A Study of the MD5 Attacks: Insights and Improvements

John Black and Martin Cochran
University of Colorado

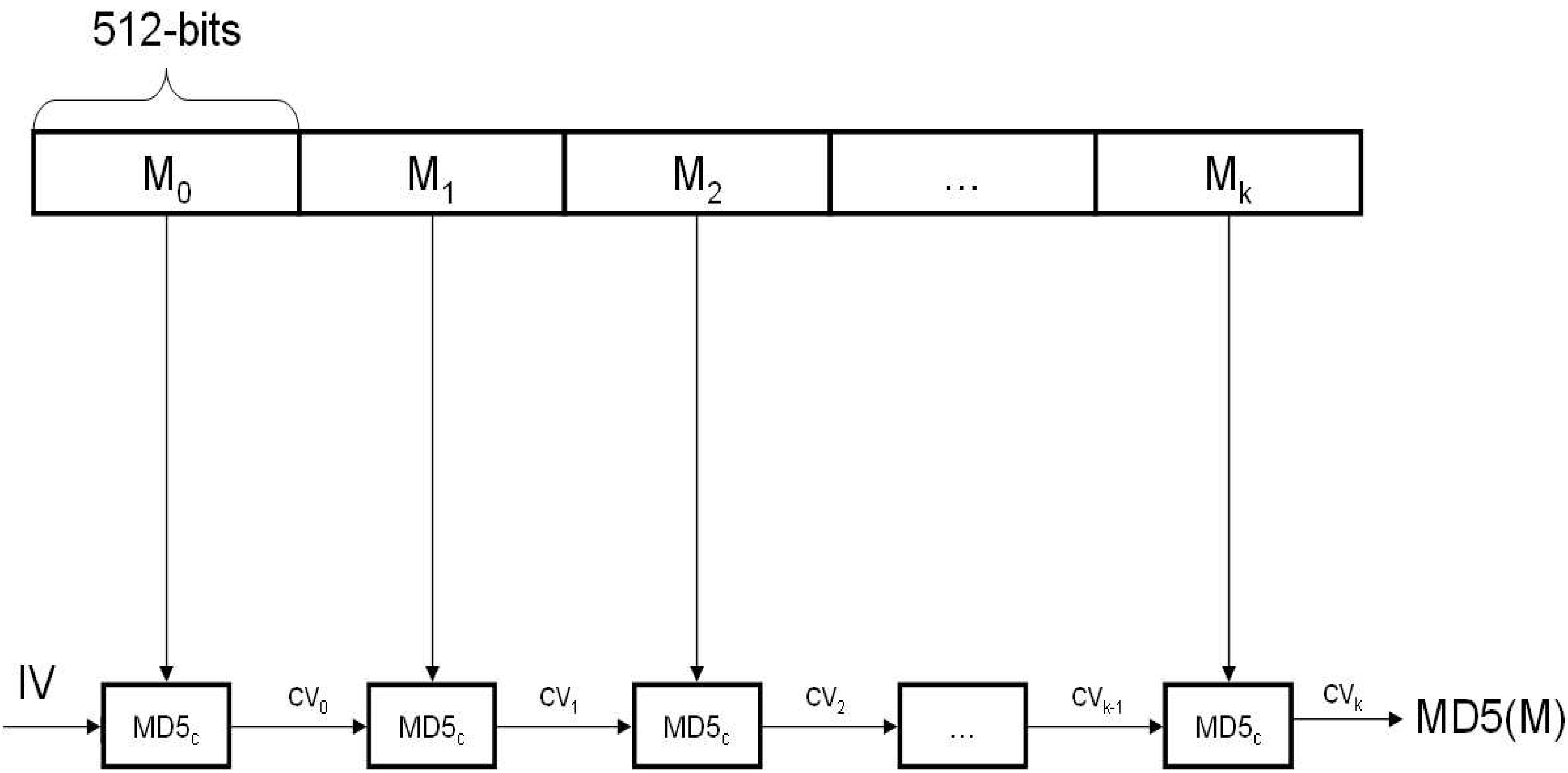
Trevor Highland
University of Texas

MD5 Algorithm

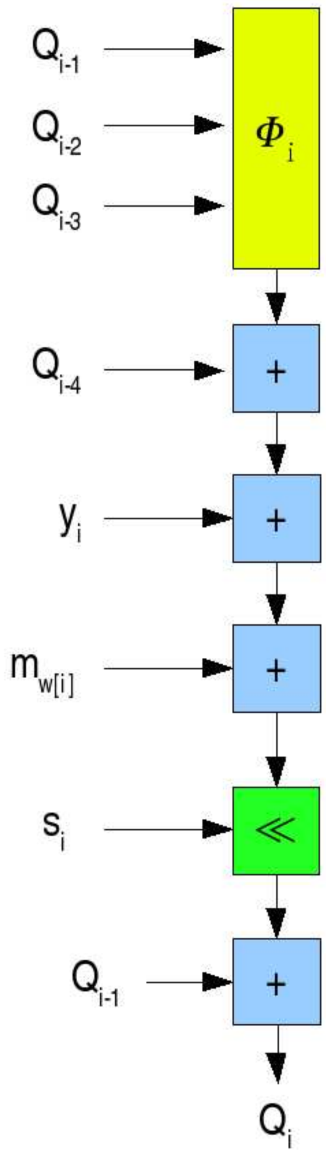
- MD5 is an iterated hash function which operates on 512-bit message blocks.
- Input into the MD5 compression function is a 128-bit chaining value CV and a 512-bit message block M
- M is split into 16 32-bit words $m_{0:15}$
- The output of the MD5 Compression function is a 128-bit chaining value CV'
- MD5 compress has 4 rounds of 16 steps. Each round uses a unique function.
- The function Φ_i for each step is defined in the following manner:

$$
\begin{aligned}
\Phi_i(x,y,z) &= F(x,y,z) = (x \wedge y) \vee (\neg x \wedge z), && 0 \le i \le 15 \\
\Phi_i(x,y,z) &= G(x,y,z) = (x \wedge z) \vee (y \wedge \neg z), && 16 \le i \le 31 \\
\Phi_i(x,y,z) &= H(x,y,z) = x \oplus y \oplus z, && 32 \le i \le 47 \\
\Phi_i(x,y,z) &= I(x,y,z) = y \oplus (x \vee \neg z), && 48 \le i \le 63
\end{aligned}
$$

MD5 Algorithm



MD5 Compress



+ is addition modulo 2^{32}

Φ_i is the Φ function to be used for the i-th step value

$\ll$ is a left rotation by s_i bits

y_i is a step dependent constant

s_i is a step dependent shift constant

w[i] is a mapping from [0,63] to [0,15]

History of MD5 Attacks

- 1993
  - den Boer and Bosselaers announced a free start collision.
