

# THE TRUSTWORTHY DIGITAL CAMERA: RESTORING CREDIBILITY TO THE PHOTOGRAPHIC IMAGE

Gary L. Friedman  
Technical Group Leader  
Advanced Information Systems  
Jet Propulsion Laboratory  
California Institute of Technology  
Pasadena, CA 91109

## Introduction

The increasing sophistication of computers has made digital manipulation of photographic images (as well as other digitally-recorded artifacts, such as audio and video) incredibly easy to perform and, as time goes on, increasingly difficult to detect. Today, the majority of pictures appearing in newspapers and magazines have been digitally altered to some degree, and the prospect of an individual manipulating images on a computer and presenting them as courtroom evidence is upon us. [1],[2],[10],[11] As the power, flexibility and ubiquity of image-altering computers continues to increase, the well-known adage that "the photograph doesn't lie" will continue to become an anachronism.

A solution to this problem comes from a concept called Digital Signatures, which incorporates modern cryptographic techniques to authenticate electronic mail messages. [3],[4] ("Authenticate" in this case means you can be sure that the message has not been altered, and that the sender's identity has not been forged.) The technique can serve not only to authenticate images, but also to help the photographer retain and enforce copyright protection when the concept of "electronic original" is no longer meaningful.

## Background on Digital Signatures

The concept of a digital signature builds upon a recent encryption technique called "Public Key Encryption" [5]. Older encryption/decryption schemes require that both the sender and receiver possess the same secret "key": the sender uses the key to transform the text message into ciphertext, and the receiver uses the same key to perform an inverse transformation on the ciphertext, revealing the original text message. If the correct key transforms the ciphertext into unreadable garbage, it is reasonable to conclude that either the wrong key is being used, the message has been altered, or the sender has been impersonated by someone ignorant of the correct key. The historic drawback to this secret key encryption scheme has been in the secure distribution of keys; key disclosure must occur out-of-

band, either transmitted via an expensive alternate path or arranged when sender and receiver were proximate.

Public key encryption techniques differ in that they enable the recipient of a message to decrypt it using a key that is different from the one used by the sender to encrypt it. All public key cryptography is based on the principle that it is easy to multiply two large prime numbers together, but extremely difficult (taking perhaps centuries using today's supercomputers) to work backwards and uncover the factors that could have been used to generate the resulting number. (Alternatively, some encryption schemes rely on the difficulty of solving discrete logarithms for their security.)

Public Key Encryption employs two different keys: a private key, which is held by the more security conscious party, and a corresponding public key, which need not be kept secret. The public key is generated based upon the private key, making the pair unique to each other.

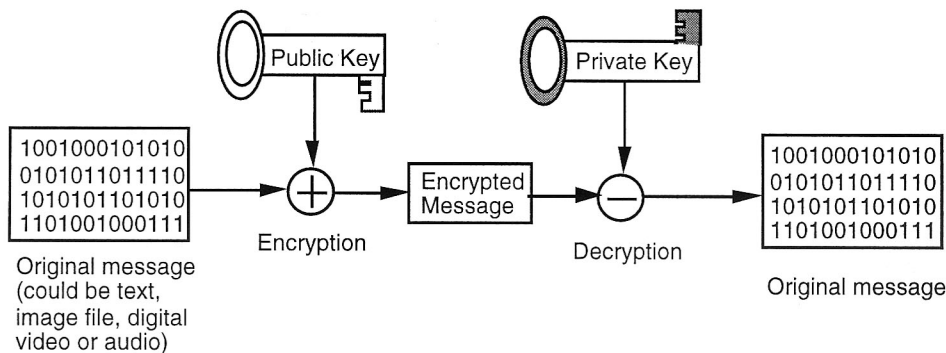
The public key scheme is illustrated in Figure 1 and works as follows: to send a secret message that only the recipient can read, the recipient would first make his/her public key known to the sender through any non-secure medium, such as a letter, a telephone conversation, or a newspaper ad. Anyone wishing to send a secure message would encrypt the message using this public key and send it to the recipient. The recipient, having sole possession of the corresponding private key, is the only one able to decrypt the message. The need to transmit a secret key that both parties must possess beforehand has been eliminated. The tradeoff in this case is that, although only the recipient can read the message, anyone who obtains the public key can send a message with anonymity.<sup>1</sup>

