SHA-1 und OpenPGP/GnuPG

Christian Aistleitner
christian@quelltextlich.at

Linuxwochenende, Wien
24. November 2012

Outline

Outline

SHA-1

SHA-1

- Hashfunktion, die eine Nachricht in einen 160-bit Wert umsetzt
  (e.g.: „`Linuxwochenende? Ja!`" → „`9b4c8f162cf6f0ae30e8bd10118e575100d46c9b`")
- *kryptographische* Hashfunktion
  (e.g.: „`Linuxwochenende? Na!`" → „`5c89e4889a10aa1ee5801ed2d7a10a550dcc2868`")
- leicht zu implementieren
- weit verbreitet (e.g.: OpenPGP, GnuPG, git)
- 1995 in FIPS PUB 180-1 veröffentlicht

SHA-1 ist kaputt!

SHA-1 has been broken. Not a reduced-round version. Not a simplified version. The real thing.

Bruce Schneier (2005)

Use of SHA-1 should now be avoided.

Apache Software Foundation (Zumindest seit 2009)

Due to weaknesses found with the SHA1 hashing algorithm Debian prefers to use keys that are at least 2048 bits and preferring SHA2.

Debian (Zumindest seit 2009)

> Although SHA1 is considered to be broken by some, [...]
That is plain nonsense.

Werner Koch (2012)

SHA-1: Status quo (alle 80 Runden)

Komplexität (compressions)	Schätzung (Jahre)[1]	Typ	Quelle	Jahr
2^{160}	$2.0 \cdot 10^{31}$	preimage	(bruteforce)	1995
2^{80}	17000000	collision	(bruteforce)	1995
(avg.) 2^{77}	2000000	chosen pfx	Stevens	2012
2^{69}	8100	identical pfx	Wang, Yin, Yu	2005
$2^{60.3}$ to $2^{65.3}$	20 to 630	identical pfx	Stevens	2012

[1] basierend auf einer Messung von 2300 Millionen compressions/Sekunde auf ATI HD 5970 in 2010.

It's time to walk, but not run, to the fire exits. You don't see smoke, but the fire alarms have gone off.

Jon Callas ($\leq$2005), quoted by Bruce Schneier

Outline

OpenPGP

- IETF RFC 4880 „OpenPGP Message Format" (STD1) (November 2007)
- Software, die OpenPGP implementiert: GnuPG, PGP, Hushmail
- Übertragenen Daten sind „OpenPGP message".
- „OpenPGP message" ist ein Abfolge von „packet"s
- „packet" ist zum Beispiel
  - Einzelner öffentlicher Schlüssel
  - Einzelner privater Schlüssel
  - Einzelne Unterschrift
  - Signierte Daten
  - Verschlüsselte Daten

OpenPGP packets mit erzwungenem SHA-1

Abschnitt	Name	Problematisch?
§5.5.3.	Secret-Key Packet Formats (usage 254)	
§5.14.	Modification Detection Code Packet	
§12.2.	Key IDs and Fingerprints	

OpenPGP packets mit wählbarer Hashfunktion

Abschnitt	Name
§3.7.1.1.	Simple S2K
§3.7.1.2.	Salted S2K
§3.7.1.3.	Iterated and Salted S2K
§5.2.2.	Version 3 Signature Packet Format
§5.2.3.	Version 4 Signature Packet Format
§5.2.3.8.	Preferred Hash Algorithms
§5.2.3.25.	Signature Target
§5.4.	One-Pass Signature Packets

$\rightarrow$ Hashfunktion wird über 8-bit Wert ausgewählt

Outline

OpenPGP packets mit wählbarer Hashfunktion

Abschnitt	Name	Bei GnuPG
§3.7.1.1.	Simple S2K	?
§3.7.1.2.	Salted S2K	?
§3.7.1.3.	Iterated and Salted S2K	?
§5.2.2.	Version 3 Signature Packet Format	?
§5.2.3.	Version 4 Signature Packet Format	?
§5.2.3.8.	Preferred Hash Algorithms	?
§5.2.3.25.	Signature Target	?
§5.4.	One-Pass Signature Packets	?

