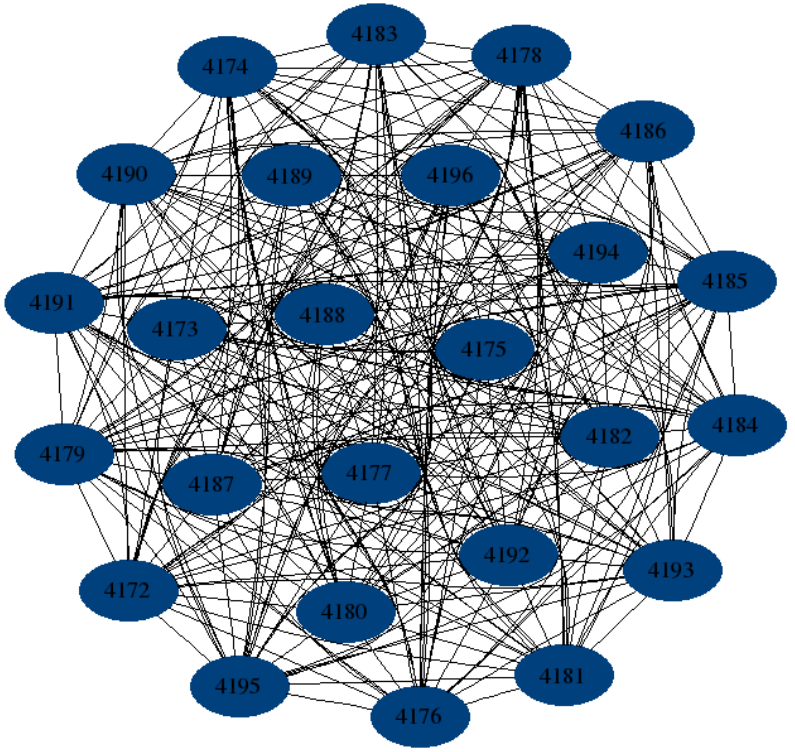
Results...

IE-20180318 Cluster 333
21 hosts sharing mail and web keys in various ways
One shared key has been browser-trusted since 2014 (crt.sh says)
Some commonality in banner names
This was cluster 10 in the IE-20171130 scan



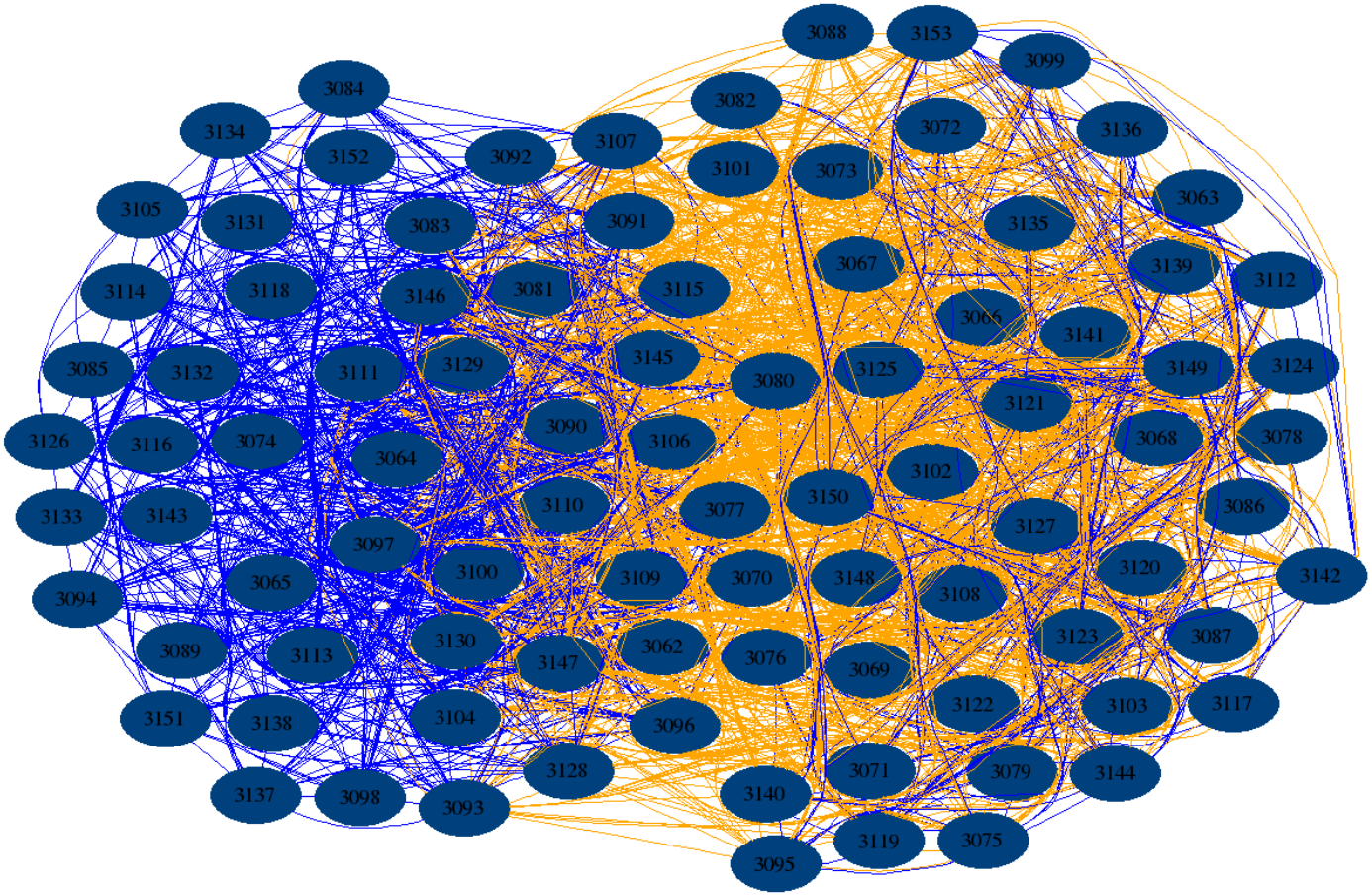
Results...