The process described above can also be implemented "backwards" to great advantage. In a second scenario, it

---

<sup>1</sup> The described scenario can also be used as the first step in a process of exchanging secret keys to allow for conventional secure message transmission, eliminating any of the drawbacks of the one-way authenticatability. [3],[6]

is the sender who maintains possession of the private key, and anyone who has the widely disseminated corresponding public key could decrypt this message.



**Figure 1:** With Public Key Encryption, the encoding (public) key and decoding (private) keys are different, and it is computationally difficult to derive one given the other. To send a message that only the receiver can read, the non-secret public key is used to encrypt; the secret private key is used to decrypt. Encrypting with the private key forgoes confidentiality in favor of authenticity: if the public key can decode it, then only the one holding the private key could have generated the message.

Although this procedure no longer performs the traditional function of encryption (which is to provide confidential communication between two parties), it does provide a way to insure that messages have not been forged: only the private key could have produced a message that is decipherable by the corresponding public key.

This gives us the foundation for message authentication: if the private key remains private, then *only* the private key holder can produce messages decipherable by the public key. Furthermore, it is extremely difficult to reverse-engineer the public key and ascertain the original private key. Without knowledge of the private key, a counterfeit message cannot be forged.

Digital signatures build upon these public key cryptographic techniques and allow you to authenticate the contents of the message as well as the identity of the sender, without obscuring the original message. The signatures are produced by creating a *hash*<sup>2</sup> of the

<sup>2</sup>A hash is a mathematical function which maps values from a large domain into a smaller range. For example, a checksum is a simple kind of hash. A more complex example of a hash algorithm would involve dividing a binary file into a collection of, say, 16 Kilobit pieces and performing a cumulative Exclusive OR function between successive pieces, producing a simple 16 Kilobit "hash" that is smaller than the original file yet is practically unique to it. (Many more complex and secure transformations are also possible.) Changing a single bit in the original message produces a very different hash output; and reverse engineering a message so it will have a given hash value and also make sense to the reader is virtually impossible. A

original plaintext message, and then encrypting the hash using the sender's private key, as shown in Figure 2. The result is a second digital file (referred to as a *signature*) which accompanies the original plaintext message. To emphasize: THE ORIGINAL MESSAGE IS UNTOUCHED; only the message's hash is encrypted. This way the original file can be read by all, yet if you wish to authenticate it you can decrypt the message's unique digital signature using the public key. If the decrypted digital signature and an independent hash on the file in question match, both the integrity of the message and the authenticity of the sender can be assured.

This digital signature technique is very general; it can be applied not only to 1-dimensional symbolic text (such as electronic mail) but also to any n-dimensional digital pattern (such as digital video, digital audio, and/or digital holograms).

### Digital Cameras

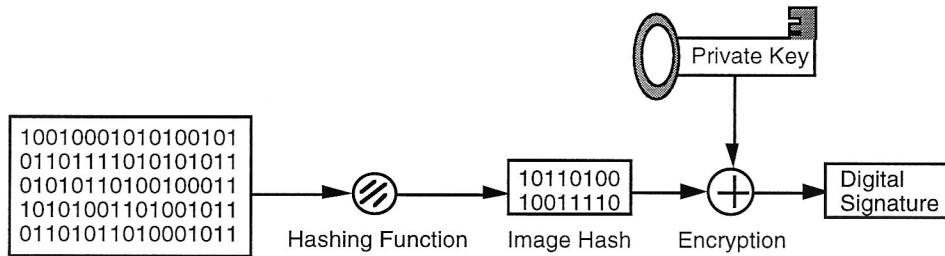
Standard digital cameras are filmless; they sense light and color via an electronic device (such as a Charge Coupled Device (CCD)), and produce as output a computer file that describes the image using 1's and 0's arranged in a meaningful, pre-defined format. Often this digital image file is stored on a small mass-storage medium inside the camera itself (such as floppy disk or magneto-optical disk) for later transference to a large computer. Alternatively, the image file can be sent directly to the computer via a transmission medium. Once inside the computer it then can be read and then easily altered in any number of different ways.

