Identidad, criptografía, confianza, y el desarrollo de una comunidad

Un estudio sobre el llavero criptográfico del proyecto Debian

Gunnar Wolf

9 de noviembre, 2017

El proyecto Debian

- Proyecto de desarrollo e integración de software libre, de ámbito mundial
- Más de 5400 personas han contribuido con el proyecto en sus 22 años
- Participación voluntaria (no corporativa, siempre a título individual

¿Qué hace Debian?

- Una de las distribuciones de Linux más empleadas
  - Muchas distribuciones derivadas (Ubuntu, Mint, Kali, etc.)
  - $\approx$ 70% instalaciones de Linux
  - `http://w3techs.com/technologies/history_details/os-linux`
- Alta responsabilidad del proyecto sobre las acciones de sus participantes
  - *«¡Tengo root en millones de computadoras!»*
- Contrato social
  - `https://www.debian.org/social_contract`

Necesidad de *vinculación fuerte*

persona $\Longleftrightarrow$ cuenta

¿Qué es la *identidad*, criptográficamente?

Al verificar de forma automatizada la *identidad* de una persona otorgando semejante nivel de acceso, el viejo esquema usuario/contraseña sencillamente no es suficiente.

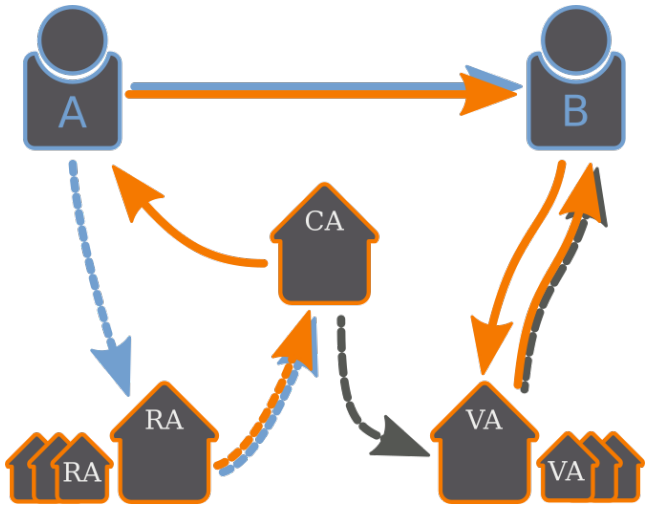
"Nube" con las contraseñas más frecuentemente utilizadas, según su frecuencia relativa (Burnett 2011)

Certificados de identidad

- La validación de identidad requiere *por lo menos* un certificado criptográfico
- Documento *extremadamente improbable* de ser falsificado
- Basado en criptografía asimétrica (o de llave pública)
- Un *tercero confiable* respalda la identidad de quien lo presenta
- Dos modelos principales

Modelo *Infraestructura de llave pública* (PKI)

- Modelo centralizado
- Depende de *autoridades certificadoras* (CAs)
  - *Confianza última* en las CAs para un sistema dado
- Cada certificado firmado por una *única* CA

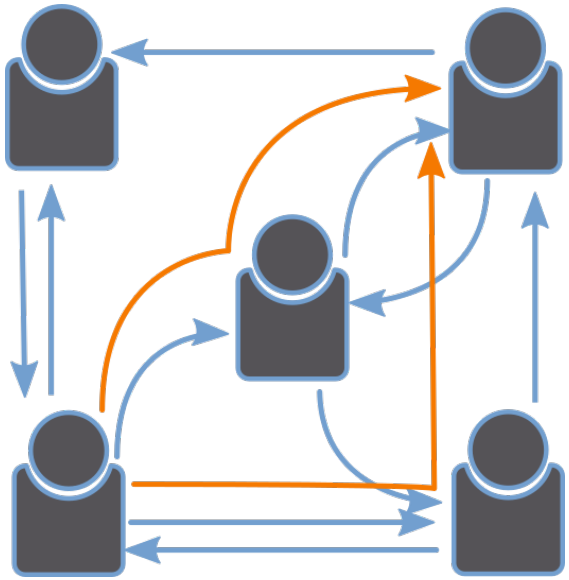


Infraestructura de llave pública (Wikipedia 2015)

- Estándar de la industria, particularmente relevante por TLS/SSL (Dierks and Rescorla 2008)
  - Presente en todos los navegadores Web, adecuable a cualquier comunicación TCP/IP
- Estándar de certificados X.509 (Cooper et al. 2008)

Modelo *Llavero de confianza* (WoT)

- Modelo distribuido
- Cada entidad puede *validar* la identidad de los demás
  - No indica confianza *en la entidad*, sino en su *identidad*
- Un conjunto de participantes $\longrightarrow$ *llavero*



Llavero de confianza

- Todo certificado puede ser firmado por *cualquier cantidad de entidades* en el llavero
  - $\rightarrow$ Confianza *transitiva*
  - Varias métricas para establecer la confianza (McBurnett 1997; Penning 2004; Penning 2015)
- Estándar de certificados y comunicación OpenPGP (Callas et al. 2007)

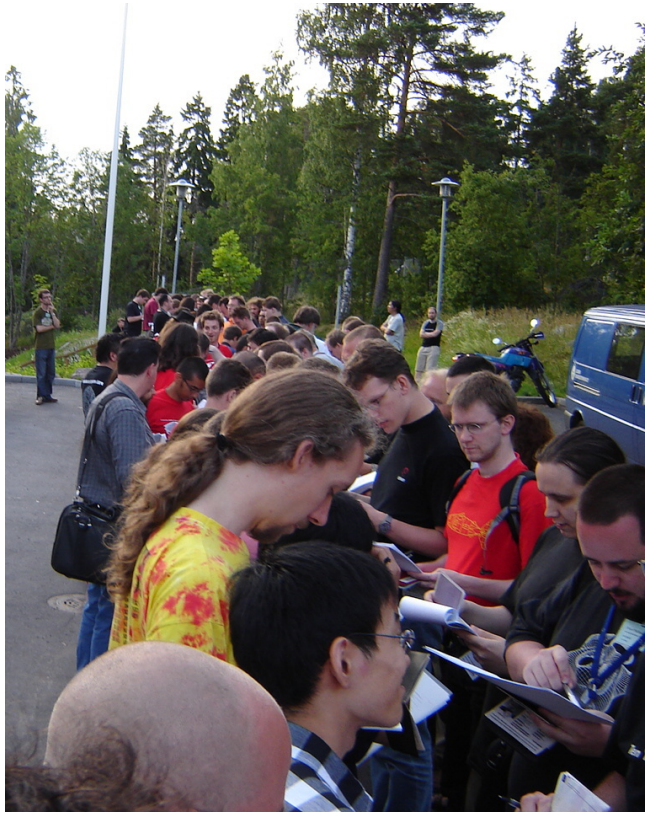
Definiendo la participación en Debian

- Debian es el proyecto (y grupo social) que mayor uso da al WoT en el mundo (Cederlöf 2004)
  - Participación en varias áreas de Debian $\rightarrow$ *firmado* por una llave OpenPGP
  - Perteneciente a uno de los *llaveros de confianza*
- Idealmente, los llaveros *deberían* guardar relación 1:1 con la participación formal en el proyecto

