Hashfunktionen gebrochen

Rüdiger Weis, Stefan Lucks

In den letzten Monaten wurden auf dem Gebiet der kryptographischen Hashfunktionen bedeutende neue Erkenntnisse gewonnen. Dies betrifft unter anderem neue Angriffe auf die schon etwas veralteten, aber noch immer in vielen Anwendungen genutzten Hashfunktionen MD4 und MD5. Noch mehr beunruhigt, dass es um die Sicherheit von SHA-1 nicht zum Besten bestellt ist. Hashfunktionen, insbesondere der SHA-1, spielen für die Fälschungssicherheit digitaler Unterschriften eine entscheidende Rolle.

Prof. Dr. rer. nat. Rüdiger Weis

zur Zeit Forscher in der Gruppe von Andrew S. Tanenbaum an der Vrije Universiteit Amsterdam und Leiter der Cryptolabs Amsterdam. Forschungsschwerpunkte: Kryptographie, Computersicherheit, Rechnernetze, Betriebssysteme.
E-Mail: rcw@cryptolabs.org

Privatdozent Dr. Stefan Lucks

arbeitet als Oberassistent am Lehrstuhl für Theoretische Informatik an der Universität Mannheim.
Forschungsschwerpunkte: Kryptographie, Computersicherheit, Komplexitätstheorie.
E-Mail: lucks@th.informatik.uni-mannheim.de

1 Kryptographische Hashfunktionen

Hashfunktionen erzeugen für Eingaben (praktisch) beliebiger Länge einen kurzen, typischerweise zwischen 128 und 512 bit langen „fingerprint".

Für kryptographische Anwendungen ist zusätzlich die Eigenschaft der Kollisionsresistenz von besonderer Bedeutung. Als Kollision bezeichnet man zwei verschiedene Nachrichten M und M', die auf den selben Hashwert h(M)=h(M') abgebildet werden. Da es (mehr oder weniger) unendlich viele Eingaben, aber nur endlich viele Hashwerte gibt, ist die Existenz von Kollisionen nicht zu verhindern.

Eine Hashfunktion ist kollisionsresistent, wenn der Rechenaufwand, irgend eine Kollision zu finden, so riesig ist, dass man ihn als praktisch unmöglich ansehen kann. Aktuell gelten 2^{80} Rechenoperationen als praktisch unmöglich. Wegen des so genannten „Geburtstagsangriffs"[1] muss eine Hashfunktion für dieses Sicherheitsniveau mindestens einen 160 bit Hashwert liefern. In dem Maß, in dem Rechner immer schneller werden („Moore's Law") muss man diese Grenze anpassen – also im Laufe der Jahre steigern. Schon bald werden 160 bit nicht mehr ausreichen. Eine Hashfunktion, die nicht kollisionsresistent ist, gilt als gebrochen.

Kryptographische Hashfunktionen werden in vielen Bereichen der IT-Sicherheit eingesetzt, beispielsweise bei Integritätsprüfungen (z. B. tripwire). Dabei wird von als „harmlos" und „nützlich" eingestuften Programmen ein Hash-Wert abgespeichert. Vor Ausführung des Programms wird dessen Hashwert mit dem Referenz-Hashwert verglichen. Stimmen die beiden nicht überein, wird angenommen, dass das Programm manipuliert wurde, z. B. von einem Computervirus. Stimmen Soll- und Ist-Hashwert dagegen überein, dann kann es sich nur um das Originalprogramm handeln – sofern die Hashfunktion kollisionsresistent ist.

Besondere Bedeutung haben Hashfunktionen für digitale Signaturen: In der Praxis unterschreibt man nicht die Nachricht, sondern ihren Hashwert („Hash-Then-Sign Paradigma"). Die Handhabung beliebig langer zu unterschreibender Nachrichten wäre sonst problematisch. Gelänge es einem Angreifer, zwei kollidierende Nachrichten M und M' zu finden, gilt also h(M)=h(M'), dann ist eine gültige digitale Signatur für M auch gültig für M'. Schwächen von Hashfunktionen bezüglich der Kollisionsresistenz können daher die Fälschungssicherheit und damit die Rechtsicherheit digital signierter Verträge in Frage stellen.

