Cryptography for the Internet

E-mail and other information sent electronically are like digital postcards—they afford little privacy. Well-designed cryptography systems can ensure the secrecy of such transmissions

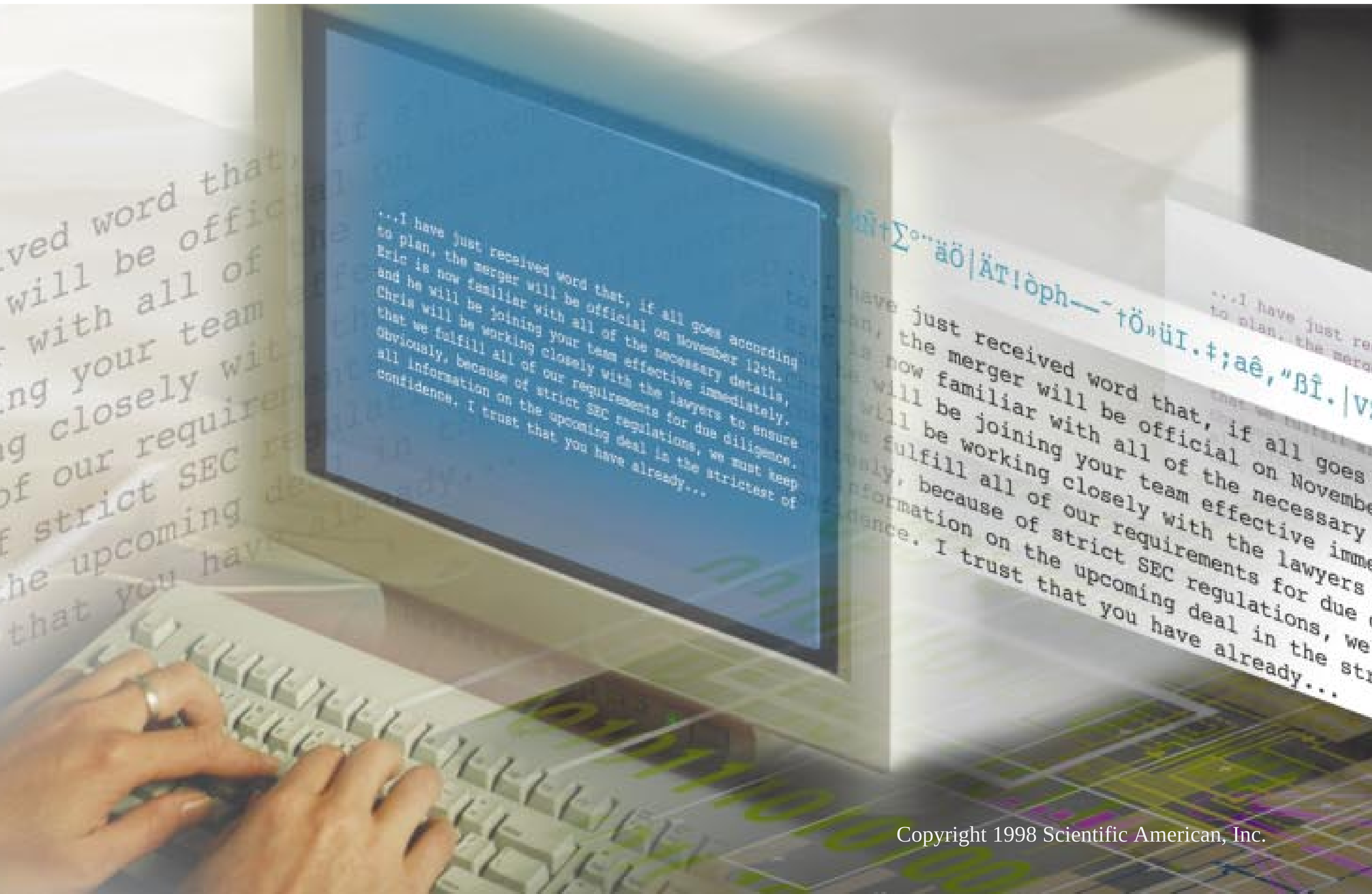
by Philip R. Zimmermann

Sending letters through the post office might take days, but at least the correspondence is guaranteed some degree of privacy. E-mail delivered over the Internet, on the other hand, can be blindingly fast but is highly susceptible to electronic eavesdroppers. One way to increase the privacy of such transmissions is to encrypt them, scrambling the information in complex ways to render it unintelligible to anyone but the intended recipient.