- 1996
  - Hans Dobbertin published documentation of a collision in the MD5 compression function.
  - The attack used chosen IV's different from MD5's.
- 2004-2005
  - At CRYPTO 2004 Xiaoyun Wang announced collisions in MD5.
  - Hawkes, Paddon, and Rose published a paper describing the derivation of conditions prior to Wang's paper being released.
  - The Wang team published a paper describing the attack at the 2005 EUROCRYPT conference.
  - Vlastimil Klima released a paper providing a detailed description of how to produce collisions.
  - Stach-Liu released an implementation of the attack that produces collisions in 45 minutes.

High Level Overview

- Each collision consists of 2 1024-bit messages M and M' which only differ in 6 specified bits.

0	0	0	0	2^{31}	0	0	0	0	0	0	2^{15}	0	0	2^{31}	0

0	0	0	0	2^{31}	0	0	0	0	0	0	-2^{15}	0	0	2^{31}	0

- Each collision follows the same general differential path. [Wa05]
- The differential path describes bit differences in each step value for the calculation MD5(M) versus MD5(M').
- A set of conditions on step values Q_i were derived. [Wa05]
- When these conditions are satisfied the differential path holds with high probability.
- The first and second block of a message can be generated independently.
- Most conditions can be satisfied deterministically. All remaining conditions are satisfied probabilistically.

