The Classification of Hash Functions

Ross Anderson, C.Math, FIMA

University of Cambridge Computer Laboratory,
Pembroke Street, Cambridge CB2 3QG, UK
Email: rja14@cl.cam.ac.uk

Abstract

When we ask what makes a hash function 'good', we usually get an answer which includes collision freedom as the main (if not sole) desideratum. However, we show here that given any collision-free function, we can derive others which are also collision-free, but cryptographically useless. This explains why researchers have not managed to find many interesting consequences of this property. We also prove Okamoto's conjecture that correlation freedom is strictly stronger than collision freedom.

We go on to show that there are actually rather many properties which hash functions may need. Hash functions for use with RSA must be multiplication free, in the sense that one cannot find X, Y and Z such that $h(X)h(Y) = h(Z)$; and more complex requirements hold for other signature schemes. Universal principles can be proposed from which all the freedom properties follow, but like most theoretical principles, they do not seem to give much value to a designer; at the practical level, the main import of our work is that one should be explicit about the properties which we require of a cryptographic algorithm. It also has some consequences for algorithm design.

1 Introduction

There are many applications where a one-way hash function is required. Digital signatures are one example: it is usually not practical to sign a whole message, as public key algorithms are rather slow, so the normal practice is to hash a message to a digest of between 128 and 160 bits, and sign this instead.

According to the seminal paper by Diffie and Hellman: 'f is a *one-way function* if, for any argument x in the domain of f, it is easy to compute the corresponding value $f(x)$, yet, for almost all y in the range of f, it is computationally infeasible to solve the equation $y = f(x)$ for any suitable argument x' [DH]. Lampson's more succinct expression is: 'you can't invert the function and compute a message with a given digest' [LABW].

Hash functions are used for much more than just generating message digests. For example, if we wish to make a signature algorithm which is also a homomorphism (such as RSA) proof against adaptive chosen-ciphertext attack, then we can break up the homomorphic property by passing the data through a hash function first. Hash functions are also used as authentication primitives in their own right: the new IBM distributed security system, 'KryptoKnight', uses the hash function MD4 as its basic building block in order to get round export restrictions on cryptographic algorithms [MTVZ]. Finally, Merkle showed that signature schemes can be constructed using hash functions alone [M2], although these are one-time schemes and not really practical.

A number of researchers therefore tried to find general security conditions which would formalise our intuitive idea that a hash function should be hard to invert. A number of abstract definitions were proposed, based on concepts such as Turing machine complexity [B] and circuit complexity [BL], but these were too theoretical to hold out much hope of practical consequences.

One problem with attempts to formalise the concept of one-wayness is the implicit temporal ordering: you can get the pair $(x, h(x))$ if you start from x but not if you start from $h(x)$. Equivalently, conditions can be placed on the context of an attack, such as the knowledge an attacker may start off with, or on the way her knowledge evolves during the attack. Such contextual properties can be rather hard to deal with, as is well known to workers in the field of cryptographic protocols [BAN]. There was thus some excitement at the appearance of a fresh approach, namely 'collision freedom'.

A function h is called *collision free* if it is unfeasible to find two input strings S and T which both hash to the same value, ie $h(S) = h(T)$. This concept was introduced in [D1], which showed that hardness assumptions about number-theoretic problems such as discrete log and factoring (or more generally the existence of claw-free permutations) imply that one can construct hash functions which are indeed collision-free. This result has been very influential, and collision freedom has seemed, to many researchers, to be a natural way to capture the behaviour which we expect from a 'good' hash function.

Many applications do indeed need a hash function to be collision free - otherwise an attacker might get a victim to sign one message hash and then substitute another [Y]. However, despite the passage of several years, only a few results have been shown to follow from collision freedom: essentially all we know is that the property is preserved under chaining [D2]; a number of equivalent definitions exist [R] [SY]; and collision free functions are enough to commit bits and construct one-time signatures. However, researchers wanting to prove security results for practical systems have had to resort to other definitions such as Okamoto's 'correlation free one-way hash functions'.

A function h is *correlation free* if it is not feasible to find X and Y such that the Hamming weight of $h(X)$ xor $h(Y)$ is less than one would expect to get from random chance if we calculated $h(M)$ for a lot of M; intuitively, it means that as well as having no collisions, we get no near misses either. Okamoto used this property to prove the security of a discrete log based identification and signature scheme, and he conjectures in his paper [O2] that correlation freedom is a stronger property than collision freedom, and we will now prove that this conjecture is in fact true. In fact, we shall show that given any collision free function h, we can construct a derived function $\overline{h}$ which is also collision free, but not correlation free.