IE-20180318 Cluster 103
25 hosts sharing SSH host keys, no TLS services are visible at these IPs
The AS here is a hyper-scaler with a local hosting presence



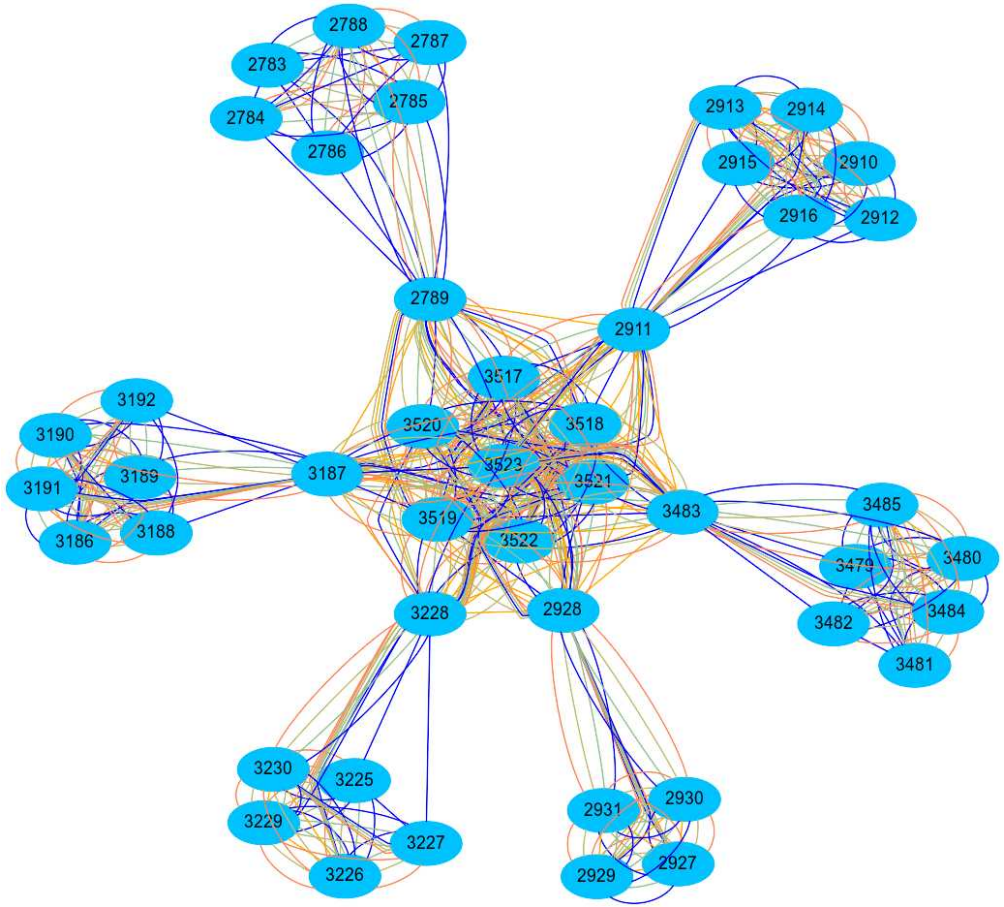
Results...

IE-20180318 Cluster 32
92 hosts sharing web and mail keys
About 50 different looking SMTP banners
Possibly a mobile web application developer doing stuff for it's customers



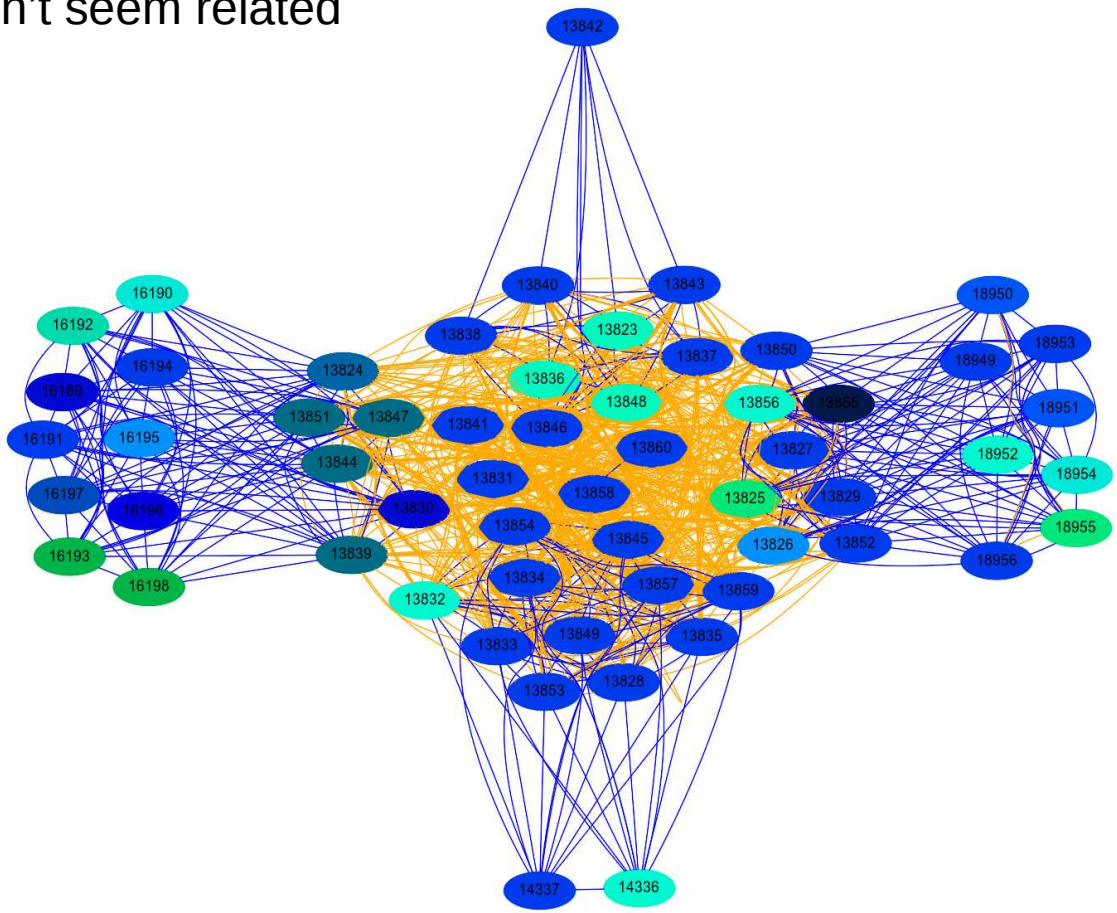
Results...