Attack Overview

Algorithm Find_Collision'
while collision_found is false **do**

1. Select values $Q_{0:15}$ arbitrarily.
2. Modify $Q_{0:15}$ to satisfy all first round conditions and differentials.
3. Compute $m_{0:15}$ from these values of $Q_{0:15}$.
4. Satisfy all possible second round conditions and differentials using multi-message modification methods.
5. Check to see if all other conditions and differentials are satisfied.
6. **if** (all differentials satisfied) **then** collision_found $\leftarrow$ true
7. **else** collision_found $\leftarrow$ false

end do
return M

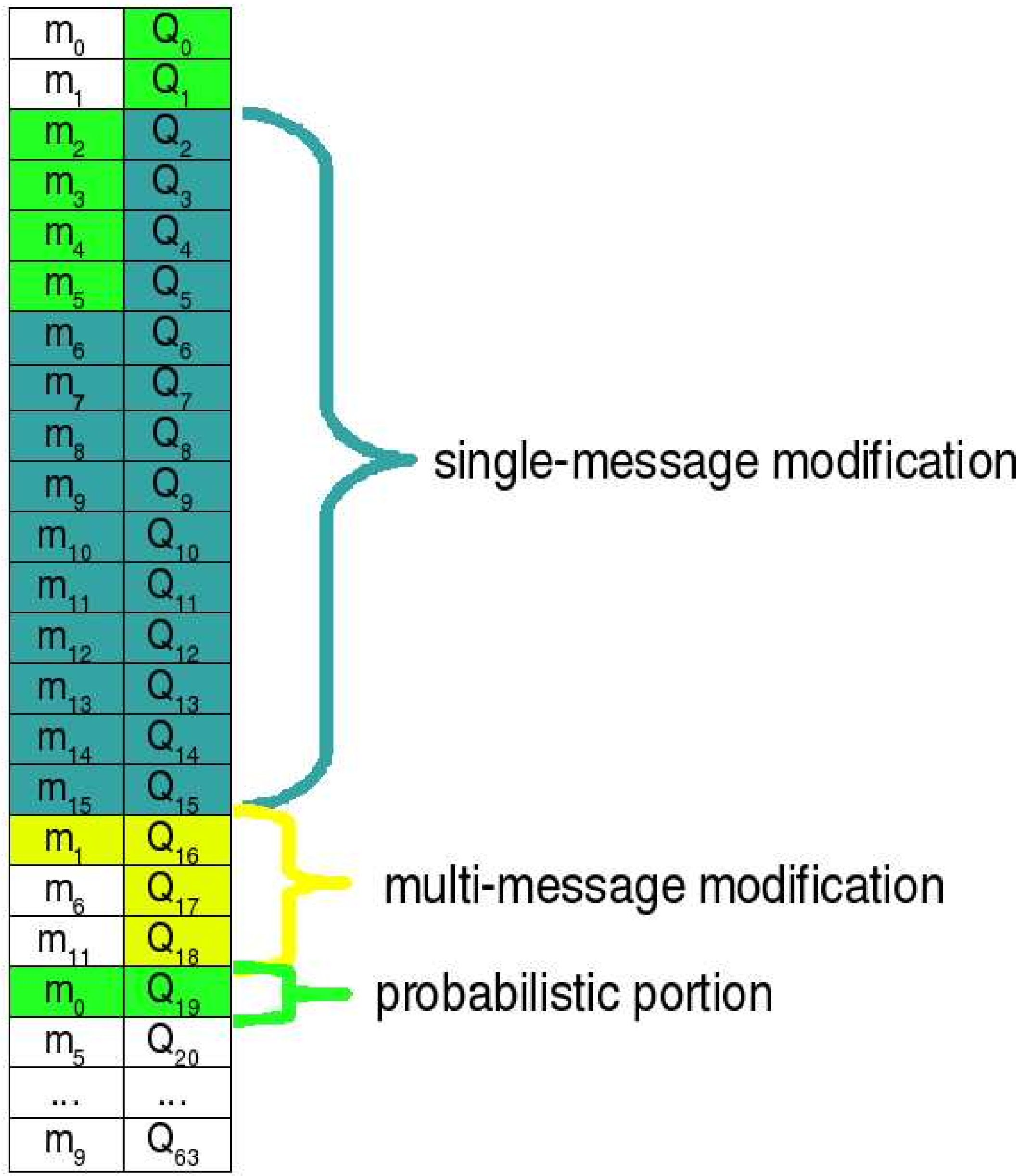
Message Modification

- **Single-message modification [Klima05]**
  - Process of satisfying all conditions on first round step values.
  - For $0 \leq i \leq 15$ do the following:
    1. Randomly select a value for Q_i which satisfies all conditions for that step value.
    2. Calculate m_i
       $m_i \leftarrow ((Q_i - Q_{i-1}) \ggg s_i) - T_i - Q_{i-4} - \Phi_i(Q_{i-1}, Q_{i-2}, Q_{i-3})$
- **Multi-message modification**
  - Process of satisfying second round conditions while leaving first round conditions satisfied.
  - This process is complicated by the fact that message words m_i are being used for the second time.
  - There are many approaches, that work with varying degrees of success.

Multi-Message Modification

- Satisfy conditions for step values $Q_{16:n}$ by modifying chosen step values in the range of $Q_{0:15}$.
- Once these conditions are satisfied the probabilistic portion of the attack begins.
- Find a method to generate a sufficient number of messages, which preserve all previously satisfied conditions.
- If all conditions are satisfied through Q_n, generated messages must change some input for the calculation Q_{n+1}
- The same process cannot be used for generating first and second block messages, because differences in conditions on $Q_{0:15}$ restrict the process.