In the proposed digital camera (Figure 3) we wish to authenticate the initial image file as it emerges from the camera. To accomplish this, the camera produces two output files for each captured image as shown in Figure 4: the first is an all-digital industry-standard file format representing the captured image. The second would be an encrypted "digital signature" produced by applying the camera's unique private key (embedded

digital signature can then be created by encrypting the hashing output using the sender's private key.

within the camera's secure microprocessor) to a hash of the captured image file, using the procedure described in [6]. It is the responsibility of the user to keep track of

If the technique is to be effective (i.e., no false positives or false negatives) and extended to larger data sets such as digital audio, digital video or even digital holograms, we must build upon the accomplishments of the computer mass storage industry, which has already achieved the ability to store and deliver extremely large binary data sets without errors.



Original message (could be text, image file, digital video or audio)

**Figure 2:** The Digital Signature is created by producing a complex checksum called a "hash", which is then encrypted using the private key. Attempting to forge this signature without knowledge of the private key would take decades using today's supercomputer technologies.

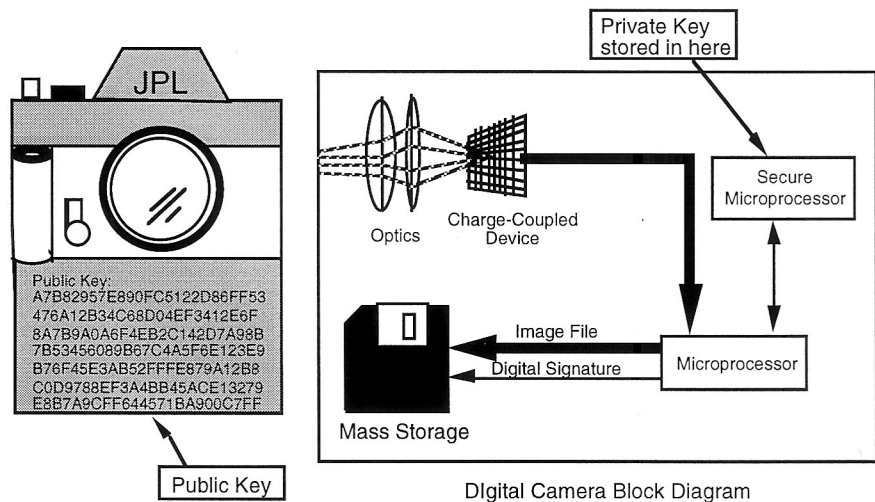
both files once they leave the camera, since both are required to authenticate the image.

Once the digital image file and the digital signature are generated by the camera and stored in computer memory, the image file's integrity can later be affirmed by using a public key decoding program, which can be freely distributed to users and certification authorities via conventional software distribution techniques. This verification program (illustrated in Figure 5), which has no knowledge of either the public or private keys, takes as input the digital image file in question, its accompanying digital signature file, and the public key which is unique to the originating camera. (It is perfectly reasonable to have the public key double as the camera's serial number.) The program then calculates its own hash on the digital image file (the hashing algorithm need not be kept a secret), and uses the public key to decode the digital signature to reveal the hash originally calculated by the camera at the time the image was taken (Figure 6). If these two hashes match, it is certain to any required degree that the digital image in question is indeed identical to what the camera originally produced. If on the other hand at least a single bit is different, the two hashes will not even closely match and the image's integrity will not be affirmed.