Since the 1980s the development of sophisticated algorithms and fast but affordable computer hardware have made powerful, military-grade cryptographic systems available to millions of people with ordinary personal computers. Recent technological improvements promise to make such systems increasingly resistant to even the most advanced cipher-cracking techniques.

Out from the Shadows

Four decades ago the Pentagon's requirements for tiny custom circuits to fit into missiles and spacecraft were the driving force behind the U.S. electronics industry. Today civilian demands dominate, and the military currently


...I have just received word that, if all goes according to plan, the merger will be official on November 12th. Eric is now familiar with all of the necessary details, and he will be joining your team effective immediately. Chris will be working closely with the lawyers to ensure that we fulfill all of our requirements for due diligence. Obviously, because of strict SEC regulations, we must keep all information on the upcoming deal in the strictest of confidence. I trust that you have already...

satisfies most of its needs with off-the-shelf products designed for the much larger consumer market. The same thing is happening with cryptography.

Until the mid-1970s the National Security Agency (NSA) had a virtual monopoly on U.S. encryption technology, a field that was kept shrouded in secrecy. Then, in 1976, the seminal article "New Directions in Cryptography," in which Whitfield Diffie and Martin E. Hellman of Stanford University first described "public-key cryptography" in the open literature, forever changed the landscape. In the years since that publication, an energetic cryptographic community in academia and industry has emerged, publishing an ever increasing number of papers and building a mature discipline. The growing popularity of the Internet—and people's concerns about the privacy of that medium—has only intensified the trend. Today some of the best ciphers and systems are being developed by cryptographers at universities and in the private sector all over the world. In fact, the NSA is now beginning to buy commercial products for a portion of its cryptographic needs.

Why was Diffie and Hellman's introduction of public-key cryptography so crucial? In conventional cryptosystems, a single key is used for both encryption and decryption. Such systems, called symmetric, require the key to be transmitted over a secure channel—a process that is often inconvenient. After all, if a secure channel exists, why is encryption needed in the first place? This limitation hobbled cryptography.

Diffie and Hellman removed that constraint. Public-key cryptography allows the participants to communicate without requiring a secret means of delivering the keys. Such asymmetric systems rely on a pair of keys that are different but complementary. Each key unlocks the message that the other key encrypts, but the process is not reversible: the key used to encrypt a message cannot be used to decrypt it. Thus, one of the complementary keys (public) can be disseminated widely, whereas the other key (private) is held only by its owner. When Bob wants to send a message to Alice, he can use her public key to encrypt the information, which she will then use her private key to decrypt.

Public-key cryptosystems are based on mathematical problems that are easy to compute in one direction but painfully slow to solve in the reverse. The two main public-key algorithms are the Diffie-Hellman (and its variants, such as the Digital Signature Standard from the National Institute of Standards and Technology, ElGamal and elliptic curve approaches) and RSA, developed at the Massachusetts Institute of Technology by computer scientists Ronald L. Rivest, Adi Shamir and Leonard M. Adleman.

ENCRYPTING A PRIVATE MESSAGE that Bob will send to Alice over the Internet requires several steps. In this conceptual schematic, Bob first computes a hash of the text [*see diagram on page 113*]. He then encrypts the hash using his private key [*see box on next page*]. The resulting information (*blue, below*) serves as Bob's "signature." Bob compresses the signature and his message electronically (*purple*) and enciphers the file (*green*) using a particular session key. Bob encrypts this key using Alice's public key, and the result (*orange*) is added to the message. Finally, the file is converted into alphanumeric characters (*red*) for transmission over the Internet. At the receiving end, the steps are essentially reversed, with Alice using her private key to decrypt the session key, which she can then use to decipher the rest of the message.

The former approach uses discrete logarithms. It is simple to compute g^x modulo p: just raise g to the x power, divide that quantity by a large prime number p, and then take the remainder of that operation. But given g, p and the value of g^x modulo p, it is infeasible to recover x [see "The Mathematics of Public-Key Cryptography," by Martin E. Hellman; SCIENTIFIC AMERICAN, August 1979].

The RSA system is based on the difficulty of factoring. It is straightforward to multiply two large prime numbers together, but it is extremely difficult to factor that huge composite back into its two primes [see "Mathematical Games," by Martin Gardner; SCIENTIFIC AMERICAN, August 1977].

