

Recent Developments In Hash Functions

*Matt Robshaw
RSA Laboratories*

*RSA Data Security Conference
January 28-31, 1997*



Overview

- **What are cryptographic hash functions (message-digest algorithms)?**
 - properties and uses of hash functions
- **Hash function design and structure**
 - collisions, pseudo-collisions and compression functions
- **Some examples of hash functions**
 - the MD-family and related hash functions
 - survey of known results
- **Summary, recommendations and conclusions**



Hash Functions

- A hash function converts an input of arbitrary length into an output of a fixed, short length



- the input is sometimes termed a pre-image
- the output is the hash value or message digest
- A hash function does not require the use of any secret information

Some Hash Function Properties

- **Finding a first or a second pre-image is hard**
 - given hash output y find some m with $h(m)=y$
 - given an input/output pair m/y to some hash function find another input m' with $h(m')=y$
- **Finding a collision is hard**
 - find any two inputs m and m' with $h(m)=h(m')$
 - note that we know collisions **must** exist!
- **The output appears to be “random”**



Some Hash Function Properties

	pre-image hard	2nd pre-image hard	collision-free
one-way hash function	yes	yes	-
collision-resistant hash function	yes	yes	yes



Some Uses of Hash Functions

- Storing a table of computer passwords
- Use as a component in a pseudo-random number generator
- Reducing some digital document prior to using a digital signature
- As a mechanism for committing to some “string”
- A component in a wide-variety of theoretical and practical cryptographic mechanisms



Uses and Properties

- It is important to know what properties we are appealing to when we use a hash function
 - storing computer passwords (one-way)
 - string-commitment (one-way)
 - resistance against signature forgery after signing is complete (one-way)
 - resistance against signature forgery at time of signing (collision-resistance)
 - etc. ...



Digital Signatures and Properties

- Consider signing the hash of some document directly
 - Diana wants a generous divorce settlement A but Charles is only willing to lose the washing machine B
 - Diana asks her butler to make two independent lists of variants to A and variants to B
 - when the hash of entries in these lists match then $(\text{hash}(B'))^{\text{sign}}$ supplied by Charles will also be a valid signature for A' since $\text{hash}(A') = \text{hash}(B')$
 - the birthday paradox can be applied to this problem
 - this attack depends on the method of digital signature



Digital Signatures and Properties

- Consider the opportunities for attack after a document has been signed
 - Charles' offer of the washing machine B has been made and signed $(\text{hash}(B))^{\text{sign}}$
 - Diana's butler is now faced with the task of finding some lucrative A' so that $\text{hash}(A') = \text{hash}(B)$
 - this is not the same problem as finding a collision, this is finding a second pre-image
 - note that B and $\text{hash}(B)$ are already fixed
 - this is a much harder problem than finding a general collision



