Global Headquarters: 5 Speen Street Framingham, MA 01701 USA P.508.872.8200 F.508.935.4015 www.idc.com

WHITE PAPER

The Coming of Age of Client Security: Top Managers Realize They Have to Lock Down the Point of Entry

Sponsored by: IBM Corporation

Roger L. Kay

January 2003

SUMMARY

Although security technology has progressed tremendously over time, awareness of the need for security on the part of people who use computers — both consumers and businesspeople — has not in general kept pace. Essentially, there is plenty of technology on hand, but the understanding of what it does and how to use it has lagged. However, much has changed since the attacks of September 11. CEOs and IT managers everywhere drew lessons from the differing fates of companies that had backup and restore procedures and those that didn't. Data recovery is, of course, only one piece of the security pie, but as political tensions have increased on the macro level, this and other security concerns have risen in visibility with top managers. "To what degree is our data — and therefore our business — safe?" CEOs are now asking in ever greater numbers and with increasing vehemence. "Just where are we with security?" they want to know of their CIOs.

"To what degree is our data — and therefore our business — safe?" CEOs are now asking.

This shift in attitude represents an evolution from the pre–September 11 state, which was characterized by a vague awareness of some subset of security issues but a misunderstanding of the complete security picture and a widespread lack of adoption and deployment.

Now managers are beginning to assess their vulnerability and to ask what their alternatives are.

In most corporations, the security infrastructure is still inadequate and full of holes. Even the most sophisticated organizations are vulnerable. In one incident, widely reported in the press, that had an impact of major but unknown proportions — the degree of penetration was difficult to assess — a hacker from St. Petersburg, the intellectual seat of the old Soviet Union, broke into Microsoft's network and absconded with a large number of important files, including, purportedly, an unknown quantity of Windows source code files. Naturally, Microsoft never advertised the extent of the damage — if, indeed, it is actually known. And if a company at the epicenter of the information technology business is vulnerable (and by inference should know better), truly, no company is safe from attack.

The security threat is growing in several dimensions at once. The amount of value flowing across the network — in the form of actual money, but also business plans, intellectual property, and strategic documents — is rising by leaps and bounds. And value is at risk in less obvious ways. A reputation can be damaged irreparably by an attack, business can be lost as a result of downtime, and the trust on which ebusiness is based can be destroyed permanently. To the growing list of imaginative crimes must be added identity theft, which has become a veritable cottage industry. In addition, malicious hackers are getting more sophisticated. Malevolent programmers are not only figuring out more effective ways to harm businesses and individuals but are also publishing their tricks on Web sites for other less creative, but perhaps more vindictive, people to find and use.

The security threat is growing in several dimensions at once.

In this environment, client security can be one of the weakest links in the chain. Despite the availability of operating systems with improved security features, desktop and notebook PCs still often have only a Windows password protecting them, and, in older Windows versions, these flimsy mechanisms are easy to crack. Once inside the organization by way of an unprotected node, a malicious hacker has the run of the place to the extent that the legitimate user of the system did. From this position, the intruder can execute transactions as if he were the victim. And worse, in this era of the Internet, the perpetrator does not even have to be physically onsite, but can reach the system remotely. And if the hacker is sufficiently sophisticated, he may be able to get at the most sensitive areas of the network, pillaging information, destroying functionality, or even potentially turning computer after computer into a rogue slave that does his bidding. Even if other security measures — such as physical access control, firewalls, network security, software security, database encryption, and server-level intrusion detection — have been instituted, the client node may indeed represent a weak point in the corporation's armor.

In this environment, client security can be one of the weakest links in the chain.

Although the mathematics of security are theoretically solid, a secure implementation depends on both the embodiment of the algorithms and the procedures for handling sensitive data and the keys used for encryption and decryption. Although modern encryption is virtually uncrackable, encryption implemented in software is an open door to hackers. In software encryption, various ways exist to sniff the most important element — the user's private key. To address this weakness, IBM has embedded the entire process in hardware. An industry group composed of all the major manufacturers and suppliers and many smaller ones has agreed to drive the standard into the marketplace. The Trusted Computing Platform Alliance (TCPA to its friends) is now in the second revision of the standard, and this revision is expected to be incorporated into Microsoft's Palladium security infrastructure, due to hit the market in 2004 or 2005. Although IBM acted unilaterally to design and implement its hardware solution, key players in the industry have acknowledged the design point. The TCPA was inaugurated with IBM, Hewlett-Packard, Compaq, Intel, and Microsoft as founding members. Since, its inception in October 1999, more than 190 firms have signed up, including Dell. TCPA wants its security technology to be universal in the computing industry, and IBM has committed to making it available via license to anyone who wants one.

Although modern encryption is virtually uncrackable, encryption implemented in software is an open door to hackers.

IBM itself has moved on from the original embodiment of the TCPA standard, a security chip or cryptographic microprocessor that was soldered onto the system board of the client and connected to the main processor by a local bus, and now offers an implementation as a modular daughter card. There is no way a Trojan horse can sniff the chip on the card because all private key operations take place within a protected hardware environment. Since its key-management structure is hierarchical, a single private key can be used to secure a large number of certificates (issued, for example, by diverse entities such as a senior citizens group, a corporate employer, Microsoft Outlook, American Express, and MasterCard).

The hardware is designed to work with a suite of other security elements, such as firewalls, antivirus software, security policy software, and Internet Protocol Security (IPSec), to provide a complete security solution. In addition to being extremely secure, the hardware is simple to use and inexpensive.

In an ebusiness world, trust, protection of privacy, and a secure operating environment are essential. The benefits of hardware-based security are obvious: Private keys are truly safe from malicious hackers, multiple secure keys can be generated to facilitate ecommerce with a wide variety of entities, and, combined with a full security suite, hardware encryption enables another layer of security, making ebusiness more viable. The simple conclusion is this: If your client-level security isn't implemented in hardware, your systems are more vulnerable.