El equipo *keyring-maint*

- Responsables del mantenimiento de los llaveros
- Establecimiento de identidad: Llavero de confianza + proceso de *curaduría*
- Una de nuestras misiones: Asegurar que los llaveros se mantengan al día ante retos y amenazas a los esquemas del cifrado asimétrico

Las *fiestas de firmado de llaves* (KSPs)



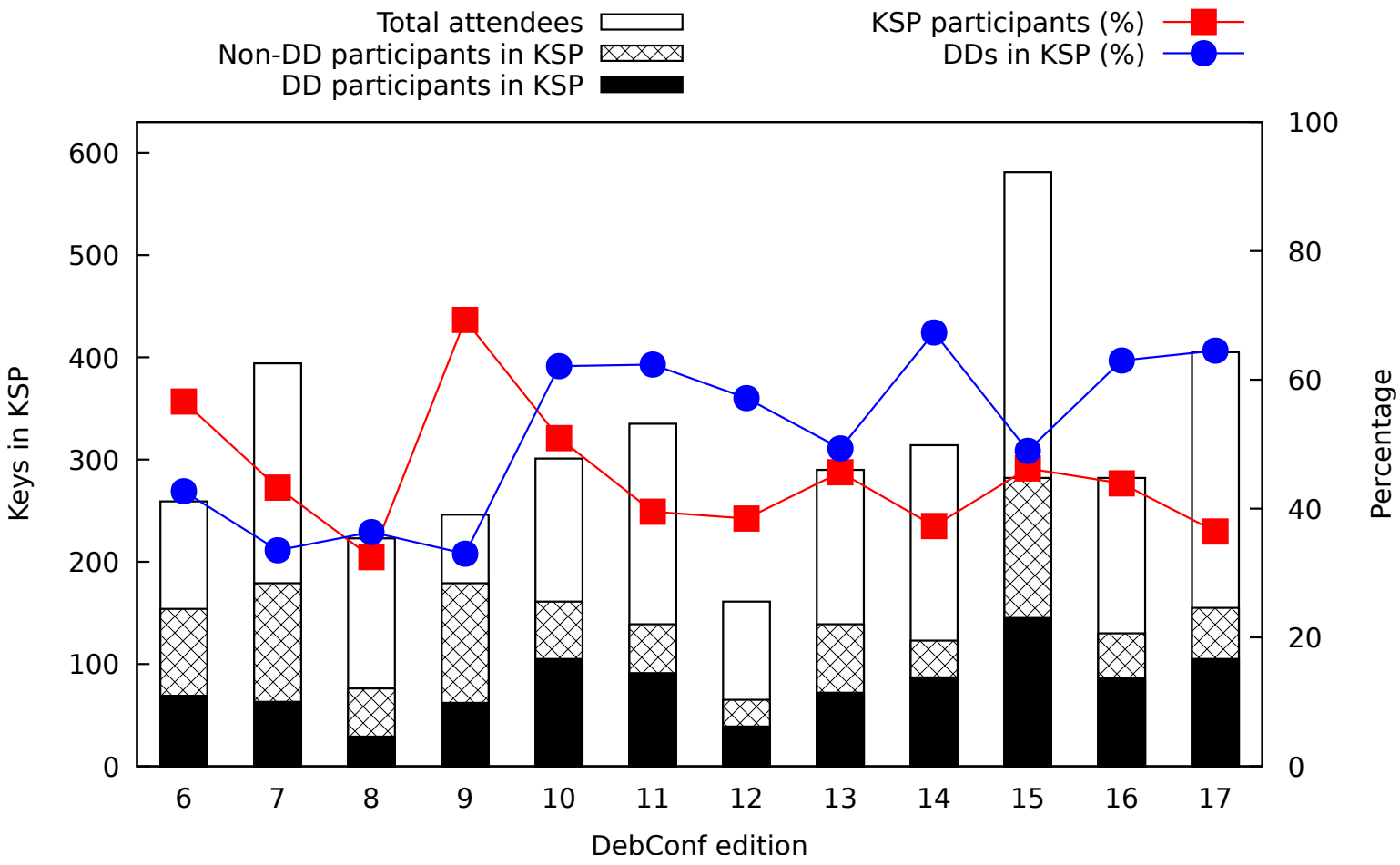
KSP en DebConf5, Helsinki, 2005

- El nivel de confianza en la identidad depende de las *firmas cruzadas*
  - En los proyectos con amplia distribución geográfica se acostumbran las KSPs
- Se busca que cada participante verifique la identidad de los demás
  - Típicamente se sigue el protocolo Sassaman and Zimmerman (2006)