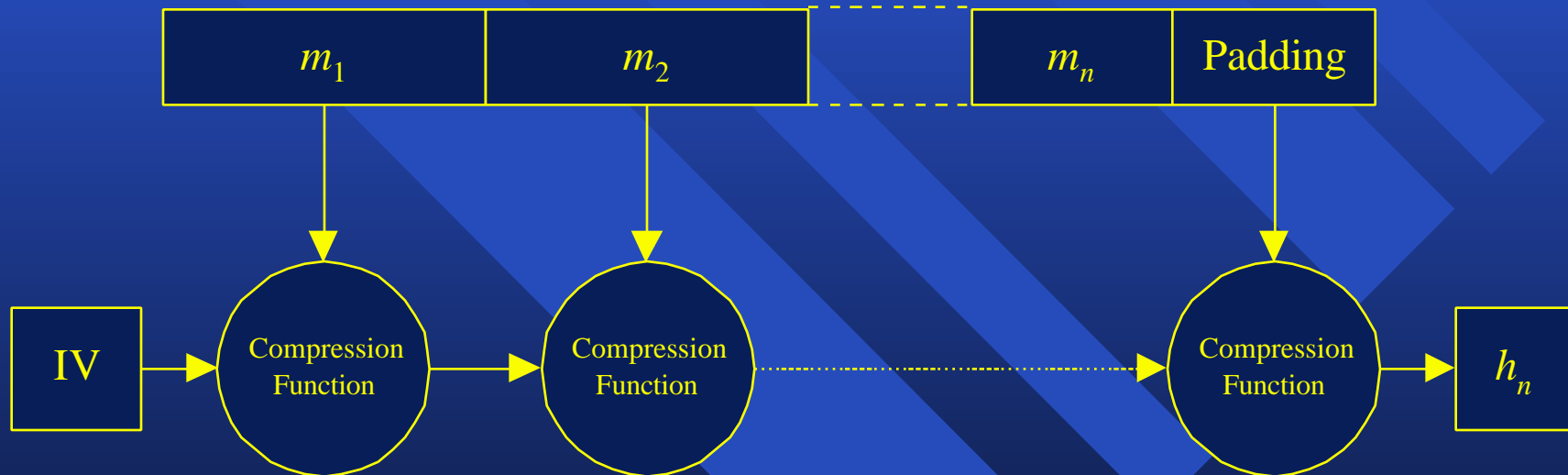
Hash Function Design

- **There are many designs for hash functions**
 - we will be concerned with those considered to be part of the MD-family
 - MD2, MD4, MD5, SHA-1, RIPEMD-160
 - these are dedicated hash functions with an iterative structure
 - they are built around the use of a chaining variable and a compression function
- **There are other interesting proposals such as HAVAL and Tiger**