Kryptographisch gilt eine Hashfunktion auch dann als nicht kollisionsresistent, also als gebrochen, wenn die Nachrichten M und M'. die ein Angreifer finden kann, mehr oder weniger zufällig sind – M und M' brauchen keine gültigen und sinnvollen Vertragstexte zu sein. In der Praxis ist das jedoch eine wichtige Voraussetzung, um einen Angriff tatsächlich umsetzen zu können.

2 MD4 und MD5

Ron Rivest veröffentliche 1990 das Design für die 128 bit Hashfunktion MD4 [Ri90]. MD4 ist eine iterierende 3-ründige Hashfunktion welche nach dem Damgård-Merkle Konstruktionsprinzip entwickelt wurde. MD4 ist recht einfach implementierbar und besonders für die schnelle Verarbeitung auf 32-bit Prozessoren optimiert.

Da schon früh kryptographische Schwächen von einfacheren MD4-Varianten aufgezeigt werden konnten, machte sich Ron Rivest bereits kurze Zeit später an die Entwicklung einer verbesserten Version.

1991 veröffentlichte Ron Rivest die Hashfunktion MD5 [Ri91]. Neben einiger Modifikationen in der Kompressionsfunktion wurde eine zusätzliche 4. Runde spezifiziert. Hierdurch ist MD5 etwas langsamer als MD4. Wie MD4 liefert MD5 auch lediglich einen 128 bit langen Message-Digest. MD5 wurde die erste in der Praxis

[1] Siehe Fox, Gateway, DuD 11/2001, S. 684.

weit verbreitete Hashfunktion, und auch heute ist sie noch in vielen Anwendungen anzutreffen – auch wenn eine Hashfunktion mit einem 128 bit Hashwert schon lange nicht mehr den aktuellen Anforderungen an die Mindestlänge genügt. Denn bei einer 128 bit Hashfunktion findet der bereits erwähnte Geburtstagsangriff eine Kollision mit einem Rechenaufwand von maximal 2^{64} Hash-Aufrufen – selbst wenn die Hashfunktion keine spezifischen Schwächen aufweist.

2.1 MD4 mit Papier und Bleistift angreifbar

Schon 1996 zeigte Hans Dobbertin, wie man auf effiziente Weise eine Kollision für MD4 finden kann [Do96a]. Vereinfacht gesagt, wählte Dobbertin Nachrichten M und M' mit einer bestimmten Differenz, die mit der Wahrscheinlichkeit 2^{32} kollidierten. Man wählt so lange neue Nachrichten M und M', bis diese kollidieren [Do97,Do98].

Dobbertin benutzte für den Angriff einen Computer – freilich keinen Superrechner, sondern einen einfachen PC. Acht Jahre später wurde nicht einmal mehr ein PC gebraucht. Auf der Rump-Session der Crypto 2004 präsentierten die chinesischen Forscher Xiaoyun Wang, Xiaoyun Lai, Dengguo Feng, Hui Chen und Hongbo Yu Angriffe auf MD4, die teilweise praktisch per Hand durchgeführt werden können.

Die neue Attacke findet eine Kollision mit einer Wahrscheinlichkeit von 2^{-6} bis 2^{-8} und einem überraschend niedrigen Aufwand von 2^8 MD4-Berechnungen.

Die Autoren geben auch eine Anwendung ihrer Techniken auf RIPEMD an, welche mit einer Wahrscheinlichkeit von 2^{-17} eine Kollision mit einer Angriffskomplexität von 2^{19} findet.

2.2 MD5 gebrochen

Am 17. August 2004 zeigten Wang, Feng, Lai und Yu Kollisionen für MD5 auf. Der praktische Angriff benötigte nach ihren Angaben nur zwischen 15 Minuten eine Stunde auf einem Workstation Cluster.