Another beauty of public-key cryptography is that it can be used for message authentication: a recipient can verify the identity of the sender. When Bob transmits a message to Alice, he first encrypts it with his private key, then reencrypts the encrypted message with Alice's public key. Alice, after receiving the transmission, reverses the steps. She first decrypts the message with her own private key, then decrypts it again with Bob's public key. If the final text is legible, Alice can be confident that Bob actually wrote the message.

Of course, all this encrypting and decrypting requires myriad mathematical calculations. But software applications, such as PGP, running on PCs can automate the process. Using one of those packages, Bob and Alice need only press the "encrypt" and "decrypt" buttons on their computers, and the number crunching is performed behind the scenes.

For all its innovation, public-key cryptography has two severe limitations. First, because of its relatively slow speed, the technology is impractical for encrypting large messages. Second, and perhaps more important, public-key cryptography sometimes allows patterns in a message to survive the encryption process. The patterns are thus detectable in the enciphered text, making the technology vulnerable to cryptanalysis. (Cryptography is the science of making ciphers, cryptanalysis is the study of breaking them, and cryptology is both disciplines.)

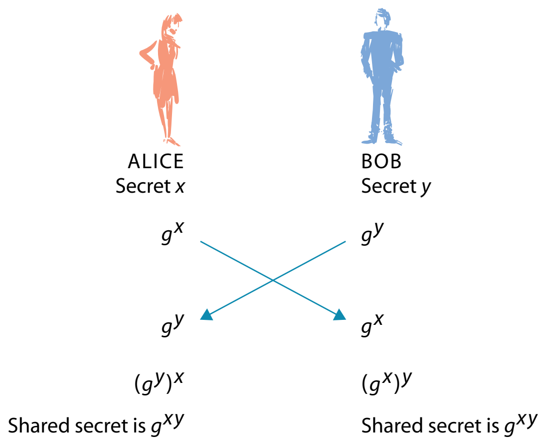
Symmetric Workhorses

Consequently, the bulk of encryption is usually performed by faster and more secure symmetric ciphers, with public-key cryptography limited to the small—but essential—function of exchanging the symmetric keys. Specifically, Bob encrypts his message with a quick and strong symmetric cipher. He then needs to send Alice the symmetric key that he used, so he enciphers it with her public key and attaches the result to his encrypted message. Alice will decrypt the symmetric key with her private key so that she can use that information to decrypt the rest of Bob's message.

For authentication, Bob again does not use public-key cryptography to "sign" his transmission directly. Instead he computes a hash, or fingerprint, of his message. Such mathematical procedures can be used to condense an input of any size into a digest of fixed length, typically 160 bits long. (A bit is the most basic unit of computer data. It stores one of two possible states, represented by 0 or 1.) Cryptographically strong hash functions, such as SHA-1, RIPEMD-160 and MD5, are designed so that a forger would find it computationally infeasible to devise a different message that would yield the same hash. In other words, the fingerprints generated are virtually unique: two different messages will almost certainly yield distinct digests.

After computing a hash of his message, Bob encrypts that information with his private key. He sends this "signature" with the rest of his encrypted transmission. Alice receives the encrypted hash and decrypts it with Bob's public key. She can then compare the result with the hash she computes herself after decrypting the message. A match proves both that the transmission has not been tampered with and that Bob is the sender.

For encrypting such information to be sent over the Internet, the most common method is to break the data into fixed-size blocks, each usually 64 or 128 bits long, so that the encryption can be performed a chunk at a time. So-



Public-Key Cryptography

For centuries, cryptography was hampered by the so-called key-exchange problem. Specifically, if Bob wanted to send Alice an enciphered message, he also somehow had to transmit to her the secret encryption key that he had used. Public-key cryptosystems overcame this limitation by relying on clever mathematics.

In the Diffie-Hellman algorithm, which helped to spawn the field of public-key cryptography, Alice uses her secret number x to calculate g^x and sends that quantity to Bob. On his end, Bob uses his secret number y to compute g^y and sends that to Alice. (Note that the value of g is publicly known.) After Alice receives this information, she can then compute $(g^y)^x$, which is equal to $(g^x)^y$, the value that Bob calculates. This quantity can become their shared, secret encryption key.

But someone who has intercepted Alice's g^x and Bob's g^y would be able to derive the secret x and y. So to thwart any eavesdroppers, Alice and Bob insert the modulo function, which calls for the remainder from a division operation. (For example, 14 modulo 4 = 2 because the remainder of 14 divided by 4 is 2.) This added twist ensures secrecy—instead of sending g^x to Bob, Alice transmits the value of g^x modulo p, from which eavesdroppers would have great difficulty in recovering x, even if they know g and p.