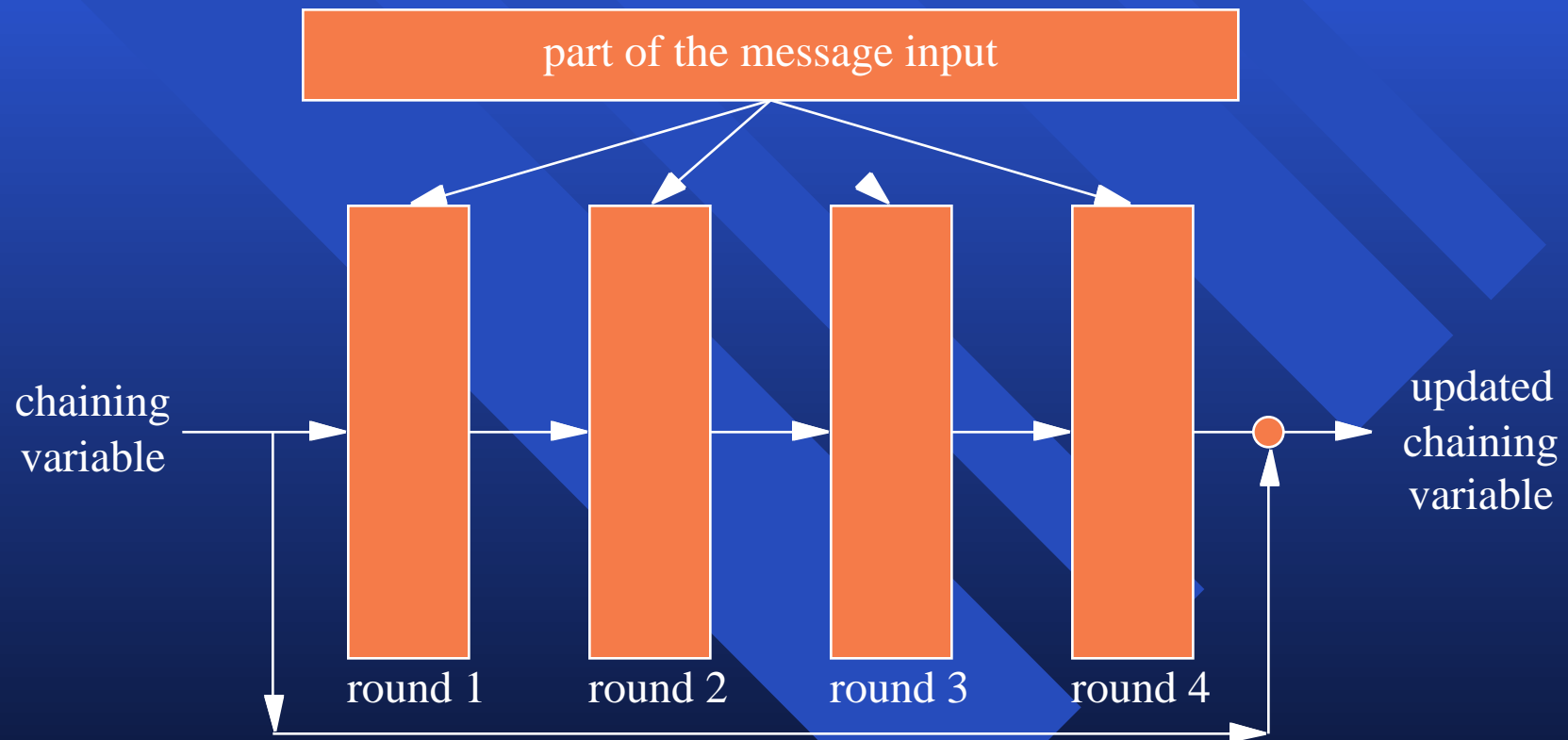
Hash Function Design

- The MD-family of hash functions follow the same iterative design

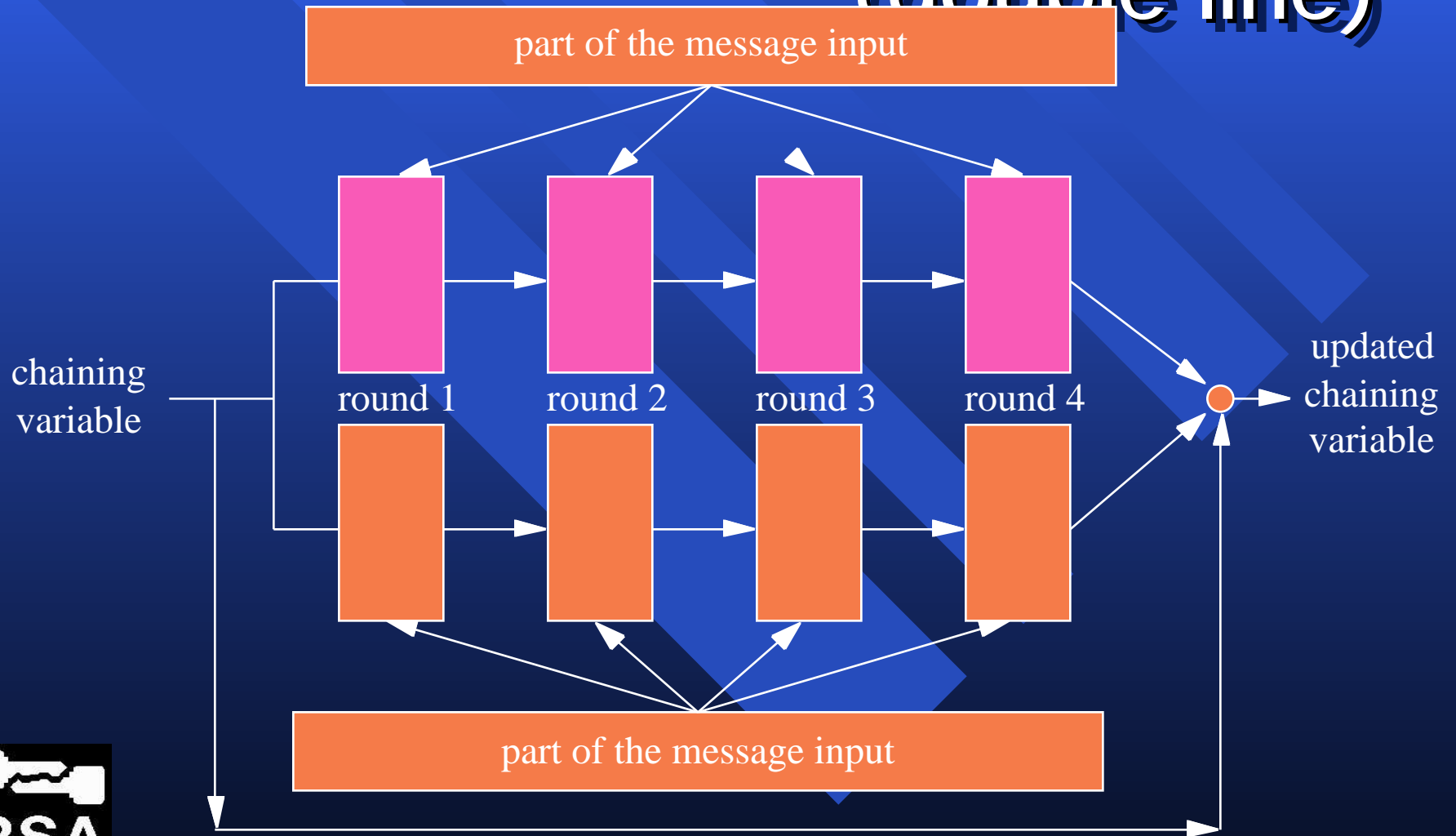


- the iterative use of a compression function has a good theoretical basis

Inside the Compression Function (single line)

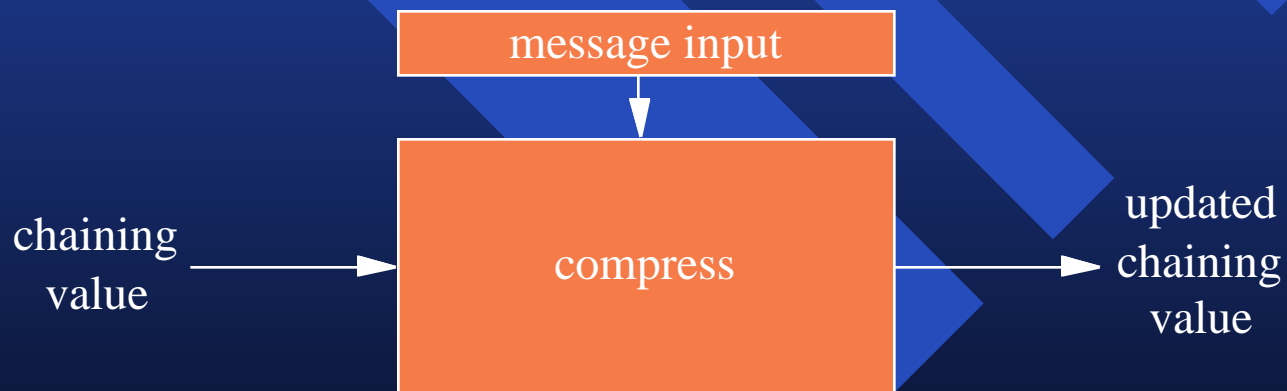


