Broadcast Encryption's Bright Future

As a vehicle for content protection, broadcast encryption promises more flexible and resilient security compared with traditional public key cryptography techniques.

Jeffrey Lotspiech
Stefan Nusser
Florian Pestoni
IBM Almaden

For 25 years, public key cryptography has reigned supreme as the foundation of secure systems. Recently, however, advances in a new cryptographic area, *broadcast encryption*, threaten to unseat this technology in certain applications. Amos Fiat and Moni Naor[1] spawned this new technique when they published a seminal paper in 1993 addressing the following question: "Can two devices, previously unknown to each other, agree upon a key if they only have a one-way communication path?" They can indeed, and the one-way nature of this communication gives broadcast encryption its name.

The conditional access application motivated Fiat and Naor's work. This technology restricts access to a TV cable system's premium channels to viewers who have paid a subscription fee. Ironically, this flagship application for broadcast encryption has proven less important than another application—media content protection. This application has the same one-way nature as an encrypted broadcast: A recorder makes an encrypted recording and, years later, a player that might not even have existed when the recording was made needs to play it back. This situation allows no opportunity for the player and recorder to communicate a cryptographic key.

To commercialize broadcast encryption for media content protection, IBM, Intel, Matsushita, and Toshiba founded the 4C Entity, LLC in 1998 and developed the Content Protection for Recordable Media technology. Devices that use CPRM have already been marketed, including DVD audio players, DVD video recorders, and some flash memory music players—specifically, those that use Secure Digital Memory Card or Secure CompactFlash.

The content protection scheme for DVD video predated Brendan Traw's insight that broadcast encryption could be applied to protect media.[2] Instead, DVD video uses the Content Scrambling System, a shared-secret scheme. According to a story that's at least 50 percent apocryphal, a 16-year-old in Norway found the shared secret and thus broke CSS in 1999.

Regardless of the actual facts, the CSS scheme *has* been broken, and it is now easy to find programs on the Internet that will decrypt a DVD protected by CSS. Thus, short of redesigning the system from scratch, the CSS security break cannot be fixed. Had the scheme been based on broadcast encryption, on the other hand, developers could have issued new discs that would exclude those circumvention programs without affecting legitimate consumer devices.

Our work focuses on various aspects of content protection, which we believe is the ideal application for broadcast encryption. However, some applications match badly with broadcast encryption, thereby revealing weaknesses of this technology.

HOW BROADCAST ENCRYPTION WORKS

There are at least four different schemes for broadcast encryption, and all share some common features. These schemes are based on a *key management block*: a block of data that is sent at the beginning of a broadcast in a broadcast application or is prerecorded on blank media during the manufacturing process. Each recipient reads the key

0018-9162/02/$17.00 © 2002 IEEE

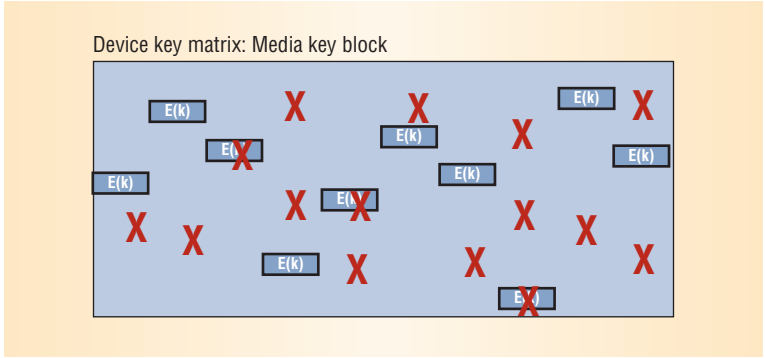

Figure 1. Content Protection for Recordable Media key matrix. CPRM uses device keys, shown as rectangles. The Xs show compromised keys: If an innocent device finds that one of its keys has been crossed out, it simply moves on to another column.

management block and processes it to yield a *management key*. Each device processes this key management block in a slightly different way, but they all get the same final answer. On the other hand, when circumvention devices attempt to process the key management block in the same way, they end up with the wrong answer.