With additional mathematics, the Diffie-Hellman algorithm has evolved into cryptosystems that generate two complementary keys, one private (for Alice, x) and the other public (consisting of g, p and the value of g^x modulo p). Ingeniously, the private key deciphers the message that was enciphered by the public key, but the key used to encrypt a message cannot be used to decrypt it. Thus, Bob can use Alice's public key (which she has disseminated to everyone) to encrypt a message to her, which she—and only she—can decrypt using her private key. *—P.R.Z.*

ILLUSTRATIONS BY DMITRY KRASNY

called block ciphers usually encrypt each chunk using multiple rounds (the exact number is dictated by the particular algorithm) of mathematical operations, with the output of one iteration fed as input to the next. Each round often involves both permutation (shuffling "xtv" to "tvx") and substitution (changing "tvx" to "cb2"). A section of the key helps to transform the data during the iterations.

Feeding identical chunks of text to a block cipher will lead to encrypted blocks that are identical to each other. To suppress any such block-aligned patterns from forming (which would make the cipher easier to crack), block algorithms typically use some kind of chaining. Blocks that have already been encrypted are looped back to help encrypt subsequent chunks. In effect, the encryption of a block of text depends on *all* the previous blocks.

Block ciphers have symmetric keys that are usually 56, 128 or 256 bits long. Well-known examples are the Data Encryption Standard (DES), triple-DES, CAST, IDEA and Skipjack. The workhorses of cryptography, block algorithms have become the focus of much recent research.

The Key Is the Key

The most sensitive operation in cryptography is the generation of keys. For a system to be as secure as possible, the keys should be numbers that are truly random, unpredictable by an attacker. Such numbers are different from the deterministic pseudorandom sequences that computers generate algorithmically for games and simulations. Truly random numbers can be derived only from the environmental "noise" of the physical world, such as the process of radioactive decay.

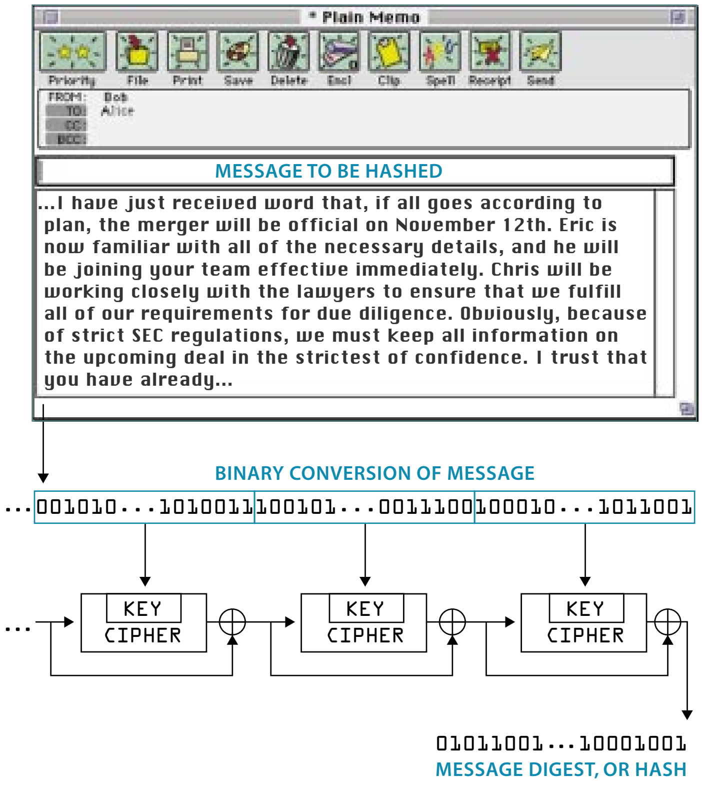
Such high-quality randomness is difficult to generate in a computer. One method is to measure the time, in microseconds, between each human-supplied keystroke, which is impossible to predict. Data gathered in this way are not quite random enough for generating keys directly, but the information can be passed through a hash function to distill the disorder.

