can determine if those assumptions are true." The answer could be important for many endangered species.

Wayne, for his part, is extracting DNA from museum pelts of a now extinct wolf-like animal that inhabited the Falkland Islands until the latter part of the last century. "Charles Darwin wrote the first description of this species," says Wayne, "but about 30 years after he reported it, hunters wiped it out." Since then, various scientists have argued that the animal was actually a wolf, a large fox, or some type of domestic dog that had gone feral. "By extracting the DNA from hairs on the pelts," Wayne says, "this can be resolved definitively."

The questions being addressed aren't limited to animal species; molecular geneticists have had good luck retrieving viable DNA from human remains and using these to address questions of migration patterns. In the May 1993 *Proceedings of the (UK) Royal Society*, Erika Hagelberg and John Clegg of Cambridge University published initial results of a 3-year study of the prehistoric colonization of the South Pacific islands of Oceania. Previous efforts to trace the settlement patterns of the early island-hoppers relied on linguistic and archeological evidence—and indicated that the original settlers were voyagers from Southeast Asia. But Hagelberg sampled mitochondrial DNA sequences taken from human skeletons at a variety of sites in Oceania (some dating as far back as 2700 years before present). By comparing these skeletal genes to mitochondrial DNA from modern inhabitants, Hagelberg has shown that some of the earliest settlers probably came from Melanesia, and not just Southeast Asia.

Other investigators are looking for 13,000-year-old DNA from ground sloths to untangle their family tree. Still others are taking genetic material from 800-year-old skeletons of early Americans—the Hohokum people—to learn if they are related to current Native Americans. Though none of these projects are as romantic as the hunt for dinosaur DNA, together they constitute what Wayne terms "the real nuts and bolts research" of ancient DNA studies. "What we really want to do is reconstruct the historical variability of species, to get a feel for the variation in gene frequency over time. It's opened up a whole new avenue of research for museum collections, a new way of studying evolutionary history," he says.

And while these cautious researchers may never inspire a Michael Crichton novel or Steven Spielberg film, they are already in a position to do what the dinosaur DNA hunters are still struggling to accomplish: get hard data and replicate it.

–Virginia Morell

COMPUTER SCIENCE

Electronic Time-Stamping: The Notary Public Goes Digital

You've just discovered the secret of cold fusion. You're not ready to go public, but you'd like to establish intellectual (not to mention patent) priority for your inspired idea. Down the hall, your colleague in the biology department is worried about fending off another congressional investigation into whether or not one of his postdocs has altered lab notes, which would be easy to do, since the notes exist solely as computer files. Across town, a surgeon is "revising" a patient's records in the hospital's computer system after a botched operation. Meanwhile, in the high-rise offices of Shady Deals Investment Corp., a ruthless corporate climber is backdating a memo to prove he warned his boss against the company's latest disastrous venture.

The paperless world in which all these things could be happening at once in a medium-sized town isn't far off—and it's getting closer every day. Along with it comes the question: How do you establish trust in documents that exist only in the easily altered memory of a computer? So far the answer has been to make paper printouts of crucial documents, which can't be altered easily without leaving traces. But two researchers at Bell Communications Research (Bellcore) in Morristown, New Jersey, have proposed what they say is a more elegant solution based on mathematical ideas in computer science.

Digital time-stamping, as co-inventors Stuart Haber and Scott Stornetta call their approach, makes it possible to prove that a particular document existed at a particular time in a specific form, without requiring the document to exist in hard copy. How is this trick pulled off? The answer is that instead of authenticating a piece of paper or a magnetic tape, the new scheme creates a time-stamp from the data themselves. "A document with its digital time-stamp not only shows when the thing was created, but also assures anybody looking at it that the document hasn't been changed since then," says Haber. And that may be just what's needed in the increasingly electronic world of finance, insurance, and scholarship. Says Mack Hicks, a vice president for data processing in the technology division of BankAmerica, "This is the first technology I've seen for notarizing electronic documents."

