Status of the Debian OpenPGP keyring

Daniel Kahn Gillmor, Jonathan McDowell, Gunnar Wolf

What do we do
Escaping algorithmic fragility: So far
Better key handling practices

Debian Project

DebConf 14 • Portland, Oregon
Status of the Debian OpenPGP keyring

Daniel Kahn Gillmor, Jonathan McDowell, Gunnar Wolf

1. What do we do

2. Escaping algorithmic fragility: So far

3. Better key handling practices
We maintain your keyrings

Maybe the naming is suboptimal...

debian-keyring-gpg 1003 keys
debian-maintainers-gpg 221 keys
debian-nonupload-gpg 10 keys
debian-role-keys-gpg 9 keys (unused)
emeritus-keyring-pgp 237 keys (unused)
removed-keys-gpg 750 keys (unused)
We maintain your keyrings

Maybe the naming is suboptimal...

debian-keyring-gpg 1003 keys
debian-maintainers-gpg 221 keys
debian-nonupload-gpg 10 keys
debian-role-keys-gpg 9 keys (unused)
ememitus-keyring-pgp 237 keys (unused)
removed-keys-gpg 750 keys (unused)
Active Debian keys

**Figura:** Evolution of the number of active keys, by type (inactive keys omitted)
Contenidos

Status of the Debian OpenPGP keyring

Daniel Kahn Gillmor, Jonathan McDowell, Gunnar Wolf

1. What do we do
2. Escaping algorithmic fragility: So far
3. Better key handling practices
Getting rid of PGPv3

- PGPv3: Weak keys (key fingerprint weakness, short keylength...)
- 2005: 261 PGPv3 keys, 903 GPG keys
- September 2010: zero PGPv3 keys
Getting rid of PGPv3

**Figura:** Number of keys in the DD keyring, by type
Forcefully removal

- Evolution of PGPv3 key migration was good
- Some people just didn't act on time
- In the end: Forcefully removed
  - 17 active keys removed
### Status of the Debian OpenPGP keyring

**Daniel Kahn Gillmor, Jonathan McDowell, Gunnar Wolf**

**What do we do**

**Escaping algorithmic fragility: So far**

**Better key handling practices**

---

**But... What's wrong with 1024D?**

---

<table>
<thead>
<tr>
<th>Security Level</th>
<th>Security Protection</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>32</td>
<td>Attacks in “real-time” by individuals</td>
</tr>
<tr>
<td>2.</td>
<td>64</td>
<td>Very short-term protection against small organizations</td>
</tr>
<tr>
<td>3.</td>
<td>72</td>
<td>Short-term protection against medium organizations, medium-term protection against small organizations</td>
</tr>
<tr>
<td>4.</td>
<td>80</td>
<td>Very short-term protection against agencies, long-term prot. against small organizations</td>
</tr>
<tr>
<td>5.</td>
<td>96</td>
<td>Legacy standard level</td>
</tr>
<tr>
<td>6.</td>
<td>112</td>
<td>Medium-term protection</td>
</tr>
<tr>
<td>7.</td>
<td>128</td>
<td>Long-term protection</td>
</tr>
<tr>
<td>8.</td>
<td>256</td>
<td>“Foreseeable future”</td>
</tr>
</tbody>
</table>
But... What’s wrong with 1024D?

Table 7.2: Key-size Equivalence.