2 A Counterexample

Let k be a fixed small integer, and let h be a collision-free hash function. Given any string S, we will write S_1 for the first k bits of S, and S_2 for the remainder. We will also write $\|$ for string concatenation. Thus

$$S = S_1 \parallel S_2 \tag{1}$$

Now define the function $\overline{h}$ as

$$\overline{h}(S) = S_1 \parallel h(S_2) \tag{2}$$

In other words, $\overline{h}$ leaves the first k bits of the input string unchanged, and hashes the rest of it using h. It is clear that $\overline{h}$ is collision-free if h is. However, h is not correlation free, as we can trivially construct X and Y such that $h(X)$ differs from $h(Y)$ in any number of bits up to k.

3 The Real Requirements for Hash Functions

Although $\overline{h}$ is collision free, it is quite unsuitable as a general purpose hash function. Some variants of the Fiat-Shamir signature scheme can be used only once unless we can ignore the probability of two hashed messages differing in only one bit; more seriously, schemes such as 'KryptoKnight', use hash functions with a secret prefix to provide one-way encryption, and a hash function such as $\overline{h}$ would leak the secret key. This brings out the implicit assumption that hash functions have information hiding properties, and shows that these are quite different from collision freedom.

Furthermore, it will often be necessary that our hash function should not interact with a given signature or authentication scheme in some possibly dangerous fashion. As such schemes have been based on a wide variety of cryptographic primitives, including factorisation [RSA], discrete log [DSA] [E], modular squaring [J], the Data Encryption Standard [MM] [M1], and knapsacks

[C] [GC], there are many ways in which hashing and other algorithms might possibly interact.

3.1 Complementation Freedom:

This can be important for schemes which are built on DES.

Definition 1 *A function h is complementation free if it is not feasible to find inputs X and Y such that $h(X) = \sim h(Y)$, where $\sim X$ is the binary complement of X.*

Collision freedom does not imply this either; following the above model, we can take any collision free hash function h, set k=1 and let $\tilde{h}(S) = h(S_2)$ where $S_1 = 1$, and $\tilde{h}(S) = \sim h(S_2)$ where $S_1 = 0$.

3.2 Addition Freedom:

Definition 2 *A function h is addition free if it is not feasible to find inputs X, Y and Z such that $h(X) + h(Y) = h(Z)$.*

This property also does not follow from collision-freedom. Again, let h be a collision free hash function, and assume that the functions whose outputs are the odd and the even bits of its output are are also collision free; let h_o and h_e be these functions. In other words

$$h_e(S) = h(S) \textit{ and } 10101010...$$

$$h_o(S) = h(S) \textit{ and } 01010101...$$

Now we can define a function h^+ which is collision free but not addition free by

$$h^+(S) = \begin{cases} S_1 \parallel h(S_2) & \text{where } S_1 \equiv 0 \pmod 3 \\ S_1 \parallel h_o(S_2) & \text{where } S_1 \equiv 1 \pmod 3 \\ S_1 \parallel h_e(S_2) & \text{otherwise} \end{cases}$$

3.3 Multiplication Freedom:

Definition 3 *A function h is multiplication free mod N if it is not feasible to find inputs X, Y and Z such that $h(X)h(Y) = h(Z) \pmod N$. It is multiplication free if it is multiplication free mod all N greater than some value.*

This is the property needed when the signature scheme is a multiplicative homomorphism such as RSA [G]. We can construct counterexamples be taking any function h^+ which is not addition-free mod N and any generator g of the multiplicative group mod N, and forming $h^*(M) = g^{h^+(M)}$.

3.4 More Complex Freedom Properties:

The above properties may be the obvious ones, but they do not begin to exhaust the range of freedom properties which may be required.

