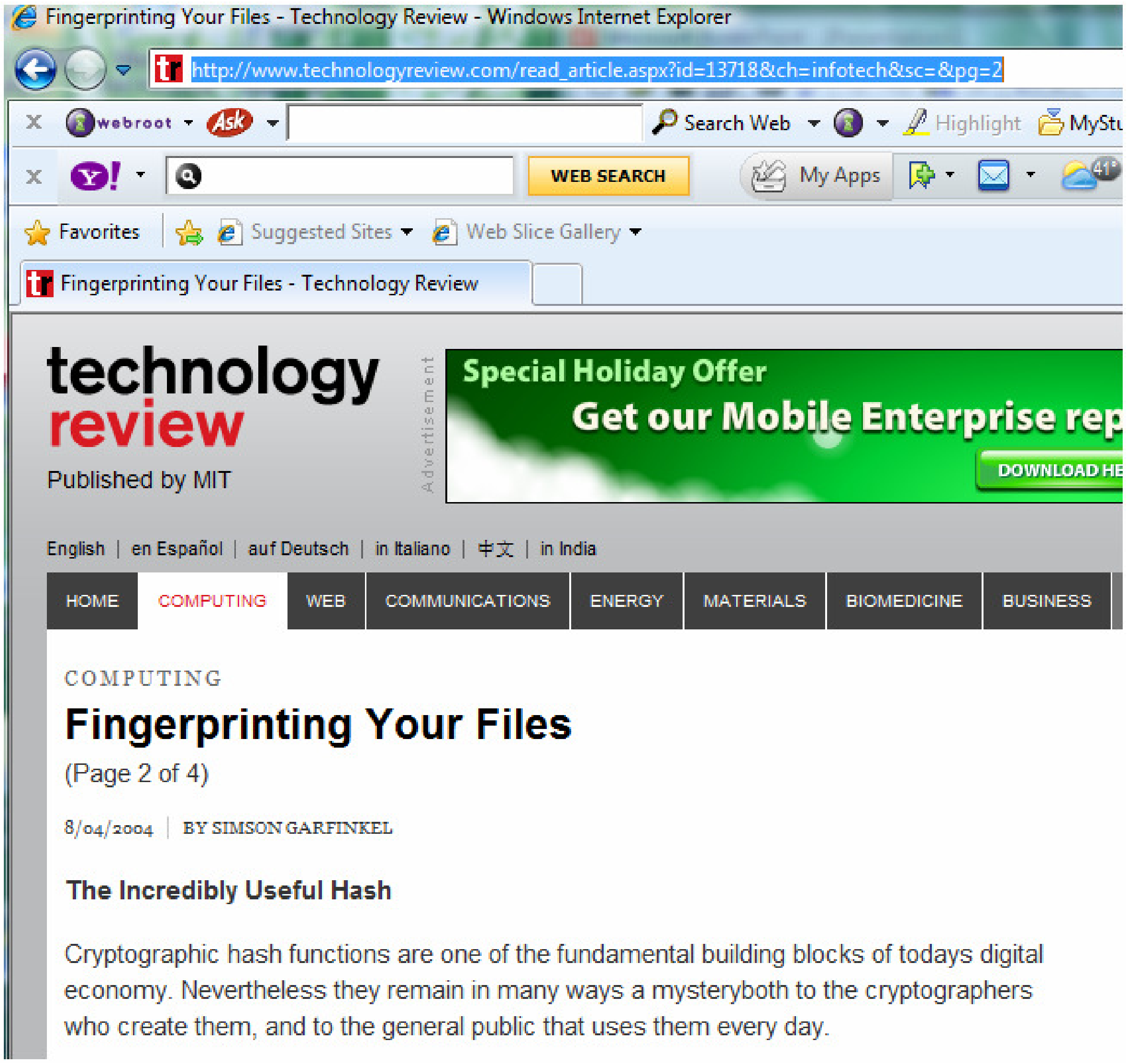
http://www.technologyreview.com/read_article.aspx?id=13718&ch=infotech&sc=&pg=2


Fingerprinting Your Files - Technology Review - Windows Internet Explorer
http://www.technologyreview.com/read_article.aspx?id=13718&ch=infotech&sc=&pg=2
webroot
Ask
Search Web
Highlight
MyStu
WEB SEARCH
My Apps
Favorites
Suggested Sites
Web Slice Gallery
Fingerprinting Your Files - Technology Review
technology review
Published by MIT
Advertisement
Special Holiday Offer
Get our Mobile Enterprise rep
DOWNLOAD HE
English | en Español | auf Deutsch | in Italiano | 中文 | in India
HOME
COMPUTING
WEB
COMMUNICATIONS
ENERGY
MATERIALS
BIOMEDICINE
BUSINESS