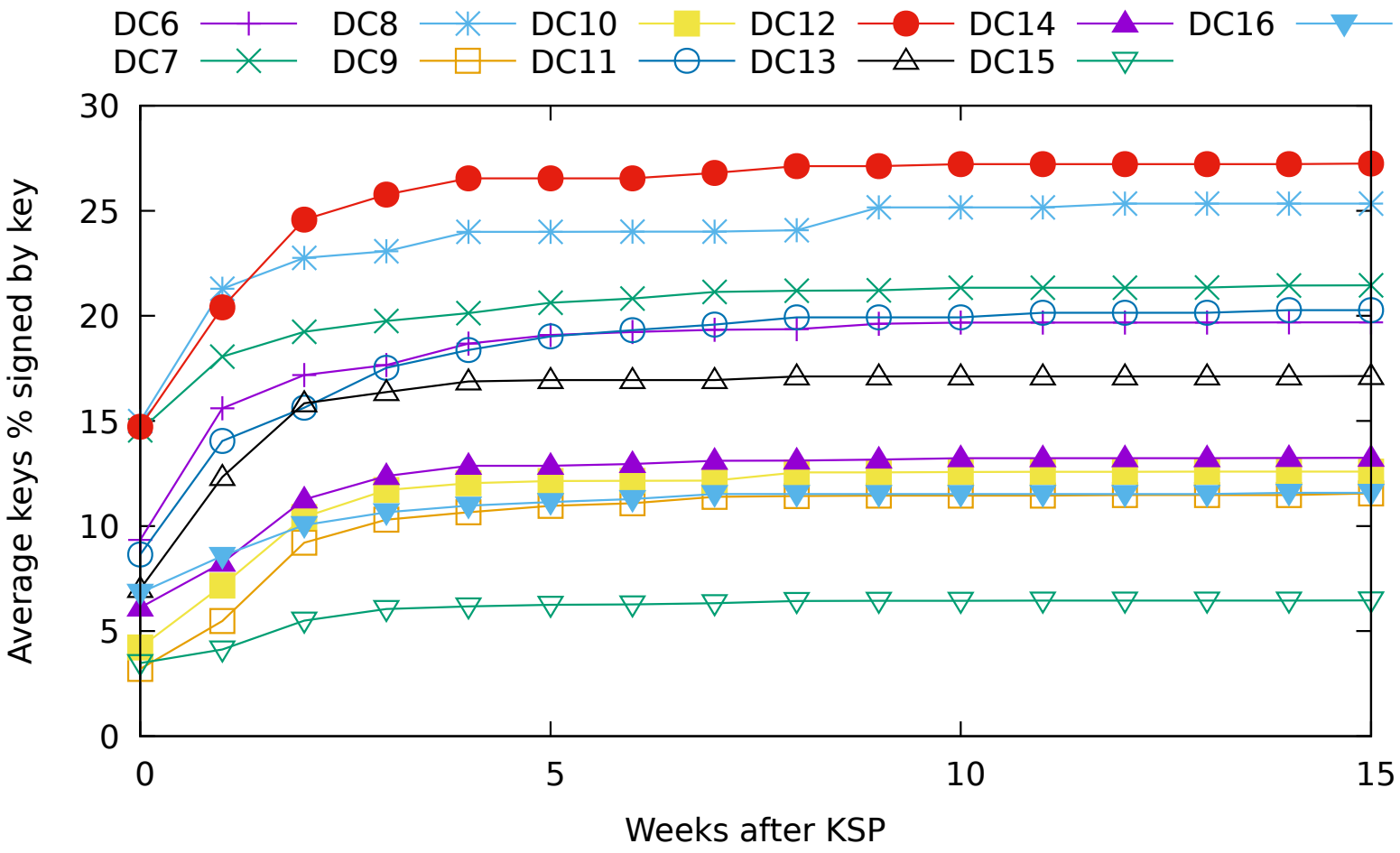
¿Cómo se verifica la identidad?

- La verificación *debe ser* suficientemente confiable para satisfacer los requisitos *personales*
  - *No hay lineamientos globales*, cada quién usa los propios
  - *Típicamente*: Documento oficial emitido por el gobierno local; muchos usan distintos criterios
- Atacando al esquema (¿académicamente?)
  - Un desarrollador bien conocido personalmente por el proyecto *probó* la seguridad de una KSP presentando una identificación falsa en 2006
  - Llevó al replanteamiento del manejo de las KSPs en grupos grandes (Srivastava 2006; Krafft 2008)

Midiendo a las KSPs: Participación



Midiendo a las KSPs: Ritmo de crecimiento



Métricas de llaves y llaveros: Conceptos

Conjunto fuerte (strong set) El mayor subconjunto de llaves entre las cuales hay (por lo menos) un camino bidireccional de confianza

Distancia mínima La longitud de la ruta más corta entre dos llaves en el llavero. Pueden o no ser del conjunto fuerte, mide la cercanía entre dos llaves cualquiera

Distancia promedio mínima (MSD) Medida utilizada para establecer qué tan confiable es la identidad de *un* participante. Sólo sobre el conjunto fuerte; promedio de la distancia de una llave a todas las demás llaves del mismo

Promedio de distancias promedio mínimas (AMSD) Promedio de la MSD de todas las llaves del conjunto fuerte (aplica al llavero completo)

Midiendo la fuerza de la criptografía

Dentro de la *criptografía de llave pública* es posible realizar implementaciones más *fuertes* y más *débiles* — Más *seguras* y más *inseguras*

- A mayor fortaleza, mayor complejidad computacional
- Obviamente, el estado del cómputo cambia con los años. . .
- Van apareciendo debilidades algorítmicas que obligan a replantear el uso de determinados algoritmos
- . . .

Nivel de seguridad en criptografía simétrica

Table 7.4: Security levels (symmetric equivalent).

Security Level	Security (bits)	Protection	Comment
1.	32	Attacks in "real-time" by individuals	Only acceptable for auth. tag size
2.	64	Very short-term protection against small organizations	Should not be used for confidentiality in new systems
3.	72	Short-term protection against medium organizations, medium-term protection against small organizations	
4.	80	Very short-term protection against agencies, long-term prot. against small organizations	Smallest general-purpose level, ≤ 4 years protection (E.g. use of 2-key 3DES, $< 2^{40}$ plaintext/ciphertexts)
5.	96	Legacy standard level	2-key 3DES restricted to $\sim 10^6$ plaintext/ciphertexts, ≈ 10 years protection
6.	112	Medium-term protection	≈ 20 years protection (E.g. 3-key 3DES)
7.	128	Long-term protection	Good, generic application-indep. recommendation, ≈ 30 years
8.	256	"Foreseeable future"	Good protection against quantum computers unless Shor's algorithm applies.

Resistencia aproximada por nivel de seguridad en criptografía simétrica (Smart 2012)

Equivalencia efectiva entre tamaños de llave

Table 7.2: Key-size Equivalence.

Security (bits)	RSA	DLOG		EC
		field size	subfield	
48	480	480	96	96
56	640	640	112	112
64	816	816	128	128
80	1248	1248	160	160
112	2432	2432	224	224
128	3248	3248	256	256
160	5312	5312	320	320
192	7936	7936	384	384
256	15424	15424	512	512

Table 7.3: Effective Key-size of Commonly used RSA/DLOG Keys.