The simple conclusion is this: If your client-level security isn't implemented in hardware, your systems are more vulnerable.

The Microsoft intrusion was a so-called "lunchtime attack," named for the archetypical scenario in which an employee goes out to lunch, leaving his or her computer on, and an intruder simply sits down at the absent worker's desk to feast on whatever privileges that user enjoys, including access to files, programs, and services.

Without having to resort to social engineering, a lunchtime attack can be thwarted quite easily by a variety of authentication methods based on client-level hardware encryption. For example, the operating system can be set to lock out access after a short period of time if it receives no further input and be reactivated only via biometric recognition or a proximity badge, or both, eliminating the need for passwords, which can be forgotten or stolen. If the network had been able to interrogate the remote client to find out whether or not it was authorized, Microsoft would likely have been able to prevent the attack. Had appropriate fail-safes been in place, the hack would likely not have been successful.

The need for stronger security is well demonstrated, and effective measures to protect data and users exist in the marketplace today. We're not talking about something two or three years down the road. IT managers should look into these technologies now.

THE SECURITY LANDSCAPE

In this paper, we will cover a number security-related topics, including:

- Business managers' growing consciousness of security issues
- How the PC client can be the weak point in the security perimeter
- The rise in the value of data stored in insecure computing systems
- The scope of security measures
- Security history and current technology
- Client security implementations
- The advantages of IBM's hardware security implementation
- The evolution of industry standards for client security

USAGE LAGS BEHIND TECHNOLOGY

Security technology has come a long way since the day in 1586 when Thomas Phellips, Queen Elizabeth's decipherer, broke Mary Queen of Scots' simple offset code, an unfortunate event that led directly to Mary's trial and execution. Today, a malicious hacker trying to break so-called "Triple DES" encoding with all the computing power currently hooked up to the Internet simultaneously would need 64 quadrillion years to do the job, plenty of time to slip back over the border into Scotland. And Triple DES is by no means the strongest code out there.

A denial-of-service attack on the Internet's 13 root servers successfully crippled traffic on the Internet as recently as October 2002.

But usage of security measures in the data world has not tracked the technology itself. People just haven't gotten the message that security is important. For example, denial-of-service attacks involve the penetration and hijacking of innocent people's PCs unbeknownst to them and then unleashing the enslaved systems' power simultaneously in a stream of requests that block legitimate traffic to targeted servers. These attacks first surfaced in 1999, but the average user still hangs out on the Internet with unencrypted connections, vulnerable to getting picked off by a sniffer,

and a denial-of-service attack on the Internet's 13 root servers successfully crippled traffic on the Internet as recently as October 2002. This attack has been connected to cyberterror, and IDC is expecting at least one major cyberterror attack on the Internet infrastructure in the not-too-distant future.

In addition, as wireless installations, home networks, and hotspots become more common, the opportunities for client penetration are only increasing. Many users don't even turn on the encryption available on their wireless connections. Picking traffic out of the air is commonplace, albeit mostly harmless. Nonetheless, on occasion, the crown jewels are exposed.

Of course, awareness and concern about security issues have risen among corporate executives since September 11, but a steady drumbeat of increasing Internet fraud and identity theft has been rising in the background as well. The multiple directions from which cyberdanger can come are among the main worries of IT managers. Access control and authentication are key for enterprises with remote employees. Physical security remains a hot topic, particularly as devices are becoming smaller and more mobile.

Although IDC surveys show that IT executives in companies engaged in ebusiness activity have always led others with respect to security, awareness and implementation are beginning to become more mainstream for enterprise networks. Security has moved from the global realm of total systems, such as the public key infrastructure (PKI), which require cooperation and trust among multiple entities, and focused on the more immediate task of authenticating users at the point of entry and encrypting local files.

While at the highest level most of the attention to security is focused on protecting the information of greatest value to the corporation — financial, personnel, and proprietary technical data — whether it lies in the mainframe, on the network, or in clients, at the low level of client protection most of the focus has shifted to ensuring that the *cordon sanitaire* is unbroken at the access point and that user files are secured. Good mainframe security implementations, particularly at the procedural level, have been in place for a long time. Network security, which makes use of techniques such as intrusion detection and firewalls, is primarily concerned with availability and integrity. Client security is now extending from antivirus products and limited password controls to robust authentication methods and protection of intellectual property.

THE IMPORTANCE OF THE CLIENT TO OVERALL SECURITY INFRASTRUCTURE

Security in a networked environment is achieved by deployment of a full set of protective measures. These measures include software-based perimeter defenses and antivirus software, which, deployed on both servers and clients, help protect computing assets from destruction, and hardware-based authentication and encryption tools, which guard against privacy loss, identity theft, and data tampering.

Almost all clients now have user log-ins and passwords, but these limited protections are sometimes left unchanged by the user from the manufacturers' uniform settings — often common words such as "user" and "password" or just plain blank. And if the password has been set properly (i.e., to a longish string, say, of at least eight characters, of mixed letters and numbers that do not make up any common words), a malicious hacker can still enter the system using a "hammering" algorithm such as L0PHT, which, by trying a multitude of combinations of characters in rapid fire, can crack open a standard corporate PC password like a coconut in less than a minute. More primitive client password schemes — still used in Windows 95 and 98 installations — can simply be bypassed by hitting the Escape key.

And even with the best of intentions, IT departments do not always upgrade all their systems with the latest security patches, sent out by application, antivirus, and operating systems companies when they discover flaws that allow outside penetration. The hacker community knows about these flaws and cruises the Internet, looking for systems that lack the updates.