SI-20180514 Cluster 152
46 hosts sharing web and mail keys
Many clusters (incl. this) include names of hosting tools in banners/SANs
Support fora for such tools do seem to indicate key mgmt is considered tedious/hard



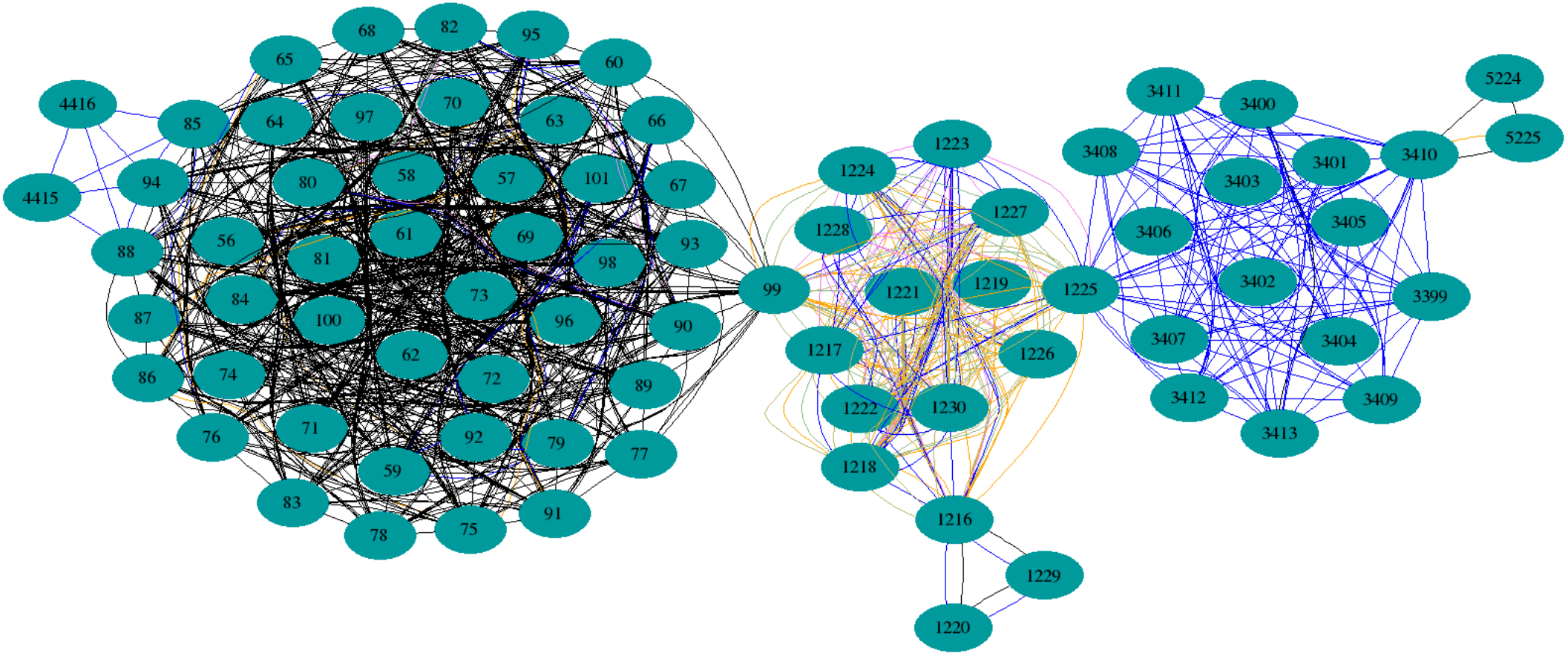
Results...

SG-20180430 Cluster 128
58 hosts sharing web and mail keys over 15 different Ases
11 TLS key pairs used for 151 host/port combinations
145 services with 1024-bit RSA, 38 with 2048 (32 SSH)
SMTP banners don't seem related



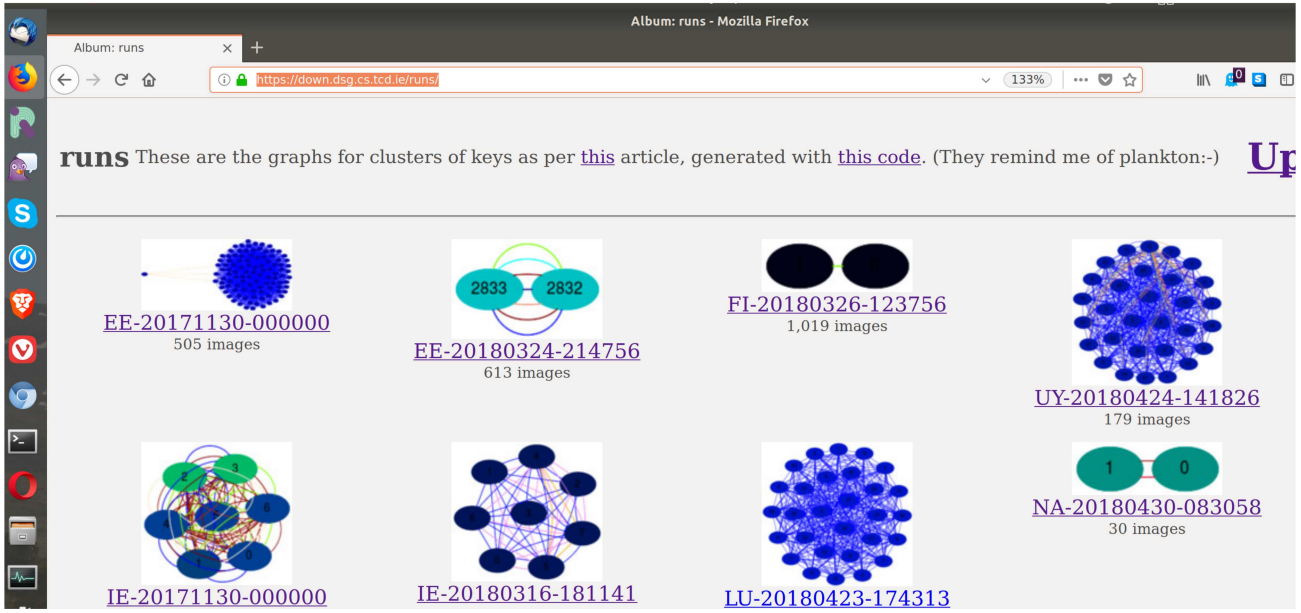
Results...

FI-20180328 Cluster 835
80 hosts sharing SSH, web and mail keys;
One SSH key, seen 46 times here, is also seen 10 times in PT!
For some reason, this is my favourite image, but I have no idea what's going on here;-)
IPs seem to relate to a local ICT consultancy – sent 'em mail on 20180708



Results...