RSA/DLOG Key	Security (bits)
512	50
768	62
1024	73
1536	89
2048	103

Equivalencia efectiva entre tamaños de llave, incluyendo simétricas y asimétricas (Smart 2012)

Necesidad de migrar a llaves *más fuertes*

- Entre los 1990s y 2000s, la recomendación para criptografía asimétrica era usar llaves de 1024 bytes
- Claramente, ya no es suficiente
  - 1024 bits asimétricos $\approx$ 73 bits simétricos
  - *Protección a corto plazo contra organizaciones medianas, protección a mediano plazo contra organizaciones pequeñas*
  - Insuficiente
- Problema: Llevar al llavero completo a migrar a llaves más fuertes

El problema, desde la perspectiva de PKI

- Un modelo centralizado puede *imponer* políticas y plazos para la migración forzosa
  - Motivación *directa y económica* para que cada cliente renueve sus certificados

(3) Subscriber Certificates

	Validity period ending on or before 31 Dec 2013	Validity period ending after 31 Dec 2013
Digest algorithm	SHA1*, SHA-256, SHA-384 or SHA-512	SHA1*, SHA-256, SHA-384 or SHA-512
Minimum RSA modulus	1024	2048

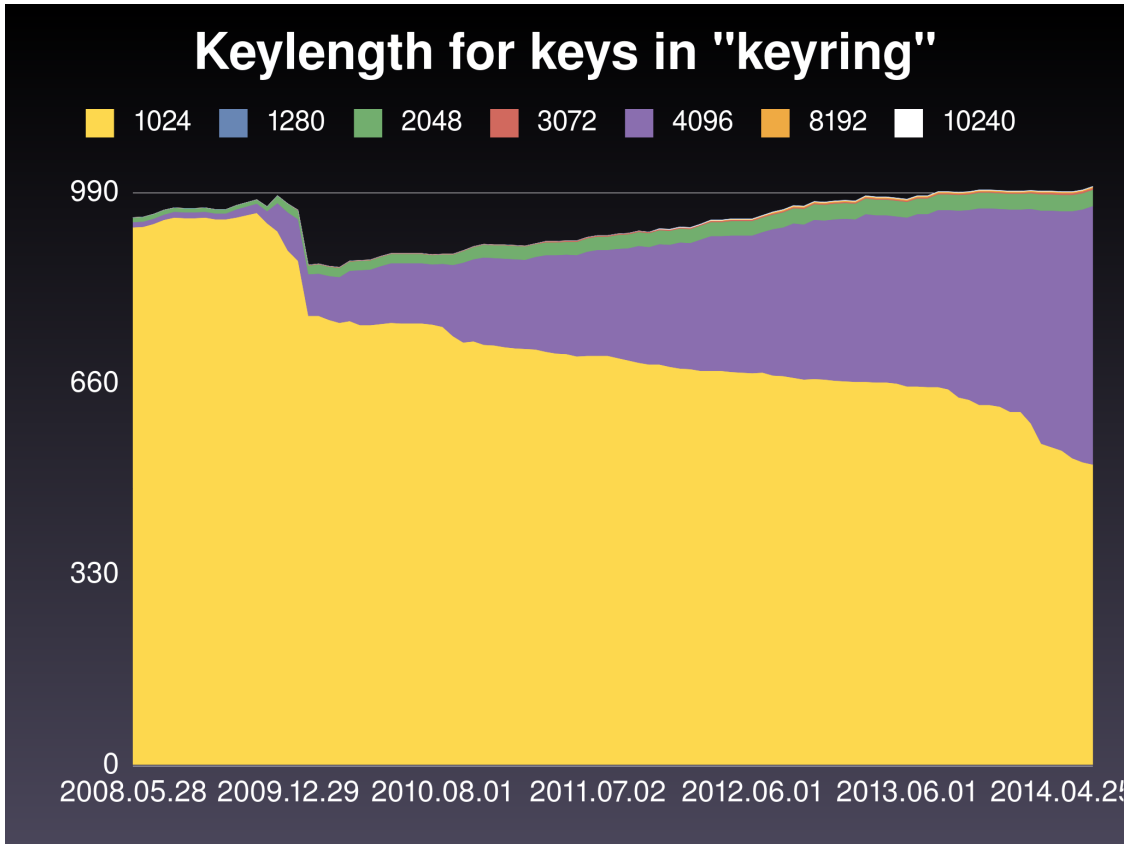
CA / Browser Forum Baseline Requirements, v. 1.1.8 (as of 5 June 2014)

Requisitos base para certificados de usuario final (The CA / Browser Forum 2011)

El problema, desde la perspectiva de Debian

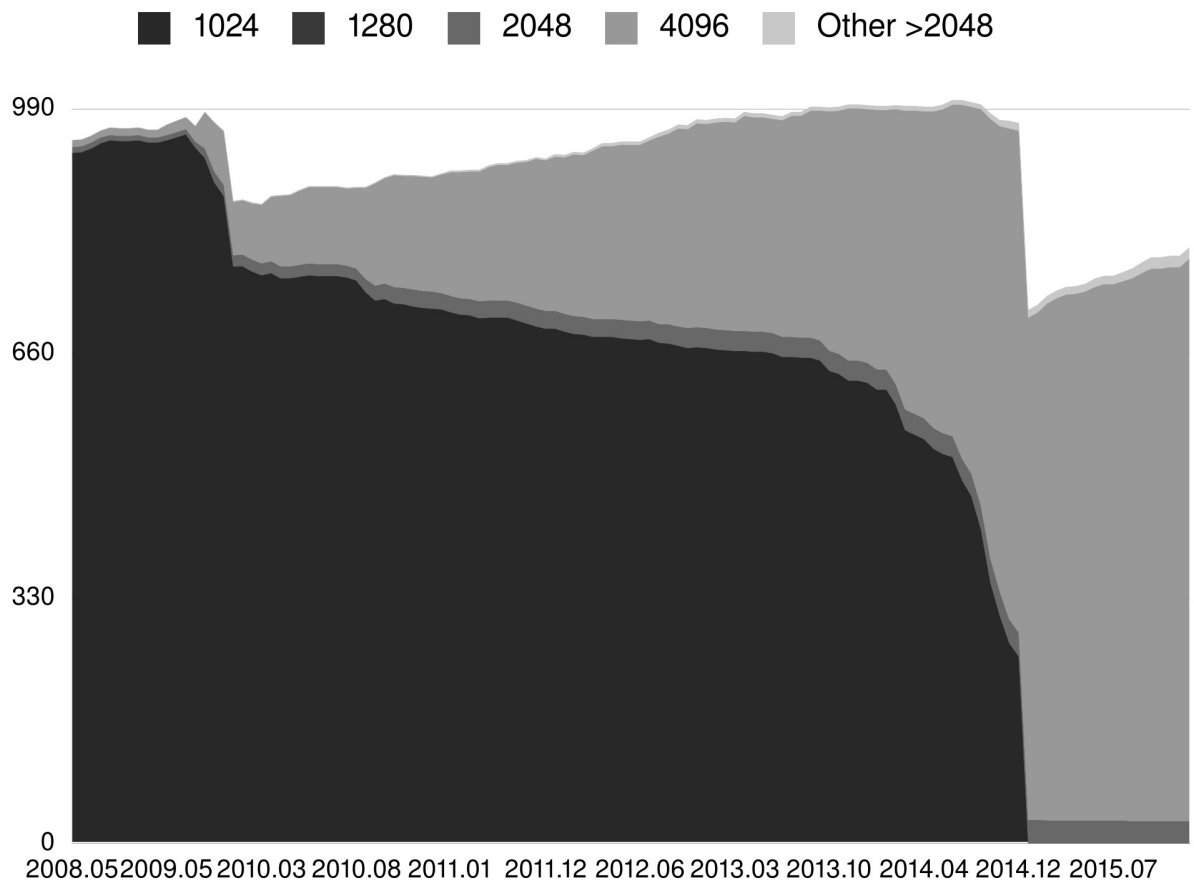
- El proyecto está interesado en *no perder* a los desarrolladores
  - Su trabajo *beneficia directamente* a Debian
  - Un desarrollador sin un certificado válido y acorde con los estándares del proyecto ve limitado o entorpecido su trabajo
  - Si bien hay cientos de desarrolladores en países centrales, hay decenas en regiones con baja densidad
  - No todo desarrollador participa del *entorno social*
- Plantear un proceso que bloquee efectivamente la participación de un amplio número de desarrolladores se arriesga a un fuerte rechazo