At this point, the different broadcast encryption schemes diverge. Content protection for recordable media, the simplest and easiest to explain, associates the key management block with a matrix of device keys. The CPRM matrix is always much taller than it is wide. For example, a recordable DVD's matrix measures 16 columns wide by roughly 2,500 rows tall, and it is pre-embossed onto the disk's lead-in area. The key management block is the repeated encryption of the management key using each different device key. A consumer electronics device that knows a device key's position in the matrix can decrypt the value found at that position. The result will be the management key.

Each device actually has 16 device keys, one for each column in the matrix. Figure 1 shows these keys as rectangles in the matrix. Every device in the world has a different set of device keys. If we pick two devices at random, they might have one or two keys in common, but no two devices will have all 16 keys in common. To process the key management block, a device uses one of its keys to decrypt the corresponding position in the matrix.

Which device key? At first, it doesn't matter—any column will do. But once the system comes under attack, circumvention devices will appear that are using some of the keys in the matrix. Figure 1 shows compromised keys as Xs. If an innocent device finds that one of its keys has been crossed out, it simply moves on to another column. Eventually, it will find a key that has not been compromised. In severe cases, the system may be so compromised that most of the matrix has Xs. In that case, the system would exclude some innocent devices, and we would probably say that the system has been broken.

Thus, the CPRM scheme has a finite, albeit large, capability to withstand attack, as was also true of the original Fiat-Naor scheme. In 1998, however, two independent groups—one led by Debby Wallner[3] and another led by Chung Kei Wong[4]—found a scheme that had unlimited revocation. The Wallner-Wong scheme, which uses trees of keys instead of a matrix, is called the *logical key hierarchy*. Unfortunately, because the LKH key management block for a given amount of revocation capability was about the same size as a matrix-based key management block, LKH did not offer any improvement for media.

In 2000, however, Dalit Naor and colleagues[5] developed an LKH approach that substantially reduced the size of the key management block to roughly the size of a public-key certificate revocation list. That improvement eliminated a remaining advantage of public key cryptography.

BROADCAST ENCRYPTION VERSUS PUBLIC KEY

Broadcast encryption is fast. All its calculations are done using simple symmetric encryptions. In contrast, actual public key calculations require exponentiation operations over a finite field. The processor load to calculate a management key in a broadcast encryption scheme literally requires less than 1,000 times the load required to perform a public key signature calculation.

Certificate revocation lists are a vital if often overlooked aspect of public key systems. Without a means to revoke compromised individual keys, a public key system degenerates into a global secret scheme: The first break defeats the entire system. If a proposed public key system contains a flaw—and, sadly, many do these days—it is almost axiomatic that the revocation information fails to travel through the system.

Granted, broadcast encryption schemes are as sensitive as public key schemes to proper revocation. However, in a broadcast encryption scheme, it is difficult to forget about the problem: The revocation information is implicit in the key management block.

Classic security usually addresses the "us versus them" problem by trying to prevent eavesdropping on the communication between two boxes that are known to be secure. Recently, however, copyright protection has become an important new cryptography application that complicates the problem because it gives keys to everyone. Both attackers and legitimate customers receive keys because there is no way to distinguish between the two. Thus, how sensitive a box is to reverse engineering becomes an important issue.

As Figure 2 shows, a public key system requires a cryptographic handshake at the link level, which tends to cause the secret keys to bubble to the surface, near the link-level code. This exposes two

weaknesses: First, the secrets might be easier to find; second, it might be easier to find other interfaces—shown as question marks—where the content is unprotected.

When Beale Screamer broke the Windows Media Player (http://cryptome.org/ms-drm.htm), this self-named hacker ridiculed the classic public-key-based module-to-module interface that Microsoft designed into the product. Beale Screamer said that while the interface was expensive in performance and code size, it provided no obstacle to his attack. In contrast, a broadcast encryption scheme can hide the secrets—the device keys—much deeper in the software, near the point of content consumption.

One limitation of broadcast encryption is that it can never provide a nonrefutable digital signature. The nonrefutable property makes it difficult for someone to deny having produced a given digital signature. Public key cryptography offers an advantage because a forger faces an intractable mathematical problem when attempting to generate a valid signature without access to the actual signer's private key. On the other hand, anybody can verify a digital signature because doing so doesn't require access to secret information.