- Want more graphs? There's 10,962 of those at https://down.dsg.cs.tcd.ie/runs
- Some graphs are too big to render using graphviz tools (at least, I've failed to find a way to do 'em;-) so 10,962 is not quite the overall number of clusters



Numeric results (1)

- Mostly self-explanatory, except:
- Analysis date: when analysis finished, sometimes after bug fixing:-)
- “Out of country”: sometimes MaxMind DB is out of date at scan time or analysis time, or maybe BGP changes happened
- HARK % == “Hosts are re-using keys” percentage is:

  #hosts-in-clusters/#hosts-doing-some-crypto

  - Could be a useful metric for repeated runs?
  - Would like it to decrease, not necessarily to zero

Numeric results (2)

Country (year)	IE(2017)	IE(2018)	EE(2017)	EE(2018)	FI(2018)	PT(2018)
Scan start	2017-11-30	2018-03-16	2017-11-30	2018-03-24	2018-03-26	2018-04-03
Analysis	2018-04-15	2018-03-25	2018-04-14	2018-03-29	2018-04-01	2018-04-05
IPs from ZMap	23616	24774	12775	17827	37012	19782
"out of county"	0	1233	0	1334	506	63
"In country" IPs	23616	23541	12775	16493	36506	19719
No crypto seen	12959	5273	796	1519	26106	4169
Some Crypto	10657	18268	11979	14974	10400	15550
Some crypto%	45%	77%	93%	90%	28%	78%
Total crypto host/ports	25935	54447	45067	80019	34263	63907
Total unique keys	12889	20053	15502	20014	11686	12202
Percent keys vs. max	49%	36%	34%	25%	34%	19%
Hosts with only local keys	5651	8570	3176	3303	4675	4143
Hosts in clusters	5006	9698	8803	11671	5725	11407
HARK	**46%**	**53%**	**73%**	**77%**	**55%**	**73%**
Number of clusters	823	1437	521	639	1029	1512
Max cluster size	671	1991	2874	2402	373	2016
Median cluster size	21	26.5	36	42	24	30
Average cluster size	63.23	87.78	121.18	98.04	50.65	117.51

Numeric results (3)

Country (year)	LU	UY	NZ	NA	SG	SI
Scan start	2018-04-23	2018-04-24	2018-04-25	2018-04-30	2018-04-30	2018-05-14
Scan end	2018-04-24	2018-04-25	2018-04-29	2018-04-30	2018-05-22	2018-05-20
IPs from ZMap	6800	1878	23333	551	73608	9759
"out of county"	4	0	3	0	14	0
"In country" IPs	6796	1878	23330	551	73594	9759
No crypto seen	686	336	11872	172	21871	1882
Some Crypto	6110	1542	11458	379	51723	7877
Some crypto%	89%	82%	49%	68%	70%	80%
Total crypto host/ports	21284	4806	33424	820	129346	26330
Total unique keys	5622	1355	10657	504	35364	7421
Percent keys vs. max	26%	28%	31%	61%	27%	28%
Hosts with only local keys	1966	720	5814	299	22644	3529
Hosts in clusters	4144	822	5644	80	29079	4348
HARK	**67%**	**53%**	**49%**	**21%**	**56%**	**55%**
Number of clusters	446	180	811	30	3050	612
Max cluster size	1012	245	281	8	2800	948
Median cluster size	19	7.5	25.5	4	52.5	19.5
Average cluster size	73	29.67	47.12	4.4	179.47	66.88

Cluster size distribution (1)

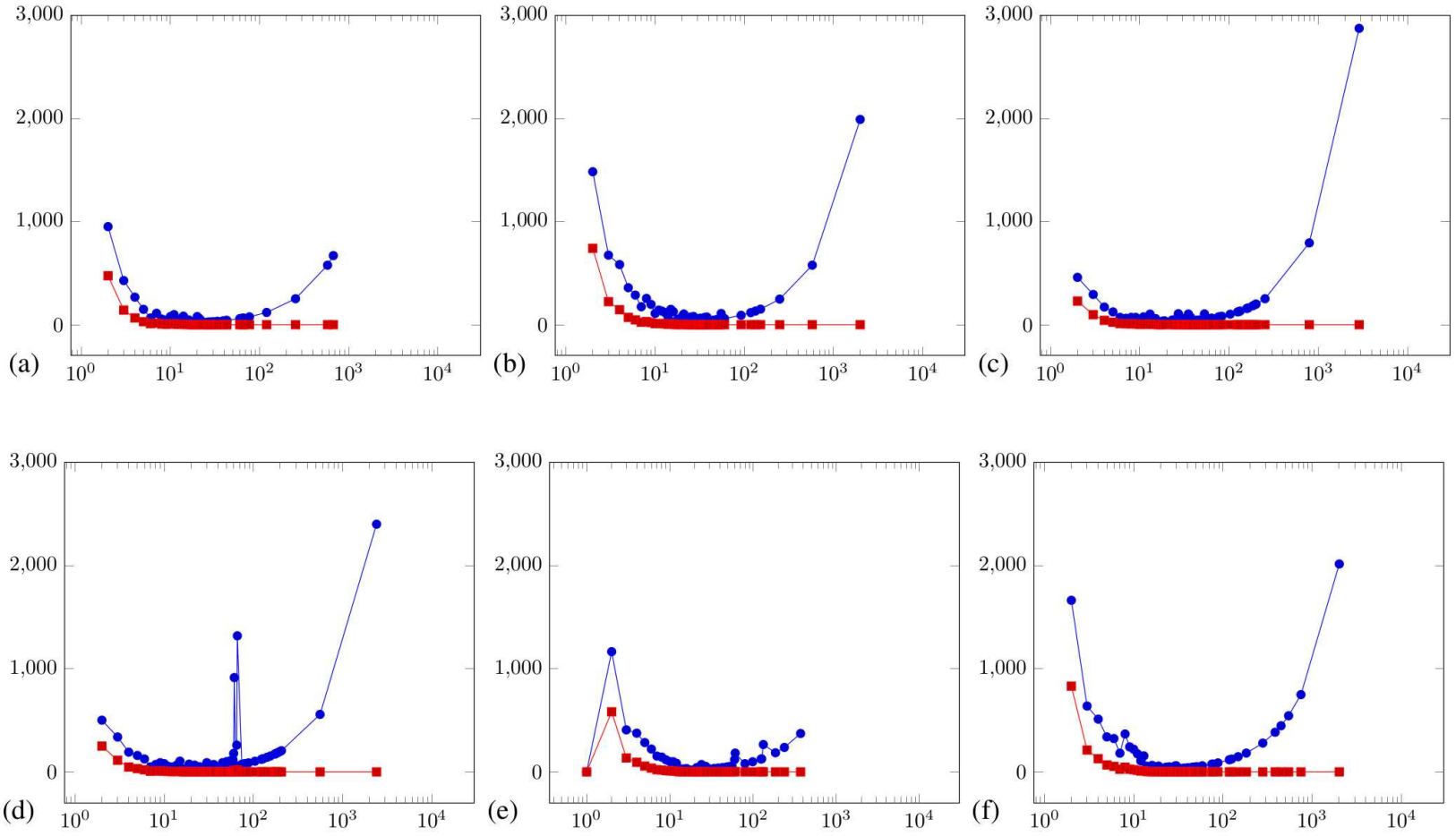


Fig. 3: Cluster size distributions for runs (a) IE-20171130, (b) IE-20180316, (c) EE-20171130, (d) EE-20180324, (e) FI-20180326, (f) PT-20180403. Blue circles show the number of hosts in clusters of given size, red squares reflect the number of clusters of given size. The x-axis is logarithmic.