Interestingly, the only cipher that cryptologists have ever proved to be perfectly secure is the one-time pad (OTP), in which the key is as long as the message itself. In an OTP, a random sequence is used to encipher a message bit for bit—that is, the 34th bit of the key is used to alter the 34th bit of the message. The key must be truly random. It cannot be a pseudorandom sequence produced by a deterministic algorithm; otherwise the cipher may be crackable. OTPs are rarely used because of their impracticality: the key must be as long as the message, and it must be sent to the receiver over a secure channel. Moreover, it can be used only once, or an attacker could break the messages.

Although many people think key size is the determining factor in cryptographic strength, an equally important criterion is the quality of the cipher's design. Consider a simple substitution cipher in which all *A*s are changed to *W*s, all *B*s turned into *K*s, all *C*s transformed to *Q*s, and so on. The number of different ways to rearrange the alphabet is given by 26 factorial (that is, 26 × 25 × 24 × … 3 × 2 × 1). That quantity is roughly equivalent to 2^{88}, a "key space" of different combinations that is regarded as fairly respectable, requiring enormous computing resources to break if every possible key must be tried. Yet when I was a kid I would crack this type of cryptogram all the time with no more than a pencil and paper. I simply looked for the most common letter and assumed it was probably *E* and then found the second most common letter and assigned *T* to it, and so on. Clearly, despite its vast key space, this type of cipher is very weak.

For a well-designed cryptography system, though, the key size does relate directly to the effort required to crack it. For block ciphers, the relation is usually exponential. Adding just one bit to the key length doubles the work the attacker must do to try all the keys. And doubling the key size squares the amount of effort. On average, a 128-bit key requires about 2^{127} (in decimal, 1.7×10^{38}) operations to break.

Public-key algorithms are less sensitive. Typically, they have subexponential but superpolynomial key spaces, which means that doubling the length of the key increases the work substantially, but the amount is less than a squaring of the work effort. To use RSA as an example, modern factoring algorithms can do much better than simply trying all the possible smaller prime numbers to factor a large composite. Diffie-Hellman is also subexponential. For com-



…I have just received word that, if all goes according to plan, the merger will be official on November 12th. Eric is now familiar with all of the necessary details, and he will be joining your team effective immediately. Chris will be working closely with the lawyers to ensure that we fulfill all of our requirements for due diligence. Obviously, because of strict SEC regulations, we must keep all information on the upcoming deal in the strictest of confidence. I trust that you have already…

HASH ALGORITHM condenses a message into a digest, a digital fingerprint that can be used to detect forgeries. The text of the message is first converted into binary form. (The letter *A* might be represented by 00000, the letter *B* by 00001, the letter *C* by 00010, and so on.) The resulting string of 0s and 1s is then separated into equal-size blocks. Next, the chunks are fed in sequence as key material into a cipher. The final output is the digest, or hash, of the original message. Note that a message of any length will always yield a digest of fixed size. The operation is called "one way" because it is virtually impossible to recover a message from its hash. Also, the algorithm is designed so that any given two messages will almost certainly yield distinct hashes, and it is computationally infeasible to find another message that produces the same hash as a given message. Thus, a digest can serve as a "fingerprint" for its corresponding message.

parison's sake, a 3,000-bit RSA or Diffie-Hellman key requires about the same amount of work to crack as a 128-bit key for a block cipher.

Still, block ciphers are hardly invincible. This year a special-purpose massively parallel machine built for less than $250,000 by the Electronic Frontier Foundation, headquartered in San Francisco, broke a DES message by exhausting its 56-bit key space in less than a week.

Brute force is not the only way to crack a cipher. Cryptanalysts can apply powerful mathematical and statistical tools to find any shortcuts, perhaps by uncovering patterns in the encrypted text. Attempts to break ciphers can be grouped into three categories, depending on how much is known about the original message (called plaintext) and the corresponding enciphered transmission (called ciphertext).