Haber and Stornetta's interest in time-stamping was prompted partly by the much-publicized, lengthy scientific fraud case over a 1986 paper published in *Cell*. During that case, Secret Service analysts testified that data in the lab notebooks of Tufts University immunologist Thereza Imanishi-Kari had been altered. Haber and Stornetta realized that if it had happened today, "she'd probably be keeping all the data in computer files, and there wouldn't be anything on paper for the Secret Service to come back and look at," says Haber. The two wondered whether it's possible to devise a digital stamp that, like an old-fashioned seal on an envelope, would reveal if an electronic document had been tampered with since the seal was applied.

Like a wax seal, the stamp had to do more than give a time; it also had to certify the document's content. The answer, Haber and Stornetta decided, lay in a mathematical procedure called a one-way hash function. One-way hash functions, of which there are many types, are procedures for taking long strings of characters (which is what all documents in a computer amount to) and boiling them down to shorter, random looking character strings. A hash value contains no clue to its input; publishing or storing a hash value at the time a document is created gives away no secrets. At the same time, however, each document's hash value is for all practical purposes unique, like a human fingerprint; alter a document by even one character, and its hash value changes completely.

If all of this seems a bit abstract, try an elementary example. A simple hash function that converts the word *science* into a string of 6 digits might work this way: First, convert the letters into 2-digit numbers, where a=01, b=02, and so on. Then add a digit to describe the position of each letter, turning *science* into the string 119, 203, 309, 405, 514, 603, 705. Next, square the numbers and add them together: $119^2 + 203^2 + 309^2 + 405^2 + 514^2 + 603^2 + 705^2 = 1439706$. Finally, keep only the last 6 digits, yielding a hash value equal to 439706. Clearly, changing even a single letter would change the hash value.

One way to time-stamp a document, Haber and Stornetta realized, would be to send its hash-value fingerprint to a central time-stamp service (say Bellcore), which would attach the time of arrival and then put both in permanent storage. Any question about a document's date or integrity could be settled by checking with the time-stamp service. But that solution was unsatisfactory, Haber and Stornetta felt, mainly because it required the time-stamp service to be absolutely trustworthy. With the connivance of the time-stamp service, after all, a customer could easily alter a document, re-

compute its hash value, and file the new value with the original time record.

What Haber and Stornetta wanted was a system that didn't rely on trust at all. Their original solution: Attach a copy of each document's hash value and time-stamp to the next document that's submitted for time-stamping. Each document's time-stamp will then affect the time-stamp of the next document, linking all of the time-stamps together and preventing anyone—including the time-stamp service itself—from slipping in a phony, backdated document later on. In effect, the time-stamp service creates a computationally unbreakable chain that can only be added to at one end.

Haber and Stornetta "time-stamped" their idea in the traditional way: by publishing it. Their first papers appeared in the proceedings of a cryptology conference in 1990 and in the *Journal of Cryptology* in 1991, and since then they and Bellcore have been issued two patents. More recently they have refined the method to make it more efficient and reliable. This approach, developed with Dave Bayer at Barnard College, blends all the time-stamp requests for a given time period together into a kind of tree, which is more compact than a chain. Each party is issued a "time-stamp certificate" to prove its document took part in the blending, and the "root" of the tree is printed in a newspaper, as a public record of when the documents in the tree were created (see sidebar).

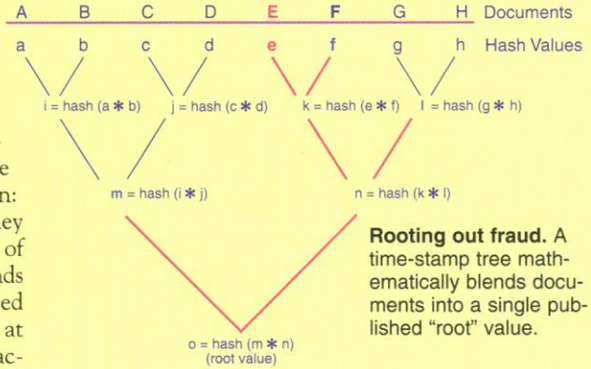
Bellcore is now experimenting internally with this time-stamp system. Haber and Stornetta have also been talking with potential customers. One expression of interest has come from Hicks at BankAmerica, who thinks time-stamping could eliminate the need to print paper copies of faxed orders for the transfer of funds. These orders involve "a lot of money and a lot of transactions," Hicks says, adding that banks currently "are just swimming in paper." Also intrigued is Peter Graham, a librarian at Rutgers University. Without some form of certification it's impossible for a scholar to know whether a document that flashes onto a computer screen is exactly the original or some altered form of it, Graham explains. "With a book we take it for granted [that the text is authentic]. With an electronic text we don't."