Once inside the network via a vulnerable client node, a hacker with malevolent intent has all the privileges accorded the legitimate user of that client: access to files, programs, system resources, and, potentially, other users' PCs. And if the hacker is sufficiently sophisticated, he may be able to grant himself privileged status and get at the most sensitive areas of the network, turning computer after computer into a captive resource. From this position, he can destroy or alter files, corrupt programs, erase nonvolatile storage devices, and co-opt system resources to carry on further mayhem.

Once inside the network via a vulnerable client node, a hacker with malevolent intent has all the privileges accorded the legitimate user of that client: access to files, programs, system resources, and, potentially, other users' PCs.

Thus, even if other security measures — such as physical access control, firewalls, network security, software security, database encryption, and server-level intrusion detection — have been instituted, the client node may represent a weak point in the corporation's armor. Improved authentication on all nodes would help mitigate this situation. No network is safer than its least-secure node. A full security perimeter necessarily involves a solid defense at the client level.

THE ADVENT OF ECOMMERCE AND THE RISE IN THE VALUE OF DATA

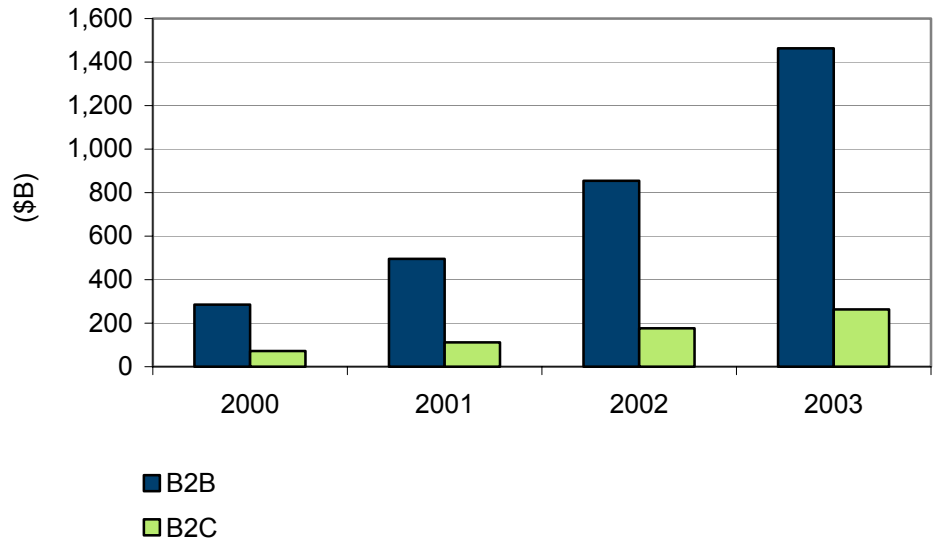
Why should client security matter more now than it has in the past? Until recently, few organizations had a need for systematic data security. Banks and other financial institutions had to ensure end-to-end security for storing and moving money around over wires. Certain government agencies could only operate in an impregnable data fortress. But the volume of valuable data being stored and transmitted by most firms was relatively low. All that is being changed by the advent of electronic commerce.

A tremendous amount of value is already flowing through the Internet. And far more is coming. IDC estimates that the value of Internet commerce was $50 billion in 1999, and this figure will grow by several orders of magnitude to $1.7 trillion worldwide in 2003 (see Figure 1).

This value takes many forms. For individuals, the stakes range from credit card number loss to identity theft. But for corporations and governments, the value of the intellectual property inside the computer can be astronomical and, as in the Microsoft case, sometimes incalculable. However large the threat is to individuals, it is far greater to corporations. In the corporate world, there are a host of values to be lost — money, first and foremost. Fraudulent actions can be enormous, in the tens of millions of dollars in a single transaction. Value is also represented by nonfinancial assets, such as intellectual property, business plans, and strategic documents. Pilferage of corporate secrets could lead to a loss of competitive advantage, potentially condemning a firm to death by slow strangulation.

FIGURE 1

WORLDWIDE ECOMMERCE SPENDING BY TYPE, 2000–2003



Source: IDC's *Internet Commerce Market Model* version 8.1, February 2002

Authorities in the United States recently cracked the case of a professional hacker based in the United Kingdom who had access to about 100 unclassified military networks during most of 2002. The case, which had been considered a high priority for a year, despite the unclassified nature of the networks, was a focus because of the skill of the hacker, who was finally snared. Although the exposed data was not particularly sensitive, the determination and talent of the hacker led authorities to believe that it would be only a matter of time before he uncovered something valuable. Among sites he was able to enter were the Pentagon Picatiny Arsenal in New Jersey, one of the Army's most delicate research facilities. And he was in the process of unfolding a multistage attack, the sign of a highly sophisticated hacker, at the time of his arrest.

In a case involving far greater value, two Russian hackers were lured to the United States by federal agents on the pretext of a commercial interest. Once in Seattle, they were arrested. But until that moment, they had been engaged in an operation that had hacked into banks and ecommerce sites and extorted the operators for money with the promise of not revealing the hacks to the public. Sometimes the value of reputation damage is difficult to assess, but it may represent the entire value of the business. Another Russian hacker was monitored for years as he downloaded millions of pages of sensitive data from defense department computers, including one colonel's email inbox.

Aiding and abetting the rise in attacks, the collective pool of hacking knowledge has risen. Hackers often trade schemes and software via Internet Relay Chat, better known by its initials, IRC, one of the least-regulated areas of the Internet and one that allows anonymous contact at the user's discretion. The packets flow to everywhere from everywhere. And although more policed areas of the Internet exist (e.g., AOL and other "communities" in cyberspace), the underlying structure still relies on real-time routing, and packet spoofing makes it possible for someone to conceal his whereabouts, particularly if he comes from a number of directions at once.