Cluster size distribution (2)

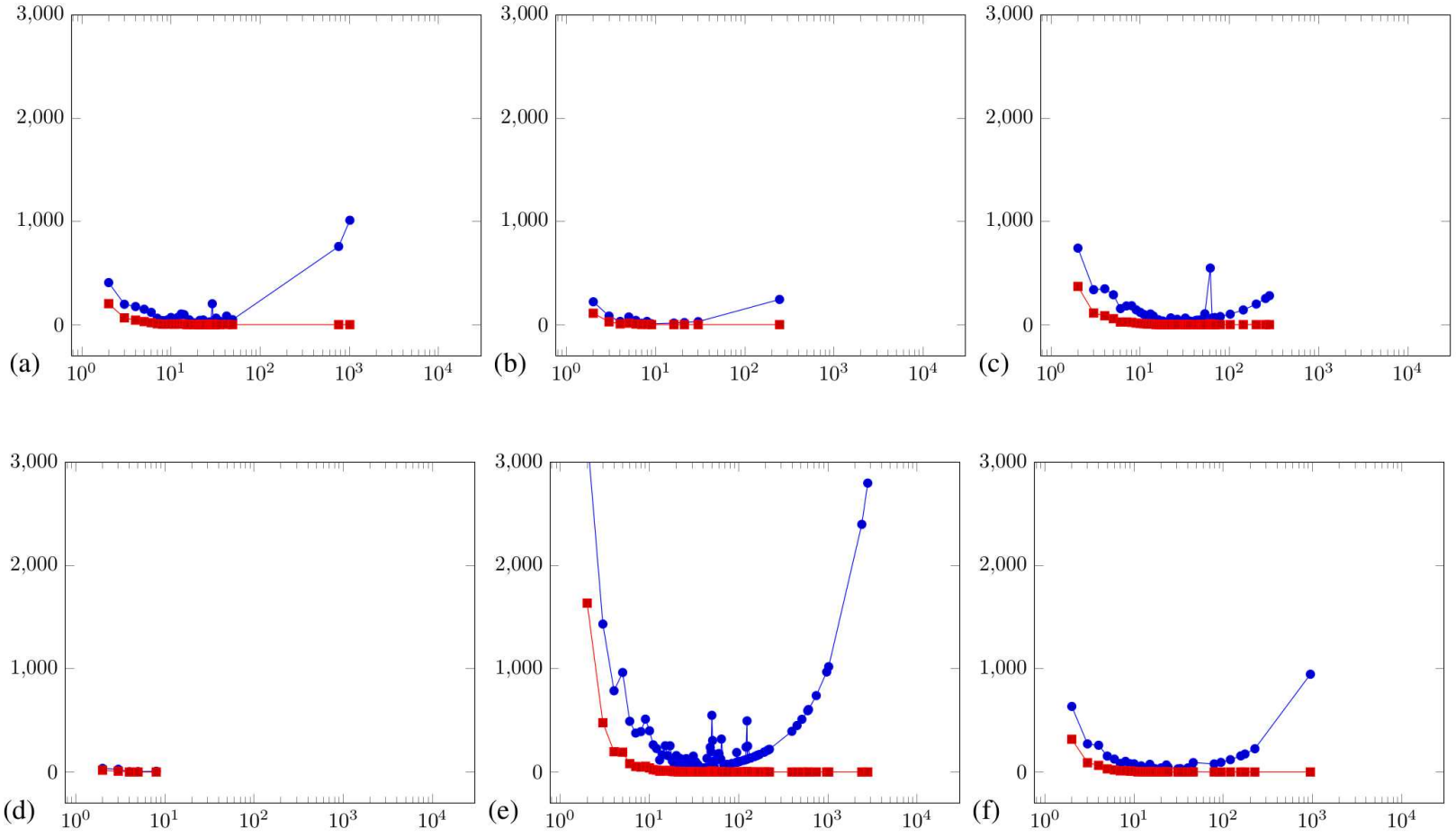


Fig. 31: Cluster size distributions for runs (a) LU-20180423, (b) UY-20180424, (c) NZ-20180425, (d) NA-20180430, (e) SG-20180430, (f) SI-20180514. Blue circles show the number of hosts in clusters of given size, red squares reflect the number of clusters of given size. The x-axis is logarithmic. NA figures are so small this rendering isn't useful so Figure 33 presents NA at a more appropriate scale.

Evolution over time (1)

- Clusters are determined by IP addresses and keys
  - Keys can be moved, deleted, generated
  - IP addresses appear, disappear or can be re-purposed
- We see pretty much every possible kind of evolution from IE-2017 to IE-2018
  - New clusters appear, old ones disappear
  - New ones can be linked to old via both keys and IP addresses, or just via keys, or just via IP addresses
  - More complex evolution is also possible (e.g. two clusters turns into 3)

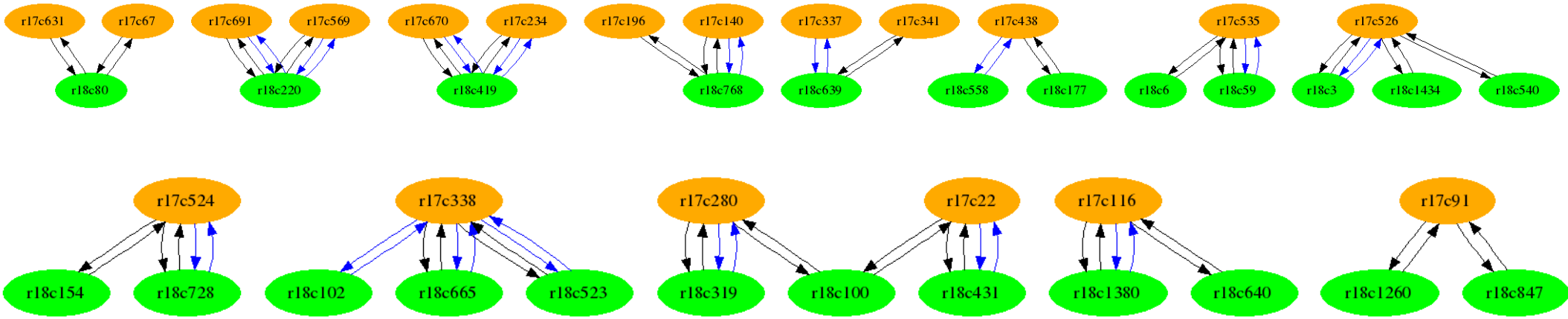
Evolution over time (2)

TABLE VII: Cluster evolution - Categories in the evolution from IE-20171130 to IE-20180316. Numbers are the number of clusters in each category.