Analog techniques (such as audio cassette tape or the NTSC encoding on today's video tape formats) or non-corrected digital formats (such as the popular audio compact disc (CD), which is so unreliable that CD player manufacturers now employ "over-sampling" to combat the problem of missed bits) introduce a large amount of errors upon playback that are normally imperceptible to the viewer or listener, but are intolerable for the purposes of image authentication.

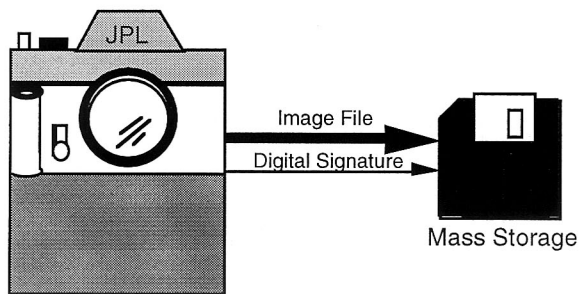
### Measures of Protection

The scheme as described above is resistant to forgery attempts since the secret private key (which is known only by the camera's manufacturer) is embedded in a probe-proof microprocessor which itself is deeply



**Figure 3:** The Trustworthy Digital Camera starts with a digital sensor instead of film, and delivers the image directly in a computer-compatible format. The secure microprocessor responsible for the encryption of the digital signature is programmed with the private key at the factory. The public key necessary for later authentication appears in the image's border as well as on the camera body.

integrated into the camera's system (Figure 3). Even if some adept pirate were to dissect the camera and replace the chip with one containing a homebrew key, the digital signature produced thereafter would not be



**Figure 4:** When a single photo is taken, two files are produced: a standard digital image file, and an encrypted digital signature. The files can be stored on a variety of media, such as a Write-Once-Read-Many (WORM) CD or the computer's mass storage device. The image can then be accessed and used just as any other computer data.

decodable by any public key published by the manufacturer. And for the truly paranoid, the manufacturer could combine the CCD, standard and secure microprocessors onto one substrate and in one integrated circuit package to remove the threat of dissecting the circuit board and feeding digital data directly into the camera.

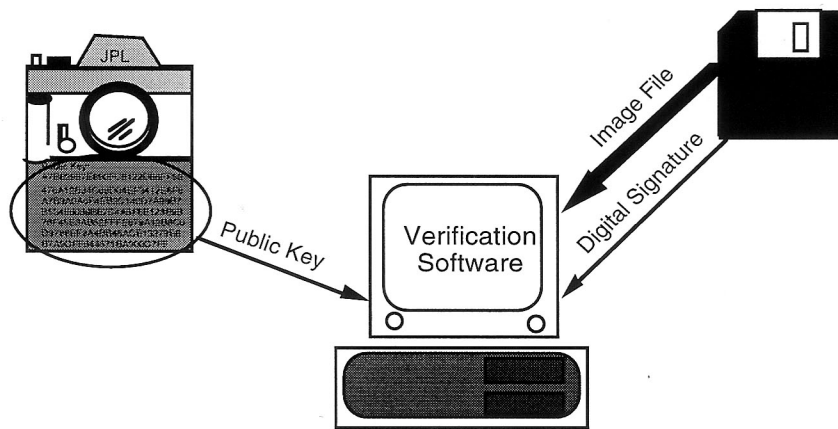
The advantages to freely distributing the verification software and valid public keys are great; with the software freely available verification can become commonplace and routine. No special certification authority need be called in for routine checks, no fees are required, no big fuss is made, and no bad-faith climate amongst the parties involved need be created as a result of being challenged. But the mass distribution of verification software does carry one danger: it would be easy for someone to create a bogus program that looks, behaves, and has the same file length as the genuine verification software, with the only difference being it always proclaims a "match" regardless of the integrity of the image being verified. With the software freely and widely available this is not a large risk, as additional copies can be easily obtained from multiple sources and a best 2-out-of-3 scheme can be employed. When the stakes are high and it is extremely important that the verification software be

known to be genuine, an independent certification authority or the manufacturer could then be called in to provide their own copies of the software and their own lists of valid public keys at the time of verification.