Aiding and abetting the rise in attacks, the collective pool of hacking knowledge has risen. Hackers often trade schemes and software via Internet Relay Chat, better known by its initials, IRC, one of the least-regulated areas of the Internet and one that allows anonymous contact at the user's discretion.

Companies are subject not only to fraud and the direct loss of assets but also to the value of lost business. When their services are denied by a deliberate overload of bogus requests, they lose the value of the potential business that would have been transacted during the period of denial. Another less tangible but perhaps ultimately more disastrous effect of such attacks is damage to reputation. The harm can be irreparable. Public confidence in a company may be shaken beyond repair by a particularly malicious attack or series of attacks. For electronic commerce to function, customers and partners need to be able to trust the ebusiness process.

And security requirements will only rise as companies turn increasingly to ebusiness. Although the encryption technologies today are sufficient to guarantee complete confidence and, mathematically, a user can have perfect assurance that a message is unique and really did come from the person who says he or she sent it, in order for the system to be a trustworthy enough medium in which to do business, the infrastructure must be whole. Given that most companies' security focus is on network servers, routers, and firewalls, it may be that the client node is the overlooked weak link in the security chain, but it is by no means the only possible point of penetration. Breaches can be internal or external. Often, depredations come from the employees themselves. Employees must be protected from each other so that all intranet users trust the system. And corporations must be shielded from external threats, hostile outsiders who may enter the castle from the Internet via the many connections most firms maintain to communicate with the outside world. For both internal and external transactions, users must be able to trust and be trusted.

SECURITY TECHNOLOGY: FROM GLOBAL TO LOCAL

Public key encryption and its associated infrastructure address the issue of trust at the global level. Of the many elements that make up a total security solution, however, PKI is the most dependent on completeness; that is, any two parties participating in secure transactions must both agree to rely on a third party, a trusted authority, sometimes called a certificate authority.

It is because of the complexity of implementing the PKI infrastructure that companies have recently turned to less ambitious tasks with respect to guaranteeing security at the client node. Encryption similar to that used to pass keys back and forth over a network between participants in a PKI scheme can be used to perform far simpler — but no less important — jobs at the local level. For example, without having to resort to the network at all, a PC client can provide its user with securely encrypted folders, the contents of which would look like gibberish to any hacker who managed to open them. Using one or more authentication techniques (e.g., some combination of biometric access control, proximity badge, and password), only the legitimate owner of the locked-away files can open them as readable data. This same type of authentication can be pressed into service to authorize the client node's user to the network and all the corporate resources it contains.

THE EVOLUTION OF SECURITY TECHNOLOGY

Security has come a long way since the need for it was first perceived. The development of security technology has followed both the leapfrog-like need to stay ahead of the competition and the availability of the means to do it.

The essence of encryption is the systematic altering of text or other data by mathematical transformations (algorithms), processes that are inherently abstract (i.e., they can be embodied in either software or hardware). Also critical to the success of any security scheme is a set of procedures for handling both the original (clear) and transformed (encrypted) text. In this area, some sets of procedures are distinctly better than others, as we shall see.

For many years, encryption algorithms were quite simple. The offsets used by Mary Queen of Scots relied on the slowness of human decipherers, who were often as much psychologists as mathematicians. If every letter in the encrypted message was, say, five letters up the alphabet from the original (wrapping around again at Z), then decoding one word was enough to break the whole text. A bit more complicated would be incrementing the offset by a fixed amount, which would take a little more doing on the part of the decipherer but would still yield to trial and error.

These types of techniques were supplanted by the use of "key texts," a method a notch further up the complexity chain. The offsets to the clear text were determined by the value of the letters of another text, which could be any written document agreed upon by both sender and receiver. The document was usually a book, and the key could start anywhere in the text (say, on the 23rd letter of page 23). This method worked pretty well, except when the encrypted message was intercepted along with one or the other of the concealing parties, at which point the shared secret could be "coaxed" out of the unfortunate soul. These algorithms all depended on the absence of computing power, which in today's world can perform, in a relatively short period, "brute force" trial-and-error sequences that a human could never hope to produce in a lifetime.

These algorithms all depended on the absence of computing power, which in today's world can perform, in a relatively short period, "brute force" trial-and-error sequences that a human could never hope to produce in a lifetime.

FROM DES TO AES

One of the first big improvements in security came in 1970, when IBM scientists developed the Data Encryption Standard (DES). DES starts with something like offsets but uses complex permutations. The algorithm itself is in the public domain, but, without the key, the result of any particular instance of usage is nearly opaque. Essentially, the clear text is broken into groups or blocks of 64 bits and then transformed using an algorithm dependent on both the message bits and a key, which is 56 bits long (8 bits being reserved for parity check). This stuff is pretty thorough. As a rule of thumb, changing one bit of input in the clear text changes the values of half the output bits in the encoded text. To break this code without the key, a decipherer has to try 2^{56} or 72,057,594,037,927,936 combinations (72 quadrillion, for those intimidated by the sight of large numerals), and because of the dynamism of the DES algorithm, it is extremely difficult to reduce the size of the search space (search-space reduction being one of the more important techniques at the disposal of decipherers) other than by luck. Until the mid-1990s, only the National Security Agency had the computational power to crack a 56-bit DES-encoded message with brute force.

In the mid-1990s, commercially available computing reached a level of performance sufficient to break DES in a matter of hours, and privacy seekers started using Triple DES, which essentially runs clear text through the DES washing machine three times, using a different key on each pass. Triple DES was considered quite secure, requiring a code breaker to cover a search space of 2^{112} combinations. The only reason the search space is not 2^{168} is that by that time complex cryptoanalytic techniques had been discovered that reduced the maximum search space. Nonetheless, Triple DES represented a reprieve for the existing standard. It would still take all the computers on the Internet more time to crack than the earth is likely to last, not to mention the human race or something as geologically transient as electricity.