Category	#	Category	#
Disappeared	168	Appeared	777
IP-linked	36	FP-linked	16
IP and FP-linked	584		
Complex-20171130	19	Complex-20180316	24

Evolution over time (3)

- The complex cluster changes in Ireland
  - Orange: 2017, Green: 2018
  - Blue arrows indicate key linkage
  - Black arrows indicate IP address linkage

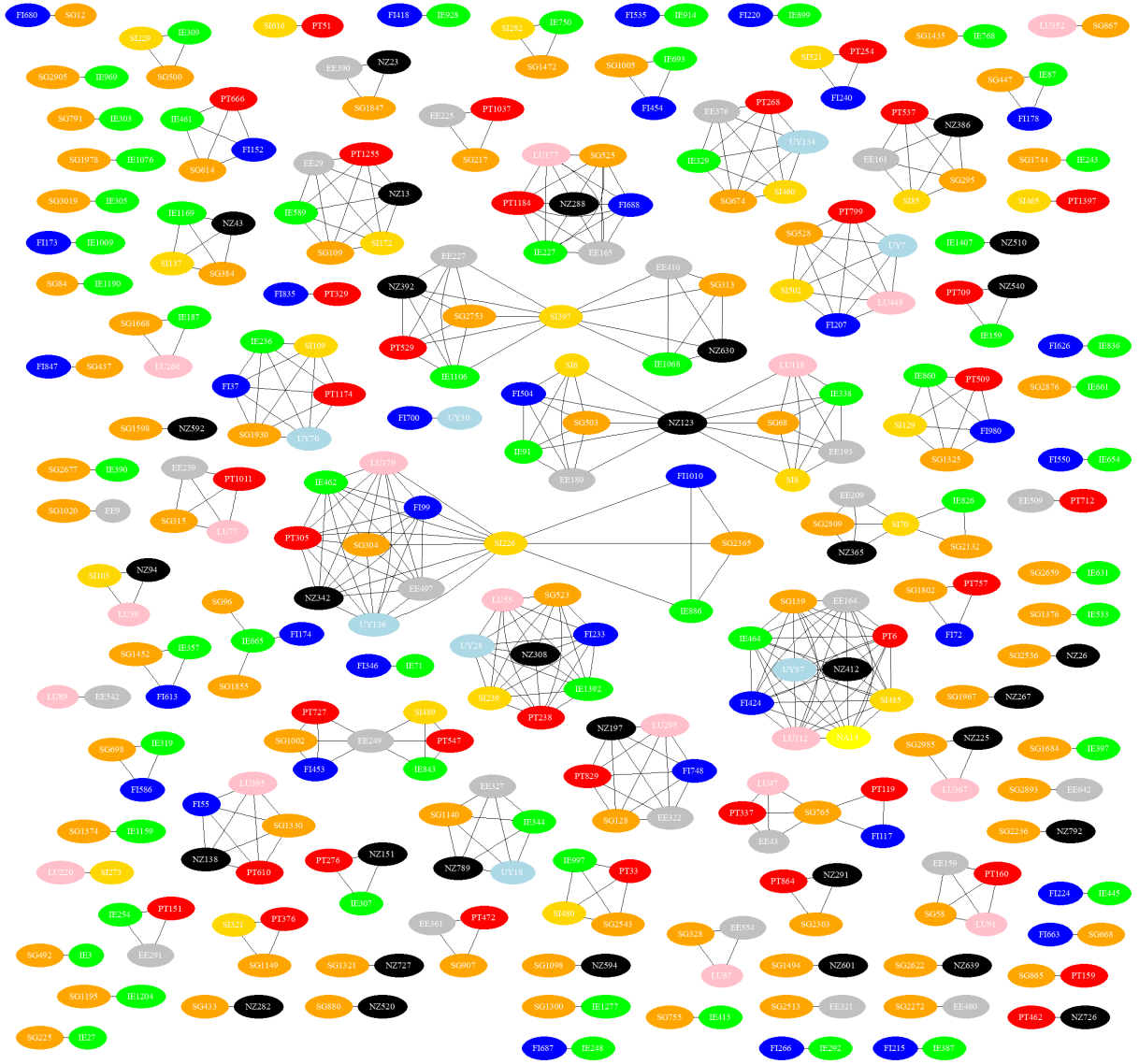


101 cross-border clusters (1)

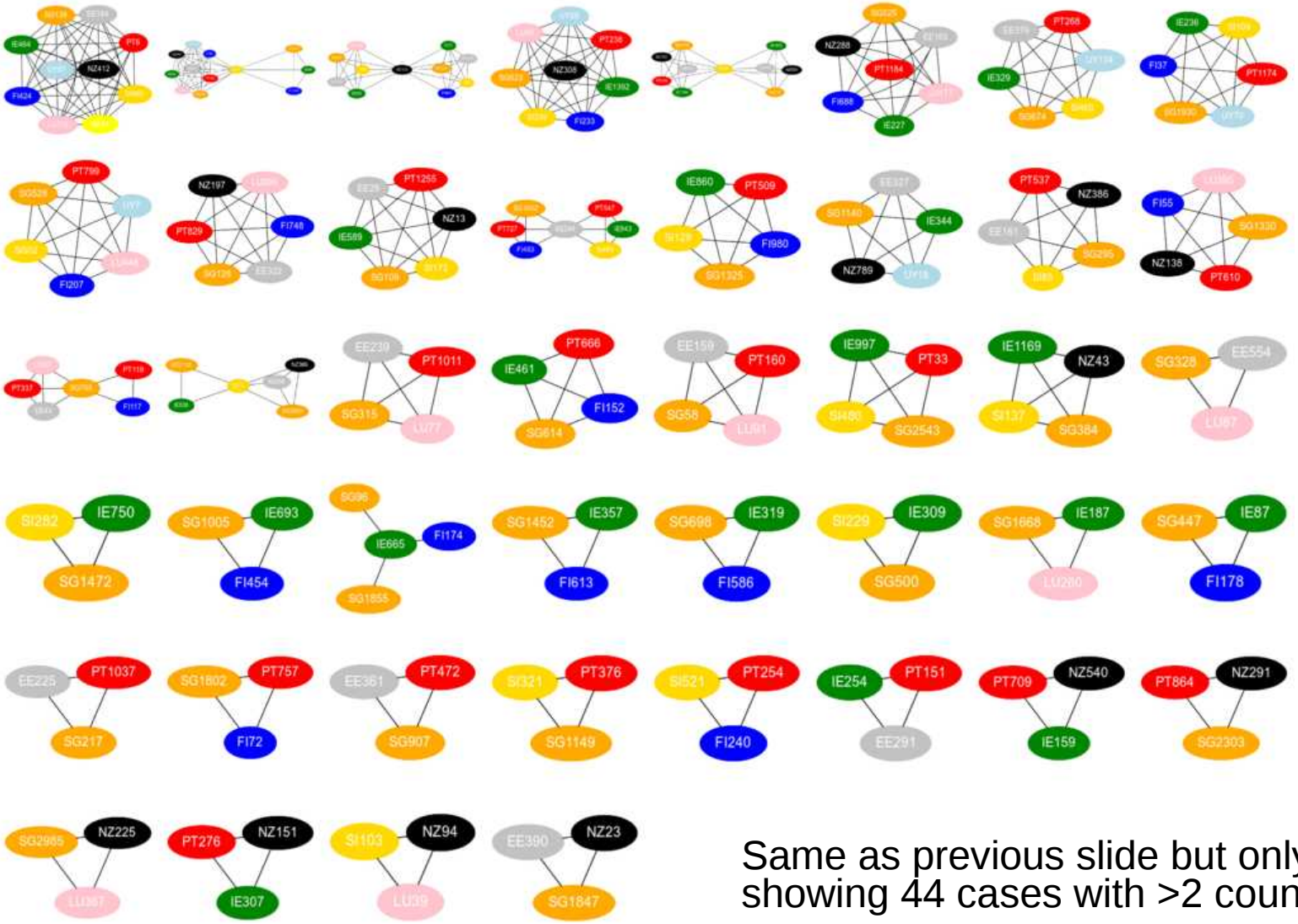
- As we see clusters that have addresses in multiple ASes we may well see cross-border links between clusters
  - And we do. In every case. In 101 different ways involving 7,468 IP addresses.