- 8-bit Wert zur Auswahl der Hashfunktion
  - 2 $\rightarrow$ SHA-1
  - 10 $\rightarrow$ SHA-512

Schlüsselerzeugung

```
$ gpg --gen-key
gpg (GnuPG) 2.0.19; Copyright (C) 2012 Free Software Foundation, Inc.
[...]
public and secret key created and signed.

gpg: checking the trustdb
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0  valid:   1  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 1u
gpg: next trustdb check due at 2012-10-18
pub   2048R/DA6AD20A 2012-10-17 [expires: 2012-10-18]
      Key fingerprint = 2FB1 8AD2 9D58 307C 51E6  CDF3 9A82 03DD[...]
uid                  John Doe
sub   2048R/32EA7448 2012-10-17 [expires: 2012-10-18]
```

String-to-key (I)

```
$ gpg --export-secret-keys 0xDA6AD20A | gpg --list-packets
:secret key packet:
[...]
        iter+salt S2K, algo: 3, SHA1 protection, hash: 2, salt: [...]
[...]
:secret sub key packet:
[...]
        iter+salt S2K, algo: 3, SHA1 protection, hash: 2, salt: [...]
[...]
```

String-to-key (II)

```
$ echo "s2k-digest-algo SHA512" >>~/.gnupg/gpg.conf
$ gpg --edit-key 0xDA6AD20A
[...]
gpg> passwd
Key is protected.

You need a passphrase to unlock the secret key for
[...]
gpg> save
$ gpg --export-secret-keys 0xDA6AD20A | gpg --list-packets
:secret key packet:
[...]
        iter+salt S2K, algo: 3, SHA1 protection, hash: 10, salt:[...]
[...]
:secret sub key packet:
[...]
        iter+salt S2K, algo: 3, SHA1 protection, hash: 10, salt:[...]
[...]
```

OpenPGP packets mit wählbarer Hashfunktion

Abschnitt	Name	Bei GnuPG
§3.7.1.1.	Simple S2K	s2k-digest-algo + passwd
§3.7.1.2.	Salted S2K	s2k-digest-algo + passwd
§3.7.1.3.	Iterated and Salted S2K	s2k-digest-algo + passwd
§5.2.2.	Version 3 Signature Packet Format	?
§5.2.3.	Version 4 Signature Packet Format	?
§5.2.3.8.	Preferred Hash Algorithms	?
§5.2.3.25.	Signature Target	?
§5.4.	One-Pass Signature Packets	?

Signieren

```
$ echo "abc" | gpg --default-key 0xDA6AD20A --sign | gpg --list-packets

You need a passphrase to unlock the secret key for
user: "John Doe"
2048-bit RSA key, ID DA6AD20A, created 2012-10-17

[...]
:compressed packet: algo=1
:onepass_sig packet: keyid 9A8203DDDA6AD20A
        version 3, sigclass 0x00, digest 2, pubkey 1, last=1
[...]
:signature packet: algo 1, keyid 9A8203DDDA6AD20A
[...]
        digest algo 2, begin of digest dc cb
[...]
```

Signieren (II)

```
$ echo "digest-algo SHA512" >>~/.gnupg/gpg.conf
$ echo "abc" | gpg --default-key 0xDA6AD20A --sign | gpg --list-packets

You need a passphrase to unlock the secret key for
user: "John Doe"
2048-bit RSA key, ID DA6AD20A, created 2012-10-17

[...]
:compressed packet: algo=1
:onepass_sig packet: keyid 9A8203DDDA6AD20A
        version 3, sigclass 0x00, digest 10, pubkey 1, last=1
[...]
:signature packet: algo 1, keyid 9A8203DDDA6AD20A
        version 4, created 1350496299, md5len 0, sigclass 0x00
        digest algo 10, begin of digest b1 40
[...]
```

OpenPGP packets mit wählbarer Hashfunktion

Abschnitt	Name	Bei GnuPG
§3.7.1.1.	Simple S2K	s2k-digest-algo + passwd
§3.7.1.2.	Salted S2K	s2k-digest-algo + passwd
§3.7.1.3.	Iterated and Salted S2K	s2k-digest-algo + passwd
§5.2.2.	Version 3 Signature Packet Format	?
§5.2.3.	Version 4 Signature Packet Format	digest-algo
§5.2.3.8.	Preferred Hash Algorithms	?
§5.2.3.25.	Signature Target	?
§5.4.	One-Pass Signature Packets	*automatisch*

Signieren (III)

```
$ echo "abc" | gpg --default-key 0xDA6AD20A --sign --force-v3-sigs \
  | gpg --list-packets

You need a passphrase to unlock the secret key for
user: "John Doe"
2048-bit RSA key, ID DA6AD20A, created 2012-10-17

[...]
:compressed packet: algo=1
:onepass_sig packet: keyid 9A8203DDDA6AD20A
        version 3, sigclass 0x00, digest 10, pubkey 1, last=1
[...]
:signature packet: algo 1, keyid 9A8203DDDA6AD20A
        version 3, created 1350496570, md5len 5, sigclass 0x00
        digest algo 10, begin of digest aa 92
[...]
```