Llavero de Desarrolladores de Debian (ago'14)



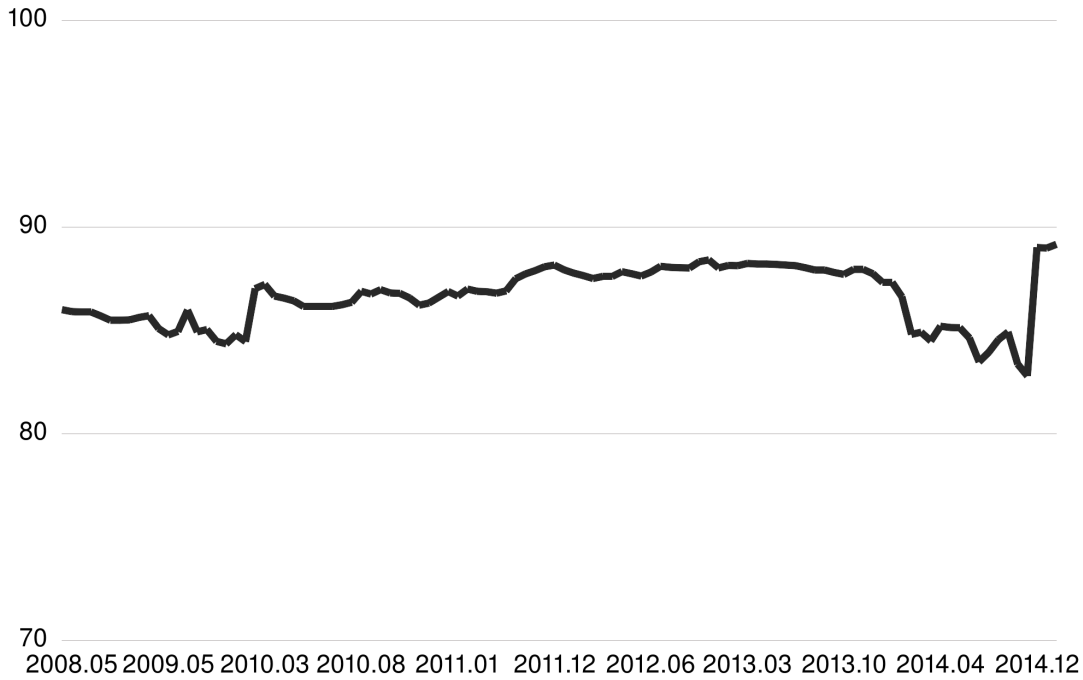
Evolución histórica del llavero de Desarrolladores de Debian, 2008-2015

Llavero de Desarrolladores de Debian (abr'16)



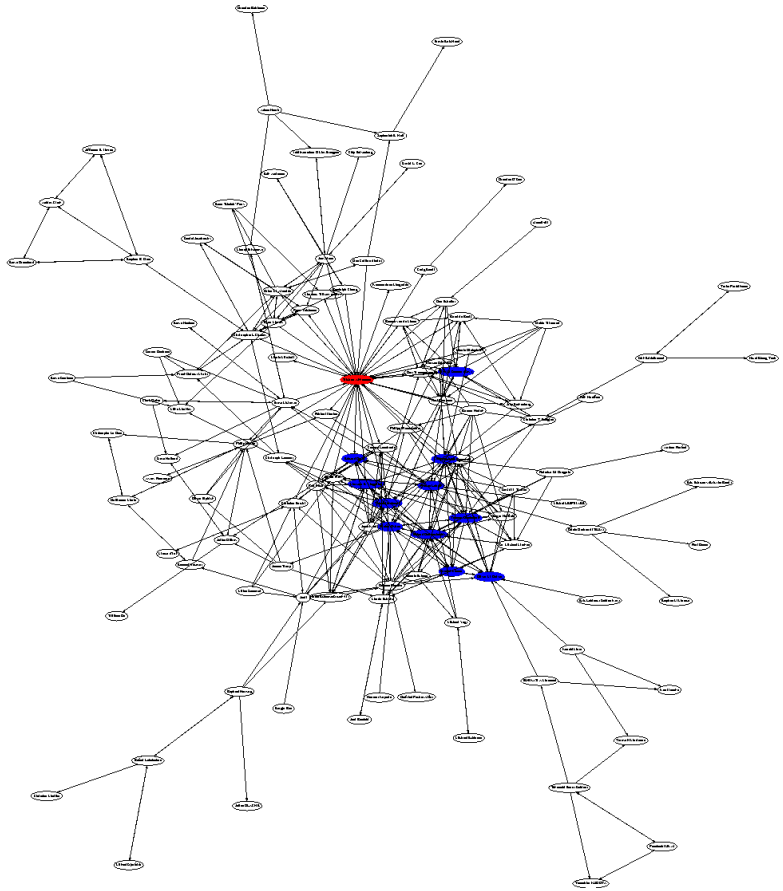
Evolución histórica del llavero de Desarrolladores de Debian, 2008-2016