Interessanterweise handelt sich es bei der vorgestellten Methode um einen speziellen differentiellen Angriff, welcher Differentiale bezüglich modularer Subtraktion verwendet. Diese Angriffsmethode findet nach Angaben der Autoren für MD4 eine Kollision mit einem Aufwand von 2^{23} MD4-Berechnungen, 2^{13} für HAVAL-128 [ZPS92], 2^{33} für RIPEMD und 2^{62} für SHA-0.

2.3 Verwendungsstopp für MD4 und MD5

Angesichts der aktuellen Entwicklungen muss die dringende Empfehlung erteilt werden, MD5 und MD4 nicht mehr zu verwenden. Im Bereich von digitalen Signaturen sollte auf die Nutzung von MD4 und MD5 – soweit diese ohnehin obsoleten Funktionen noch genutzt werden – sofort verzichtet werden. Darüber hinaus ist eine Analyse möglicher Auswirkungen bezüglich bereits geleisteter Signaturen angeraten.

Das US-amerikanische National Institute of Standards and Technology (NIST) William Burr, Manager Security Technology Group, mahnte ebenfalls in einem Interview vom Februar 2005 insbesondere für die sensiblen Bereiche von Zertifikaten und Digitalen Signaturen einen sofortigen Verwendungsstopp von MD5 an:

> „*If by some chance you are still using MD5 in certificates or for digital signatures, you should stop.*" [Ol05]

3 SHA-0 und SHA-1

Der SHA-Algorithmus (Secure Hash Standard, inzwischen als SHA-0 bezeichnet) [NI92] basiert auf einen von der amerikanischen National Security Agency (NSA) stammenden Design, dessen Designprinzipien und Kriterien nicht veröffentlicht wurden. Schon kurz nach seiner Veröffentlichung wurde dieser Standard mit dem Hinweis auf (unveröffentlichte) Sicherheitslücken zu SHA-1 [NI94] modifiziert:

> „*SHA-1 is a technical revision of SHA (FIPS 180). A circular left shift operation has been added to the specifications in section 7, line b, page 9 of FIPS 180 and its equivalent in section 8, line c, page 10 of FIPS 180. This revision improves the security provided by this standard.*"

SHA-0 und SHA-1 lehnen sich nahe an den MD4 Hashalgorithmus an: „*The SHA-1 is based on principles similar to those used by Professor Ronald L. Rivest of MIT when designing the MD4 message digest algorithm (…), and is closely modelled after that algorithm.*"

3.1 SHA-0 gebrochen

Nach dem Rückzug von SHA(-0) dauerte es nicht lange, bis erste Schwächen von dieser Hashfunktion auch von den öffentlich forschenden und frei publizierenden Forschern bestätigt und publiziert wurden. Im Jahr 2004 präsentierten dann mehreren Gruppen unabhängig voneinander gravierende Angriffe auf SHA-0.

Dabei stellten manche auch die Frage, ob diese Angriffe auch auf SHA-1 übertragbar sind [We05]. Schließlich unterscheidet sich der SHA-1 von SHA-0 ja nur in einer zusätzlichen, extrem einfachen Operation, einer zyklischen Linksrotation eines 32-bit Wortes.

3.2 SHA-1 wahrscheinlich gebrochen

Die Antwort ließ nur einige Monate auf sich warten. Mitte Februar 2005 verbreiteten die chinesischen Forscher Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu eine dreiseitige Notiz, dass sie Kollisionen für SHA-1 mit einem Zeitaufwand von 2^{69} Aufrufen der Hashfunktion berechnen können [Sc05]. Sollte sich dieses Ergebnis bestätigen, gerät das Finden derartiger Kollisionen in die Reichweite moderner Supercomputer. Zum Redaktionsschluss dieses Artikels war die wissenschaftliche Diskussion über diese Arbeit allerdings noch nicht abgeschlossen.