A recent example underlines this neatly. If p and q are primes with $q \mid (p-1)$, g is of order q, and the user has secret key x and public key $y = g^x$, and the message key is k, then the Yen-Laih signature on message m is r, s such that

$$r = g^k \pmod p, \qquad s = x + krm \pmod q$$

These signatures are verified by $g^s = yr^{rm \pmod q} \pmod p$. However, Nyberg recently pointed out [N] that this will also be satisfied by m', r' and s' such that:

$$m' = \frac{rm}{rg^t \pmod p} \pmod q$$

$$r' = rg^t \pmod p, \qquad s' = s + trm \pmod q$$

for all t between 0 and q. Thus we need a hash function with a strange freedom property, namely that you can never find two messages m and m' such that

$$h(m') = \frac{rh(m)}{rg^t \pmod p} \pmod q \qquad (3)$$

In other words, for a hash function to be used with the Yen-Laih scheme, we must be confident that given A and B, we cannot find m and m' such that $h(m) = Ah(m')/B \pmod q$. The temporal ordering here is of course what we had been trying to escape by using freedom properties, and it raises doubt about whether any set of freedom properties can ever be enough.

Even if they could be, they are likely to be complicated. Consider the case of the US Digital Signature Standard [DSA]; here we need that for all functions

f_1 and f_2 which are practical to compute, we will never find m, m' such that where $s = k^{-1}(m + xr) \pmod q$, we have

$$r = g^{m/s} y^{r/s} \pmod q \tag{4}$$

and

$$f_1(r) = g^{m'/f_2(s)} y^{f_1(r)/f_2(s)} \pmod q \tag{5}$$

Now it is well known that if an opponent can ever get his hands on the message key used to create a DSA signature, then he can recover the signer's secret key. For this reason, implementers often use a hash function to generate message keys for DSA and other ElGamal type schemes; but then there is another freedom property which must be satisfied in order to ensure that a dependency between two message keys does not reveal the user's secret signing key.

In this regard, the freedom property required by DSA appears too complicated to be of practical interest, but other schemes are not so robust: in Schnorr's scheme [S] we must not have $h(g^r, m)$ equal to $f(k) + h(g^{r+k}, m)$ for any computable function f.

Thus, even if we do not expect to find a general freedom property which will cover every case, writing down specific freedom requirements explicitly would seem to be good practice; it would at least have saved Yen and Laih from embarrassment.

3.5 Other Security Properties

As we pointed out in section 2 above, hash functions are often used to provide one-way encryption, and a typical implementation prefixes the secret key to the string being encrypted. If $\overline{h}$ were used in such a system, it would leak the first k bits of the secret key directly. This shows that hash functions must often posses local as well as global one-wayness properties.

We should also point out that the *universal one-way hash functions* of [NY] are a subset of the collision free functions, and thus $\overline{h}$ is universal; and that the *uniformity* of [CW], which is implied by correlation freedom, is rather weak, as $\overline{h}$ is uniform too. Thus neither of these definitions implies local one-wayness.

4 Algorithm Design

Freedom properties have a rôle to play in algorithm design as well. Indeed, most of the attacks on both block and stream ciphers rely to some extent on the designer's failure to ensure that there should be no usable affine approximation to the nonlinear part of their function. Thus correlation freedom is closely related both to correlation attacks on stream ciphers [MS] and to linear [M3] and differential [BS] attacks on block ciphers; these attacks exploit correlations between the input and output of a block cipher round function, between the key and keystream of a stream cipher, or (in any kind of cipher) between the ciphertexts obtained if inputs (plaintexts or keys) are changed in some known way.

Consider for example the Luby-Rackoff construction [LR] which can be used to obtain a block cipher from a hash function. This is essentially the first three rounds of a Feistel cipher; where our hash function is h, the left and right halves of the input block are m_1 and m_2, and the left and right halves of the output block are c_1 and c_2, we have

$$i = h(k_1 \parallel m_1) \oplus m_2, \qquad c_1 = h(k_2 \parallel i) \oplus m_1, \qquad c_2 = h(k_3 \parallel c_1) \oplus i$$

The interesting results concerning constructions of this kind have so far been proved using the (very strong) assumption of pseudorandomness. It should be clear why collision freedom will not be sufficient; the function h must be correlation free in order to prevent linear attacks, and its binary 'derivative' must be correlation free in order to prevent differential attacks. We leave it as an open problem whether there is a set of freedom properties which enables one to prove corresponding results about the derived block cipher.

In the case of stream ciphers, there appear to be a number of related robustness properties. Divide and conquer correlation attacks happen because of a correlation between some function of the keystream and some other function of part of the key. Much the same occurs in the case of fast correlation attacks, at least where the nonzero coefficients of the polynomial are bunched together [A]; and most of the practical fast correlation attacks appear to be reducible to this case.

5 Is There an Interesting Universal Property?

One might wish to find a security property which is universal, or at least implies all the freedom properties we are likely to need. However, such a property is unlikely to be constructive, as many signature schemes have nonconstructive freedom requirements similar to those exhibited for DSA above. It will also imply at least some one-wayness properties, as we saw from considering Nyberg's attack on the Yen-Laih scheme.