<table>
<thead>
<tr>
<th>Security (bits)</th>
<th>RSA field (size)</th>
<th>RSA subfield</th>
<th>DLOG field</th>
<th>DLOG subfield</th>
<th>EC field</th>
<th>EC subfield</th>
</tr>
</thead>
<tbody>
<tr>
<td>48</td>
<td>480</td>
<td>480</td>
<td>96</td>
<td>96</td>
<td></td>
<td></td>
</tr>
<tr>
<td>56</td>
<td>640</td>
<td>640</td>
<td>112</td>
<td>112</td>
<td></td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>816</td>
<td>816</td>
<td>128</td>
<td>128</td>
<td></td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>1248</td>
<td>1248</td>
<td>160</td>
<td>160</td>
<td></td>
<td></td>
</tr>
<tr>
<td>112</td>
<td>2432</td>
<td>2432</td>
<td>224</td>
<td>224</td>
<td></td>
<td></td>
</tr>
<tr>
<td>128</td>
<td>3248</td>
<td>3248</td>
<td>256</td>
<td>256</td>
<td></td>
<td></td>
</tr>
<tr>
<td>160</td>
<td>5312</td>
<td>5312</td>
<td>320</td>
<td>320</td>
<td></td>
<td></td>
</tr>
<tr>
<td>192</td>
<td>7936</td>
<td>7936</td>
<td>384</td>
<td>384</td>
<td></td>
<td></td>
</tr>
<tr>
<td>256</td>
<td>15424</td>
<td>15424</td>
<td>512</td>
<td>512</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>RSA/DLOG Key</th>
<th>Security (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>512</td>
<td>50</td>
</tr>
<tr>
<td>768</td>
<td>62</td>
</tr>
<tr>
<td>1024</td>
<td>73</td>
</tr>
<tr>
<td>1536</td>
<td>89</td>
</tr>
<tr>
<td>2048</td>
<td>103</td>
</tr>
</tbody>
</table>
But... What’s wrong with 1024D?

<table>
<thead>
<tr>
<th>(3) Subcriber Certificates</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Validity period ending</strong></td>
</tr>
<tr>
<td>Digest algorithm</td>
</tr>
<tr>
<td>Minimum RSA modulus</td>
</tr>
</tbody>
</table>

CA / Browser Forum Baseline Requirements, v. 1.1.8 (as of 5 June 2014)
The situation WRT 1024D (1/6)

Figura: Number of Nonuploading DD keys, by key length — Absolute
The situation WRT 1024D (2/6)

**Figura:** Number of Nonuploading DD keys, by key length — Absolute
The situation WRT 1024D (3/6)

Figura: Number of Maintainer keys, by key length — Absolute
The situation WRT 1024D (4/6)

**Figura:** Number of Maintainer keys, by key length — Absolute
The situation WRT 1024D (5/6)

**Figura:** Number of DD keys, by key length — Absolute
The situation WRT 1024D (6/6)

**Figure**: Number of DD keys, by key length — Absolute
Warning

Until this point, we have stated facts.
Warning

Until this point, we have stated facts.

From this point on, it’s all a proposal for discussion.
The way out...?

Some ideas we put on the table

- Set a hard-cutoff date
  - Say, `Time.now() + 6.months`?
  - Or rather, the last day of this year?
  - Whatever: +- that timeframe

- But... What about key migration difficulties?
  - People *socially* disconnected from Debian
  - People *geographically* disconnected
  - *Consideration* to special cases

- But aren’t we all somehow... *Special*?
The way out...?

Some ideas we put on the table

- Set a hard-cutoff date
  - Say, `Time.now() + 6.months`?
  - Or rather, the last day of this year?
  - Whatever: +- that timeframe

- But... What about key migration difficulties?
  - People *socially* disconnected from Debian
  - People *geographically* disconnected
  - *Consideration* to special cases

- But aren’t we all somehow... *Special*?
What about signing based on... 

- Migration documents? 
- Non-personal contact? 

Personal identification: Unenforceable, but widely expected
What about signing based on... 

- Migration documents?
- Non-personal contact?

Personal identification: Unenforceable, but widely expected (And mostly honored)

Where should we encode this expectation? (i.e. DMUP and friends?)
Status of the Debian OpenPGP keyring

Daniel Kahn Gillmor, Jonathan McDowell, Gunnar Wolf

1. What do we do

2. Escaping algorithmic fragility: So far

3. Better key handling practices
Key handling practices should improve

- Many people don’t handle their keys carefully enough
  - Separating master keyring from key *du jour*
  - Key expiration
  - Revocation certificates
  - Proper offline storage for master private key material
  - . . .
- Cannot have technical solutions for social issues. . .
Could we require keys to have a set expiration date?
- Say, requiring 3 years expiration (+maintaining the key updated, of course)
- Demonstrable key update activity (HKPS)
- Set a timeframe for expiring keys to be enforced
- Periodic service where we inform you your expiration is soon...
Questions?

keyring-maint@debian.org