However, even Triple DES had a couple of major weaknesses. It was a symmetric key encryption method, an Achilles' heel that in some ways makes it no stronger than the old key-text method. The algorithm is called symmetric because the math to encrypt a message is simply run backward to decrypt it. This scheme requires both the sender and recipient to have the same key. Both parties have to share a secret, and they must be able to exchange that secret secretly. And so the possibility exists that clever Internet sniffers or bad men with pointy sticks can extract the secret at either end of the transmission or even in the middle and pop open the message. After all, the key is just a series of numbers, albeit long ones. In addition, because of the

nature of computer operating environments, DES slows down data flow considerably when executed in software, and Triple DES slows down the system three times more.

Thus, in the late 1990s, the National Institute of Standards Technology (NIST), formerly the National Bureau of Standards, put out a call for new algorithms, and a competition ensued. The specification for the new standard, called Advanced Encryption Standard (AES), required that it be easily implemented in software, that the key length be bumped up from 56 to 128 or 256 bits, and that the block size be increased to 128 bits. With these specifications, AES would be far too large for anyone using any method to search the key space. After several years, the competition was narrowed to a few finalists. IBM championed an algorithm called MARS; cryptographers in Cambridge, England, put forth Serpent; and Schneier produced a viable competitor, as did RSA Labs. All the finalists' algorithms were considered more than secure enough, but one written by a couple of cryptographers in Belgium, Joan Daemen and Vincent Rijmen, called Rijndael (a euphonious, if not cryptographic, mixing of their names) was chosen partly because it was both fast, even in software environments, and small.

Rijndael was chosen partly because it was both fast, even in software environments, and small.

PUBLIC KEY — STILL BETTER

Despite the speed issue, symmetric key methods are relatively fast because they are computationally less intensive than other more secure methods. Because they have relatively less impact on the data rate, they are desirable for encrypting bulk data for storage and transmission. However, the problem of the shared secret is left unsolved, even with AES. And so, the best encryption techniques involve doing three things, which are a combination of technology and procedures: wrapping the shared AES secret in a much more robustly encrypted envelope, encoding the main message with AES, and throwing the whole thing away after a single use. One-time usage makes the value of decryption low to an interceptor, even as the cost is high. As a matter of jargon, a one-time key is called "ephemeral."

The more robust method used to encode the AES keys is called asymmetric or public key cryptography. The asymmetry refers to the fact that mathematically related but different keys are used for encoding and decoding. When the private key is used to encrypt a message, only the associated public key can be used to decrypt it. When the public key is used to encrypt a message, only the associated private key can be used to decrypt it. The public key can be shared with anyone, but the private key must be kept secret and should only be available to the owner of the key. Knowledge of the public key does not disclose any information about the private key. The first asymmetric encryption method to reach commercial usage was brought to market in the late 1970s by three MIT professors, Ron Rivest, Adi Shamir, and Leonard Adleman, whose initials just happened to combine to make the name RSA, which is now the moniker for the de facto standard in public key encryption.

When the private key is used to encrypt a message, only the associated public key can be used to decrypt it.

Here are two illustrations of how this type of encryption can be useful. Let's say that sometime in the near future, you'll be able to vote over the Internet. If every voter has a pair of private keys safely tucked away in his or her computer, and for every voter a pair of public keys resides at the statehouse, the courthouse, and the White House, then when an encrypted vote from you comes in, only the public key associated with you and only you will be able to decrypt it. Thus, if a vote purports to come from you, and the vote counter pops it open with your key, then that vote can be guaranteed to have come from you — assuming your client node is inviolate, which underscores the need to secure the network at the client end. Going the other way, if I want to send you a secret note that only you can open, I can encrypt it with your public key, which I can get because it is public, and only you can open the message. These examples illustrate two important aspects of security: authentication and privacy.

Public key encryption is based on the idea that some mathematical operations are easy to do — but hard to undo. A simple example is a square versus a square root. If you already have the square root of three (which, although approximately 1.73205080756888, has no finite answer), multiplying it by itself easily yields three, but trying to find the root given only the number three is a lot more difficult. The essence of the RSA algorithm is the same: Two large prime numbers are easily multiplied, but factoring the result to find the original numbers is extremely difficult. Asymmetric encryption starts with two randomly chosen 100-digit prime numbers. The sender "knows" them or at least has possession and usage of them. They are multiplied together, and the product becomes one of the two elements in both the keys. For the other two elements (one each for the public and private keys), one is chosen from a restricted set that relates to the first two and the other is derived algorithmically from the product of the first two and the third (the one just chosen). These four numbers (three, really, since one is shared) are the kernels for RSA asymmetric encryption. If the public key pair is used to transform clear text via complex mathematics, then only the private pair can be used to decrypt, via a similar set of calculations. By the same logic, if the private key pair is used to encode the clear text, only the public key can be used to decipher it. Although the two key pairs are interrelated, neither can be derived from the other.

The strength of public key encryption is that it is fantastically robust. Anyone can send a message encrypted with a public key, but only the holder of the associated private key can decrypt it. The weakness of asymmetric cryptography is that it is computationally intensive and would slow down data traffic unacceptably if it were applied promiscuously. So, as previously mentioned, in practical circumstances it is used only to encode the symmetric key (i.e., the AES key) used for bulk data encryption. The result of encoding the symmetric key with an asymmetric public key is called a "digital envelope," and the process is referred to as "PKI key exchange."