A broadcast encryption system replaces the digital signature with a weaker concept: a *message authentication code*. To verify a MAC, devices must know a shared secret—in this case, the management key. Every device that can process the key management block knows the management key, not just the device that generated the MAC.

Another advantage of public key cryptography is that the system design does not require a central authority. This is not possible in broadcast encryption, which requires a licensing agency to produce the key management blocks and assign the device keys. Many public key systems do use a central authority, called a certificate authority, but some do not. For example, the Pretty Good Privacy system[6] uses a completely democratic approach. Users create their own certificates, which move throughout the system as users exchange them. To establish trust, users vouch for other users' certificates.

Given the tradeoffs between the two approaches, broadcast encryption is not the optimal choice for all applications. Content protection, however, is well suited to this technology. All of broadcast encryption's advantages are important in content protection, which is primarily oriented toward consumer electronics devices, where features such as low overhead, an implicit revocation list, and strong resistance to reverse engineering are essential considerations.

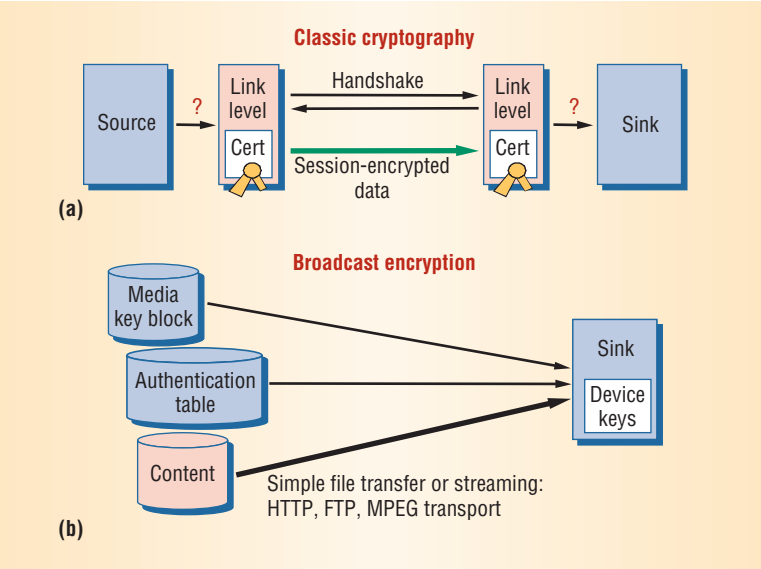

Figure 2. Classic cryptography versus broadcast encryption. (a) Classic security uses a cryptographic handshake at the link level to prevent eavesdropping on the communication between two secure boxes. (b) Broadcast encryption hides device keys deeper in the software, near the point of content consumption.

On the other hand, broadcast encryption's disadvantages, such as nonrefutable signatures, are irrelevant in content protection. A device cares only about connecting to another compliant device, not exactly which device it connects to. Calculating a correct MAC proves to another device that the original device is compliant. After all, if a device misbehaves, it will be excluded in future key management blocks. Finally, the need for a central licensing authority is a plus in content protection systems, which fundamentally focus on licensing, not technology.

CONTENT PROTECTION IN A HOME NETWORK

Most observers agree that home networking is inevitable: Soon, a family's personal computer, television, stereo, DVD player, and similar devices will all be connected to each other. These devices could also be seamlessly connected to the Internet, offering the advantage of an added content source. Internet access also provides this application's greatest threat, however, at least from the entertainment companies' viewpoint.

If allowed to run unchecked, a single purchased movie could be replayed on literally millions of televisions worldwide, generating no income for the title's owners beyond the initial sale. All three of the technologies we describe give consumers complete freedom to enjoy purchased content within the home, but they raise obstacles to the wide redistribution of protected content outside of it.