OpenPGP packets mit wählbarer Hashfunktion

Abschnitt	Name	Bei GnuPG
§3.7.1.1.	Simple S2K	s2k-digest-algo + passwd
§3.7.1.2.	Salted S2K	s2k-digest-algo + passwd
§3.7.1.3.	Iterated and Salted S2K	s2k-digest-algo + passwd
§5.2.2.	Version 3 Signature Packet Format	digest-algo
§5.2.3.	Version 4 Signature Packet Format	digest-algo
§5.2.3.8.	Preferred Hash Algorithms	?
§5.2.3.25.	Signature Target	?
§5.4.	One-Pass Signature Packets	*automatisch*

Bevorzugte Hashfunktion

```
$ gpg --export 0xDA6AD20A | gpg --list-packets
:public key packet:
[...]
:user ID packet: "John Doe"
:signature packet: algo 1, keyid 9A8203DDDA6AD20A
[...]
        hashed subpkt 21 len 5 (pref-hash-algos: 8 2 9 10 11)
[...]
```

Weitere Hashfunktionen-Ids:

- 8 → SHA256
- 9 → SHA384
- 11 → SHA224

Bevorzugte Hashfunktion (II)

```
$ echo "default-preference-list SHA512 SHA384 SHA256 SHA224 SHA1 "\
"AES256 AES192 AES CAST5 ZLIB BZIP2 ZIP Uncompressed" \
>>~/.gnupg/gpg.conf
$ gpg --edit-key 0xDA6AD20A
[...]
gpg> setpref
Set preference list to:
     Cipher: AES256, AES192, AES, CAST5, 3DES
     Digest: SHA512, SHA384, SHA256, SHA224, SHA1
     Compression: ZLIB, BZIP2, ZIP, Uncompressed
     Features: MDC, Keyserver no-modify
Really update the preferences? (y/N) y
[...]
gpg> save
```

Bevorzugte Hashfunktion (III)

```
$ gpg --export 0xDA6AD20A | gpg --list-packets
:public key packet:
[...]
:user ID packet: "John Doe"
:signature packet: algo 1, keyid 9A8203DDDA6AD20A
[...]
        hashed subpkt 21 len 5 (pref-hash-algos: 10 9 8 11 2)
[...]
```

OpenPGP packets mit wählbarer Hashfunktion

Abschnitt	Name	Bei GnuPG
§3.7.1.1.	Simple S2K	s2k-digest-algo + passwd
§3.7.1.2.	Salted S2K	s2k-digest-algo + passwd
§3.7.1.3.	Iterated and Salted S2K	s2k-digest-algo + passwd
§5.2.2.	Version 3 Signature Packet Format	digest-algo
§5.2.3.	Version 4 Signature Packet Format	digest-algo
§5.2.3.8.	Preferred Hash Algorithms	default-preference-list + setpref
§5.2.3.25.	Signature Target	?
§5.4.	One-Pass Signature Packets	*automatisch*

Signature Target

- OpenPGP:

```
29 = Reason for Revocation
30 = Features
31 = Signature Target
32 = Embedded Signature
```

- GnuPG (common/openpgpdefs.h):

```
SIGSUBPKT_REVOC_REASON  = 29, /* Reason for revocation. */
SIGSUBPKT_FEATURES      = 30, /* Feature flags. */

SIGSUBPKT_SIGNATURE     = 32, /* Embedded signature. */
```

OpenPGP packets mit wählbarer Hashfunktion

Abschnitt	Name	Bei GnuPG
§3.7.1.1.	Simple S2K	s2k-digest-algo + passwd
§3.7.1.2.	Salted S2K	s2k-digest-algo + passwd
§3.7.1.3.	Iterated and Salted S2K	s2k-digest-algo + passwd
§5.2.2.	Version 3 Signature Packet Format	digest-algo
§5.2.3.	Version 4 Signature Packet Format	digest-algo
§5.2.3.8.	Preferred Hash Algorithms	default-preference-list + setpref
§5.2.3.25.	Signature Target	*nicht implementiert*
§5.4.	One-Pass Signature Packets	*automatisch*

Cheater!

```
$ gpg --export 0xDA6AD20A | gpg --list-packets
:public key packet:
[...]
:user ID packet: "John Doe"
:signature packet: algo 1, keyid 9A8203DDDA6AD20A
        version 4, created 1350497669, md5len 0, sigclass 0x13
        digest algo 2, begin of digest 6a c6
:public sub key packet:
[...]
:signature packet: algo 1, keyid 9A8203DDDA6AD20A
        version 4, created 1350493810, md5len 0, sigclass 0x18
        digest algo 2, begin of digest 54 65
```