IDENTIFYING THE SENDER AND GUARANTEEING DATA INTEGRITY

We now have an infrastructure robust enough to guarantee the identity of the sender. The sender is fairly confident of the recipient because only the proper recipient has the correct private key pair and can turn the message back into clear text. But a trusted third party (sometimes called the "certificate authority") is required as well — one that knows all participants and guarantees the identity of the sender. Everybody has access to the authority's public key pair. Once the sender has proven his or her identity (through, for example, a handwritten signature, iris scan, voiceprint, or fingerprint), the authority is able to return a copy of the sender's public key, "signed" with the authority's private key, and the sender can include this "certificate" in his or her outgoing message. Thus, you are proven to be you for the purposes of ebusiness.

The signature is simply a secure one-way "hash" of the message itself, encrypted with a sender's private key. Analogous to a Cyclic Redundancy Check (CRC), the hash is produced by reducing the message through an algorithm to a "digest," a string of between 64 and 256 bits. The original message cannot be reconstructed from the hash by any means because most of the information has been destroyed in the hashing process. However, the hash is uniquely related to the original message mathematically. As with the AES algorithm, changing a single bit in the message will change half the bits in the hash. The hash, encrypted with the sender's private key, is attached to the message, and at the recipient end, the same math is performed on the message itself. The recipient decrypts the signature with the sender's public key and compares the result to the local hash of the message. If the two strings match perfectly, then the recipient is sure that the sender is authentic and that the message has not been altered during transmission. Thus, data integrity is assured.

In the public version of security, a world of ecommerce, where people can freely trust the Net and all the clients and services that they run into, there would be no inelegance. But in the real world, people of a certain disposition and skill can game the system and filch unprotected private keys, forcing the owners to go back to the authority and ask it to disallow that pair. The authority has to maintain a list of revoked "certificates," records that contain details about senders' identities, details about the authority, senders' public keys, expiration dates, and digests of the certificates themselves. Certificates, obtained from and validated through the authority, are used to vouch for the sender's public key and irrefutably connect the sender to a set of public-private key pairs. A valid certificate provides the recipient with a guarantee that the sender cannot repudiate the transaction contained in the message, a critical feature for doing ebusiness. Another complication with PKI is the existence of more than one certificate authority, and participants must have a public key for each. Common certificate issuers include VeriSign, Baltimore, Entrust, and Xcert. Finally, certificate authorities need clear procedures to verify that participants are who they say they are.

However, the good news on the procedural side is that the only keys each participant has to worry about are his or her private key, his or her public key, and the public key of the certificate authority. With symmetric keys, participants have to keep a copy of the keys of everybody they ever correspond with.

So, that's great. We have an unbeatable algorithm and a theoretically robust infrastructure to operate it in (if we can find a third party to trust). But what about implementation? In what medium do we utilize this powerful math?

CLIENT SECURITY IMPLEMENTATIONS

Because of the unresolved procedural issues involved with implementing a fully secure infrastructure, some of the grander visions of secure computing have been scaled back, at least in the short term. Companies need not wait until all parties agree on all aspects in order to shore up their security perimeters. Even if it is not yet feasible to send and receive data from all customers and all suppliers over secure, verified links, it is possible and even easy to set up basic security at the client node.

User authentication at the client end can be performed adequately with smart cards, strong passwords, or biometric identification systems. The ideal client-protection procedure involves some combination of two or more of what you have, what you are, and what you know, which are defined as follows:

- **What you have.** Can be a smart card or proximity badge, with which the client system must interact in order to operate
- **What you are.** Can be a biometric measure of your iris, voice, or fingerprint
- **What you know.** Can be a password or your mother's maiden name

Smart cards are credit card–size cards that carry their own microchip. Smart cards can carry keys and, in conjunction with authentication software, can be used to identify a user trying to log on to a particular client. One of the benefits of this type of user verification system is that, as a hardware implementation, it can be outfitted with a counter to prevent hammering. Any user attempting to log on too many times can trigger a lockout of the smart card. However, smart cards have certain drawbacks. For one thing, the number of keys a smart card can hold is limited, which is a problem from the perspective of likely developments in ebusiness, for which users will likely require a large number of secure keys to conduct transactions with a variety of entities. Also, a smart card and smart-card reader, which are necessary add-ons, are relatively expensive.

Biometry — authentication by fingerprint, retinal scan, voice, or facial geometry — is particularly good for matching employees or customers with systems and data records. While biometry represents a key piece of the security puzzle, biometric information carries no data and cannot in itself support PKI. An improvement over passwords, biometry provides better security because users cannot alter their biological qualities.

Passwords are ever useful as an added security step, even though biometric entry can be a complete substitute for passwords. Still, a password can help prevent ID spoofing, which hackers can still sometimes practice successfully against systems protected by only "what you have" methods.

THE WEAKNESS OF SOFTWARE-ONLY SOLUTIONS

A key distinction between core security implementations is whether they are software or hardware based.

There are a number of reasons why hardware-based security is better than software-based security, speed being among them, but you really only need one good one. And here it is: Software security is hackable.

In January 2000, researchers at nCipher in Cambridge, England, came up with an algorithm that can search main memory, looking for a high degree of entropy. A good private key is going to be exceedingly entropic; that is, the random numbers in the key will be well dispersed in numeric space. Other elements in memory — such as the clear text to be encrypted and the encryption program itself — won't be. All three — the program, the data, and the key — have to be in main memory at the same time for software encryption to take place. The nCipher algorithm, in combination with a Trojan horse such as Back Orifice, which, as mentioned earlier, allows someone on the Internet to commandeer a PC, will let the intruder scan the contents of main memory and find the user's private key. Back Orifice is good at masking itself, encrypts its own outgoing traffic, and was released in source code about two years ago at a hackers' conference. The nCipher program can find a 1,024-bit private key, the best in commercial use. And if a malicious hacker can get your private key, he can get your identity — and your right to do business.

Another weakness of software solutions is that they cannot prevent hammering because they are unable to keep a counter. A hacker can always freeze the state of the machine and continue to bombard it with attempts. But this flaw pales beside the problem of leaving highly entropic private keys around in main memory.