By its very name, broadcast encryption implies a one-to-many technology that's poorly suited to peer-to-peer applications. Yet broadcast encryption can work surprisingly well in a P2P situation, as long as MACs suffice in place of digital signatures. To support this assertion with examples, we consider three different approaches to the problem of protecting entertainment content within a home

Broadcast encryption technologies raise obstacles to the redistribution of protected content outside the home network.

network. The first two use conventional public key cryptography. The third, the xCP cluster protocol, uses broadcast encryption.

DTCP: Security for the digital bus

Digital Transmission Content Protection (http://www.dtcp.com/), the most developed of the three technologies, has been around for several years and can be found in several types of consumer devices. The DTCP protocol provides authentication and data encryption for devices connected with a digital bus—primarily the IEEE 1394 Firewire bus. The 5C companies—Hitachi, Intel, Matsushita, Sony, and Toshiba—developed DTCP, which allows the digital connection of consumer electronic devices such as DVD players, digital televisions, or set-top boxes.

DTCP provides a straightforward application of state-of-the-art public key cryptography to the problem of content protection on a digital bus. It defines two handshake protocols that let the participating devices authenticate each other and establish a shared session key. Both of these authentication and key exchange protocols are based on elliptic curve certificates that follow the new Digital Signature Algorithm standard and a Diffie-Hellman key exchange.[7]

The full authentication process requires both devices to exchange and check the signature on each other's certificates. Next, the Diffie-Hellman key exchange results in the calculation of a session key. Protected content will not flow between two devices in the home network unless both are bona fide, unrevoked, licensed entities that can be counted on to obey the content protection rules. Depending on the type of content, the devices might agree on the restricted authentication protocol. This process allows devices with limited resources to participate in the protocol and establish a secure channel using a shared-secret algorithm.

Unlike other public key approaches, DTCP has a well-thought-out revocation scheme that uses system renewability messages to identify revoked devices by listing their certificate numbers. The licensing agency distributes these messages in new devices, in broadcast channels, or on prerecorded media.

A link-level technology, DTCP has potential reverse-engineering exposure. Although DTCP offers the restricted authentication protocol to address the overhead of public key cryptography, this is little better than a shared-secret scheme. The well-reported DTCP break was not a break at all, just an exposé that revealed the restricted authentication's relative weakness.

DTCP does not completely address the problem of redistributing protected content in a home network application. It offers no technology to protect the content once it leaves the link and resides on hard disks within the network. The transient DTCP session key lives only during the conversation between two devices. To be fair to DTCP's designers, they never intended for it to protect media. They envisioned that some other technology—perhaps the broadcast-encryption-based CPRM—would actually protect the content on media.

OCCAM: End-to-end content security

Open Conditional Content Access Management[8] offers a more general approach to content security. An open standard proposed by Cisco Systems, OCCAM protects content during transmission over public networks, in home networks, and while stored on removable or fixed media.

Cisco bases OCCAM on a hierarchical public key infrastructure with a central authority, the OCCAM Certification and Revocation Authority. OCRA assigns device IDs and private keys to device manufacturers and maintains a database of the corresponding public key certificates, which it distributes periodically to content owners along with the revocation information.

To access a piece of content, the OCCAM-certified device must request a ticket from the content owner. Among other things, the ticket contains an encryption of the content key using the device's public key. Only that device can use the ticket to decrypt and access the content. Doing so requires connecting the device to the ticket-issuing agency, at least at the beginning of the transaction. This connected model lets the content provider check the device's revocation status and other restrictions, such as regional coding or time-based usage constraints, before returning an authorization ticket to the device.

An OCCAM-certified device provides only limited support for operating in a disconnected environment. Although the standard includes the predownload of authorization tickets for later playback, this feature essentially consists of a connected system that requires devices to go online to retrieve an authorization ticket for each piece of content.

OCCAM makes no apologies for being a pure public key system, thus it offers no shortcuts for reducing device overhead. Given the connectivity requirement, it has excellent revocation properties. The content remains encrypted with the same key

throughout the system until the moment of rendering, solving the problem of protecting the content on network storage. For the same reason, OCCAM offers the same degree of reverse-engineering protection as a broadcast encryption scheme.