Evolución del llavero de Debian: Conjunto fuerte



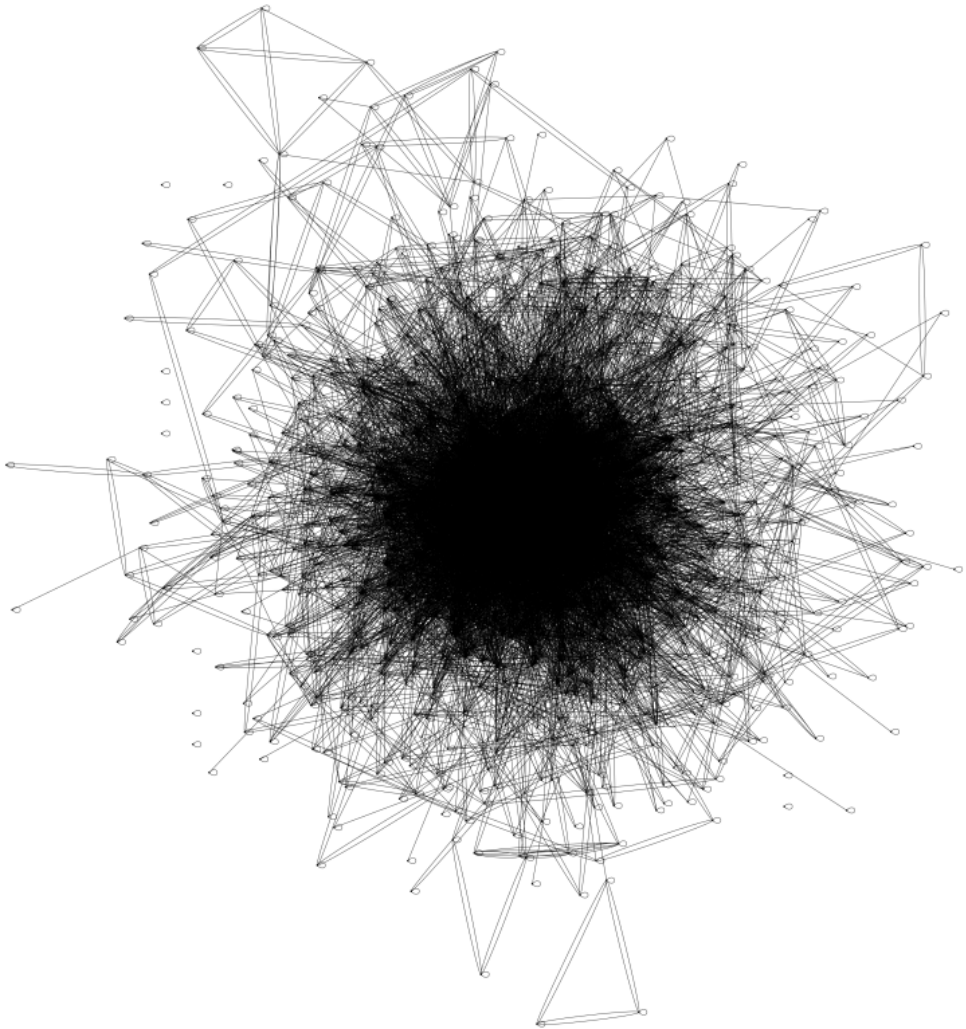
Conjunto fuerte en el llavero de Desarrolladores de Debian, 2008-2015: 82 – 89%

Conjunto fuerte de Debian en 2000



Conjunto fuerte en el llavero de Desarrolladores de Debian, 2000

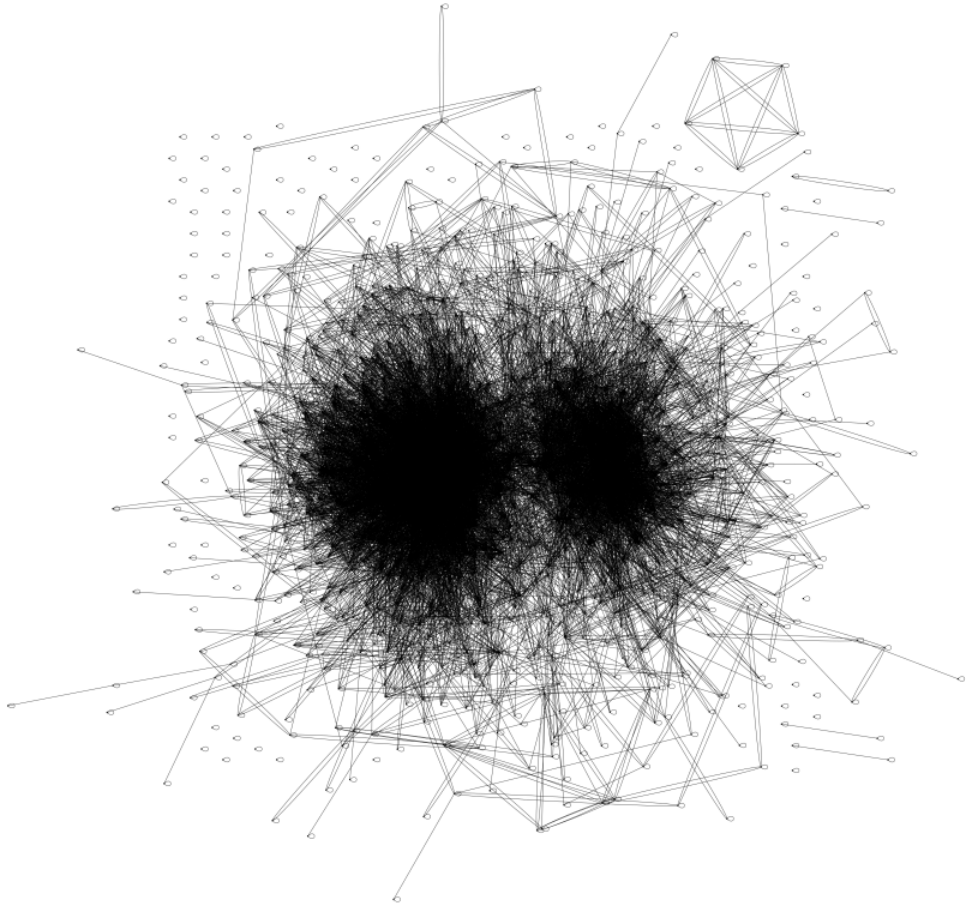
Foto del llavero después de la migración, 2016.06.19



keyring at 2016.06.19 (816 keys)