-	NA	NZ	UY	PT	IE	EE	SI	SG	LU	FI
NA	x	1	1	1	1	1	1	1	1	1
NZ	1	x	4	13	14	13	12	28	9	7
UY	1	4	x	6	6	4	6	7	4	6
PT	1	13	6	x	15	17	16	26	10	15
IE	1	14	6	15	x	12	18	45	6	25
EE	1	13	4	17	12	x	11	25	10	6
SI	1	12	6	16	18	11	x	21	7	9
SG	1	28	7	26	45	25	21	x	15	22
LU	1	9	4	10	6	10	7	15	x	7
FI	1	7	6	15	25	6	9	22	7	x

101 cross-border clusters (2)



101 cross-border clusters (3)

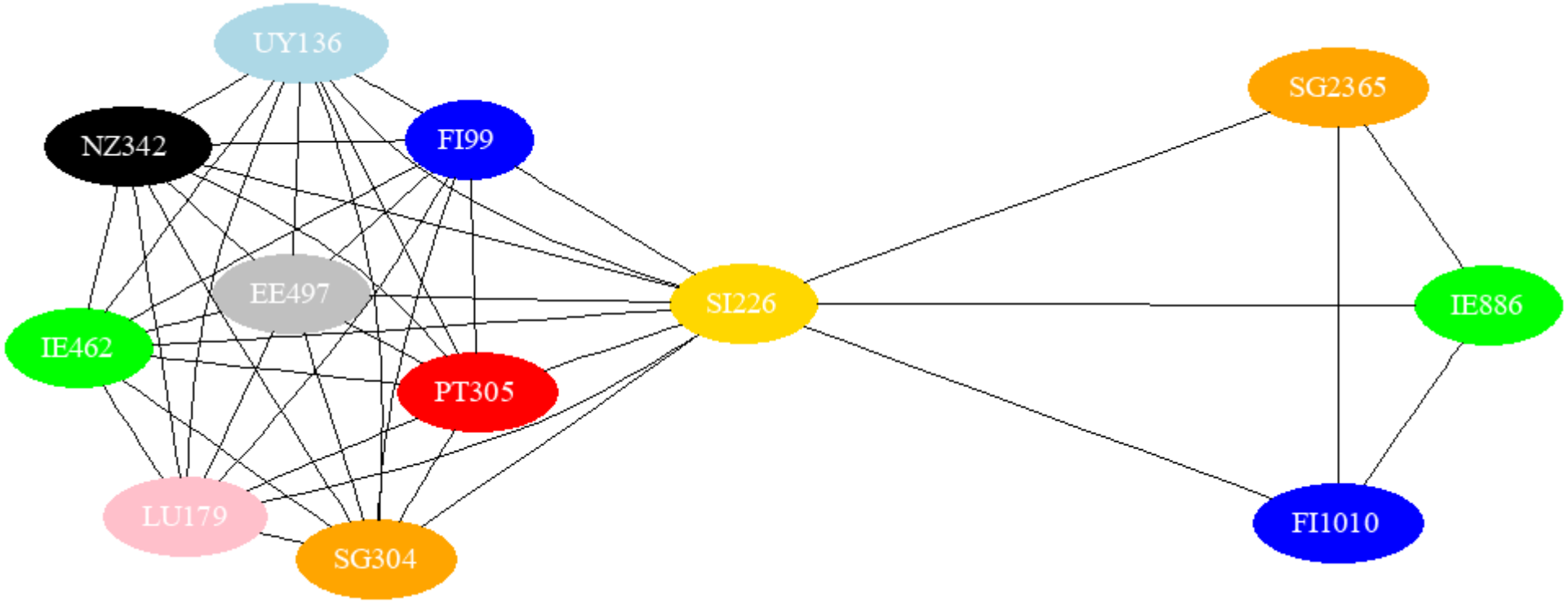


Same as previous slide but only showing 44 cases with >2 countries

101 cross-border clusters (4)

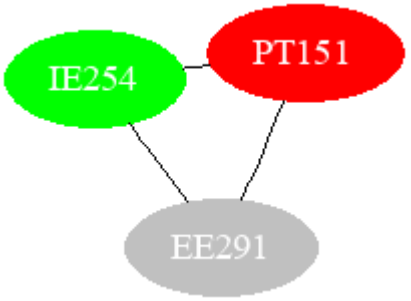
- All (but one) of the cross-border “super” clusters so far are fully connected, **or**, are two fully connected graphs linked by one cluster
  - Example on next slide
- That might indicate that there are common causes here
  - Smells like hard-coded keys?
  - Or maybe wild-card certs used in many places
  - Analysis here is still a work-in-progress
- Note: cross-border analysis code only considers hosts already in clusters in one country, so some links won’t be seen, e.g. if just one IP address in each country has a copy of a re-used key

Cross-border example (1)