COMPUTING

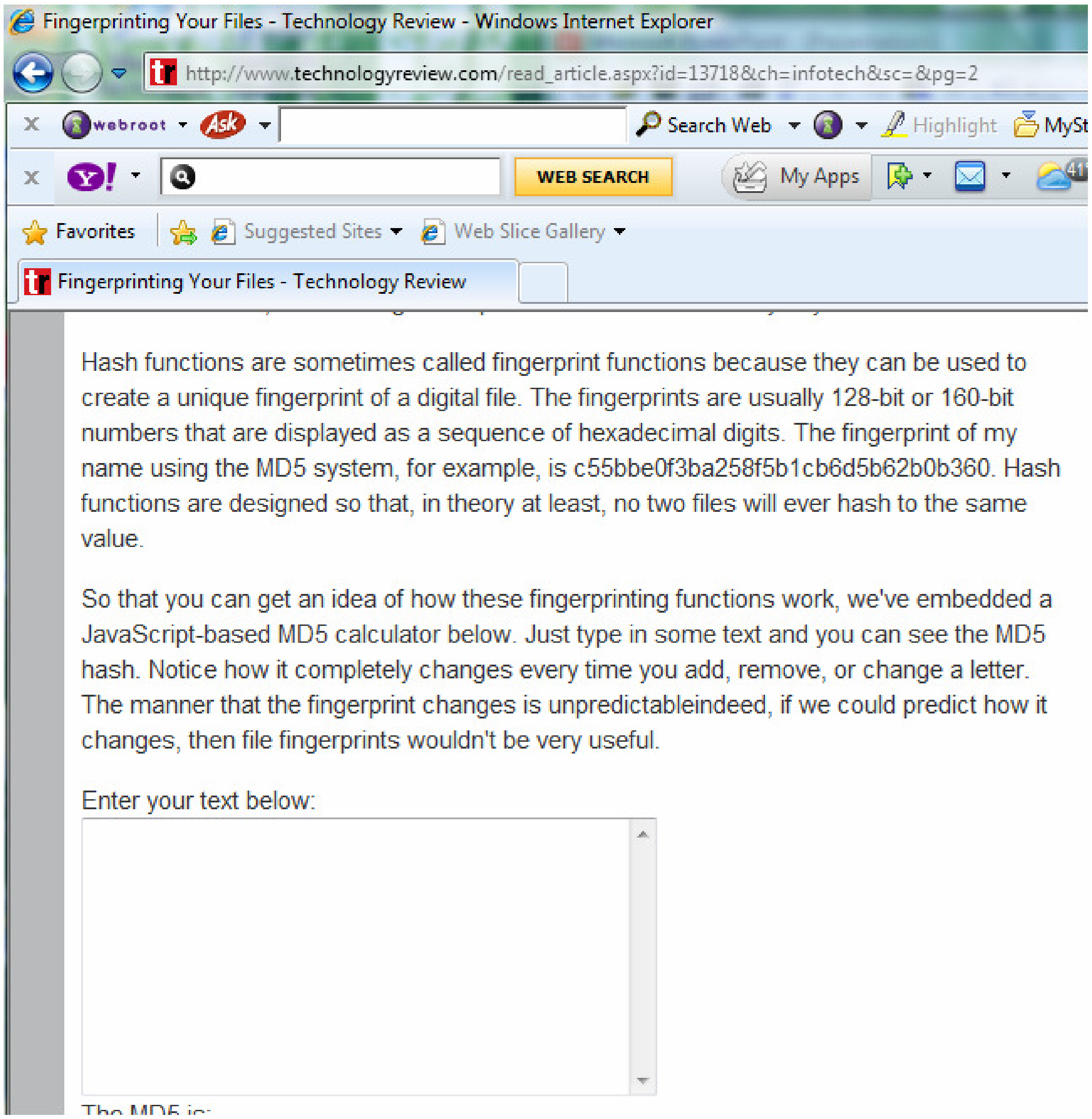
Fingerprinting Your Files

(Page 2 of 4)

8/04/2004 | BY SIMSON GARFINKEL

The Incredibly Useful Hash

Cryptographic hash functions are one of the fundamental building blocks of todays digital economy. Nevertheless they remain in many ways a mysteryboth to the cryptographers who create them, and to the general public that uses them every day.