Inside the Compression Function (double line)



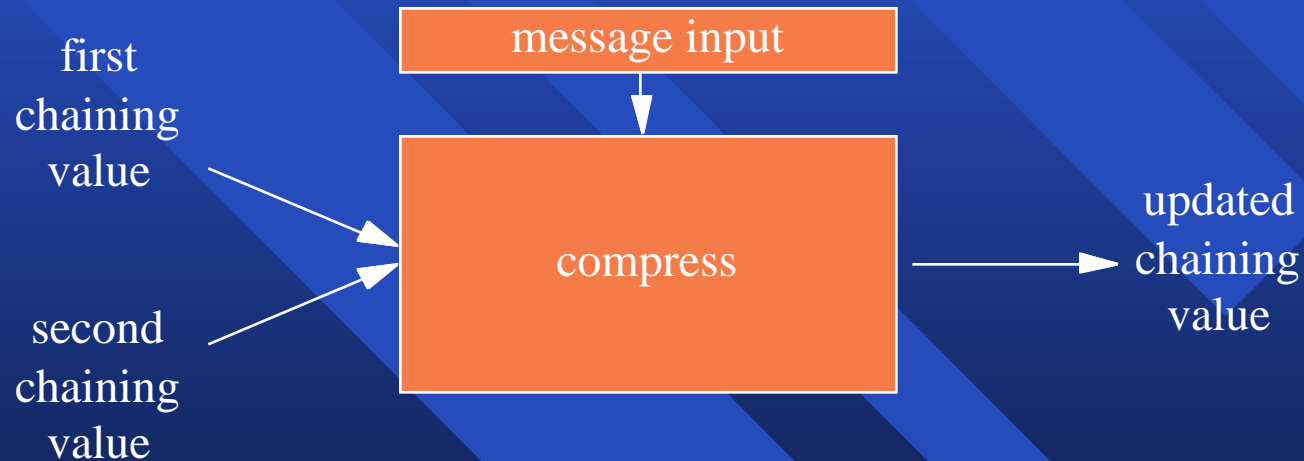
Collisions and Pseudo-Collisions

- Hash functions of the type we consider modify the value of a chaining value iteratively
 - the starting value of the chaining value is provided as part of the algorithm definition
- Consider the compression function as follows