Cheater! (II)

```
$ echo "cert-digest-algo SHA512" >>~/.gnupg/gpg.conf
$ gpg --edit-key 0xDA6AD20A
[...]
gpg> setpref
[...]
gpg> key 1
[...]
gpg> expire
[set expiration time again.
 For this to update the signature's algorithm, use patches from
 gnupg-dev mailinglist]
gpg> save
```

Cheater! (III)

```
$ gpg --export 0xDA6AD20A | gpg --list-packets
:public key packet:
[...]
:user ID packet: "John Doe"
:signature packet: algo 1, keyid 9A8203DDDA6AD20A
        version 4, created 1350500867, md5len 0, sigclass 0x13
        digest algo 10, begin of digest 9c 74
[...]
:public sub key packet:
[...]
:signature packet: algo 1, keyid 9A8203DDDA6AD20A
        version 4, created 1350501179, md5len 0, sigclass 0x18
        digest algo 10, begin of digest 09 b9
[...]
```

OpenPGP packets mit wählbarer Hashfunktion

Abschnitt	Name	Bei GnuPG
§3.7.1.1.	Simple S2K	s2k-digest-algo + passwd
§3.7.1.2.	Salted S2K	s2k-digest-algo + passwd
§3.7.1.3.	Iterated and Salted S2K	s2k-digest-algo + passwd
§5.2.2.	Version 3 Signature Packet Format	{cert-,}digest-algo
§5.2.3.	Version 4 Signature Packet Format	{cert-,}digest-algo
§5.2.3.8.	Preferred Hash Algorithms	default-preference-list + setpref
§5.2.3.25.	Signature Target	*nicht implementiert*
§5.4.	One-Pass Signature Packets	*automatisch*

- Existierende Signaturen aktualisieren
  - Primärer Schlüssel: setpref (benötigt Patch)
    or Signatur löschen und neu signieren
  - Unterschlüssel: expire (benötigt den gleichen Patch)
  - Schlüssel/User-Ids von anderen: neu signieren

OpenPGP §9.4. Hash algorithms

ID	Algorithm	Text Name
1	MD5	MD5
2	SHA-1	SHA1
3	RIPE-MD/160	RIPEMD160
4	Reserved	
5	Reserved	
6	Reserved	
7	Reserved	
8	SHA256	SHA256
9	SHA384	SHA384
10	SHA512	SHA512
11	SHA224	SHA224
100 to 110	Private/Experimental algorithm	

Implementations MUST implement SHA-1. Implementations MAY implement other algorithms. MD5 is deprecated.

Interoperabilität von SHA2 Schlüsseln

	GnuPG-Version	Klappt's?	Verfügbar in
	master (v1)	ja	
	v1.4.12	ja	Gentoo, Debian wheezy
	v1.4.11	ja	Ubuntu natty–quantal
2009-09-02	v1.4.10	ja	Debian squeeze, Ubuntu lucid
	master (v2)	ja	
	v2.0.19	ja	Gentoo, Arch, openSUSE 12.2, Debian wheezy
	v2.0.18	ja	openSUSE 12.1
	v2.0.17	ja	Ubuntu oneiric–quantal
	v2.0.16	ja	openSUSE 11.4
	v2.0.15	ja	
	v2.0.14	ja	Ubuntu lucid–natty
2009-09-04	v2.0.13	ja	Debian squeeze

Zusammenfassung

- RFC 4880 (OpenPGP) erlaubt SHA-1 auszuweichen
- Interoperabilität zwar typischerweise, aber nicht automatisch gegeben
  (Kein Problem für die GnuPGs der letzten drei Jahre)
- Wechsel zu SHA512 dauert $<$10 Minuten
- Letzte Schwierigkeit: Aktualisieren bestehender Signaturen

- Aufgemerkt!
  - Das bewahrt dich nicht davor SHA-1 Summen zu bekommen.
  - Das bewahrt dich nicht davor SHA-1 Summen zu versenden.

SHA2 für GnuPG Zusammenfassung

- Folgende Zeilen an gpg.conf anhängen

```
s2k-digest-algo SHA512
digest-algo SHA512
default-preference-list SHA512 SHA384 SHA256
   SHA224 SHA1 AES256 AES192 AES CAST5 ZLIB
   BZIP2 ZIP Uncompressed
cert-digest-algo SHA512
```

- Neue Schlüssel
  - Wie üblich erzeugen. Sie werden nun auf SHA512 voreingestellt.
- Bereits existierende Schlüssel
  - passwd für die Schlüssel ausführen
  - setpref für die Schlüssel ausführen
  - Signaturen aktualisieren (Patches!)