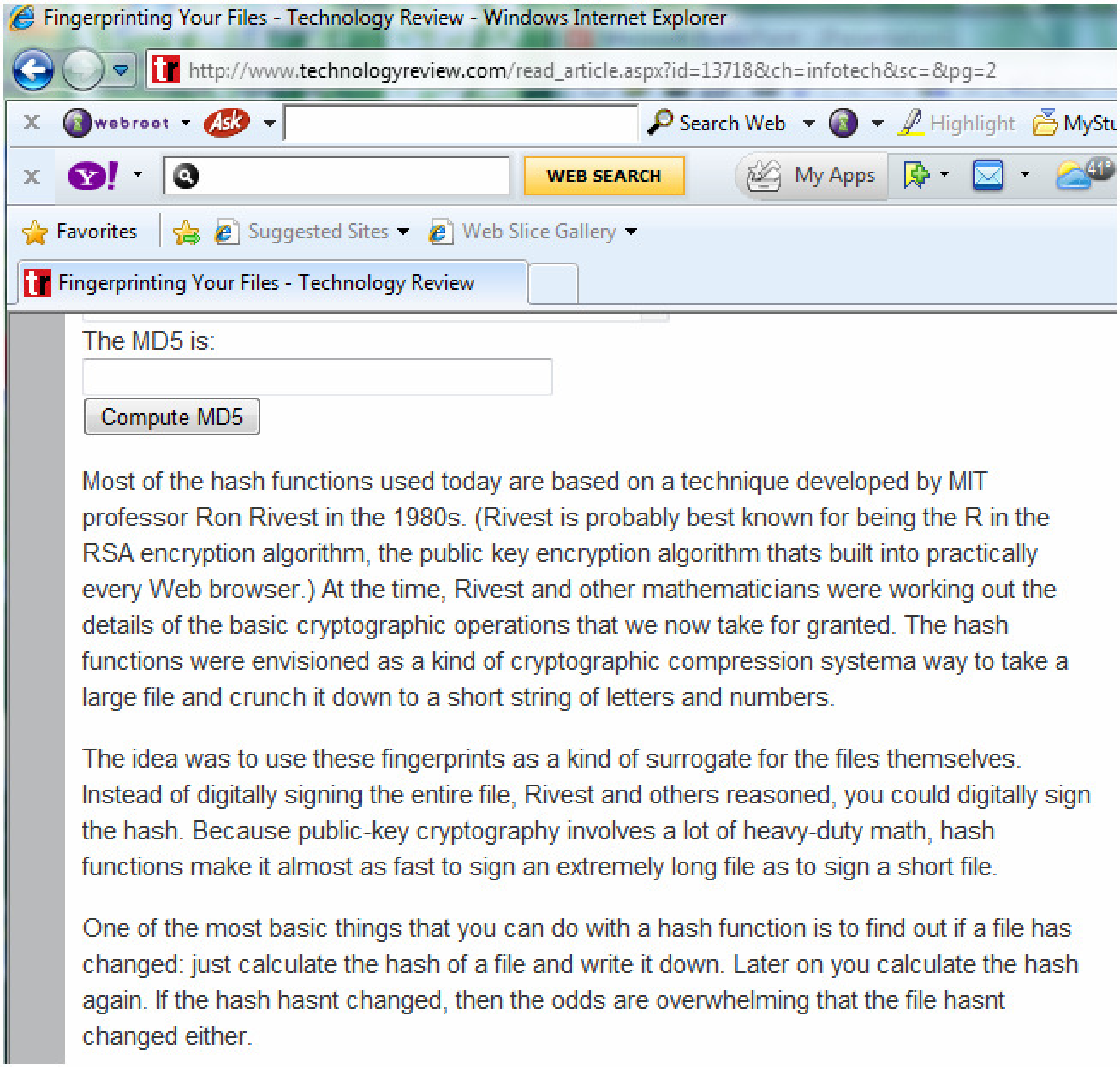

Fingerprinting Your Files - Technology Review - Windows Internet Explorer
http://www.technologyreview.com/read_article.aspx?id=13718&ch=infotech&sc=&pg=2
webroot
Ask
Search Web
Highlight
MySt
WEB SEARCH
My Apps
Favorites
Suggested Sites
Web Slice Gallery
Fingerprinting Your Files - Technology Review

Hash functions are sometimes called fingerprint functions because they can be used to create a unique fingerprint of a digital file. The fingerprints are usually 128-bit or 160-bit numbers that are displayed as a sequence of hexadecimal digits. The fingerprint of my name using the MD5 system, for example, is c55bbe0f3ba258f5b1cb6d5b62b0b360. Hash functions are designed so that, in theory at least, no two files will ever hash to the same value.

So that you can get an idea of how these fingerprinting functions work, we've embedded a JavaScript-based MD5 calculator below. Just type in some text and you can see the MD5 hash. Notice how it completely changes every time you add, remove, or change a letter. The manner that the fingerprint changes is unpredictableindeed, if we could predict how it changes, then file fingerprints wouldn't be very useful.

Enter your text below:

The MD5 is:


Fingerprinting Your Files - Technology Review - Windows Internet Explorer
http://www.technologyreview.com/read_article.aspx?id=13718&ch=infotech&sc=&pg=2
webroot
Ask
Search Web
Highlight
MySt
WEB SEARCH
My Apps
Favorites
Suggested Sites
Web Slice Gallery
Fingerprinting Your Files - Technology Review

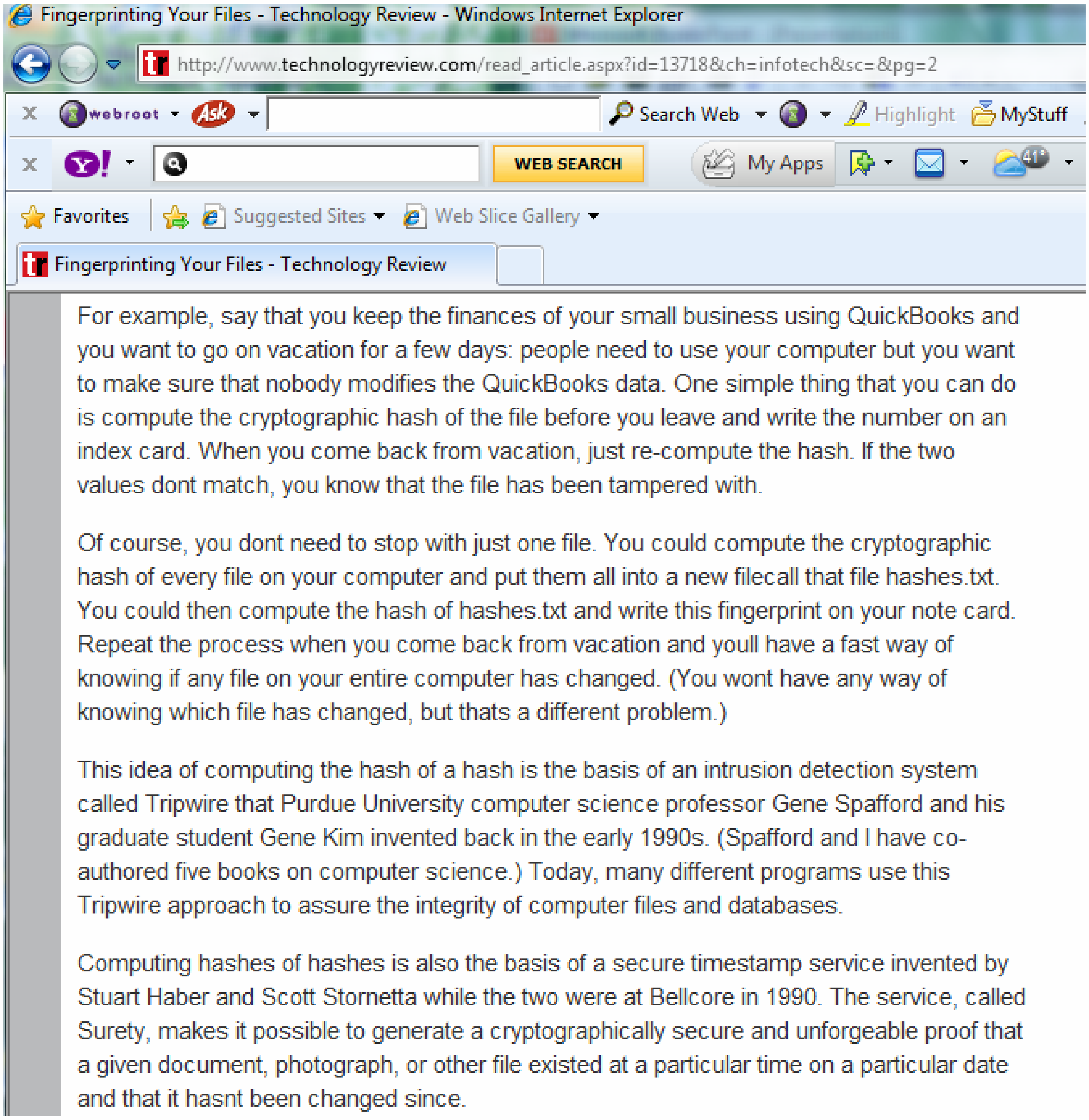
The MD5 is:

Compute MD5

Most of the hash functions used today are based on a technique developed by MIT professor Ron Rivest in the 1980s. (Rivest is probably best known for being the R in the RSA encryption algorithm, the public key encryption algorithm thats built into practically every Web browser.) At the time, Rivest and other mathematicians were working out the details of the basic cryptographic operations that we now take for granted. The hash functions were envisioned as a kind of cryptographic compression systema way to take a large file and crunch it down to a short string of letters and numbers.

The idea was to use these fingerprints as a kind of surrogate for the files themselves. Instead of digitally signing the entire file, Rivest and others reasoned, you could digitally sign the hash. Because public-key cryptography involves a lot of heavy-duty math, hash functions make it almost as fast to sign an extremely long file as to sign a short file.

One of the most basic things that you can do with a hash function is to find out if a file has changed: just calculate the hash of a file and write it down. Later on you calculate the hash again. If the hash hasnt changed, then the odds are overwhelming that the file hasnt changed either.

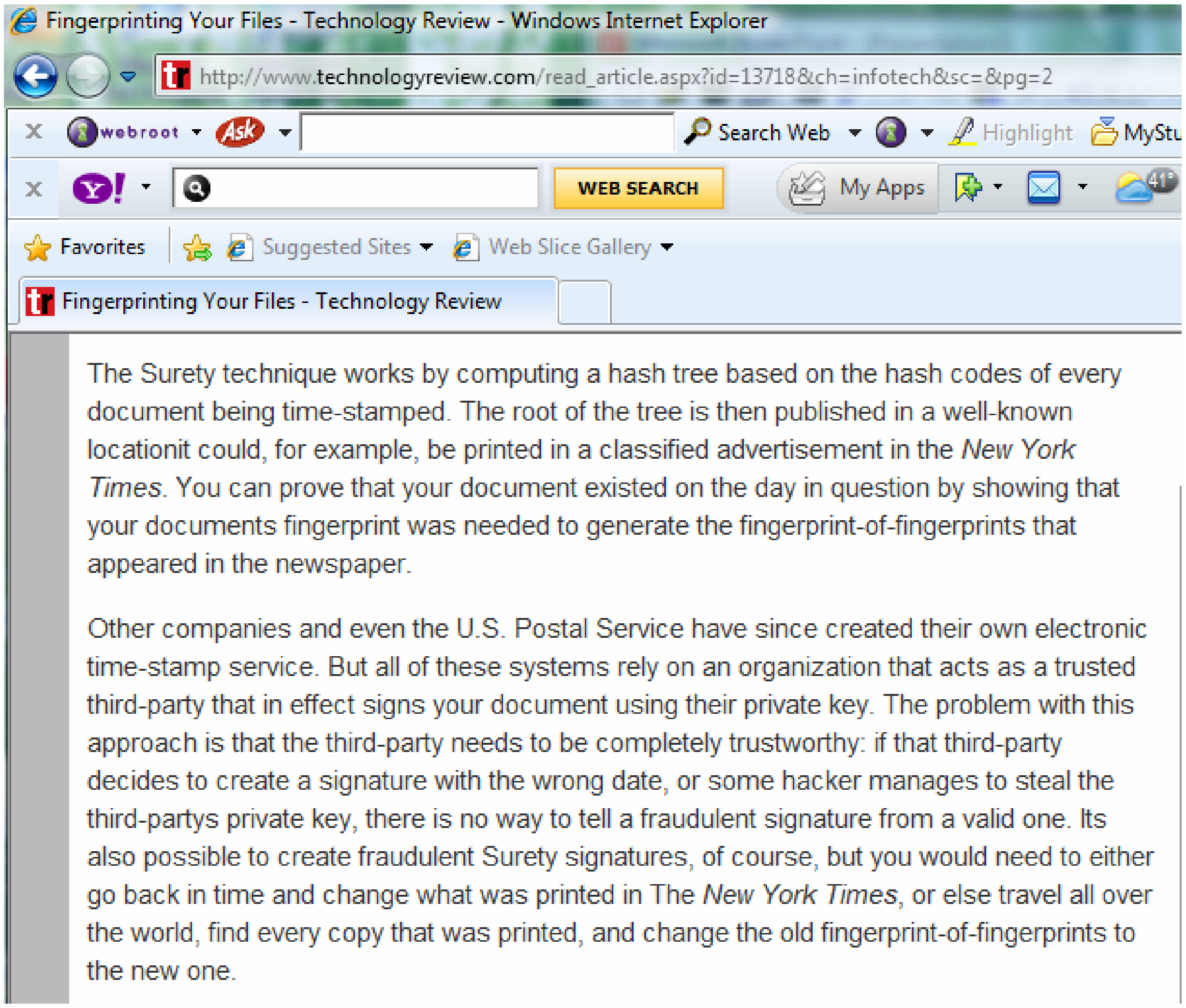

Fingerprinting Your Files - Technology Review - Windows Internet Explorer
http://www.technologyreview.com/read_article.aspx?id=13718&ch=infotech&sc=&pg=2
webroot
Ask
Search Web
Highlight
MyStuff
WEB SEARCH
My Apps
41°
Favorites
Suggested Sites
Web Slice Gallery
Fingerprinting Your Files - Technology Review

For example, say that you keep the finances of your small business using QuickBooks and you want to go on vacation for a few days: people need to use your computer but you want to make sure that nobody modifies the QuickBooks data. One simple thing that you can do is compute the cryptographic hash of the file before you leave and write the number on an index card. When you come back from vacation, just re-compute the hash. If the two values dont match, you know that the file has been tampered with.