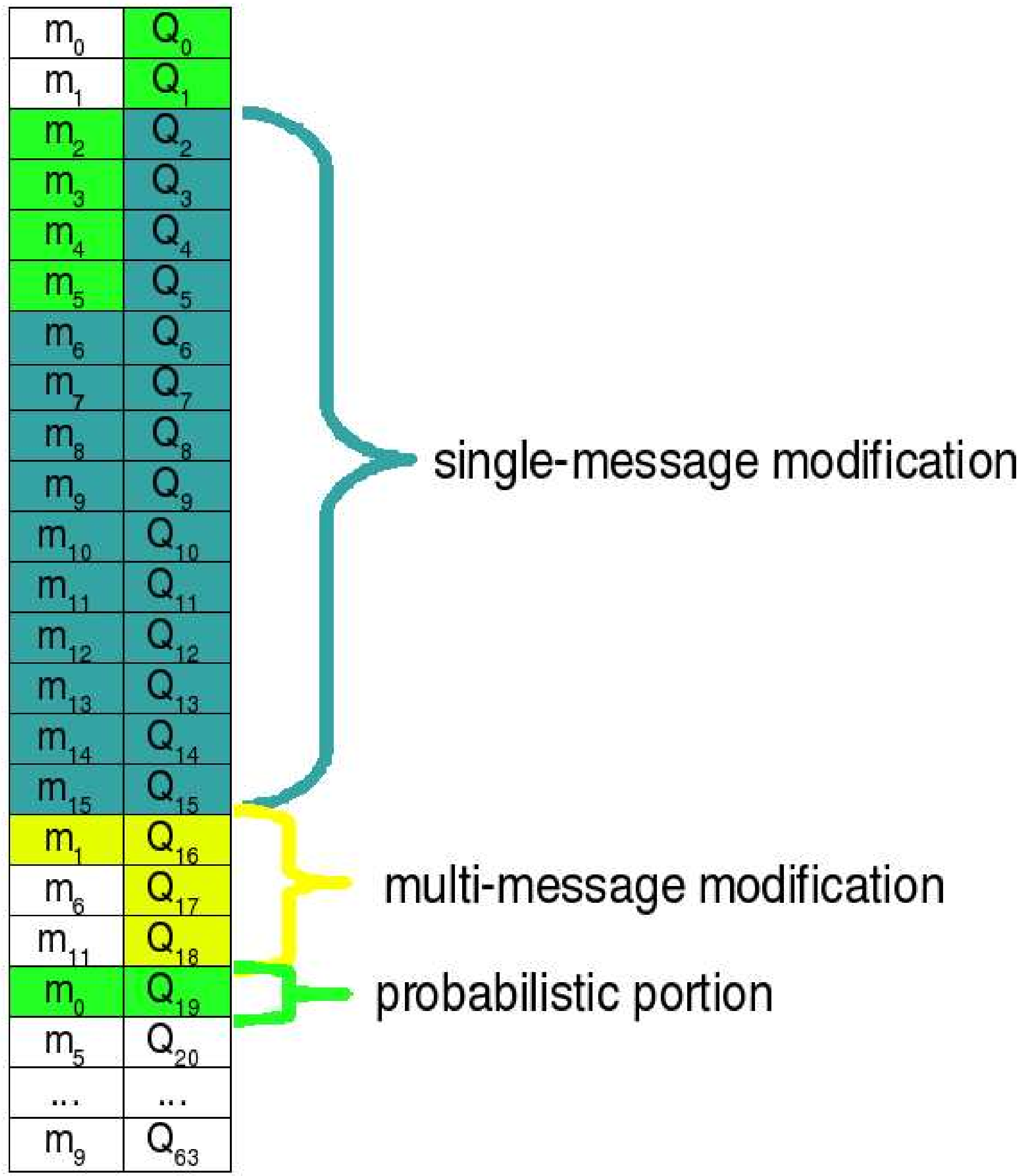
Message Modification



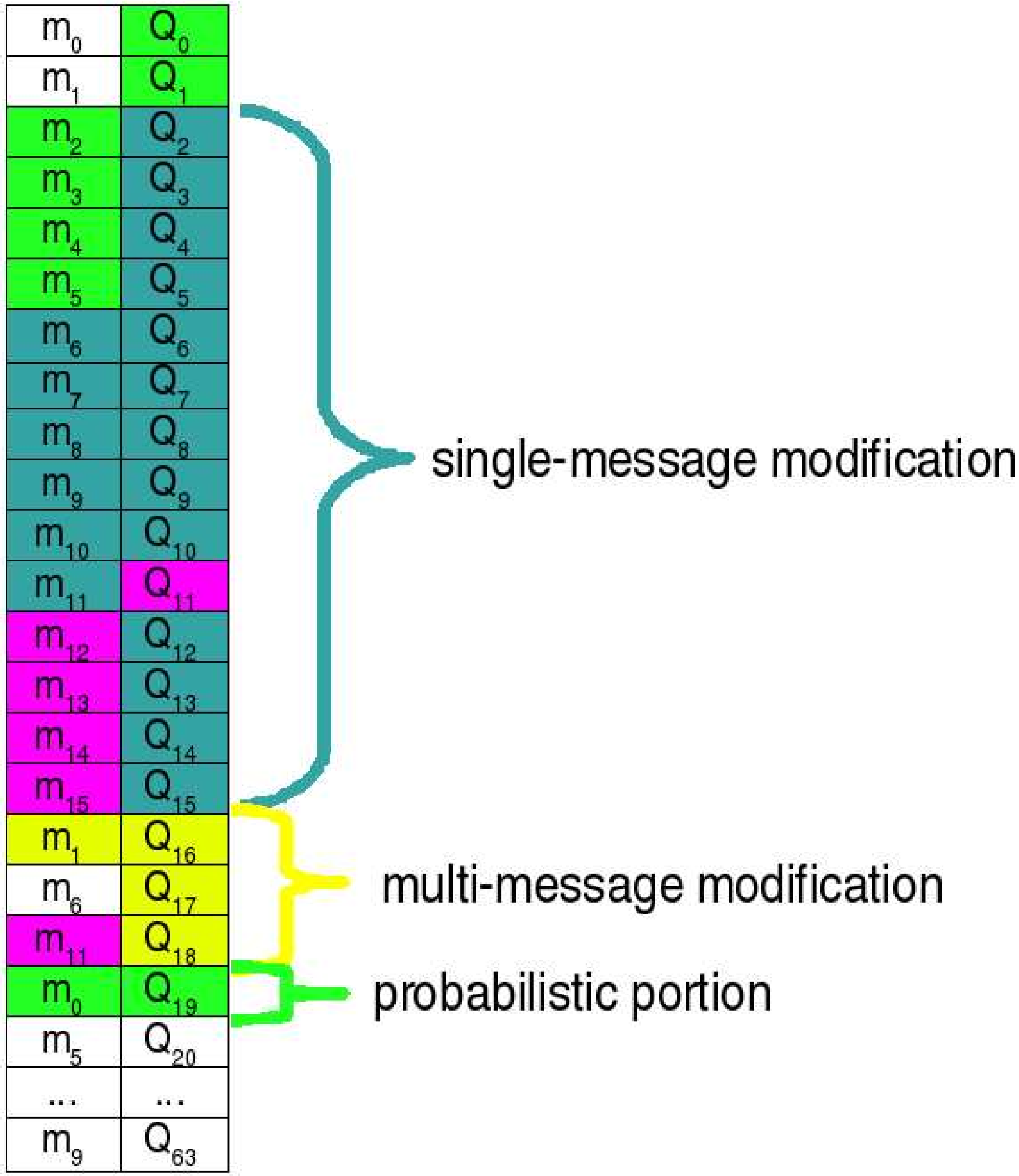
New Message Modification Techniques

- We developed a method that satisfies the 3 conditions for Q_{20} and Q_{21} with probability 15/16.
- We add 12 conditions which are all easy to satisfy.
  - provide a method to change the bits that the conditions are placed on
  - provide a way to update the message while preserving all previously satisfied conditions

Message Modification



Message Modification



New Message Modification Techniques

- Analysis of the probabilistic portion of the attack showed that the satisfaction of certain conditions has a predictable distribution.
- Messages for the probabilistic portion of the first block method are generated by iterating through values of Q_{19}
- Conditions on Q_{20} are often unsatisfied for 128 consecutive values of Q_{19}.
- Add 127 to Q_{19} anytime conditions on Q_{20} are not satisfied.
- This method leads to 97% of messages satisfying both conditions on Q_{20}, a vast improvement from the expected 25%.

Performance

- Our implementation takes 9 minutes on average to produce a collision using a Pentium 4 running Linux.