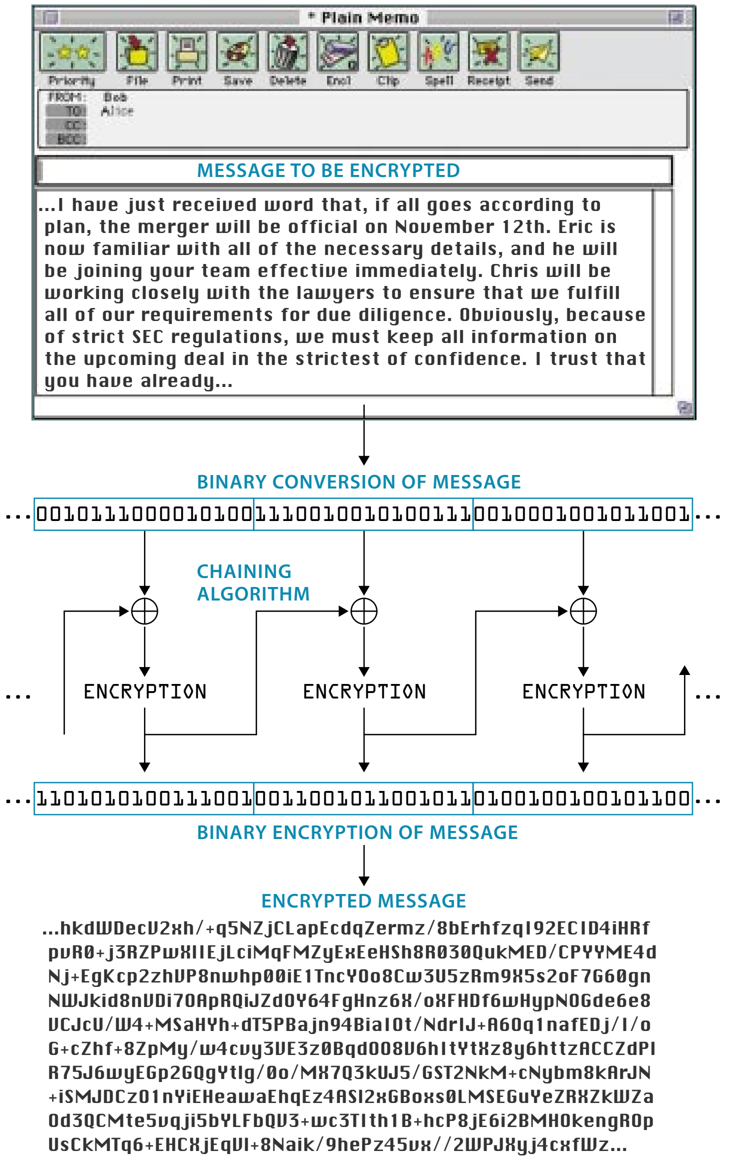
In some cases, all that the attackers have to work with is the ciphertext, so they have little to guide their efforts in guessing the key. Even a poorly designed cipher might be able to withstand such ciphertext-only attacks.

But if the attackers know at least a part of the message—for instance, that the text begins with "Dear Mr. Jones"—the opportunities for success increase significantly. At a minimum, they can try different keys until they find one that decrypts the "Dear Mr. Jones" part of the plaintext. Even if the attacker knows only the language (Russian or French or COBOL) of the plaintext, that information can be exploited. If the message is in English, for example, the most common word is probably "the." To thwart such known-plaintext attacks, some cryptography systems electronically compress the message, squeezing out easily predictable patterns in the plaintext, before encrypting it.

Often an attacker knows much more. If a person steals a "smart" card containing crypto hardware, the thief can present perhaps billions of carefully chosen messages to the card and study the ciphertext output. Such chosen-plaintext attacks will crack a poorly designed cipher easily. Another example is public-key systems. An attacker can write a message, encrypt it with the public key (which is, after all, public) and then analyze the resulting ciphertext.

Two very effective methods of cryptanalysis, differential and linear, have recently been developed. Both approaches have been used to crack a number of well-known block ciphers and to show that DES can be broken hundreds or thousands of times faster than by key exhaustion.

In differential cryptanalysis, introduced by Shamir and Eli Biham of Technion Israel Institute of Technology, many pairs of plaintext messages with carefully chosen differences are encrypted to find a corresponding pair of ciphertexts that have a certain dissimilarity. When such a pair is found, it reveals information about the key. Linear cryptanalysis, developed by Mitsuru Matsui of Mitsubishi Electric Corporation, searches for correlations between plaintext, ciphertext and key that are true slightly more often than not. The method then gathers statistics on large numbers of known plaintext-ciphertext pairs, looking for biases that will disclose clues about the key.