Furthermore, if we want to base our universal property on global one-wayness, then we will have to add quite a lot to the intuitive definitions of Lampson and Diffie. Consider, for example, a hash function which ignores some of its input; if h is non-invertible, and $h_I(S) = h(S_2)$, then h_I is also non-invertible, but finding collisions for it is trivial.

Some authors escape this problem by including collision freedom explicitly in their definition of one-wayness [P], but this is not enough, as $\overline{h}$ shows. Another approach is Damgård's 'maximal security' property [D1]: he calls a signature scheme maximally secure if it withstands an adaptive chosen message attack, and showed that we can use any collision free function to compress messages before signing without loss of security.

We could define a general property which covers both freedom and one-wayness by combining the ideas of maximal security and local one-wayness. If $h(S) = c$, and an attacker does not know S, but knows c and some other function of S which distinguishes it from the other preimages of c, then no amount of side knowledge about c will enable her to find S. That is, even if she has an oracle which, for any functions F and G, will tell her every set of S_i (not containing S) such that $F(h(S_1), h(S_2), ...)$ is nontrivially equal to $G(c)$, then she still cannot find S. The case where F and G are the identity gives us collision freedom; multiplication and other freedoms are similarly straightforward; and choosing G to be a projection gives us local one-wayness.

However, this definition is only of theoretical interest: in effect, it agglutinates all possible freedom and one-wayness properties as axioms. In the practical world, signature schemes with maximal security are constructed by concentrating on specific properties; the usual method is to use a hash function to remove homomorphic features from an existing signature scheme [ZS].

6 Conclusion

It appears that the way forward lies in explicitness rather than generality. Individual freedom properties are fairly straightforward to define in practical applications, and using them in security specifications has the further advantage that metaprotocol issues such as side knowledge are easier to control. This was the initial attraction of collision freedom; our work shows that the lack of results to date is not so much because collision freedom is a wrong track, but rather because it is only one among a very large number of freedom properties. These freedom properties are central to controlling interactions between cryptographic algorithms, and have the potential to be useful in algorithm design as well.

References

[A] RJ Anderson, "Faster attack on certain stream ciphers", in *Electronics Letters* **29** no 15 (22nd July 1993) pp 1322 - 1323

[B] G Brassard, "Relativised Cryptography", in *IEEE Transactions on Information Theory*, **IT-29** no 6 (November 1984) pp 877 - 894

[BAN] M Burrows, M Abadi RM Needham, *'A logic of Authentication'*, Technical Report no 39, Digital Systems Research Center, Palo Alto, Ca.

[BL] RB Boppana and JC Lagarias, "One-way functions and circuit complexity", in *Information and Computation*, **74** (1987) pp 226 - 240

[BS] E Biham, A Shamir, *'Differential cryptanalysis of the Data Encryption Standard'*, Springer 1993

[C] BZ Chor, *'Two issues in public-key cryptography: RSA bit security and a new knapsack type system'*, MIT 1985

[CW] JL Carter and MN Wegman, "Universal Classes of Hash Functions", in *Journal of Computer and Systems Sciences* **18** (1979) pp 143 - 154

[D1] IB Damgård, "Collision free hash functions and public key signature schemes", in *Advances in Cryptology - EUROCRYPT 87*, Springer LNCS **304**, pp 203 - 216

[D2] IB Damgård, "A Design principle for Hash Functions", in *Advances in Cryptology - CRYPTO 89*, Springer LNCS **435**, pp 416 - 427

[DH] W Diffie and ME Hellman, "New Directions in Cryptography", in *IEEE Transactions on Information Theory*, **IT-22** no 6 (November 1976) p 650

[DSA] National Institute of Standards and Technology, "A Proposed Digital Signature Standard", posted on usenet news group sci.crypt on 24/9/91

[E] T El-Gamal, "A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms", in *IEEE Transactions on Information Theory* **IT-31** no 4 (July 1985) pp 469 - 472

[FS] A. Fiat and A. Shamir, "How To Prove Yourself: Practical Solutions to Identification and Signature Problems", in em Advances in Cryptology - Crypto 86, Springer LNCS **263**, pp 186 - 194

[G] M Girault, "Hash-functions using modulo-N operations", in *Advances in Cryptology - EUROCRYPT 87*, Springer LNCS **304**, pp 217 - 226