The algorithms and private key necessary for encrypting the additional digital signature file from within the camera are to be embedded inside a new breed of secure microprocessors whose ROM contents cannot be observed outside of the factory. Because the private key used for encryption is hard-coded into this chip by the manufacturer (who must then ensure that the private key remains secret), credibility of the camera's output becomes an extension of that of the manufacturer; a digital signature from the camera can be considered to be just as reliable and secure as if the signature had been generated by the manufacturer.<sup>3</sup>

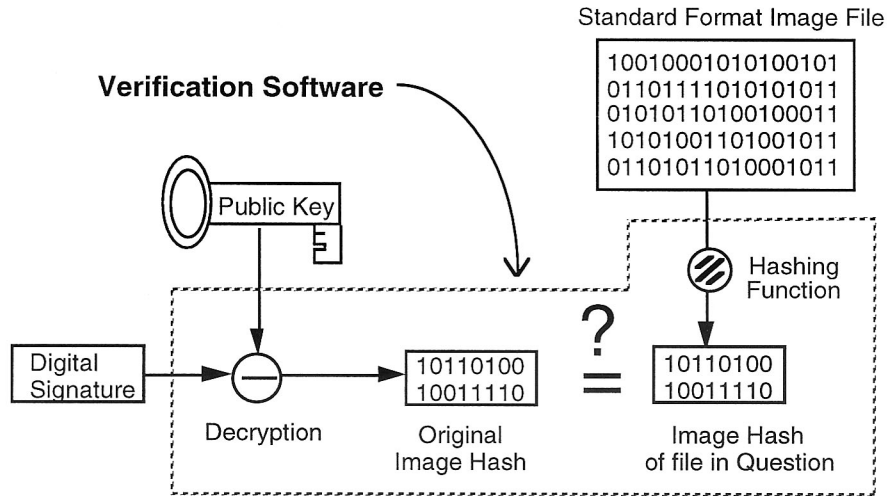
Each camera should possess its own unique pair of private and public keys, with the private key etched into the camera's secure microcontroller and the public key stored in three places: in a public key list

kept by the manufacturer, on the camera body itself (which can then also double as the camera's serial number), and in the colorful border that contains more data about the captured image (see "A Special Border" section below for more details on this idea). Assigning unique keys to each camera has the benefit of avoiding



**Figure 5:** To authenticate the image, public domain verification software is run on a standard computer platform. The program takes as input the image file in question, the digital signature, and the camera's serial number (which doubles as its public key).

<sup>3</sup> Any company involved with the development of a Trustworthy Digital Camera would have to address the issue of liability, for if the security of the private key were ever to be compromised (for example by a disgruntled employee who steals a private key and uses it to generate false authenticatable images), the lawsuits brought on as the result of a false positive would necessitate significant insurance coverage.



**Figure 6:** The verification software computes its own hash of the image in question, and compares it to the original hash which has been decrypted using the public key. If the image in question has not been manipulated, the decrypted digital signature and the program's own hashing function will match, resulting in an authentication. If even a single bit is different, the two hashes will not even closely match, yielding an authentication failure.

instant obsolescence which would occur if only one private key were used for all cameras, and that key were to be compromised. An even higher level of security would occur if the manufacturer were to destroy all records of the private key once the camera is produced. (At that point the private key is no longer needed.) This would eliminate the possibility of compromise via industrial espionage or theft.