Bottom line: Private keys, symmetric keys, credit card numbers, and anything else stored on clients or servers protected by only software encryption are more vulnerable than those protected in hardware.

Bottom line: Private keys, symmetric keys, credit card numbers, and anything else stored on clients or servers protected by only software encryption are more vulnerable than those protected in hardware.

THE STRENGTH OF HARDWARE SECURITY

Because of the weakness of software-only solutions, IBM set out in the direction of implementing encryption operations in hardware. Initially an in-house project, the resulting architecture and silicon designs have been widely adopted in the information technology industry.

The IBM security chip is extremely secure, simple to use, and inexpensive. The chip, actually a cryptographic microprocessor, can be embedded in the system board of the client. It supports RSA PKI operations and includes electronically erasable programmable read-only memory (EEPROM) for storing key pairs. The chip communicates with the main processor via a local bus and also has a link to the system BIOS during boot up. An application program interface (API) routes

cryptographic operations through the chip. Cryptographic middleware automatically routes function calls to the hardware.

The chip is compliant with Microsoft's CAPI and PKCS #11, industry-standard interfaces, which many of the PKI providers, such as Entrust, Baltimore, and Microsoft itself, use for applications such as email (e.g., Outlook and Notes), VPN clients (e.g., Cisco, SonicWALL, and L2TP), or network log-on clients (e.g., NetWare).

The chip library supports 512-, 1024-, and 2,048-bit key generation; encryption; decryption; and digital signature operations as well as 256-bit decryption for symmetric key operations. All private key operations take place within the protected environment of the chip. The keys, which are generated internally and stored on the chip, never appear in main memory. So there is no way a Trojan horse can sniff it.

When a system with the a hardware security chip is first booted up, the chip must be enabled with a BIOS setting (the BIOS itself is protected by an integrity procedure). No one can give a user his or her identity. Each system owner configures his or her own personalized subsystem identity by initializing it. This subsystem identity key pair is called the "hardware key pair." Once inside, the private key is used only to hide other keys and never to identify the system or owner.

In its most recent implementation, the security chip has been paired with external hardware, such as a PC Card–slot or USB-attached fingerprint reader from Targus, a USB-connected proximity badge from Ensure Technologies, or even a smart card. IBM's focus has shifted from providing fully authenticated PKI communications and guaranteed ebusiness transactions to the more straightforward tasks of making sure that the individual logging on to a particular client node is the authorized user and that his or her local data is protected from intruders.

A HIERARCHY OF KEYS

One of the greatest strengths of hardware security architecture is the hierarchical nature of its key-management system. The first key pair generated is used to protect another key pair, called the "platform identity key pair." This key pair is created under the system owner's control and can be used by the system owner to definitively identify the PC.

As part of the subsystem initialization, the owner of the system can make an archive copy of the platform private key. The platform private key is encrypted with the administrative public key. The corresponding administrative private key can be split into up to five parts, allowing the restoration responsibility to be secured among multiple administrators. This archive data, including the administrative private key, can be stored on external removable media or on a network server. If the system, chip, or motherboard dies, or the system needs to be upgraded, the owner has the ability to securely migrate all of his or her key information from the old system to the new system. The security administrator might or might not want to use this sort of backup scheme, which represents a back door to the system, but it is there for corporate implementations. Without it, the whole system could become inaccessible. With it, as with any archive system, a potential security exposure exists if the administrator's private key is ever compromised. Each firm has to assess its circumstances and risk profile.

Next, a "user key pair" is created. The private key of the user pair is encrypted with the public key of the platform pair. Before encrypting the private key of the user pair, a "passphrase" (up to 128 characters) is associated with it. Then the private key and passphrase are encrypted with the public key of the platform pair. As another level of protection, the chip will not execute any operations if it doesn't receive the correct passphrase for that key.

Unlike software encryption, which can't keep a counter, the chip can keep track of log-in attempts, and it won't let the count-per-time rise too high, interpreting repeated assays as hammering behavior. Each failed attempt increases the length of the delay before a user can try again — up to 28 days. Although this feature can be reset with an administrative passphrase, it functions as a good antihacking mechanism.

The user key is not used for signing anything but allows the chip to load keys from elsewhere. Unlike a smart card, the chip can work with multiple certificates (issued, for example, by a senior citizens group, a corporate employer, Microsoft Outlook, American Express, or MasterCard). The number of keys can get quite large since each organization a user might interact with will have its own.

ONE ELEMENT OF A SECURITY SUITE

With one of the security factors thus based in embedded hardware, dual-factor client security systems can include, as mentioned previously, a biometric authenticator or proximity badge to further hinder identity spoofing and lunchtime attacks. Tied to third-party authentication tools, embedded hardware security can plug some of the more vulnerable holes in the security perimeter. For example, the range of a proximity badge, which operates over a radio frequency link, can be configured from five feet — for really secure — to 30 feet — for still pretty secure protection against lunchtime attacks.

With one of the security factors thus based in embedded hardware, dual-factor client security systems can include, as mentioned previously, a biometric authenticator or proximity badge to further hinder identity spoofing and lunchtime attacks.

In the Targus biometric recognition implementation, a spring-loaded PC Card–based device with a small reader on it pops out with a finger push. The device reads the user's fingerprint, which is used initially to set up access, and if it finds a match, permits log-on. The software included with the device lets the user map any application requiring a password to this surefire authentication system.

The security chip, which is now available worldwide, is designed to be used with other security elements. For example, it will not protect against a virus that can wipe the hard disk clean. Firewalls and antivirus software are required for that type of defense. The chip just keeps data private and confidential and provides for PKI operations. IBM and other vendors offer suites of interrelated security products to create a fully secure environment. For example, IPSec protects communications links by securing the Ethernet controller.