[GC] P Godlewski and P Camion, "Manipulations and errors, detection and localization", in *Advances in Cryptology - EUROCRYPT 88*, Springer LNCS **330**, pp 97 - 106

[J] RR Juenemann, "A high speed manipulation detection code", in *Advances in Cryptology - CRYPTO 86*, Springer LNCS **263**, pp 327 - 346

[LABW] B Lampson, M Abadi, M Burrows, E Wobber, "Authentication in Distributed Systems: Theory and Practice," in *ACM Transactions on Computer Systems*, **10** no 4 (Nov 1992) pp 265 - 310

[LR] M Luby, C Rackoff, "How to extract pseudorandom permutations from pseudorandom functions", *SIAM Journal of Computing* **17** *no 2 (April 1988)* pp 373 - 386

[LY] AK Lenstra, Y Yacobi, "User Impersonation in Key Certification Schemes", in *Journal of Cryptology* **6** *no 4 (Autumn 1993) pp 225 - 232*

[M1] RC Merkle, "One Way Hash Functions and DES", in *Advances in Cryptology - CRYPTO 89*, Springer LNCS **435**, pp 428 - 446

[M2] RC Merkle, "A Digital Signature Based on a Conventional Encryption Function", in *Advances in Cryptology - CRYPTO 87*, Springer LNCS **293**, pp 369 - 378

[M3] M Matsui, "Linear Cryptanalysis Method for DES Cipher", in *Pre-Proceedings of Eurocrypt 93* pp W112 - W123

[MM] CH Meyer and SM Matyas, *'Cryptography - A New Dimension in Computer Data Security'*, Wiley 1982

[MS] W Meier, O Staffelbach, "Fast Correlation Attacks on certain Stream Ciphers", in *Journal of Cryptology* **1** *(1989)* pp 159 - 176

[MTVZ] R Molva, G Tsudik, E Van Herreweghen, S Zatti, "Kryptoknight Authentication and Key Distribution System", in *Computer Security - ESORICS 92*, Springer LNCS **648**, pp 155 - 174

[N] K Nyberg, "The Digital Signature Scheme of Yen and Laih is Not Secure Without Hashing" *Electronics Letters (to appear)*

[NY] M Naor and M Yung, "Universal one-way hash functions and their cryptographic applications", in *Proc 21st STOC*, ACM (1989), pp 33 - 43

[O1] T Okamoto, "A Fast Signature Scheme Based on Congruential Polynomial Operations", in *IEEE Transactions on Information Theory* **IT-36** (Jan 1990) pp 47 - 53

[O2] T Okamoto, "Provably Secure and Practical Identification Schemes and Corresponding Signature Schemes", in *Abstracts of Crypto 92*, pp 1-15 to 1-25

[OFF] T Okamoto, A Fujioka and E Fujisaki, "An Efficient Digital Signature Scheme Based on an Elliptic Curve over the Ring Z_n", in *Abstracts of Crypto 92*, pp 1-26 to 1-31

[P] B Preneel, *'Analysis and Design of Cryptographic Hash Functions'*, PhD Thesis, Katholieke Universiteit Leuven, January 1993

[R] A Russell, "Necessary and Sufficient Conditions for Collision-free Hashing", in *Abstracts of Crypto 92*, pp 10-22 to 10-27

[RSA] RL Rivest, A Shamir and L Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", in *Communications of the ACM* **21** (1978) pp 120 - 126

[S] CP Schnorr, "Efficient identification and signatures for smart cards", in *Advances in Cryptology - CRYPTO 89*, Springer LNCS **435**, pp 239 - 251

[SY] A de Santis and M Yung, "On the design of provably-secure cryptographic hash functions", in *Advances in Cryptology - EUROCRYPT 90*, Springer LNCS **473**, pp 412 - 431

[W] RS Winternitz, "Producing one-way hash functions from DES", in *Proceedings of Crypto 83*, Plenum Press, p 203 - 207

[Y] G Yuval, "How to Swindle Rabin", in *Cryptologia* **3** *(1979)* p 107

[YL] SM Yen, CS Laih, "New Digital Signature Scheme Based on Discrete Logarithm", in *Electronics Letters* **29** *no 12 (1993)* pp 1120 - 1121

[ZS] YL Zheng, J Seberry, "Practical Approaches to Attaining Security against Adaptively Chosen Ciphertext Attacks", in *Abstracts of Crypto 92 pp 7-1 - 7-10*