To Stornetta, time-stamping should become as routine with electronic documents as making a backup file is now. With a combination of scientific and entrepreneurial excitement, he asks, "Isn't this what we should be doing with all the world's data?"

–Barry Cipra

All the Hash That's Fit to Print

"Time-stamping" data to guarantee their authenticity and prevent fraud is an idea whose time is coming (see main story). An early sign can be found tucked away each Sunday among the Public and Commercial Notices in the inside back page of *The New York Times* Metro section: three or four lines that look like they were typeset by the proverbial team of monkeys. The gibberish actually stands for hundreds of documents generated that week in an experimental effort at Bellcore. Publishing this single character string, the researchers believe, can serve the same function as signing the documents in ink, notarizing them, and "securing" them in safe deposit boxes. If any of the documents identified by the gibberish were to be altered and then backdated, the string of nonsense would, in theory, reveal the breach.



Rooting out fraud. A time-stamp tree mathematically blends documents into a single published "root" value.


13:45 6/11/93 Bellcore TimeLink Digital Time-Stamp Markers: 30//2.v2021022/.C/F5/5/1 G99345LD22D0g97JB9K45U/B f18O/1P87z5Bz20W91oD5s5A These markers prevent computer document tampering. 445 South St, Morristown, NJ, 07960, Rm. 2Q-346.

ALL WHO KNEW

The mysterious code in the *Times* is the "root" of a tree of digital time-stamps—a system developed by Stuart Haber and Scott Stornetta at Bellcore as a fraud-proof way of establishing the integrity of computer files for everything from financial transactions to laboratory notes. On each of the tree's uppermost branches is a hash value—a string of characters that provides a unique fingerprint of a document, computed by running the much longer digital string of the document through a "one-way hash function." By mathematically combining each hash value on the tree with neighboring ones to produce the root value that appears in the *Times*, the Bellcore workers create what they hope is an insurmountable obstacle to altering a document after the tree has been produced.

In simplified outline, here's how the time-stamp tree works: Eight customers submit hash values *a*-*h* of their documents A-H (*a*=hash(A), *b*=hash(B), etc.). The time-stamp service pairs these values and computes hash values. It then pairs *those* values, and so on until it ends up with a single hash value, which is what appears in the Sunday *Times*. Each customer is then sent a "time-stamp certificate" consisting of the hash values that were blended with the hash value of his or her document to compute the published root. The certificate for document E, for example, includes the hash values *m*, *f*, and *l*. The same idea works if there are a million documents in the tree instead of just eight, except that each customer's certificate contains approximately 20 hash values instead of just 3.

To backdate an altered document, a forger would have to create a phony time-stamp certificate listing values that, when blended with the hash value for the document itself, yield the root published on the desired day. But hash functions are "one-way"—a hash value contains no clue to its input. If all you have is a hash value, essentially the only way to find its input is to look at all possible inputs, apply the hash function to each, and keep going till you duplicate the given value—in practical terms, an impossible task.

It would be easier for a would-be forger to track down every last copy of the *Times* and change the published root—but, of course, that would raise a cry among all the other participants in the tree. Likewise, any mistake made by the time-stamp service (or by the *Times*'s typesetter) would be recognized immediately by users checking their certificates. In effect, the integrity of the time-stamp service is regulated by the individual self-interest of the participants. Jokes Haber: "The service could be run by the Mafia, and it would still be worth trusting!"

True, what's computationally daunting today may be child's play tomorrow. As soon as computer power, or some programmer's cleverness, catches up with a particular hash function, all the time-stamps issued using that function become worthless. But Haber and Stornetta think that by periodically restamping documents and their certificates with new hash functions, they have a good chance of staying ahead of the challenges.

–B.C.