Of course, you dont need to stop with just one file. You could compute the cryptographic hash of every file on your computer and put them all into a new filecall that file hashes.txt. You could then compute the hash of hashes.txt and write this fingerprint on your note card. Repeat the process when you come back from vacation and youll have a fast way of knowing if any file on your entire computer has changed. (You wont have any way of knowing which file has changed, but thats a different problem.)

This idea of computing the hash of a hash is the basis of an intrusion detection system called Tripwire that Purdue University computer science professor Gene Spafford and his graduate student Gene Kim invented back in the early 1990s. (Spafford and I have co-authored five books on computer science.) Today, many different programs use this Tripwire approach to assure the integrity of computer files and databases.

Computing hashes of hashes is also the basis of a secure timestamp service invented by Stuart Haber and Scott Stornetta while the two were at Bellcore in 1990. The service, called Surety, makes it possible to generate a cryptographically secure and unforgeable proof that a given document, photograph, or other file existed at a particular time on a particular date and that it hasnt been changed since.



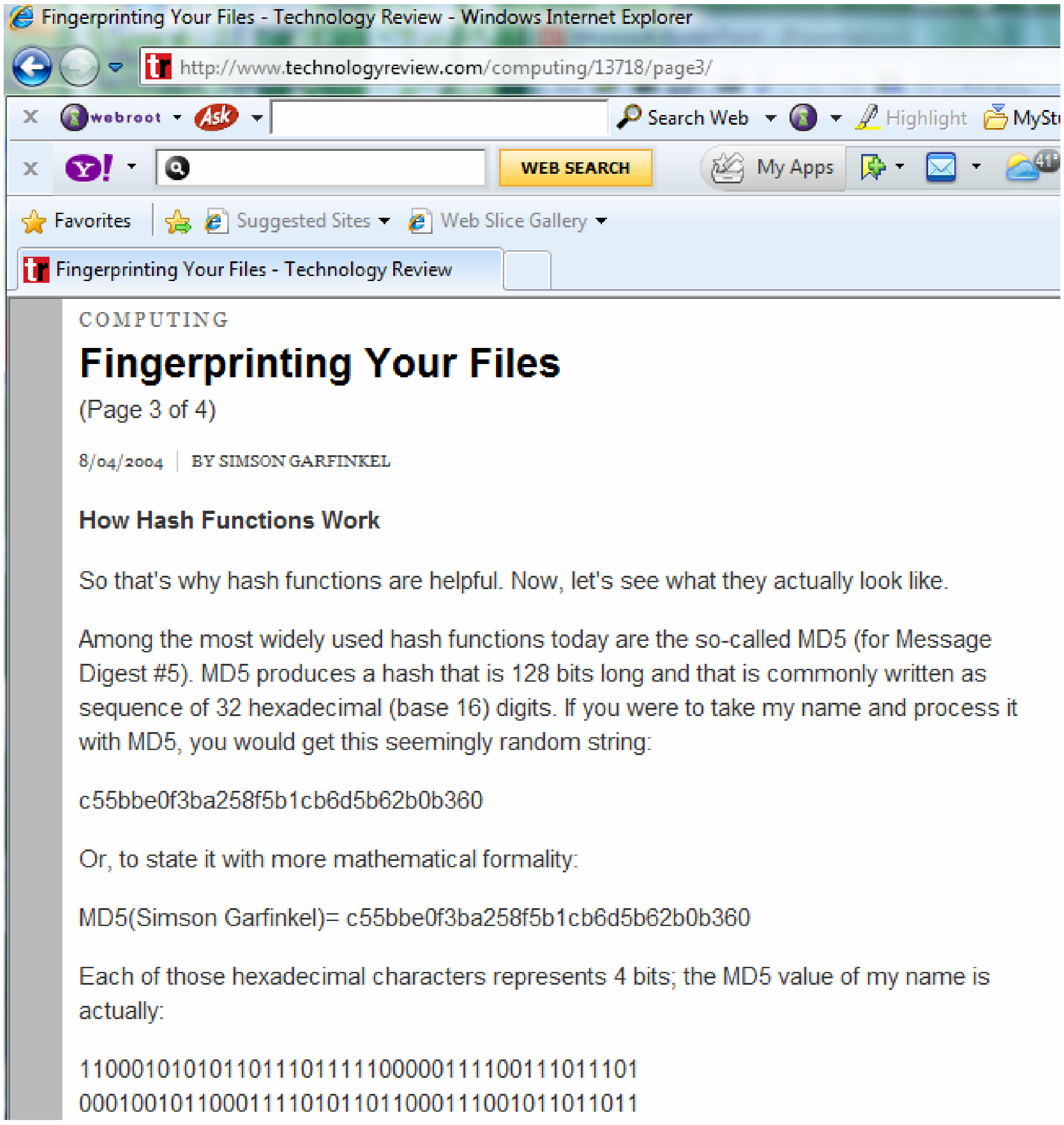
Fingerprinting Your Files - Technology Review - Windows Internet Explorer

http://www.technologyreview.com/read_article.aspx?id=13718&ch=infotech&sc=&pg=2

Fingerprinting Your Files - Technology Review

The Surety technique works by computing a hash tree based on the hash codes of every document being time-stamped. The root of the tree is then published in a well-known locationit could, for example, be printed in a classified advertisement in the *New York Times*. You can prove that your document existed on the day in question by showing that your documents fingerprint was needed to generate the fingerprint-of-fingerprints that appeared in the newspaper.