3.3 SHA-1 und digitale Signaturen

SHA-1 ist für qualifizierte Signaturen nach dem Signaturgesetz zugelassen. Alternativen, wie z. B. die Hashfunktionen des SHA-2 Standards (s.u.), werden von signaturfähigen Smartcards oft nicht unterstützt – der Nutzer hat dann keine Alternative zur Nutzung des SHA-1. Oder sollte er auf digitale Signaturen ganz verzichten? Wie sicher kann man sich bei Signaturen noch sein?
Zunächst einige gute Nachrichten:

- Der Angriff der chinesischen Forscher impliziert keine Gefahr für Unterschriften, die in der Vergangenheit geleistet wurden. (Es sei denn, Kriminelle kannten den Angriff schon länger und habe ihn in der Vergangenheit bereits benutzt).

Denn die Aufgabe, zu einer bereits unterschriebenen Nachricht M eine Nachricht M' zu finden, zu der die Unterschrift wegen einer Kollision H(M)=H(M') passt, ist

schwieriger als die Aufgabe, eine allgemeine Kollision zu finden. Der Angriff der chinesischen Forscher leistet nur letzteres.

- Der Rechenaufwand für den Angriff von Wang und Co-Autoren ist mit 2^{69} Aufrufen der Hashfunktion extrem groß, und nur von einer hervorragend finanzierten und sehr stark motivierten Organisation zu leisten.
- Bisher scheinen die chinesischen Forscher nur in der Lage zu sein, mehr oder weniger zufällige kollidierende Nachrichten M und M' zu finden – nicht jedoch sinnvolle und verständliche Vertragstexte (gleich welcher Sprache).

Leider gibt es auch schlechte Nachrichten:

- Der Angriff von Wang und Co-Autoren lässt es bedrohlicher erscheinen, Nachrichten zu unterschreiben, die von einer anderen Partei ausformuliert wurden.
- Die Erfahrung aus der Vergangenheit ist, dass Angriffe immer besser werden. Es wäre also wenig überraschend, wenn die Analyse von SHA-1 in den nächsten Monaten und Jahren noch verbessert und der Rechenaufwand für das Problem, Kollisionen für SHA-1 zu finden, noch verringert werden würde.
- Die Hauptbedingung für den Angriff von Wang und Co auf SHA-0 und MD5 besteht darin, dass die Nachrichten M und M' eine bestimmte Differenz aufweisen. Das lässt dem Angreifer große Freiheiten, die Wahl von M und M' einzuschränken, statt sie dem Zufall zu überlassen.

  Ein Schritt in diese Richtung gelang Arjen Lenstra, Wang und Benne de Weger, die am 1. März meldeten, dass die Methode zum Berechnen von MD5-Kollisionen genutzt haben, um spezifische Kollisionen zu berechnen, die die formalen Anforderungen an X.509 Zertifikate erfüllen. [LWW05]

  Aus ähnlichen Gründen ist es auch Dobbertin bei MD4 gelungen, gezielte Kollisionen zu finden (zwei Fassungen eines Kaufvertrages mit sehr unterschiedlichen Kaufpreisen). Es ist anzunehmen, dass der Angriff auf SHA-1 ähnlich funktioniert wie der Angriff auf SHA-0.

Doch unabhängig davon, ob sich die Angaben der chinesischen Forscher bestätigen, ist es an der Zeit, die Hashfunktion SHA-1 durch Alternativen zu ersetzen. Die Angriffe auf andere Hashfunktionen, einschließlich SHA-0, sind zu eindrucksvoll, als dass man SHA-1 noch lange vertrauen sollte. Insbesondere wenn sich die Angaben von Wang und ihren Co-Autoren bestätigen sollten, ist jedoch eine zögerliche Haltung bei der Umstellung von SHA-1 auf Alternativen nicht mehr zu rechtfertigen.