- Previously Stach-Liu had the fastest running code which found collisions in 45 minutes.
  - This code was unable to generate a second block message for many first block messages.
- First block algorithm
  - 30 conditions must be satisfied probabilistically.
  - We were able to genereate 80 first block messages in 10 hours.
  - Each message took an average of 455 seconds..
- Second block algorithm
  - 24 conditions must be satisfied probabilistically.
  - 80 second block messages were generated in 2 hours 6 minutes.
  - Each message took an average of 95 seconds.

Changes to Conditions and other Corrections

- Conditions previously labeled as sufficient are not actually sufficient.
  - This does not pose any problems if differentials are verified at critical points.
  - Many updates to the list of sufficient conditions appear on eprint.
  - The combination of all updates are still not quite sufficient.
  - One update to the list of conditions successfully removes a condition on the 1st block chaining values[YS05].
- Our collisions had small differences in the subtraction differentials. Our tables have been updated to reflect these differences.

Future Work

- Formalization of new message modification techniques.
  - Our analysis for the distribution of when conditions are satisfed was limited to just a few conditions.
  - Further analysis could reveal why the satisfaction of certain conditions has a predictable distribution.
  - The process of analyzing the distribution of conditions being satisfed could be automated.
- Finding a general method to find new differential paths through MD5.
  - This work could likely be extended to more powerful hash functions such as SHA-1.
- Applying techniques described in this paper to other hash functions with similar attacks.

Paper and Source Code

- The full version of the paper located at:

  http://www.cs.colorado.edu/~jrblack/papers.html

  - insights into how the differential path was likely produced
  - second block multi-message modification techniques
  - full description of all new multi-message modification techniques

- The software is available at:

  www.cs.colorado.edu/~jrblack/md5toolkit.tar.gz

  - source code written in C
  - conditions can easily be updated
  - IV used to produce the collision can be changed