Collisions and Pseudo-Collisions

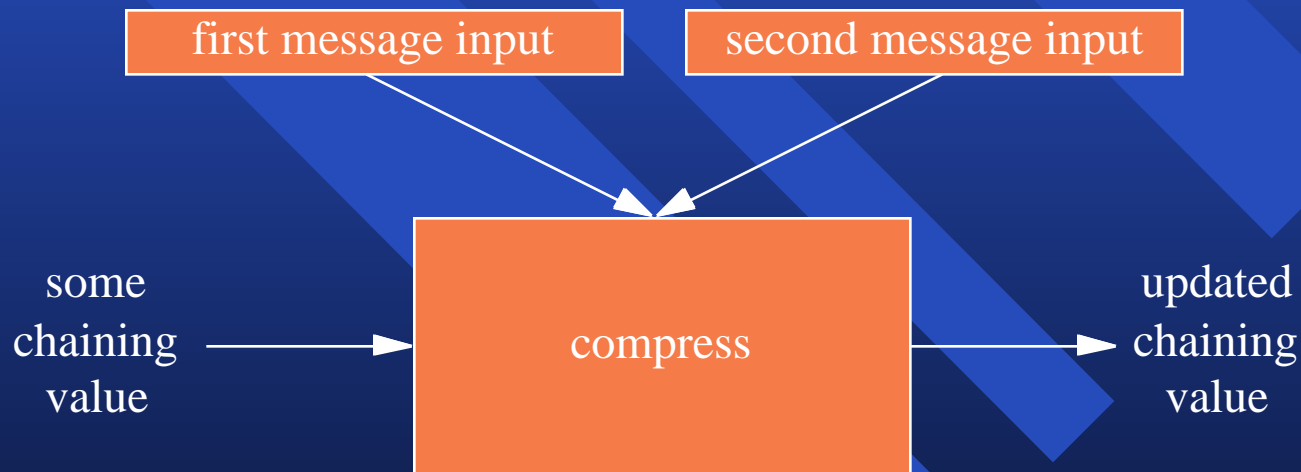
- A pseudo-collision to the compression function can be represented graphically as:



- a pseudo-collision can also be formed with two chaining values and two message inputs