4 Multi-Kollisionen

Buchstäblich alle in der Praxis eingesetzten Hashfunktionen, MD4, MD5, SHA-0, SHA-1, SHA-XXX, RIPEMD, RIPEMD-169, ... orientieren sich in ihrem Design an den theoretischen Grundlagenarbeiten von Merkle [Me89] und Damgård [Da89]. Das Merkle-Damgård-Designprinzip besteht darin, eine Hashfunktion für beliebig lange Eingaben dadurch zu realisieren, dass man eine Mini-Hashfunktion (die sogenannte „Kompressionsfunktion“) für Eingaben fester Länge iteriert. Merkle und Damgård bewiesen, dass die iterierte Hashfunktion kollisionsresistent ist, wenn die Kompressionsfunktion ihrerseits kollisionsresistent ist. Die Aufgabe, eine kollisionsresistente Hashfunktion zu konstruieren, wurde damit auf die etwas überschaubarere Aufgabe reduziert, eine kollisionsresistente Kompressionsfunktion zu entwerfen.

Joux [Jo04] präsentierte auf der Crypto 04 einen generischen Angriff, der eine Schwäche des Merkle-Damgård Designprinzips aufdeckt. Etwas vereinfacht gesagt zeigte Joux, dass, wenn man Kollisionen für die Hashfunktion (bzw. die Kompressionsfunktion) finden kann, es dann sogar leicht ist, K-fache Kollisionen für H zu berechnen. Dies sind verschiedene Nachrichten M[1], ..., M[K] mit dem gleichen Hashwert H(M[1])=...=H(M[K]). Der Aufwand dafür ist logarithmisch – Joux braucht $\log_2(K)$ Kollisionen der Kompressionsfunktion für eine K-fache Kollision. Z. B. kann er mit nur 20 Kollisionen der Kompressionsfunktion mehr als eine Million kollidierender Nachrichten erzeugen.

Das Merkle-Damgård Design für Hashfunktionen hat sich damit als fragil erwiesen: Wenn eine bestimmte Schwäche existiert, dann existiert sogar eine noch viel größere Schwäche. Anders ausgedrückt: Wenn der Damm einen Riss hat, dann dauert es nicht mehr lange, bis er ganz bricht.

Es gibt jedoch Erweiterungen des Merkle-Damgård Designs, die beweisbar sicher gegen den Joux Angriff sind. In [Lu04] wird sogar formal bewiesen, dass einige dieser Erweiterungen sogar sicher gegen alle generischen Angriffe zum Finden von Multi-Kollisionen sind.

5 „Doppelt genäht“

Eine wenig elegante, aber kryptographisch robuste Methode, welche ohne großen Aufwand anwendbar ist, ist das zusätzliche Verwenden mehrerer Hashfunktionen. Beispielsweise kann man einen 512 bit Hash H durch das Aneinanderhängen des 128-bittigen MD5-Hashes, des 160-bitten SHA-1 Hashes und des 256-bit langen SHA-224 Hashes bilden:

$$H(x)=(MD5(x),SHA\text{-}1(x),SHA\text{-}224(x)).$$

Eine Kollision für H ist eine Kollision für jede der drei Hashfunktionen, aus denen H zusammengesetzt wurde. Eine derartige „Kaskade“ von Hashfunktionen hat deshalb die Eigenschaft, dass der zusammengesetzte Hash mindestens so stark ist, wie der stärkste einzelne Hash. In diesem Sinne ist die Methode kryptographisch robust.

Leider bietet diese Konstruktion darüber hinaus aber nur einen überraschend geringen Sicherheitsgewinn. Auf der Crypto 2004 präsentierte Joux [Jo04] eine Anwendung seinem Multi-Kollisionsangriff auch auf Kaskaden von Hashfunktionen. Wenn die verwendeten Funktionen Merkle-Damgråd Hashfunktionen sind, ist der Zugewinn an Sicherheit weit geringer als bisher erwartet. Praktisch alle zur Zeit in der Praxis verwendet Hashfunktionen sind Merkle-Damgråd Hashfunktionen.