The concerns about OCCAM focus on the practicality of the high connectivity it requires. Every piece of content the user acquires needs a separate ticket for each device that accesses it. Although some analysts have suggested giving consumers a single ticket for multiple content pieces, such as "HBO programs for the month of September," these proposals have not been convincing. The key weakness of these simplifications is that the ticket becomes an important shared secret, with all the weaknesses that implies.

Consumer privacy concerns, however, may be OCCAM's most troublesome aspect. Even if manufacturers make a conscious effort to disassociate device certificates from things like device serial numbers—which would revert to warranty cards—the home network application will still need to identify the cluster of devices in a single home. A list of device manufacturers and model numbers would probably be sufficient to identify the individual. Adding to that a list of purchased content items—which the ticketing agency would maintain—would uniquely identify the user.

xCP cluster protocol

IBM developed the xCP cluster protocol—a broadcast encryption-based technology—as part of its participation in the 4C group.

Figure 3 shows an xCP cluster in which the devices have agreed upon a common key management block. All the protected content is encrypted with keys based on the management key. The compliant devices only process content associated with the cluster they belong to. A device joins a cluster and proves its compliance by verifying the MAC for the join request based on the cluster's management key. The existing devices in the cluster allow the new device to join because they know that a circumvention device could not calculate the correct MAC.

A device doesn't need to converse with other devices to calculate the content keys. As long as it can find the key management block—which is a simple file in the network, a duplicate of which might even be in the device's local persistent storage—the device can decrypt any piece of content in the network. The usage rules may forbid the device from doing certain things with the cryptographically protected content. A compliant device will not perform a forbidden action. For example, a recorder would not record content marked *do not copy*.

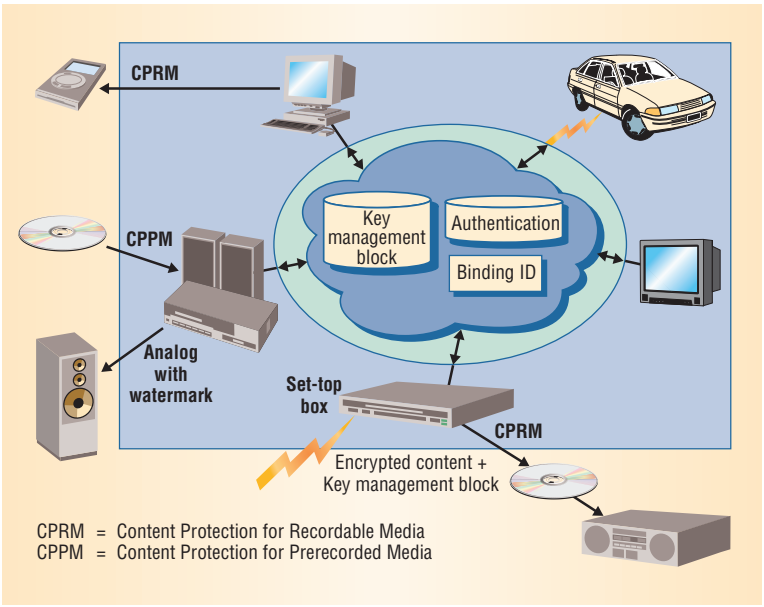

Figure 3. xCP cluster. All the connected devices have agreed upon a common key management block. All protected content is encrypted with keys based on this management key so that compliant devices only process content associated with their cluster.

Intuitively, the DTCP approach sounds stronger than xCP, but in cryptographic terms, the two approaches are equivalent. In DTCP, a source device like a DVD player would not send content marked *do not copy* to a device that a device certificate identifies as a recorder. In xCP, the recorder can access and decrypt the content, but it voluntarily chooses not to record it. A DTCP circumvention device will never identify itself as a recorder, even if it is one. Thus, the revocation list is the only guarantee that a source is not sending content to a recorder. Likewise, when an xCP source encrypts content marked *do not copy* with the management key, it has the same guarantee: No compliant recorder will record that content. If a recording device has keys that do not play by the rules, the keys will be revoked.