...I have just received word that, if all goes according to plan, the merger will be official on November 12th. Eric is now familiar with all of the necessary details, and he will be joining your team effective immediately. Chris will be working closely with the lawyers to ensure that we fulfill all of our requirements for due diligence. Obviously, because of strict SEC regulations, we must keep all information on the upcoming deal in the strictest of confidence. I trust that you have already...

```
...0010111000010100|1110010010100111|0010001001011001...
```

```
...1101010100111001|0011001011001011|0100100100101100...
```

```
...hkdWDecV2xh/+q5NZjCLapEcdqZermz/8bErhfzqI92ECID4iHRf
pvR0+j3RZPwXIIEjLciMqFMZyExEeHSh8R030QukMED/CPYYME4d
Nj+EgKcp2zhVP8nwhp00iE1TncY0o8Cw3U5zRm9X5s2oF7G60gn
NWJkid8nVDi70ApRQiJZd0Y64FgHnz6X/oXFHDf6wHypN0Gde6e8
VCJcU/W4+MSaHYh+dT5PBajn94Bial0t/NdrIJ+A60q1nafEDj/I/o
G+cZhf+8ZpMy/w4cvy3VE3z0Bqd008V6hItYtXz8y6httzACCZdPI
R75J6wyEGp2GQgYtIg/0o/MX7Q3kUJ5/GST2NkM+cNybm8kArJN
+iSMJDCz01nYiEHeawaEhqEz4ASI2xGBoxs0LMSEGuYeZRXZkWZa
Od3QCMte5vqji5bYLFbQV3+wc3TIth1B+hcP8jE6i2BMH0kengR0p
UsCkMTq6+EHCXjEqVI+8Naik/9hePz45vx//2WPJXyj4cxfWz...
```

CHAINING ALGORITHM increases the security of block ciphers. A message is converted into a string of 0s and 1s, and the long sequence is then broken into blocks of equal size. Before each of these chunks is encrypted, it is first mathematically combined with the enciphered previous block. Thus, the encryption of the 23rd chunk depends on the enciphered 22nd block, which itself was affected by the encryption of the 21st block, and so on. Because of this feedback chain, an encrypted block depends on all the previous blocks, making the cipher more difficult for cryptanalysts to crack.

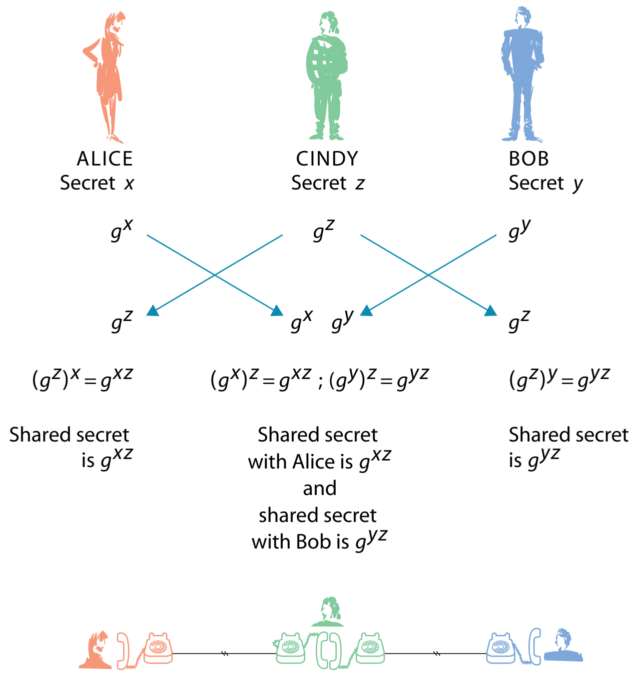
Beware of Middlemen

Though powerful, cryptanalysis techniques usually require a backbreaking number of computations. Often instead of trying to crack a cipher, it is easier to attack the protocol, or implementation, of that cipher.

One potential threat is man-in-the-middle attacks, which are the biggest vulnerability of public-key cryptosystems. When Bob wants to send a message to Alice, he may be unaware that Cindy is attempting to impersonate Alice. If Cindy can trick Bob into using her public key instead of Alice's, she will be able to decrypt Bob's message.