Other companies and even the U.S. Postal Service have since created their own electronic time-stamp service. But all of these systems rely on an organization that acts as a trusted third-party that in effect signs your document using their private key. The problem with this approach is that the third-party needs to be completely trustworthy: if that third-party decides to create a signature with the wrong date, or some hacker manages to steal the third-partys private key, there is no way to tell a fraudulent signature from a valid one. Its also possible to create fraudulent Surety signatures, of course, but you would need to either go back in time and change what was printed in The *New York Times*, or else travel all over the world, find every copy that was printed, and change the old fingerprint-of-fingerprints to the new one.



Fingerprinting Your Files - Technology Review - Windows Internet Explorer
http://www.technologyreview.com/computing/13718/page3/
Search Web
Highlight
WEB SEARCH
My Apps
Favorites
Suggested Sites
Web Slice Gallery
Fingerprinting Your Files - Technology Review

COMPUTING

Fingerprinting Your Files

(Page 3 of 4)

8/04/2004 | BY SIMSON GARFINKEL

How Hash Functions Work

So that's why hash functions are helpful. Now, let's see what they actually look like.

Among the most widely used hash functions today are the so-called MD5 (for Message Digest #5). MD5 produces a hash that is 128 bits long and that is commonly written as sequence of 32 hexadecimal (base 16) digits. If you were to take my name and process it with MD5, you would get this seemingly random string:

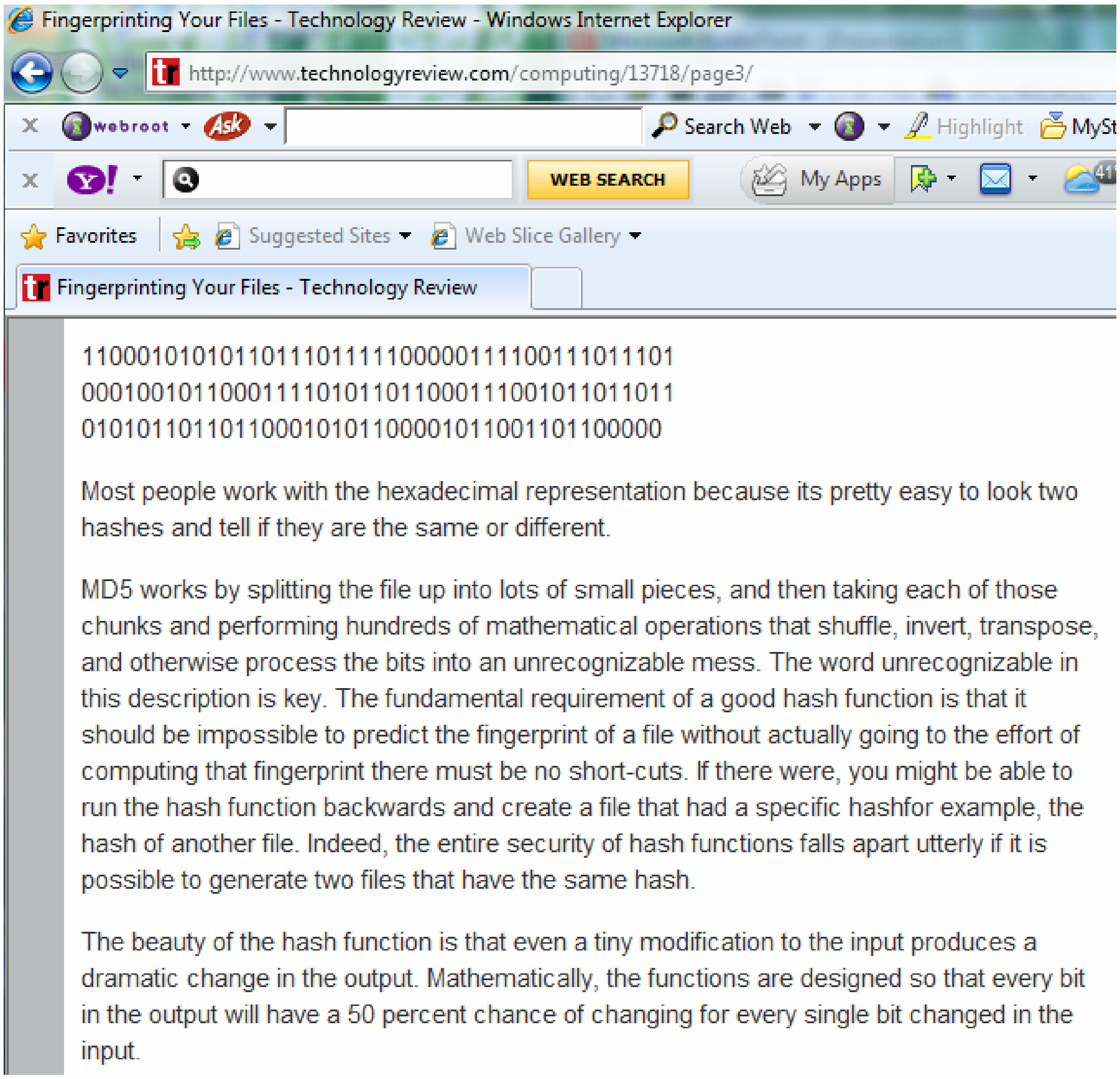
c55bbe0f3ba258f5b1cb6d5b62b0b360

Or, to state it with more mathematical formality:

MD5(Simson Garfinkel)= c55bbe0f3ba258f5b1cb6d5b62b0b360

Each of those hexadecimal characters represents 4 bits; the MD5 value of my name is actually:

11000101010110111011111000001111001110111011
00010010110001111010110110001110010110110110