Another key feature of the IBM-embedded security chip is that it is inexpensive — to the point where IBM has included it in select client systems at no additional charge to the buyer. The company charges about $25 for the chip to commercial buyers, which is less than the cost of the simplest hardware token (e.g., a USB key) and one-half to one-third the cost of the least-expensive smart card. For the degree of utility it provides in *de novo* installations, nothing else can match it on a price-performance basis. Hardware-based solutions implemented as cards are more expensive — in some cases up to $2,000 — and a perpetrator could put a sniffer on some aftermarket cards. Also, the chip ties the trust to an actual PC rather than to a removable card. The only possible way to hack the chip is by direct physical attack (and even this involves such "high-spook" work that only a very few cryptoanalysts, mostly employed by the dark sectors of governments, can even think of mounting such as assault), which involves sensing voltage changes on the power lead and gives only an indirect view of activity inside the chip. A successful malicious hack cannot be launched remotely.

The only penalty that an organization might pay for using encryption of any sort — the IBM chip or another hardware or software implementation — is that the process creates some computing overhead. However, today's PC systems — based on multigigahertz processors, generous and faster memory, and wider and faster system buses — have more than enough power to compensate for this performance "tax."

THE TRUSTED COMPUTING PLATFORM ALLIANCE EVOLVES

IBM has put together one of the most comprehensive suites of security products in the computer industry. Many of the elements evolved from the company's own R&D and others have been adapted from other firms, such as RSA and Intel. Although IBM acted unilaterally to design and implement its embedded solution, the design point has been acknowledged by key players in the industry. The Trusted Computing Platform Alliance (TCPA), which was inaugurated with IBM, HP, Compaq, Intel, and Microsoft as founding partners in October 1999, now has more than 190 members, essentially everybody who's anybody in the PC business. TCPA's position on the technology is that it wants it to be universal in the computing industry, and IBM is committed to making its development available via license to anyone who wants one. More important, though, the success of any security strategy depends on its comprehensiveness and universality, and it is in IBM's interest that this solution become as widespread as possible. The platform specification, which has been agreed upon by the general membership, is now shipping in version 1.1. Atmel, based in Colorado, was the first manufacturer, and then Infineon, a captive semiconductor fabricator owned by Siemens, came aboard. The Siemens connection opens the door for a smart-card implementation of embedded security. Other manufacturers include STMicroelectronics in Europe and California-based National Semiconductor. The 1.1 specification is available at **www.trustedpc.org**.

The next revision of the specification, version 1.2, is currently being refined. It is envisioned as part of an overarching security infrastructure, code named Palladium, now being created by Microsoft. Palladium, which will incorporate TCPA's work, will handle a wide variety of content and client security functions, including many — such as digital rights management for copyrighted material — outside the scope of the TCPA specification. Version 1.2 will be implemented in conjunction with future processor and chipset families from Intel and others and will have to wait for Microsoft's Longhorn generation of operating system, currently scheduled for release in 2004.

CONCLUSION

In a trusted computing environment, the most important thing a participant owns is his or her private key pair. It proves identity. At the level of data interchange, it can be used to sign messages and exchange symmetric keys and it forms the basis for participation in nonrepudiatable ecommerce. At the level of the local client node, it can be used to uniquely authenticate the owner and store his or her files privately. The private key must be kept absolutely secure.

In a trusted computing environment, the most important thing a participant owns is his or her private key pair. It proves identity.

A public key pair is open to everyone and need not be secured. Since the symmetric keys used for bulk message encoding operate only once, the loss of any one key exposes at most a single message. For these reasons, keys other than the user's private pair have relatively low security requirements. But it is difficult to stress sufficiently the importance of keeping a private key secret. And the only way to ensure that the private key is totally safe is to implement security in embedded hardware.

In an ebusiness world, trust, protection of privacy, and a secure operating environment are essential. The benefits of the TCPA-embedded security chip are obvious:

- Private keys are truly safe from malicious hackers.
- Multiple secure keys can be generated to facilitate ecommerce with a wide variety of entities.

- Combined with a full security suite, the chip enables the peace of mind necessary to make ebusiness viable.

But while widespread adoption of PKI is still some way off in the future, security implementations that require cooperation between fewer parties are here now, such as secure support for email and for Microsoft's Outlook via CAPI. Since the first version of the chip, the industry has learned that this technology can be used more like pliers for general work rather than like a wrench of a specific gauge for a narrowly defined task. The chip can provide the encryption element for diverse operations:

- In a TCPA-enabled system, the chip can be used to determine if the BIOS has been changed since the previous boot.

- The embedded chip can perform the same authentication functions as an RSA secure ID keyfob, a device that costs in the range of $55–80. Without the requirement to have a hard token, chip-based authentication can be done for less than half that price. Today, about 10 million systems in the installed base have such keyfobs.

- Encryption for sending bits over the air in a wireless LAN via 802.1x, which ships with Microsoft's Windows XP, works flawlessly with the chip. The embedded chip is tied to the Microsoft code so that if the user chooses Wireless Application Protocol (WAP) encryption, the Wireless Transport Layer Security (WTLS) protocol, which is a derivative of Secure Sockets Layer (SSL), is invoked. This protocol begins with a secure certificate exchange between wireless nodes.

- Within a single node, the chip can be used at will for individual local file and folder encryption. Files and folders can also be encrypted or decrypted on the fly when saved or opened by the authorized user.

- The chip can be used along with the IBM Client Password Manager software to replace most of the user's passwords with a single passphrase or a fingerprint or a combination of both.

The simple conclusion is this: If your client-level protection isn't implemented in embedded hardware, you haven't achieved the best and lowest-cost security solution.

The simple conclusion is this: If your client-level protection isn't implemented in embedded hardware, you haven't achieved the best and lowest-cost security solution.