The only way to prevent this type of attack is for Bob to confirm somehow that Alice's public key is really Alice's. Most of the complexity of well-designed implementations of public-key cryptosystems is devoted to this one particular vulnerability. One solution is to have a trusted third party verify and sign the keys. This approach, however, begs the

ILLUSTRATIONS BY DMITRY KRASNY


ALICE
Secret x
CINDY
Secret z
BOB
Secret y
g^x
g^z
g^y
g^z
g^x g^y
g^z
$(g^z)^x = g^{xz}$
$(g^x)^z = g^{xz}$; $(g^y)^z = g^{yz}$
$(g^z)^y = g^{yz}$
Shared secret is g^{xz}
Shared secret with Alice is g^{xz} and shared secret with Bob is g^{yz}
Shared secret is g^{yz}

MIDDLEMAN ATTACK is the greatest vulnerability of public-key cryptosystems. If Cindy, an eavesdropper, can intercept transmissions between Alice and Bob, she can trick Bob into using her g^z instead of Alice's g^x and similarly deceive Alice into using g^z instead of Bob's g^y [*see box on page 112*]. Cindy would then be able to decrypt and reencrypt Alice's and Bob's messages to each other—all unbeknownst to the couple. The process is analogous to Alice and Bob talking on special, encrypted telephones while Cindy listens in by using a pair of such phones to decrypt, then reencrypt, the transmission.

major—and politically controversial—question: Should the keys be certified in a top-down manner by government authorities or in a decentralized grassroots method by different entities, including various private companies and individuals, allowing people to choose for themselves which key signers to trust? In fact, this issue is so crucial that I could have written this entire article on it.

As cipher-breaking techniques have improved, so have the algorithms for stronger cryptography. Recently the National Institute of Standards and Technology solicited designs for the Advanced Encryption Standard (AES), a new block cipher to replace the DES, which has reached the end of its useful life, mainly because of its short 56-bit key and 64-bit block size. The AES, which has been generating considerable excitement in the cryptography field, will use a key size of 128, 192 or 256 bits to encrypt data in 128-bit blocks.

Good AES designs will meet several criteria. They will offer flexibility in various key and block sizes; they will be efficient in setting up keys and in encrypting and decrypting, particularly when implemented on 32-bit processors as well as on eight-bit microprocessors, such as in "smart" cards, and on other hardware; and they will perform well in a wide range of applications, from satellite communications to high-definition television.

Several of the AES candidates appear to be extremely well designed. The better proposals have capitalized on the experience of cryptographers who have studied block ciphers for the past 20 years, including their knowledge of how to defend against linear and differential cryptanalyses.

Of the 15 submissions, I believe more than a few would make credible encryption standards. MARS, which draws on the experience of IBM's original DES team, uses two very different structures for the encryption rounds. The mixed approach, the IBM cryptographers claim, will result in better security than that achieved with a homogeneous cipher. CAST-256 extends the earlier CAST architecture to a 256-bit key and 128-bit block size. Twofish is more mathematically rigorous than its predecessor, Blowfish. Serpent deploys an unusual parallel design to make it as fast as DES, with a short time for key setup, which should enable the cipher to be used efficiently as a hash function.

Deciphering the Future

Whichever candidate is selected, the AES promises to tip the balance further in favor of cryptographers in their ongoing arms race against cryptanalysts. Today the very best cryptosystems are beyond the reach of the best cryptanalytic methods known. Still, it is conceivable that powerful, new cipher-breaking techniques will be developed in the coming years. Even so, many cryptologists contend that the gap between cipher makers and cipher breakers will only widen.

I agree with that assertion, in part because of the active community of cryptographers in academia and the private sector, which has grown and matured to reach parity with military expertise in the field. Evidence of this was supplied by the recent declassification of the Skipjack cipher, which the NSA had developed in secrecy for the Clipper chip. A review by Technion's Biham, an academic cryptologist, revealed the algorithm to be less conservative, with a smaller margin of safety, than the best designs from academia. It appears that cryptography—like the Internet itself—has stepped from the dark shadows of the military into the bright sunshine of the free market. SA

The Author

PHILIP R. ZIMMERMANN is the author of Pretty Good Privacy (PGP) encryption software, for which he received the Chrysler Award for Innovation in Design. He is a software engineer with more than 20 years of experience in cryptography, data communications and real-time embedded systems. He has testified before Congress, urging the U.S. government to loosen its control of encryption technology. Selected in 1995 by *Time* magazine as one of the 50 most influential people on the Internet, Zimmermann is currently a senior fellow with Network Associates.

Further Reading

THE OFFICIAL PGP USER'S GUIDE. Philip R. Zimmermann. MIT Press, 1995.
APPLIED CRYPTOGRAPHY. Second edition. Bruce Schneier. John Wiley & Sons, 1996.
Additional information can be found at http://www.pgpi.com, http://www.pgpinternational.com, http://www.pgp.com/phil, http://csrc.nist.gov/encryption/, http://www.epic.org, http://www.eff.org and http://www.cdt.org on the World Wide Web.