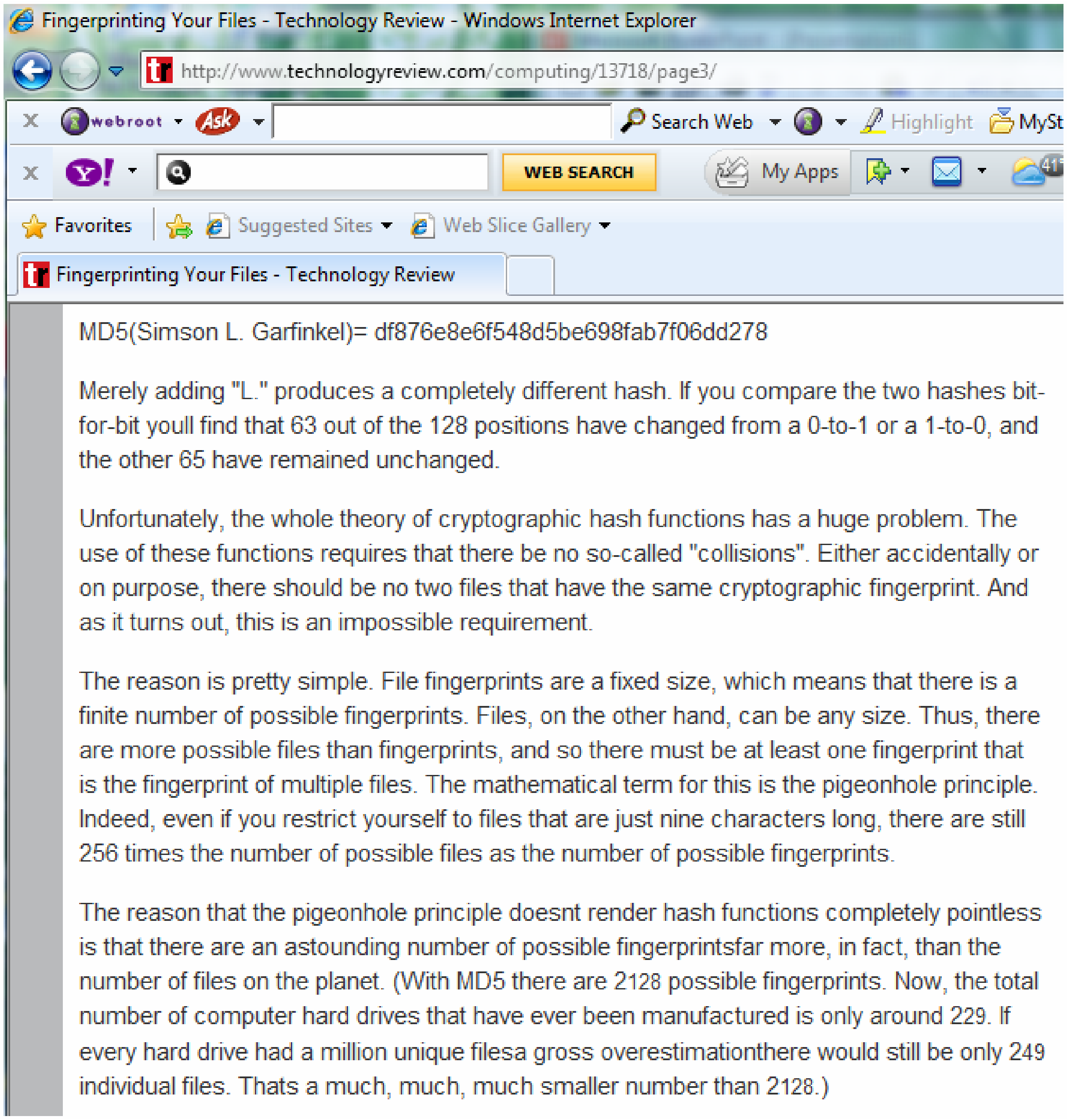

Fingerprinting Your Files - Technology Review - Windows Internet Explorer
http://www.technologyreview.com/computing/13718/page3/
webroot
Ask
Search Web
Highlight
MySt
WEB SEARCH
My Apps
Favorites
Suggested Sites
Web Slice Gallery
Fingerprinting Your Files - Technology Review

```
1100010101011011101111100000111100111011101
0001001011000111101011011000111001011011011
0101011011011000101011000010110011011100000
```

Most people work with the hexadecimal representation because its pretty easy to look two hashes and tell if they are the same or different.

MD5 works by splitting the file up into lots of small pieces, and then taking each of those chunks and performing hundreds of mathematical operations that shuffle, invert, transpose, and otherwise process the bits into an unrecognizable mess. The word unrecognizable in this description is key. The fundamental requirement of a good hash function is that it should be impossible to predict the fingerprint of a file without actually going to the effort of computing that fingerprint there must be no short-cuts. If there were, you might be able to run the hash function backwards and create a file that had a specific hashfor example, the hash of another file. Indeed, the entire security of hash functions falls apart utterly if it is possible to generate two files that have the same hash.

The beauty of the hash function is that even a tiny modification to the input produces a dramatic change in the output. Mathematically, the functions are designed so that every bit in the output will have a 50 percent chance of changing for every single bit changed in the input.


Fingerprinting Your Files - Technology Review - Windows Internet Explorer
http://www.technologyreview.com/computing/13718/page3/
webroot
Ask
Search Web
Highlight
MySt
WEB SEARCH
My Apps
Favorites
Suggested Sites
Web Slice Gallery
Fingerprinting Your Files - Technology Review