Finally, regular and free distribution of all valid public keys is desirable to defeat a counterfeiter who has learned of the encryption algorithm employed and has written a program to produce digital signatures based on his own private key. Decoding these forgeries would require the use of a public key not generated by the manufacturer. Freely distributing updated public key lists would make it easy to identify and thwart such attempts.

### Uses

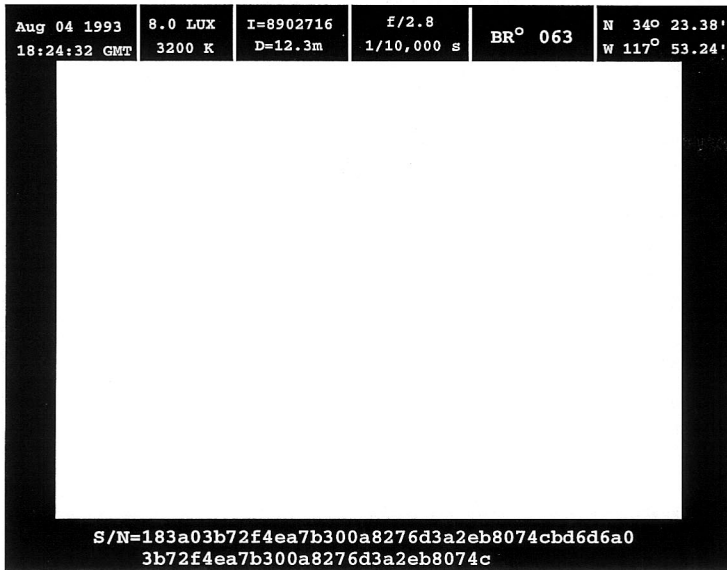
The single most obvious use of a trustworthy camera would be in situations where proof of image authenticity is necessary; such as for legal evidence or insurance claims. The inevitable transition to digital cameras and electronically-transmitted images will also make it more difficult for the professional photographer to protect his or her image copyright, since with electronic cameras there is no "original" to control, and works stored in computer format tend to proliferate faster and with less control from the author than the traditional distribution method (which places image control in the hands of whoever holds the original negative or transparency).

Just as it is common practice today to obtain model releases for any published picture containing a recognizable face, it is foreseeable that no electronic image in the future will be published without first having authenticated the image using the digital signature of the camera which was or is registered to the photographer.

This technique need not be limited to still digital images. Because digital signatures can be used to verify any block of digital data, it can also be engineered into digital video cameras and digital audio tape recorders. In both these devices, a digital signature can be generated and recorded onto the medium each time the recording process stops or pauses; this way each sound byte or video "take" (along with other information, such as date, time, and sequence number) is hashed, encoded and recorded onto the medium at the time it's created.

### A Special Border

Since the proposed camera is being initially targeted towards legal authentication, a few additional features can be implemented to better serve this use. A brightly-colored border could automatically be generated as part of each captured image file (Figure 7). Within the border would appear textual information about the image: the date and time it was taken, the ambient light level seen by the camera at the time of exposure, the original color temperature of the scene, the lens' aperture and sampling speed (equivalent to a film-based camera's shutter speed), the software version of the camera's firmware, the camera's serial number (or a



**Figure 7:** A textural border would be part of each image, containing information about the time, position, and environment at the time the image was taken. Because it is part of the picture, authenticating the image file will also authenticate the accompanying information.

barcoded representation thereof), the focusing distance of the lens at the time of exposure, a unique sequence number, the direction in which the camera was pointing, and (when the technology allows for a Global Positioning System (GPS) receiver to be built into the camera) the geographical coordinates of the camera, indicating where in the world you were when the picture was taken. The ambient light level and color temperature readings would be useful for getting a feel for exactly what the scene was like at the time of exposure; something a sensitive optical element might inadvertently hide via automatic exposure and color correction. The lens' focused distance is there to help detect potential abuse of the trustworthy camera: taking close-up pictures of a modified photo and trying to pass it off as an unaltered original. Since all these textual data in the colored border are part of the authenticated image file, their credibility is also upheld when authenticated by the verification software.