Eine Alternative hierzu ist die Kaskade der internen Kompressionsfunktionen [Lu04]. Diese Variante ist aufwändiger zu implementieren und weniger effizient als die Kaskade der Hashfunktionen, bietet aber beweisbare Sicherheit gegen alle generischen Angriffe, eingeschlossen den Multi-Kollisionsangriff von Joux.

Die Kaskade von mehreren Hashfunktionen kann, trotz der diskutierten Einschränkungen, als eine einfach zu implementierende Übergangslösung angesehen werden, solange neue und (hoffentlich!)

bessere Hashfunktionen noch nicht zur Verfügung stehen.

Auf keinen Fall sollte die jedoch die Kaskade an sich unsicherer Hashfunktionen als Alternative zum Einsatz neuer stärkerer Hashfunktionen missverstanden werden!

6 Gibt es Alternativen?

Keine praktischen Angriffe wurden bisher für die gestärkte RIPEMD Variante RIPEMD-160 [DBP96] gefunden. Dennoch lassen die Angriffe auf die Vorversion RIPEMD und die Mitgliedschaft in der MD4 Familie zur Vorsicht raten. Eine 160-bit Hashfunktion sollte ohnehin nur als temporäre Lösung für noch wenige weitere Jahre angesehen werden.

Der neue NIST-Standard SHA-2 [NI02] definiert eine ganze Reihe von Funktionen mit längerer Hashwertlänge. Sie liefern zwischen 224 bit (SHA-224) und 512 bit (SHA-512) lange Hashwerte. Einige moderne Sicherheitslösungen wie GPG, bieten heute bereits SHA-256 als Option [GP05] oder wenden wie Cryptophone [Cr03] SHA-256 mehrfach paramaterisiert hintereinander an.

Allerdings existieren für diese Funktionenfamilie erst sehr wenige Analysen [GH03, We05].

Leider gab es auch, im Gegensatz zum sehr erfolgreichen Verfahren um die Blockchiffre AES, im Falle der Hashfunktionen keinen derartigen Wettbewerb – vielmehr hat das NIST, wie bei SHA-0 und SHA-1, auf ein von der NSA stammendes Design zurückgegriffen, dessen Designprinzipien und -kriterien nicht veröffentlicht wurden.

Henri Gilbert und Helena Handschuh zeigten, dass die Funktionen gegen eine Reihe von bei SHA-1 und MD4 möglichen Angriffstechniken sehr widerstandsfähig sind. Beunruhigend ist allerdings, dass modifizierte Versionen gravierend schwach sind – auch bei nur geringen Vereinfachungen: "*However we show that slightly simplified versions of the hash functions are surprisingly weak: whenever symmetric constants and initialization values are used throughout the computations, and modular additions are replaced by exclusive or operations, symmetric messages hash to symmetric digests.*" [GH03]

Für Kryptographen sind die Hashalgorithmen der SHA-2 Familie damit unangenehm fragil – sehr kleine Änderungen der Hashfunktion sollten eigentlich nur moderate Auswirkungen auf die Sicherheit der Hashfunktion haben. Dennoch ist die Verwendung der Funktionen des SHA-2 Standards eine akzeptable Alternative.

Als Alternative bieten sich beispielsweise algebraische Kombinationen von Funktionen mit unterschiedlichem Basisdesigns untersucht werden (s.a. [We00]). Interessante Ansätze stellen unserer Meinung hierfür beispielsweise Tiger [AB96] und Whirlpool [BR00] aus dem europäischen NESSIE Projekt dar.

Wir befürworten mit Nachdruck die Initiierung eines Wettbewerbes zur Findung einer Standard-Hashfunktion.

Literatur

[AB96] Anderson, R., Biham, E., Tiger: A Fast New Hash Function, Fast Software Encryption – FSE'96, LNCS 1039, Springer-Verlag (1996), pp. 89-97.