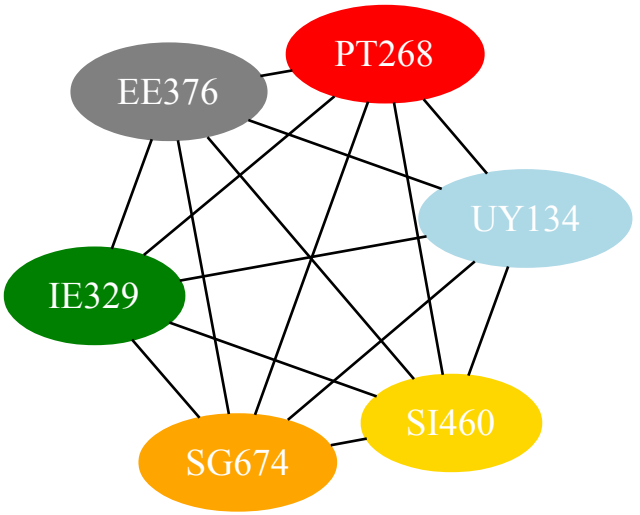
SCNZ342 is a set of linked cross-border clusters – the LHS of this includes the IE462 cluster previously shown and due to a vendor product that ships with a “demonstration” key pair. The RHS of this seems to be caused by the same product with a different “demonstration” key, perhaps an earlier or later version. There are 85 hosts in total in these clusters over 42 ASes.
On 20180607 the vendor concerned stated that they would contact affected customers and fix the issue in the product. (again: Yay!)

Cross-border example (2)



SCPT151 is a cross-border cluster of SSH host keys involving 21 hosts on 3 countries over 6 ASes.
It turns out I “own” one of the IE hosts. That’s a VPS I started renting in 2013. The offending ECDSA host key file is dated in 2012, a little more than a year before I starting paying for service.
I opened a ticket with the hoster – they answered, nothing else odd seen.
I expect the ECDSA key was in the image used in 2012.

Cross-border example (3)



SCUY134 is a cross-border cluster of keys involving 99 hosts in 6 countries over 24 ASes.
This is (partly) caused by an open-source package intended for developers that ships with a 1024-bit RSA key. The package is not intended for real deployment, just for easy development, but of course users do what they want later;-)
The hard-coded key is visible on port 443 on 27 of the 99 IP addresses in the super-cluster. It may be (not sure) that other OSS packages use this one and inherit the infelicity.
I contacted the maintainers via their security notification mail address. They answered the next day (20180614) saying that they would add key-generation at install-time to a future version.

Confirmed causes

- SSH Host Key generation prior to virtualisation
  - Seems like a mistake that happens over and over, in various **different** ways
- Products that ship with default or “demonstration” keys
- Large scale use of wildcard certs (is a 1991 address cluster for ~250 customers of a marketing campaign tool reasonable?)
- Mega-SANs: Saw one cert with >1500 SubjectAltNames (SANs) and had to write code to stop name analysis after 100 SANs because they slow down the analysis
- Multi-homed hosts with up to ~20 IP addresses (an offering from one IE hoster)
- The above have all been confirmed with asset-holders, the preprint describes some more not-yet-confirmed possible causes
  - Confirming more is a work-in-progress – help appreciated!

Multi-Homed Hosts

- Multi-homed hosts aren't in general detectable via IPv4 address scans, so add "noise" to our results. That may affect about 29% of clusters.

TABLE VI: Numbers of possible non-Multi-Homed and Multi-Homed Host Clusters. Note that one cluster can have more than one indicator (mixed/AS/multikey) that the cluster is not solely due to a multi-homed host, so the first three numeric columns don't add up to the 4th.

Run	Mixed	AS	multikey	non-MH	Possible MH	No Info	Total
EE-20171130	85	39	124	192 (36%)	191 (36%)	138 (26%)	521
EE-20180324	81	46	127	204 (31%)	242 (37%)	193 (30%)	639
FI-20180326	70	105	105	225 (21%)	300 (29%)	504 (48%)	1029
IE-20171130	75	68	134	240 (29%)	190 (23%)	393 (47%)	823
IE-20180316	97	133	202	385 (26%)	317 (22%)	735 (51%)	1437
LU-20180423	34	44	20	78 (17%)	90 (20%)	278 (62%)	446
NA-20180430	3	15	1	19 (63%)	2 (6%)	9 (30%)	30
NZ-20180425	62	98	39	165 (20%)	235 (28%)	411 (50%)	811
PT-20180403	66	153	50	208 (13%)	502 (33%)	802 (53%)	1512
SG-20180430	157	178	287	484 (15%)	1053 (34%)	1513 (49%)	3050
SI-20180514	34	96	25	127 (20%)	95 (15%)	390 (63%)	612
UY-20180424	8	31	3	38 (21%)	23 (12%)	119 (66%)	180
Totals	772	1006	1117	2365 (21%)	3240 (29%)	5485 (49%)	11090

TLS Versions

- TLSv1.0 is still high! Don’t have rate-of-change info yet.

TABLE IX: TLS Versions seen overall

Run	SSLv3	TLSv1.0	TLSv1.1	TLSv1.2	Total
IE-20180316	7	7323	6	40209	47545
EE-20180324	10	5672	10	58713	64405
FI-20180326	16	5562	4	23801	29383
PT-20180403	21	6169	19	52111	58320
LU-20180423	2	1278	1	18790	20071
UY-20180424	1	771	3	3804	4579
NZ-20180425	18	6377	2	23838	30235
NA-20180430	5	171	0	505	681
SG-20180430	16	9319	15	168209	177559
SI-20180514	25	5927	2	19048	25002
Total	121	48569	62	409028	457780
Percent	0.02%	10.61%	0.01%	89.35%	100%

Mitigations

- Rotate keys – for TLS, certbot/LE combination should break up clusters in a few months if used
- Measure – check your network/customers etc to see if keys are being re-used in ways you don’t expect
- SSH clients could react to multiple copies of public key in known-hosts
- CAs could (in principle) be less tolerant of key re-use
- SSH protocol could (in principle) be changed to do key rotation by default for host keys and client keys

Conclusion

- Key re-use is much more widespread than (at least I) envisaged
- Rotating keys would result in improvements
- It's likely this is part of us all learning how to manage crypto at scale
- Measurement always does seem to turn up new stuff, in this case structure, go do some!
  - Don't assume you won't be found, if doing something odd;-)
- If you're a relevant asset-holder, do get in touch – I'd love to try help you improve your network posture and learn more about how this happens with your help

Thanks/Questions

- Offline questions are welcome too...
  - stephen.farrell@cs.tcd.ie
  - PGP Key ID: 0x5AB2FAF17B172BEA
- Preprint: https://eprint.iacr.org/2018/299
- Code: https://github.com/sftcd/surveys/
- Graphs: https://down.dsg.cs.tcd.ie/runs/
- This: https://down.dsg.cs.tcd.ie/misc/hark.pdf