A maze of twisty passages, all alike

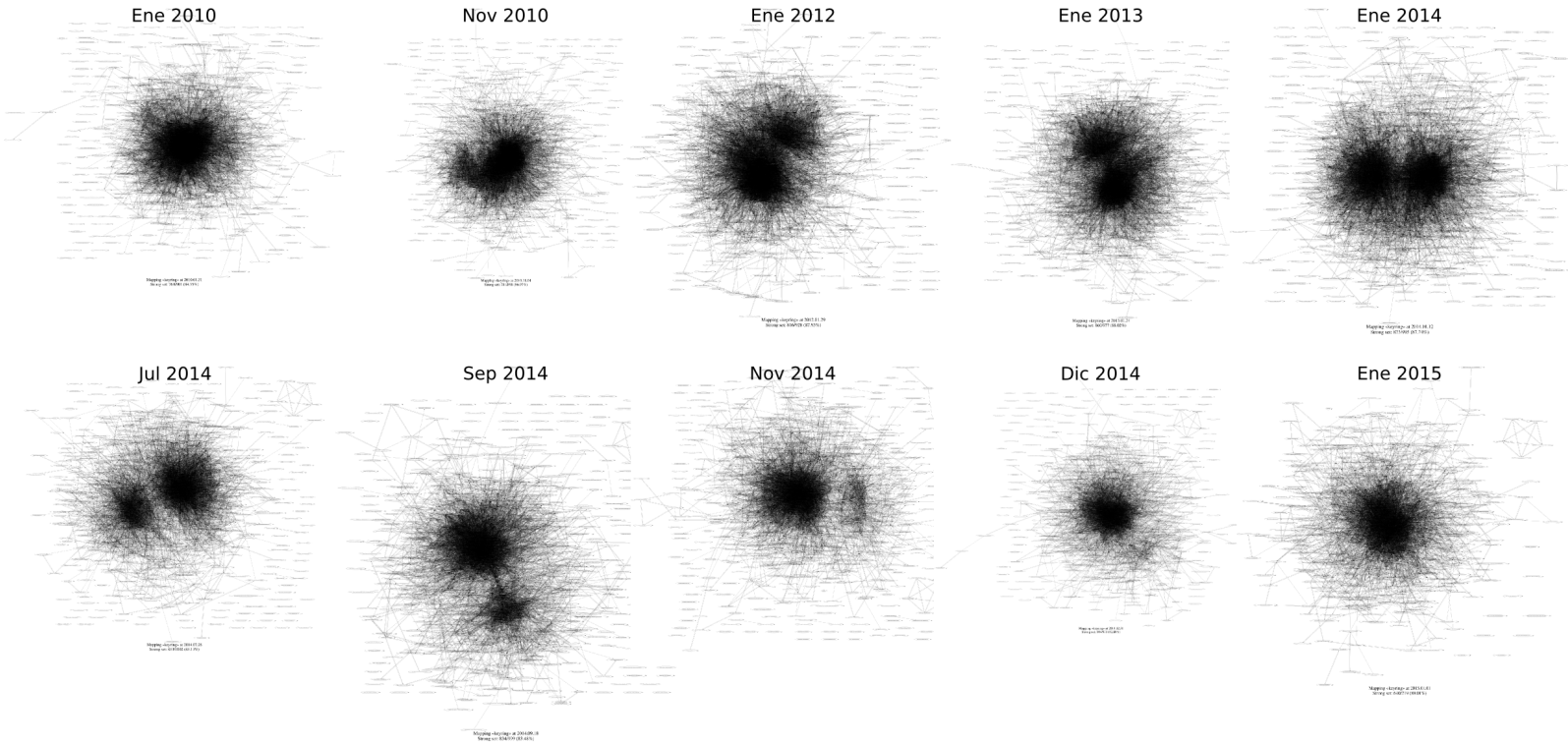
Un momento más *divertido*: Mediados del 2014



keyring at 2014.07.28 (1002 keys)

¡¡Está VIVO!!!

Observando la migración: *¡Está vivo!*



Observamos que el conjunto fuerte se iba /separando/, llegando a casi dos mitades iguales a inicios del 2014; el fuerte empuje hacia la migración hecho por /keyring-maint/ en la segunda mitad del 2014 llevó a su casi-convergencia.

Lanzando hipótesis: *¿envejecimiento?*

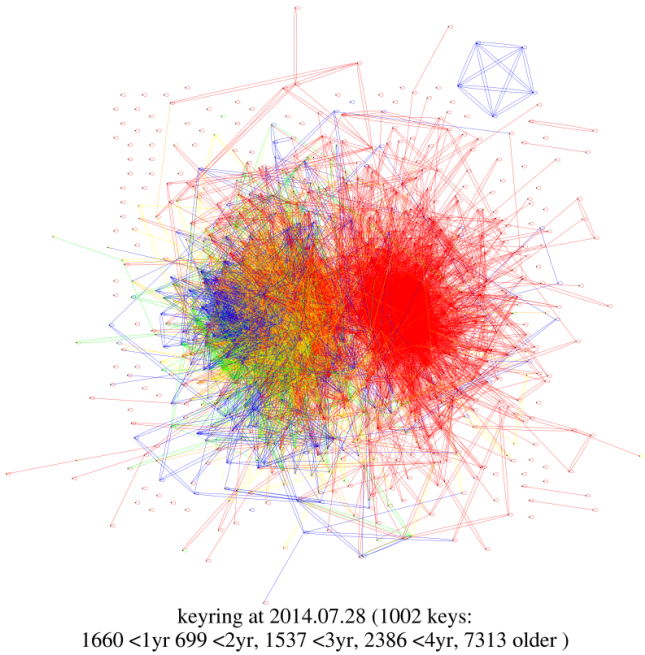
- La división comienza hacia 2011, mucho antes del reemplazo de llaves ≤ 1024
- ¿Llaves más viejas, desconexión social con las más nuevas?
  - Gráfica generada con un *modelo de tensión*, las firmas *acortan* la distancia entre grupos de llaves relacionadas
- ¿Generación de desarrolladores viejos que lentamente *se caen*?
- Interesante: El segundo conjunto se desvanece gradualmente, al corte de fines de 2014 era ya relativamente pequeño