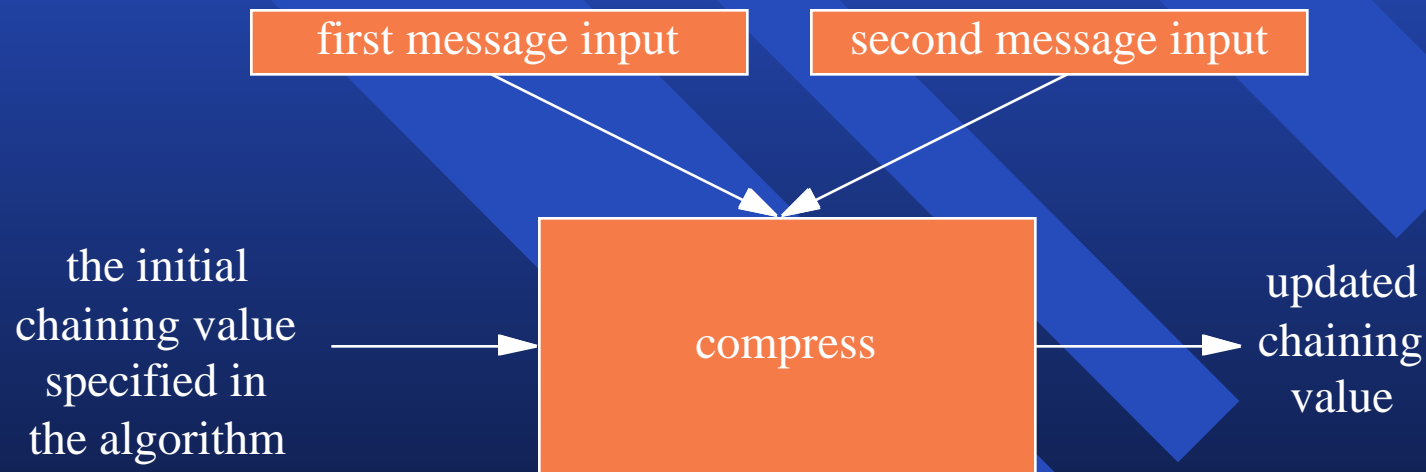
Collisions and Pseudo-Collisions

- A collision for the compression function can be represented graphically as:



Collisions and Pseudo-Collisions

- A collision for the entire hash function can be represented graphically as:



Some Example Hash Functions

- MD2
- MD4
- MD5
 - successor to MD4
- SHA-1
 - successor to SHA
- RIPEMD-128 and RIPEMD-160
 - successors to RIPEMD



Summary of Known Results

	pseudo-collisions	collisions for compress	collisions for reduced rounds	collisions for hash function
MD2	-	yes	-	-
MD4	-	yes	yes	yes
MD5	yes	yes	-	-
SHA-1	-	-	-	-
RIPEMD	-	-	yes	-
RIPEMD-128	-	-	-	-
RIPEMD-160	-	-	-	-



Status of MD2, MD4 and MD5

■ MD2

- considerable progress in finding collisions but full collisions have not yet been discovered

■ MD4

- extensive research shows that the behavior of MD4 is not sufficiently complex to resist analysis

■ MD5

- considerable progress in finding collisions but full collisions have not yet been discovered
- pseudo-collisions have been discovered



Status of MD2, MD4 and MD5

- **MD4 should not be used for any purpose**
 - collisions have been found and other work suggests that further advanced analysis of MD4 is possible
- **MD2 and MD5 should not be used for applications that require collision-resistance**
 - collisions have not yet been found, but this advance should be expected
 - the one-way properties of MD2 and MD5 have not been questioned in any of the existing literature



Some Alternatives To MD5

- SHA-1 and RIPEMD-160 can be recommended as good replacements for MD5 in applications that require collision-resistance

	Portable C Mbits/sec	x86 Assembly Mbits/sec
MD5	59.7	113.7
SHA-1	21.2	48.7
RIPEMD-128	35.6	64.0
RIPEMD-160	19.3	39.9



New Signature Techniques

- New signature techniques have been proposed by Bellare and Rogaway
 - signing requires the use of random numbers
 - even though $\text{hash}(A')$ and $\text{hash}(B')$ might be identical, the signature block contains $\text{hash}(A', r)$ or $\text{hash}(B', r)$ where r is randomly chosen by the signer
 - it is possible that collisions generated off-line will be of little use in forging such signatures
 - it might be possible to safely use hash functions that are not strictly speaking collision-resistant



Recommendations

- Check the application to be sure exactly which properties of the hash function are appealed to
 - if collision-resistance is not required then MD5 is as suitable for use as other recommended hash functions
 - for collision-resistance a prudent option would be to use either SHA-1 or RIPEMD-160
- If an existing application uses MD5 and collision-resistance is required
 - while the application is not immediately at risk it should be upgraded when convenient



Future Trends

- **Van Oorschot and Wiener have demonstrated that 128-bit message digests might be too short for future use**
 - they describe the design of a collision-search machine
 - for \$10 million a dedicated machine could find an MD5 collision in around 24 days
- **To allow for future advances 160-bit message digests will become increasingly appropriate**



Conclusions

- Hash function analysis has been revolutionized in the past two years
- One of the most popular hash functions (MD5) is no longer suitable when collision-resistance is required
 - SHA-1 and RIPEMD-160 are good alternatives
- We need to consider closely exactly what we require of a hash function in some application
 - few applications require that the hash function be collision-resistant
 - MD5 remains suitable for a variety of applications