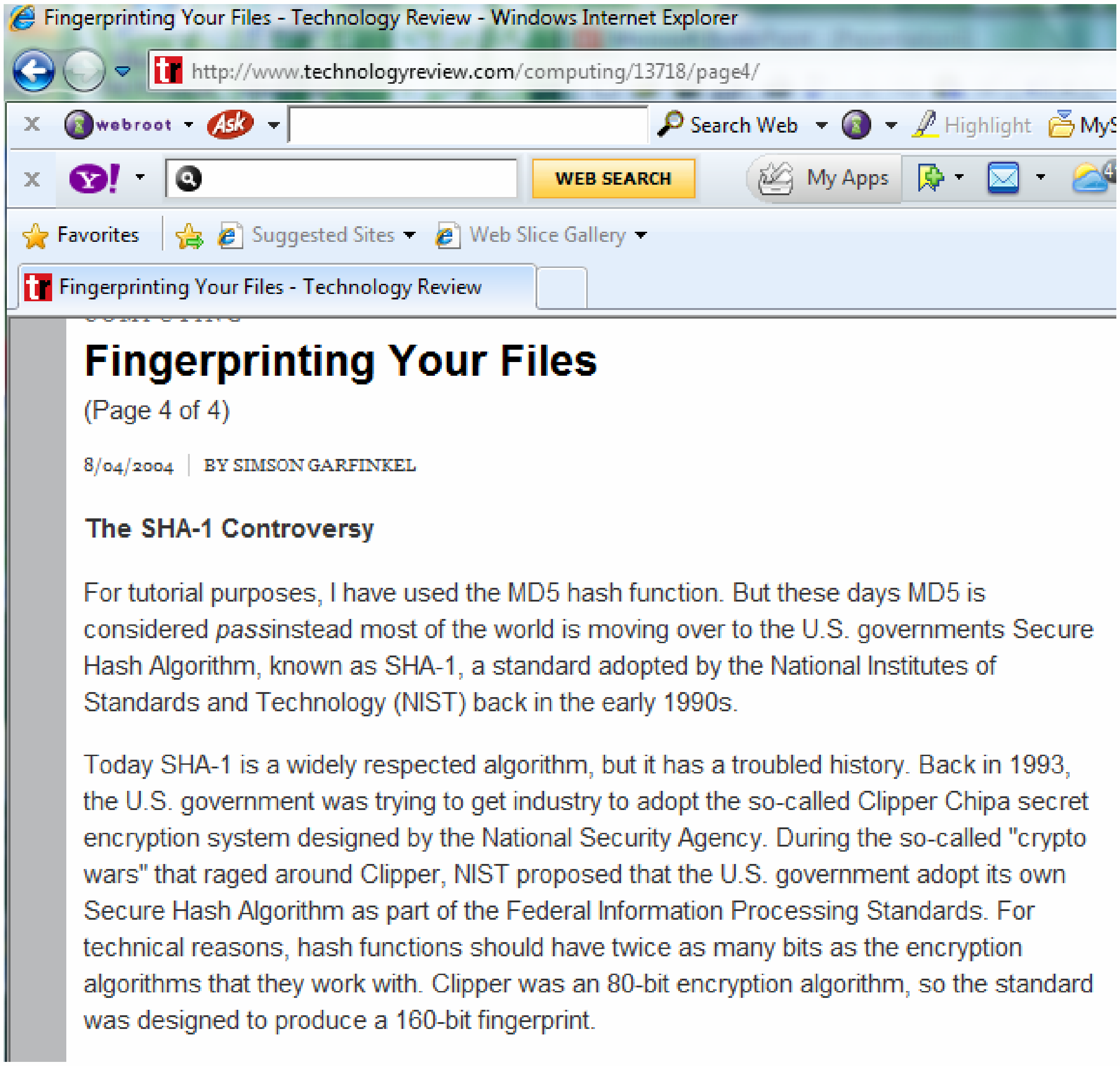
MD5(Simson L. Garfinkel)= df876e8e6f548d5be698fab7f06dd278

Merely adding "L." produces a completely different hash. If you compare the two hashes bit-for-bit youll find that 63 out of the 128 positions have changed from a 0-to-1 or a 1-to-0, and the other 65 have remained unchanged.

Unfortunately, the whole theory of cryptographic hash functions has a huge problem. The use of these functions requires that there be no so-called "collisions". Either accidentally or on purpose, there should be no two files that have the same cryptographic fingerprint. And as it turns out, this is an impossible requirement.

The reason is pretty simple. File fingerprints are a fixed size, which means that there is a finite number of possible fingerprints. Files, on the other hand, can be any size. Thus, there are more possible files than fingerprints, and so there must be at least one fingerprint that is the fingerprint of multiple files. The mathematical term for this is the pigeonhole principle. Indeed, even if you restrict yourself to files that are just nine characters long, there are still 256 times the number of possible files as the number of possible fingerprints.

The reason that the pigeonhole principle doesnt render hash functions completely pointless is that there are an astounding number of possible fingerprintsfar more, in fact, than the number of files on the planet. (With MD5 there are 2128 possible fingerprints. Now, the total number of computer hard drives that have ever been manufactured is only around 229. If every hard drive had a million unique filesa gross overestimationthere would still be only 249 individual files. Thats a much, much, much smaller number than 2128.)


Fingerprinting Your Files - Technology Review - Windows Internet Explorer
http://www.technologyreview.com/computing/13718/page4/
webroot
Ask
Search Web
Highlight
WEB SEARCH
My Apps
Favorites
Suggested Sites
Web Slice Gallery
Fingerprinting Your Files - Technology Review

Fingerprinting Your Files

(Page 4 of 4)

8/04/2004 | BY SIMSON GARFINKEL

The SHA-1 Controversy

For tutorial purposes, I have used the MD5 hash function. But these days MD5 is considered *pass*instead most of the world is moving over to the U.S. governments Secure Hash Algorithm, known as SHA-1, a standard adopted by the National Institutes of Standards and Technology (NIST) back in the early 1990s.

Today SHA-1 is a widely respected algorithm, but it has a troubled history. Back in 1993, the U.S. government was trying to get industry to adopt the so-called Clipper Chipa secret encryption system designed by the National Security Agency. During the so-called "crypto wars" that raged around Clipper, NIST proposed that the U.S. government adopt its own Secure Hash Algorithm as part of the Federal Information Processing Standards. For technical reasons, hash functions should have twice as many bits as the encryption algorithms that they work with. Clipper was an 80-bit encryption algorithm, so the standard was designed to produce a 160-bit fingerprint.