Coloreando por edad

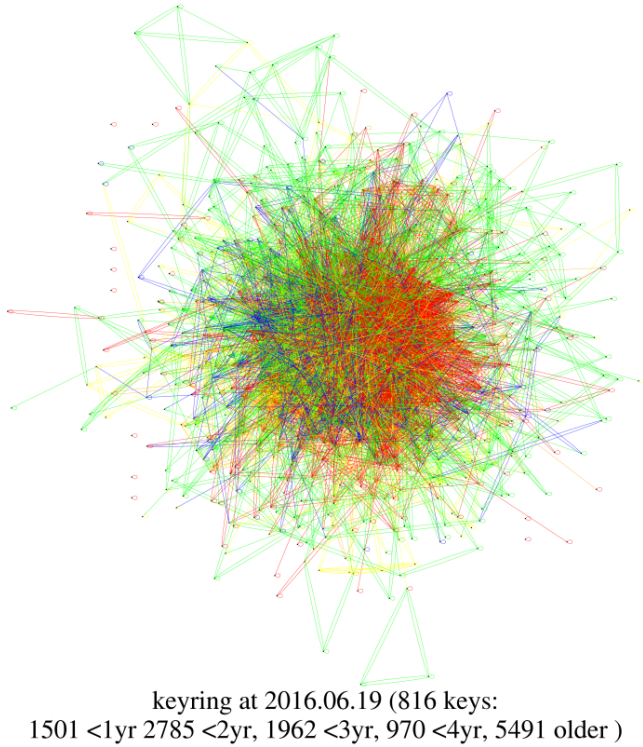
- Generamos nuevamente las mismas gráficas, pero indicando con color la *edad* de cada *firma*
  - Para esto, la llave en particular (cada nodo) no es ya relevante, nos interesan sólo los vértices
- Edad *relativa* al punto en el tiempo que se presenta
- Código de color, en años:

$$x \leq 1 < x \leq 2 < x \leq 3 < x \leq 4 < x$$

Los dos mismos llaveros: 2014.07.28 y 2016.06.19

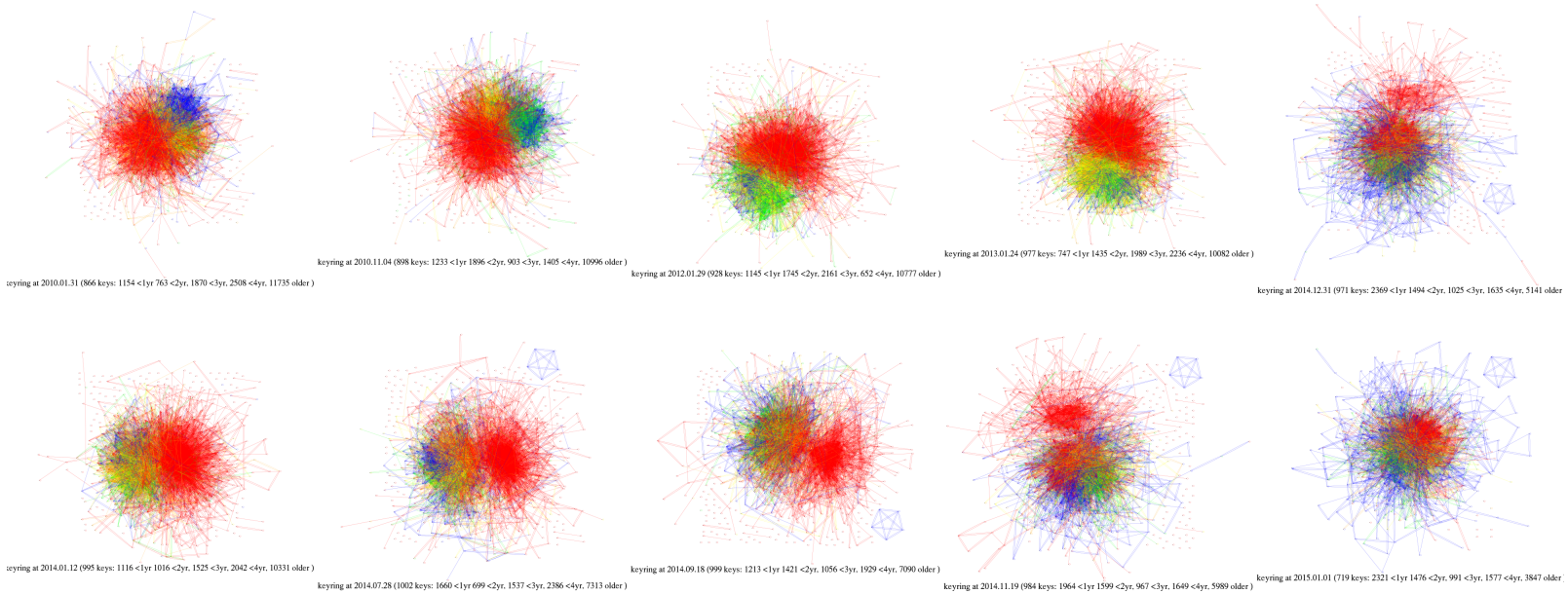


(a) Bola grande, roja, desconectada



(b) Ahora, una distribución más uniforme

Mismos diez vistazos



A lo largo de los años, la "bola" roja se mantiene separada, hasta desvanecerse y desaparecer

Aparentemente confirma la hipótesis

¿Qué sigue?

- Los últimos puntos presentados son aún trabajo en proceso
- ¿Qué busco analizando esta información?
  - Divulgar el modelo de confianza de *WoT curada*
  - Comprender al proyecto en el que colaboro
  - Explicar esta muestra de las relaciones sociales que lo sostienen
  - Presentar posibles mejoras a las prácticas de firmado al interior del proyecto Debian (¿y al exterior?)

¿Dudas, comentarios?

¡Gracias!

Gunnar Wolf
Debian, IIEc-UNAM, FI-UNAM

http://gwolf.org • http://keyring.debian.org • gwolf@debian.org

AB41 C1C6 8AFD 668C A045 EBF8 673A 03E4 C1DB 921F

Referencias I

Burnett, Mark (2011). *10,000 Top Passwords*.

Callas, Jon et al. (2007). *OpenPGP Message Format*. Tech. rep. RFC 4880. Internet Engineering Task Force.

Cederlöf, Jörgen (2004). *Dissecting the leaf of trust*.

Cooper, David et al. (2008). *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. Tech. rep. RFC 5280. Internet Engineering Task Force.

Debian Project (2015). *Developer locations*.

Dierks, Tim and Eric Rescorla (2008). *The Transport Layer Security (TLS) Protocol Version 1.2*. Tech. rep. RFC 5246. Internet Engineering Task Force.

Krafft, Martin (2008). *On the point of keysigning*.

McBurnett, Neal (1997). *PGP Web of Trust Statistics*.

Penning, Henk P. (2004). *Computing shortest paths in WOTs*.

– (2015). *analysis of the strong set in the PGP web of trust*.

Referencias II

Sassaman, Len and Phil Zimmerman (2006). *Efficient Group Key Signing Method*.

Smart, Nigel (2012). *ECRYPT II Yearly Report on Algorithms and Keysizes (2011-2012)*. Tech. rep. 7th Framework Programme, European Commission.

Srivastava, Manoj (2006). *Please revoke your signatures from Martin Krafft's keys*.

The CA / Browser Forum (2011). *Baseline Requirements for the Issuance and Management of Publicly-Trusted Certificates, v1.0*. Tech. rep.

Wikipedia (2015). *Public key infrastructure*.