Compared to the roar of cryptographic conversations that fly about the network when DTCP establishes session keys or when OCCAM issues tickets, xCP clusters seem almost silent. In xCP, a cryptographic conversation occurs only when a device joins the cluster. As must be expected when comparing xCP to the two public-key-based systems, this broadcast encryption technology excels as the low-overhead approach. More surprisingly, xCP seems to have two functional advantages as well. Compared to DTCP, xCP protects not just the link in the home, but also the content on the media itself. Compared to OCCAM, xCP offers an unconnected, high-privacy approach.

WHERE BROADCAST ENCRYPTION FAILS

Unfortunately, broadcast encryption falls short of being a panacea. In at least a couple of instances, broadcast encryption fails to convincingly address the full application.

A successful broadcast encryption system requires a way to avoid the big-score phenomenon.

Electronic funds transfer

In this application, broadcast encryption suffers because it lacks a nonrefutable signature. In public key cryptography, the signature not only provides validity, it uniquely identifies the signer. When a client transfers funds, both the sender and receiver want to be confident that they are communicating with the proper entity.

In broadcast encryption, the MAC only proves that it was "signed" by a member in good standing of the approved devices club. This capability is not totally useless, however—one club bylaw might be that members are not allowed to lie about their identity. Liars would be excluded from further participation in the system by revoking their keys in the system's key management blocks.

Although it is possible to use broadcast encryption for electronic fund transfers, it is impractical. By the time the miscreant can be identified, the damage has been done. An attacker can play by the rules while preparing for the big score. When ready, the intruder can use a false identify to execute a one-time attack and then never participate in the system again. Among other things, it is impossible to determine which device keys an intruder used unless the fraud is performed systematically and repeatedly. In that case, the only way to reveal the attacker's identity is by sending a series of "forensic" key management blocks.

Why don't these problems occur in the content protection application? Why couldn't a hit-and-run circumvention device break the content protection rules and remain undetected? Athough such a device could exist, it would be of no economic consequence because a low level of unauthorized copying will always occur, regardless of the technologies used to guard against it. If millions of such devices circulate, the rare successful intrusions would become obvious, and access for that class of devices could be revoked. Simply put, content protection schemes that do not rely on global secrets allow no opportunity for a single big score.

Thus, a successful broadcast encryption system requires a way to avoid the big-score phenomenon. This requirement suggests that a micropayment system based on broadcast encryption, which charges fractions of a cent for minor events like reading a Web page, is reasonable. The inherent anonymity of a broadcast-encryption-based system might actually be an advantage when simulating cash. At the very least, such a system would be better than one based on a shared secret.

Secure socket layer

Another common application of public key cryptography involves providing link-level security for communication channels. Secure socket layer technology offers a good example of this category. Many businesses use SSL to provide encryption, authentication, and integrity-verification services for a variety of application-level Internet protocols. The best-known implementation of this concept is the SSL-secured version of HTTP supported by all standard Web browsers as the HTTPS protocol.

While SSL supports two-way authentication, it is primarily used in a one-way mode that lets the browser use an X.509 certificate issued for the server's DNS name to authenticate the Web server. This lets the browser make sure it has connected to the site the user asked for, and it provides a way to establish a session key that protects the conversation.

For renewability, SSL relies on X.509 certificate revocation lists. Although most Web browsers properly process revocation information, there is no strict requirement for the presence of current certificate revocation data and no well-defined automated process for the browser to download certificate revocation lists from the issuing certificate authority. Renewability does not seem to be an issue for SSL in the server authentication mode.

In a system that provides server authentication and communication-channel security with broadcast encryption, the server applications need to obtain a unique set of device keys, and the client applications need to obtain and periodically update key management blocks. The client sends its key management block to the server when it connects, and the server proves its compliance by using the management key to encrypt a response. This response might contain the server's domain name and a proposed session key. Unlike an SSL client that can operate completely without a revocation list, the system requires the key management block initialization at least once to set up the client. After completing that step, the same considerations apply for updating key management blocks as for downloading certificate revocation list updates.

Clearly, this system is more lightweight than a conventional SSL-based server system—in terms of both the communication flow required during the initial handshake and the CPU requirements for clients and servers. For commerce servers, the SSL overhead usually dwarfs all other processing that the site does.