[BR00] Barreto, P., Rijmen, V., The Whirlpool Hashing Function, First open NESSIE Workshop, Leuven, Belgium, 13-14 November 2000.

[Cr05] Cryptophone FAQ http://www.gsmk.de/html/faq_en.html

[Da89] Damgård, I. A design principle for hash functions. Crypto 89, LNCS 435, pp. 416-427.

[DBP96] Dobbertin, H., Bosselaers, A., Preneel, B., RIPEMD-160, a strengthened version of RIPEMD, Fast Software Encryption 1996, LNCS 1039, D. Gollmann, Ed., Springer-Verlag, 1996, pp. 71-82.

[Do96a] Dobbertin, H., Cryptoanalysis of MD4, Fast Software Encryption, 1996.

[Do96b] Dobbertin, H., The Status of MD5 After a Recent Attack, CryptoBytes 2 (2), 1996.

[Do97] Dobbertin, H.: Digitale Fingerabdrücke. Sichere Hashfunktionen für digitale Signatursysteme. Datenschutz und Datensicherheit (DuD), 2/1997, S. 82-87.

[Do98] Dobbertin, H., Cryptanalysis of MD4, Journal of Cryptology 11:4 (1998), pp. 253-271.

[GH03] Henri Gilbert, H., Handschuh, H., Security analysis of SHA-256 and sisters, in M. Matsui and R. Zuccherato, Eds., Selected Areas in Cryptography (SAC 2003), vol. 3006 of Lecture Notes in Computer Science, pp. 175-193, Springer-Verlag, 2004.

[GP05] GNU Privacy Guard, www.gnupg.org

[Jo04] Joux, A., Multicollisions in iterated hash functions, application to cascaded constructions. Crypto 04, LNCS 3152, pp. 306-316.

[Lu04] Lucks, S., Design Principles for Iterated Hash Functions, Cryptology ePrint Archive: Report 2004/253.

[LWW05] Arjen Lenstra and Xiaoyun Wang and Benne de Weger. Colliding X.509 Certificates. http://eprint.iacr.org/2005/067

[Me89] Merkle, R., One-way hash functions and DES. Crypto 89, LNCS 435, pp. 428-446.

[NE03] NESSIE, Portfolio of recoendatet cryptographic primitives, https://www.cosic.esat.kuleuven.ac.be/nessie/deliverables/decision-final.pdf

[NI92] NIST, "Proposed FIPS Secure Hash Standard", Washington D.C., 11. Sept 1992.

[NI94] NIST, "Proposed Revision FIPS Secure Hash Standard", Washington D.C., 11. Juni 1994.

[NI02] NIST, "FIPS 180-2: Secure Hash Standard (SHS)", August 2002 (change notice: February 2004).

[Ol05] Olsen, F., NIST moves to stronger hashing, Federal Computer Week, Feb. 7, 2005.

[Ri90] Rivest, Ron, The MD4 Message Digest Algorithm, Advances in Cryptology – CRYPTO '90 Proceedings, Springer-Verlag, 1991, pp. 303-311.

[Ri91] Rivest, Ron, xxx, 1991

[Sc05] Schneier, B., SHA-1 Broken. http://www.schneier.com/blog/archives/2005/02/sha1_broken.html

[Wea05] Wang, X., Lai, X., Feng, D., Chen, H., Yu, X., Cryptoanalysis for Hash Functions MD4 and RIPEMD, preprint, 2005.

[We00] Weis, R., Protocols and Algorithms, PhD Thesis, 2000.

[We05] Weis, R., Hash Problems, CCC Datenschleuder, 2005.

[ZPS92] Y. Zheng, J. Pieprzyk, and J. Seberry, HAVAL – a one-way hashing algorithm with variable length of output, Advances in Cryptology – Auscrypt'92, LNCS 718, Springer-Verlag (1993), pp. 83-104.