The accuracy of the date and time information would again be the responsibility of the secure microprocessor; in addition to being able to keep its programming a secret, it also would have a lithium battery powering a system clock that was set to Universal (Greenwich Mean) Time at the time of manufacture. If the timer circuit ever fails or is tampered with, the system will be programmed to detect the inconsistencies and fill the time and date fields with XXXX's, eliminating the chance of a random time stamp being mistaken for the actual time.

## Higher Level of Security

Although the proposed Trustworthy Digital Camera offers a satisfactory level of security, nevertheless there still exists a small possibility that a determined saboteur will be able to crack the camera's private key given an extended amount of time. (No cryptographic scheme will protect your data forever; given sufficient time, advancements in code breaking or improved computer horsepower will be enough to render any given level of cryptographic protection obsolete.) If the discovered private key were then to be published, it would allow an individual to generate authentic-looking digital signatures on altered image files, essentially undermining the credibility offered by the compromised camera. (The security level of other cameras in use, and of images taken with those cameras, will be unaffected.)

Because of this risk, it would be wise for a manufacturer of such cameras to regularly upgrade and enhance the sophistication of the encryption implementation as newer camera models are introduced, typically using longer encryption/decryption key lengths and improved encryption/decryption algorithms. It is expected that evolving verification software (the public domain software component of this authentication scheme which is freely distributed) will then be designed to recognize, identify and authenticate all previous versions.

Because the encryption details must necessarily be changed often (depending on the technological capabilities of the day), no single image format, key length or digital signature algorithm is being specified in this disclosure.

## Conclusion

The Trustworthy Digital Camera is an application of existing technology toward the solution of an ever-more-troubling social problem, the eroding credibility of the photographic image. Although it will always be possible to lie with a photograph (using such time-honored techniques as false perspective and misleading captions), this proposed device will prevent the explosion of very capable personal computers from creating a plethora of doctored photographs that are passed off as truth.

## Bibliography

- [1] "When Is Seeing Believing?: Digital technology for manipulating images has subverted the certainty of photographic evidence", Mitchell, W., *Scientific American*, February, 1994, pp. 68-73.
- [2] Cantor, Ben, "Changing Times: How Will Computers Impact Forensic Photography?", *Photo>Electronic Imaging*, February, 1993
- [3] "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithm", *IEEE Transactions on Information Theory*, Vol. IT-31 no. 4, (July 1985), Taher ElGamal, pp. 473-81.
- [4] "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", *Communications of the ACM*, Vol. 21 no. 2 (February, 1978), R.L. Rivest, A. Shamir, L. Adleman, pp. 120-26.
- [5] "New Directions in Cryptography", *IEEE Transactions on Information Theory*, Vol. IT-22 no. 6 (November, 1976), Whitfield Diffie and Martin E. Hellman, pp. 644-54.
- [6] "The Proposal for a U.S. Standard for Digital Signature Encoding", *IEEE Spectrum*, August, 1992, Dennis K. Branstad, pp. 30.
- [7] "8051 Tackles Secure Smart-card Applications", *EDN*, December 10, 1992, pp. 46-48.
- [8] "Photography by the Numbers", *BYTE*, January 1993, Howard Eglowstein, pp. 241-244.
- [9] "Electronic Copyright Protection," *Photo>Electronic Imaging*, Sapwater, E., and Wood, K., V37n6, 1994, p. 18.
- [10] "Ask It No Questions; The Photo Can Lie", Grundberb, A., *The New York Times*, August 12, 1990 Section 2, pp. 1, 29
- [11] "Alterations of Photos Raise Host of Legal, Ethical Issues", Ansberry, C., *The Wall Street Journal*, Jan 26, 1989, Section B, p. 1.

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

First revision: December, 1992.  
Paper last revised Nov. 30, 1994