The problems with this setup resemble those that beset the electronic funds transfer application. A server would have a license obligation to use its set

of device keys for its site only, but there is no cryptographic enforcement of that obligation. Any server with a set of device keys could lie about its identity and masquerade as another site.

Arguably, on the Internet, an imposter would have to make repeated intrusions to get a big enough payback to justify the attack. However, repeated intrusions can be detected, and new key management blocks can be issued to exclude the imposter.

More insidious and more difficult to detect is a silent, man-in-the-middle attacker who has a set of device keys. Such an attacker can read all the traffic to the legitimate site and could, for example, collect credit card numbers as users send them to a merchandising site like Amazon.com.

At first glance, this vulnerability makes the use of broadcast encryption for communication-channel security seem inappropriate. For some applications, a reduced level of authentication is acceptable or even preferable—especially if they only issue device keys based on well-defined criteria, such as existing certification systems or customer-feedback ratings. We believe that this approach raises the barrier of entry for potential attackers considerably, and it would provide a meaningful level of authentication and confidentiality for consumers. For example, a customer ordering medication over the Internet is more interested in knowing that a registered pharmacy operates the server than in knowing the Web site's name.

Broadcast encryption provides just as much protection as public key cryptography. However, broadcast encryption's cryptographic guarantee differs fundamentally from the protection that public key cryptography offers. Broadcast encryption guarantees a participant that another participant belongs to the same group. Public key cryptography guarantees the other participant's actual identity. It is not clear that knowing the actual identity is better than knowing whether that individual belongs to a particular group. In both cases, the guarantees are only as strong as the encompassing system's functional revocation mechanism.

The increased worldwide interest in individual privacy will motivate developers to rethink many existing applications. In any event, developers currently underutilize broadcast encryption as a cryptographic tool, and we believe that this decade will likely see an increased emphasis on this method of content protection. ■

References

1. A. Fiat and M. Naor, "Broadcast Encryption," *Advances in Cryptology* (Crypto 93), Lecture Notes in Computer Science 773, Springer-Verlag, New York, 1994, pp. 480-491.
2. B. Traw, "Protecting Digital Content Within the Home," *Computer*, Oct. 2001, pp. 42-47.
3. D.M. Wallner, E.J. Harder, and R.C. Agee, "Key Management for Multicast: Issues and Architectures," RFC 2627 (informational), July 1999; ftp://ftp.isi.edu/in-notes/rfc2627.txt.
4. C.K. Wong, M. Gouda, and S. Lam, "Secure Group Communications Using Key Graphs," *Proc. SIGCOMM 1998*, ACM Press, New York, pp. 68-79.
5. D. Naor, M. Naor, and J. Lotspiech, "Revocation and Tracing Routines for Stateless Receivers," *Advances in Cryptology* (Crypto 2001), Lecture Notes in Computer Science 2139, Springer-Verlag, New York, 2001, pp. 41-62.
6. P. Zimmermann, *The Official PGP User's Guide*, MIT Press, Cambridge, Mass., 1995.
7. W. Diffie and M. Hellman, "New Directions in Cryptography," *IEEE Trans. Information Theory*, Nov. 1976, pp. 644-654.
8. J. Yoshida, "Content Protection Plan Targets Wireless Home Networks," *EE Times*, Jan. 2002; http://www.eetimes.com.

Jeffrey Lotspiech *is a research staff member at IBM Almaden. His research interests include content protection and software tamper resistance. Lotspiech received an MS in electrical engineering/computer science from the Massachusetts Institute of Technology. He is a member of the IEEE. Contact him at lotspiech@almaden.ibm.com.*

Stefan Nusser *is a senior software engineer in IBM's Software Group. His research interests include content protection and digital rights management. Nusser received a PhD in management information systems from Vienna University of Business Administration and Economics. Contact him at nusser@us.ibm.com.*

Florian Pestoni *is a software engineer at IBM Almaden. His research interests include digital content distribution. Pestoni received an MS in electrical engineering from the University of Buenos Aires and an MBA from the University of California, Berkeley. Contact him at fpestoni@almaden.ibm.com.*